Article

Experimental Nonlinear and Incremental Control Stabilization of a Tail-Sitter UAV with Hardware-in-the-Loop Validation

Alexandre Athayde , Alexandra Moutinho , José Raul Azinheira *

Posted Date: 8 February 2024

doi: 10.20944/preprints202402.0467.v1

Keywords: tail-sitter; unmanned aerial vehicles (UAV); vertical take-off and landing (VTOL); nonlinear control; incremental control; attitude control

doi:10.20944/preprints202402.0467.v1

Article

Experimental Nonlinear and Incremental Control Stabilization of a Tail-Sitter UAV with Hardware-in-the-Loop Validation

Alexandre Athayde , Alexandra Moutinho and José Raúl Azinheira *

IDMEC, Instituto Superior Técnico, Universidade de Lisboa, 1049-001 Lisboa, Portugal

* Correspondence: jose.raul.azinheira@tecnico.ulisboa.pt

Abstract: Tail-sitters aim to combine the advantages of fixed-wing and rotor-craft, but demand a robust and fast stabilization strategy to perform vertical maneuvers and transitions to and from aerodynamic flight. The research conducted in this work intends to assess the performance of nonlinear control strategies to stabilize the attitude of a X-Vert VTOL aircraft when hovering, comparing existing solutions to applications of Nonlinear Dynamics Inversion (NDI) and its incremental version, INDI. Such controllers are implemented and tuned in simulation in order to stabilize a model of the tail-sitter , complemented by estimation methods that allow to feed back the necessary variables. These estimators and controllers are then implemented in a microcontroller, validating them in a Hardware-in-the-Loop (HITL) scenario, with simple maneuvers in vertical flight. Lastly, the developed control solutions are used to stabilize the aircraft in experimental flight, being monitored by a motion capture system. The experimental results allow to validate the model of the X-Vert and compare the effectiveness of the different control solutions in stabilizing it, with the INDI presenting itself as a more robust control strategy, with better tracking capabilities and less actuator demands.

Keywords: tail-sitter; unmanned aerial vehicles (UAV); vertical take-off and landing (VTOL); nonlinear control; incremental control; attitude control

1. Introduction

Unmanned Aerial Vehicles (UAV) have been increasing in popularity in the last decades, with broad scientific, industrial and military usages [1]. From a wide-range of urban applications [2], to topographic mapping and surveillance operations [3], useful for agricultural purposes [4] for example, these aircraft are undeniable aids, and are becoming virtually indispensable in many fields. It is natural that a given application may influence the requirements for the UAV to be used. Rotor-craft are usually chosen when high manoeuvrability is intended, whilst fixed-wings are more common when needing to cover large distances due to their higher endurance, with each having their own drawbacks [5]. As a middle-ground between these two types of aircraft, hybrid UAVs have also been a subject of intense research [6], aiming to combine the advantages of fixed-wings and rotor-craft, simultaneously avoiding or diminishing their shortcomings. These sort of vehicles are also commonly designated as Vertical Take-Off and Landing (VTOL) aircraft, as they have the capability of taking off vertically and then perform a transition to cruise flight, and then transition back to vertical flight for landing operations. These UAVs can be classified according to a number of aspects, namely how they perform their transition maneuvers, for example, rotating their rotors or wings, respectively for tilt-rotors or tilt-wing aircraft, or tilting themselves in landing maneuvers, such as tail-sitters, that owe their denomination to the fact that they land on their tails. The reviews in [5–7] provide additional insights over VTOL aircraft. From the different hybrid and convertible UAVs mentioned in these works, tail-sitters have a simplified mechanical design, in general requiring less actuators and moving parts, although at the cost of being susceptible to crosswinds when performing vertical flight as the wing is perpendicular to the ground, and thus requiring complex transition maneuvers [6].

The state-of-the-art on aircraft control starts on model-based linear feedback controllers: a nonlinear model of the system is designed, comprising its different subsystems with a varying degree of complexity, which is then linearized at a certain operating point, usually hovering flight for rotorcraft, or constant airspeed leveled flight for fixed-wings. Afterwards, linear controllers are designed using adequate methods, of which Proportional-Integral-Derivative (PID) and Linear Quadratic Regulators (LQR) are examples [8,9]. However, the performance of these controllers is highly influenced by nonlinearities and model mismatches, both of which not infrequent in UAVs that face large angles-of-attack. Furthermore, as the models used to compute such controllers depend heavily on the airspeed and air density, these linear control strategies are commonly paired with a scheduling mechanism [10,11], which may become computationally expensive for the flight controller. As a way of circumventing many of these issues, nonlinear strategies are the object of development and discussion in flight control, as they allow to incorporate the nonlinearities of the model in the control design phase, thus being less-susceptible to the aforementioned loss of performance [12,13]. The Nonlinear Dynamics Inversion (NDI) is a well-known example used in aircraft stabilization, which works by inverting the model - along with many of its nonlinearities - in order to determine the control action to take, leading to more robust controllers [14,15]. Notwithstanding, NDI, along with similar nonlinear control strategies like Backstepping [16], still require an accurate model of the UAV to be controlled, thus requiring extensive parameters identification either in wind tunnel or flight testing, and being susceptible to model inaccuracies [17]. As a way to counteract such limitations, a reformulation of these control strategies can be performed so that they rely more on sensor data instead of the information provided by the model, leading to incremental versions of these controllers - INDI [18,19] and IBKS [20,21]. These control strategies have the advantage of only requiring the modeling of the actuation components - like propellers, rotors and control surfaces - to compute the control action, and have demonstrated to be more robust to model mismatches and parameter uncertainty [22–24]. In particular, tail-sitters benefit from control strategies that are less model dependant, as they face high angles-of-attack (AOA) and complex propeller-fuselage interactions, both of which are difficult aspects to portray. Some previous research works address the application of incremental control laws to tail-sitters [25–28], but these do not draw a comparison with conventional nonlinear controllers in order to highlight the advantages of their incremental versions.

This work provides a thorough model of the E-Flite/Horizon Hobby X-Vert VTOL, a small and highly-manoeuvrable bi-rotor tail-sitter UAV, which has been previously modeled in [29], together with a systematic comparison of the performance of different controllers in the presence of sensor noise and possible model mismatches.Three control strategies were considered: i) a simplified version of the nonlinear control strategy proposed in [29], consisting on an adaptation to specific conditions of vertical flight, and designated as Benchmark Nonlinear Controller (BNC), ii) the conventional Nonlinear Dynamics Inversion, and iii) its incremental counterpart, INDI. These controllers are implemented to control the X-Vert when performing vertical flight, and compared in three distinct environments: i) in simulation, considering the comprehensive model of the tail-sitter, in order to provide for a baseline comparison of the controllers performance, ii) in Hardware-In-The-Loop (HITL), where these control solutions are implemented in a micro-controller unit (MCU) enabling to control the computer-simulated aircraft by an external flight controller using Ethernet communications which was specifically designed and assembled for the control of the X-Vert VTOL, and iii) in a controlled indoor environment, in order to assess the controllers performance and robustness when controlling the real UAV to perform simple maneuvers in vertical flight.

2. Aircraft Simulator

The simulator developed in this work encompasses a model of the X-Vert tail-sitter UAV as well as the necessary elements to replicate an experimental scenario. A generic overview of the simulation environment is provided in Figure 1, accounting for these diverse components, where $\mathbf{x}$, $\mathbf{u}$ and $\mathbf{z}$ respectively stand for the state, input and output vectors. The subset of $\mathbf{x}$ that is used for vertical flight control is represented by $\mathbf{y}$, and thus $\mathbf{y}_{ref}$ and $\mathbf{y}_{est}$ are used to respectively denote its reference and estimation.

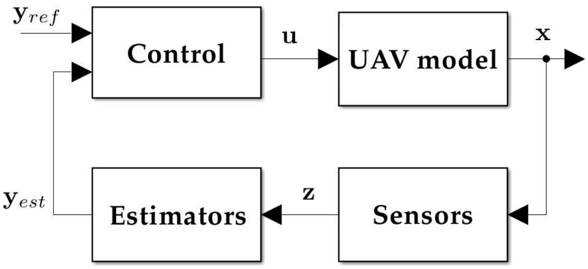


Figure 1. Simulation environment.

The current section addresses the X-Vert model, starting with a short description of this UAV, then describing the equations of motion of the model in detail, accounting for its aerodynamics, propulsion system, ground contact forces and gravity. Since the sensors do not yield direct measurements for all the components of $\mathbf{x}$, an estimation step, described in the end of this section, is necessary, allowing to reconstruct the remaining necessary states from the available sensor data. The control block, representing the centerpiece of this research work, is addressed separately in section 3. $\mathbf{y_{ref}}$ consists on a set of simple maneuvers in vertical flight, which are described in detail in section 4.

Throughout this section, multiple constants and coefficients will be defined and used to describe the simulated aircraft, but their values are omitted for presentation purposes, being provided in Appendix A with their respective sources.

2.1. *Tail-Sitter Prototype and Nonlinear Model*

The X-Vert VTOL is a radio-controlled aircraft manufactured by E-Flite/Horizon Hobby [30]. It is a half-meter wingspan tail-sitter with two elevons - control surfaces that combine the traditional functions of aileron and elevator - and two proprotors - which also have their denomination as they work simultaneously as rotors and propellers. This aircraft was chosen for this research work due to its VTOL capabilities and small dimensions, allowing for indoor flight. The prototype used for this work was retrofitted with additional sensors and electronics that are described in section 5, and is shown in Figure 2, where the elevons and proprotors can be seen.

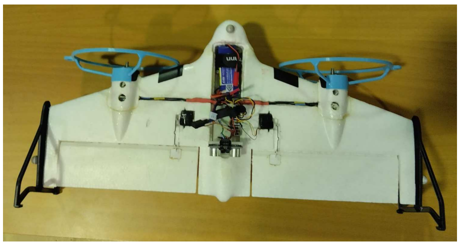


Figure 2. Experimental prototype based on the frame of the X-Vert VTOL.

2.1.1. Equations of Motion

The aircraft model used in the developed simulator is largely based in [29] as it already provides a comprehensive model for the specific tail-sitter UAV used in this work, the X-Vert VTOL. This model is described briefly, with additional emphasis on the aspects that differ from the aforementioned model.

This UAV requires four inputs, these being the deflections of the two elevons, δ_R and δ_L, and the throttle signals for the two proprotors, τ_R and τ_L, with the subscripts "R" and "L", denoting the *right* and *left* side of the wing, respectively. Notwithstanding, it is more convenient to represent these inputs as the more traditional ones in flight control: δ_a and δ_e for respectively standing for the differential and simultaneous elevon deflection, representing the functions of *ailerons* and *elevator*;

doi:10.20944/preprints202402.0467.v1

and τ_r and τ_t being the analogous application to the proprotors, taking the role of *rudder* and *throttle*. Therefore, the input vector $\mathbf{u}$ can be obtained from the original four inputs $[\delta_R, \delta_L, \tau_R, \tau_L]^T$ by an adequate transformation [31], and is represented by:

$$\mathbf{u} = [\delta_a, \delta_e, \tau_r, \tau_t]^T \tag{1}$$

The state vector $\mathbf{x} = [\mathbf{v}_g^B, \mathbf{w}_g^B, \mathbf{p}^{NED}, \mathbf{q}^{NED}, \mathbf{\Omega}]^T$ is represented by the linear and angular velocity vectors in relation to the ground and expressed in the body-frame, $\mathbf{v}_g^B$ and $\mathbf{w}_g^B$, respectively, the position of the aircraft in the North-East-Down (fixed) frame, $\mathbf{p}^{NED}$, and its orientation in relation to this same frame, expressed in quaternion form, $\mathbf{q}^{NED}$. Naturally, the usage of quaternions for attitude representation intends to avoid the well-known singularity issues related to more familiar representations, like the Euler angles [31]. Two additional states are included in $\mathbf{x}$ to account for the angular velocities of the left and right motors that rotate the proprotors, represented briefly by $\mathbf{\Omega} = [\Omega_R, \Omega_L]^T$.

It should be noted that it is common to represent aforementioned velocity vectors with respect to the air - $\mathbf{v}_a^B = \mathbf{v}_g^B - \mathbf{v}_w^B$ and $\mathbf{w}_a^B = \mathbf{w}_g^B - \mathbf{w}_w^B$ - especially when accounting for the effects of aerodynamics, as these are impacted by the wind by providing additional components for these velocities, $\mathbf{v}_w^B$ and $\mathbf{w}_w^B$. Although this work assumes negligible wind and therefore resulting in $\mathbf{v}_a^B = \mathbf{v}_g^B$ and $\mathbf{w}_a^B = \mathbf{w}_g^B$, the subscript a will still be kept when air velocities should be used for generalisation purposes. Examples of this are the angle-of-attack (AOA) α, the sideslip angle β, and airspeed V_t, which are computed using the air velocity vector $\mathbf{v}_a^B$ [31–33].

The dynamics and kinematics of the UAV are expressed by

$$\dot{\mathbf{v}}_g^B = \frac{1}{m_{uav}}\mathbf{f}^B - (\mathbf{w}_g^B \times \mathbf{v}_g^B) \tag{2a}$$

$$\dot{\mathbf{w}}_g^B = \mathbf{J}_{uav}^{-1}(\mathbf{m}^B - \mathbf{w}_g^B \times \mathbf{J}_{uav}\mathbf{w}_g^B) \tag{2b}$$

$$\dot{\mathbf{p}}^{NED} = \mathbf{R}_B^{NED}\, \mathbf{v}_g^B \tag{3a}$$

$$\dot{\mathbf{q}}^{NED} = \frac{1}{2}\left(\mathbf{\Omega}_{\mathbf{q}}^{\mathbf{w}}\, \mathbf{q}^{NED}\right) \tag{3b}$$

where m_{uav} represents the mass of the X-Vert and $\mathbf{J}_{uav}$ its inertia matriX, whilst $\mathbf{f}^B$ and $\mathbf{m}^B$ stand for the resulting force and moment vectors acting on it. Additionally, $\mathbf{R}_B^{NED}$ and $\Omega_{\mathbf{q}}$ denote auxiliary matrices that depict the influence of $\mathbf{v}_g^B$ and $\mathbf{w}_g^B$ respectively in $\mathbf{p}^{NED}$ and $\mathbf{q}^{NED}$, having been gathered from [31].

Four different aspects contribute to the forces and moments acting on the UAV, these being the propulsion system, the aerodynamics, the ground contact effects and the gravity. Since it is assumed that the force and moment balances are expressed in the Center-of-Gravity (CG) of the aircraft, the moment that results from the gravity effects is neglected, and therefore $\mathbf{f}^B$ and $\mathbf{m}^B$ are expressed by

$$\mathbf{f}^B = \mathbf{f}_p^B + \mathbf{f}_a^B + \mathbf{f}_c^B + \mathbf{f}_g^B \tag{4a}$$

$$\mathbf{m}^B = \mathbf{m}_p^B + \mathbf{m}_a^B + \mathbf{m}_c^B \tag{4b}$$

with each of the components being described in the following sections.

2.1.2. Propulsion Forces and Moments

The rotation of each proprotor of the X-Vert generates a thrust T and a torque Q, which influence the propulsion force $\mathbf{f}_p^B$ and moment $\mathbf{m}_p^B$ accounting for the position of the right and left proprotors [29]:

$$\mathbf{f}_p^B = \begin{bmatrix} T_R + T_L \\ 0 \\ 0 \end{bmatrix} \tag{5a}$$

$$\mathbf{m}_p^B = \begin{bmatrix} Q_R - Q_L \\ 0 \\ 0 \end{bmatrix} + \left(\mathbf{d}_{pr,R} \times \mathbf{f}_{p,R}^B\right) + \left(\mathbf{d}_{pr,L} \times \mathbf{f}_{p,L}^B\right) \tag{5b}$$

In turn, the thrust and torque of each proprotor depend on its angular velocity and incoming airspeed. Taking V_t as the vectorial norm of $\mathbf{v}_a^B$ and defining φ as the angle this vector makes with the rotational axis of each proprotor, the advance ratio J is obtainable, which can then be used to compute the thrust and power coefficients required for T and Q [29]:

$$J = \frac{\pi \cos(\varphi) \cdot V_t}{\Omega R_{prop}} \tag{6a}$$

$$T = \frac{4}{\pi^2} \rho_a \, \Omega^2 \, R_{prop}^4 C_T(J) \tag{6b}$$

$$Q = \frac{4}{\pi^3} \rho_a \, \Omega^2 \, R_{prop}^5 C_P(J) \tag{6c}$$

It should be noted that C_T and C_P depend on the advance ratio by a second-order equation following the form $C_{T/P} = c_{T/P,2} J^2 + c_{T/P,1} J + c_{T/P,0}$, and these are the parameters provided for the aircraft in Appendix A.

The relationship between the angular velocity of the motors Ω and the throttle input τ is modelled as the dynamics of a Brushless Direct Current (BLDC) electric motor regulated by an Electronic Speed Controller (ESC) [34], but a simplification is made by assuming the steady-state solution for the dynamics of the electric current, resulting in the first-order model

$$\dot{\Omega} = \frac{1}{J_{pr}} \left(K_t I - Q - B_m \Omega\right) \tag{7a}$$

$$I = \frac{V_{bat} \tau - K_e \Omega}{R_m} \tag{7b}$$

which assumes a constant battery voltage V_{bat} of 7.4 V, and with the torque constant, K_t; the back-electromotive force, K_e; the motor resistance, R_m; the rotational inertia of the proprotor, J_{pr}; and the damping constant B_m. The values for these constants were gathered from a similar motor to the one that is used in the X-Vert, and it can be verified that the steady-state solution of (7) agrees with the motor model in [29] for the aforementioned value of V_{bat}.

Lastly, the rotation of the proprotors originates an induced airspeed V_{ind} dependant on the generated thrust and inflow airspeed, which can be found solving [35]:

$$V_{ind}^4 + 2\cos(\varphi) V_t V_{ind}^3 + V_t^2 V_{ind}^2 = \left(\frac{T}{2\rho\pi R_{prop}^2}\right)^2 \tag{8}$$

Once V_{ind} is computed, the slipstream velocity $\mathbf{v}_{slip}$ and radius $\mathbf{r}_{slip}$ are determined by:

$$\mathbf{v}_{slip} = \mathbf{v}_a + \begin{bmatrix} -2V_{ind} \\ 0 \\ 0 \end{bmatrix} \tag{9a}$$

$$r_{slip} = \sqrt{R_{prop}^2 \left(\frac{V_t + V_{ind}}{V_t + 2\,V_{ind}} \right)} \tag{9b}$$

assuming fully-developed flow. The aerodynamic angles for the slipstream, α_{slip} and β_{slip}, and the absolute slipstream airspeed, V_{slip} are obtainable from $\mathbf{v}_{slip}$ [31,32].

2.1.3. Aerodynamic Forces and Moments

The wing of the X-Vert is modeled as a set of flat-plate segments for the left and right sides, following a similar approach to the ones taken in [29,36,37], but including additional aerodynamic derivatives to account for lateral aerodynamics and effects of the angular rates [31]:

$$\mathbf{f}_a^B = \mathbf{f}_{a,R}^B + \mathbf{f}_{a,L}^B + \mathbf{f}_{a,lat+derivs}^B \tag{10a}$$

$$\mathbf{m}_a^B = \mathbf{m}_{a,R}^B + \mathbf{m}_{a,L}^B + \mathbf{m}_{a,lat+derivs}^B \tag{10b}$$

Each side of the wing is divided into three zones for the computation of aerodynamic forces and moments, as illustrated in Figure 3: zone 1 (z_1) - the elevon section which is in the slipstream of the proprotors; zone 2 (z_2) - the elevon section outside said slipstream; and zone 3 (z_3) - the remainder of the side of the wing.

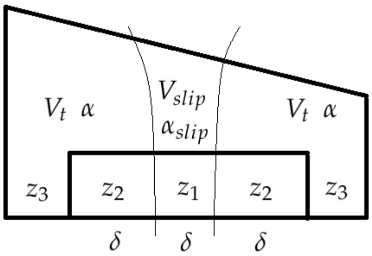


Figure 3. Division of wing into three zones, for the computation of the aerodynamic forces and moments according to different angles-of-attack and airspeed values, as well as elevon deflection.

Accounting for such division, the drag D, lift L and pitching moment m for each zone can be expressed in the body frame as

$$\mathbf{f}_{a,R/L}^B = \mathbf{R}_{W,slip}^B \begin{bmatrix} -D_{z_1} \\ 0 \\ -L_{z_1} \end{bmatrix}_{R/L} + \mathbf{R}_W^B \begin{bmatrix} -D_{z_2} - D_{z_3} \\ 0 \\ -L_{z_2} - L_{z_3} \end{bmatrix}_{R/L} \tag{11a}$$

$$\mathbf{m}_{a,R/L}^B = \begin{bmatrix} 0 \\ m_{z_1} + m_{z_2} + m_{z_3} \\ 0 \end{bmatrix}_{R/L} + \left(\mathbf{d}_{AC,R/L} \times \mathbf{f}_{a,R/L}^B \right) \tag{11b}$$

where $\mathbf{R}_W^B$ introduces the influence of the angle-of-attack and sideslip angles, α and β, and $\mathbf{R}_{W,slip}^B$ takes on an analogous role but for the angles for zone 1, α_{slip} and β_{slip}, which account for the slipstream effects. The distance of the aerodynamic centre (AC) of the right and left sides of the wing are denoted respectively by $\mathbf{d}_{AC,R}$ and $\mathbf{d}_{AC,R/L}$, which are assumed to be the same for each side of the wing, regardless of the zone.

The drag, lift and pitching moment for each zone are obtained using the data for the aerodynamic coefficients - C_D, C_L and C_m - from [29] for the angle-of-attack and elevon deflection, and combining it with the airspeed at each zone and its span. For example, zone 1 has a span of $b_{z_1} = 2\,r_{slip}$, it is influenced by the slipstream air velocity V_{slip}, and the aerodynamic coefficients account for the slipstream AOA α_{slip} and elevon deflection δ, resulting in:

$$D_{z_1} = \frac{1}{2}\rho_a V_{slip}^2 c_w b_{z_1} \left(C_D\right)_{\alpha_{slip},\delta} \tag{12a}$$

$$L_{z_1} = \frac{1}{2}\rho_a V_{slip}^2 c_w b_{z_1} \left(C_L\right)_{\alpha_{slip},\delta} \tag{12b}$$

$$m_{z_1} = \frac{1}{2}\rho_a V_{slip}^2 c_w^2 b_{z_1} \left(C_m\right)_{\alpha_{slip},\delta} \tag{12c}$$

where c_w denotes the mean aerodynamic chord (MAC) of the wing and ρ_a denotes the air density, both assumed to be constant. Zones 2 and 3 use V_t for the airspeed value and α as the angle-of-attack to obtain the aerodynamic coefficients, but zone 2 accounts for the elevon deflection whilst zone 3 does not. The spans for each zone to be used in analogous computations to (12) are defined directly from the wingspan of the X-Vert, b_w, and elevon span, b_e: $b_{z_2} = b_e - 2\,r_{slip}$, $b_{z_2} = \frac{b_w}{2} - b_e$.

The last aspect of (10) to describe is the influence of lateral aerodynamics and aerodynamic derivatives by the means of $\mathbf{f}^B_{a,lat+derivs}$ and $\mathbf{m}^B_{a,lat+derivs}$ given by:

$$\mathbf{f}^B_{a,lat+derivs} = \frac{1}{2}\rho_a V_t^2 b_w c_w \mathbf{R}^B_W \begin{bmatrix} 0 \\ C_{Y_\beta}\sin(\beta) + \frac{b_w}{2V_t}\left(C_{Y_p} p_a + C_{Y_r} r_a\right) \\ \frac{c_w}{2V_t} C_{L_q} q_a \end{bmatrix} \tag{13a}$$

$$\mathbf{m}^B_{a,lat+derivs} = \frac{1}{2}\rho_a V_t^2 b_w c_w \begin{bmatrix} b_w C_{l_\beta}\sin(\beta) + \frac{b_w^2}{2V_t}\left(C_{l_p} p_a + C_{l_r} r_a\right) \\ \frac{c_w^2}{2V_t}\left(C_{m_q} q_a\right) \\ b_w C_{n_\beta}(2\beta) + \frac{b_w^2}{2V_t}\left(C_{n_p} p_a + C_{n_r} r_a\right) \end{bmatrix} \tag{13b}$$

having been adapted from [31] but modified to allow for larger values of β.

Since no specific data for the X-Vert was available for these coefficients, the values were gathered from a flying-wing UAV [38] and used according to (13). Despite the fact that the aircraft in the aforementioned research work is not the X-Vert, it is still a flying-wing UAV which the X-Vert resembles, and thus the resulting values for the coefficients should be satisfactory since adequate determination of these often requires experimental identification and/or computational methods [31], which undoubtedly fall out of the scope of this work. This may lead to the introduction of eventual modeling errors, but it is acknowledged for the purpose of designing control strategies for the X-Vert, expecting these should be robust to eventual model mismatches.

2.1.4. Ground Contact Forces and Moments

The interaction of the ground is gathered from [29], which employs a spring-damper analogy for representing the ground contact force at each contact point k, expressed in the inertial frame:

$$\mathbf{f}^{NED}_{c,k} = \begin{bmatrix} 0 \\ 0 \\ -m_{uav} k_{c,p} d_k \end{bmatrix} - m_{uav} k_{c,v} \mathbf{v}^{NED}_k \tag{14}$$

where d_k stands for the depth of point k, and $\mathbf{v}^{NED}_k = \mathbf{R}^B_{NED}(\mathbf{v}^{NED}_{g,k} + \mathbf{w}^{NED}_{g,k} \times \mathbf{r}_{c,k})$ its velocity expressed in the inertial frame, with $\mathbf{r}_{c,k}$ being the position of the point in relation to the CG, where "c" denotes "*contact*". $k_{c,p}$ and $k_{c,v}$ are gains for the spring-damper system, and their values were kept from [29].

$\mathbf{f}_{c,k}^{NED}$ also accounts for an upped limit of zero, representing the loss of contact with the ground for each point.

Five contact points were used, corresponding the four corners of the main wing and the noise of the UAV as shown in Appendix A, and thus the ground force and moment vectors can be computed by expressing $\mathbf{f}_{c,k}^{NED}$ for each in the body frame:

$$\mathbf{f}_c^B = \sum_{k=1}^{5} \mathbf{R}_B^{NED} \mathbf{f}_{c,k}^{NED} \tag{15a}$$

$$\mathbf{m}_c^B = \sum_{k=1}^{5} \mathbf{r}_{c,k} \times \mathbf{R}_B^{NED} \mathbf{f}_{c,k}^{NED} \tag{15b}$$

2.1.5. Gravity Force

The last force acting on the simulated model of the X-Vert is the gravity. As explained before, it is assumed that this force acts on the CG of the UAV, thus producing no moment, and therefore its effects can be modeled by:

$$\mathbf{f}_g^B = m_{uav} \mathbf{R}_{NED}^B \mathbf{g}^{NED} \tag{16}$$

where $\mathbf{g}^{NED} = [0, 0, 9.8065]^T$ m/s^2 denotes the gravity acceleration vector expressed in the fixed frame.

2.2. Sensors

The simulator used in this work also accounts for the sensors onboard the UAV (see Figure 2, namely an accelerometer and gyroscope - a common combination for estimating the attitude of aircraft - and a sonar used to assist in the vertical takeoff and landing maneuvers. The output vector $\mathbf{z} = [\mathbf{a}_{g,acc}^B, \mathbf{w}_{g,gyr}^B, d_{son}^B]^T$ comprises the acceleration vector provided by the accelerometer, $\mathbf{a}_{g,acc}^B$, the angular velocity from gyroscope readings, $\mathbf{w}_{g,gyr}^B$, and distance measured by the sonar, d_{son}^B, all expressed in the body frame. It is noted that the sensors are assumed to be coincident with the centre of gravity (CG) of the UAV.

The models for the accelerometer and gyroscope [39] are provided respectively by:

$$\mathbf{a}_{g,acc}^B = \mathbf{a}^B + \mathbf{w}_g^B \times \mathbf{v}_g^B + \mathbf{R}_{NED}^B \mathbf{g}^{NED} + \mathbf{b}_{acc} + \eta_{acc} \tag{17}$$

$$\mathbf{w}_{g,gyr}^B = \mathbf{w}_g^B + \mathbf{b}_{gyr} + \eta_{gyr} \tag{18}$$

where $\mathbf{b}$ denotes the bias of the respective sensor and η its zero-mean Gaussian noise vector.

The sonar is mounted in the underside of the X-Vert and pointing to its tail, parallel to the x axis of the body frame, and its model is given by

$$d_{son}^B = \frac{P_D || \mathbf{u}_{son}^{NED} ||}{\mathbf{u}_{son}^{NED} \cdot \mathbf{e}_z} + b_{son} + \eta_{son} \tag{19}$$

where d_{son}^B represents the measurements of the sensor, P_D is the Down coordinate expressed in the NED frame, $\mathbf{u}_{son}^{NED} = \mathbf{R}_B^{NED}(-\mathbf{e}_x)$ depicts the orientation vector of the sonar expressed in this same frame, $\mathbf{e}_x$ and $\mathbf{e}_z$ represent the respective unit vectors and η_{son} denotes the noise of the sensor.

When it comes to implementing the models of these sensors, some considerations should be provided regarding their respective biases and noise vectors. Firstly, the biases are assumed to be constant and capable of being removed by means of a suitable calibration process in an experimental scenario. Consequently, they take null values for simulation purposes, as shown in Appendix A. Secondly, regarding the specific noise characteristics of each sensor, these are usually characterized by their respective variance σ^2, where a combination of theoretical values from [31] and measurements from the real sensors were used (see section 5). Since the goal is not extensive sensor identification but only to portray the effects of their characteristic noise for simulation purposes, their specific values are illustrative but kept in their respective order of magnitude.

2.3. Attitude and Vertical Velocity Estimators

For the purpose of stabilization of the X-Vert in vertical flight, the variables related with the attitude - $\mathbf{w}_g^B$ and $\mathbf{q}^{NED}$ - and altitude - u_g^B and P_D - require observation, and therefore the estimated output vector is defined as

$$\mathbf{y}_{est} = [\hat{u}_g^B, \mathbf{w}_{g,gyr}^B, \hat{P}_D, \hat{\mathbf{q}}^{NED}]^T \tag{20}$$

In order to save computational resources, a choice was made to use simple and fast methods to estimate these states of complementary nature. Starting with the attitude, an estimate of $\mathbf{q}^{NED}$ is obtainable through the combination of accelerometer and gyroscope data using the Madgwick algorithm [40]:

$$\hat{\mathbf{q}}_k^{NED} = \hat{\mathbf{q}}_{k-1}^{NED} + T_s\left(\frac{1}{2}\hat{\mathbf{q}}_{k-1}^{NED} \otimes [0\ ,\ \mathbf{w}_{g,gyr}^B]^T - \beta_{CF}\frac{\mathbf{J}_g\mathbf{f}_g}{||\mathbf{J}_g\mathbf{f}_g||}\right) \tag{21}$$

where $\mathbf{f}_g$ is an objective function to be minimised and $\mathbf{J}_g$ its Jacobian matrix, the expressions of both having been omitted in this work but readily available in the original research [40]. This estimator includes the accelerometer readings $\mathbf{a}_{g,acc}^B$ in $\mathbf{f}_g$ and combines it with the integration of the gyroscope measurements $\mathbf{w}_{g,gyr}$, balancing the relative weight between both with the scalar β_{CF}, which takes the role of the single design variable for adjusting this estimator. Since the accelerometer does not perceive any change in a rotation over an axis aligned with the gravity vector, the estimator is subject to some drift as it relies only on the gyroscope integration for these cases. Nonetheless, for small flight times it provided satisfactory results, presenting a fast and simple estimation strategy for the attitude in vertical flight, whilst acting as a filter for the noise present in the sensors.

Following an analogous reasoning, the readings from the accelerometer, discarding the gravity contribution, can be integrated to estimate the velocity over the x-axis of the UAV, whilst another estimate of it is obtainable by deriving sonar readings. Pairing both of these in the form of a more conventional complementary filter [31], the longitudinal velocity u_g can be computed by filtering and combining both of these estimates according to

$$\hat{u}_{g,k}^B = \alpha_{CF} \cdot LPF(\hat{u}_{son,k}^B) + (1 - \alpha_{CF}) \cdot HPF(\hat{u}_{acc,k}^B) \tag{22}$$

in which α_{CF} is a design variable, and where $\hat{u}_{son,k}$ and $\hat{u}_{acc,k}$ are the previously described estimates of u_g from the sonar and accelerometer, respectively, defined by:

$$\hat{u}_{son,k}^B = \frac{d_{son,k}^B - d_{son,k-1}^B}{T_s} \tag{23a}$$

$$\hat{u}_{acc,k}^B = \hat{u}_{acc,k-1}^B + T_s(a_{acc,X}^B + g_0 \cdot 2(q_0q_2 - q_1q_3)) \tag{23b}$$

and LPF and HPF denote respectively *Low-Pass Filter* and *High-Pass Filter*, both first-order as defined in [31]. Despite being a somewhat rudimentary estimation method when compared to more complex sensor-fusion algorithms like the Kalman filter [31], the combination of these two estimators provide observations of all the necessary variables, at a relatively low computational cost, with the added benefit of requiring only two design variables - α_{CF} and β_{CF}.

3. Nonlinear Control Strategies for Tail-Sitter UAV Vertical Flight

This section describes the design of different nonlinear control strategies based on the model of the X-Vert and applied to its vertical flight. As the focus of this work is to compare the different control methods stabilizing the UAV, a choice was made to test these with the same velocity control strategy defined in [29]. With this in mind, it is useful to reorganize the state and input vectors to account for a decoupled design of attitude and velocity controllers. The velocity controller focuses on keeping a desired forward velocity and altitude $\mathbf{x}_{vel} = [u_g^B, P_D]^T$ using the throttle input $\mathbf{u}_{vel} = \tau_t$. Similarly, the attitude stabilization corresponds to the states $\mathbf{x}_{att} = [\mathbf{w}_g^B, \mathbf{q}^{NED}]^T$ and inputs $\mathbf{u}_{att} = [\delta_a, \delta_e, \tau_r]^T$.

3.1. Equilibrium at Hover

Under the assumption that no wind is present, the equilibrium conditions for the X-Vert in hovering flight can be evaluated computing the steady-state solutions of (2) at a given nonzero altitude and knowing that $\mathbf{q}^{NED} = [\sqrt{2}/2, 0, \sqrt{2}/2, 0]^T$, $u_g^B = 0$ m/s. Under such conditions, the thrust generated by both motors must be equal to the opposing forces, namely the gravity and aerodynamic drag as a result of a nonzero slipstream velocity behind the propellers, and the ground contact force will be zero. Thus, the value for Ω_0 can be obtained by numerically solving the equation:

$$2T(\Omega) - 2D(\alpha_w, \delta_e = 0) - m_{uav} g_0 = 0 \tag{24}$$

Once Ω_0 has been determined, the necessary throttle to maintain a hovering condition, $tau_{t,0}$, can be found by computing the steady-state solution of (7). Performing these two steps, the values of $\Omega_0 = 1167.167$ rad/s and $\tau_{t,0} = 0.831$ were determined accounting for the parameters of the X-Vert in Appendix A. Therefore, the state and input vectors when hovering become:

$$\mathbf{x}_0 = [\mathbf{0}_{1\times 3}, \mathbf{0}_{1\times 3}, 0, 0, P_{D,0}, \sqrt{2}/2, 0, \sqrt{2}/2, 0, 1167.167, 1167.167]^T \tag{25a}$$

$$\mathbf{u}_0 = [0, 0, 0, 0.831]^T \tag{25b}$$

in the corresponding units.

The last aspects of relevance for the equilibrium at hover are the advance ratio of the propellers, the slipstream velocity and respective radius, as the first allows the computation of C_T and C_P, whilst the latter two directly influence the authority of the elevons. Under the assumption of $V_t = 0$ m/s, it follows that $J = 0$, and $\mathbf{v}_{slip,0}$ and $R_{slip,0}$ can be obtained directly from (8) and (9).

3.2. Rotational Dynamics in Affine Form

The NDI control strategy requires that the system to be controlled is expressed in affine form. Addressing the rotational subsystem of (2), it can be reorganized into:

$$\dot{\mathbf{w}}_g^B = \mathbf{F}(\mathbf{w}_g^B) + \mathbf{G}\begin{bmatrix}\delta_a \\ \delta_e \\ \tau_r\end{bmatrix} \tag{26}$$

which, in turn, demands that the resultant moment $\mathbf{m}^B$ be divided into its wing and actuator contributions, as the following equations suggest [41]:

$$\mathbf{F}(\mathbf{w}_g^B) = \mathbf{J}_{uav}^{-1}(\mathbf{m}_{wing}^B - \mathbf{w}_g^B \times \mathbf{J}_{uav}\mathbf{w}_g^B) \tag{27a}$$

$$\mathbf{G}\begin{bmatrix}\delta_a \\ \delta_e \\ \tau_r\end{bmatrix} = \mathbf{J}_{uav}^{-1}(\mathbf{m}_{cont}^B) \tag{27b}$$

Since $\mathbf{m}^B$ depends solely on the propulsion and aerodynamics of the X-Vert, each of these must be mathematically manipulated in order to allow for the implementation of (26). Starting with (5) and linearizing it for the operating condition defined by Ω_0, the moment contribution from the propulsion subsystem becomes:

$$\mathbf{m}_p^B \approx \begin{bmatrix} k_Q 2\Omega_0(\Omega_R - \Omega_L) \\ 0 \\ -d_{pr,y} k_T 2\Omega_0(\Omega_R - \Omega_L) \end{bmatrix} = \begin{bmatrix} 4k_Q \frac{\Omega_0^2}{\tau_{t,0}} \\ 0 \\ -4d_{pr,y} k_T \frac{\Omega_0^2}{\tau_{t,0}} \end{bmatrix} \tau_r \tag{28}$$

where $k_T = \frac{4}{\pi^2}\rho_a R_{prop}^4 C_T(0)$ and $k_Q = \frac{4}{\pi^3}\rho_a R_{prop}^5 C_P(0)$ were used to simplify the notation. The second equality of (28) is achieved by performing the approximation $\Omega_{R/L} \approx \frac{\Omega_0}{\tau_{t,0}}\tau_{R/L}$.

A similar process must be obtained for the aerodynamics contribution for $\mathbf{m}^B$ expressed by (10). Since $V_t = 0$, the lift, drag and pitching moment will only be influenced by the slipstream velocity on the control surfaces, and $\alpha_{slip} = 0$ and $\beta_{slip} = 0$ as $\mathbf{v}_{slip}$ will be aligned with the axes of the propellers. Using a linear approximation of the aerodynamic coefficients defined generically by $C_{D/L/m} \approx k_{D/L/m}\delta$ with $k_{D/L/m} = \left(C_{D/L/m}(\alpha = 0, \delta = \delta_{max}) - C_{D/L/m}(\alpha = 0, \delta = 0)\right)/\delta_{max}$, the aerodynamic contribution to $\mathbf{m}^B$ can be approximated as:

$$\mathbf{m}_a^B \approx \rho_a V_{slip}^2 R_{slip} \bar{c} \begin{bmatrix} -2d_{AC,y}k_L & 0 \\ 0 & 2\bar{c}k_m + 2d_{AC,x}k_L \\ 2d_{AC,y}k_D & 0 \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_e \end{bmatrix} \tag{29}$$

It is evident from (28) and (29) that $\mathbf{m}_{wing}^B = \mathbf{0}_{3\times 1}$, and therefore the $\mathbf{F}$ and $\mathbf{G}$ components of (26) can be determined adequately merging (28) and (29) into:

$$\mathbf{F}(\mathbf{w}_g^B) = \mathbf{J}_{uav}^{-1}(-\mathbf{w}_g^B \times \mathbf{J}_{uav}\mathbf{w}_g^B) \tag{30a}$$

$$\mathbf{G} = \mathbf{J}_{uav}^{-1} \begin{bmatrix} -2d_{AC,y}k_L & 0 & 4k_Q \frac{\Omega_0^2}{\tau_{t,0}} \\ 0 & 2\bar{c}k_m + 2d_{AC,x}k_L & 0 \\ 2d_{AC,y}k_D & 0 & -4d_{p,y}k_T \frac{\Omega_0^2}{\tau_{t,0}} \end{bmatrix} \tag{30b}$$

Accounting for the aforementioned assumptions, the $\mathbf{G}$ matrix in (30) is constant, and thus can be determined with the parameters and constants of the X-Vert model supplied in Appendix A. Furthermore, this matrix can be further simplified considering only its diagonal, assuming that each of the components of $\mathbf{u}_{att} = [\delta_a, \delta_e, \tau_r]^T$ predominantly affects p_g, q_g and r_g, respectively. Therefore, its numerical value is

$$\mathbf{G} \approx \begin{bmatrix} -25.492 & 0 & 0 \\ 0 & -95.726 & 0 \\ 0 & 0 & -274.151 \end{bmatrix} \tag{31}$$

in appropriate units, as it will be used in NDI and INDI controllers.

3.3. Velocity Control

In order to draw an objective comparison among different stabilization methods of the X-Vert, a suitable forward velocity controller must be chosen, and care must be taken that it ensures a minimal slipstream velocity to provide control authority to the elevons. As the design of such controller falls out of the scope of this work, and the control strategy of [29], taken as the benchmark nonlinear controller, already provides an adequate solution, this was adopted. It consists on determining a desired thrust that tracks references for P_D and u_g^B,

$$F_d = m_{uav}(2(q_0q_2 - q_1q_3))(g_0 - k_{P_D}P_{D,err}) + m_{uav}k_u u_{err} \tag{32}$$

in which $u_{err} = u_{g,ref}^B - u_g^B$ and $P_{D,err} = P_{D,ref} - P_D$.

Knowing the desired forward velocity, the propeller model (6) can be used to determine the angular velocity and motor torque, which in turn allow to compute the throttle input τ_t. Nonetheless, as explained in [29], it is useful to provide lower boundary on F_d, to ensure a minimum slipstream airspeed $V_{slip,min}$ of 7 m/s in the control surfaces, and an upper limit so it can have some yawing authority:

$$F_d \in [\rho_a \pi R_{prop}^2 V_{slip,min}^2 \quad , \quad 2 \cdot 0.95 \cdot \frac{4}{\pi^2} \rho_a \, R_{prop}^4 C_T(0) \Omega_{max}^2] \tag{33}$$

The maximum angular velocity of $\Omega_{max} = 1367.665$ rad/s was retrieved by numerically solving (7) for $\tau = 1$.

The expression in (32) was used to compute τ_t in order to complement the NDI and INDI attitude stabilization methods with a forward velocity control option. However, the Benchmark Nonlinear Controller computes separately the left and right throttle signals, τ_R and τ_L, together with the attitude stabilization, which was chosen to be maintained, as explained next.

3.4. Attitude Stabilization

3.4.1. Benchmark Nonlinear Controller (BNC)

The nonlinear controller used as benchmark for the attitude stabilization is an adaptation of the one in [29], made by applying the previously referred assumption of zero airspeed, $V_t = 0$. Similarly to the velocity controller, a desired set of moments can be defined by:

$$\mathbf{m}_d^B = \mathbf{J}_{uav}(\mathbf{K}_{ap}\mathbf{q}_{err,1:3} + \mathbf{K}_{ad}\mathbf{w}_g^B) \tag{34}$$

where $\mathbf{K}_{ap}$ and $\mathbf{K}_{ad}$ are 3-by-3 diagonal gain matrices, with $k_{i,i} > 0$ for $i = 1, 2, 3$, and $\mathbf{q}_{err,1:3}$ standing for the vectorial components of $\mathbf{q}_{err} = \mathbf{q}^{NED*} \otimes \mathbf{q}_{ref}^{NED}$ - the error quaternion - which results from the quaternion product of the conjugate of $\mathbf{q}^{NED}$ with $\mathbf{q}_{ref}^{NED}$ [29,32]. Knowing $\mathbf{m}_d$, and obtaining F_d from (32), the required thrust for each motor is computed from:

$$\begin{bmatrix} T_L \\ T_R \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & \frac{1}{d_{p,y}} \\ 1 & -\frac{1}{d_{p,y}} \end{bmatrix} \begin{bmatrix} F_d \\ m_{d,Z} \end{bmatrix} \tag{35}$$

From T_R and T_L, the angular velocity of each motor can be computed by inverting the propeller model in (6). In turn, knowing Q_R and Q_L enables the calculation of τ_R and τ_L solving (7) under the assumption of steady-state conditions. Lastly, the elevon deflections are obtained from

$$\begin{bmatrix} \delta_L \\ \delta_R \end{bmatrix} = \frac{R_{prop}^2 \pi}{2k_T} \begin{bmatrix} \frac{1}{c_x \Omega_L^2} & \frac{-1}{c_y \Omega_L^2} \\ \frac{-1}{c_x \Omega_R^2} & \frac{-1}{c_y \Omega_R^2} \end{bmatrix} \begin{bmatrix} m_{d,X} - Q_R + Q_L \\ m_{d,Y} \end{bmatrix} \tag{36}$$

where c_x and c_y represent respectively the rolling and pitching moment deflection coefficients in the slipstream, as defined in the original research work [29].

It is useful to reorganize the inputs that result from the BNC controller in an analogous expression to (1), allowing to compare these with their respective counterparts that result from the remaining controllers:

$$\begin{bmatrix} \delta_a \\ \delta_e \\ \tau_r \\ \tau_t \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & -1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} \delta_R \\ \delta_L \\ \tau_R \\ \tau_L \end{bmatrix} \tag{37}$$

doi:10.20944/preprints202402.0467.v1

3.4.2. Nonlinear Dynamics Inversion (NDI) Controller

The application of NDI to the attitude stabilization problem consists on the inversion of the model of the system so that the resulting inputs $[\delta_a, \delta_e, \tau_r]^T$ enable the aircraft to follow a desired dynamics, which can be made to depend on the relevant error variables:

$$\dot{\mathbf{w}}^B_{g,d} = \mathbf{K_w}(\mathbf{K_q}\mathbf{q}_{err,1:3} + \mathbf{w}^B_g) \tag{38}$$

Afterwards, (26) can then be inverted to solve for the control inputs accounting for this desired dynamics, consisting on a straightforward control law as depicted in (39) [14]:

$$\mathbf{u}_{NDI} = \begin{bmatrix} \delta_a \\ \delta_e \\ \tau_r \end{bmatrix} = \mathbf{G}^{-1}(\dot{\mathbf{w}}^B_{g,d} - \mathbf{F}(\mathbf{w}^B_g)) \tag{39}$$

3.4.3. Incremental Nonlinear Dynamics Inversion (INDI) Controller

The third attitude controller is the incremental version of NDI, INDI. Generically, the INDI control is deduced from (26) under the assumptions that the control inputs have higher impact in the dynamics of the aircraft and a high sample-rate is possible [13]. An increment for the control action $\Delta\mathbf{u}$ can then be computed accounting for the control-effectiveness matrix $\mathbf{G}$ to allow to track $\dot{\mathbf{w}}^B_{g,d}$, which is defined by (38) [17]:

$$\mathbf{u}_{INDI} = \begin{bmatrix} \delta_a \\ \delta_e \\ \tau_r \end{bmatrix}_k = \lambda\mathbf{G}^{-1}(\dot{\mathbf{w}}^B_{g,d} - \dot{\mathbf{w}}^B_g) + \begin{bmatrix} \delta_a \\ \delta_e \\ \tau_r \end{bmatrix}_{k-1} \tag{40}$$

where an additional scaling factor is included as it acts as a low-pass filter in the computation of the increment [13,42]. Nonetheless, (40) assumes that the angular acceleration is available, and therefore must be estimated. A second-order derivative filter presents a solution for this aspect, as it consists on a method of estimating $\dot{\mathbf{w}}^B_g$ from the measurements of the gyroscope $\mathbf{w}^B_{g,gyr}$, illustrated by [43]:

$$SD(s) = \frac{\omega^2_{SD}s}{s^2 + 2\zeta\omega_{SD}s + \omega^2_{SD}} \tag{41}$$

Assuming the damping coefficient $\zeta = 2$, the trade-off value for the cutoff frequency ω_{SD} must be found when tuning the INDI controller, aiming to find a balance between an acceptable level of noise and the delay introduced by the filtering operations [42]. An additional tool also helps in achieving a control action robust to noise is a command filter:

$$CF(s) = \frac{1}{\tau_{CF}s + 1} \tag{42}$$

which acts as a low-pass filter and a saturation to enforce the limits of the actuators, defined by a single parameter, τ_{CF} [43].

Figure 4 illustrates the different steps in computing the INDI control action $\mathbf{u}_{INDI}$, from the desired dynamics $\dot{\mathbf{w}}^B_g$ and gyroscope measurements $\mathbf{w}^B_{g,gyr}$.

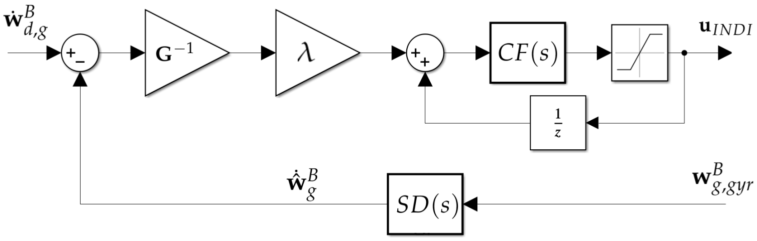


Figure 4. Block diagram representing the implementation of the INDI controller: the angular acceleration $\dot{\mathbf{w}}_{g}^{B}$ is estimated from gyroscope measurements using the second-order derivative filter, and compared to the desired dynamics $\dot{\mathbf{w}}_{g,d}^{B}$, and the difference is used to compute the required control increment. This is then added to the value of the control action of the previous time-step, and then filtered by the command-filter and subject to actuator saturation.

4. Hardware-in-the-Loop Simulation

The simulation for this work was developed in the Simulink environment of MATLAB 2021a, aiming to test the different attitude controllers with the simulator of the X-Vert and its sensors, and to validate the respective implementation of these in a microcontroller unit (MCU), performing a Hardware-in-the-Loop (HITL) simulation.

4.1. Hardware and Communications

The chosen microcontroller for this work was the Arduino Nano 33 IOT [44], as it fulfilled the requirements for both for the hardware-in-the-loop validation and experimental flight testing: it has a built-in 6 degrees-of-freedom (DOF) Inertial Measurement Unit (IMU), more specifically the LSM6DS3, connected via Inter-Integrated Circuit (I2C or I^{2}C), for the previously described attitude estimation; Wi-Fi and Bluetooth capacities for communications provided by the built-in NINA-W102 module; five pulse width modulation (PWM) pins, specially helpful for servomotor and BLDC motor control; a 32-bit Cortex-M0+ main processor, functioning at 48 Mhz, for the required onboard computations for the estimation and controller implementations; and additional wired communication capabilities, highlighting the Serial Peripheral Interface (SPI), which is of relevance due to its high velocity when compared to I2C.

Regarding the communications' method between the computer running MATLAB and the Arduino, the choice was to employ the User Datagram Protocol (UDP) over Ethernet, as this combination is relatively simple to implement, yet allowing the fast stream of data packets, particularly useful for real-time systems. To use UDP over Ethernet for HITL validation, an assembly was made that comprised the Arduino Nano 33 IOT and a W5500 Ethernet module [45], which are connected via SPI. Such assembly is shown in Figure 5, where an additional printed circuit board (PCB) can also be seen, designed to allow the usage of two 4-by-2 connectors, which are more convenient than the original layout on the W5500.

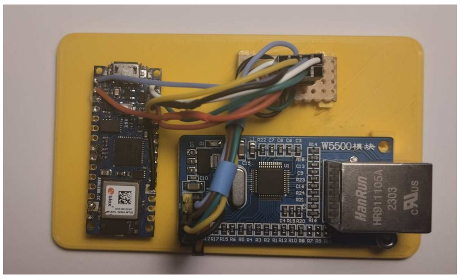


Figure 5. Assembly designed for the HITL development comprising the Arduino Nano 33 IOT board on the left, the W5500 Ethernet adapter on the bottom-right, and the adapter PCB on the top-right.

4.2. Benchmark Maneuver for Vertical Flight

To validate the different attitude control solutions, a set of maneuvers had to be designed, allowing to explore the degrees of freedom of the controlled UAV, while stabilized. As depicted in figure 1, these maneuvers will correspond to a time-varying vector of references, $\mathbf{y}_{ref}(t)$, described by:

$$\mathbf{y}_{ref}(t) = [u^B_{ref}, \mathbf{w}^B_{g,ref}, P_{D,ref}, \mathbf{q}^{NED}_{ref}]^T \tag{43}$$

comprising forward velocity, angular velocity, altitude and attitude values to be tracked by the controllers, where $\mathbf{w}^B_{g,ref}$ was taken as zero for attitude stabilization.

For this end, the set of maneuvers in vertical flight starts by take-off after 5 seconds, to ensure the communications and estimators have reached their steady-state, and holding an altitude of 2 meters. Afterwards, a 4 stage maneuver is run for every axis of the aircraft: a rotation of 15 degrees ($\frac{\pi}{12}$ rad) for 5 seconds, returning to fully-vertical orientation for another 5 seconds, then rotating in the opposite direction for the same time, and ending with another 5 second stop. The UAV executes this maneuver for each axis following a Y-Z-X order (corresponding to different pitch, yaw and roll angles), and, once concluded, performs a vertical landing. Using dashed lines to illustrate this set of references for $h = -P_D$, q_1, q_2 and q_3, Figure 6 presents a graphical interpretation of the described set of maneuvers. Lastly, u_{ref} was considered to be zero throughout the maneuvers, except for the take-off and landing procedures, and thus it is omitted as a reference for these plots.

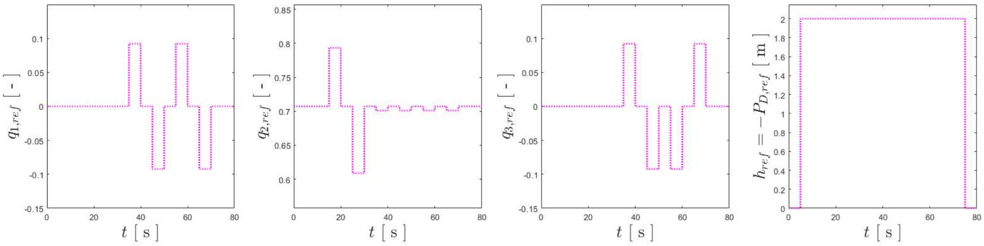


Figure 6. References for the benchmark maneuver for vertical flight. From left to right, in dashed pink: references for q_1, q_2, q_3 and altitude ($-P_D$).

4.3. Simulation Environment

The simulation in Simulink for HITL follows the general layout of a feedback-controlled system, as initially shown in Figure 1, but adapted to a Simulink environment (see Figure 7). The first block, in blue, generates the reference $\mathbf{r}(t)$ to be tracked; the green blocks consist on the control techniques and estimators, which were respectively addressed in section 3 and in the end of section 2; the yellow block implements the X-Vert model, together with models for its sensors; the communications with the Arduino board via Ethernet/UDP are managed by the orange-colored block; and the last block in red is included for visualization purposes. The controller and estimator blocks are shown in the same colour as they represent the software that must be implemented externally in the MCU. An additional switch (in white) is included to allow to interchange between the control action being generated by the Arduino and the one computed directly by MATLAB, allowing to test the control solutions in a purely-simulated environment or with the HITL implementation.

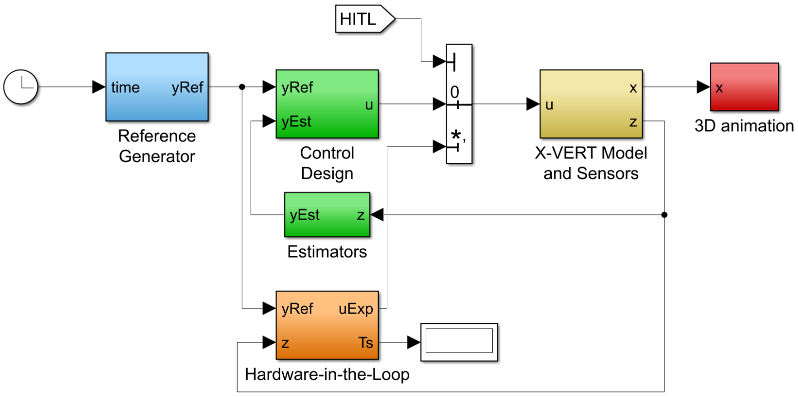


Figure 7. Overview of the simulated environment developed for testing the attitude control solutions in Simulink.

Some remarks regarding the simulation and HITL implementation:

- Sampling times: The simulation was run at 200 Hz, representing a fixed sample time of $T_{s,sim} = 0.005$ s, as it was considered as significantly low whilst still allowing the simulation to run in real-time without requiring high computational power. Regarding the implementations of the controllers and estimators, both in MATLAB and in Arduino, these were also kept at $T_{s,cont} = 0.005$ s for the same reasons, particularly for ensuring the Nano 33 IOT could perform all the tasks for each cycle under this time.
- Filter discretization: Despite the fact that MATLAB allows the implementation of transfer functions in continuous-time, a discrete form is desired to validate the MCU implementation of the HPF and LPF of the velocity estimation in (22), and the SD and CF filters - respectively (41) and (42) - for the INDI controller. The bilinear transformation:
$$s(z) = \frac{2}{T_{s,cont}} \left(\frac{1 - z^{-1}}{1 + z^{-1}} \right) \tag{44}$$
is employed for this end, and the deduction of the discrete expressions for each of the previously-referred filters is omitted.
- Estimators and altitude controller: Effort was taken to maintain the same parameters for the estimators and altitude controller, in both pure simulation and HITL, allowing for simulation runs to focus on the attitude controllers. With this in mind, the estimators were tuned using the values of $\alpha_{CF} = 0.99$ and $\beta_{CF} = 0.05$, respectively for (21) and (22), and the gains of the altitude controller in (32) were $k_{P_D} = 18$ and $k_u = 8$, kept from their original work [29].
- MCU implementation: The implementation in the Arduino Nano 33 IOT board follows a standard Arduino program flow: in the *setup*, the Ethernet communications are established and the relevant variables are initialized; and then an infinite *loop* is run every instant according to the sample-time $T_{s,cont}$, which consists on receiving the UDP packet (which comprises both the references and the simulated sensor data as can be seen in figure 7), performs the estimation of the attitude and vertical velocity, uses these to compute the control action, and sends it via an UDP packet back to MATLAB, completing the cycle. The controllers can be changed via the Simulink interface to allow for the same program to run all the different control strategies. Lastly, regarding all the code implementation, it should be noted that only the built-in functions available for Arduino, namely for Ethernet, Wi-Fi, UDP and SPI, were used, having been written the remaining necessary ones for the estimators and controllers.
- Visualization: For the goal of having a visual interface with the simulation, mainly for inspecting the attitude of the UAV, a three-dimensional environment was developed using the Simulink 3D Animation tools, and a computer-aided design (CAD) model of a bi-rotor tail-sitter from [46] was

included in it, as can be seen in Figure 8. In order to not overburden the simulation, a low update frequency of 10 Hz is used in this visualization block.

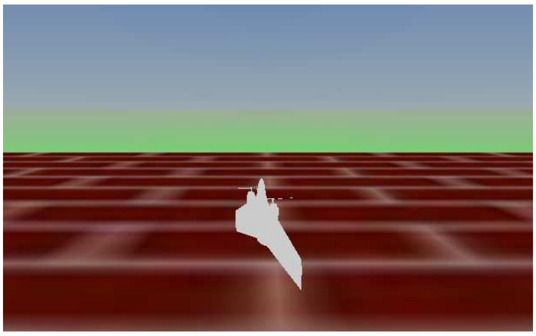


Figure 8. Simulink 3D environment with a tailsitter model [46].

4.4. Simulation and HITL Results

The simulations were run accounting for the considerations provided in this section. A significant delay was noticeable when the HITL simulations ran, which demanded some parameter re-tuning. The most noticeable case of this is the INDI controller running with HITL, where it was verified that the command filter introduced additional delay in the computed control action, which demonstrated to be unmanageable when added to the already-existing delay of the communications. Therefore, a value of $\tau_{CF}^{HITL} = 0$ was used, effectively removing the command filter component of the INDI controller.

Regarding the metrics to evaluate in order to be able to draw an objective comparison amongst the different control solutions for the X-Vert, two were chosen. The first was the root-mean-square error, RMS, of the three vectorial components of the quaternion, q_1, q_2 and q_3, when compared to the respective references:

$$RMS_{q_i} = \sum_{t=5}^{75} (q_{i,t}^{ref} - q_{i,t})^2 \quad , \quad i = 1,2,3 \tag{45}$$

to analyse the tracking performance. The second was the oscillation of the actuators, represented by μ, and obtained by computing the RMS of the control action with the one obtained after applying a tenth-order median filter, which aims to attest the smoothness of the control action of each controller, and its robustness to sensor noise:

$$\mu_{\delta_{e/a}} = \sum_{t=5}^{75} (\delta_{e/a,t} - \delta_{e/a,t}^{filt})^2 \quad \text{or} \quad \mu_{\tau_r} = \sum_{t=5}^{75} (\tau_{r,t} - \tau_{r,t}^{filt})^2 \tag{46}$$

The average of each metric is also provided to facilitate the aforementioned comparison. Lastly, it should be noted that the computations of these metrics are only made for the time interval comprised between 5 and 75 seconds, which mark respectively the take-off and the landing maneuvers.

4.4.1. Simulation Results

Firstly, simulation only tests were made, meaning that the model, the controllers and estimators were running inside the Simulink environment, without external hardware. The parameters for the attitude controllers used for simulation were the following:

- BNC (simulation): $\mathbf{K}_{ad}^{sim} = \text{diag}[60\,,\,60\,,\,60]$, $\mathbf{K}_{ap}^{sim} = \text{diag}[700\,,\,700\,,\,700]$
- NDI (simulation): $\mathbf{K}_{\mathbf{w}}^{NDI,sim} = \text{diag}[10\,,\,50\,,\,10]$, $\mathbf{K}_{\mathbf{q}}^{NDI,sim} = \text{diag}[5\,,\,20\,,\,5]$
- INDI (simulation): $\mathbf{K}_{\mathbf{w}}^{INDI,sim} = \text{diag}[10\,,\,10\,,\,10]$, $\mathbf{K}_{\mathbf{q}}^{INDI,sim} = \text{diag}[5\,,\,5\,,\,5]$, $\omega_{SD}^{sim} = 50$, $\tau_{CF}^{sim} = 0.01$, $\lambda^{sim} = 0.2$

Figure 9 presents the results obtained when tracking the three vectorial components of $\mathbf{q}_{ref}$, comparing these with the reference described previously, illustrated by Figure 6. It can be verified that all the controllers follow the references generically, with some small discrepancies. Similarly, Figure 10 presents the input vectors for attitude control, $\mathbf{u}_{att}$, for the different controllers, overlapping the plots so a comparison regarding their oscillation can be made. The results for this case are summarized in Table 1, rounded to the fourth decimal place.

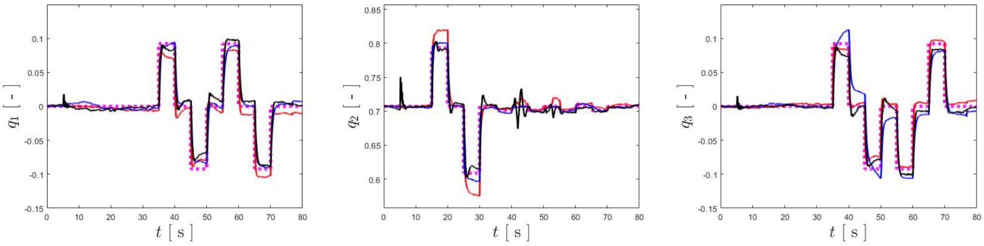


Figure 9. Tracking results for q_1, q_2 and q_3, respectively from left to right. The references are shown in dashed pink, the results for the BNC, NDI, and INDI are represented in red, blue and black, respectively.

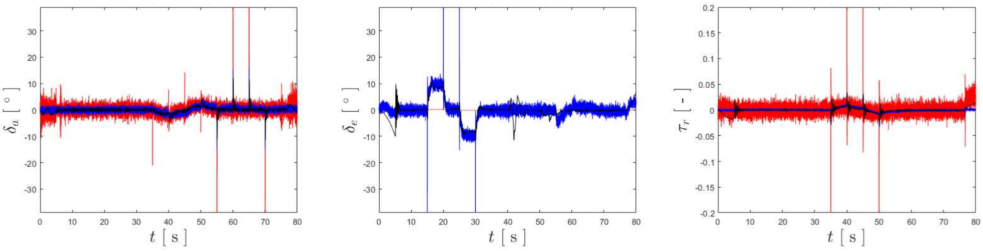


Figure 10. Plots for δ_a, δ_e and τ_r, respectively from left to right. The inputs generated by the BNC controller are represented in red, the ones from the NDI are shown in blue, and INDI in black.

Table 1. Simulation results.

Cont.	$RMS_{q_1}^{sim}$	$RMS_{q_2}^{sim}$	$RMS_{q_3}^{sim}$	$\overline{RMS}_q^{sim}$	$\mu_{\delta_a}^{sim}$	$\mu_{\delta_e}^{sim}$	$\mu_{\tau_r}^{sim}$	$\overline{\mu}_{u_{att}}^{sim}$
BNC	0.0133	0.0139	0.0116	0.0129	0.0273	0.0176	0.0104	0.0184
NDI	0.0178	0.0072	0.0221	0.0157	0.0122	0.0216	0.0011	0.0116
INDI	0.0171	0.0119	0.0166	0.0152	0.0012	0.0007	0.0001	0.0007

Inspecting Table 1 and figures 9-10, it can be seen that the BNC controller provides the overall best tracking performance, although demonstrating some issues when tracking q_2 and at the cost of significant oscillation of actuators. On the other hand, INDI demonstrates smoothness for all the required control inputs, showing excellent robustness to sensor noise, whilst also accurately tracking $\mathbf{q}_{ref}$. Lastly, the NDI controller has the highest value for $\overline{RMS}_q^{sim}$, but provides a middle-ground regarding oscillation of the actuators. It should also be noted that the NDI controller provides the best results for tracking q_2 of the three control solutions implemented, although at the cost of a higher value for $\mu_{\delta_e}^{sim}$.

4.4.2. Hardware-in-the-Loop Results

Taking the previous parameters as a starting point, the controllers were re-tuned to accommodate the HITL configuration, namely due to the communications delay between MATLAB and the Arduino board. For the BNC and NDI controllers this was a matter of performing a slight decrease in the gains, but for the INDI, not only the gains were adjusted, but also the cutoff frequency ω_{SD} had to be changed, and its command filter component had to be removed. Accounting for this, the controllers for HITL simulation account for the following values:

- BNC (HITL): $\mathbf{K}_{ad}^{HITL} = \text{diag}[60\,,\,60\,,\,60]$, $\mathbf{K}_{ap}^{HITL} = \text{diag}[500\,,\,500\,,\,500]$
- NDI (HITL): $\mathbf{K}_{\mathbf{w}}^{NDI,HITL} = \text{diag}[10\,,\,20\,,\,10]$, $\mathbf{K}_{\mathbf{q}}^{NDI,HITL} = \text{diag}[5\,,\,20\,,\,5]$
- INDI (HITL): $\mathbf{K}_{\mathbf{w}}^{INDI,HITL} = \text{diag}[5\,,\,5\,,\,5]$, $\mathbf{K}_{\mathbf{q}}^{INDI,HITL} = \text{diag}[5\,,\,5\,,\,5]$, $\omega_{SD}^{HITL} = 100$, $\tau_{CF}^{HITL} = 0$, $\lambda^{HITL} = 0.1$

For such values, the hardware-in-the-loop simulations with the Arduino as part of the control loop were run as specified before, providing the results on Table 2.

Table 2. Hardware-in-the-Loop results.

Cont.	$RMS_{q_1}^{HITL}$	$RMS_{q_2}^{HITL}$	$RMS_{q_3}^{HITL}$	$\overline{RMS}_{q}^{HITL}$	$\mu_{\delta_a}^{HITL}$	$\mu_{\delta_e}^{HITL}$	$\mu_{\tau_r}^{HITL}$	$\overline{\mu}_{u_{att}}^{HITL}$
BNC	0.0246	0.0215	0.0142	0.0201	0.0295	0.0182	0.0108	0.0195
NDI	0.0201	0.0117	0.0227	0.0182	0.0117	0.0083	0.0011	0.0070
INDI	0.0212	0.0143	0.0188	0.0181	0.0028	0.0011	0.0003	0.0014

Omitting the figures for the HITL scenario, Table 2 allows to draw slightly different conclusions of the ones taken for the simulation-only scenario: both the NDI and INDI controllers demonstrated excellent tracking capacity, with the INDI standing out due to the smoothness of actuation. The BNC controller provides marginally worse results when it comes to $\overline{RMS}_{q}^{HITL}$, but evidences the highest oscillation of actuation as evidenced by the comparison of the values of $\overline{\mu}_{u_{att}}^{HITL}$ for all controllers.

Such results hint that the delay in communications impacted negatively the performances of both the NDI and BNC, but the first appears to suffer the consequences of this the most, whilst the INDI appears to be more robust to this delay.

5. Experimental Validation

The last component of the research developed for this work is the experimental implementation and testing of the control solutions described in section 3, after these have been validation in simulation and HITL. The different aspects of this experimental component are described now, providing the results from the flight trials and a comparison among the different nonlinear control strategies.

5.1. Flight Controller Design

The original flight controller (FC) of the X-Vert acted not only as a FC by itself with a dedicated board of sensors, but as a receiver for the transmitter and as a pair of Electronic Speed Controllers (ESC) for the BLDC motors. Since this board was not re-programmable, these different functions had to be covered in a different way. Starting with a dedicated FC, the following components were used, many of which being native to the Nano 33 IOT board [44]:

- Microcontroller Unit (MCU): Cortex M0+ (native to Arduino board)
- Inertial Measurement Unit (IMU): LSM6DS3 accelerometer and gyroscope (native to Arduino board)
- Sonar: HC-SR04 ultrasonic sensor
- Communications: UDP/Wi-Fi via NINA W102 module (native to Arduino board)

For the experimental scenario, the UDP communications were reformulated to work via Wi-Fi instead of Ethernet, which required minimal adjustment. A dedicated case was designed and 3D printed to accommodate the Arduino board, the HC-SR04 ultrasonic sensor, and a support PCB for the servo and ESC connectors, which can be seen in Figure 11.

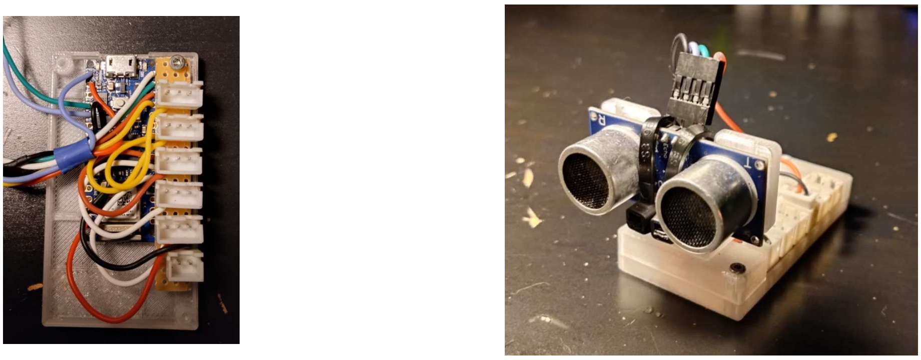


Figure 11. Flight controller based on the Arduino Nano 33 IOT (left) and the HC-SR04 ultrasonic sensor (right).

The integration of the aforementioned FC into the frame of the X-Vert required additional steps as well, namely to provide power to the required components. A power distribution board was included to regulate the voltage provided from the battery, powering the ESCs, servos and flight controller in their adequate levels. The electronic components used in the experimental assembly were the following:

- Servos: Spektrum A220 4g servos [47]
- ESC: SkyRC 20A Nano ESC with BLHeli firmware [48]
- BLDC motors: BL280 2600Kv Brushless Outrunner Motor [49]
- PDB: Matek Mini Power Hub [50]
- Battery: Gens Ace 450 mAh 7.4V battery [51]

Figure 12 presents a diagram of the connections of the different electronic components, which were assembled in the frame of the X-Vert as shown previously in Figure 2.

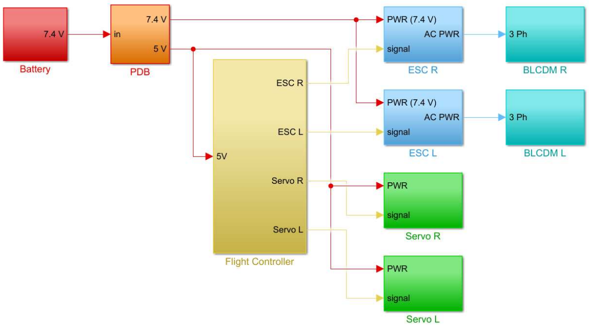


Figure 12. Diagram of electronics of the experimental assembly. From left to right: battery (red); PDB (orange); FC (yellow); ESCs (blue) and BLDC motors (cyan); and servos (green). The power supply lines can be seen in red displaying the respective voltage levels, the signal lines are shown provided from the FC are shown in yellow, and the three-phase ones are shown in blue.

The management of the simulation was once again made resorting to Simulink, which was responsible for providing the references to the flight controller, and to register the control action and the estimated attitude computed by it.

In order to verify the effectiveness of this communication strategy, while also taking the opportunity to partially validate the different control strategies, a test assembly as designed and 3D printed, which comprised the electronics (with exception of the servos) and propellers, as shown in Figure 13. Since the beam-like frame with the two motors and propellers is able to pivot around its center by the means of a bearing, mounting the FC in it allows the testing of the stabilization algorithms of the z axis of the UAV, and also ensures that any errors and problems in the implementation are detected. This proved to be of vital importance since the X-Vert is a fragile aircraft, and initial crashes demonstrated to significantly hinder the development.

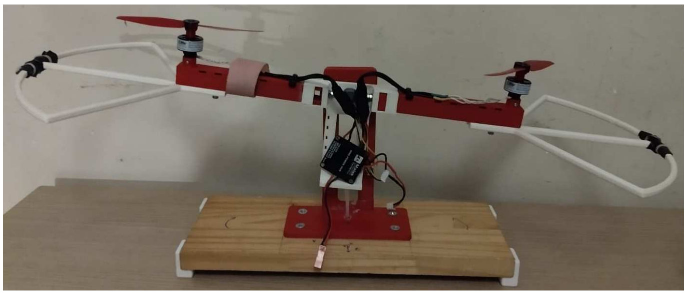


Figure 13. 3D printed test assembly, designed to validate the communications and controller implementation.

5.2. Groundtruth

In order to evaluate the performance of the controllers in the experimental scenario, a method of effectively tracking the position and attitude of the X-Vert is required. A motion capture system (MCS) covering an area of 12 by 4 metres enveloped in a safety net, available at the host facility IDMEC - Instituto de Engenharia Mecânica, was used for this end. Using this set of cameras and the dedicated Qualysis Track Manager software by Qualysis [52], together with reflective markers placed on the surface of the X-Vert, it was possible to track both $\mathbf{p}^{NED}$ and $\mathbf{q}^{NED}$ with high-precision inside the aforementioned area. Figure 14 shows the utilized installations and also provides the illustration of the area using the MCS, where the reconstruction of the X-Vert can be seen. The coordinate frame of the MCS does not match the previously-described NED orientation, but care was taken to process the gathered data to account for the necessary conversion. Accounting for this, the position and orientation of the X-Vert provided by the MCS are assumed as the ground-truth for the experimental trials, therefore represented directly by $\mathbf{p}^{NED}$ and $\mathbf{q}^{NED}$, respectively.

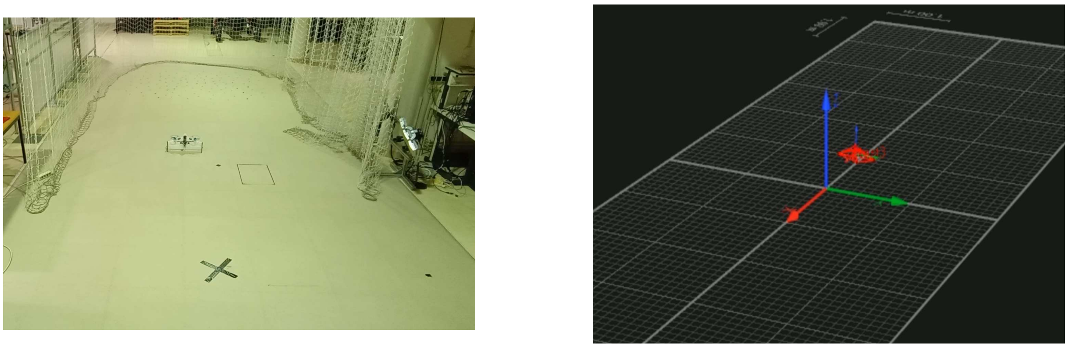


Figure 14. On the left: 12-by-4 meter net-protected arena used. On the right: 3D reconstruction of the X-Vert using the MCS.

5.3. Benchmark Maneuver for Experimental Vertical Flight

The benchmark maneuver described in section 4 had to be adapted to the context of experimental tests, where the available flight area is limited to the arena previously described. A decision was taken to focus on the pitch control of the aircraft, as it was considered the most demanding to stabilize when in vertical flight, and presents significant challenges for the transition to aerodynamic flight in future work. Therefore, the solution found for performing the experimental trials was to order the X-Vert to perform a take-off and hold altitude at 0.5 metres, manually guide it to the center of the arena if a substantial deviation happened during the initial maneuvers, and only then provide a set of references for q_2. This set consisted on a rotation of -10 degrees around the y axis of the aircraft when it is fully vertical, hold this attitude for 2 seconds, return to vertical orientation for 2 seconds, and then perform a symmetric rotation for another 2 seconds. This was made to ensure that the same reference would be provided during the tests for different controllers, allowing to draw suitable conclusions. Nonetheless, depending on the actual performance of the controller, additional lateral corrections were sometimes provided in order to avoid collisions, but this was avoided as most as possible.

An example of references of an experimental test, particularly the one made for the BNC controller, is shown in Figure 15, which evidences some small corrections during the process of guiding the UAV for the center of the arena, together with the aforementioned references of 2 seconds for q_2 and the constant reference altitude of 0.5 metres.

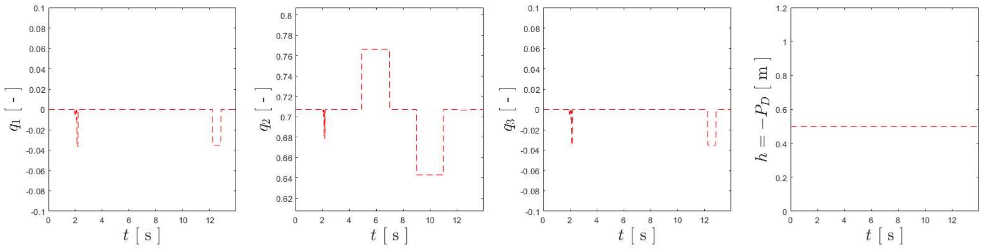


Figure 15. References for the experimental trial for the BNC controller.

5.4. Parameter Tuning for Experimental Flight

The adaptation for an experimental scenario demanded some adaptation of the parameter tuning in order to accommodate such change. It was soon realized that the Arduino Nano 33 IOT could not perform the cycles at 200 [Hz] when using UDP over Wi-Fi, and thus a sample time $T_{s,exp} = 0.01$ [s] was used instead, as it proved to be sufficiently low to stabilize the aircraft. The estimators and controllers were subject to a re-tuning as well, imposed not only by the transition from simulation to experimental, but also because of the change in the sampling frequency. Similarly to the simulation and HITL results, effort was taken to ensure that the parameters of the estimators and altitude controller were the same for the trials using different attitude controllers, and their values are provided in (47).

$$\alpha_{CF}^{exp} = 0.905 \quad , \quad \beta_{CF}^{exp} = 0.05 \quad , \quad k_{p_D}^{exp} = 10 \quad , \quad k_u^{exp} = 5 \tag{47}$$

Regarding the controller gains and remaining parameters, these also had to be adjusted for the experimental trials, and the values are shown next:

- BNC (experimental): $\mathbf{K}_{ad}^{exp} = \mathrm{diag}[20\,,\,60\,,\,20]$, $\mathbf{K}_{ap}^{exp} = \mathrm{diag}[200\,,\,500\,,\,200]$
- NDI (experimental): $\mathbf{K}_{\mathbf{w}}^{NDI,exp} = \mathrm{diag}[20\,,\,20\,,\,20]$, $\mathbf{K}_{\mathbf{q}}^{NDI,exp} = \mathrm{diag}[10\,,\,20\,,\,10]$
- INDI (experimental): $\mathbf{K}_{\mathbf{w}}^{INDI,exp} = \mathrm{diag}[2\,,\,5\,,\,2]$, $\mathbf{K}_{\mathbf{q}}^{INDI,exp} = \mathrm{diag}[5\,,\,5\,,\,5]$, $\omega_{SD}^{sim} = 100$, $\tau_{CF}^{exp} = 0$, $\lambda^{exp} = 0.2$

As can be seen, $\tau_{CF}^{exp} = 0$ implies that there was no filtering made by the command-filter described in section 3. It was not deemed necessary since the INDI already provided a smooth actuation, as it will be described next, and the additional filtering was introducing unwanted delay.

5.5. Vertical Flight Results

Using the described setup, multiple trials were run for each of the controllers, and the data of each was registered using the following process: the references $\mathbf{r}$, generated by MATLAB, were saved in a file generated at the end of each flight trial; in the same file, the quaternion $\hat{\mathbf{q}}^{NED}$ estimated by the FC and the control action $\mathbf{u}$ computed by it and sent to MATLAB via Wi-Fi/UDP were also saved; the orientation and position provided by the MCS, $\mathbf{q}^{NED}$ and $\mathbf{p}^{NED}$, were exported from the QTM software at the end of the trials to a separate. Matching the timestamps of both files, it was possible to overlap them, effectively allowing to compare the reference quaternion, the respective estimation by the onboard FC, and the quaternion provided by the MCS. Using the same experimental trial for the

BNC controller as shown in Figure 15, the curves for $\mathbf{q}_{ref}^{NED}$, $\hat{\mathbf{q}}^{NED}$ and $\mathbf{q}^{NED}$ - from take-off to landing - are shown in Figure 16, where it can also be seen the comparison between the reference altitude $P_{D_{ref}} = -0.5$ [m] and its value provided by the MCS, assumed as ground-truth. It should be noted that the estimated altitude was assumed to be of less relevance for the purpose of comparing attitude controllers, and thus was omitted in order to reduce the amount of data being transmitted.

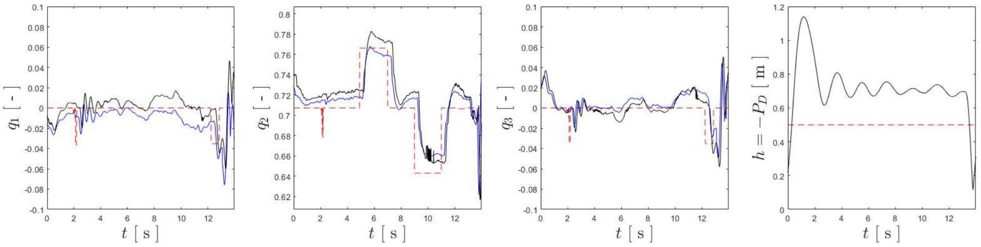


Figure 16. Full-length experimental trial for the BNC controller, where the references are shown in dashed red, the ground-truth is represented by black, and the estimated values are presented in blue.

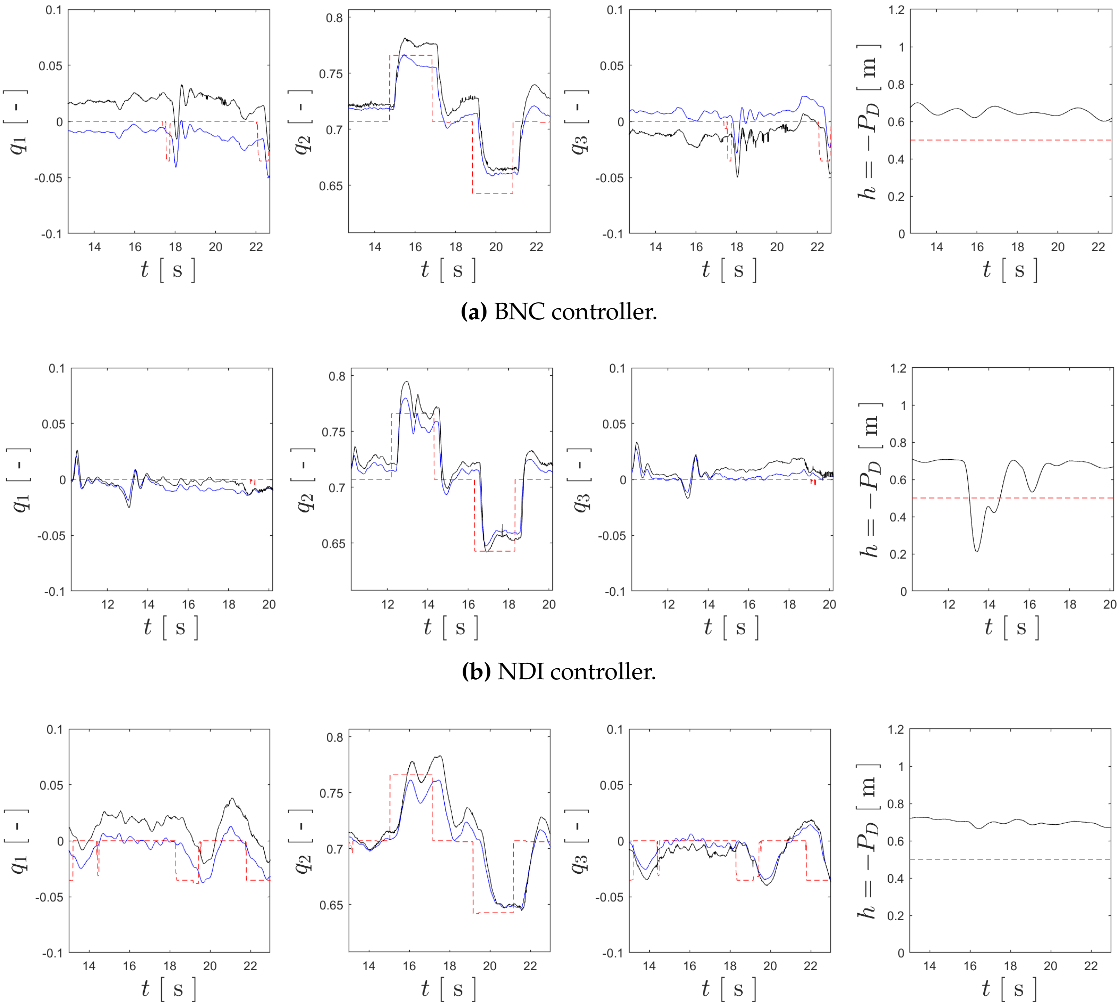
For the purposes of computing the evaluation metrics RMS_q^{exp} and μ^{exp}, a 10 second window was considered for each trial, corresponding to an interval of 2 seconds before and 2 seconds after the references being sent. This ensured that the comparison was fair, regardless of the length of the trials for each controller. Figure 17 present the tracking results for each controller, according to the previously-described 10 second window of each flight trial.

Analysing the above images, it is clear that all the controllers are capable of tracking the references in q_2, while also stabilizing the remaining degrees-of-freedom of the attitude. The BNC controller provides excellent tracking of q_2 but its performance worsens for q_1 and q_3, while the NDI controller manages to keep these more closely at null values, but has some some spikes when tracking q_2, although still tracking it with precision. In the trial for the INDI controller strategy, additional lateral references were provided in order to avoid collisions, but the controller still managed to track the references. Nonetheless, since these lateral references were provided simultaneously to the q_2 references, some performance loss may have happened. Table 3 provides the evaluation metrics that summarize the tracking results for the experimental trials, where it can be seen that the NDI has the best overall performance, whilst the INDI has the worse, possibly evidencing the aforementioned loss of performance for this controller, and the BNC provides the middle-ground among the three controllers. Lastly, regarding altitude tracking, it is clear that there is a sudden drop in altitude in the trial for the NDI controller, which was duly analysed, and the explanation is provided at the end of after the analysis regarding actuator oscillation.

Table 3. Experimental results in tracking $\mathbf{q}_{ref}^{NED}$.

Cont.	$RMS_{q_1}^{exp}$	$RMS_{q_2}^{exp}$	$RMS_{q_3}^{exp}$	$\overline{RMS}_q^{exp}$
BNC	0.0219	0.0292	0.0149	0.0220
NDI	0.0071	0.0264	0.0115	0.0150
INDI	0.0271	0.0352	0.0232	0.0285

Moving on to the analysis over actuator oscillation, Figure 18 provides the plots of $\mathbf{u} = [\delta_a, \delta_e, \tau_r, \tau_t]^T$ registered for the trial of each controller.



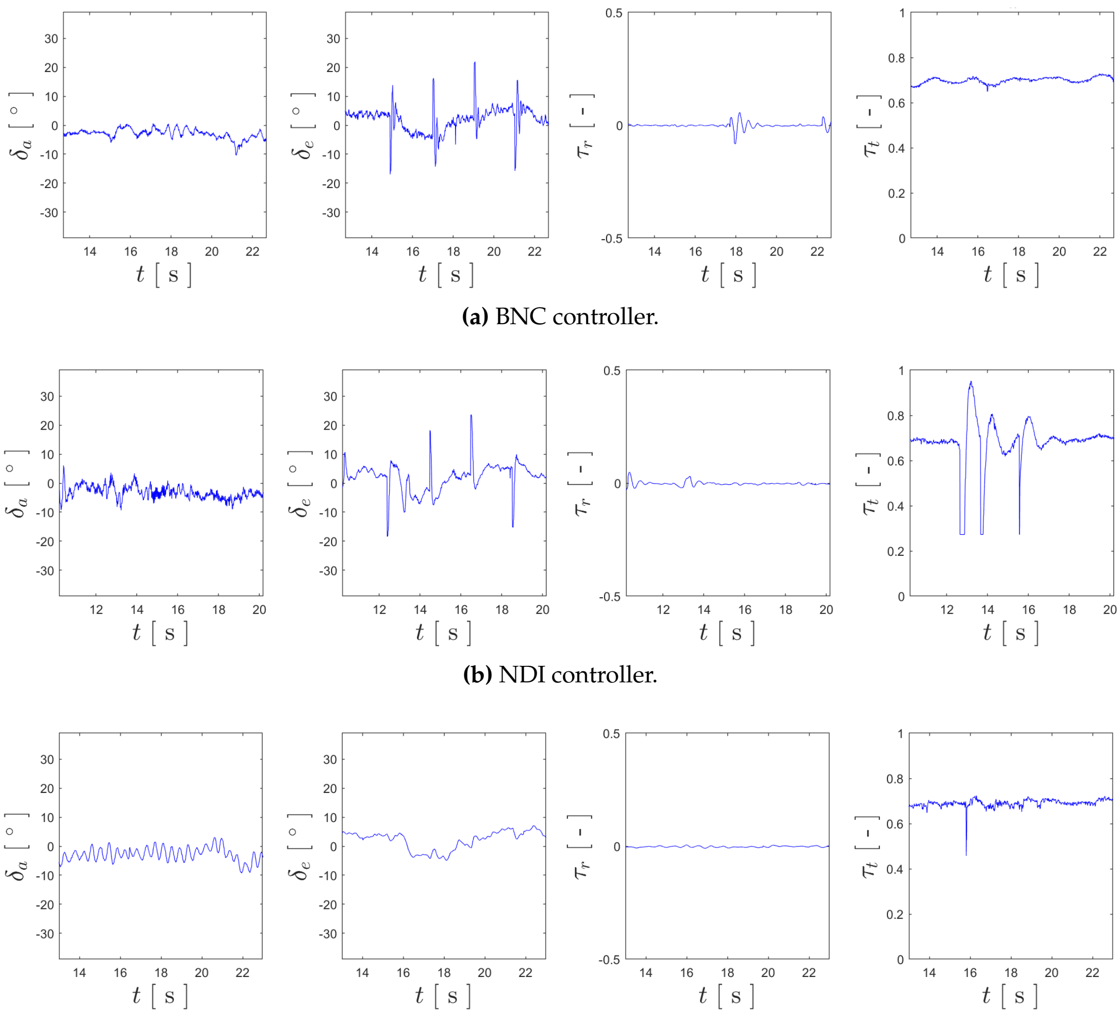
(a) BNC controller.

(b) NDI controller.

(c) INDI controller.

Figure 17. Tracking results for the different attitude controllers, where the references are shown in dashed red, the ground-truth is represented by black, and the estimated values are presented in blue.

Focusing the analysis on the inputs used for attitude stabilization - δ_a, δ_e and τ_r - the first obvious conclusion that can be drawn is that there is far less noise in the respective plots when compared to the simulation and HITL results. This may be justified by the reduction of the values of the gains, but could also evidence excessive noise when modeling the sensors. Analysing the smoothness of the curves of Figure 18, it can be stated that the NDI and BNC appear to be similar, with the BNC being more noisy for δ_a, and the NDI for δ_e, and that there is virtually absence of oscillation for τ_r for all the three controllers. For the elevator input, the INDI controller has a far smoother actuation when compared to the remaining two control strategies, not showing any of the sudden spikes that appear in the δ_e curves of the trials for the NDI and BNC. Nonetheless, the curve for δ_a has an almost constant oscillation, which could have been cause by the previously-mentioned simultaneous lateral references during the maneuvers. Summarizing such results, Table 4 provides the values for $\mu_{\delta_a}^{HITL}$, $\mu_{\delta_e}^{HITL}$ and $\mu_{\tau_r}^{HITL}$, as well as their average value.



(a) BNC controller.

(b) NDI controller.

(c) INDI controller.

Figure 18. Plots of each component of $\mathbf{u} = [\delta_a, \delta_e, \tau_r, \tau_t]^T$ recorded for the flight trials of each controller.

Table 4. Experimental results in oscillation of $\mathbf{u}_{att}$.

Cont.	$\mu^{exp}_{\delta_a}$	$\mu^{exp}_{\delta_e}$	$\mu^{exp}_{\tau_r}$	$\overline{\mu}^{exp}_{u_{att}}$
BNC	0.0056	0.0213	0.0018	0.0095
NDI	0.0144	0.0156	0.0012	0.0104
INDI	0.0039	0.0018	0.0002	0.0020

Lastly, some considerations about the altitude control should be provided, namely the sudden loss of altitude shown in Figure 17 for the experimental trial of the NDI controller. As Figure 18 also demonstrates for the same trial, the curve of τ_t has abrupt decreases, which cannot be justified by excessive altitude. A similar event also happens for the experimental flight for the INDI controller, although to a smaller degree. After thorough analysis, it was found that this was caused by the partial obstruction of the sonar by the tail of the X-Vert when its pitch went beyond 90 degrees. Since this obstruction caused for the sonar not to receive a response signal (activating a timeout), it assumed a far greater altitude than the real one, and the controller responds accordingly, decreasing the throttle input. This issue was acknowledged and in general the controllers demonstrated robustness to its impact, but it should be addressed in future work.

Conclusions

In this research work, a simulator for a tail-sitter UAV was developed, allowing for the testing of different control strategies in a simulated scenario. The attitude controllers tested were of nonlinear nature, one from a previous work developed for the same aircraft [29], one based on Nonlinear Dynamics Inversion principles, and the last being its incremental version, INDI. The controllers and estimators were implemented in an Arduino board and integrated into the simulated environment, in order to enable hardware-in-the-loop testing of the control solutions. The simulation and HITL results for performing standard maneuvers in vertical flight demonstrated that all the implemented controllers manage to track the references, although with the INDI control strategy standing out by providing a far smoother actuation.

Advancing to an experimental scenario, a tail-sitter was adapted in order to accept the Arduino-based flight controller, and indoor experimental trials for vertical flight were conducted. The results of these highlight once again the smoothness of actuation of the INDI control strategy, nonetheless demonstrating that all the implemented controllers are capable of stabilizing the X-Vert in vertical flight.

Author Contributions: Conceptualization, A.A., A.M. and J.R.A.; methodology, A.A., A.M. and J.R.A.; software, A.A. and J.R.A.; validation, A.M. and J.R.A.; formal analysis, A.A.; investigation, A.A.; resources, A.M.; data curation, A.A.; writing—original draft preparation, A.A.; writing—review and editing, A.M. and J.R.A.; visualization, A.A.; supervision, A.M. and J.R.A.; project administration, A.M.; funding acquisition, A.M.. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by National Funds by FCT – Fundação para a Ciência e Tecnologia, I.P., through IDMEC under project Eye in the Sky (PCIF/SSI/0103/2018), and under LAETA, project UIDB/50022/2020.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AC	Aerodynamic Center
AOA	Angle-of-attack
BLDC	Brushless Direct Current
BNC	Benchmark Nonlinear Controller
CAD	Computer-Aided Design
CF	Complementary/Command Filter
CG	Center-of-Gravity
DOF	Degree-of-Freedom
ESC	Electronic Speed Controller
FC	Flight Controller
HITL	Hardware-In-The-Loop
HPF	High-Pass Filter
I2C	Inter-Integrated Circuit
IMU	Inertial Measurement Unit
INDI	Incremental Nonlinear Dynamics Inversion
LPF	Low-Pass Filter
LQR	Linear Quadratic Regulators
MAC	Mean Aerodynamic Chord
MCS	Motion Capture System
MCU	Micro-controller Unit

MDPI	Multidisciplinary Digital Publishing Institute
NDI	Nonlinear Dynamics Inversion
NED	North-East-Down
PCB	Prrinted Circuit Board
PID	Proportional Integral-Derivative
PWM	Pulse WidthModulation
QTM	Qualysis Track Manager
RMS	Root-Mean-Square
SD	Second-(order) Derivative
SPI	Serial Peripheral Interface
UAV	Unmanned Aerial Vehicles
UDP	User Datagram Protocol
VTOL	Vertical Takeoff and Landing

Appendix A. X-Vert Simulator Parameters and Constants

Appendix A.1. General Parameters

Table A1. Geometry and mass properties of the X-VERT VTOL [29].

Prop	Value	SI	Description
b_w	0.500	m	wingspan
c_w	0.154	m	MAC
S_w	0.077	m^2	wing surface area
Λ_w	19.80	$^{\mathrm{o}}$	sweep angle of the wing
$\mathbf{d}_{CG}$	$[0.130, 0, 0]^T$	m	distance of the CG to the TE
m_{uav}	0.220	kg	aircraft mass
$\mathbf{J}_{uav}$	$\begin{bmatrix} 3.0{\cdot}10^{-3} & 0 & -14{\cdot}10^{-6} \\ 0 & 6.2{\cdot}10^{-4} & 0 \\ -14{\cdot}10^{-6} & 0 & 3.5{\cdot}10^{-3} \end{bmatrix}$	$\mathrm{kg{\cdot}m}^2$	aircraft inertia matrix
c_f	0.062	m	elevon chord
b_f	0.190	m	elevon span
δ_{max}	0.681	rad	elevon maximum deflection

Appendix A.2. Propulsion Subsystem

Table A2. Parameters of the propulsion components of the X-VERT VTOL [29].

Prop	Value	SI	Description
$\mathbf{d}_{pr,R}$	$[0.037, 0.144, 0]^T$	m	distance of the right proprotor to the CG
$\mathbf{d}_{pr,L}$	$[0.037, -0.144, 0]^T$	m	distance of the left proprotor to the CG
r_{prop}	0.0625	m	radius of the propeller
$c_{T,2}$	-0.1281	-	second order parameter of C_T
$c_{T,1}$	-0.1196	-	first order parameter of C_T
$c_{T,0}$	0.1342	-	zeroth order parameter of C_T
$c_{P,2}$	-0.0602	-	second order parameter of C_P
$c_{P,1}$	-0.0146	-	first order parameter of C_P
$c_{P,0}$	0.0522	-	zeroth order parameter of C_P
V_{bat}	7.400	V	voltage of the battery
R_m	0.250	Ω	resistance of the electric motor
K_e	$3.7 \cdot 10^{-3}$	V/(rad·s)	back-electromotive force of the electric motor
J_{pr}	$4.2 \cdot 10^{-7}$	kg·m^2	rotational inertia of the proprotor
K_t	$2.8 \cdot 10^{-3}$	(N·m)/A	torque constant of the electric motor
B_m	$8.4 \cdot 10^{-6}$	(N·m·s)/rad	damping constant of the electric motor

Appendix A.3. Aerodynamics Subsystem

Equations for C_{L_α}, C_{D_α} and C_{m_α} aerodynamic curves from [29]:

$$C_{L_\alpha}(\alpha,\delta) = 0.7\left(\sin(2\alpha) + \frac{1.5\sin(2\alpha)}{1+100\sin^4(\alpha)}\right) + \left(-0.2\sin\left(|\alpha|\right) + 0.2\cos^2(\alpha)\right)\left(\frac{\delta}{\delta_{max}}\right) \quad [\text{ - }] \tag{A1a}$$

$$C_{D_\alpha}(\alpha,\delta) = 0.1 + 1.1\sin^2\left(\alpha + \frac{c_f}{c_w}\delta\right) \quad [\text{ - }] \tag{A1b}$$

$$C_{m_\alpha}(\alpha,\delta) = -0.35\sin\left(\alpha + 0.2\frac{\delta}{\delta_{max}}\right) - 0.5\left(\frac{\sin(\alpha)}{1+100\sin^4(\frac{\alpha-\pi}{2})}\right) + \left(\frac{\delta}{\delta_{max}}\right)\left(\frac{-0.1\sin\left(\alpha + \left|0.8\frac{\delta}{\delta_{max}}\right|\right)}{1+400\sin^6(\frac{\alpha-\pi}{2})}\right) \quad [\text{ - }] \tag{A1c}$$

Table A3. Aerodynamic parameters for the X-Vert VTOL [38].

Prop	Value	SI	Description
$\mathbf{d}_{AC,R}$	$[-0.0037, 0.1250, 0]^T$	m	aerodynamic centre of the right wing
$\mathbf{d}_{AC,L}$	$[-0.0037, -0.1250, 0]^T$	m	aerodynamic centre of the left wing
C_{L_q}	3.1851	-	lift stability derivative for pitch
C_{m_q}	-2.4487	-	pitch damping derivative
C_{Y_β}	-0.0025	-	side-force derivative for side-slip
C_{Y_p}	0.2620	-	side-force stability derivative for roll
C_{Y_r}	-0.0673	-	side-force derivative for yaw
C_{l_β}	-0.1604	-	roll static stability derivative
C_{l_p}	-0.4506	-	roll damping derivative
C_{l_r}	0.3107	-	rolling moment derivative for yaw
C_{n_β}	0.0390	-	yaw static stability derivative
C_{n_p}	-0.1890	-	yawing moment derivative for roll
C_{n_r}	0.0028	-	yaw damping derivative

Appendix A.4. Ground-Contact Subsystem

Table A4. Constants for the ground contact forces [29].

Prop	Value	SI	Description
$\mathbf{r}_{c,1}$	$[0.117, 0, 0]^T$	m	distance of the tip of the fuselage to the CG
$\mathbf{r}_{c,2}$	$[-0.147, 0.250, -0.073]^T$	m	distance of corner 1 to the CG
$\mathbf{r}_{c,3}$	$[-0.147, 0.250, 0.073]^T$	m	distance of corner 2 to the CG
$\mathbf{r}_{c,4}$	$[-0.147, -0.250, 0.073]^T$	m	distance of corner 3 to the CG
$\mathbf{r}_{c,5}$	$[-0.147, -0.250, -0.073]^T$	m	distance of corner 4 to the CG
$k_{c,p}$	100	$1/\text{s}$	position gain for the contact forces
$k_{c,v}$	5	$1/\text{s}^2$	velocity gain for the contact forces

Appendix A.5. Sensors

Table A5. Sensor properties of the X-VERT VTOL [31].

Prop	Value	SI	Description
$\mathbf{b}_{acc}$	$[0,0,0]^T$	m/s^2	bias in the accelerometer readings
σ_{acc}	0.05	-	standard deviation of the noise of the accelerometer
$\mathbf{b}_{gyr}$	$[0,0,0]^T$	rad/s	bias in the gyroscope readings
σ_{gyr}	0.03	-	standard deviation of the noise of the gyroscope
b_{son}	0	m	bias in the sonar readings
σ_{acc}	0.01	-	standard deviation of the noise of the sonar

References

1. Telli, K., Kraa, O., Himeur, Y., Ouamane, A., Boumehraz, M., Atalla, S., Mansoor, W.,. A comprehensive review of recent research trends on unmanned aerial vehicles (uavs) *Systems* **2023**, *11(8)*, 400.
2. Xu, H., Wang, L., Han, W., Yang, Y., Li, J., Lu, Y., Li, J., A, Survey on UAV Applications in Smart City Management: Challenges, Advances, and Opportunities. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* **2023**, *16*.
3. Chehreh , B., Moutinho , A., Viegas , C., Latest Trends on Tree Classification and Segmentation Using UAV Data—A Review of Agroforestry Applications. *Remote Sensing* **2023**, *15(9)*, 2263.
4. Rakesh, D., Kumar, N. A., Sivaguru, M., Keerthivaasan, K. V. R., Janaki, B. R., Raffik, R., ole of UAVs in innovating agriculture with future applications: A review. *2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)* **2021**.
5. Saeed, A. S., Younes, A. B., Cai, C., Cai, G. A survey of hybrid unmanned aerial vehicles. *Progress in Aerospace Sciences* **2018**, *98*.
6. Ducard, G. J., Allenspach, M. Review of designs and flight control techniques of hybrid and convertible VTOL UAVs. *Aerospace Science and Technology* **2021**, *118*.
7. Misra, A., Jayachandran, S., Kenche, S., Katoch, A., Suresh, A., Gundabattini, E., ... Legesse, A. A. A Review on Vertical Take-Off and Landing (VTOL) Tilt-Rotor and Tilt Wing Unmanned Aerial Vehicles (UAVs). *Journal of Engineering* **2022**.
8. Okasha, M., Shah, J., Fauzi, W., Hanouf, Z. Gain scheduled linear quadratic control for quadcopter. *IOP conference series: materials science and engineering* **2017**.
9. Poksawat, P., Wang, L., Mohamed, A. Gain scheduled attitude control of fixed-wing UAV with automatic controller tuning. *IEEE Transactions on Control Systems Technology* **2017**, 26(4).
10. Qiao, J., Liu, Z., Zhang, Y. Gain scheduling based PID control approaches for path tracking and fault tolerant control of a quad-rotor UAV. *International Journal of Mechanical Engineering and Robotics Research* **2018**, *7(4)*.
11. Melo, A. G., Andrade, F. A., Guedes, I. P., Carvalho, G. F., Zachi, A. R., Pinto, M. F. Fuzzy gain-scheduling PID for UAV position and altitude controllers. *Sensors* **2022**, *22(6)*.
12. Pashilkar, A. A., Ismail, S., Ayyagari, R., Sundararajan, N. Design of a nonlinear dynamic inversion controller for trajectory following and maneuvering for fixed wing aircraft. *2013 IEEE symposium on computational intelligence for security and defense applications (CISDA)* **2013**.
13. Azinheira, J.R., Moutinho, A. Hover Control of an UAV With Backstepping Design Including Input Saturations. *Journal Abbreviation* **2014**.
14. Horn, J. F. Non-linear dynamic inversion control design for rotorcraft. *Aerospace* **2019**, *6(3)*.

15. Wang, F., Wang, P., Deng, H., Chen, B. Nonlinear Dynamic Inversion Control of VTOL Tilt-Wing UAV. *018 Eighth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC)* **2018**.
16. Y. Fu, Y. Zhang, Z. Yu, Z. Liu. A Backstepping Control Strategy for Fixed Wing UAV under Actuator Failure. *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)* **2019**.
17. Smeur, E. J., Chu, Q., De Croon, G. C.. Adaptive incremental nonlinear dynamic inversion for attitude control of micro air vehicles. *of Guidance, Control, and Dynamics* **2016**, *39(3)*.
18. Azinheira, J. R., Moutinho, A., Carvalho, J. R. Lateral control of airship with uncertain dynamics using incremental nonlinear dynamics inversion. *IFAC-PapersOnLine* **2015**, *48(19)*.
19. Sieberling, S., Chu, Q. P., Mulder, J. A. Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction. *Journal of guidance, control, and dynamics* **2019**, *33(6)*.
20. Acquatella, P., van Kampen, E., Chu, Q. P. Incremental backstepping for robust nonlinear flight control. *Proceedings of the EuroGNC* **2013**.
21. Guerreiro, N. M., Moutinho, A. Robust incremental backstepping controller for the attitude and airspeed tracking of a commercial airplane. *2019 IEEE 10th International Conference on Mechanical and Aerospace Engineering (ICMAE)* **2019**.
22. Cordeiro, R. A., Azinheira, J. R., Moutinho, A. Robustness of incremental backstepping flight controllers: The boeing 747 case study. *EEE Transactions on Aerospace and Electronic Systems* **2021**, *57(5)*.
23. Tal, E., Karaman, S. Accurate tracking of aggressive quadrotor trajectories using incremental nonlinear dynamic inversion and differential flatness. *IEEE Transactions on Control Systems Technology* **2020**, *29(3)*.
24. Jayaraman, B., Giri, D. K., Ghosh, A. K. Attitude Tracking Control of a Light Aircraft Using Classical and Incremental Nonlinear Dynamic Inversion Approaches. *2021 International Conference on Unmanned Aircraft Systems (ICUAS)* **2021**.
25. Smeur, E. J., Bronz, M., de Croon, G. C. Incremental control and guidance of hybrid aircraft applied to a tailsitter unmanned air vehicle. *Journal of Guidance, Control, and Dynamics* **2020**, *43(2)*.
26. Tal, E. A., Karaman, S. Global trajectory-tracking control for a tailsitter flying wing in agile uncoordinated flight. *In AIAA Aviation 2021 Forum* **2021**.
27. Tal, E., Karaman, S. Global incremental flight control for agile maneuvering of a tailsitter flying wing. *Journal of Guidance, Control, and Dynamics* **2022**, *45(2)*.
28. Yang, Y., Zhu, J., Yang, J. INDI-based transitional flight control and stability analysis of a tail-sitter UAV. *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* **2020**.
29. Chiappinelli, R., Nahon, M. Modeling and Control of a Tailsitter UAV. *2018 International Conference on Unmanned Aircraft Systems (ICUAS)* **2018**.
30. E-Flite XVERT VTOL webpage. Available online: https://www.horizonhobby.de/en_DE/product/x-vert-vtol-bnf-basic-504mm/EFL1850.html (accessed on 24 January 2024).
31. Beard, Randal W.; McLain, Timothy W. *Small Unmanned Aircraft: Theory and Practice*, 1st ed.; Publisher: Princeton University Press, 2018.
32. Stevens, Brian L; Lewis, Frank L.; *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3 ed.; Publisher: John Wiley & Sons, 2016.
33. Nelson, Robert C. *Flight Stability and Automatic Control*, 2nd ed.; Publisher: McGraw-Hill, 1998;.
34. Khan, W., Nahon, M. Toward an accurate physics-based UAV thruster model. *IEEE/ASME Transactions on Mechatronics*, **2013**, *18(4)*.
35. McCormick, Barnes W. *Aerodynamics, Aeronautics and Flight Mechanics*, 2nd ed.; Publisher: John Wiley & Sons, 1979;.
36. Khan, W., Nahon, M. Real-time modeling of agile fixed-wing UAV aerodynamics. *2015 international conference on unmanned aircraft systems (ICUAS)* **2015**.
37. Khan, W., Nahon, M. Modeling dynamics of agile fixed-wing UAVs for real-time applications. *2016 international conference on unmanned aircraft systems (ICUAS)* **2016**.
38. Prisacariu, V., Cîrciu, I., Boscoianu, M. Flying wing aerodynamic analysis. *Review of the Air Force Academy* 2012, *(2)*.
39. Esteves, D. J., Moutinho, A., Azinheira, J. R. Stabilization and altitude control of an indoor low-cost quadrotor: design and experimental results. *2015 IEEE International Conference on Autonomous Robot Systems and Competitions* **2015**.

40. Madgwick, S. O., Harrison, A. J., Vaidyanathan, R. Estimation of IMU and MARG orientation using a gradient descent algorithm. *2011 IEEE international conference on rehabilitation robotics* **2011**.
41. Acquatella, P., van Ekeren, W., Chu, Q. P. Pi (d) tuning for flight control systems via incremental nonlinear dynamic inversion. *IFAC-PapersOnLine* **2017**, *50(1)*.
42. Cordeiro, R. A., Azinheira, J. R., Moutinho, A. Cascaded incremental backstepping controller for the attitude tracking of fixed-wing aircraft. *5th CEAS Conference on Guidance, Navigation and Control.* **2019**.
43. Cordeiro, R. A., Marton, A. S., Azinheira, J. R., Carvalho, J. R., Moutinho, A. Increased robustness to delay in incremental controllers using input scaling gain. *IEEE Transactions on Aerospace and Electronic Systems* **2021**, *58(2)*.
44. Arduino Nano 33 IOT product page. Available online: https://docs.arduino.cc/hardware/nano-33-iot/ (accessed on 24 January 2024).
45. Wiznet W5500 product page. Available online: https://www.wiznet.io/product-item/w5500/ (accessed on 24 January 2024).
46. CAD model *UAV Tailsitter* at GrabCAD website. Available online: https://grabcad.com/library/uav-tailsitter-1 (accessed on 24 January 2024).
47. Spektrum A220 4g Aircraft Servo product page, Horizon Hobby website. Available online: https://www.horizonhobby.com/product/a220-4g-aircraft-servo-x-vert/SPMSA220.html (accessed on 24 January 2024).
48. 20A Nano ESC product page, SkyRC website. Available online: https://www.skyrc.com/Discontinued_Products/20A_ESC (accessed on 24 January 2024).
49. BL280 Brushless Outrunner Motor 2600Kv product page, Horizon Hobby website. Available online: https://www.horizonhobby.com/product/bl280-brushless-outrunner-motor-2600kv/EFLM1809.html (accessed on 24 January 2024).
50. Mini Power Hub product page, MatekSys website. Available online: http://www.mateksys.com/?portfolio=hub5v12v#tab-id-1 (accessed on 24 January 2024).
51. Gens ace G-Tech Soaring 450mAh 7.4V battery product page, Gens Ace website. Available online: https://www.gensace.de/gens-ace-g-tech-soaring-450mah-7-4v-30c-2s1p-lipo-battery-pack-with-jst-syp-plug.html (accessed on 24 January 2024).
52. Qualysis Track Manager software, Qualysis website. Available online: https://www.qualisys.com/software/qualisys-track-manager/ (accessed on 24 January 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.