UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Lucas Cunha Duarte

Unified Control Law Strategy for a Piloted Vertical Take-off and Landing Transitioning Aircraft

Uberlândia

2022

Lucas Cunha Duarte

Unified Control Law Strategy for a Piloted Vertical Take-off and Landing Transitioning Aircraft

Final thesis for the graduation course of Aeronautical Engineering of the Universidade Federal de Uberlândia for the degree of Bachelor in Aeronautical Engineering.
Supervisor: Prof. Dr. Pedro Augusto Queiroz de Assis

Uberlândia
2022

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

D812 Duarte, Lucas Cunha, 1998-
2022 Unified Control Law Strategy for a Piloted Vertical Take-off and Landing Transitioning Aircraft [recurso eletrônico] / Lucas Cunha Duarte. - 2022.

Orientador: Pedro Augusto Queiroz de Assis.
Trabalho de Conclusão de Curso (graduação) - Universidade Federal de Uberlândia, Graduação em Engenharia Aeronáutica.
Modo de acesso: Internet.
Inclui bibliografia.
Inclui ilustrações.

1. Aeronáutica - Engenharia de sistemas. I. Assis, Pedro Augusto Queiroz de,1989-, (Orient.). II. Universidade Federal de Uberlândia. Graduação em Engenharia Aeronáutica. III. Título.

CDU: 629.7.054

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:
Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074

Lucas Cunha Duarte

Unified Control Law Strategy for a Piloted Vertical Take-off and Landing Transitioning Aircraft

This final thesis has been considered suitable for obtaining the Bachelor's Degree in Aeronautical Engineering and approved in its final form by the Aeronautical Engineering Graduate Course.

Uberlândia, August 10, 2022.

Reviewing Board:

Prof. Dr. Pedro Augusto Queiroz de Assis
Universidade Federal de Uberlândia

Prof. M.Sc. Giuliano Gardolinski Venson
Universidade Federal de Uberlândia

Dr. Felipe Machini Malachias Marques

ACKNOWLEDGEMENTS

Gostaria de agradecer primeiramente a meu querido professor orientador, Pedro Assis, pelas incontáveis revisões dessa tese. Toda sua atenção aos pequenos detalhes e conhecimento foram fundamentais para a qualidade final desse trabalho, e eu sou muito grato por todo tempo que você se dedicou para isso. Também gostaria de exaltar suas aulas de controle linear, as quais tive o prazer de presenciar e que são indubitavelmente uma das bases para este trabalho.

Também gostaria de expressar meus sinceros agradecimentos a meu segundo orientador, Burak. Muito obrigado pela confiança em meu trabalho e todo o suporte durante minha estadia em Munique. Todo o entusiasmo e paixão que você expressava no dia a dia (especialmente nos dias de voo) sempre me motivaram e eu sou extremamente grato pelas oportunidades que você me proporcionou. Gostaria também de exaltar meu brilhante colega de trabalho Ashwath. Nossas incontáveis discussões a respeito de teoria de controle e aeronaves sempre foram muito agradaveis e eu aprendi muito com elas. Obrigado por todo o conhecimento compartilhado e motivação.

Agradeço à coordenação do curso de Engenharia Aeronáutica, em especial ao Giuliano, pelo suporte durante meu estágio e à Leandra que sempre distribuiu todo o carinho possível aos alunos do nosso curso e sempre se mostrou disposta a nos ajudar.

Obrigado à equipe Tucano Aerode-

First, I would like to thank my advisor professor, Pedro Assis, for the countless revisions of this thesis. All your deep knowledge and attention to small details were fundamental to the final quality of this work, and I am sincerely grateful for all the time you dedicated to this. I would also like to acknowledge your lectures on linear control, which I had the pleasure to attend and are undoubtedly one of the cornerstones of this work.

I would also like to express my sincere gratitude to my second advisor, Burak. Thanks for your trust in my work and all the support during my stay in Munich. All the enthusiasm and passion that you expressed on a daily basis (especially on flight-test days) always motivated me and I am extremely grateful for the opportunities you provided me with. I would also like to recognize my brilliant co-worker Ashwath. Our countless discussions about control theory and aircraft were always pleasant and I learned a lot from it. Thank you for all the shared knowledge and motivation.

I would like to thank the coordinators of the Aeronautic Engineering course, especially Giuliano, for the support during my internship and Leandra, who always gave all the love possible to the students of our course and was always willing to help us.

I thank to the Tucano Aerodesign

sign por todas as oportunidades que me foram proporcionadas e pelo papel exemplar no desenvolvimento dos alunos do nosso curso. Desejo todo sucesso possível ao futuro da equipe e a todos os membros.

Gostaria de expressar minha gratidão aos amigos e colegas que conheci em Munique e que fizeram da minha experiencia aqui inesquecível. Meus sinceros cumprimentos a Robin, Tim, Mani, Krishna, Shashi, Bastie, Rene, Marwan, Greeshma e Pranav. Obrigado por todo o suporte e bons momentos que compartilhamos.

Além disso, agradeço a todos os meus maravilhosos amigos do Brasil que sempre estiveram em contato: Alfredo, Ísis, Daniel, Torezan, Alves, Freitas, Iago, Phillemon, Lutffalla, Filipe, Amanda, Josh, Honorato, Dimas, Yasmin, Guilherme, Lucas, Emanuelly e Ana. Agradeço especialmente à Natalia que sempre esteve comigo nos momentos difíceis. Sou muito grato por ter você durante essa parte da minha vida e desejo apenas o melhor em seu futuro.

Agradeço a toda minha família, em especial às minhas tias Augusta, Rita, Margarida e Maria Helena, aos meus irmãos Samira e André, a minha pequenina sobrinha Ana Júlia, por trazer tanta alegria à nossa casa. Por fim, sou eternamente grato a meus pais Vanda e Carlos por todas as oportunidades e privilégios que me foram proporcionados e por todo amor e carinho recebido durante toda minha vida.

team for all the opportunities it has provided me, as well as the exemplary role in the education of the students of the Aeronautic Engineering course. I wish all possible success to the future of the team and all members.

I would also like to express my gratitude to the friends and colleagues I met in Munich, who made my experience here unforgettable. Many thanks to Robin, Tim, Mani, Krishna, Shashi, Bastie, Rene, Marwan, Greeshma and Pranav. Thanks for all the support, beers and good times we had together.

In addition, I thank all my lovely friends from Brazil who have always been in touch: Alfredo, Ísis, Daniel, Torezan, Alves, Freitas, Iago, Phillemon, Lutffalla, Filipe, Amanda, Josh, Dimas, Guilherme, Lucas, Emanuelly and Ana. Special thanks to Natalia, who always supported me during tough times. I am deeply grateful for having you during this part of my life and I wish only the best in your future.

My sincere gratitude to my whole family, especially my aunties Augusta, Rita, Margarida and Maria Helena, my brothers Samira and André and my little niece Ana Júlia, for bringing so much joy to our home. Finally, I am eternally grateful to my parents Vanda and Carlos for all the opportunities and privileges they have provided me, as well as the love and care they have shown me throughout my life.

ABSTRACT

The development of VTOL (Vertical Take-Off and Landing) transitioning aircraft has been increasing in recent years. Such vehicles utilize both the ability to fly using lift from a fixed wing and to fly vertically using power-lift actuators, generally in the form of rotors. These vehicles face several challenges from a flight control perspective due to large changes in the system dynamics from vertical take-off to hover and wingborne flight, and back to vertical landing.

This thesis proposes a unified controller strategy to enable such aircraft to transition between those different flight phases without the need of switches, using the INDI (Incremental Non-Linear Dynamic Inversion) control technique. A velocity and attitude controller loops are designed to compute desired linear and angular accelerations, which are then sent to a control allocation framework to compute the actuators commands. The designed controller is validated through simulation of a high-fidelity model of a scaled drone developed at Volocopter GmbH. The results demonstrated that the developed strategy can track the reference commands and transition from hover to wingborne flight and back.

Keywords: Unified control; incremental nonlinear dynamic inversion; control allocation; VTOL.

LIST OF FIGURES

LIST OF ABBREVIATIONS AND ACRONYMS

AAM	Advanced Air Mobility
AMS	Attainable Moments Subset
CFD	Computational Fluid Dynamics
CG	Center of Gravity
GNSS	Global Navigation Satellite System
IMU	Inertial Measurement Unit
INDI	Incremental Non-Linear Dynamic Inversion
MPC	Model Predictive Controller
NED	North-East-Down
PCH	Pseudo Control Hedging
RSPI	Redistributed Scaled Pseudo Inverse
SISO	Single-Input and Single-Output
UAV	Unmanned Aerial Vehicle
VTOL	Vertical Take-Off and Landing

CONTENTS

NOTATION

In this work, scalars are presented using normal weighted characters, e.g. $\lambda \in \mathbb{R}$, vectors and matrices are denoted with bold symbols, e.g. $\boldsymbol{\nu} \in \mathbb{R}^{m \times n}$ and coordinate transformation matrices are given with the following subscript notation:

$$\boldsymbol{R}_{AB}$$

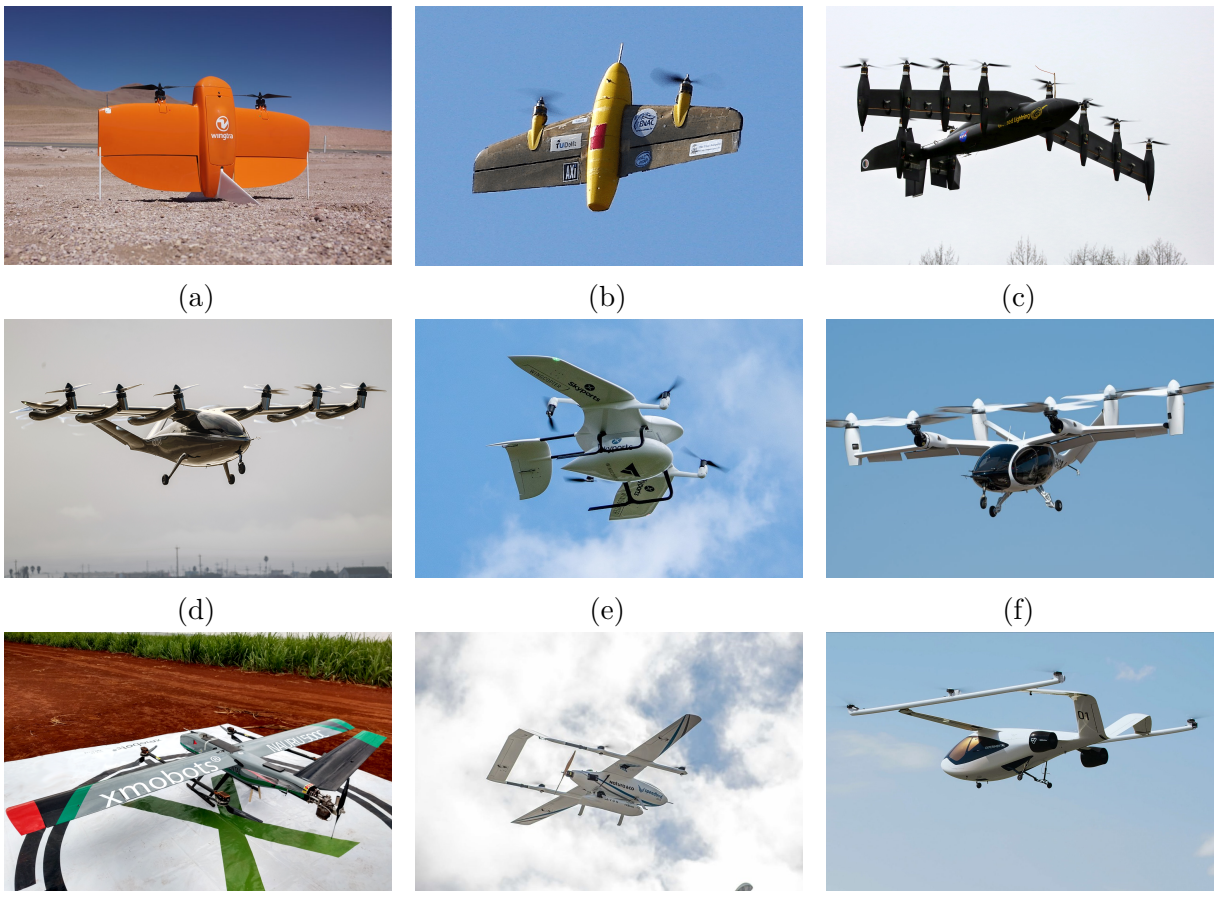
where $\boldsymbol{R}_{AB}$ represents the matrix transformation from frame B to A.

Geometric vectors are denoted with capital subscripts to indicate at which coordinate system the vector is denoted in. As an example, $\boldsymbol{V}_B$ represent a velocity vector denoted in the Body frame (B).

1 INTRODUCTION

A whole new generation of aircraft is expected to operate in the so-called AAM (Advanced Air Mobility), an air transportation ecosystem aimed to transport people and cargo in urban areas (NASA, 2022). For this purpose, VTOL (Vertical Take-Off and Landing) transitioning aircraft are becoming a common solution as this configuration combine the characteristics of multi-rotor and fixed wing aircraft. By using a fixed wing during cruise, those aircraft have a greater range and cruise speed compared to multi-rotor aircraft (STRAUBINGER et al., 2020). Moreover, using propellers facing upwards, they are capable of performing vertical flight, which can be useful to operate in non-aviation areas where conventional wingborne aircraft can't land or take-off. As examples, Figure 1 shows some pictures of real VTOL transitioning aircraft.

Figure 1 – Different VTOL transitioning aircraft configurations. (a) and (b) perform hover flight by tilting its body, in a configuration called tailsitter. (c) utilizes a tilt-wing to align the wing and the rotors attached in the direction of the airflow for wingborne flight. (d), (e) and (f) utilize tilt-rotors to generate thrust in the forward direction for wingborne flight. (g), (h) and (i) use fixed-rotors, where different sets of motors are used for hover and wingborne flight.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Source: Wingtra (a), MAVLab TUDelft (b), NASA (c), Archer (d), Wingcopter (e), Joby Aviation (f), XMobots (g), Speedbird (h), Volocopter GmbH (i)

The transition dynamics from hover to wingborne flight is highly nonlinear, and modeling the aerodynamics at low airspeed, specially in the stall regime, requires additional complexity (YEO; LIU, n.d.). Therefore, the development of a control strategy for the whole flight envelope can be complex and challenging.

Several approaches were proposed to overcome such challenges. For example, Öznalbant and Kavsaoğlu (2018) successfully controlled a tilt-rotor UAV by projecting different controllers for specific flight phases and switching between them based on the aircraft speed. Although it is a straightforward implementation, ensuring the stability between the transition from one controller to another can be a challenging task. Yeo and Liu (n.d.) and Hartmann, Meyer, and Moormann (2017) developed a single controller by using gain scheduling and defined trim operation points based on the aircraft speed. Bauersfeld et al. (2021) used a MPC (Model Predictive Controller) outer loop to define optimal trajectories and transition from hover to wingborne. The problem with those approaches is that they rely on extensive modelling of the system.

This thesis follows the approach of Raab et al. (2018), which uses a unified controller structure with only few model parameters required. This unified structure is composed by an attitude controller, a velocity controller and a control allocation framework to distribute the actuators commands. In this thesis some improvements are proposed in terms of a new control allocation and a new command mapping architecture to address some of the issues observed in their work, such as the performance during transition.

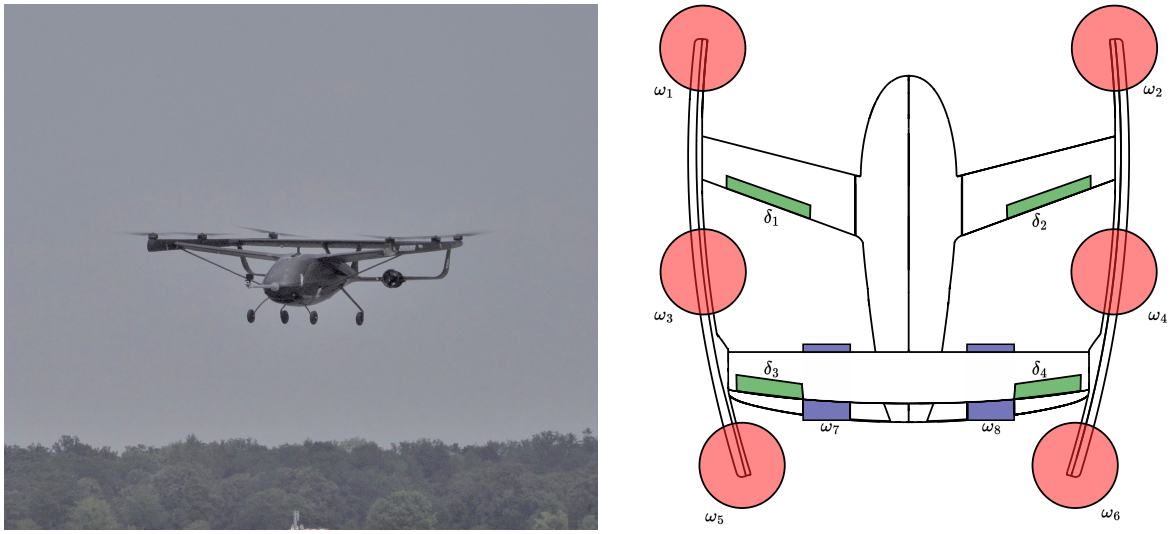
The remainder of this thesis is structured in the following way: Chapter 2 describes the aircraft used herein, the adopted reference frames and the equations of motion. Chapter 3 details the control strategy to perform the transition from hover to wingborne flight. Chapter 4 presents simulations results using a six degree of freedom high fidelity model developed in another project at Volocopter GmbH. Lastly, Chapter 5 presents a brief conclusion and suggestions for future work.

This thesis work was performed at Volocopter GmbH under the supervision of Dr.-Ing. Burak Yüksel.

2 SYSTEM DESCRIPTION

To validate the proposed control strategy, a scaled drone developed at Volocopter GmbH is used, which is presented in Figure 2(a). This aircraft can perform vertical take-off and landing using its lifting rotors (vertical motors) that are facing upwards, and transition to wingborne, where it can use its control surfaces for controlling the aircraft in fixed-wing flight phase. Figure 2(b) shows a sketch of this aircraft, where the details on the actuator location is depicted.

Figure 2 – On the left side (a), a picture showing the scaled prototype used in this thesis. On the right side (b), a sketch of the same aircraft with its actuators highlighted.



(a) (b)

Source: Volocopter GmbH.

Notice from Figure 2, that there are eight motors (six pointing upwards and two forward). Additionally, four control surfaces can be manipulated. The speed of such motors (denoted by $\omega_i,\ i = 1,\ 2,\ \cdots,\ 8$) and the displacement of the control surfaces (represented by $\delta_i,\ i = 1,\ 2,\ 3,\ 4$) compose the real actuators control vector, which is denoted by

$$\boldsymbol{u}_{act} = [\ \overbrace{\omega_1(t) \quad \cdots \quad \omega_6(t)}^{\text{Vertical Motors}} \quad \overbrace{\omega_7(t) \qquad \omega_8(t)}^{\text{Horizontal Motors}} \quad \overbrace{\delta_1(t) \quad \cdots \quad \delta_4(t)}^{\text{Control Surfaces}}\ ]^T$$

The vertical motors are used during the hover phase to lift the aircraft while on the wingborne phase, the fixed wings produce the necessary lift. The horizontal motors are used during both phases to produce forward thrust as well as yawing moments by means of differential thrust. The control surfaces are employed during wingborne flight to produce rolling and pitching moments.

The aircraft is equipped with an IMU (Inertial Measurement Unit) and a GNSS (Global Navigation Satellite System) sensor. Thus, the available measurement (system

outputs) compose the following vector:

$$\boldsymbol{\xi}(t) = [\boldsymbol{\Phi}(t) \quad \boldsymbol{\omega}_B(t) \quad \boldsymbol{f}_B(t) \quad \boldsymbol{V}_O(t)\ ]^T$$

where $\boldsymbol{\Phi}$, $\boldsymbol{\omega}_B$ and $\boldsymbol{f}_B$ are the Euler angles, angular rates and specific forces in body frame, respectively. These outputs are estimated by means of an already existing sensor fusion solution. Moreover, $\boldsymbol{V}_o$ depicts the kinematic velocity vector in the NED frame, which is calculated by using the GNSS informations.

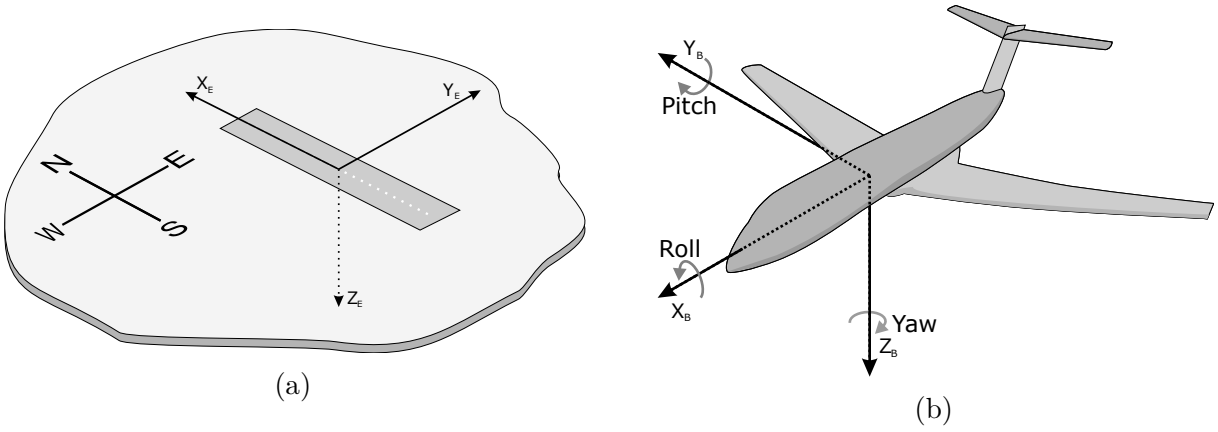
In order to reduce the pilot workload, a proper control strategy must be employed. To design such strategy, firstly, a system model is detailed in the following section.

2.1 SYSTEM MODELLING

To develop the equations of motion of the aircraft, a set of frames is defined. Figure 3(a) shows the earth frame, depicted with the subscript E. The X-axis is chosen to points North while the Y-axis points East and the Z-axis down. Herein, the earth frame is treated as an inertial frame and, therefore, the earth curvature and rotation are neglected due to the flight envelope considered in this context.

Additionally, Figure 3(b) depicts the body frame, denoted with subscript B. Its origin is coincident with the CG (Center of Gravity) of the aircraft. Its X-axis points towards the aircraft nose, the Y-axis points along the right wing and Z points down.

Figure 3 – Inertial frame (a) and Body frame (b).



(a) (b)

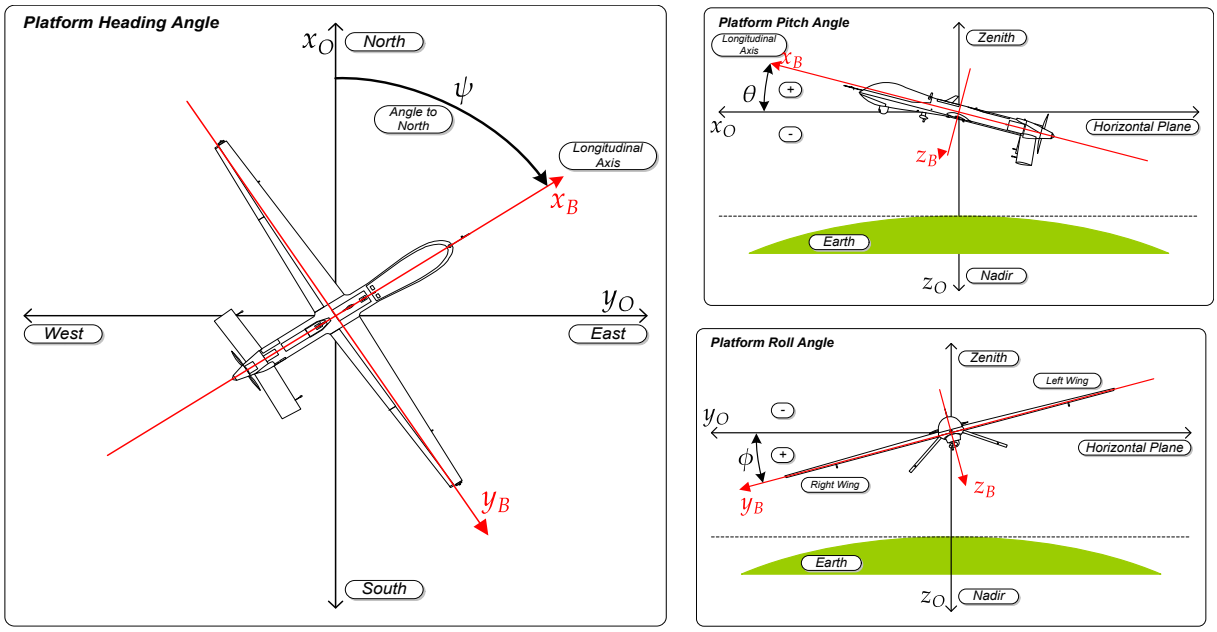
Source: Kriel (2008).

The NED (North-East-Down) frame is depicted in Figure 4 with the subscript O. It has same orientation as the Earth frame but with origin at the vehicle CG. The orientation of the aircraft with respect to the NED frame is given by a series of three sequential rotations from the NED frame, which are represented by the Euler angles ψ (yaw), θ (pitch) and ϕ (roll or bank), respectively. To transform a vector from body to

NED frame, the following transformation can be used:

$$\boldsymbol{R}_{OB} = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \quad (1)$$

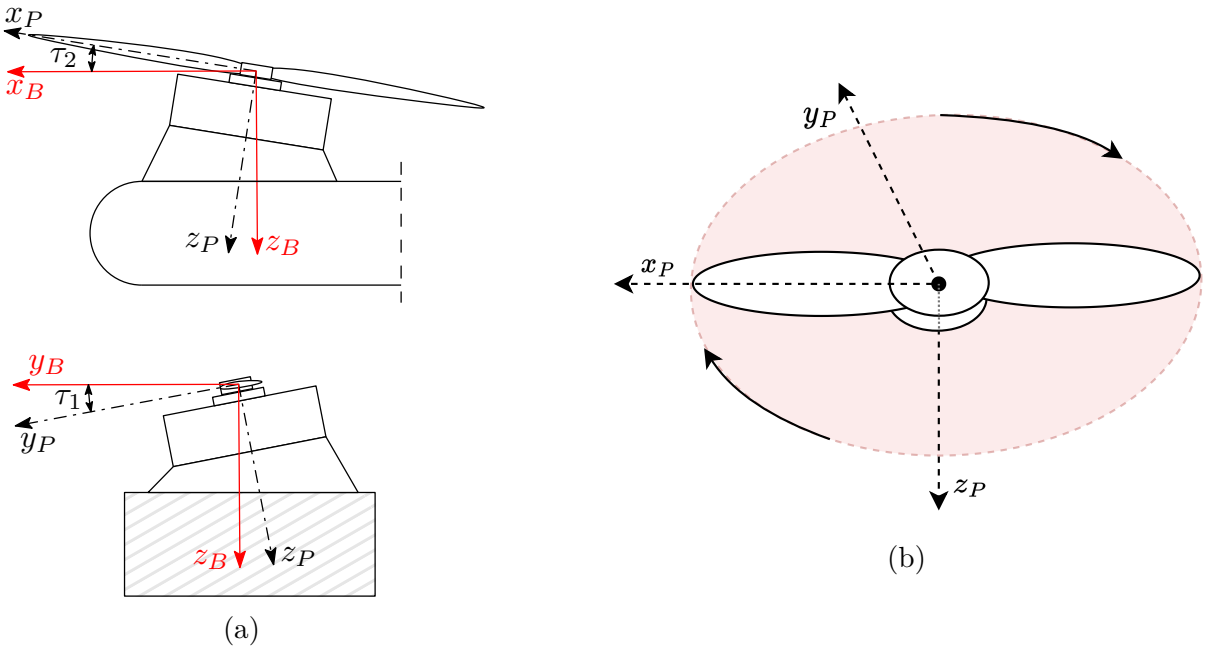
Figure 4 – Illustration of the NED frame and the Euler angles.



Source: Adapted from (MISB, 2014)

The Propeller frame, denoted with subscript P, is a local coordinate system defined for each rotor. Its origin is at the center of the propeller and its Z-axis points perpendicular to the plane in which the propellers spin, the rotor plane, as depicted in Figure 5(b). The angles between the rotor plane and the Body frame are the tilt angles of the motors, depicted as τ_1 and τ_2 and illustrated in Figure 5(a).

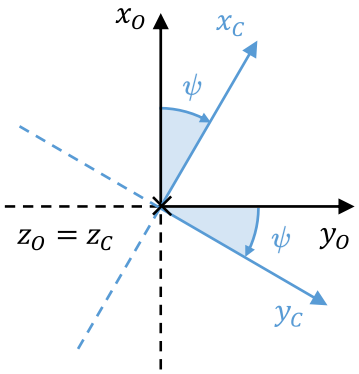
Figure 5 – Illustration of the Propeller frame. The Body frame origin is translated to the center of the propeller for better visualization of the tilt angles. On the right side is depicted the convention for a positive rotation direction.



Source: Author.

The Controller frame, pointed by the subscript C is obtained by rotating the NED frame about the z_O axis by the yaw angle ψ, as depicted in Figure 6. The velocity commands are given using this set of axes, as will be explained in Section 3.4.

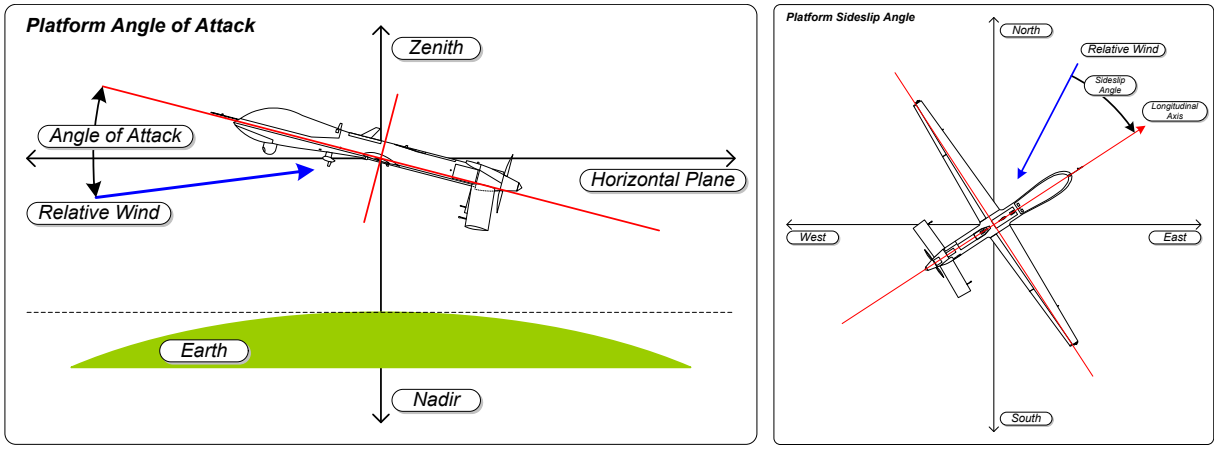
Figure 6 – Illustration of the relation between the NED and the Controller frames.



Source: Raab et al. (2018).

The angle of attack (α) and the sideslip (β) are the angles between the relative wind vector and the Body frame, as shown in Figure 7.

Figure 7 – Illustration of the angle of attack and sideslip angle.



Source: MISB (2014).

The translational equations of motion in the Controller frame are given by (RAAB et al., 2018):

$$\dot{\boldsymbol{V}}_C(\boldsymbol{u}(t), \boldsymbol{x}(t)) = \begin{bmatrix} \dot{u}_C(\boldsymbol{u}(t), \boldsymbol{x}(t)) \\ \dot{v}_C(\boldsymbol{u}(t), \boldsymbol{x}(t)) \\ \dot{w}_C(\boldsymbol{u}(t), \boldsymbol{x}(t)) \end{bmatrix} = \frac{1}{m} \cdot [\boldsymbol{F}_C(\boldsymbol{u}(t), \boldsymbol{x}(t)) - \boldsymbol{\omega}_C(t) \times \boldsymbol{V}_C(t)] \tag{2}$$

where m is the vehicle mass, $\boldsymbol{F}_C \in \mathbb{R}^3$ is the external force vector written in the Controller frame, $\boldsymbol{V}_C \in \mathbb{R}^3$ and $\boldsymbol{\omega}_C \in \mathbb{R}^3$ are the kinematic velocity and angular rate with respect to the NED frame, respectively. Lastly, $\boldsymbol{x}$ represents the system states. Henceforth the time dependency of the variables will be omitted for clarity.

The rotational equations of motion are given by (RAAB et al., 2018):

$$\dot{\boldsymbol{\omega}}_B(\boldsymbol{u}, \boldsymbol{x}) = \begin{bmatrix} \dot{p}(\boldsymbol{u}, \boldsymbol{x}) \\ \dot{q}(\boldsymbol{u}, \boldsymbol{x}) \\ \dot{r}(\boldsymbol{u}, \boldsymbol{x}) \end{bmatrix} = \boldsymbol{I}_{BB}^{-1} \cdot (\boldsymbol{M}_B(\boldsymbol{u}, \boldsymbol{x}) - \boldsymbol{\omega}_B \times \boldsymbol{I}_{BB} \cdot \boldsymbol{\omega}_B) \tag{3}$$

where $\boldsymbol{I}_{BB} \in \mathbb{R}^{3\times3}$ is the aircraft inertia matrix written in body frame, $\boldsymbol{M}_B \in \mathbb{R}^3$ denotes the external moments and $\boldsymbol{\omega}_B \in \mathbb{R}^3$ is the angular rate vector.

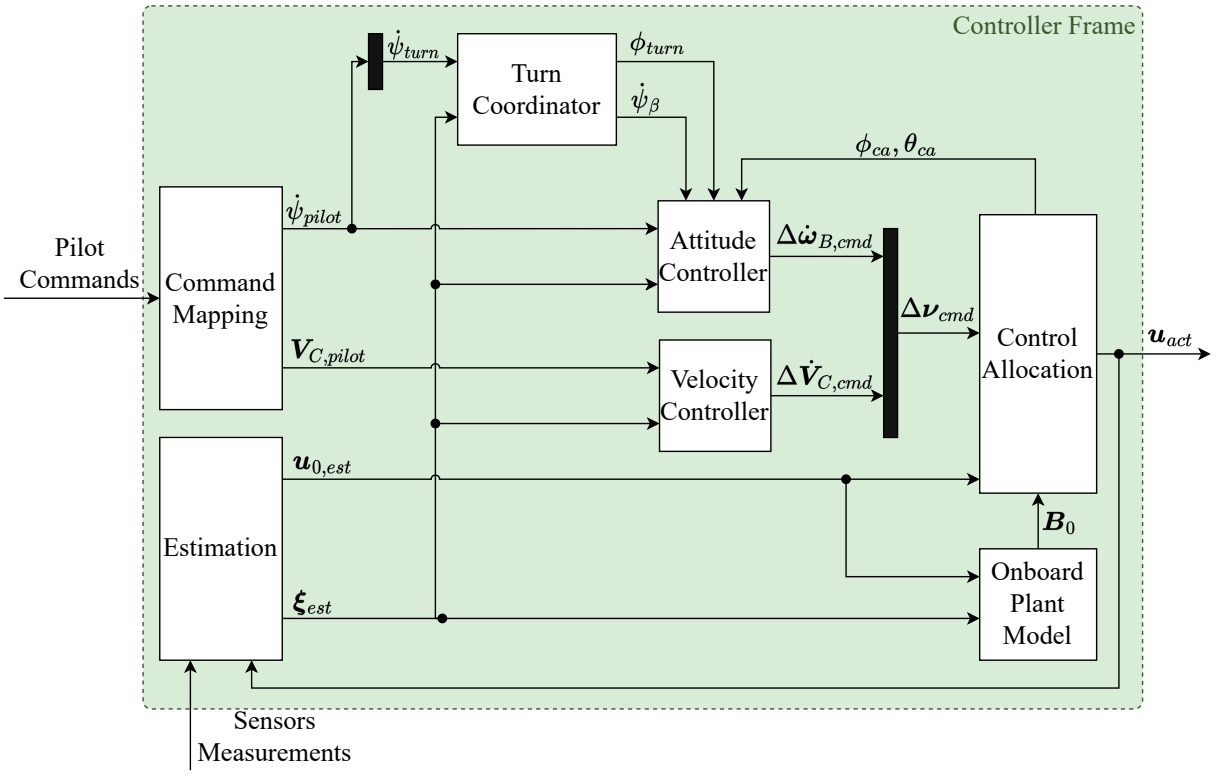
Equations (2) and (3) are used to develop the inversion law for the attitude and velocity controller, as described in Sections 3.3 and 3.4.

3 UNIFIED CONTROL STRATEGY

Figure 8 shows a block diagram of the adopted control strategy. The pilot commands are mapped into velocities commands in the Controller frame and heading rate commands. The attitude and velocity controllers receive those commands and the feedback from the sensors and then calculate desired increment on the angular and linear accelerations, the pseudo controls. These pseudo controls (which are denoted by $\Delta\boldsymbol{\nu} = [\Delta\dot{\boldsymbol{V}}_C^T,\ \Delta\dot{\boldsymbol{\omega}}_B^T]^T$) are transformed into the real control vector by using a control allocation technique. Moreover, it also generates roll and pitch angle commands to the attitude controller.

The estimation block represents all sensor data filtering as well as estimating the state of the actuators ($\boldsymbol{u}_{0,est}$), since no feedback from them is available. The onboard plant model calculates the current effectiveness matrix $\boldsymbol{B}_0$ (details are given in Section 3.2). Lastly, the Turn Coordinator block generates roll angle and heading rate commands for coordinated turns during high-speed flight.

Figure 8 – Overall controller structure.



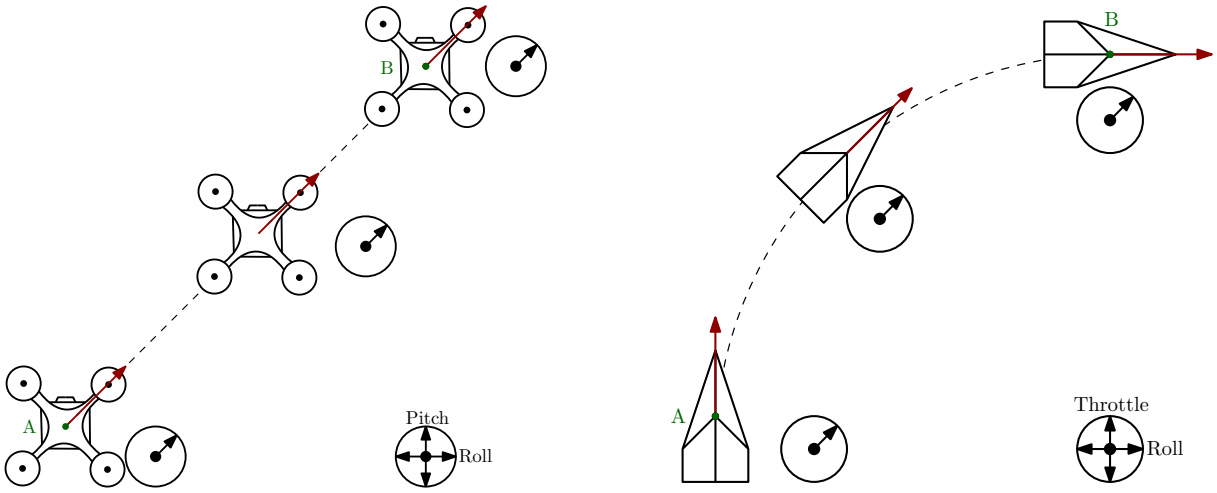
Source: Author.

A detailment of each block is given in the sequence.

3.1 COMMAND MAPPING

One of the challenges when designing a control law for a transitioning vehicle is the pilot control assignment, since multi-rotor and fixed-wing aircraft have fundamentally different dynamics. To illustrate this, Figure 9 depicts possible command sequences to go from point A to B for both type of aircraft. The multi-rotor vehicle, shown on the left side, can create a lateral velocity and go straight to point B without changing its heading direction, while the fixed-wing aircraft, shown on the right side of the figure, typically performs changes on its heading to go from A to B.

Figure 9 – Comparison between typical trajectories of multi-rotor and fixed-wing aircraft. On the bottom part of the two images is depicted a common radio control assignment and besides the aircraft the commands given by the pilot.

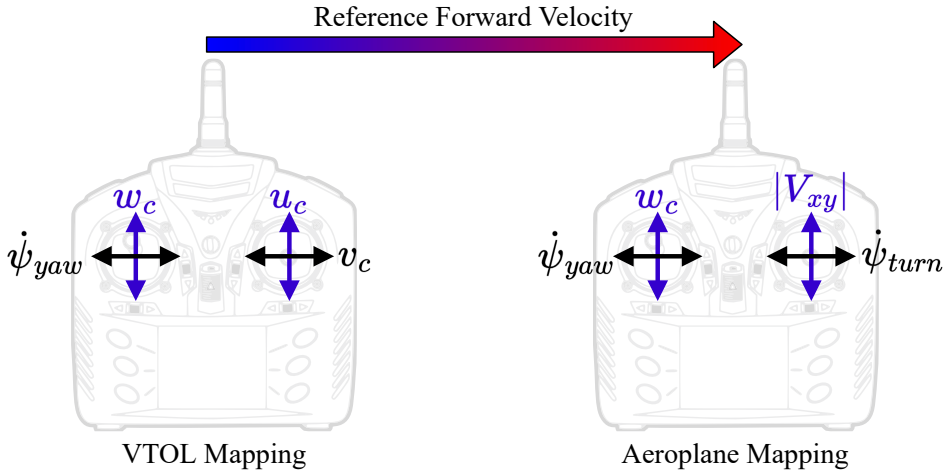


(a) Multi-rotor trajectory. (b) Fixed-Wing trajectory.

Source: Author.

Rather than choosing one of these dynamics for the whole flight envelope, the proposed control strategy is able to transition from one to another depending on the flight regime. This is accomplished by using different command mappings during VTOL and aeroplane modes, as shown in Figure 10. During VTOL mode, the linear velocities on the three axes (u_C, v_C and w_C) and the heading rate $\dot{\psi}_{yaw}$ are commanded, as a common drone configuration. In aeroplane mode, the absolute horizontal velocity, depicted as $|V_{xy}|$ and the climb rate are controlled, mimicking a throttle and elevator command, respectively. In addition, the two remaining channels are mapped to heading rate command, but the right hand stick is a mix of heading rate and roll angle, denoted by $\dot{\psi}_{turn}$, such that the aircraft has a coordinate turn dynamics.

Figure 10 – Adopted command mapping for both VTOL and Aeroplane flight modes.



Source: Author.

To transition between the control modes, a sigmoid function is used to slowly blend the commands and transition from one to another, based on the reference forward velocity $u_{C,ref}$. Mathematically, the blending factor λ is defined as

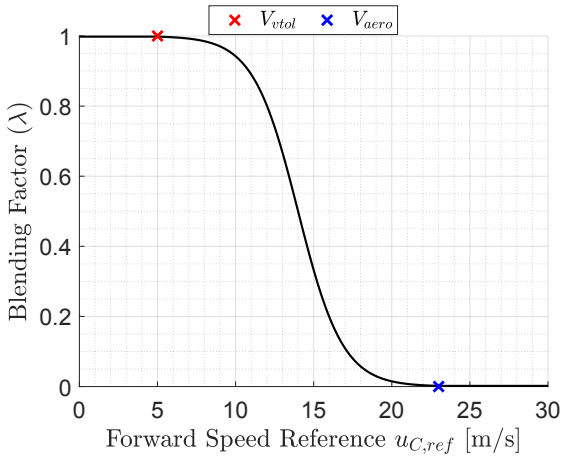
$$\lambda(u_{C,ref}) = \begin{cases} 0 & if\ u_{C,ref} \leq V_{vtol} \\ G(u_{C,ref}) & if\ V_{vtol} < u_{C,ref} < V_{aero} \\ 1 & if\ u_{C,ref} \geq V_{aero} \end{cases} \tag{4}$$

$$G(u_{C,ref}) = \frac{1}{2} \cdot \left[\tanh\left(2 \cdot \pi \cdot \frac{(V_{aero} - u_{C,ref})}{(V_{aero} - V_{vtol})} - \pi \right) + 1 \right]$$

where V_{vtol} and V_{aero} are breakpoints which define the transition between VTOL and aeroplane modes, respectively. The reference speed was used instead of the measured since it is not susceptible to noise. This value is calculated by the velocity controller and corresponds to a feasible reference trajectory. Furthermore, by using it, the final decision to transition or not the aircraft depends only on the pilot's commands.

Figure 11 depicts the evolution of λ in terms of the reference forward speed $u_{C,ref}$.

Figure 11 – Blending factor as a function of the reference forward speed ($u_{C,ref}$). Note that the transition to aeroplane mode starts at V_{vtol} and is completed at V_{aero}.



Source: Author.

Herein, the attitude and velocity controllers are designed based on INDI (Incremental Non-Linear Dynamic Inversion) control technique. The INDI was adopted due to its lower model dependency and robustness against disturbances (VELD, 2016). A brief description of this technique is performed in the sequence. For details about the INDI the reader is refereed to Veld (2016).

3.2 INCREMENTAL NONLINEAR DYNAMIC INVERSION

The INDI formulation considers a non-linear system described by

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{5}$$

where $\boldsymbol{x} \in \mathbb{R}^{n_x \times 1}$ and $\boldsymbol{u} \in \mathbb{R}^{n_u \times 1}$ are the state and control vectors, respectively. Henceforth, the time dependency notation is dropped for better readability. The non-linear system (5) is then linearized at the current operation condition, given by $(\boldsymbol{x}_0, \boldsymbol{u}_0)$, using the first order Taylor series. As a result, it follows that

$$\dot{\boldsymbol{x}} \approx \underbrace{\boldsymbol{f}(\boldsymbol{x}_0,\ \boldsymbol{u}_0)}_{\dot{\boldsymbol{x}}_0} + \underbrace{\frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{x}}\bigg|_{\boldsymbol{x}_0, \boldsymbol{u}_0}}_{\boldsymbol{A}_0} \cdot (\boldsymbol{x} - \boldsymbol{x}_0) + \underbrace{\frac{\partial \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u})}{\partial \boldsymbol{u}}\bigg|_{\boldsymbol{x}_0, \boldsymbol{u}_0}}_{\boldsymbol{B}_0} \cdot (\boldsymbol{u} - \boldsymbol{u}_0)$$

$$\dot{\boldsymbol{x}} \approx \dot{\boldsymbol{x}}_0 + \boldsymbol{A}_0 \cdot (\boldsymbol{x} - \boldsymbol{x}_0) + \boldsymbol{B}_0 \cdot (\boldsymbol{u} - \boldsymbol{u}_0) \tag{6}$$

Assuming that the change in the control input is significantly faster than the plant dynamics, that is $\boldsymbol{x} \approx \boldsymbol{x}_0$, the linear system (6) can be simplified as

$$\dot{\boldsymbol{x}} \approx \dot{\boldsymbol{x}}_0 + \boldsymbol{B}_0 \cdot (\boldsymbol{u} - \boldsymbol{u}_0) \tag{7}$$

If the matrix $\boldsymbol{B}_0$ is known and invertible, a control law can be developed by defining a desired dynamics given by the pseudo control $\boldsymbol{\nu} = \dot{\boldsymbol{x}}$:

$$\boldsymbol{u} = \boldsymbol{u}_0 + \boldsymbol{B}_0^{-1} \cdot (\boldsymbol{\nu} - \underbrace{\dot{\boldsymbol{x}}_0}_{\boldsymbol{\nu}_0})$$

$$\boldsymbol{u} = \boldsymbol{u}_0 + \boldsymbol{B}_0^{-1} \cdot \Delta\boldsymbol{\nu} \tag{8}$$

In practice, the application of the INDI controller comprises the design of the pseudo command $\boldsymbol{\nu}$, generally using a reference model and an error controller, and the computation of $\boldsymbol{B}_0^{-1} \cdot \Delta\boldsymbol{\nu}$, done by a control allocation framework.

To illustrate this, a SISO (Single-Input and Single-Output) system is considered. Let y be the controlled variable and x the state of the system. To use the control law (8), y is differentiated until a relation with the input variable u is obtained, that is

$$y^{(r)} = f(x, u) \tag{9}$$

where $y^{(r)}$ is the r-th derivative of y. Thus, the pseudo control is defined as

$$\nu = y^{(r)} \tag{10}$$

A reference model is used to generate smooth and physically feasible trajectories for the controlled variable to follow. Since only the r-th derivative can be commanded, the reference model must be smooth enough to provide the reference signal's r-th derivative. If the system perfectly tracks the pseudo command ν, then the input output dynamics is given by

$$y = \frac{1}{s^r} \cdot \nu \tag{11}$$

In practice, model mismatches, disturbances, time-delays and non-ideal actuators will make the system deviate from the reference trajectory. Since the linearized system is a chain of integrators, the plant output y will drift away from the reference trajectory. Hence an error controller has to be designed to move the poles into the left half complex plane and bring stability to the system, compensating for the deviations. The error controller dynamics is given by (RAMESH, 2021)

$$\nu_{ec} = \begin{bmatrix} K_{p,y} & K_{p,\dot{y}} & \cdots & K_{p,y^{(r-1)}} \end{bmatrix} \cdot \begin{bmatrix} y_{ref} - y \\ \dot{y}_{ref} - \dot{y} \\ \vdots \\ y_{ref}^{(r-1)} - y^{(r-1)} \end{bmatrix} \tag{12}$$

where the subscript ref indicates the reference values generated by the reference model and the gains are positive constants. The INDI control law does not need a integral error part since it already has an implicit integrating behavior (RAMESH, 2021) (the

demonstration is given in Appendix A). Thus, the commanded pseudo control is obtained by adding the reference and the error controller part, that is

$$\nu_{cmd} = \underbrace{\nu_{ref}}_{y_{ref}^{(r)}} + \nu_{ec} \tag{13}$$

Lastly, the measured r-th derivative of the output (ν_0) is subtracted and a control allocation framework is used to compute the input variable u, as given by (8).

In the scope of this thesis, the attitude and velocity controllers are considered to be the models involved in the computation of the pseudo commands, that is, the reference model and error controller.

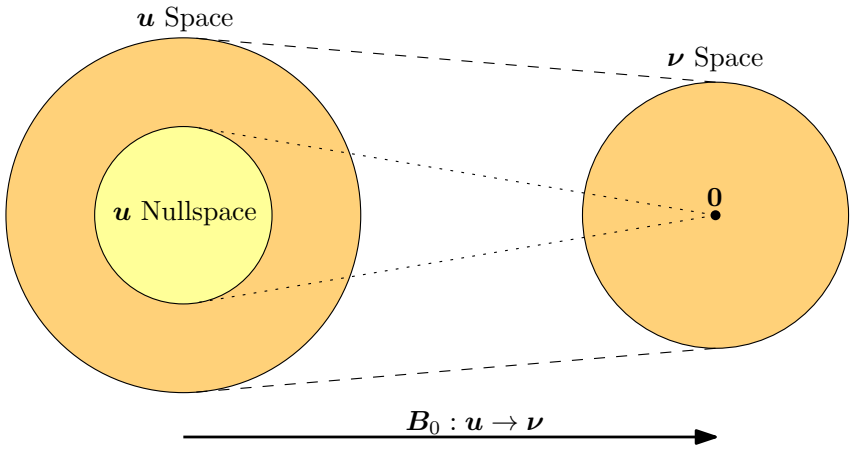
3.2.1 Path Dependency Problem

A redundant system is equipped with more actuators than axes to control (OPPENHEIMER; DOMAN; BOLENDER, 2006). In the scope of this thesis, let $\boldsymbol{\nu} \in \mathbb{R}^m$ and $\boldsymbol{u} \in \mathbb{R}^n$, the system is redundant if $n > m$ and if the matrix $\boldsymbol{B}_0$ has full rank. It follows from this definition that if the system is redundant, then the nullspace of the transformation

$$\boldsymbol{B}_0 : \boldsymbol{u} \rightarrow \boldsymbol{\nu}$$

has dimension greater than zero[1]. What this means is that exist non-zero vectors $\boldsymbol{u}$ that satisfy $\boldsymbol{B}_0 \cdot \boldsymbol{u} = \boldsymbol{0}$, as illustrated in Figure 12.

Figure 12 – Illustration of the nullspace of a redundant system. It consists of all the vectors $\boldsymbol{u}$ that map into the null pseudo command vector. Physically this means that the system can trim with multiple actuators commands.



Source: Author.

[1] This follows from the rank-nullity theorem (LIMA, 2014). The dimension of the nullspace is equal to $dim(\boldsymbol{u}) - rank(\boldsymbol{B}_0)$, that is, $n - m$.

One significant implication is that if the system is redundant, the control law (8) is path dependent. This means that the actuators are not guaranteed to return to their trimmed positions after maneuvering the aircraft and returning to level flight. This follows from the fact that the control law is incremental and the nullspace dimension is greater than zero (DURHAM; BORDIGNON; BECK, 2017). A more detailed explanation can be found in Appendix B.

This is a known problem and the common way to solve it is by adding another command in the nullspace of $\boldsymbol{B}_0$ to restore the actuators to their trimmed position or accordingly to some other objective (DURHAM; BORDIGNON; BECK, 2017). This step can be computationally expensive since it requires augmenting the matrix $\boldsymbol{B}_0$ with another row and performing a new matrix inversion. Instead, an easier and simpler method is employed to solve the path dependency problem by multiplying (8) by $\boldsymbol{B}_0^{-1} \cdot \boldsymbol{B}_0$:

$$\begin{aligned} \boldsymbol{B}_0^{-1} \cdot \boldsymbol{B}_0 \cdot \boldsymbol{u} &= \boldsymbol{B}_0^{-1} \cdot \boldsymbol{B}_0 \cdot \boldsymbol{u}_0 + \boldsymbol{B}_0^{-1} \cdot \boldsymbol{B}_0 \cdot \boldsymbol{B}_0^{-1} \cdot \Delta\boldsymbol{\nu} \\ \boldsymbol{u} &= \boldsymbol{B}_0^{-1} \cdot (\boldsymbol{B}_0 \cdot \boldsymbol{u}_0 + \Delta\boldsymbol{\nu}) \end{aligned} \tag{14}$$

A detailed explanation of why (14) is not path dependent is given in Appendix B. In short, multiplying $\boldsymbol{u}_0$ by $\boldsymbol{B}_0$ prevents the integration of terms in the nullspace of $\boldsymbol{B}_0$, which is the reason why the actuators drift away from their initial trimmed state in the classical control law (8).

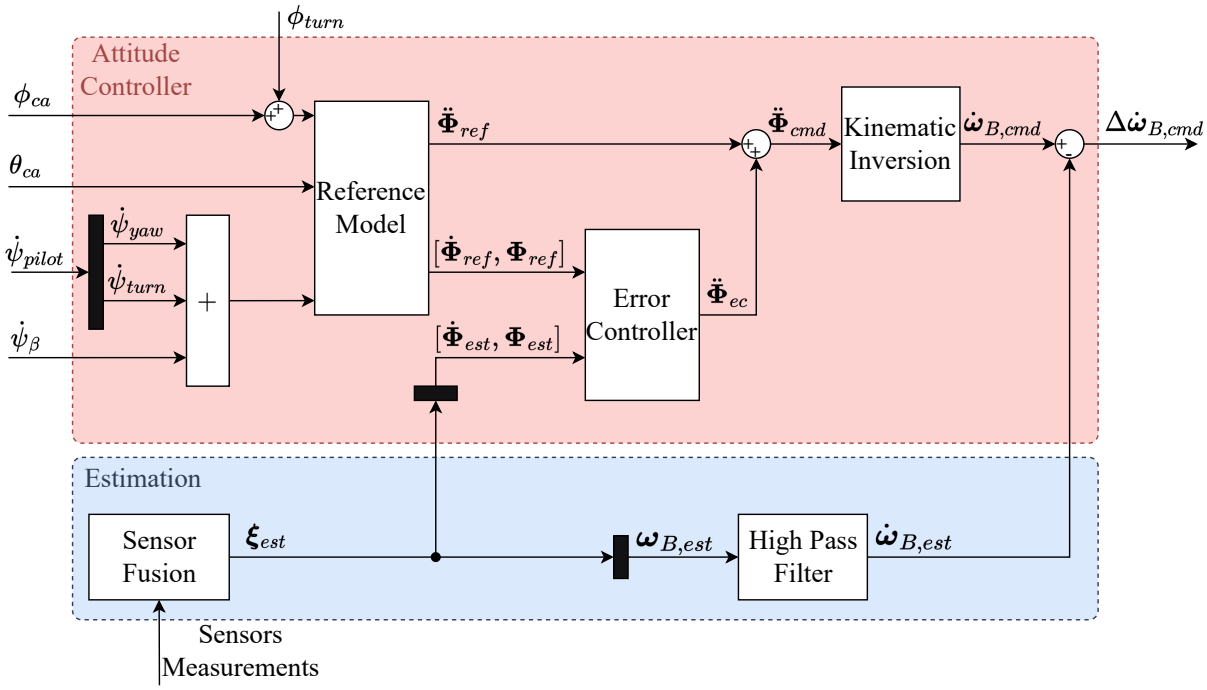
It's worth noting that this method only requires a new matrix multiplication, rather than augmenting $\boldsymbol{B}_0$ and performing a new matrix inversion. This is particularly useful for systems with a large number of actuators, in which computing $\boldsymbol{B}_0^{-1}$ can require a significant amount of computational power.

3.3 ATTITUDE CONTROLLER

Figure 13 shows the attitude controller structure. The controller receives pitch, roll and heading rate commands, which come from the command mapping, turn coordinator and control allocation. The input from those blocks are summed and the input vector $\boldsymbol{y}$ is then comprised of

$$\underbrace{\begin{bmatrix} \phi_{cmd} \\ \theta_{cmd} \\ \dot{\psi}_{cmd} \end{bmatrix}}_{\boldsymbol{y}} = \begin{bmatrix} \phi_{ca} + \phi_{turn} \\ \theta_{ca} \\ \dot{\psi}_{yaw} + \dot{\psi}_{turn} + \dot{\psi}_{\beta} \end{bmatrix} \tag{15}$$

Figure 13 – Attitude controller block diagram.



Source: Author.

By analysing the rotational equation of motion (3), the control law (14) can be used by defining $\boldsymbol{\nu} = \dot{\boldsymbol{\omega}}_B$. To control $\boldsymbol{y}$, the reference model computes reference $\ddot{\mathbf{\Phi}}$ values since they are in the same algebraic level as $\dot{\boldsymbol{\omega}}_B$, that is

$$\dot{\boldsymbol{\omega}}_B = \boldsymbol{f}(\ddot{\Phi}, \boldsymbol{x}) \tag{16}$$

The transfer functions of each channel are given by

$$\begin{aligned}
\underbrace{R(s)_{\phi_{ref},\ \phi_{cmd}}}_{R_\phi} &= \frac{\omega_\phi^2}{s^2 + 2 \cdot \xi_\phi \cdot \omega_\phi \cdot s + \omega_\phi^2} \\
\underbrace{R(s)_{\theta_{ref},\ \theta_{cmd}}}_{R_\theta} &= \frac{\omega_\theta^2}{s^2 + 2 \cdot \xi_\theta \cdot \omega_\theta \cdot s + \omega_\theta^2} \\
\underbrace{R(s)_{\dot{\psi}_{ref},\ \dot{\psi}_{cmd}}}_{R_{\dot{\psi}}} &= \frac{\omega_{\dot{\psi}}}{s + \omega_{\dot{\psi}}}
\end{aligned} \tag{17}$$

$$\begin{bmatrix} \phi_{ref} \\ \dot{\phi}_{ref} \\ \ddot{\phi}_{ref} \\ \theta_{ref} \\ \dot{\theta}_{ref} \\ \ddot{\theta}_{ref} \\ \dot{\psi}_{ref} \\ \ddot{\psi}_{ref} \end{bmatrix} = \begin{bmatrix} R_\phi & 0 & 0 \\ R_\phi \cdot s & 0 & 0 \\ R_\phi \cdot s^2 & 0 & 0 \\ 0 & R_\theta & 0 \\ 0 & R_\theta \cdot s & 0 \\ 0 & R_\theta \cdot s^2 & 0 \\ 0 & 0 & R_{\dot{\psi}} \\ 0 & 0 & R_{\dot{\psi}} \cdot s \end{bmatrix} \cdot \begin{bmatrix} \phi_{cmd} \\ \theta_{cmd} \\ \dot{\psi}_{cmd} \end{bmatrix} \tag{18}$$

where the parameters are chosen accordingly with the vehicle capabilities and from a handling qualities perspective. Second order transfer functions are used in the roll and pitch channel since they have to be differentiated twice to reach the dynamic level of the input.

The error controller of each channel is given by

$$\underbrace{\begin{bmatrix} \ddot{\phi}_{ec} \\ \ddot{\theta}_{ec} \\ \ddot{\psi}_{ec} \end{bmatrix}}_{\ddot{\boldsymbol{\Phi}}_{ec}} = \begin{bmatrix} K_{p,\,\phi} & K_{d,\,\phi} & 0 & 0 & 0 \\ 0 & 0 & K_{p,\,\theta} & K_{d,\,\theta} & 0 \\ 0 & 0 & 0 & 0 & K_{p,\,\dot{\psi}} \end{bmatrix} \cdot \begin{bmatrix} e_\phi \\ e_{\dot{\phi}} \\ e_\theta \\ e_{\dot{\theta}} \\ e_{\dot{\psi}} \end{bmatrix} \tag{19}$$

where e depicts the tracking error and the subscript denotes the respective state. Lastly, the commanded Euler accelerations are obtained by

$$\ddot{\boldsymbol{\Phi}}_{cmd} = \ddot{\boldsymbol{\Phi}}_{ec} + \ddot{\boldsymbol{\Phi}}_{ref}. \tag{20}$$

Next, the commanded Euler angle accelerations ($\ddot{\boldsymbol{\Phi}}_{cmd}$) are transformed into Body frame angular accelerations ($\dot{\boldsymbol{\omega}}_{B,cmd}$) using a kinematic inversion. This transformation is done because $\dot{\boldsymbol{\omega}}_B$ contains an algebraic relation with the actuators, as given by (3). The relation between $\ddot{\boldsymbol{\Phi}}$ and $\dot{\boldsymbol{\omega}}_B$ is derived from the strap down equation (ROSKAM, 2001)

$$\underbrace{\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}}_{\dot{\boldsymbol{\Phi}}} = \underbrace{\begin{bmatrix} 1 & \sin\phi \cdot \tan\theta & \cos\phi \cdot \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \dfrac{\sin\phi}{\cos\theta} & \dfrac{\cos\phi}{\cos\theta} \end{bmatrix}}_{\boldsymbol{M}(\boldsymbol{\Phi})} \cdot \underbrace{\begin{bmatrix} p \\ q \\ r \end{bmatrix}}_{\boldsymbol{\omega}_B} \tag{21}$$

To obtain a relation between the accelerations, (21) is differentiated with respect to time and $\dot{\boldsymbol{\omega}}_B$ is isolated, obtaining

$$\begin{aligned} \ddot{\boldsymbol{\Phi}} &= \dot{\boldsymbol{M}}(\boldsymbol{\Phi}, \dot{\boldsymbol{\Phi}}) \cdot \boldsymbol{\omega}_B + \boldsymbol{M}(\boldsymbol{\Phi}) \cdot \dot{\boldsymbol{\omega}}_B \\ \dot{\boldsymbol{\omega}}_B &= [\boldsymbol{M}(\boldsymbol{\Phi})]^{-1} \cdot (\ddot{\boldsymbol{\Phi}} - \dot{\boldsymbol{M}}(\boldsymbol{\Phi}, \dot{\boldsymbol{\Phi}}) \cdot \boldsymbol{\omega}_B) \end{aligned} \tag{22}$$

Note that this relation uses the angular rates and the Euler derivatives. As those measurements are usually noisy, the reference values are used, obtaining

$$\dot{\boldsymbol{\omega}}_{B,cmd} = [\boldsymbol{M}(\boldsymbol{\Phi}_{ref})]^{-1} \cdot (\ddot{\boldsymbol{\Phi}}_{cmd} - \dot{\boldsymbol{M}}(\boldsymbol{\Phi}_{ref}, \dot{\boldsymbol{\Phi}}_{ref}) \cdot \boldsymbol{\omega}_{B,ref}) \tag{23}$$

To obtain $\boldsymbol{\omega}_{B,ref}$, Equation (23) is solved for $\ddot{\boldsymbol{\Phi}}_{B,ref}$, that is

$$\dot{\boldsymbol{\omega}}_{B,ref} = [\boldsymbol{M}(\boldsymbol{\Phi}_{ref})]^{-1} \cdot (\ddot{\boldsymbol{\Phi}}_{ref} - \dot{\boldsymbol{M}}(\boldsymbol{\Phi}_{ref}, \dot{\boldsymbol{\Phi}}_{ref}) \cdot \boldsymbol{\omega}_{B,ref}) \tag{24}$$

and the result is integrated to obtain $\boldsymbol{\omega}_{B,ref}$.

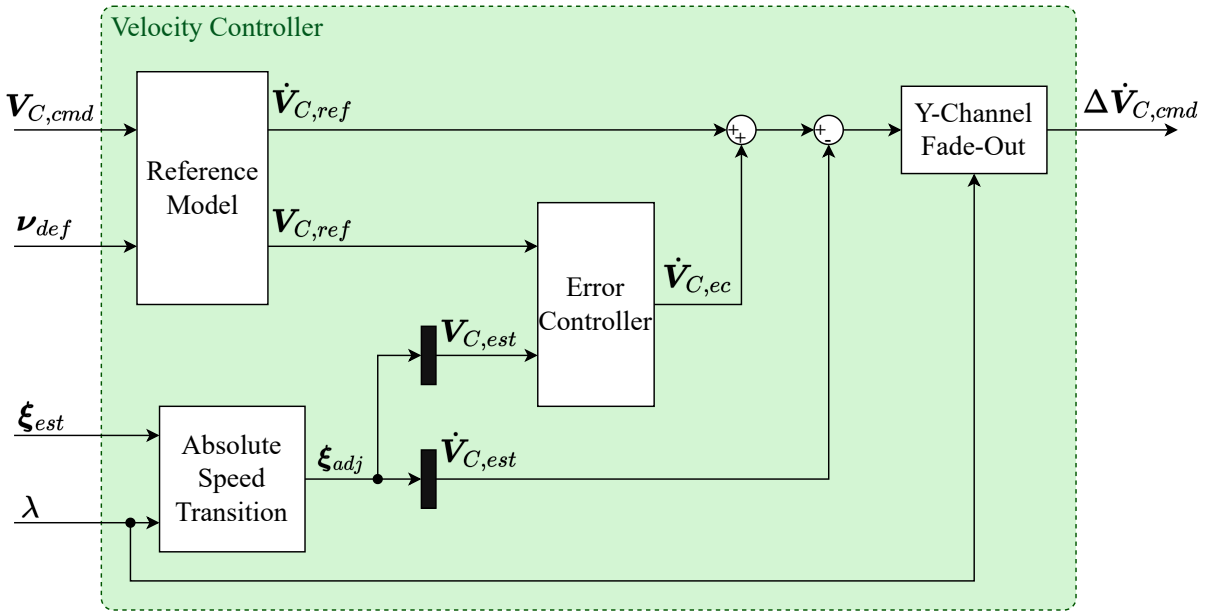
As illustrated in Figure 13, the estimated angular accelerations $\dot{\boldsymbol{\omega}}_{B,est}$ are calculated by using a high pass filter on the angular rates measurements, following the approach of E.J.J Smeur (2018). The filter transfer function is shown in (25). An important point to notice is that the actuator estimation signal has to be delayed using the same filter in order to synchronize the two signals, otherwise, unsynchronized delays will cause oscillatory behavior (SMEUR, E.J.J, 2018). Hence, the low pass filter shown in Figure 17 uses the same bandwidth.

$$F(s)_{\dot{\boldsymbol{\omega}}_{B,est},\ \boldsymbol{\omega}_{B,est}} = \frac{\omega_{filt}^2 \cdot s}{s^2 + 2 \cdot \xi_{filt} \cdot \omega_{filt} \cdot s + \omega_{filt}^2} \tag{25}$$

3.4 VELOCITY CONTROLLER

The velocity controller structure is depicted in Figure 14. This controller receives velocity commands directly from the pilot inputs and, similar to how the attitude controller generates the pseudo commands, a reference model and an error controller are employed. Additionally, two new blocks are used to transition the controller to its aeroplane mode: the Absolute Speed Transition is responsible to switch the controlled variable from u_c to the absolute horizontal speed $|V_{xy}|$ and the other switches off the Y-Channel during aeroplane mode.

Figure 14 – Velocity controller block diagram.



Source: Author.

The velocity controller vector command $\boldsymbol{y}$ is given by

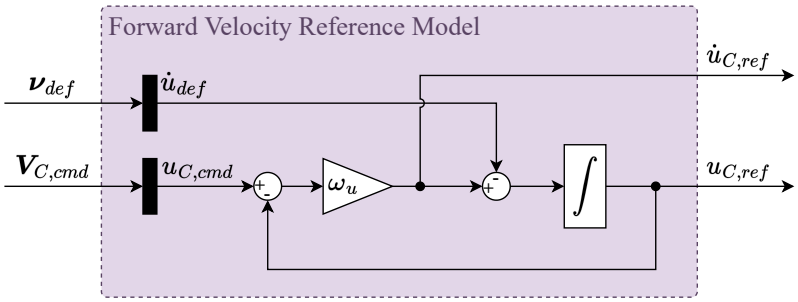
$$\underbrace{\begin{bmatrix} u_{C,cmd} \\ v_{C,cmd} \\ w_{C,cmd} \end{bmatrix}}_{\boldsymbol{y}} = \underbrace{\begin{bmatrix} u_{C,pilot} \\ v_{C,pilot} \\ w_{C,pilot} \end{bmatrix}}_{\boldsymbol{V}_{C,pilot}} \tag{26}$$

Note that (2) gives a relation between the control vector $\boldsymbol{u}$ and $\dot{\boldsymbol{y}}$. Therefore, the pseudo control is defined as $\dot{\boldsymbol{V}}_C$. A first order reference model was initially designed to compute the reference trajectories, but unfeasible trajectories were being generated by the forward speed reference model, especially during the back-transition phase (transition from aeroplane to VTOL mode) due to the absence of actuators to break the aircraft, such as spoilers. Since this is a physical limitation of the system, it was chosen to simply utilize a PCH (Pseudo Control Hedging) to slow down the reference model and avoid high control actions. The way hedging works is by taking the information of saturation from the control allocation and using it to compute new reference trajectories. This is done by subtracting the pseudo command deficiency $\boldsymbol{\nu}_{def}$ from the highest derivative of the reference model, as depicted in Figure 15. Mathematically, this command deficiency is calculated by

$$\boldsymbol{\nu}_{def} = \boldsymbol{\nu}_{cmd} - \boldsymbol{\nu}_{alloc} \tag{27}$$

where $\boldsymbol{\nu}_{cmd}$ is the commanded pseudo command and $\boldsymbol{\nu}_{alloc}$ is the allocated pseudo command. Note that $\boldsymbol{\nu}_{alloc}$ is not always equal to $\boldsymbol{\nu}_{cmd}$ due to the actuators limits.

Figure 15 – Block diagram of the forward velocity reference model with PCH.



Source: Author.

The transfer functions of each channel are given by

$$\begin{aligned} \underbrace{R(s)_{u_{ref},\, u_{cmd}}}_{R_u} &= \frac{\omega_u - \dot{u}_{C,\, def}}{s + \omega_u} \\ \underbrace{R(s)_{v_{ref},\, v_{cmd}}}_{R_v} &= \frac{\omega_v}{s + \omega_v} \\ \underbrace{R(s)_{w_{ref},\, w_{cmd}}}_{R_w} &= \frac{\omega_w}{s + \omega_w} \end{aligned} \tag{28}$$

$$\begin{bmatrix} u_{C,\, ref} \\ \dot{u}_{C,\, ref} \\ v_{C,\, ref} \\ \dot{v}_{C,\, ref} \\ w_{C,\, ref} \\ \dot{w}_{C,\, ref} \end{bmatrix} = \begin{bmatrix} R_u & 0 & 0 \\ R_u \cdot s + \dot{u}_{C,\, def} & 0 & 0 \\ 0 & R_v & 0 \\ 0 & R_v \cdot s & 0 \\ 0 & 0 & R_w \\ 0 & 0 & R_w \cdot s \end{bmatrix} \cdot \begin{bmatrix} u_{C,\, cmd} \\ v_{C,\, cmd} \\ w_{C,\, cmd} \end{bmatrix} \tag{29}$$

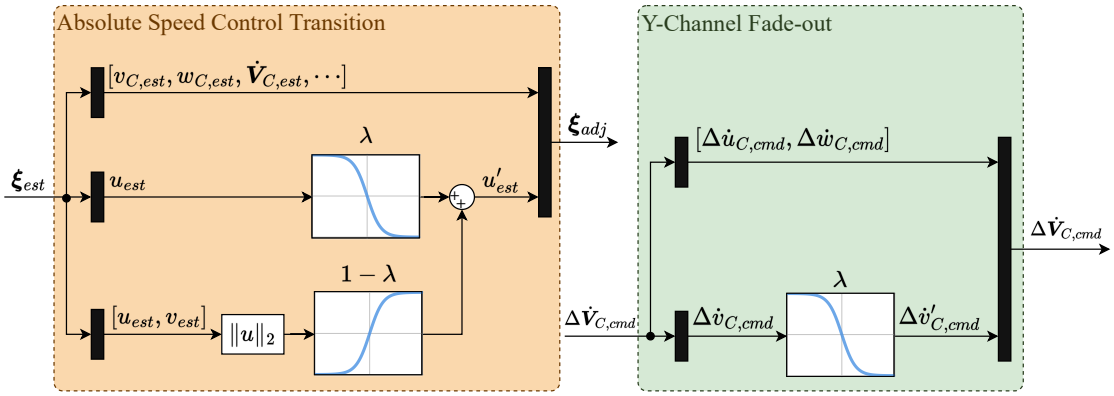
The error controller of each channel is given by

$$\underbrace{\begin{bmatrix} \dot{u}_{C,\, ec} \\ \dot{v}_{C,\, ec} \\ \dot{w}_{C,\, ec} \end{bmatrix}}_{\dot{\boldsymbol{V}}_{C,\, ec}} = \begin{bmatrix} K_{p,\, u} & 0 & 0 \\ 0 & K_{p,\, v} & 0 \\ 0 & 0 & K_{p,\, w} \end{bmatrix} \cdot \begin{bmatrix} e_u \\ e_v \\ e_w \end{bmatrix} \tag{30}$$

As aforementioned, during aeroplane mode the absolute horizontal velocity is controlled, rather than its components u_C and v_C. This is achieved by simply using the absolute horizontal velocity as a feedback for the X-channel during aeroplane mode. Moreover, during high speed flight, lateral control is achieved by changing the direction of the velocity vector rather than producing a lateral velocity. Therefore, the velocity controller Y-channel is turned off during aeroplane mode by multiplying the pseudo command $\Delta\dot{v}_C$ by the blending factor. In that way, the lateral control is purely done by means of the

turn coordinator block during aeroplane mode. Figure 16 shows how both transitions were implemented.

Figure 16 – Block diagram of the aeroplane mode transition logic in the velocity controller.



Source: Author.

3.5 TURN COORDINATOR

During wingborne flight, changes in the heading direction are accomplished by rolling the aircraft rather than yawing. Hence, the turn coordinator shown in Figure 13 is used to generate roll angle commands for turns during aeroplane mode.

According to Roskam (2001), during a steady level turn, the following relation holds:

$$\dot{\psi} = \frac{g \cdot \tan\phi}{V_{kin}} \tag{31}$$

where g is the acceleration of gravity and V_{kin} is aircraft absolute kinematic velocity.

Thus, a steady roll angle for the desired heading rate command is calculated by solving (31) for the roll angle:

$$\phi_{turn} = \arctan\left(\frac{\dot{\psi}_{turn} \cdot \|\boldsymbol{V}_{O,est}\|_2}{g}\right) \tag{32}$$

where $\boldsymbol{V}_{O,est}$ is the estimated kinematic velocity vector denoted in the NED frame.

Note that the commands during hover flight are small since (32) is approximately proportional to the velocity. As such, (32) provides a natural transition to wingborne flight behavior as the speed increases.

As shown in Figure 13, the $\dot{\psi}_{turn}$ command is also sent to the attitude controller, which provides a coordination between the roll and yaw commands. If the pilot wants to yaw the aircraft, a separate heading rate command, depicted as $\dot{\psi}_{yaw}$ is sent directly to the attitude controller, bypassing the turn coordinator.

In order to maintain zero sideslip angle during wingborne flight, a sideslip controller is employed. The sideslip angle is estimated based on the lateral accelerations measure-

ments, since no direct measurements were available. This logic follows from the relation that, for small sideslip angles, the lateral force due to a sideslip angle is approximate as (ROSKAM, 2001)

$$Y_B(\beta) = \frac{1}{2} \cdot \rho \cdot V_{wind}^2 \cdot S_{ref} \cdot C_{y\beta} \cdot \beta \tag{33}$$

where Y_B is the y-axis force denoted in the body frame, ρ is the air density, S_{ref} is the reference area, V_{wind} is the relative wind speed and $C_{y\beta}$ is the derivative of the lateral force coefficient with respect to the sideslip angle. Dividing (33) by the vehicle mass, the relation is written as a function of the lateral specific force:

$$f_{B,y}(\beta) = \frac{1}{m} \cdot \frac{1}{2} \cdot \rho \cdot V_{wind}^2 \cdot S_{ref} \cdot C_{y\beta} \cdot \beta \tag{34}$$

This relation shows that the lateral acceleration is proportional to the sideslip angle and, thus, by using the lateral specific force measurements, the heading rate can be changed to eliminate the sideslip (SMEUR, E.J.J, 2018). This is done by adding a heading rate command proportional to the estimated lateral specific force:

$$\dot{\psi}_{cmd} = \dot{\psi}_{turn} + \dot{\psi}_{yaw} - \underbrace{f_{B,y,est} \cdot K_{p,\ \beta} \cdot (1 - \lambda)}_{\dot{\psi}_\beta} \tag{35}$$

where $\dot{\psi}_{yaw}$ and $\dot{\psi}_{turn}$ are commands given by the pilot and $K_{p,\ \beta}$ is a positive gain. Note that the multiplication by $(1 - \lambda)$ guarantees that the controller does not act during the VTOL phase.

3.6 CONTROL ALLOCATION

The control allocation block is used to determine the actuator commands so that the commanded pseudo controls are achieved. Mathematically, the real actuator commands are chosen to satisfy

$$\boldsymbol{\nu}_{cmd} = \boldsymbol{B}_0 \cdot \boldsymbol{u}_{real} \tag{36}$$

$$\text{Subject to } \boldsymbol{u}_{real} \in \mathbb{U} = \{\boldsymbol{u}_{real} \in \mathbb{R}^n \mid \underline{\boldsymbol{u}} < \boldsymbol{u}_{real} < \bar{\boldsymbol{u}}\}$$

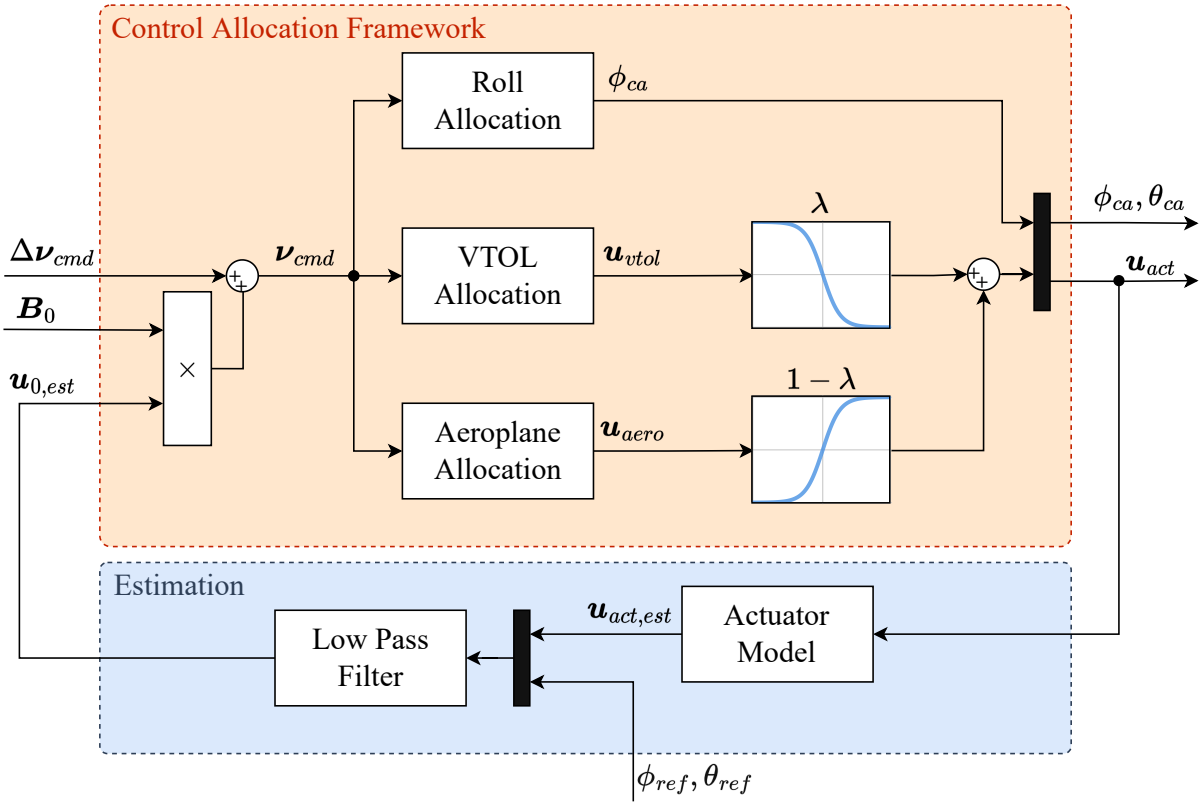
where $\underline{\boldsymbol{u}}$ and $\bar{\boldsymbol{u}}$ represent the lower and upper limits for the actuators, respectively and n is the number of actuators. Therefore, if (36) is feasible, the accelerations generated by the attitude and velocity controllers are imposed to the aircraft.

As shown in Figure 8, the roll and pitch angle commands are calculated in the control allocation as they are used to produce accelerations. For this reason, they are treated as virtual actuators in the control allocation framework, as adopted by Zhang et al. (2018). Therefore, the considered real control vector contains not only the actuators, but the roll and pitch angle as well. That is,

$$\boldsymbol{u}_{real} = \begin{bmatrix} \omega_1 & \cdots & \omega_8 & \delta_1 & \cdots & \delta_4 & \phi_{ca} & \theta_{ca} \end{bmatrix}^T_{14\times 1}$$

The developed allocation strategy is illustrated in Figure 17. To transition the actuators from VTOL to aeroplane mode, two control allocation blocks are run in parallel, and their solutions are interpolated using the blending factor λ. The VTOL allocation runs with all the motors and the pitch angle while the aeroplane allocation runs with the control surfaces, horizontal motors and pitch angle. The actuators commands calculated by the VTOL and aeroplane allocation are denoted by $\boldsymbol{u}_{vtol}$ and $\boldsymbol{u}_{aero}$, respectively. The allocation to the roll angle is done separately since it's the only virtual control with lateral acceleration authority.

Figure 17 – Control allocation framework. Note that the incremental pseudo command $\Delta\boldsymbol{\nu}_{cmd}$ is added with $\boldsymbol{B}_0 \cdot \boldsymbol{u}_0$ in order to solve the path dependency problem, as explained in Section 3.2.1. Following to the blending logic, the command vector from the VTOL allocation ($\boldsymbol{u}_{vtol}$) and from the aeroplane allocation ($\boldsymbol{u}_{aero}$) are linearly interpolated using the blending factor λ such that the controller continuously transition from one allocation mode to the other. In the estimation framework, a first order actuator model is used to estimate the actuators states. The virtual controls states are estimated using the reference values. Although measurements were available, the reference values were chosen since they are not affected by noise. The estimated vector is filtered using a low pass filter with same bandwidth as the filter used in (25), such that the estimated accelerations and the actuators states have the same time delay.



Source: Author.

Assuming that the commanded pseudo control is feasible for both VTOL and aeroplane allocation, the blending is performed as follows:

$$\begin{aligned} \boldsymbol{\nu}_{cmd} &= \boldsymbol{B} \cdot \boldsymbol{u}_{vtol} = \boldsymbol{B} \cdot \boldsymbol{u}_{aero} \\ \boldsymbol{\nu}_{cmd} &= \boldsymbol{B} \cdot \boldsymbol{u}_{vtol} \cdot \lambda + \boldsymbol{B} \cdot \boldsymbol{u}_{aero} \cdot (1 - \lambda) \end{aligned} \tag{37}$$

Moreover, it can also be proved that $\boldsymbol{u}_{vtol} \cdot \lambda + \boldsymbol{u}_{aero} \cdot (1 - \lambda) \in \mathbb{U}$ if both $\boldsymbol{u}_{vtol}$ and $\boldsymbol{u}_{aero}$ are within the actuators limits:

$$\begin{aligned} \underline{\boldsymbol{u}} < \boldsymbol{u}_{vtol} < \bar{\boldsymbol{u}} \\ \underline{\boldsymbol{u}} < \boldsymbol{u}_{aero} < \bar{\boldsymbol{u}} \end{aligned} \tag{38}$$

Considering $\lambda \in [0,\, 1]$,

$$\begin{aligned} \underline{\boldsymbol{u}} \cdot \lambda \quad &< \quad \boldsymbol{u}_{vtol} \cdot \lambda \quad < \quad \bar{\boldsymbol{u}} \cdot \lambda \\ \underline{\boldsymbol{u}} \cdot (1 - \lambda) &< \boldsymbol{u}_{aero} \cdot (1 - \lambda) < \bar{\boldsymbol{u}} \cdot (1 - \lambda) \end{aligned} \tag{39}$$

Adding these two inequalities, the following is obtained:

$$\underline{\boldsymbol{u}} < \boldsymbol{u}_{vtol} \cdot \lambda + \boldsymbol{u}_{aero} \cdot (1 - \lambda) < \bar{\boldsymbol{u}} \tag{40}$$

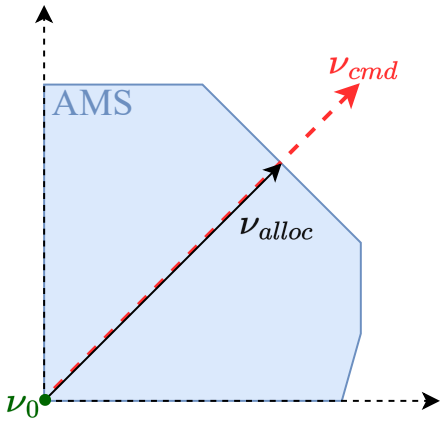
Equations (37) and (40) show that, in despite of the allocation of both modes being done separately, if the desired commands are feasible in both allocations, the blended commands are also feasible and correctly map to the desired pseudo command.

However, if the commanded pseudo control is not feasible, the two techniques described in Sections 3.6.1 and 3.6.2 can be used to find a solution withing the actuators bounds. These two techniques are employed in both allocations, as detailed in Sections 3.6.3 and 3.6.4.

3.6.1 Redistributed Scaled Pseudo Inverse

The RSPI (Redistributed Scaled Pseudo Inverse) method is a recursive algorithm for accounting for actuator limits. It provides the maximum attainable solution considering the actuator limits, as illustrated in Figure 18. The AMS (Attainable Moments Subset) is a nontrivial set that represents the maximum pseudo controls the vehicle can produce and it's given by the physical limitations of the system, such as the actuators limits. The RSPI is capable of reaching the whole control volume, while simpler methods like the pseudoinverse have a limited usable volume (STEPHAN; FICHTER, 2019). A detailed explanation of the algorithm is presented in Appendix C. For a mathematical explanation, the reader is refereed to Stephan and Fichter (2019).

Figure 18 – Redistributed pseudo inverse solution in a bi-dimensional subspace.



Source: Author.

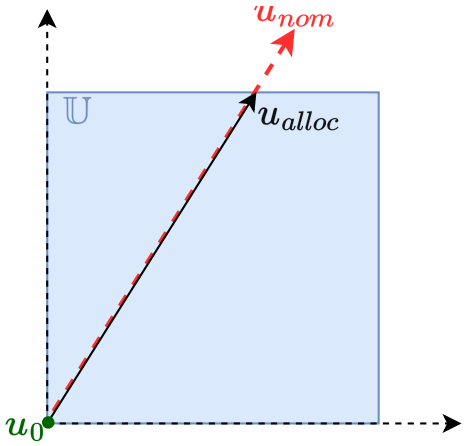
3.6.2 Scaled Pseudo Inverse

Another technique employed in this work is the scaled pseudo inverse. This method starts by taking the nominal pseudo inverse solution $\boldsymbol{u}_{nom}$, that is

$$\boldsymbol{u}_{nom} = \boldsymbol{B}_0^{-1} \cdot \boldsymbol{\nu}_{cmd}$$

if $\boldsymbol{u}_{nom}$ is within the actuators limits, the solution is used, otherwise, $\boldsymbol{u}_{nom}$ is scaled to the actuators bounds, as illustrated in Figure 19. It's worth noting that there is no guarantee that the scaled solution will be on the edge of the AMS and, therefore, some physically achievable commands are not found using this method (STEPHAN; FICHTER, 2019). Besides that, this method has faster computation compared to the RSPI and is adopted in less critical parts.

Figure 19 – Scaled pseudo inverse solution. Note the difference with Figure 18, where the pseudo command is scaled to the AMS edge.

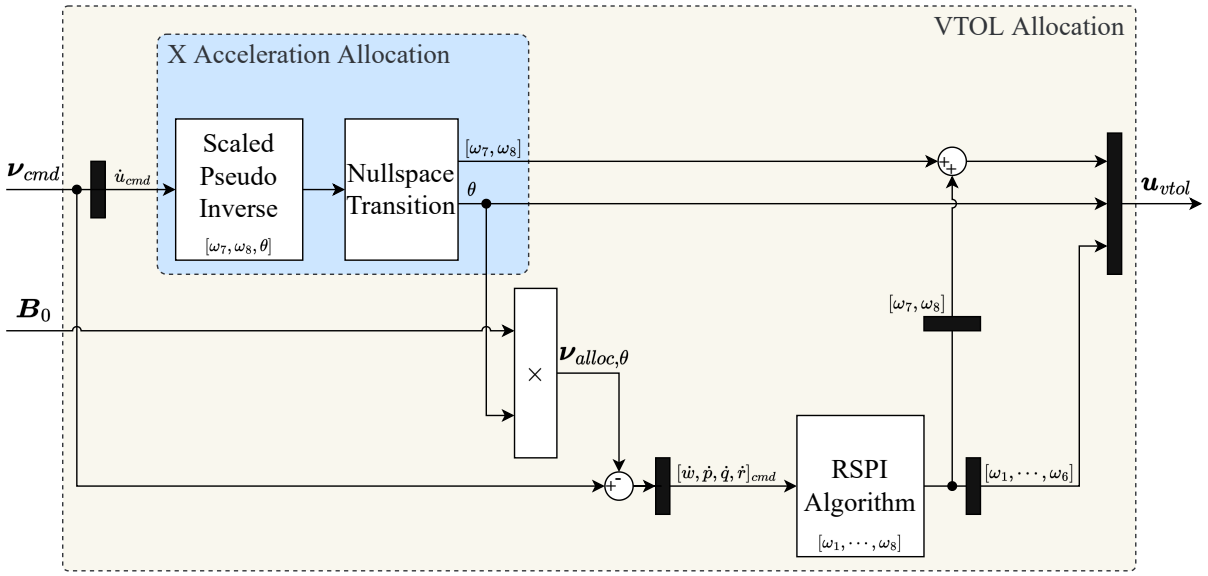


Source: Author.

3.6.3 VTOL Allocation

Figure 20 depicts the control allocation for the VTOL mode. For better readability, each allocation block has in its bottom part a indication of which actuators are used to calculate the desired pseudo command.

Figure 20 – VTOL control allocation. Following the block diagram, the allocation to the forward acceleration command ($\dot{u}_C$) is done first by using the pitch angle and the horizontal motors. A nullspace transition is performed in order to preferentially tilt the aircraft or use the motors to go forward depending on the flight regime. The effects of this first allocation are then accounted by multiplying the pitch angle command by the proper elements of $\boldsymbol{B}_0$ and subtracting the result from the pseudo commands before the RSPI algorithm, where the remaining pseudo commands are allocated.



Source: Author.

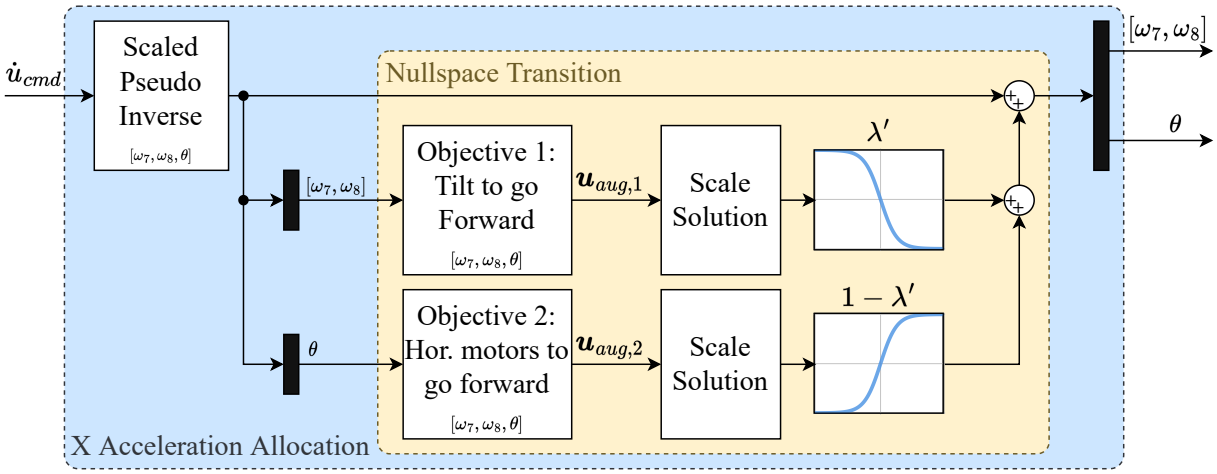
The $\dot{u}$ allocation is done separately for several reasons: first, this structure prevents that saturation in the X acceleration channel affects the allocation on the other axes, since the RSPI algorithm scales the whole command vector. Moreover, only the horizontal motors and the pitch angle can produce changes in the X acceleration. Therefore, by doing this step separately, the matrix inversion is simpler and computationally cheaper.

The nullspace transition block was designed such that, at low speeds, the aircraft preferentially tilts and at higher speeds, the use of the horizontal motors is preferred. This choice creates a more continuous behavior during hover since the aircraft can only accelerate backwards by tilting and, at higher speeds, this creates a more natural transition to wingborne flight.

In practice, the nullspace transition is done by adding an augmenting command to the scaled pseudo inverse solution. Figure 21 depicts in more details how it is performed.

Two augmenting commands are calculated using different objectives and their solutions are interpolated using a blending factor λ'.

Figure 21 – Nullspace transition block diagram. The results from the scaled pseudo inverse are used to calculate two augmenting commands $\boldsymbol{u}_{aug,1}$ and $\boldsymbol{u}_{aug,2}$, these commands are scaled in case of saturation and the final result is a linear combination of both based on the blending factor λ'.



Source: Author.

The augmenting command 1, that is, control the aircraft by tilting, is calculated by the following matrix inversion:

$$\begin{bmatrix} 0 \\ -(\omega_{7,\ alloc} + \omega_{8,\ alloc}) + 2 \cdot \omega_{idle} \end{bmatrix} = \begin{bmatrix} \frac{\partial \dot{u}_c}{\partial \omega_7} & \frac{\partial \dot{u}_c}{\partial \omega_8} & \frac{\partial \dot{u}_c}{\partial \theta} \\ 1 & 1 & 0 \end{bmatrix} \cdot \underbrace{\begin{bmatrix} \omega_{7,\ aug,\ 1} \\ \omega_{8,\ aug,\ 1} \\ \theta_{aug,\ 1} \end{bmatrix}}_{u_{aug,1}} \tag{41}$$

where $\omega_{7,alloc}$ and $\omega_{8,alloc}$ are the horizontal motors command calculated by the scaled pseudo inverse block and ω_{idle} is the idle rotation speed of the horizontal motors. The multiplication of this term by two is due to the fact that the constraint command is given as a sum of the rotation speed of the two motors. Note that this objective actually tries to set the horizontal motors to idle speed and, as a result, having to tilt to control the aircraft.

Similarly, the second augmenting command is calculated by the following matrix inversion:

$$\begin{bmatrix} 0 \\ -\theta_{alloc} + \theta_{des} \end{bmatrix} = \begin{bmatrix} \frac{\partial \dot{u}_c}{\partial \omega_7} & \frac{\partial \dot{u}_c}{\partial \omega_8} & \frac{\partial \dot{u}_c}{\partial \theta} \\ 0 & 0 & 1 \end{bmatrix} \cdot \underbrace{\begin{bmatrix} \omega_{7,\ aug,\ 2} \\ \omega_{8,\ aug,\ 2} \\ \theta_{aug,\ 2} \end{bmatrix}}_{u_{aug,2}} \tag{42}$$

where θ_{alloc} is the pitch angle calculated by the scaled pseudo inverse block and θ_{des} is a constant. This objective tries to set the pitch angle to θ_{des} and, hence, the allocation has to compensate that by using the horizontal motors.

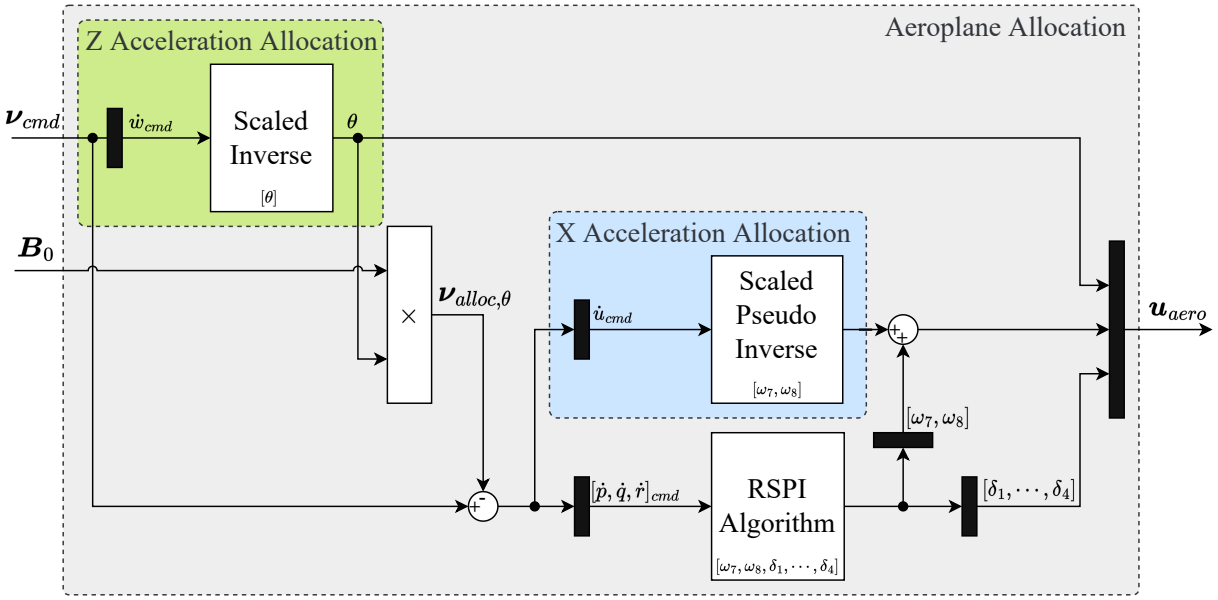
The blending factor λ' is calculated using (4), but with different breakpoints in a way that the transition to objective 2 is already completed before Aeroplane mode.

Note that the nullspace transition does not impose a hard constraint for the allocation. As a result, the horizontal motors can still be commanded to push the aircraft forward if the pseudo command $\dot{u}_C$ is high enough and the aircraft can still tilt backwards in order to break at higher speeds.

3.6.4 Aeroplane Allocation

The aeroplane allocation is illustrated in Figure 22. For the same reasons explained previously, the $\dot{w}_C$ and $\dot{u}_C$ allocations are done separately, while the RSPI allocation takes care of the remaining pseudo commands.

Figure 22 – Aeroplane control allocation. First the Z-axis acceleration ($\dot{w}_C$) is allocated using the pitch angle. As will be explained in the following section, the changes in the pitch angle are approximated as changes in the angle of attack. Therefore, the Z-axis acceleration is controlled by changing the amount of lift generated. Continuing the block diagram, the effects of the allocated pitch angle are then accounted by subtracting the pseudo commands generated. Lastly, a scaled pseudo inverse and a RSPI algorithm are used to allocate the remaining pseudo commands.



Source: Author.

3.7 ONBOARD MODEL

The onboard plant model is used to calculate the effectiveness matrix $\boldsymbol{B}_0$, which has the algebraic relation between the pseudo commands $\boldsymbol{\nu} = [\dot{\boldsymbol{V}}_{C,cmd}^T \; \dot{\boldsymbol{\omega}}_{B,cmd}^T]^T$ and the real and virtual actuators:

$$\boldsymbol{B}_0 = \begin{bmatrix} \frac{\partial \dot{u}_c}{\partial \omega_1} & \cdots & \frac{\partial \dot{u}_c}{\partial \omega_8} & \frac{\partial \dot{u}_c}{\partial \delta_1} & \cdots & \frac{\partial \dot{u}_c}{\partial \delta_4} & \frac{\partial \dot{u}_c}{\partial \phi} & \frac{\partial \dot{u}_c}{\partial \theta} \\ \frac{\partial \dot{v}_c}{\partial \omega_1} & \cdots & \frac{\partial \dot{v}_c}{\partial \omega_8} & \frac{\partial \dot{v}_c}{\partial \delta_1} & \cdots & \frac{\partial \dot{v}_c}{\partial \delta_4} & \frac{\partial \dot{v}_c}{\partial \phi} & \frac{\partial \dot{v}_c}{\partial \theta} \\ \frac{\partial \dot{w}_c}{\partial \omega_1} & \cdots & \frac{\partial \dot{w}_c}{\partial \omega_8} & \frac{\partial \dot{w}_c}{\partial \delta_1} & \cdots & \frac{\partial \dot{w}_c}{\partial \delta_4} & \frac{\partial \dot{w}_c}{\partial \phi} & \frac{\partial \dot{w}_c}{\partial \theta} \\ \frac{\partial \dot{p}}{\partial \omega_1} & \cdots & \frac{\partial \dot{p}}{\partial \omega_8} & \frac{\partial \dot{p}}{\partial \delta_1} & \cdots & \frac{\partial \dot{p}}{\partial \delta_4} & \frac{\partial \dot{p}}{\partial \phi} & \frac{\partial \dot{p}}{\partial \theta} \\ \frac{\partial \dot{q}}{\partial \omega_1} & \cdots & \frac{\partial \dot{q}}{\partial \omega_8} & \frac{\partial \dot{q}}{\partial \delta_1} & \cdots & \frac{\partial \dot{q}}{\partial \delta_4} & \frac{\partial \dot{q}}{\partial \phi} & \frac{\partial \dot{q}}{\partial \theta} \\ \frac{\partial \dot{r}}{\partial \omega_1} & \cdots & \frac{\partial \dot{r}}{\partial \omega_8} & \frac{\partial \dot{r}}{\partial \delta_1} & \cdots & \frac{\partial \dot{r}}{\partial \delta_4} & \frac{\partial \dot{r}}{\partial \phi} & \frac{\partial \dot{r}}{\partial \theta} \end{bmatrix}\Bigg|_{\boldsymbol{x}_0, \boldsymbol{u}_0} \tag{43}$$

In order to utilize this matrix with the RSPI algorithm, small elements on the $\boldsymbol{B}_0$ matrix are set to zero due to the fact that it's not desired to use inefficient actuators to control determined axes. As an example, the vertical motors are mounted with non-zero tilt angles, which means that they are physically capable of producing forces in the lateral axis without tilting the aircraft. Since this effect is minimal, is better to assume a zero effectiveness to avoid the allocation of them for this task. In light of this, the following simplified matrix is used:

$$\boldsymbol{B}_0 \approx \begin{bmatrix} 0 & \cdots & 0 & \frac{\partial \dot{u}_c}{\partial \omega_7} & \frac{\partial \dot{u}_c}{\partial \omega_8} & 0 & \cdots & 0 & 0 & \frac{\partial \dot{u}_c}{\partial \theta} \\ 0 & \cdots & 0 & 0 & 0 & 0 & \cdots & 0 & \frac{\partial \dot{v}_c}{\partial \phi} & 0 \\ \frac{\partial \dot{w}_c}{\partial \omega_1} & \cdots & \frac{\partial \dot{w}_c}{\partial \omega_6} & 0 & 0 & 0 & \cdots & 0 & 0 & \frac{\partial \dot{w}_c}{\partial \theta} \\ \frac{\partial \dot{p}}{\partial \omega_1} & \cdots & \frac{\partial \dot{p}}{\partial \omega_6} & 0 & 0 & \frac{\partial \dot{p}}{\partial \delta_1} & \cdots & \frac{\partial \dot{p}}{\partial \delta_4} & 0 & 0 \\ \frac{\partial \dot{q}}{\partial \omega_1} & \cdots & \frac{\partial \dot{q}}{\partial \omega_6} & 0 & 0 & \frac{\partial \dot{q}}{\partial \delta_1} & \cdots & \frac{\partial \dot{q}}{\partial \delta_4} & 0 & 0 \\ \frac{\partial \dot{r}}{\partial \omega_1} & \cdots & \frac{\partial \dot{r}}{\partial \omega_6} & \frac{\partial \dot{r}}{\partial \omega_7} & \frac{\partial \dot{r}}{\partial \omega_8} & 0 & \cdots & 0 & 0 & 0 \end{bmatrix}\Bigg|_{\boldsymbol{x}_0, \boldsymbol{u}_0} \tag{44}$$

The next sections details how the different elements in (44) were estimated. The formulation is presented with respect to the moments in Body frame (denoted by $\boldsymbol{M}_B$)

and forces in the Controller frame ($\boldsymbol{F}_C$), instead of the pseudo commands $\dot{\boldsymbol{\omega}}_B$ and $\dot{\boldsymbol{V}}_C$. This is done because these derivatives are generally described using forces and moments in the literature. From (2) and (3), it's easy to compute the relation between $\boldsymbol{F}_C$, $\boldsymbol{M}_B$ and $\dot{\boldsymbol{\omega}}_B$, $\dot{\boldsymbol{V}}_C$. That is,

$$\begin{aligned} \frac{\partial \dot{\boldsymbol{\omega}}_B}{\partial \boldsymbol{u}}\Big|_{\boldsymbol{x}_0,\boldsymbol{u}_0} &= \frac{\partial}{\partial \boldsymbol{u}}\left[\boldsymbol{I}_{BB}^{-1}\cdot \boldsymbol{M}_B(\boldsymbol{x},\boldsymbol{u})\right]\Big|_{\boldsymbol{x}_0,\boldsymbol{u}_0} \\ \frac{\partial \dot{\boldsymbol{\omega}}_B}{\partial \boldsymbol{u}}\Big|_{\boldsymbol{x}_0,\boldsymbol{u}_0} &= \boldsymbol{I}_{BB}^{-1}\cdot \frac{\partial \boldsymbol{M}_B}{\partial \boldsymbol{u}}\Big|_{\boldsymbol{x}_0,\ \boldsymbol{u}_0} \end{aligned} \tag{45}$$

$$\begin{aligned} \frac{\partial \dot{\boldsymbol{V}}_C}{\partial \boldsymbol{u}}\Big|_{\boldsymbol{x}_0,\boldsymbol{u}_0} &= \frac{\partial}{\partial \boldsymbol{u}}\left[\frac{1}{m}\cdot \boldsymbol{F}_C(\boldsymbol{x},\boldsymbol{u})\right]\Big|_{\boldsymbol{x}_0,\boldsymbol{u}_0} \\ \frac{\partial \dot{\boldsymbol{V}}_C}{\partial \boldsymbol{u}}\Big|_{\boldsymbol{x}_0,\boldsymbol{u}_0} &= \frac{1}{m}\cdot \frac{\partial \boldsymbol{F}_C}{\partial \boldsymbol{u}}\Big|_{\boldsymbol{x}_0,\ \boldsymbol{u}_0} \end{aligned} \tag{46}$$

the derivatives $\partial \boldsymbol{F}_C/\partial \boldsymbol{u}$ and $\partial \boldsymbol{M}_B/\partial \boldsymbol{u}$ are presented in the next section and then (45) and (46) are used to generate the $\boldsymbol{B}_0$ matrix.

3.7.1 Motors Derivatives

The propeller force and moment vector can be represented using the approximations:

$$\boldsymbol{F}_P(\omega_i) = \begin{bmatrix} 0 \\ 0 \\ -\omega_i^2 \cdot K_{T,i} \end{bmatrix},\ i = 1, 2, \cdots, 8 \tag{47}$$

$$\boldsymbol{M}_P(\omega_i) = \begin{bmatrix} 0 \\ 0 \\ -\omega_i^2 \cdot K_{Q,i} \cdot sign(\omega_i) \end{bmatrix},\ i = 1, 2, \cdots, 8 \tag{48}$$

where $K_{T,i}$ and $K_{Q,i}$ are positive constants and $\boldsymbol{F}_P$ and $\boldsymbol{M}_P$ are the propeller force and moment vectors, respectively, written in the Propeller frame. The inclusion of the term $sign(\omega_i)$ in (48) is to account for the propeller rotation direction. With those relations, the moments in the Body frame can be calculated by

$$\boldsymbol{M}_B(\omega_i) = \boldsymbol{R}_{BP,i}\cdot \begin{bmatrix} 0 \\ 0 \\ -\omega_i^2 \cdot K_{Q,i} \cdot sign(\omega_i) \end{bmatrix} + \boldsymbol{r}_{B,i} \times \boldsymbol{R}_{BP,i}\cdot \begin{bmatrix} 0 \\ 0 \\ -\omega_i^2 \cdot K_{T,i} \end{bmatrix} \tag{49}$$

where $\boldsymbol{R}_{BP,i}$ is a transformation matrix from the Propeller frame i to the Body frame and $\boldsymbol{r}_{B,i}$ is the position vector of the motor i with respect to the vehicle CG, denoted in

Body frame. Taking the derivative of (49) with respect to ω_i^2, it follows that

$$\frac{\partial \boldsymbol{M}_B}{\partial \omega_i^2} = \boldsymbol{R}_{BP} \cdot \begin{bmatrix} 0 \\ 0 \\ -K_{Q,i} \cdot sign(\omega_i) \end{bmatrix} + \boldsymbol{r}_B \times \boldsymbol{R}_{BP} \cdot \begin{bmatrix} 0 \\ 0 \\ -K_{T,i} \end{bmatrix} \tag{50}$$

The coefficients K_Q and K_T are assumed to be approximately constant during the whole flight envelope and, therefore, (50) is calculated offline and the result is stored to be used during the flight.

Note that (50) gives a linear relation with respect to the square of the rotor speed ω_i. This means that the control allocation actually calculates the rotors speed command squared. In practice, the proper conversions have to be made before using the results from the control allocation, such as taking the square root of the rotors output to get the actual commands not squared.

3.7.2 Control Surfaces Derivatives

The moments generated by the control surfaces are approximated using linear relations (ROSKAM, 2001):

$$\boldsymbol{M}_B(\delta_i) = \frac{1}{2} \cdot \rho \cdot V_{wind}^2 \cdot S_{ref} \cdot \delta_i \cdot \begin{bmatrix} C_{l\delta_i} \cdot B_{ref} \\ C_{m\delta_i} \cdot C_{ref} \\ C_{n\delta_i} \cdot B_{ref} \end{bmatrix}, \; i = 1, 2, 3, 4 \tag{51}$$

where B_{ref} is the reference wingspan, C_{ref} is the reference chord, and the coefficients $C_{l\delta_i}$, $C_{m\delta_i}$ $C_{n\delta_i}$ are the derivatives of the moments coefficients in the X, Y and Z axes, respectively, with respect to the deflection of the control surface δ_i. These coefficients can be easily calculated using low fidelity aerodynamic models or CFD (Computational Fluid Dynamics). The derivative of (51) with respect to the deflection δ_i is taken, obtaining

$$\frac{\partial \boldsymbol{M}_B}{\partial \delta_i} = \frac{1}{2} \cdot \rho \cdot V_{wind}^2 \cdot S_{ref} \cdot \begin{bmatrix} C_{l\delta_i} \cdot B_{ref} \\ C_{m\delta_i} \cdot C_{ref} \\ C_{n\delta_i} \cdot B_{ref} \end{bmatrix} \tag{52}$$

In practice, the kinematic velocity V_{kin} was used since airspeed measurements were not available.

3.7.3 Virtual Controls Derivatives

During hover flight the pitch and roll angle can be used to produce forward and lateral force, respectively. To get a relation between the forces and the aircraft tilt angles, the Z force produced by the propellers in the Body frame, depicted as Z_B is transformed

to the Controller frame using a rotation matrix. Mathematically,

$$\boldsymbol{F}_c(\phi,\theta)_{vtol} = \begin{bmatrix} \cos\theta & \sin\phi\cdot sin\theta & \cos\phi\cdot\sin\theta \\ 0 & \cos\phi & -\sin\phi \\ -\sin\theta & \sin\phi\cdot\cos\theta & \cos\phi\cdot\cos\theta \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ Z_B \end{bmatrix}$$

$$\boldsymbol{F}_c(\phi,\theta)_{vtol} = \begin{bmatrix} \cos\phi\cdot\sin\theta\cdot Z_B \\ -\sin\phi\cdot Z_B \\ \cos\phi\cdot\cos\theta\cdot Z_B \end{bmatrix} \tag{53}$$

The term Z_B is calculated from the estimated rotation speed of the motors as

$$\begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} = \sum_{i=1}^{8} \left(\boldsymbol{R}_{BP,i} \cdot \begin{bmatrix} 0 \\ 0 \\ -\omega_{i,est}\cdot K_{T,i} \end{bmatrix} \right) \tag{54}$$

Using small angle approximations, the following linear relation between the virtual controls and the pseudo controls is determined:

$$\boldsymbol{F}_C(\phi,\theta)_{vtol} \approx \begin{bmatrix} \theta\cdot Z_B \\ -\phi\cdot Z_B \\ Z_B \end{bmatrix} \tag{55}$$

During aeroplane mode, the lift from the wings is used to produce lateral forces by rolling the aircraft and upward forces by changing the angle of attack, thus increasing or decreasing the amount of lift. Assuming a small climb rate compared to the forward velocity, lift acts approximately on the $Y_c - Z_c$ plane, that is

$$\boldsymbol{F}_C(\phi)_{aero} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ -L \end{bmatrix}$$

$$\boldsymbol{F}_C(\phi)_{aero} = \begin{bmatrix} 0 \\ \sin\phi\cdot L \\ -\cos\phi\cdot L \end{bmatrix} \tag{56}$$

where L is the total lift. Using small angle approximations Equation (56) is further simplified as

$$\boldsymbol{F}_C(\phi)_{aero} \approx \begin{bmatrix} 0 \\ \phi\cdot L \\ -L \end{bmatrix} \tag{57}$$

To get a relation between the lift and the pitch angle, the lift is first written using a linear approximation, as described by Roskam (2001):

$$L = 0.5 \cdot \rho \cdot V_{wind}^2 \cdot S_{ref} \cdot (CL_0 + CL_\alpha \cdot \alpha) \tag{58}$$

where CL_0 is the lift coefficient at zero angle of attack and CL_α is the derivative of the lift coefficient with respect to the angle of attack. (58) is then written with respect to a steady state angle of attack α_0:

$$L = 0.5 \cdot \rho \cdot V_{wind}^2 \cdot S_{ref} \cdot (CL_0 + CL_\alpha \cdot (\alpha_0 + \Delta\alpha)) \tag{59}$$

Considering that changes in the pitch angle are achieved considerably fast, it's assumed that $\Delta\theta \approx \Delta\alpha$. With this assumption, the lift produced by the wing is controlled by changes in the pitch angle:

$$L \approx 0.5 \cdot \rho \cdot V_{wind}^2 \cdot S_{ref} \cdot (CL_0 + CL_\alpha \cdot (\alpha_0 + \Delta\theta)) \tag{60}$$

Taking the derivative with respect to θ:

$$\left.\frac{\partial L}{\partial \theta}\right|_{\theta_0} \approx 0.5 \cdot \rho \cdot V_{wind}^2 \cdot S_{ref} \cdot CL_\alpha \tag{61}$$

Finally, combining (55), (57) and (61), the derivative of the forces with respect to θ are given by:

$$\frac{\partial \boldsymbol{F}_c}{\partial \theta} = \begin{bmatrix} Z_B \\ 0 \\ -0.5 \cdot \rho \cdot V_{wind}^2 \cdot S_{ref} \cdot CL_\alpha \end{bmatrix} \tag{62}$$

In the same manner, the derivative with respect to ϕ is obtained:

$$\frac{\partial \boldsymbol{F}_c}{\partial \phi} = \begin{bmatrix} 0 \\ -Z_B + L \\ 0 \end{bmatrix} \tag{63}$$

Assuming that the force from the propellers and the lift from the wings is approximately always balancing the aircraft, that is $-Z_B + L \approx m \cdot g$, Equation (63) is further simplified as

$$\frac{\partial \boldsymbol{F}_c}{\partial \phi} = \begin{bmatrix} 0 \\ m \cdot g \\ 0 \end{bmatrix} \tag{64}$$

4 SIMULATION RESULTS

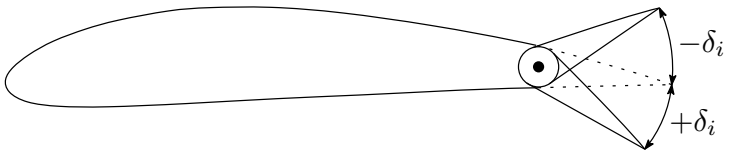
This chapter shows simulation results obtained by implementing the control strategy of Chapter 3 on the aircraft described in Chapter 2. A high-fidelity six degree of freedom model of the aircraft is adopted for the simulations. The model was developed in another project at Volocopter GmbH, and it uses full aerodynamic polars, that is, the coefficients were calculated within the range ± 180 degrees angle of attack and ± 90 degrees sideslip, such that hover maneuvers can be simulated with the proper aerodynamic effects. For more details about flight dynamics, the reader is referred to Roskam (2001). The aerodynamic coefficients were obtained from CFD simulations and the propellers coefficients were obtained from wind tunnel testing.

For better readability, the motors commands in the graphs are normalized using their maximum rotation speed, that is

$$\omega_{i,norm} = \frac{\omega_i}{\omega_{i,max}},\ i = 1, 2, \cdots, 8$$

where $\omega_{i,max}$ depicts the maximum rotation speed of the i-th motor. The deflection of the control surfaces are given in degrees and all the surfaces are constrained to ± 20 degrees. The positive and negative convention are illustrated in Figure 23.

Figure 23 – A downwards deflection (increasing the lift) is considered positive.



Source: Author.

Each of the following sections presents different pilot commands to show different aspects of the developed control strategy. As such, transition and back-transition (aeroplane to VTOL) scenarios are presented to show the controller transition behavior. Additionally, a scenario in VTOL mode is presented to show the system performance during hover. At last, a scenario in aeroplane mode is presented to show the action of the turn coordinator controller.

The controller parameters are shown in Tables 1, 2 and 3.

Table 1 – Error controllers Parameters.

Parameter	Value	Unit
$K_{p,\ u}$	2.0	-
$K_{p,\ v}$	2.0	-
$K_{p,\ w}$	10.0	-
$K_{p,\ \phi}$	12.0	-
$K_{d,\ \phi}$	6.0	-
$K_{p,\ \theta}$	8.0	-
$K_{d,\ \theta}$	4.5	-
$K_{p,\ \dot{\psi}}$	2.2	-
$K_{p,\ \beta}$	$6.5 \cdot 10^{-2}$	-

Source: Author.

Table 2 – Referece models parameters.

Parameter	Value	Unit
ω_u	0.5	rad/s
ω_v	0.5	rad/s
ω_w	1.5	rad/s
ω_ϕ	4.2	rad/s
ξ_ϕ	1.0	-
ω_θ	3.5	rad/s
ξ_θ	1.0	-
$\omega_{\dot{\psi}}$	4.2	rad/s

Source: Author.

Table 3 – Other parameters.

Parameter	Value	Unit
V_{vtol}	5	m/s
V_{aero}	23	m/s
Maximum pitch angle command	$\pm$ 20	$\circ$
Maximum roll angle command	$\pm$ 30	$\circ$

Source: Author.

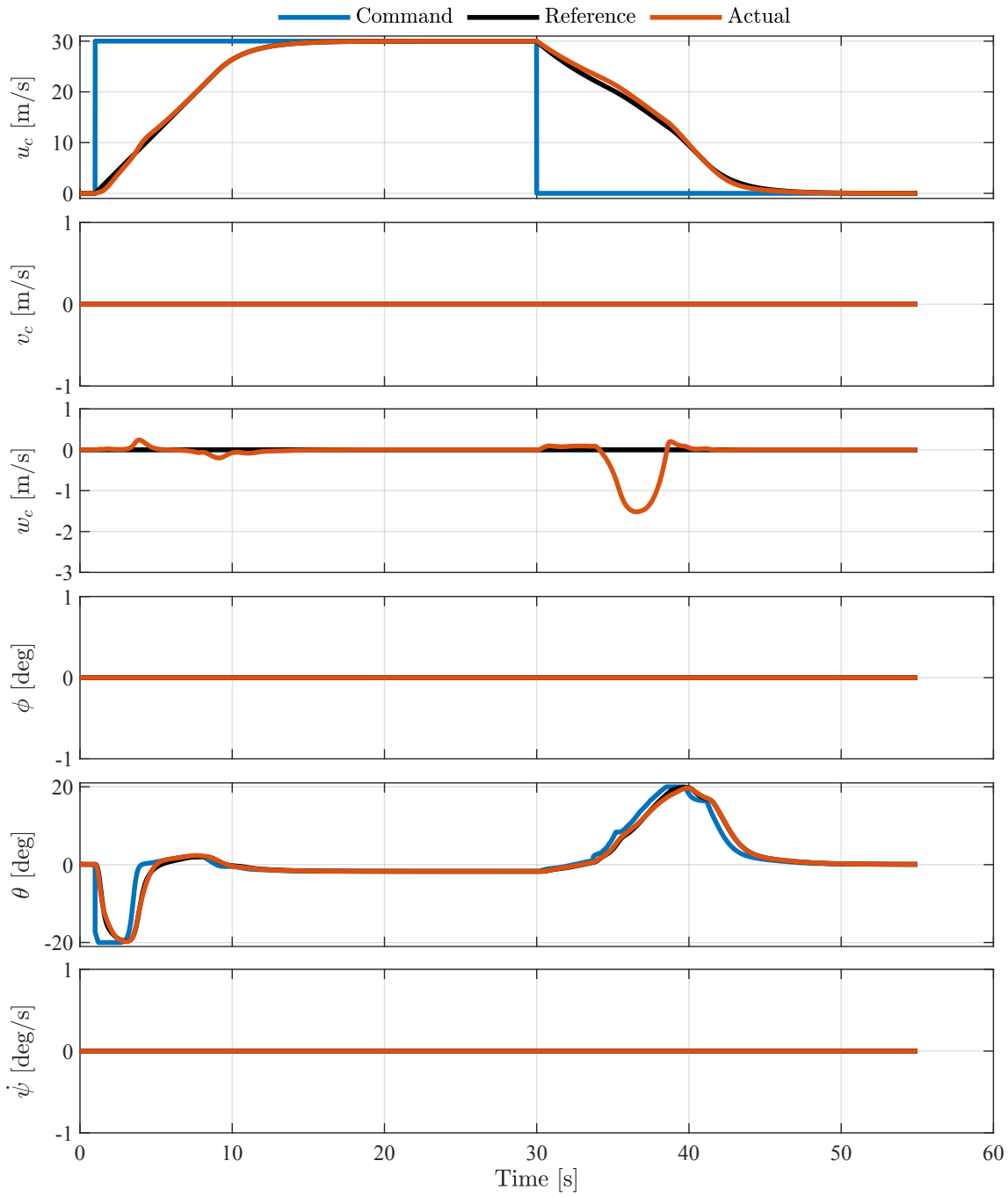
4.1 TRANSITION AND BACK-TRANSITION

Figure 24 shows the commanded and actual states of the controlled variables during a transition and back-transition commands. In this scenario, the pilot command is comprised only by the forward velocity u_C, while the actuators commands, shown in Figure 25, and the pitch angle are commanded by the control strategy.

It can be noticed that the controlled variables reached the reference without steady state errors. During the back-transition (around 35 seconds), the aircraft starts to climb, causing the major deviation from the reference vertical velocity w_C. This happens because, as the speed decreases and the control allocation transitions back to its VTOL part, the pitch angle is allocated without taking into account the lift produced. As a result, a high pitch angle, intended to reduce the aircraft forward velocity, increases the lift and the aircraft climbs.

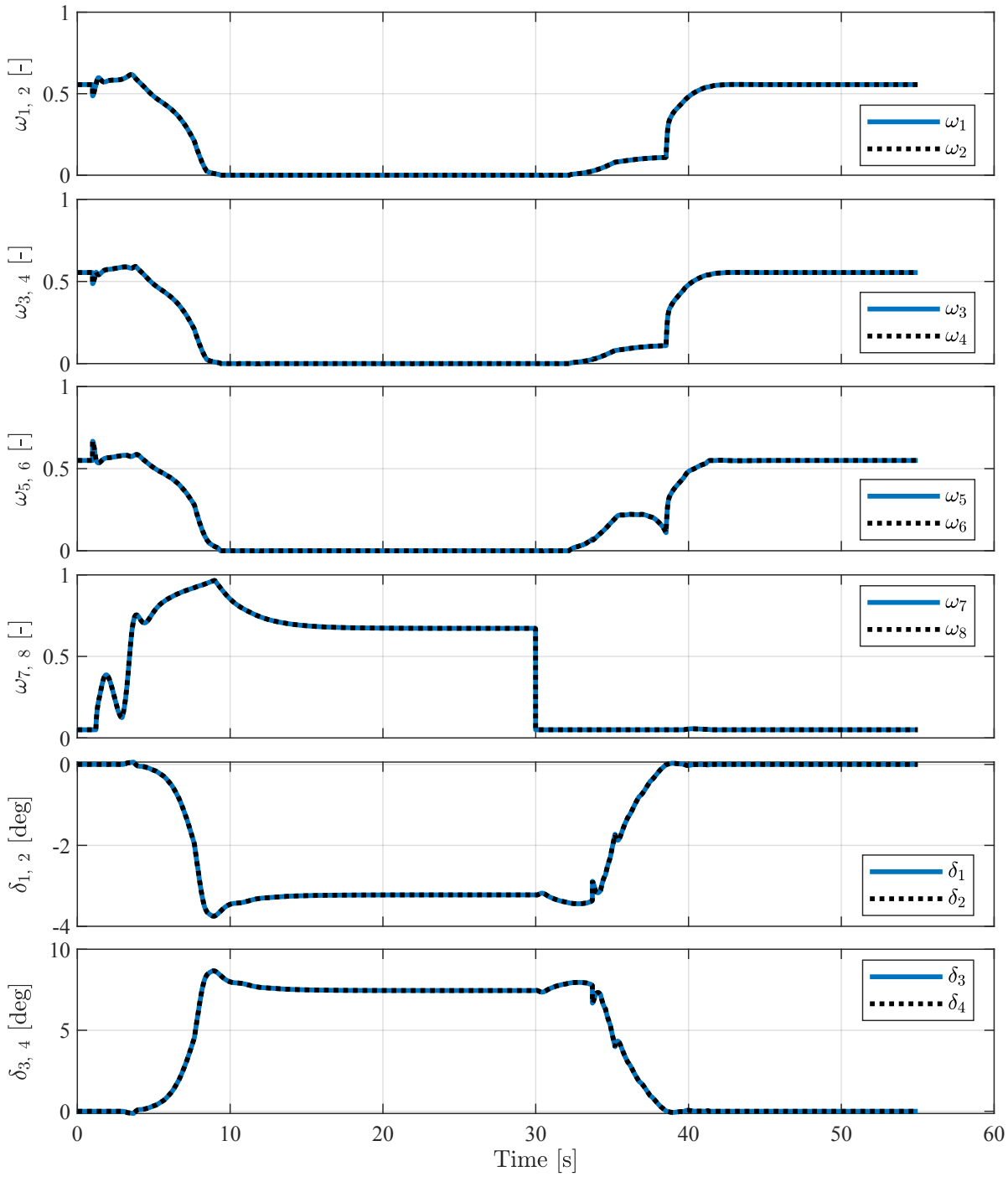
In Figure 25, it can be verified that the aircraft successfully completed the transition by turning off the vertical motors and using the control surfaces. Another interesting thing to notice is that both the pitch angle and the horizontal motors are commanded during the initial acceleration of the aircraft. As explained in Section 3.6.3, the aircraft should preferentially only tilt during low speeds, but this is not a hard constraint. As a result, high acceleration demands from the velocity controller caused the horizontal motors to act in order to produce more forward acceleration.

Figure 24 – States during transition and back-transition maneuver. At around 35 seconds, there is a big error in the climbing rate (w_C) due to the high pitch angle command demanded to break. As shown in Figure 21, the pitch angle allocation in VTOL mode does not take into account the lift generated by pitching, leading to this effect.



Source: Author.

Figure 25 – Actuators during transition and back-transition maneuver. The same effect mentioned earlier can be noticed in the actuators commands. When the aircraft starts to climb due to the amount of lift, the vertical motors reach the idle speed and only after the speed decreases, they are commanded to hover speed.



Source: Author.

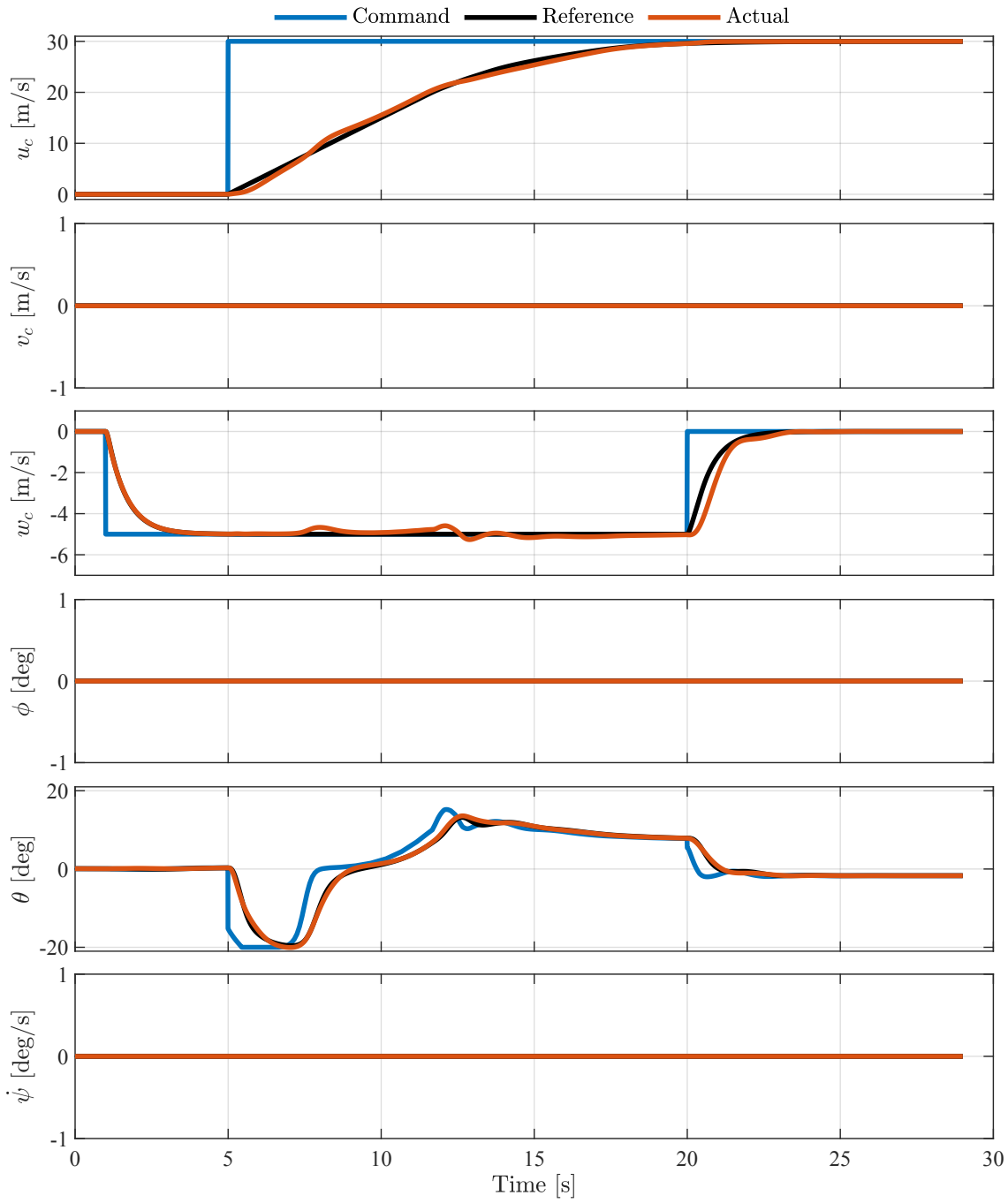
4.2 CLIMB-TRANSITION

In this scenario, pilot commands a forward speed u_C and a climb rate w_C, demanding the aircraft has to transition while maintaining the climb velocity. Figure 26 shows the controller tracking performance, where it can be noticed the tracking of both, the forward and vertical velocity commands without steady state errors. In this graph, it can be also noticed small oscillations in the vertical velocity w_C, caused by a variety of factors, such as aerodynamic effects, non-ideal actuators and mismatches between the actual matrix $\boldsymbol{B}_0$ and the estimated.

An interesting point to notice is the tracking performance of the vertical velocity commands at 2 seconds and 20 seconds. The first is commanded during VTOL mode, which means the vertical acceleration is controlled by means of the vertical motors, while the second command is given during aeroplane mode, in which the vertical acceleration is controlled by changing the pitch angle. Since the vertical motors have a much quicker dynamics compared to the pitch angle reference model, the tracking is better during VTOL mode.

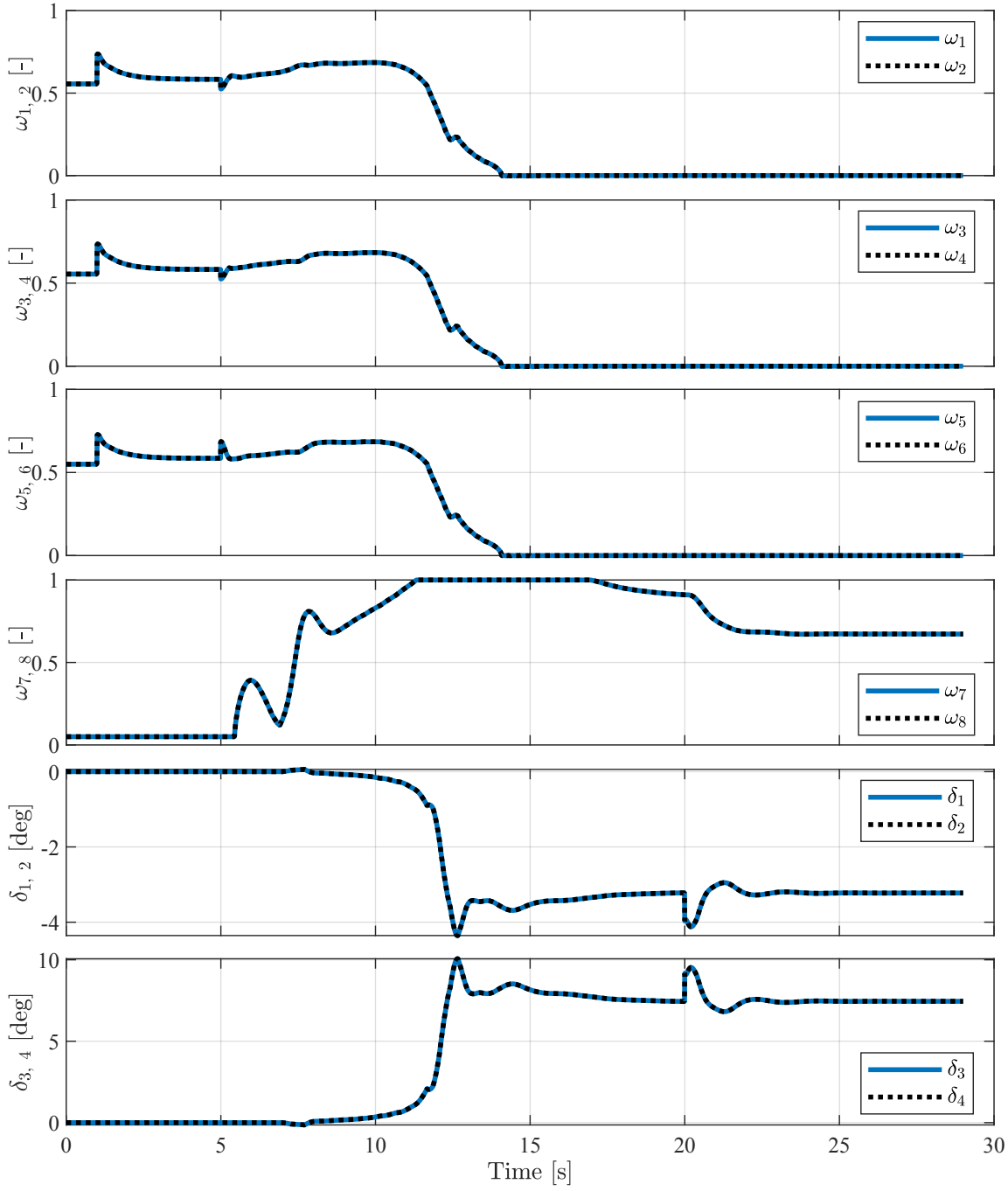
Figure 27 shows the actuators commands, where it can be verified the full transition at around 15 seconds, when the vertical motors are turned off. It is worth noting that during the forward acceleration of the aircraft (5-10 seconds), the command to the vertical motors increase due to the negative angle of attack, caused by the combination of the climb velocity and the commanded pitch angle.

Figure 26 – States during climb-transition maneuver. The order of the commands were chosen in a way to emulate the expected operation of a transition vehicle: first the aircraft is commanded to climb vertically to get some minimum ground and obstacle clearance, then a forward speed command is given while keeping the climbing rate until the cruise altitude is reached.



Source: Author.

Figure 27 – Actuators during climb-transition maneuver. Notice that, although the aircraft is pitching up during the transition, the control surfaces generate a pitch down moment to compensate for the inherent aerodynamic moment of the aircraft.



Source: Author.

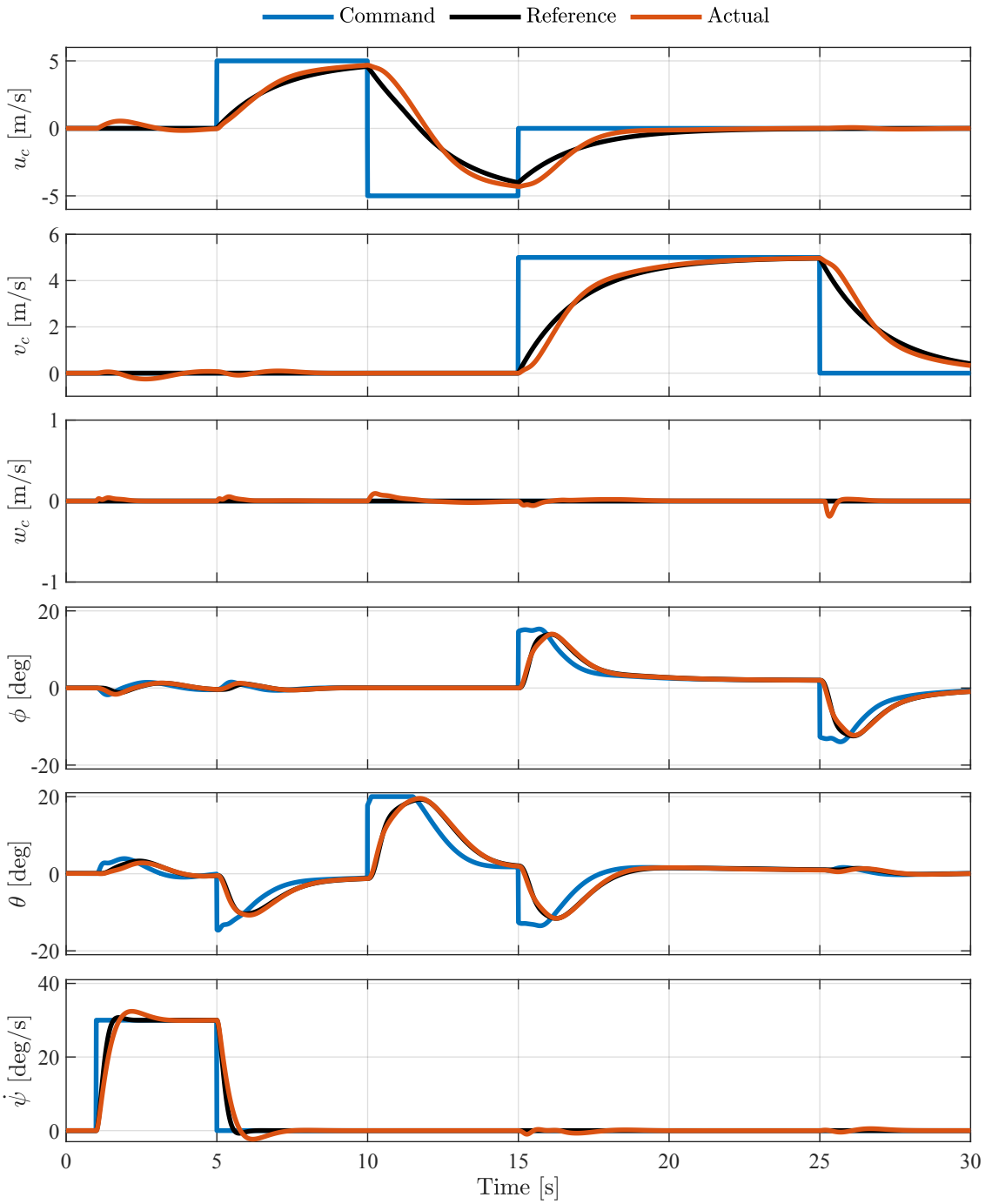
4.3 HOVER MANEUVERS

Figure 28 depicts the controller's performance only in VTOL mode. The pilot commands consists of slow speed forward and lateral velocities, u_C and v_C, respectively, and heading rate commands. The forward command is given to a maximum of 5 m/s, which assures the aircraft does not start to transition.

The pitch and roll angle are generated by the control strategy, more specifically, by the control allocation framework. As explained in Section 3.6.3, the aircraft preferentially tilts in order to control its forward speed in low speed flight. This behavior can be noticed by the pitch angle commands and by checking the actuators commands in Figure 29, where the horizontal motors are only commanded diferentially in order to produce yawing moments.

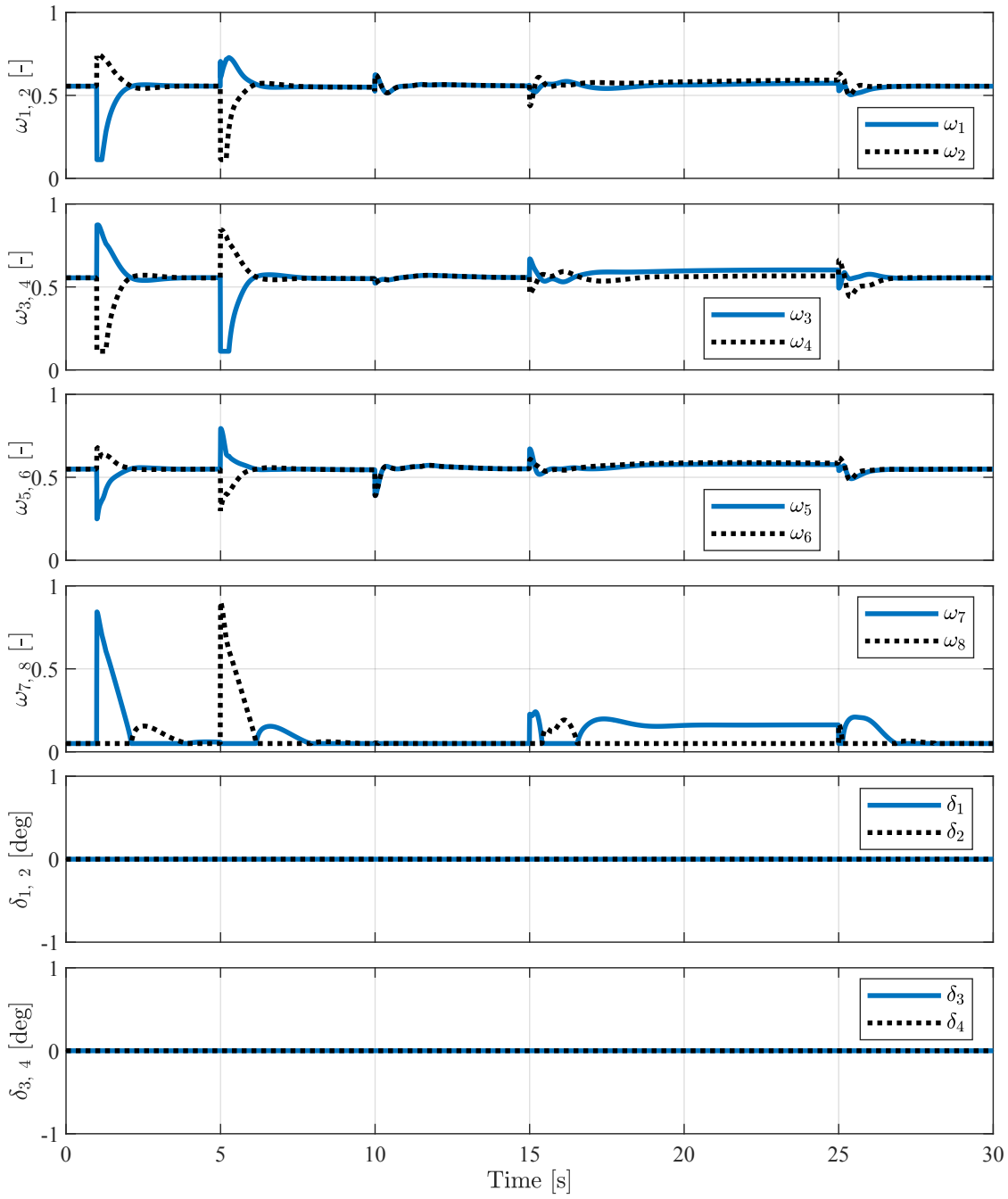
In Figure 28, it can be noticed deviations from the reference forward and lateral velocities. This happens because they are controlled by changes in the pitch and roll angle, and since they have slow reference models, the performance is degraded. Besides that, the commanded velocities are tracked without steady state errors.

Figure 28 – States during hover maneuvers. In this simulation low-speed and yaw commands are given to emulate commands during a VTOL operation. Note that, as explained in Section 3.6.4, the aircraft preferentially tilts to control its forward speed during low speed flight.



Source: Author.

Figure 29 – Actuators during hover maneuvers. Note that since the aircraft preferentially tilts to control its forward speed in VTOL, the horizontal motors are used for yaw control, which can be noticed by the differential commands.



Source: Author.

4.4 AEROPLANE TURN

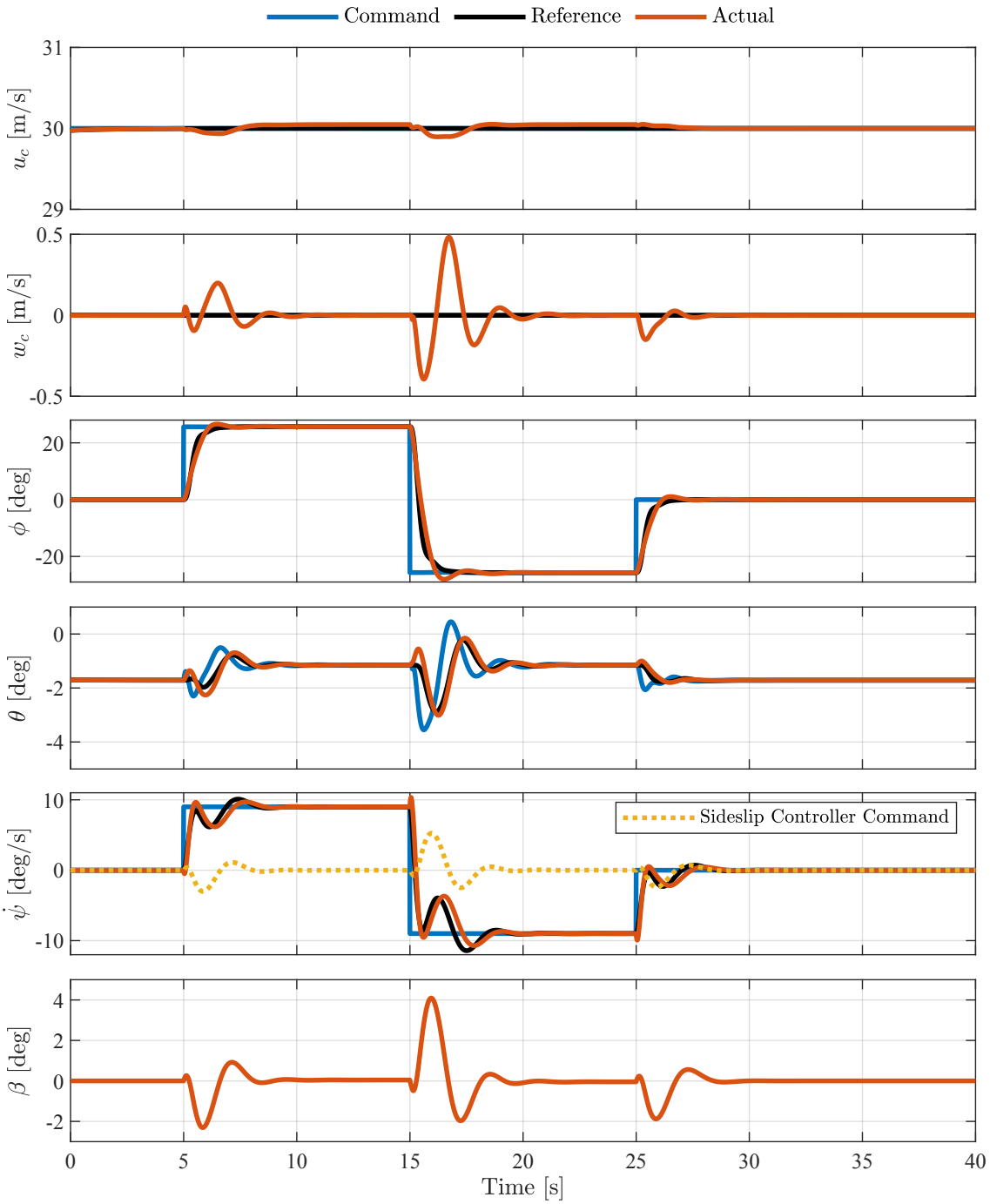
In this last scenario, the aircraft is initially in aeroplane mode and a heading rate command $\dot{\psi}_{turn}$ is given. The states are shown in Figure 30, where the roll angle in this phase is purely produced by the turn coordinator. Because the lateral velocity v_C is not controlled during aeroplane mode, the sideslip angle is plotted to show the action of the sideslip controller.

The heading rate commands are tracked without steady state errors by means of a combination of roll and yaw commands, as explained in Section 3.5. Furthermore, the sideslip controller, which uses feedback from the lateral specific forces, is capable of vanishing the sideslip angle during the turns.

During the initial moments of the turn commands, deviations in the vertical velocity can be noticed due to the fact that the control allocation does not take into account the inclination of the lift vector when rolling. To perform a level turn, the aircraft has to not only roll, but also increase its angle of attack to compensate for the inclination of the lift vector. With the simplifications made in Section 3.7, the control allocation only commands a roll angle, rather than a combination of roll and pitch. As a result, the aircraft has to first deviate from the reference trajectory so that the controller can command an increase in the pitch angle.

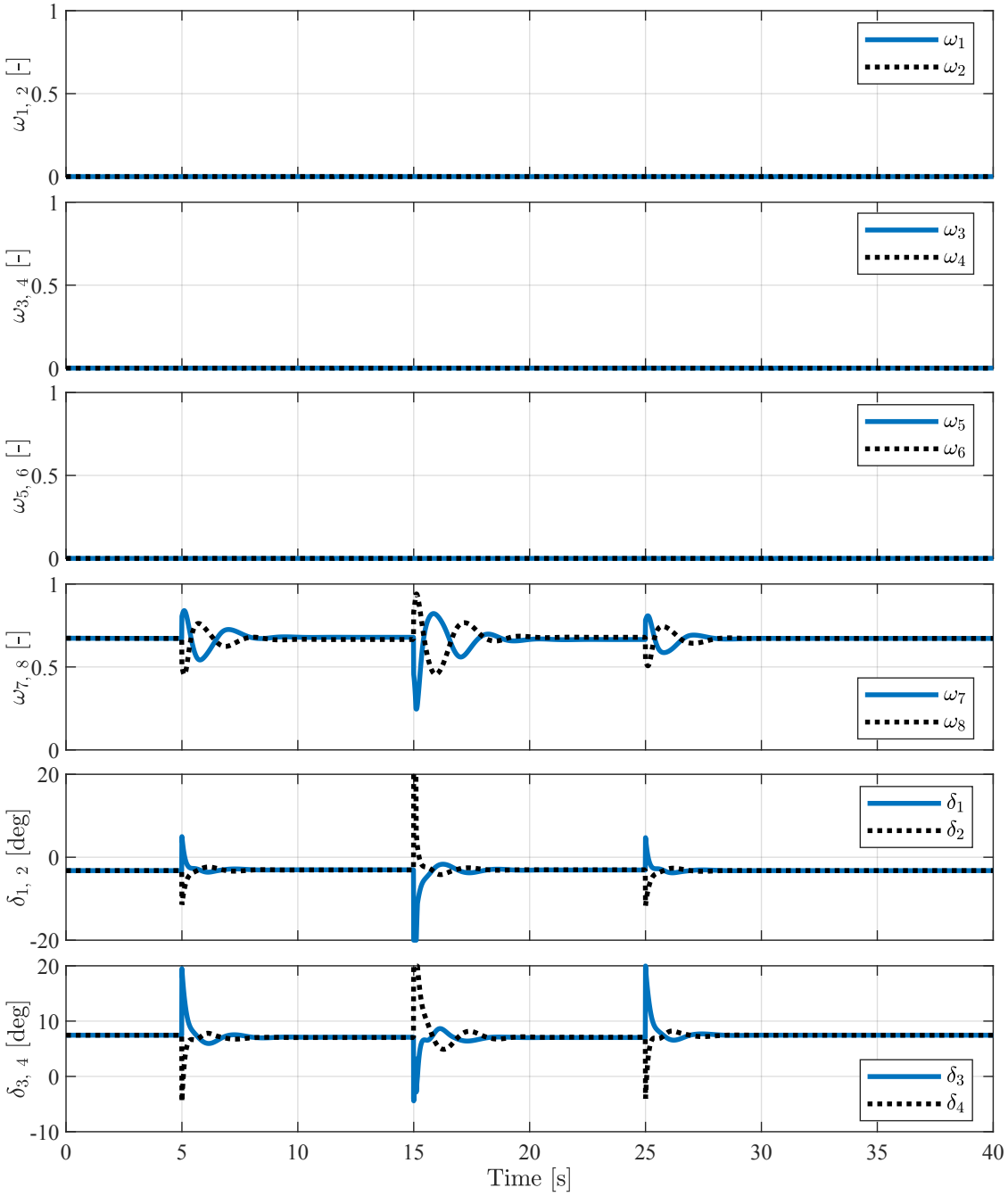
The actuators are shown in Figure 31. Notice that the horizontal motors are used to produce yawing moments during the turns. Furthermore, the control surfaces return to their initial trimmed position after the turn maneuvers, even after reaching saturation at 15 seconds, showing that the technique employed can indeed solve the path dependency problem.

Figure 30 – States during aeroplane turn. In this simulation, the aircraft is initially in Aeroplane mode and a turn command ($\dot{\psi}_{turn}$) is given. The roll commands are purely generated by the turn compensator, as explained in Section 3.5. The sideslip angle (β) is plotted to show the sideslip controller action.



Source: Author.

Figure 31 – Actuators during aeroplane turn. The horizontal motors are being commanded differentially to yaw the aircraft in the direction of the turn during the turn commands. Another interesting thing to notice is that the control surfaces return to their initial trimmed positions after the maneuvers, showing that the control law (14) is not path dependent.



Source: Author.

5 CONCLUSIONS

With the increasing interest and investment in AAM, several VTOL transitioning aircraft are in development. Those aircraft are capable of performing VTOL to operate in urban areas and wingborne flight during cruise, which increases the range performance compared to multi-rotor aircraft.

In this work, a unified control strategy was proposed for a VTOL transitioning aircraft that enables a continuous transition between the two flight modes. An INDI attitude and velocity controllers were developed, which calculates desired angular and linear accelerations, the pseudo controls. The pseudo controls are transformed into the real control vector using a control allocation framework. The control strategy was validated through simulations to show the transition and back-transition between the two flight modes as well as the tracking performance in pure hover and aeroplane modes.

The transition from hover to aeroplane showed that the controller successfully tracked the reference commands and transitioned the actuators commands from the vertical motors to the control surfaces. During the back transition, a high pitch angle command, meant to reduce the forward velocity, caused a considerable deviation in the vertical velocity due to the lift generated by the high angle of attack. This shows that a compromise has to be made between the tracking performance of the forward velocity and the vertical velocity during the back-transition, since a high pitch angle command will reduce the aircraft velocity faster, but at the cost of causing the aircraft to climb.

The hover maneuvers showed good tracking of the commands at low speeds, but a relative worst tracking of the forward and lateral velocities can be observed due to the slower dynamics of the virtual controls. This suggests that the assumptions used to derive the INDI may not be negligible in this case. The last simulated maneuver showed a turn in aeroplane mode, where the tracking of the heading rate commands can be observed as a combination with the roll angle commands. Moreover, the sideslip angle was vanished by the action of the sideslip controller.

For future work, the effect of the virtual controls dynamics could be explored, as well as other techniques to take into account the actuators dynamics, such as the Extended INDI (RAAB et al., 2019). Another point of action is the back-transition dynamics, where a new VTOL control allocation architecture could be designed to avoid the command of high pitch angles and improve the tracking of the vertical velocity. Finally, since the simulations never capture all real-world effects, flight tests must be performed to validate the system behavior.

REFERENCES

BAUERSFELD, Leonard; SPANNAGL, Lukas; DUCARD, Guillaume J. J.; ONDER, Christopher H. MPC Flight Control for a Tilt-Rotor VTOL Aircraft. **IEEE Transactions on Aerospace and Electronic Systems**, v. 57, n. 4, p. 2395–2409, 2021. eprint: https://doi.org/10.1109/TAES.2021.3061819.

DURHAM, Wayne; BORDIGNON, Kenneth A.; BECK, Roger. **Aircraft Control Allocation**. First. [S.l.]: John Wiley & Sons, Ltd., 2017.

HARTMANN, Philipp; MEYER, Carsten; MOORMANN, Dieter. Unified Velocity Control and Flight State Transition of Unmanned Tilt-Wing Aircraft. **Journal of Guidance, Control, and Dynamics**, v. 40, n. 6, p. 1348–1359, 2017. eprint: https://doi.org/10.2514/1.G002168.

KRIEL, Steven Cornelius. **A Comparison of Control Systems for the Fligth Transition of VTOL Unmanned Aerial Vehicles**. Mar. 2008. MA thesis – University of Stellenbosch. Faculty of Engineering. Dept. of Electrical and Electronic Engineering. eprint: http://hdl.handle.net/10019.1/1535.

LIMA, Elon Lages. **Álgebra linear**. [S.l.]: IMPA, 2014. (Coleção matemática universitária). ISBN 9788524400896.

MISB, Motion Imagery Standards Board. **MISB Standard 0601 - UAS Datalink Local Set**. [S.l.], Oct. 2014. Available from: https://upload.wikimedia.org/wikipedia/commons/1/19/MISB_Standard_0601.pdf.

NASA. **Advanced Air Mobility**. June 2022. Available from: https://www.nasa.gov/aam/overview/. Visited on: 27 May 2022.

OPPENHEIMER, Michael W.; DOMAN, David B.; BOLENDER, Michael A. Control Allocation for Over-actuated Systems. In: 2006 14th Mediterranean Conference on Control and Automation. [S.l.: s.n.], 2006. P. 1–6. eprint: https://doi.org/10.1109/MED.2006.328750.

ÖZNALBANT, Zafer; KAVSAOĞLU, Mehmet Ş. Flight control and flight experiments of a tilt-propeller VTOL UAV. **Transactions of the Institute of Measurement and Control**, v. 40, n. 8, p. 2454–2465, 2018. eprint: https://doi.org/10.1177/0142331218754618.

RAAB, Stefan A.; ZHANG, Jiannan; BHARDWAJ, Pranav; HOLZAPFEL, Florian. Consideration of Control Effector Dynamics and Saturations in an Extended INDI Approach. In: AIAA Aviation 2019 Forum. [S.l.: s.n.], 14 June 2019. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2019-3267.

RAAB, Stefan A.; ZHANG, Jiannan; BHARDWAJ, Pranav; HOLZAPFEL, Florian. Proposal of a Unified Control Strategy for Vertical Take-off and Landing Transition Aircraft Configurations. In: 2018 Applied Aerodynamics Conference. [S.l.: s.n.], 2018. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2018-3478.

RAMESH, Ashwath. **Augmentation of an Incremental Non-Linear Dynamic Inversion Controller with a Modified Linear Extended State Observer**. Jan. 2021. MA thesis – Technische Universität München.

ROSKAM, Jan. **Airplane Flight Dynamics and Automatic Flight Controls**. Third. [S.l.]: Design, Analysis and Research Corporation (DARcorporation), 2001. v. 1.

SMEUR, E.J.J. **Incremental Control of Hybrid Micro Air Vehicles**. Nov. 2018. MA thesis – Delft University of Technology. eprint: https://doi.org/10.4233/uuid:23c338a1-8b34-40a6-89e9-997adbdafd75.

SMEUR, E.J.J.; DE CROON, G.C.H.E.; CHU, Q. Cascaded incremental nonlinear dynamic inversion for MAV disturbance rejection. **Control Engineering Practice**, v. 73, p. 79–90, 2018. ISSN 0967-0661. eprint: https://www.sciencedirect.com/science/article/pii/S0967066118300030.

STEPHAN, Johannes; FICHTER, Walter. Fast Exact Redistributed Pseudoinverse Method for Linear Actuation Systems. **IEEE Transactions on Control Systems Technology**, v. 27, n. 1, p. 451–458, 2019. eprint: https://doi.org/10.1109/TCST.2017.2765622.

STRAUBINGER, Anna; ROTHFELD, Raoul; SHAMIYEH, Michael; BÜCHTER, Kai-Daniel; KAISER, Jochen; PLÖTNER, Kay Olaf. An overview of current research and developments in urban air mobility – Setting the scene for UAM introduction. **Journal of Air Transport Management**, v. 87, p. 101852, 2020. ISSN 0969-6997. eprint: https://doi.org/10.1016/j.jairtraman.2020.101852.

VELD, R.C. van't. **Incremental Nonlinear Dynamic Inversion Flight Control: Stability and Robustness Analysis and Improvements**. Oct. 2016. MA thesis –

Delft University of Technology. eprint: http://resolver.tudelft.nl/uuid:f85a9c88-7bdb-42cd-a01c-aa85251d365c.

YEO, Yih Tang; LIU, Hugh H. Transition Control of a Tilt-Rotor VTOL UAV. In: 2018 AIAA Guidance, Navigation, and Control Conference. [S.l.: s.n.]. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2018-1848.

ZHANG, Jiannan; BHARDWAJ, Pranav; RAAB, Stefan A.; SABOO, Saurabh; HOLZAPFEL, Florian. Control Allocation Framework for a Tilt-rotor Vertical Take-off and Landing Transition Aircraft Configuration. In: 2018 Applied Aerodynamics Conference. [S.l.: s.n.], 2018. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2018-3480.

APPENDIX A – INDI implicit integrating behavior

When considering the transfer function from the tracking error $\boldsymbol{e}$ to the control input $\boldsymbol{u}$, the INDI control law implicitly specifies an integrating behavior, and thus the error controller in INDI does not require an integrating element. (RAMESH, 2021). For simplicity, a SISO system is considered, where y is the system output and u is the controlled variable. Considering the control law

$$u(s) = u_0(s) + B_0^{-1} \cdot (\nu_{cmd}(s) - \nu_0(s)) \tag{A.1}$$

where $\nu_0(s) = y^{(r)}(s)$. Considering that the actuator transfer function from the input u to its actual state u_0 is given by

$$G(s)_{u_0,\, u} = \frac{\omega}{s+\omega} \tag{A.2}$$

equation (A.1) is rewritten using (A.2), obtaining

$$\begin{aligned} u(s) &= \frac{\omega}{s+\omega} \cdot u(s) + B_0^{-1} \cdot (\nu_{cmd}(s) - \nu_0(s)) \\ \left(1 - \frac{\omega}{s+\omega}\right) \cdot u(s) &= B_0^{-1} \cdot (\nu_{cmd}(s) - \nu_0(s)) \\ \frac{s}{s+\omega} \cdot u(s) &= B_0^{-1} \cdot (\nu_{cmd}(s) - \nu_0(s)) \\ \frac{s}{s+\omega} \cdot u(s) &= B_0^{-1} \cdot (\nu_{ref}(s) + \nu_{ec}(s) - \nu_0(s)) \end{aligned} \tag{A.3}$$

The error controller part (ν_{ec}) is given by

$$\nu_{ec}(s) = \underbrace{\begin{bmatrix} K_{p,y} & K_{p,\dot{y}} & \cdots & K_{p,y^{(r-1)}} \end{bmatrix}}_{\boldsymbol{K}_e} \cdot \underbrace{\begin{bmatrix} y_{ref}(s) - y(s) \\ \dot{y}_{ref}(s) - \dot{y}(s) \\ \vdots \\ y_{ref}^{(r-1)}(s) - y^{(r-1)}(s) \end{bmatrix}}_{\boldsymbol{e}(s)} \tag{A.4}$$

Combining (A.4) and (A.3), the following is obtained:

$$\begin{aligned} u(s) &= \frac{s+\omega}{s} \cdot B_0^{-1} \cdot (\nu_{ref}(s) - \nu_0(s) + \boldsymbol{K}_e \cdot \boldsymbol{e}(s)) \\ u(s) &= \frac{s+\omega}{s} \cdot B_0^{-1} \cdot (\nu_{ref}(s) - \nu_0(s)) + \frac{s+\omega}{s} \cdot B_0^{-1} \cdot \boldsymbol{K}_e \cdot \boldsymbol{e}(s) \\ u(s) &= \frac{s+\omega}{s} \cdot B_0^{-1} \cdot (\nu_{ref}(s) - \nu_0(s)) + B_0^{-1} \cdot (\boldsymbol{K}_e \cdot \boldsymbol{e}(s) + \boxed{\frac{\omega}{s} \cdot \boldsymbol{K}_e \cdot \boldsymbol{e}(s)}) \end{aligned} \tag{A.5}$$

where the highlighted term provides the integrating behavior.

Two things have to be noticed in this derivation. First, it considers that there is no bias in the actuators. If measurements from the actuators are used, then a integral term is required to compensate for measurements bias, as observed by Veld (2016). Secondly, a bias in the measurement of ν_0 will also lead to a steady state error, as explained by E.J.J. Smeur, de Croon, and Chu (2018).

APPENDIX B – Proposed non-path dependent INDI

Some properties of linear transformations are first introduced to demonstrate that the control law (14) is not path dependent. Firstly, consider the linear transformation between the real actuator inputs $\boldsymbol{u}$ and the pseudo controls $\boldsymbol{\nu}$

$$\boldsymbol{B}_0 : \boldsymbol{u} \rightarrow \boldsymbol{\nu} \tag{B.1}$$

where $\boldsymbol{B}_0 \in \mathbb{R}^{m \times n}$, $\boldsymbol{u} \in \mathbb{R}^n$ and $\boldsymbol{\nu} \in \mathbb{R}^m$. Let $\mathcal{N}(\boldsymbol{B}_0)$ represent the nullspace and $\mathcal{I}m(\boldsymbol{B}_0)$ the image of $\boldsymbol{B}_0$. If $\mathcal{G}_{\mathcal{N}} = \{\boldsymbol{u}_{1,\mathcal{N}}, \boldsymbol{u}_{2,\mathcal{N}}, \cdots, \boldsymbol{u}_{n-m,\mathcal{N}}\}$[1] is a basis of $\mathcal{N}(\boldsymbol{B}_0)$ and $\mathcal{G}_{\mathcal{I}m} = \{\boldsymbol{B}_0 \cdot \boldsymbol{u}_{1,\mathcal{I}m}, \boldsymbol{B}_0 \cdot \boldsymbol{u}_{2,\mathcal{I}m}, \cdots, \boldsymbol{B}_0 \cdot \boldsymbol{u}_{m,\mathcal{I}m}\}$ is a basis of $\mathcal{I}m(\boldsymbol{B}_0)$, then $\{\mathcal{G}_{\mathcal{N}}, \boldsymbol{u}_{1,\mathcal{I}m}, \boldsymbol{u}_{2,\mathcal{I}m}, \cdots, \cdot\boldsymbol{u}_{m,\mathcal{I}m}\}$ is a basis of the space vector of $\boldsymbol{u}$ (LIMA, 2014). The important implication for the following demonstration is that any vector $\boldsymbol{u}$ can be written as a sum of linear independent vectors in the nullspace and image of $\boldsymbol{B}_0$, that is

$$\begin{aligned} \boldsymbol{u} =& \alpha_1 \cdot \boldsymbol{u}_{1,\mathcal{N}} + \alpha_2 \cdot \boldsymbol{u}_{2,\mathcal{N}} + \cdots + \alpha_{n-m} \cdot \boldsymbol{u}_{n-m,\mathcal{N}} \\ &+ \beta_1 \cdot \boldsymbol{u}_{1,\mathcal{I}m} + \beta_2 \cdot \boldsymbol{u}_{2,\mathcal{I}m} + \cdots + \beta_m \cdot \boldsymbol{u}_{m,\mathcal{I}m} \end{aligned} \tag{B.2}$$

where

$$\boldsymbol{B}_0 \cdot (\alpha_1 \cdot \boldsymbol{u}_{1,\mathcal{N}} + \alpha_2 \cdot \boldsymbol{u}_{2,\mathcal{N}} + \cdots + \alpha_{n-m} \cdot \boldsymbol{u}_{n-m,\mathcal{N}}) = \boldsymbol{0} \tag{B.3}$$

for all $\alpha_1, \alpha_2, \cdots, \alpha_{n-m}$, since $\mathcal{G}_{\mathcal{N}}$ is a basis of $\mathcal{N}(\boldsymbol{B}_0)$ and

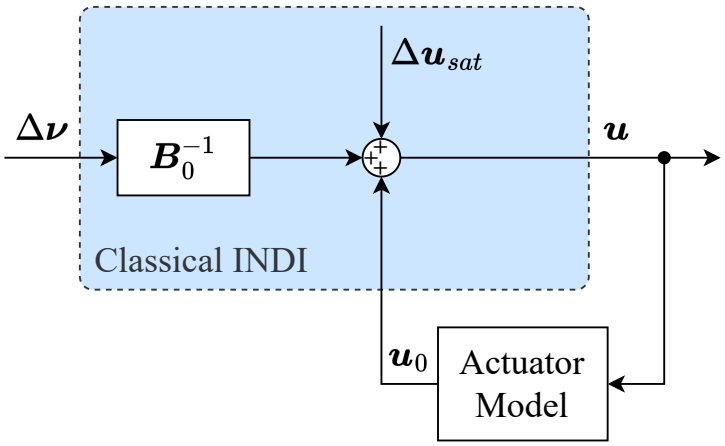
$$\boldsymbol{B}_0 \cdot (\beta_1 \cdot \boldsymbol{u}_{1,\mathcal{I}m} + \beta_2 \cdot \boldsymbol{u}_{2,\mathcal{I}m} + \cdots + \beta_m \cdot \boldsymbol{u}_{m,\mathcal{I}m}) = \boldsymbol{0} \tag{B.4}$$

only if $\beta_1 = \beta_2 = \cdots = \beta_m = 0$, since the vectors $\{\boldsymbol{B}_0 \cdot \boldsymbol{u}_{1,\mathcal{I}m}, \boldsymbol{B}_0 \cdot \boldsymbol{u}_{2,\mathcal{I}m}, \cdots, \boldsymbol{B}_0 \cdot \boldsymbol{u}_{m,\mathcal{I}m}\}$ form a basis of $\mathcal{I}m(\boldsymbol{B}_0)$ and, thus, are linearly independent.

With those relations, consider the block diagram of the classical INDI control law (8), depicted in Figure B.1. The term $\Delta\boldsymbol{u}_{sat}$ represents nonlinear effects on the inversion $\boldsymbol{B}_0^{-1} \cdot \Delta\boldsymbol{\nu}$, such as saturation of the actuators.

[1] The nullspace basis is formed by $n - m$ vectors due to its dimension. From the rank nullity theorem (LIMA, 2014), it follows that the nullspace dimension of this transformation is equal to $dim(\boldsymbol{u}) - rank(\boldsymbol{B}_0)$. Considering a full rank matrix $\boldsymbol{B}_0$, the nullspace dimension is then equal to $n - m$

Figure B.1 – Block diagram of the classical INDI control law.



Source: Author.

The command vector $\boldsymbol{u}$ is given by:

$$\boldsymbol{u}(s) = \boldsymbol{B}_0^{-1} \cdot \Delta\boldsymbol{\nu}(s) + \boldsymbol{G}(s) \cdot \boldsymbol{u}(s) + \Delta\boldsymbol{u}_{sat}(s) \tag{B.5}$$

$$\boldsymbol{u}(s) = \frac{\boldsymbol{B}_0^{-1} \cdot \Delta\boldsymbol{\nu}(s) + \Delta\boldsymbol{u}_{sat}(s)}{1 - \boldsymbol{G}(s)} \tag{B.6}$$

where $\boldsymbol{G}(s)$ represents the actuators dynamics. In general, $\boldsymbol{G}(s)$ can be well represented by a first order model, that is

$$\boldsymbol{G}(s) = \frac{\omega}{s + \omega} \tag{B.7}$$

where ω is the actuator bandwidth. Combining (B.7) and (B.6), the following is obtained:

$$\boldsymbol{u}(s) = \Delta\boldsymbol{u}_{sat}(s) + \boldsymbol{B}_0^{-1} \cdot \Delta\boldsymbol{\nu}(s) + \boldsymbol{B}_0^{-1} \cdot \Delta\boldsymbol{\nu}(s) \cdot \frac{\omega}{s} + \Delta\boldsymbol{u}_{sat}(s) \cdot \frac{\omega}{s} \tag{B.8}$$

$\Delta\boldsymbol{u}_{sat}$ is then written using the nullspace and image vectors basis, that is

$$\Delta\boldsymbol{u}_{sat}(t) = \Delta\boldsymbol{u}_{sat,\mathcal{I}m}(t) + \Delta\boldsymbol{u}_{sat,\mathcal{N}}(t) \tag{B.9}$$

where

$$\Delta\boldsymbol{u}_{sat,\mathcal{I}m} = \beta_1 \cdot \boldsymbol{u}_{1,\mathcal{I}m} + \beta_2 \cdot \boldsymbol{u}_{2,\mathcal{I}m} + \cdots + \beta_m \cdot \boldsymbol{u}_{m,\mathcal{I}m}$$

$$\Delta\boldsymbol{u}_{sat,\mathcal{N}} = \alpha_1 \cdot \boldsymbol{u}_{1,\mathcal{N}} + \alpha_2 \cdot \boldsymbol{u}_{2,\mathcal{N}} + \cdots + \alpha_{n-m} \cdot \boldsymbol{u}_{n-m,\mathcal{N}}$$

Now, combining the last term of (B.8) with (B.9), the following is obtained:

$$\Delta\boldsymbol{u}_{sat}(s) \cdot \frac{\omega}{s} = \Delta\boldsymbol{u}_{sat,\mathcal{I}m}(s) \cdot \frac{\omega}{s} + \Delta\boldsymbol{u}_{sat,\mathcal{N}}(s) \cdot \frac{\omega}{s} \tag{B.10}$$

In the same fashion, $\boldsymbol{u}$ is written using the nullspace and image basis, that is

$$\boldsymbol{u}(t) = \boldsymbol{u}_{\mathcal{I}m}(t) + \boldsymbol{u}_{\mathcal{N}}(t) \tag{B.11}$$

Finally, combining (B.11) and (B.10) into (B.8), the following results are obtained:

$$\begin{aligned}\boldsymbol{u}_{\mathcal{N}}(s) = &- \boldsymbol{u}_{\mathcal{I}m}(s) + \Delta\boldsymbol{u}_{sat,\mathcal{I}m}(s) + \boldsymbol{B}_0^{-1}\cdot\Delta\boldsymbol{\nu}(s) + \boldsymbol{B}_0^{-1}\cdot\Delta\boldsymbol{\nu}(s)\cdot\frac{\omega}{s} \\ &+ \Delta\boldsymbol{u}_{sat,\mathcal{N}}(s) + \Delta\boldsymbol{u}_{sat,\mathcal{I}m}(s)\cdot\frac{\omega}{s} + \Delta\boldsymbol{u}_{sat,\mathcal{N}}(s)\cdot\frac{\omega}{s}\end{aligned} \tag{B.12}$$

The linear transformation $\boldsymbol{B}_0$ is applied, resulting in

$$\begin{aligned}\boldsymbol{0} = &\boldsymbol{B}_0\cdot(-\boldsymbol{u}_{\mathcal{I}m}(s) + \Delta\boldsymbol{u}_{sat,\mathcal{I}m}(s) + \boldsymbol{B}_0^{-1}\cdot\Delta\boldsymbol{\nu}(s) \\ &+ \boldsymbol{B}_0^{-1}\cdot\Delta\boldsymbol{\nu}(s)\cdot\frac{\omega}{s} + \Delta\boldsymbol{u}_{sat,\mathcal{I}m}(s)\cdot\frac{\omega}{s})\end{aligned} \tag{B.13}$$

Note that all these terms can be written using only the image basis. Thus, it follows from (B.4) that

$$\begin{aligned}&-\boldsymbol{u}_{\mathcal{I}m}(s) + \Delta\boldsymbol{u}_{sat,\mathcal{I}m}(s) + \boldsymbol{B}_0^{-1}\cdot\Delta\boldsymbol{\nu}(s) \\ &+\boldsymbol{B}_0^{-1}\cdot\Delta\boldsymbol{\nu}(s)\cdot\frac{\omega}{s} + \Delta\boldsymbol{u}_{sat,\mathcal{I}m}(s)\cdot\frac{\omega}{s} = \boldsymbol{0}\end{aligned} \tag{B.14}$$

Combining this relation with (B.12), the following is obtained:

$$\boldsymbol{u}_{\mathcal{N}}(s) = \Delta\boldsymbol{u}_{sat,\mathcal{N}}(s) + \Delta\boldsymbol{u}_{sat,\mathcal{N}}(s)\cdot\frac{\omega}{s} \tag{B.15}$$

Thus, if $\Delta\boldsymbol{u}_{sat,\mathcal{N}} \neq \boldsymbol{0}$, the actuators will drift away from their initial trimmed state due to the integrating term. To restore the actuators to their initial state, a command in the opposite direction of $\Delta\boldsymbol{u}_{sat,\mathcal{N}}$ has to be given. The problem is that the linear transformation

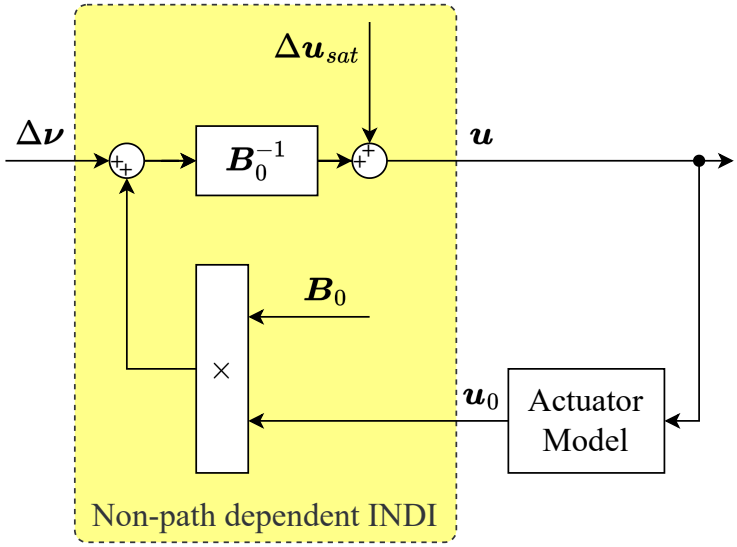
$$\boldsymbol{B}_0^{-1} : \boldsymbol{\nu} \rightarrow \boldsymbol{u}$$

can't generate a command in the direction of $-\Delta\boldsymbol{u}_{sat,\mathcal{N}}$ since $\boldsymbol{\nu} = \boldsymbol{0}$ only maps to $\boldsymbol{u} = \boldsymbol{0}$[2].

Therefore, the path dependency problem occurs due to the integration of nullspace components in the command vector. Now, consider the block diagram of the control law (14), illustrated in the figure bellow.

[2] This follows from the rank-nullity theorem (LIMA, 2014). The linear transformation $\boldsymbol{B}_0^{-1} : \boldsymbol{\nu} \rightarrow \boldsymbol{u}$ has nullspace of dimension zero, that is, it contains only the $\boldsymbol{0}$ vector.

Figure B.2 – Block diagram of the proposed not path dependent INDI control law.



Source: Author.

The command vector $\boldsymbol{u}$ is given by

$$\boldsymbol{u}(s) = \boldsymbol{B}_0^{-1} \cdot [\Delta \boldsymbol{\nu}(s) + \boldsymbol{B}_0 \cdot \boldsymbol{G}(s) \cdot \boldsymbol{u}(s)] + \Delta \boldsymbol{u}_{sat}(s) \tag{B.16}$$

the term $\boldsymbol{u}$ is written using the nullspace and image basis

$$\boldsymbol{u}(t) = \boldsymbol{u}_{\mathcal{I}m}(t) + \boldsymbol{u}_{\mathcal{N}}(t) \tag{B.17}$$

where

$$\boldsymbol{u}_{\mathcal{I}m} = \beta_1 \cdot \boldsymbol{u}_{1,\mathcal{I}m} + \beta_2 \cdot \boldsymbol{u}_{2,\mathcal{I}m} + \cdots + \beta_m \cdot \boldsymbol{u}_{m,\mathcal{I}m}$$
$$\boldsymbol{u}_{\mathcal{N}} = \alpha_1 \cdot \boldsymbol{u}_{1,\mathcal{N}} + \alpha_2 \cdot \boldsymbol{u}_{2,\mathcal{N}} + \cdots + \alpha_{n-m} \cdot \boldsymbol{u}_{n-m,\mathcal{N}}$$

Combining (B.16) and (B.17), it's obtained

$$\boldsymbol{u}_{\mathcal{I}m}(s) + \boldsymbol{u}_{\mathcal{N}}(s) = \boldsymbol{B}_0^{-1} \cdot [\Delta \boldsymbol{\nu}(s) + \boldsymbol{B}_0 \cdot \boldsymbol{G}(s) \cdot (\boldsymbol{u}_{\mathcal{I}m}(s) + \boldsymbol{u}_{\mathcal{N}}(s))] + \Delta \boldsymbol{u}_{sat}(s)$$

$$\boldsymbol{u}_{\mathcal{I}m}(s) + \boldsymbol{u}_{\mathcal{N}}(s) = \boldsymbol{B}_0^{-1} \cdot [\Delta \boldsymbol{\nu}(s) + \boldsymbol{B}_0 \cdot \boldsymbol{G}(s) \cdot \boldsymbol{u}_{\mathcal{I}m}(s) + \cancelto{\mathbf{0}}{\boldsymbol{B}_0 \cdot \boldsymbol{G}(s) \cdot \boldsymbol{u}_{\mathcal{N}}(s)}] + \Delta \boldsymbol{u}_{sat}(s)$$

$$\boldsymbol{u}_{\mathcal{I}m}(s) + \boldsymbol{u}_{\mathcal{N}}(s) = \boldsymbol{B}_0^{-1} \cdot [\Delta \boldsymbol{\nu}(s) + \boldsymbol{B}_0 \cdot \boldsymbol{G}(s) \cdot \boldsymbol{u}_{\mathcal{I}m}(s)] + \Delta \boldsymbol{u}_{sat}(s)$$

$$\boldsymbol{u}_{\mathcal{N}}(s) = \boldsymbol{B}_0^{-1} \cdot \Delta \boldsymbol{\nu}(s) + \boldsymbol{G}(s) \cdot \boldsymbol{u}_{\mathcal{I}m}(s) + \Delta \boldsymbol{u}_{sat}(s) - \boldsymbol{u}_{\mathcal{I}m}(s) \tag{B.18}$$

$\Delta \boldsymbol{u}_{sat}$ is then written using the nullspace and image vectors basis, that is

$$\boldsymbol{u}_{\mathcal{N}}(s) = \boldsymbol{B}_0^{-1} \cdot \Delta \boldsymbol{\nu}(s) + \boldsymbol{G}(s) \cdot \boldsymbol{u}_{\mathcal{I}m}(s) + \Delta \boldsymbol{u}_{sat,\mathcal{I}m}(s) + \Delta \boldsymbol{u}_{sat,\mathcal{N}}(s) - \boldsymbol{u}_{\mathcal{I}m}(s) \tag{B.19}$$

The linear transformation $\boldsymbol{B}_0$ is applied, obtaining

$$\mathbf{0} = \boldsymbol{B}_0 \cdot \left(\boldsymbol{B}_0^{-1} \cdot \Delta\boldsymbol{\nu}(s) + \boldsymbol{G}(s) \cdot \boldsymbol{u}_{\mathcal{I}m}(s) + \Delta\boldsymbol{u}_{sat,\mathcal{I}m}(s) - \boldsymbol{u}_{\mathcal{I}m}(s) \right) \tag{B.20}$$

Since all these vectors can be written using the image basis, it follows from (B.4) that

$$\boldsymbol{B}_0^{-1} \cdot \Delta\boldsymbol{\nu}(s) + \boldsymbol{G}(s) \cdot \boldsymbol{u}_{\mathcal{I}m}(s) + \Delta\boldsymbol{u}_{sat,\mathcal{I}m}(s) - \boldsymbol{u}_{\mathcal{I}m}(s) = \mathbf{0} \tag{B.21}$$

Therefore, (B.19) reduces to

$$\boldsymbol{u}_{\mathcal{N}}(s) = \Delta\boldsymbol{u}_{sat,\mathcal{N}}(s) \tag{B.22}$$

Finally, comparing (B.22) and (B.15) the difference can be clearly noticed. (B.22) provides a path free solution since there is no integration of terms in the nullspace of $\boldsymbol{B}_0$ in the equivalent transfer function.

APPENDIX C – Redistributed pseudo inverse algorithm

Let $\boldsymbol{u}_0 \in \mathbb{U}$ be an arbitrary input vector and $\boldsymbol{\nu}_0$ the pseudo control generated by $\boldsymbol{u}_0$, that is

$$\boldsymbol{\nu}_0 = \boldsymbol{B}_0 \cdot \boldsymbol{u}_0$$

Now, let $\boldsymbol{\nu}_{cmd}$ be the commanded pseudo control, $\Delta\boldsymbol{\nu}$ is defined as

$$\Delta\boldsymbol{\nu} = \boldsymbol{\nu}_{cmd} - \boldsymbol{\nu}_0$$

The RSPI algorithm starts by computing the pseudo inverse solution $\Delta\boldsymbol{u}$, that is

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

If $\boldsymbol{u}_0 + \Delta\boldsymbol{u}$ is whithin the actuators limits, that is

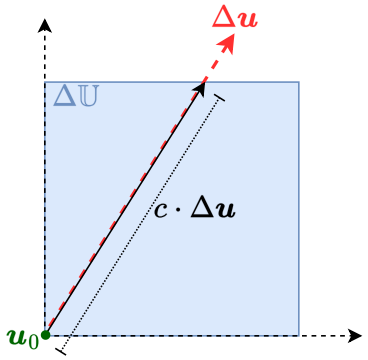
$$\boldsymbol{u}_0 + \Delta\boldsymbol{u} \in \mathbb{U} = \{\boldsymbol{u}_0 + \Delta\boldsymbol{u} \in \mathbb{R}^n \mid \underline{\boldsymbol{u}} < \boldsymbol{u}_0 + \Delta\boldsymbol{u} < \bar{\boldsymbol{u}}\} \tag{C.2}$$

or in an equivalent form as

$$\Delta\boldsymbol{u} \in \Delta\mathbb{U} = \{\Delta\boldsymbol{u} \in \mathbb{R}^n \mid \underbrace{\underline{\boldsymbol{u}} - \boldsymbol{u}_0}_{\Delta\underline{\boldsymbol{u}}} < \Delta\boldsymbol{u} < \underbrace{\bar{\boldsymbol{u}} - \boldsymbol{u}_0}_{\Delta\bar{\boldsymbol{u}}}\} \tag{C.3}$$

the code stops and the solution $\boldsymbol{u}_0 + \Delta\boldsymbol{u}$ is used. If saturation occurs, $\Delta\boldsymbol{u}$ is scaled using the scaling function illustrated in the figure bellow.

Figure C.1 – Scaling function in a bi dimensional space. The calculated command vector is scaled to the border of $\Delta\mathbb{U}$.



Source: Author.

The scale factor $c \in [0,\ 1]$ is the largest scalar such that at least one actuator command i within $c \cdot \Delta\boldsymbol{u}$ is located at its limit value $\Delta\bar{\boldsymbol{u}}$ or $\Delta\underline{\boldsymbol{u}}$ and all other actuator commands are within their respective limits.

Next, several signals are updated for the next iteration:

- The i-th column of the $\boldsymbol{B}_0$ matrix is replaced by zeros, such that the i-th actuator is not used in the next iterations.
- The vector $\boldsymbol{u}_0$ is updated: $\boldsymbol{u}_0 = \boldsymbol{u}_0 + c \cdot \Delta\boldsymbol{u}$.
- $\boldsymbol{\nu}_0$ is updated in the same fashion: $\boldsymbol{\nu}_0 = \boldsymbol{\nu}_0 + c \cdot \Delta\boldsymbol{\nu}$.

Then, the algorithm restarts from (C.1), where this recursive formula is done until one of the following conditions is met:

- $c = 1$ at some iteration, which means the pseudo command $\boldsymbol{\nu}_{cmd}$ is feasible.
- $\boldsymbol{B}_0$ has a rank drop due to the succesive column modifications. Then $\boldsymbol{\nu}_{cmd}$ is not feasible.

If $\boldsymbol{\nu}_{cmd}$ is not feasible, the calculated input vector $\boldsymbol{u}$ gives the maximum physically possible pseudo command in the direction of $\boldsymbol{\nu}_{cmd}$ (STEPHAN; FICHTER, 2019), as illustrated in Figure 18.