2024 European Control Conference (ECC)
June 25-28, 2024. Stockholm, Sweden

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

N. Adelina Ghindaoanu Kristoffer Gryte Dirk Reinhardt Tor A. Johansen
Department of Engineering Cybernetics, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

***Abstract*— This paper explores nonlinear model predictive control (NMPC) for an unmanned aerial vehicle (UAV) operating in icing conditions, simulated as asymmetric icing on the wings and icing on the propeller. First, a NMPC flight controller based on a nominal model is tested together with a disturbance observer to handle unmodelled effects. Second, we test a NMPC that includes the effect of asymmetric icing in the prediction model to explore how it affects its performance and robustness, and simulations are performed to compare this NMPC to the NMPC without icing knowledge, and to a conventional PID controller. The results show a clear improvement in the performance of the NMPC when the NMPC prediction model includes icing, as well as better performance and robustness in extreme icing asymmetry cases compared to the PID controller. Additional simulations were performed, indicating a significant degree of robustness.**

I. Introduction

As the use of unmanned aerial vehicles (UAVs) increase, controllers that are able to operate under severe weather conditions are needed. This paper focuses on in-flight icing, which has been shown to lead to a degradation of the aerodynamic performance of the aircraft, with a significant decrease in lift, increase in drag, and a deterioration of the stall limits [1]. This affects a small UAV more severely than a larger aircraft, as ice accretion will make up a more significant part of its total weight, lift and drag. As the ice conditions cannot be identified visually by a pilot in this case, the controller must be able to handle them and ensure the safety of the UAV. Knowledge about the icing model is therefore needed and progress has been made, where Winter's work [2] improved the aerodynamic model of the Skywalker X8 [3], the UAV used in this paper, with its aerodynamic coefficient data given for iced and clean airfoils. Kleiven [4] extended the model to account for asymmetric icing on the wings, as ice accretion and ice shedding might not occur at the same time on both wings. Additionally, propeller icing results in a significant decrease in thrust and an increase in torque, and a propeller icing model was developed by Müller [5] for a propeller similar to the Skywalker X8's.

For UAV control, inner-loop PID controllers are typically used. Previous work by Högnadottir [6] explores model reference adaptive control (MRAC), which was found to perform better than the PID controller under certain conditions. Although having overall similar tracking performance, both struggling the most with severely asymmetric icing. Nonlinear Model Predictive Control (NMPC), on the other hand, takes nonlinear effects into account and has the advantage of taking constraints as part of its optimization problem, being able to consider actuator and safety limits. Due to the fast dynamics of a UAV, NMPCs are usually used for guidance [7], while a PID controller handles the low-level control. Few works such as [8] explore inner-loop MPC control of a fixed-wing UAV. Moreover, the NMPC developed in Reinhardt's work [9], [10] also considers nonlinearities, and it showed in experiments that real-time control was feasible using state-of-the-art numerical algorithms and hardware, while having a similar or superior performance compared to industry-standard PID controllers.

In this paper, Reinhardt's NMPC is tested in icing conditions in a Matlab/Simulink simulator, including Winter's and Kleiven's extensions to the icing model, [2], [4]. The simulator is also expanded to include Müller's model for icing on the propeller, adapted to the propeller of the Skywalker X8. Similar to [10], the NMPC is used with a disturbance observer to handle disturbances and unmodelled effects. The disturbance observer's ability to handle icing is explored here, as well as how the NMPC can be modified to increase its robustness and performance when subject to icing and wind disturbances, particularly asymmetric icing on the wings, as the PID controller had its greatest performance degradation in this case. More specifically, the performance of the NMPC with icing knowledge is compared to the one of the PID and the NMPC without icing knowledge in a relatively low airspeed simulation in different icing levels. The NMPC with icing knowledge was additionally modified to show how its performance might change when run in practice, considering that the perfect information on the states and icing level used in the simulation might not be achieved. Except for improvements in the tuning of the NMPC controllers and some additional simulations, the findings and results presented in this paper originate from the Master's thesis on the same topic [11].

II. Aerodynamic model

The kinematic and dynamic equations of the UAV are given in the body-fixed frame $\{b\}$ and NED frame $\{n\}$, which we assume to be inertial, where the state of the model is given by $\boldsymbol{x} = [\boldsymbol{p}_{nb}^{n\top}, \Theta_{nb}^{\top}, \boldsymbol{v}_{nb}^{b\top}, \boldsymbol{\omega}_{nb}^{b\top}]^{\top}$. We use the notation from [12], where e.g. $\boldsymbol{p}_{nb}^{n}$ denotes the position of $\{b\}$ in $\{n\}$ expressed in coordinates of $\{n\}$. This model is used to simulate the UAV, and can be found in [11]. The vector $\boldsymbol{p}_{nb}^{n}$ describes the position of the UAV in the inertial frame $\{n\}$,

This work was partly supported by the Norwegian Research Council and through the AutoFly project (grant no. 261791), the icing protection project (grant no. 316425), and the Center for Autonomous Marine Operations and Systems (NTNU AMOS) (grant no. 223254). Corresponding Author: gryte@ieee.org

$\boldsymbol{v}_{nb}^b = [u,\, v,\, w]^\top$ is the velocity vector in the body-fixed frame $\{b\}$. The Euler angles roll, pitch and yaw, $\boldsymbol{\Theta}_{nb} = [\phi,\, \theta,\, \psi]^\top$, in $\{n\}$, describe the orientation of the UAV, and $\boldsymbol{\omega}_{nb}^b = [p,\, q,\, r]^\top$ is the angular velocity vector, given in $\{b\}$, where p, q, and r are the roll, pitch and yaw rates. Reinhardt [9] utilizes an alternative dynamic model, given in the stability $\{s\}$ and wind $\{w\}$ frames, with respect to the aerodynamic quantities airspeed V_a, angle of attack (AOA) α and sideslip angle (SSA) β, given by:

$$V_a = ||\boldsymbol{v}_r^b|| = \sqrt{u_r^2 + v_r^2 + w_r^2}, \tag{1a}$$

$$\alpha = \arcsin\left(\frac{w_r}{u_r}\right), \quad \beta = \arcsin\left(\frac{v_r}{V_a}\right). \tag{1b}$$

The relative velocity vector $\boldsymbol{v}_r^b = [u_r,\, v_r,\, w_r]^\top$ is given by $\boldsymbol{v}_r^b = \boldsymbol{v}_{nb}^b - \boldsymbol{R}_{nb}^\top \boldsymbol{v}_{nw}^n$, where $\boldsymbol{v}_{nw}^n$ is the wind velocity vector in the inertial frame, and $\boldsymbol{R}_{nb} = \boldsymbol{R}(\boldsymbol{\Theta}_{nb})$ is the rotation matrix from $\{b\}$ to $\{n\}$. To rotate vectors in the frames $\{b\}$, $\{s\}$, $\{w\}$, we use the rotation matrices $\boldsymbol{R}_{sb}(\alpha)$, $\boldsymbol{R}_{ws}(\beta)$ and $\boldsymbol{R}_{wb}(\alpha,\beta) = \boldsymbol{R}_{ws}(\beta)\boldsymbol{R}_{sb}(\alpha)$, described in [13].

A. The UAV Model

Following Reinhardt's work [9], the UAV model used in the formulation of the NMPC is given in stability-wind frame representation. In this model, $\boldsymbol{p}_{nb}^n$ is not included in the state vector since it is not needed to control speed and attitude. The orientation is given by the rotation matrix $\boldsymbol{R}_{nb}$ instead of the Euler angles $\boldsymbol{\Theta}_{nb}$, in order to obtain a globally unique and non-singular attitude representation. The velocity is represented in wind frame using V_a, α and β, and the angular velocity is given in the stability frame by $\boldsymbol{\omega}_{nb}^s = \boldsymbol{R}_{sb}\boldsymbol{\omega}_{nb}^b$. The model is then given by:

$$\begin{bmatrix} \dot{V}_a \\ \dot{\beta} V_a \\ \dot{\alpha} V_a \cos\beta \end{bmatrix} = \frac{1}{m}\boldsymbol{R}_{wb}(\boldsymbol{F}_a^b + \boldsymbol{F}_t^b + \boldsymbol{F}_g^b) - \boldsymbol{\omega}_{nb}^w \times \boldsymbol{v}_r^w \tag{2a}$$

$$\dot{\boldsymbol{R}}_{nb} = \begin{bmatrix} \dot{\boldsymbol{r}}_x, & \dot{\boldsymbol{r}}_y, & \dot{\boldsymbol{r}}_z \end{bmatrix}^\top = \boldsymbol{R}_{nb}\boldsymbol{S}(\boldsymbol{R}_{sb}^\top \boldsymbol{\omega}_{nb}^s) \tag{2b}$$

$$\dot{\boldsymbol{\omega}}_{nb}^s = -\boldsymbol{\omega}_{bs}^s \times \boldsymbol{\omega}_{nb}^s + (\boldsymbol{J}^s)^{-1}(\boldsymbol{R}_{sb}\boldsymbol{M}_a^b - \boldsymbol{\omega}_{nb}^s \times \boldsymbol{J}^s \boldsymbol{\omega}_{nb}^s), \tag{2c}$$

where $\boldsymbol{\omega}_{nb}^w$ is the angular velocity vector decomposed in the wind frame $\{w\}$ and given by $\boldsymbol{\omega}_{nb}^w = \boldsymbol{R}_{wb}\boldsymbol{\omega}_{nb}^b$. The matrix $\boldsymbol{J}^s = \boldsymbol{R}_{sb}\boldsymbol{J}^b\boldsymbol{R}_{sb}^\top$ is the inertia matrix decomposed in the stability frame $\{s\}$, $\boldsymbol{\omega}_{bs}^s = [0, \dot{\alpha}, 0]^\top$ is the angular velocity of $\{s\}$ relative to the body-fixed reference frame and decomposed in $\{s\}$, and $\boldsymbol{v}_r^w = [V_a, 0, 0]^\top$ is the relative velocity vector decomposed in $\{w\}$.

B. Forces and Moments

The vector $\boldsymbol{M}_a^b = [l,\, m,\, n]^\top$ is the aerodynamic moment in body-fixed reference frame, and the forces $\boldsymbol{F}_a^b$, $\boldsymbol{F}_g^b$, $\boldsymbol{F}_t^b$ are the aerodynamic, gravitational and propulsion forces, where

$$\boldsymbol{F}_a^b = \boldsymbol{R}_{wb}^\top \boldsymbol{F}_a^w = \boldsymbol{R}_{wb}^\top \begin{bmatrix} -F_{\text{drag}} \\ F_{\text{side}} \\ -F_{\text{lift}} \end{bmatrix}. \tag{3}$$

The aerodynamic model is given by the quasi-linear approximation:

$$\begin{bmatrix} F_{\text{drag}} \\ F_{\text{side}} \\ F_{\text{lift}} \end{bmatrix} = \bar{q}S \begin{bmatrix} C_D(\alpha) + C_{D_q}\frac{c}{2V_a}q + C_{D_{\delta_e}}\delta_e \\ C_S(\beta) + C_{S_p}\frac{b}{2V_a}p + C_{S_r}\frac{b}{2V_a}r + C_{S_{\delta_a}}\delta_a \\ C_L(\alpha) + C_{L_q}(\alpha)\frac{c}{2V_a}q + C_{L_{\delta_e}}\delta_e \end{bmatrix} \tag{4}$$

$$\begin{bmatrix} l \\ m \\ n \end{bmatrix} = \bar{q}S \begin{bmatrix} b(C_l(\beta) + C_{l_p}\frac{b}{2V_a}p + C_{l_r}\frac{b}{2V_a}r + C_{l_{\delta_a}}\delta_a) \\ c(C_m(\alpha) + C_{m_q}(\alpha)\frac{c}{2V_a}q + C_{m_{\delta_e}}\delta_e) \\ b(C_n(\beta) + C_{n_p}\frac{b}{2V_a}p + C_{n_r}\frac{b}{2V_a}r + C_{n_{\delta_a}}\delta_a) \end{bmatrix}, \tag{5}$$

where S, b, c and $\bar{q} = \frac{1}{2}\rho V_a^2$ are physical parameters of the UAV, and $\delta_e \in [\delta_{e,min}, \delta_{e,max}]$ and $\delta_a \in [\delta_{a,min}, \delta_{a,max}]$ are the elevator and aileron deflections. Notice that a rudder deflection δ_r is not included in this model as the Skywalker X8 UAV used in the case study does not have a rudder. The aerodynamic coefficients C_* are given by Winter *et. al.* [2], based on wind tunnel data for the Skywalker X8 [3] and CFD analysis of the UAV in icing. The parameters used can be found in [11].

The gravitational force $\boldsymbol{F}_g^b$ follows Newton's second law and is given by $\boldsymbol{F}_g^b = \boldsymbol{R}_{nb}^\top m\boldsymbol{g}^n = \boldsymbol{R}_{nb}^\top [0,\, 0,\, mg]^\top$, where m is the mass of the UAV and g is the gravitational constant. The propulsion force $\boldsymbol{F}_t^b$ is based on [14] and given by:

$$\boldsymbol{F}_t^b = \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \frac{\rho D^4}{2\pi^2} C_T(J)\omega^2 \\ 0 \\ 0 \end{bmatrix}, \tag{6}$$

where ρ is the air density, D is the propeller diameter, and $\omega = \delta_t \omega_{max}$ is the propeller speed that is proportional to throttle $\delta_t \in [0, 1]$. The thrust coefficient $C_T(J)$ is

$$C_T(J) = C_{T,0} + C_{T,1}J, \quad J = \frac{2\pi V_a}{\omega D}, \tag{7}$$

where J is the advance ratio, and $C_{T,0} = 0.126$, $C_{T,1} = -0.1378$, [14].

III. Icing Model

A. Asymmetric Icing for Aerodynamic Forces and Moments

The model for icing on the wings is based on aerodynamic coefficient data from [2] given for the clean and iced states as $C_{*,clean}$ and $C_{*,iced}$. The icing data corresponds to mixed ice, the most severe icing type in terms of aerodynamic performance degradation [1], and is considered valid for this model. The coefficients in intermediate icing states are then approximated as in [4]:

$$C_*(\zeta) = \zeta C_{*,iced} + (1 - \zeta)C_{*,clean}, \tag{8}$$

with $\zeta \in [0, 1]$ denoting the icing level on each wing, where $\zeta = 0$ = clean wing, and $\zeta = 1$ = fully iced wing. As ice can build up or shed asymmetrically, the aerodynamic forces and moments on each wing are considered by extending the icing model to the asymmetric model found by Kleiven [4].

B. Propulsion Icing Model

Based on the work of Müller [5], a model for the thrust coefficient in icing conditions can be given by:

$$C_{T,iced} = C_T(J)\,(1 + \min(TWC, TWC_{max})\Delta C_T(T)), \tag{9}$$

$$\Delta C_T(T) = \Delta C_{T,0} + \Delta C_{T,1}T + \Delta C_{T,2}T^2 \tag{10}$$

where $C_T(J)$ is given by (7), T is the temperature in $[^\circ C]$, TWC is the total water amount collected on the propeller, and TWC_{max} is the maximum amount of water collected on the propeller before ice shedding is expected. They are given by:

$$TWC = t \cdot LWC \cdot \omega \frac{D}{2}, \tag{11}$$

$$TWC_{max} = \frac{A_{max}}{\frac{D}{2}\omega^2}, \tag{12}$$

$$A_{max} = A_{max,0} + A_{max,1} T^2, \tag{13}$$

where t is the icing accretion time, LWC is the liquid water content, ω is the rotation rate of the propeller and D is the propeller diameter. Eq. (13) describes the adhesion forces of the ice on the propeller. It can be seen that there is an equilibrium between A_{max} and TWC_{max}, as ice shedding occurs when the amount of ice collected is too large for the adhesion forces to hold it. For information about the values used as well as details on how they were found, see [11].

IV. NMPC Controllers

Following [9], the NMPC's dynamic model is given by Eq. (2), and can be written as $\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})$ where the state vector is defined as $\boldsymbol{x} = [V_a, \beta, \alpha, \boldsymbol{r}_x^\top, \boldsymbol{r}_y^\top, \boldsymbol{r}_z^\top, {\boldsymbol{\omega}_{nb}^s}^\top, \delta_a, \delta_e, \delta_t]^\top \in \mathbb{R}^{n_x}$ and the input as $\boldsymbol{u} = [\dot{\delta}_a, \dot{\delta}_e, \dot{\delta}_t]^\top \in \mathbb{R}^{n_u}$. Asymmetric icing is included as an option in the NMPC prediction model, whereas the propeller icing model is only used in simulation model. The model is discretized using an explicit Runge-Kutta method of order 4, represented as $\boldsymbol{x}(k+1) = \boldsymbol{f}_{RK4}(\boldsymbol{x}(k), \boldsymbol{u}(k))$. In order to account for unmodeled dynamics, modeling inaccuracies and disturbances, and to obtain offset-free control, a disturbance observer based on [10] for the states $[V_a,\ \beta,\ \alpha,\ {\boldsymbol{\omega}_{nb}^s}^\top]^\top$ leads to:

$$\boldsymbol{f}_d(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{d}) \triangleq \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) + \begin{bmatrix} \boldsymbol{d}_f^\top, & \boldsymbol{0}_{1\times 9}, & \boldsymbol{d}_m^\top, & \boldsymbol{0}_{1\times 3} \end{bmatrix}^\top.$$

The disturbance estimates $\boldsymbol{d}_f$ and $\boldsymbol{d}_m$ are initiated as $\boldsymbol{d}_f(0) = \boldsymbol{d}_m(0) = \boldsymbol{0}_{3\times 1}$ and updated with the NMPC:

$$\boldsymbol{d}_f(t) \leftarrow \boldsymbol{d}_f(t) + \boldsymbol{L}_f \begin{bmatrix} \Delta V_a(t), & \Delta\beta(t), & \Delta\alpha(t) \end{bmatrix}^\top, \tag{14}$$

$$\boldsymbol{d}_m(t) \leftarrow \boldsymbol{d}_m(t) + \boldsymbol{L}_m \Delta\omega_{nb}^s(t), \tag{15}$$

where $\Delta z(t) = z(t) - z_{nmpc}(t)$ represents the error between a state and the state predicted by the NMPC, and $\boldsymbol{L}_f$ and $\boldsymbol{L}_m$ are the observer gain matrices, given by:

$$\boldsymbol{L}_f = \text{diag}(l_{V_a},\ l_\beta,\ l_\alpha), \quad \boldsymbol{L}_m = \text{diag}(l_p,\ l_q,\ l_r). \tag{16}$$

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

$$\boldsymbol{r} \triangleq \begin{bmatrix} V_{a,ref}, & \boldsymbol{\Gamma}_{ref}^\top, & \boldsymbol{\omega}_{ref}^\top \end{bmatrix}^\top, \tag{17}$$

where $\boldsymbol{\Gamma}_{ref} \in \mathbb{S}^2$ is the reference for the reduced attitude vector, used in order to represent roll angle $\phi \in [-\pi,\ \pi]$ and pitch angle $\theta \in [-\frac{\pi}{2},\ \frac{\pi}{2}]$, and can be parameterized as:

$$\boldsymbol{\Gamma}(\phi, \theta) = \begin{bmatrix} -\sin\theta, & \cos\theta\sin\phi, & \cos\theta\cos\phi \end{bmatrix}^\top. \tag{18}$$

The reduced attitude vector is used to avoid singularities in the Euler angles, and to ensure that the NMPC can handle the full range of the attitude space without trigonometric functions. The reference vector $\boldsymbol{r}$ also contains $\boldsymbol{\omega}_{ref}$ to add a damping effect and achieve a smoother response. The vector $\boldsymbol{\omega}_{ref}$ is derived from ϕ_{ref} and θ_{ref}, as explained in [15]. In addition, constraints on the state $\boldsymbol{x}$, input $\boldsymbol{u}$ and slack variables $\boldsymbol{s}$ are added to the problem to ensure that the operational and physical limits of the UAV are not surpassed:

$$\begin{aligned}
V_a - \underline{V}_a + \underline{s}_{V_a} &\geq 0 & -V_a + \overline{V}_a + \overline{s}_{V_a} &\geq 0 \\
\beta - \underline{\beta} + \underline{s}_\beta &\geq 0 & -\beta + \overline{\beta} + \overline{s}_\beta &\geq 0 \\
\alpha - \underline{\alpha} &\geq 0 & -\alpha + \overline{\alpha} &\geq 0 \\
p_s - \underline{p}_s &\geq 0 & -p_s + \overline{p}_s &\geq 0 \\
q_s - \underline{q}_s &\geq 0 & -q_s + \overline{q}_s &\geq 0 \\
r_s - \underline{r}_s &\geq 0 & -r_s + \overline{r}_s &\geq 0 \\
\delta_a + \delta_e - \underline{\delta}_{el} &\geq 0 & -\delta_a - \delta_e + \overline{\delta}_{el} &\geq 0 \\
-\delta_a + \delta_e - \underline{\delta}_{er} &\geq 0 & \delta_a - \delta_e - \overline{\delta}_{er} &\geq 0 \\
\delta_t - \underline{\delta}_t &\geq 0 & -\delta_t + \overline{\delta}_t &\geq 0 \\
\boldsymbol{u} - \underline{\boldsymbol{u}} &\geq 0 & -\boldsymbol{u} + \overline{\boldsymbol{u}} &\geq 0,
\end{aligned} \tag{19}$$

where the slack variables $\boldsymbol{s} = [\underline{s}_{V_a},\ \underline{s}_\beta,\ \overline{s}_{V_a},\ \overline{s}_\beta]^\top \geq 0$ have been added to ensure the feasibility of the problem. Slack variables were not deemed necessary for the angular velocity, whereas for α they were omitted to prevent it from violating the stall limit, although they could have been added with a higher weight as the unfeasibility risk remains. Ineq. (19) is summarised by the function $\boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{s}) \geq 0$. The optimal control problem (OCP) over a prediction horizon T_h, discretized into N steps with shooting interval $\Delta t = T_h/N$ gives the nonlinear problem (NLP):

$$\min_{\boldsymbol{x}(\cdot), \boldsymbol{u}(\cdot)} \quad \sum_{k=0}^{N-1} \ell(\boldsymbol{x}(k|t), \boldsymbol{u}(k|t), \boldsymbol{r}(k|t)) + \frac{1}{2}\boldsymbol{s}^\top \boldsymbol{P}\boldsymbol{s}, \tag{20a}$$

$$\text{s. t.} \quad \boldsymbol{x}(0|t) = \boldsymbol{x}(t) \tag{20b}$$

$$\boldsymbol{x}(k+1|t) = f_{RK4}(\boldsymbol{x}(k|t), \boldsymbol{u}(k|t), \boldsymbol{d}(0|t)) \tag{20c}$$

$$\boldsymbol{h}(\boldsymbol{x}(k|t), \boldsymbol{u}(k|t), \boldsymbol{s}) \geq 0, \quad \text{for } k \in \{0, ..., N-1\}, \tag{20d}$$

The stage cost $\ell(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{r})$ is defined as

$$\begin{aligned}
\ell(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{r}) \quad = \quad & q_{V_a}(V_a - V_{a,ref})^2 + ||\boldsymbol{\Gamma} - \boldsymbol{\Gamma}_{ref}||^2_{\boldsymbol{Q}_{\boldsymbol{\Gamma}}} \\
& + ||\boldsymbol{\omega} - \boldsymbol{\omega}_{ref}||^2_{\boldsymbol{Q}_{\boldsymbol{\omega}}} + ||\boldsymbol{u}||^2_{\boldsymbol{R}},
\end{aligned} \tag{21}$$

where $q_{V_a} > 0$ is the weighting scalar that penalizes the V_a error, and $\boldsymbol{Q}_{\boldsymbol{\Gamma}} \in \mathbb{R}^{3\times 3}$, $\boldsymbol{Q}_{\boldsymbol{\omega}} \in \mathbb{R}^{3\times 3}$ and $\boldsymbol{R} \in \mathbb{R}^{n_u \times n_u}$ are the symmetric and positive-definite weighting matrices for attitude error, angular velocity and input. The term $\frac{1}{2}\boldsymbol{s}^\top \boldsymbol{P}\boldsymbol{s}$ was added to penalize the vector of slack variables $\boldsymbol{s}$ with the symmetric, positive-definite matrix $\boldsymbol{P} \in \mathbb{R}^{n_s \times n_s}$. The vectors $\boldsymbol{x}(\cdot|t) \in \mathbb{R}^{n_x \cdot (N+1)}$ and $\boldsymbol{u}(\cdot|t) \in \mathbb{R}^{n_u \cdot N}$ are the predicted state and input sequence, respectively. The MPC scheme is based on solving the NLP given by (20) at time t. As the variables $[\delta_a, \delta_e, \delta_t]^\top$ are included in the state vector $\boldsymbol{x}$, the control input applied to the UAV is extracted from the optimal state after one shooting interval, given by $\boldsymbol{x}^*(1|t)$, obtained after solving the NLP:

$$\boldsymbol{u}_{\text{uav}}(t) = \begin{bmatrix} \boldsymbol{0}_{n_u \times (n_x - n_u)} & \boldsymbol{I}_{n_u \times n_u} \end{bmatrix} \boldsymbol{x}^*(1|t) \tag{22}$$

V. Simulation Results

Several simulations in varying ice configurations were performed in [11], including nominal airspeed ($V_a = 20m/s$),

TABLE I: Tuning parameters of the NMPC controllers.

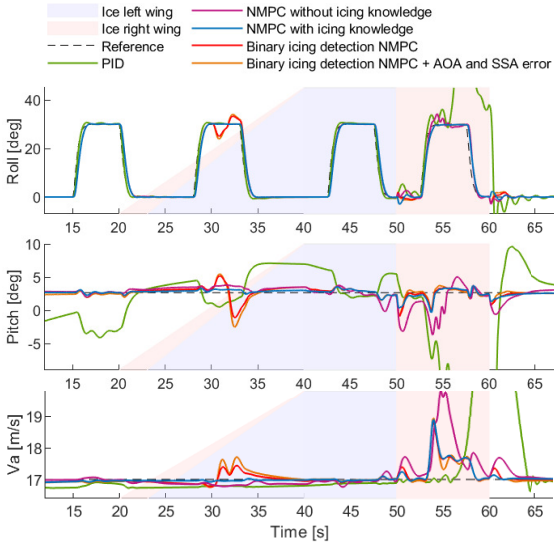
Parameters	NMPC w/o icing	NMPC w/ icing
$q_{V_a}, q_{\Gamma_1}, q_{\Gamma_2}, q_{\Gamma_3}$	0.1, 50, 200, 200	0.1, 100, 200, 200
$q_p, q_q, q_r, r_{\dot{\delta}_a}, r_{\dot{\delta}_e}, r_{\dot{\delta}_t}$	1, 1, 1, 0.2, 10, 0.08	1, 1, 1, 0.2, 1, 0.01
$p_{\underline{s}_{V_a}}, p_{\underline{s}_\beta}, p_{\overline{s}_{V_a}}, p_{\overline{s}_\beta}$	1, 1, 1, 1	1, 1, 1, 1
$l_{V_a}, l_\beta, l_\alpha, l_l, l_m, l_n$	0.03, 0.01, 0.01, 0.4, 0.1, 0.1	0.01, 0.01, 0.01, 0.4, 0.1, 0.1

relatively low airspeed ($V_a = 17m/s$), severe wind conditions, and path-following simulations. The relatively low airspeed scenario is presented here, being an interesting case as it maximizes endurance, and with results showing notable differences in tracking performance between the PID and the different NMPC controllers, with the PID controller being tuned for nominal airspeed, as described in [6].

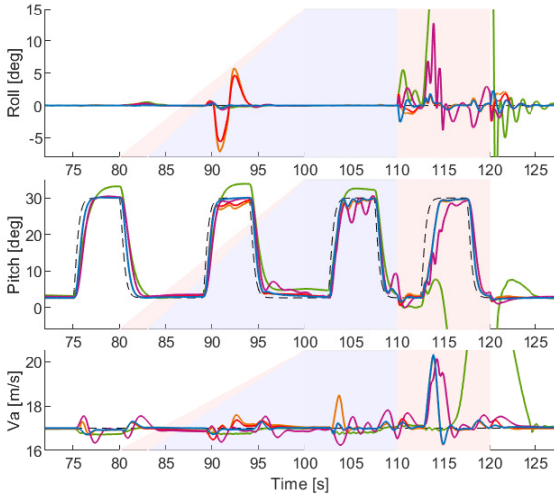
The UAV model and the NMPC controllers are simulated and tested in Matlab/Simulink. The NMPC is implemented in Python using the package acados [16], and interfaced with the simulator through an S-function, which uses C code generated from the Python program. The OCP with $T_h = 3.5\ s$ is discretized into $N = 35$ steps with shooting interval $\Delta t = 0.1s$, using direct multiple-shooting. This horizon was chosen to balance the computational load and the tracking performance of the NMPC. Stability and feasibility of the NMPC are not ensured by terminal constraints or costs, but by choosing a sufficiently long prediction horizon. This is often done in practice and discussed in [17]. To solve the NLP, real-time iteration sequential quadratic programming (RTI-SQP) is used with the high-performance interior-point method (HPIPM) [18] to solve the underlying quadratic problems.

First, the asymmetric icing model from Section III-A is included in the NMPC prediction model, where the simulated icing level on each wing $\zeta_{\text{left}}, \zeta_{\text{right}} \in [0, 1]$ is taken as a known input. This is referred to as the *NMPC with icing knowledge*. Second, to simulate the *NMPC without icing knowledge*, the icing level taken as an input is always 0, so the model only uses the nominal (no icing) aerodynamic coefficients. The tuning process is explained in [11], and the final tuning parameters used for the two NMPC controllers are given in Table I. In the simulator, the NMPC can have perfect information about the icing level simulated on each wing, but in a real-life situation it must be measured or estimated [19]. A third case is only being able to differentiate between clean and iced. This is considered by the *binary icing detection NMPC*, which is tuned as the NMPC with icing knowledge, but takes the icing level as 0 if it is under 50%, or 1 if it is over. In addition, the aerodynamic forces and moments are always dependent on AOA/SSA, which are usually difficult to estimate accurately for small UAVs. This can typically be done with an RMSE of around 0.58 deg [20] and 1.25 deg [21], respectively. To account for this, a fourth simulation case was included with the *binary icing detection NMPC including +1.5 deg offset error in AOA and SSA*.

Figures 1 and 2 show how the controllers performed under the relatively low airspeed simulation with different icing configurations, where the roll and pitch reference varied in the intervals $t \in (10,\ 70)s$ and $t \in (70,\ 130)s$, respectively. During the two intervals, the ice accretion on each wing is simulated until both wings are 100% iced, followed by a 100% asymmetric icing interval as the ice on the left wing sheds, and finally shedding of the ice on the right wing.



(a) Simulation during the interval $t \in (10, 70)s$.



(b) Simulation during the interval $t \in (70, 130)s$.

Fig. 1: Roll, pitch and airspeed tracking response of the controllers during the relatively low airspeed simulation.

The results in Figures 1 and 2 show performance degradation during the 100% asymmetry interval, especially in the case of the PID controller, whose throttle δ_t saturates to 0, as well as its aileron δ_a and elevator δ_e to -17.5 and -12.5 degrees, respectively. The NMPC without icing knowledge also struggles during this interval, especially while trying to keep roll and airspeed to their constant references while pitch changes to 30 degrees. Oscillations in the control signals and AOA can be seen, as well as oscillations in pitch during the 100% icing interval in Figure 1b. Adding icing knowledge to the model of the NMPC shows a clear improvement in the controller's performance during the icing and icing asymmetry intervals, where the oscillations are almost eliminated. In the case of the binary icing detection NMPC, the largest deviation is seen when the ice level on

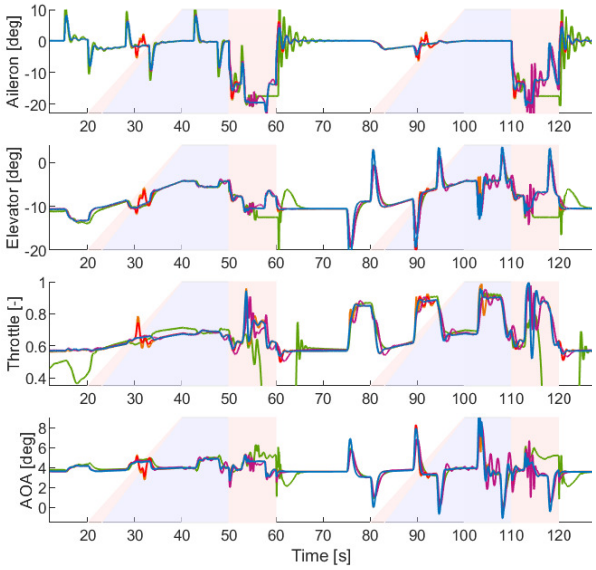


Fig. 2: Control signals and AOA of the PID and NMPC controllers throughout the relatively low airspeed simulation.

TABLE II: The IAE of the controllers in the simulation.

Controllers	IAE roll	IAE pitch	IAE Va
PID	9.6	15.5	82.2
NMPC without icing knowledge	1.6	4.0	29.8
NMPC with icing knowledge	1.3	2.0	13.3
Binary icing detection NMPC	1.6	2.3	16.4
Binary icing detection NMPC with AOA/SSA offset error	1.8	2.5	18.5

each wing exceeds 50%. This is because the ice level the controller predicts suddenly changes from 0 to 1. During the rest of the simulation, it is similar to the behavior of the NMPC with icing knowledge. Adding a $+1.5$ deg error in AOA and SSA to the binary icing detection NMPC worsens its performance slightly, and introduces some oscillations in δ_e and δ_t during the 100% icing interval. In Figure 3, it can be seen that this NMPC also requires the highest computation time during the icing and asymmetry intervals, while the NMPC with full icing knowledge usually requires the lowest.

To evaluate the performance of the controllers quantitatively, the Integral Absolute Error (IAE) of the roll, pitch and airspeed reference error are used, where the results are found in Table II and are consistent with the previously seen results.

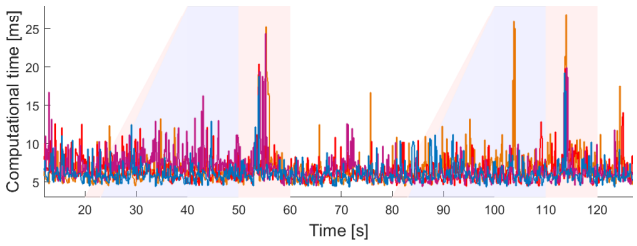


Fig. 3: Computation time of the NMPC controllers in the relatively low airspeed simulation.

VI. Discussion

The results show that icing asymmetry, when one of the wings remains iced while the ice sheds from the other, compromises the controllers' performance the most, as both the PID controller and the NMPC without icing knowledge had significant performance degradation during this interval. Although the mentioned NMPC can control the UAV in icing conditions, its performance and robustness are considerably improved by adding the icing model from Section III-A in its prediction model, reducing the oscillations in the control signals and giving a tighter roll and pitch tracking.

The propulsion icing model [5], however, was not added to the NMPC model. The model was developed for a different propeller than the one the Skywalker X8 uses, and although it was modified as explained in [11] to be valid in this case, many assumptions were made. The model also depends on variables the UAV might not have access to, such as the temperature and the LWC. Given its uncertainty, complexity and the good results obtained without it, implementing it in the NMPC model was not considered necessary, and it was only implemented in the Matlab/ Simulink simulator.

In less severe icing cases both the NMPC with and without icing knowledge perform well, as the disturbance observer handles modeling errors. This can be seen in the ice accretion intervals in the simulations. However, during these intervals the binary ice detection NMPCs have their worst performance, with observed transients in their response, as the ice level the controller sees suddenly changes from 0 to 1 when the ice level on the given wing surpasses 50%. After the initial transient, the icing level error is handled by the disturbance observer, and the binary icing detection controllers then perform similarly as in the 100% icing interval. This can be seen in Figure 4, where a) shows an ice accretion scenario similar to the one the binary icing detection NMPC detects in the reduced airspeed simulation, while b) shows the NMPC's behavior after the transient. This shows that the main cause for the performance degradation of the binary icing detection NMPCs is the sudden asymmetry the controller predicts in that instant, rather than the error between the real and the detected icing level. To make the transition from 0 to 1 icing level less abrupt, it was low-pass filtered. The filter was tuned to respond relatively quickly, but the transients can be further reduced by making it slower, as shown in plot c) in Figure 4. Adding possible AOA and SSA errors to the binary icing detection NMPC was not shown to significantly worsen its performance. The most visible difference is an increased airspeed during the 100% icing interval in Figure 1b, likely as a result of trying to keep the pitch reference.

The scenarios tested were rather extreme, as the iced coefficients correspond to the most severe icing type, and 100% icing asymmetry was simulated, although less severe asymmetry can also be expected. The binary icing detection NMPC with additional AOA/SSA errors was also implemented for this purpose, considering the worst-case scenario regarding icing detection and AOA/SSA estimation. This NMPC closely follows the response of the NMPC with icing knowledge, handling the icing and asymmetry intervals better

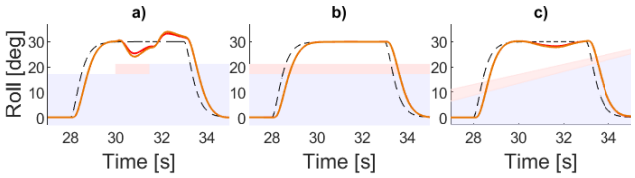


Fig. 4: Binary icing detection NMPCs in different ice accretion scenarios. The response is shown with a) similar icing as the one detected by the controllers in the original (low airspeed) simulation; b) constant icing (50% left wing, 60 % right wing), interpreted by the controllers as icing level 0 and 1; c) same icing as in the original simulation, but a slower low-pass filter to transition from icing level 0 to 1.

than the PID and the NMPC without icing knowledge. The low computation time, under 30 ms, indicates that real-time embedded implementation is feasible (see also [9]), and given the results as well as the discussed improvements during the ice accretion interval, the NMPC with icing knowledge is expected to perform well in practice.

Compared to the PID controller, the NMPC has the advantage of constraints in its formulation. This way, the stall angle of the UAV ($\alpha = 10$ deg [2]) can be taken into account. This was considered in this paper by setting $\overline{\alpha} = 8$ deg. Although the NMPC predictions always respect this limit, the actual AOA violates it at some point, possibly due to modeling inaccuracies. The upper limit for the AOA was set lower than the stall limit to prevent it from being violated, but finding a less conservative solution remains part of the future work.

VII. Conclusion

In this paper, we explore how an NMPC controller can be improved to be better suited for the control of a fixed-wing UAV operating in icing conditions, with icing asymmetry on the wings leading to the greatest performance degradation. The NMPC controller uses a disturbance observer to handle unmodelled effects and is improved by including an asymmetric icing model in its formulation. The NMPC with icing knowledge performs significantly better in a relatively low airspeed simulation than an NMPC without icing knowledge and a conventional PID controller, especially in the case of severe icing asymmetry. The NMPC with icing knowledge was further modified to simulate its expected behavior in conditions where we considered possible errors in AOA and SSA, as well as reduced information about the icing level on each wing (binary, i.e. either iced or clean). During asymmetric icing, it performed similarly to the NMPC with icing knowledge, and its worse transient performance during the ice accretion interval could be mitigated by filtering the abrupt jump in icing level input to the NMPC. As its computation time remains under 30 ms, the NMPC with icing knowledge is expected to run in practice with similar computational performance. This remains part of the future work, as well as finding a better way to ensure that the stall limit is kept.

Acknowledgment

The authors thank Nicolas Müller for his insight on propeller icing and his help in adapting the propeller icing model [5] to the propeller of the Skywalker X8.

References

[1] R. Hann and T. A. Johansen, "Unsettled Topics in Unmanned Aerial Vehicle Icing," *SAE International, SAE EDGE Research Report EPR2020008*, 2020.

[2] A. Winter, R. Hann, A. Wenz, K. Gryte, and T. A. Johansen, "Stability of a Flying Wing UAV in Icing Conditions," in *8th European Conference for Aeronautics and Space Sciences (EUCASS)*, 2019.

[3] 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)*. IEEE, 2018, pp. 826–835.

[4] R. Kleiven, K. Gryte, and T. A. Johansen, "Robust and Gain-Scheduled Flight Control of Fixed-Wing UAVs in Wind and Icing Conditions," *IEEE Aerospace Conference*, 2021.

[5] N. Müller and R. Hann, "UAV Icing: A Performance Model for a UAV Propeller in Icing Conditions," *AIAA Atmospheric and Space Environments Conference*, 06 2022.

[6] S. Högnadottir, K. Gryte, R. Hann, and T. A. Johansen, "Inner-Loop Control of Fixed-Wing Unmanned Aerial Vehicles in Icing Conditions," *AIAA Sci-Tech Conference*, 2023.

[7] T. Stastny and R. Siegwart, "Nonlinear Model Predictive Guidance for Fixed-wing UAVs Using Identified Control Augmented Dynamics," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2018, pp. 432–442.

[8] P. Oettershagen, A. Melzer, S. Leutenegger, K. Alexis, and R. Siegwart, "Explicit model predictive control and L1-navigation strategies for fixed-wing UAV path tracking," in *22nd Mediterranean Conference on Control and Automation*, 2014, pp. 1159–1165.

[9] D. Reinhardt and T. A. Johansen, "Control of Fixed-Wing UAV Attitude and Speed based on Embedded Nonlinear Model Predictive Control," *IFAC-PapersOnLine*, vol. 54, no. 6, pp. 91–98, 2021, 7th IFAC Conference on Nonlinear Model Predictive Control (NMPC).

[10] ——, "Nonlinear Model Predictive Control combined with Geometric Attitude and Speed Control for Fixed-Wing UAVs," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2021, pp. 465–475.

[11] N. A. Ghindaoanu, *Control of Fixed-Wing UAVs in Icing Conditions Using Nonlinear Model Predictive Control*. M.S. thesis, Norwegian University of Science and Technology, 2023.

[12] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons, 2015.

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

[14] E. M. Coates, A. Wenz, K. Gryte, and T. A. Johansen, "Propulsion System Modeling for Small Fixed-Wing UAVs," in *Int. Conf. Unmanned Aircraft Systems (ICUAS)*, 2019, pp. 748–757.

[15] D. Reinhardt, E. M. Coates, and T. A. Johansen, "Hybrid Control of Fixed-Wing UAVs for Large-Angle Attitude Maneuvers on the Two-Sphere," vol. 53, no. 2, pp. 5717–5724, 21st IFAC World Congress.

[16] R. Verschueren, G. Frison, D. Kouzoupis, J. Frey, N. van Duijkeren, A. Zanelli, B. Novoselnik, T. Albin, R. Quirynen, and M. Diehl, "acados—a modular open-source framework for fast embedded optimal control," pp. 147–183.

[17] L. Grüne and J. Pannek, "Nonlinear Model Predictive Control," in *Nonlinear Model Predictive Control: Theory and Algorithms*, L. Grüne and J. Pannek, Eds. Springer International Publishing, pp. 45–69.

[18] G. Frison and M. Diehl. HPIPM: A high-performance quadratic programming framework for model predictive control.

[19] B. Løw-Hansen, R. Hann, B. N. Stovner, and T. A. Johansen, "UAV Icing: A survey of recent developments in ice detection methods," *IFAC World Congress, Yokohama*, 2023.

[20] 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.

[21] 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.