

Nadine Adelina Ghindaoanu

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

Master's thesis in Cybernetics and Robotics

Supervisor: Tor Arne Johansen

Co-supervisor: Kristoffer Gryte

June 2023



Nadine Adelina Ghindaoanu

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

Master's thesis in Cybernetics and Robotics  
Supervisor: Tor Arne Johansen  
Co-supervisor: Kristoffer Gryte  
June 2023

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





# Abstract

As the applications of unmanned aerial vehicles (UAVs) increase, controllers that are able to operate under severe weather conditions are sought. In this thesis, nonlinear model predictive control (NMPC) is explored, taking into account a UAV operating in icing conditions, including asymmetric icing on the wings.

In previous work, the NMPC controller was used together with a disturbance observer to handle the effects of icing, considering it a disturbance the NMPC has no knowledge of. The focus of this thesis lies on increasing the robustness of the NMPC controller, making it more suitable for icing conditions. This was done by updating the aerodynamic model used in the NMPC formulation, as well as including the effects of icing and icing asymmetry in its model. The NMPC controller without icing in its model is compared to the NMPC controller with icing in a series of simulations, testing their performance with disturbances such as reduced airspeed and severe wind conditions.

The NMPC controllers are also compared to previously developed PID and MRAC controllers. The results show a similar performance of the PID, MRAC and NMPC without icing controllers, depending on the simulation run. The NMPC without icing is shown to be better suited to handle reduced airspeed, but shows a slower response in the other simulations, compared to the PID and MRAC. However, the results show a clear improvement in performance when icing and asymmetry are included in the model of the NMPC. This NMPC outperforms all controllers in all simulations when it comes to pitch and airspeed tracking, and has the second-best roll tracking, only surpassed by the PID.

The path-following abilities of the four controllers are also tested with severe wind conditions, where the NMPC with icing and the MRAC have the best performances. Knowing that the NMPC run in practice might not be able to detect the icing level or estimate the disturbance observer states to the same degree of precision, additional simulations are performed where this is taken into account. These scenarios were not found to compromise the performance of the NMPC controller.



# Sammendrag

*Norwegian translation of the abstract*

Etterhvert som bruken av små ubemannede luftfartøy (UAV) øker, etterlyses regulatorer som er i stand til å operere under krevende værforhold. I denne oppgaven blir NMPC (*nonlinear model predictive control*) utforsket, hvor det er tatt i betraktning at UAV-en kan operere under isingsforhold, og asymmetrisk ising på vingene er inkludert.

I tidligere arbeid ble NMPC-regulatoren brukt sammen med en *disturbance observer* for å håndtere effekten av ising, som var sett på som en forstyrrelse NMPC-en ikke har kjennskap til. Fokuset i denne oppgaven ligger på å øke robustheten til NMPC-regulatoren, og gjøre den mer egnet for isingsforhold. Dette ble gjort ved å oppdatere den aerodynamiske modellen brukt i NMPC-formuleringen, samt inkludere effektene av ising og isingsasymmetri i modellen sin. NMPC-regulatoren uten ising i modellen sammenlignes med NMPC-regulatoren med ising i en serie simuleringer, som tester ytelsen med forstyrrelser som redusert lufthastighet og harde vindforhold.

NMPC-regulatoren sammenlignes også med tidligere utviklede PID- og MRAC-regulatorer. Resultatene viser en lignende ytelse av PID, MRAC og NMPC uten ising-regulatoren, avhengig av hvilken simulering det er kjørt. NMPC uten ising er vist å være bedre egnet til å håndtere redusert lufthastighet, men viser en langsommere respons i de andre simuleringene, sammenlignet med PID og MRAC. Resultatene viser imidlertid en klar forbedring i ytelsen når ising og asymmetri er inkludert i modellen til NMPC-en. Denne NMPC-en viser en bedre evne til å følge pitch og lufthastighet enn de andre regulatorene i alle simuleringer, og er den nest beste til å følge rullreferansen, kun overgått av PID-en.

Banefølgingssevnen til de fire regulatorene er også testet med harde vindforhold, der NMPC-en med ising og MRAC-en viser den beste ytelsen. I tillegg, NMPC-en kjørt i praksis vil kanskje ikke være i stand til å oppdage isingsnivået eller estimere forstyrrelsesobservatørens tilstander med samme grad av presisjon. Derfor blir ytterligere simuleringer utført, der dette er tatt i betraktning. Disse scenariene ble ikke funnet å kompromittere ytelsen til NMPC-regulatoren.





# Preface

This thesis completes my Master's degree in Cybernetics and Robotics at the Norwegian University of Science and Technology (NTNU). The work presented in this thesis was carried out at the Department of Engineering Cybernetics (ITK) under the supervision of Tor Arne Johansen and Kristoffer Gryte. This thesis is a continuation of my pre-project [1], in which the control of a UAV that is subject to wind and icing conditions using nonlinear model predictive control (NMPC) was investigated. The background for both reports is the same, which is reflected in the introduction and the background theory. Some of the simulations are repeated as well for comparison. However, the focus of this thesis lies in the improvement and robustness of the NMPC controller.

I would like to thank my supervisors Tor Arne Johansen and Kristoffer Gryte for their help and guidance throughout the past year. Their ideas, insight on the icing problem and model predictive control have been very valuable and helped me progress in my work. I would also like to thank Nicholas Müller for his help and knowledge on propeller icing, helping me include it in the model.



# Contents

<b>Abstract</b> . . . . .	<b>iii</b>
<b>Sammendrag</b> . . . . .	<b>v</b>
<b>Preface</b> . . . . .	<b>vii</b>
<b>Contents</b> . . . . .	<b>ix</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Previous Work and Contributions . . . . .	2
1.3 Report Outline . . . . .	3
<b>2 Theory</b> . . . . .	<b>5</b>
2.1 Coordinate Frames . . . . .	5
2.2 The UAV Model . . . . .	7
2.3 The UAV Model in Stability-Wind Frame Representation . . . . .	8
2.4 Forces and Moments . . . . .	9
2.4.1 Aerodynamic forces and moments . . . . .	9
2.4.2 Gravitational forces . . . . .	10
2.4.3 Propulsion forces . . . . .	10
2.5 Actuation . . . . .	11
2.6 Icing Model . . . . .	11
2.6.1 Asymmetric Icing Model for Aerodynamic Forces and Mo- ments . . . . .	12
2.6.2 Propulsion Icing Model . . . . .	13
2.7 Wind model . . . . .	13
<b>3 Nonlinear Model Predictive Control (NMPC)</b> . . . . .	<b>15</b>
3.0.1 Reduced attitude parametrization . . . . .	16
3.0.2 Disturbance observer . . . . .	16
3.0.3 Nonlinear Program . . . . .	17
3.0.4 Constraints . . . . .	18
<b>4 Method</b> . . . . .	<b>21</b>
4.1 Reference Generation . . . . .	22
4.1.1 Roll and Pitch Reference Model . . . . .	22
4.1.2 Angular Velocity Reference Model . . . . .	22
4.1.3 Commanded Roll and Pitch from the Guidance Controller . . . . .	23
4.2 Implementation of the NMPC Controllers . . . . .	25
4.2.1 Tuning of the NMPC . . . . .	28

4.2.2	NMPC without icing: updated model . . . . .	34
4.2.3	NMPC with icing in its model . . . . .	34
4.2.4	Binary NMPC with icing in its model . . . . .	35
4.3	Implementation of the Physical Model . . . . .	36
4.3.1	Icing on the Wings . . . . .	36
4.3.2	Propeller Icing . . . . .	36
4.4	Simulations . . . . .	38
4.4.1	Performance Metric . . . . .	39
4.4.2	Baseline simulation . . . . .	39
4.4.3	Reduced airspeed and wind conditions . . . . .	40
4.4.4	Guidance simulations . . . . .	40
4.4.5	Additional simulations . . . . .	41
<b>5</b>	<b>Results . . . . .</b>	<b>43</b>
5.1	Baseline simulation . . . . .	43
5.2	Reduced airspeed simulation . . . . .	47
5.3	Wind disturbance simulation . . . . .	50
5.4	Additional simulations . . . . .	53
5.5	Guidance simulation . . . . .	59
<b>6</b>	<b>Discussion . . . . .</b>	<b>65</b>
6.1	The improved NMPC controllers . . . . .	65
6.2	Simulations and performance of the controllers . . . . .	66
6.3	Tuning of the controllers . . . . .	66
6.4	Stall angle . . . . .	67
<b>7</b>	<b>Conclusion . . . . .</b>	<b>69</b>
	<b>Bibliography . . . . .</b>	<b>71</b>
<b>A</b>	<b>The PID and MRAC Controllers . . . . .</b>	<b>75</b>
A.1	Roll, Pitch and Airspeed Control with PID and MRAC . . . . .	75
A.1.1	PID . . . . .	75
A.1.2	Model Reference Adaptive Control (MRAC) . . . . .	75
A.1.3	Airspeed Controller . . . . .	77
A.2	Implementation of the PID and MRAC Controllers . . . . .	77

# Chapter 1

## Introduction

### 1.1 Motivation

Unmanned Aerial Vehicles (UAVs), as the name suggests, are aircraft that do not operate with a pilot on board. They are smaller in size than manned aircraft, giving them the possibility to access remote or challenging locations, and their use across various applications is rapidly increasing as a result. However, because of the difference in size and weight compared to manned aircraft, they are also more severely affected by weather conditions. UAVs are an emerging technology, so a way to combat these challenges has not been researched to the same extent. This thesis focuses on one of these challenges in particular: in-flight icing [2]. In-flight icing has been shown to lead to a degradation of the aerodynamic performance of the aircraft, including a significant decrease in lift, an increase in drag, and a deterioration of the stall limits [3], [4]. The severity of the performance degradation depends on the type of ice being formed, where complex ice geometries, such as horn-like shapes, as shown to increase it [4]. In the case of a UAV in icing conditions, the ice accretion will make up a more significant part of its total weight, so its performance would be affected more severely than in the case of a manned aircraft. In addition, the ice conditions cannot be identified by a pilot in this case, so the controllers used must be able to handle the effects of icing and ensure the safety of the UAV.

Due to the fast dynamics of a UAV, the controllers typically used are inner-loop PID controllers, where the model is linearized around trim states. Because nonlinearities are neglected, the model is less accurate the further away from the trim states. Nonlinear Model Predictive Control (NMPC), on the other hand, would allow for the nonlinear effects to be taken into account, but due to large computational time, they have typically been too slow to be used. Progress has been made on the topic, where the NMPC controllers developed by Reinhardt [5], for example, were shown to have equal or superior performance when tested together with the ArduPilot [6] controller, a widely used open-source drone software. In the author's pre-master project [1], the NMPC controller was shown to also work with the icing problem and have similar performance as previously developed PID and

MRAC controllers. This thesis explores how the NMPC controller can be modified to increase the controller's robustness and performance when subject to icing and other disturbances.

## 1.2 Previous Work and Contributions

The Nonlinear Model Predictive Controller used as a starting point in this thesis is based on the research performed by Reinhardt in his doctoral thesis [5]. His controllers were developed in Python using the package Acados [7], and showed a great performance, but they were not tested in icing conditions. This was done in the pre-master project [1], and compared to the performances of previously developed PID and Model Reference Adaptive Control (MRAC) controllers. The PID controller, based on Beard & McLain [8], and the MRAC, based on Lavretsky and Wise [9], were tuned for icing conditions and implemented in a Matlab/ Simulink simulator by Högnadóttir [10]. The simulator was first developed by Gryte [11], and later improved with the UAV's aerodynamic data in icing conditions, found by Winter [12], and with Kleiven's extended model to account for asymmetric icing on the wings. [13]. In the pre-project thesis, the simulator was also extended to include the effects of icing on the propeller, which is shown to result in a significant decrease in thrust and increase in torque [14],[15]. This was done following Müller's work [16]. The propulsion model was also further improved by implementing the model proposed by Coates [17], which was shown to predict thrust better than the Fitzpatrick model [18],[19] previously used, with a root mean square error of 2.20 – 4.52 percent, compared to Fitzpatrick's 6.56 percent [17].

This thesis continues the work from the project thesis [1], in which the NMPC developed by Reinhardt was implemented in the Matlab/ Simulink simulator and compared to the PID and MRAC controllers. However, its aim is to improve the NMPC controller, focusing on the icing problem, and making it more robust against the effects of icing and icing asymmetry. The previously developed NMPC is first improved by updating its model with the "clean" aerodynamic coefficients from Winter's work [12], matching the model used in the simulator. This NMPC controller relies on a disturbance observer to counteract the effects of icing, as well as the effects of other disturbances. In this case, the NMPC has no knowledge of icing. In this thesis, a new NMPC is explored, in which the icing model and asymmetry model implemented in the simulator are also implemented in the model of the NMPC. Its performance is compared to the one of the NMPC without icing in its model, and to the previously mentioned PID and MRAC controllers.

### 1.3 Report Outline

This thesis is organized in several parts. In the first part, Chapter 2, the theory used throughout the thesis will be introduced, including the mathematical model of the UAV with its forces and moments, as well as the icing and wind models. The NMPC controller is introduced in Chapter 3. In Chapter 4, the simulator is explained with the implementation of its different parts (reference generation, controllers, physical model). The tuning and implementation of the NMPC controllers is also explained here, as well as the different simulation cases. Finally, the simulation results will be presented and discussed in chapters 5 and 6.

In this thesis, the NMPC controller will be compared to a proportional-integral-derivative (PID) controller, and to a model reference adaptive controller (MRAC) controller, based on Gryte [11] and Högnadóttir [10]. As the purpose is to compare the NMPC to existing, well-tuned controllers, their background theory and implementation are outside the scope of this thesis, but they were described in the author's project thesis, and given in Appendix A.





## Chapter 2

# Theory

As this master thesis is a continuation of the author's project thesis [1], the theory necessary to understand the UAV system and the forces, moments and disturbances acting on it, is the same as described in the project thesis. Therefore, this chapter's content is repeated from [1] and slightly updated where more information was needed. The mathematical models explained here are the background for the simulations, and some will also be used in formulating 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  $\mathbf{v}$  described in reference frame  $\{a\}$  will be written as  $\mathbf{v}^a$ , and a rotation transforming vector  $\mathbf{v}^a$  to vector  $\mathbf{v}^b$  (described in reference frame  $\{b\}$ ), will be given by the rotation matrix  $\mathbf{R}_{ba}$ , so that  $\mathbf{v}^b = \mathbf{R}_{ba}\mathbf{v}^a$ . Similarly,  $\mathbf{v}^a = \mathbf{R}_{ab}\mathbf{v}^b = \mathbf{R}_{ba}^\top\mathbf{v}^b$  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  $\mathbf{x}^n$  axis pointing to the North, the  $\mathbf{z}^n$  axis pointing to the direction of gravity, and  $\mathbf{y}^n = \mathbf{z}^n \times \mathbf{x}^n$ . 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  $\mathbf{x}^b$  axis points forward through the nose of the UAV, the  $\mathbf{z}^b$  axis points downwards, and  $\mathbf{y}^b = \mathbf{z}^b \times \mathbf{x}^b$ .

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  $\alpha$ , and the sideslip angle  $\beta$ , given by:

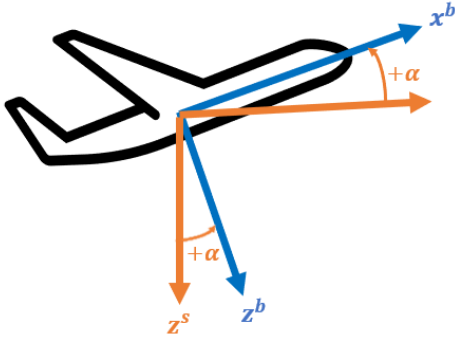
$$V_a = \|\mathbf{v}_r^b\| = \sqrt{u_r^2 + v_r^2 + w_r^2}, \quad (2.1a)$$

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

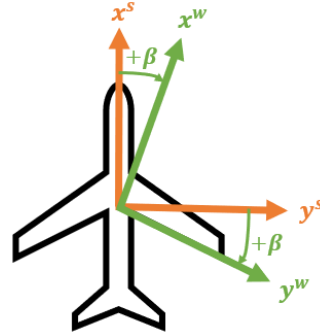
$$\beta = \arcsin\left(\frac{v_r}{V_a}\right), \quad (2.1c)$$

where the vector  $\mathbf{v}_r^b = [u_r, v_r, w_r]^\top$  is the relative velocity vector, given by  $\mathbf{v}_r^b = \mathbf{v}_{nb}^b - \mathbf{R}_{nb}^\top \mathbf{v}_{nw}^n$ ,  $\mathbf{v}_{nb}^b$  is the velocity vector of the UAV decomposed in body frame, and  $\mathbf{v}_{nw}^n$  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  $\mathbf{y}^b = \mathbf{y}^s$ , with rotation angle equal to the angle of attack  $\alpha$ , 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  $\mathbf{z}^s = \mathbf{z}^w$ , with rotation angle equal to the sideslip angle  $\beta$ , 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  $\mathbf{R}_{sb}(\alpha)$ :

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

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

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

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

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

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

## 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 [8] and are given by:

$$\dot{\mathbf{p}}_{nb}^n = \mathbf{R}(\Theta_{nb})\mathbf{v}_{nb}^b \quad (2.5a)$$

$$\dot{\Theta}_{nb} = \mathbf{T}(\Theta_{nb})\boldsymbol{\omega}_{nb}^b \quad (2.5b)$$

$$\dot{\mathbf{v}}_{nb}^b = \frac{1}{m}(\mathbf{F}_a^b + \mathbf{F}_g^b + \mathbf{F}_t^b) - \boldsymbol{\omega}_{nb}^b \times \mathbf{v}_{nb}^b \quad (2.5c)$$

$$\mathbf{J}^b \dot{\boldsymbol{\omega}}_{nb}^b = \mathbf{S}(\mathbf{J}^b \boldsymbol{\omega}_{nb}^b)\boldsymbol{\omega}_{nb}^b + \mathbf{M}^b. \quad (2.5d)$$

The kinematics describe the relationship between position and velocity, and are given by (2.5a) and (2.5b). The vector  $\mathbf{p}_{nb}^n = [p_n, p_e, p_d]^\top$  describes the inertial position of the UAV in the NED coordinate frame  $\{n\}$ , and  $\mathbf{v}_{nb}^b = [u, v, w]^\top$  is the velocity vector in  $\{b\}$ , where  $u$ ,  $v$  and  $w$  are the forward, lateral and vertical velocities, respectively. The vector  $\Theta_{nb} = [\phi, \theta, \psi]^\top$  contains the Euler angles, where  $\phi$ ,  $\theta$ , and  $\psi$  are the roll, pitch and yaw angles, respectively, and are given in  $\{n\}$ . The rotational motion can be described by the angular velocity vector  $\boldsymbol{\omega}_{nb}^b = [p, q, r]^\top$ , 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  $\mathbf{F}^b = \mathbf{F}_a^b + \mathbf{F}_g^b + \mathbf{F}_t^b$ , and  $\mathbf{M}^b$  is the total moment, given in  $\{b\}$ . They will be further described in section 2.4.1.

From (2.5) we also have  $\mathbf{J}^b = (\mathbf{J}^b)^\top$ , which denotes the inertia matrix and is given by

$$\mathbf{J}^b = \begin{bmatrix} J_{xx} & 0 & -J_{xz} \\ 0 & J_{yy} & 0 \\ -J_{xz} & 0 & J_{zz} \end{bmatrix}, \quad (2.6)$$

with the parameters in the inertia matrix given in Table 4.9. The matrices  $\mathbf{R}(\Theta_{nb})$

and  $T(\Theta_{nb})$  are given by

$$\mathbf{R}(\Theta_{nb}) = \begin{bmatrix} c_\theta c_\psi & s_\phi s_\theta c_\psi - c_\phi s_\psi & c_\phi s_\theta c_\psi + s_\phi s_\psi \\ c_\theta s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi \\ -s_\theta & s_\phi c_\theta & c_\phi c_\theta \end{bmatrix}, \quad (2.7)$$

$$\mathbf{T}(\Theta_{nb}) = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix}. \quad (2.8)$$

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 [5], 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  $\alpha$  and sideslip angle  $\beta$  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 this model,  $\delta_r$  is not included, as the UAV studied in this thesis does not have a rudder.

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  $\{n\}$ ,  $\mathbf{R}_{nb}$ , instead of the Euler angles that were used in (2.5), and is equivalent with  $\mathbf{R}_{nb} = \mathbf{R}(\Theta_{nb})$  given by (2.7). The matrix  $\mathbf{R}_{nb}$  can be written as  $\mathbf{R}_{nb} = [\mathbf{r}_x, \mathbf{r}_y, \mathbf{r}_z]$ , where  $\mathbf{r}_{\{x,y,z\}}$  are the axis of the body-fixed frame expressed in the coordinates of the inertial frame. Then, we define the state vector  $\mathbf{x} \in \mathbb{R}^{n_x}$  and the input  $\mathbf{u} \in \mathbb{R}^{n_u}$  as:

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

$$\mathbf{u} = [\dot{\delta}_a \ \dot{\delta}_e \ \dot{\delta}_t]^\top, \quad (2.10)$$

with  $n_x = 19$  and  $n_u = 3$ . The states  $V_a$ ,  $\alpha$  and  $\beta$  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  $\boldsymbol{\omega}_{nb}^s = \mathbf{R}_{sb} \boldsymbol{\omega}_{nb}^b$ , with  $\mathbf{R}_{sb}$  given by (2.2). The dynamic and kinematic equations for the state variables are given by:

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

$$\dot{\mathbf{R}}_{nb} = [\dot{r}_x \ \dot{r}_y \ \dot{r}_z] = \mathbf{R}_{nb} \mathbf{S} (\mathbf{R}_{sb}^\top \boldsymbol{\omega}_{nb}^s) \quad (2.11b)$$

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

where we notice that  $\boldsymbol{\omega}_{nb}^w$  is the angular velocity vector decomposed in the wind frame  $\{w\}$  and given by  $\boldsymbol{\omega}_{nb}^w = \mathbf{R}_{wb} \boldsymbol{\omega}_{nb}^b$ , with  $\mathbf{R}_{wb}$  given by (2.4). The matrix  $\mathbf{J}^s$  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  $\mathbf{v}_r^w$  is the relative velocity vector decomposed in  $\{w\}$ . They are given by

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

with  $\mathbf{J}^b$  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:

$$\mathbf{F}^b = \mathbf{F}_g^b + \mathbf{F}_a^b + \mathbf{F}_t^b, \quad (2.13)$$

$$\mathbf{M}^b = \mathbf{M}_a^b + \mathbf{M}_p^b, \quad (2.14)$$

where the total force in body-fixed reference frame  $\mathbf{F}^b$  is given by the sum of the gravity force  $\mathbf{F}_g^b$ , the aerodynamic force  $\mathbf{F}_a^b$  and the propulsion force  $\mathbf{F}_p^b$ . The total moment in body-fixed frame is equal to the sum of the aerodynamic moment  $\mathbf{M}_a^b$  and the propulsion moment  $\mathbf{M}_p^b$ . 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  $\mathbf{M}_p^b \approx 0$ .

### 2.4.1 Aerodynamic forces and moments

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

$$\mathbf{F}_a^b = \mathbf{R}_{wb}^\top \mathbf{F}_a^w = \mathbf{R}_{wb}^\top \begin{bmatrix} -F_{drag} \\ F_{side} \\ -F_{lift} \end{bmatrix}, \quad (2.15)$$

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

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

where  $\rho$  is the air density and  $S$  is the wing area, given in Table 4.9. Based on Beard & McLain [8] and Winter [12], the aerodynamic coefficients are given by:

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

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

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

where  $b$  is the wingspan and  $c$  is the mean chord, given in Table 4.9. The aerodynamic moment vector  $\mathbf{M}_a^b$ , modelled in the body-fixed frame, is given by:

$$\mathbf{M}_a^b = \begin{bmatrix} l \\ m \\ n \end{bmatrix} = \frac{1}{2} \rho V_a^2 S \begin{bmatrix} b C_l(\beta, p, r, \delta_a, \delta_r) \\ c C_m(\alpha, q, \delta_e) \\ b C_n(\beta, p, r, \delta_a, \delta_r) \end{bmatrix}, \quad (2.20)$$

where the aerodynamic coefficients are given by:

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

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

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

As the Skywalker X8 does not have a rudder, it is to be noted that  $\delta_r = 0$  in this case.

#### 2.4.2 Gravitational forces

The gravitational force acts on the  $\mathbf{z}^n$  axis direction, and following Newton's second law it is equal to  $\mathbf{g}_z^n = m\mathbf{g}$ , 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:

$$\mathbf{F}_g^b = \mathbf{R}_{nb}^\top m \mathbf{g}^n = \mathbf{R}_{nb}^\top \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}, \quad (2.24)$$

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

#### 2.4.3 Propulsion forces

The propeller model used is based on Coates [17], and describes the propeller thrust  $T$  as a function of the thrust coefficient  $C_T$ :

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

where  $\rho$  is the air density,  $\omega$  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  $J$ :

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

$$J = \frac{2\pi V_a}{\omega D}, \quad (2.27)$$

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:

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

The propeller speed  $\omega$  is proportional to throttle  $\delta_t \in [0,1]$ , where  $\delta_t = 1$  is equivalent to  $\omega = \omega_{max}$ . Therefore, Eq. (2.28) can be rewritten to include  $\delta_t$  in the expression:

$$\mathbf{F}_{prop}^b = \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{\rho D^4}{2\pi^2} C_T(J) (\delta_t \omega_{max})^2 \\ 0 \\ 0 \end{bmatrix}. \quad (2.29)$$

## 2.5 Actuation

The actuators of a UAV typically include a throttle,  $\delta_t$ , together with the control surfaces aileron, elevator and rudder, represented by  $\delta_a$ ,  $\delta_e$ ,  $\delta_r$ . These variables have been recurrent in various equations throughout Section 2.4.1. Deflecting the control surfaces can change the aerodynamic forces and moments of the UAV, following equations (2.17)-(2.23), which leads to attitude control. Although there is coupling between the states,  $\delta_a$  is mainly in charge of controlling roll, while  $\delta_e$  and  $\delta_r$  mainly control pitch and yaw, respectively. The throttle, on the other hand, is responsible for altering the propulsion forces.

The Skywalker X8 is a type of UAV called a flying wing, which does not have a rudder and 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,  $\delta_{el}$  and  $\delta_{er}$ . This relationship is given by:

$$\begin{bmatrix} \delta_{er} \\ \delta_{el} \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_a \end{bmatrix}. \quad (2.30)$$

Because of this,  $\delta_a$  and  $\delta_e$  are still used when calculating the forces and moments, but they are mapped to the pair  $\delta_{er}$  and  $\delta_{el}$  when applying the actuation to the physical system.

## 2.6 Icing Model

Ice accretion on the UAV affects the aerodynamic forces and moments acting on the UAV, as well as the propulsion forces. Using Kleiven's work [13], a linear interpolation can be used to describe how the aerodynamic coefficients change based on the icing level on the wings:

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

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. The coefficients  $C_{k,iced}$  and  $C_{k,clean}$  are found in the work of Winter [12], and more details are given in Section 4.3.1. In addition, as ice can fall from one wing while the other remains iced, icing can have an asymmetric effect on the aerodynamic forces and moments of the UAV. Therefore, the icing model used in this thesis includes an asymmetric aerodynamic icing model, explained in the next section, as well as a model that describes the effects of icing on the propulsion forces, introduced in Subsection 2.6.2.

### 2.6.1 Asymmetric Icing Model for Aerodynamic Forces and Moments

The asymmetric icing model for the wings was developed in Kleiven's thesis [13] by considering asymmetry in the aerodynamic forces and moments acting on the aircraft. In the model, the UAV is divided into a left and right side, so that the aerodynamic forces acting on it are given by

$$\mathbf{F}_k = \mathbf{F}_{k,r} + \mathbf{F}_{k,l}, \quad \text{for } k = \text{drag, lift, side}, \quad (2.31)$$

where  $\mathbf{F}_{k,r}$  and  $\mathbf{F}_{k,l}$  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

$$\mathbf{M}_{a,asym} = \mathbf{M}_{a,0} + \sum_k (\mathbf{r}_k \times \mathbf{F}_{k,r} + \mathbf{l}_k \times \mathbf{F}_{k,l}), \quad \text{for } \mathbf{F}_{k,r}, \mathbf{F}_{k,l} \notin \mathbf{M}_{a,0}, \quad (2.32)$$

where  $\mathbf{M}_{a,0}$  is the symmetric aerodynamic moment, given in (2.20), and the second term describes the asymmetry. The vectors  $\mathbf{r}_k$  and  $\mathbf{l}_k$  represent the distance from the center 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 [13].

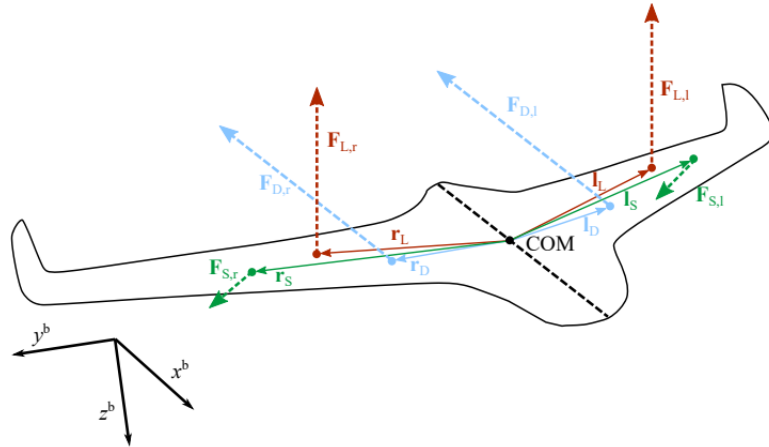


Figure 2.3: Asymmetric forces, from Kleiven [13].



### 2.6.2 Propulsion Icing Model

Based on the work of Müller [16], a model for the thrust coefficient in icing conditions can be given by:

$$C_{T,iced} = C_T(J)(1 + \min(TWC, TWC_{max})\Delta C_T(T)), \quad (2.33)$$

$$\Delta C_T(T) = \Delta C_{T,0} + \Delta C_{T,1}T + \Delta C_{T,2}T^2 \quad (2.34)$$

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:

$$TWC = tLWC\omega\frac{D}{2}, \quad (2.35)$$

$$TWC_{max} = \frac{A_{max}}{\frac{D}{2}\omega^2}, \quad (2.36)$$

$$A_{max} = A_{max,0} + A_{max,1}T^2. \quad (2.37)$$

Eq. (2.37) 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. (2.36). 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 [16]:

$$P = C_p\rho n^3 D^5, \quad n = \frac{\omega}{2\pi} \quad (2.38a)$$

$$C_{p,iced} = C_p(J)(1 + \min(TWC, TWC_{max})\Delta C_p(T)), \quad (2.38b)$$

$$C_p(T) = \Delta C_{p,0} + \Delta C_{p,1}T + \Delta C_{p,2}T^2, \quad (2.38c)$$

$$\Delta C_p(T) = \Delta C_{p,0} + \Delta C_{p,1}T + \Delta C_{p,2}T^2. \quad (2.38d)$$

## 2.7 Wind model

To simulate wind disturbances, the wind gusts are usually modeled by the von Karmen model, given in Beard and McLain [8]. In this thesis, similarly to Högnadóttir and Kleiven's previous work [10], [13], 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 [20] as:

$$\begin{aligned}
 H_u(s) &= \sigma_u \sqrt{\frac{2L_u}{\pi V_a} \frac{1}{1 + \frac{L_u}{V_a}s}}, & H_p(s) &= \sigma_w \sqrt{\frac{0.8}{V_a} \frac{(\frac{\pi}{4b})^{\frac{1}{6}}}{L_w^{\frac{1}{3}}(1 + \frac{4b}{\pi V_a}s)}}, \\
 H_v(s) &= \sigma_v \sqrt{\frac{L_v}{\pi V_a} \frac{1 + \frac{\sqrt{3}L_v}{V_a}s}{(1 + \frac{L_v}{V_a}s)^2}}, & H_q(s) &= \frac{-\frac{s}{V_a}}{1 + \frac{4b}{\pi V_a}s} H_w(s), \\
 H_w(s) &= \sigma_w \sqrt{\frac{L_w}{\pi V_a} \frac{1 + \frac{\sqrt{3}L_w}{V_a}s}{(1 + \frac{L_w}{V_a}s)^2}}, & H_r(s) &= \frac{\frac{s}{V_a}}{1 + \frac{3b}{\pi V_a}s} H_v(s),
 \end{aligned}$$

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

$$\begin{aligned}
 L_w &= h, & \sigma_w &= 0.1V_{20}, \\
 L_u = L_v &= \frac{h}{(0.177 + 0.000823h)^{1.2}}, & \frac{\sigma_u}{\sigma_w} = \frac{\sigma_v}{\sigma_w} &= \frac{1}{(0.177 + 0.000823h)^{0.4}},
 \end{aligned}$$

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

## Chapter 3

# Nonlinear Model Predictive Control (NMPC)

This chapter describes the Nonlinear Model Predictive Controller (NMPC) proposed by Reinhardt in his work [5]. Based on the background theory described here, it will later be explored in Chapter 4 how this NMPC controller can be improved to be better suited for the icing problem. As Reinhardt's NMPC controller was used in the author's project thesis [1], the theory is repeated in the following sections but updated where relevant. As seen in Section 3.0.3, the reference vector was updated, and an angular velocity reference was added to the cost function of the NMPC to add a damping effect to the response.

The Model Predictive Control (MPC) solves an optimization problem in order to find the optimal input needed to follow a desired reference. The optimal input minimizes a cost function, which is chosen as the error between the state and the desired reference, and is subject to constraints on the states, inputs, or other system variables. This is a key feature of the MPC, as actuator and safety limits can be taken into account. The MPC solves the optimization problem within a defined time horizon and predicts the optimal states and inputs over the given horizon length. Only the first input in the sequence is executed, allowing for the predicted states and optimal input to be adjusted in the next time step when the optimization problem is solved again. As the kinematic and dynamic equations that form the mathematical model of the UAV are nonlinear, the MPC used in this thesis for the inner-loop control of roll, pitch and airspeed is a nonlinear model predictive controller (NMPC).

As proposed by Reinhardt [5], the NMPC's dynamic model is described in Section 2.3, where Eq. 2.11 can be written as the continuous ODE  $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u})$ . This is discretized using an explicit Runge-Kutta method of order 4, becoming  $\mathbf{x}(k+1) = f_{RK4}(\mathbf{x}(k), \mathbf{u}(k))$ , and used where applicable.

### 3.0.1 Reduced attitude parametrization

Following Reinhardt's work [5], the reduced attitude vector  $\Gamma \in \mathbb{S}^2$  is used in order to control roll and pitch. It is defined as the representation of the vertical axis of the inertial frame  $\mathbf{e}_3 = [0, 0, 1]^\top$ , expressed in the body frame:

$$\Gamma = \mathbf{R}_{nb}^\top \mathbf{e}_3. \quad (3.1)$$

The reduced attitude vector can be parameterized as:

$$\Gamma(\phi, \theta) = [-\sin \theta \quad \cos \theta \sin \phi \quad \cos \theta \cos \phi]^\top, \quad (3.2)$$

where roll is given by  $\phi \in [-\pi, \pi]$ , pitch is given by  $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ , and, as observed, it is independent of yaw.

### 3.0.2 Disturbance observer

In order to account for unmodeled dynamics, modeling inaccuracies as well as disturbances, a disturbance observer is included in the NMPC model with the purpose of obtaining offset-free attitude stabilization. The disturbance observer, based on Reinhardt's work [5], affects airspeed  $V_a$ , sideslip angle  $\beta$ , angle of attack  $\alpha$  and angular velocity  $\omega_{nb}^s$ . The difference between the observed state and the one predicted by the controller at each time instant is given by:

$$\begin{bmatrix} \Delta V_a(t) \\ \Delta \beta(t) \\ \Delta \alpha(t) \end{bmatrix} = \begin{bmatrix} V_a(t) \\ \beta(t) \\ \alpha(t) \end{bmatrix} - \begin{bmatrix} V_{a,mpc}(t) \\ \beta_{mpc}(t) \\ \alpha_{mpc}(t) \end{bmatrix} \quad (3.3)$$

$$\Delta \omega_{nb}^s(t) = \omega_{nb}^s(t) - \omega_{nb,mpc}^s(t) \quad (3.4)$$

The disturbance estimates are then updated together with the NMPC by

$$\mathbf{d}_f(t) \leftarrow \mathbf{d}_f(t) + \mathbf{L}_f [\Delta V_a(t) \quad \Delta \beta(t) \quad \Delta \alpha(t)]^\top, \quad (3.5)$$

$$\mathbf{d}_m(t) \leftarrow \mathbf{d}_m(t) + \mathbf{L}_m \Delta \omega_{nb}^s(t), \quad (3.6)$$

where  $\mathbf{d}_f$  and  $\mathbf{d}_m$  are the force and moment disturbances, respectively, and initialized as  $\mathbf{d}_f(0) = \mathbf{d}_m(0) = \mathbf{0}_{3 \times 1}$ , and with  $\mathbf{L}_f$  and  $\mathbf{L}_m$  given by

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

The gains  $l_{V_a}, l_\beta, l_\alpha, l_p, l_q, l_r$  are given in Table 4.6, and the disturbances are finally added to  $\mathbf{f}$ :

$$\mathbf{f}(\mathbf{x}, \mathbf{u}, \mathbf{d}) \triangleq \mathbf{f}(\mathbf{x}, \mathbf{u}) + [\mathbf{d}_f^\top \quad \mathbf{0}_{1 \times 3} \quad \mathbf{d}_m^\top]^\top \quad (3.8)$$

### 3.0.3 Nonlinear Program

The NMPC tracks the reference vector  $\mathbf{r}$ , defined as

$$\mathbf{r} \triangleq [V_{a,ref} \quad \mathbf{\Gamma}_{ref}^\top \quad \boldsymbol{\omega}_{ref}^\top]^\top, \quad (3.9)$$

where the angular velocity vector  $\boldsymbol{\omega}_{ref}$  is added to the reference vector and to the cost function to add a damping effect, in order to achieve a smoother response, and  $\mathbf{\Gamma}_{ref}$  is parameterized using the references for roll and pitch,  $\phi_{ref}$  and  $\theta_{ref}$ , as shown in Eq. (3.2). The way the reference 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

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

which is a sum of quadratic terms, where  $q_{V_a}$  is the weighting scalar that penalizes the  $V_a$  error, and  $\mathbf{Q}_\Gamma \in \mathbb{R}^{3 \times 3}$ ,  $\mathbf{Q}_\omega \in \mathbb{R}^{3 \times 3}$  and  $\mathbf{R} \in \mathbb{R}^{n_u \times n_u}$  are the weighting matrices for attitude error, angular velocity and input, respectively. The cost function in Eq. (3.11) can also be written in a more compact form as:

$$l(\mathbf{x}, \mathbf{u}, \mathbf{r}) = \mathbf{r}^\top \mathbf{Q} \mathbf{r} + \mathbf{u}^\top \mathbf{R} \mathbf{u}, \quad (3.11)$$

where  $\mathbf{Q}$  and  $\mathbf{R}$  are given as:

$$\mathbf{Q} = \begin{bmatrix} q_{V_a} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & q_{\Gamma_1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & q_{\Gamma_2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_{\Gamma_3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & q_p & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & q_q & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & q_r \end{bmatrix}, \quad \mathbf{R} = \begin{bmatrix} r_{\dot{\delta}_a} & 0 & 0 \\ 0 & r_{\dot{\delta}_e} & 0 \\ 0 & 0 & r_{\dot{\delta}_t} \end{bmatrix} \quad (3.12)$$

The OCP over a prediction horizon  $T$  is then given by:

$$\min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot)} \int_0^T l(\mathbf{x}(\tau), \mathbf{u}(\tau), \mathbf{r}(\tau)) d\tau + \frac{1}{2} \mathbf{s}^\top \mathbf{P} \mathbf{s}, \quad t \in [0, T] \quad (3.13a)$$

$$\text{s. t. } \mathbf{x}(0) = \mathbf{x}_0 \quad (3.13b)$$

$$\dot{\mathbf{x}}(t) = \mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(0)), \quad t \in [0, T] \quad (3.13c)$$

$$\mathbf{h}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{s}) \geq \mathbf{0}, \quad t \in [0, T] \quad (3.13d)$$

where  $\mathbf{f}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(0))$  is the augmented system given in equation (3.8), and the term  $\frac{1}{2} \mathbf{s}^\top \mathbf{P} \mathbf{s}$  was added to penalize the vector of slack variables  $\mathbf{s} = [\mathbf{s}_x^\top \ \mathbf{s}_x^\top]^\top$  with the symmetric, positive-definite matrix  $\mathbf{P} \in \mathbb{R}^{n_s \times n_s}$ . All constraints on state  $\mathbf{x}$  and output  $\mathbf{u}$ , as well as their respective slack variables, are included in the constraint  $\mathbf{h}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{s})$ . This is explained in Section 3.0.4. Discretizing 3.13

into  $N$  steps with shooting interval  $\Delta t = T/N$  gives the nonlinear problem (NLP):

$$\min_{\mathbf{x}(\cdot), \mathbf{u}(\cdot)} \sum_{k=0}^{N-1} l(\mathbf{x}(k|t), \mathbf{u}(k|t), \mathbf{r}(k|t)) + \frac{1}{2} \mathbf{s}^\top \mathbf{P} \mathbf{s}, \quad k \in [0, \dots, N] \quad (3.14a)$$

$$\text{s. t. } \mathbf{x}(0|t) = \mathbf{x}(t) \quad (3.14b)$$

$$\mathbf{x}(k+1|t) = f_{RK4}(\mathbf{x}(k|t), \mathbf{u}(k|t), \mathbf{d}(0|t)), \quad k \in [0, \dots, N] \quad (3.14c)$$

$$\mathbf{h}(\mathbf{x}(k|t), \mathbf{u}(k|t), \mathbf{s}) \geq 0, \quad k \in [0, \dots, N], \quad (3.14d)$$

where  $\mathbf{x}(\cdot|t) \in \mathbb{R}^{n_x \times (N+1)}$  and  $\mathbf{u}(\cdot|t) \in \mathbb{R}^{n_u \times N}$  are the predicted state and input sequence, respectively. The MPC scheme is based on solving the NLP given by Eq. (3.14) at time  $t$ , for the predictions  $k \in [0, \dots, N]$ .

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

$$\mathbf{u}_{uav}(t) = [\mathbf{0}_{n_u \times (n_x - n_u)} \quad \mathbf{I}_{n_u \times n_u}] \mathbf{x}^*(1|t) \quad (3.15)$$

### 3.0.4 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 of the UAV. Let the constraints on the state  $\mathbf{x}$  and the constraints on the input  $\mathbf{u}$  to be given by  $\mathcal{X}$  and  $\mathcal{U}$ , respectively:

$$\mathcal{X} \triangleq \{\mathbf{x} \in \mathbb{R}^{n_x} | \mathbf{h}(\mathbf{x}, \mathbf{s}) \geq 0\} \quad (3.16)$$

$$\mathcal{U} \triangleq \{\mathbf{x} \in \mathbb{R}^{n_u} | \mathbf{u} - \underline{\mathbf{u}} \geq \mathbf{0} \wedge -\mathbf{u} + \bar{\mathbf{u}} \geq \mathbf{0}\} \quad (3.17)$$

When choosing the constraints on the state  $\mathbf{x}$ , limits on airspeed, sideslip 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 the actuators, which express the actual surface deflection limits of the elevons. All state constraints are given by:

$$\begin{aligned} V_a - \underline{V}_a + \underline{s}_{V_a} &\geq 0 & -V_a + \bar{V}_a + \bar{s}_{V_a} &\geq 0 \\ \beta - \underline{\beta} + \underline{s}_\beta &\geq 0 & -\beta + \bar{\beta} + \bar{s}_\beta &\geq 0 \\ \alpha - \underline{\alpha} &\geq 0 & -\alpha + \bar{\alpha} &\geq 0 \\ p_s - \underline{p}_s &\geq 0 & -p_s + \bar{p}_s &\geq 0 \\ q_s - \underline{q}_s &\geq 0 & -q_s + \bar{q}_s &\geq 0 \\ r_s - \underline{r}_s &\geq 0 & -r_s + \bar{r}_s &\geq 0 \\ \delta_a + \delta_e - \underline{\delta}_{el} &\geq 0 & -\delta_a - \delta_e + \bar{\delta}_{el} &\geq 0 \\ -\delta_a + \delta_e - \underline{\delta}_{er} &\geq 0 & \delta_a - \delta_e - \bar{\delta}_{er} &\geq 0 \\ \delta_t - \underline{\delta}_t &\geq 0 & -\delta_t + \bar{\delta}_t &\geq 0, \end{aligned} \quad (3.18)$$

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

$$\mathbf{s} = [\mathbf{s}_{V_a}^\top \quad \mathbf{s}_\beta^\top]^\top \geq 0. \quad (3.19)$$

Eq. (3.18) can be summarised by the function  $\mathbf{h}(\mathbf{x}, \mathbf{s}) \geq 0$ , and (Eq. 3.17) with (Eq. 3.18) can be written together as  $\mathbf{h}(\mathbf{x}, \mathbf{u}, \mathbf{s}) \geq 0$ .



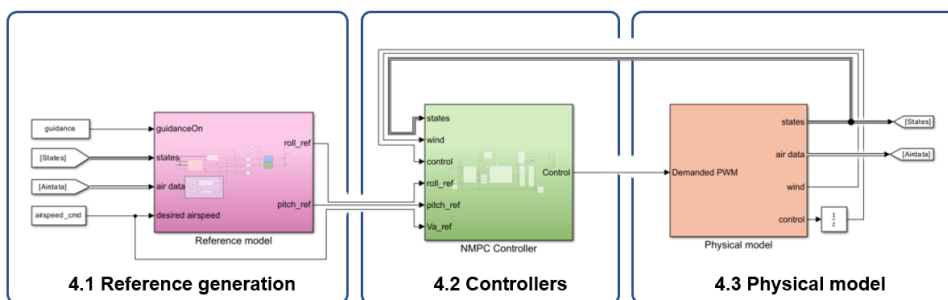


# Chapter 4

## Method

Given the theory in Chapter 3, this chapter explores different variants of the NMPC controller, evaluating what changes can be made to improve its performance in icing conditions. This chapter explains how the controllers were implemented, which simulations were performed, as well as the simulator in which the controllers were tested. The NMPC controllers were compared to previously developed PID and MRAC controllers [10], and their implementation and tuning are given in Appendix A, as no changes to them were made in this thesis.

In a classic control loop, the reference is generated, and the error between the reference and the current state of the UAV is found and sent to the controller, which finds the appropriate input and sends it to the physical model. The Matlab/ Simulink simulator used in this thesis, first developed by Gryte [11] and then extended by Högnadóttir [10], can be divided into three blocks: the Reference Generation block, the Controller block and the Physical Model block. This is illustrated in Figure 4.1. Following the same structure, the reference generation will be explained in Section 4.1, the NMPC controllers are explored in 4.2, and the implementation of the icing models into the simulator are described in Section 4.3. In addition, Section 4.4 will describe the simulations performed.



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

## 4.1 Reference Generation

In order to simulate the performance of the two NMPC controllers developed in this thesis, and compare them to the PID and MRAC controllers, two types of simulations are performed. First, the controllers need to be able to follow some commanded roll and pitch angles. Next, it is interesting to compare their abilities to follow a given path as well, by following commanded roll and pitch angles found by a guidance controller. Given a commanded roll and pitch angle, either chosen by us or by the guidance controller, the simulator generates a reference for the controllers to follow, using the methods described in this section. Parts of this section were described in the author's project thesis [1] and are repeated below.

### 4.1.1 Roll and Pitch Reference Model

The reference model that drives the roll and pitch angles to certain commanded angles,  $\phi_{cmd}$  and  $\theta_{cmd}$ , is described as the following second-order transfer function:

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

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

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 [10], they are set as  $\omega_{n,\phi} = \omega_{n,\theta} = 4$  and  $\zeta_\phi = \zeta_\theta = 1$ .

### 4.1.2 Angular Velocity Reference Model

In addition to roll and pitch, the NMPC controllers also follow a desired angular velocity  $\omega_{ref}^s$ . The reference model for  $\dot{\phi}_{ref}$  and  $\dot{\theta}_{ref}$  can be generated from the roll and pitch reference model, described in equation (4.1), by representing it as a linear time invariant (LTI) system:

$$\dot{\phi}_{ref} = \mathbf{A}_\phi \phi_{ref} + \mathbf{B}_\phi \phi_{cmd}, \quad (4.2)$$

$$\dot{\theta}_{ref} = \mathbf{A}_\theta \theta_{ref} + \mathbf{B}_\theta \theta_{cmd}, \quad (4.3)$$

$$(4.4)$$

where

$$\mathbf{A}_\phi = \begin{bmatrix} 0 & 1 \\ -\omega_{n,\phi}^2 & -2\zeta_\phi \omega_{n,\phi} \end{bmatrix}, \quad \mathbf{B}_\phi = \begin{bmatrix} 0 \\ \omega_{n,\phi}^2 \end{bmatrix}, \quad (4.5)$$

$$\mathbf{A}_\theta = \begin{bmatrix} 0 & 1 \\ -\omega_{n,\theta}^2 & -2\zeta_\theta \omega_{n,\theta} \end{bmatrix}, \quad \mathbf{B}_\theta = \begin{bmatrix} 0 \\ \omega_{n,\theta}^2 \end{bmatrix}. \quad (4.6)$$

In this case,  $\dot{\psi}$  can be found by using the expression for coordinated turn, from Beard & McLain [8]:

$$\dot{\psi}_{ref} = \frac{g}{V_a} \tan \phi_{ref} \quad (4.7)$$

Finally,  $\omega_{ref}^s$  is found by first transforming  $\dot{\Theta}_{ref}$  to body frame, and then to stability frame:

$$\omega_{ref}^s = R_{sb}(\alpha)^\top \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \dot{\Theta}_{ref}, \quad (4.8)$$

where  $R_{sb}(\alpha)$  is given by 2.2 and  $\alpha$  is the angle of attack.

### 4.1.3 Commanded Roll and Pitch from the Guidance Controller

The path-following abilities of the controllers are tested by letting them follow the commanded roll and pitch angles found by a guidance controller. When the guidance controller is used, it is included as part of the Reference Generation block, as seen in Figure 4.1, since it will be providing the commanded roll and pitch angles,  $\phi_{cmd}$  and  $\theta_{cmd}$ , that the UAV should follow. Given  $\phi_{cmd}$  and  $\theta_{cmd}$ , the references  $\phi_{ref}$  and  $\theta_{ref}$ , and  $p_{ref}^s$  and  $q_{ref}^s$ , are found following the procedures described in the Subsection 4.1.2. This section describes the guidance controller used to find the commanded roll and pitch angles, based on the path to be followed, which is given as a set of waypoints. The guidance laws are given in the lateral and longitudinal directions.

Based on Fossen [21], the proportional line-of-sight (LOS) guidance law is given by a course controller, which tracks a desired course angle  $\chi_d$  and calculates the corresponding roll angle reference as:

$$\phi_{cmd} = k_{p_\chi}(\chi_d - \chi) + \frac{k_{i_\chi}}{s}(\chi_d - \chi). \quad (4.9)$$

The desired course angle  $\chi_d$  is given by:

$$\chi_d = \chi_p - \tan^{-1} \left( \frac{y_e^p}{\Delta_{lat}} \right), \quad (4.10)$$

where the path-tangential angle  $\chi_p$  and the cross-track error  $y_e$  are found by:

$$\chi_p = \text{atan2}(y_{k+1} - y_k, x_{k+1} - x_k) \quad (4.11)$$

$$y_e = -(x(t) - x_k) \sin \chi_p + (y(t) - y_k) \cos \chi_p. \quad (4.12)$$

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 [22] and You [23], a flight path angle controller, which tracks a desired flight path angle  $\gamma_d$  and calculates the corresponding pitch angle reference is given by:

$$\theta_{cmd} = -c_1(\gamma - \gamma_d) + \gamma_d + \alpha_{trim}, \quad (4.13)$$

where the desired flight path angle is given by:

$$\gamma_d = \gamma_p + \tan^{-1} \left( \frac{k_{ph} z_e + k_{ih} \int z_e d\tau}{\Delta_{lon}} \right). \quad (4.14)$$

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

$$\gamma_p = \text{atan2}(-(z_{k+1} - z_k), L_{xy}), \quad (4.15)$$

$$z_e = S_{xy} \sin(\gamma_p) + (z(t) - z_k) \cos(\gamma_p). \quad (4.16)$$

$S_{xy}$  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:

$$S_{xy} = \cos(\chi_p)(x(t) - x_k) + \sin(\chi_p)(y(t) - y_k) \quad (4.17)$$

$$L_{xy} = \sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2}. \quad (4.18)$$

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

$$\delta_t = \delta_t^* + k_{p_v}(V_{a,cmd} - V_a) + \frac{k_{i_v}}{s}(V_{a,cmd} - V_a) + k_{p_{z_e}} z_e(t). \quad (4.19)$$

The tuning parameters used in the guidance controller come from the results of Högnadottir's work [10], and given in Table 4.1.

Parameter	Value
$\Delta_{lat}$	33
$k_{p\chi}$	2
$k_{i\chi}$	0.1
$k_{ph}$	0.8
$k_{ih}$	0.1
$R_{max}$	20°
$c_1$	0.001
$k_{pV}$	0.6
$k_{iV}$	0.01
$k_{pze}$	0.1
$\delta^*$	0.44

**Table 4.1:** Tuning parameters of the guidance controller

When using the guidance controller, a reference for  $\dot{\psi}_{ref}$  can be found more directly than the one from Eq. (4.7) by derivating the LOS guidance law in Eq. (4.10), and using the relationship between yaw, course and crab angle:

$$\psi = \chi - \beta_c \quad (4.20)$$

Following Gryte's derivations in his dissertation [24], derivating Eq. (4.10) gives

$$\dot{\chi}_d = -\frac{\Delta}{\Delta^2 - y_e^2} \dot{y}_e + \dot{\chi}_p. \quad (4.21)$$

By derivating Eq. (4.20) and inserting it into Eq. (4.21), it results in

$$\dot{\psi}_d = \dot{\chi}_d - \dot{\beta}_{c,d} = -\frac{\Delta}{\Delta^2 - y_e^2} \dot{y}_e + \dot{\chi}_p - \dot{\beta}_{c,d} \quad (4.22)$$

$$\dot{\psi}_d = -\frac{\Delta}{\Delta^2 - y_e^2} \dot{y}_e, \quad \dot{\chi}_p \approx 0, \quad \dot{\beta}_{c,d} \approx 0, \quad (4.23)$$

as both  $\dot{\chi}_p$  and  $\dot{\beta}_{c,d}$  can be assumed to be 0. Finally,  $\omega_{ref}^s$  is found by following Eq. (4.8).

## 4.2 Implementation of the NMPC Controllers

This thesis explores how the accuracy of the NMPC's model, more specifically how including the modeled icing effects, affects the robustness and performance of the controller when subjected to the mentioned disturbances. For this purpose, an NMPC controller without icing in its model is compared to an NMPC controller which includes the effects of icing on the aerodynamic forces and moments in its model, including asymmetry, as explained in Section 2.6. Both NMPC controllers are based on the NMPC controller developed by Reinhardt in his doctoral

thesis [5], written in Python using the software package Acados [7]. However, Reinhardt's NMPC used the parameters from Gryte [25] to model the aerodynamic forces and moments. This was updated in this thesis to use Winter's model, with the parameters found in his work [12], to match the model used in the Matlab/Simulink simulator and reduce model mismatches. In Sections 4.2.2 and 4.2.3 this is explained in more detail, as well as how the icing model is implemented in the case of the NMPC with icing in its model.

The NMPC in Python, based on Reinhardt's work [5], discretizes Eq. (3.13), in this case into  $N = 35$  steps, using direct multiple-shooting with a fourth-order explicit Runge-Kutta integrator. The nonlinear problem in Eq. (3.14) is solved using a real-time iteration sequential quadratic programming (RTI SQP) 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.

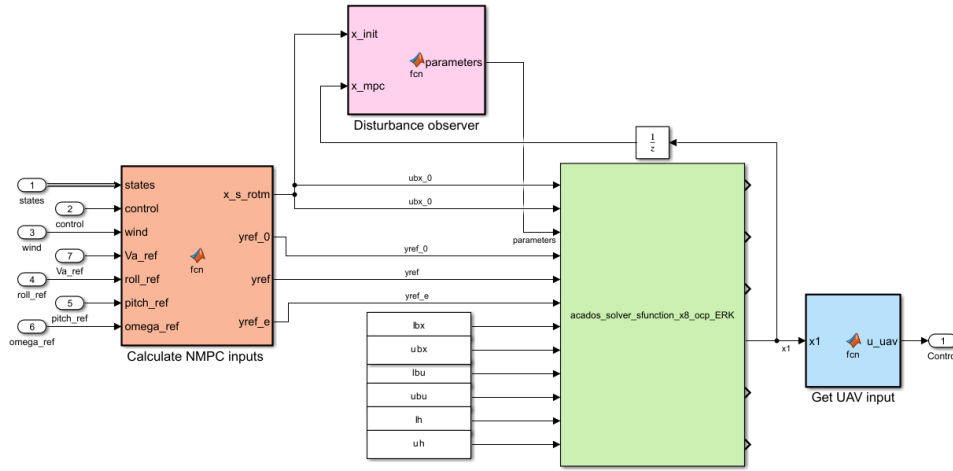
The constraints used in the NLP are given in Table 4.2, where the constraints on  $\delta_x$  are given by the real actuator limits. When it comes to the angle of attack  $\alpha$ , the upper and lower bounds should be chosen as the stall limit for a fully iced wing. The stall angle is however not yet clearly determined in the existing literature. Winter finds  $\alpha_{stall}$  to be  $\approx 10^\circ$  [12], but Högnadóttir describes it might be closer to  $\alpha_{stall} \approx 4^\circ$  [10]. In the author's project thesis [1],  $\bar{\alpha}$  was set to  $4^\circ$  to test the abilities of the NMPC controller. Although it worked, more research needs to be made in order to conclude that the stall limit is in fact as low as  $4^\circ$ , so in this thesis, the limit  $10^\circ$  was assumed instead. To ensure that it is not crossed,  $\bar{\alpha}$  was set to  $8^\circ$ . When it comes to airspeed, the lower and upper limits are set as the safety limits of  $\underline{V}_a = 15 \text{ m/s}$  and  $\bar{V}_a = 25 \text{ m/s}$ .

Variable	Value	
$\underline{V}_a, \bar{V}_a$	15, 25	m/s
$\underline{\beta}, \bar{\beta}$	-90, 90	deg
$\underline{\alpha}, \bar{\alpha}$	-8, 8	deg
$\underline{p}^s, \bar{p}^s; \underline{q}^s, \bar{q}^s; \underline{r}^s, \bar{r}^s$	-180, 180; -180, 180; -180, 180	deg/s
$\underline{\delta}_a, \bar{\delta}_a$	-35, 35	deg
$\underline{\delta}_e, \bar{\delta}_e$	-35, 35	deg
$\underline{\delta}_r, \bar{\delta}_r$	0, 0	deg
$\underline{\delta}_t, \bar{\delta}_t$	0, 1	-
$\underline{\delta}_{el}, \bar{\delta}_{el}, \underline{\delta}_{er}, \bar{\delta}_{er}$	-30, 20, -30, 20	deg

**Table 4.2:** Inequality constraints in the NMPC controller

To implement the NMPC controller into the Matlab/ Simulink simulator, C code was generated from the Python code and interfaced with the simulator through an S-function, following the examples in the Acados documentation [26]. To solve the optimization problem, the S-function containing the NMPC takes in a series of

inputs, containing information about the reference to be followed, the constraints and the disturbances, as illustrated in Figure 4.2. A list with all the input signals to the S-function block and the information they contain is given in Table 4.3. The disturbance observer, as well as the information about the icing level, are implemented through the input **parameters**. When the NMPC controller without icing implemented in its model is run, the icing level input is always set to 0 (clean state). This way, the controller assumes that there is no ice on the wings, and icing is treated as another disturbance that the NMPC has no knowledge of, to be handled by the disturbance observer.



**Figure 4.2:** The NMPC controller implemented as an S-function block, with its inputs and outputs, calculated as described in Table 4.3.

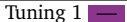
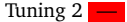
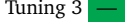
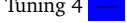
Input	Input
ubx_0	$[V_a \ \alpha \ \beta \ r_x^T \ r_y^T \ r_z^T \ \omega^s{}^T \ \delta_a \ \delta_e \ \delta_t]^T$
lbx_0	$[V_a \ \alpha \ \beta \ r_x^T \ r_y^T \ r_z^T \ \omega^s{}^T \ \delta_a \ \delta_e \ \delta_t]^T$
parameters	$[V_{wind}^n \ d_P^T \ d_M^T \ \text{icing\_left\_wing} \ \text{icing\_right\_wing}]^T \times N + 1$
yref_0	$[V_{a,ref} \ \Gamma_{1,ref} \ \Gamma_{2,ref} \ \Gamma_{3,ref} \ \omega_{ref}^s{}^T \ 0 \ 0 \ 0]^T$
yref	$[V_{a,ref} \ \Gamma_{1,ref} \ \Gamma_{2,ref} \ \Gamma_{3,ref} \ \omega_{ref}^s{}^T \ 0 \ 0 \ 0]^T \times N - 1$
yref_e	$[V_{a,ref} \ \Gamma_{1,ref} \ \Gamma_{2,ref} \ \Gamma_{3,ref} \ \omega_{ref}^s{}^T]^T$
lbx	$[V_a \ \alpha \ \beta \ \omega^s{}^T \ \delta_a \ \delta_e \ \delta_t]^T \times N - 1$
ubx	$[\bar{V}_a \ \bar{\alpha} \ \bar{\beta} \ \bar{\omega}^s{}^T \ \bar{\delta}_a \ \bar{\delta}_e \ \bar{\delta}_t]^T \times N - 1$
lbu	$[\dot{\delta}_a \ \dot{\delta}_e \ \dot{\delta}_t]^T \times N$
ubu	$[\bar{\delta}_a \ \bar{\delta}_e \ \bar{\delta}_t]^T \times N$
lh	$[\delta_{el} \ \delta_{er}]^T \times N$
uh	$[\bar{\delta}_{el} \ \bar{\delta}_{er}]^T \times N$

**Table 4.3:** Inputs into the S-function block that contains the NMPC controller. Using these inputs, the NMPC finds an optimal solution to the NLP, and the input to be used by the actuators.

### 4.2.1 Tuning of the NMPC

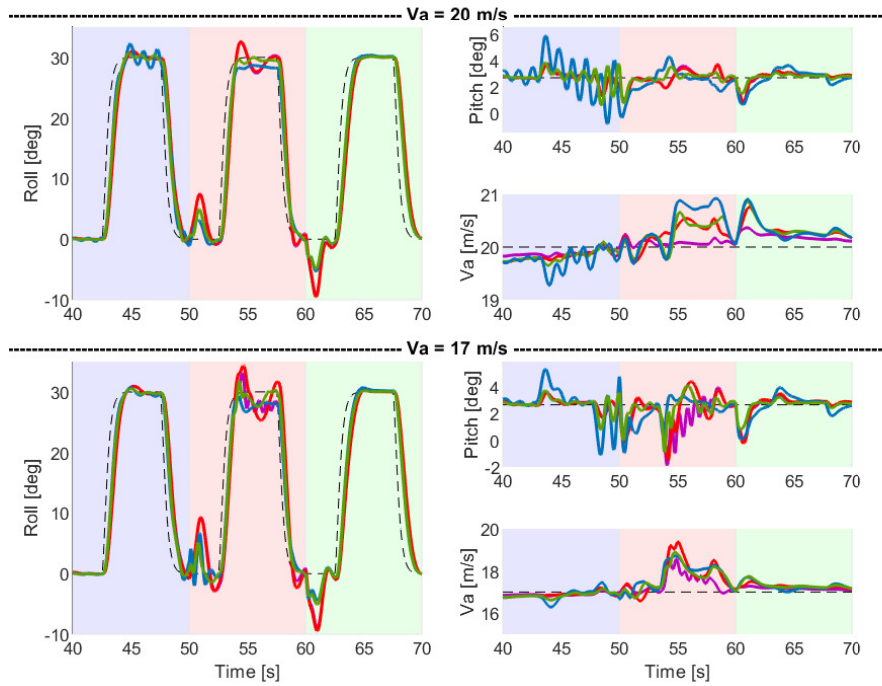
The NMPC was tuned by adjusting the weighting matrices  $\mathbf{Q}$  and  $\mathbf{R}$ , given by Eq. (3.12), as well as the disturbance observer's weighting matrices  $\mathbf{L}_f$  and  $\mathbf{L}_m$ , given by Eq. (3.7). The values for the disturbance observer were found first, to correct a large offset error that was observed in roll, pitch and airspeed when the disturbance observer was turned off ( $l_x = 0$ ). The original weights chosen were  $l_{V_a} = l_\beta = 0.01$ ,  $l_\alpha = l_n = 0.1$ ,  $l_l = l_m = 0.5$ , since the largest offset error was observed in roll and pitch, and a lower weight on airspeed seemed sufficient to remove the airspeed offset error. Then, the first set of  $\mathbf{Q}$  and  $\mathbf{R}$  weights were found. The tuning process involved changing individual weights of the two matrices at a time based on the response obtained, to observe the impact it has on the performance and whether it improves as a result. To find an optimal tuning, the roll angle reference was changed while pitch was kept constant at 0 degrees and vice versa, similar to the simulations performed later in Section 4.4. The performance was considered in the case of 100 % icing on both wings, 100 % asymmetry, and in the case of no ice on the wings. In addition, reduced airspeed was also tested and, as the tuning of the NMPC improved, severe wind conditions were taken into consideration as well. The first four tuning cases are shown in Table 4.4, and their results can be seen in Figure 4.3, where the blue, red and green background colors represent the time intervals in which the wings were iced, asymmetric (one iced and one clean) and clean, respectively.

With Tuning 1, an aggressive pitch response was observed in Figure 4.3b during the 100 % asymmetry, reduced airspeed case. This was improved in Tuning 2 by reducing the weight on airspeed  $q_{V_a}$ , although the airspeed performance worsened, as expected. As throttle had a lower weight than aileron and elevator, and with the idea to improve the roll and pitch tracking performances,  $r_{\delta_a}$  and  $r_{\delta_e}$  were reduced in Tuning 3. This improved the roll performance significantly in Figure 4.3a, but the pitch response during the asymmetry, low airspeed interval became again very aggressive, as well as the roll during this interval, as seen in Figure 4.3b. This suggests that the weight on  $r_{\delta_e}$  is perhaps too low. However, since angular velocity is part of the cost function, increasing the weight on pitch rate  $q_q$  should lead to a damping effect, so this approach was inspected next in Tuning 4. This resulted in a slower, more controlled pitch response during the asymmetry interval, but oscillations appeared during the iced interval in both pitch and roll, as seen in Figure 4.3.

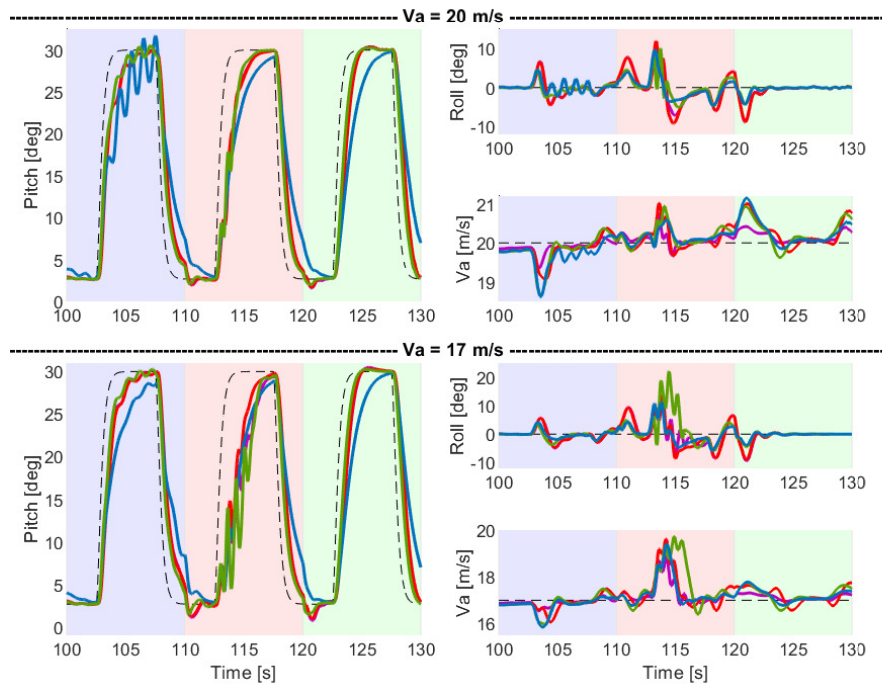
	$q_{V_a}$	$q_{r_1}$	$q_{r_2}$	$q_{r_3}$	$q_p$	$q_q$	$q_r$	$r_{\delta_a}$	$r_{\delta_e}$	$r_{\delta_l}$	$l_{V_a}$	$l_\beta$	$l_\alpha$	$l_l$	$l_m$	$l_n$
Tuning 1 	0.1	10	10	10	1	1	1	1	1	0.01	0.01	0.01	0.1	0.5	0.5	0.1
Tuning 2 	<b>0.01</b>	10	10	10	1	1	1	1	1	0.01	0.01	0.01	0.1	0.5	0.5	0.1
Tuning 3 	0.01	10	10	10	1	1	1	<b>0.1</b>	<b>0.1</b>	0.01	0.01	0.01	0.1	0.5	0.5	0.1
Tuning 4 	0.01	10	10	10	1	<b>10</b>	1	0.1	0.1	0.01	0.01	0.01	0.1	0.5	0.5	0.1

**Table 4.4:** First round of tuning.





(a) Roll response.



(b) Pitch response.

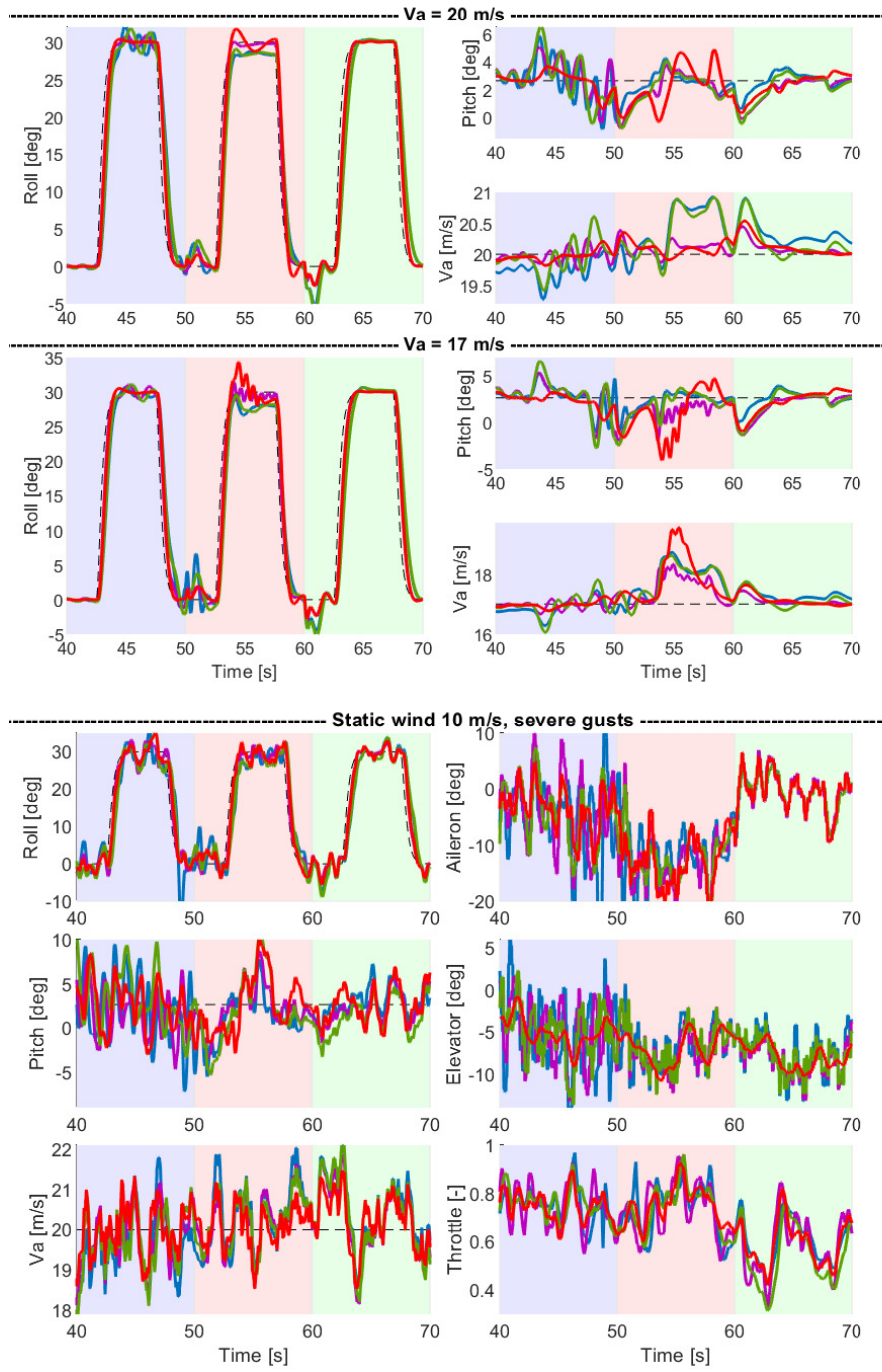
**Figure 4.3:** Roll and pitch responses with the tuning parameters given in Table 4.4, and with two different airspeed references. The background colors represent the icing level simulated: blue for 100 % icing, red for 100 % asymmetry and green for the clean state (no ice).

A second round of tuning options is presented in Table 4.5, which in this case were tested with severe wind conditions as well to ensure the robustness of the final tuning.

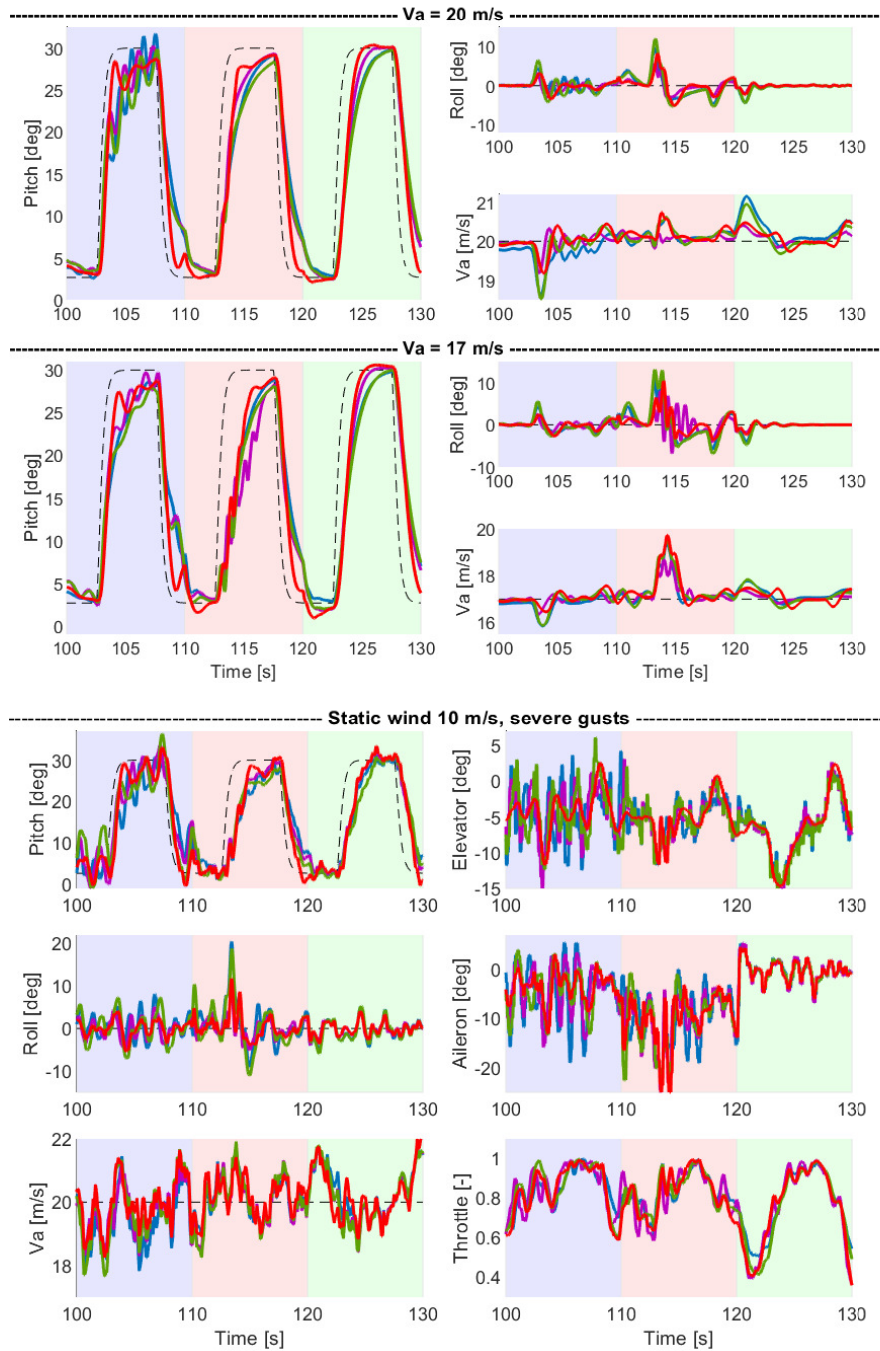
The results are shown in Figures 4.4 and 4.5. With the added challenge of wind disturbances, it was observed that modifying some of the weights of the NMPC was not sufficient to both improve the performance and keep the NLP feasible under these severe wind conditions. Therefore, slightly adjusting the disturbance observer was found to be more beneficial, as shown by Tuning 5. The weights were found by trial and error, but increasing  $l_{V_a}$  helped reduce the offset error in airspeed, as seen more clearly during the iced and clean time intervals in Figure 4.4. As the weight on the angle of attack was high compared to the one on airspeed and sideslip angle,  $l_\alpha$  was decreased and it was found to work well. The weights on roll and pitch moment were also decreased, which helped reduce the "noisiness" when the wind disturbances were on, as seen in Figures 4.4 and 4.5 in the roll, pitch, aileron and elevator responses. In Tuning 6, the effects of the weights on  $\Gamma$ , the reduced-attitude vector, are explored. Following Reinhardt's explanations in his work [5], higher weights on  $\Gamma_1$  and  $\Gamma_2$  would result in a tighter tracking of the pitch and roll angles, respectively, but higher  $\Gamma_1$  and  $\Gamma_2$  compared to  $\Gamma_3$  can also temporarily increase the cost function, and might require a longer horizon  $N$  to be able to converge. Increasing all three  $\Gamma_1$ ,  $\Gamma_2$  and  $\Gamma_3$  to 100 was found to be too aggressive, while the final Tuning 6 gave more favorable results. As pitch is also a part of  $\Gamma_2$  and  $\Gamma_3$ , this tuning keeps the horizon  $N$  short while improving both pitch and roll, although favoring roll. In addition, increasing  $q_{V_a}$  improved the airspeed tracking, as seen in both Figure 4.4 and 4.5. A more aggressive pitch tracking, however, brings back the problem of oscillations during the reduced airspeed, iced and asymmetry intervals.

	$q_{V_a}$	$q_{\Gamma_1}$	$q_{\Gamma_2}$	$q_{\Gamma_3}$	$q_p$	$q_q$	$q_r$	$r_{\dot{\delta}_a}$	$r_{\dot{\delta}_s}$	$r_{\dot{\delta}_i}$	$l_{V_a}$	$l_\beta$	$l_\alpha$	$l_l$	$l_m$	$l_n$
Tuning 4	0.01	10	10	10	1	10	1	0.1	0.1	0.01	0.01	0.01	0.1	0.5	0.5	0.1
Tuning 5	0.01	10	10	10	1	10	1	0.1	0.1	0.01	<b>0.03</b>	0.01	<b>0.01</b>	<b>0.4</b>	<b>0.2</b>	0.1
Tuning 6	<b>0.1</b>	10	<b>100</b>	<b>100</b>	1	10	1	0.1	0.1	0.01	0.03	0.01	0.01	0.4	0.2	0.1
Tuning 7	0.1	<b>20</b>	100	100	1	<b>1</b>	1	<b>0.2</b>	<b>7</b>	<b>0.05</b>	0.03	0.01	0.01	0.4	<b>0.1</b>	0.1

**Table 4.5:** Second round of tuning.



**Figure 4.4:** Roll response with the tuning parameters given in Table 4.5, given two different airspeed references and severe wind conditions. The background colors represent the icing level simulated: blue for 100 % icing, red for 100 % asymmetry and green for the clean state (no ice).



**Figure 4.5:** Pitch response with the tuning parameters given in Table 4.5, given two different airspeed references and severe wind conditions. The background colors represent the icing level simulated: blue for 100 % icing, red for 100 % asymmetry and green for the clean state (no ice).

Finally, Tuning 7 was found by examining how changing the different weights of  $\mathbf{Q}$  and  $\mathbf{R}$  affected the performance under the different disturbances, based on the other tuning options presented. Generally, more aggressive roll and pitch controllers worsen the performance in the reduced airspeed case, as they add oscillations and, in some cases, can make the NLP unfeasible. However, a faster controller is necessary under severe wind conditions. As the oscillations in pitch were more severe, the pitch tuning was prioritized. Increasing  $q_q$  resulted in oscillations during the iced interval, as seen by Tuning 4 in Figure 4.5, as well as a slower pitch response. Therefore, in Tuning 7,  $r_{\delta_e}$  is increased instead in order to restrict elevator, and  $q_{\Gamma_1}$  is increased to improve the pitch tracking. Aileron and throttle are also slightly more restricted. Increasing the elements in the  $\mathbf{R}$  matrix reduces many of the oscillations in aileron, elevator and throttle during the wind conditions, as seen in Figures 4.4 and 4.5. These improvements, however, slightly reduce the roll tracking performance.

The final weights that performed the best among the other options tested were found by trial and error. Tuning 7 is the final tuning chosen for the NMPC controller without icing included in its model, and it is the one used in the future simulations, described in Section 4.4. When it comes to the NMPC with icing in its model, which is to be explained in Section 4.2.3, it was tuned more aggressively, as it will be shown to be able to handle the disturbances caused by the icing conditions better. In its case, Tuning 7 was taken as the starting point, and the  $q_{\Gamma}$  weights were increased as  $\mathbf{R}$  was decreased to make the controller more aggressive and improve its performance. This was done by trial and error. The tuning parameters for both controllers are presented in Table 4.6.

(a) Tuning matrices of NMPC without icing		(b) Tuning parameters of the disturbance observer with the NMPC without icing	
	Parameters		Parameters
$\mathbf{Q}$	diag[0.1, 20, 100, 100, 1, 1, 1]	$l_{V_a}, l_{\beta}, l_{\alpha}$	0.03, 0.01, 0.01
$\mathbf{R}$	diag[0.2, 7, 0.05]	$l_l, l_m, l_n$	0.4, 0.1, 0.1
$\mathbf{P}$	diag[1, 1, 1, 1]		

(c) Tuning matrices of NMPC with icing		(d) Tuning parameters of the disturbance observer with the NMPC without icing	
	NMPC without icing		Value
$\mathbf{Q}$	diag[0.1, 100, 200, 200, 1, 1, 1]	$l_{V_a}, l_{\beta}, l_{\alpha}$	0.03, 0.01, 0.01
$\mathbf{R}$	diag[0.2, 1, 0.01]	$l_l, l_m, l_n$	0.4, 0.1, 0.1
$\mathbf{P}$	diag[1, 1, 1, 1]		

**Table 4.6:** Tuning parameters of the NMPC controller.

### 4.2.2 NMPC without icing: updated model

In the NMPC controller developed by Reinhardt, the model that describes the aerodynamic forces and moments is based on the model and aerodynamic coefficients found by Gryte [25]. Compared to the Matlab simulator, which uses Winter's model [12], including iced and clean values for the aerodynamic coefficients, the NMPC's model is outdated. Therefore, although the first NMPC controller does not include the effects of icing, its model is updated based on the results from Winter's work [12], where only the "clean" aerodynamic coefficients are implemented into equations (2.17)-(2.23). In his work, the aerodynamic coefficient data was found by experiment for a given angle of attack and sideslip angle, and interpolation was used to find the aerodynamic coefficients between these data points. As the NMPC controller is written in Python, which is later converted into C code and used in the Matlab simulator, using interpolation in the same way as described in Section 4.3.1 is not possible, as look-up tables were not supported as a part of the OCP problem. The NMPC model uses therefore first-order polynomial approximations of the aerodynamic coefficients, found by curve fitting in Winter [12], which are generally good in the angle of attack interval  $\alpha \in [-5, 10]$  degrees. The coefficients are then modeled as

$$C_k(x) = C_{k_1}x + C_{k_0}, \quad (4.24)$$

where  $x = \alpha, \beta$ , based on which angle the coefficient is dependent on. When it comes to the drag coefficient  $C_D$ , Winter did not include an approximation in his work, so a third-order polynomial was found to ensure a good fit:

$$C_D(\alpha) = C_{D_3}\alpha^3 + C_{D_2}\alpha^2 + C_{D_1}\alpha + C_{D_0}. \quad (4.25)$$

The parameters used are given in Table 4.7.

	$C_{k_3}$	$C_{k_2}$	$C_{k_1}$	$C_{k_0}$		Value		Value
$C_D(\alpha)$	1.605	0.823	0.010	0.016	$C_{D_q}$	0.0		
$C_L(\alpha)$	0.0	0.0	4.06	0.03	$C_{S_r}$	0.005	$C_{L_{\delta_e}}$	0.2780
$C_{L_q}(\alpha)$	0.0	0.0	-0.381	4.653	$C_{S_p}$	-0.085	$C_{D_{\delta_e}}$	0.0633
$C_m(\alpha)$	0.0	0.0	-0.61	0.0	$C_{l_r}$	0.039	$C_{m_{\delta_e}}$	-0.2060
$C_{m_q}(\alpha)$	0.0	0.0	-0.0955	-1.987	$C_{l_p}$	-0.409	$C_{S_{\delta_a}}$	0.0433
$C_S(\beta)$	0.0	0.0	-0.27	0.0	$C_{n_r}$	-0.022	$C_{l_{\delta_a}}$	0.1200
$C_l(\beta)$	0.0	0.0	-0.101	0.0	$C_{n_p}$	0.027	$C_{n_{\delta_a}}$	-0.00339
$C_n(\beta)$	0.0	0.0	0.0297	0.0				

Table 4.7: Parameters of the clean aerodynamic coefficients.

### 4.2.3 NMPC with icing in its model

This thesis aims to improve the robustness of the previously described NMPC and make it more suitable for icing conditions. For this purpose, the mathematical

model of the NMPC has been extended to include the aerodynamic, asymmetric icing model developed by Kleiven [13] and described in Section 2.6, together with the iced aerodynamic coefficients found in Winter's work [12]. The iced coefficients were implemented in the same way as the clean coefficients, following the same procedure described in Section 4.2.2. Curve fitting, more specifically first or third-order polynomials were used again to approximate the measured coefficients. The parameters used are given in Table 4.8.

	$C_{k_3}$	$C_{k_2}$	$C_{k_1}$	$C_{k_0}$		Value		Value
$C_D(\alpha)$	-6.454	4.041	0.043	0.0428	$C_{D_q}$	0.0		
$C_L(\alpha)$	0.0	0.0	3.26	0.01	$C_{S_r}$	0.002	$C_{L_{\delta_e}}$	0.2780
$C_{L_q}(\alpha)$	0.0	0.0	-1.9099	-3.243	$C_{S_p}$	-0.133	$C_{D_{\delta_e}}$	0.0633
$C_m(\alpha)$	0.0	0.0	-0.32	0.0	$C_{l_r}$	0.158	$C_{m_{\delta_e}}$	-0.2060
$C_{m_q}(\alpha)$	0.0	0.0	-1.528	-1.957	$C_{l_p}$	-0.407	$C_{S_{\delta_a}}$	0.0433
$C_S(\beta)$	0.0	0.0	-0.23	0.0	$C_{n_r}$	-0.049	$C_{l_{\delta_a}}$	0.1200
$C_l(\beta)$	0.0	0.0	-0.861	0.0	$C_{n_p}$	0.017	$C_{n_{\delta_a}}$	-0.00339
$C_n(\beta)$	0.0	0.0	0.0348	0.0				

**Table 4.8:** Parameters of the iced aerodynamic coefficients.

#### 4.2.4 Binary NMPC with icing in its model

The NMPC with icing in its model takes the icing level on each wing as a parameter and calculates the corresponding aerodynamics coefficients, following Eq. (2.6). The forces and moments are calculated taking asymmetry into account, following the icing model presented in Section 2.6. Throughout this thesis, the icing level on each wing has been assumed to be known, so it has been sent from the simulator to the NMPC as it is. In practice, the icing level can be estimated [27][28] or measured, but the same precision will not be achieved. Because of that, it is interesting to see how the NMPC with icing in its model performs when the information it gets about the icing level is less accurate. Assuming that the UAV will use an icing sensor to detect icing, its minimal requirement would be to detect whether the wing is in a clean or iced state. Taking this scenario into account, another NMPC is explored, which, as the title of this section suggests, only gets minimal information about the icing level, either clean (0) or iced (1). In this case, it is assumed that the shift in the state happens somewhere in the middle, so that under 50 % icing level the wing is assumed to be clean, and over 50 % it is assumed to be iced. By placing a sensor on each wing, the effects of asymmetry would also be taken into account. The performance of the NMPC in this scenario is tested in Section 5.4. Under the simulations, a spike was observed when the controller would suddenly get a shift in icing state, so a low-pass filter was used in this case to make the icing level transition smoothly into a different state.

### 4.3 Implementation of the Physical Model

This section describes how the physical model of the UAV, following the equations of motion described in Section 2.2, as well as the icing models described in Section 2.6, were implemented. As this was done in the project thesis and no changes were made, this section is repeated from [1].

#### 4.3.1 Icing on the Wings

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 4.9. 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* [29], as done in Högnadóttir's work [10]. As the aerodynamic coefficients are given for the either clean or iced case, a linear interpolation is used, following Kleiven's work [13], to find the coefficients in an intermediate ice state:

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

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	$\rho$	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 4.9: Physical parameters of the Skywalker X8.

#### 4.3.2 Propeller Icing

The propeller icing model was implemented following Müller's explanations [16] and his help. For the development of the model, the wind-tunnel experiments



Müller conducted were performed on the Mejlík 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 [17]. The clean power coefficients  $C_{P, clean}$  are taken from [30], 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 4.10.

(a) $C_T$ parameters		(b) $C_P$ parameters	
Parameter	Value	Parameter	Value
$C_{T,0}$	0.126	$C_{P,0}$	0.032989
$C_{T,1}$	-0.1378	$C_{P,1}$	0.0866
$C_{T,2}$	0	$C_{P,2}$	-0.1623

**Table 4.10:** Parameters the clean thrust and power coefficients estimation.

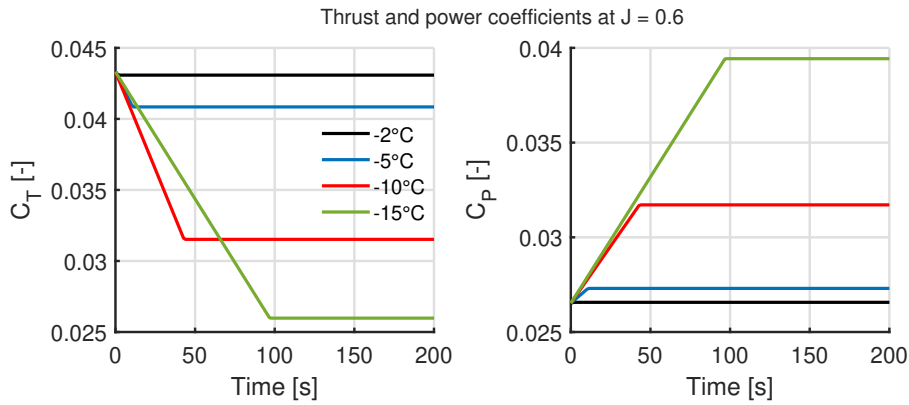
When it comes to the adhesion force  $A_{max}$ , it depends on the material and surface finish of the propeller, so it is assumed to be the same, as both the Mejlík and Aeronaut propellers are carbon fiber propellers. The  $\Delta C_T$  and  $\Delta C_P$ , however, depend on the twist and chord distribution of the propeller blades. The Mejlík 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  $\Delta C_T$  and  $\Delta C_P$  found by Müller were assumed to be transferable as well. The final parameters used are given in Table 4.11.

(a) $\Delta C_T$ parameters		(b) $\Delta C_P$ parameters		(c) $A_{max}$ parameters	
Parameter	Value	Parameter	Value	Parameter	Value
$\Delta C_{T,0}$	0.0233	$\Delta C_{P,0}$	-0.00890	$A_{max,0}$	37.250
$\Delta C_{T,1}$	0.0254	$\Delta C_{P,1}$	-0.0166	$A_{max,1}$	1223
$\Delta C_{T,2}$	0.00140	$\Delta C_{P,2}$	-5.79e-04		

**Table 4.11:** Parameters for the  $\Delta C_T$ ,  $\Delta C_P$  and  $A_{max}$  estimation.

Looking at Eq. (2.33) and (2.35), 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  $LWC = 0.44 \text{ g/m}^3$ . Given the results in Figure 4.6, the worst final perform-

ance degradation was given at  $T = -15^\circ\text{C}$  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. (2.33), choosing  $T = -15^\circ\text{C}$ ,  $LWC = 0.44\text{ g/m}^3$ , and  $t = 200\text{s}$ , 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 4.6, the power coefficient, and subsequently the torque, were not shown to increase substantially, so the previous claim is assumed to remain valid.



**Figure 4.6:** Thrust and power coefficient of the propeller for an advance ratio of  $J = 0.6$  and LWC of  $0.44\text{ g/m}^3$ .

## 4.4 Simulations

This section describes the simulations performed in this thesis to test the performance of the NMPC controllers when subjected to icing conditions, together with other severe disturbances, and compare it to the PID and MRAC controllers, given in Appendix A. To test the performance of the controllers, a set of simulations was run in which the roll and pitch angles were driven to a given angle, while the icing level and disturbance type varied. These simulations are given in Section 4.4.2 and 4.4.3. Section 4.4.4 describes the guidance simulations, in which the path-following performance of the controllers was tested, while subjected to icing and wind disturbances. In Section 4.4.5, additional simulations that test the performance of the NMPC controllers specifically are described. To be able to objectively analyze the performance of each controller, the simulations in Section 4.4.2 use the Integral Absolute Error (IAE) performance metric, presented in the next section.

#### 4.4.1 Performance Metric

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

$$IAE = \int_0^t |e_a(\tau)| d\tau. \quad (4.26)$$

Since the three controllers are following references for roll, pitch and airspeed, the reference errors, in this case, are  $e_\phi = \phi_{ref} - \phi$ ,  $e_\theta = \theta_{ref} - \theta$  and  $e_{V_a} = V_{a,ref} - V_a$ . To assess performance, the IAE values of each controller are compared under a certain time interval. A higher IAE indicated a larger error. By comparing how fast the IAE changes during, for example, a time interval in which icing asymmetry is simulated, we can see which controllers are more affected by it.

#### 4.4.2 Baseline simulation

The first simulation that is run is referred to as the baseline simulation, as it does not include any disturbances other than icing, and serves as a baseline for the next two simulations in Section 4.4.3. Throughout the simulation, the airspeed reference is set to  $V_a = 20 \text{ m/s}$ . As illustrated in Figure 4.7, the simulation starts with an excitation of the MRAC controller that lasts for 50 seconds, and which is run to ensure that its states converge to a good tuning. This is done while the references for roll and pitch of the NMPC and PID controllers are kept at the trim states of  $\phi_{cmd} = 0^\circ$  and  $\theta_{cmd} = 2.659^\circ$ , respectively. After the excitation period, the first half of the simulation is run. Here, the reference in roll varies between 0 and 30 degrees while pitch is kept at 2.66 degrees, throughout different icing level intervals: once without icing, the second time as the icing level on each wing builds up, the third time during the iced state in which both wings have reached a maximum ice accretion point, and the last time during the 100 % icing asymmetry interval. In the second part of the simulation, the same is repeated this time with pitch, varying between 2.66 and 30 degrees while roll is kept at 0 degrees.

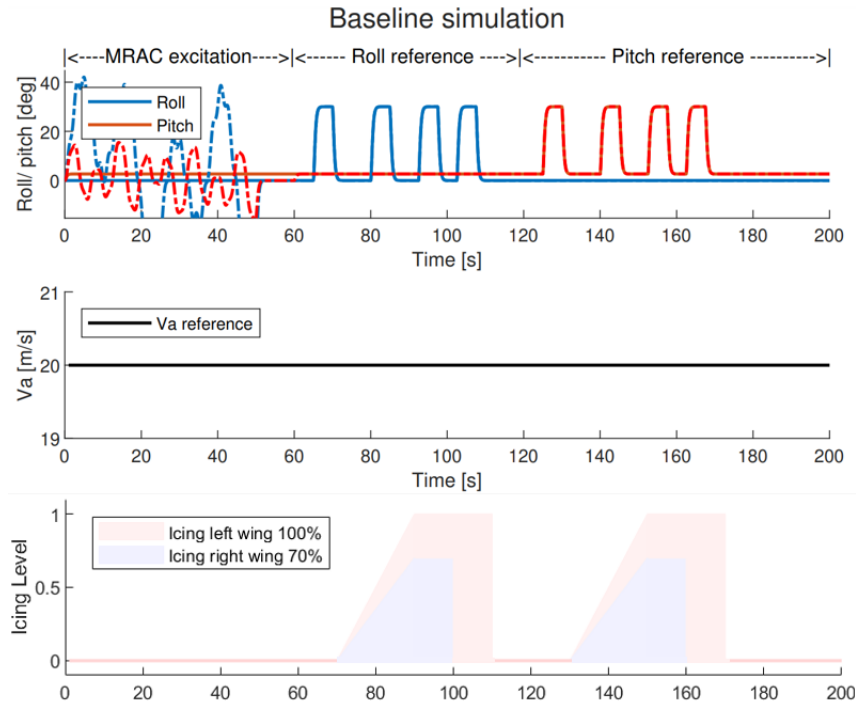


Figure 4.7: Roll, pitch, airspeed and icing level references in the baseline simulation.

#### 4.4.3 Reduced airspeed and wind conditions

Based on the baseline simulation described in the previous section, two more simulations are performed to test the performance of the controllers with additional disturbances: reduced airspeed and severe wind conditions. The references for roll and pitch, as well as the icing level, are the same as in the previous simulation, and illustrated in Figure 4.7. In the reduced airspeed simulation, the airspeed is set to  $V_a = 20 \text{ m/s}$ . When it comes to the simulation with severe wind conditions, the airspeed reference is set back to  $V_a = 17 \text{ m/s}$ , the wind static velocity component is set to  $V_{wind} = 10 \text{ m/s}$ , and severe wind gusts are turned on. To simulate severe gusts, the wind gusts chosen correspond to a static velocity of  $V_{wind} = 23 \text{ m/s}$ . For this simulation, the wind direction was set to  $\psi_w = 180$  degrees.

#### 4.4.4 Guidance simulations

Various guidance simulations were run in order to test the path-following performance of the different controllers. The path was chosen as a set of waypoints, as illustrated in Figure 4.8. The roll and pitch references to be tracked were given by the guidance controller described in Section 4.1.3. In addition, the guidance simulations were run under the same severe wind conditions described in the previous section. It was observed that the performance varied significantly based on

the wind direction angle,  $\psi_w$ , as well as on the simulated icing state of the wings. Different scenarios have therefore been tested. During these simulations, the icing state of the wings remained the same during the entire duration of the simulation, in order to see whether poor performance could be attributed to the wind conditions or to the icing conditions. A total of six simulations were run, all with a static wind of  $V_{wind} = 10 \text{ m/s}$  together with severe gusts, as done in the previous section. Three simulations were run with 100 % icing on both wings, each with a wind direction of  $\psi_w = 90^\circ$ ,  $\psi_w = 180^\circ$  or  $\psi_w = 270^\circ$ , as they were found to give different performances, and, in some cases, severe deviations from the path. The next three simulations were run with 100 % icing asymmetry instead, keeping one wing fully iced while assuming the other one is fully clean, and with each of the previously mentioned wind directions.

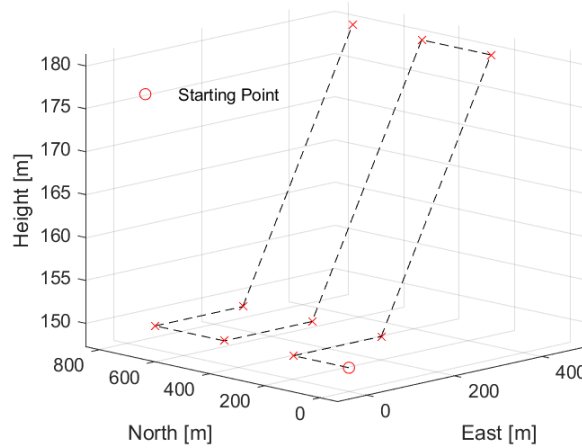


Figure 4.8: Path to be followed.

#### 4.4.5 Additional simulations

This section goes through the additional simulations performed to test the NMPC controllers specifically, and how they perform with possible limitations taken into account. The two NMPC variants explored are an NMPC which does not include the effects of icing in its model and an NMPC which does. The NMPC without icing relies solely on a disturbance observer, described in Section 3.0.2, to handle the icing effects, by considering icing as any other unknown disturbance. To illustrate why the disturbance observer is necessary, and how it combats the effects of icing, an additional baseline simulation is performed, as explained in Section 4.4.2, using the NMPC without icing in its model and having the disturbance observer turned off.

The disturbance observer removes the offset error between the reference and the NMPC response, where the error is found as the difference between the state of

the UAV and the state predicted by the NMPC. While simulating the response in the simulator, the state of the UAV is always known. In practice, the state would have to be measured or estimated. While airspeed  $V_a$  and the angular velocity can be measured, the UAV is not equipped with sensors that can measure the angle of attack  $\alpha$  or sideslip angle  $\beta$ , as these sensors are often too expensive, large and heavy. Instead,  $\alpha$  and  $\beta$  can be estimated using the standard sensor suite of a UAV, composed of a GNSS (Global Navigation Satellite System), IMU (Inertial Measurement Unit), and pitot-static tube, and kinematic relationships together with a Kalman filter [31][32]. Research on this topic finds that the angle of attack can be estimated with an RMSE of 0.56 degrees [32], while another article [33] finds an angle of attack and sideslip angle estimation accuracy of 1.2-1.3 degrees. Implementing these methods is outside the scope of this thesis, but these results are taken into account. In order to ensure that the disturbance observer would manage to perform similarly in practice, additional simulations are performed, in which the angle of attack and sideslip angles include an error of +1.5 degrees first, and -1.5 degrees next, simulating an estimation error. In the case of the NMPC with icing in its model, this was tested on a baseline simulation. When it comes to the NMPC without icing, it was assumed to be affected more, so the reduced airspeed simulation was tested with the estimation error in order to push the controller.

Additionally, the scenario in which the NMPC with icing can only take binary information about the icing level, either 0 for clean or 1 for iced, as explained in Section 4.2.4, is also tested with the baseline simulation.

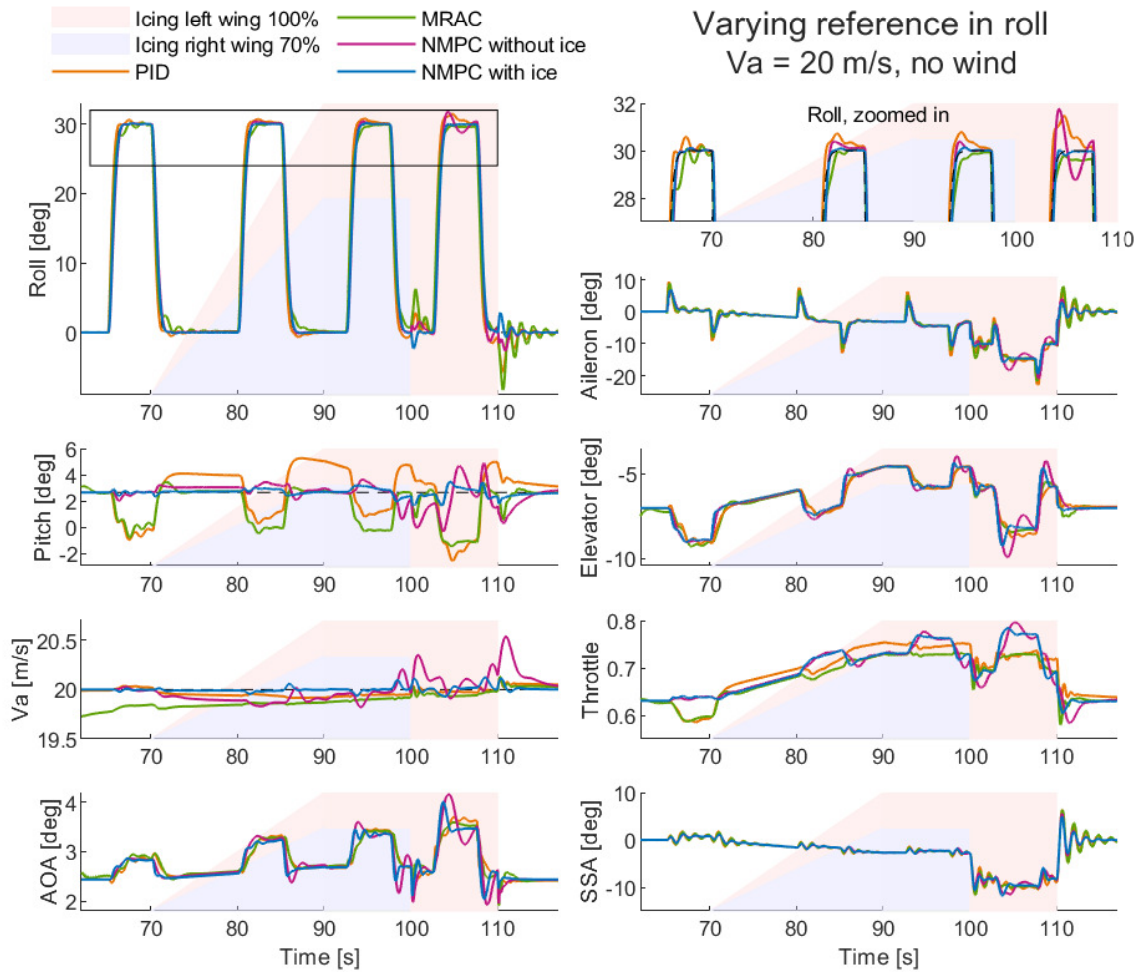
## Chapter 5

# 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, where the airspeed reference is kept at an optimal  $20\text{ m/s}$ , and no wind disturbances are present. The response after the first half of the simulation is shown in Figure 5.1, with  $t \in (60, 120)$ . The reference for roll oscillates between  $0^\circ$  and  $30^\circ$ , while keeping the pitch angle at  $2.6595^\circ$  and varying the icing level on the wings. During this time period, both the PID and MRAC have a faster response to the changes in the roll angle, resulting in a smaller roll-tracking error, as seen in Figure 5.1 and 5.3, but they react more aggressively to the sudden changes in icing level, at  $t = 100$  and  $t = 110$ , when it is simulated that the UAV loses all the ice on the right and left wing, respectively. It can also be seen that the pitch response of the PID and MRAC varies more with the changes in roll angle than the two NMPCs, and this increases during the ice asymmetry interval at  $t \in (100, 110)$ , especially with the PID controller. During this interval, the roll, pitch and airspeed tracking performance of the NMPC controller that does not include icing in its model also worsens visibly, where the most significant errors can be seen in pitch and airspeed, with a maximum airspeed error of  $e_{v_a} = 0.52\text{ m/s}$ . As this NMPC is tuned to be less aggressive, the aileron, elevator and angle of attack responses also vary more during the asymmetry interval, where the highest angle of attack of  $\alpha = 4.13^\circ$  is reached. Including icing in the mathematical model of the NMPC reduced these issues significantly and enabled a more aggressive tuning, giving a better and less affected by asymmetry, pitch and airspeed tracking performance.



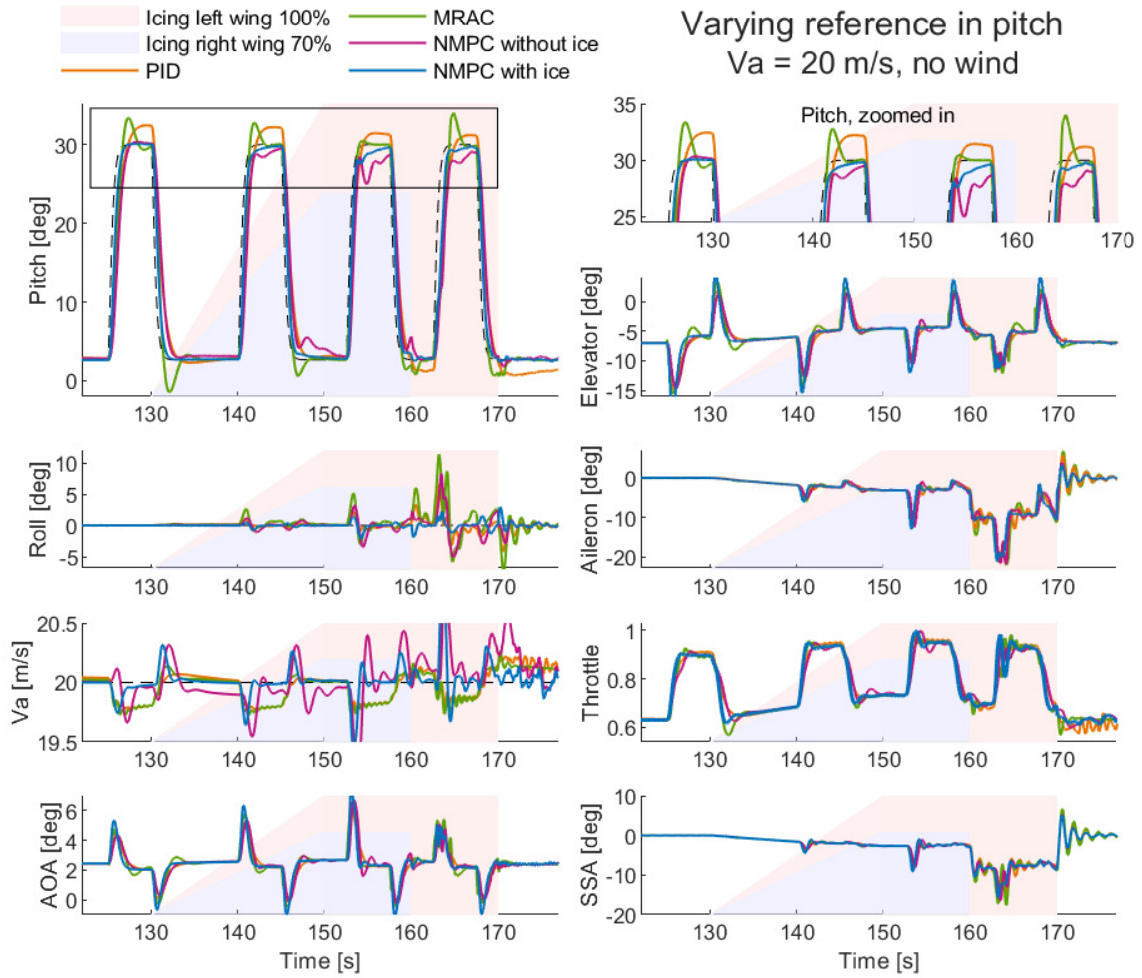
**Figure 5.1:** Baseline simulation for  $t \in (60, 120)$ , with a varying reference in roll from  $\phi_{cmd} = 0^\circ$  to  $\phi_{cmd} = 30^\circ$ , constant  $\theta_{cmd} = 2.659^\circ$ , constant  $V_{a,ref} = 20 \text{ m/s}$ , and no wind disturbances.

The roll simulation finishes at  $t = 120 \text{ s}$ , and in Figure 5.3 it can be seen that up until this point, the two NMPCs have a larger error in roll tracking performance, as they are slower, but a better pitch response than the PID and MRAC controllers. The airspeed response is more varied, with the NMPC without icing being better than the MRAC but worse than the PID. Generally, the NMPC that includes icing in its model always performs better than the NMPC that does not, and it also surpasses both the PID and MRAC controllers in terms of pitch and airspeed tracking performance.

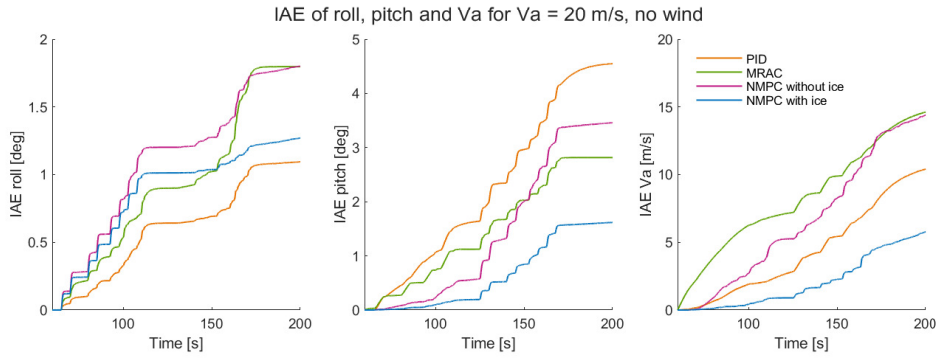
Figure 5.2 shows the response of the baseline simulation between  $t \in (120, 180)$ , where the roll reference is kept at  $0^\circ$  while the pitch oscillates between  $2.6585^\circ$  and  $30^\circ$ , and the icing level on the wings varies. In this case, the airspeed tracking performance of both NMPCs is more affected by the changes in pitch than they



were by the previous changes in roll, with the largest errors during the iced and asymmetric time intervals, as well as right after the ice on the left wing "falls", at  $t = 170$  s. The error is smaller for the NMPC that includes icing in its model, which quickly goes back to the airspeed reference. The PID and MRAC airspeed tracking performance is also more affected in this case, and the errors persist longer than with the NMPC controllers. The iced and asymmetry intervals also induce some oscillations in airspeed with these two controllers. In this figure as well as in Figure 5.3 it can be seen that the NMPC with icing in its model has a significantly better roll performance, while the most affected is the MRAC controller, followed by the NMPC without icing. The NMPC with icing has also the best pitch-tracking performance, while the other three controllers have similar responses, with the exception of a drop in performance by the NMPC without icing during the iced state. During the iced time interval, the highest angle of attack is achieved by the NMPC with icing, with  $\alpha_{max} = 7.57^\circ$ . When it comes to elevator, aileron and throttle it can be seen that the response of the two NMPC controllers generally oscillates less.



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



**Figure 5.3:** IAE of roll, pitch and airspeed throughout the baseline simulation, with a constant  $V_{a,ref} = 20 \text{ m/s}$  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.

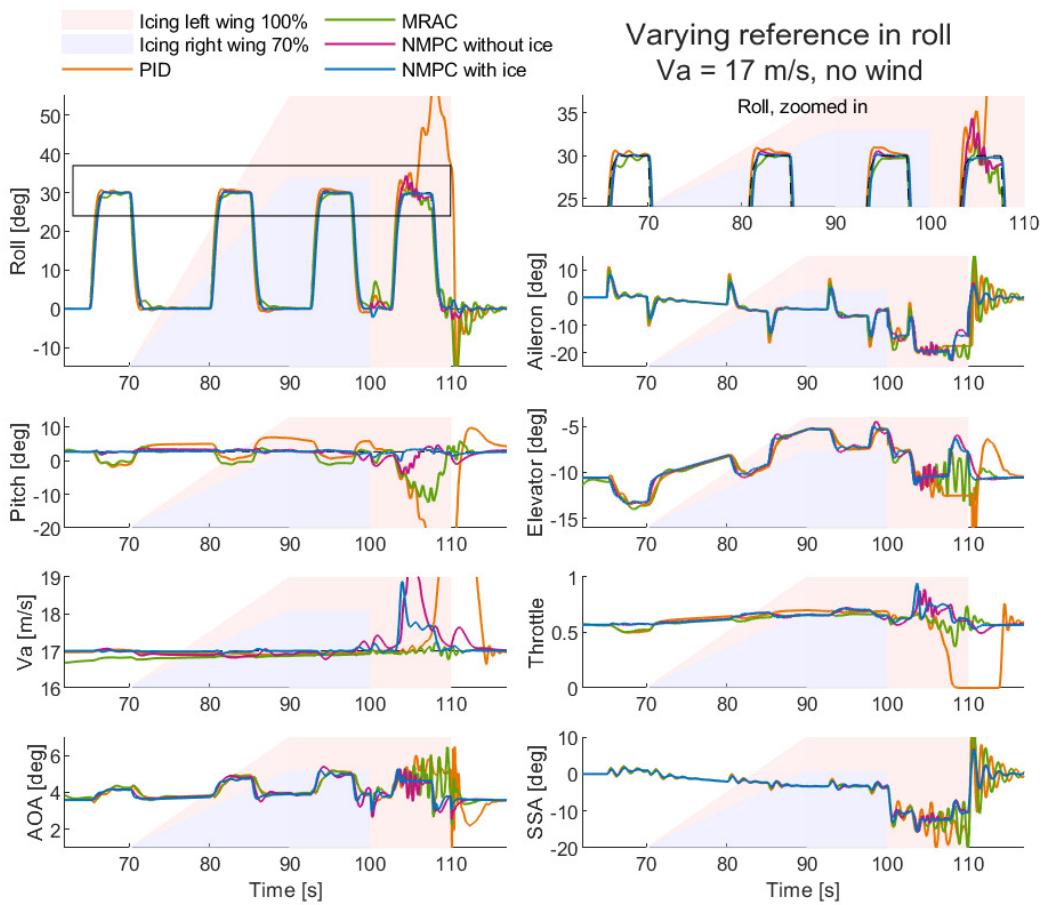
The results of the baseline simulation can be summarized by Figure 5.3. The best roll tracking performance was achieved by the PID controller, with the NMPC with icing in its model coming second due to its good performance in the second part of the simulation, when it was the least affected by the changes in pitch reference, with the MRAC controller being the most affected. The NMPC with icing is also the best at tracking pitch and airspeed, while the NMPC without icing has generally the worse performance, only surpassed by the PID's pitch and the MRAC's airspeed tracking performance.

## 5.2 Reduced airspeed simulation

The reduced airspeed simulation is run as described in Section 4.4.3, with the airspeed reference set to  $V_{a,ref} = 17 \text{ m/s}$  and without any wind disturbances. As in the baseline simulation, in the first half of the simulation ( $t \in (60, 120)$ ) the reference in roll changes while pitch and airspeed remain constant, this is shown in Figure 5.4. When the airspeed is reduced, the effect of icing asymmetry on the wings is very apparent in the case of the PID controller, which can barely handle the roll maneuver, causing a spike to  $\phi_{max} = 60.65^\circ$ , a drop in pitch to  $\theta_{min} = -53.1^\circ$  and increase in airspeed to  $V_{a,max} = 24.69 \text{ m/s}$ , as throttle saturates to 0. This is solely caused by the asymmetry, since the PID handles the iced state well. The asymmetry interval also affects the MRAC more apparently than the two NMPC controllers, as seen by a drop in pitch, and significant oscillations in elevator, angle of attack and sideslip angle. The highest angles of attack are achieved by the PID and MRAC controllers, at  $\alpha_{max} = 6.49^\circ$  and  $\alpha_{max} = 6.22^\circ$ , respectively. In addition, both the MRAC and PID are quite sensitive to the sudden loss of ice on the left wing at  $t = 110 \text{ s}$ , causing oscillations in roll, aileron and sideslip angle. When it comes to the NMPC controllers, the NMPC which includes icing in the model is, as expected, less affected by the asymmetry, being able to

follow the roll and pitch references well. The airspeed tracking performance is however more impacted by the asymmetry than with the MRAC controller, but less than with the NMPC without icing in its model.

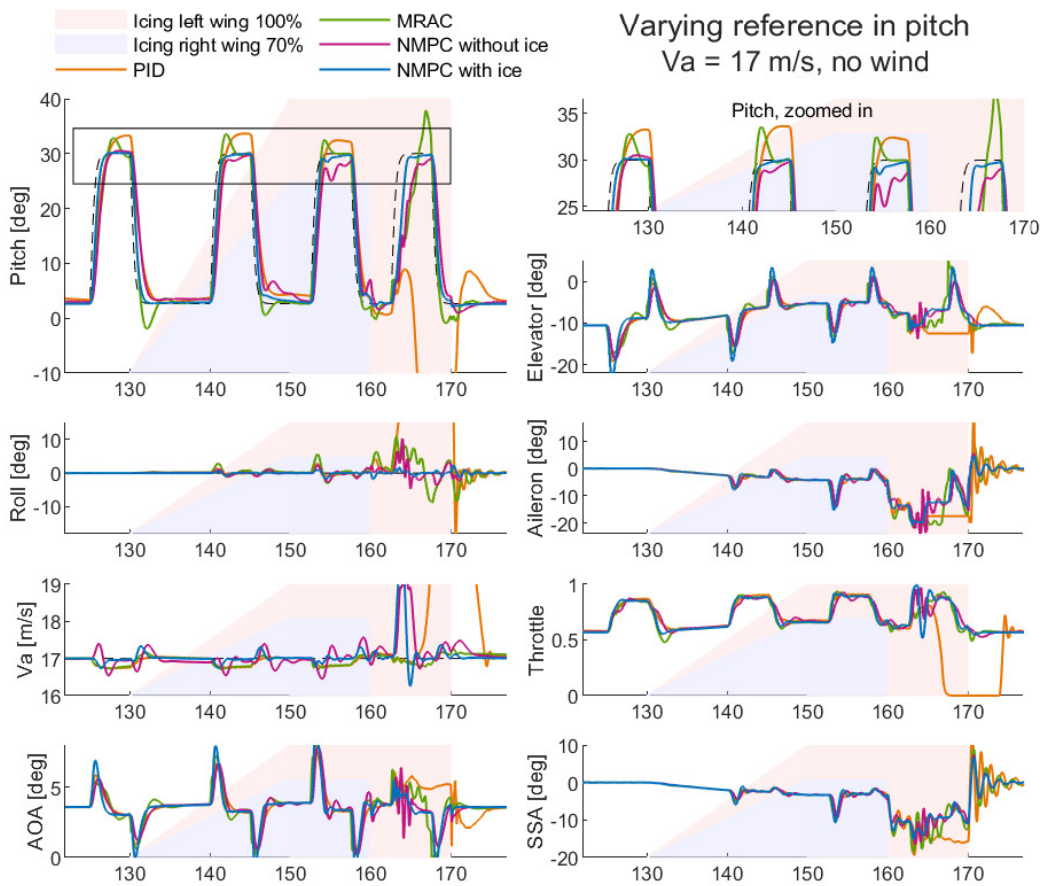
Up until this point, the roll, pitch and airspeed performances can also be seen more objectively in Figure 5.6, in the interval  $t \in (60, 120)$ . Here too it can be seen that the PID's roll, pitch and airspeed performances deteriorated drastically after the severe asymmetric case. Asymmetry did not affect the two NMPC's roll and pitch tracking performance, but it did affect the airspeed. The opposite is true in the MRAC controller's case, where the roll and pitch tracking performances were affected, but the airspeed was not. Other than in the asymmetric time interval, the reduced airspeed does not impact the performance of the controllers significantly.



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

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 PID controller's inability to handle asymmetry

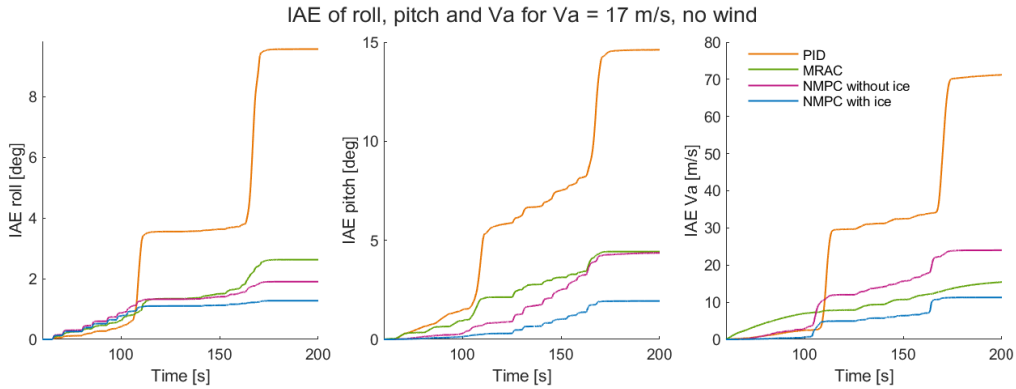
and reduced airspeed simultaneously persists here as well, with severe spikes in pitch ( $\theta_{min} = -62.63^\circ$ ), airspeed ( $V_{a_{max}} = 25.1 \text{ m/s}$ ) and roll ( $\phi_{min} = -79.28^\circ$ ). The PID's throttle saturates in this case too. The asymmetry interval also affects the MRAC controller's pitch and roll-tracking performance more than the clean and iced intervals, as well as the NMPC controllers' airspeed performance, with a maximum airspeed of  $V_{a_{max}} = 20.26 \text{ m/s}$  by the NMPC with icing in its model. During the asymmetry interval, more oscillations can however be observed in the NMPC without icing's overall response. These issues are fixed by including icing and asymmetry in the NMPC model, as seen by the response of the NMPC with icing. This controller however also has the highest angle of attack, at  $\alpha_{max} = 8.6^\circ$ , during the iced interval. The PID and MRAC controllers have their maximum at  $\alpha_{max} = 7.59^\circ$  and  $\alpha_{max} = 8.19^\circ$ , respectively.



**Figure 5.5:** Reduced airspeed simulation for  $t \in (60, 120)$ , with a varying reference in roll from  $\phi_{cmd} = 0^\circ$  to  $\phi_{cmd} = 30^\circ$ , constant  $\theta_{cmd} = 2.659^\circ$ , constant  $V_{a,ref} = 17 \text{ m/s}$ , and no wind disturbances.

The results of the reduced airspeed simulation are summarized in Figure 5.6.

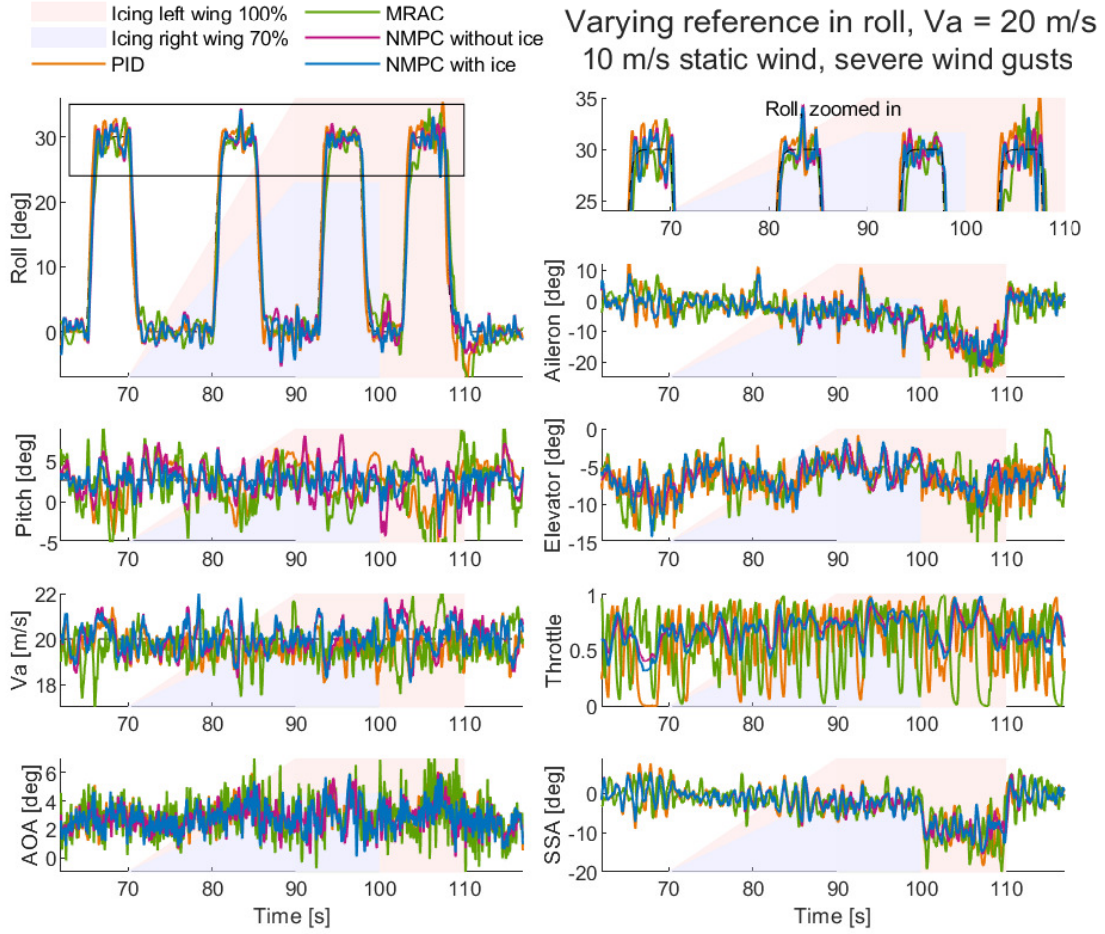
It can be seen that the PID performance is overall severely impaired compared to the baseline simulation shown in Figure 5.3, and to the other two controllers, whose performance is not affected to the same degree. In this simulation, both NMPC controllers become the best at tracking roll and pitch, with the NMPC with icing in its model also being the best at tracking airspeed. In this simulation as well it can be seen that the NMPC's performance improves greatly when the effects of icing and asymmetry become a part of its model.



**Figure 5.6:** IAE of roll, pitch and airspeed throughout the reduced airspeed simulation, with a constant  $V_{a,ref} = 17 \text{ m/s}$  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.

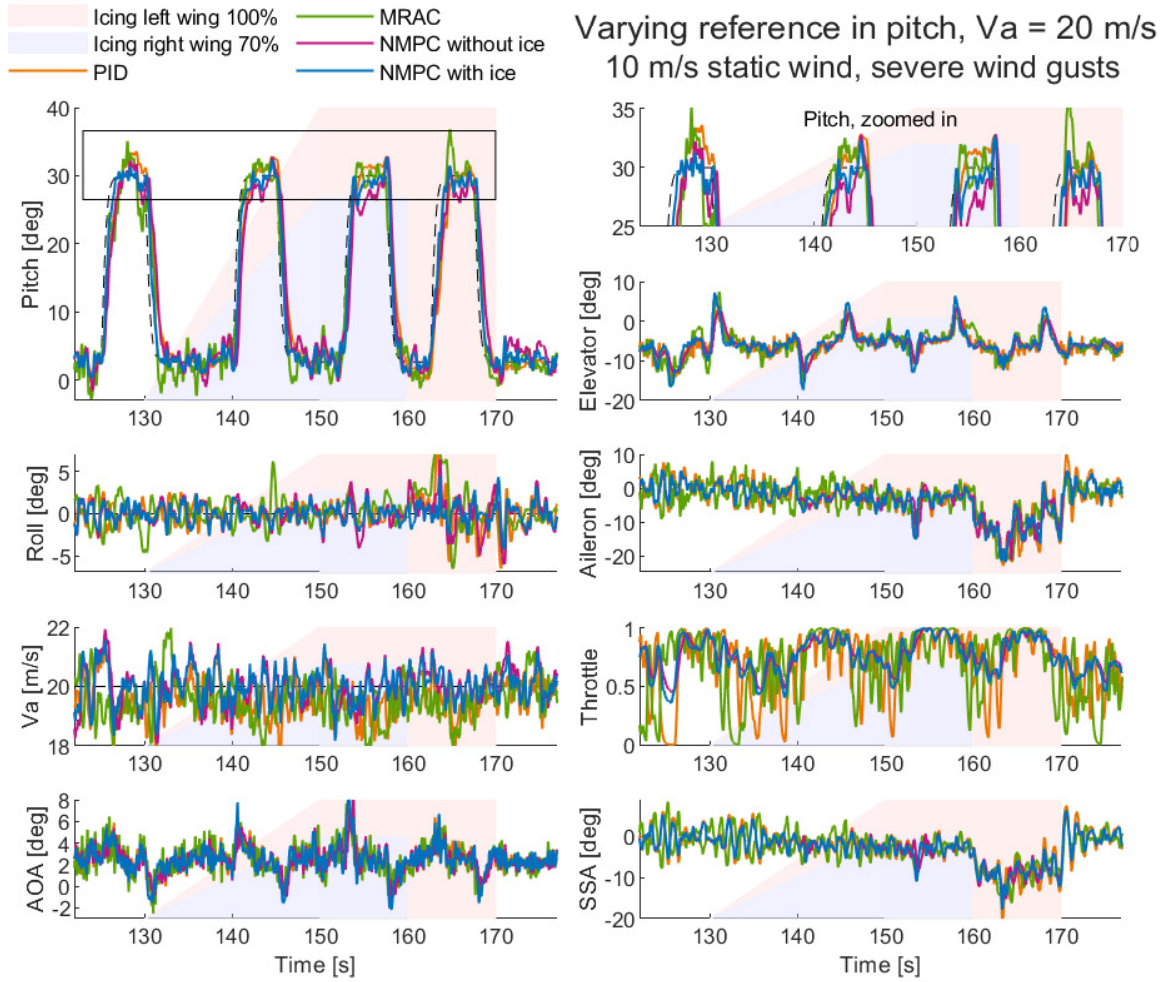
### 5.3 Wind disturbance simulation

The wind disturbance simulation is run as described in Section 4.4.3, by keeping the airspeed constant at the optimal  $V_a = 20 \text{ m/s}$ , while adding wind disturbances, with a static wind of  $V_{wind} = 10 \text{ m/s}$  together with severe wind gusts. The response of the first half of the simulation, in the interval  $t \in (60, 120) \text{ s}$ , is shown in Figure 5.7. The roll tracking performance is difficult to judge from Figure 5.7, but Figure 5.9 shows that the PID is the least affected, while the other three controllers show similar performance. When it comes to pitch tracking, both figures show that the two NMPC controllers follow the reference better. During this interval, the MRAC controller is the one which struggles the most with roll, pitch and airspeed tracking in the case of severe wind conditions. The most notable difference between the two NMPCs and the PID and MRAC is the throttle response, which oscillates significantly more with the PID and MRAC controllers. The overall aileron, elevator and angle of attack responses also seem to be less "noisy" with the NMPC controllers than with the PID and MRAC. In this simulation, the highest angles of attack are reached by the MRAC and PID controllers, at  $\alpha_{max} = 7.31^\circ$  and  $\alpha_{max} = 6.34^\circ$ , respectively.



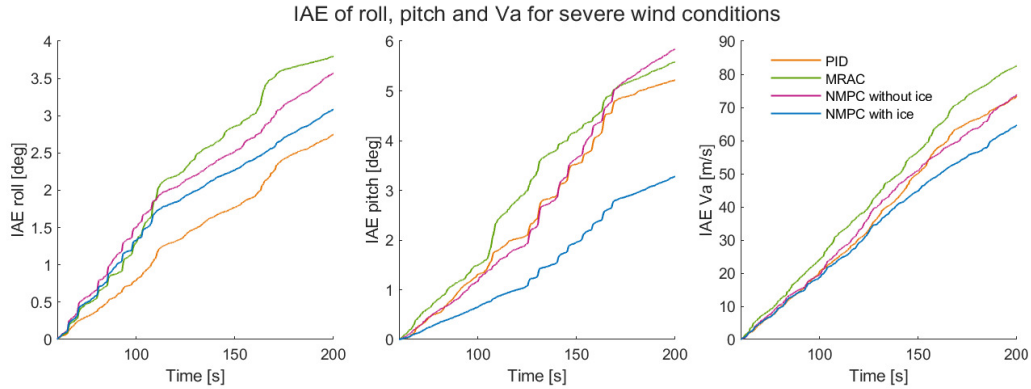
**Figure 5.7:** Wind conditions simulation for  $t \in (60, 120)$ , with a varying reference in roll from  $\phi_{ref} = 0^\circ$  to  $\phi_{ref} = 30^\circ$ , constant  $\theta_{ref} = 2.659^\circ$ , constant  $V_{a,ref} = 20 \text{ m/s}$ , and  $V_{wind} = 23 \text{ m/s}$ .

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. The pitch response seems similar for all controllers, but Figure 5.9 shows that the NMPC without icing in its model has a notable drop in performance during this time interval. However, the performance improves significantly if icing and asymmetry are added to the model of the NMPC. The NMPC without icing in its model also struggles more to keep the roll reference during the icing asymmetry interval. This is also the case for the MRAC and PID controllers. It can be seen that the two NMPC controllers are better at tracking airspeed, and their throttle and aileron responses oscillate significantly less than the PID and MRAC controllers. The angle of attack performance is similar, with the highest angles given by  $\alpha_{max} = 8.29^\circ$ ,  $\alpha_{max} = 8.19^\circ$ ,  $\alpha_{max} = 8.01^\circ$  and  $\alpha_{max} = 7.03^\circ$  by the NMPC without icing, the NMPC with icing, the MRAC and the PID, respectively. The minimum angle of attack,  $\alpha_{min} = -2.71^\circ$ , is given by the MRAC.



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





**Figure 5.9:** Baseline simulation for  $t \in (60, 120)$ , with a varying reference in roll from  $\phi_{cmd} = 0^\circ$  to  $\phi_{cmd} = 30^\circ$ , constant  $\theta_{cmd} = 2.659^\circ$ , constant  $V_{a,ref} = 20 \text{ m/s}$ , and  $V_{wind} = 23 \text{ m/s}$ . 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.

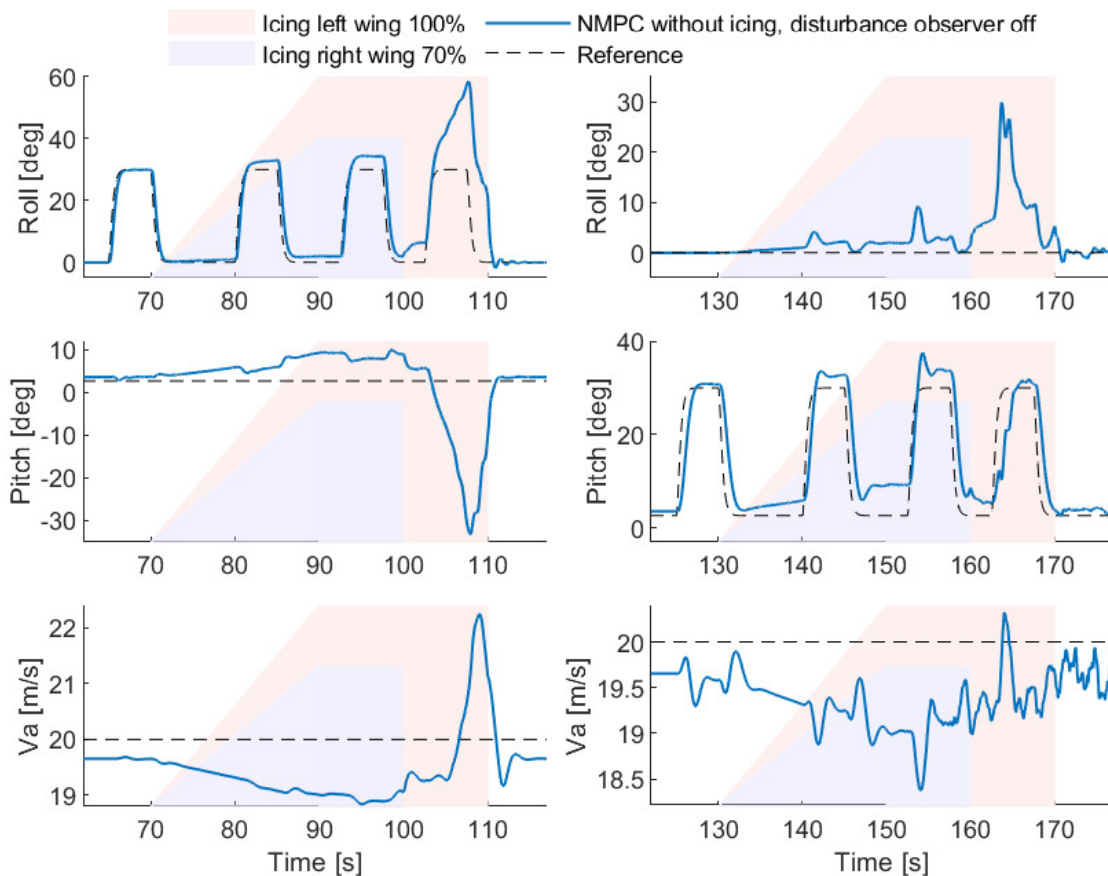
As a result of the severe wind conditions, the response of all controllers oscillates more, but the effects of the 100% asymmetric icing case do not impact the performance as severely in this case as they did in the reduced airspeed case. Looking at the IAE of roll, pitch and airspeed in Figure 5.9, and especially at the two NMPC controllers, the error increases more linearly as time passes, indicating that the severe wind conditions affect the performance of the controllers more than the changes in icing level. The MRAC controller is generally the most affected by the wind conditions, while the PID is the best at tracking roll and the NMPC with icing in its model is the best at tracking pitch and airspeed. The NMPC without icing has a similar performance to the PID and MRAC controllers, but here too it can be seen that its performance is improved when icing is included in its model.

## 5.4 Additional simulations

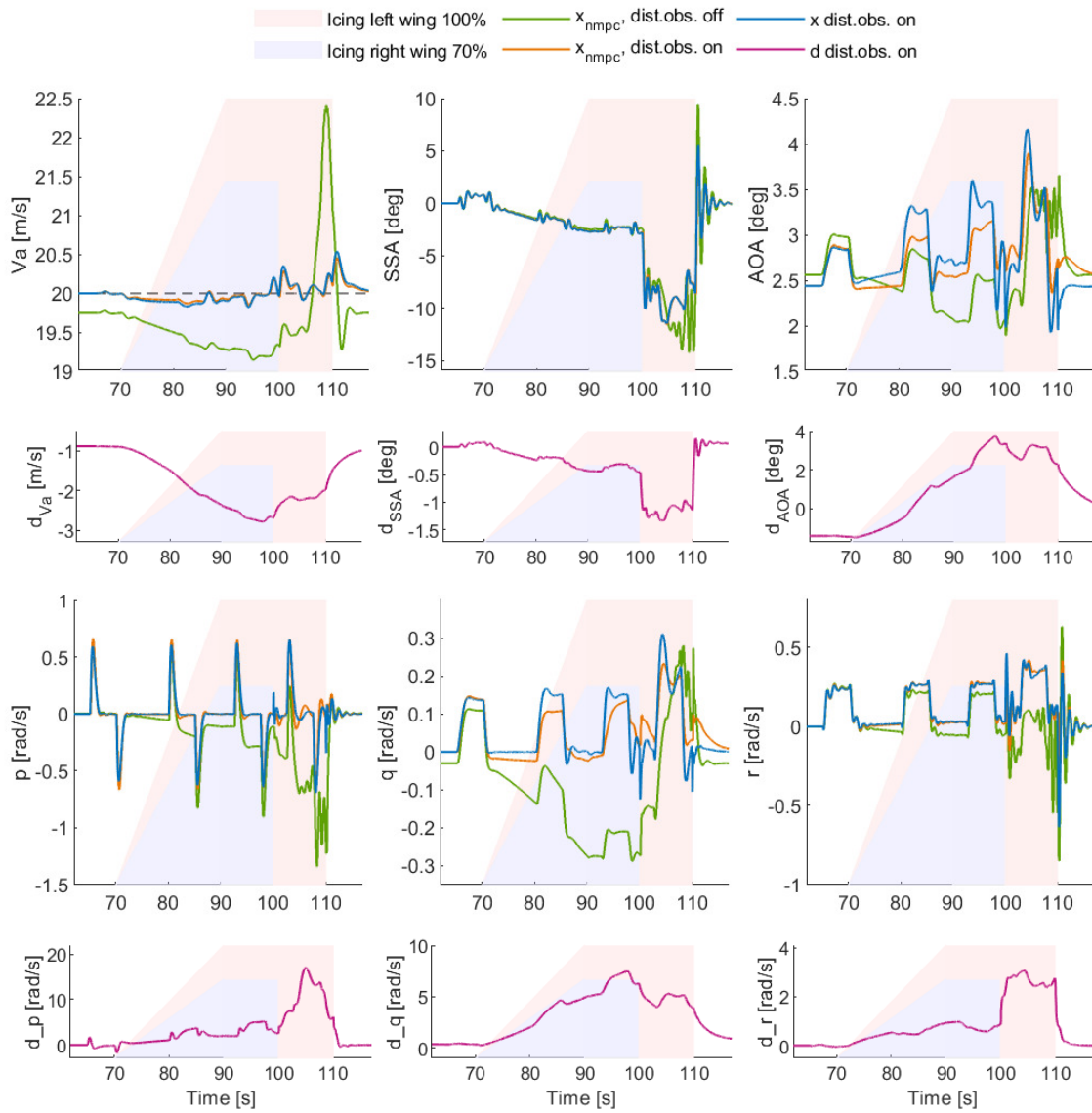
To further evaluate the performance of the two NMPC controllers, one which includes icing in its model and the other which does not, additional simulations have been performed. In the case of the NMPC without icing, it might be interesting to see what the output of the disturbance observer is, and how it manages to combat the effects of icing and icing asymmetry. For this purpose, the baseline simulation was run with the disturbance observer turned off. The roll, pitch and airspeed responses of the NMPC in this case are given in Figure 5.10.

The results show an offset error in airspeed and pitch primarily, which increases with the level of icing. The roll error also increases with the icing level, becoming very large during the 100 % icing asymmetry case, resulting in large errors in pitch and airspeed as well. Figures 5.11 and 5.12 show the NMPC predictions of the states controlled by the disturbance observer when this is turned off and on, as well as the disturbances predicted by the observer when it is on. It can be seen

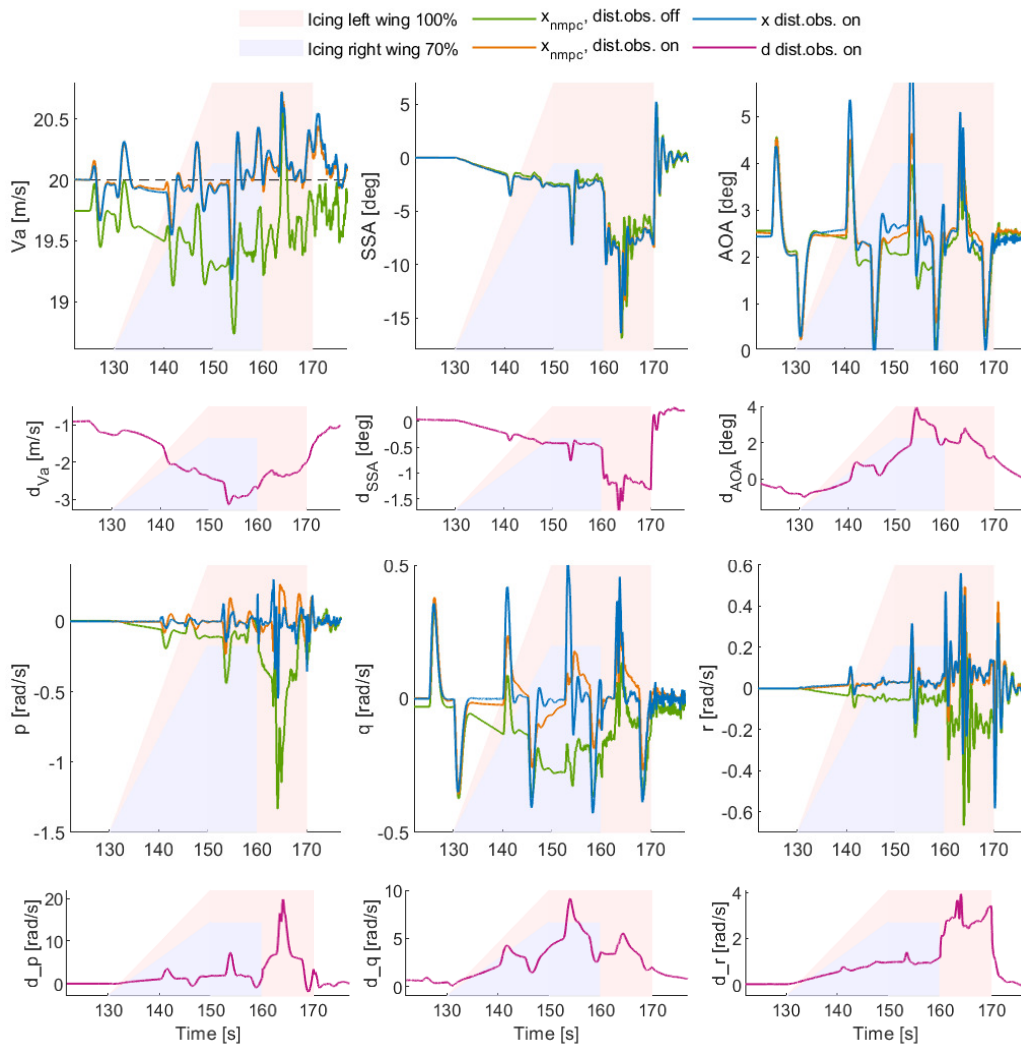
that there is an increasing error between the NMPC prediction  $\mathbf{x}_{nmpc}$  and the state  $\mathbf{x}$  as the icing level increases while the disturbance observer is off. This is expected considering the response shown in Figure 5.10. When the disturbance observer is turned on, it can be seen that the NMPC predictions are corrected, in most cases almost entirely as they overlap with the state  $\mathbf{x}$ . The disturbances predicted by the observer are also plotted, and it can be seen that they correspond to the observed improvement of the NMPC predictions: an increase or decrease of  $d$  follows the previous errors between  $\mathbf{x}_{nmpc}$  and  $\mathbf{x}$ .



**Figure 5.10:** Response of the NMPC without icing in its model when the baseline simulation is run while the disturbance observer is off.



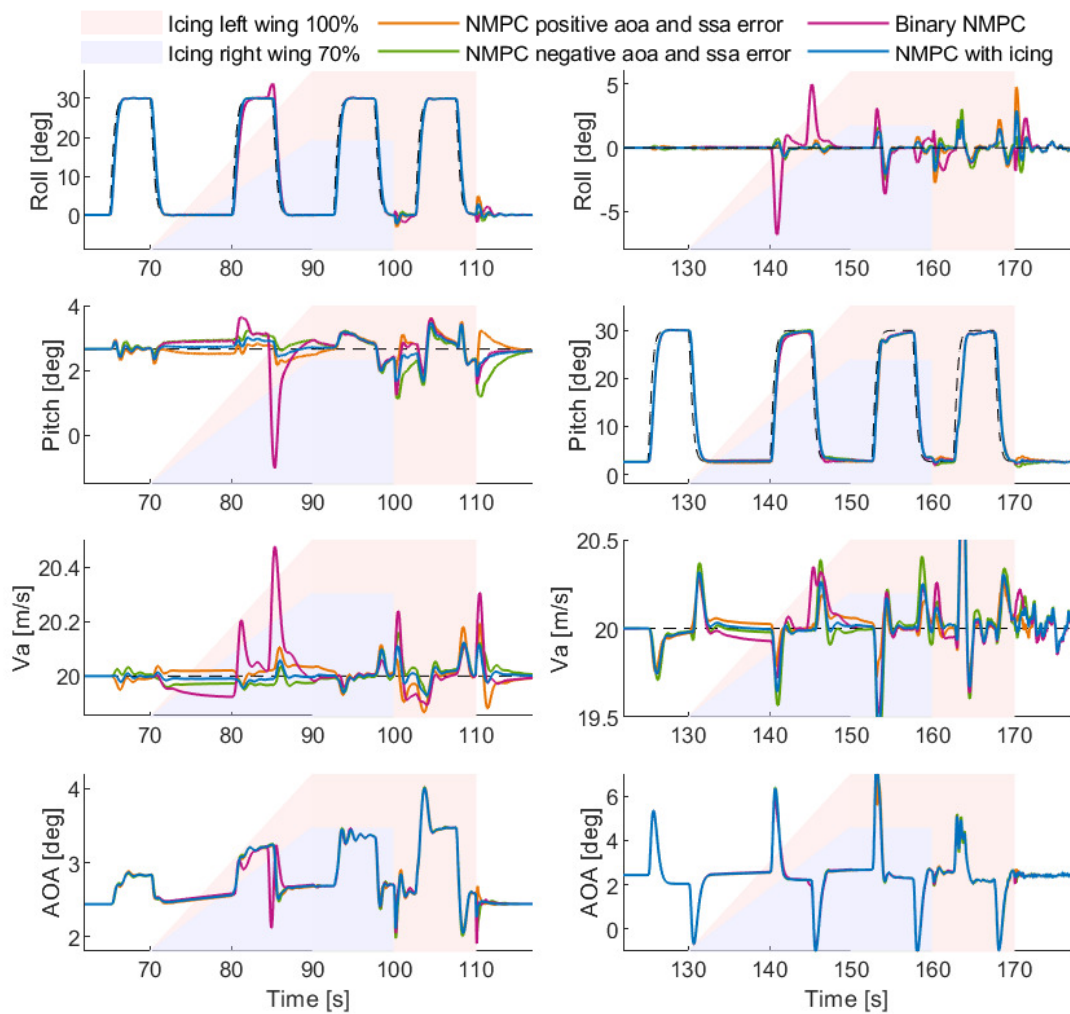
**Figure 5.11:** The first half of the baseline simulation run with the disturbance observer on and off. It can be seen how the NMPC predictions of the states controlled by the disturbance observer change with the observer being on or off. The disturbances  $d$  from Eq. Eq. (3.5)-(3.6) when the observer is on are also plotted.



**Figure 5.12:** The second half of the baseline simulation run with the disturbance observer on and off. It can be seen how the NMPC predictions of the states controlled by the disturbance observer change with the observer being on or off. The disturbances  $d$  from Eq. Eq. (3.5)-(3.6) when the observer is on are also plotted.

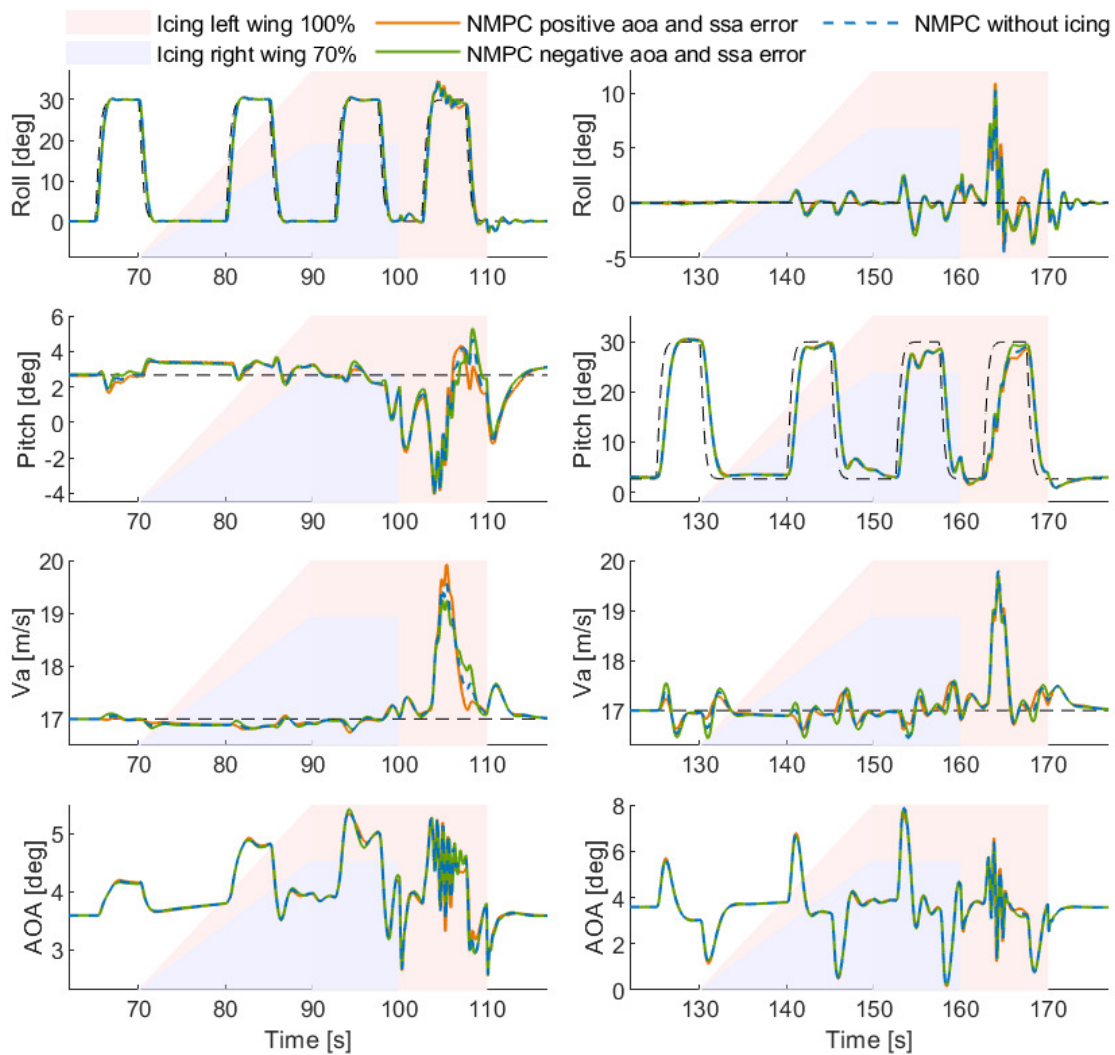
Next, possible limitations of the NMPC controllers are explored. Because the NMPC controllers are tested on the simulator, all information about the states is available. When tested in practice, the angle of attack and sideslip angles, which are corrected by the disturbance observer, cannot be measured and would need to be estimated instead. When it comes to the icing level, it can be measured, but it is safe to assume that not to the same level of precision. Therefore, as explained in Section 4.4.5, two simulations are performed to test these limitations. Figure 5.13 shows the roll, pitch, airspeed and angle of attack response of the NMPC with icing in its model with the limitations mentioned. It can be seen that the binary

NMPC has an identical response to the NMPC with access to full information about the icing level, except for the moments when the icing state of the binary NMPC flips from clean to iced as the reference in roll changes. In those cases, there is a fast, drastic change in the states, but it gets resolved quickly. When it comes to the NMPC controllers that have a positive or negative error of 1.5 degrees in both angle of attack and sideslip, it can be seen that this has a minimal impact on the performance. These controllers have almost the same response as the "perfect" NMPC with icing, except for the moments when the 100 % asymmetric icing interval starts and ends.



**Figure 5.13:** Baseline simulation response of the NMPC with icing included in its model, compared to more limited NMPC controllers with icing. Two of them include simulated estimation errors of  $\pm 1.5$  degrees in angle of attack and sideslip angle, while the binary NMPC only has binary information about the icing level, either clean (0) or iced (1).

The estimation error of angle of attack and sideslip is also tested with the NMPC without icing in its model. In this case, the airspeed simulation is performed, as it is more challenging, with a positive or negative error of 1.5 degrees in both angle of attack and sideslip. The results are given in Figure 5.14, where it can be seen in this case too that the estimation errors have a minimal effect on the performance, although they increase slightly during the asymmetry interval.



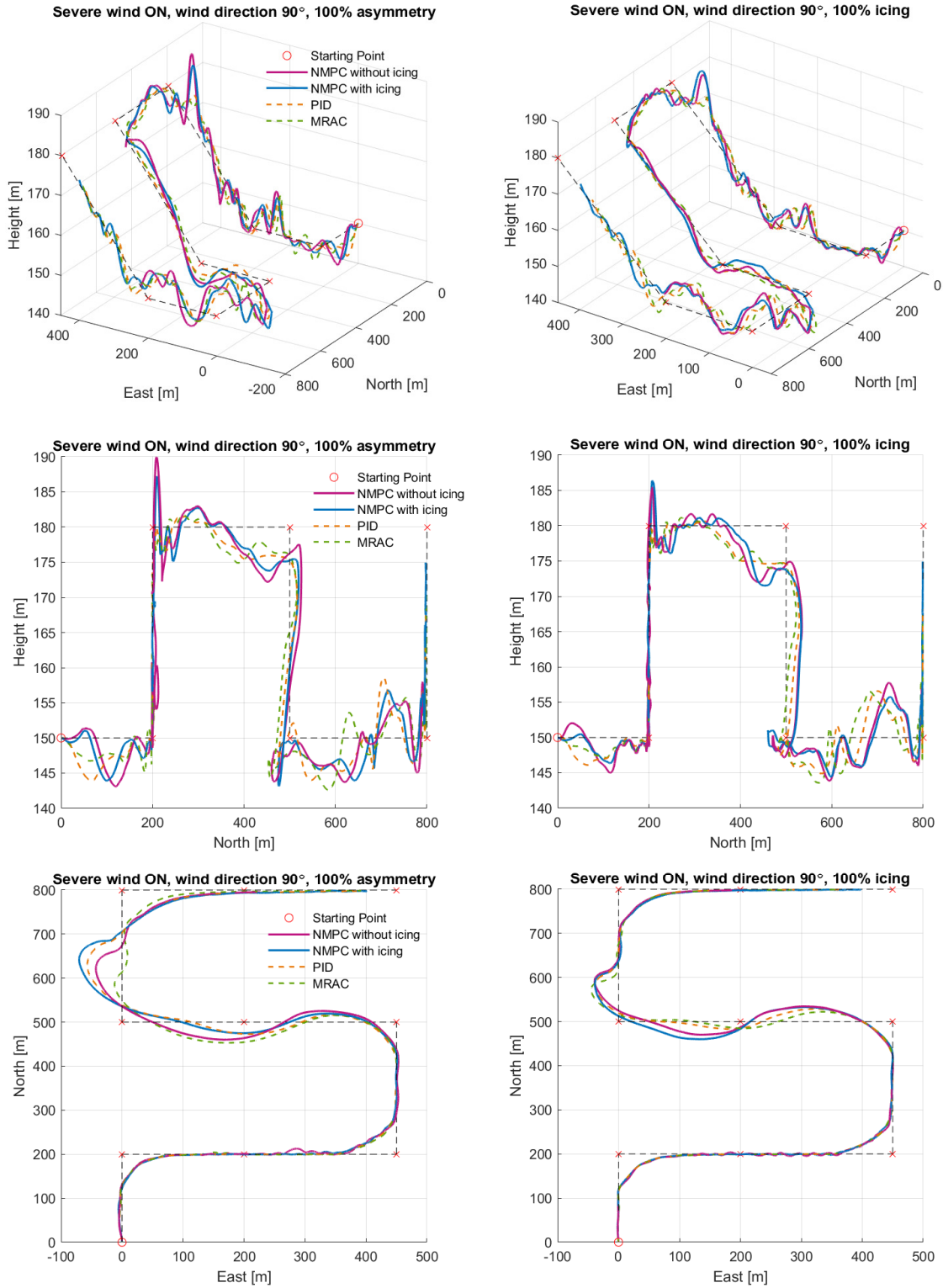
**Figure 5.14:** Reduced airspeed simulation response of the NMPC without icing included in its model, compared to two more limited NMPC controllers with simulated estimation errors of  $\pm 1.5$  degrees in angle of attack and sideslip angle.

## 5.5 Guidance simulation

In this section, the results from the guidance simulation described in section 4.4.4 are presented. The simulations are run with 10 *m/s* wind and severe wind gusts, changing the wind direction and the icing level, as the results were found to vary a lot depending on those two factors. The performance of the NMPC with icing in its model is compared to the one of the NMPC without icing, and to the PID and MRAC controllers.

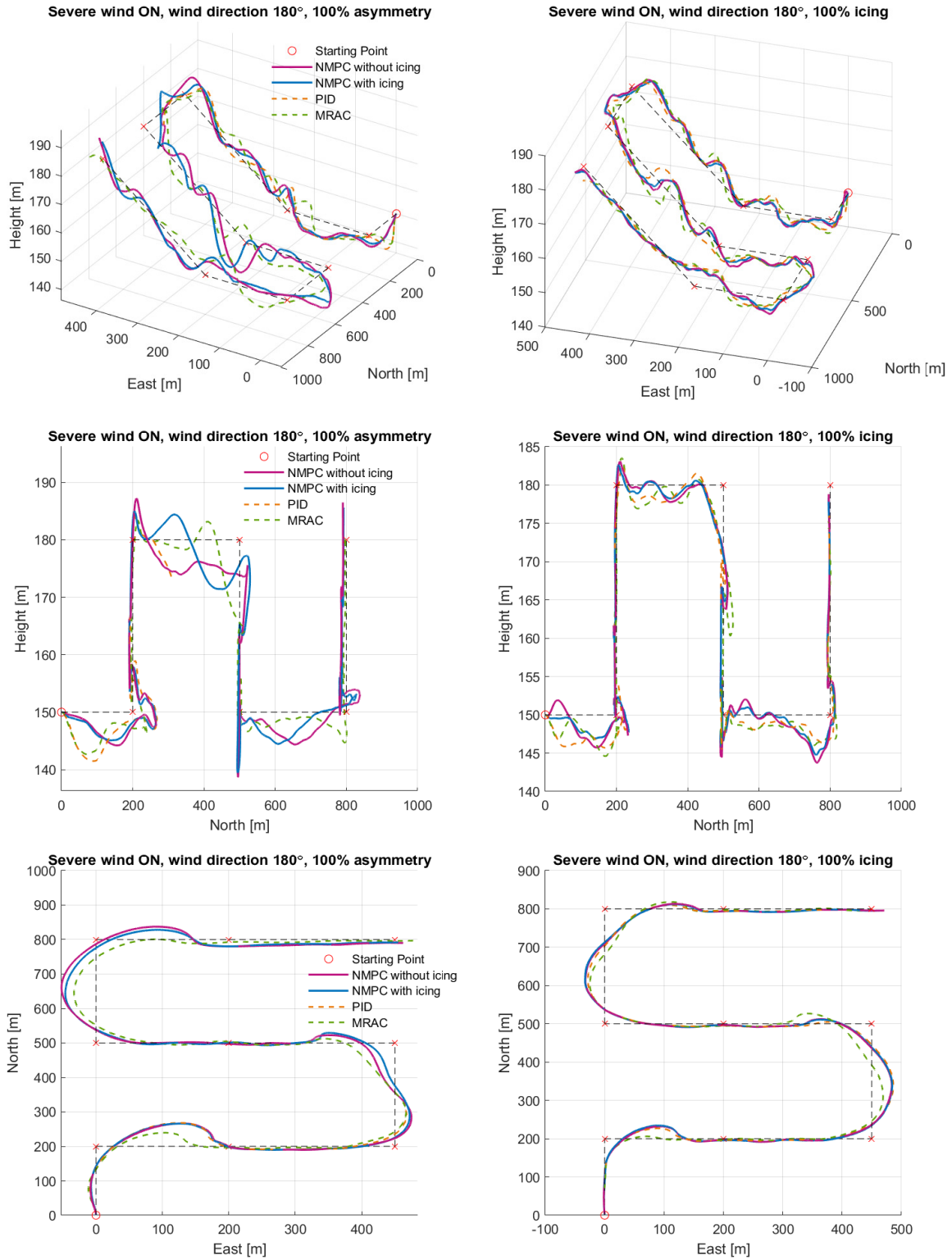
Figure 5.15 shows the performance of the four controllers when the wind direction is  $\psi_w = 90^\circ$ . The performance in the case of 100 % asymmetry throughout the simulation time is presented on the left, while the 100 % icing case is presented on the right. It can be seen that all controllers struggle more to follow the path in the case of asymmetry than when the wings are fully iced, especially when the UAV gets closer to take a turn. As observed, the asymmetry also leads to more height variations across the path. The NMPC with icing in its model is slightly better at keeping the height closer to the target than the NMPC without icing, but it is outperformed by both the PID and the MRAC.

When the wind direction is changed to  $\psi_w = 180^\circ$ , the PID controller is no longer able to keep the path during the 100 % icing asymmetry simulation, and crashes while trying to take the second turn, as seen in Figure 5.16. The performance of the two NMPC controllers is also notably impaired, as they struggle to keep the height after the second turn more than in the 100 % icing simulation on the right. In general, the controllers are also forced to take larger turns when the direction of the path changes. When it comes to the difference between the NMPC controllers, it is not very apparent which one is better in this case, given the described icing and wind conditions.

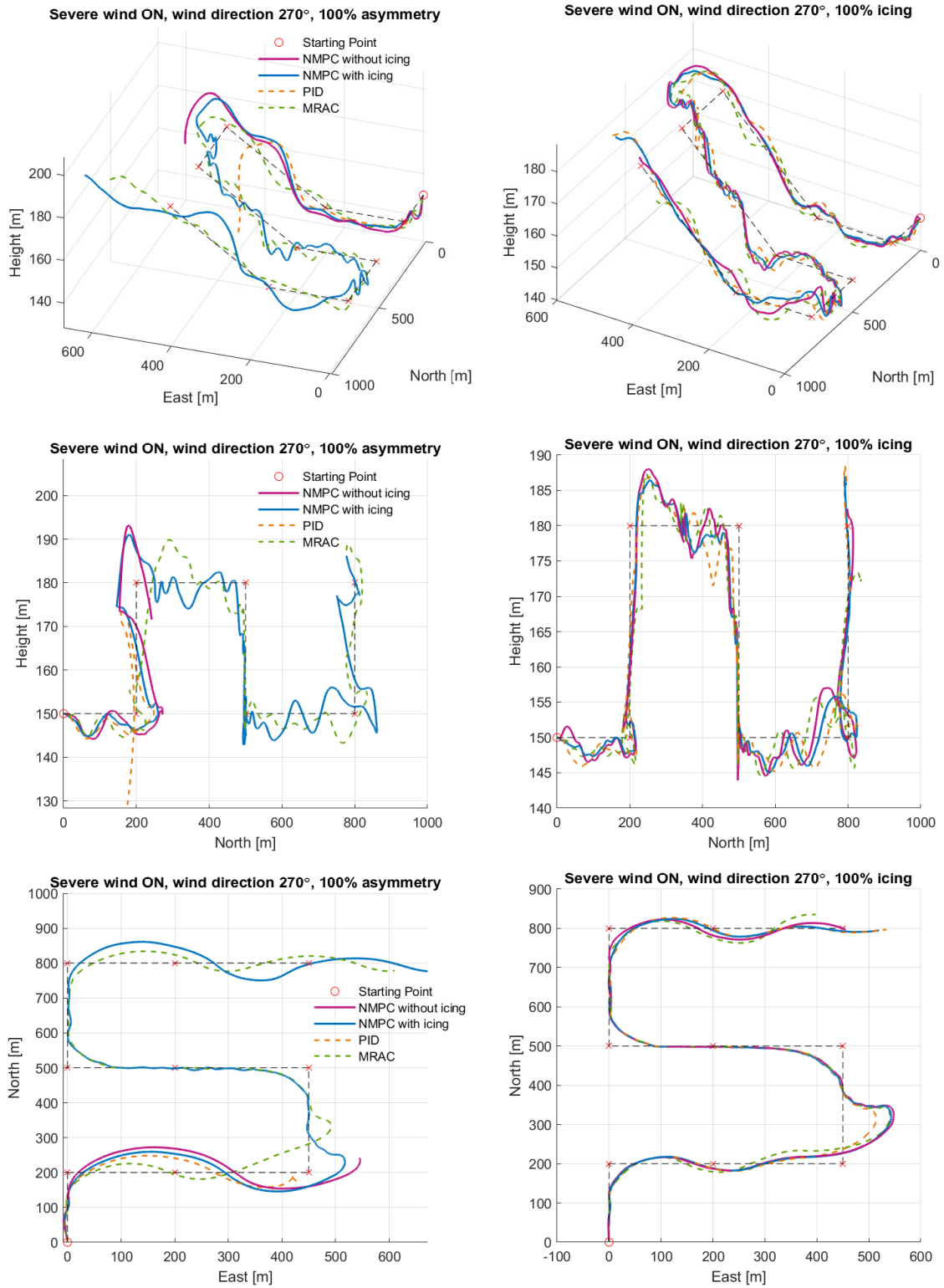


**Figure 5.15:** Guidance simulation with a static wind velocity of  $V_{wind} = 10 \text{ m/s}$ , severe wind gusts and wind direction  $\psi_w = 90^\circ$ . Run with 100 % asymmetry and 100 % icing level.





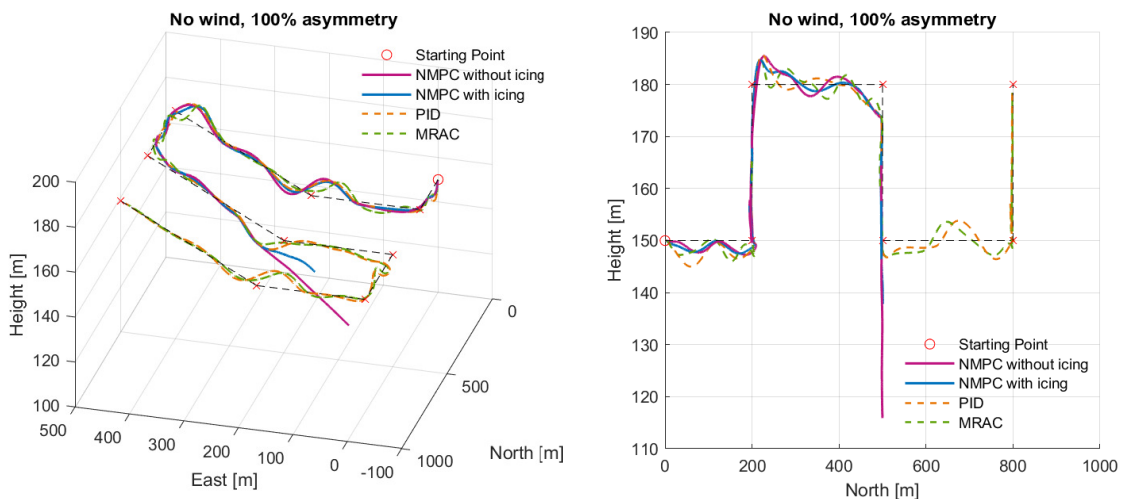
**Figure 5.16:** Guidance simulation with a static wind velocity of  $V_{wind} = 10 \text{ m/s}$ , severe wind gusts and wind direction  $\psi_w = 180^\circ$ . Run with 100 % asymmetry and 100 % icing level.



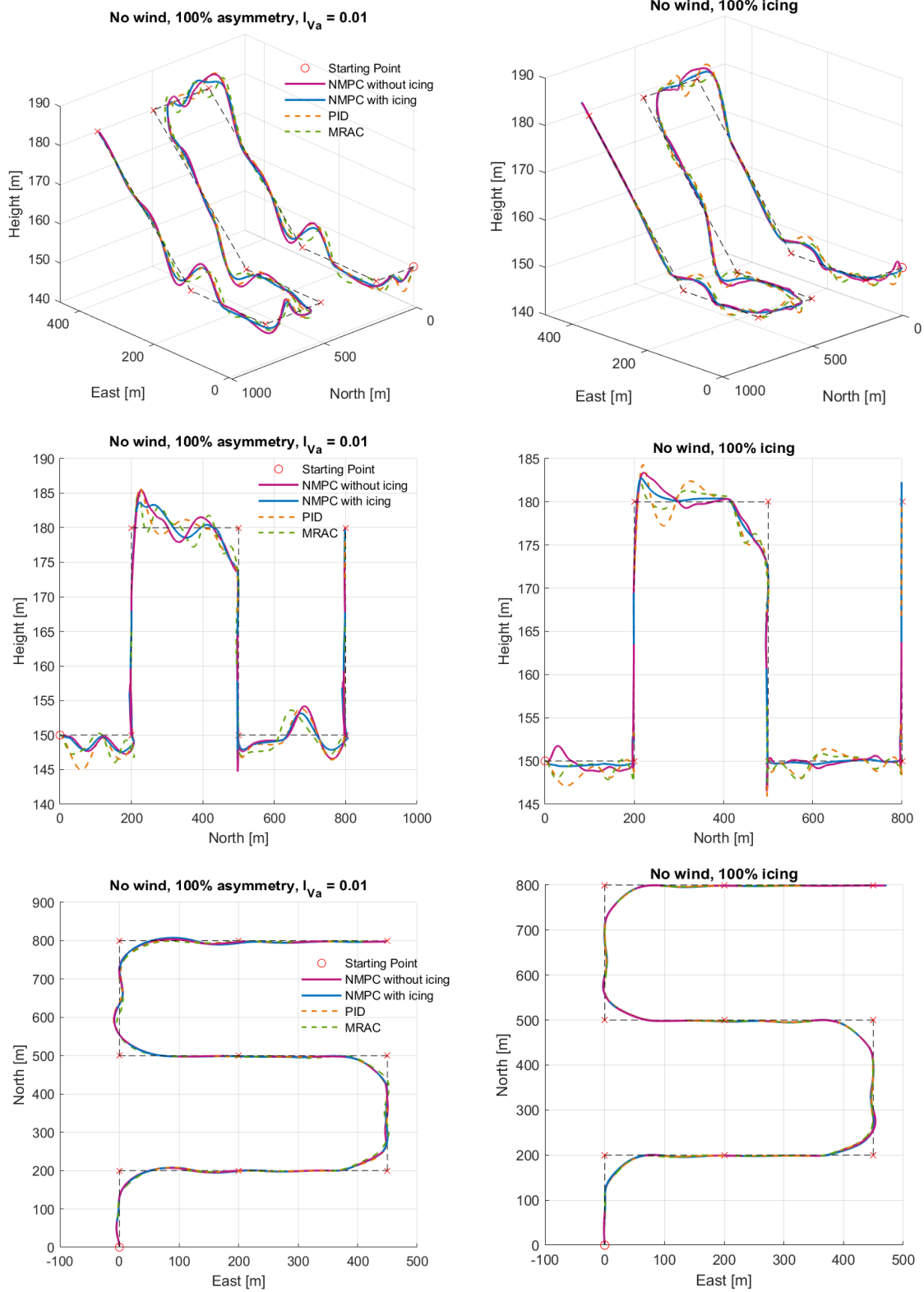
**Figure 5.17:** Guidance simulation with a static wind velocity of  $V_{wind} = 10 \text{ m/s}$ , severe wind gusts and wind direction  $\psi_w = 270^\circ$ . Run with 100 % asymmetry and 100 % icing level.

In Figure 5.17 we see how well the controllers follow the path when the wind direction is changed to  $\psi_w = 270^\circ$ . As can be seen, in the 100 % asymmetry case, both the NMPC without icing (nor asymmetry) included in its model and the PID controller crash. As the wind pushes the UAV further from the path during the swing at the third waypoint, the NMPC without icing does not manage to converge back. The PID crashes before the swing, unable to both ascend and prepare to turn left. Although the wind direction pushes the UAV towards the east when taking the second turn, the effects are more severe when the icing level on the wings is asymmetric. Including icing in the NMPC model makes it able to follow the path, but it seems more disturbed than the MRAC controller. In the case of 100 % icing, on the other hand, the performance of both NMPC controllers improves, and their response seem to oscillate less than with the MRAC. This can be seen the two times the UAV ascends to 180 meters, going towards the fourth and the last waypoint.

After running the guidance simulations in different wind conditions and directions, it was interesting to see how the different controllers perform without any wind conditions or disturbances present, other than the icing conditions. During this simulation, it was surprising to see that both NMPC controllers crashed during the part of the path that descends from 180 to 150 meters, during the 100 % asymmetry case. This is because the throttle optimal input would saturate at  $\delta_t = 0$ . By reducing the weight on the disturbance observer airspeed error,  $l_{V_a}$ , from 0.03 to 0.01, this problem is solved, as shown in Figure 5.19. However,  $l_{V_a} = 0.03$  is needed when wind disturbances are present, and due to the lack of time, a tuning that works in both cases has not been found. Using  $l_{V_a} = 0.01$ , Figure 5.19 shows that the NMPC controllers are better at keeping the height reference than the PID and MRAC controllers.



**Figure 5.18:** Guidance simulation with the wind turned off. Run with 100 % asymmetry and 100 % icing level.



**Figure 5.19:** Guidance simulation with the wind turned off, and  $l_{V_a} = 0.01$ . Run with 100 % asymmetry and 100 % icing level. In this case, the NMPC controllers no longer crash.

## Chapter 6

# Discussion

Based on the simulation results presented in Chapter 5, the performance of the NMPC controllers will be analyzed and discussed in this chapter, comparing them to the PID and MRAC controllers, as well as the challenges related to the controller designs.

### 6.1 The improved NMPC controllers

The aim of this thesis was to improve the NMPC controller used in the project thesis [1] to be more suitable in icing conditions. In this thesis, two NMPC controllers were explored: an NMPC with the effects of icing and asymmetry included in its model, and an NMPC without icing, based on the NMPC developed in the project thesis [1]. Updating the model of the latter to match the aerodynamic model by Winter [12] used in the Matlab/ Simulink simulator helped reduce the offset error in the NMPC response when the disturbance observer was turned off. The disturbance observer was then mainly used to account for the effects of icing asymmetry and other disturbances. The most notable improvement, however, was adding the effects of icing on the wings and asymmetry to the NMPC model. Having knowledge on the icing model and how it affects the aerodynamic performance of the UAV ensured an optimal input even in the case of severe asymmetry, as the NMPC with icing in its model understands how it works. The effects of icing on the propeller, however, were not added to this NMPC. This is because the effects are dependent on the propeller type, and including it in the NMPC model limits the accuracy of the NMPC to only be valid for the Skywalker X8's propeller. In addition, the model is dependent on the temperature and the liquid water content (LWC), which the UAV might not have access to. When adapting the propeller icing model developed by Müller [16], and valid for the Mejzlik propeller that he used, to the UAV used in this thesis, there were also many assumptions made. Therefore, although the model is assumed to be good enough to simulate the additional effect of icing on the propeller, given possible inaccuracies and the mentioned challenges above, the NMPC might not benefit from including it in its model.

## 6.2 Simulations and performance of the controllers

Throughout the simulations performed, it was clear that the effects of asymmetry in particular deteriorated the performance of the controllers significantly more than any other icing interval. Adding icing to the model of the NMPC was found to notably reduce the degradation in performance during the asymmetry interval, and reduce oscillations compared to the NMPC without icing. The NMPC with icing in its model had the best IAE pitch and airspeed tracking performance during the baseline, reduced airspeed and wind conditions simulations, and the second best roll performance. However, asymmetry still affected the performance of this controller to some degree, as could be seen more clearly during the guidance simulations. This could be because of differences between the model used in the simulator and the model of the NMPC, or possible implementation errors of the asymmetry model into the NMPC. The guidance simulation with 100 % asymmetry also crashes with the tuning proposed, and although better tuning was found for this case, it was not tested for the other simulations. The tuning of the NMPC controllers needs therefore to be improved. When it comes to the simulations, all disturbances were planned and simulated in the Matlab/ Simulink simulator, and had a maximum length of 200 seconds. In the real world, ice would not build and shed that quickly, leading to the question of whether the scenarios tested were realistic or too strict. When it comes to the icing conditions, the icing data found in Winter [12] and used in this model is valid for the most severe icing type, the mixed icing case. Although the UAV could encounter less severe icing conditions, the controllers had a bigger issue when severe asymmetry was simulated than with 100 % iced wings. When it comes to the 100 % asymmetry simulated, when all ice sheds from one of the wings while the other one remains fully iced, it is the worst-case scenario but it is also fairly unlikely. How long it would last if it were to happen is also unknown, so the guidance simulations in which asymmetry lasts for over 100 seconds might be too strict. However, the NMPC controller with icing in its model still performed well under almost all simulations.

## 6.3 Tuning of the controllers

The tuning of the NMPC controllers, and especially of the controller without icing in its model, was in the end more challenging than expected. As the controller did not have any information about the icing, trade-offs were needed in the tuning to ensure it worked in both of the most severe icing cases (the 100 % icing and 100 % asymmetry cases), and together with other severe disturbances. A more aggressive controller worked well under severe wind conditions, but it often did not handle reduced airspeed in the asymmetry interval. Improving the roll performance during this interval often came at the expense of pitch, and vice versa. The performance during the reduced airspeed simulation was very similar to the one during the baseline simulation, excepting the icing asymmetry interval. Therefore, choosing a less aggressive tuning because of this interval resulted

in a slower baseline response as well. In addition, different disturbance observer parameters had also a great impact on the performance, where different weights would perform better under certain disturbances or parts of the icing interval, but poor during others. For example, too little weight on pitch angular velocity  $l_q$  would not be sufficient with severe wind conditions, but too much weight would increase oscillations, as the NMPC "trusts" the disturbance observer more than its predictions. As seen in Figures 5.18 and 5.19, a too high weight on the disturbance observer airspeed value makes the NMPC controller crash in the absence of disturbances other than icing asymmetry. This questions the robustness of the NMPC, since the controller was not proven to work perfectly under all simulation cases with the tuning chosen. However, given how well it was able to perform in the rest of the simulations, especially under the asymmetry intervals, it is safe to say that it was the least affected by the effects of icing asymmetry on the wings, and that a better tuning is possible to be found.

When it comes to the PID and MRAC controllers, they were used the way they were developed by Gryte [11] and Högnadóttir [10], and further tuning was left outside the scope of this thesis, as the focus was on developing robust NMPC controllers. As some changes were made in the Simulink model, the question arises of whether the PID and MRAC are fairly tuned, and whether their performance is worsened with the updated model. This topic was explored in the author's project thesis [1], but it was found that the updated Simulink model had little impact on the performance of the PID and the MRAC controllers. This is probably because, compared to greater disturbances like reduced airspeed and wind conditions, the differences between the new and old models are not as significant.

## 6.4 Stall angle

One of the key features of the NMPC controller is that it allows for constraints to be put on the states when formulating the optimization problem, so that physical and safety limits are taken into account. This way, stalling of the aircraft can be considered by constraining the angle of attack to not go above the upper bound  $\alpha_{stall}$ . This is not the case with the PID and MRAC controllers, which currently do not employ any stalling measures, and the stall limit needs to be checked after running each simulation to verify it is not crossed. Although stalling was considered for the NMPC, the actual stall limit is not clear in the literature, with Winter suggesting that  $\alpha \approx 10^\circ$  in his work [12], while Högnadóttir writes in her thesis that it might be closer to  $\alpha \approx 4^\circ$  [10]. This could be due to the differences between the configurations used in their simulations, but  $\alpha \approx 10^\circ$  was chosen as it is the one she used in her thesis as well. However, it was observed that although the predicted state of the NMPC never crossed the upper bound for  $\alpha$ , the actual state did not always follow this limit, due to remaining differences in the simulator model and the model used by the NMPC. To ensure that the angle of attack stays well below  $\alpha \approx 10^\circ$ , the upper bound for  $\alpha$  was set to  $\bar{\alpha} = 8^\circ$ . During the reduced

airspeed simulation, however, the angle of attack reached  $\alpha = 8.6^\circ$ . It is unknown whether it would have reached 10 degrees under different circumstances, but it is possible to reduce  $\bar{\alpha}$  even more to ensure it does not happen. In the project thesis [1] the limit was set to  $\bar{\alpha} = 4^\circ$ , which kept the maximum angle of attack lower (around  $\alpha_{max} = 6^\circ$ ), but at the expense of the airspeed tracking performance in order to keep the optimization problem feasible.



## Chapter 7

# Conclusion

This thesis explores how the NMPC controller for inner-loop control of roll, pitch and airspeed, developed by Reinhardt [5] and implemented in the Matlab/ Simulink simulator developed by Gryte [11], can be modified to be better suited for the icing problem. The model in the NMPC is first updated to use the aerodynamic clean coefficients found in Winter's work [12], to match the model used in the simulator. Although the model is updated, this NMPC does not include icing in its model, so the icing conditions are handled by its disturbance observer. Then, the effects of icing and asymmetry are implemented in the model of the NMPC, following Winter's and Kleiven's work [13], to see to what degree its performance and robustness improve. The two NMPC controllers, the one with ice in its model and the one without, are tested and compared in different simulations, run with icing conditions in addition to other disturbances such as reduced airspeed and severe wind conditions. Their performance is also compared to the PID and MRAC controllers, previously developed by Högnadóttir in her thesis [10].

The simulation results have been assessed using the IAE performance metric, and they show a similar performance of the PID, MRAC and NMPC without icing controllers, depending on the simulation run. The NMPC without icing performs better than the PID and MRAC under the reduced airspeed simulation, but shows a slower response in the other simulations. However, a clear improvement in performance can be seen when icing and asymmetry are included in the model of the NMPC. In this case, the NMPC outperforms all controllers in all simulations when it comes to pitch and airspeed tracking, and has the second-best roll tracking, only surpassed by the PID. Additional simulations were also performed to test the performance of the NMPC assuming that the detected icing level or estimated disturbance observer states are less precise in practice. These scenarios were not found to compromise the performance of the NMPC controller.

When it comes to the path-following abilities of the controllers, guidance simulations are performed with different wind directions and icing conditions. When the simulations are performed with 100 % icing asymmetry, the performance de-

gradation is the most severe. In this case, the NMPC with icing and the MRAC have the best performances. The PID controller crashes in two of these simulations. During one of the guidance simulations, the two NMPC controllers crash as well, due to the weight on the airspeed error of the disturbance observer being too high. Although it is shown that reducing  $l_{V_a}$  solves this issue, it has not been tested with the other simulations. A better tuning should therefore be found and tested. However, considering the overall better performance of the NMPC with icing in its model compared to the other controllers throughout the different simulations, it can still be concluded that this NMPC controller is more suited to handle the effects caused by icing asymmetry.

When it comes to further work, better tuning of the NMPC should be found to ensure that it works in all simulation cases. When it comes to improvements to the model, both the stall limit and the aerodynamic coefficients between the iced and clean state of the wing are uncertain, so more experiments are needed to expand our knowledge on this topic. In addition, the icing on the propeller model also needs more research, as the original model was developed for another propeller than the one the Skywalker X8 uses. Although the model was assumed to be valid, a greater understanding of the differences between the two propellers, or experiments with the correct propeller, would increase the accuracy of the model. Finally, the NMPC controllers should be tested in practice.

# Bibliography

- [1] N. A. Ghindaoanu, *Control of Fixed-Wing UAVs in Icing Conditions Using Nonlinear Model Predictive Control*. unpublished, 2023.
- [2] 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>.
- [3] 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.
- [4] M. Bragg, A. Broeren **and** L. Blumenthal, ‘Iced-Airfoil Aerodynamics,’ *Progress in Aerospace Sciences*, 41:323-362, 2005.
- [5] D. P. Reinhardt, *On Nonlinear and Optimization-based Control of Fixed-Wing Unmanned Aerial Vehicles*. Doctoral thesis, Norwegian University of Science **and** Technology, 2022.
- [6] *MS Windows NT kernel description*, <https://ardupilot.org/>, Accessed: 2021-12-03.
- [7] *Acados solver for nonlinear optimal control*, <https://github.com/acados/acados>, Accessed: 2022-12-15.
- [8] R. W. Beard **and** T. W. McLain, *Small Unmanned Aircraft: Theory and Practice*. Princeton University Press, 2012.
- [9] 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.
- [10] S. Högnadottir, *Inner-Loop Adaptive Control of FixedWing Unmanned Aerial Vehicles in Icing Conditions*. M.S. thesis, Norwegian University of Science **and** Technology, 2022.
- [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] 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.
- [14] 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*, **journal** 102, **pages** 101–112, 2019, ISSN: 0894-1777. DOI: <https://doi.org/10.1016/j.expthermflusci.2018.11.008>.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] P Fitzpatrick, *Calculation of Thrust in a Ducted Fan Assembly for Hovercraft*. Technical report, Hovercraft Club of Great Britain, 2003.
- [19] R. W. Beard, *Uavbook Supplement. Additional thoughts on propeller thrust model*. Technical report, 2014.
- [20] *Flying qualities of piloted airplanes*. U.S. Military Specification MIL-F8785C, 1980.
- [21] T. I. Fossen, *Handbook of marine craft hydrodynamics and motion control*. Second edition. Wiley-Blackwell, 2021.
- [22] 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.
- [23] 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.
- [24] K. Gryte, *Precision control of fixed-wing UAV and robust navigation in GNSS-denied environments*. Doctoral thesis, Norwegian University of Science and Technology, 2020.
- [25] 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.

- [26] *Matlab + Simulink and Octave Interface*, [https://docs.acados.org/matlab\\_octave\\_interface/index.html](https://docs.acados.org/matlab_octave_interface/index.html), Accessed: 2022-11-01.
- [27] A. Cristofaro, A. P. Aguiar **and** T. A. Johansen, 'Icing Detection and Identification for Unmanned Aerial Vehicles using Adaptive Nested Multiple Models,' *International Journal of Adaptive Control and Signal Processing*, , Vol. 31, pp. 1584–1607, 2017. DOI: 10.1002/acs.2787.
- [28] D. Rotondo, A. Cristofaro **and** T. A. Johansen, 'Icing diagnosis in unmanned aerial vehicles using an LPV multiple model estimator,' *IFAC World Congress, Toulouse*, 2017. DOI: 10.1002/acs.2787.
- [29] *N-D Lookup Table*, <https://se.mathworks.com/help/simulink/slref/ndlookuptable.html>, Accessed: 2022-12-15.
- [30] *UIUC Propeller Database - Volume 3*, <https://m-selig.ae.illinois.edu/props/volume-3/propDB-volume-3.html>, Accessed: 2022-12-05.
- [31] T. A. Johansen, A. Cristofaro, K. L. Sørensen, J. M. Hansen **and** T. I. Fosse, 'On estimation of wind velocity, angle-of-attack and sideslip angle of small UAVs using standard sensors,' *International Conference on Unmanned Aircraft Systems*, Denver, 2015. DOI: 10.1109/ICUAS.2015.7152330.
- [32] A. W. Wenz **and** T. A. Johansen, 'Moving Horizon Estimation of Air Data Parameters for UAVs,' *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, pp. 2101-2121, 2020. DOI: <https://doi.org/10.1109/TAES.2019.2946677>.
- [33] K. T. Borup, B. B. Stovner, T. I. Fossen **and** T. A. Johansen, 'Kalman Filters for Air Data System Bias Correction for a Fixed-Wing UAV,' *IEEE Transactions on Control Systems Technology*, Vol. 28, pp. 2164-2176, 2020. DOI: <https://doi.org/10.1109/TCST.2019.2931672>.



# Appendix A

## The PID and MRAC Controllers

### A.1 Roll, Pitch and Airspeed Control with PID and MRAC

The NMPC controllers were compared to two inner-loop, roll and pitch controllers: a proportional-integral-derivative (PID) controller and a model reference adaptive controller (MRAC), developed in the previous work of Gryte [11] and Högnadóttir [10], and explained in Subsections A.1.1 and A.1.2. When roll and pitch are controlled using the PID or the MRAC controllers, the airspeed is controlled using a proportional-integral (PI) controller, introduced in Subsection A.1.3.

#### A.1.1 PID

The PID controller used in Högnadóttir's work [10] is based on Beard & McLain [8], and it is formed by two PID controllers, one for roll and one for pitch. The roll PID controller determines the aileron  $\delta_a$  needed to drive  $\phi$  to the commanded angle  $\phi_{cmd}$ :

$$\delta_a = k_{p_\phi}(\phi_{cmd} - \phi) + \frac{k_{i_\phi}}{s}(\phi_{cmd} - \phi) - k_{d_\phi}p, \quad (\text{A.1})$$

where the control gains  $k_{p_\phi}$ ,  $k_{i_\phi}$  and  $k_{d_\phi}$  are given in Table A.2a. Similarly, the controller for pitch is a PID controller which determines the elevator  $\delta_e$  needed to drive  $\theta$  to the commanded angle  $\theta_{cmd}$ :

$$\delta_e = k_{p_\theta}(\theta_{cmd} - \theta) + \frac{k_{i_\theta}}{s}(\theta_{cmd} - \theta) - k_{d_\theta}q, \quad (\text{A.2})$$

where the control gains  $k_{p_\theta}$ ,  $k_{i_\theta}$  and  $k_{d_\theta}$  are given in Table A.2b.

#### A.1.2 Model Reference Adaptive Control (MRAC)

The MRAC controllers used in Högnadóttir's work [10] are based on Lavretsky and Wise [9], and the most important parts of the theory will be explained in this section.

A nonlinear system can be given on the form:

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\Lambda(\mathbf{u} + \Theta^\top \Phi(\mathbf{x})), \quad (\text{A.3})$$

where  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is the unknown state matrix,  $\mathbf{B} \in \mathbb{R}^{n \times m}$  is the known control matrix,  $\Lambda \in \mathbb{R}^{m \times m}$  is the unknown control effectiveness matrix,  $\Theta \in \mathbb{R}^{N \times m}$  is constant and unknown and  $\Phi(\mathbf{x}) \in \mathbb{R}^N$  is the known regressor vector [9]. The objective is to track the reference model given by

$$\dot{\mathbf{x}}_{ref} = \mathbf{A}_{ref}\mathbf{x}_{ref} + \mathbf{B}_{ref}\mathbf{r}(t), \quad (\text{A.4})$$

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

$$\mathbf{u} = \hat{\mathbf{K}}_x^\top \mathbf{x} + \hat{\mathbf{K}}_r^\top \mathbf{r} - \hat{\Theta}^\top \Phi(\mathbf{x}), \quad (\text{A.5})$$

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

$$\dot{\hat{\mathbf{K}}}_x = \text{Proj}(\hat{\mathbf{K}}_x, -\Gamma_x \mathbf{x} \mathbf{e}^\top \mathbf{P} \mathbf{B}), \quad (\text{A.6})$$

$$\dot{\hat{\mathbf{K}}}_r = \text{Proj}(\hat{\mathbf{K}}_r, -\Gamma_r \mathbf{r} \mathbf{e}^\top \mathbf{P} \mathbf{B}), \quad (\text{A.7})$$

$$\dot{\hat{\Theta}}_x = \text{Proj}(\hat{\Theta}_x, -\Gamma_\Theta \Phi \mathbf{e}^\top \mathbf{P} \mathbf{B}). \quad (\text{A.8})$$

$\text{Proj}(\cdot)$  is a projection operator, defined in [9], the symmetric and positive-definite matrices  $\Gamma_x$ ,  $\Gamma_r$  and  $\Gamma_\Theta$  are the adaptive rates, and  $\mathbf{e} = \mathbf{x} - \mathbf{x}_{ref}$  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 [8], the roll dynamics are linearized as:

$$\dot{\phi} = p + d_{\phi_1}, \quad (\text{A.9})$$

$$\ddot{\phi} = -a_{\phi_1} \dot{\phi} + a_{\phi_2} \delta_a + d_{\phi_2}, \quad (\text{A.10})$$

where  $d_{\phi_1}$  and  $d_{\phi_2}$  are considered the disturbances of the system. The dynamics written on the same form as Eq. (A.3) are given by

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\Lambda(\mathbf{u} + \Theta^\top \Phi(\mathbf{x})), \quad (\text{A.11})$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & a_1 \end{bmatrix} \begin{bmatrix} \phi \\ p \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \lambda_1 (\delta_a + [\theta_{bias,roll}][1]), \quad (\text{A.12})$$

with  $a_1 = -a_{\phi_1}$ ,  $\lambda_1 = a_{\phi_2}$ ,  $\theta_{bias,roll} = d_{\phi_2}$ , where  $(\mathbf{A}, \mathbf{B}\Lambda)$  is controllable for all  $a_{\phi_2} \neq 0$ . Likewise, the linearized pitch dynamics [8] are given as

$$\dot{\theta} = q + d_{\theta_1}, \quad (\text{A.13})$$

$$\ddot{\theta} = -a_{\theta_1} \dot{\theta} - a_{\theta_2} \theta + a_{\theta_3} \delta_e + d_{\theta_2}, \quad (\text{A.14})$$



which on the same form as Eq. (A.3) they become

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{\Lambda}(\mathbf{u} + \mathbf{\Theta}^\top \mathbf{\Phi}(\mathbf{x})), \quad (\text{A.15})$$

$$\begin{bmatrix} \dot{\theta} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ a_2 & a_3 \end{bmatrix} \begin{bmatrix} \theta \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \lambda_2 (\delta_e + [\theta_{bias,pitch}][1]), \quad (\text{A.16})$$

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

### A.1.3 Airspeed Controller

When the PID and MRAC controllers are used to control roll and pitch, the airspeed is controlled by the following PI controller, which determines the throttle  $\delta_t$  needed to drive  $V_a$  to the commanded airspeed  $V_{a,cmd}$ :

$$\delta_t = \delta_t^* + k_{p_v}(V_{a,cmd} - V_a) + \frac{k_{i_v}}{s}(V_{a,cmd} - V_a), \quad (\text{A.17})$$

where  $\delta_t^*$  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.

## A.2 Implementation of the PID and MRAC Controllers

The PID and MRAC controllers were tuned and implemented in the Matlab/ Simulink simulator as described in Högnadóttir's work [10]. The PID controller for roll and pitch is given by equations (A.1) and (A.2), where the commanded angles  $\phi_{cmd}$  and  $\theta_{cmd}$  are set equal to the angles  $\phi_{ref}$  and  $\theta_{ref}$ , 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 [10], and can be seen in Table A.1.

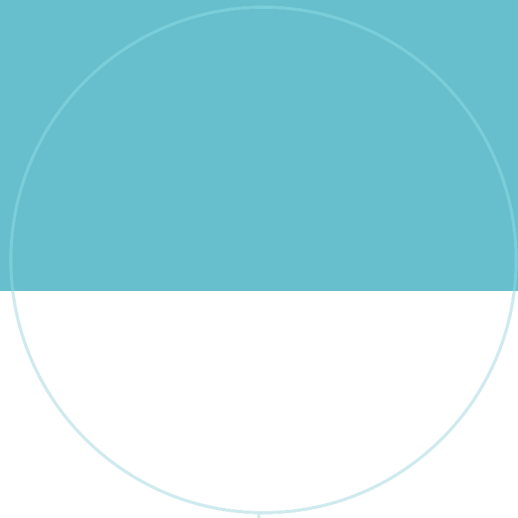
(a) Roll controller gains		(b) Pitch controller gains	
Parameter	Value	Parameter	Value
$k_{p_\phi}$	2.5	$k_{p_\theta}$	2.5
$k_{i_\phi}$	2	$k_{i_\theta}$	2
$k_{d_\phi}$	0.01	$k_{d_\theta}$	0.01

**Table A.1:** Tuning parameters of the PID controller.

When it comes to the MRAC controller, described in Section A.1.2, the reference model  $\mathbf{x}_{ref}$  is found as described in Section 4.1, where  $\phi_{cmd}$  and  $\theta_{cmd}$  are the commanded reference  $\mathbf{r}(t)$  in the MRAC equations. The MRAC was tuned as described in Högnadóttir's work [10], and the resulting tuning parameters can be seen in Table A.2.

(a) Roll adaptive rates		(b) Pitch adaptive rates	
Parameter	Value	Parameter	Value
$\mathbf{Q}$	diag[3, 1]	$\mathbf{Q}$	diag[4, 0.4]
$\Gamma_x$	diag[12, 4]	$\Gamma_x$	diag[6, 0.01]
$\Gamma_r$	10	$\Gamma_r$	5
$\Gamma_\Theta$	15	$\Gamma_\Theta$	10

**Table A.2:** Tuning parameters of the MRAC controller.



 **NTNU**

Norwegian University of  
Science and Technology