Nonlinear model predictive control and guidance for a propeller-tilting hybrid unmanned air vehicle

Automatica 132 (2021) 109790

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

Nonlinear model predictive control and guidance for a propeller-tilting hybrid unmanned air vehicle ☆

Mike Allenspach [a,*], Guillaume Jacques Joseph Ducard [b]

[a] Autonomous Systems Lab ETH Zürich, 8092 Zürich, Switzerland
[b] I3S, University Côte d'Azur, CNRS, 06903 Sophia Antipolis, France

ARTICLE INFO

Article history:
Received 1 August 2020
Received in revised form 28 February 2021
Accepted 7 June 2021
Available online xxxx

Keywords:
Nonlinear control
MPC
Convertible UAV

ABSTRACT

Hybrid unmanned aerial vehicles (UAVs) provide an interesting combination of the vertical takeoff and landing (VTOL) capabilities of rotary-wing (RW) and of the efficient forward flight of fixed-wing (FW) vehicles. The employed controllers must be able to handle the highly nonlinear dynamics and changing control authorities resulting from this combination, especially during the transition between the two flight modes. In this paper, a nonlinear model predictive control (MPC) structure is designed and applied to a tiltrotor convertible UAV. Full-flight envelope trajectory tracking and optimal exploitation of the aircraft's VTOL and FW properties is achieved. A common approach is to design a set of stable controllers for various trim-points in the flight envelope and make use of Gain Scheduling (GS) or controller mixing, in order to select the appropriate control law for a given flight configuration. In this paper, a unified control approach is developed. A nonlinear MPC structure is designed and applied to a tiltrotor convertible UAV whose nonlinear model is derived and presented. Thus, full-flight envelope trajectory tracking and optimal exploitation of the aircraft's VTOL and FW properties are achieved, without the need for controller switching or scheduling policies. The proposed multi-stage control allocation handles the changing control authorities of the actuators in a continuous manner and efficiently distributes the required control actions between propellers, tilt servos and control-surfaces. The feasibility and performance of this novel control approach are successfully evaluated in real-world flight experiments. Simulation results for comparison with state-of-the-art approaches are presented.

1. Introduction

Over the last years, the field of hybrid unmanned aerial vehicles (UAVs) has gained increasing interest in the robotics community. Also known as "convertibles", these vehicles are designed to accomplish gravity compensation through both propeller-induced thrust and airspeed-induced lift forces, essentially combining the concepts behind the standard rotary-wing (RW) and fixed-wing (FW) platforms. Combining the benefits of both groups, hybrid UAVs effectively pair vertical takeoff and landing (VTOL) and hover capabilities with efficient cruise flight (Liu, He, Yang, & Han, 2017; Saeed, et al., 2015).

In this paper, a tiltrotor hybrid UAV is considered, meaning that pivoting propellers are used for thrust vectoring. Depending on the flight mode, the thrust can be directed upwards for lift generation (RW mode) or rotated forward for horizontal acceleration (FW mode), while the fuselage is controlled to remain horizontal throughout the mission. However, the thrust-vectoring capabilities make these vehicles difficult to control during the transition from one mode to the other, since highly-nonlinear models with changing aerodynamic effects and control authorities must be considered.

These challenges are usually tackled by linearizing the model around different trim-points in the feasible flight envelope and designing a stable linear control law for each of them. Continuous control over the full flight envelope is achieved through the combination of these linear control laws with adaptive methodologies such as Gain Scheduling (GS). A selection of common combinations is provided in Table 1.

Other control approaches involve model predictive control (MPC), in order to compute more optimal control inputs. However, these control structures generally focus exclusively on hover (Papachristos, Alexis, & Tzes, 2013) or FW flight (Kvaternik, et al., 2001). Indeed, most nonlinear control laws employed in tiltrotor UAV control are either limited to a single flight mode or

☆ The material in this paper was partially presented at the 28th Mediterranean Conference on Control and Automation, September 15-18, 2020, Saint-Raphaël, France. This paper was recommended for publication in revised form by Associate Editor Abdelhamid Tayebi under the direction of Editor Thomas Parisini.

* Corresponding author.
E-mail addresses: mike.allenspach@mavt.ethz.ch (M. Allenspach), ducard@i3s.unice.fr (G.J.J. Ducard).

https://doi.org/10.1016/j.automatica.2021.109790

Table 1
Common linear control laws and adaptive approaches for tiltrotor hybrid UAV control.

Linear control law	Adaptive methodology	Source
Proportional–Integral–Derivative	Gain Scheduling	Cetinsoy (2015), Nakamura, Arakawa, Watanabe, and Nagai (2018)
	Control Law Weighting	Anglade, Kai, Hamel, and Samson (2019), Bauersfeld and Ducard (2020)
	Dynamic Inversion	Chao, Lincheng, Daibing, and Jiyang (2016), Yatsun, Lushnikov, and Emelyanova (2018)
State Feedback	Gain Scheduling	Hernández-García and Rodríguez-Cortés (2015), Yangping and Honggang (2018)
Linear–Quadratic-Regulator	Gain Scheduling	Park, Bae, Kim, and Kim (2013), Peng, et al. (2010)

lack experimental evaluation. As an example, although the MPCs presented in Mehra, Prasanth, and Gopalaswamy (1998) and Mehra, Wasikowski, Prasanth, Bennett, and Neckels (2001) are theoretically capable of handling the transition maneuver, their performance has only been evaluated in simulation. The same is true for the backstepping controller designed in Kendoul, Fantoni, and Lozano (2006) or the sliding mode controller in Zhu, Fan, and Yang (2010).

To sum up, most controllers found in literature are based on linearized models, limited to specific areas of the flight envelope due to a finite number of linearization points. As a consequence, aerodynamic effects acting on the vehicle during transition can often not be considered adequately, meaning that stable and feasible trajectory generation is not guaranteed. On the other hand, nonlinear controllers capable of capturing the mentioned effects are often only evaluated in simulation, meaning that their performance in the presence of modeling uncertainties and external disturbances remains unclear.

Contrarily, the main contribution of this work is the development of a nonlinear guidance-MPC module, capable of handling the nonlinear dynamics and covering the entire flight envelope without the need for controller switching or scheduling policies. Compared to the simulation proof-of-concept provided in Allenspach and Ducard (2020), the controller in this work has been modified and extended to allow on-board and real-time deployment on a small form-factor computer aboard the considered prototype and is evaluated in real-world flight experiments. The theoretical background and key ideas of this novel approach are explained in the remainder of the paper as follows:

- The nonlinear model of the hybrid UAV used for controller synthesis is derived in Section 2.
- Key ideas behind the nonlinear MPC formulation, as well as the underlying optimization problem and incorporated solver are presented in Section 3.
- The proposed multi-stage control allocation presented in Section 4 accounts for changing control authorities and efficiently exploits the overactuation of the system. A theoretical derivation of feasible thrust and torque sets is performed.
- The real-world flight experiments conducted to evaluate the feasibility and performance of the proposed approach are discussed in Section 5. Additionally, simulation results for comparison against state-of-the-art approaches are included.
- Finally, Section 6 concludes with remarks and possible directions for future work.

To the best of the authors knowledge, this is the first time that on-board, real-time, nonlinear MPC has been deployed successfully on a hybrid tiltrotor UAV.

2. Vehicle description

The nonlinear model presented below bases on preliminary work in Ducard and Hua (2014) and will be used for controller design. Important variables are defined in Table 2 and the considered tiltrotor UAV is sketched in Figs. 1 and 2, with rotor–tilt angles in both RW and FW configurations. Note that the construction of the tilt mechanism allows for arbitrary tilt angles in between these two positions.

Table 2
Important variables and definitions.

Symbol	Definition
$\mathcal{I} : \{O_\mathcal{I}, \boldsymbol{e}_n, \boldsymbol{e}_e, \boldsymbol{e}_d\}$	Inertial frame: origin and primary axes
$\mathcal{B} : \{O_\mathcal{B}, \boldsymbol{e}_x, \boldsymbol{e}_y, \boldsymbol{e}_z\}$	Body-fixed frame: origin and primary axes
$\boldsymbol{q}_{\mathcal{IB}} \in \mathbb{H}$	Attitude quaternion of $\mathcal{B}$ relative to $\mathcal{I}$
$\boldsymbol{\omega}_{\mathcal{IB}} \in \mathbb{R}^3$	Angular velocity of $\mathcal{B}$ relative to $\mathcal{I}$
$\boldsymbol{p} \in \mathbb{R}^3$	center of mass position relative to $O_\mathcal{I}$
$\boldsymbol{v} \in \mathbb{R}^3$	center of mass velocity relative to $O_\mathcal{I}$
$m,\ I^\mathcal{B}$	Mass and body-frame inertia matrix
$\boldsymbol{\omega} \in \mathbb{R}^4$	Vector of rotor spinning velocities
$\boldsymbol{\chi} \in \mathbb{R}^2$	Vector of rotor–tilt angles
$\boldsymbol{\delta} \in \mathbb{R}^5$	Vector of control surface deflections
$\boldsymbol{F}_r \in \mathbb{R}^3,\ \boldsymbol{M}_r \in \mathbb{R}^3$	Total propeller forces and torques
$\boldsymbol{T}_i \in \mathbb{R}^3$	ith propeller thrust contribution
$\boldsymbol{Q}_i \in \mathbb{R}^3$	ith propeller resisting torque contribution
$\boldsymbol{F}_a \in \mathbb{R}^3,\ \boldsymbol{M}_a \in \mathbb{R}^3$	Total aerodynamic forces and torques
$C_{L,k},\ C_{D,k}$	kth aerodynamic lift and drag coefficient
$\boldsymbol{v}_{a,k} \in \mathbb{R}^3,\ \alpha_k$	kth airspeed vector and angle-of-attack
$\boldsymbol{M}_\delta \in \mathbb{R}^3$	Torques produced by control surfaces

2.1. Conventions and nomenclature

Two coordinate frames are defined as shown in Figs. 1 and 2: the inertial North-East-Down (NED) frame $\mathcal{I}$ with origin $O_\mathcal{I}$ and the body frame $\mathcal{B}$ attached to the aircraft's center of mass (CoM) with origin $O_\mathcal{B}$. Here, the quaternion $\boldsymbol{q}_{\mathcal{IB}} = (\, q_0 \ \boldsymbol{q}_v^\top \,)^\top \in \mathbb{H}$ and vector $\boldsymbol{\omega}_{\mathcal{IB}} \in \mathbb{R}^3$ denote the orientation and angular rate of $\mathcal{B}$ relative to $\mathcal{I}$, whereby $\mathbb{H}$ is the Hamilton space. The position and velocity of the vehicle's CoM with respect to $O_\mathcal{I}$ are described by $\boldsymbol{p} \in \mathbb{R}^3$ and $\boldsymbol{v} \in \mathbb{R}^3$, respectively.

A right-hand superscript, e.g. $\boldsymbol{p}^\mathcal{I}$ is used to specify the frame in which a vector is represented.

2.2. Actuator dynamics

The vehicle encompasses eleven actuation degree-of-freedoms (DoFs), namely four propeller speeds $\boldsymbol{\omega}$, five control-surfaces $\boldsymbol{\delta}$ and two tilt angles $\boldsymbol{\chi}$. The propeller speed and control-surface dynamics are fast and thus neglected compared to the dynamics of the rotor–tilt angles, which move much slower due to the combined inertia of the propeller, motor and arm assembly. Note that rotors 1 and 2 are mechanically constraint to the same tilt

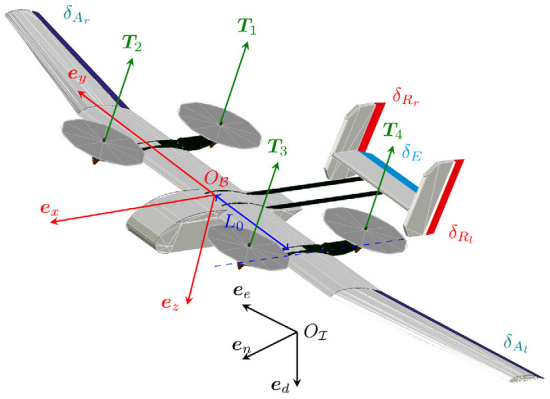

Fig. 1. Tiltrotor UAV propeller forces and control-surfaces with rotor–tilt angles $\chi_r = \chi_l = 0\,\text{rad}$ in RW configuration.

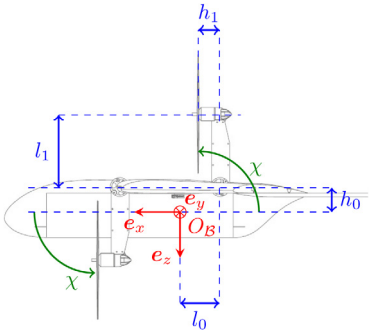

Fig. 2. Tiltrotor UAV side view with rotor–tilt angles $\chi_r = \chi_l = \frac{\pi}{2}\,\text{rad}$ in FW configuration.

angle, i.e. $\chi_1 = \chi_2 = \chi_r$ and similarly rotors 3 and 4 $\chi_3 = \chi_4 = \chi_l$. Their dynamics exhibit a slew-rate behavior as follows:

$$\dot{\chi}_i = \text{sgn}\left(\chi_{i,c} - \chi_i\right)\dot{\chi}_{max} \qquad \forall i \in \{1,\dots,4\} \tag{1}$$

where the subscript c denotes the commanded actuator values. According to Fig. 2, tilt angles are measured between the x–y-plane of $\mathcal{B}$ and the propeller plane.

2.3. Center-of-mass dynamics

The continuous-time equations of motion (EoM) for the vehicles CoM are derived based on the standard Newton-Euler equations. The six DoF model looks as follows:

$$\dot{\boldsymbol{p}}^{\mathcal{I}} = \boldsymbol{v}^{\mathcal{I}} \tag{2a}$$

$$\dot{\boldsymbol{v}}^{\mathcal{I}} = g\boldsymbol{e}_d{}^{\mathcal{I}} + \frac{1}{m}R_{\mathcal{I}\mathcal{B}}(\boldsymbol{q}_{\mathcal{I}\mathcal{B}})(\boldsymbol{F}_r{}^{\mathcal{B}} + \boldsymbol{F}_a{}^{\mathcal{B}}) \tag{2b}$$

$$\dot{\boldsymbol{q}}_{\mathcal{I}\mathcal{B}} = \frac{1}{2}\begin{pmatrix} 0 & -(\boldsymbol{\omega}_{\mathcal{I}\mathcal{B}}{}^{\mathcal{B}})^\top \\ \boldsymbol{\omega}_{\mathcal{I}\mathcal{B}}{}^{\mathcal{B}} & -[\boldsymbol{\omega}_{\mathcal{I}\mathcal{B}}{}^{\mathcal{B}}]_\times \end{pmatrix}\boldsymbol{q}_{\mathcal{I}\mathcal{B}} \tag{2c}$$

$$I^{\mathcal{B}}\dot{\boldsymbol{\omega}}_{\mathcal{I}\mathcal{B}}{}^{\mathcal{B}} = -[\boldsymbol{\omega}_{\mathcal{I}\mathcal{B}}{}^{\mathcal{B}}]_\times I^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{I}\mathcal{B}}{}^{\mathcal{B}} + \boldsymbol{M}_r{}^{\mathcal{B}} + \boldsymbol{M}_a{}^{\mathcal{B}} + \boldsymbol{M}_\delta{}^{\mathcal{B}} \tag{2d}$$

with gravitational acceleration g, mass m and moment of inertia $I^{\mathcal{B}}$. The rotation matrix $R_{\mathcal{I}\mathcal{B}} \in SO(3)$ is computed from the attitude quaternion using the Rodrigues' rotation formula $R_{\mathcal{I}\mathcal{B}}(\boldsymbol{q}_{\mathcal{I}\mathcal{B}}) = \mathbb{I}_3 + 2q_0[\boldsymbol{q}_v]_\times + 2[\boldsymbol{q}_v]_\times^2$, whereby $[\cdot]_\times : \mathbb{R}^3 \to \mathfrak{so}(3)$ is the skew-symmetric operator.

All external forces and moments are caused either by the rotor and tilt mechanism system $(\boldsymbol{F}_r, \boldsymbol{M}_r)$ or by aerodynamic effects $(\boldsymbol{F}_a, \boldsymbol{M}_a, \boldsymbol{M}_\delta)$. These contributions will be explained in more detail in the next two sections. Numerical values for vehicle dimensions and coefficients required for controller synthesis are listed in Table 4.

2.3.1. Propeller forces and moments

As shown in Ducard and Hua (2014), proper modeling of propeller generated forces $\boldsymbol{F}_r \in \mathbb{R}^3$ and torques $\boldsymbol{M}_r \in \mathbb{R}^3$ requires the inclusion of several different effects, e.g. gyroscopic torques and free-stream velocity over the propeller blades. Knowledge of these contributions is used in the development of a high-fidelity simulator in Section 5.

However, the high nonlinearity and sometimes implicit formulation of these terms largely increase the computational complexity of model-based optimal control approaches. Since the model derived in (2) serves primarily as a starting point for the controller design in Section 3, only the most prominent terms, namely the thrust $\boldsymbol{T}_i^{\mathcal{B}} \in \mathbb{R}^3$ and resisting torque $\boldsymbol{Q}_i{}^{\mathcal{B}} \in \mathbb{R}^3$ of each propeller, are considered here:

$$\boldsymbol{F}_r{}^{\mathcal{B}} = \sum_{i=1}^{4} \boldsymbol{T}_i{}^{\mathcal{B}} \tag{3a}$$

$$\boldsymbol{M}_r{}^{\mathcal{B}} = \sum_{i=1}^{4} \boldsymbol{Q}_i{}^{\mathcal{B}} + \boldsymbol{d}_i{}^{\mathcal{B}} \times \boldsymbol{T}_i{}^{\mathcal{B}} \tag{3b}$$

where the vector $\boldsymbol{d}_i$ from $O_\mathcal{B}$ to the ith rotor's center can be constructed, using the vehicle dimensions l_1, h_1, l_0, h_0, L_0 listed in Table 4. Using a standard modeling approximation, the propeller forces and torques are proportional to the rotor speed ω_i squared:

$$\boldsymbol{T}_i{}^{\mathcal{B}} = -c_T\omega_i^2\boldsymbol{n}(\chi_i) \tag{4a}$$

$$\boldsymbol{Q}_i{}^{\mathcal{B}} = (-1)^i c_Q\omega_i^2\boldsymbol{n}(\chi_i) \tag{4b}$$

with c_T and c_Q being thrust and torque coefficients respectively and direction vector $\boldsymbol{n}(\chi) = (-\sin\chi\ 0\ \cos\chi)^\top$. The resisting torque is oriented parallel/anti-parallel to the thrust for the rotors (1 and 3)/(2 and 4), since these rotors spin in CW/CCW manner, respectively.

2.3.2. Aerodynamic forces and moments

Six main sources of aerodynamic effects can be identified: right wing W_r, left wing W_l, fuselage F, horizontal tail element T_h, vertical tail element T_v and the control-surfaces. Note that the separate consideration of the right and left sides of the wing W_r, W_l simplifies the modeling of roll- and yaw-induced moments.

The aerodynamic forces and torques generated by the wing, fuselage and tail are collected in the terms $\boldsymbol{F}_a \in \mathbb{R}^3$ and $\boldsymbol{M}_a \in \mathbb{R}^3$ from Eq. (2):

$$\boldsymbol{F}_a{}^{\mathcal{B}} = \sum_{k\in\{W_r,W_l,F,T_h,T_v\}} \boldsymbol{F}_{a,k}{}^{\mathcal{B}} \tag{5a}$$

$$\boldsymbol{M}_a{}^{\mathcal{B}} = \sum_{k\in\{W_r,W_l,F,T_h,T_v\}} \boldsymbol{r}_k{}^{\mathcal{B}} \times \boldsymbol{F}_{a,k}{}^{\mathcal{B}} \tag{5b}$$

where $\boldsymbol{r}_k \in \mathbb{R}^3$ is the lever-arm vector between $O_\mathcal{B}$ and the location where $\boldsymbol{F}_{a,k} \in \mathbb{R}^3$ applies.

According to Kai (2018), this includes lift and drag effects that depend on the air density ρ, the characteristic surface area S_k and the relative airspeed $\boldsymbol{v}_{a,k} \in \mathbb{R}^3$:

$$\boldsymbol{F}_{a,k}{}^{\mathcal{B}} = \frac{\rho}{2}S_k\left|\boldsymbol{v}_{a,k}{}^{\mathcal{B}}\right|\left(C_{L,k}(\alpha_k)\boldsymbol{v}_{a,k}^{\perp\,\mathcal{B}} - C_{D,k}(\alpha_k)\boldsymbol{v}_{a,k}{}^{\mathcal{B}}\right) \tag{6}$$

$$\boldsymbol{v}_{a,k}{}^{\mathcal{B}} = R_{\mathcal{B}\mathcal{I}}\boldsymbol{v}_W{}^{\mathcal{I}} - \boldsymbol{v}^{\mathcal{B}} - \boldsymbol{\omega}_{\mathcal{I}\mathcal{B}}{}^{\mathcal{B}} \times \boldsymbol{r}_k{}^{\mathcal{B}} \tag{7}$$

where the direction of the lift force is $\boldsymbol{v}_{a,k}^{\perp}$, which is orthogonal to $\boldsymbol{v}_{a,k}$ and the airfoil span. The Angle-of-Attack (AoA) α_k is defined as the angle between $\boldsymbol{v}_{a,k}$ and the airfoil chord and $\boldsymbol{v}_W \in \mathbb{R}^3$ is the wind vector.

Table 3
Aerodynamic parameters required for the computation of drag and lift coefficient.

	Pre-stall				Post-stall		Blending	
	C_{D0}	$C_{D\alpha}$	C_{L0}	$C_{L\alpha}$	C_0	C_1	k	α_s
Wing (W_r, W_l)	0.03	0.2	0.25	5.62	0.025	1.0	20.0	0.227
Fuselage (F)	1.28	0.0	0.0	0.0	1.28	0.0	20.0	0.0
Tail (T_v, T_h)	0.0	1.24	0.0	0.885	0.0	0.314	1.0	0.698

Due to the vehicle's VTOL capabilities, the drag and lift coefficients $C_{D,k}$, $C_{L,k}$ must not only be modeled up to the stall AoA $\alpha_{s,k}$ but also in the post-stall region, in order to cover the entire flight envelope. It has been demonstrated in Ducard and Hua (2014) and Stengel (2004), how a pseudo-sigmoid function

$$\sigma(\alpha_k) = \frac{1 + \tanh\left(k_k(\alpha_{s,k}^2 - \alpha_k^2)\right)}{1 + \tanh(k_k\alpha_{s,k})} \tag{8}$$

can be used to blend pre- and post-stall regions together:

$$\begin{aligned} C_{D,k}(\alpha_k) &= \sigma(\alpha_k)(C_{D0,k} + C_{D\alpha_k,k}\alpha_k^2) \\ &\quad + (1 - \sigma(\alpha_k))(C_{0,k} + 2C_{1,k}\sin(\alpha_k)^2) \end{aligned} \tag{9}$$

$$\begin{aligned} C_{L,k}(\alpha_k) &= \sigma(\alpha_k)(C_{L0,k} + C_{L\alpha_k,k}\alpha_k) \\ &\quad + (1 - \sigma(\alpha_k))(C_{1,k}\sin(2\alpha_k)) \end{aligned} \tag{10}$$

where a polynomial function is used in pre-stall (Stengel, 2004) and the trigonometric post-stall terms are modeled as shown in Pucci, Hamel, Morin, and Samson (2013). The parameters are determined by augmenting and refining the data obtained from XFoil Software,[1] in Ducard and Hua (2014) with wind-tunnel like practical tests (Bauersfeld & Ducard, 2020; Spannagl & Ducard, 2020). Hereby, the wing is treated as a Clark-Y profile and the remaining passive sources as flat plates. The final values are presented in Table 3.

The torque generated by the aerodynamic-control surfaces $\boldsymbol{M}_\delta \in \mathbb{R}^3$ is controlled by changing the angle of the ailerons around $\boldsymbol{e}_x$ (δ_{A_r}, δ_{A_l}), the elevator around $\boldsymbol{e}_y$ (δ_E) and the rudders around $\boldsymbol{e}_z$ (δ_{R_r}, δ_{R_l}) as follows:

$${}^{\mathcal{B}}\boldsymbol{M}_\delta = \frac{\rho}{2}\begin{pmatrix} C_A\left(\left|{}^{\mathcal{B}}\boldsymbol{v}_{a,A_r}\right|^2\delta_{A_r} + \left|{}^{\mathcal{B}}\boldsymbol{v}_{a,A_l}\right|^2\delta_{A_l}\right) \\ C_E\left|{}^{\mathcal{B}}\boldsymbol{v}_{a,E}\right|^2\delta_E \\ C_R\left(\left|{}^{\mathcal{B}}\boldsymbol{v}_{a,R_r}\right|^2\delta_{R_r} + \left|{}^{\mathcal{B}}\boldsymbol{v}_{a,R_l}\right|^2\delta_{R_l}\right) \end{pmatrix} \tag{11}$$

where the velocity vectors $\boldsymbol{v}_{a,A_r}, \ldots, \boldsymbol{v}_{a,R_l}$ are computed according to (7) and the constant coefficients C_A, C_E, C_R depend on the geometry of each control surface. Note that the deflection angles are limited $\delta_k \in [-30°, 30°]$, $\forall k \in \{A_r, \ldots, R_l\}$. Since the control authority is dependent on the airspeed, it changes drastically across the flight envelope which must be considered during the actuator allocation process (see Section 4).

3. Guidance model predictive controller

As already mentioned in Section 1, the controller must be able to handle the changing lift generation and control authorities associated with the highly-nonlinear model derived in Section 2.3. We propose the control structure shown in Fig. 3 to tackle these challenges.

This section focuses on (a) the guidance-MPC which ensures feasible trajectory generation under the nonlinear EoM and (b) covers the rationale behind the attitude controller. Section 4 addresses the issue of changing control authorities and introduces the custom control allocation.

[1] http://airfoiltools.com/airfoil/

Table 4
System dimensions and parameters required for controller synthesis (determined by authors of Bauersfeld & Ducard, 2020; Spannagl & Ducard, 2020).

Parameter	Value
m, $\{I_{xx}^{\mathcal{B}}, I_{yy}^{\mathcal{B}}, I_{zz}^{\mathcal{B}}\}$	2.7 [kg], $\{0.1,\ 0.1,\ 0.1\}$[kgm^2]
$\{l_1, h_1, l_0, h_0, L_0\}$	$\{0.16,\ 0.05,\ 0.105,\ 0.015,\ 0.29\}$[m]
$\dot{\chi}_{max}$	$\frac{\pi}{2}$[rad/s]
$\{c_T, c_Q\}$	$\{1.11 \times 10^{-5},\ 1.99 \times 10^{-7}$ [m]$\}$[Ns^2/rad^2]
${}^{\mathcal{B}}r_{Wr}$ ${}^{\mathcal{B}}r_{Wl}$	$\begin{pmatrix}0 & \pm 0.5 & -0.015\end{pmatrix}^\top$[m]
${}^{\mathcal{B}}r_F$	$\begin{pmatrix}0.036 & 0 & -0.015\end{pmatrix}^\top$[m]
${}^{\mathcal{B}}r_{T_v}$	$\begin{pmatrix}-0.71 & 0 & -0.04\end{pmatrix}^\top$[m]
${}^{\mathcal{B}}r_{T_h}$	$\begin{pmatrix}-0.71 & 0 & -0.015\end{pmatrix}^\top$[m]
$\{S_{Wr}, S_{Wl}, S_F, S_{T_v}, S_{T_h}\}$	$\{0.21,\ 0.21,\ 0.055,\ 0.074,\ 0.047\}$ [m^2]
$\{C_A, C_E, C_R\}$	$\{0.095,\ 0.088,\ 0.088\}$ [m^3]

3.1. Problem formulation

The guidance-MPC algorithm is responsible for generating actuator–constraint-respecting control commands based on the vehicles EoM to smoothly track a set of reference waypoints. The following nonlinear optimization problem is solved at each iteration of the controller:

$$\min_{\left\{{}^2\boldsymbol{x}_n, \ldots, {}^{N+1}\boldsymbol{x}_n \atop {}^1\boldsymbol{u}_n, \ldots, {}^N\boldsymbol{u}_n\right\}} \sum_{j=1}^{N} h\left({}^j\Delta\boldsymbol{x}, {}^j\Delta\boldsymbol{u}\right) + h_{N+1}\left({}^{N+1}\Delta\boldsymbol{x}\right) \tag{12a}$$

$$s.t. \quad \dot{\boldsymbol{x}}_n = f(\boldsymbol{x}_n, \boldsymbol{u}_n) \tag{12b}$$

$${}^1\boldsymbol{x}_n = \boldsymbol{x}(t) \tag{12c}$$

$${}^j\boldsymbol{x}_n \in [\boldsymbol{x}_{min}, \boldsymbol{x}_{max}]\ \forall j \in \{1, \ldots, N+1\} \tag{12d}$$

$${}^j\boldsymbol{u}_n \in [\boldsymbol{u}_{min}, \boldsymbol{u}_{max}]\ \forall j \in \{1, \ldots, N\} \tag{12e}$$

3.2. System dynamics

The tiltrotor UAV dynamics are captured in the constraint (12b) to ensure that the obtained optimal solution respects the EoM (2). Solver convergence time is improved by reducing the model complexity as follows:

- Position states are neglected, as they do not affect the forces and moments acting on the vehicle.
- Actuator effects are lumped into a single virtual control torque $\boldsymbol{M}_n \in \mathbb{R}^3$:

$${}^{\mathcal{B}}\boldsymbol{M}_n = {}^{\mathcal{B}}\boldsymbol{M}_r + {}^{\mathcal{B}}\boldsymbol{M}_\delta = \sum_{i=1}^{4} {}^{\mathcal{B}}\boldsymbol{Q}_i + {}^{\mathcal{B}}\boldsymbol{d}_i \times {}^{\mathcal{B}}\boldsymbol{T}_i + {}^{\mathcal{B}}\boldsymbol{M}_\delta \tag{13}$$

- The propeller thrust vectors $\boldsymbol{T}_i$ in (4a) are combined into a single thrust T_n and tilt angle χ_n:

$${}^{\mathcal{B}}\boldsymbol{F}_n = T_n\boldsymbol{n}(\chi_n) = \sum_{i=1}^{4} {}^{\mathcal{B}}\boldsymbol{T}_i \tag{14}$$

with direction vector $\boldsymbol{n}$ as defined in Section 2.3.1.

Section 4 will show how to recover the individual actuator commands from $\boldsymbol{M}_n$ and $\boldsymbol{F}_n$.

Based on these points, the MPC system state $\boldsymbol{x}_n \in \mathbb{R}^{11}$ is constructed as $\boldsymbol{x}_n = \left({}^{\mathcal{I}}\boldsymbol{v}_n^\top\ \chi_n\ \boldsymbol{q}_{\mathcal{IB},n}^\top\ ({}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{IB},n})^\top\right)^\top$ where the index n is used to differentiate the MPC variables from the real-system

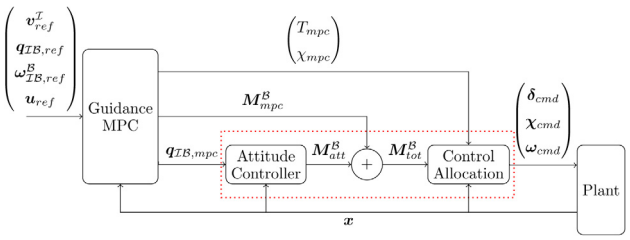

Fig. 3. Tiltrotor UAV control structure with guidance MPC controller running at 20 Hz and inner loop (red rectangle) attitude controller and allocation running at 250 Hz.

states. The first-stage state ${}^1\boldsymbol{x}_n$ is initialized with the system state $\boldsymbol{x}(t)$, according to (12c).

The control input vector $\boldsymbol{u}_n \in \mathbb{R}^5$ of the MPC system is defined to be $\boldsymbol{u}_n = \left(T_n \ \dot{\chi}_{n,cmd} \ (\boldsymbol{M}_n{}^{\mathcal{B}})^\top\right)^\top$. It is important to note that the rotor–tilt angle χ_n is treated as a state, while its rate $\dot{\chi}_{n,cmd}$ is used as a control input. This formulation allows for trivial inclusion of the maximum propeller tilt-rate $\dot{\chi}_{max}$ (see Eq. (18b)), which is the limiting factor for the minimum duration of the transition maneuver.

The dynamics of the MPC system state are summarized as:

$$\dot{\boldsymbol{v}}_n{}^{\mathcal{I}} = g\boldsymbol{e}_d{}^{\mathcal{I}} + \frac{1}{m} R_{\mathcal{IB},n}(\boldsymbol{q}_{\mathcal{IB},n}) \left(\boldsymbol{F}_n^{\mathcal{B}} + \boldsymbol{F}_a^{\mathcal{B}}\right) \tag{15a}$$

$$\dot{\chi}_n = \dot{\chi}_{n,cmd} \tag{15b}$$

$$\dot{\boldsymbol{q}}_{\mathcal{IB},n} = \frac{1}{2} \begin{pmatrix} 0 & -(\boldsymbol{\omega}_{\mathcal{IB},n}{}^{\mathcal{B}})^\top \\ \boldsymbol{\omega}_{\mathcal{IB},n}{}^{\mathcal{B}} & -[\boldsymbol{\omega}_{\mathcal{IB},n}{}^{\mathcal{B}}]_\times \end{pmatrix} \boldsymbol{q}_{\mathcal{IB},n} \tag{15c}$$

$$\dot{\boldsymbol{\omega}}_{\mathcal{IB},n}{}^{\mathcal{B}} = (I^{\mathcal{B}})^{-1} \left(-[\boldsymbol{\omega}_{\mathcal{IB},n}{}^{\mathcal{B}}]_\times I^{\mathcal{B}} \boldsymbol{\omega}_{\mathcal{IB},n}{}^{\mathcal{B}} + \boldsymbol{M}_a^{\mathcal{B}} + \boldsymbol{M}_n^{\mathcal{B}}\right) \tag{15d}$$

or in short $\dot{\boldsymbol{x}}_n = f(\boldsymbol{x}_n, \boldsymbol{u}_n)$, where the vectors $\boldsymbol{F}_a$, $\boldsymbol{M}_a$ are computed as shown in Eq. (5).

Potential issues with $\boldsymbol{q}_{\mathcal{IB},n}$ being no longer a unit quaternion due to solver discretization are resolved by re-normalization prior to the computation of $R_{\mathcal{IB},n}$.

3.3. Objective function

The objective function described in (12a) sums up individual stage costs h over the prediction horizon N and adds an additional terminal cost h_{N+1}. To simplify the tuning and reduce the solver convergence time, a quadratic stage cost is defined:

$$h\left({}^j\Delta\boldsymbol{x}, {}^j\Delta\boldsymbol{u}\right) = {}^j\Delta\boldsymbol{x}^\top Q \, {}^j\Delta\boldsymbol{x} + {}^j\Delta\boldsymbol{u}^\top R \, {}^j\Delta\boldsymbol{u} \tag{16a}$$

$$h_{N+1}\left({}^{N+1}\Delta\boldsymbol{x}\right) = {}^{N+1}\Delta\boldsymbol{x}^\top Q \, {}^{N+1}\Delta\boldsymbol{x} \tag{16b}$$

where $Q \in \mathbb{R}^{9\times9}$ and $R \in \mathbb{R}^{5\times5}$ are weighting matrices. The term ${}^j\Delta\boldsymbol{x}$ denotes the deviation between the predicted nominal and the desired state at MPC-prediction stage j, i.e. at time $t_j = t + (j-1)T_{MPC}$, where the sampling time is T_{MPC}. This tracking error is defined as:

$$\Delta\boldsymbol{x} = \begin{pmatrix} \boldsymbol{v}_n^{\mathcal{I}} - \boldsymbol{v}_{ref}^{\mathcal{I}} \\ 2\,\mathrm{sgn}(q_{e,0})\boldsymbol{q}_{e,v} \\ \boldsymbol{\omega}_{\mathcal{IB},n}^{\mathcal{B}} - \boldsymbol{\omega}_{\mathcal{IB},ref}^{\mathcal{B}} \end{pmatrix} \tag{17}$$

with error quaternion $\boldsymbol{q}_e = \boldsymbol{q}_{IB,n}^{-1}\boldsymbol{q}_{IB,ref}$. Note that the propeller-tilt angle χ_n does not accumulate any cost, since the controller should be able to choose it freely based on the current flight conditions. Similarly, the control input deviation ${}^j\Delta\boldsymbol{u}$ is defined as $\Delta\boldsymbol{u} = \boldsymbol{u}_n - \boldsymbol{u}_{ref}$.

3.4. Constraints

The actuator limitations of the aircraft impose box constraints on rotor speeds, tilt angles and rates, as well as surface deflection. Thus, for the MPC control input $\boldsymbol{u}_n$, the constraints in (12e) take the form:

$$T_n \in [0, 4c_T\omega_{max}^2] \tag{18a}$$

$$\dot{\chi}_{n,cmd} \in [-\dot{\chi}_{max}, \dot{\chi}_{max}] \tag{18b}$$

$$\boldsymbol{M}_n{}^{\mathcal{B}} \in [-M_{max}, M_{max}] \tag{18c}$$

which must hold at any stage $j \in \{1, \ldots, N\}$ and where ω_{max} is the maximum possible propeller speed. Hereby, simulations and experiments have confirmed the torque limit of $M_{max} = 2$ Nm to allow dynamic maneuvers in the MPC, while still leaving enough control authority for fast disturbance compensation in the inner-loop attitude controller during all flight conditions.

Regarding the state vector $\boldsymbol{x}_n$ and its constraints (12d), only the tilt angle χ_n needs to be limited within the actual physical range of χ_r, χ_l: $\chi_n \in \left[-\frac{\pi}{18}, \frac{\pi}{2}\right]$, which must hold for all stages $j \in \{1, \ldots, N+1\}$.

3.5. Control input synthesis

In order to use the solutions from the optimization problem (12) as control inputs, a real-time nonlinear MPC solver is required. In this work, the code generation feature of the ACADO Toolkit[2] and its inherent real-time iteration (RTI) algorithm (Houska, Ferreau, & Diehl, 2011) are used to derive a nonlinear solver with solving times in the millisecond range, without the need for linearization or excessive simplifications.

Once the solver has converged and an optimal and feasible solution is found, the following values will be extracted for further processing in the control structure:

$$T_{mpc} = {}^1u_{n,1} \tag{19a}$$

$$\chi_{mpc} = {}^2x_{n,4} \tag{19b}$$

$$\boldsymbol{M}_{mpc}{}^{\mathcal{B}} = \begin{pmatrix} {}^1u_{n,3} & {}^1u_{n,4} & {}^1u_{n,5} \end{pmatrix}^\top \tag{19c}$$

$$\boldsymbol{q}_{\mathcal{IB},mpc} = \begin{pmatrix} {}^2x_{n,5} & {}^2x_{n,6} & {}^2x_{n,7} & {}^2x_{n,8} \end{pmatrix}^\top \tag{19d}$$

where e.g. ${}^2x_{n,4}$ is the 4th element of the 2nd stage state vector in the MPC solution. Hereby, values from ${}^2\boldsymbol{x}_n$ are used, since the state ${}^1\boldsymbol{x}_n$ is given by the initialization constraint and not part of the computed MPC solution.

The commanded total thrust T_{mpc} and tilt angle χ_{mpc} are fed directly into the control allocation (see Section 4), while the attitude of the solution trajectory $\boldsymbol{q}_{\mathcal{IB},mpc}$ is used as a reference in the inner-loop attitude controller (see Section 3.5.1). The torque command of this low-level controller $\boldsymbol{M}_{att}{}^{\mathcal{B}}$ is then added to the MPC control torque $\boldsymbol{M}_{mpc}{}^{\mathcal{B}}$ before performing the allocation.

3.5.1. Attitude controller

Despite the high-quality solver, the MPC problem's high dimensionality and nonlinearity still result in convergence times that are incompatible with the fast dynamics of the vehicle, especially in RW mode.

Besides using an Euler-Forward state predictor to account for this policy lag, simulations and real hover experiments have shown tremendous performance improvements with the addition of a high-frequency attitude controller (geometric approach) and rate controller (PID approach). While tracking the optimal quaternion trajectory $\boldsymbol{q}_{\mathcal{IB},mpc}$, this low-level cascaded controller can adapt quickly to external disturbances and attitude-deviations caused by modeling errors.

[2] https://acado.github.io/index.html

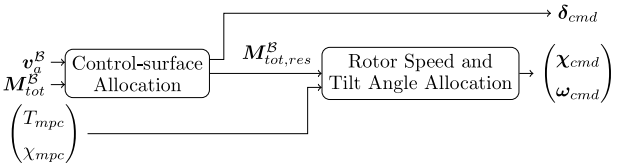

Fig. 4. Two-step daisy-chaining control allocation.

4. Control allocation

The eleven actuator commands are computed through the inversion of Eqs. (13) and (14), based on the current airspeed ${\boldsymbol{v}_a}^{\mathcal{B}} = {\boldsymbol{v}_W}^{\mathcal{B}} - {\boldsymbol{v}}^{\mathcal{B}}$ and the thrust T_{mpc}, tilt angle χ_{mpc} and total torque ${\boldsymbol{M}_{tot}}^{\mathcal{B}} = {\boldsymbol{M}_{mpc}}^{\mathcal{B}} + {\boldsymbol{M}_{att}}^{\mathcal{B}}$ commands of the MPC and attitude controller.

In order to efficiently resolve the overactuation and handle the changing control authorities, a two-step daisy-chaining approach (Johansen & Fossen, 2013) is used, as shown in Fig. 4.

4.1. Control-surface deflection

The first step of the daisy-chaining procedure focuses on the allocation of the control-surfaces. From an energy point of view, these control inputs are cheap compared to rotor speeds and are given priority in control allocation.

4.1.1. Allocation matrix

The complexity of the problem can be reduced by commanding symmetric deflections, i.e. $\delta_A = \delta_{A_r} = \delta_{A_l}$ and $\delta_R = \delta_{R_r} = \delta_{R_l}$. Furthermore, by neglecting the effect of $\boldsymbol{\omega}_{\mathcal{IB}}$ on the airspeed in Eq. (7) due to its small magnitude compared to $\boldsymbol{v}$, the resulting control torque in Eq. (11) can then be written as:

$$ {\boldsymbol{M}_\delta}^{\mathcal{B}} = \underbrace{\frac{\rho}{2}\left|{\boldsymbol{v}_a}^{\mathcal{B}}\right|^2 \operatorname{diag}(2C_A, C_E, 2C_R)}_{A_\delta} \begin{pmatrix}\delta_A \\ \delta_E \\ \delta_R\end{pmatrix} \tag{20} $$

where $A_\delta \in \mathbb{R}^{3\times 3}$ is called the allocation matrix.

4.1.2. Actuator allocation

In order to allocate as much moment as possible to the different aerodynamic-control surfaces, without violating the actuator constraints, the following operation is performed:

$$ \boldsymbol{\delta}_{cmd} = \begin{pmatrix}\delta_{A,cmd} \\ \delta_{E,cmd} \\ \delta_{R,cmd}\end{pmatrix} = \max\left(\min\left(\frac{\pi}{6}, A_\delta^{-1}{\boldsymbol{M}_{tot}}^{\mathcal{B}}\right), -\frac{\pi}{6}\right) \tag{21} $$

Note that the inverse of A_δ can be computed very efficiently at each iteration, since it is a diagonal matrix. Based on $\boldsymbol{\delta}_{cmd}$, the residual moment $\boldsymbol{M}_{tot,res} \in \mathbb{R}^3$ which still needs to be allocated is computed as:

$$ {\boldsymbol{M}_{tot,res}}^{\mathcal{B}} = {\boldsymbol{M}_{tot}}^{\mathcal{B}} - A_\delta \boldsymbol{\delta}_{cmd} \tag{22} $$

4.2. Rotor speed and tilt angle

The second step in the daisy-chaining allocation is related to rotor speed and rotor–tilt angle commands. Hereby, the right- and left-side tilt angles are allowed to deviate from the guidance-MPC command χ_{mpc} by an angle $\Delta\chi_r$, $\Delta\chi_l$ according to:

$$ \chi_r = \chi_{mpc} + \Delta\chi_r \qquad \chi_l = \chi_{mpc} + \Delta\chi_l \tag{23} $$

The freedom to have different tilt angles on both sides increases the overall control authority, especially torque generation around the axis $\boldsymbol{n}(\chi_{mpc})$ which would otherwise be limited to the comparatively weak propeller-resisting torque.

4.2.1. Allocation matrix

Neglecting the already allocated control surface torque $\boldsymbol{M}_\delta$, Eqs. (13) and (14) can be rewritten using the thrust-rotor-speed and torque–rotor–speed relationships in Eq. (4):

$$ \begin{pmatrix} T_{mpc}\cdot\sin\chi_{mpc} \\ T_{mpc}\cdot\cos\chi_{mpc} \\ {\boldsymbol{M}_{tot,res}}^{\mathcal{B}}\end{pmatrix} = A(\chi_r, \chi_l)\begin{pmatrix}\omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2\end{pmatrix} \tag{24} $$

However, solving this equation for the rotor speeds and tilt angles is hard, due to its nonlinearity in the matrix $A \in \mathbb{R}^{5\times 4}$ and the fact that we have six DoF for five equations.

Using the tilt angle parametrization in (23), as well as trigonometric identities and addition theorems, the expression can be transformed into a system of linear equations, which is easier to handle:

$$ \underbrace{\begin{pmatrix} T_{mpc}\sin\chi_{mpc} \\ T_{mpc}\cos\chi_{mpc} \\ {\boldsymbol{M}_{tot,res}}^{\mathcal{B}}\end{pmatrix}}_{\boldsymbol{f}_{des}} = A_\omega(\chi_{mpc}) \underbrace{\begin{pmatrix}\omega_1^2\sin\Delta\chi_r \\ \omega_1^2\cos\Delta\chi_r \\ \vdots \\ \omega_4^2\sin\Delta\chi_l \\ \omega_4^2\cos\Delta\chi_l\end{pmatrix}}_{\boldsymbol{v}} \tag{25} $$

where the approximation $\cos\Delta\chi_i \approx 1$ is used. Note that the allocation matrix $A_\omega \in \mathbb{R}^{5\times 8}$ is a function of the desired tilt angle χ_{mpc}, which is known at the time of the allocation. Also, the tilt angle deviations $\Delta\chi_r$, $\Delta\chi_l$ are constrained to be smaller than 10°, such that tilt angle dynamics can be neglected as they have already been considered in the computation of χ_{mpc}.

4.2.2. Actuator allocation

Based on Eq. (25), the rotor speed and tilt angle allocation can be done in two steps. First, $\boldsymbol{v}$ is computed as $\boldsymbol{v} = A_\omega^\dagger \boldsymbol{f}_{des}$. Hereby, the pseudo-inverse matrix $A_\omega^\dagger$ minimizes the term $\boldsymbol{v}^\top\boldsymbol{v}$, which is equal to the sum of squared rotor speeds and thus minimizes the energy consumption. Then, $\boldsymbol{v} = (v_1 \cdots v_8)^\top$ is used to compute the actuator commands for the tilt angles

$$ \Delta\chi_{r,cmd} = \arctan\left(\frac{v_1+v_3}{v_2+v_4}\right) \tag{26a} $$

$$ \Delta\chi_{l,cmd} = \arctan\left(\frac{v_5+v_7}{v_6+v_8}\right) \tag{26b} $$

$$ \boldsymbol{\chi}_{cmd} = \begin{pmatrix}\chi_{r,cmd} \\ \chi_{l,cmd}\end{pmatrix} = \begin{pmatrix}\chi_{mpc}+\Delta\chi_{r,cmd} \\ \chi_{mpc}+\Delta\chi_{l,cmd}\end{pmatrix} \tag{26c} $$

and rotor speeds

$$ \boldsymbol{\omega}_{cmd} = \begin{pmatrix}\omega_{1,cmd} \\ \omega_{2,cmd} \\ \omega_{3,cmd} \\ \omega_{4,cmd}\end{pmatrix} = \begin{pmatrix}\sqrt{v_1\sin\Delta\chi_{r,cmd} + v_2\cos\Delta\chi_{r,cmd}} \\ \sqrt{v_3\sin\Delta\chi_{r,cmd} + v_4\cos\Delta\chi_{r,cmd}} \\ \sqrt{v_5\sin\Delta\chi_{l,cmd} + v_6\cos\Delta\chi_{l,cmd}} \\ \sqrt{v_7\sin\Delta\chi_{l,cmd} + v_8\cos\Delta\chi_{l,cmd}}\end{pmatrix} \tag{27} $$

These formulations were chosen to minimize the effect of numerical inaccuracies. For example, it can happen that $\frac{v_1}{v_2} \neq \frac{v_3}{v_4}$ even though both terms algebraically evaluate to $\tan\Delta\chi_r$.

4.2.3. Actuator constraints

The described allocation method must respect the physical rotor speed and tilt angle constraints. Additionally, it has been mentioned, how the differential tilt angles $\Delta\chi_r$, $\Delta\chi_l$ should be kept small to justify the neglect of dynamic effects.

These limits are enforced by applying box constraints to Eqs. (26) and (27). However, this might cause deviations between the

Table 5
MPC weights used in flight experiments.

Weight	Value
$Q_{\Delta v}$	diag(50 50 50)[m^{-2} s^2]
$Q_{\Delta R_{\mathcal{IB}}}$	diag(1000 0.1 100)
$Q_{\Delta \omega_{\mathcal{IB}}}$	diag(10 0.1 10)[s^2]
$R_{\Delta T}$	0.05[N^{-2}]
$R_{\Delta \dot{\chi}}$	10[s^2]
$R_{\Delta \boldsymbol{M}}$	diag(50 50 50)[N^{-2} m^{-2}]

desired force–torque vector $\boldsymbol{f}_{des}$ and the actually produced one $\boldsymbol{f}_{act}$ (see Section 5).

Alternatively, it is possible to derive a set $\mathbb{F}$, such that $\boldsymbol{f}_{des} \in \mathbb{F}$ guarantees the allocated differential tilt angles and rotor speeds to be feasible. In case $\boldsymbol{f}_{des} \notin \mathbb{F}$, a constrained version can be constructed on the boundary of $\mathbb{F}$, e.g. through orthogonal projection or scaling. Compared to box constraints, this method allows $\boldsymbol{f}_{act}$ to be determined before the allocation is performed. The derivation of the set $\mathbb{F}$ is shown in the Appendix.

5. Results

The feasibility and performance of the proposed control structure are evaluated in real-world flight experiments. The general experimental setup is introduced in Section 5.1. Section 5.2 focuses exclusively on RW flight mode tests, while the experiments conducted in Section 5.3 cover the full flight envelope, including RW-to-FW and FW-to-RW transitions. Comparison against state-of-the-art approaches is discussed in Section 5.4 for simulation only.

Videos showcasing transition experiments from both onboard and offboard perspectives can be downloaded from the ETH Polybox at https://polybox.ethz.ch/index.php/s/oUtnNjgtCNTmuPk.

5.1. Experimental setup

All flight experiments were conducted outdoors with wind disturbances between 5 km h^{-1} and 10 km h^{-1}, using the tiltrotor UAV shown in Section 2.

The ACADO-generated MPC solver is setup with a sampling time of $T_{MPC} = 50$ ms and prediction horizon $N = 20$. The resulting time horizon of 1 s allows coverage of a full transition maneuver at maximum propeller-tilt rate. The code runs on a Raspberry Pi 3 B with average convergence time of 40 ms, thereby allowing sufficient buffer for deviations in the solve time when running at 20 Hz. The controller weights used during the experiments are listed in Table 5. In case the solver fails to converge in time or throws an error, it is reinitialized with an attitude corresponding to level hover and the control commands are taken from the last valid solution instead.

The inner-control loop is embedded into the PX4 software[3] and runs, together with the native state and wind estimator, at 250 Hz on a Pixhawk® 1 autopilot. Zero-order hold is used if no new MPC commands are available.

Velocity and yaw rate references were directly provided through the sticks on the RC, according to the PX4 MC Position Mode. Pitch and roll angles and rates are set to zero and similarly all control input references. The respective MPC weights are tuned to allow appropriate attitudes for a given maneuver (e.g. acceleration, deceleration) and optimize for efficient control commands.

[3] https://px4.io/

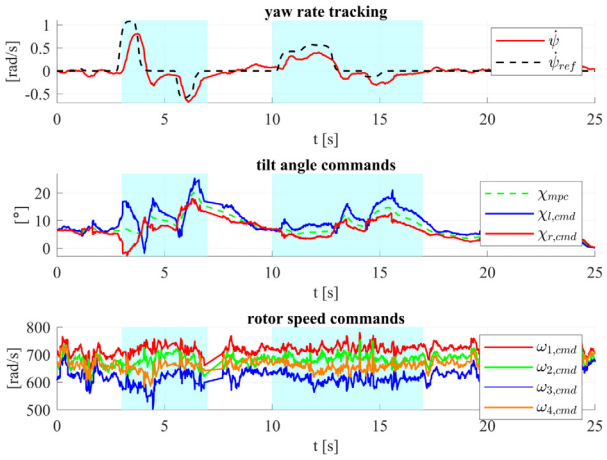

Fig. 5. Yaw-rate tracking performance and corresponding tilt angle and rotor speed commands for real-world hover tests in presence of wind disturbances (≈ 5 km h^{-1}) with non-zero yaw-rate references highlighted in cyan.

Table 6
Mean μ and standard deviation σ statistics of the relative error e_r between produced and desired thrusts and torques.

e_r	μ	σ
$e_r(T_x{}^{\mathcal{B}})$	0.7%	9.6%
$e_r(T_z{}^{\mathcal{B}})$	0.4%	6.4%
$e_r(M_x{}^{\mathcal{B}})$	0.7%	7.5%
$e_r(M_y{}^{\mathcal{B}})$	0.6%	11%
$e_r(M_z{}^{\mathcal{B}})$	0.4%	8.3%

5.2. Hover flight tests

The goal of the hover flight tests is to evaluate validity and efficiency of the thrust and tilt angle allocation. Validity addresses how well the requested forces and moments are matched by the actuator commands, despite the box constraints, numerical approximations and inaccuracies associated with the linear representation of the allocation equations. Efficiency of the commands is measured by comparing them to optimal solutions of Eq. (24) obtained in post-processing.

5.2.1. Results

Different yaw rate changes are commanded to excite the vehicle in hover. The references and corresponding tilt angle and rotor speed commands are displayed in Fig. 5 with regions of non-zero yaw-rate references highlighted in cyan. The differential-tilting performance of the allocation is well visible in these regions.

In order to validate the proposed control allocation, the actually-produced thrust and torque vector $\boldsymbol{f}_{act}$ is reconstructed from the actuator commands using Eq. (24) and compared to the originally desired one $\boldsymbol{f}_{des}$. According to the relative error statistics in Table 6, the values are almost identical, thus verifying the validity of the used approach. This conclusion allows the assumption that the poor yaw-rate tracking performance in Fig. 5 is related to insufficiently counteracted wind disturbances. Since the control allocation outputs match the desired forces and torques, the required yaw–torque magnitude must be too small to reject all disturbances, most likely due to the relation between the respective weights in the guidance MPC.

Concerning efficiency, solutions of Eq. (24) with minimum norm of squared rotor speeds were obtained using MATLAB's nonlinear `fmincon` optimizer. Mean and standard deviation of the relative error between these optimal $\boldsymbol{\omega}_{opt}$, $\boldsymbol{\chi}_{opt}$ and the commanded $\boldsymbol{\omega}_{cmd}$, $\boldsymbol{\chi}_{cmd}$ values are shown in Table 7.

Table 7
Mean μ and standard deviation σ statistics of the relative error e_r between optimal and commanded rotor speeds and tilt angles.

e_r	μ	σ
$e_r(\omega_1)$	0.4%	0.9%
$e_r(\omega_2)$	0.4%	0.8%
$e_r(\omega_3)$	0.5%	1.5%
$e_r(\omega_3)$	0.5%	0.8%
$e_r(\chi_r)$	0.8%	1.2%
$e_r(\chi_l)$	1%	1.7%

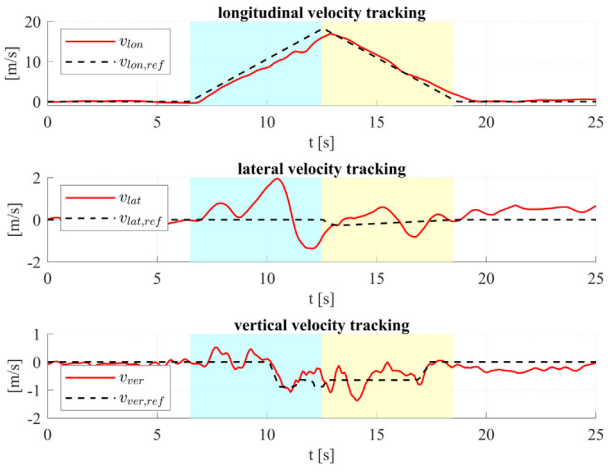

Fig. 6. Velocity reference and state trajectories for real-world transition experiments in presence of wind disturbances ($\approx 10\,\mathrm{km\,h^{-1}}$) with acceleration and deceleration phase highlighted in cyan and yellow, respectively.

These statistics suggest that the custom allocation outputs almost optimal and highly efficient actuator commands. Furthermore, the preference of using differential tilt angles rather than propeller resisting torque to produce yaw rotation in hover is confirmed by the optimal values.

5.3. Transition flight tests

The performance of the proposed control structure during the critical RW-to-FW and FW-to-RW transition phases is evaluated through an appropriate flight experiment. The vehicle is commanded to accelerate at a constant rate until FW mode is reached and lift is generated purely by the main wing. Once in FW mode, a constant deceleration rate is provided as a reference, in order to start the transition from FW back to RW mode.

The guidance-MPC algorithm should ensure sufficient lift generation at all times.

5.3.1. Results

The data collected during the transition experiment is plotted in Fig. 6, Figs. 7 and 8 with acceleration and deceleration phase highlighted in cyan and yellow, respectively. During acceleration, the tilt angle and rotor speed trajectories display a successful transition from RW to FW flight mode with increasing rotor–tilt angle and decreasing rotor speed, as lift generation is gradually taken over by the main wing. Fig. 6 further suggests that the guidance-MPC algorithm not only manages to maintain altitude but is also capable of increasing it ($t = 10$s) while still tracking the longitudinal velocity reference.

As the control surfaces become more effective with increasing airspeed, less residual torque $\boldsymbol{M}_{tot,res}$ needs to be produced by the propellers. A corresponding decrease in differential tilt and in the difference between rotor speeds can be observed as the

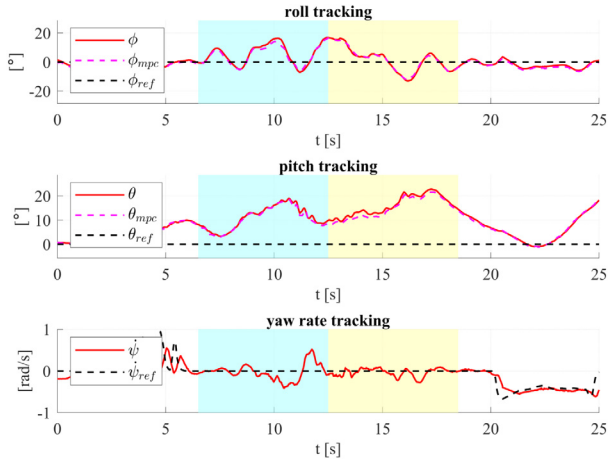

Fig. 7. Attitude reference and state trajectories for real-world transition experiments in presence of wind disturbances ($\approx 10\,\mathrm{km\,h^{-1}}$) with acceleration and deceleration phase highlighted in cyan and yellow, respectively.

Fig. 8. Actuator commands for real-world transition experiments in presence of wind disturbances ($\approx 10\,\mathrm{km\,h^{-1}}$) with acceleration and deceleration phase highlighted in cyan and yellow, respectively.

vehicle accelerates (see Fig. 8). Note that the control surfaces are switched off at low airspeeds to avoid bang–bang behavior.

In the deceleration phase, the guidance-MPC algorithm makes use of increased AoA-resulting drag forces by pitching up. Looking at Fig. 8, the rotors tilt back into RW configuration and spin up again to counteract gravity, as the airspeed-induced lift forces from the main wing decrease. Thereby, sufficient lift generation is ensured and allows vertical velocity tracking during the entire transition maneuver.

The deviations in lateral velocity originate from the non-zero roll angles observed in Fig. 7, which redirect the total aerodynamic lift force and thus cause lateral acceleration. These undesired non-zero roll angles are likely caused by unmodeled disturbance torques due to side winds. In particular, wind disturbances from the left side of the vehicle also explain the lateral drifting observed towards the end of the experiment ($t \in [20, 25]$s). Based on the references sent to the low-level attitude controller (see Fig. 7), it can again be stated that the MPC does not deploy enough control authority to fully counteract these disturbances.

5.4. Comparison with state-of-the-art

Based on the detailed vehicle model derived in Ducard and Hua (2014), a high-fidelity simulator was developed for initial

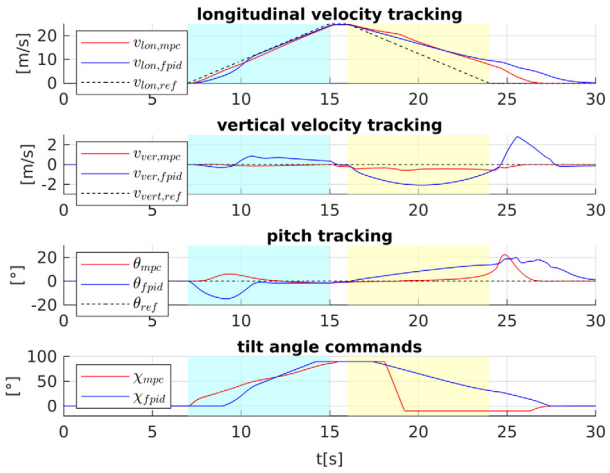

Fig. 9. Simulation results comparing MPC and FPID performance during transition phases with acceleration and deceleration phase highlighted in cyan and yellow, respectively.

testing of the controller and to resolve the trade-off between model complexity and computation time mentioned in Section 2. As the aircraft model used for simulation is more elaborate than the one used for control design, some robustness could already be accredited to the proposed approach. Indeed, the results in the previous section verify that one does not require a fully detailed system model for stable performance on a real-life platform in the presence of external disturbances and potential modeling errors.

This simulator is also used to compare our novel control algorithm to state-of-the-art approaches. As a representative example, the Fused PID (FPID) controller presented in Bauersfeld and Ducard (2020) is considered. Hereby, control commands from separate FW and RW controllers are mixed together according to the desired tilt angle, which in turn is scheduled based on the current airspeed. As mentioned in Section 1, such control authority weighting is a common control method for hybrid tiltrotor UAVs and has already been verified in real-world experiments. The performance of our MPC and the FPID approach in RW-to-FW and FW-to-RW transition simulations is shown in Fig. 9. While the numerical results strongly depend on the respective tuning, some fundamental differences between the two approaches can still be observed.

The simultaneous tracking performance and efficiency optimization in the MPC controller results in a nonlinear relationship between the longitudinal velocity and rotor–tilt angle values, as shown in Fig. 9. This contradicts the general intuition in the FPID and other GS-based control approaches. Additionally, it can be seen how the MPC accelerates only through increasing tilt angles and not by pitching down like the FPID, thus contributing to a better altitude tracking. Although the body dynamics are generally faster than the tilt-angle dynamics, negative pitch results in undesired downwards lift forces being generated by the main wing. Finally, the shortcoming of a static mapping between tilt angle and velocity used in FPID becomes clear during the deceleration phase, where the non-zero airspeed prevents the rotors from being rotated back into a RW configuration. Being able to use some upward thrust would have been beneficial to prevent the drop in altitude towards the end of the transition ($t \in [25, 27]$s) (Bauersfeld & Ducard, 2020). This FPID limitation does not appear with the novel MPC approach.

6. Conclusion

In this paper, a new MPC-based control structure for a tiltrotor UAV has been introduced and evaluated, running on-board the vehicle in real-time.

Using a nonlinear MPC solver, the guidance module computes optimal thrust, tilt angle and body torque control inputs for the provided reference velocity, thereby achieving automatic transition from RW to FW mode and back, all while considering the full nonlinear dynamics of the vehicle. Stability and tracking performance are improved through the addition of a high-frequency attitude controller. The allocation algorithm is capable of handling changing control authorities and ensures energy efficient actuation through the use of daisy-chaining control allocation and differential propeller tilt. This novel approach represents a unified flight control solution, valid throughout the whole flight-envelope without the need for controller switching or scheduling policies.

Practical experiments have verified the success of the proposed control structure. Validity and efficiency of the proposed control allocation were evaluated and confirmed in real experiments. Successful automatic transition from RW to FW flight mode and back was achieved in the presence of wind disturbances, showing the MPC's capabilities to ensure sufficient lift generation at all times despite the changing aerodynamic effects. Simulation results for comparison with a representative state-of-the-art approach were provided and showed that the novel MPC approach outperforms current mainstream approaches.

Regarding future work, it is expected that adding integral control or disturbance estimation capabilities to the MPC will significantly improve the overall tracking performance and result in an overall smoother transition maneuver. Further improvements in modeling error and disturbance robustness are anticipated when increasing the frequency of the MPC guidance module through more efficient solver implementations or additional simplifications in the optimization problem.

Acknowledgments

The authors would like to thank Leonard Bauersfeld and Lukas Spannagl for their help in measuring the aerodynamic coefficients, as well as in preparing and maintaining the prototype hardware.

Appendix. Derivation of feasible thrust and torque set $\mathbb{F}$

Based on (26), the following constraints must hold for the differential tilt angles $\Delta\chi_r$, $\Delta\chi_l$:

$$\Delta\chi_r = \arctan\left(\frac{v_1 + v_3}{v_2 + v_4}\right) \in \left[-\frac{\pi}{18}, \frac{\pi}{18}\right] \tag{A.1a}$$

$$\Delta\chi_l = \arctan\left(\frac{v_5 + v_7}{v_6 + v_8}\right) \in \left[-\frac{\pi}{18}, \frac{\pi}{18}\right] \tag{A.1b}$$

which are rewritten in matrix form:

$$C_{v,\Delta\chi}\boldsymbol{v} \leq \boldsymbol{b}_{v,\Delta\chi} \qquad C_{v,\Delta\chi} \in \mathbb{R}^{4\times 8} \tag{A.2}$$

Similarly, rotor speeds $\boldsymbol{\omega}$ must fulfill:

$$\omega_1^2 = v_1 s_{\Delta\chi_{r,cmd}} + v_2 c_{\Delta\chi_{r,cmd}} \in [0, \omega_{max}^2] \tag{A.3a}$$

$$\omega_2^2 = v_3 s_{\Delta\chi_{r,cmd}} + v_4 c_{\Delta\chi_{r,cmd}} \in [0, \omega_{max}^2] \tag{A.3b}$$

$$\omega_3^2 = v_5 s_{\Delta\chi_{l,cmd}} + v_6 c_{\Delta\chi_{l,cmd}} \in [0, \omega_{max}^2] \tag{A.3c}$$

$$\omega_4^2 = v_7 s_{\Delta\chi_{l,cmd}} + v_8 c_{\Delta\chi_{l,cmd}} \in [0, \omega_{max}^2] \tag{A.3d}$$

for all $\Delta\chi_r, \Delta\chi_l \in [-\frac{\pi}{18}, \frac{\pi}{18}]$. Considering only the angles which result in the smallest and largest rotor speeds, (A.3) is rewritten as shown on the example of (A.3a):

$$v_1 > 0: \quad \begin{matrix} v_1 \sin\frac{\pi}{18} + v_2 \\ -v_1 \sin\frac{\pi}{18} + v_2 \cos\frac{\pi}{18} \end{matrix} \quad \begin{matrix} \leq \omega_{max}^2 \\ \geq 0 \end{matrix} \tag{A.4a}$$

$$v_1 < 0: \quad \begin{matrix} -v_1 \sin\frac{\pi}{18} + v_2 & \leq \omega_{max}^2 \\ v_1 \sin\frac{\pi}{18} + v_2 \cos\frac{\pi}{18} & \geq 0 \end{matrix} \tag{A.4b}$$

Note that v_2 is always non-negative by definition. Applying (A.4) to each line in (A.3) and collecting the resulting 16 constraints in a matrix yield:

$$C_{v,\omega}\boldsymbol{v} \leq b_{v,\omega} \qquad C_{v,\omega} \in \mathbb{R}^{16\times 8} \tag{A.5}$$

The set of thrust and torque vectors

$$\mathbb{F} = \{\boldsymbol{f} \in \mathbb{R}^5 | C_f \boldsymbol{f} \leq \boldsymbol{b}_f\} \tag{A.6}$$

that result in feasible differential tilt angles and rotor speeds, i.e.

$$\boldsymbol{f}_{des} \in \mathbb{F} \Rightarrow \begin{matrix} \Delta\chi_{r,cmd}, \Delta\chi_{l,cmd} \in \left[-\frac{\pi}{18}, \frac{\pi}{18}\right] \\ \omega_{1,cmd}, \ldots, \omega_{4,cmd} \in [0, \omega_{max}] \end{matrix} \tag{A.7}$$

when applying the proposed allocation algorithm is then given as:

$$C_f = \begin{pmatrix} C_{v,\Delta\chi} \\ C_{v,\omega} \end{pmatrix} A_\omega^\dagger \qquad b_f = \begin{pmatrix} b_{v,\Delta\chi} \\ b_{v,\omega} \end{pmatrix} \tag{A.8}$$

Here, $C_f \in \mathbb{R}^{20\times 5}$ and $b_f \in \mathbb{R}^{20}$ form a conservative set, since the pseudo-inverse is an optimal solution to (25) in the unconstrained case.

References

Allenspach, M., & Ducard, G. (2020). Model predictive control of a convertible tiltrotor unmanned aerial vehicle. In *Proceedings of the 28th IEEE Mediterranean conference on control and automation*.

Anglade, A., Kai, J.-M., Hamel, T., & Samson, C. (2019). Automatic control of convertible fixed-wing drones with vectorized thrust. In *2019 IEEE 58th conference on decision and control*.

Bauersfeld, L., & Ducard, G. (2020). Fused PID control for tilt-rotor VTOL aircraft. In *Proceedings of the 28th IEEE Mediterranean conference on control and automation*.

Cetinsoy, E. (2015). Design and control of a gas-electric hybrid quad tilt-rotor UAV with morphing wing. In *2015 International conference on unmanned aircraft systems*.

Chao, C., Lincheng, S., Daibing, Z., & Jiyang, Z. (2016). Mathematical modeling and control of a tiltrotor UAV. In *2016 IEEE international conference on information and automation*.

Ducard, G., & Hua, M. (2014). Modeling of an unmanned hybrid aerial vehicle. In *2014 IEEE conference on control applications*.

Hernández-García, R. G., & Rodríguez-Cortés, H. (2015). Transition flight control of a cyclic tiltrotor UAV based on the gain-scheduling strategy. In *2015 International conference on unmanned aircraft systems*.

Houska, B., Ferreau, H. J., & Diehl, M. (2011). An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range. *Automatica*.

Johansen, T. A., & Fossen, T. I. (2013). Control allocation - A survey. *Automatica*.

Kai, J.-M. (2018). *Nonlinear automatic control of fixed-wing aerial vehicles* (Ph.D. theses), Université Côte d'Azur.

Kendoul, F., Fantoni, I., & Lozano, R. (2006). Modeling and control of a small autonomous aircraft having two tilting rotors. *IEEE Transactions on Robotics*.

Kvaternik, R., Piatak, D., Nixon, M., Langston, C., Singleton, J., Bennett, R., et al. (2001). An experimental evaluation of generalized predictive control for tiltrotor aeroelastic stability augmentation in airplane mode of flight. *Journal of the American Helicopter Society*.

Liu, Z., He, Y., Yang, L., & Han, J. (2017). Control techniques of tilt rotor unmanned aerial vehicle systems: A review. *Chinese Journal of Aeronautics*.

Mehra, R. K., Prasanth, R. K., & Gopalaswamy, S. (1998). XV-15 tiltrotor flight control system design using model predictive control. In *1998 IEEE aerospace conference proceedings (Cat. No.98TH8339)*.

Mehra, R., Wasikowski, M., Prasanth, R., Bennett, R., & Neckels, D. (2001). Model predictive control design for XV-15 tilt rotor flight control. In *AIAA guidance, navigation, and control conference and exhibit*.

Nakamura, Y., Arakawa, A., Watanabe, K., & Nagai, I. (2018). Transitional flight simulations for a tilted quadrotor with a fixed-wing. In *2018 IEEE international conference on mechatronics and automation*.

Papachristos, C., Alexis, K., & Tzes, A. (2013). Model predictive hovering-translation control of an unmanned Tri-TiltRotor. In *2013 IEEE international conference on robotics and automation*.

Park, S., Bae, J., Kim, Y., & Kim, S. (2013). Fault tolerant flight control system for the tilt-rotor UAV. *Journal of the Franklin Institute*.

Peng, C., Hwang, T., Chen, S., Chang, C., Lin, Y., Wu, Y., et al. (2010). ZPETC path-tracking gain-scheduling design and real-time multi-task flight simulation for the automatic transition of tilt-rotor aircraft. In *2010 IEEE conference on robotics, automation and mechatronics*.

Pucci, D., Hamel, T., Morin, P., & Samson, C. (2013). Nonlinear control of aerial vehicles subjected to aerodynamic forces. In *Proceedings of the IEEE conference on decision and control*.

Saeed, A. S., Younes, A. B., Islam, S., Dias, J., Seneviratne, L., & Cai, G. (2015). A review on the platform design, dynamic modeling and control of hybrid UAVs. In *2015 International conference on unmanned aircraft systems*.

Spannagl, L., & Ducard, G. (2020). Control allocation for an unmanned hybrid aerial vehicle. In *Proceedings of the 28th IEEE Mediterranean conference on control and automation*.

Stengel, R. F. (2004). *Flight dynamics*. Princeton University Press.

Yangping, D., & Honggang, G. (2018). Transition flight control and test of a new kind tilt prop box-wing VTOL UAV. In *2018 9th international conference on mechanical and aerospace engineering*.

Yatsun, A., Lushnikov, B., & Emelyanova, O. (2018). Motion control automation in the quadcopter convertiplane in a transient mode. In *2018 International Russian automation conference*. RusAutoCon.

Zhu, X. P., Fan, Y. H., & Yang, J. (2010). Design of tiltrotor flight control system using fuzzy sliding mode control. In *2010 International conference on measuring technology and mechatronics automation*.

Mike Allenspach received his Master degree in Robotics, Systems and Control in 2020 from ETH Zurich, Switzerland. He is currently pursuing a Ph.D. degree at the Autonomous Systems Lab at ETH Zurich. His research interests include estimation and optimal control of UAVs.

Guillaume Jacques Joseph Ducard received his Master degree in electrical engineering from ETH Zurich, Switzerland, in 2004. He completed his doctoral work in 2007 and his two-year Post-Doctoral Course in 2009, both about unmanned aerial vehicles (UAVs) at ETH Zurich. He is currently an Associate Professor at University of Nice Sophia Antipolis, France. His research activity is about nonlinear control, estimation and fault-tolerant flight control of UAVs.