Proposal of a Unified Control Strategy for Vertical Take-off and Landing Transition Aircraft Configurations

Stefan Raab*, Jiannan Zhang †, Pranav Bhardwaj‡ and Florian Holzapfel§
Institute of Flight System Dynamics, Technische Universität München, Garching, 85748, Germany

This paper proposes a unified controller based on an Incremental Nonlinear Dynamic Inversion (INDI) for vertical take-off and landing (VTOL) transition aircraft configurations. The proposed strategy attempts to encapsulate the control of different flight phases into one universal controller. The INDI control approach is used for dealing with systems that are non-affine in control inputs, which makes it suitable for controlling various kinds of VTOL aircraft configurations. Compared to a conventional cascaded loop, this concept enables the allocation of all desired forces and moments simultaneously. The controller comprises of a reference model, an Onboard Plant-Model, and error controller and a control allocation. While the reference model provides smooth and reasonable trajectories of the command values, the Onboard Plant-Model computes online estimates of the control effectiveness with respect to pseudo controls. Furthermore, the error controller is designed to achieve the desired error dynamics by a proper choice of feedback gains. The control allocation computes physically feasible control effector commands, providing the possibility of considering secondary constraints. The overall approach is extended by the concept of virtual control inputs – nested control feedback loops of the control allocation into the reference model. The control concept is applied to an experimental transition VTOL configuration including tilt rotors and aerodynamic surfaces. The control architecture as well as the aircraft configuration allows the unified design approach of the controller. Consequently the complete flight envelope utilizes only one baseline controller, eliminating the need for distinct controller variants for different maneuvers. Initial aeropropulsive interaction in the form of closed loop pitch oscillations are addressed and thereby enabled flight tests and corresponding results are presented.

I. Introduction

A. Motivation

TRANSITION vehicles have grown more and more popular over the last years, especially in the segment of Unmanned Aerial Vehicles (UAV). Recently even manned aircraft development contemplates with the idea of electric VTOL transition vehicles. This development emphasizes the urge for a modular, easy-to-adapt and intuitive control strategy. Within this article we propose our attempt of such a strategy. We want to provide one *unified* controller that is able to control continuously over the whole flight envelope of transition vehicles - hover, low speed and high speed. There shall neither be predefined maneuvers necessary nor a *switch* between different flight state specific controllers. If physically feasible, the controller shall be able to use all control effectors in an optimal way. There shall be no daisy chaining or consecutive effector allocation in order to enable the possibility to exchange vehicle configurations easier. The control from an operator point of view shall be intuitive over the whole flight envelope and not require deep knowledge in quadrocopter and fixed-wing aircraft control. The proposed strategy is based on Incremental Nonlinear Dynamic Inversion (INDI). The next section presents a rough overview of recent developments in INDI and VTOL UAV control, followed by an overview of the structure of the article.

B. State of the Art and Contribution

The origins of INDI trace back to [1–4] and are based on "nominal" nonlinear dynamic inversion (NDI), which is described in high detail in [5]. There is a broad usage of INDI for quadrocopters [6, 7, 9, 10].

*Research Assistant, E-Mail: stefan.raab@tum.de.
†Research Assistant, E-Mail: jiannan.zhang@tum.de.
‡Research Assistant, E-Mail: pranav.bhardwaj@tum.de.
§Professor, E-Mail: Florian.Holzapfel@tum.de, Associate Fellow AIAA.

Recent papers furthermore focus on robustness and stability analyses of INDI controlled systems [8, 11]. The control of VTOL transition UAVs using INDI was presented in [12–14] and successful in numerous simulations.

The difference to these approaches are the following: First, we want to propose a *unified* control strategy that uses one controller over the whole flight envelope, no predefined maneuvers, no switches and no *hard* changes of control from an operator point of view. Second, we want a full utilization of all available control effectors in a local, optimal way - no daisy chaining and no enabling/disabling over the flight envelope. Of course there is a global logic, which drives the control effectors naturally, but all effectors can be utilized if their control authority is required. Furthermore, we will use a combined inner and outer loop allocation as introduced in [6]. It shall be mentioned that there are two correlated papers accepted and published simultaneously, the first providing detailed explanation of the intuitive operator control and generation of reference trajectories [15] and the second dealing with the complex control allocation algorithms used in the proof of concept example for this control strategy [16].

C. Structure of the Article

The remainder of the article is structured in the following way: Chapter II presents required preliminaries, including a general description of the considered systems, an introduction to INDI and an introduction to the concept of virtual control inputs. The following Chapter III explains the proposed controller structure with all required submodules, which is applied on a VTOL transition vehicle for a proof of concept in Chapter IV. For this proof of concept example actual flight tests were performed, which is discussed in Chapter V. Furthermore, this chapter discusses a special case of aeropropulsive interaction, which was encountered during flight testing. The last Chapter VI presents a brief conclusion. Robustness and stability of the closed loop system are not assessed within this paper and is subject to future work.

II. Preliminaries

This section gives an overview of required preliminaries. Based on [5], at first the general set of considered systems and the respective nomenclature is introduced, followed by an introduction to INDI. The subsequent part explains the concept of virtual control inputs, which is based on [6].

A. System Description

Considered are multidimensional, nonlinear and time invariant systems, which can mathematically be described as a set of differential algebraic equations as follows:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)), \\ \boldsymbol{y}(t) &= \boldsymbol{h}(\boldsymbol{x}(t)), \end{aligned} \tag{1}$$

where $\boldsymbol{x} \in \mathbb{R}^{n_x}$ denotes the state vector, $\boldsymbol{u} \in \mathbb{R}^{n_u}$ the (control) inputs and $\boldsymbol{y} \in \mathbb{R}^{n_y}$ the outputs (or control variables) of the system. Furthermore, the vector functions $\boldsymbol{f} : \mathcal{D}_{x,u} \rightarrow \mathbb{R}^{n_x}$ and $\boldsymbol{h} : \mathcal{D}_x \rightarrow \mathbb{R}^{n_y}$ are sufficiently smooth and differentiable mappings. The notation of the explicit dependency on time t is omitted in the remainder for better readability. More detailed information can be found in [5]. In order to forge a bridge to the INDI introduction, the system described in Eq. (1) is transformed as explained in the following.

The number of times that a certain output y_i, with $i \in [1, ..., n_y]$, has to be differentiated with respect to time t until that derivative is directly affected by the input vector $\boldsymbol{u}$ will be denoted as relative degree r_i of the respective output. Mathematically this can be written as

$$\begin{aligned} \frac{\partial \overset{(r_i)}{y}_i}{\partial \boldsymbol{u}} &\neq \boldsymbol{0}, \\ \frac{\partial \overset{(k)}{y}_i}{\partial \boldsymbol{u}} &= \boldsymbol{0}, \end{aligned} \tag{2}$$

for $k \in [0, ..., r_i - 1]$. In this, $\overset{(r_i)}{y}_i$ reflects the r_ith and $\overset{(k)}{y}_i$ the kth derivative of y_i with respect to time t. The overall relative degree of the system is given by $r = r_1 + \cdots + r_{n_y}$ [5]. The highest derivatives of each output channel will be abbreviated with ν_i and the corresponding function is denoted as $F_i(\boldsymbol{x}, \boldsymbol{u})$.

Summarized, the transformed system description is given as

$$\nu_i = F_i(\boldsymbol{x}, \boldsymbol{u}) = {}^{(r_i)}y_i. \tag{3}$$

All output channels concatenated furthermore result in

$$\boldsymbol{\nu} = \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}). \tag{4}$$

The resulting vectorial function $\boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})$ therefore represents the direct algebraic relation between the control inputs $\boldsymbol{u}$ and the outputs $\boldsymbol{y}$ - to be more precise, their change with respect to time. Another important entity is given by the *external states*, which are defined for each output as follows

$$\boldsymbol{\xi}^i = \begin{bmatrix} \xi_1^i \\ \vdots \\ \xi_{r_i}^i \end{bmatrix} = \begin{bmatrix} {}^{(0)}y_i \\ \vdots \\ {}^{(r_i-1)}y_i \end{bmatrix} \in \mathbb{R}^{r_i}. \tag{5}$$

The external states describe the whole trajectory of the outputs and their time derivatives up to $r_i - 1$. The overall external states are concatenated by all output channel external states and are given by

$$\boldsymbol{\xi} = \begin{bmatrix} \boldsymbol{\xi}^1 \\ \vdots \\ \boldsymbol{\xi}^{n_y} \end{bmatrix} \in \mathbb{R}^r. \tag{6}$$

For more information the reader shall refer to [5]. The system equation given by Eq. (4), as well as the external states are used in the following section for the introduction of INDI.

B. Incremental Nonlinear Dynamic Inversion

The basic idea of dynamic inversion is to utilize the relation given by Eq. (4) to determine the control inputs $\boldsymbol{u}$ in a way such that the highest derivatives given by $\boldsymbol{\nu}$ follow a desired command $\boldsymbol{\nu}_{des}$. If the $\boldsymbol{\nu}$ of the plant could be specified directly, arbitrary desired reference dynamics can be imprinted. As the name already suggests, the concept of dynamic inversion plans to invert the dynamics given by Eq. (4) in order to achieve this goal. In general, the inversion can be written as

$$\boldsymbol{u} = \boldsymbol{F}^{-1}(\boldsymbol{\nu}, \boldsymbol{x}). \tag{7}$$

This assumes that the inversion is actually possible and the equation can be directly solved for $\boldsymbol{u}$. Applying this inversion law utilizing a desired $\boldsymbol{\nu}_{des}$, the system dynamics becomes

$$\begin{aligned} \boldsymbol{\nu} &= \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u}) \\ &= \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{F}^{-1}(\boldsymbol{\nu}_{des}, \boldsymbol{x})) \\ &= \boldsymbol{\nu}_{des}, \end{aligned} \tag{8}$$

and the plant would follow the specified $\boldsymbol{\nu}_{des}$ perfectly. However, this direct inversion is not applicable in general - the function might not be known perfectly and furthermore, the existence of global invertibility might not be guaranteed. Before continuing, the term *pseudo control* is introduced. As indicated in Eq. (8), the $\boldsymbol{\nu}$ of the plant will follow the desired $\boldsymbol{\nu}_{des}$ - the control input itself is solely used for canceling the system's dynamics. The term $\boldsymbol{\nu}_{des}$ therefore represents a *new* input to the closed loop system, which we will call *pseudo control*. To be more precise, when we use $\boldsymbol{\nu}$, we mean the current pseudo control or the output channels highest derivatives of the plant itself. When we use $\boldsymbol{\nu}_{des}$, we mean the current *desired* pseudo control.

The INDI approach differs in the sense, that the inversion is not applied globally but *incrementally*, or *locally*. For the derivation, the Taylor series of Eq. (4) with respect to $\boldsymbol{x}_0$ and $\boldsymbol{u}_0$ is considered [17]. It can be written as

$$\boldsymbol{\nu} = \boldsymbol{F}(\boldsymbol{x}_0, \boldsymbol{u}_0) + \frac{\partial \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}_0, \boldsymbol{u}_0} \cdot (\boldsymbol{x} - \boldsymbol{x}_0) + \frac{\partial \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{u}}\Big|_{\boldsymbol{x}_0, \boldsymbol{u}_0} \cdot (\boldsymbol{u} - \boldsymbol{u}_0) + \tilde{\boldsymbol{\Delta}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{x}_0, \boldsymbol{u}_0). \tag{9}$$

Using the substitutions

$$\begin{aligned}
\boldsymbol{\nu}_0 &= \boldsymbol{F}(\boldsymbol{x}_0, \boldsymbol{u}_0), \\
\boldsymbol{A}_0 &= \frac{\partial \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{x}}\Big|_{\boldsymbol{x}_0, \boldsymbol{u}_0}, \\
\boldsymbol{B}_0 &= \frac{\partial \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{u}}\Big|_{\boldsymbol{x}_0, \boldsymbol{u}_0}, \\
\Delta\boldsymbol{x} &= \boldsymbol{x} - \boldsymbol{x}_0, \\
\Delta\boldsymbol{u} &= \boldsymbol{u} - \boldsymbol{u}_0,
\end{aligned} \tag{10}$$

Eq. (9) follows as

$$\boldsymbol{\nu} = \boldsymbol{\nu}_0 + \boldsymbol{A}_0 \cdot \Delta\boldsymbol{x} + \boldsymbol{B}_0 \cdot \Delta\boldsymbol{u} + \tilde{\boldsymbol{\Delta}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{x}_0, \boldsymbol{u}_0), \tag{11}$$

where $\boldsymbol{\nu}$ represents the pseudo control of the plant given state $\boldsymbol{x}$ and input $\boldsymbol{u}$, and $\boldsymbol{\nu}_0$ the pseudo control of the plant given state $\boldsymbol{x}_0$ and input $\boldsymbol{u}_0$. The matrix $\boldsymbol{A}_0$ denotes the current (linear) sensitivity of the pseudo control with respect to a deviation $\Delta\boldsymbol{x}$ in the state vector, $\boldsymbol{B}_0$ the current (linear) sensitivity of the pseudo control with respect to a deviation $\Delta\boldsymbol{u}$ in the input vector. The term $\tilde{\boldsymbol{\Delta}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{x}_0, \boldsymbol{u}_0)$ describes the remaining higher order terms.

A linear approximation of Eq. (11) is obtained by simply neglecting the higher order terms $\tilde{\boldsymbol{\Delta}}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{x}_0, \boldsymbol{u}_0)$, which is valid in the vicinity of $\boldsymbol{x}_0$ and $\boldsymbol{u}_0$. Here we can furthermore trace an arc to the discrete nature of the latter controller design: If the sample time t_S of the controller is small enough, the change in state and input over one time step is sufficiently small. This may be seen as a first important assumption of INDI.

Finally, the linear approximation follows as

$$\boldsymbol{\nu} \approx \boldsymbol{\nu}_0 + \boldsymbol{A}_0 \cdot \Delta\boldsymbol{x} + \boldsymbol{B}_0 \cdot \Delta\boldsymbol{u}. \tag{12}$$

If we further assume that the increment $\Delta\boldsymbol{u}$ can be achieved instantaneous, thus without delay or dynamics, the term $\boldsymbol{A}_0 \cdot \Delta\boldsymbol{x}$ can be neglected as well. This follows from the fact, that a sudden change in $\boldsymbol{u}$ acts on $\dot{\boldsymbol{x}}$ rather than directly generating a $\Delta\boldsymbol{x}$. Stated like this, the assumption is not valid for real applications, as the control effectors themselves are most likely subject to at least some kind of dynamics (actuator dynamics, motor dynamics, etc.). However, if the control effector dynamics is fast enough in comparison to the plant dynamics, the assumption may still be seen as justified (compare [4]). Thus, we further omit the term $\boldsymbol{A}_0 \cdot \Delta\boldsymbol{x}$, which will lead to

$$\boldsymbol{\nu} \approx \boldsymbol{\nu}_0 + \boldsymbol{B}_0 \cdot \Delta\boldsymbol{u}. \tag{13}$$

In the following we assume a full row rank of the Jacobian $\boldsymbol{B}_0$, which means that its span includes the entire output space - by a proper choice of $\Delta\boldsymbol{u}$, any $\boldsymbol{\nu}$ may be reached for a fixed $\boldsymbol{\nu}_0$. Given this assumption, Eq. (13) can be solved for $\Delta\boldsymbol{u}$ by inverting $\boldsymbol{B}_0$. If the number of inputs n_u matches the number of outputs n_y a unique solution for the inverse exists. If there are more inputs than outputs ($n_u > n_y$) there may exist infinite solutions for the inverse. For simplification we will use $\boldsymbol{B}_0^{-1}$ in both cases, without specifying what inverse is used and how it is obtained. We can then write

$$\Delta\boldsymbol{u} = \boldsymbol{B}_0^{-1} \cdot (\boldsymbol{\nu} - \boldsymbol{\nu}_0). \tag{14}$$

Eq. (14) specifies the inversion law. Let $\hat{\boldsymbol{B}}$ denote our estimation of the true matrix $\boldsymbol{B}_0$ at a specified point in time (for later: at the current time step). The index 0 is dropped for simplicity. Then the required command $\boldsymbol{u}_{cmd}$ in order to achieve a desired $\boldsymbol{\nu}_{des}$ follows as

$$\begin{aligned}
\Delta\boldsymbol{u} &= \hat{\boldsymbol{B}}^{-1} \cdot (\boldsymbol{\nu}_{des} - \boldsymbol{\nu}_0), \\
\boldsymbol{u}_{cmd} &= \boldsymbol{u}_0 + \Delta\boldsymbol{u},
\end{aligned} \tag{15}$$

where $\boldsymbol{\nu}_0$ denotes the current plants pseudo control and $\boldsymbol{u}_0$ the current control input [17]. We will furthermore use the following abbreviation:

$$\Delta\boldsymbol{\nu} = \boldsymbol{\nu}_{des} - \boldsymbol{\nu}_0. \tag{16}$$

The term $\boldsymbol{\nu}_{des}$ consists of two parts, a feed forward part $\boldsymbol{\nu}_{ref}$ and a feedback part $\boldsymbol{\nu}_{ec}$. The feed forward part $\boldsymbol{\nu}_{ref}$ is provided by the reference model and specifies the pseudo control that is required such that the plant's output $\boldsymbol{y}$ follows the reference trajectory $\boldsymbol{y}_{ref}$ resulting from a command $\boldsymbol{y}_{cmd}$. The feedback part $\boldsymbol{\nu}_{ec}$ is calculated using the error between reference trajectory $\boldsymbol{\xi}_{ref}$ and plant trajectory $\boldsymbol{\xi}$ and is required for pulling back the plant to the desired trajectory in case of deviations [5].

C. Virtual Control Inputs

This section introduces the concept of virtual controls, which is used in this control structure to deal with changing relative degrees. The concept is based on [6]. We introduce the virtual control inputs as $\boldsymbol{y}_v$, which can be obtained by multiplying the state vector $\boldsymbol{x}$ with a respective selection matrix $\boldsymbol{C}_v$. We furthermore introduce a complementary matrix $\boldsymbol{C}_c$, such that

$$\begin{bmatrix} \boldsymbol{y}_v \\ \boldsymbol{y}_c \end{bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{C}_v \\ \boldsymbol{C}_c \end{bmatrix}}_{\boldsymbol{C}_{vc}} \cdot \boldsymbol{x}, \tag{17}$$

with $\boldsymbol{C}_{vc}$ being a unitary matrix. $\boldsymbol{y}_c$ reflects the complementary part of the transformed state. Using Eq. (16) and (17) and inserting them into Eq. (12) (with $\boldsymbol{\nu} = \boldsymbol{\nu}_{des}$) yields

$$\Delta\boldsymbol{\nu} \approx \underbrace{\boldsymbol{A}_0 \cdot \boldsymbol{C}_{vc}^{-1}}_{\boldsymbol{A}_{vc}} \cdot \begin{bmatrix} \Delta\boldsymbol{y}_v \\ \Delta\boldsymbol{y}_c \end{bmatrix} + \boldsymbol{B}_0 \cdot \Delta\boldsymbol{u}. \tag{18}$$

The matrix $\boldsymbol{A}_{vc}$ is furthermore separated in its parts corresponding to the inputs $\boldsymbol{y}_v$ and $\boldsymbol{y}_c$, which leads to

$$\Delta\boldsymbol{\nu} \approx \begin{bmatrix} \boldsymbol{A}_v & \boldsymbol{A}_c \end{bmatrix} \cdot \begin{bmatrix} \Delta\boldsymbol{y}_v \\ \Delta\boldsymbol{y}_c \end{bmatrix} + \boldsymbol{B}_0 \cdot \Delta\boldsymbol{u}. \tag{19}$$

Extending the equation and using the correlation $\Delta\boldsymbol{y}_c = \boldsymbol{C}_c \cdot \Delta\boldsymbol{x}$ furthermore yields

$$\Delta\boldsymbol{\nu} \approx \boldsymbol{A}_c \cdot \boldsymbol{C}_c \cdot \Delta\boldsymbol{x} + \boldsymbol{A}_v \cdot \Delta\boldsymbol{y}_v + \boldsymbol{B}_0 \cdot \Delta\boldsymbol{u}. \tag{20}$$

Neglecting the first term $\boldsymbol{A}_c \cdot \boldsymbol{C}_c \cdot \Delta\boldsymbol{x}$ due to the aforementioned reasons and rearranging finally results in

$$\Delta\boldsymbol{\nu} \approx \begin{bmatrix} \boldsymbol{A}_v & \boldsymbol{B}_0 \end{bmatrix} \cdot \begin{bmatrix} \Delta\boldsymbol{y}_v \\ \Delta\boldsymbol{u} \end{bmatrix}, \tag{21}$$

which can be inverted in order to achieve

$$\begin{bmatrix} \Delta\boldsymbol{y}_v \\ \Delta\boldsymbol{u} \end{bmatrix} = \left(\begin{bmatrix} \boldsymbol{A}_v & \boldsymbol{B}_0 \end{bmatrix} \right)^{-1} \cdot \Delta\boldsymbol{\nu}. \tag{22}$$

It should be mentioned, that the matrix $\boldsymbol{A}_v$ represents the Jacobian of $\boldsymbol{\nu}$ with respect to $\boldsymbol{y}_v$ and is given by

$$\boldsymbol{A}_v = \left. \frac{\partial \boldsymbol{F}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{y}_v} \right|_{\boldsymbol{x}_0, \boldsymbol{u}_0}. \tag{23}$$

The reason behind this derivation is to allow the desired increment in pseudo control to be distributed not just to the actual control inputs but also to certain states of the system. The virtual controls may be seen as slow actuators from the inversion, which of course might lead to transient errors, depending on how fast the controller is able to achieve the change in these states. As an example the reader shall be referred to [6], where this concept is used in a quadrocopter control, using Euler angles as virtual controls. A similar approach is chosen in the controller structure presented in Chapter IV. Within the inversion, the virtual controls are treated as nominal control inputs, but the procedure after the distribution is different. The increments $\Delta\boldsymbol{y}_v$ are added - similar as $\boldsymbol{u}_{cmd} = \boldsymbol{u}_0 + \Delta\boldsymbol{u}$ - to the current value $\boldsymbol{y}_{v,0}$ in order to obtain a command $\boldsymbol{y}_{v,cmd} = \boldsymbol{y}_{v,0} + \Delta\boldsymbol{y}_v$. This command is fed back to the reference model alongside the pilot commands, which are denoted by $\boldsymbol{y}_{p,cmd}$.

The main advantages are twofold. The first advantage is the possibility to treat a changing relative degree in another way than applying dynamic extension ([18]). Especially for the VTOL transition vehicle, the following has to be considered: In hover flight, the lift force - or vertical load factor n_z - is generated by propellers, which means a direct force control. Neglecting motor dynamics, there is a direct algebraic relation between propeller rotational rate ω_{prop} (control input) and load factor n_z (pseudo control).

In wingborne flight, the load factor shall be provided solely by aerodynamic lift, which is mainly depending on the angle of attack. This angle has to be built up by a pitch rate, which follows a pitch acceleration. The control inputs (here the elevator η) is therefore on an algebraic level with the pitch acceleration $\dot{q}$, or the second derivative of the angle of attack, respectively. There are two more integrations from control input to load factor response in wingborne flight, increasing the relative degree by two. This circumstance could be treated with dynamic extension ([18]), which would be applied as follows: The load factor is differentiated with respect to time two more times, revealing a direct algebraic relation between $\dddot{n}_z$ and $\ddot{\omega}_{prop}$. The latter $\ddot{\omega}_{prop}$ is treated as new control input, which is calculated utilizing the aforementioned inversion law and then integrated twice to obtain the actual control effector command ω_{prop}. The disadvantage is that INDI requires a current pseudo control measurement or estimation $\boldsymbol{\nu}_0$ for the inversion law, which would be given by the second derivative of the load factor in this case. Especially for multirotor systems, already the load factor measurements are subject to major vibrations. For higher order estimates, high pass filters have to be utilized, which even increase the effect of measurement vibrations on the estimate, degrading the performance furthermore.

The second advantage correlates to the specified reference dynamics. If the virtual control inputs are commanded, envelope protections within the reference model ([15]) can be used to ensure the specified state envelope. The saturations of the virtual control inputs can be taken into account at the same time by the control allocation ([16]), such that there are no commands outside the envelope. Using dynamic extension, the latter might be included as well, but involve more complexity and lose the "straight forward nature".

The next section illustrates and discusses the resulting controller structure.

III. Controller Structure

A. Overall Structure

Fig. 1 shows the considered, simplified closed loop system. The *Plant* includes the *PlantDynamics* itself, as well as the control effector dynamics represented by the block *Control Effectors* and simplified, linear relationship $\boldsymbol{u} = \boldsymbol{G}_A(s) \cdot \boldsymbol{u}_{cmd}$ with $\boldsymbol{G}_A(s)$ being the control effector transfer matrix. The block *Sensors & INS* shall represent all possible and available feedbacks $\boldsymbol{z}$ to the plant. This may include actuator and control effector measurements, state and even state derivative measurements. The signals that are available for the controller presented in this article are explained in more detail in later sections.

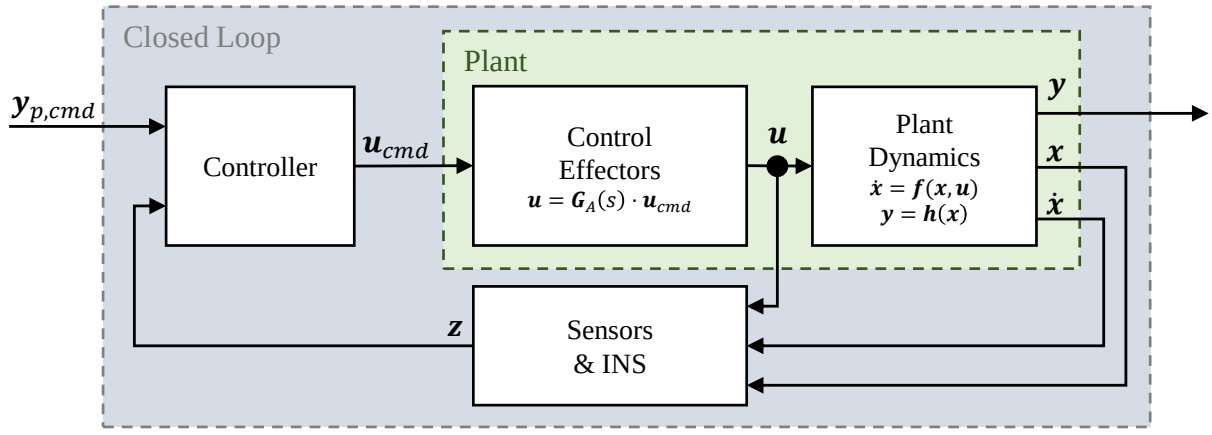


Fig. 1 Simplified closed loop block diagram.

The *Controller* is provided with this plant feedback $\boldsymbol{z}$, as well as with the pilot commands denoted with $\boldsymbol{y}_{p,cmd}$. The aforementioned virtual feedback commands $\boldsymbol{y}_{v,cmd}$ is only existing within the Controller and is thus not shown in this block diagram. A more detailed overview on the controller itself is presented in Fig. 2.

Special attention will be paid to the highlighted parts, *E* (Estimation), *EC* (Error Controller) and *OBPM* (Onboard-Plant-Model). The remaining modules are introduced in more detail in separate articles, the reader may refer to [15] for the module *RM* (Reference Model) and to [16] for the module *CA* (Control Allocation). The following subsections explain the different submodules in more detail.

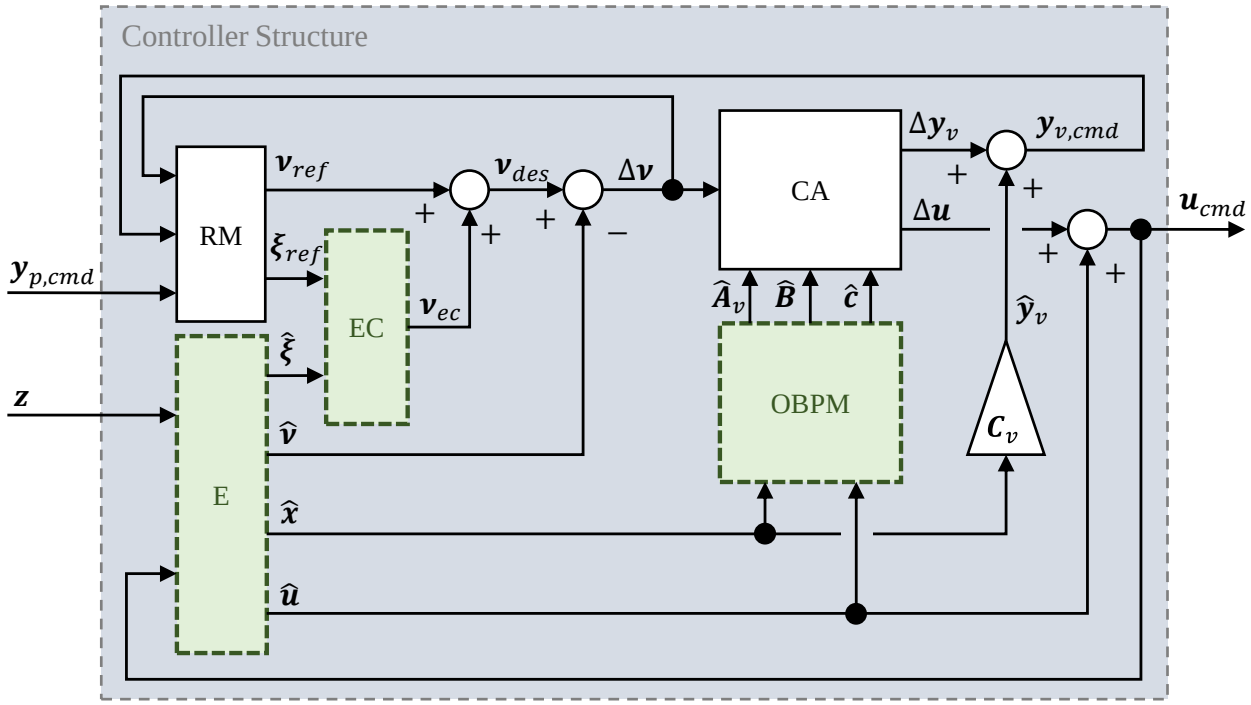


Fig. 2 Controller structure with highlighted modules.

B. Estimation (E)

The *Estimation* submodule is a placeholder for all required onboard data estimation. Given the plant feedback $\boldsymbol{z}$ and the controller commands $\boldsymbol{u}_{cmd}$, the *Estimation* calculates and provides an estimation of the current external states $\hat{\boldsymbol{\xi}} = \hat{\boldsymbol{\xi}}_0$, the current pseudo control $\hat{\boldsymbol{\nu}} = \hat{\boldsymbol{\nu}}_0$, the current states $\hat{\boldsymbol{x}} = \hat{\boldsymbol{x}}_0$ and the current control inputs $\hat{\boldsymbol{u}} = \hat{\boldsymbol{u}}_0$. The index 0 is dropped for simplicity in the remainder. Whether the estimations are purely based on measurement, on models or using a combination of both (f.e. in complementary filters) is dependent on the application.

C. Reference Model (RM)

As already mentioned before, the *Reference Model* is provided with commands of the desired control variables. The control variables are built up of two different kinds: pilot commands $\boldsymbol{y}_{p,cmd}$, which are directly forwarded from the pilot (or operator), and virtual feedback commands $\boldsymbol{y}_{v,cmd}$, which are commanded from the control allocation. Given these commands, the reference model provides smooth trajectories for each output (control variable) channel, given by reference pseudo control $\boldsymbol{\nu}_{ref}$ and by the reference external state trajectory $\boldsymbol{\xi}_{ref}$. Besides the virtual control feedback there is an additional feedback of $\Delta\boldsymbol{\nu}$, which is used for Pseudo Control-Hedging (PCH). As mentioned above, a very detailed explanation on the reference model, in particular - integrated reference model - may be found in [15].

D. Error Controller (EC)

The *Error Controller* is required for pulling back the plant external state trajectory to the reference trajectory in case of a deviation. The reference model only provides the feed forward command in order to follow the reference trajectory, there is no feedback or information about the actual deviation. The inversion is canceling the dynamics of the plant and the resulting system in an ideal case is given by a chain of integrators between the desired $\boldsymbol{\nu}_{des}$ and the plant output $\boldsymbol{y}$

in each of the output channels. Any deviation in the highest derivative (pseudo control) would act as an output error, integrated r_i times [5]. Purpose of the error controller is to move the closed loop poles of the resulting system, which all lay in the origin given a perfect inversion, into the left complex half plane in order to ensure disturbance rejection and tracking. The resulting error dynamics can be specified by feeding back the trajectory error χ with proper gains. Following [5], the error is defined as

$$\chi = \xi_{ref} - \hat{\xi}, \tag{24}$$

where we already use the external state estimate rather than the "true" external state vector ξ. The error trajectory of each channel given by $\chi^i = \xi^i_{ref} - \hat{\xi}^i$ is thus multiplied with a gain matrix $\left(c^T_{ec}\right)_i$, which is represented by a row vector in this case. The gains have to be specified based on the physical capabilities of the plant and control effector dynamics. Furthermore, the errors in the outputs, which are given by χ^i_1, are used for a integral feedback in order to compensate for steady state errors in the pseudo control channel. The respective gains are denoted with $(c_I)_i$. A block diagram of the error controller is presented in Fig. 3.

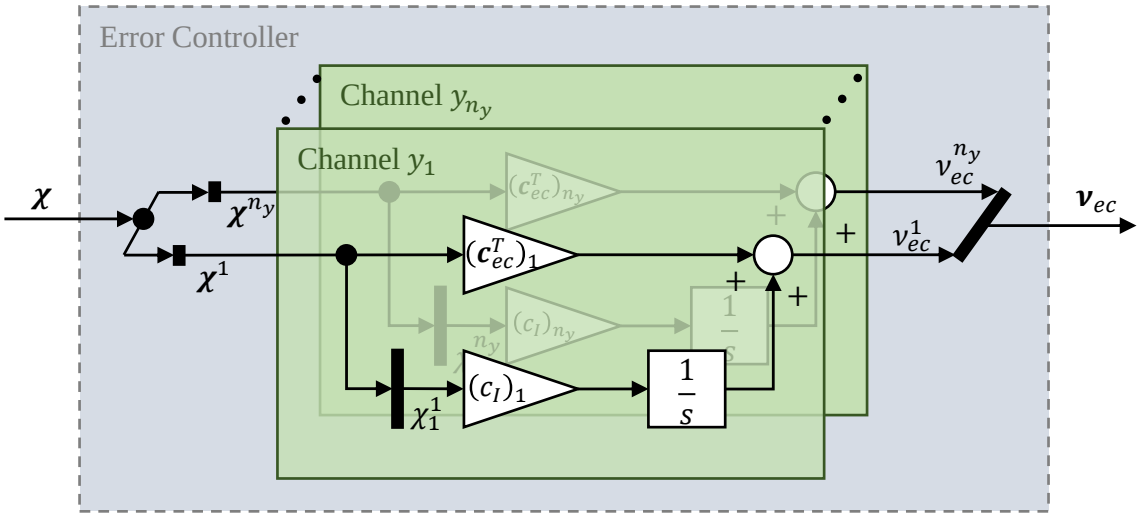


Fig. 3 Error controller block diagram.

The overall operation can be simplified in the following matrix equation

$$\begin{aligned}\nu_{ec} = &\begin{bmatrix} \left(c^T_{ec}\right)_1 & \mathbf{0}^{1\times r_2} & \mathbf{0}^{1\times r_3} & \cdots & \mathbf{0}^{1\times r_{n_y}} \\ \mathbf{0}^{1\times r_1} & \left(c^T_{ec}\right)_2 & \mathbf{0}^{1\times r_3} & \cdots & \mathbf{0}^{1\times r_{n_y}} \\ \mathbf{0}^{1\times r_1} & \mathbf{0}^{1\times r_2} & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \mathbf{0}^{1\times r_{n_y}} \\ \mathbf{0}^{1\times r_1} & \mathbf{0}^{1\times r_2} & \cdots & \mathbf{0}^{1\times (r_{n_y}-1)} & \left(c^T_{ec}\right)_{n_y} \end{bmatrix} \cdot \begin{bmatrix} \chi^1 \\ \chi^2 \\ \vdots \\ \vdots \\ \chi^{n_y} \end{bmatrix} \\ &+ \int_0^t \begin{bmatrix} (c_I)_1 & 0 & 0 & \cdots & 0 \\ 0 & (c_I)_2 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & 0 & \cdots & 0 & (c_I)_{n_y} \end{bmatrix} \cdot \begin{bmatrix} \chi^1_1(\tau) \\ \chi^2_1(\tau) \\ \vdots \\ \vdots \\ \chi^{n_y}_1(\tau) \end{bmatrix} \mathrm{d}\tau,\end{aligned} \tag{25}$$

with $\left(c^T_{ec}\right)_i$ and $(c_I)_i$ being the tuning parameters ("Gains"), specifying the error dynamics of the respective output channel. Note that countermeasures for integrator wind up are not explained explicitly, although the controller of the presented application incorporates saturation protections and PCH.

E. Onboard-Plant-Model (OBPM)

The *Onboard-Plant-Model* (OBPM) provides the required Jacobian matrices $\hat{\boldsymbol{A}}_v$ and $\hat{\boldsymbol{B}}$. Furthermore, it provides values $\hat{\boldsymbol{c}}$ for secondary objective constraints, which are utilized by the control allocation. E.g. they are used to drive control effectors in certain desired positions. More details on the constraints can be found in [16]. The Jacobian matrices are calculated using one sided numerical perturbation. In words, the system function $\hat{\boldsymbol{F}}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}})$ is evaluated once normally in order to achieve a "nominal" pseudo control output $\hat{\boldsymbol{\nu}} = \hat{\boldsymbol{F}}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}})$ and once one particular input is perturbed in order to obtain a "perturbed" output $\hat{\boldsymbol{\nu}}_{\Delta u_k} = \hat{\boldsymbol{F}}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}_{\Delta u_k})$. The respective Jacobian with respect to the control effectors inputs follows as

$$\begin{aligned}\hat{\boldsymbol{B}} = \frac{\partial \hat{\boldsymbol{F}}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{u}}\Big|_{\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}} \approx (&\begin{bmatrix} \hat{\boldsymbol{F}}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}_{\Delta u_1}) & \cdots & \hat{\boldsymbol{F}}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}_{\Delta u_{n_u}}) \end{bmatrix} \\ &- \begin{bmatrix} \hat{\boldsymbol{F}}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}) & \cdots & \hat{\boldsymbol{F}}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}) \end{bmatrix}) \cdot \begin{bmatrix} \Delta u_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \Delta u_{n_u} \end{bmatrix}^{-1},\end{aligned} \tag{26}$$

where, $\hat{\boldsymbol{u}}_{\Delta u_i}$ represents the input vector with a perturbation of Δu_i on the ith input $\hat{u}_i$. Using Eq. (17), the Jacobian with respect to the virtual control inputs can be found in a similar way, following

$$\begin{aligned}\hat{\boldsymbol{A}}_v = \frac{\partial \hat{\boldsymbol{F}}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{y}_v}\Big|_{\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}} \approx (&\begin{bmatrix} \hat{\boldsymbol{F}}(\boldsymbol{C}_{vc}^{-1} \cdot \begin{bmatrix} \hat{\boldsymbol{y}}_{v_{\Delta y_{v,1}}}^T & \hat{\boldsymbol{y}}_c^T \end{bmatrix}^T, \hat{\boldsymbol{u}}) & \cdots & \hat{\boldsymbol{F}}(\boldsymbol{C}_{vc}^{-1} \cdot \begin{bmatrix} \hat{\boldsymbol{y}}_{v_{\Delta y_{v,n_{y_v}}}}^T & \hat{\boldsymbol{y}}_c^T \end{bmatrix}^T, \hat{\boldsymbol{u}}) \end{bmatrix} \\ &- \begin{bmatrix} \hat{\boldsymbol{F}}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}) & \cdots & \hat{\boldsymbol{F}}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}) \end{bmatrix}) \cdot \begin{bmatrix} \Delta y_{v,1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \Delta y_{v,n_{y_v}} \end{bmatrix}^{-1}.\end{aligned} \tag{27}$$

In this equation, n_{y_v} specifies the number of virtual control inputs and $\Delta y_{v,k}$ for $k \in [1, ..., n_{y_v}]$ represents the perturbation increment on the kth virtual control input. As already mentioned before, the OBPM provides estimates of secondary constraints, which can be used by the Control Allocation. In order to do so, not just the actual values have to be estimated, but also the Jacobians of the constraints with respect to control inputs and virtual controls. For this, the following vector function is introduced, which represents the mathematical formulation of the constraints. It is given by a general function

$$\hat{\boldsymbol{c}} = \hat{\boldsymbol{F}}_c(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}), \tag{28}$$

with mapping $\hat{\boldsymbol{F}}_c : \mathcal{D}_{x,u} \rightarrow \mathbb{R}^{n_c}$ and n_c being the number of constraints. It should be mentioned, that the number of constraints is less or equal than the control effector excess, thus $n_y + n_c \leq n_u$.
If there are less constraints, the closed loop system is not fully determined and the solutions of the absolute control effector values might diverge in an undesired way. More details are presented in [16]. Since the vector function is defined, the Jacobians follow similar as before. The sensitivity with respect to the control inputs yields

$$\begin{aligned}\hat{\boldsymbol{B}}_c = \frac{\partial \hat{\boldsymbol{F}}_c(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{u}}\Big|_{\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}} \approx (&\begin{bmatrix} \hat{\boldsymbol{F}}_c(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}_{\Delta u_1}) & \cdots & \hat{\boldsymbol{F}}_c(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}_{\Delta u_{n_u}}) \end{bmatrix} \\ &- \begin{bmatrix} \hat{\boldsymbol{F}}_c(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}) & \cdots & \hat{\boldsymbol{F}}_c(\hat{\boldsymbol{x}}, \hat{\boldsymbol{u}}) \end{bmatrix}) \cdot \begin{bmatrix} \Delta u_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \Delta u_{n_u} \end{bmatrix}^{-1},\end{aligned} \tag{29}$$

and the sensitivity with respect to the virtual control inputs follows as

$$\hat{\boldsymbol{A}}_{c_v} = \frac{\partial \hat{\boldsymbol{F}}_c(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{y}_v}\Big|_{\hat{\boldsymbol{x}},\hat{\boldsymbol{u}}} \approx (\begin{bmatrix} \hat{\boldsymbol{F}}_c(\boldsymbol{C}_{vc}^{-1}\cdot\begin{bmatrix} \hat{\boldsymbol{y}}_{v_{\Delta y_{v,1}}}^T & \hat{\boldsymbol{y}}_c^T \end{bmatrix}^T,\hat{\boldsymbol{u}}) & \cdots & \hat{\boldsymbol{F}}_c(\boldsymbol{C}_{vc}^{-1}\cdot\begin{bmatrix} \hat{\boldsymbol{y}}_{v_{\Delta y_{v,n_{y_v}}}}^T & \hat{\boldsymbol{y}}_c^T \end{bmatrix}^T,\hat{\boldsymbol{u}}) \end{bmatrix} - \begin{bmatrix} \hat{\boldsymbol{F}}_c(\hat{\boldsymbol{x}},\hat{\boldsymbol{u}}) & \cdots & \hat{\boldsymbol{F}}_c(\hat{\boldsymbol{x}},\hat{\boldsymbol{u}}) \end{bmatrix}) \cdot \begin{bmatrix} \Delta y_{v,1} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \Delta y_{v,n_{y_v}} \end{bmatrix}^{-1}. \tag{30}$$

The OBPM provides the Control Allocation with these matrices $\hat{\boldsymbol{B}}_c$ and $\hat{\boldsymbol{A}}_{c_v}$, although it is not explicitly shown in Fig. 2.

F. Control Allocation (CA)

The *Control Allocation* (CA) deals with inverting the obtained Jacobians and finding solutions $\Delta\boldsymbol{y}_v$ and $\Delta\boldsymbol{u}$ to the problem given by

$$\Delta\boldsymbol{\nu} = \begin{bmatrix} \hat{\boldsymbol{A}}_v & \hat{\boldsymbol{B}} \end{bmatrix} \cdot \begin{bmatrix} \Delta\boldsymbol{y}_v \\ \Delta\boldsymbol{u} \end{bmatrix}, \tag{31}$$

including secondary constraints formulated by $\hat{\boldsymbol{c}}$. More details on the actual control allocation algorithm can be found in the associated paper [16].

The next section presents the application of the previous explained control strategy on a VTOL transition UAV and highlights further adaptions that were required for that special kind of vehicle.

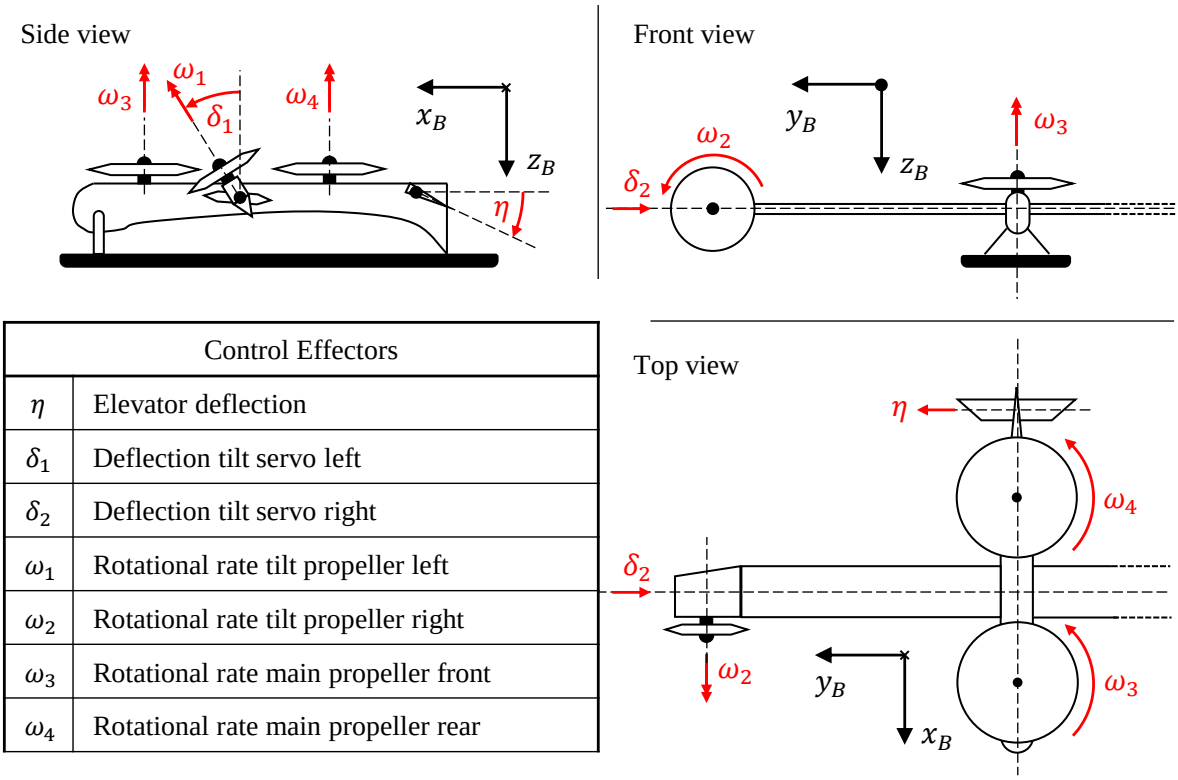
IV. Application on VTOL Transition UAV

This section presents the application of the proposed control strategy on an experimental transition vehicle. The first part introduces the aircraft configuration and the corresponding equations of motion. The second part highlights control specific variables such as pseudo controls and control variables. The subsequent section presents details on the control design and the last part gives a brief overview of hardware related aspects.

A. Aircraft Configuration and and Equations of Motion

The aircraft that is considered within this article is a VTOL transition UAV with movable tilts. A sketch of the vehicle and the available control effectors can be found in Fig. 4. The vehicle comprises four propellers, whereas two of them - named *tilt propellers* - are attached to two moving surfaces, the *tilt servos*. The rotational rates of the propellers are seen as control input to the system, where ω_1 denotes the rotational rate of the left propeller (view towards front of the vehicle) and ω_2 the rotational rate of the right tilt propeller. The tilt angles are accordingly denoted with δ_1 and δ_2. Pointing upwards, with tilt propellers producing force in negative body z-direction (z_B), is defined as zero angle $\delta_{1,2} = 0$, whereas pointing forward is reflected by $\delta_{1,2} = 90°$. In the latter case, the tilt propellers produce forces in positive body x-direction (x_B). The two remaining propellers, denoted *main propellers*, are body fixed and produce force in negative z_B-direction (drag and side forces neglected). The front propeller rotational rate is denoted with ω_3 and the rear propeller rotational rate with ω_4. The rotational directions are not highlighted explicitly, but it shall be mentioned that each of the pairs rotate in opposite directions and that the motors do not support reverse thrust. The last control effector is given by the elevator, whose deflection is denoted as η.

For low speeds and hover, the elevator has no effect and is not used for control. The lift force is solely produced by the four propellers. Forward and backward acceleration is achieved by symmetric deflection of the tilt servos, a pitch maneuver is not planned and only commanded if required. Lateral accelerations are obtained by a respective roll maneuver, as no effector can produce a control force in y_B-direction. Roll and pitch are achieved by differential thrust of the two propeller pairs: differential tilt propeller rate for roll and differential main propeller rate for pitch accelerations. Yaw acceleration is obtained by an asymmetric deflection of the tilt angles.



Control Effectors	
η	Elevator deflection
δ_1	Deflection tilt servo left
δ_2	Deflection tilt servo right
ω_1	Rotational rate tilt propeller left
ω_2	Rotational rate tilt propeller right
ω_3	Rotational rate main propeller front
ω_4	Rotational rate main propeller rear

Fig. 4 Vehicle sketch and control effectors.

As the considered aircraft is a transition vehicle and therefore able to fly at higher aerodynamic speeds, the control authority and control effectiveness of the different effectors change over the flight envelope. For high speed flight, the elevator dominates in producing pitch moments. Furthermore, the required lift will be produced by the wings, such that the main propellers can be disabled in order to reduce power consumption. The tilts point forward into flight direction, which changes the generation of roll and yaw moments compared to the hover case. In wingborne flight, differential tilt propeller rates produce a yaw acceleration, whereas an asymmetric tilt servo deflection is associated with producing roll moments. Aim of the control design will be to use the previous introduced control strategy for controlling this vehicle in a "unified" way, without the necessity of predefined (transition) maneuvers.

In the following, the equations of motion of the presented vehicle are derived. There are three main assumptions made for the further derivations:

First, we assume that the vehicle is always controlled within an adequate Euler angle range, thus the pitch angle Θ never reaches $\pm 90°$. This ensures a singularity free derivation of the equations of motion.

Second, we assume a non rotating earth and therefore earth's rotation rate follows as $\boldsymbol{\omega}^{IE} = \mathbf{0}$. The superscript I stands for the earth centered inertial frame and E for the earth centered earth fixed frame.

Third, earth is assumed to be flat, leading to a transport rate $\boldsymbol{\omega}^{EO} = \mathbf{0}$, with O denoting the NED frame (North East Down). These assumptions also imply that the E-frame can be treated as inertial frame.

An important frame that needs to be introduced before specifying the equations of motion is given by the C-frame (*Control* frame). It is obtained by rotating the NED-frame around its z_O-axis with the azimuth angle Ψ. The x_C-axis reflects the vertical projection of the body x_B-axis onto the horizontal plane. The y_C-axis similarly reflects the projection of the y_B-axis onto the horizontal plane. A sketch of the C-frame can be seen in Fig. 5.

The C-frame is required for the specified translational states of the system. We use the velocities with respect to the E-frame, but written in C-frame as our translational states.

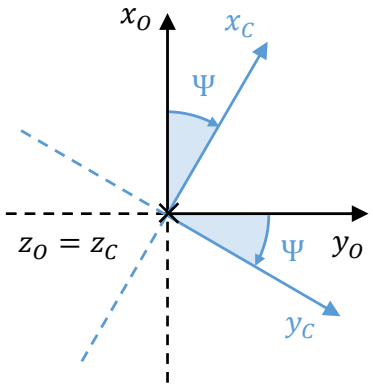


Fig. 5 Control (C)-Frame.

They are denoted as

$$\left(\boldsymbol{V}_K^G\right)_C^E = \begin{bmatrix} \left(u_K^G\right)_C^E \\ \left(v_K^G\right)_C^E \\ \left(w_K^G\right)_C^E \end{bmatrix} \stackrel{!}{=} \begin{bmatrix} u_C \\ v_C \\ w_C \end{bmatrix}, \tag{32}$$

where G specifies the velocity at the center of gravity and K that the kinematic velocity is used. The term on the right denotes the simplified description of the translational states.
The states for the attitude are given by the Euler angles,

$$\boldsymbol{\Phi} = \begin{bmatrix} \Phi \\ \Theta \\ \Psi \end{bmatrix}, \tag{33}$$

and the rotational states by the body rotational rates written in the body B-frame, following

$$\left(\boldsymbol{\omega}^{OB}\right)_B = \begin{bmatrix} \left(\omega_x^{OB}\right)_B \\ \left(\omega_y^{OB}\right)_B \\ \left(\omega_z^{OB}\right)_B \end{bmatrix} \stackrel{!}{=} \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \tag{34}$$

The overall states then can be concatenated as

$$\boldsymbol{x} = \begin{bmatrix} u_C & v_C & w_C & \Phi & \Theta & \Psi & p & q & r \end{bmatrix}^T. \tag{35}$$

The position states are not taken into account, as their effect on the dynamics is neglected. As already mentioned before, the control effectors or control inputs to the system are given by elevator deflection η, tilt deflections of left and right tilt servos δ_1 and δ_2, rotational rates of left and right tilt propeller ω_1 and ω_2, and rotational rates of front and rear main propeller ω_3 and ω_4. The input vector thus follows as

$$\boldsymbol{u} = \begin{bmatrix} \eta & \delta_1 & \delta_2 & \omega_1 & \omega_2 & \omega_3 & \omega_4 \end{bmatrix}^T. \tag{36}$$

As the states and the inputs are specified, we can state the equations of motion, starting with the attitude. Based on [5], these equations of motion are given by the *strapdown* equation following

$$\begin{bmatrix} \dot{\Phi} \\ \dot{\Theta} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\Phi \cdot \tan\Theta & \cos\Phi \cdot \tan\Theta \\ 0 & \cos\Phi & -\sin\Phi \\ 0 & \sin\Phi/\cos\Theta & \cos\Phi/\cos\Theta \end{bmatrix} \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix}. \tag{37}$$

Furthermore, based on [5] and adapted, the translational equations of motion with respect to the center of gravity G and written in C-frame follow as

$$\left(\dot{\boldsymbol{V}}_K^G\right)_C^{EC} = \frac{1}{m} \cdot \left[\left(\boldsymbol{F}_P^G\right)_C + \left(\boldsymbol{F}_A^G\right)_C + \left(\boldsymbol{F}_G^G\right)_C\right] - \left(\boldsymbol{\omega}^{OC}\right)_C \times \left(\boldsymbol{V}_K^G\right)_C^E. \tag{38}$$

In this equation, m denotes the mass of the vehicle, $\left(\boldsymbol{F}_P^G\right)_C$ the propulsion forces written in C-frame, $\left(\boldsymbol{F}_A^G\right)_C$ the aerodynamic forces written in C-frame and $\left(\boldsymbol{F}_G^G\right)_C$ the gravity force written in C-frame. The rotational rate between O-frame and C-frame follows as $\left(\boldsymbol{\omega}^{OC}\right)_C = \begin{bmatrix} 0 & 0 & \dot{\Psi} \end{bmatrix}^T$. Using the substitution for the simplified load factor $\boldsymbol{n}_C = 1/(g \cdot m) \cdot \left[\left(\boldsymbol{F}_P^G\right)_C + \left(\boldsymbol{F}_A^G\right)_C\right]$, the translational state representation and the equation for $\dot{\Psi}$ (given in Eq. (37)), the translational equations of motion can be simplified to

$$\dot{\boldsymbol{V}}_C = \begin{bmatrix} \dot{u}_C \\ \dot{v}_C \\ \dot{w}_C \end{bmatrix} = g \cdot \boldsymbol{n}_C(\boldsymbol{x}, \boldsymbol{u}) + \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \sin\Phi/\cos\Theta \cdot q + \cos\Phi/\cos\Theta \cdot r \end{bmatrix} \times \begin{bmatrix} u_C \\ v_C \\ w_C \end{bmatrix}. \tag{39}$$

The term $\boldsymbol{n}_C(\boldsymbol{x}, \boldsymbol{u})$ indicates that the load factor is a function of states and control effectors. g represents the gravitational constant. The rotational equations of motion follow from [5] as

$$\dot{\boldsymbol{\omega}}_B^{OB} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \boldsymbol{I}_{BB}^{-1} \cdot \left(\boldsymbol{M}_B(\boldsymbol{x}, \boldsymbol{u}) - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \boldsymbol{I}_{BB} \cdot \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right), \tag{40}$$

where $\boldsymbol{I}_{BB}$ specifies the moment of inertia written in B-frame and $\boldsymbol{M}_B(\boldsymbol{x}, \boldsymbol{u})$ denotes the external moments. As indicated in the brackets, they are dependent on the states and the control inputs.

Summarized, the attitude dynamics is independent of any control inputs, as well as any model uncertainties - there is a pure kinematic relation. The translational derivatives can be controlled by the load factors $\boldsymbol{n}_C(\boldsymbol{x}, \boldsymbol{u})$, which are functions of the states and the control inputs. The rotational derivatives can be specified by the moments $\boldsymbol{M}_B(\boldsymbol{x}, \boldsymbol{u})$, which are functions of the control inputs as well. The next section introduces the control variables, virtual controls and pseudo controls for the presented aircraft configuration control.

B. Control Variables, Virtual Control Inputs and Pseudo Controls

The pilot control variables were chosen to be the velocities in the C-frame, as well as $\dot{\Psi}$:

$$\boldsymbol{y}_p = \begin{bmatrix} u_C & v_C & w_C & \dot{\Psi} \end{bmatrix}^T. \tag{41}$$

Note that the vertical velocity reflects the negative height change, $w_C = -\dot{h}$. The control variables were chosen in order to ensure an intuitive behavior for the pilot over the whole flight envelope. Details can be found in [15]. The virtual control inputs follow as the two remaining angular degrees of freedom, the bank angle Φ and the pitch angle Θ, resulting in

$$\boldsymbol{y}_v = \begin{bmatrix} \Phi & \Theta \end{bmatrix}^T. \tag{42}$$

Given Eq. (41) and Eq. (42), the overall control variables follow as

$$\boldsymbol{y} = \begin{bmatrix} u_C & v_C & w_C & \dot{\Psi} & \Phi & \Theta \end{bmatrix}^T. \tag{43}$$

The time derivatives specifying the algebraic level of the pseudo controls can be obtained by using the equations of motion (Eq. (37) - Eq. (40)), which will lead to

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \\ \dot{y}_3 \\ \dot{y}_4 \\ \ddot{y}_5 \\ \ddot{y}_6 \end{bmatrix} = \begin{bmatrix} \dot{u}_C \\ \dot{v}_C \\ \dot{w}_C \\ \ddot{\Psi} \\ \ddot{\Phi} \\ \ddot{\Theta} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_{\dot{V}}(\boldsymbol{n}_C, \boldsymbol{x}) \\ \boldsymbol{f}_{\ddot{\boldsymbol{\Phi}}}(\dot{\boldsymbol{\omega}}_B^{OB}, \boldsymbol{x}) \end{bmatrix}. \tag{44}$$

The nominal pseudo controls are given by the left term. However, it is more practical to use $\boldsymbol{n}_C$ and $\dot{\boldsymbol{\omega}}_B^{OB}$ as pseudo controls instead, as they reflect the physical capabilities of the vehicle in a better way. They are on the same dynamical level and utilizing $\boldsymbol{x}$, they can directly be translated into the nominal pseudo controls. The resulting "mapping" is illustrated in Fig. 6. As can be seen, the reference model and error controller provide nominal pseudo control commands as derived before. The desired pseudo controls are mapped to the "new pseudo controls" $\boldsymbol{n}_C$ and $\dot{\boldsymbol{\omega}}_B^{OB}$, which are then forwarded incrementally to the control allocation.

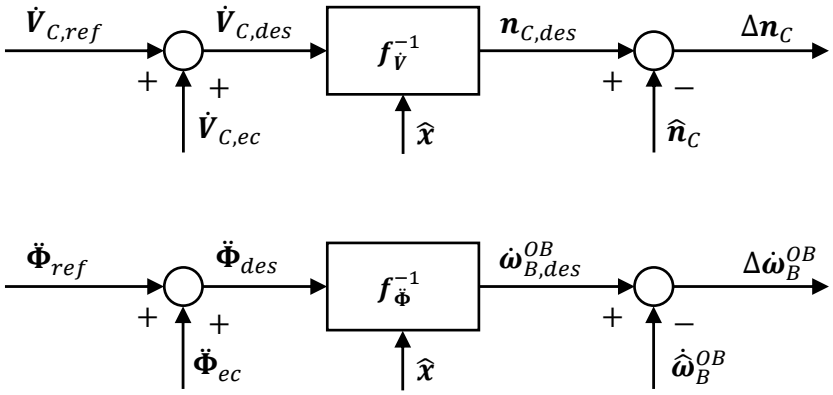


Fig. 6 Pseudo control mapping.

Summarized, the pseudo controls that are allocated to control effectors and virtual controls are given by

$$\boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{n}_C & \dot{\boldsymbol{\omega}}_B^{OB} \end{bmatrix}^T = \begin{bmatrix} n_{x,C} & n_{y,C} & n_{z,C} & \dot{p} & \dot{q} & \dot{r} \end{bmatrix}^T. \tag{45}$$

The next section presents the control design based on the structure presented in the previous chapters.

C. Control Design

This section shall highlight the resulting structures for the different submodules of the controller structure. As aforementioned, a detailed explanation on the functionality of the *Reference Model* can be found in [15]. The important facts are that the reference model takes the pilot commands $u_{C,cmd}$, $v_{C,cmd}$, $w_{C,cmd}$ and Ψ_{cmd}, and the virtual control input commands Φ_{cmd} and Θ_{cmd} as main input. It provides trajectories, or external states, $u_{C,ref}$, $v_{C,ref}$, $w_{C,ref}$, Φ_{ref}, $\dot{\Phi}_{ref}$, Θ_{ref}, $\dot{\Theta}_{ref}$ and $\dot{\Psi}_{ref}$, and nominal pseudo controls $\dot{\boldsymbol{V}}_{C,ref}$ and $\ddot{\boldsymbol{\Phi}}_{ref}$ - compare Fig. 6. The reference model furthermore provides $\boldsymbol{n}_{C,ref}$ and $\dot{\boldsymbol{\omega}}_{B,ref}^{OB}$, which equivalently represents $\dot{\boldsymbol{V}}_{C,ref}$ and $\ddot{\boldsymbol{\Phi}}_{ref}$.

The *Estimation* submodule is used for two main tasks. As there are no actuator and control effector measurements in the considered experimental aircraft, the current values have to be estimated. This is done based on control effector models, which provide the control input estimate $\hat{\boldsymbol{u}}$ based on the command $\boldsymbol{u}_{cmd}$. Furthermore, the *Estimation* module takes inputs from an already existing INS solution, consisting of Kalman filtered specific forces, angular rates, velocity and position. From these inputs the respective states and external states can be calculated without any further estimation required. The estimation of the pseudo controls differs. As there are no angular acceleration sensors on the vehicle, they have to be estimated. For this specific controller, a second order high pass filter on the measured angular rates is used to obtain a low pass filtered angular acceleration estimation. This is a similar approach as used in [9], which is based on [2].

Due to high noise and vibration levels, there has to be a trade off between estimation bandwidth and vibration transfer. One important point to mention is that the pseudo control estimate is delayed due to this filter. As presented in [9], further signals (as the estimate of the current control input $\hat{\boldsymbol{u}}$) have to be delayed using the same filter in order to "match the time" - omitting these filters will lead to errors. We will not present these circumstances in any further detail and the additional required filters are not drawn in the presented block diagrams explicitly.

The next considered submodule, the *Error Controller*, provides a nominal pseudo control part as presented in Fig. 6 that follows Eq. 25 and is given by

$$\begin{aligned}\begin{bmatrix} \dot{\boldsymbol{V}}_{C,ec} \\ \ddot{\boldsymbol{\Phi}}_{ec} \end{bmatrix} = &\begin{bmatrix} k_u & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & k_v & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & k_w & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & k_\Phi & k_{\dot{\Phi}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & k_\Theta & k_{\dot{\Theta}} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & k_\Psi \end{bmatrix} \cdot \begin{bmatrix} e_u \\ e_v \\ e_w \\ e_\Phi \\ e_{\dot{\Phi}} \\ e_\Theta \\ e_{\dot{\Theta}} \\ e_\Psi \end{bmatrix} \\ &+ \int_0^t \begin{bmatrix} k_{I,u} & 0 & 0 & 0 & 0 & 0 \\ 0 & k_{I,v} & 0 & 0 & 0 & 0 \\ 0 & 0 & k_{I,w} & 0 & 0 & 0 \\ 0 & 0 & 0 & k_{I,\Phi} & 0 & 0 \\ 0 & 0 & 0 & 0 & k_{I,\Theta} & 0 \\ 0 & 0 & 0 & 0 & 0 & k_{I,\dot{\Psi}} \end{bmatrix} \cdot \begin{bmatrix} e_u(\tau) \\ e_v(\tau) \\ e_w(\tau) \\ e_\Phi(\tau) \\ e_\Theta(\tau) \\ e_{\dot{\Psi}}(\tau) \end{bmatrix} \mathrm{d}\tau.\end{aligned} \tag{46}$$

The actual gains are still subject to optimization, values for the first flight tests were obtained by taking into account controllability of the system and time scale separation and actuator/control effector dynamics.

The *Onboard-Plant-Model* runs a model of the functions given by

$$\begin{bmatrix} \boldsymbol{n}_C \\ \dot{\boldsymbol{\omega}}_B^{OB} \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_{\boldsymbol{n}_C}(\boldsymbol{x}, \boldsymbol{u}) \\ \boldsymbol{f}_{\dot{\boldsymbol{\omega}}}(\boldsymbol{x}, \boldsymbol{u}) \end{bmatrix}, \tag{47}$$

which is based on the equations of motion, the aircraft configuration and aerodynamic and propulsive lookup tables, which were generated offline utilizing aerodynamic calculation methods and blade element theory [19]. As aforementioned, the required Jacobians are obtained using one sided numerical perturbation. Eventually, the desired Jacobian with respect to the control effectors results in

$$\hat{\boldsymbol{B}} = \begin{bmatrix} \partial n_{x,C}/\partial\eta & \partial n_{x,C}/\partial\delta_1 & \partial n_{x,C}/\partial\delta_2 & \partial n_{x,C}/\partial\omega_1 & \partial n_{x,C}/\partial\omega_2 & \partial n_{x,C}/\partial\omega_3 & \partial n_{x,C}/\partial\omega_4 \\ \partial n_{y,C}/\partial\eta & \partial n_{y,C}/\partial\delta_1 & \partial n_{y,C}/\partial\delta_2 & \partial n_{y,C}/\partial\omega_1 & \partial n_{y,C}/\partial\omega_2 & \partial n_{y,C}/\partial\omega_3 & \partial n_{y,C}/\partial\omega_4 \\ \partial n_{z,C}/\partial\eta & \partial n_{z,C}/\partial\delta_1 & \partial n_{z,C}/\partial\delta_2 & \partial n_{z,C}/\partial\omega_1 & \partial n_{z,C}/\partial\omega_2 & \partial n_{z,C}/\partial\omega_3 & \partial n_{z,C}/\partial\omega_4 \\ \partial\dot{p}/\partial\eta & \partial\dot{p}/\partial\delta_1 & \partial\dot{p}/\partial\delta_2 & \partial\dot{p}/\partial\omega_1 & \partial\dot{p}/\partial\omega_2 & \partial\dot{p}/\partial\omega_3 & \partial\dot{p}/\partial\omega_4 \\ \partial\dot{q}/\partial\eta & \partial\dot{q}/\partial\delta_1 & \partial\dot{q}/\partial\delta_2 & \partial\dot{q}/\partial\omega_1 & \partial\dot{q}/\partial\omega_2 & \partial\dot{q}/\partial\omega_3 & \partial\dot{q}/\partial\omega_4 \\ \partial\dot{r}/\partial\eta & \partial\dot{r}/\partial\delta_1 & \partial\dot{r}/\partial\delta_2 & \partial\dot{r}/\partial\omega_1 & \partial\dot{r}/\partial\omega_2 & \partial\dot{r}/\partial\omega_3 & \partial\dot{r}/\partial\omega_4 \end{bmatrix}, \tag{48}$$

whereas the Jacobian with respect to the virtual control inputs follows as

$$\hat{\boldsymbol{A}}_v = \begin{bmatrix} \partial n_{x,C}/\partial\Phi & \partial n_{x,C}/\partial\Theta \\ \partial n_{y,C}/\partial\Phi & \partial n_{y,C}/\partial\Theta \\ \partial n_{z,C}/\partial\Phi & \partial n_{z,C}/\partial\Theta \\ \partial\dot{p}/\partial\Phi & \partial\dot{p}/\partial\Theta \\ \partial\dot{q}/\partial\Phi & \partial\dot{q}/\partial\Theta \\ \partial\dot{r}/\partial\Phi & \partial\dot{r}/\partial\Theta \end{bmatrix}. \tag{49}$$

The performed calculations use the modeled functions given by Eq. (47) and their application within the functions given by Eq. (26) and Eq. (27). Without going into further details, the numerical perturbation increments are chosen to be around $1/100$ of the respective ranges.

The "translational" pseudo controls are reflected by the load factor in the C-frame. A more natural way to see the available control load factor is of course given by considering the B-frame: In hover, the vehicle can actively produce forces in x_B and z_B-direction, it cannot produce any lateral force. The same holds for wingborne flight when the side force of aerodynamics is neglected (since it is not used as *direct* control force). For hover and low speed flight, the load factors are dominated by the propulsive forces of the propellers. Their magnitude, as well as their direction within the plane spanned by x_B and z_B can be changed by only utilizing the control effectors $\boldsymbol{u}$. However, this is restricted to a two dimensional subspace of the three dimensional space and in order to access forces in all directions, an additional rotation of the body of the vehicle has to be utilized.

If the load factors are written in C-frame, still only their magnitude and direction in the x_B/z_B-plane can be changed by $\boldsymbol{u}$, but the direction in the three dimensional space can be changed using the bank angle Φ and the pitch angle Θ. In other words, a lateral acceleration $n_{y,C}$ can be "allocated" into a bank angle Φ.

The aerodynamic effects of the vehicle increase for higher velocities, until they eventually dominate the main lift force generation. Main effects on the aerodynamic forces and moments are given by the angle of attack α, which can be simplified as $\alpha = \arctan(w_B/u_B)$. The body velocities can be calculated from the velocities given in C-frame by using rotational matrices, following

$$\begin{bmatrix} u_B \\ v_B \\ w_B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\Phi & \sin\Phi \\ 0 & -\sin\Phi & \cos\Phi \end{bmatrix} \cdot \begin{bmatrix} \cos\Theta & 0 & -\sin\Theta \\ 0 & 1 & 0 \\ \sin\Theta & 0 & \cos\Theta \end{bmatrix} \cdot \begin{bmatrix} u_C \\ v_C \\ w_C \end{bmatrix}. \tag{50}$$

The translational states that are used in the OBPM are given by the velocities in the C-frame. A perturbation of Φ and Θ therefore affects the body velocities and the aerodynamic angles. In case of a flight with zero bank, a perturbation of Θ directly reflects a perturbation of α (neglecting wind). Using this correlation, the virtual control input Θ can be used in wingborne flight for the allocation of the desired vertical load factor - lift. The virtual control inputs therefore not just change the direction of the load factors, but also their magnitude.

As there are six pseudo controls in total, but nine effectors (including virtual control inputs), there are three degrees of freedom left over for the control allocation to take into account constraints. In general, they specify an "intuitive" flight operation:

The first constraint specifies the lift distribution between the main and the tilt propellers in hover flight, as well as a reduction and eventually shutting down of the main propeller usage in faster flight.

The second constraint specifies the desired pitch angle. As the tilts can rotate forward and backward, an acceleration in x_B-direction (also x_C) is not force-coupled to a pitch maneuver. The desired philosophy here is to keep the pitch angle at zero until the aerodynamic effects start to produce enough lift.

The third constraint deals with the usage of the elevator and keeps its deflection at zero for low speeds.

As mentioned before, the OBPM provides the Jacobians $\hat{\boldsymbol{B}}_c$ and $\hat{\boldsymbol{A}}_{c_v}$, as specified in Eq. (29) and Eq. (30). The constraints are described in more detail in [16].

The last submodule *Control Allocation* uses the specified Jacobians and the desired increments in pseudo controls as presented in Fig. 6 and calculates corresponding increments in control effectors and virtual control inputs. The whole control allocation algorithm is quite complex and explained separate in more detail in [16].

The structure and design as presented so far reflects the basic concept of the implemented controller. However, the actual controller is more complex and includes further functionalities, which might be presented in the future. The control algorithms were modeled using MATLAB/Simulink and executable code was generated using the automatic code generation functionality of Simulink. The next section gives a brief introduction to the utilized hardware, before the last section presents flight test results and additional remarks.

D. Avionics and Hardware

The hardware of the presented vehicle comprises commercial of the shelf components for INS solution (EKF) and microcontroller, as well as control effectors and actuators. The controller sampling time is $t_S = 0.005$ s, which represents an update rate of 200 Hz.

An important aspect to mention is the resulting computational load, which is mainly caused by the control allocation and the multiple evaluation of the OBPM functions per time step. The latter has been improved by taking into account

the comparatively slow change of the Jacobian matrices. As their change during a small number of time steps is found to be negligible, the number of columns of the Jacobian that are to be calculated per time step can and have been decreased. Thus, every time step only one instead of the all columns of the Jacobians is updated. In other words, instead of 10 evaluations (1 nominal, 9 perturbed) of the OBPM functions, only 2 evaluations (1 nominal, 1 perturbed) have to be performed each time step.

The implemented controllers were verified in Model-in-the-loop (MIL) and Hardware-in-the-loop (HIL) simulations including high fidelity models of reality to increase confidence before performing actual flight tests. The next section highlights conclusions drawn from first flight tests and a brief presentation of a recent flight.

V. Flight Tests and further Remarks

A. Aeropropulsive Interaction

First forward flight tests with the experimental vehicle have shown massive pitch up moments when increasing forward velocity u_C. Although a pitch up was expected due to the horizontal inflow to the main propellers, the experienced moment was above all expectations. The controller canceled the moment by decreasing the front propeller rotational rate and increasing the rear propeller rotational rate. However, at around 12 to 15 m/s, an oscillation in the pitch channel built up, which was (most likely depending on wind) either slighly diverging, persistent but stable or converging within three to four periods. As the oscillation grew stronger for higher forward velocities, the vehicle could not fully transition to wingborne flight - at this point, wingborne flight defined a flight state at around 20 m/s with main propellers at idle. As we saw from the flight logs, the main propellers were highly involved in the oscillation, which triggered further investigation. Apparently, the main propeller commands were reasonable - asymmetric increase/decrease for moment generation and symmetric for vertical load factor. However, there was a non-desired moment building up in the same oscillatory behavior. We concluded that the estimations of the effectiveness of the main propellers were significantly wrong for higher speeds. Therefore, a symmetric increase/decrease of the main propeller rates causes pitch moments. There were two main countermeasures to this behavior so far:

First, we used the elevator and changed the default deflection to the positive saturation. For increasing speed, the elevator is accordingly producing a natural pitch down, which counteracts the pitch up by the main propellers. This measure only helps in the acceleration phase, but there is no impact on the oscillation, as this is a closed loop behavior of main propellers and the pitch channel.

The second countermeasure relates to the undesired pitch moment itself. The forces and moments generated by the propellers were calculated using blade element theory (as presented in [19] for instance). It does not take into account the phase delayed inclination of the tip path plane due to the propellers rotation but the resulting moment from an inflow from the front is a pure roll moment. As the propeller blades are elastic, this elasticity can be seen as a hinge, which leads to a delayed response in the tip path plane inclination. However, we were facing two problems: First, we didn't know proper propeller parameters like stiffness etc., and second the mounting of the propellers itself allowed certain bending - this might change the resulting stiffness. However, we assumed that the undesired moment is related to this flapping moment and thus can be modeled using the calculated roll moment. Consequently, an optimization based on the measured flight data was performed. For each propeller we fitted phase delay of the tip path plane inclination and a gain on the roll moment magnitude. Using this fit, the undesired moment could be approximated adequately using only the roll moment calculated from the blade element theory. Including these results in the OBPM stabilized the transition behavior and we were able to reach wingborne flight. It has to be mentioned that there are still undesired transients during transition, which are subject to future work. The optimization reflects an interim-solution for the time being.

B. Flight Test Results

The performed flight test included a low speed approach phase, in which the pilot flew the vehicle to a specified height and distance. The approach was followed by a short hover and alignment phase in order to adjust the vehicles position and heading for the subsequent transition. The next flight phase was a full wingborne circling above the operator position, followed by back transition, approach to land and landing maneuver. The tacking logs can be found in Fig. 7, which shows *commanded*, *reference* and *estimated* signals for the velocities in the C-frame, the Euler angles Φ and Θ, as well as Ψ. The subsequent Fig. 8 shows the corresponding control effector commands. The red area (*reset*) represents the time in which the controller was inactive.

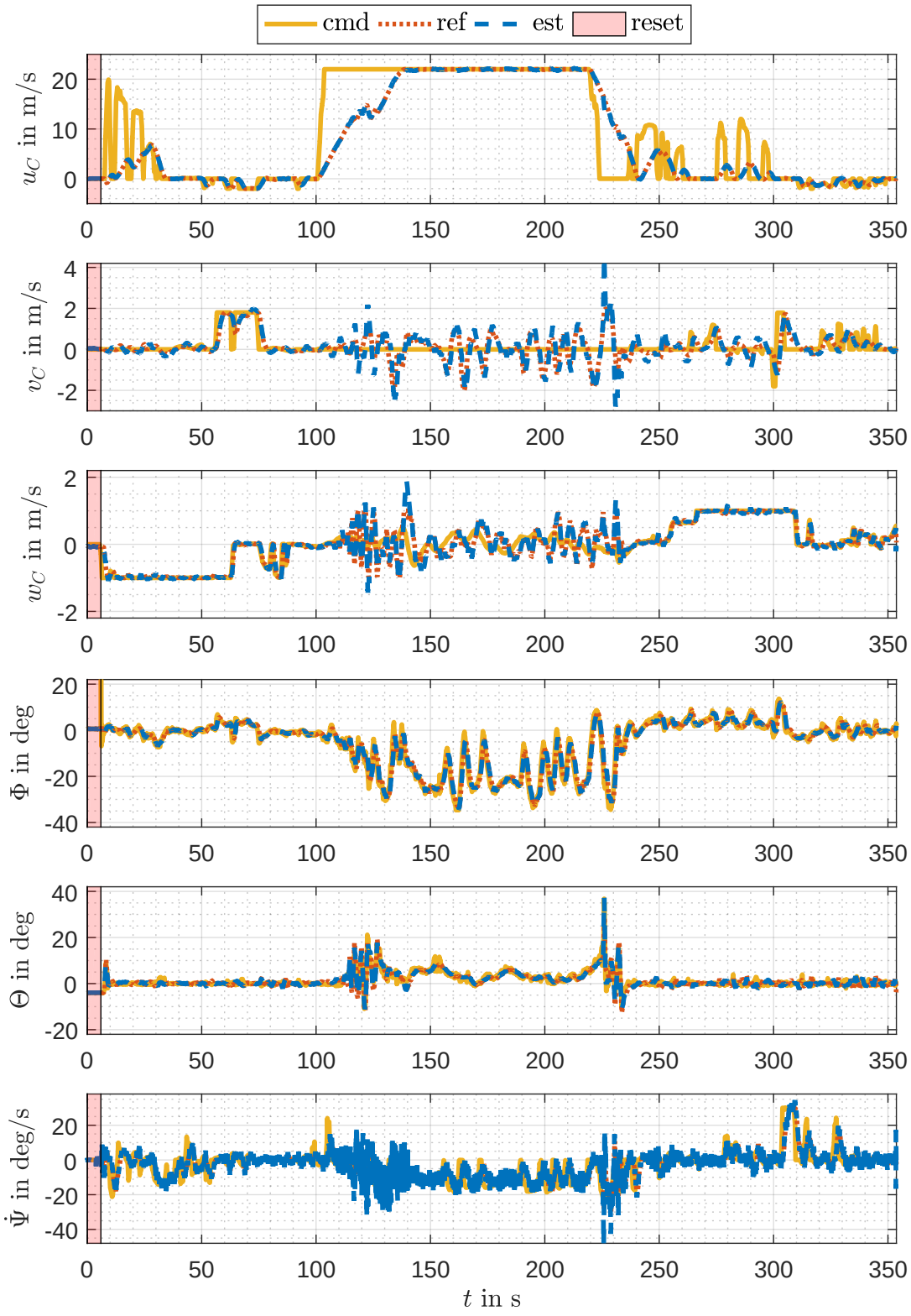


Fig. 7 Tracking of velocity and angles in flight test.

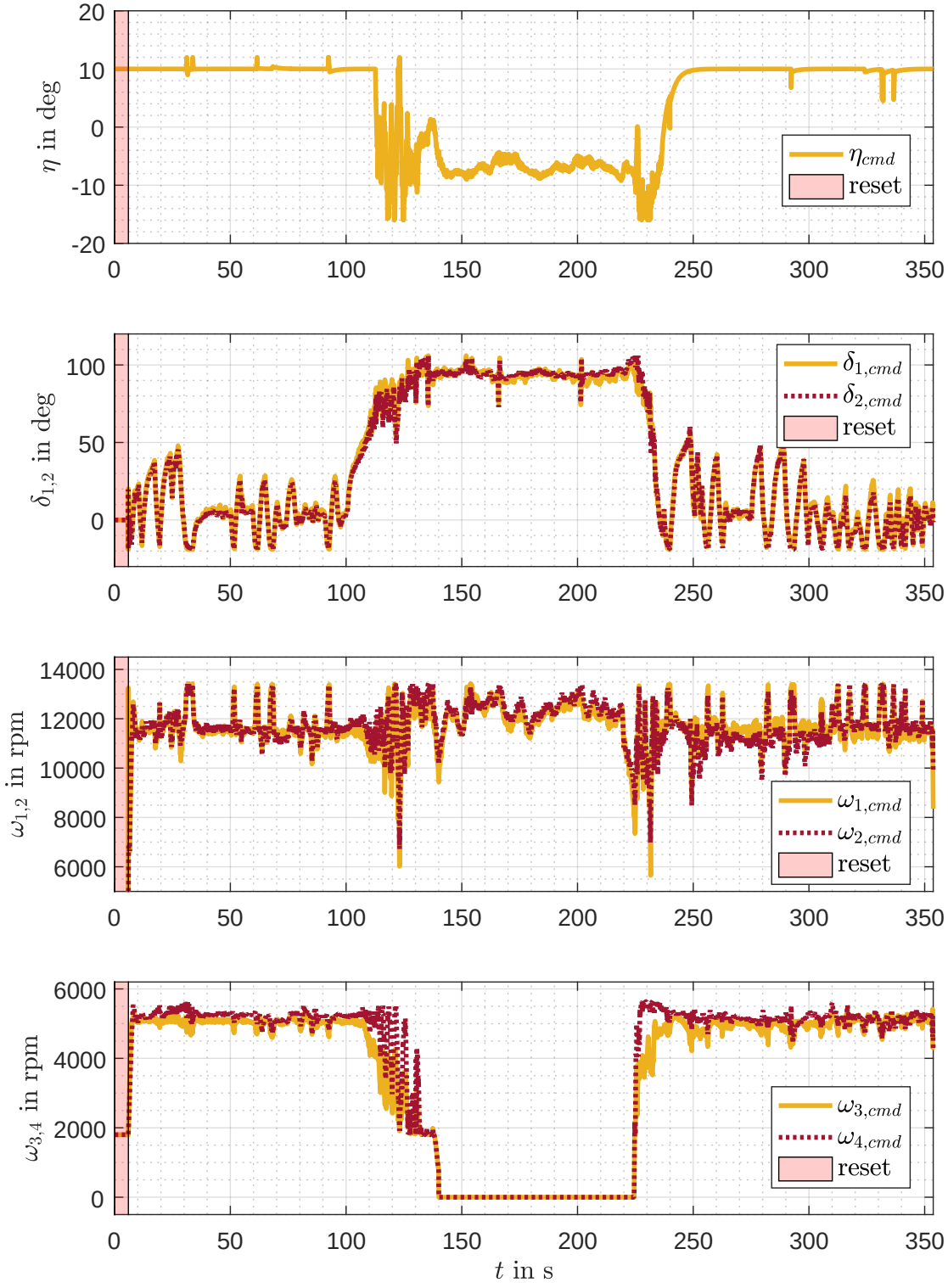


Fig. 8 Control effector commands in flight test.

In general can be seen that the velocity tracking in low speed (0 to 100 s, 250 to 350 s) is more accurate in comparison to wingborne flight. This is affiliated to the higher impact of plant dynamics in wingborne flight and the initial conservative design of the controller parameters. In the region of transition and back-transition (100 to 140 s and 220 to 240 s) higher performance loss can be observed - improvement is subject to future work. From approximately 150 to 220 s, the vehicle flies wingborne, circling counter-clockwise (top view). During wingborne flight, the main propellers shut down and the tilt propellers point forward in body x_B-direction. Furthermore, the elevator is used for pitch control during this period of time, whereas it is kept at a constant deflection of 10° in low speed.

The vehicle is able to track the commanded velocity of 22 m/s without major errors. The lateral velocity error in wingborne flight stays within a bound of 2 m/s, which reflects an angle of sideslip $|\beta| \leq 5°$. The error in vertical velocity stays within a bound of 1 m/s during wingborne flight.

VI. Conclusion

A general unified control strategy for VTOL transition vehicles has been proposed. The structure includes estimation, reference model, error controller, onboard-plant-model and control allocation as modular subparts, enabling an easy adaption to different configurations. Further key aspects comprise the combined allocation of inner and outer loop, and the internal feedback as virtual control input, as well as a continuous controller operation without switching logics and required predefined maneuvers. The control strategy was applied for the control of an experimental transition vehicle and verified in real flights as proof of concept. First flight test showed high aeropropulsive interaction at around 12 to 15 m/s, visible in a closed loop pitch oscillation. A first successful mitigation attempt was based on data based optimization, which enabled a safe transition in subsequent flights. Flight test results of a full mission flight including hover phase, transition, wingborne flight, back-transition and landing have been presented, including plots of command tracking and control effector commands. Future work will focus on improving the transition behavior and increasing performance, as well as the overall bandwidth.

References

[1] Smith, P., "A simplified approach to nonlinear dynamic inversion based flight control," *23rd Atmospheric Flight Mechanics Conference*, 1998, p. 4461.

[2] Bacon, B., and Ostroff, A., "Reconfigurable flight control using nonlinear dynamic inversion with a special accelerometer implementation," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2000, p. 4565.

[3] Walker, G., and Allen, D., "X-35B STOVL Flight Control Law Design and Flying Qualities," *2002 Biennial International Powered Lift Conference and Exhibit*, American Institute of Aeronautics and Astronautics, Reston, Virigina, 2002. doi: 10.2514/6.2002-6018.

[4] Sieberling, S., Chu, Q. P., and Mulder, J. A., "Robust Flight Control Using Incremental Nonlinear Dynamic Inversion and Angular Acceleration Prediction," *Journal of Guidance, Control, and Dynamics*, Vol. 33, No. 6, 2010, pp. 1732–1742. doi:10.2514/1.49978.

[5] Holzapfel, F., *Nichtlineare adaptive Regelung eines unbemannten Fluggerätes: Zugl.: München, Techn. Univ., Diss., 2004*, 1st ed., Luftfahrt, Verl. Dr. Hut, München, 2004.

[6] Akkinapalli, V. S., and Holzapfel, F., "Incremental Dynamic Inversion based Velocity Tracking Controller for a Multicopter System," *2018 AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2018. doi:10.2514/6.2018-1345.

[7] Smeur, E., Croon, G. d., and Chu, Q., "Cascaded incremental nonlinear dynamic inversion for MAV disturbance rejection," *Control Engineering Practice*, Vol. 73, 2018, pp. 79–90. doi:10.1016/j.conengprac.2018.01.003.

[8] Wang, X., van Kampen, E.-J., Chu, Q. P., and Lu, P., "Stability Analysis for Incremental Nonlinear Dynamic Inversion Control," *2018 AIAA Guidance, Navigation, and Control Conference*, 2018, p. 1115.

[9] Smeur, E. J. J., Chu, Q., and Croon, G. C. H. E. d., "Adaptive Incremental Nonlinear Dynamic Inversion for Attitude Control of Micro Air Vehicles," *Journal of Guidance, Control, and Dynamics*, Vol. 39, No. 3, 2016, pp. 450–461. doi:10.2514/1.G001490.

[10] Wang, X., van Kampen, E.-J., Breuker, R. d., and Chu, Q. P., "Flexible Aircraft Gust Load Alleviation with Incremental Nonlinear Dynamic Inversion," *2018 AIAA Atmospheric Flight Mechanics Conference*, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2018. doi:10.2514/6.2018-0774.

[11] van 't Veld, R., van Kampen, E.-J., and Chu, Q. P., "Stability and Robustness Analysis and Improvements for Incremental Nonlinear Dynamic Inversion Control," *2018 AIAA Guidance, Navigation, and Control Conference*, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2018. doi:10.2514/6.2018-1127.

[12] Di Francesco, G., Mattei, M., and D'Amato, E., "Incremental nonlinear dynamic inversion and control allocation for a tilt rotor UAV," *AIAA Guidance, Navigation, and Control Conference*, 2014, p. 963.

[13] Di Francesco, G., D'Amato, E., and Mattei, M., "INDI Control with Direct Lift for a Tilt Rotor UAV," *IFAC-PapersOnLine*, Vol. 48, No. 9, 2015, pp. 156–161. doi:10.1016/j.ifacol.2015.08.076.

[14] Di Francesco, G., and Mattei, M., "Modeling and Incremental Nonlinear Dynamic Inversion Control of a Novel Unmanned Tiltrotor," *Journal of Aircraft*, Vol. 53, No. 1, 2016, pp. 73–86. doi:10.2514/1.C033183.

[15] Bhardwaj, P., Raab, S., Zhang, J., and Holzapfel, F., "Integrated Reference Model for a Tilt-rotor Vertical Take-off and Landing Transition UAV," *2018 AIAA Aviation*, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2018 (submitted for publication).

[16] Zhang, J., Bhardwaj, P., Raab, S., and Holzapfel, F., "Control Allocation Framework for a Tilt-rotor Vertical Take-off and Landing Transition Aircraft Configuration," *2018 AIAA Aviation*, American Institute of Aeronautics and Astronautics, Reston, Virginia, 2018 (submitted for publication).

[17] Chen, H. B., and Zhang, S. G., "Robust dynamic inversion flight control law design," *Systems and Control in Aerospace and Astronautics, 2008. ISSCAA 2008. 2nd International Symposium on*, 2008, pp. 1–6.

[18] Isidori, A., *Nonlinear Control Systems*, third edition ed., Communications and Control Engineering, Springer, London, 1995. doi:10.1007/978-1-84628-615-5, URL http://dx.doi.org/10.1007/978-1-84628-615-5.

[19] Johnson, W., *Helicopter Theory*, Dover Books on Aeronautical Engineering, Dover Publications, Newburyport, 2012. URL http://gbv.eblib.com/patron/FullRecord.aspx?p=1894902.