Incremental Backstepping Control of an Overactuated Tilt-Quadrotor

André Macara Reis
andre.macara.reis@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

October 2020

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

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" [14].

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 [6] modeling, simulating and building the first iteration of the ALIV.

It was reiterated by Pedro [13], making the platform more rigid with a substantial emphasis on the theoretical design and improvement of the structure and the rotor performance.

The project was progressed by Fernandes [8], who built the platform using carbon fibre composite materials and obtained actuator components and avionics.

In 2013, Mateos [12], 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 [11] 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 Linear Quadratic Regulator (LQR) method. He obtained a control law for the actuators in order to stabilize the platform and validated the solution experimentally.

2. Background Theory

Command Filtered Backstepping control design and its incremental counterpart are based on Lyapunov theory, which can be studied in more detail in [9], as well as command filtering. This section includes a small introduction to the design methods and their foundations.

2.1. Backstepping

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

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

$$\dot{\boldsymbol{x}}_n = \boldsymbol{f}_n(\boldsymbol{x}) + \boldsymbol{g}_n(\boldsymbol{x}) \cdot \boldsymbol{u} \quad (2)$$

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 $\boldsymbol{w}_i^T = [\boldsymbol{x}_1^T, .., \boldsymbol{x}_i^T] \in \mathbb{R}^{ip}$. $\boldsymbol{f}_i$ and $\boldsymbol{g}_i$ are assumed to be known, bounded and differentiable.

This design method is recursive and each step can be divided into three parts [3, 10]:

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 $\boldsymbol{x}_{i+1,des}$, i.e. makes the time-derivative of the Lyapunov function, $\dot{V}_i$ negative (semi-)definite if $\boldsymbol{x}_{i+1} = \boldsymbol{x}_{i+1,des}$.

This process is repeated for each subsystem i from 1 to n, in the last step the stabilizing control law is designed directly for $\boldsymbol{u}$, i.e. $\boldsymbol{u} = \boldsymbol{x}_{n+1,des}$, 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 [3, 7, 2].

2.2. 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." [3, p. 135]

The IBKS controller design method is applied to systems in the same form as Backstepping (1), (2) with the added prerequisite that $\boldsymbol{g}_i$ must be smooth, i.e. exists higher than first order derivatives.

Applying the *Taylor* expansion to a $\dot{\boldsymbol{x}}_i$ around a previous time instant, t_0, (reference) point $\boldsymbol{x}_{i+1} = \boldsymbol{x}_{i+1,0}$ and $\boldsymbol{w}_i = \boldsymbol{w}_{i,0}$:

$$\begin{aligned}\dot{\boldsymbol{x}}_i =& \boldsymbol{f}_i(\boldsymbol{w}_{i,0}) + \boldsymbol{g}_i(\boldsymbol{w}_{i,0}) \cdot \boldsymbol{x}_{i+1,0} \\ &+ \frac{\partial}{\partial \boldsymbol{w}_i}\left[\boldsymbol{f}_i(\boldsymbol{w}_i) + \boldsymbol{g}_i(\boldsymbol{w}_i) \cdot \boldsymbol{x}_{i+1}\right](\boldsymbol{w}_i - \boldsymbol{w}_{i,0}) \\ &+ \frac{\partial}{\partial \boldsymbol{x}_{i+1}}\left[\boldsymbol{f}_i(\boldsymbol{w}_i) + \boldsymbol{g}_i(\boldsymbol{w}_i) \cdot \boldsymbol{x}_{i+1}\right](\boldsymbol{x}_{i+1} - \boldsymbol{x}_{i+1,0}) \\ &+ \text{H.O.T.}\end{aligned} \quad (3)$$

where "H.O.T." stands for Higher Order Terms, for sake of simplicity and notation, $\boldsymbol{x}_{n+1}$ is the control input $\boldsymbol{u}$ and the partial derivatives are taken for $\boldsymbol{w}_i = \boldsymbol{w}_{i,0}$ and $\boldsymbol{x}_{i+1} = \boldsymbol{x}_{i+1,0}$. For each integration step i the approximate subsystem is

$$\dot{\boldsymbol{x}}_i \approx \dot{\boldsymbol{x}}_{i,0} + \boldsymbol{A}_{i,0} \cdot \Delta\boldsymbol{w}_i + \boldsymbol{B}_{i,0} \cdot \Delta\boldsymbol{x}_{i+1,0} \quad (4)$$

where

$$\dot{\boldsymbol{x}}_{i,0} = \boldsymbol{f}_i(\boldsymbol{w}_{i,0}) + \boldsymbol{g}_i(\boldsymbol{w}_{i,0}) \cdot \boldsymbol{x}_{i+1,0}$$

$$\boldsymbol{A}_{i,0} = \frac{\partial}{\partial \boldsymbol{w}_i}\left[\boldsymbol{f}_i(\boldsymbol{w}_i) + \boldsymbol{g}_i(\boldsymbol{w}_i) \cdot \boldsymbol{x}_{i+1}\right]$$

$$\boldsymbol{B}_{i,0} = \boldsymbol{g}_i(\boldsymbol{w}_{i,0})$$

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

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

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:

$$\dot{\boldsymbol{x}}_i \approx \dot{\boldsymbol{x}}_{i,0} + \boldsymbol{B}_{i,0} \cdot \Delta\boldsymbol{x}_{i+1} \quad (5)$$

Note that the term on $\boldsymbol{f}_i(\boldsymbol{w}_i)$ 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, $\boldsymbol{f}_i(\boldsymbol{w}_i)$ may not be known. However, $\boldsymbol{g}_i(\boldsymbol{w}_i)$ 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.

2.3. Command Filtering

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" [7, 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. Further analysis on command filtering can be found in [7].

3. Tilt-Quadrotor Model and Controller Implementation

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.

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.

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, 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.

3.1. Actuators Model

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.

3.1.1. Brushless Motors

The motors present in the ALIV3 platform are BrushLess Direct Current (BLDC) motors. These require a driver, Electronic Speed Controller (ESC). The desired angular speed is sent to the ESCs digitally 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:

$$\omega_i = \frac{K_\omega}{\tau_\omega s + 1} u_{\omega i}, \tag{6}$$

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

$$u_{\omega i} = PWM_{\omega i} - u_{\omega,d}, \tag{7}$$

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.

3.1.2. 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.

$$\boldsymbol{\nu}^T = \begin{bmatrix} \phi_2 & \phi_4 & \vartheta_2 & \vartheta_4 \end{bmatrix}, \tag{8}$$

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

$$\nu_i = \frac{K_\nu}{\tau_\nu s + 1} u_{\nu i}, \tag{9}$$

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.

$$u_{\nu i} = PWM_{\nu i} - \frac{PWM_{min} + PWM_{max}}{2}, \tag{10}$$

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

3.1.3. Propellers

The Blade Momentum Theory [5] states that the thrust T and the torque Q produced by a propeller are

$$T = C_T \rho A_p r^2 \omega^2, \tag{11a}$$
$$Q = C_Q \rho A_p r^3 \omega^2, \tag{11b}$$

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:

$$T_i = K_T \omega_i^2, \tag{12a}$$
$$Q_i = K_Q \omega_i^2, \tag{12b}$$

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.

3.1.4. 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 [6, 11]:

$$\boldsymbol{F}^T = \begin{bmatrix} F_x & F_y & F_z \end{bmatrix} \tag{13}$$

$$F_x = -K_T \left(\omega_2^2 \sin\vartheta_2 \cos\phi_2 + \omega_4^2 \sin\vartheta_4 \cos\phi_4 \right) \tag{14}$$

$$F_y = K_T \omega_2^2 \sin\phi_2 \cos\vartheta_2 + \omega_4^2 \sin\phi_4 \cos\vartheta_4 \tag{15}$$

$$F_z = -K_T \left(\omega_1^2 + \omega_2^2 \cos\phi_2 \cos\vartheta_2 + \omega_3^2 + \omega_4^2 \cos\phi_4 \cos\vartheta_4 \right) \tag{16}$$

$$\boldsymbol{M}^T = \begin{bmatrix} M_x & M_y & M_z \end{bmatrix} \tag{17}$$

$$\begin{aligned} M_x = & \omega_2^2 \cos\phi_2 \left(dK_T \cos\vartheta_2 + K_Q \sin\vartheta_2 \right) \\ & - \omega_4^2 \cos\phi_4 \left(dK_T \cos\vartheta_4 + K_Q \sin\vartheta_4 \right) \end{aligned} \tag{18}$$

$$\begin{aligned} M_y = & dK_T \left(\omega_1^2 - \omega_3^2 \right) - K_Q \omega_2^2 \cos\vartheta_2 \sin\phi_2 \\ & + K_Q \omega_4^2 \cos\vartheta_4 \sin\phi_4 \end{aligned} \tag{19}$$

$$\begin{aligned} M_z = & K_Q \left(\omega_3^2 - \omega_1^2 \right) + \omega_2^2 \cos\phi_2 \left(K_Q \cos\vartheta_2 - dK_T \sin\vartheta_2 \right) \\ & + \omega_4^2 \cos\phi_4 \left(dK_T \sin\vartheta_4 - K_Q \cos\vartheta_4 \right) \end{aligned} \tag{20}$$

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

3.2. Equations of Motion

The ALIV3 platform equations of motion can be separated into two parts: Kinematics and Dynamics.

3.2.1. 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 [4].

The relative position of quadrotor to the inertial frame is:

$$\boldsymbol{P} = \begin{bmatrix} x & y & z \end{bmatrix}^T. \tag{21}$$

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

$$\boldsymbol{V} = \begin{bmatrix} u & v & w \end{bmatrix}^T. \tag{22}$$

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

$$\boldsymbol{\Phi} = \begin{bmatrix} \varphi & \theta & \psi \end{bmatrix}^T \tag{23}$$

The angular velocity is defined by:

$$\boldsymbol{\Omega} = \begin{bmatrix} p & q & r \end{bmatrix}^T, \tag{24}$$

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

$$\boldsymbol{S} = \begin{bmatrix} \cos\theta\cos\psi & \cos\psi\sin\theta\sin\varphi - \cos\varphi\sin\psi & \sin\varphi\sin\psi + \cos\varphi\cos\psi\sin\theta \\ \cos\theta\sin\psi & \cos\varphi\cos\psi + \sin\varphi\sin\theta\sin\psi & \cos\varphi\sin\theta\sin\psi - \cos\psi\sin\varphi \\ -\sin\theta & \cos\theta\sin\varphi & \cos\varphi\cos\theta \end{bmatrix}. \tag{25}$$

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

$$\dot{\boldsymbol{P}} = \boldsymbol{S}\,\boldsymbol{V} \tag{26}$$

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

$$\dot{\boldsymbol{\Phi}} = \boldsymbol{T}\,\boldsymbol{\Omega}, \tag{27}$$

where

$$\boldsymbol{T} = \begin{bmatrix} 1 & \sin\varphi\tan\theta & \cos\varphi\tan\theta \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi\sec\theta & \cos\varphi\sec\theta \end{bmatrix}. \tag{28}$$

3.2.2. 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 body-fixed frame [4]

$$\dot{\boldsymbol{V}} = \frac{1}{m}\boldsymbol{F} - \boldsymbol{\Omega} \times \boldsymbol{V} + \boldsymbol{S}^T\boldsymbol{g} \tag{29}$$

where $\boldsymbol{g} = \begin{bmatrix} 0 & 0 & g_0 \end{bmatrix}$, $g_0 = 9.81\,m/s^2$ is the Earth gravitational acceleration, $\boldsymbol{F}$ is composed by the three components of the forces applied to the quadrotor defined in (13) and $\boldsymbol{S}$ the rotation matrix from the body-fixed frame to the inertial frame, defined in (25).

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

$$\boldsymbol{J}\dot{\boldsymbol{\Omega}} = \boldsymbol{M} - \boldsymbol{\Omega} \times (\boldsymbol{J}\,\boldsymbol{\Omega}) \tag{30}$$

where M is composed by the three components of the external moments applied to the platform defined in (17) and $\boldsymbol{J}$ is its inertia matrix.

3.3. Backstepping Controller

In this section, a controller for the Tilt-Quadrotor model is 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

$$\begin{bmatrix} \dot{\boldsymbol{\Phi}} \\ \dot{\boldsymbol{P}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{T} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{S} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Omega} \\ \boldsymbol{V} \end{bmatrix} \tag{31}$$

$$\begin{bmatrix} \dot{\boldsymbol{\Omega}} \\ \dot{\boldsymbol{V}} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{J}^{-1}\left(\boldsymbol{\Omega} \times (\boldsymbol{J}\boldsymbol{\Omega})\right) \\ -\boldsymbol{\Omega} \times \boldsymbol{V} + \boldsymbol{S}^T \cdot \boldsymbol{g} \end{bmatrix} + \begin{bmatrix} \boldsymbol{J}^{-1} & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & m^{-1} \cdot \boldsymbol{I}_3 \end{bmatrix} \begin{bmatrix} \boldsymbol{M} \\ \boldsymbol{F} \end{bmatrix} \tag{32}$$

The dynamics above is equivalent to:

$$\dot{\boldsymbol{s}}_1 = \boldsymbol{g}_1(\boldsymbol{s}_1) \cdot \boldsymbol{s}_2 \tag{33}$$

$$\dot{\boldsymbol{s}}_2 = \boldsymbol{f}_2(\boldsymbol{s}_1, \boldsymbol{s}_2) + \boldsymbol{g}_2\boldsymbol{N} \tag{34}$$

which is in the form of (1) and (2). Note that $\boldsymbol{f}_1 \equiv \boldsymbol{0}_{6\times1}$, $\boldsymbol{g}_2$ 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.

3.3.1. Kinematics Loop

As aforementioned, in section 2.1, the first part of each step is to define the tracking error

$$\boldsymbol{\zeta}_1 = \begin{bmatrix} \boldsymbol{\zeta}_\Phi \\ \boldsymbol{\zeta}_p \end{bmatrix} = \boldsymbol{s}_1 - \boldsymbol{s}_{1,des} = \begin{bmatrix} \boldsymbol{\Phi} \\ \boldsymbol{P} \end{bmatrix} - \begin{bmatrix} \boldsymbol{\Phi}_{des} \\ \boldsymbol{P}_{des} \end{bmatrix}, \tag{35}$$

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

$$\dot{\boldsymbol{\zeta}}_1 = \boldsymbol{g}_1(\boldsymbol{s}_1)\boldsymbol{s}_2 - \dot{\boldsymbol{s}}_{1,des} \tag{36}$$

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 ζ_1, i.e.

$$V_1 = \frac{1}{2}\zeta_1^T\zeta_1 + \frac{1}{2}\gamma^T \boldsymbol{K}_i\gamma, \tag{37}$$

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.

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

The integrative term is defined as:

$$\gamma = \begin{bmatrix} \gamma_\Phi \\ \gamma_p \end{bmatrix} = \int_0^t \zeta_1\, dt. \tag{38}$$

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 $\boldsymbol{s}_2 = \boldsymbol{s}_{2,des}$.

$$\begin{aligned} \dot{V}_1 &= \boldsymbol{\zeta}_1^T\left(\boldsymbol{g}_1(\boldsymbol{s}_1)\boldsymbol{s}_2 - \dot{\boldsymbol{s}}_{1,des}\right) \\ &= \boldsymbol{\zeta}_\Phi^T\left(\boldsymbol{T}\,\boldsymbol{\Omega} - \dot{\boldsymbol{\Phi}}_{des} + \boldsymbol{K}_{\Phi i}\boldsymbol{\gamma}_\Phi\right) \\ &\quad + \boldsymbol{\zeta}_p^T\left(\boldsymbol{S}\boldsymbol{V} - \dot{\boldsymbol{P}}_{des} + \boldsymbol{K}_{pi}\boldsymbol{\gamma}_p\right) \end{aligned} \tag{39}$$

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

$$\boldsymbol{\Omega}_{des} = \boldsymbol{T}^{-1}\left(-\boldsymbol{K}_\Phi\boldsymbol{\zeta}_\Phi - \boldsymbol{K}_{\Phi i}\boldsymbol{\gamma}_\Phi + \dot{\boldsymbol{\Phi}}_{des}\right) \tag{40}$$

$$\boldsymbol{V}_{des} = \boldsymbol{S}^T\left(-\boldsymbol{K}_p\boldsymbol{\zeta}_p - \boldsymbol{K}_{pi}\boldsymbol{\gamma}_p + \dot{\boldsymbol{P}}_{des}\right) \tag{41}$$

where $\boldsymbol{K}_\Phi$ and $\boldsymbol{K}_p$ are positive-definite matrices, then, when $\boldsymbol{s}_2 = \boldsymbol{s}_{2,des}$, $\dot{V}_1$ is simplified to

$$-\boldsymbol{\zeta}_1^T\boldsymbol{K}_1\boldsymbol{\zeta}_1 \leq 0$$

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 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

$$\boldsymbol{T}^{-1} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\varphi & \cos\theta\sin\varphi \\ 0 & -\sin\varphi & \cos\varphi\cos\theta \end{bmatrix} \tag{42}$$

3.3.2. Dynamics Loop

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

$$\boldsymbol{\zeta}_2 = \begin{bmatrix}\boldsymbol{\zeta}_\Omega \\ \boldsymbol{\zeta}_v\end{bmatrix} = \boldsymbol{s}_2 - \boldsymbol{s}_{2,des,f} = \begin{bmatrix}\boldsymbol{\Omega} \\ \boldsymbol{V}\end{bmatrix} - \begin{bmatrix}\boldsymbol{\Omega}_{des,f} \\ \boldsymbol{V}_{des,f}\end{bmatrix}, \tag{43}$$

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

$$\dot{\boldsymbol{\zeta}}_2 = \boldsymbol{f}_2(\boldsymbol{s}_1, \boldsymbol{s}_2) + \boldsymbol{g}_2\boldsymbol{N} - \dot{\boldsymbol{s}}_{2,des,f}. \tag{44}$$

With $\boldsymbol{\zeta}_2$ defined, (39) 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 $(\boldsymbol{\zeta}_v + \boldsymbol{V}_{des,f} + \boldsymbol{V}_{des} - \boldsymbol{V}_{des})$, simplifying $\dot{V}_1$ to

$$\begin{aligned} \dot{V}_1 = &\underbrace{-\boldsymbol{\zeta}_1^T\boldsymbol{K}_1\boldsymbol{\zeta}_1}_{\leq 0} + \boldsymbol{\zeta}_1^T\boldsymbol{g}_1\left(\boldsymbol{s}_1\right)\boldsymbol{\zeta}_2 \\ &+ \boldsymbol{\zeta}_1^T\boldsymbol{g}_1\left(\boldsymbol{s}_1\right)\left(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des}\right) \end{aligned} \tag{45}$$

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

$$V_2 = V_1 + \frac{1}{2}\boldsymbol{\zeta}_2^T\boldsymbol{\zeta}_2 \tag{46}$$

The third part is to design a virtual stabilizing control law, $\boldsymbol{N}_{des}$, that makes $\dot{V}_2$ negative definite for $\boldsymbol{\zeta}_1,\, \boldsymbol{\zeta}_2 \neq 0$ when $\boldsymbol{N} = \boldsymbol{N}_{des}$, $\dot{V}_2$ is simplified to

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

In order to guarantee stability, one needs to compensate the difference $(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des})$ with $\boldsymbol{\zeta}_{1,c}$ [7], defined by

$$\begin{aligned} &\dot{\boldsymbol{\zeta}}_{1,c} = -\boldsymbol{K}_1\boldsymbol{\zeta}_{1,c} - \boldsymbol{K}_i\boldsymbol{\gamma} + \boldsymbol{g}_1(\boldsymbol{s}_1)\left(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des}\right) \\ &\boldsymbol{\zeta}_{1,c}(t=0) = \boldsymbol{0} \end{aligned} \tag{47}$$

If $\boldsymbol{N}_{des}$ is defined as

$$\boldsymbol{N}_{des} = \begin{bmatrix} \boldsymbol{\Omega}\times(\boldsymbol{J}\boldsymbol{\Omega}) + \boldsymbol{J}\left(\dot{\boldsymbol{\Omega}}_{des,f} - \boldsymbol{K}_\Omega\boldsymbol{\zeta}_\Omega\right) \\ m\cdot\left(\dot{\boldsymbol{V}}_{des,f} + \boldsymbol{\Omega}\times\boldsymbol{V} - \boldsymbol{S}^T\boldsymbol{g} - \boldsymbol{K}_v\boldsymbol{\zeta}_v\right) \end{bmatrix} - \boldsymbol{g}_2^{-1}\cdot\boldsymbol{g}_1^T(\boldsymbol{s}_1)(\boldsymbol{\zeta}_1 - \boldsymbol{\zeta}_{1,c}), \tag{48}$$

where $\boldsymbol{K}_\Omega$ and $\boldsymbol{K}_v$ are positive definite matrices, then the control law satisfies the conditions for system stability, more information can be found in [7]. The tracking error dynamics is

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

where

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

3.3.3. Control Allocation

Above, the expressions for the desired applied moments and forces to the platform were defined and it is now necessary to know what motor speeds and servo angles produce them.

$$\Delta \boldsymbol{f}(\boldsymbol{x}) \approx \frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}} \cdot \Delta \boldsymbol{x}, \tag{49}$$

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

$$\upsilon_i = \begin{cases} S\left(\frac{1}{10}\,\omega_i, 0, 100\right) & \text{, for } i = 1,2,3,4 \\ S\left(\frac{300}{\pi}\,\nu_{i-4}, -100, 100\right) & \text{, for } i = 5,6,7,8 \end{cases} \tag{50}$$

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

$$\boldsymbol{\Upsilon} = \begin{bmatrix} \upsilon_1 & \upsilon_2 & \upsilon_3 & \upsilon_4 & \upsilon_5 & \upsilon_6 & \upsilon_7 & \upsilon_8 \end{bmatrix}^T. \tag{51}$$

To know what value of $\boldsymbol{\Upsilon}$ produces $\boldsymbol{N}_{des}$, use the linear approximation of $\boldsymbol{N}(\boldsymbol{\Upsilon})$:

$$\boldsymbol{\Upsilon} = \boldsymbol{\Upsilon}_0 + \Delta\boldsymbol{\Upsilon} \tag{52}$$

$$\Delta\boldsymbol{\Upsilon} = -\left[\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}}\bigg|_{\boldsymbol{\Upsilon}_0}\right]^\dagger \cdot (\boldsymbol{N}(\boldsymbol{\Upsilon}_0) - \boldsymbol{N}_{des}) \tag{53}$$

where † denotes a pseudo-inverse, and $\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}}\big|_{\boldsymbol{\Upsilon}_0}$ is the partial derivative of $\boldsymbol{N}$ with respect to $\boldsymbol{\Upsilon}$ at a given point $\boldsymbol{\Upsilon}$, which is defined analytically. For constant $\boldsymbol{\Upsilon}_0$, the Jacobian matrix, $\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}}\big|_{\boldsymbol{\Upsilon}_0}$, is constant as so is its pseudo-inverse. Meaning

$$\boldsymbol{\Upsilon} = \boldsymbol{\Upsilon}_0 + \boldsymbol{K}_{Alloc}\left(\boldsymbol{N}(\boldsymbol{\Upsilon}_0) - \boldsymbol{N}_{des}\right), \tag{54}$$

where

$$\boldsymbol{K}_{Alloc} = -\left[\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}}\bigg|_{\boldsymbol{\Upsilon}_0}\right]^\dagger. \tag{55}$$

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

$$PWM_{\omega i} = a\upsilon_i + u_{\omega,d}, i = 1,2,3,4 \tag{56}$$

$$PWM_{\nu i} = b\upsilon_{i+4} + c, i = 1,2,3,4 \tag{57}$$

where

$$a = \frac{PWM_{max} - u_{\omega,d}}{100}$$

$$b = \frac{PWM_{max} - PWM_{min}}{200}$$

$$c = \frac{PWM_{max} + PWM_{min}}{2}$$

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 $\upsilon_{i+4} = -100$, $i = 1,2,3,4$.

3.4. Incremental Backstepping Controller

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 3.3.1. 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.

3.4.1. Dynamics Loop

The second step is similar to the BKS controller design up to the $\boldsymbol{s}_2$ dynamics, which is approximated to

$$\begin{aligned} \dot{\boldsymbol{s}}_2 &\approx \dot{\boldsymbol{s}}_{2,0} + \boldsymbol{g}_2 \cdot \Delta\boldsymbol{N} \\ &\approx \dot{\boldsymbol{s}}_{2,0} + \boldsymbol{g}_2 \cdot \left[\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}}\bigg|_{\boldsymbol{\Upsilon}_0}\right] \cdot \Delta\boldsymbol{\Upsilon} \end{aligned} \tag{58}$$

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

$$\dot{\boldsymbol{\zeta}}_2 \approx \dot{\boldsymbol{s}}_{2,0} + \boldsymbol{g}_2 \cdot \left[\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}}\bigg|_{\boldsymbol{\Upsilon}_0}\right] \cdot \Delta\boldsymbol{\Upsilon} - \dot{\boldsymbol{s}}_{2,des,f}. \tag{59}$$

For notation simplicity define

$$\boldsymbol{B}_0 = \overline{\boldsymbol{B}}_0 \cdot \left[\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}}\bigg|_{\boldsymbol{\Upsilon}_0}\right],\ \overline{\boldsymbol{B}}_0 = \boldsymbol{g}_2$$

Similarly to the Backstepping design method, with $\boldsymbol{\zeta}_2$ defined, $\dot{V}_1$ and V_2 can be rewritten in the same way, (45) and (46), respectively.

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

$$\begin{aligned} \dot{V}_2 &= \dot{V}_1 + \boldsymbol{\zeta}_2^T\left(\dot{\boldsymbol{s}}_{2,0} - \dot{\boldsymbol{s}}_{2,des,f} + \boldsymbol{B}_0 \cdot \Delta\boldsymbol{\Upsilon}\right) \\ &= -\boldsymbol{\zeta}_1^T \boldsymbol{K}_1 \boldsymbol{\zeta}_1 + \boldsymbol{\zeta}_2^T\left(\dot{\boldsymbol{s}}_{2,0} - \dot{\boldsymbol{s}}_{2,des,f} + \boldsymbol{g}_1^T(\boldsymbol{s}_1)\,\boldsymbol{\zeta}_1 + \boldsymbol{B}_0 \cdot \Delta\boldsymbol{\Upsilon}\right) \\ &\quad + \boldsymbol{\zeta}_1^T \boldsymbol{g}_1(\boldsymbol{s}_1)\left(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des}\right) \end{aligned} \tag{60}$$

In order to guarantee stability, one needs to compensate the difference $(\boldsymbol{s}_{2,des,f} - \boldsymbol{s}_{2,des})$ with $\boldsymbol{\zeta}_{1,c}$ [7], defined by (47).

3.4.2. Matrix Pseudoinverse

In order to be able to compute $\Delta\boldsymbol{\Upsilon}$, it is firstly needed to compute $\boldsymbol{B}_0^{\dagger}$, where $\dagger$ represents a pseudo-inverse. 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

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

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

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

where δ_Υ is a scalar whose magnitude is less than the smallest non-zero eigenvalue of $\boldsymbol{X}_0\boldsymbol{X}_0^T$, guaranteeing that $(\boldsymbol{X}_0\boldsymbol{X}_0^T + \delta_\Upsilon^2\boldsymbol{I}_6)$ is non-singular, its inverse exists and approximately equal to $\boldsymbol{X}_0^{\dagger}$.

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 the solution 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 [15, page 80]

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

with arbitrary $\boldsymbol{z}$, which means that it 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.

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

$$\boldsymbol{X}_0^{\dagger} = \left[\left.\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}}\right|_{\boldsymbol{\Upsilon}_0}\right]^T \left(\left.\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}}\right|_{\boldsymbol{\Upsilon}_0} \cdot \left[\left.\frac{\partial \boldsymbol{N}}{\partial \boldsymbol{\Upsilon}}\right|_{\boldsymbol{\Upsilon}_0}\right]^T + \delta_\Upsilon^2 \cdot \boldsymbol{I}_6\right)^{-1} \tag{61}$$

$$\begin{aligned}\Delta\boldsymbol{\Upsilon} = \boldsymbol{X}_0^{\dagger}\overline{\boldsymbol{B}}_0^{-1}\left(-\dot{\boldsymbol{s}}_{2,0} + \dot{\boldsymbol{s}}_{2,des,f} - \boldsymbol{g}_1^T\left(\boldsymbol{\zeta}_1 - \boldsymbol{\zeta}_{1,c}\right) - \boldsymbol{K}_2\boldsymbol{\zeta}_2\right) \\ + \left(\boldsymbol{I}_8 - \boldsymbol{X}_0^{\dagger}\boldsymbol{X}_0\right)\boldsymbol{K}_\Upsilon\left(\boldsymbol{\Upsilon}_0 - \boldsymbol{\Upsilon}_r\right)\end{aligned} \tag{62}$$

$$\boldsymbol{\Upsilon} = \boldsymbol{\Upsilon}_0 + \Delta\boldsymbol{\Upsilon} \tag{63}$$

Achieving the stability properties in [7] (where more information can be found), assuming model correctness and sufficiently high control update and sampling frequency, but also avoiding actuator saturation.

The PWM signals can then be calculated as in (56) and (57).

4. Simulation Results

For a first performance comparison, a series of step references are given to the controllers. Both control strategies show excellent results, with zero steady-state error for the three Cartesian coordinates and three Euler angles.

The Backstepping controlled system has a response overshoot of 1.6% for φ reference tracking and less than 0.86% for θ or ψ tracking. Position reference tracking has smaller response overshoot, up to 0.0066%.

The Incremental Backstepping controlled system has higher response overshoot: approximately 1.96% for all attitude, 1.78% for z and 1.44% for x or y reference tracking.

In order to compare control allocation and system decoupling, the responses to φ and x tracking, including the actuation values are recorded in figures 1 and 2. Note that only two motor and two servo responses are included for each controller response. This way, it is included one motor with servo, one without, one ϕ servo and one ϑ servo, presenting no loss of generalization because the other actuators of the same kind have the same response. Changes in roll angle, φ, directs the force in the y direction, having almost no effect on the other two directions, hence the plotting of the y, φ in the same figure 1. Analogously for the variables θ and x in figure 2.

Figures 1 and 2 show that BKS and IBKS have different control allocation solutions. The first one

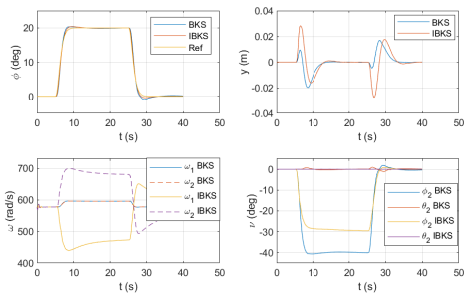

Figure 1: Roll Angle Reference Tracking System Responses

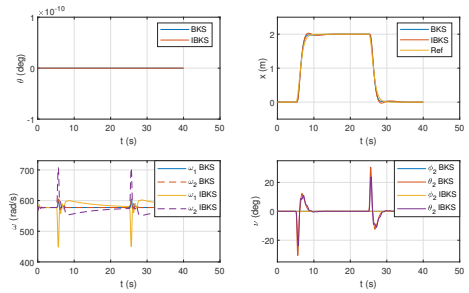

Figure 2: x Coordinate Reference Tracking System Responses

tends to have the same speed on all motors, while its incremental counterpart demands a higher effort of the motors with servos, decreasing the necessary servo angle to maintain the tilted attitude. In figure 1, it can be seen that a change in attitude of 20 degrees, has a rather small impact of 0.02 meters on the position, with the same effect on x for changes in θ. However, changes in the x, y and z coordinates have little to no effect on the Tilt-Quadrotors attitude, see figure 2.

For coupled reference tracking, with φ and θ reference values of up to 40 degrees and ψ values of up to 80 degrees, both controllers have good tracking capabilities for position, as can be seen in figure 3. The BKS system showing an average distance from the reference of 0.0703 meters and the IBKS system of 0.1080 meters while both being able to track the attitude references closely.

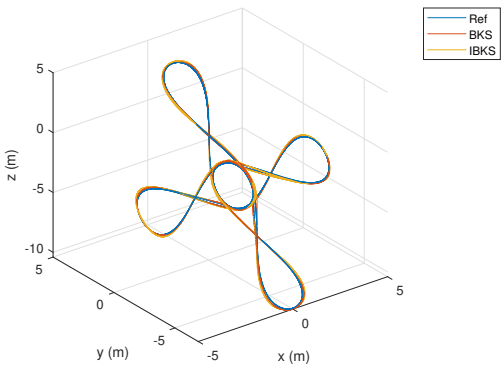

Figure 3: Coupled Reference Tracking System Responses

4.1. Robustness

When it comes to controller robustness to measurement noise, testing is made by simulation with added zero-mean white noise to the system's outputs $\boldsymbol{s}_1$, $\boldsymbol{s}_2$ and $\dot{\boldsymbol{s}}_2$ before being fed to the controller. The linear position, velocity and acceleration measurement noise variance is $4 \cdot 10^{-3}$. In the case of the attitude and angular velocity, the variance is of 10^{-4}. Lastly, the variance of the noise added to the angular acceleration measurement is $2 \cdot 10^{-3}$.

In order to evaluate performance, the root mean square residue between the responses with measurement noise and the noise-free responses is calculated.

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

where s is the state response being studied, which can stand for φ, θ, ψ, x, y or z, s_b is the baseline, noise-free, response for the same state, $0 \leq k \leq N$ 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.

For attitude tracking, the BKS controlled system response RMSR values are of the order of magnitude of 10^{-1} degrees and of 10^{-2} degrees for the IBKS controlled system. For position tracking, the BKS and IBKS controlled systems responses RMSR have the same order of magnitude of 10^{0} millimeters, showing that the IBKS controller is more robust to measurement noise in general.

Testing shows that the IBKS controller is robust, i.e. can stabilize and track references, to large parameter changes in center of gravity location, arm length, mass, inertia, propeller constants, motor and servo constants and single motor or servo constants with respect to the other motors or servos. The BKS controller can not stabilize and track references to as large parameter changes, especially for the location of the center of gravity or one of the motors or servos to be different from the others.

5. Conclusions

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.

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 stably 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.

Robustness analysis shows that the Incremental Backstepping controller is extremely robust to model errors, being able to stabilize and track references with severe model parameter changes. 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.

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.

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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