Incremental Backstepping Control of an Overactuated Tilt-Quadrotor

André Macara Reis

Thesis to obtain the Master of Science Degree in

Mechanical Engineering

Supervisors: Prof.ª Alexandra Bento Moutinho
Dr. Rafael de Angelis Cordeiro

Examination Committee

Chairperson:	Prof. Paulo Jorge Coelho Ramalho Oliveira
Supervisor:	Dr. Rafael de Angelis Cordeiro
Member of the Committee:	Prof. Miguel Afonso Dias de Ayala Botto

October 2020

Acknowledgments

I would like to thank my supervisor Prof.ª Alexandra Moutinho and co-supervisor Rafael Cordeiro for the opportunity to work on this amazing project and for their constant help and support.

I would like to thank my friends Inês Santos and João Silva for their support and company, as well as their help and companionship.

I would like to thank my family for all the counseling and encouragement.

Finally, I would like to thank my father, who has always been there for me, even in the toughest moments.

Resumo

Esta dissertação continua o trabalho de desenvolvimento do projecto ALIV, propondo controladores não-lineares de atitude e posição para o ALIV-3, um veículo aéreo capaz de descolar e aterrar verticalmente que pode mudar a sua atitude sem mudar a posição no espaço e vice-versa.

Os controladores não lineares são desenvolvidos usando dois métodos: Backstepping com Filtro de Comando e Backstepping Incremental com Filtro de Comando.

As leis de controlo são obtidas usando os dois métodos, e seguindo-se a implementação dos controladores e do modelo do ALIV-3 no Simulink, uma ferramenta de simulação do `Matlab`, onde os parâmetros dos controladores são ajustados para obter o desempenho desejado.

Desta forma, o modelo e os controladores são validados em simulação. Ambas as soluções mostram boas capacidades de estabilização e seguimento de referências assim como robustez a ruído de medição e erros de modelo. Adicionalmente, são feitos testes extensivos à robustez dos dois controladores de modo a assegurar que, em sendo implementados no ALIV-3 real, são capazes de alcançar um óptimo desempenho de estabilização e seguimento de referências.

Palavras-chave: Backstepping Incremental, Backstepping, Filtro de Comando, ALIV, Controlo de Atitude, Controlo de Posição, Controlo Não-linear, Sistema Não-linear, Sistema Sobreactuado, Alocação de Controlo

Abstract

This dissertation continues the work on the development of the ALIV project, proposing nonlinear attitude and position controllers for the ALIV-3, a vertical take-off and landing capable aerial vehicle that can change attitude without changing its position in space and vice-versa.

The nonlinear controllers are developed using two design methods: Command Filtered Backstepping and Command Filtered Incremental Backstepping.

The control laws are obtained using the two methods, followed by implementing the controllers and the ALIV-3 model in Simulink, a `Matlab`'s simulation tool, where the controller parameters are adjusted to achieve the desired performance.

Thus, the model and controllers are validated in simulation. Both solutions show good stabilizing and reference tracking capabilities as well as measurement noise and model error robustness after implementation in simulation. Additionally, extensive testing on the robustness of the two controllers is made in order to ensure that, if they are implemented on the real ALIV-3, they are able to achieve great stabilizing and reference tracking performance.

Keywords: Incremental Backstepping, Backstepping, Command Filtering, ALIV, Attitude Control, Position Control, Nonlinear Control, Nonlinear System, Over actuated System, Control Allocation

Contents

List of Tables

List of Figures

q

Nomenclature

Acronyms

UAV	Unmanned Aerial Vehicle
ALIV	Autonomous Locomotive Individual Vehicle
BKS	Backstepping
IBKS	Incremental Backstepping
NED	North-East-Down
PWM	Pulse Width Modulation
BLDC	BrushLess Direct Current motor

Variables and Constants

Symbol	Description	Unit
x	x axis coordinate	m
y	y axis coordinate	m
z	z axis coordinate	m
$\boldsymbol{P}$	Linear position	m
u	Velocity in the x axis direction	m/s
v	Velocity in the y axis direction	m/s
w	Velocity in the z axis direction	m/s
$\boldsymbol{V}$	Linear velocity	m/s
φ	Roll angle	rad
θ	Pitch angle	rad
ψ	Yaw angle	rad
$\mathbf{\Phi}$	Angular position	rad
p	Roll rate	rad/s
q	Pitch rate	rad/s
r	Yaw rate	rad/s
$\mathbf{\Omega}$	Angular velocity	rad/s
$\boldsymbol{s}_1$	Attitude and Position state vector	rad and m

Symbol	Description	Unit
$\boldsymbol{s}_2$	Angular and Linear Velocity state vector	rad/s and m/s
$\boldsymbol{S}$	Rotation matrix	
$\boldsymbol{T}$	Kinematic Angles Relation	
ϕ_2	Servo angle on motor two in the roll angle direction	rad
ϕ_4	Servo angle on motor four in the roll angle direction	rad
ϑ_2	Servo angle on motor two in the pitch angle direction	rad
ϑ_4	Servo angle on motor four in the pitch angle direction	rad
ν_i	Angle of servo i	rad
$\boldsymbol{\nu}$	Servo angles vector	rad
ω_i	Rotational speed of motor i	rad/s
$PWM_{\omega i}$	Input PWM signal to motor i	μs
$PWM_{\nu i}$	Input PWM signal to servo i	μs
T_i	Thrust produced by propeller of motor i	N
Q_i	Moment produced by propeller of motor i	$N.m$
$\boldsymbol{F}$	Applied force to the center of mass of the platform in the body-fixed frame	N
F_x	Applied force to the center of mass of the platform in the x axis direction of the body-fixed frame	N
F_y	Applied force to the center of mass of the platform in the y axis direction of the body-fixed frame	N
F_z	Applied force to the center of mass of the platform in the z axis direction of the body-fixed frame	N
$\boldsymbol{M}$	Applied moment to the center of mass of the platform in the body-fixed frame	$N.m$
M_x	Applied moment to the center of mass of the platform in the x axis direction of the body-fixed frame	$N.m$
M_y	Applied moment to the center of mass of the platform in the y axis direction of the body-fixed frame	$N.m$
M_z	Applied moment to the center of mass of the platform in the z axis direction of the body-fixed frame	$N.m$
$\boldsymbol{N}$	Concatenation of the applied moments and forces to the center of mass of the platform in the body-fixed frame	$N.m$ and N
$\boldsymbol{\gamma}_\Phi$	Time-integration of the attitude tracking error	$rad.s$
$\boldsymbol{\zeta}_\Phi$	Attitude tracking error	rad
$\boldsymbol{\zeta}_\Omega$	Angular velocity tracking error	rad/s
$\boldsymbol{\zeta}_{\Phi,c}$	Compensated attitude tracking error	rad
$\boldsymbol{\gamma}_p$	Time-integration of the position tracking error	$m.s$
$\boldsymbol{\zeta}_p$	Position tracking error	m

Symbol	Description	Unit
$\boldsymbol{\zeta}_v$	Linear velocity tracking error	m/s
$\boldsymbol{\zeta}_{p,c}$	Compensated position tracking error	m
$\boldsymbol{\gamma}$	Concatenation of the time-integration of attitude and position tracking errors	$rad.s$ and $m.s$
$\boldsymbol{\zeta}_1$	Concatenation of the attitude and position tracking errors	rad and m
$\boldsymbol{\zeta}_2$	Concatenation of the angular and linear velocities tracking errors	rad/s and m/s
$\boldsymbol{\zeta}_{1,c}$	Concatenation of the compensated attitude and position tracking errors	rad and m
$\boldsymbol{\Upsilon}$	Demanded actuation in percentage of the maximum actuation	%
V	Generic *Lyapunov* function	
α	Generic desired states	
ω_n	Generic natural frequency of a filter	rad/s
ξ	Generic damping coefficient of a filter	
τ	Generic time-constant	s
$\boldsymbol{x}$	Generic state vector	
$\boldsymbol{w}$	Generic partial state vector	
$\boldsymbol{f}$	Generic vector function	
$\boldsymbol{g}_i$	Generic matrix function	
$\boldsymbol{u}$	Generic input vector	
$\boldsymbol{A}_0$	Linearization of $\boldsymbol{f}$ about $\boldsymbol{x}_0$	
$\boldsymbol{B}_0$	Linearization of $\boldsymbol{g}$ about $\boldsymbol{x}_0$	
$\overline{\boldsymbol{B}}_0$	Constant part of $\boldsymbol{B}_0$	
m	ALIV3 mass	kg
d	ALIV3 arm length	m
$\boldsymbol{J}$	ALIV3 inertia matrix	$kg.m^2$
τ_ω	BLDC motors time constant	s
K_ω	BLDC motors steady-state gain	$rad/(s.\mu s)$
$u_{\omega,d}$	BLDC motors PWM dead zone	μs
τ_ν	Servo motors time constant	s
K_ν	Servo motors steady-state gain	$rad/\mu s$
K_T	Propeller thrust constant	$N.s^2/rad^2$
K_Q	Propeller moment constant	$N.m.s^2/rad^2$
g_0	Absolute gravity acceleration	m/s^2
$\boldsymbol{g}$	Acceleration of gravity	m/s^2
$\boldsymbol{K}_{pi}$	Position tracking error integration constant gain matrix	
$\boldsymbol{K}_p$	Position tracking error constant gain matrix	
$\boldsymbol{K}_v$	Linear velocity tracking error constant gain matrix	
$\boldsymbol{K}_{\Phi i}$	Attitude tracking error integration constant gain matrix	

Symbol	Description	Unit
$\boldsymbol{K}_{\Phi}$	Attitude tracking error constant gain matrix	
$\boldsymbol{K}_{\Omega}$	Angular velocity tracking error constant gain matrix	
$\boldsymbol{K}_1$	Velocities tracking error constant gain matrix	
$\boldsymbol{K}_2$	Velocities tracking error constant gain matrix	
$\boldsymbol{K}_{Alloc}$	Allocation gain matrix	
δ_{Υ}	Matrix inversion determinant enlarger constant	
$\boldsymbol{K}_{\Upsilon}$	IBKS Control Allocation correction matrix	
$\boldsymbol{\Upsilon}_r$	IBKS Control Allocation reference	
$\boldsymbol{I}_k$	$k \times k$ identity matrix	

Constants

BLDC Motors

Symbol	Value	Unit
τ_ω	0.162	s
K_ω	0.5	$rad/(s.\mu s)$
$u_{\omega,d}$	538	μs
PWM_{min}	1000	μs
PWM_{max}	2000	μs

Servo Motors

Symbol	Value	Unit
τ_ν	0.2308	s
K_ν	$\frac{1}{1500}\pi$	$rad/\mu s$
ν_{max}	$\frac{1}{3}\pi$	rad
$\dot{\nu}_{max}$	$\frac{3}{2}\pi$	rad/s
PWM_{min}	1000	μs
PWM_{max}	2000	μs

Propellers

Symbol	Value	Unit
K_T	$1.40 \cdot 10^{-5}$	$N.s^2/rad^2$
K_Q	$4.20 \cdot 10^{-7}$	$N.m.s^2/rad^2$

ALIV3 Platform

Symbol	Value	Unit
$\boldsymbol{J}$	$\begin{bmatrix} 0.0367 & 0 & 0 \\ 0 & 0.0262 & 0 \\ 0 & 0 & 0.0504 \end{bmatrix}$	$kg.m^2$
m	1.903	kg
d	0.3	m

BKS Controller

Symbol	Value	Unit
$\boldsymbol{K}_\Omega$	$\begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$	
$\boldsymbol{K}_\Phi$	$\begin{bmatrix} 0.3 & 0 & 0 \\ 0 & 0.3 & 0 \\ 0 & 0 & 0.3 \end{bmatrix}$	
$\boldsymbol{K}_{\Phi i}$	$\begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$	
$\boldsymbol{K}_v$	$\begin{bmatrix} 18.5 & 0 & 0 \\ 0 & 18.5 & 0 \\ 0 & 0 & 18.5 \end{bmatrix}$	
$\boldsymbol{K}_p$	$\begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 1.5 & 0 \\ 0 & 0 & 1.5 \end{bmatrix}$	
$\boldsymbol{K}_{pi}$	$\begin{bmatrix} 0.7 & 0 & 0 \\ 0 & 0.7 & 0 \\ 0 & 0 & 0.7 \end{bmatrix}$	
$\omega_{n\Omega}$	17	rad/s
ξ_Ω	1.1	Dimensionless
ω_{nv}	9.8	rad/s
ξ_v	1.1	Dimensionless

Symbol	Value	Unit
$\boldsymbol{K}_{Alloc}$	$\begin{bmatrix} -9.7204 \cdot 10^{-4} & -10.8125 & 1.0608 \cdot 10^{-2} & 1.2446 \cdot 10^{-15} & -4.4952 \cdot 10^{-16} & 1.6234 \\ -10.7154 & -1.0705 \cdot 10^{-1} & -1.0714 & -1.2123 \cdot 10^{-15} & 4.41 \cdot 10^{-16} & 1.6234 \\ 9.7204 \cdot 10^{-4} & 10.8125 & -1.0608 \cdot 10^{-2} & 2.9109 \cdot 10^{-16} & -4.1964 \cdot 10^{-16} & 1.6234 \\ 10.7154 & 1.0705 \cdot 10^{-1} & 1.0714 & -1.1735 \cdot 10^{-15} & 4.4129 \cdot 10^{-16} & 1.6234 \\ -3.3754 \cdot 10^{-2} & 3.4821 \cdot 10^{-1} & 3.6837 \cdot 10^{-1} & 2.5143 \cdot 10^{-15} & -11.2743 & 1.2097 \cdot 10^{-18} \\ 3.3754 \cdot 10^{-2} & -3.4821 \cdot 10^{-1} & -3.6837 \cdot 10^{-1} & 2.4924 \cdot 10^{-15} & -11.2743 & -1.2097 \cdot 10^{-18} \\ -3.7205 & 3.7174 & 37.2051 & 11.2743 & 7.9276 \cdot 10^{-16} & -1.5696 \cdot 10^{-16} \\ 3.7205 & -3.7174 & -37.2051 & 11.2743 & 7.8249 \cdot 10^{-16} & 2.47078 \cdot 10^{-16} \end{bmatrix}$	
$\boldsymbol{\Upsilon}_0$	$\begin{bmatrix} 55 & 55 & 55 & 55 & 0 & 0 & 0 & 0 \end{bmatrix}^T$	%
$\boldsymbol{N}(\boldsymbol{\Upsilon}_0)$	$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -16.94 \end{bmatrix}^T$	$N.m$ and N

IBKS Controller

Symbol	Value	Unit
$\boldsymbol{K}_2$	$\begin{bmatrix} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \begin{bmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{bmatrix} \end{bmatrix}$	
$\boldsymbol{K}_{\Phi}$	$\begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}$	
$\boldsymbol{K}_{\Phi i}$	$\begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.1 \end{bmatrix}$	
$\boldsymbol{K}_p$	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	
$\boldsymbol{K}_{pi}$	$\begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix}$	
$\omega_{n\Omega}$	14.5	rad/s
ξ_{Ω}	1.05	Dimensionless
ω_{nv}	9	rad/s
ξ_v	1.05	Dimensionless
τ_{Υ}	$2 \cdot 10^{-2}$	s
δ_{Υ}	$5 \cdot 10^{-2}$	Dimensionless

Symbol	Value	Unit
$\boldsymbol{K}_{\Upsilon}$	$-10^{-5}\begin{bmatrix}400 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 400 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 400 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 400 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 125 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 125 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 125 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 125\end{bmatrix}$	
$\boldsymbol{\Upsilon}_r$	$\begin{bmatrix}55 & 55 & 55 & 55 & 0 & 0 & 0 & 0\end{bmatrix}^T$	%

Chapter 1

Introduction

In recent years, Unmanned Aerial Vehicles (UAV) have been used for many applications, from recreational and surveillance vehicles to cargo carrier, with increasingly popularity. With advancing technology, it is now possible to build more accurate and efficient sensors as well as smaller and higher-performing computers, allowing for better and cheaper controllers than ever before. Implementing nonlinear controllers digitally was unlikely some decades ago due to its high processing power needs compared to traditional linear controllers.

Despite their inherent unstable nature, quadrotors are an especially popular type of UAV due to their reliability, relatively cheap manufacturing and platform configuration versatility.

Linear control design achieves good stability results around the equilibrium point of quadrotors, hovering with zero angular and linear speeds. However, if a wider ranged performance is wanted, this control strategy may have a reduced attraction domain, affecting stability. This reason motivates the design of nonlinear controllers for this type of UAV.

This work aims to contribute with possible nonlinear control strategies for a specific quadrotor developed at Instituto Superior Técnico since 2008. It continues the ALIV project on a fully actuated and innovative quadrotor configuration.

1.1 The ALIV Concept and History

The Autonomous Locomotive Individual Vehicle (ALIV) concept is based on the patent US8128033B2 - ”System and Process of Vector Propulsion with Independent Control of three Translation and Three Rotation Axis” [32].

The ALIV platform is similar to a classical quadrotor with a cross-shaped structure. The difference is that two of the main arms are equipped with two servo motors each along with shifted propeller rotation direction. The servo motors can tilt the attached propellers in two different axis, which allows the ALIV to move in space without tilting its main core. The platform can also maintain a tilted attitude while not changing its position in space.

Being developed at Instituto Superior Técnico (IST), the ALIV - IST is a project which several students have contributed to. The project started in 2008 with Costa [13] modeling, simulating and building the first iteration of the ALIV.

It was reiterated by Pedro [31], making the platform more rigid with a substantial emphasis on the theoretical design of the platform by employing a Genetic Algorithm and Finite Element Method in order to improve the structure and the rotor performance.

The project was progressed by Fernandes [19], who built the platform using carbon fibre composite materials and obtained actuator components and avionics.

In 2013, Mateos [30], updated the platform by improving its balance and stiffness. The platform was modelled, identified and simulated. Linear Quadratic Regulator (LQR) controllers were designed for the platform without the action of the servos, for proof of concept. Motor curves and moments of inertia of the platform were identified.

Marques [29] was the last student to work on the ALIV3 platform, in 2018. He designed and implemented a linear controller for the Tilt-Quadrotor. The model-based control design method was used, starting by linearizing the system around the intended operation point and applying the Linear Quadratic Regulator (LQR) method. He obtained a control law for the actuators in order to stabilize the platform and validated the solution experimentally.

1.2 ALIV Background

Quadrotos are typically fast and agile UAVs, which high maneuverability requires powerful and robust control strategies. The increased UAV configuration complexity further increases this requirement.

Conventional quadrotor UAVs are well studied [22] and the Backstepping method has been applied, producing good results [28, 23].

As stabilizing and reference tracking problems are solved for conventional quadrotors, more sophisticated manouvers, for example squeezing through smaller spaces or higher flight speeds, are ambitioned. Thus, new configurations created, emerging new problems to solve, such as the fixed-wing tilt-rotor UAV [9], the hexarotor UAV [17] where Incremental Backstepping was applied or even folding quadcopters to alter their dimensions mid-flight [7, 16].

The oddity of the ALIV is that it is a non-conventional quadrotor with two tilting rotors in two directions and no wings, thriving at relatively low speeds with high maneuverability. An LQR controller was already developed for this UAV, but a wider region of non model-dependent stability is desired. It can be provided by Incremental Backstepping, which has promising characteristics even compared to other nonlinear control design methods such as the less model-dependent Incremental Nonlinear Dynamics Inversion (INDI) [5]. It achieves so by seeking to work with the nonlinear dynamics and not against it.

1.3 Objectives and Contributions

The main purpose of this work is to develop a non-linear controller that achieves good stabilization, reference tracking, measurement noise robustness and model error robustness. It continues the work on the ALIV project thus far.

The steps to reach this goal are:

- Development of a simulator model in Matlab/Simulink;
- Development of a Backstepping controller.
- Development of an Incremental Backstepping Controller;
- Validation of the model and controllers through simulation;
- Perform extensive robustness testing.

This works contributes with a Matlab/Simulink simulator of the nonlinear ALIV-3 model and two promising nonlinear controllers that have been tested and validated through simulation.

1.4 Thesis Structure

This dissertation is mainly composed by four sections.

In chapter 2, the ALIV-3 model is introduced inclunding the actuator models and rigid-body dynamics.

In chapter 3, the background theory for the control design is introduced and the control laws for the two controllers are obtained. It includes extensive proving and symbolic manipulation of expressions.

Chapters 4 and 5 include the simulation results and analysis for both controllers. The results in chapter 4 are for validation, thus the model includes no disturbances or parameter uncertainties. An extensive robustness analysis is made in chapter 5, considering parameter uncertainties such as measurement noise, center of gravity deviation and changes in mass, inertia, arm length, motor parameters or servo parameters. Additionally, the situations where one of the motors or servos is different from the others is considered.

Lastly, in chapter 6, the most relevant conclusions are presented and some future work recommendations are made.

Chapter 2

The ALIV3 Model

In this chapter, the assumptions and mathematical model of the ALIV-3 platform are introduced. This includes the actuators model and the equations of motion. A simplified illustration of the ALIV-3 can be seen in figure 2.1, in which d is the arm length of the ALIV-3, the numbers denote the number of the motor-rotor set and x, y, z are the Cartesian coordinate axis, better explained in the next sections.

2.1 General Modeling Assumptions

In order to model the ALIV-3 tilt-quadrotor, some assumptions are made:

- The platform is a rigid body, i.e. the platform is nondeformable,
- The platform is symmetric about the vertical planes containing the UAV arms,
- Wind resistance is neglected,
- The ground effect is neglected, which is the effect the presence of the ground has on the platform, which tends to increase lift the closer the propeller is to the ground,
- All sets of motors, servos and propellers (or rotors) are identical,
- There is no slip between the motor shafts and the propellers,
- The geometrical center of the platform is coincident with the center of mass,
- The ALIV-3 is a planar body.

2.2 Coordinate Systems

It is crucial to define the coordinate systems the variables are related to in order to describe the UAV motion. Two different reference frames are adopted in this project: the Earth-fixed reference frame, considered (approximately) inertial, and the (local) body-fixed frame.

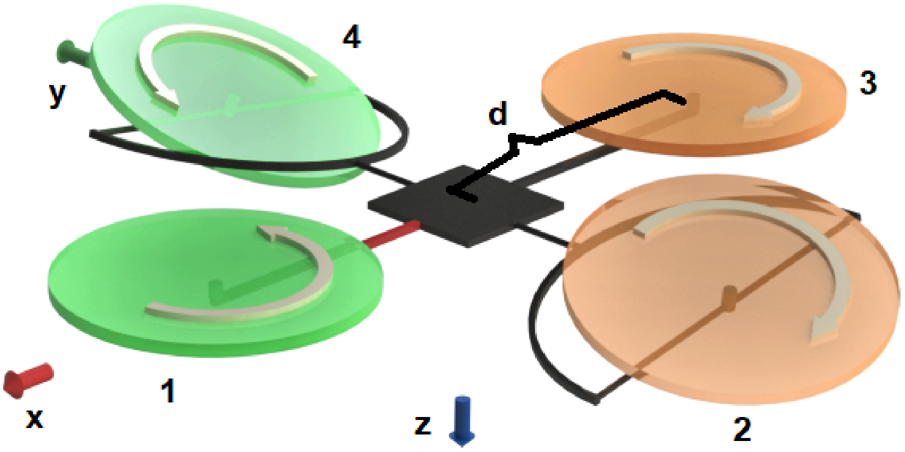

Figure 2.1: ALIV-3 Body-fixed Coordinate System

The adopted Earth-fixed frame uses the North-East-Down (NED) convention. As the name indicates, the three perpendicular axis are pointing North, East and to the center of Earth respectively. The frame origin is coincident with the quadrotor initial position.

The body-fixed frame origin is coincident with the quadrotor center of mass, the x axis pointing forward, represented by the red arm in figure 2.1, y axis pointing to its right and the z axis pointing downward.

The roll angle φ is the angle of rotation around the x axis, the pitch angle θ around the y axis and the yaw angle ψ around the z axis, all with respect to the inertial frame following the usual signs convention.

2.3 Actuation

For the ALIV-3 model, the actuators are four motor-propeller sets and four servo motors for vectoring two of the rotors. The actuators, whose states are controlled slave controllers, named Electronic Speed Controllers (ESC) in the case of the motors, which have Pulse Width Modulation (PWM) signals as inputs, are the means to affect the linear and angular positions, velocities and accelerations in order to stabilize and control the tilt-quadrotor.

2.3.1 Brushless Motors

The motors present in the ALIV-3 platform are BrushLess Direct Current (BLDC) motors. These require a driver, Electronic Speed Controller (ESC), in order for them to rotate, in contrast to brushed direct current motors which only require a direct current voltage applied to their terminals.

The desired angular speed is sent to the ESCs using PWM signals. PWM signals are square waves whose duration is interpreted by the ESCs as the desired speed. The ESCs receive pulses ranging from PWM_{min} to PWM_{max}, setting the motors, linearly, to the minimum and maximum speed respectively.

Neglecting the dynamics of the electrical components of the BLDC, as it has much faster dynamics than the mechanical part, the dynamics of the motors may be approximated by:

$$\omega_i = \frac{K_\omega}{\tau_\omega s + 1} u_{\omega i}, \tag{2.3.1}$$

where ω_i is the angular speed of the motor i in radians per second, K_ω is its steady-state gain in $rad/(s.\mu s)$, τ_ω is its time constant in seconds and $u_{\omega i}$ the input which is dependent on the pulse duration of the PWM sent to the motor i in μs, which can be defined as

$$u_{\omega i} = PWM_{\omega i} - u_{\omega,d}, \tag{2.3.2}$$

where $PWM_{\omega i}$ is the pulse duration of the PWM sent to the motor i in μs and $u_{\omega,d}$ is the ESCs PWM dead-zone in μs.

2.3.2 Servo Motors

Similarly to the BLDC motors, the inputs to the servo motors are PWM signals with the same range, PWM_{min} to PWM_{max}. However, the operation of the servo motors is symmetric, i.e. PWM_{min} pulse duration sets the servo motor to its minimum angle of $-\nu_{max}$ and PWM_{max} pulse duration sets it to its maximum angle of ν_{max}. The angular speed of each of the servo motors is limited to $\dot{\nu}_{max}$.

For each of two motors with servos there is one servo which tilts the motor in the roll direction, ϕ_2 for motor 2 and ϕ_4 for motor 4, and one servo that tilts the motor in the pitch direction, ϑ_2 for motor 2 and ϑ_4 for motor 4.

$$\boldsymbol{\nu} = \begin{bmatrix} \nu_1 \\ \nu_2 \\ \nu_3 \\ \nu_4 \end{bmatrix} = \begin{bmatrix} \phi_2 \\ \phi_4 \\ \vartheta_2 \\ \vartheta_4 \end{bmatrix},$$

where $\boldsymbol{\nu}$ is the concatenation of the four servo angles ϕ_2, ϕ_4, ϑ_2 and ϑ_4.
Each of the servo dynamics is modelled by:

$$\nu_i = \frac{K_\nu}{\tau_\nu s + 1} u_{\nu i}, \tag{2.3.3}$$

where ν_i is the servo angle i in radians, K_ν its steady-state gain in $rad/\mu s$, τ_ν its time constant in seconds and $u_{\nu i}$ the input, in μs, which is dependent on the pulse duration of the PWM signal sent to each servo.

$$u_{\nu i} = PWM_{\nu i} - \frac{PWM_{min} + PWM_{max}}{2}, \tag{2.3.4}$$

where $PWM_{\nu i}$ is the pulse duration of the PWM sent to servo i in μs.

2.3.3 Propellers

The Blade Momentum Theory [8] states that the thrust T and the torque Q produced by a propeller are

$$T = C_T \rho A_p r^2 \omega^2, \tag{2.3.5a}$$

$$Q = C_Q \rho A_p r^3 \omega^2, \tag{2.3.5b}$$

where C_T is the coefficient of thrust of the propeller, C_Q is the coefficient of torque, ρ is the fluid density, A_p is the circular area the propeller sweeps, r is the the radius of A_p and ω is the angular speed of the propeller.

Assuming the air density, propeller dimensions and configurations to be constant allows the produced torque and thrust of each propeller to be obtained as:

$$T_i = K_T \omega_i^2, \tag{2.3.6a}$$

$$Q_i = K_Q \omega_i^2, \tag{2.3.6b}$$

where T_i is the thrust produced by the propeller i, in N, Q_i is the torque produced by propeller i, in $N.m$, ω_i is the rotational speed of the propeller i in radians per second, K_T, in $N.s^2.rad^{-2}$, and K_Q, in $N.m.s^2.rad^{-2}$, are the thrust and moment coefficients, respectively. Note that the thrust and moment of a propeller are related in the following manner

$$T_i = \frac{K_T}{K_Q} Q_i. \tag{2.3.7}$$

2.3.4 Actuation Induced Forces and Moments

The aforementioned propeller thrusts and moments combined with the servo angles and the Tilt-Quadrotor configuration produce forces and moments on the platform.

For the ALIV platform, the resultant force $\boldsymbol{F}$ and moment $\boldsymbol{M}$ are given as [13, 29]:

$$\boldsymbol{F} = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} -\left(T_2 \sin\vartheta_2 \cos\phi_2 + T_4 \sin\vartheta_4 \cos\phi_4\right) \\ T_2 \sin\phi_2 \cos\vartheta_2 + T_4 \sin\phi_4 \cos\vartheta_4 \\ -\left(T_1 + T_2 \cos\phi_2 \cos\vartheta_2 + T_3 + T_4 \cos\phi_4 \cos\vartheta_4\right) \end{bmatrix}, \tag{2.3.8}$$

$$\boldsymbol{M} = \begin{bmatrix} M_x \\ M_y \\ M_z \end{bmatrix} = \begin{bmatrix} T_2 d \cos\phi_2 \cos\vartheta_2 - T_4 d \cos\phi_4 \cos\vartheta_4 + Q_2 \sin\vartheta_2 \cos\phi_2 - Q_4 \sin\vartheta_4 \cos\phi_4 \\ \left(T_1 - T_3\right) d - Q_2 \sin\phi_2 \cos\vartheta_2 + Q_4 \sin\phi_4 \cos\vartheta_4 \\ -Q_1 + Q_2 \cos\phi_2 \cos\vartheta_2 + Q_3 - Q_4 \cos\phi_4 \cos\vartheta_4 - T_2 d \cos\phi_2 \sin\vartheta_2 + T_4 d \cos\phi_4 \sin\vartheta_4 \end{bmatrix}; \tag{2.3.9}$$

where d is the distance from the quadrotor geometrical center to the rotors, or replacing T_i and Q_i by their definitions in (2.3.6a) and (2.3.6b) respectively,

$$\boldsymbol{F} = K_T \begin{bmatrix} -\left(\omega_2^2 \sin\vartheta_2 \cos\phi_2 + \omega_4^2 \sin\vartheta_4 \cos\phi_4\right) \\ \omega_2^2 \sin\phi_2 \cos\vartheta_2 + \omega_4^2 \sin\phi_4 \cos\vartheta_4 \\ -\left(\omega_1^2 + \omega_2^2 \cos\phi_2 \cos\vartheta_2 + \omega_3^2 + \omega_4^2 \cos\phi_4 \cos\vartheta_4\right) \end{bmatrix}, \tag{2.3.10}$$

$$\boldsymbol{M} = \begin{bmatrix} \omega_2^2 \cos\phi_2 \left(dK_T \cos\vartheta_2 + K_Q \sin\vartheta_2\right) - \omega_4^2 \cos\phi_4 \left(dK_T \cos\vartheta_4 + K_Q \sin\vartheta_4\right) \\ dK_T \left(\omega_1^2 - \omega_3^2\right) - K_Q\omega_2^2 \cos\vartheta_2 \sin\phi_2 + K_Q\omega_4^2 \cos\vartheta_4 \sin\phi_4 \\ K_Q \left(\omega_3^2 - \omega_1^2\right) + \omega_2^2 \cos\phi_2 \left(K_Q \cos\vartheta_2 - dK_T \sin\vartheta_2\right) + \omega_4^2 \cos\phi_4 \left(dK_T \sin\vartheta_4 - K_Q \cos\vartheta_4\right) \end{bmatrix}, \tag{2.3.11}$$

where d is the distance from the quadrotor geometrical center to the rotors.

2.4 Equations of Motion

The ALIV-3 platform equations of motion can be separated into two parts: Kinematics and Dynamics.

2.4.1 Kinematics

Kinematics is the study of geometry of motion which is used to relate linear and angular position and velocity with no mention to origin of the motion [6].

The relative position of quadrotor to the inertial frame is:

$$\boldsymbol{P} = \begin{bmatrix} x & y & z \end{bmatrix}^T. \tag{2.4.1}$$

The velocity, relative to the body-fixed frame, is:

$$\boldsymbol{V} = \begin{bmatrix} u & v & w \end{bmatrix}^T. \tag{2.4.2}$$

The attitude, described by the *Euler* angles, of the platform is

$$\boldsymbol{\Phi} = \begin{bmatrix} \varphi & \theta & \psi \end{bmatrix}^T. \tag{2.4.3}$$

The angular velocity is defined by:

$$\boldsymbol{\Omega} = \begin{bmatrix} p & q & r \end{bmatrix}^T, \tag{2.4.4}$$

where p, q and r represent the roll, pitch and yaw rates, respectively, relative to the body-fixed frame.

These quantities relate to each other through relations depending on the attitude of the quadrotor.

Define the rotation matrix $\boldsymbol{S}$ from the body-fixed frame to the inertial frame as

$$\boldsymbol{S} = \begin{bmatrix} \cos\theta\cos\psi & \cos\psi\sin\theta\sin\varphi - \cos\varphi\sin\psi & \sin\varphi\sin\psi + \cos\varphi\cos\psi\sin\theta \\ \cos\theta\sin\psi & \cos\varphi\cos\psi + \sin\varphi\sin\theta\sin\psi & \cos\varphi\sin\theta\sin\psi - \cos\psi\sin\varphi \\ -\sin\theta & \cos\theta\sin\varphi & \cos\varphi\cos\theta \end{bmatrix}. \tag{2.4.5}$$

Note that $\boldsymbol{S}^{-1} = \boldsymbol{S}^T$, a property of rotation matrices.

The velocity relative to the inertial frame is related to the body-fixed frame velocity by

$$\dot{\boldsymbol{P}} = \boldsymbol{S}\,\boldsymbol{V}. \tag{2.4.6}$$

The roll, pitch and yaw rates relate to the *Euler* angles time-derivatives through a different kinematic relation:

$$\dot{\boldsymbol{\Phi}} = \boldsymbol{T}\,\boldsymbol{\Omega}, \tag{2.4.7}$$

where

$$\boldsymbol{T} = \begin{bmatrix} 1 & \sin\varphi\tan\theta & \cos\varphi\tan\theta \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi\sec\theta & \cos\varphi\sec\theta \end{bmatrix}. \tag{2.4.8}$$

2.4.2 Dynamics

Dynamics, opposed to kinematics, takes into account the mass and inertia of the body as well as the forces and moments applied to it to describe its motion.

When it comes to the linear motion dynamics, one can define it with relation to the inertial frame

$$\ddot{\boldsymbol{P}} = \frac{1}{m}\boldsymbol{S}\,\boldsymbol{F} + \boldsymbol{g}, \tag{2.4.9}$$

where

$$\boldsymbol{g} = \begin{bmatrix} 0 & 0 & g_0 \end{bmatrix}^T,$$

or to the body-fixed frame [6]

$$\dot{\boldsymbol{V}} = \frac{1}{m}\boldsymbol{F} - \boldsymbol{\Omega}\times\boldsymbol{V} + \boldsymbol{S}^T\boldsymbol{g}, \tag{2.4.10}$$

where $\boldsymbol{F}$ is composed by the three components of the forces applied to the quadrotor defined in (2.3.10), $g_0 = 9.81\,m/s^2$ is the Earth gravitational acceleration and $\boldsymbol{S}$ the rotation matrix from the body-fixed frame to the inertial frame, defined in (2.4.5).

The tilt-quadrotor is also subject to rotations around the three axis, described by *Euler* second law of motion

$$\boldsymbol{J}\dot{\boldsymbol{\Omega}} = \boldsymbol{M} - \boldsymbol{\Omega} \times (\boldsymbol{J}\,\boldsymbol{\Omega}) \tag{2.4.11}$$

where $\boldsymbol{M}$ is composed by the three components of the external moments applied to the platform defined in (2.3.11) and $\boldsymbol{J}$ is the ALIV-3 inertia matrix.

A simplified block diagram of the system can be found in figure 2.2, where the signal ”States” refers to

$$\left[\boldsymbol{\Phi}^T, \boldsymbol{P}_1^T, \boldsymbol{\Omega}^T, \boldsymbol{V}^T\right]^T,$$

”States Time Derivative” refers to

$$\left[\dot{\boldsymbol{\Omega}}^T, \dot{\boldsymbol{V}}^T\right]^T,$$

”Reference” and ”Reference Time Derivative” refer to the desired path for $\boldsymbol{\Phi}^T, \boldsymbol{P}_1^T$ to follow and its time-derivative, respectively. Note that each number reference in each block refers to a equation or set of equations previously defined. The ”Controllers Block” outputs the control signals for the actuators. The ”Rigid-body Dynamics” Block, explored in figure 2.3, includes the dynamics described in this section.

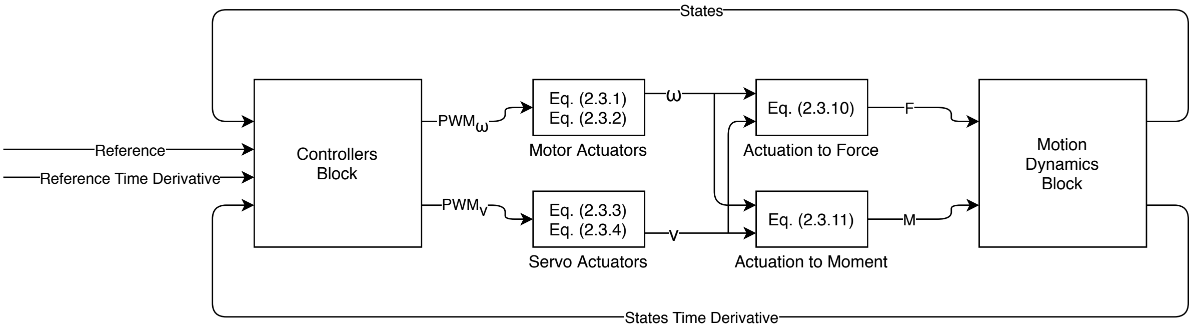

Figure 2.2: System Block Diagram

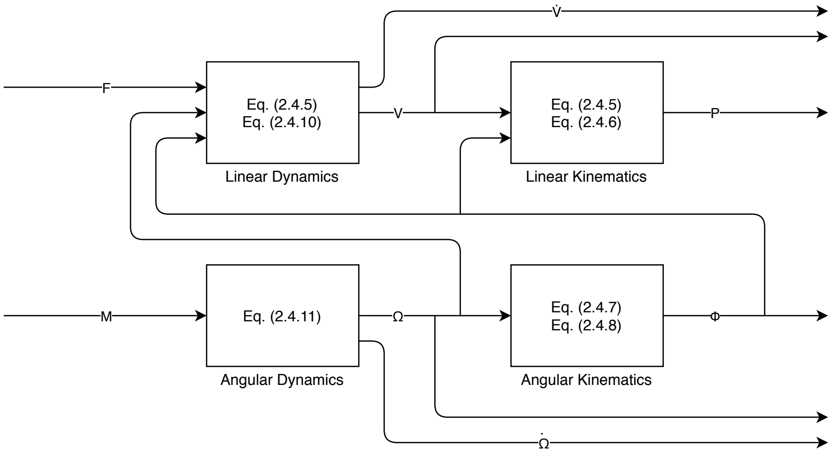

Figure 2.3: Motion Dynamics Block Diagram

Chapter 3

Control Design

In this work, two different methods are used for control design: Command Filtered Backstepping (BKS) and Command Filtered Incremental Backstepping (IBKS). A small introduction to these control design methods and to their theory foundations is included in section 3.1. In sections 3.2 and 3.3, these methods are applied to obtain control laws for the ALIV-3 platform model.

3.1 Background Theory

Command Filtered Backstepping control design and its incremental counterpart are based on Lyapunov theory as well as command filtering. This section includes a small introduction to the design methods and their foundations.

3.1.1 Lyapunov Theory

Stability is the main concept about system control and engineering. The Russian mathematician and engineer Alexandr Lyapunov introduced many of the fundamental concepts of stability in a doctoral thesis in the 19th century [27].

Lyapunov theory is used to describe the stability of dynamic systems. Consider a n-th order system with state vector $\boldsymbol{x}(t)$ described by

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}\left(\boldsymbol{x}(t)\right) \tag{3.1.1}$$

with an, possibly not the only one, equilibrium point $\boldsymbol{x}_0$, i.e. $\boldsymbol{f}\left(\boldsymbol{x}_0\right) = 0$ [5]. Assume there exists a scalar function $V\left(\boldsymbol{x}(t)\right)$ with the following properties:

- $V\left(\boldsymbol{x}(t)\right) > 0, \forall\, \boldsymbol{x}(t) \neq \boldsymbol{x}_0$;
- $V\left(\boldsymbol{x}(t)\right) \rightarrow \infty, \|\boldsymbol{x}(t)\| \rightarrow \infty$
- $\frac{d}{dt}\left[V\left(\boldsymbol{x}(t)\right)\right] = \frac{\partial V(\boldsymbol{x}(t))}{\partial \boldsymbol{x}(t)} \cdot \boldsymbol{f}\left(\boldsymbol{x}(t)\right) \leq 0.$

If $V(\boldsymbol{x}(t))$, which can be interpreted as a generalized distance from $\boldsymbol{x}(t)$ to $\boldsymbol{x}_0$, satisfies these conditions then the distance between the state vector and the equilibrium point used for the definition of $V(\boldsymbol{x}(t))$ decreases with time [20], i.e. the state vector tends to the equilibrium point.

In the sense of Lyapunov, stability has a broader definition. It is considered for an equilibrium point to be stable if, for a $\epsilon \geq 0$, there exists a $\delta(\epsilon, t_0) \geq 0$ such that

$$|\boldsymbol{x}(t_0) - \boldsymbol{x}_0| \leq \delta(\epsilon, t_0) \Rightarrow |\boldsymbol{x}(t) - \boldsymbol{x}_0| \leq \epsilon.$$

This may be interpreted as, if the distance between initial state and the equilibrium point is less than or equal to $\delta(\epsilon, t_0)$, then the distance between the state and the equilibrium point will always be smaller than or equal to some value ϵ, meaning that the distance may increase but will be bounded.

The equilibrium point may have additional conditions and is considered to be:

- Uniformly stable if δ does not depend on the initial time value t_0,
- Asymptotically stable if $|\boldsymbol{x}(t) - \boldsymbol{x}_0| \rightarrow 0$ as $t \rightarrow \infty$,
- Globally asymptotically stable if it fulfills the conditions for asymptotic stability for any initial state $\boldsymbol{x}(t_0)$.

For a more in-depth analysis see [5, 20].

A Lyapunov function is a function that satisfies these properties in a neighborhood of $\boldsymbol{x}_0$ which is used to assess the stability of a system [20] depending on the region of $\boldsymbol{x}(t)$ that makes $\dot{V}(\boldsymbol{x}(t)) < 0$.

Consider a system in the form:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), \boldsymbol{u}(t)) \tag{3.1.2}$$

where $\boldsymbol{u}(t)$ is the control input. Using Lyapunov theory, the control engineer can choose a $\boldsymbol{u}(\boldsymbol{x}(t))$ that makes the time-derivative of the chosen Lyapunov function, $V(\boldsymbol{x}(t), \boldsymbol{u}(\boldsymbol{x}(t)))$, negative semi-definite, stabilizing the system [20].

3.1.2 Backstepping

Backstepping is a Lyapunov-based method for nonlinear control design which is applied to $(n \cdot p)$-th order dynamic systems in the form:

$$\dot{\boldsymbol{x}}_i = \boldsymbol{f}_i(\boldsymbol{w}_i) + \boldsymbol{g}_i(\boldsymbol{w}_i) \cdot \boldsymbol{x}_{i+1},\ i = 1, .., n-1 \tag{3.1.3}$$

$$\dot{\boldsymbol{x}}_n = \boldsymbol{f}_n(\boldsymbol{x}) + \boldsymbol{g}_n(\boldsymbol{x}) \cdot \boldsymbol{u} \tag{3.1.4}$$

where t is omitted for simplicity as the input and all state vectors are time-dependant, $\boldsymbol{x}_i \in \mathbb{R}^p$ is the state vector of order i, $\boldsymbol{u} \in \mathbb{R}^m$, $\boldsymbol{x}^T = [\boldsymbol{x}_1^T, .., \boldsymbol{x}_n^T] \in \mathbb{R}^{np}$ and $\boldsymbol{w}_i^T = [\boldsymbol{x}_1^T, .., \boldsymbol{x}_i^T] \in \mathbb{R}^{ip}$. $\boldsymbol{f}_i$, $\boldsymbol{g}_i$, $\boldsymbol{f}_n$ and $\boldsymbol{g}_n$ are assumed to be known, bounded and differentiable.

This design method is recursive and each step can be divided into three parts [5, 21]:

1. Define the i-th step tracking errors $\boldsymbol{\zeta}_i = \boldsymbol{x}_i - \boldsymbol{x}_{i,des}$, where $\boldsymbol{x}_{i,des}$ is the desired path for $\boldsymbol{x}_i$, and their error dynamics;

2. Define a Lyapunov function candidate V_i containing the tracking error $\boldsymbol{\zeta}_i$;

3. Design a virtual stabilizing control law $\boldsymbol{x}_{i+1,des}$ that makes the time-derivative of the Lyapunov function, $\dot{V}_i$, negative (semi-)definite for $\boldsymbol{x}_{i+1} = \boldsymbol{x}_{i+1,des}$.

This systematic procedure starts by defining a error variable on $\boldsymbol{x}_i$ which is intended to be zero, later defining what path should the next state $\boldsymbol{x}_{i+1}$ follow so that the error variable is asymptotically stable. In the next step, it is assumed that $\boldsymbol{x}_{i+1}$ can't be always equal to $\boldsymbol{x}_{i+1,des}$, so a new error variable $\boldsymbol{\zeta}_{i+1}$ and the procedure continues.

This process is repeated for each step i from 1 to $n-1$. In the last step, the procedure is repeated once more but the stabilizing control law is designed directly for $\boldsymbol{u}$, the control input to the system. This model-based design method is model dependent and may be model sensitive. Further analysis on the design theory of Backstepping control can be found in [5, 18, 4].

3.1.3 Incremental Backstepping

Incremental Backstepping (IBKS) follows the assumptions that the control action is instantaneous and that its change produces changes in the states derivatives. For the attitude control case, ”for small time increments and fast actuators, a change in control input has a change in moment, which in turn directly affects angular accelerations.” [5, p. 135]

The IBKS controller design method is applied to systems in the same form as Backstepping with the added prerequisite that $\boldsymbol{g}_i$ must be smooth, i.e. exists higher than first order derivatives. Applying the *Taylor* expansion to any $\dot{\boldsymbol{x}}_i$ around a previous time instant, t_0, (reference) point $\boldsymbol{x}_{i+1} = \boldsymbol{x}_{i+1,0}$ and $\boldsymbol{w}_i = \boldsymbol{w}_{i,0}$:

$$\begin{aligned}\dot{\boldsymbol{x}}_i =& \boldsymbol{f}_i\left(\boldsymbol{w}_{i,0}\right) + \boldsymbol{g}_i\left(\boldsymbol{w}_{i,0}\right) \cdot \boldsymbol{x}_{i+1,0} \\ &+ \frac{\partial}{\partial \boldsymbol{w}_i}\left[\boldsymbol{f}_i(\boldsymbol{w}_i) + \boldsymbol{g}_i(\boldsymbol{w}_i) \cdot \boldsymbol{x}_{i+1}\right]\Bigg|_{\substack{\boldsymbol{w}_i = \boldsymbol{w}_{i,0} \\ \boldsymbol{x}_{i+1} = \boldsymbol{x}_{i+1,0}}} \cdot \left(\boldsymbol{w}_i - \boldsymbol{w}_{i,0}\right) \\ &+ \frac{\partial}{\partial \boldsymbol{x}_{i+1}}\left[\boldsymbol{f}_i(\boldsymbol{w}_i) + \boldsymbol{g}_i(\boldsymbol{w}_i) \cdot \boldsymbol{x}_{i+1}\right]\Bigg|_{\substack{\boldsymbol{w}_i = \boldsymbol{w}_{i,0} \\ \boldsymbol{x}_{i+1} = \boldsymbol{x}_{i+1,0}}} \cdot \left(\boldsymbol{x}_{i+1} - \boldsymbol{x}_{i+1,0}\right) \\ &+ \text{H.O.T.}\end{aligned} \tag{3.1.5}$$

where ”H.O.T.” stands for Higher Order Terms and, for sake of simplicity and notation $\boldsymbol{x}_{n+1}$ is the control input $\boldsymbol{u}$. For each integration step i the approximate subsystem is

$$\dot{\boldsymbol{x}}_i \approx \underbrace{\boldsymbol{f}_i\left(\boldsymbol{w}_{i,0}\right) + \boldsymbol{g}_i\left(\boldsymbol{w}_{i,0}\right) \cdot \boldsymbol{x}_{i+1,0}}_{\dot{\boldsymbol{x}}_{i,0}} + \underbrace{\frac{\partial}{\partial\, \boldsymbol{w}_i}\left[\boldsymbol{f}_i(\boldsymbol{w}_i) + \boldsymbol{g}_i(\boldsymbol{w}_i) \cdot \boldsymbol{x}_{i+1}\right]\Bigg|_{\substack{\boldsymbol{w}_i = \boldsymbol{w}_{i,0} \\ \boldsymbol{x}_{i+1} = \boldsymbol{x}_{i+1,0}}}}_{\boldsymbol{A}_{i,0}} \cdot (\boldsymbol{w}_i - \boldsymbol{w}_{i,0}) + \underbrace{\boldsymbol{g}_i(\boldsymbol{w}_{i,0})}_{\boldsymbol{B}_{i,0}} \cdot (\boldsymbol{x}_{i+1} - \boldsymbol{x}_{i+1,0})$$

$$= \dot{\boldsymbol{x}}_{i,0} + \boldsymbol{A}_{i,0} \cdot \Delta\boldsymbol{w}_i + \boldsymbol{B}_{i,0} \cdot \Delta\boldsymbol{x}_{i+1,0} \tag{3.1.6}$$

where

$$\Delta\boldsymbol{w}_i = \boldsymbol{w}_i - \boldsymbol{w}_{i,0};$$

$$\Delta\boldsymbol{x}_{i+1} = \boldsymbol{x}_{i+1} - \boldsymbol{x}_{i+1,0}.$$

In a small neighborhood of the reference state, the nonlinear system can be approximated by its linearization about that reference state. For small time increments and sufficiently fast control update, $\Delta\, \boldsymbol{w}_i$ is negligible, allowing the subsystem to be represented as:

$$\dot{\boldsymbol{x}}_i \approx \dot{\boldsymbol{x}}_{i,0} + \boldsymbol{B}_{i,0} \cdot \Delta\boldsymbol{x}_{i+1} \tag{3.1.7}$$

Note that the term on $\boldsymbol{f}_i\left(\boldsymbol{w}_i\right)$ was dropped and that $\dot{\boldsymbol{x}}_{i,0}$ may be measured (directly or indirectly), which makes this control design method less dependent on the model and very advantageous because, effectively, $\boldsymbol{f}_i\left(\boldsymbol{w}_i\right)$ may not be known. However, $\boldsymbol{g}_i\left(\boldsymbol{w}_i\right)$ must still be known.

After obtaining the intended approximated subsystem, the procedure is analogous to the BKS design method, designing $\Delta\, \boldsymbol{x}_{i+1,des}$ and incrementing it with $\boldsymbol{x}_{i+1,0}$ instead of designing $\boldsymbol{x}_{i+1,des}$ directly, i.e.

$$\boldsymbol{x}_{i+1} = \boldsymbol{x}_{i+1,0} + \Delta\, \boldsymbol{x}_{i+1,des},$$

which intrinsically gives integrative properties to this design method.

3.1.4 Command Filtering

The issue with using the backstepping or its incremental counterpart control design methods ”for an n-th order system is that the desired output and its first n derivatives must be available for use in the control law implementation” [18, section I]. This case is extended to the virtual stabilizing control laws $\boldsymbol{x}_{i+1,des}$, whose first $n-i$ time-derivatives must also be available.

Command filtering serves the purpose of filtering the influence of measurement noise on the control law as well as avoiding the, rather complicated, analytical computation of the time-derivatives of the virtual control laws $\boldsymbol{x}_{i+1,des}$ for $i = 1,..,n-1$ because the control signal $\boldsymbol{u}$ includes the time-derivative of $\boldsymbol{x}_{n,des}$, which needs the second time-derivative of $\boldsymbol{x}_{n-1,des}$ and so on.

This section will be restricted to first and second order filters. Further analysis on command filtering can be found in [18].

Let the state-space form of a second-order filter be introduced, whose input is the desired path

$\boldsymbol{x}_{des}(t)$ for the state $\boldsymbol{x}(t)$ as:

$$\begin{aligned} \dot{\boldsymbol{\chi}}_1(t) &= \boldsymbol{\chi}_2(t) \\ \dot{\boldsymbol{\chi}}_2(t) &= -2\xi\omega_n\boldsymbol{\chi}_2(t) - \omega_n^2\left(\boldsymbol{\chi}_1(t) - \boldsymbol{x}_{des}(t)\right) \\ \boldsymbol{x}_{des,f}(t) &= \boldsymbol{\chi}_1(t) \\ \dot{\boldsymbol{x}}_{des,f}(t) &= \boldsymbol{\chi}_2(t) \\ \boldsymbol{\chi}_1(0) &= \boldsymbol{x}_{des}(0) \\ \boldsymbol{\chi}_2(0) &= 0 \end{aligned} \tag{3.1.8}$$

where $\boldsymbol{\chi}_1$ and $\boldsymbol{\chi}_2$ are the filter states, ω_n is the filter natural frequency, ξ is the filter damping coefficient, $\boldsymbol{x}_{des,f}(t)$ is the filtered desired path for $\boldsymbol{x}(t)$ and $\dot{\boldsymbol{x}}_{des,f}(t)$ its time-derivative. Note that $\dot{\boldsymbol{x}}_{des,f}(t)$ is obtained without the need to differentiate $\boldsymbol{x}_{des,f}(t)$ analytically.

The control engineer must now choose the constants $\omega_n > 0$ and $\xi > 0$ to specify the desired bandwidth and transient response of the system. The higher the ω_n, the closer $\boldsymbol{x}_{des,f}(t)$ is to $\boldsymbol{x}_{des}(t)$ as well as their time-derivatives, sacrificing the low-pass filtering property. In this work, the ξ constant should be chosen greater than 1 to avoid filter response overshoot but low enough not to slow down the system transient response.

First-order command filters, opposed to second-order, do not output the derivative of the signal but are also low-pass filters. Let the space-state form of a first-order filter be introduced, whose input is the desired path $\boldsymbol{x}_{des}(t)$ for the state $\boldsymbol{x}(t)$ as:

$$\begin{aligned} \dot{\boldsymbol{\chi}}(t) &= -\frac{1}{\tau}\left(\boldsymbol{\chi}(t) - \boldsymbol{x}_{des}(t)\right) \\ \boldsymbol{x}_{des,f}(t) &= \boldsymbol{\chi}(t) \\ \boldsymbol{\chi}(0) &= \boldsymbol{x}_{des}(0) \end{aligned} \tag{3.1.9}$$

where $\boldsymbol{\chi}$ is the filter state and τ is the filter time constant. The control engineer only needs to choose a suitable large enough τ in order to filter the high-frequency noise but small enough not to slow down the system transient response.

3.2 Backstepping Implementation

In this section, a controller for the model described in section 2.4 will be designed by applying the Backstepping control design method. For controller design, it is assumed that the actuators do not have dynamics due to their relative small response time constants.

The ALIV3 system model presents the necessary characteristics in order to apply the BKS method, as it can be written as

$$\begin{bmatrix} \dot{\boldsymbol{\Phi}} \\ \dot{\boldsymbol{P}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{T} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \boldsymbol{S} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Omega} \\ \boldsymbol{V} \end{bmatrix} \tag{3.2.1}$$

$$\begin{bmatrix} \dot{\boldsymbol{\Omega}} \\ \dot{\boldsymbol{V}} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{J}^{-1}\left(\boldsymbol{\Omega} \times (\boldsymbol{J}\boldsymbol{\Omega})\right) \\ -\boldsymbol{\Omega} \times \boldsymbol{V} + \boldsymbol{S}^T \cdot \boldsymbol{g} \end{bmatrix} + \begin{bmatrix} \boldsymbol{J}^{-1} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & m^{-1} \cdot \boldsymbol{I}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{M} \\ \boldsymbol{F} \end{bmatrix}. \tag{3.2.2}$$

The dynamics above is equivalent to:

$$\dot{\boldsymbol{s}}_1 = \boldsymbol{g}_1(\boldsymbol{s}_1)\boldsymbol{s}_2 \tag{3.2.3}$$

$$\dot{\boldsymbol{s}}_2 = \boldsymbol{f}_2(\boldsymbol{s}_1, \boldsymbol{s}_2) + \boldsymbol{g}_2\boldsymbol{N} \tag{3.2.4}$$

which is in the form of (3.1.3) and (3.1.4) for

$$\begin{aligned} \boldsymbol{s}_1 &= \begin{bmatrix} \boldsymbol{\Phi} \\ \boldsymbol{P} \end{bmatrix}; \\ \boldsymbol{s}_2 &= \begin{bmatrix} \boldsymbol{\Omega} \\ \boldsymbol{V} \end{bmatrix}; \\ \boldsymbol{g}_1(\boldsymbol{s}_1) &= \begin{bmatrix} \boldsymbol{T} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \boldsymbol{S} \end{bmatrix}; \\ \boldsymbol{f}_2(\boldsymbol{s}_1, \boldsymbol{s}_2) &= \begin{bmatrix} -\boldsymbol{J}^{-1}\left(\boldsymbol{\Omega} \times (\boldsymbol{J}\boldsymbol{\Omega})\right) \\ -\boldsymbol{\Omega} \times \boldsymbol{V} + \boldsymbol{S}^T \cdot \boldsymbol{g} \end{bmatrix}; \\ \boldsymbol{g}_2 &= \begin{bmatrix} \boldsymbol{J}^{-1} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & m^{-1} \cdot \boldsymbol{I}_3 \end{bmatrix}; \\ \boldsymbol{N} &= \begin{bmatrix} \boldsymbol{M} \\ \boldsymbol{F} \end{bmatrix}. \end{aligned}$$

Note that $\boldsymbol{f}_1 \equiv \mathbf{0}_{6\times1}$, $\boldsymbol{g}_2$ is constant and t was suppressed for notation simplicity. This model contains 12 states. However, it is composed by two loops: the kinematics, $\boldsymbol{s}_1$, (outer) loop and the dynamics, $\boldsymbol{s}_2$, (inner) loop which means that the Backstepping design method is composed of two steps.

3.2.1 Kinematics Loop

As aforementioned, in section 3.1.2, the first part of each step is to define the tracking error

$$\boldsymbol{\zeta}_1 = \begin{bmatrix} \boldsymbol{\zeta}_\Phi \\ \boldsymbol{\zeta}_p \end{bmatrix} = \boldsymbol{s}_1 - \boldsymbol{s}_{1,des} = \begin{bmatrix} \boldsymbol{\Phi} \\ \boldsymbol{P} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\Phi}_{des} \\ \boldsymbol{P}_{des} \end{bmatrix}, \tag{3.2.5}$$

where $\boldsymbol{\Phi}_{des}$ and $\boldsymbol{P}_{des}$ are the reference attitude and position, respectively. $\boldsymbol{\zeta}_1$ dynamics is

$$\dot{\boldsymbol{\zeta}}_1 = \boldsymbol{g}_1(\boldsymbol{s}_1)\boldsymbol{s}_2 - \dot{\boldsymbol{s}}_{1,des} \tag{3.2.6}$$

The second part of the first step is to define a Lyapunov function V_1 containing $\boldsymbol{\zeta}_1$. In order to avoid steady-state error due to model errors, the Lyapunov function candidate will be augmented with an integrative term of $\boldsymbol{\zeta}_1$, i.e.

$$V_1 = \frac{1}{2}\boldsymbol{\zeta}_1^T\boldsymbol{\zeta}_1 + \frac{1}{2}\boldsymbol{\gamma}^T\boldsymbol{K}_i\boldsymbol{\gamma}, \tag{3.2.7}$$

where $\boldsymbol{K}_i$ is a diagonal matrix whose elements are positive. Note that if there are large model uncertainties the integrative term may introduce a wind-up problem. However, for smaller uncertainties, it may be beneficial to counteract them. The integrative term is defined as:

$$\boldsymbol{\gamma} = \begin{bmatrix} \gamma_\Phi \\ \gamma_p \end{bmatrix} = \int_0^t \boldsymbol{\zeta}_1 \, dt. \tag{3.2.8}$$

Equation (3.2.7) may be rewritten as:

$$V_1 = \frac{1}{2}\boldsymbol{\zeta}_\Phi^T\boldsymbol{\zeta}_\Phi + \frac{1}{2}\boldsymbol{\gamma}_\Phi^T\boldsymbol{K}_{\Phi i}\boldsymbol{\gamma}_\Phi + \frac{1}{2}\boldsymbol{\zeta}_p^T\boldsymbol{\zeta}_p + \frac{1}{2}\boldsymbol{\gamma}_p^T\boldsymbol{K}_{pi}\boldsymbol{\gamma}_p, \tag{3.2.9}$$

with

$$\boldsymbol{K}_i = \begin{bmatrix} \boldsymbol{K}_{\Phi i} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{K}_{pi} \end{bmatrix}.$$

This Lyapunov function candidate fulfills the conditions in section 3.1.1, as

$$V_1(\boldsymbol{\zeta}_1, \boldsymbol{\gamma}) \to \infty,\ (\boldsymbol{\zeta}_1, \boldsymbol{\gamma}) \to \infty$$

$$V_1 > 0,\ \text{for } \boldsymbol{\zeta}_1 \neq \boldsymbol{0} \text{ and } \boldsymbol{\gamma} \neq \boldsymbol{0}$$

The third stage is to design a virtual stabilizing control law, $\boldsymbol{s}_{2,des}$, that makes $\dot{V}_1 < 0$ for $\boldsymbol{\zeta}_1 \neq \boldsymbol{0}$ when $\boldsymbol{s}_2 = \boldsymbol{s}_{2,des}$.

$$\begin{aligned}
\dot{V}_1 &= \boldsymbol{\zeta}_1^T\left(\boldsymbol{g}_1(\boldsymbol{s}_1)\boldsymbol{s}_2 - \dot{\boldsymbol{s}}_{1,des}\right) \\
&= \boldsymbol{\zeta}_\Phi^T\left(\boldsymbol{T}\boldsymbol{\Omega} - \dot{\boldsymbol{\Phi}}_{des}\right) + \boldsymbol{\zeta}_\Phi^T\boldsymbol{K}_{\Phi i}\boldsymbol{\gamma}_\Phi + \boldsymbol{\zeta}_p^T\left(\boldsymbol{S}\boldsymbol{V} - \dot{\boldsymbol{P}}_{des}\right) + \boldsymbol{\zeta}_p^T\boldsymbol{K}_{pi}\boldsymbol{\gamma}_p \\
&= \boldsymbol{\zeta}_\Phi^T\left(\boldsymbol{T}\boldsymbol{\Omega} - \dot{\boldsymbol{\Phi}}_{des} + \boldsymbol{K}_{\Phi i}\boldsymbol{\gamma}_\Phi\right) + \boldsymbol{\zeta}_p^T\left(\boldsymbol{S}\boldsymbol{V} - \dot{\boldsymbol{P}}_{des} + \boldsymbol{K}_{pi}\boldsymbol{\gamma}_p\right)
\end{aligned} \tag{3.2.10}$$

If $\boldsymbol{s}_{2,des} = \begin{bmatrix} \boldsymbol{\Omega}_{des}^T & \boldsymbol{V}_{des}^T \end{bmatrix}^T$, which will be command filtered later, is chosen as

$$\boldsymbol{\Omega}_{des} = \boldsymbol{T}^{-1}\left(-\boldsymbol{K}_\Phi\boldsymbol{\zeta}_\Phi - \boldsymbol{K}_{\Phi i}\boldsymbol{\gamma}_\Phi + \dot{\boldsymbol{\Phi}}_{des}\right) \tag{3.2.11}$$

$$\boldsymbol{V}_{des} = \boldsymbol{S}^T\left(-\boldsymbol{K}_p\boldsymbol{\zeta}_p - \boldsymbol{K}_{pi}\boldsymbol{\gamma}_p + \dot{\boldsymbol{P}}_{des}\right) \tag{3.2.12}$$

where $\boldsymbol{K}_\Phi$ and $\boldsymbol{K}_p$ are positive-definite matrices, then, when $\boldsymbol{s}_2 = \boldsymbol{s}_{2,des}$,

$$\begin{aligned}\dot{V}_1 &= \boldsymbol{\zeta}_\Phi^T \left(\boldsymbol{T} \cdot \boldsymbol{T}^{-1} \left(-\boldsymbol{K}_\Phi \boldsymbol{\zeta}_\Phi - \boldsymbol{K}_{\Phi i} \boldsymbol{\gamma}_\Phi + \dot{\boldsymbol{\Phi}}_{des}\right) - \dot{\boldsymbol{\Phi}}_{des} + \boldsymbol{K}_{\Phi i} \boldsymbol{\gamma}_\Phi\right) \\ &+ \boldsymbol{\zeta}_p^T \left(\boldsymbol{S} \cdot \boldsymbol{S}^T \left(-\boldsymbol{K}_p \boldsymbol{\zeta}_p - \boldsymbol{K}_{pi} \boldsymbol{\gamma}_p + \dot{\boldsymbol{P}}_{des}\right) - \dot{\boldsymbol{P}}_{des} + \boldsymbol{K}_{pi} \boldsymbol{\gamma}_p\right) \\ &= -\boldsymbol{\zeta}_\Phi^T \boldsymbol{K}_\Phi \boldsymbol{\zeta}_\Phi - \boldsymbol{\zeta}_p^T \boldsymbol{K}_p \boldsymbol{\zeta}_p \\ &= -\boldsymbol{\zeta}_1^T \boldsymbol{K}_1 \boldsymbol{\zeta}_1 \leq 0\end{aligned}$$

where

$$\boldsymbol{K}_1 = \begin{bmatrix} \boldsymbol{K}_\Phi & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{K}_p \end{bmatrix}.$$

Note that the $\det(\boldsymbol{T}) = \sec(\theta) \neq 0$ but arises problems when $\theta = \frac{\pi}{2} + k\pi,\ k \in \mathbb{Z}$. It could have been avoided by, for example, using the quaternion angle representation instead of the Euler angles but it should be pointed that it is not expected for the platform to reach pitch attitudes near $\frac{\pi}{2}$ radians (or 90 degrees), meaning that the singularities will be avoided and $\boldsymbol{\Omega}_{des}$ can be defined as above. $\boldsymbol{T}^{-1}$ is defined analytically as

$$\boldsymbol{T}^{-1} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\varphi & \cos\theta\sin\varphi \\ 0 & -\sin\varphi & \cos\varphi\cos\theta \end{bmatrix} \tag{3.2.13}$$

3.2.2 Dynamics Loop

Similarly to the kinematics loop, the dynamics loop control design starts by defining the tracking error

$$\boldsymbol{\zeta}_2 = \begin{bmatrix} \boldsymbol{\zeta}_\Omega \\ \boldsymbol{\zeta}_v \end{bmatrix} = \boldsymbol{s}_2 - \boldsymbol{s}_{2,des,f} = \begin{bmatrix} \boldsymbol{\Omega} \\ \boldsymbol{V} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\Omega}_{des,f} \\ \boldsymbol{V}_{des,f} \end{bmatrix}, \tag{3.2.14}$$

where $\boldsymbol{\Omega}_{des,f}$ and $\boldsymbol{V}_{des,f}$ are the command filtered versions of $\boldsymbol{\Omega}_{des}$ and $\boldsymbol{V}_{des}$ respectively. The dynamics of $\boldsymbol{\zeta}_2$ are

$$\dot{\boldsymbol{\zeta}}_2 = \boldsymbol{f}_2(\boldsymbol{s}_1, \boldsymbol{s}_2) + \boldsymbol{g}_2 \boldsymbol{N} - \dot{\boldsymbol{s}}_{2,des,f}. \tag{3.2.15}$$

As can be seen in (3.2.15), the time-derivative of the desired angular and linear velocities is present in the tracking error $\boldsymbol{\zeta}_2$ dynamics.

With $\boldsymbol{\zeta}_2$ defined, (3.2.10) can now be rewritten substituting $\boldsymbol{\Omega}$ by $(\boldsymbol{\zeta}_\Omega + \boldsymbol{\Omega}_{des,f} + \boldsymbol{\Omega}_{des} - \boldsymbol{\Omega}_{des})$ and $\boldsymbol{V}$

by $(\boldsymbol{\zeta}_v + \boldsymbol{V}_{des,f} + \boldsymbol{V}_{des} - \boldsymbol{V}_{des})$

$$\begin{aligned}
\dot{V}_1 &= \boldsymbol{\zeta}_\Phi^T \left(\boldsymbol{T}\left(\boldsymbol{\zeta}_\Omega + \boldsymbol{\Omega}_{des,f} + \boldsymbol{\Omega}_{des} - \boldsymbol{\Omega}_{des}\right) - \dot{\boldsymbol{\Phi}}_{des} + \boldsymbol{K}_{\Phi i}\boldsymbol{\gamma}_\Phi\right) \\
&+ \boldsymbol{\zeta}_p^T \left(\boldsymbol{S}\left(\boldsymbol{\zeta}_v + \boldsymbol{V}_{des,f} + \boldsymbol{V}_{des} - \boldsymbol{V}_{des}\right) - \dot{\boldsymbol{P}}_{des} + \boldsymbol{K}_{pi}\boldsymbol{\gamma}_p\right) \\
&= \boldsymbol{\zeta}_\Phi^T \left(\boldsymbol{T}\,\boldsymbol{\Omega}_{des} - \dot{\boldsymbol{\Phi}}_{des} + \boldsymbol{K}_{\Phi i}\boldsymbol{\gamma}_\Phi\right) + \boldsymbol{\zeta}_\Phi^T\boldsymbol{T}\boldsymbol{\zeta}_\Omega + \boldsymbol{\zeta}_\Phi^T\boldsymbol{T}\left(\boldsymbol{\Omega}_{des,f} - \boldsymbol{\Omega}_{des}\right) \\
&+ \boldsymbol{\zeta}_p^T \left(\boldsymbol{S}\boldsymbol{V}_{des} - \dot{\boldsymbol{P}}_{des} + \boldsymbol{K}_{pi}\boldsymbol{\gamma}_p\right) + \boldsymbol{\zeta}_p^T\boldsymbol{S}\boldsymbol{\zeta}_v + \boldsymbol{\zeta}_p^T\boldsymbol{S}\left(\boldsymbol{V}_{des,f} - \boldsymbol{V}_{des}\right) \\
&= \underbrace{-\boldsymbol{\zeta}_1^T\boldsymbol{K}_1\boldsymbol{\zeta}_1}_{\leq 0} + \boldsymbol{\zeta}_1^T\boldsymbol{g}_1\left(\boldsymbol{s}_1\right)\boldsymbol{\zeta}_2 + \boldsymbol{\zeta}_1^T\boldsymbol{g}_1\left(\boldsymbol{s}_1\right)\left(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des}\right)
\end{aligned} \tag{3.2.16}$$

The second stage is to define a Lyapunov function candidate

$$V_2 = V_1 + \frac{1}{2}\boldsymbol{\zeta}_2^T\boldsymbol{\zeta}_2 \tag{3.2.17}$$

This Lyapunov function candidate fulfills the conditions in section 3.1.1, as

$$V_2(\boldsymbol{\zeta}_2, \boldsymbol{\zeta}_1, \boldsymbol{\gamma}) \to \infty, \ (\boldsymbol{\zeta}_2, \boldsymbol{\zeta}_1, \boldsymbol{\gamma}) \to \infty$$

$$V_2 > 0, \text{ for } \boldsymbol{\zeta}_2 \neq \boldsymbol{0}, \ \boldsymbol{\zeta}_1 \neq \boldsymbol{0} \text{ and } \boldsymbol{\gamma} \neq \boldsymbol{0}$$

The third stage is to design a virtual stabilizing control law, $\boldsymbol{N}_{des}$

$$\begin{aligned}
\dot{V}_2 &= \dot{V}_1 + \boldsymbol{\zeta}_2^T\left(\boldsymbol{f}_2(\boldsymbol{s}_1, \boldsymbol{s}_2) + \boldsymbol{g}_2\boldsymbol{N} - \dot{\boldsymbol{s}}_{2,des,f}\right) \\
&= -\boldsymbol{\zeta}_1^T\boldsymbol{K}_1\boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_1^T\boldsymbol{g}_1(\boldsymbol{s}_1)\boldsymbol{\zeta}_2 + \boldsymbol{\zeta}_1^T\boldsymbol{g}_1(\boldsymbol{s}_1)\left(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des}\right) + \boldsymbol{\zeta}_2^T\left(\boldsymbol{f}_2(\boldsymbol{s}_1, \boldsymbol{s}_2) + \boldsymbol{g}_2\boldsymbol{N} - \dot{\boldsymbol{s}}_{2,des,f}\right) \\
&= -\boldsymbol{\zeta}_1^T\boldsymbol{K}_1\boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_1^T\boldsymbol{g}_1(\boldsymbol{s}_1)\left(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des}\right) + \boldsymbol{\zeta}_2^T\left(\boldsymbol{f}_2(\boldsymbol{s}_1, \boldsymbol{s}_2) + \boldsymbol{g}_1^T(\boldsymbol{s}_1)\boldsymbol{\zeta}_1 + \boldsymbol{g}_2\boldsymbol{N} - \dot{\boldsymbol{s}}_{2,des,f}\right) \\
&= -\boldsymbol{\zeta}_1^T\boldsymbol{K}_1\boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_1^T\boldsymbol{g}_1(\boldsymbol{s}_1)\left(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des}\right) \\
&+ \boldsymbol{\zeta}_\Omega^T\left(-\boldsymbol{J}^{-1}\left(\boldsymbol{\Omega} \times (\boldsymbol{J}\boldsymbol{\Omega})\right) + \boldsymbol{T}^T\boldsymbol{\zeta}_\Phi + \boldsymbol{J}^{-1}\boldsymbol{M} - \dot{\boldsymbol{\Omega}}_{des,f}\right) \\
&+ \boldsymbol{\zeta}_v^T\left(-\boldsymbol{\Omega} \times \boldsymbol{V} + \boldsymbol{S}^T\boldsymbol{g} + \boldsymbol{S}^T\boldsymbol{\zeta}_p + \frac{1}{m}\boldsymbol{F} - \dot{\boldsymbol{V}}_{des,f}\right)
\end{aligned} \tag{3.2.18}$$

In order to guarantee stability, one needs to compensate the difference $(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des})$ with $\boldsymbol{\zeta}_{1,c}$ [18], defined by

$$\begin{aligned}
\dot{\boldsymbol{\zeta}}_{1,c} &= -\boldsymbol{K}_1\boldsymbol{\zeta}_{1,c} - \boldsymbol{K}_i\boldsymbol{\gamma} + \boldsymbol{g}_1(\boldsymbol{s}_1)\left(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des}\right) \\
\boldsymbol{\zeta}_{1,c}(t = 0) &= \boldsymbol{0}
\end{aligned} \tag{3.2.19}$$

Let $\boldsymbol{N}_{des}$ be defined as

$$\boldsymbol{N}_{des} = \begin{bmatrix} \boldsymbol{M}_{des} \\ \boldsymbol{F}_{des} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Omega} \times (\boldsymbol{J}\boldsymbol{\Omega}) + \boldsymbol{J}\left(\dot{\boldsymbol{\Omega}}_{des,f} - \boldsymbol{K}_\Omega\boldsymbol{\zeta}_\Omega\right) \\ m \cdot \left(\dot{\boldsymbol{V}}_{des,f} + \boldsymbol{\Omega} \times \boldsymbol{V} - \boldsymbol{S}^T\boldsymbol{g} - \boldsymbol{K}_v\boldsymbol{\zeta}_v\right) \end{bmatrix} - \boldsymbol{g}_2^{-1} \cdot \boldsymbol{g}_1^T(\boldsymbol{s}_1)(\boldsymbol{\zeta}_1 - \boldsymbol{\zeta}_{1,c}), \tag{3.2.20}$$

where $\boldsymbol{K}_\Omega$ and $\boldsymbol{K}_v$ are positive definite matrices that compose

$$\boldsymbol{K}_2 = \begin{bmatrix} \boldsymbol{K}_\Omega & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{K}_v \end{bmatrix}.$$

Substituting $\boldsymbol{N}$ by $\boldsymbol{N}_{des}$ in (3.2.18)

$$\begin{aligned} \dot{V}_2 &= -\boldsymbol{\zeta}_1^T \boldsymbol{K}_1 \boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_1^T \boldsymbol{g}_1(\boldsymbol{s}_1)\left(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des}\right) \\ &\quad + \boldsymbol{\zeta}_\Omega^T \left(-\boldsymbol{K}_\Omega \boldsymbol{\zeta}_\Omega + \boldsymbol{T}^T \boldsymbol{\zeta}_\Phi\right) + \boldsymbol{\zeta}_v^T \boldsymbol{K}_v \left(-\boldsymbol{K}_v \boldsymbol{\zeta}_v + \boldsymbol{S}^T \boldsymbol{\zeta}_p\right) \\ &\quad - \boldsymbol{\zeta}_2^T \boldsymbol{g}_1^T(\boldsymbol{s}_1)(\boldsymbol{\zeta}_1 - \boldsymbol{\zeta}_{1,c}) \\ &= -\boldsymbol{\zeta}_1^T \boldsymbol{K}_1 \boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_1^T \boldsymbol{g}_1(\boldsymbol{s}_1)\left(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des}\right) \\ &\quad - \boldsymbol{\zeta}_2^T \boldsymbol{K}_2 \boldsymbol{\zeta}_2 + \boldsymbol{\zeta}_2^T \boldsymbol{g}_1^T(\boldsymbol{s}_1)\boldsymbol{\zeta}_1 - \boldsymbol{\zeta}_2^T \boldsymbol{g}_1^T(\boldsymbol{s}_1)(\boldsymbol{\zeta}_1 - \boldsymbol{\zeta}_{1,c}) \\ &= -\boldsymbol{\zeta}_1^T \boldsymbol{K}_1 \boldsymbol{\zeta}_1 - \boldsymbol{\zeta}_2^T \boldsymbol{K}_2 \boldsymbol{\zeta}_2 \\ &\quad + \boldsymbol{\zeta}_1^T \boldsymbol{g}_1(\boldsymbol{s}_1)\left(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des}\right) + \boldsymbol{\zeta}_{1,c}^T \boldsymbol{g}_1(\boldsymbol{s}_1)\boldsymbol{\zeta}_2 \end{aligned}$$

which does not appear to be negative (semi-)definite. However, $\boldsymbol{\zeta}_{1,c}$ is introduced due to the difference between $\boldsymbol{s}_{2,des}$ and its command filtered version. Additionally, the control law satisfies the conditions for system stability, more information can be found in [18].

The tracking error $\boldsymbol{\zeta}_1$ dynamics is obtained by substituting $\boldsymbol{s}_2$ by $(\boldsymbol{\zeta}_2 + \boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des} + \boldsymbol{s}_{2,des})$ in (3.2.6):

$$\dot{\boldsymbol{\zeta}}_1 = \boldsymbol{g}_1(\boldsymbol{s}_1)\boldsymbol{\zeta}_2 + \boldsymbol{g}_1(\boldsymbol{s}_1)\left(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des}\right) + \boldsymbol{g}_1(\boldsymbol{s}_1)\boldsymbol{s}_{2,des} - \dot{\boldsymbol{s}}_{1,des}$$

substituting $\boldsymbol{s}_{2,des}$ in the third term by the definition in (3.2.11) and (3.2.12),

$$\begin{aligned} \dot{\boldsymbol{\zeta}}_1 &= \boldsymbol{g}_1(\boldsymbol{s}_1)\boldsymbol{\zeta}_2 + \boldsymbol{g}_1(\boldsymbol{s}_1)\left(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des}\right) + \boldsymbol{g}_1(\boldsymbol{s}_1)\left[\boldsymbol{g}_1^{-1}(\boldsymbol{s}_1)\left(-\boldsymbol{K}_1\boldsymbol{\zeta}_1 - \boldsymbol{K}_i\boldsymbol{\gamma} + \dot{\boldsymbol{s}}_{1,des}\right)\right] - \dot{\boldsymbol{s}}_{1,des} \\ &= -\boldsymbol{K}_1\boldsymbol{\zeta}_1 - \boldsymbol{K}_i\boldsymbol{\gamma} + \boldsymbol{g}_1(\boldsymbol{s}_1)\boldsymbol{\zeta}_2 + \boldsymbol{g}_1(\boldsymbol{s}_1)\left(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des}\right). \end{aligned}$$

The tracking error $\boldsymbol{\zeta}_2$ dynamics is obtained by substituting $\boldsymbol{N}$ in (3.2.15) by $\boldsymbol{N}_{des}$ defined in (3.2.20):

$$\begin{aligned} \dot{\boldsymbol{\zeta}}_2 &= \boldsymbol{f}_2(\boldsymbol{s}_1, \boldsymbol{s}_2) + \boldsymbol{g}_2\left[\boldsymbol{g}_2^{-1}\left(-\boldsymbol{f}_2(\boldsymbol{s}_1, \boldsymbol{s}_2) - \boldsymbol{K}_2\boldsymbol{\zeta}_2 - \boldsymbol{g}_1^T(\boldsymbol{s}_1)(\boldsymbol{\zeta}_1 - \boldsymbol{\zeta}_{1,c}) + \dot{\boldsymbol{s}}_{2,des,f}\right)\right] - \dot{\boldsymbol{s}}_{2,des,f} \\ &= -\boldsymbol{K}_2\boldsymbol{\zeta}_2 - \boldsymbol{g}_1^T(\boldsymbol{s}_1)(\boldsymbol{\zeta}_1 - \boldsymbol{\zeta}_{1,c}). \end{aligned}$$

The total tracking error dynamics is then

$$\dot{\boldsymbol{\gamma}} = \boldsymbol{\zeta}_1 \qquad (3.2.21)$$

$$\dot{\boldsymbol{\zeta}}_1 = -\boldsymbol{K}_1\boldsymbol{\zeta}_1 - \boldsymbol{K}_i\boldsymbol{\gamma} + \boldsymbol{g}_1(\boldsymbol{s}_1)\boldsymbol{\zeta}_2 + \boldsymbol{g}_1(\boldsymbol{s}_1)\left(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des}\right) \qquad (3.2.22)$$

$$\dot{\boldsymbol{\zeta}}_2 = -\boldsymbol{K}_2\boldsymbol{\zeta}_2 - \boldsymbol{g}_1^T(\boldsymbol{s}_1)\left(\boldsymbol{\zeta}_1 - \boldsymbol{\zeta}_{1,c}\right) \qquad (3.2.23)$$

An approach for implementation of the control loop in section 3.2.1 and is represented by a block diagram in figure 3.1.

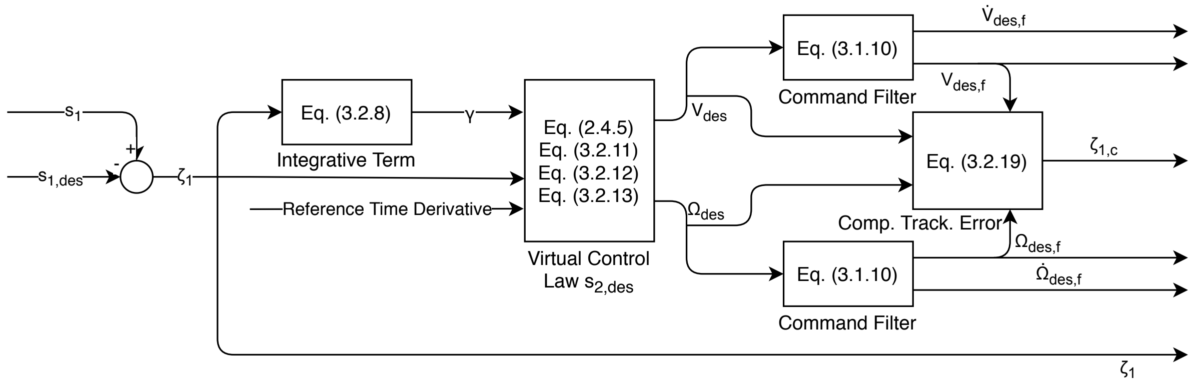

Figure 3.1: Kinematics Loop Controller Block Diagram

3.2.3 Control Allocation

The expressions for the desired applied moments and forces to the platform were defined in (3.2.20). As the moments and forces are functions of the motor speeds and servo angles, (2.3.11) and (2.3.10) respectively, it is now necessary to know what control allocation produces them. A first approach may be applying the *Newton's Method* in order to find them.

Newton's Method is an iterative root-finding algorithm, i.e. it is an algorithm that computes in each step, a $\boldsymbol{x}_i$ in order to make $\boldsymbol{f}(\boldsymbol{x}_i)$ tend to zero as i tends to infinity. It uses the linear approximation of the function that is intended to find the root for by assuming

$$\Delta \boldsymbol{f}(\boldsymbol{x}) \approx \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}} \cdot \Delta \boldsymbol{x}, \tag{3.2.24}$$

where $\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}$ is analytically defined.

Let us introduce the saturation function

$$S(x, m, M) = \begin{cases} m & \text{, if } x < m \\ x & \text{, if } m \leq x \leq M \\ M & \text{, if } x > m \end{cases} .$$

Introduce a variable change from ω_i and ν_i

$$\upsilon_i = \begin{cases} S\left(\frac{1}{10}\,\omega_i, 0, 100\right) & \text{, for } i = 1,2,3,4 \\ S\left(\frac{300}{\pi}\,\nu_{i-4}, -100, 100\right) & \text{, for } i = 5,6,7,8 \end{cases} \tag{3.2.25}$$

where it is assumed that the motors rotate at a minimum and maximum speeds of 0 and 1000 radians per second, respectively, and the servo motors achieve angles from $-\frac{\pi}{3}$ to $\frac{\pi}{3}$ radians. This variable change expresses the actuation effort in the same scale, as a percentage of the maximum values. Define the associated vector to the new variables υ_i

$$\boldsymbol{\Upsilon} = \begin{bmatrix} \upsilon_1 & \upsilon_2 & \upsilon_3 & \upsilon_4 & \upsilon_5 & \upsilon_6 & \upsilon_7 & \upsilon_8 \end{bmatrix}^T . \tag{3.2.26}$$

Rewriting the definition for $\boldsymbol{N}$:

$$\boldsymbol{N} = 100 \cdot \begin{bmatrix} \upsilon_2^2 \cos\frac{\pi\upsilon_5}{300}\left(dK_T\cos\frac{\pi\upsilon_7}{300} + K_Q\sin\frac{\pi\upsilon_7}{300}\right) - \upsilon_4^2\cos\frac{\pi\upsilon_6}{300}\left(dK_T\cos\frac{\pi\upsilon_8}{300} + K_Q\sin\frac{\pi\upsilon_8}{300}\right) \\ dK_T(\upsilon_1^2 - \upsilon_3^2) - K_Q\upsilon_2^2\cos\frac{\pi\upsilon_7}{300}\sin\frac{\pi\upsilon_5}{300} + K_Q\upsilon_4^2\cos\frac{\pi\upsilon_8}{300}\sin\frac{\pi\upsilon_6}{300} \\ K_Q(\upsilon_3^2 - \upsilon_1^2) + \upsilon_2^2\cos\frac{\pi\upsilon_5}{300}\left(K_Q\cos\frac{\pi\upsilon_7}{300} - dK_T\sin\frac{\pi\upsilon_7}{300}\right) + \upsilon_4^2\cos\frac{\pi\upsilon_6}{300}\left(-K_Q\cos\frac{\pi\upsilon_8}{300} + dK_T\sin\frac{\pi\upsilon_8}{300}\right) \\ -K_T\upsilon_2^2\cos\frac{\pi\upsilon_5}{300}\sin\frac{\pi\upsilon_7}{300} - K_T\upsilon_4^2\cos\frac{\pi\upsilon_6}{300}\sin\frac{\pi\upsilon_8}{300} \\ K_T\upsilon_2^2\sin\frac{\pi\upsilon_5}{300}\cos\frac{\pi\upsilon_7}{300} + K_T\upsilon_4^2\sin\frac{\pi\upsilon_6}{300}\cos\frac{\pi\upsilon_8}{300} \\ -K_T\left(\upsilon_1^2 + \upsilon_2^2\cos\frac{\pi\upsilon_5}{300}\cos\frac{\pi\upsilon_7}{300} + \upsilon_3^2 + \upsilon_4^2\cos\frac{\pi\upsilon_6}{300}\cos\frac{\pi\upsilon_8}{300}\right) \end{bmatrix}, \tag{3.2.27}$$

which allows for its partial derivative with respect to $\boldsymbol{\Upsilon}$ to be defined as

$$\frac{\partial\boldsymbol{N}(\boldsymbol{\Upsilon})}{\partial\boldsymbol{\Upsilon}} = \begin{bmatrix} \frac{\partial\boldsymbol{M}}{\partial\boldsymbol{\Upsilon}_{1-4}} & \frac{\partial\boldsymbol{M}}{\partial\boldsymbol{\Upsilon}_{5-8}} \\ \frac{\partial\boldsymbol{F}}{\partial\boldsymbol{\Upsilon}_{1-4}} & \frac{\partial\boldsymbol{F}}{\partial\boldsymbol{\Upsilon}_{5-8}} \end{bmatrix}, \tag{3.2.28}$$

where $\boldsymbol{\Upsilon}_{1-4}$ is composed of the first four elements of $\boldsymbol{\Upsilon}$ and $\boldsymbol{\Upsilon}_{5-8}$ the last four, with

$$\frac{\partial\boldsymbol{M}}{\partial\Upsilon_{1-4}} = 200\begin{bmatrix} 0 & \upsilon_2c_5\left(dK_Tc_7 + K_Qs_7\right) & 0 & -\upsilon_4c_6\left(dK_Tc_8 + K_Qs_8\right) \\ dK_T\upsilon_1 & -K_Q\upsilon_2s_5c_7 & -dK_T\upsilon_3 & K_Q\upsilon_4s_6c_8 \\ -K_Q\upsilon_1 & \upsilon_2c_5\left(K_Qc_7 - dK_Ts_7\right) & K_Q\upsilon_3 & \upsilon_4c_6\left(dK_Ts_8 - K_Qc_8\right) \end{bmatrix}; \tag{3.2.29}$$

$$\frac{\partial\boldsymbol{M}}{\partial\Upsilon_{5-8}} = \frac{\pi}{3}\begin{bmatrix} -\upsilon_2^2s_5\left(dK_Tc_7 + K_Qs_7\right) & \upsilon_4^2s_6\left(dK_Tc_8 + K_Qs_8\right) & \upsilon_2^2c_5\left(K_Qc_7 - dK_Ts_7\right) & \upsilon_4^2c_6\left(dK_Ts_8 - K_Qc_8\right) \\ -K_Q\upsilon_2^2c_5c_7 & K_Q\upsilon_4^2c_6c_8 & K_Q\upsilon_2^2s_5s_7 & K_Q - \upsilon_4^2s_6s_8 \\ \upsilon_2^2s_5\left(dK_Ts_7 - K_Qc_7\right) & \upsilon_4^2s_6\left(K_Qc_8 - dK_Ts_8\right) & -\upsilon_2^2c_5\left(dK_Tc_7 + K_Qs_7\right) & \upsilon_4^2c_6\left(dK_Tc_8 + K_Qs_8\right) \end{bmatrix}; \tag{3.2.30}$$

$$\frac{\partial\boldsymbol{F}}{\partial\Upsilon_{1-4}} = 200K_T\begin{bmatrix} 0 & -\upsilon_2c_5s_7 & 0 & -\upsilon_4c_6s_8 \\ 0 & \upsilon_2s_5c_7 & 0 & \upsilon_4s_6c_8 \\ -\upsilon_1 & -\upsilon_2c_5c_7 & -\upsilon_3 & -\upsilon_4c_6c_8 \end{bmatrix}; \tag{3.2.31}$$

$$\frac{\partial\boldsymbol{F}}{\partial\Upsilon_{5-8}} = \frac{\pi}{3}K_T\begin{bmatrix} \upsilon_2^2s_5s_7 & \upsilon_4^2s_6s_8 & -\upsilon_2^2c_5c_7 & -\upsilon_4^2c_6c_8 \\ \upsilon_2^2c_5c_7 & \upsilon_4^2c_6c_8 & \upsilon_2^2s_5s_7 & \upsilon_4^2s_6s_8 \\ \upsilon_2^2s_5c_7 & \upsilon_4^2s_6c_8 & \upsilon_2^2c_5s_7 & \upsilon_4^2c_6s_8 \end{bmatrix}, \tag{3.2.32}$$

where

$$c_i = \cos\frac{\pi\upsilon_i}{300},$$
$$s_i = \sin\frac{\pi\upsilon_i}{300}.$$

The algorithm starts with an initial guess $\boldsymbol{\Upsilon}_0$ and computes the error $\boldsymbol{E}_0 = \boldsymbol{N}(\boldsymbol{\Upsilon}_0) - \boldsymbol{N}_{des}$. The desired value for this error is zero, i.e. $\boldsymbol{N}(\boldsymbol{\Upsilon}_0) = \boldsymbol{N}_{des}$. The algorithm continues by assessing if $\boldsymbol{E}_0$ reached the stopping condition, e.g. $||\boldsymbol{E}_0||^2 \leq tol$ for some given tolerance tol. If the condition was reached, then $\boldsymbol{\Upsilon}_0$ is the output of the algorithm. If not, then the next guess, $\boldsymbol{\Upsilon}_1$, is computed by

$$\Delta\boldsymbol{\Upsilon} = -\left[\left.\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}}\right|_{\boldsymbol{\Upsilon}_0}\right]^{\dagger} \cdot \boldsymbol{E}_0$$

$$\boldsymbol{\Upsilon}_1 = \boldsymbol{\Upsilon}_0 + \Delta\boldsymbol{\Upsilon}$$

where † denotes a pseudo-inverse. The algorithm continues by assessing if $\boldsymbol{E}_1 = \boldsymbol{N}(\boldsymbol{\Upsilon}_1) - \boldsymbol{N}_{des}$ reached the stopping condition, repeating the process. Note that the stopping condition for the iterative process can also include a maximum number of iterations.

It should be pointed that the *Newton's Method* may not converge. Even if it converges to a solution, it may not be feasible (e.g. due to actuator saturation) or it may not be the most elegant (or efficient) solution, for example having the servos at a non-zero position or having unsymmetrical motor speeds while hovering. The possibility of applying this method will be addressed in the simulation results chapter.

If the first guess, $\boldsymbol{\Upsilon}_0$, is constant and only one iteration is allowed for the algorithm then the Jacobian matrix, $\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}}|_{\boldsymbol{\Upsilon}_0}$, is constant as so is its pseudo-inverse. If one iteration only is enough to stabilize the system, then it is possible to avoid the controller to invert matrices, which is computationally expensive. In this case, $\boldsymbol{\Upsilon}$ is calculated as

$$\boldsymbol{\Upsilon} = \boldsymbol{\Upsilon}_0 + \boldsymbol{K}_{Alloc}\left(\boldsymbol{N}(\boldsymbol{\Upsilon}_0) - \boldsymbol{N}_{des}\right), \tag{3.2.33}$$

where

$$\boldsymbol{K}_{Alloc} = -\left[\left.\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}}\right|_{\boldsymbol{\Upsilon}_0}\right]^{\dagger}. \tag{3.2.34}$$

Note that the choice of $\boldsymbol{\Upsilon}_0$ may have a significant influence on stability and performance. If the chosen $\boldsymbol{\Upsilon}_0$ makes $\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}}|_{\boldsymbol{\Upsilon}_0}$ singular or poorly conditioned, the resulting $\boldsymbol{K}_{Alloc}$ may cause the control signals change too aggressively, ultimately making the system unstable.

Each strategy has advantages and disadvantages which will be addressed in the simulation results chapter.

Having obtained $\boldsymbol{\Upsilon}$, the PWM signal on-time can then be calculated through a linear relation as

$$PWM_{\omega i} = \frac{PWM_{max} - u_{\omega,d}}{100}\upsilon_i + u_{\omega,d}, i = 1, 2, 3, 4 \tag{3.2.35}$$

$$PWM_{\nu i} = \frac{PWM_{max} - PWM_{min}}{200}\upsilon_{i+4} + \frac{PWM_{max} + PWM_{min}}{2}, i = 1, 2, 3, 4 \tag{3.2.36}$$

because it is wanted for $PWM_{\omega i}$ to be equal to PWM_{max} when $\upsilon_i = 100$ and $u_{\omega,d}$ when $\upsilon_i = 0$, $i = 1, 2, 3, 4$. Similarly for $PWM_{\upsilon i}$, it is meant to be PWM_{max} when $\upsilon_{i+4} = 100$ and PWM_{min} when $\upsilon_{i+4} = -100$, $i = 1, 2, 3, 4$.

The $\boldsymbol{K}_{Alloc}$ approach for implementation of the control loop in this section and section 3.2.2 and is represented by a block diagram in figure 3.2. Note that if the *Newton Method* approach is used, the block ”F and M to Actuation” does not use the equation stated and should employ the mentioned root-finding algorithm instead.

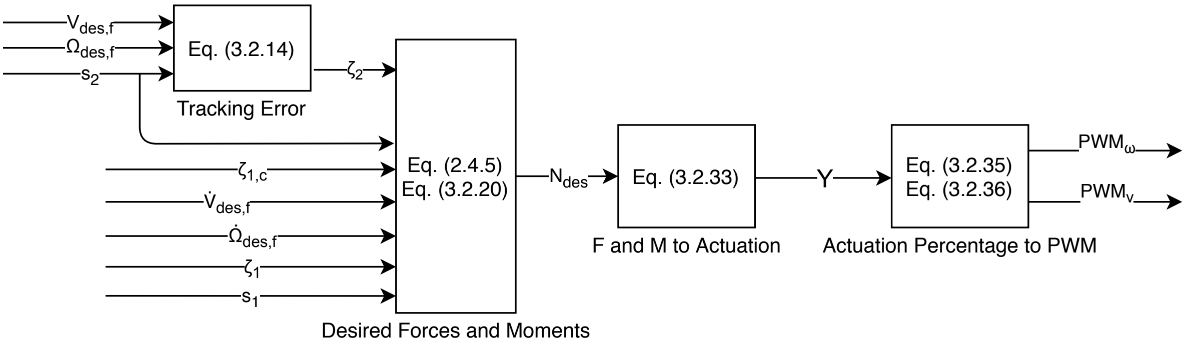

Figure 3.2: Backstepping Dynamics Loop Controller Block Diagram

3.3 Incremental Backstepping Implementation

In this section, a controller for the model described in section 2.4 will be designed by applying the Incremental Backstepping design method.

The kinematics of the ALIV3 model does not present uncertainty and as such, the incremental design will not be applied for the first step of the method (kinematics loop). This means that the first step is identical to the Backstepping method, see section 3.2.1 and figure 3.1. However, for the second step (dynamics loop) it may be advantageous to. The dynamics is the loop that is affected directly by model or actuator parameter uncertainty and disturbance forces such as wind resistance, which is not modeled.

3.3.1 Dynamics Loop

The dynamics loop control design starts by defining the tracking error as in (3.2.14).

The $\boldsymbol{s}_2$ dynamics is now approximated to

$$\begin{aligned} \dot{\boldsymbol{s}}_2 &\approx \dot{\boldsymbol{s}}_{2,0} + \boldsymbol{g}_2 \cdot \Delta \boldsymbol{N} \\ &\approx \dot{\boldsymbol{s}}_{2,0} + \boldsymbol{g}_2 \cdot \left[\left. \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}} \right|_{\boldsymbol{\Upsilon}_0} \right] \cdot \Delta \boldsymbol{\Upsilon} \end{aligned} \tag{3.3.1}$$

which makes the $\boldsymbol{\zeta}_2$ dynamics to be defined as

$$\dot{\boldsymbol{\zeta}}_2 \approx \dot{\boldsymbol{s}}_{2,0} + \boldsymbol{g}_2 \cdot \left[\left. \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}} \right|_{\boldsymbol{\Upsilon}_0} \right] \cdot \Delta \boldsymbol{\Upsilon} - \dot{\boldsymbol{s}}_{2,des,f}. \tag{3.3.2}$$

For notation simplicity define

$$\overline{\boldsymbol{B}}_0 = \boldsymbol{g}_2,$$

the constant part of

$$\boldsymbol{B}_0 = \overline{\boldsymbol{B}}_0 \cdot \left[\left. \frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}} \right|_{\boldsymbol{\Upsilon}_0} \right].$$

Similarly to the backstepping design method, with $\boldsymbol{\zeta}_2$ defined, $\dot{V}_1$ can be rewritten the same as (3.2.16)

$$\dot{V}_1 = \underbrace{-\boldsymbol{\zeta}_1^T \boldsymbol{K}_1 \boldsymbol{\zeta}_1}_{\leq 0} + \boldsymbol{\zeta}_1^T \boldsymbol{g}_1 \left(\boldsymbol{s}_1\right) \boldsymbol{\zeta}_2 + \boldsymbol{\zeta}_1^T \boldsymbol{g}_1 \left(\boldsymbol{s}_1\right) \left(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des}\right) \tag{3.3.3}$$

The second part of the second step is to define a new Lyapunov function candidate

$$V_2 = V_1 + \frac{1}{2} \boldsymbol{\zeta}_2^T \boldsymbol{\zeta}_2, \tag{3.3.4}$$

which fulfills the conditions in section 3.1.1 for a Lyapunov function candidate, as

$$V_2(\boldsymbol{\zeta}_2, \boldsymbol{\zeta}_1, \boldsymbol{\gamma}) \to \infty, \ (\boldsymbol{\zeta}_2, \boldsymbol{\zeta}_1, \boldsymbol{\gamma}) \to \infty$$

$$V_2 > 0, \text{ for } \boldsymbol{\zeta}_2 \neq \boldsymbol{0}, \ \boldsymbol{\zeta}_1 \neq \boldsymbol{0} \text{ and } \boldsymbol{\gamma} \neq \boldsymbol{0}$$

The third part is to design a virtual stabilizing control law, $\Delta \boldsymbol{\Upsilon}$, that makes $\dot{V}_2$ negative definite for $\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2 \neq 0$.

$$\begin{aligned} \dot{V}_2 &= \dot{V}_1 + \boldsymbol{\zeta}_2^T \left(\dot{\boldsymbol{s}}_{2,0} - \dot{\boldsymbol{s}}_{2,des,f} + \boldsymbol{B}_0 \cdot \Delta \boldsymbol{\Upsilon}\right) \\ &= -\boldsymbol{\zeta}_1^T \boldsymbol{K}_1 \boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2^T \left(\dot{\boldsymbol{s}}_{2,0} - \dot{\boldsymbol{s}}_{2,des,f} + \boldsymbol{g}_1^T \left(\boldsymbol{s}_1\right) \boldsymbol{\zeta}_1 + \boldsymbol{B}_0 \cdot \Delta \boldsymbol{\Upsilon}\right) + \boldsymbol{\zeta}_1^T \boldsymbol{g}_1 \left(\boldsymbol{s}_1\right) \left(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des}\right) \end{aligned} \tag{3.3.5}$$

In order to guarantee stability, one needs to compensate the difference $(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des})$ with $\boldsymbol{\zeta}_{1,c}$ [18], defined in (3.2.19):

$$\begin{aligned} &\dot{\boldsymbol{\zeta}}_{1,c} = -\boldsymbol{K}_1 \boldsymbol{\zeta}_{1,c} - \boldsymbol{K}_i \boldsymbol{\gamma} + \boldsymbol{g}_1(\boldsymbol{s}_1) \left(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des}\right) \\ &\boldsymbol{\zeta}_{1,c}(t = 0) = \boldsymbol{0} \end{aligned} \tag{3.2.19}$$

If $\Delta \boldsymbol{\Upsilon}$ is defined as

$$\Delta \boldsymbol{\Upsilon} = \boldsymbol{B}_0^{\dagger} \left(-\dot{\boldsymbol{s}}_{2,0} + \dot{\boldsymbol{s}}_{2,des,f} - \boldsymbol{g}_1^T \left(\boldsymbol{\zeta}_1 - \boldsymbol{\zeta}_{1,c}\right) - \boldsymbol{K}_2 \boldsymbol{\zeta}_2\right), \tag{3.3.6}$$

where † represents a pseudo-inverse, assuming the model is correct and the sampling frequency is high enough, then the control law fulfills the requirements for stability, more information can be found in [18]. The control law is defined as

$$\boldsymbol{\Upsilon} = \boldsymbol{\Upsilon}_0 + \Delta \boldsymbol{\Upsilon}. \tag{3.3.7}$$

The $\boldsymbol{\zeta}_1$ error dynamics are obtained the same way as for the Backstepping controller. As for the $\boldsymbol{\zeta}_2$ error dynamics, replace $\Delta \boldsymbol{\Upsilon}$ by its definition, equation (3.3.6), in equation (3.3.2), note that it is assumed that $\boldsymbol{B}_0 \cdot \boldsymbol{B}_0^\dagger = \boldsymbol{I}_6$,

$$\begin{aligned}\dot{\boldsymbol{\zeta}}_2 &\approx \dot{\boldsymbol{s}}_{2,0} - \dot{\boldsymbol{s}}_{2,des,f} + \boldsymbol{B}_0 \cdot \boldsymbol{B}_0^\dagger \left(-\dot{\boldsymbol{s}}_{2,0} + \dot{\boldsymbol{s}}_{2,des,f} - \boldsymbol{g}_1^T \left(\boldsymbol{\zeta}_1 - \boldsymbol{\zeta}_{1,c}\right) - \boldsymbol{K}_2 \boldsymbol{\zeta}_2\right) \\ &= -\boldsymbol{g}_1^T \left(\boldsymbol{\zeta}_1 - \boldsymbol{\zeta}_{1,c}\right) - \boldsymbol{K}_2 \boldsymbol{\zeta}_2,\end{aligned}$$

which leads to the same total error dynamics as for the Backstepping controller:

$$\dot{\boldsymbol{\gamma}} = \boldsymbol{\zeta}_1 \tag{3.3.8}$$

$$\dot{\boldsymbol{\zeta}}_1 = -\boldsymbol{K}_1 \boldsymbol{\zeta}_1 - \boldsymbol{K}_i \boldsymbol{\gamma} + \boldsymbol{g}_1(\boldsymbol{s}_1)\boldsymbol{\zeta}_2 + \boldsymbol{g}_1(\boldsymbol{s}_1)\left(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des}\right) \tag{3.3.9}$$

$$\dot{\boldsymbol{\zeta}}_2 = -\boldsymbol{K}_2 \boldsymbol{\zeta}_2 - \boldsymbol{g}_1^T(\boldsymbol{s}_1)\left(\boldsymbol{\zeta}_1 - \boldsymbol{\zeta}_{1,c}\right) \tag{3.3.10}$$

Having defined $\boldsymbol{\Upsilon}$, the PWM signals are calculated, the same way as for the Backstepping controller, as in (3.2.35) and (3.2.36).

3.3.2 Matrix Pseudoinverse

In order to be able to compute (3.3.6), it is firstly needed to compute $\boldsymbol{B}_0^\dagger$. Since

$$\boldsymbol{B}_0 = \overline{\boldsymbol{B}}_0 \cdot \left.\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}}\right|_{\boldsymbol{\Upsilon}_0},$$

then

$$\boldsymbol{B}_0^\dagger = \left[\left.\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}}\right|_{\boldsymbol{\Upsilon}_0}\right]^\dagger \cdot \overline{\boldsymbol{B}}_0^{-1}.$$

It is possible to define analytically

$$\overline{\boldsymbol{B}}_0^{-1} = \begin{bmatrix} \boldsymbol{J} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & m \cdot \boldsymbol{I}_3 \end{bmatrix}$$

because it is a constant diagonal matrix with non-zero elements in the principal diagonal. For notation simplicity, let us define

$$\boldsymbol{X}_0 = \left.\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}}\right|_{\boldsymbol{\Upsilon}_0}.$$

According to [1, pages 19-21], the pseudo-inverse of $\boldsymbol{X}_0$ can be defined as

$$\boldsymbol{X}_0^\dagger = \lim_{\delta_\Upsilon \to 0} \boldsymbol{X}_0^T (\boldsymbol{X}_0 \boldsymbol{X}_0^T + \delta_\Upsilon^2 \boldsymbol{I}_6)^{-1}.$$

If a non-zero scalar δ_Υ whose magnitude is less than the smallest non-zero eigenvalue of $\boldsymbol{X}_0 \boldsymbol{X}_0^T$ is chosen, it is guaranteed that $(\boldsymbol{X}_0 \boldsymbol{X}_0^T + \delta_\Upsilon^2 \boldsymbol{I}_6)$ is non-singular and that its inverse exists. Choosing the right sized δ_Υ it may be possible to avoid singularities and

$$\boldsymbol{X}_0^T (\boldsymbol{X}_0 \boldsymbol{X}_0^T + \delta_\Upsilon^2 \boldsymbol{I}_6)^{-1} \approx \boldsymbol{X}_0^T (\boldsymbol{X}_0 \boldsymbol{X}_0^T)^{-1},$$

if $\boldsymbol{X}_0 \boldsymbol{X}_0^T$ is non-singular.

The tilt-quadrotor is overactuated, which means that there may be more than one $\Delta\boldsymbol{\Upsilon}$ that solves the equation

$$\boldsymbol{X}_0 \Delta\boldsymbol{\Upsilon} = \boldsymbol{b}$$

for an arbitrary $\boldsymbol{b}$. Note that $\boldsymbol{X}_0$ is an 6×8 matrix, meaning that its rank is at most 6, i.e. has at most 6 linearly independent columns. Let us define

- The range (or column space) of a matrix $\boldsymbol{A}$

$$\mathrm{Sp}(\boldsymbol{A}) = \{y | \boldsymbol{A}x = y\};$$

- The kernel (or null space) of a matrix $\boldsymbol{A}$

$$\mathrm{Ker}(\boldsymbol{A}) = \{x | \boldsymbol{A}x = 0\}.$$

Let the solution for $\Delta\boldsymbol{\Upsilon}$ be split into

$$\Delta\boldsymbol{\Upsilon} = \boldsymbol{y}_1 + \boldsymbol{y}_2,$$

where

$$\boldsymbol{y}_1 = \boldsymbol{X}_0^{\dagger} \cdot \boldsymbol{b}$$

and

$$\boldsymbol{X}_0 \cdot \boldsymbol{y}_2 = \boldsymbol{0}.$$

This is possible if $\boldsymbol{y}_2$ is obtained by [36, page 80]

$$\boldsymbol{y}_2 = \left(\boldsymbol{I}_8 - \boldsymbol{X}_0^{\dagger}\boldsymbol{X}_0\right)\boldsymbol{z}$$

with arbitrary $\boldsymbol{z}$. This is due to $\mathrm{Sp}(\boldsymbol{I} - \boldsymbol{X}_0^{\dagger}\boldsymbol{X}_0) = \mathrm{Ker}(\boldsymbol{X}_0)$. One may apply this property to use the null space of $\boldsymbol{X}_0$ to force the $\Delta\boldsymbol{\Upsilon}$ solution to follow a path, which in this case is to avoid actuator saturation. This means that $\boldsymbol{z}$ can be chosen to penalize when $\boldsymbol{\Upsilon}_0$ is close to saturation, defining

$$\boldsymbol{z} = \boldsymbol{K}_{\Upsilon}\left(\boldsymbol{\Upsilon}_0 - \boldsymbol{\Upsilon}_r\right),$$

where $\boldsymbol{\Upsilon}_r$ is a reference vector for $\boldsymbol{\Upsilon}$, a design parameter, and $\boldsymbol{K}_{\Upsilon}$ is a constant 8×8 diagonal matrix whose elements are non-zero. Note that $\boldsymbol{y}_2$ does not alter the solution for $\Delta\boldsymbol{\Upsilon}$, maintaining the desired properties.

Finally, $\boldsymbol{X}_0^{\dagger}$, $\Delta\boldsymbol{\Upsilon}$ and $\boldsymbol{\Upsilon}$ are computed in the following way:

$$\boldsymbol{X}_0^{\dagger} = \left[\left.\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}}\right|_{\boldsymbol{\Upsilon}_0}\right]^T \left(\left.\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}}\right|_{\boldsymbol{\Upsilon}_0} \cdot \left[\left.\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}}\right|_{\boldsymbol{\Upsilon}_0}\right]^T + \delta_{\Upsilon}^2 \cdot \boldsymbol{I}_6\right)^{-1}, \tag{3.3.11}$$

$$\Delta\boldsymbol{\Upsilon} = \boldsymbol{X}_0^{\dagger} \begin{bmatrix} \boldsymbol{J} & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 3} & m\cdot\boldsymbol{I}_3 \end{bmatrix} \Bigg(\Bigg(\begin{bmatrix} \dot{\boldsymbol{\Omega}}_{des,f} \\ \dot{\boldsymbol{V}}_{des,f} \end{bmatrix} - \begin{bmatrix} \dot{\boldsymbol{\Omega}} \\ \dot{\boldsymbol{V}} \end{bmatrix} - \begin{bmatrix} \boldsymbol{T}^T & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{S}^T \end{bmatrix} \Bigg(\Bigg(\begin{bmatrix} \boldsymbol{\zeta}_{\Phi} \\ \boldsymbol{\zeta}_p \end{bmatrix} - \boldsymbol{\zeta}_c \Bigg) - \boldsymbol{K}_2\boldsymbol{\zeta}_2 \Bigg) \\ + \Big(\boldsymbol{I}_8 - \boldsymbol{X}_0^{\dagger}\boldsymbol{X}_0\Big)\boldsymbol{K}_{\Upsilon}\left(\boldsymbol{\Upsilon}_0 - \boldsymbol{\Upsilon}_r\right), \tag{3.3.12}$$

$$\boldsymbol{\Upsilon} = \boldsymbol{\Upsilon}_0 + \Delta\boldsymbol{\Upsilon}. \tag{3.3.13}$$

Maintaining the properties of (3.3.6) but also avoiding actuator saturation.

An approach for implementation of the control loop in this section and section 3.3.1 and is represented by a block diagram in figure 3.3.

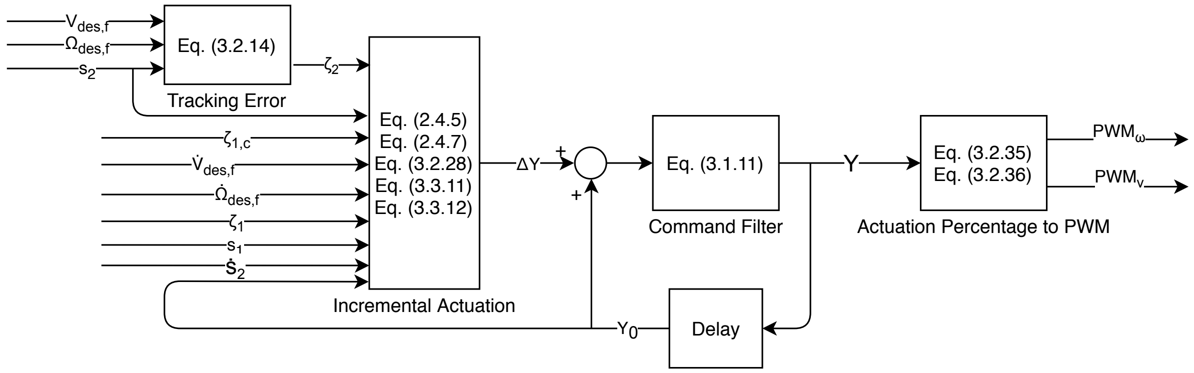

Figure 3.3: Incremental Backstepping Dynamics Loop Controller Block Diagram

Chapter 4

Simulation Results

The control strategies in the previous section are implemented and tested on *Simulink*, a `Matlab` simulation tool. A simplified block diagram of the system can be found in figure 2.2. The ”Controllers Block”, better described in figures 3.1, 3.2 and 3.3, includes each of the designed controllers in the previous sections 3.2 and 3.3, and the ”Rigid-body Dynamics” Block, explored in figure 2.3, includes the dynamics described in section 2.4.

For the Backstepping controller a continuous sampling and actuation is considered in simulation, which not the case with the Incremental Backstepping controller that takes samples at a the rate of 200 Hz. The actuation is updated at the same frequency with a delay of 5 ms, due to the need to access the previous time-sample actuation $\boldsymbol{\Upsilon}_0$. Although this choice does not stem from a theoretical study, a practical approach was used: several rates were tested and this one seems to perform as expected for this work. Longer delays led to more unstable behaviour and shorter delays strained the calculation capability without an equivalent improvement. No further investigation on this subject was taken.

The parameters and constants used for the model and controllers are included in section Constants.

In order to assess and compare controller performance, a first approach is feeding a series of step references into the controllers, see section 4.1. Then, the controller is evaluated for a more challenging maneuver, with coupled motion, see section 4.2.

4.1 Step Response

The simpler reference signals are composed of different step references every 20 seconds in order to let the system achieve and remain in steady-state for approximately 10 seconds. It starts with an ascending step reference in the x direction followed by a descending one. Then the same two step references in the y direction and lastly in the z direction. Both for position, $\mathbf{P}$, and attitude, $\boldsymbol{\Phi}$.

4.1.1 Backstepping Controller

For the Backstepping controller, two control allocation strategies are considered: a constant allocation matrix $\boldsymbol{K}_{Alloc}$, introduced in (3.2.34), and the *Newton's Method* in order to find the correct motor speeds and servo angles for the desired Forces and Moments.

The stabilizing and reference tracking results depend on the chosen gains $\boldsymbol{K}_i$, $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ for the BKS controller. Although the choices do not stem from a theoretical study, a practical approach was used: a combination of the three matrix gains that delivered a satisfactory performance for both control allocation strategies.

Simulation results with $\boldsymbol{K}_{Alloc}$ for linear position demands are presented in figure 4.1. It can be seen that the $\boldsymbol{P}$ and $\boldsymbol{\Phi}$ systems are practically decoupled, i.e. that the linear movement practically does not affect the attitude of the platform.

The reference is followed by the system throughout the simulation with an absolute error smaller than 1%, zero steady-state error and response overshoot up to $7 \cdot 10^{-3}\%$.

The response for attitude steps, in figure 4.2, shows a different behaviour with respect to $\boldsymbol{P}$ and $\boldsymbol{\Phi}$ systems decoupling, but the influence of the attitude is rather small on the position.

A positive step in roll angle, φ, makes the platform oscillate 0.02 meters in the y direction and fall 0.02 meters (positive z direction). The negative step has the mirrored response on the linear position.

The response is very similar for the step in pitch angle, θ, with the difference that the positive pitch angle makes the platform move in the negative direction of x.

Yaw angle step response has a much smaller influence on the linear position, up to 0.002 meters. The references in the three directions are followed closely, with an error smaller than 1%, zero steady-state error and response overshoot up to 1.6%.

The two figures 4.1 and 4.2 show that the controller tends to force the four motor speeds to be equal and the servo angles to be equal in pairs, $\phi_2 \approx \phi_4$ and $\vartheta_2 \approx \vartheta_4$. Note that unwanted angular position oscillations occur when it is requested for the platform to hover, at around 140 seconds in figure 4.1. In order to assess if the platform becomes unstable, longer simulations (up to 1000 seconds) holding the hover request were run. These revealed that the platform does oscillate after 140 seconds of simulation. However, the oscillation peaks do not surpass 0.15 degrees, with a period of at least 1.5 seconds.

Applying the *Newton's Method* for the control allocation has a similar result to the constant matrix method but not with the same actuation.

Figure 4.3 shows similar response, peaks in servo angle and motor speeds to the constant matrix method in figure 4.1.

This approach does not tend to force the motor speeds to be equal, unlike the previous. In figure 4.4, it can be seen that the controller tends to increase the effort on the fixed motors (without servos), i.e. motors 1 and 3, increasing the necessary servo angle to maintain the tilted position compared to figure 4.2.

However, the *Newton's Method* approach introduces increased unwanted oscillations on the attitudes, when comparing to the $\boldsymbol{K}_{Alloc}$ approach, as can be seen in figure 4.3 at around 140 seconds and figure 4.4 at around 80 seconds. Although these oscillations are quite small, around 0.25 degrees and 2 degrees respectively, the additional computational burden is not worthwhile because it is not offset by an equivalent benefit on the results.

In order to compare computational effort, 20 simulations were run for each control allocation ap-

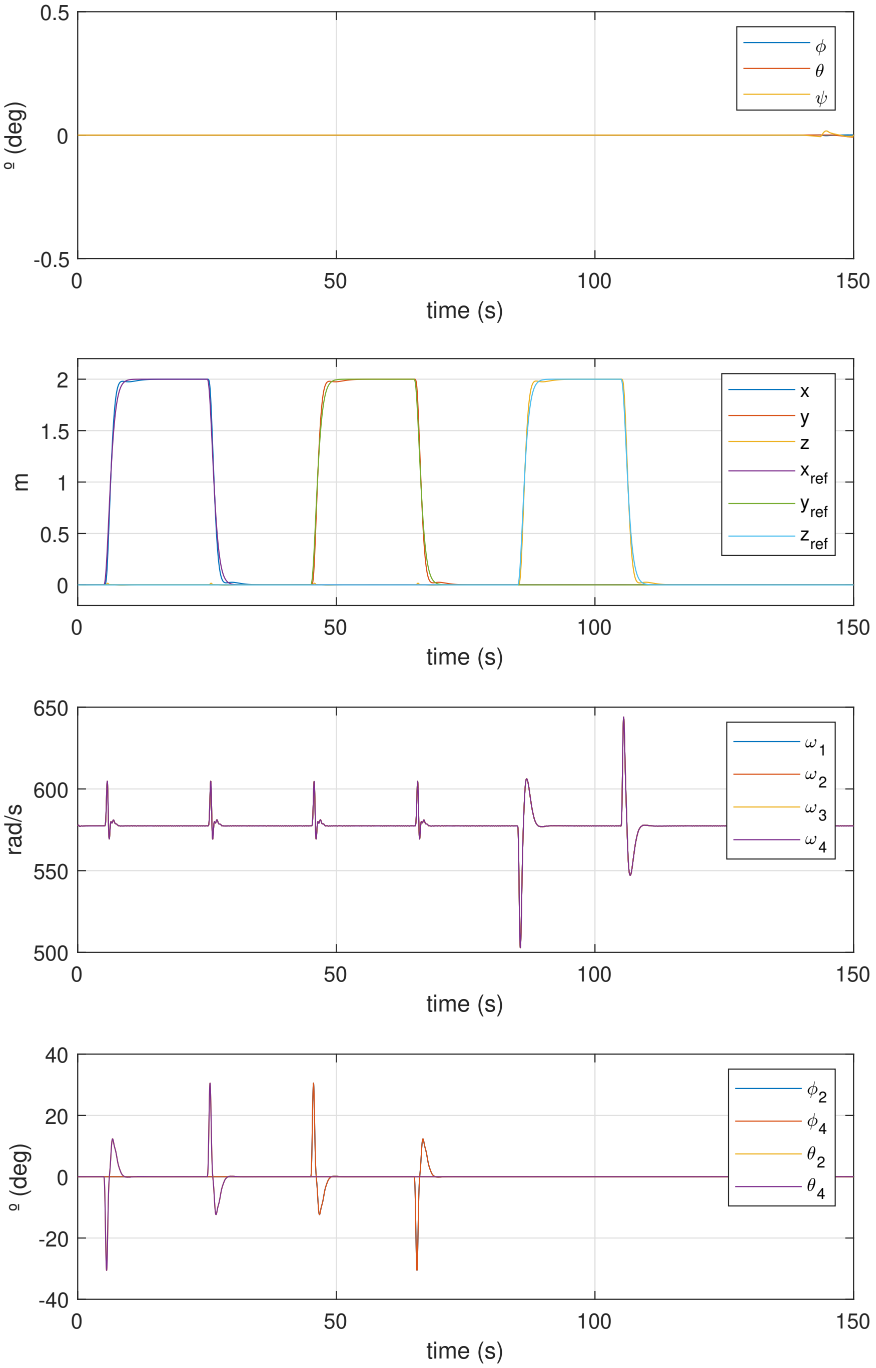

Figure 4.1: UAV response for different linear position steps using BKS with $\boldsymbol{K}_{Alloc}$.
From top to bottom: attitude, linear position and their references, motor speeds, servo angles.

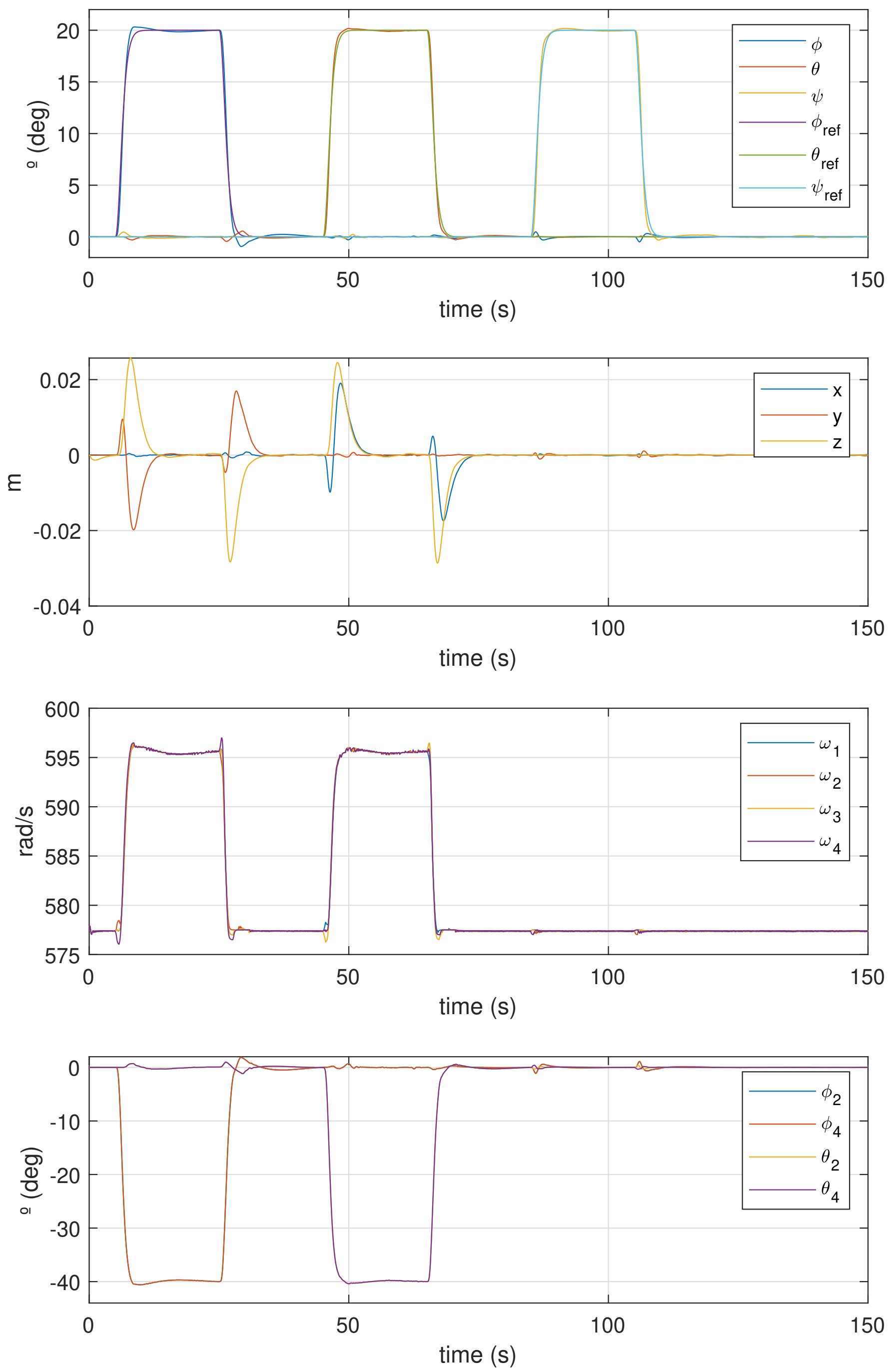

Figure 4.2: UAV response for different angular position steps using BKS with $\boldsymbol{K}_{Alloc}$.
From top to bottom: attitude and their references, linear position, motor speeds, servo angles.

proach on the same computer and approximately same conditions: 10 with position step reference and other 10 with attitude step reference. The average run time for the constant allocation matrix approach was 14.3 seconds which is less than half the average run time for the *Newton's Method* approach of 30.5 seconds.

4.1.2 Incremental Backstepping Controller

The Incremental Backstepping control strategy performs an almost decoupled tracking of the linear and angular positions, $\boldsymbol{P}$ and $\boldsymbol{\Phi}$ respectively, as can be seen in figures 4.5 and 4.6. The references are followed closely and zero steady-state error is achieved.

The stabilizing and reference tracking results depend on the chosen gains $\boldsymbol{K}_i$, $\boldsymbol{K}_1$ and $\boldsymbol{K}_2$ for the IBKS controller. Although the choices do not stem from a theoretical study, a practical approach was used: a combination of the three matrix gains that delivered a satisfactory performance.

Figure 4.5 shows a system response overshoot of 1.8% on the tracking of the UAV linear position in x, y, and z directions. Comparing the motor speeds and servo angles in figure 4.2 with figure 4.6, it can be seen that the IBKS controller requires more effort of motors 2 and 4 up more than the BKS controller, decreasing the needed angle for the servo motors.

Figure 4.6 shows that a tilting of the platform introduces a small spike on the position of the platform up to 0.03 meters. The response shows an overshoot of 2%.
After the platform returns to the horizontal position, the controller makes the motor speeds and servo angles tend to the initial conditions slowly, i.e. equal motor speeds and zero servo angles.

4.2 Coupled Reference

The more elaborate reference include position and attitude requests simultaneously. The platform must move in a certain predefined path while changing its attitude in order to test the decoupling of the two subsystems, attitude and linear position.

4.2.1 Backstepping Controller

The BKS controller with the constant allocation matrix shows a larger error tracking the attitude reference compared to the position reference, as seen in figure 4.8. The same figure shows that the speeds of the four motors and each pair of servos tend to be the same, complying with the results of the simple references. The yaw angle oscillates around the reference with an amplitude of up to 4 degrees.

Figure 4.7 shows that the path is followed closely. The average distance from the quadcopter to its reference position is 0.0703 meters.

4.2.2 Incremental Backstepping Controller

The IBKS controller results in figure 4.9 show better performance than the BKS controller, in figure 4.8, while tracking attitude references. The speed of the motors with servos have a higher demand by the IBKS controller than the two other motors, which leads to actuation saturation seen multiple times during the simulation time. Nevertheless, the controller compensates with higher servo angles on those situations.

Figure 4.7 shows that the Tilt-Quadrotor follows the path closely while changing its attitude. The average distance from the quadcopter to its reference position is 0.1080 meters, larger than the average distance for the BKS controller.

Although the IBKS controller appears to have similar performance to the BKS controller, this testing is executing using the nominal model, i.e. with no model parameter changes and no measurement noise, which may not be determinant for controller choice. Robustness to model errors and measurement noise may be decisive, hence the robustness analysis in chapter 5.

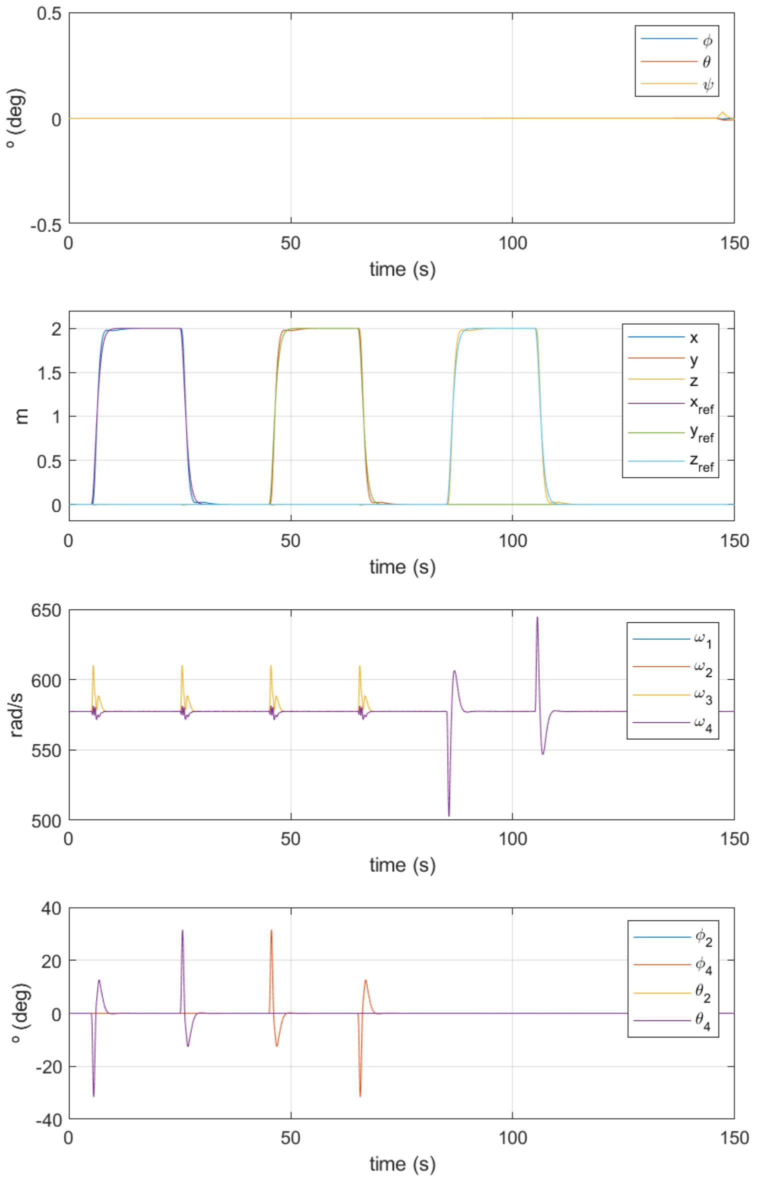

Figure 4.3: UAV response for different linear position steps using BKS with the *Newton Method*. From top to bottom: attitude, linear position and their references, motor speeds, servo angles.

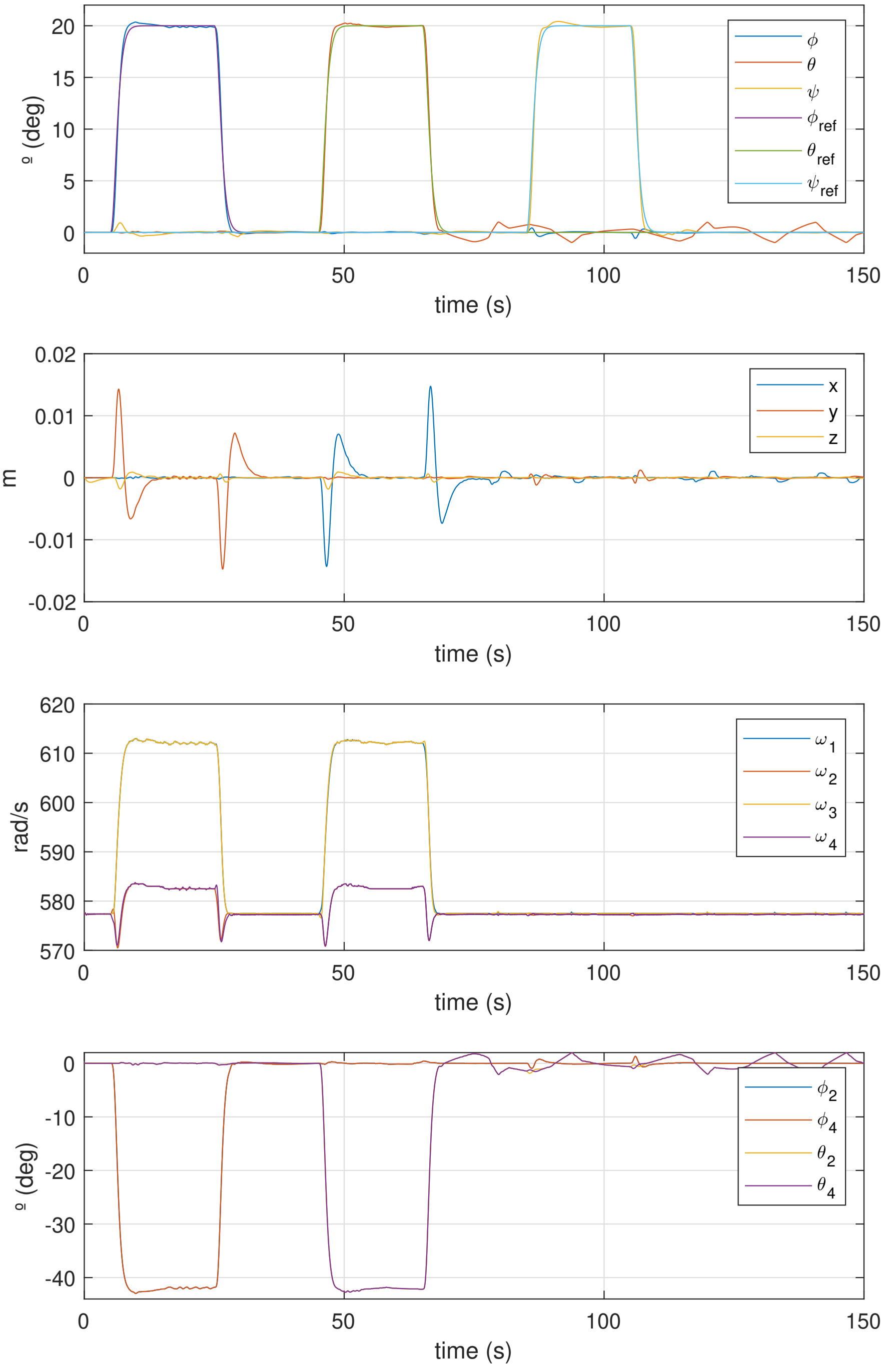

Figure 4.4: UAV response for different angular position steps using BKS with the *Newton Method*. From top to bottom: attitude and their references, linear position, motor speeds, servo angles.

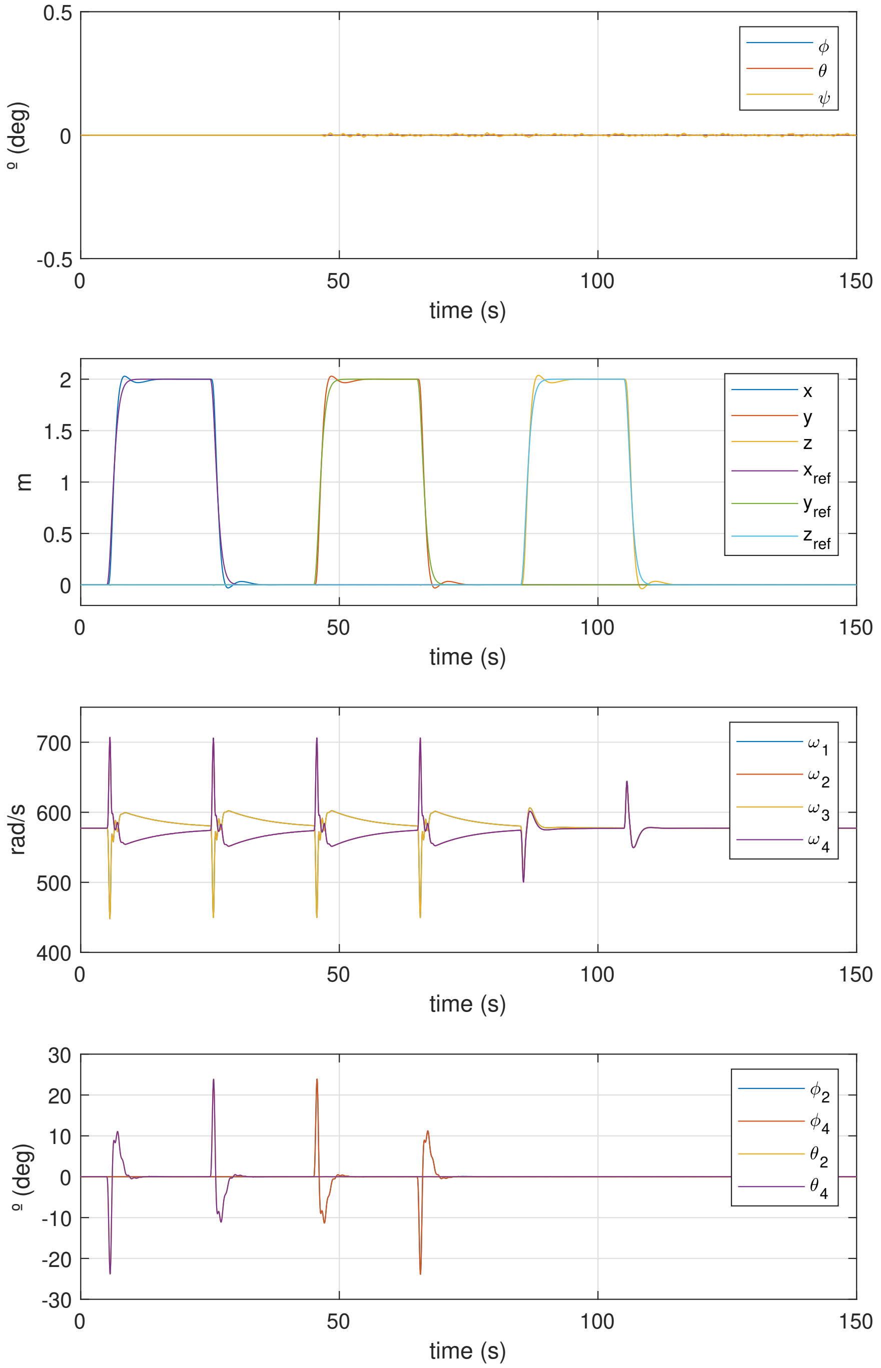

Figure 4.5: UAV response for different linear position steps using IBKS.
From top to bottom: attitude, linear position and their references, motor speeds, servo angles.

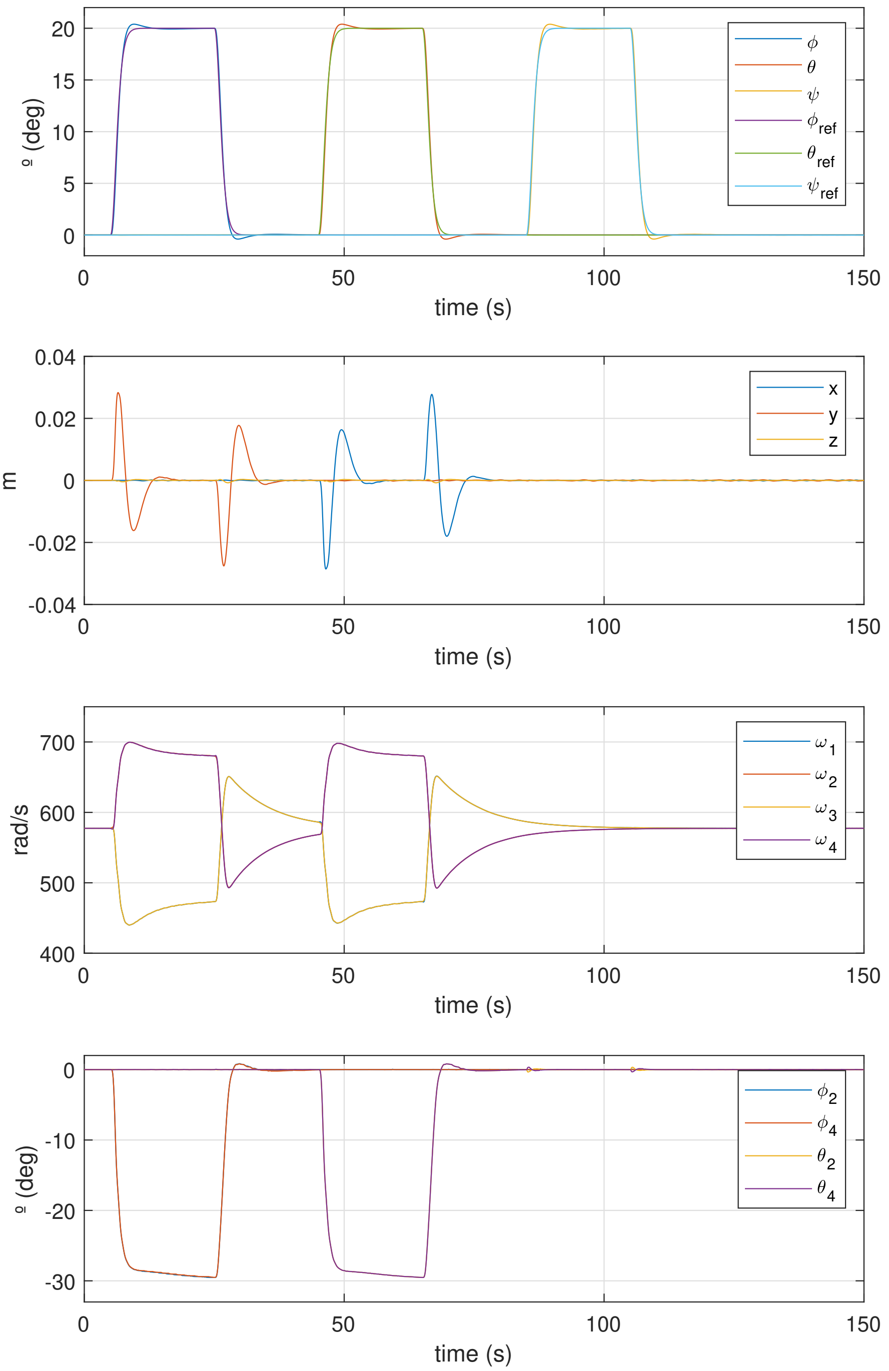

Figure 4.6: UAV response for different angular position steps using IBKS.
From top to bottom: attitude and their references, linear position, motor speeds, servo angles.

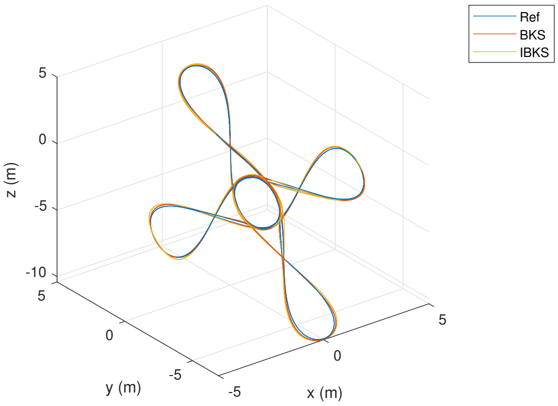

Figure 4.7: Linear Position System Response for the Coupled Reference for both BKS and IBKS solutions

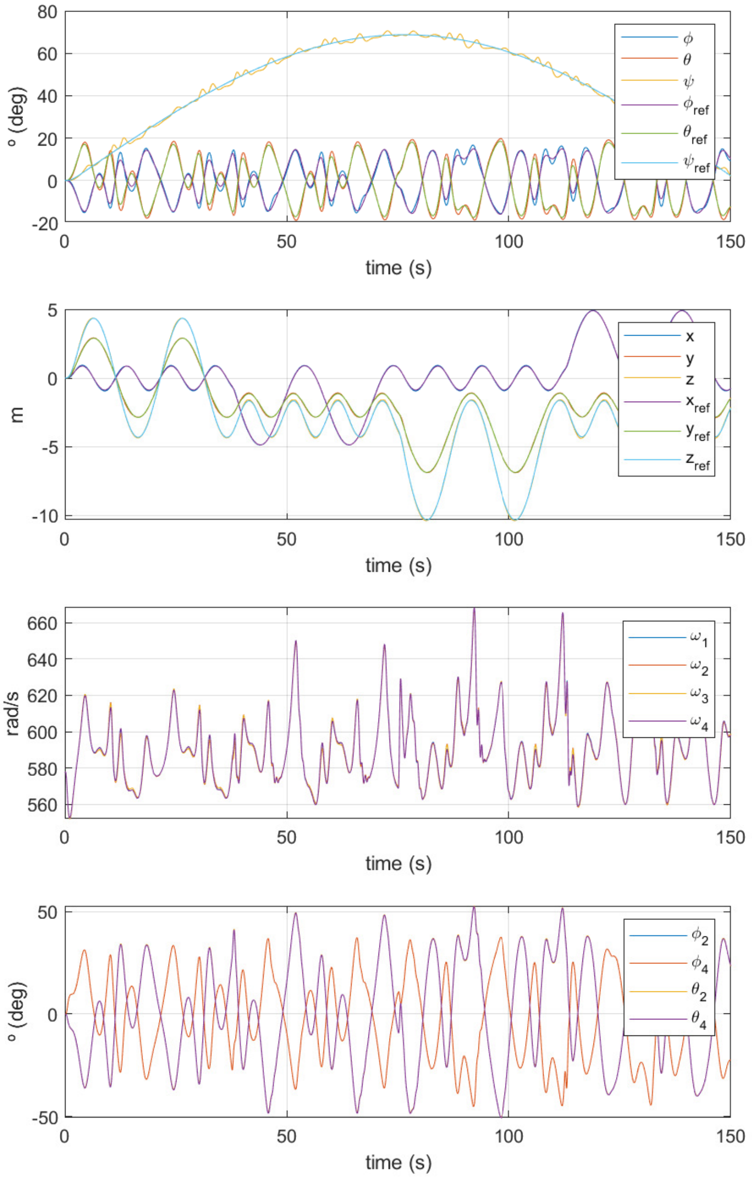

Figure 4.8: Coupled Reference BKS Controlled System Response with $\boldsymbol{K}_{Alloc}$

From Top to Bottom: Attitude, Position, Motor Speeds, Servo Angles

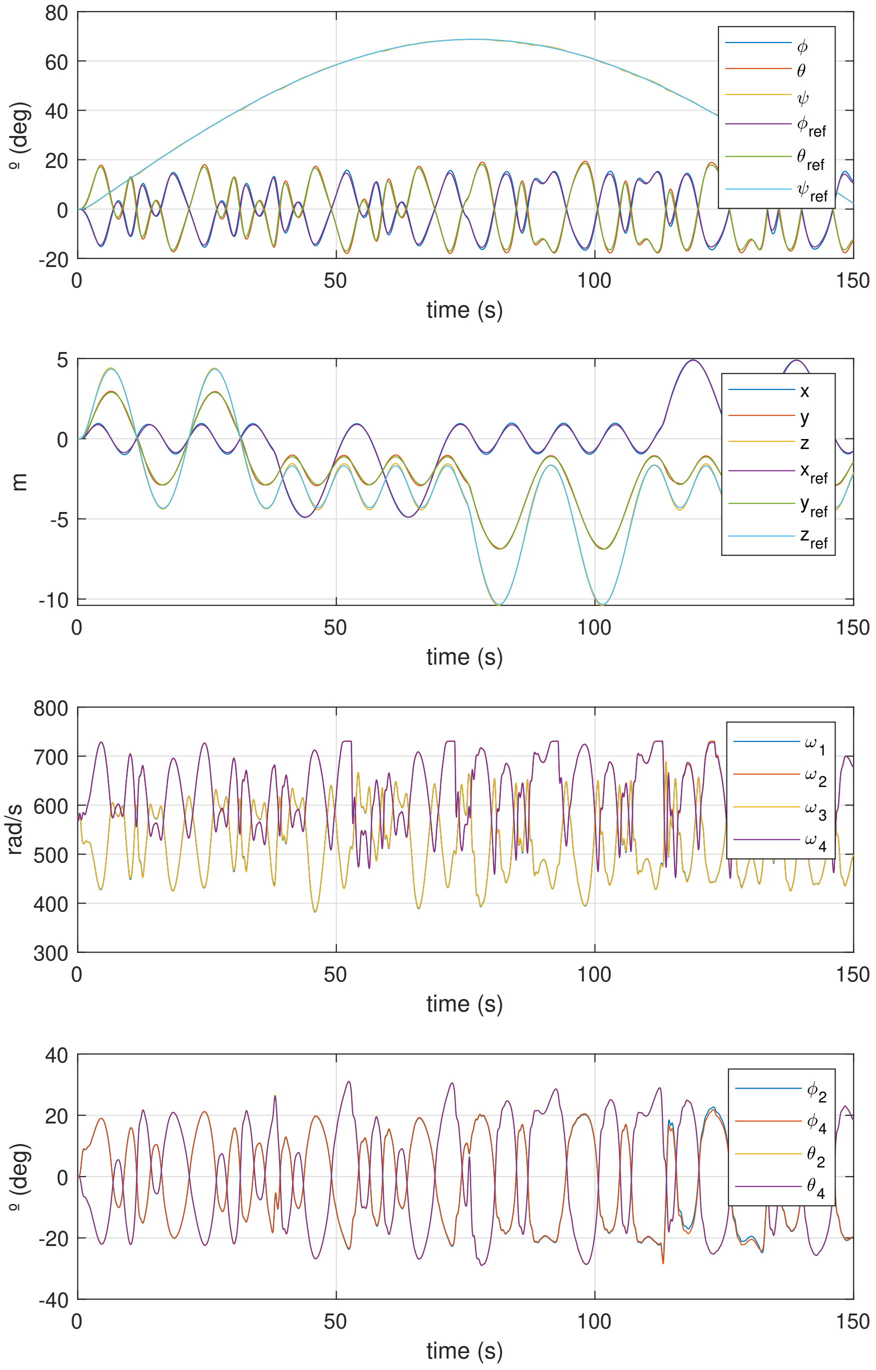

Figure 4.9: Coupled Reference IBKS Controlled System Response

From Top to Bottom: Attitude, Position, Motor Speeds, Servo Angles

Chapter 5

Robustness Analysis

When identifying system parameters, it is expected some error relative to the real values. This is due to some parameters being relatively hard to identify, see the ALIV-3 parameter identification in [30, Chapter 3], when compared with measuring the total mass with a scale. There may also be model simplifications such as assuming that the inertia matrix $\boldsymbol{J}$ is diagonal. For example, if a UAV is carrying cargo its total mass changes, so its inertia matrix (also depending where the cargo is placed) and the position of the center of gravity relative to the geometric center. Robustness to parameter uncertainty and measurement noise may be relevant and decisive for controller choice or design.

This section includes results from testing the controllers for robustness to measurement noise, parameter uncertainty in the actuator and UAV model by changing the model parameters in the simulator and using the controllers designed in sections 3.2 and 3.3.

The method used for comparison is the response root mean squared residue (RMSR) relative to the baseline response of each controller, present in figure 5.1. The RMSR is only calculated for the scenarios in which the reference is non-zero (the scenarios in the principal diagonal of the figure) between responses originated from the same controller, assessing the response difference. The response root mean squared residue is defined as:

$$RMSR = \sqrt{\frac{1}{T}\int_0^T \left(s(t) - s_b(t)\right)^2 \, dt},$$

where $s(t)$ is the state response being studied, which can stand for $\varphi(t)$, $\theta(t)$, $\psi(t)$, $x(t)$, $y(t)$ or $z(t)$, $s_b(t)$ is the state baseline response for the same state and T is the total simulation time. Or equivalently, in the discrete case

$$RMSR = \sqrt{\frac{1}{N}\sum_{k=0}^{N} \left(s(k \cdot T_s) - s_b(k \cdot T_s)\right)^2},$$

where $0 \leq k \leq N$ for $k \in \mathbb{Z}$ is the sample number, T_s is the sample time and $N \in \mathbb{N}$ is the total number of samples.

Looking at table 5.1, both controllers present response overshoot, with the IBKS controlled system having higher response overshoot than the BKS controlled system for all scenarios.

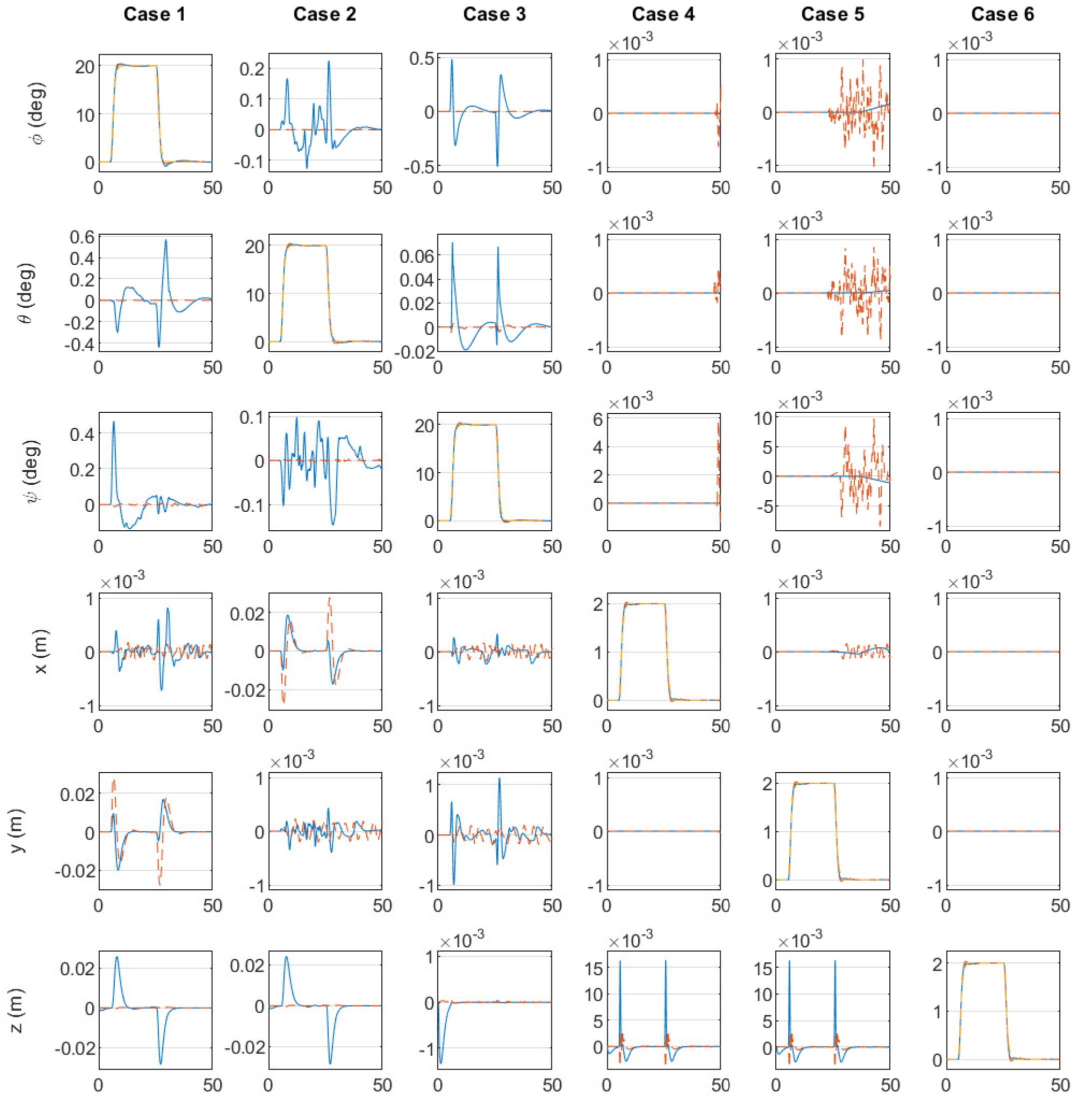

Figure 5.1: Robustness analysis baseline
Blue line for the BKS controller, orange dashed line for the IBKS controller and the yellow dashed-dot line for the reference signal

Case (state)	1 (φ)	2 (θ)	3 (ψ)	4 (x)	5 (y)	6 (z)
BKS	1.6	0.86	0.68	$5.2 \cdot 10^{-3}$	$6.6 \cdot 10^{-3}$	$2.0 \cdot 10^{-3}$
IBKS	1.96	1.96	1.95	1.44	1.44	1.78

Table 5.1: Baseline Response Overshoot Percentage Values

Figure 5.1 shows system responses for six cases for each controller. For the first case (first column of the figure), a φ step reference is requested for the platform to track and the development over time of the other five states can be seen. The second case (second column of the figure), is similar but the θ state is the one the reference tracking is requested for. ψ for the third case, x for the forth case and so on.

Analysing each of the plots in the diagonal of figure 5.1, it can be seen that both controllers show no steady-state error and have approximately the same settling time.

In each of the first three columns of the figure, it can be seen that the IBKS controlled system is generally better decoupled than the BKS controlled system, i.e. changes in one of the states has less effect on the others, with the exception for the y response in case 1 and the x response in case 2. One reason that may explain these situations is that the IBKS controlled system has delayed measurements and actuation while the BKS controlled system does not. Note that the increase in roll angle increases the force in the positive y direction and the increase in pitch angle increases the force in the negative x direction.

In the last three columns, there is an almost perfect decoupling between all variables for both controllers. For the BKS controlled system, the platform has a 15 millimeter drop (positive z position) when x or y position reference tracking is requested, while for the IBKS controlled system the drop is less than 5 millimeters.

5.1 Measurement Noise

In order to test robustness to measurement noise, noise is added to all system outputs, $\boldsymbol{s}_1$, $\boldsymbol{s}_2$ and $\dot{\boldsymbol{s}}_2$ before being fed into the controller.

The ALIV controller has embedded accelerometer, gyroscope, magnetometer and barometer [29]. The linear acceleration measurement may be obtained directly by the accelerometer and the z measurement may be obtained by the barometer, through a mathematical relation. As the ALIV controller does not have Global Posititon Systems (GPS), it is expected that the measurements are obtained by sensor fusion and filtering algorithms. However, the linear position and velocity may be obtained by integration of the accelerations and combination with the barometer measurements and their time-derivatives.

The angular position and velocity may also be obtained by sensor fusion and filtering algorithms but using the gyroscope, magnetometer and accelerometer. However, there is no included sensor for angular acceleration, then it must be derived from the angular position and velocity which means taking its time-derivative, inherently increasing high-frequency noise.

Normally distributed random noise with a variance of $4 \cdot 10^{-3}$ is added to the linear position, velocity and acceleration. These values are overestimated and it should be expected for the actual noise variance on the linear position to be smaller than for the velocity, which should be smaller than the variance for the linear acceleration.

With the same reasoning, the added noise to the attitude and angular velocity measurements has a variance of 10^{-4}. Lastly, the variance of the noise added to the angular acceleration measurement is $2 \cdot 10^{-3}$, as it is expected for the accelerations to have more measurement noise. Overestimating the angular acceleration may give a more meaningful result.

The RMSR for each case and controller can be found in table 5.2. It can be seen that the IBKS controller is more robust to measurement noise when tracking attitude references (cases one to three) than the BKS controller despite the fact that the latter does not use the linear and angular acceleration measurements, i.e. the noise from these measurements does not affect the computation of the control signal. When tracking position references (cases four to six), the two controllers have a similar response, as the RMSR have the same order of magnitude.

Control	RMSR					
	Case 1 φ (deg)	Case 2 θ (deg)	Case 3 ψ (deg)	Case 4 x (mm)	Case 5 y (mm)	Case 6 z (mm)
BKS	$1.60 \cdot 10^{-1}$	$1.24 \cdot 10^{-1}$	$9.83 \cdot 10^{-2}$	5.65	5.67	6.15
IBKS	$8.26 \cdot 10^{-2}$	$7.58 \cdot 10^{-2}$	$7.75 \cdot 10^{-2}$	6.80	6.70	5.55

Table 5.2: RMSR for Responses With Measurement Noise

5.2 Center of Gravity Deviation

Robustness to center of gravity (CG) deviation relative to the geometric center, in the body-fixed frame, is important because no mechanical construction is perfect, meaning that the weight is never perfectly centered. In order to quantify the deviation of the center of gravity, a term is added to equation (2.4.11), rewriting it as

$$\boldsymbol{J}\dot{\boldsymbol{\Omega}} = \boldsymbol{M} - \boldsymbol{\Omega} \times (\boldsymbol{J}\,\boldsymbol{\Omega}) + \underbrace{\boldsymbol{r}_{CG} \times \left(\boldsymbol{S}^T \boldsymbol{g} \cdot m\right)}_{\text{torque due to the CG deviation}} \quad , \tag{5.2.1}$$

where $\boldsymbol{r}_{CG}$ is the displacement vector from the geometric center to the center of gravity, composed of the x, y and z components $r_{CG,x}$, $r_{CG,y}$ and $r_{CG,z}$, respectively. Center of gravity deviation alters the inertia matrix $\boldsymbol{J}$. In these tests, it is assumed that the inertia matrix is correctly identified while having uncertainty about the location of the center of gravity.

The method for testing robustness to these parameters is to assign non-zero values to each one at a time.

Table 5.3 and 5.4 shows the highest RMSR values for the tested CG deviations in the x and y directions, for each interval, that do not make the system unstable. The Incremental Backstepping controller is more robust to model errors, as expected. The BKS controlled system becomes unstable with more than 2 and 1 millimeter deviation in either positive or negative direction when demanding roll angle tracking and pitch angle tracking, respectively. However, if yaw angle or linear position are demanded, the controller can stabilize and track the references despite having large peaks (up to ± 30 degrees for pitch, ± 10 degrees for roll and ± 2 meters in the x) before countering the imbalance.

The IBKS controller can stabilize the system and have a similar response for up to 20 millimeters deviation in either positive or negative direction, measured by the relatively small RMSR values in table 5.3. The system becomes unstable for higher $r_{CG,x}$ values due to saturation of the motors rather to the controller limitation.

For the Backstepping controller, deviation of the center of gravity in the y direction has a larger impact on stability than deviation in the x direction. The system becomes unstable for a deviation of 2 millimeters in the y direction (positive and negative). For a deviation of 1 millimeter, the response presents peaks of up to ± 15 degrees for roll angle, up to ± 2 degrees for pitch angle, up to ± 5 degrees for yaw angle and

$r_{CG,x}$ (mm)	Control	RMSR					
		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
		φ (deg)	θ (deg)	ψ (deg)	x (mm)	y (mm)	z (mm)
$[-1, 1]$	BKS	$8.03 \cdot 10^{-1}$	6.60	$3.08 \cdot 10^{-1}$	8.09	$7.32 \cdot 10^{-1}$	9.77
	IBKS	$1.30 \cdot 10^{-3}$	$3.83 \cdot 10^{-3}$	$4.59 \cdot 10^{-3}$	$6.54 \cdot 10^{-1}$	$3.96 \cdot 10^{-1}$	$9.14 \cdot 10^{-1}$
$[-2, 2]$	BKS	1.81	$1.28 \cdot 10^{1}$	4.09	$8.72 \cdot 10^{2}$	$5.35 \cdot 10^{1}$	$4.29 \cdot 10^{2}$
	IBKS	$2.27 \cdot 10^{-3}$	$7.69 \cdot 10^{-3}$	$5.00 \cdot 10^{-3}$	$6.17 \cdot 10^{-1}$	$3.82 \cdot 10^{-1}$	$8.97 \cdot 10^{-1}$
$[-3, 3]$	BKS	unstable	unstable	unstable	unstable	unstable	unstable
	IBKS	$3.38 \cdot 10^{-3}$	$1.29 \cdot 10^{-2}$	$5.94 \cdot 10^{-3}$	$7.91 \cdot 10^{-1}$	$7.68 \cdot 10^{-1}$	1.04
$[-20, 20]$	BKS	unstable	unstable	unstable	unstable	unstable	unstable
	IBKS	$9.98 \cdot 10^{-3}$	$5.17 \cdot 10^{-1}$	$1.70 \cdot 10^{-2}$	2.09	$6.45 \cdot 10^{-1}$	3.06

Table 5.3: RMSR for responses with center of gravity deviation in the x direction

$r_{CG,y}$ (mm)	Control	RMSR					
		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
		φ (deg)	θ (deg)	ψ (deg)	x (mm)	y (mm)	z (mm)
$[-1, 1]$	BKS	4.89	$4.80 \cdot 10^{-1}$	$6.61 \cdot 10^{-1}$	$7.78 \cdot 10^{-1}$	6.07	7.30
	IBKS	$2.38 \cdot 10^{-3}$	$1.50 \cdot 10^{-3}$	$4.57 \cdot 10^{-3}$	$6.60 \cdot 10^{-1}$	$4.67 \cdot 10^{-1}$	$9.18 \cdot 10^{-1}$
$[-2, 2]$	BKS	unstable	unstable	unstable	unstable	unstable	unstable
	IBKS	$4.35 \cdot 10^{-3}$	$2.41 \cdot 10^{-3}$	$4.68 \cdot 10^{-3}$	$7.00 \cdot 10^{-1}$	$4.88 \cdot 10^{-1}$	$8.71 \cdot 10^{-1}$
$[-3, 3]$	BKS	unstable	unstable	unstable	unstable	unstable	unstable
	IBKS	$5.94 \cdot 10^{-3}$	$3.57 \cdot 10^{-3}$	$5.75 \cdot 10^{-3}$	$9.61 \cdot 10^{-1}$	$5.27 \cdot 10^{-1}$	$9.88 \cdot 10^{-1}$
$[-20, 20]$	BKS	unstable	unstable	unstable	unstable	unstable	unstable
	IBKS	$4.34 \cdot 10^{-1}$	$1.61 \cdot 10^{-2}$	$2.74 \cdot 10^{-2}$	1.73	2.65	3.38

Table 5.4: RMSR for responses with center of gravity deviation in the y direction

up to 20 millimeters in the x, y or z directions.

Similarly to $r_{CG,x}$, $r_{CG,y}$ has a smaller effect on the Incremental Backstepping controller than on the Backstepping controller. Table 5.4 shows relatively small RMSR values for 20 millimeters CG deviation in the y direction, meaning that the IBKS controller is able to perform similarly to the baseline. However, the system becomes unstable for larger deviations in the y direction due to saturation of the motors, similarly to deviation in the x direction.

Robustness to deviation of the CG in the z direction is quite important because it may affect significantly the stability of the system. In the positive direction, downwards with respect to the body-fixed frame, it should help to stabilize the platform, i.e. maintaining zero roll and pitch angles, forcing it to behave similarly to a three-dimensional pendulum. However, in the negative direction, upwards with

$r_{CG,z}$ (mm)	Control	RMSR					
		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
		φ (deg)	θ (deg)	ψ (deg)	x (mm)	y (mm)	z (mm)
$[-60, -10]$	BKS	unstable	unstable	unstable	unstable	unstable	unstable
	IBKS	$5.01 \cdot 10^{-1}$	$7.95 \cdot 10^{-2}$	$4.11 \cdot 10^{-3}$	$2.64 \cdot 10^{-16}$	$2.69 \cdot 10^{-1}$	0
$[-5, -2]$	BKS	unstable	unstable	$1.06 \cdot 10^{-1}$	0	$8.71 \cdot 10^{-2}$	0
	IBKS	$4.40 \cdot 10^{-3}$	$6.82 \cdot 10^{-3}$	$3.65 \cdot 10^{-3}$	$6.28 \cdot 10^{-17}$	$8.94 \cdot 10^{-2}$	0
$[2, 10]$	BKS	7.65	8.48	$2.91 \cdot 10^{-1}$	0	$7.72 \cdot 10^{-2}$	0
	IBKS	$9.18 \cdot 10^{-3}$	$1.29 \cdot 10^{-2}$	$3.40 \cdot 10^{-3}$	$1.72 \cdot 10^{-16}$	$6.08 \cdot 10^{-1}$	0
$[25, 80]$	BKS	$1.12 \cdot 10^{1}$	$1.15 \cdot 10^{1}$	$1.36 \cdot 10^{-1}$	0	$7.10 \cdot 10^{-2}$	0
	IBKS	$3.57 \cdot 10^{-1}$	$1.04 \cdot 10^{-1}$	$4.10 \cdot 10^{-3}$	$3.54 \cdot 10^{-17}$	$6.34 \cdot 10^{-1}$	0

Table 5.5: RMSR for responses with center of gravity deviation in the z direction

respect to the body-fixed frame, it affects considerably the stability of the system, forcing the platform to behave similarly to an inverted pendulum.

In quadcopter design, it may be helpful to have a positive $r_{CG,z}$ for robustness while hovering. Additionally, if the platform is carrying cargo below with a rigid link, e.g. a camera, that will force a positive $r_{CG,z}$ but if that cargo is above, it may introduce a negative $r_{CG,z}$ that should be counteracted to assure system stability.

The Backstepping controller can not stabilize the system when the roll or pitch references are non-zero for negative $r_{CG,z}$. However, for up to -5 millimeters $r_{CG,z}$ the Backstepping controller is able to stabilize and track position and yaw references closely, as can be seen in table 5.5. Positive $r_{CG,z}$ does not make the system unstable up to 80 millimeters but, settling time can increase significantly for roll or pitch angle demand and even may make the system unstable due to wind-up.

The Incremental Backstepping controller is able to stabilize the platform for a wider range of $r_{CG,z}$. Actuator saturation is the limiting factor, i.e. the motors and servos saturate and the system becomes unstable, for $r_{CG,z} < -60$ mm or $r_{CG,z} > 80$ mm. Table 5.5 shows relatively small RMSR values. For the most negative $r_{CG,z}$, -60 millimeters, the largest RMSR value is for roll angle tracking, φ. In this case the φ response presents a 25% overshoot. This is due to the saturation of the motors with servos, the ones needed to counteract moments in the x-axis direction as can be seen in equation (2.3.11). Additionally, in section 4.1.2 is pointed out that the IBKS controller tends to speed those two motors up (motors 2 and 4).

5.3 Arm Length

The method used to test robustness to arm length variation is by adding (or subtracting) to the model parameter a percentage of its original value (from -75% to +100%), meaning that if that percentage is -50%, then the arm length is half of what is assumed it is for control design. This type of robustness may be relevant if it is intended to have the same controller for different platforms of different sizes.

With the Backstepping controller, for a arm length reduction of 75%, the system becomes unstable for pitch angle requests as it can not produce moments in the y-axis quickly enough to correct its pitch angle. For other attitude requests, the system presents oscillations of up to ± 10 degrees for roll and yaw angle before stabilizing. Position control does not seem to be affected, as can be seen with the relatively small values of RMSR in table 5.6. The system performs better with larger arm length than with smaller. If the arm length is doubled (+100%), it presents an overshoot of 15% when tracking yaw angle references.

The Incremental Backstepping controlled system performs closely to the baseline confirmed by the relatively small RMSR values in table 5.6. However, oppositely to the Backstepping controlled system, if the arm length is larger the system behaves worse than if it smaller, presenting overshoots of up to 15% for attitude references and oscillations in the other two angles of up to ± 3 degrees with doubled arm length.

d (%)	Control	RMSR					
		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
		φ (deg)	θ (deg)	ψ (deg)	x (mm)	y (mm)	z (mm)
$[-25, +25]$	BKS	$3.68 \cdot 10^{-1}$	$4.75 \cdot 10^{-1}$	$4.12 \cdot 10^{-1}$	$6.67 \cdot 10^{-2}$	$5.37 \cdot 10^{-2}$	0
	IBKS	$5.77 \cdot 10^{-4}$	$1.03 \cdot 10^{-3}$	$4.91 \cdot 10^{-3}$	$1.02 \cdot 10^{-1}$	$8.62 \cdot 10^{-2}$	0
$[-50, +50]$	BKS	1.35	1.34	$9.49 \cdot 10^{-1}$	$5.48 \cdot 10^{-2}$	$6.44 \cdot 10^{-2}$	0
	IBKS	$2.28 \cdot 10^{-1}$	$3.11 \cdot 10^{-3}$	$7.60 \cdot 10^{-3}$	$7.43 \cdot 10^{-2}$	1.79	0
$[-75, +75]$	BKS	4.46	unstable	3.71	$7.36 \cdot 10^{-2}$	$6.77 \cdot 10^{-2}$	0
	IBKS	$4.48 \cdot 10^{-1}$	$5.14 \cdot 10^{-3}$	$1.59 \cdot 10^{-2}$	$1.06 \cdot 10^{-1}$	2.19	0
$+100$	BKS	$6.48 \cdot 10^{-1}$	$5.42 \cdot 10^{-1}$	$6.34 \cdot 10^{-1}$	$6.94 \cdot 10^{-2}$	$7.23 \cdot 10^{-2}$	0
	IBKS	$6.00 \cdot 10^{-1}$	$1.98 \cdot 10^{-2}$	$7.63 \cdot 10^{-2}$	$1.24 \cdot 10^{-1}$	5.81	0

Table 5.6: RMSR for responses with arm length variation

5.4 Mass

Robustness to mass variation is important as there should not be a need for a controller change when improving the platform by reducing its mass or when the quadcopter is carrying cargo. When the mass is increased by +75%, the platform can not produce enough thrust to hover due to motor saturation.

The Backstepping controlled system is able to stabilize and track references for up to a 50% increase in mass with similar responses to the baseline except for when the mass is halved (presenting attitude overshoots of up to 25%). However, the RMSR values are still relatively small as seen in table 5.7, meaning that the system can still track references for all states.

The Incremental Backstepping controlled system can not track the roll and pitch references for a 50% increase in mass despite being able to track other state references, see table 5.7, due to saturation of motors 2 and 4, which this controller tends to require more effort from, see section 4.1.2. Reduced mass presents no issue for the Incremental Backstepping controlled system.

m (%)	Control	RMSR					
		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
		φ (deg)	θ (deg)	ψ (deg)	x (mm)	y (mm)	z (mm)
$[-25, +25]$	BKS	$1.89 \cdot 10^{-1}$	$2.74 \cdot 10^{-1}$	$3.86 \cdot 10^{-1}$	$1.02 \cdot 10^{-1}$	$8.36 \cdot 10^{-2}$	$1.71 \cdot 10^{1}$
	IBKS	$1.69 \cdot 10^{-1}$	$6.26 \cdot 10^{-3}$	$6.11 \cdot 10^{-3}$	8.21	7.79	5.79
$[-50, +50]$	BKS	$3.37 \cdot 10^{-1}$	$2.21 \cdot 10^{-1}$	$6.97 \cdot 10^{-1}$	$3.34 \cdot 10^{-1}$	$3.10 \cdot 10^{-1}$	$1.57 \cdot 10^{2}$
	IBKS	unstable	unstable	$7.01 \cdot 10^{-3}$	$2.88 \cdot 10^{1}$	$2.84 \cdot 10^{1}$	$1.86 \cdot 10^{2}$

Table 5.7: RMSR for responses with mass variation

5.5 Inertia

Similarly to mass variation, robustness to inertia variation is important as there should not be a need for a controller change when improving the platform or when carrying cargo. In order to test robustness to this parameter change, a percentage of the original value is added (or subtracted) to the inertia matrix, $\boldsymbol{J}$, from -75% to +100%, keeping the testing simple with no loss of generalization.

As it is expected, inertia changes have little to no effect on position reference tracking for both controllers when no change in attitude is requested, confirmed by the relatively small RMSR values in table 5.8 for the x, y and z cases.

Both IBKS and BKS controlled systems destabilize for inertia reduction by -75% when pitch reference tracking is requested. The IBKS controlled system performs similarly to the baseline for any other case, see table 5.8, and increased inertia up to +100%.

The Backstepping controlled system presents yaw angle increased overshoot for all scenarios, up to 20% for reduced inertia. The overshoot for roll and pitch angles reference tracking increases with increasing inertia for up to 20%, showing no overshoot for reduced inertia.

		RMSR					
$\boldsymbol{J}$ (%)	Control	Case 1 φ (deg)	Case 2 θ (deg)	Case 3 ψ (deg)	Case 4 x (mm)	Case 5 y (mm)	Case 6 z (mm)
$[-25, +25]$	BKS	$3.68 \cdot 10^{-1}$	$4.75 \cdot 10^{-1}$	$4.12 \cdot 10^{-1}$	$6.67 \cdot 10^{-2}$	$5.37 \cdot 10^{-2}$	0
	IBKS	$5.72 \cdot 10^{-4}$	$1.18 \cdot 10^{-3}$	$5.17 \cdot 10^{-3}$	$5.74 \cdot 10^{-2}$	$3.86 \cdot 10^{-1}$	0
$[-50, +50]$	BKS	$6.48 \cdot 10^{-1}$	$5.47 \cdot 10^{-1}$	$8.98 \cdot 10^{-1}$	$4.62 \cdot 10^{-2}$	$7.36 \cdot 10^{-2}$	0
	IBKS	$6.17 \cdot 10^{-1}$	$1.64 \cdot 10^{-2}$	$3.17 \cdot 10^{-2}$	$1.20 \cdot 10^{-1}$	$1.61 \cdot 10^{1}$	0
$[-75, +75]$	BKS	$8.77 \cdot 10^{-1}$	unstable	$8.52 \cdot 10^{-1}$	$5.41 \cdot 10^{-2}$	$7.34 \cdot 10^{-2}$	0
	IBKS	6.540	unstable	$3.22 \cdot 10^{2}$	$8.13 \cdot 10^{-2}$	$2.16 \cdot 10^{2}$	0
$+100$	BKS	1.340	1.190	1.100	$5.81 \cdot 10^{-2}$	$5.88 \cdot 10^{-2}$	0
	IBKS	$1.88 \cdot 10^{-3}$	$1.32 \cdot 10^{-3}$	$1.49 \cdot 10^{-2}$	$1.22 \cdot 10^{-1}$	$4.32 \cdot 10^{-1}$	0

Table 5.8: RMSR for responses with inertia variation

5.6 Propeller Thrust Coefficient

The robustness analysis of changes in the thrust coefficient K_T allows the evaluation of the controller if an update of the motors is performed, e.g. from 3 blade propellers to 4 blade propellers. The parameter cannot be reduced to less than 75% of its nominal value in order to ensure enough thrust to sustain the UAV weight.

The Backstepping controlled system behaves similarly to its baseline for K_T ranging from 75% of its original value to 200% (-25% to +100%) as can be seen in table 5.9.

The Incremental Backstepping controlled system performs closely to its baseline when the motors do not saturate. For roll and pitch reference tracking with a 25% decrease in K_T value, motors 2 and 4 saturate making these responses present overshoots of up to 10%.

5.7 Propeller Moment Coefficient

Variation in the propeller moment coefficient, K_Q, (from -75% to +100%) have little effect on system performance, as can be seen by the relatively low RMSR values in table 5.10. This means that both controllers are less influenced by the propeller K_Q parameter than by the K_T parameter.

K_T (%)	Control	RMSR					
		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
		φ (deg)	θ (deg)	ψ (deg)	x (mm)	y (mm)	z (mm)
$[-25, +25]$	BKS	$2.67 \cdot 10^{-1}$	$2.27 \cdot 10^{-1}$	$1.18 \cdot 10^{-1}$	$1.16 \cdot 10^{-1}$	$9.36 \cdot 10^{-2}$	$2.26 \cdot 10^{1}$
	IBKS	$4.43 \cdot 10^{-1}$	$3.62 \cdot 10^{-1}$	$4.32 \cdot 10^{-3}$	$1.41 \cdot 10^{1}$	$1.32 \cdot 10^{1}$	$1.43 \cdot 10^{1}$
$+50$	BKS	$3.12 \cdot 10^{-1}$	$2.01 \cdot 10^{-1}$	$9.68 \cdot 10^{-2}$	$9.67 \cdot 10^{-2}$	$1.00 \cdot 10^{-1}$	$2.38 \cdot 10^{1}$
	IBKS	$7.44 \cdot 10^{-4}$	$1.15 \cdot 10^{-3}$	$7.23 \cdot 10^{-3}$	2.85	2.54	8.45
$+75$	BKS	$3.55 \cdot 10^{-1}$	$2.27 \cdot 10^{-1}$	$5.61 \cdot 10^{-2}$	$1.10 \cdot 10^{-1}$	$9.41 \cdot 10^{-2}$	$3.26 \cdot 10^{1}$
	IBKS	$7.93 \cdot 10^{-4}$	$3.54 \cdot 10^{-3}$	$9.43 \cdot 10^{-3}$	3.57	3.23	$1.46 \cdot 10^{1}$
$+100$	BKS	$4.85 \cdot 10^{-1}$	$2.70 \cdot 10^{-1}$	$3.21 \cdot 10^{-1}$	$1.26 \cdot 10^{-1}$	$1.15 \cdot 10^{-1}$	$4.11 \cdot 10^{1}$
	IBKS	$1.27 \cdot 10^{-3}$	$1.69 \cdot 10^{-3}$	$8.51 \cdot 10^{-3}$	3.59	3.07	$2.42 \cdot 10^{1}$

Table 5.9: RMSR for responses with K_T variation

K_Q (%)	Control	RMSR					
		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
		φ (deg)	θ (deg)	ψ (deg)	x (mm)	y (mm)	z (mm)
$[-75, +75]$	BKS	$7.15 \cdot 10^{-2}$	$5.09 \cdot 10^{-2}$	$8.03 \cdot 10^{-2}$	$5.67 \cdot 10^{-2}$	$6.27 \cdot 10^{-2}$	0
	IBKS	$7.63 \cdot 10^{-4}$	$8.78 \cdot 10^{-4}$	$3.99 \cdot 10^{-3}$	$1.10 \cdot 10^{-1}$	$1.44 \cdot 10^{-1}$	0
$+100$	BKS	$7.83 \cdot 10^{-2}$	$4.83 \cdot 10^{-2}$	$7.84 \cdot 10^{-2}$	$7.01 \cdot 10^{-2}$	$5.38 \cdot 10^{-2}$	0
	IBKS	$3.73 \cdot 10^{-4}$	$8.10 \cdot 10^{-4}$	$4.20 \cdot 10^{-3}$	$6.05 \cdot 10^{-2}$	$9.98 \cdot 10^{-2}$	0

Table 5.10: RMSR for responses with K_Q variation

5.8 Motor Steady-State Gain

For this testing, all motors are identical and their K_ω is varied by adding a percentage of its nominal value, from -15% to 50%, similarly to the previous testing. For K_ω lower than 75% its nominal value, the motors do not produce enough thrust to sustain the weight of the platform.

Lowering K_ω worsens the performance of the system, presenting higher overshoots of up to 10% for attitude reference tracking for both BKS and IBKS controlled systems.

Increasing K_ω makes both systems track the reference more closely, presenting less overshoot than the baseline.

K_ω (%)	Control	RMSR					
		Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
		φ (deg)	θ (deg)	ψ (deg)	x (mm)	y (mm)	z (mm)
$[-15, +15]$	BKS	$2.17 \cdot 10^{-1}$	$6.08 \cdot 10^{-1}$	$1.98 \cdot 10^{-1}$	$1.05 \cdot 10^{-1}$	$1.01 \cdot 10^{-1}$	$3.36 \cdot 10^{1}$
	IBKS	$3.76 \cdot 10^{-1}$	$3.79 \cdot 10^{-1}$	$4.10 \cdot 10^{-3}$	$1.83 \cdot 10^{1}$	$1.81 \cdot 10^{1}$	$3.14 \cdot 10^{1}$
$[+25, +50]$	BKS	$5.90 \cdot 10^{-1}$	$2.96 \cdot 10^{-1}$	$1.15 \cdot 10^{-1}$	$1.27 \cdot 10^{-1}$	$1.18 \cdot 10^{-1}$	$4.05 \cdot 10^{1}$
	IBKS	$1.42 \cdot 10^{-3}$	$1.30 \cdot 10^{-2}$	$1.02 \cdot 10^{-2}$	$1.01 \cdot 10^{1}$	8.93	$1.24 \cdot 10^{1}$

Table 5.11: RMSR for responses with K_ω variation

5.9 Motor Time Constant

This section evaluates the control robustness to changes in the motors time constant τ_ω, which may become uncertain due to possible deterioration of the motors over time.

Testing shows that for small changes in τ_ω (from -25% to +75%), the IBKS controlled system is less affected than the BKS controlled system but for larger changes (-75% to -50%), the latter is less affected, as can be seen in table 5.12. It is expected for the IBKS controller to be more actuator model dependent inherently due to having to predict the state of the actuators with the command filter, see section 3.1.3.

These results mean that the motors can be up to four times as fast to respond or take 175% of the time to respond that the system is still stable and able to track attitude or position references.

		RMSR					
τ_ω (%)	Control	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
		φ (deg)	θ (deg)	ψ (deg)	x (mm)	y (mm)	z (mm)
$[-25, +25]$	BKS	$8.50 \cdot 10^{-2}$	$2.14 \cdot 10^{-1}$	$1.13 \cdot 10^{-1}$	$9.39 \cdot 10^{-2}$	$8.85 \cdot 10^{-2}$	$8.99 \cdot 10^{-1}$
	IBKS	$8.19 \cdot 10^{-4}$	$1.18 \cdot 10^{-3}$	$4.09 \cdot 10^{-3}$	$9.64 \cdot 10^{-1}$	$6.59 \cdot 10^{-1}$	$8.62 \cdot 10^{-1}$
$[-50, +50]$	BKS	$2.87 \cdot 10^{-1}$	$2.15 \cdot 10^{-1}$	$8.12 \cdot 10^{-2}$	$1.20 \cdot 10^{-1}$	$1.10 \cdot 10^{-1}$	2.00
	IBKS	$3.57 \cdot 10^{-1}$	$1.26 \cdot 10^{-2}$	$1.10 \cdot 10^{-2}$	1.37	4.93	1.60
$[-75, +75]$	BKS	$4.34 \cdot 10^{-1}$	$2.11 \cdot 10^{-1}$	$2.16 \cdot 10^{-1}$	$1.64 \cdot 10^{-1}$	$1.62 \cdot 10^{-1}$	2.47
	IBKS	1.24	4.63	$1.93 \cdot 10^{-1}$	$4.77 \cdot 10^{1}$	7.13	2.32

Table 5.12: RMSR for responses with τ_ω variation

5.10 Servo Time Constant

This section evaluates the control robustness to changes in the servos time constant τ_ν, which may become uncertain due to possible deterioration of the servos over time.

Both Backstepping controlled and Incremental Backstepping controlled systems are robust to the alteration of this parameter from -50% to +50%, confirmed by the relatively small RMSR values in table 5.13.

Despite being expected for the IBKS controller to be more dependent on the actuator model, the worst case scenario for the IBKS controlled system has lower RMSR values than for the BKS controlled system when tracking attitude references.

		RMSR					
τ_ν (%)	Control	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
		φ (deg)	θ (deg)	ψ (deg)	x (mm)	y (mm)	z (mm)
$[-25, +25]$	BKS	$1.77 \cdot 10^{-1}$	$4.58 \cdot 10^{-2}$	$8.16 \cdot 10^{-1}$	3.45	3.33	0
	IBKS	$4.60 \cdot 10^{-4}$	$8.22 \cdot 10^{-4}$	$5.36 \cdot 10^{-3}$	5.12	4.83	0
$[-50, +50]$	BKS	$1.78 \cdot 10^{-1}$	$1.13 \cdot 10^{-1}$	$4.87 \cdot 10^{-1}$	1.39	1.36	0
	IBKS	$5.25 \cdot 10^{-4}$	$9.65 \cdot 10^{-4}$	$5.72 \cdot 10^{-3}$	6.08	5.64	0

Table 5.13: RMSR for responses with τ_ν variation

5.11 Actuator Asymmetry

In reality, there are no two motors or servos exactly the same. It is relevant to know how different their action can be before the system becomes unstable. Actuator asymmetry, in this case, is tested by adding (or subtracting) the steady-state gain of a motor or servo by a percentage of its nominal value. This analysis tells if replacement or (partly) failure of an actuator makes the system become unstable.

5.11.1 Motors

Motor asymmetry robustness analysis is split into two: motors with servos and motors without servos. To represent motors with servos, motor 4 is used while motor 1 is used to represent the other pair.

The Backstepping controlled system becomes unstable for variations larger 1% applied to any of the two motors. Testing shows that motor 4 affects the response more than motor 1 because, for the same deviation, the response difference from the baseline is larger. The BKS controlled system can tolerate a difference of 1% from one motor to the others and maintain stability and reference tracking capabilities.

The Incremental Backstepping controlled system destabilizes for a larger decreases than 30% ($< -30\%$) and 10% ($< -10\%$) in K_ω of motor 1 and motor 4, respectively. This, in combination with the results for the BKS controlled system, shows that the variation of the motors with servos has a larger impact on stability than the variation of the motors without servos.

The IBKS controlled system is able to track the references for all the cases and percentages from -30% to +15% for motor 1 and from -10% to +15% for motor 4. Increased motor speed has a smaller effect than decreased motor speed, generally.

Analytically, based on the considered UAV model, it is possible for the platform to sustain a still flight when motor 1 or 3 fail, i.e. remain at zero speed. For example,

$$\boldsymbol{\Upsilon}_{\omega 1} = \begin{bmatrix} 0 & 82.21 & 8.49 & 82.21 & -5.10 & 5.10 & 0.05 & -0.05 \end{bmatrix}^T$$

originates

$$\boldsymbol{N}\left(\boldsymbol{\Upsilon}_{\omega 1}\right) = \begin{bmatrix} 3 \cdot 10^{-3} & 0 & -10^{-3} & 0 & 0 & -19 \end{bmatrix}^T,$$

which the lift force of 19 Newton is more than the necessary upward force, $-m \cdot g_0 = -18.67$ Newton, with low enough moment not to rotate uncontrollably but there would have to be way to detect motor failure.

The same is not true for the motors with servos. Despite existing a solution, $\boldsymbol{\Upsilon}_{\omega 4}$ with $\upsilon_4 = 0$, for the platform to hover, i.e. $\boldsymbol{N}\left(\boldsymbol{\Upsilon}_{\omega 1}\right) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -18.67 \end{bmatrix}^T$, there is no solution for the platform to hover and produce moments in the x or y direction. Meaning if moments are needed, due to any model imperfection or disturbance, for the platform to remain horizontal, it is likely that the system becomes unstable.

5.11.2 Servos

Servo asymmetry robustness analysis is split into two: servos that tilt the motors in the roll direction and servos that tilt the motors in the pitch direction. To represent the first and second parts, servo 1 and 4 are used, respectively.

The Backstepping controller is unable to guarantee stability when one servo has deviation from the other servos larger than 1%. For 1% decrease of the nominal servo 1 steady-state gain, the system can not track roll angle references. However, other state references are tracked for percentage deviations between -1% and 1%. For servo 4, all cases and percentage variations between -1% and 1% the references are tracked closely to the baseline.

Testing indicates that the Incremental Backstepping controlled system has higher robustness to servo asymmetry than the Backstepping controlled system. Except for φ and y reference tracking, the system can behave similarly to the baseline with a servo 1 failure (i.e. stuck at position zero degrees). With a 50% decrease in servo amplitude, the system can not track roll angle references but is able to track references for any other state. For all other percentages for servo 1, the system can still behave closely to the baseline. The system can not stabilize with servo 4 failure (i.e. stuck at position zero degrees). At 50% decrease in servo amplitude, the system response presents a 50% overshoot when tracking pitch angle references and 10 degrees roll and yaw angle oscillations before stabilizing. For percentage values between -30% and +10%, the IBKS controlled system behaves closely to the baseline.

Chapter 6

Conclusion

The main purpose of this dissertation is to propose an Incremental Backstepping controller for the ALIV platform in order to take full advantage of its maneuverability and theoretical robustness due to the actuator redundancy. A Backstepping controller is also developed due to their similarity, offering a second option for nonlinear control.

Two approaches for the control allocation of the Backstepping controller are taken into consideration, a constant matrix gain and a root-finding algorithm. The root-finding algorithm approach proved too computationally heavy for the relatively small, if any, gain in performance, and so it was not investigated further.

For the nominal model, i.e. no model parameter variations, both controllers, Backstepping and Incremental Backstepping, offer excellent stabilization and reference tracking results showing small response overshoot and a decoupled system impression despite knowing that the linear position and velocity subsystem is coupled with the attitude and angular velocity subsystem.

Concerning the nominal model, i.e. no model parameter variations, the Backstepping controller shows very similar, if not better, performance than the Incremental Backstepping, with lower computational effort, because the BKS controller uses a constant allocation matrix while the IBKS controller needs matrix inversion algorithms at each time step. However, the BKS controller may have larger wind-up problems as an integrative term is introduced in the control law, while the IBKS controller has integrative properties inherently.

The Backstepping controller splits the thrust demand between all motors in an almost identical way, which may be more energy efficient than the Incremental Backstepping controller, that splits the load unevenly between the motors, demanding more from the motors with servos. However, this approach allows for the platform to maintain stability for demands up to forty-degree roll and pitch attitude without actuator saturation, while the Backstepping controller struggles to go over thirty degrees, due to servo saturation.

Both controllers are able to accurately track combined state references. It is relevant to mention that the coupled references were of high demand. In this case, the Backstepping controller performed better at tracking the position reference while its incremental counterpart tracked the attitude reference better.

Robustness analysis shows that the Incremental Backstepping controller is extremely robust to model errors, being able to stabilize and track references with severe model changes such as the failure of the first or second servos or 30% rotational speed reduction in one of the motors without servos, relative to the other three motors.

The Backstepping controller is not as robust to model errors. If the model does not differ significantly from the real platform, it can still be applied to the platform with no additional tuning. With respect to measurement noise robustness, both show good results.

This work was successful at solving the stabilization and reference tracking problem for the ALIV Tilt-Quadrotor using these two nonlinear control design methods. However, points of improvement regarding this work are possible.

As the relations between the Incremental Backstepping (or Backstepping) controller gains and their performance are not trivial, they were chosen by testing combinations. A study relating the controller performance with the gain matrices would be of interest.

The ALIV platform is controlled digitally, meaning that the controller to implement has to be discrete and not continuous. It would be interesting to study how the sampling frequency affects both controllers, especially the Incremental Backstepping as it relies on the assumption of sufficiently low time between samples.

Bibliography

[1] A. Albert. *Regression and The Moore-Penrose Pseudoinverse*. Academic Press, 1972.

[2] R. Almeida. Incremental nonlinear dynamic inversion applied to quadrotor uav control. Master's thesis, Instituto Superior Técnico - Universidade de Lisboa, November 2017.

[3] J. Azinheira, A. Moutinho, and J. Carvalho. Lateral control of airship with uncertain dyanmics using incremental nonlinear dynamics inversion. In *IFAC-PapersOnLine*, volume 48-19, pages 69 – 74, 2015.

[4] P. A. B., E.-J. van Kampen, and Q. P. Chu. Incremental backstepping for robust nonlinear flight control. 2013.

[5] P. J. A. B. Robust nonlinear spacecraft attitude control. Master's thesis, Delft University of Technology, 2011.

[6] F. P. Beer, E. R. J. Jr., D. F. Mazurek, P. J. Cornwell, and E. R. Eisenberg. *Vector Mechanics For Engineers*. McGraw-Hill, 2010.

[7] N. Bucki and M. W. Mueller. Design and control of a passively morphing quadcopter. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9116–9122, 2019.

[8] P. Castillo-García, L. E. M. Hernandez, and P. G. Gil. Indoor navigation strategies for aerial autonomous systems. Butterworth-Heinemann, 2017.

[9] C. Chen, J. Zhang, D. Zhang, and L. Shen. Control and flight test of a tilt-rotor unmanned aerial vehicle. *International journal of Advanced Robotic Systems*, pages 1–12, 02 2017.

[10] R. Coelho. Quadrotor control using incremental nonlinear dynamics inversion. Master's thesis, Instituto Superior Técnico - Universidade de Lisboa, June 2017.

[11] R. Cordeiro, J. Azinheira, and A. Moutinho. Addressing actuation redundacies in incremental controllers for attitude tracking of fixed-wing aircraft. In *IFAC-PapersOnLine*, volume 52-12, pages 417 – 422, 2019.

[12] R. D. A. Cordeiro, J. Azinheira, and A. Moutinho. Cascaded incremental backstepping controller for the attitude tracking of fixed-wing aircraft. In *5th CEAS Conference on Guidance, Navigation and Control (EuroGNC)*, 04 2019.

[13] S. Costa. Controlo e simulação de um quadrirotor convencional. Master's thesis, Instituto Superior Técnico - Universidade de Lisboa, 2008.

[14] W. Durham, K. A. Bordignon, and R. Beck. *Aircraft Control Allocation*. John Wiley & Sons, 2017.

[15] L. Eldén. A weighted pseudoinverse, generalized singular values, and constrained least squares problems. *BIT Numerical Mathematics*, 22:487–502, 01 1982. doi: 10.1007/BF01934412.

[16] D. Falanga, K. Kleber, S. Mintchev, D. Floreano, and D. Scaramuzza. The foldable drone: A morphing quadrotor that can squeeze and fly. *IEEE Robotics and Automation Letters*, 4(2):209–216, 2019.

[17] G. P. Falconí, V. A. Marvakov, and F. Holzapfel. Fault tolerant control for a hexarotor system using incremental backstepping,. In *2016 IEEE Conference on Control Applications (CCA)*, pages 237–242, 2016.

[18] J. A. Farrell, M. Polycarpou, M. Sharma, and W. Dong. Command filtered backstepping. *IEEE Transactions on Automatic Control, vol. 54, no. 6*, June 2009.

[19] N. Fernandes. Design and construction of a multi-rotor with various degrees of freedom. Master's thesis, Instituto Superior Técnico - Universidade de Lisboa, 2011.

[20] T. Glad and L. Ljung. *Control Theory*. Taylor & Francis, 2000.

[21] N. Guerreiro. Incremental backstepping control of fixed-wing commercial aircraft under uncertainties. Master's thesis, Instituto Superior Técnico - Universidade de Lisboa, 2019.

[22] S. Gupte, Paul Infant Teenu Mohandas, and J. M. Conrad. A survey of quadrotor unmanned aerial vehicles. In *2012 Proceedings of IEEE Southeastcon*, pages 1–6, 2012.

[23] X. Huo, M. Huo, and H. R. Karimi. Attitude stabilization control of a quadrotor uav by using backstepping approach. 2014.

[24] T. Keijzer, G. Looye, Q. P. Chu, and E.-J. V. Kampen. Design and flight testing of incremental backstepping based control laws with angular accelerometer feedback. In *AIAA Scitech 2019 Forum*, January 2019.

[25] P. Kokotović and M. Arcak. Constructive nonlinear control: a historical perspective. *Automatica*, 37 (5):637 – 662, 2001. ISSN 0005-1098.

[26] E. Lima. Incremental nonlinear dynamic inversion for quadrotor control. Master's thesis, Instituto Superior Técnico - Universidade de Lisboa, November 2015.

[27] A. Lyapunov. *The General Problem of the Stability of Motion (In Russian)*. PhD thesis, University of Kharkov, 1892. English translations: (1) Stability of Motion, Academic Press, New-York & London, 1966 (2) The General Problem of the Stability of Motion, (A. T. Fuller trans.) Taylor & Francis, London 1992.

[28] T. Madani and A. Benallegue. Backstepping control for a quadrotor helicopter. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3255–3260, 2006.

[29] R. Marques. Modelling and attitude stabilization of a tilt-quadrotor. Master's thesis, Instituto Superior Técnico - Universidade de Lisboa, 2018.

[30] E. Mateos. Estimation and control of a tilt-quadrotor attitude. Master's thesis, Instituto Superior Técnico - Universidade de Lisboa, 2013.

[31] F. Pedro. Projecto preliminar de um quadrirotor. Master's thesis, Instituto Superior Técnico - Universidade de Lisboa, 2009.

[32] S. Raposo. System and process of vector propulsion with independent control of three translation and three rotation axis, 2012. U.S. patent no. US8128033B2.

[33] P. Simplício. Helicopter nonlinear flight control using incremental nonlinear dynamic inversion. Master's thesis, Instituto Superior Técnico - Universidade de Lisboa, October 2011.

[34] P. van Gils. Adaptive incremental backstepping flight control. Master's thesis, Delft University of Technology, May 2015.

[35] T. Xie, J. Zheng, and S.-C. Chu. A joint attitude control method for small unmanned aerial vehicles based on prediction incremental backstepping. *Journal of Information Hiding and Multimedia Signal Processing*, 7(2), August 2015.

[36] H. Yanai, K. Takeuchi, and Y. Takane. *Projection Matrices, Generalized Inverse Matrices, and Singular Value Decomposition*. Springer Science+Business Media, 2011.

[37] M. Zamani, N. V. de Wouw, and R. Majumdar. Backstepping controller synthesis and characterizations of incremental stability. *Systems & Control Letters*, 62(10):949 – 962, 2013.

[38] C. Zhou, J. Zhu, H. Lei, X. Yuan, and W. Wang. Robust constraint backstepping control for high-performance aircraft with account of unsteady aerodynamic effects. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 230(13):2484–2503, 2016.