Nonlinear Model Predictive Attitude Control for Fixed-Wing Unmanned Aerial Vehicle based on a Wind Frame Formulation

Dirk Reinhardt and Tor Arne Johansen

Abstract— **Exploiting the physical limitations on the maneuverability of a fixed-wing Unmanned Aerial Vehicle (UAV) and simultaneously respecting its flight envelope and actuator constraints is a demanding task, for which Nonlinear Model Predictive Control (NMPC) is well-suited. This paper presents an NMPC for attitude control of a fixed-wing UAV, which is based on the vehicle model in the wind-frame formulation and includes critical flight variables such as relative speed and angle of attack in its control objective and constraint formulation. The proposed controller is evaluated in a simulation study and compared against a set of conventional PID controllers.**

I. Introduction

Nonlinear Model Predictive Control (NMPC) is based on repetitively solving open-loop optimal control problems to obtain a sequence of control inputs that, when applied, makes the plant converge to the desired state. The obtained solution is optimal in the ideal scenario in which the the mathematical model of the process is a perfect description of its physical reality. Employing NMPC for attitude control is established in spacecraft applications, see for example [1], [2], [3]. A more recent development uses a geometric formulation and designs the attitude controller directly on SO(3) [4].

Regarding NMPC for attitude control on Unmanned Aerial Vehicles (UAVs), most work is done on multicopters. In [5] the authors propose a cascaded controller with a low-level NMPC for attitude control on a multicopter. Employing geometrically exact integration (cf. [6]) they design the controller on SO(3) and in experiments show computation times averaging to 1 millisecond, using a rather powerful hardware setup. More recently, [7] designed an NMPC for attitude controller on an embedded system with limited computational resources on a multicopter. The authors employ a partially tightened real-time iteration scheme [8] to achieve 10 ms control intervals on an 800 MHz processor.

Looking at fixed-wing UAVs however, most NMPC publications focus on high-level control strategies while relying on attitude references to be tracked by low-level attitude control loops as part of an off-the-shelf autopilot. Early work on fixed-wing UAV attitude control using NMPC is presented in [9]. The authors design an NMPC that takes the general nonlinear dynamics equations of an aircraft which are non-affine in the control inputs. They expand output and control variables in a truncated Taylor series to obtain a closed-form solution for the control inputs. This makes it applicable to real-time embedded systems, given that most of the gain matrices may be computed offline. However, actuator usage and trajectory errors are penalized through the cost function, but actuator and state constraints are not included in this formulation, which means that we can not include the operational envelope in the form of inequality constraints in the Optimal Control Problem (OCP).

In [10] the authors present an NMPC for path-following control that assumes a low-level attitude control system and requires longitudinal velocity, turn rate and desired altitude as input signals. Using a desired trajectory, they define cross track error and yaw angle error to transform the path-following problem into a regulation problem. They show closed-loop stability based on conditions formulated via linear matrix inequalities. In simulation convergence to a desired direction has been achieved as long as the initial direction is within 90 degrees.

A recent work is presented by [11] where the authors achieve 3D path-following. With a low-level attitude control system in place, they identify second order models that are then being used by the NMPC to design attitude references. Docile maneuvers are assumed to yield tractable references for the attitude controller. However, the controller is later challenged with curved paths that exceed the physical limit of the UAV (30 degree roll angles), which are shown to be followed within 0.5 meter and 2 meter error in longitudinal and lateral direction, respectively. In [12] the authors focus on waypoint and path-planning to achieve collision avoidance of a system of multiple cooperating UAVs in an environment that is subject to atmospheric disturbances and physical obstacles. Work in [13] uses NMPC for guidance of a fixed-wing UAV in deep stall landing maneuvers which expose the UAV to high angles of attack. Trajectory tracking via fast NMPC is done in [14].

Through the OCP formulation, NMPCs are very reconfigurable, which can be made use of in the design of fault-tolerant aircraft systems, as in e.g. [15].

In the remainder of this section we will motivate the use of NMPC for the attitude control problem over conventional methods before discussing some mathematical preliminaries. Based on these, in Sec. II we will look at the six degrees of freedom dynamic models of the UAV in both body-fixed and wind axes.

This work has been conducted with support by AMOS (grant number 223254) and the Research Council of Norway (project Autofly, 261791/O70).

Dirk Reinhardt and Tor Arne Johansen are with the Centre for Autonomous Marine Operations and Systems (AMOS), Department of Engineering Cybernetics, Norwegian University of Science and Technology, NO-7491 Trondheim, Norway
Corresponding author: dirk.p.reinhardt@ntnu.no

The purpose of the first model is to use it in simulations whereas the latter is directly entering the NMPC together with the attitude error function introduced in Sec. III which will be included in the the OCP and the derived Nonlinear Program (NLP). In order to evaluate the controller design, we will discuss a simulation study in Sec. IV and Sec. V where we compare the NMPC against convential PID controllers, before finishing with some concluding remarks in Sec. VI.

A suitable formulation of the NMPC scheme to enforce constraints on the relative velocity vector, specifically the angle of attack and relative airspeed, through the state vector of the vehicle together with a decoupled weighting of the roll dynamics using quaternions as attitude representation is the original contribution of this paper.

A. Motivation

Conventional attitude control methods that can be found in the literature usually assume decoupled dynamics that are being controlled separately in lateral and longitudinal direction (cf. [16], [17]). In the decoupled approach, airspeed and pitch angle are controlled via a longitudinal control system and roll and yaw angle through a lateral control system. These are usually based on the successive loop-closure of PID control loops that are tuned based on desired damping ratio and bandwidth of canonical transfer functions of first or second order. This procedure typically limits the bandwidth of each loop to a maximum of 10%-20% of the preceding lower-level loop. Moreover, they are usually tuned for a limited set of trim conditions, which poses restrictions on the flight-envelope that the UAV is capable of operating in. A turn maneuver in the horizontal plane is then conducted in a coordinated turn for slip compensation, in which the turning-rate is controlled through the roll channel dynamics (bank-to-turn). This makes for a computationally efficient and well studied control system in which performance and robustness can be directly related to the design process. However, it comes at the price of conservative maneuverability and does not explicitly take actuator saturation into account.

In contrast, an NMPC may use the full nonlinear dynamics in order to achieve a higher-level control objective without bandwidth limitations imposed through the design process. In fact, using an NMPC scheme, the UAV may be operated at its physical limits and thus exploit the full flight-envelope. In UAVs with redundant actuators, the control allocation problem is solved as a by-product. One simple example would be to increase the relative airspeed to obtain a higher control authority of the ailerons, which results in an increased roll rate and at the end a faster convergence to the desired attitude in a turn maneuver. However, this comes at the cost of a significantly higher demand for computational resources which results in the controller not being usable in real-time applications. This justifies the dominant use of PID controllers in state of the art autopilots where high update rates are required. Attitude NMPCs can however be suitable as an offline tool to explore a range of feasible attitude maneuvers under actuator and state constraints, and thus to assess performance of other nonlinear controllers that are less computationally expensive but yield suboptimal results.

The idea of the proposed work is to design an NMPC to control the relative speed together with yaw and pitch angle, which is often an intermediate step to a higher-level control objective such as path-following, shown in Fig. 1. The current estimate of the static component of the wind will be taken into account as a disturbance which can be compensated for through the model formulation. Other path-following algorithms use a reference velocity vector represented by ground speed, course angle and flight-path angle. With knowledge of the wind-velocity vector in the inertial frame these can be converted to the desired airspeed, yaw and pitch angle (cf. [17]), which then enter the OCP.

The controller model will be formulated in the wind frame and thus includes angle of attack, side-slip and relative speed. Therefore hard constraints may be imposed to prevent stall while simultaneously optimize performance that is subject to actuator constraints (limits on aileron, elevator, rudder and throttle). In the same manner, we can penalize side-slip. As stated earlier, controlling the roll angle is often used as an intermediate step to control turn-rate or yaw angle to a desired reference generated by a higher-level path-following controller. The formulation of the NMPC allows for defining the desired yaw angle in the control objective directly, thus incorporating the control of the needed roll angle into the optimization problem. As we include the side-slip angle in the state vector, it is controlled by the controller and high side-slip angles can be penalized through the definition of the OCP. This makes it possible to drop the enforcement of the coordinated turn kinematics.

B. Notation and Preliminaries

Throughout this paper, vectors will be denoted by bold letters. The Euclidean norm on $\mathbb{R}^n$ is given by $\|x\| := \sqrt{\boldsymbol{x}^\top \boldsymbol{x}}$. The identity matrix of dimension $n \times n$ will be denoted by $\boldsymbol{I}_{n\times n}$. For any $\boldsymbol{x} = [x_1, x_2, x_3]^\top$, the skew-symmetric matrix-valued function $\boldsymbol{S} : \mathbb{R}^3 \mapsto \mathbb{R}^{3\times 3}$ is given by

$$\boldsymbol{S}(\boldsymbol{x}) = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}. \tag{1}$$

The vector cross product for two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^3$ can be written in matrix form $\boldsymbol{x} \times \boldsymbol{y} = S(\boldsymbol{x})\boldsymbol{y}$, where $\times$ denotes the cross product operator. For brevity, the time argument may be dropped where it is clear from context.

We represent attitude with a unit quaternion

$$\boldsymbol{q} = \begin{bmatrix} \eta & \boldsymbol{\epsilon}^\top \end{bmatrix}^\top \in \mathcal{S}^3, \tag{2}$$

which is decomposed into its scalar part η and vector part $\boldsymbol{\epsilon} \in \mathbb{R}^3$. The quaternion space is the three-sphere. The general n-sphere is given by

$$\mathcal{S}^n = \{\boldsymbol{x} \in \mathbb{R}^{n+1} \ : \ \boldsymbol{x}^\top \boldsymbol{x} = 1\}. \tag{3}$$

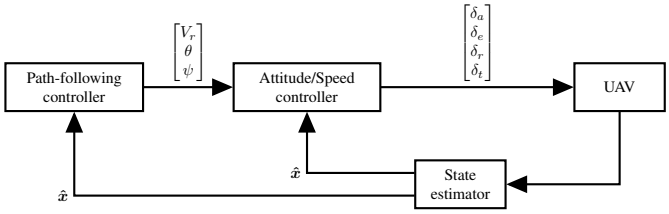

Fig. 1. Block diagram of the guidance, navigation and control system of a UAV. Input to the Attitude and Speed controller are references on relative speed, pitch and yaw angle from the path-following controller to achieve a higher-order control objective, which may include values ahead in time. In addition to the estimated states the controller uses this information to determine control surface deflections and throttle position.

The inverse of a unit quaternion is given by its conjugate $\bar{\boldsymbol{q}}$ divided by its Euclidean norm

$$\boldsymbol{q}^{-1} = \frac{\bar{\boldsymbol{q}}}{\|\boldsymbol{q}\|}, \qquad \bar{\boldsymbol{q}} = \begin{bmatrix} \eta & -\boldsymbol{\epsilon}^\top \end{bmatrix}^\top . \tag{4}$$

The special orthogonal group $SO(3)$ denotes the set of rotation matrizes with dimension 3×3 and is given by

$$SO(3) = \{\boldsymbol{R} \in \mathbb{R}^{3\times 3} : \boldsymbol{R}^\top \boldsymbol{R} = \boldsymbol{R}\boldsymbol{R}^\top = \boldsymbol{I}_{3\times 3},\, \det \boldsymbol{R} = 1\} . \tag{5}$$

A parametrization of an element of SO(3) by a unit quaternion $\boldsymbol{R} : \mathcal{S}^3 \mapsto SO(3)$ is given by [18]

$$\boldsymbol{R}(\boldsymbol{q}) = (\eta^2 - \|\boldsymbol{\epsilon}\|^2)\boldsymbol{I}_{3\times 3} + 2\eta \boldsymbol{S}(\boldsymbol{\epsilon}) + 2\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top \tag{6}$$

which can also be expressed as the product of two matrices

$$\boldsymbol{R}(\boldsymbol{q}) = \boldsymbol{\Xi}(\boldsymbol{q})^\top \boldsymbol{\Psi}(\boldsymbol{q}) \tag{7}$$

where $\boldsymbol{\Xi}(\boldsymbol{q})$ and $\boldsymbol{\Psi}(\boldsymbol{q})$ are 4×3 matrizes given by

$$\boldsymbol{\Xi}(\boldsymbol{q}) = \begin{bmatrix} -\boldsymbol{\epsilon}^\top \\ \eta \boldsymbol{I}_{3\times 3} - \boldsymbol{S}(\boldsymbol{\epsilon}) \end{bmatrix}, \quad \boldsymbol{\Psi}(\boldsymbol{q}) = \begin{bmatrix} -\boldsymbol{\epsilon}^\top \\ \eta \boldsymbol{I}_{3\times 3} + \boldsymbol{S}(\boldsymbol{\epsilon}) \end{bmatrix} . \tag{8}$$

The product of two quaternions $\boldsymbol{q}_1 = [\eta_1,\ \boldsymbol{\epsilon}_1]^\top$, $\boldsymbol{q}_2 = [\eta_2,\ \boldsymbol{\epsilon}_2]^\top$ is defined as

$$\boldsymbol{q}_1 \otimes \boldsymbol{q}_2 \triangleq \begin{bmatrix} \eta_1\eta_2 - {\boldsymbol{\epsilon}_1}^\top \boldsymbol{\epsilon}_2 \\ \eta_1\boldsymbol{\epsilon}_2 + \eta_2\boldsymbol{\epsilon}_1 + \boldsymbol{S}(\boldsymbol{\epsilon}_1)\boldsymbol{\epsilon}_2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{q}_1 & \boldsymbol{\Psi}(\boldsymbol{q}_1) \end{bmatrix} \boldsymbol{q}_2 . \tag{9}$$

Given the angular velocity vector $\boldsymbol{\omega} \in \mathbb{R}^3$, a kinematic equation of the quaternion can be given by

$$\dot{\boldsymbol{q}} = \frac{1}{2}\boldsymbol{\Psi}(\boldsymbol{q})\boldsymbol{\omega} . \tag{10}$$

II. Dynamic Model

The aim of this section is to describe the dynamic and kinematic equations of the vehicle. First the model in body-fixed frame will be presented which will be used in the forward-propagation in the simulation study. Then a formulation based in the wind frame will be presented which will be part of the OCP.

A. Vehicle Model

We model position, velocity and attitude (PVA) in the North-East-Down (NED) frame $\{n\}$. The state $\boldsymbol{x} \in \mathbb{R}^{13}$ and input vector $\boldsymbol{u} \in \mathbb{R}^4$ of the complete model in the body-fixed frame are given by

$$\boldsymbol{x} = \begin{bmatrix} {\boldsymbol{p}_{nb}^n}^\top & {\boldsymbol{q}_b^n}^\top & {\boldsymbol{v}_{nb}^n}^\top & {\boldsymbol{\omega}_{nb}^b}^\top \end{bmatrix}^\top \tag{11a}$$

$$\boldsymbol{u} = \begin{bmatrix} \delta_a & \delta_e & \delta_r & \delta_t \end{bmatrix}^\top , \tag{11b}$$

where $\boldsymbol{p}_{nb}^n, \boldsymbol{v}_{nb}^n \in \mathbb{R}^3$ are the position and linear velocity of the vehicle's body-fixed frame $\{b\}$ relative to $\{n\}$ decomposed in NED. The quaternion $\boldsymbol{q}_b^n \in \mathcal{S}^3$ represents the orientation of the body-fixed frame relative to NED, and $\boldsymbol{\omega}_{nb}^b \in \mathbb{R}^3$ is the angular velocity vector decomposed in the body-fixed frame that describes the angular velocity of the body-fixed frame relative to the NED frame. The input vector includes the deflections of the ailerons $\delta_a \in [\delta_{a,\min},\ \delta_{a,\max}]$, elevators $\delta_e \in [\delta_{e,\min},\ \delta_{e,\max}]$, rudder $\delta_r \in [\delta_{r,\min},\ \delta_{r,\max}]$ and the throttle position $\delta_t \in [0,\ 1]$.

We assume the NED frame to be inertial which yields the following kinematic and dynamic equations for the vehicle model:

$$\dot{\boldsymbol{p}}_{nb}^n = \boldsymbol{R}(\boldsymbol{q}_b^n)\boldsymbol{v}_{nb}^b \tag{12a}$$

$$\dot{\boldsymbol{q}}_b^n = \frac{1}{2}\boldsymbol{\Psi}(\boldsymbol{q}_b^n)\boldsymbol{\omega}_{nb}^b \tag{12b}$$

$$\dot{\boldsymbol{v}}_{nb}^b = \frac{1}{\mathrm{m}}(\boldsymbol{F}_A^b + \boldsymbol{F}_T^b) + \boldsymbol{R}(\boldsymbol{q}_b^n)^\top \boldsymbol{g}^n - \boldsymbol{S}(\boldsymbol{\omega}_{nb}^b)\boldsymbol{v}_{nb}^b \tag{12c}$$

$$\dot{\boldsymbol{\omega}}_{nb}^b = \boldsymbol{I}_b^{-1}(-\boldsymbol{S}(\boldsymbol{\omega}_{nb}^b)\boldsymbol{I}_b\boldsymbol{\omega}_{nb}^n + \boldsymbol{M}_A^b) . \tag{12d}$$

The model includes the mass of the vehicle $\mathrm{m} \in \mathbb{R}$, the gravity vector in NED $\boldsymbol{g}^n = [0,\ 0,\ g]^\top \in \mathbb{R}^3$ and the matrix of inertia in the body-fixed frame $\boldsymbol{I}_b \in \mathbb{R}^{3\times 3}$ given by

$$\boldsymbol{I}_b = \begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{xz} & 0 & I_z \end{bmatrix} . \tag{13}$$

The aerodynamic and thrust forces are represented by $\boldsymbol{F}_A^b, \boldsymbol{F}_T^b \in \mathbb{R}^3$ and the aerodynamic moments are given by $\boldsymbol{M}_A^b \in \mathbb{R}^3$, which are functions of the velocity of the UAV relative to the surrounding air mass, its angular velocity, control surface deflections and throttle position. To compute them, the relative velocity needs to be expressed in terms of its magnitude $V_r \in \mathbb{R}$, angle of attack $\alpha \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ and side-slip angle $\beta \in [-\pi, \pi]$, which are given by the equations

$$\boldsymbol{v}_r = \begin{bmatrix} u_r & v_r & w_r \end{bmatrix}^\top = \boldsymbol{v}_{nb}^b - \boldsymbol{R}(\boldsymbol{q}_b^n)^\top \boldsymbol{w}^n \tag{14a}$$

$$V_r = \|\boldsymbol{v}_r\|_2 \tag{14b}$$

$$\alpha = \arctan(w_r/u_r) \tag{14c}$$

$$\beta = \arcsin(v_r/V_r) \tag{14d}$$

with $\boldsymbol{w}^n = [w_n,\ w_e,\ w_d]^\top \in \mathbb{R}^3$ as the wind velocity vector in NED.

The transformation from the body-fixed frame to the stability frame $\{s\}$ is a rotation in the body-fixed frame

around the y-axis by the angle of attack

$$\boldsymbol{R}_b^s(\alpha) = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix}, \tag{15}$$

and the transformation from the stability frame to the wind frame $\{w\}$ is a rotation around the z-axis of the stability frame by the side-slip angle

$$\boldsymbol{R}_s^w(\beta) = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{16}$$

The transformation from the body-fixed frame to the wind frame is then given by $\boldsymbol{R}_b^w(\alpha,\beta) = \boldsymbol{R}_s^w(\beta)\boldsymbol{R}_b^s(\alpha)$.

With these transformations we can use V_r, α and β to express the relative velocity vector in terms of its direction and magntitude as $\boldsymbol{v}_r = \boldsymbol{R}_b^w(\alpha,\beta)^\top [V_r,\ 0,\ 0]^\top$.

We model the moment vector in body-fixed components $\boldsymbol{M}_A^b$ as

$$\boldsymbol{M}_A^b = \frac{1}{2}\rho V_r^2 S \begin{bmatrix} bC_l(V_r,\alpha,\beta,p,r,\delta_a,\delta_r) \\ cC_m(V_r,\alpha,q,\delta_e) \\ bC_n(V_r,\alpha,\beta,p,r,\delta_a,\delta_r) \end{bmatrix} \tag{17}$$

where $\rho \in \mathbb{R}$ is the air density and $b,\ c,\ S \in \mathbb{R}$ are the span, chord and surface area of the wings, respectively.

The model for the thrust force vector is given as a function of relative speed V_r and throttle position δ_t acting along the body-fixed x-axis

$$\boldsymbol{F}_T^b = \begin{bmatrix} T(V_r,\delta_t) \\ 0 \\ 0 \end{bmatrix} \tag{18}$$

with the thrust model

$$T(V_r,\delta_t) = \frac{1}{2}\rho S_{\text{prop}} C_{\text{prop}} V_{\text{dis}}(V_{\text{dis}} - V_r) \tag{19a}$$

$$V_{\text{dis}} = V_r + \delta_t(k_{\text{motor}} - V_r)\,. \tag{19b}$$

The parameters in the model are the washout area and the chord of the propeller $S_{\text{prop}},\ C_{\text{prop}} \in \mathbb{R}$ together with a motor-specific constant $k_{\text{motor}} \in \mathbb{R}$.

The model of the aerodynamic force vector in the wind frame is given by

$$\boldsymbol{F}_A^w = \begin{bmatrix} -D \\ Y \\ -L \end{bmatrix} = \frac{1}{2}\rho V_r^2 S \begin{bmatrix} C_D(V_r,\alpha,\beta,q,\delta_e) \\ C_Y(V_r,\beta,p,r,\delta_a,\delta_r) \\ C_L(V_r,\alpha,q,\delta_e) \end{bmatrix} \tag{20}$$

and needs to be transformed to the body-fixed frame by using the equation $\boldsymbol{F}_A^b = \boldsymbol{R}_b^w(\alpha,\beta)^\top \boldsymbol{F}_A^w$.

B. Vehicle Model in Stability and Wind axes

While we use the previous model for simulation, we will formulate the OCP based on the model expressed in the stability frame with different dynamic equations. The state vector $\boldsymbol{x} \in \mathbb{R}^{10}$ of this model is

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{q}_b^{n\top} & V_r & \alpha & \beta & \boldsymbol{\omega}_{nb}^{s\top} \end{bmatrix}^\top \tag{21}$$

where $\boldsymbol{q}_{nb}^n \in \mathcal{S}^3$ again represents the attitude. In this model we represent velocity in the wind frame using the relative speed, angle of attack and side-slip angle as defined in Sec. II-A. Note that the angular velocity vector is rotated to the stability frame, i.e $\boldsymbol{\omega}_{nb}^s = \boldsymbol{R}_b^s(\alpha)\boldsymbol{\omega}_{nb}^b = [p_s,\ q,\ r_s]$. The state vector dues not include the position of the vehicle since it is not needed in the attitude and speed controller. The input vector $\boldsymbol{u}$ is the same as in the body-fixed frame given in (11). The kinematic and dynamic equations follow from the derivations in [16], which yield

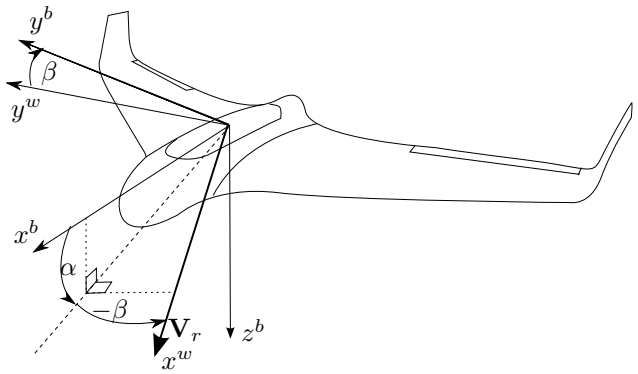

Fig. 2. Drawing of the Skywalker X8 including body-fixed axes $\{x^b, y^b, z^b\}$ and wind axes $\{x^w, y^w, z^w\}$, as well as the angle of attack α and side-slip angle β. The relative velocity vector $\boldsymbol{v}_r$ is aligned with x^w. With courtesy of [19].

$$\dot{\boldsymbol{q}}_b^n = \frac{1}{2}\boldsymbol{\Psi}(\boldsymbol{q}_b^n)\boldsymbol{R}_b^s(\alpha,\beta)^\top \boldsymbol{\omega}_{nb}^s \tag{22a}$$

$$\dot{V}_r = \frac{T\cos\alpha\cos\beta - D + \text{m}g_1}{\text{m}} \tag{22b}$$

$$\dot{\beta} = \frac{T\cos\alpha\sin\beta + Y + \text{m}g_2}{\text{m}V_r} - r_s \tag{22c}$$

$$\dot{\alpha} = \frac{-T\sin\alpha - L + \text{m}g_3}{\text{m}\cos\beta V_r} - p_s\tan\beta + q \tag{22d}$$

$$\dot{\boldsymbol{\omega}}_{nb}^s = (\boldsymbol{I}^s)^{-1}(-\boldsymbol{S}(\boldsymbol{\omega}_{nb}^s)\boldsymbol{I}^s\boldsymbol{\omega}_{nb}^s + \boldsymbol{R}_b^s(\alpha)\boldsymbol{M}^b) - \boldsymbol{S}(\boldsymbol{\omega}_{bs}^s)\boldsymbol{\omega}_{nb}^s \tag{22e}$$

where $\boldsymbol{g}^w$ denotes the gravity vector in the wind frame and is given by

$$\boldsymbol{g}^w = \begin{bmatrix} g_1, & g_2, & g_3 \end{bmatrix}^\top = \boldsymbol{R}_w^b(\alpha,\beta)\boldsymbol{R}(\boldsymbol{q}_b^n)^\top \boldsymbol{g}^n\,. \tag{22}$$

The inertia matrix I^s is given by a similarity transformation applied to I^b resulting in

$$I^s = \boldsymbol{R}_b^s(\alpha) I^b \boldsymbol{R}_b^s(\alpha)^\top = \begin{bmatrix} I_x^s & 0 & -I_{xz}^s \\ 0 & I_y^s & 0 \\ -I_{xz}^s & 0 & I_z^s \end{bmatrix} \tag{23}$$

and with the individual components given by

$$I_x^s = I_x\cos^2\alpha + I_z\sin^2\alpha - I_{xz}\sin 2\alpha \tag{24a}$$

$$I_y^s = I_y \tag{24b}$$

$$I_z^s = I_x\sin^2\alpha + I_z\cos^2\alpha + I_{xz}\sin 2\alpha \tag{24c}$$

$$I_{xz}^s = \frac{1}{2}(I_x - I_z)\sin 2\alpha + I_{xz}\cos 2\alpha\,. \tag{24d}$$

III. Speed/Attitude Controller

In this section, we will first look at the design of the attitude error function which will be included in the cost function of the OCP from which we will then derive the NLP.

The control objective is to steer the attitude towards a desired yaw and pitch angle given by ψ_d and θ_d while simultaneously controlling the relative speed to a desired reference $V_{r,d}$. In order to weigh errors in pitch and yaw direction independently from the roll angle, we need to represent the attitude error in a reduced attitude form which is invariant to changes in the roll angle. By doing this, we can treat reduction of yaw and pitch errors as a primary objective and the roll angle reduction as an intermediate secondary objective. Physically, this means that we assign a higher priority to controlling the direction of the longitudinal axis of the vehicle while treating the rotation around this axis separately. We will look at the reduced attitude representation in the following before we discuss inequality constraints and the OCP.

A. Reduced Attitude Control

Given a unit vector $\boldsymbol{b} \in \mathbb{R}^3$ which represents an arbitrary axis in NED, we define a reduced attitude in the body-fixed frame [20] through a projection $\boldsymbol{\Gamma} : \mathcal{S}^3 \times \mathbb{R}^3 \to \mathcal{S}^2$ given by

$$\boldsymbol{\Gamma}(\boldsymbol{q}, \boldsymbol{b}) = \boldsymbol{R}(\boldsymbol{q})^\top \boldsymbol{b}\,. \tag{25}$$

The reduced attitude is invariant to rotations about $\boldsymbol{b}$ which we make use of when composing the cost function. Now we use the reduced attitude to compose an attitude error function between the desired reduced attitude and the reduced attitude of the vehicle. We assume a desired attitude represented by a quaternion $\boldsymbol{q}_d$ which is parametrized using ψ_d, θ_d and an arbitrary roll angle. The first part of the attitude error function $\Phi_1 : \mathcal{S}^3 \times \mathcal{S}^3 \times \mathbb{R}^3 \to \mathbb{R}$ is defined by the equation

$$\Phi_1(\boldsymbol{q}, \boldsymbol{q}_d, \boldsymbol{b}) = 1 - \boldsymbol{\Gamma}(\boldsymbol{q}_d, \boldsymbol{b})^\top \boldsymbol{\Gamma}(\boldsymbol{q}, \boldsymbol{b}) \tag{26}$$

This equation is equivalent to $\Phi_1(\boldsymbol{q}, \boldsymbol{q}_d, \boldsymbol{b}) = 1 - \cos\vartheta$ where ϑ is the angle between $\boldsymbol{\Gamma}(\boldsymbol{q}_d, \boldsymbol{b})$ and $\boldsymbol{\Gamma}(\boldsymbol{q}, \boldsymbol{b})$ using the fact that $\|\boldsymbol{\Gamma}(\boldsymbol{q}_d, \boldsymbol{b})\| = \|\boldsymbol{\Gamma}(\boldsymbol{q}, \boldsymbol{b})\| = 1$. We see that that the there exists a global minimum at $\vartheta = 0$, where the desired reduced attitude coincides with the reduced attitude of the vehicle.

In the definition of the OCP in the next section, we will use the weighted squared norm of the output of the attitude error function. With only Φ_1 as attitude error function, this would lead to small gradients close to the minimum and may result in a slow convergence to the solution to the NLP when the attitude is close to its reference. In order to overcome this issue, we extend the attitude error function with the cross product term between the reduced attitudes

$$\boldsymbol{\Phi}_2(\boldsymbol{q}, \boldsymbol{q}_d, \boldsymbol{b}) = \boldsymbol{S}(\boldsymbol{\Gamma}(\boldsymbol{q}_d, \boldsymbol{b}))\boldsymbol{\Gamma}(\boldsymbol{q}, \boldsymbol{b}) \tag{27}$$

and concatenate as $\boldsymbol{\Phi} = \left[\Phi_1, {\boldsymbol{\Phi}_2}^\top\right]^\top$.

To show the increased magnitude of the gradient around the minimum, we introduce $\boldsymbol{n}$ as a unit vector that is orthogonal to $\boldsymbol{\Gamma}(\boldsymbol{q}_d, \boldsymbol{b})$ and $\boldsymbol{\Gamma}(\boldsymbol{q}, \boldsymbol{b})$. By the definition of the cross product it then follows that an equivalent expression of (27) is given by $\boldsymbol{\Phi}_2(\boldsymbol{q}, \boldsymbol{q}_d, \boldsymbol{b}) = \sin(\vartheta)\boldsymbol{n}$. For the extended attitude error function $\boldsymbol{\Phi}$ and the attude error function Φ_1, their squared magnitudes are $\boldsymbol{\Phi}^\top\boldsymbol{\Phi} = -2\cos\vartheta$ and $\Phi_1 = (1 - \cos\vartheta)^2$. The resulting inequality

$$\frac{d\boldsymbol{\Phi}^\top\boldsymbol{\Phi}}{d\vartheta} = 2\sin\vartheta \geq 2\sin\vartheta(1 - \cos\vartheta) = \frac{d\Phi_1^2}{d\vartheta} \tag{28}$$

is true for $\vartheta \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ which shows steeper gradient for $\boldsymbol{\Phi}^\top\boldsymbol{\Phi}$ around the minimum.

When we use desired pitch angle θ_d and yaw angle ψ_d to parametrize the desired direction of the vehicle x-axis $\boldsymbol{x}_b^n$ given by

$$\boldsymbol{x}_d^n(\theta_d, \psi_d) = \left[\cos\theta_d\cos\psi_d, \cos\theta_d\sin\psi_d, -\sin\theta_d\right]^\top, \tag{29}$$

then as a result the desired reduced attitude simplfies to $\boldsymbol{\Gamma}(\boldsymbol{q}_d, \boldsymbol{x}_d^n) = \boldsymbol{x}_b^b = \left[1, 0, 0\right]^\top$. Injecting this result into (26) and (27), we see that they reduce to

$$\tilde{\Phi}_1(\boldsymbol{q}, \psi_d, \theta_d) = 1 - \left[1, 0, 0\right]\boldsymbol{\Gamma}(\boldsymbol{q}, \boldsymbol{x}_d^n) \tag{30}$$

and

$$\begin{aligned}\tilde{\boldsymbol{\Phi}}_2(\boldsymbol{q}, \psi_d, \theta_d) &= \boldsymbol{S}(\boldsymbol{\Gamma}(\boldsymbol{q}_d, \boldsymbol{x}_d^n))\boldsymbol{\Gamma}(\boldsymbol{q}, \boldsymbol{x}_d^n) \\ &= \boldsymbol{S}(\left[1, 0, 0\right]^\top)\boldsymbol{\Gamma}(\boldsymbol{q}, \boldsymbol{x}_d^n)\,.\end{aligned} \tag{31}$$

The attitude error function will be part of the quadratic cost of the OCP which we use to design the NMPC. It is therefore worth looking at the quadratic cost introduced by the attitude error function $J_{\tilde{\boldsymbol{\Phi}}}(\boldsymbol{q}, \psi_d, \theta_d) = \tilde{\boldsymbol{\Phi}}(\boldsymbol{q}, \psi_d, \theta_d)^\top \boldsymbol{Q}_{\tilde{\boldsymbol{\Phi}}} \tilde{\boldsymbol{\Phi}}(\boldsymbol{q}, \psi_d, \theta_d)$, where $\boldsymbol{Q}_{\tilde{\boldsymbol{\Phi}}} \in \mathbb{R}^{4\times4}$ is a positive definite and diagonal weighting matrix. A plot over the desired yaw and pitch angle is given in Fig. 3, which shows that a quadratic cost on the derived attitude error function has a well-defined minimum where the yaw and pitch angle coincide with their desired values.

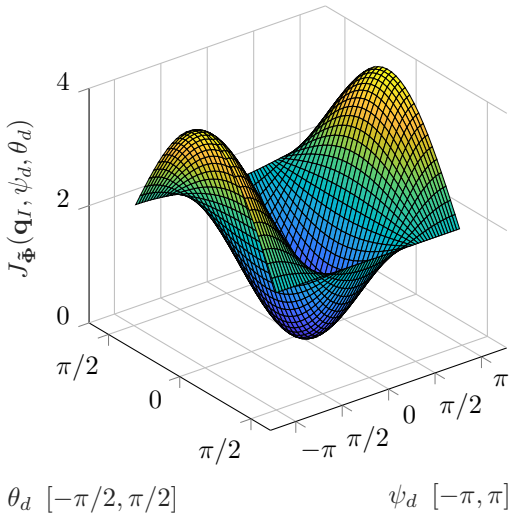

Fig. 3. The cost function $J_{\tilde{\boldsymbol{\Phi}}}(\boldsymbol{q}_I, \psi_d, \theta_d)$ evaluated at the identity quaternion $\boldsymbol{q}_I$ and $\boldsymbol{Q}_{\tilde{\boldsymbol{\Phi}}} = \boldsymbol{I}_{4\times4}$.

B. Inequality constraints

Inequality constraints are imposed by the flight envelope which defines limits on the allowable relative airspeed and angle of attack in order to prevent the UAV from stalling and to ensure a sufficient lift force. With given lower bounds $V_{r,\min}$ and $\alpha_{\min}$ as well as upper bounds $V_{r,\max}$ and $\alpha_{\max}$, we have the inequalities

$$V_{r,\min} \leq V_r \leq V_{r,\max} \tag{32a}$$
$$\alpha_{\min} \leq \alpha \leq \alpha_{\max} \tag{32b}$$

which we can represent as the vector inequality

$$\boldsymbol{h}(\boldsymbol{x},\boldsymbol{u}) \triangleq \begin{bmatrix} -V_r \\ V_r \\ -\alpha \\ \alpha \end{bmatrix} \leq \begin{bmatrix} -V_{r,\min} \\ V_{r,\max} \\ -\alpha_{\min} \\ \alpha_{\max} \end{bmatrix} \triangleq \boldsymbol{h}_{\max}\,. \tag{33}$$

To guarantee the feasibility of the solution under perturbations we follow the approach in [14] and introduce slack variables $S_i \in \mathbb{R}$ and constant back-off parameters ϵ_i to apply the following constraint relaxation:

$$\frac{h_i(\boldsymbol{x},\boldsymbol{u})}{(1-\epsilon_i)h_{\max,i}} - S_i \leq 1, \;\; S_i \geq 0\,. \tag{34}$$

The slack variables will be zero in most cases and become positive as soon as $h_i(\boldsymbol{x},\boldsymbol{u}) > (1-\epsilon_i)h_{\max,i}$. The back-off parameter can be used to introduce a safety margin to the actual constraints, which will be used to operate the UAV in a near-stall situation with turbulent wind conditions.

C. Optimal Control Problem & Discretization

The OCP over a given time horizon T can now be stated as

$$\boldsymbol{J} = \min_{\boldsymbol{x},\boldsymbol{u}} \int_t^{t+T} \tilde{\boldsymbol{x}}^\top \boldsymbol{Q_x} \tilde{\boldsymbol{x}} + \boldsymbol{u}^\top \boldsymbol{Q_u} \boldsymbol{u} d\tau \tag{35a}$$
$$+ \tilde{\boldsymbol{x}}^\top(T) \boldsymbol{P} \tilde{\boldsymbol{x}}(T) + \boldsymbol{W}^\top \boldsymbol{S}$$
$$\text{s.t. } \boldsymbol{x}(t) = \hat{\boldsymbol{x}}(t) \tag{35b}$$
$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u}) \tag{35c}$$
$$\frac{h_i(\boldsymbol{x},\boldsymbol{u})}{(1-\epsilon_i)h_{\max,i}} - S_i \leq 1, \;\; S_i \geq 0 \tag{35d}$$
$$\boldsymbol{u}_{\min} \leq \boldsymbol{u} \leq \boldsymbol{u}_{\max} \tag{35e}$$

where the time-varying error state $\tilde{\boldsymbol{x}}(t) \in \mathbb{R}^{n_x}$ is given by

$$\tilde{\boldsymbol{x}}(t) = \begin{bmatrix} \tilde{\boldsymbol{\Phi}}(\boldsymbol{q},\psi_d,\theta_d) \\ V_r - V_{r,\mathrm{d}} \\ \beta - \beta_d \\ \phi - \phi_d \\ \boldsymbol{\omega}_s - \boldsymbol{\omega}_{s,d} \end{bmatrix}. \tag{36}$$

The first two elements of this vector drive the reduced attitude and relative speed to their desired references, whereas the other terms serve to penalize high roll and side-slip angles and to avoid fast attitude transitions. Note that when the desired attitude reference is constant, we set $\beta_d = \varphi_d = \boldsymbol{\omega}_{s,d} = 0$.

The vector $\boldsymbol{u}(t)$ is the time-varying control input with lower and upper bounds $\boldsymbol{u}_{\min}$ and $\boldsymbol{u}_{\max}$, respectively. The matrices $\boldsymbol{Q},\boldsymbol{P} \in \mathbb{R}^{n_x \times n_x}$ and $\boldsymbol{Q_u} \in \mathbb{R}^{n_u \times n_u}$ are symmetric and positive definite weighting matrices. The vectors $\boldsymbol{S},\boldsymbol{W} \in \mathbb{R}^{n_S}$ contain the slack variables and the corresponding weights, respectively. The vector function $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})$ represents the system dynamics of the model (22) and (35) - (35) are the known state and actuator constraints.

In order to discretize the OCP we use multiple shooting [21] which means that we split up the time horizon T into N control intervals. The result is a uniform time grid $t \in \{t_0, t_1, \ldots, t_N\}$ with piecewise constant control inputs $u \in \{u_0, u_1, \ldots, u_{N-1}\}$. To integrate the system dynamics an explicit fourth order Runge-Kutta integrator is used which defines $\boldsymbol{x}_{k+1} = \boldsymbol{F}(\boldsymbol{x}_k,\boldsymbol{u}_k)$. The resulting NLP has the form

$$\min_{\boldsymbol{\chi}} \quad \boldsymbol{\phi}(\boldsymbol{\chi})' \tag{37a}$$
$$\text{s.t.} \quad \boldsymbol{x}_0 = \hat{\boldsymbol{x}}(t_0) \tag{37b}$$
$$\boldsymbol{u}_0 = \boldsymbol{u}(t_0) \tag{37c}$$
$$\boldsymbol{x}_{k+1} - \boldsymbol{F}(\boldsymbol{x}_k,\boldsymbol{u}_k) = 0 \tag{37d}$$
$$\frac{h_i(\boldsymbol{x}_k,\boldsymbol{u}_k)}{(1-\epsilon_i)h_{\max,i}} - S_i \leq 1 \tag{37e}$$
$$S_i \geq 0, \;\; i = [1, n_S] \tag{37f}$$
$$\boldsymbol{u}_{\min} \leq \boldsymbol{u}_k \leq \boldsymbol{u}_{\max} \tag{37g}$$

where the optimization variable $\boldsymbol{\chi} \in \mathbb{R}^{n_w}$ is given by

$$\boldsymbol{\chi} = \begin{bmatrix} \boldsymbol{x}_0^\top & \boldsymbol{u}_0^\top & \ldots & \boldsymbol{x}_{N-1}^\top & \boldsymbol{u}_{N-1}^\top & \boldsymbol{x}_N^\top & \boldsymbol{S}^\top \end{bmatrix}^\top. \tag{38}$$

The resulting dimension of the optimization variable is $n_w = N \cdot (n_x + n_u) + n_x + n_S$. The cost of the NLP in (37) is given by

$$\boldsymbol{\phi}(\boldsymbol{\chi})' = \sum_{k=0}^{N-1} (\tilde{\boldsymbol{x}}_k^\top \boldsymbol{Q_x} \tilde{\boldsymbol{x}}_k + {\boldsymbol{u}_k}^\top \boldsymbol{Q_u} \boldsymbol{u}_k) + {\boldsymbol{x}_N}^\top \boldsymbol{P} \boldsymbol{x}_N + \boldsymbol{W}^\top \boldsymbol{S}\,. \tag{39}$$

Equations (37) - (37) are the initial conditions for the current timestep and (37) is the shooting gap constraint. It is desirable to add a cost to the dynamics of the control input in order to avoid fast changes in the actuators which would lead to increased energy consumption and wear and tear. Since the system dynamics do not include actuator dynamics it was not possible to include fast dynamics of the control inputs in the OCP. We can, however, add an additional cost to the NLP based on the difference of the control input between the shooting intervals, which yields the cost function

$$\boldsymbol{\phi}(\boldsymbol{\chi}) = \boldsymbol{\phi}(\boldsymbol{\chi})' + \frac{N^2}{T^2} \sum_{k=0}^{N-2} \Delta{\boldsymbol{u}_k}^\top \boldsymbol{Q}_{\Delta \boldsymbol{u}} \Delta \boldsymbol{u}_k \tag{40}$$

with $\Delta \boldsymbol{u}_k = \boldsymbol{u}_{k+1} - \boldsymbol{u}_k$ and the weighting matrix $\boldsymbol{Q}_{\Delta \boldsymbol{u}} \in \mathbb{R}^{n_u \times n_u}$ is symmetric, diagonal and positive.

D. Implementation & Limitations

We implement the OCP and multiple shooting algorithm in MATLAB using the algorithmic differentiation tool CasADi [22]. In order to find a numerical solution to the NLP, an interior-point optimization method implemented

in the open-source software package IPOPT [23] is used together with the linear solver ma97 from the HSL package [24]. The implementation does not include upper bounds on the time allowed for solving the underlying NLPs, which means that hard guarantees on algorithm convergence time can not be given. The current setup therefore does not allow for a real-time implementation. Furtheremore, the algorithm in the current form does not take into account the problem of model inaccuracies which is one of the major challenges of model-based controllers. Development towards a real-time applicable solution in which model inaccuracies will be taken into acount will be part of future work. A a real-time iteration scheme together with a sequential quadratic programming approach (see e.g. [25]) will be the most likely modifications to the presented algorithm.

IV. Results

In this section we will discuss implementation of the designed controller and compare its performance to that of a set of conventional PID controllers in a simulation study.

A. Implementation

The vehicle model for the simulation study is the Skywalker X8 shown previously in Fig. 2. The structure of the model matches the structure of the aerodynamic equations discussed in Sec. II and the parameters of the aerodynamic model are given in [19]. The parameters for the thrust model are taken from [26]. Both the NMPC and the forward simulation are based on the same set of parameters. The dynamic equations are integrated at 100 Hertz using an explicit fourth-order Runge-Kutta integrator.

For the NMPC, the time horizon $T = 10\,\text{s}$ is chosen which is divided into $N = 40$ control intervals. In order to prioritize attitude control over actuator usage we use larger weights on $\boldsymbol{Q_x}$ compared to $\boldsymbol{Q_u}$. After tuning, the weighting matrices

$$\begin{aligned}\boldsymbol{Q_x} &= \text{diag}([1 \;\; 10^2 \;\; 3.2 \;\; 3.2 \;\; 3.2 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1]) \\ \boldsymbol{Q_u} &= \text{diag}(\begin{bmatrix}10^{-3} & 10^{-3} & 10^{-3}\end{bmatrix}) \\ \boldsymbol{Q_{\Delta u}} &= \text{diag}(\begin{bmatrix}0.16 & 0.16 & 0.16\end{bmatrix}) \\ \boldsymbol{P} &= \boldsymbol{Q_x} \\ \boldsymbol{W} &= [10 \; 10 \; 10^4 \; 10^4]^\top\end{aligned} \tag{41}$$

were found to result in a desriable behaviour of the UAV. The back-off parameter is the same for all inequality constraints, $\epsilon_i = 0.3,\ i \in [1, n_S]$ and the update rate of the NMPC is set to 20 Hertz.

B. PID controllers

For comparison, we run the simulation with a set of PID controllers. The roll and pitch angle are controlled via PID controllers for aileron and elevator deflection, respectively. The desired roll angle is generated by a PI controller which seeks to reduce the error in the yaw angle. The relative speed is controlled through the throttle position, which is determined by another PI controller. The outputs of the controllers are thus given by

$$\delta_a = k_{p\phi}(\phi_d - \phi) + k_{i\phi}\int_0^t (\phi_d - \phi)d\tau + k_{d\phi}p \tag{42a}$$

$$\phi_d = k_{p\psi}(\psi_d - \psi) + k_{i\psi}\int_0^t (\psi_d - \psi)d\tau \tag{42b}$$

$$\delta_e = k_{p\theta}(\theta_d - \theta) + k_{i\theta}\int_0^t (\theta_d - \theta)d\tau + k_{d\theta}q \tag{42c}$$

$$\delta_t = k_{pV_{r,d}}(V_{r,d} - V_r) + k_{iVr}\int_0^t (V_{r,d} - V_r)d\tau. \tag{42d}$$

The PID controllers are tuned to the resulting gains summarized in Table I together with the bandwidth and damping ratio of the canonical second-order transfer functions. See [17] for details on the design process. The trim values for the controller design are obtained from a flight at the initial references for attitude and relative speed. A simple anti-windup strategy is used in which the integrator gain is constrained to the interval $[-I_{\max,}, I_{\max,}]$ and the update rate of the PID controllers is set to 100 Hertz. The limits for the states and actuators are summarized in Table II.

TABLE I
PID controller gains.

Parameter	Value
$k_{p\phi}$, $k_{p\theta}$, $k_{p\psi}$, k_{pVr}	0.78, -0.78, 1.08, 0.69
$k_{i\phi}$, $k_{i\theta}$, $k_{i\psi}$, k_{iVr}	0.01, -0.30, 0.36, 10.00
$k_{d\phi}$, $k_{d\theta}$	-0.11, -0.16
$I_{\max,\phi}$,$I_{\max,\theta}$,$I_{\max,\psi}$,$I_{\max,V}$,	0.09, 0.09, 0.09, 1.00,
$\omega_{n\phi}$,$\omega_{n\theta}$,$\omega_{n\psi}$,ω_{nV},	9.54, 17.49, 0.48, 19.88,
ζ_ϕ,ζ_θ,ζ_ψ,ζ_V,	0.71, 0.71, 0.71, 0.71

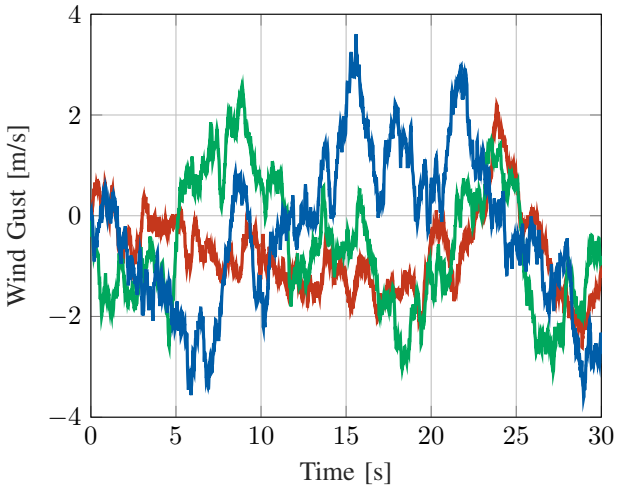

Fig. 4. Gust component of the wind velocity vector in the body-fixed frame with longitudinal (red), lateral (green) and along the z-axis of the UAV (blue).

C. Flight Scenario

In the simulation study we test the capabilities of the controller to recover from extreme attitude deviations with high angular velocities. We therefore set the initial roll angle

TABLE II

STATE AND ACTUATOR CONSTRAINTS.

Parameter	Value	Unit
$V_{r,\min},\ V_{r,\max}$	$10,\ 30$	m/s
$\alpha_{\min},\ \alpha_{\max}$	$-12,\ 12$	deg
$\delta_{a,\min},\ \delta_{a,\max}$	$-35,\ 35$	deg
$\delta_{e,\min},\ \delta_{e,\max}$	$-35,\ 35$	deg
$\delta_{t,\min},\ \delta_{t,\max}$	$0,\ 1$	-

to 140 degrees and the initial pitch angle to -40 degrees while keeping the initial yaw angle at zero. The initial linear and angular velocity vector in the body-fixed frame are set to $\boldsymbol{v}^b_{nb,0} = [18,\ 0,\ 0]^\top$ and $\boldsymbol{\omega}^b_{nb,0} = [50,\ 50,\ -50]^\top$ with units meter per second and degree per second, respectively.

The wind velocity vector in the body-fixed frame is modelled as the sum of a static component $\boldsymbol{w}^n_{\text{s}}$ and a gust component $\boldsymbol{w}^b_{\text{g}}$ by the equation

$$\boldsymbol{w}^b = \boldsymbol{R}(\boldsymbol{q}^n_b)^\top \boldsymbol{w}^n_{\text{s}} + \boldsymbol{w}^b_{\text{g}} \tag{43}$$

with $\boldsymbol{w}^n_{\text{s}} = [-5,\ -3,\ 0]^\top$ in meter peter second. The model for the gust component is the Dryden gust model which is used to generate a moderate turbulence [17], [27]. A plot of the gust component is shown in Fig. 4.

This results in the initial states in the wind frame given by

$$\begin{aligned} \boldsymbol{q}^n_{b,0} &= \begin{bmatrix} 0.32 & 0.88 & -0.12 & 0.32 \end{bmatrix}^\top \\ V_{r,0} &= 22.27 \\ \alpha_0 &= 1.40 \\ \beta_0 &= -11.30 \\ \boldsymbol{\omega}^s_{nb,0} &= \begin{bmatrix} 48.76 & 50.00 & -51.21 \end{bmatrix}^\top, \end{aligned} \tag{44}$$

with α_0, β_0 and $\boldsymbol{\omega}^s_{nb,0}$ in degree and degree per second, respectively.

The attitude reference is initially set to $\phi_d = \theta_d = \psi_d = 0$. After 15 seconds, we command an aggressive climb and turn maneuver and set the desired yaw and pitch angle to -135 degrees and 45 degrees, respectively. An FIR low-pass filter with 0.1 Hertz cutoff frequency and a 2 seconds Chebyshev window is applied to keep the reference step tractable. The reference for the relative airspeed is set to 15 meter per second and remains constant throughout the simulation. The aim of the maneuver is to create a situation in which a sudden change in pitch may lead to an angle of attack above the stall angle. This way, the NMPC is tested for following a desired attitude reference while keeping the angle of attack within the given bounds for a safe operation. Note that this maneuver may lead to load factors that can not be tolerated by a small fixed-wing UAV in real flight.

V. DISCUSSION

The attitude response in Fig. 7 and angular velocities in Fig. 8 show that both controllers are able to recover from the initial attitude and angular velocities. The PID controllers show a faster convergence to the attitude reference than the NMPC. The main reason for this is that the NMPC

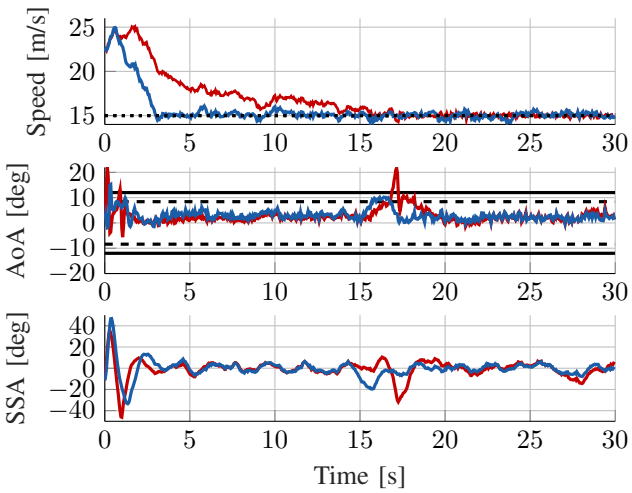

Fig. 5. The relative velocity represented by the relative speed (Speed), the angle of attack (AOA) and the side-slip angle (SSA). Results are shown for the NMPC (blue) and the PID controllers (red). The plots include the reference for the relative speed (dotted), limits on the angle of attack (solid, black) and the reduced limit using the back-off parameter ϵ (dashed, black).

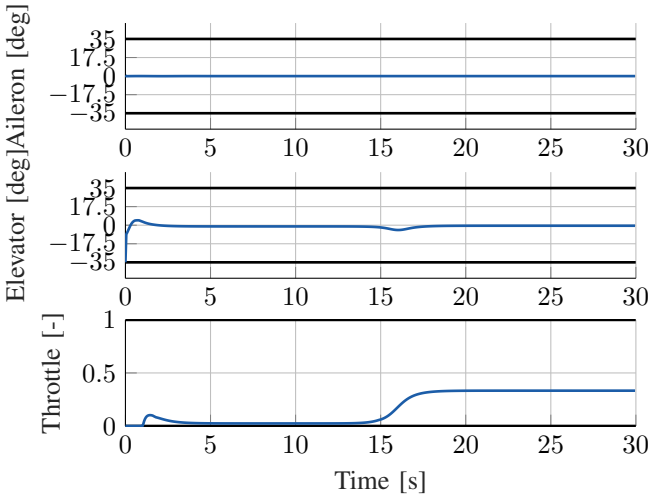

Fig. 6. UAV control inputs as output of the NMPC (blue) and the PID controllers (red). Surface deflection of the aileron δ_a and elevator δ_e with their limits (black, solid). The throttle position δ_t is in the interval [0,1].

simultaneously drives the relative speed to its reference (cf. Fig. 5), which it does by increasing the pitch angle and using the elevators close to their limits (cf. Fig. 6). It thus mediates between the objective to fly close to the attitude reference and the reference for the relative speed, which results in a faster exponential decrease of the objective value compared to the PID controllers as shown Fig. 9. Though PID control loops in which ground speed or relative speed are controlled by generating pitch angle references are common, the PID controllers here control the relative speed only through the throttle position, which in consequence is driven to its lower limit until the desired relative speed is reached, as shown in Fig. 6. The stall angle is exceeded by both controllers during the recovery phase. The PID controllers

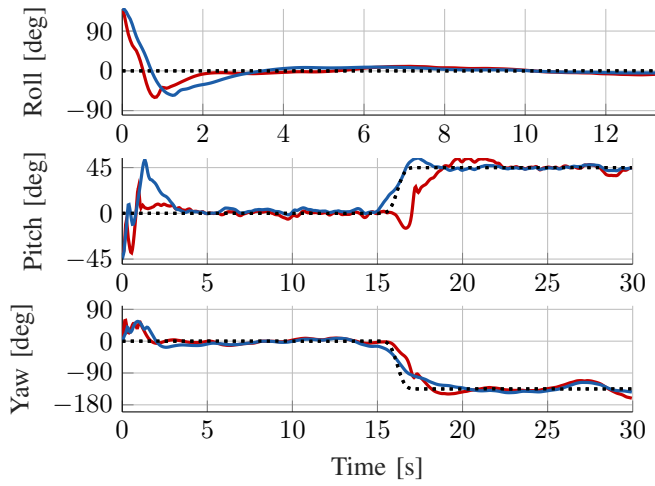

Fig. 7. Attitude response of the UAV given in Euler angles when using the NMPC (blue) and the PID controllers (red) given the attitude reference (dotted, black).

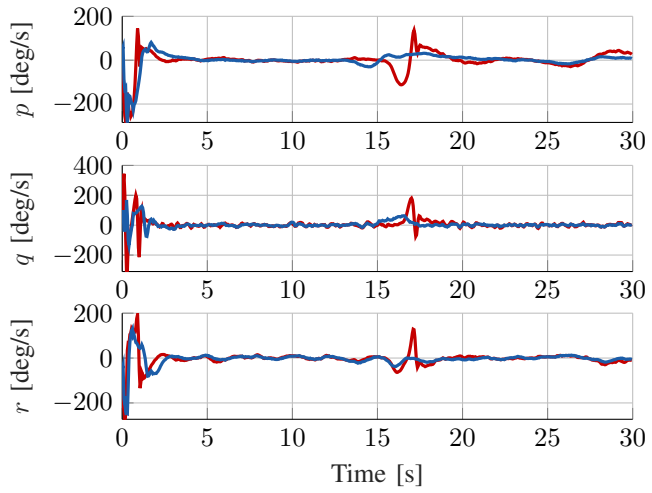

Fig. 8. Angular rates given by roll rate p, pitch rate q and yaw rate r. Results are shown for the NMPC (blue) and the PID controllers (red).

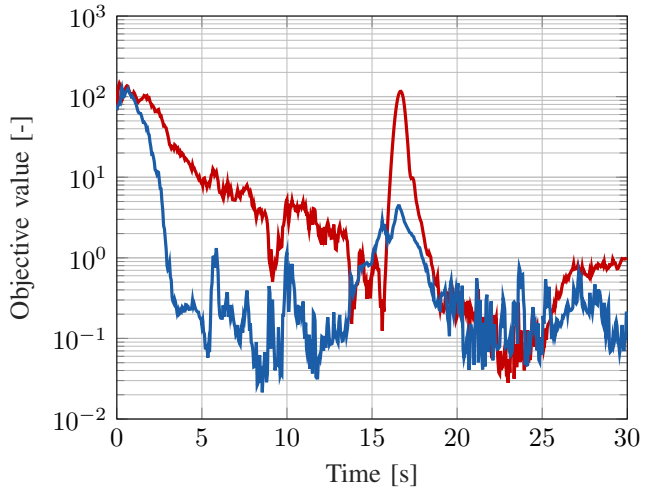

Fig. 9. Objective value for the NMPC (blue) and the PID controllers (red).

do not include angle of attack constraints and therefore do not account for them. The NMPC violates the upper bound due to the constraint relaxation and prioritizes the decrease of the attitude error and the error in the relative speed over the constraint violation. A higher weighting of the slack variables associated to the angle of attack can be used to prevent this.

The following climb and turn maneuver starts at 15 seconds. The NMPC starts the attitude change at about 14 seconds, which is a trade-off in the deviation from the current and future reference. Information about the future reference enables the NMPC to keep the attitude closer to it, which can be seen for all three axes (cf. Fig. 7). The PID controllers, on the other hand, are purely reactive and start to act as soon as there is a difference between the current attitude and its reference.

The transient behaviour of both controllers is notably different. The PID controllers show a faster transient reponse in both the lateral and longitudinal channels with significantly larger angular velocities (cf. Fig. 8). This comes at the cost of exceeding the maximum angle of attack as shown in and also larger side-slip angles Fig. 5. The NMPC in constrast explicitly deals with angle of attack, side-slip angle and relative speed and clearly lowers the increase in the angle of attack as soon as it reaches the margin defined by the back-off parameter. Furthermore, the transient of the NMPC is more monotonic since it is based on the full nonlinear dynamics in contrast to the decoupled assumption of the PID controllers which introduces a disturbance between lateral and longitudinal channel.

VI. Conclusion

In this paper, we presented a design for an NMPC for attitude control of a fixed-wing UAV and utilized a vehicle model formulated in the wind frame to include relative airspeed, angle of attack and side-slip angle as states in the control objective. A reduced attitude formulation is employed in order to treat the control of yaw and pitch angle as primary control objectives and the roll angle as an intermediate secondary objective. The designed NMPC shows good performance in a simulation study compared to conventional PID controllers. This includes its capabilities to recover the UAV from disturbed initial conditions as well as flying aggressive attitude maneuvers while keeping the angle of attack within allowed constraints imposed by the flight envelope as well as keeping relative speed and side-slip angle closer to desirable values.

Regarding future work, a more realistic scenario with mismatch of the used vehicle model including neglected dynamics will be considered. When looking at the NMPC as a potential real-time applicable solution, a trade-off between model complexity and the available computational resources together with implementation aspects will need to be investigated.

REFERENCES

[1] O. Hegrenaes, J. T. Gravdahl, and P. Toendel, "Spacecraft attitude control using explicit model predictive control," *Automatica*, 2005. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109805002657

[2] P. Sopasakis, D. Bernardini, H. Strauch, S. Bennani, and A. Bemporad, "A hybrid model predictive control approach to attitude control with minimum-impulse-bit thrusters," in *2015 European Control Conference (ECC) July 15-17, 201. Linz, Austria*.

[3] A. Sakamoto, Y. Ikeda, I. Yamaguchi, and T. Kida, "Nonlinear model predictive control for large angle attitude maneuver of spacecraft with rw and rcs," in *2016 IEEE 55th Conference on Decision and Control (CDC) ARIA Resort & Casino December 12-14, 2016, Las Vegas, USA*. IEEE, 2016.

[4] U. V. Kalabić, R. Gupta, S. Di Cairano, A. M. Bloch, and I. V. Kolmanovsky, "MPC on manifolds with an application to the control of spacecraft attitude on SO(3)," *Automatica*, vol. 76, pp. 293–300, 2017. [Online]. Available: http://dx.doi.org/10.1016/j.automatica.2016.10.022

[5] M. Kamel, K. Alexis, M. Achtelik, and R. Siegwart, "Fast nonlinear model predictive control for multicopter attitude tracking on SO(3)," *2015 IEEE Conference on Control and Applications, CCA 2015 - Proceedings*, no. 3, pp. 1160–1166, 2015.

[6] T. Lee, M. Leok, and N. H. McClamroch, "Geometric tracking control of a quadrotor UAV on SE(3)," *Proceedings of the IEEE Conference on Decision and Control*, pp. 5420–5425, 2010.

[7] A. Zanelli, "Nonlinear Model Predictive Control of a Humand-sized Quadrotor," *2018 European Control Conference (ECC)*, pp. 1542–1547, 2018. [Online]. Available: https://upcommons.upc.edu/handle/2117/98503

[8] A. Zanelli, R. Quirynen, G. Frison, and M. Diehl, "A partially tightened real-time iteration scheme for nonlinear model predictive control," in *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. IEEE, 2017, pp. 4388–4393.

[9] N. Slegers, J. Kyle, and M. Costello, "Nonlinear Model Predictive Control Technique for Unmanned Air Vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 29, no. 5, pp. 1179–1188, 2006. [Online]. Available: http://arc.aiaa.org/doi/10.2514/1.21531

[10] Y. Kang and J. K. Hedrick, "Linear tracking for a fixed-wing UAV using nonlinear model predictive control," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1202–1210, 2009.

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

[12] T. J. Stastny, G. A. Garcia, and S. S. Keshmiri, "Collision and obstacle avoidance in unmanned aerial systems using morphing potential field navigation and nonlinear model predictive control," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 1, p. 014503, 2015.

[13] S. H. Mathisen, K. Gryte, T. Johansen, and T. I. Fossen, "Non-linear Model Predictive Control for Longitudinal and Lateral Guidance of a Small Fixed-Wing UAV in Precision Deep Stall Landing," *AIAA Infotech @ Aerospace*, no. January, pp. 1–16, 2016. [Online]. Available: http://arc.aiaa.org/doi/10.2514/6.2016-0512

[14] S. Gros, R. Quirynen, and M. Diehl, "Aircraft control based on fast non-linear mpc & multiple-shooting," in *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, Dec 2012, pp. 1142–1147.

[15] D. K. Kufoalor and T. A. Johansen, "Reconfigurable fault tolerant flight control based on nonlinear model predictive control," in *American Control Conference (ACC), 2013*. IEEE, 2013, pp. 5128–5133.

[16] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft control and simulation : dynamics, controls design, and autonomous systems*. John Wiley & Sons, 2016.

[17] R. W. Beard and T. W. McLain, *Small unmanned aircraft: Theory and practice*. Princeton University Press, 2012.

[18] J. Sola, "Quaternion kinematics for the error-state kalman filter," *arXiv preprint arXiv:1711.02508*, 2017.

[19] K. Gryte, R. Hann, M. Alam, J. Rohác, T. A. Johansen, and T. I. Fossen, "Aerodynamic modeling of the Skywalker X8 Fixed-Wing Unmanned Aerial Vehicle," *International Conference of Unmanned Aerial Systems*, 2018.

[20] N. A. Chaturvedi, T. Lee, M. Leok, and N. H. McClamroch, "Nonlinear dynamics of the 3D pendulum," *Journal of Nonlinear Science*, vol. 21, no. 1, pp. 3–32, 2011.

[21] H. G. Bock and K.-J. Plitt, "A multiple shooting algorithm for direct solution of optimal control problems," *IFAC Proceedings Volumes*, vol. 17, no. 2, pp. 1603–1608, 1984.

[22] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, In Press, 2018.

[23] A. Wächter and L. T. Biegler, "On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[24] "Hsl, a collection of fortran codes for large-scale scientific computation." [Online]. Available: http://www.hsl.rl.ac.uk/

[25] S. Gros, M. Zanon, R. Quirynen, A. Bemporad, and M. Diehl, "From linear to nonlinear mpc: bridging the gap via the real-time iteration," *International Journal of Control*, vol. 0, no. 0, pp. 1–19, 2016. [Online]. Available: https://doi.org/10.1080/00207179.2016.1222553

[26] 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.

[27] J. W. Langelaan, N. Alley, and J. Neidhoefer, "Wind field estimation for small unmanned aerial vehicles," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 4, pp. 1016–1030, 2011.