An Enhanced Incremental Nonlinear Dynamic Inversion Control Strategy for Advanced Unmanned Aircraft Systems

Maryam Taherinezhad * and Alejandro Ramirez-Serrano

Department of Mechanical Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada; aramirez@ucalgary.ca
* Correspondence: maryam.taherinezhad@ucalgary.ca

Abstract: A cascade control architecture was proposed for the control of advanced unmanned aerial vehicles targeted for operation inside confined hazardous spaces. The aircraft of interest is a highly maneuverable system presenting highly coupled dynamics, enabling it to perform unique flight maneuvers that no other aircraft can. To deal with the complexities inherent to such an aircraft, a novel nonlinear control architecture was proposed. Two outer loop NDI controllers and an inner loop INDI controller were employed for position/velocity and attitude/rotation rate control, respectively. These controllers assist each other, enabling them to decouple the aircraft dynamics while coping with complex external disturbances. The obtained results demonstrate an effective control of the targeted aircraft system to accurately track complex flight maneuvers.

Keywords: Tiltrotor UAV; INDI; trajectory tracking; Navig8 UAV

1. Introduction

Urban search and rescue (USAR) robots and drones have the potential to save lives and hold the promise to limit the monetary consequences of disasters. Specifically, numerous conferences and journal articles have highlighted the potential benefits of unmanned aerial vehicles (UAVs) for SAR operations, as well as the need for their further development. To reduce the number of casualties from urban disasters (among other things), new aircraft configurations and advanced control systems are needed to enable UAVs to fly and maneuver inside helicopter-impenetrable environments and cope with the associated aerodynamic disturbances that the aircraft might encounter inside hazardous spaces. This paper provides novel theoretical and practical control formulations for the next phase of UAV evolution.

To enable robots to navigate inside confined environments, there is also a need to increase their motion capabilities so that robots can penetrate and maneuver through geometrically complex 3D spaces (e.g., voids and crevices). Examples of suitable motions include pitch hover maneuvers and aerial manipulation. Aircraft having such capabilities would be beneficial to numerous other applications, including mining, construction, forestry (flying within tree-covered areas), and inspection tasks inside critical infrastructures. Although UAVs have been the subject of numerous developments, the biggest limitation to the widespread deployment of such systems for complex operations in restricted environments is their limited agility to move inside confined spaces while coping with the associated aerodynamic ground and wall effects. Therefore, improvements are needed to control new UAV designs formulated for complex autonomous maneuvers in tight spaces with high speed, complex trajectory tracking, and other aspects.
2. Literature Review

This paper presents a novel control architecture towards developing highly maneuverable UAVs, enabling them to perform agile maneuvers inside restricted GPS-denied spaces in the presence of a priori unknown disturbances and coupled aircraft dynamics. Due to the diverse applications of UAVs in a wide variety of areas, such as surveillance, environmental monitoring, drug delivery, and SAR, there has been a substantial body of work developing diverse control systems for the position and orientation (a.k.a. pose) of UAVs, as well as their trajectory and resistance to internal and external disturbances. Although such aspects have been extensively considered in past works, such mechanisms have mainly been developed for flying in open uncluttered spaces which lack effectiveness inside the type of spaces that have been considered in this paper. For such unrestricted (open) spaces, different control methods have been developed, from classical linear \[1–3\] to adaptive \[4\], fuzzy \[5\], and multi-variable nonlinear control \[6\] approaches. These control methods have been mostly applied to typical rotorcraft (e.g., helicopters and quad-rotors) and fixed-wing type aerial vehicles, with very little attention to new aircraft concepts. Such linear and nonlinear control methods have also employed numerous complementary techniques, including tools from the area of artificial intelligence (AI).

Due to the fact that aircraft systems (especially those targeted to fly inside cluttered spaces) are affected by diverse aspects (e.g., propeller performance, changing atmospheric conditions, drag, etc.), many of which change over time and/or are based on the aircraft’s motion, obtaining an exact mathematical model of aircraft for all flying conditions is practically impossible. Under such conditions, and knowing that most aircraft models are nonlinear, researchers have employed diverse processes to improve the aircraft representation while simultaneously enabling the model to facilitate for control development using model simplification techniques \[7\]. Generating linear models based on the corresponding nonlinear models has enabled the use of simpler linear control techniques (e.g., PID, $\text{H}_\infty$, and LQR). In the study published by the authors of \[8\], a feed-forward control algorithm that led to a Lyapunov function with asymptotic stability was developed. In the study published by the authors of \[7\], Funtoni et al. proposed a linearization-based technique to enable effective control of under-actuated mechanical systems in cases where typical control mechanisms could not effectively be used, or when typical controllers (e.g., PID) severely limit the operational capabilities of aircraft restricted to fly under benign wind conditions).

Despite the success of previous controllers in controlling aircraft under typical flight missions (cruise, take-off, hover, etc.), such controllers have been proved to be ineffective when controlling UAVs performing untypical or aggressive flight maneuvers. Examples of such maneuvers include acrobatic flight maneuvers, such as hammerheads and lomce-vaks (a family of extreme acrobatic maneuvers where the aircraft, with almost no forward speed, rotates on a chosen axes using the gyroscopic precession and torque of the rotating propellers). Furthermore, controlling aircraft under unexpected or changing flight conditions, such as flying under system failure and landing on an oscillatory moving ship deck (a common problem faced by military and maritime SAR operations), have been proved to be difficult. In the study published by the authors of \[9\], Marconi et al. proposed a control algorithm for landing VTOL (vertical take-off and landing) aircraft on an oscillatory deck ship using an internal model-based feedback dynamic regulator. Such a controller proved to be robust in the presence of uncertainties affecting the system; however, the controller was only analyzed in open spaces, with the aircraft performing typical flight maneuvers. Similarly, in trying to resolve such problems, other mechanisms have been proposed e.g., those published by the authors of \[9–11\]. Olfati presented an algorithm with a smooth static state feedback approach for stabilizing a VTOL with a complex input \[10\]. A simple Lyapunov analysis-based controller for a VTOL limited to perform simple maneuvers, such as take-off, hover, linear flight trajectories, and landing, was presented and assessed by the authors of \[11\].
In the study published by the authors of [5], Coza et al. provided a summary of UAV stabilization techniques aiming to identify effective tools to solve the problems of controlling UAVs having model uncertainties—a problem none of the previous developments could fully resolve. The observation of such a review led to the development of new adaptive fuzzy methodologies to control quadrotor helicopters in the presence of both model uncertainty and sinusoidal wind disturbances without the need for a (precise) mathematical formulation of the aircraft e.g., those published by the authors of [5,12]. In the study published by the authors of [13], an adaptive nonlinear robust controller was designed to enable UAVs to follow desired tracking missions. Although such approaches have reduced the complexities associated with creating precise aircraft mathematical models, they require the use of arduous methods to find their needed control parameters (e.g., drag coefficients), requiring the control engineer to be highly knowledgeable in numerous areas outside the control domain, including computational fluid dynamics, engine/motor characterization, etc. To reduce such complexities, engineers have used fewer formal mechanisms to model the behaviour of aircraft, including AI methodologies. Similar to the study published by the authors of [5], an adaptive neural network control method was presented in the study published by the authors of [12] to stabilize the attitude of quadrotor helicopters in the presence of model uncertainty and considerable wind disturbances. Morel and Leonessa [14] proposed a direct adaptive controller for altitude and Euler angels tracking of a quadrotor UAV in the presence of model uncertainties. Amiri et al. [15] proposed an integral backstepping control technique to control both the position and orientation of a highly maneuverable UAV, improving the stability of the vehicle, but no aerodynamic or other disturbance rejection was considered. A backstepping control approach combined with two neural networks to approximate aerodynamic uncertainties was used by the authors of [16]. Xu and Ozquner proposed a combination of PID control for the fully actuated part of the dynamic model of UAVs, with a sliding mode controller for the underactuated part [17].

In the study published by the authors of [18], a sliding mode disturbance observer was presented as a robust flight controller of a quadrotor, which provided robustness to external disturbances, uncertainties of the dynamic model, and actuator failure. Similar to the study published by the authors of [18], researchers have proposed other control mechanisms to increase flight performance using robust adaptive fuzzy controllers, e.g., in the study published by the authors of [19], which have demonstrated satisfactory results against sinusoidal and other types of wind disturbances. Mokhtari et al. presented a robust feedback linearization method with a linear generalized H∞ controller with a weighting function that resulted in an improved overall robustness against disturbances and uncertainties [20]. Dunfied et al. proposed the use of a pre-trained neural network to stabilize quadrotor UAVs while hovering without disturbances [21]. In the work reported by the authors of [22,23], the capabilities of adaptive neural network controllers in VTOL aircraft stabilization were shown. Although promising and somewhat effective, most of the mentioned control developments have focused on the control of traditional unmanned aircraft (e.g., quadrotors, helicopters, and fixed-wing aircraft) executing somewhat simple flight maneuvers under the presence of simple external disturbances modelled as impulse and sinusoidal signals. Despite the satisfactory results of previous works, such approaches have not considered the control of new aircraft concepts that are currently being developed (e.g., supersonic and hybrid or transitional UAVs) [24–27]. As a result, previously developed controllers provide somewhat limited control capabilities when applied to unconventional (complex) aircraft. Such control formulations have challenges to enable aircraft to perform complex (e.g., acrobatic) and stable maneuvers in the presence of rapidly changing disturbances, which are typically found when navigating inside confined spaces or under extreme weather conditions.

Based on the proliferation in the development of advanced unmanned aircraft targeted to be deployed in hazardous environments, there is a need to reduce the depend-
ency on having accurate aircraft models required for control. Sensor-based control approaches have been formulated that are capable of dealing with model uncertainties. In such techniques, accurate knowledge of the system is exchanged with precise sensing of the system’s dynamics, decreasing system identification efforts. A promising sensor-based strategy is the use of incremental dynamics (ID) techniques, which ease the generation of controllers that are less dependent on the system model e.g., that proposed by the authors of [28]. The incremental nonlinear dynamic inversion (INDI) method, in particular, combined with a set of PID/PD controllers, has been successfully employed in the literature to control the attitude and position of advanced tilt-rotor bi-rotor drones, which cannot be fully controlled via traditional control formulations [28].

Although INDI has shown great promise for the control of novel (non-typical) aircraft subject to disturbances, INDI control remains sensitive to several model uncertainties. To deal with such characteristics, the original INDI control formulation has been combined with AI paradigms, such as artificial neural networks, and other mechanisms. The authors of [28] proposed a novel INDI with neural networks (INDI-NNs) control scheme to deal with inherent system model uncertainties. Numerous INDI control schemes have been developed with diverse objectives in mind, including position, attitude, and trajectory tracking for diverse UAVs. In the study published by the authors of [29], an INDI controller for the 3D acceleration trajectory tracking of tail-sitter UAVs (capable of transitioning between VTOL and fixed-wing flight) was developed. The results presented by the authors of [30] show that INDI control is effective to control aircraft operating under diverse flight modes (e.g., VTOL and fixed wing), which is not possible, or at least very challenging, with other control mechanisms. By adding a sideslip controller to align the wings of a given aircraft with the sensed airspeed vector, this has provided the means to neglect the flight phase that the UAV is executing. Following such advances, Wang et al. proposed an incremental sliding mode control mechanism driven by a sliding mode disturbance observer (INDI-SMC/SMDO) based on the control structure of INDI. This developed controller has been shown to have the ability to passively resist a wider variety of faults and external disturbances using continuous inputs with lower control and observer gains [31].

Based on the cited works, INDI control formulations seem to be the most appropriate mechanism developed, thus far, for the control of advanced unmanned aircraft systems, especially those aircraft having coupled dynamics capable to execute complex flight maneuvers. When compared to other control formulations for either linear or nonlinear systems, the INDI control method is less dependent and less sensitive to model dynamics and model uncertainties, and therefore increasing both system robustness and flexibility with respect to changes in the aircraft’s operation mode. Based on previous work conducted by our research group (e.g., the study published by the authors of [28]), an extension of previous INDI formulations for the control of highly maneuverable VTOL UAVs has been proposed in this paper to enhance the operation of such an aircraft targeted to operate inside GPS-denied confined spaces subject to aerodynamic ground and wall effects. This proposed control method decouples the control parameters, which reduces the time to compute a proper control signal and hence improves the performance of the aircraft. The controller was shown to be robust to external disturbances, capable of independent control of the system’s coupled degrees of freedom, and convenient for independent position/orientation trajectory tracking subject to motion and state constraints.

The following sections of this paper are organized as follows: the nonlinear mathematical model of the aircraft of interest (termed the Navig8) is described in Section 3; the proposed dynamic inversion control method is described in Section 4, while Section 5 presents illustrative numerical simulations showing the performance of the proposed controller, followed by the conclusions in Section 6.
3. Aircraft’s Dynamic Model

3.1. Flight Mechanism

The drone under consideration in this paper is a six degrees of freedom (DOF) tilt rotorcraft. Each of the aircraft’s ducted or shrouded propellers which rotate in opposite directions have independent tilt angles, speed, and variable pitch angles. The aircraft also has a variable pitch propeller (VPP) tail rotor, which, in contrast to helicopters, produces thrust perpendicular to the horizontal reference plane of the aircraft, enabling the aircraft to perform pitch maneuvers (Figure 1).

The thrust produced by each of the three propellers can be independently controlled in three possible ways: i) by changing the propeller’s rotational speed \( \omega_1, \omega_2, \text{and} \omega_3 \), ii) by changing the propeller’s angle of attack \( \beta_1, \beta_2, \text{and} \beta_3 \), and iii) a combination of “i” and “ii” (a combination of propeller rpm and VPP changes) via the control signals \( \alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \text{and} \beta_3 \), which correspond to the propellers’ speed and the angle of attack of the left, tail, and right propellers, respectively. Additionally, each of the two main (left and right) propellers can independently tilt by an angle \( \alpha_1 \), providing the capability for thrust vectoring. These characteristics result in the use of eight control signals \( \omega_1, \alpha_1, \text{and} \beta_1 \), which provides the opportunity to control the aircraft in different ways via proper control allocation mechanisms. Optimal control allocation would also enable the aircraft to cope with internal failures (e.g., damage of the rpm control) under a high reaction time in diverse ways. Due to the control allocation challenges that are available for the control of such an UAV, in this paper, we only considered controlling the thrust produced by each of the UAV’s propellers via option “i” (i.e., keeping all \( \beta_i \) parameters unchanged throughout the given mission). Controlling the UAV via options “ii” and “iii” will be the subject of a subsequent paper focusing on the corresponding control allocation formulations needed to determine the optimal use of the propellers’ rpm, VPP, or a combination of both. As a result of the reduced control complexities employed in this paper, the UAV was left with five control inputs: \( \omega_1, \omega_2, \omega_3, \alpha_1, \text{and} \alpha_3 \). In order to minimize the adverse aerodynamic ground and wall effects, which the UAV would experience when flying close to surfaces (e.g., ground), the two main propellers were set to have a constant dihedral angle, \( \gamma \), (see Figure 2). Such a dihedral angle was determined from extensive CFD (computational fluid dynamics) analysis [32].

![Figure 1. Schematic diagram of the Navig8 drone.](image)
The combination of the above-mentioned characteristics enables the aircraft to perform untypical flight maneuvers that other aircraft cannot execute, including pitch hover at angles within the range from $-90^\circ$ (nose down) to $90^\circ$ (nose up). With effective control and mechanical design changes, these flight characteristics can be augmented to perform enhanced maneuvers, such as perching and inverted pitch hover maneuvers suitable for time-extended surveillance, as well as other critical missions inside dynamic cluttered spaces.

The drone, having a mass “m”, can fly sideways, forwards, and backwards in diverse attitudes using the five control inputs that were considered in this paper. The sideways motion (in the vehicle’s “y” direction—see Figure 2) was achieved through generating a differential thrust/speed of the two (left and right) main (shrouded) propellers. The tilt angles, $\alpha_1$ and $\alpha_3$, of such propellers were changed in equal amounts to achieve fore and aft motions (i.e., motion in the “x” direction). Tilting the angle of the propellers in opposite directions created the yaw motion (rotation about the vehicle’s “z” axis). The described motions are, however, highly coupled, that is, requesting the aircraft to execute a given maneuver produces a side effect (sometimes undesirable) motion. Such coupled motions have been summarized in Table 1, where the number of Xs in each box denotes the degree of the corresponding coupled motions. For example, if the aircraft has been requested to execute a pure sideways (translational) motion, the differential rotational speeds of the main propellers needed to achieve such a motion also generate roll and yaw aircraft motions. Such coupled motions and the number of simultaneous motion(s) associated with the given (desired) motion have been described via the drone’s mathematical model.

**Table 1. Motions and respective coupling effects.**

<table>
<thead>
<tr>
<th>Motion</th>
<th>Surge</th>
<th>Sway</th>
<th>Heave</th>
<th>Roll</th>
<th>Pitch</th>
<th>Yaw</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surge (front/back)</td>
<td>-</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sway (side-to-side)</td>
<td>-</td>
<td>XX</td>
<td></td>
<td>XX</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Heave (up/down)</td>
<td>X</td>
<td>-</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Roll</td>
<td>-</td>
<td>X</td>
<td>XXX</td>
<td>X</td>
<td>XXX</td>
<td>X</td>
</tr>
<tr>
<td>Pitch</td>
<td>XXX</td>
<td>XX</td>
<td>-</td>
<td>XX</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.2. Dynamic Model

The Newton–Euler formulation was used to derive the system’s dynamic model. Two frames of reference were used: an inertial reference frame, denoted by \( OXYZ \), and a body reference frame positioned at the center of mass of the UAV, denoted by \( OX'Y'Z' \), as shown in Figure 2.

In order to describe the rotation of the drone, the Euler angles (roll (\( \phi \)), pitch (\( \theta \)), and yaw (\( \psi \))) were used and, therefore, the transformation matrix from the inertial to the body reference frame was defined by Equation (1) as:

\[
R^B_T = \begin{bmatrix}
   c\theta c\psi & c\theta s\psi & -s\theta \\
   -c\phi s\psi + s\phi c\theta c\psi & c\phi c\psi + s\phi c\theta s\psi & c\phi s\theta \\
   s\phi s\psi + c\phi c\theta c\psi & -s\phi c\psi + c\phi c\theta s\psi & c\phi c\theta
\end{bmatrix}
\]  

(1)

where \( c \) and \( s \) represent the cos and sin functions of the corresponding angle (e.g., \( c\phi = \cos \phi \)), respectively. The matrix describing the relationship between the body’s angular velocity (\( p \), \( q \), or \( r \)) and the rotational speed of the Euler angles was defined as per Equation (2):

\[
\omega = \begin{bmatrix}
   \dot{\phi} \\
   \dot{\theta} \\
   \dot{\psi}
\end{bmatrix} = \begin{bmatrix}
   1 & \tan \theta \sin \phi & \tan \theta \cos \phi \\
   0 & \cos \phi & -\sin \phi \\
   0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta
\end{bmatrix} \begin{bmatrix}
   p \\
   q \\
   r
\end{bmatrix} = J(\theta) \dot{\Omega}
\]  

(2)

Due to the complexities related to the operational capabilities of the Navig8 UAV, the following three assumptions (\( Ak \)) and two constraints (\( Ck \)) were made to simplify the drone’s equations of motion:

(C1) The body reference frame and the aircraft center of mass are coincident.
(C2) The aircraft’s structure is symmetric to its XZ (longitudinal) plane, resulting in \( I_{yy} = I_{yz} = 0 \).
(A1) All aerodynamic coefficients, such as drag and lift, remain constant throughout the given flight mission.
(A2) The thrust, \( T_i \), generated by propeller \( I \), is proportional to the square of its rotational speed.
(A3) The thrust and torque coefficients for each of the three propellers (\( i = 1, 2, \text{ or } 3 \)) is assumed constant, as described by Equations (3) and (4).

\[
T_i = \begin{bmatrix}
   0 \\
   0 \\
   k_{T_i} \omega_i^2
\end{bmatrix}^T
\]  

(3)

\[
\tau_i = \begin{bmatrix}
   0 \\
   0 \\
   K_{T_0i} \omega_i^2
\end{bmatrix}^T
\]  

(4)

and

\[
k_{T_i} = \frac{1}{2} C_T \rho A r_i^2
\]

\[
K_{T_0i} = \frac{1}{2} C_{T_0} \rho A r_i^3
\]

Where \( C_T \) and \( C_{T_0} \) represent the propellers’ thrust and torque coefficients, \( r \) and \( A \) specify the radius and footprint area of each propeller, and \( \rho \) denotes the air density.

Based on the previously mentioned aspects, the equations of the motion for the Navig8 can be expressed as Equation (5).
The terms $X$, $Y$, $Z$, $L$, $M$, and $N$ in Equation (5) denote the total forces and torques in the $x$, $y$, and $z$ directions, respectively. Such terms were obtained as per Equation (6):

$$
\begin{bmatrix}
L \\
M \\
N \\
X \\
Z
\end{bmatrix} =
\begin{bmatrix}
k_{T1}b \cos \gamma & -k_{T01} & -k_{T3}b \cos \gamma & k_{T03} & 0 \\
-k_{T1}a \cos \gamma & -k_{T1}c & -k_{T3}a \cos \gamma & -k_{T3}c & k_{T2}d \\
k_{T01} \cos \gamma & k_{T1}b & -k_{T03} \cos \gamma & -k_{T3}b & k_{T02} \\
k_{T1} \cos \gamma & 0 & -k_{T1} & 0 & k_{T2} \\
k_{T1} \cos \gamma & 0 & k_{T3} \cos \gamma & 0 & k_{T2}
\end{bmatrix}
\begin{bmatrix}
\omega_1^2 \cos \alpha_1 \\
\omega_1^2 \sin \alpha_1 \\
\omega_2^2 \cos \alpha_3 \\
\omega_2^2 \sin \alpha_3 \\
\omega_2^2
\end{bmatrix}
$$

(6)

where $a$, $b$, $c$, $d$, and $e$ represent the corresponding orthogonal distances between the center of mass of the UAV and the left, right, and back propellers described by the vectors $p_1 = (a, b, c), p_2 = (-d, 0, -e)$, and $p_3 = (a, -b, c)$, respectively, as illustrated in Figure 2. The five expressions provided in Equation (6), $X$, $Z$, $L$, $M$, and $N$, define a coupled set of equations, which are simplified by defining the five control variables $u_1, u_2, u_3, u_4$, and $u_5$, as per Equation (7).

$$
U =
\begin{bmatrix}
u_1 \\
u_2 \\
u_3 \\
u_4 \\
u_5
\end{bmatrix} =
\begin{bmatrix}
\omega_1^2 \cos \alpha_1 \\
\omega_1^2 \sin \alpha_1 \\
\omega_2^2 \cos \alpha_3 \\
\omega_2^2 \sin \alpha_3 \\
\omega_2^2
\end{bmatrix}
$$

(7)

It is worthwhile to clarify that based on the assumptions and the drone’s symmetrical geometry, the expression for $Y$ is not involved in Equation (6). As a result, Equation (6) can be rewritten as Equation (8):

$$
\begin{bmatrix}
L \\
M \\
N \\
X \\
Z
\end{bmatrix} =
\begin{bmatrix}
k_{T1}b \cos \gamma & -k_{T01} & -k_{T3}b \cos \gamma & k_{T03} & 0 \\
-k_{T1}a \cos \gamma & -k_{T1}c & -k_{T3}a \cos \gamma & -k_{T3}c & k_{T2}d \\
k_{T01} \cos \gamma & k_{T1}b & -k_{T03} \cos \gamma & -k_{T3}b & k_{T02} \\
k_{T1} \cos \gamma & 0 & -k_{T1} & 0 & k_{T2} \\
k_{T1} \cos \gamma & 0 & k_{T3} \cos \gamma & 0 & k_{T2}
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
u_3 \\
u_4 \\
u_5
\end{bmatrix} = KU
$$

(8)

in which $K$ is a fixed matrix defined by the physical characteristics of the UAV, as per Equation (8). Therefore, the control signals vector, $U$, can be calculated if the inverse of $K$ exists, as represented by Equation (9) below.

$$
U = [U_1] = K^{-1}
$$

(9)
Therefore, considering Equation (7), the five control inputs of interest are computed per Equation (10) as:

\[
\begin{bmatrix}
    \omega_1^2 \\
    \omega_2^2 \\
    \omega_3^2 \\
    \alpha_1 \\
    \alpha_3
\end{bmatrix} = K^{-1} \begin{bmatrix}
    u_1^2 + u_2^2 \\
    u_3^2 + u_4^2 \\
    u_5 \\
    \tan^{-1}\frac{u_2}{u_1} \\
    \tan^{-1}\frac{u_4}{u_3}
\end{bmatrix}
\] (10)

4. Control Technique

The UAV’s dynamic model can be represented in generic form of a nonlinear state-space system:

\[
\dot{x} = F(x, u)
\] (11)

where \( x_t \) and \( u_t \) represent the state and input transpose vectors, at time \( t \), respectively:

\[
x_t = [x, y, z, \phi, \theta, \psi, \dot{x}, \dot{y}, \dot{z}, \dot{\phi}, \dot{\theta}, \dot{\psi}]'
\]

\[
u_t = [\omega_1, \omega_2, \omega_3, \alpha_1, \alpha_3, \beta_1, \beta_2, \beta_3]'
\]

By calculating the first-order Taylor series of Equation (11), Equation (11) can be written as:

\[
\dot{x} = \dot{x}_0 + \frac{\partial F}{\partial x} |_{x_0, u_0} (x - x_0) + \frac{\partial F}{\partial u} |_{x_0, u_0} (u - u_0)
\] (12)

in which \( x_0 \) and \( u_0 \) represent the state and input vectors of the system at the previous time step, \( (t_0) \), respectively. An INDI description of a system described by Equation (12) is possible under the following two assumptions:

(a) The state of the UAV and input, \( u \), are bounded, and the function \( F \) is continuous.
(b) The amount of time that has passed between \( x_0 \) and \( x \) (sampling time) is sufficiently small.

By using a small sampling time (assumption b) and employing high performance actuators in the drone, the changes in the state of the UAV \( (\Delta x) \) with the given time step can be considered negligible (i.e., \( \Delta x = x - x_0 \)) with respect to the large changes in the input parameters represented as \( (\Delta u = u - u_0) \), given that state changes arise as a result of the integration of input changes and are, therefore, slower.

The assumptions listed above imply that the difference \( (x - x_0) \) can be considered negligible (ignored) and therefore, Equation (12) becomes:

\[
\dot{x} = \dot{x}_0 + \frac{\partial F}{\partial x} |_{x_0, u_0} (x - x_0) + \frac{\partial F}{\partial u} |_{x_0, u_0} (u - u_0)
\] (13)

To be efficient, this INDI approach requires a suitable timescale separation, which was considered as 0.01 s in this paper. To achieve the required timescale separation, the approach taken in this paper was to employ a combination of NDI, INDI, and PID controllers, forming an inner and outer loop control strategy, as illustrated in Figure 3.
Figure 3. The proposed cascade NDI/INDI controller architecture.

The developed control architecture can be applied to diverse aircraft systems, especially those with complex dynamics operating in the presence of internal (e.g., system failures) and/or external disturbances (e.g., wind gusts). Overall, the controller would receive, as a control input, a desired path/trajectory to follow, which includes the desired position \((x_d, y_d, z_d)\), orientation \((\phi_d, \theta_d, \psi_d)\), and time conditions, defining the UAV’s maneuvers from where the required control signals will be computed.

4.1. The Attitude Controller (INDI)

For the INDI attitude controller, the state of the drone was defined as a subset of the state variables used in Equation (11). Such a subset, \(\bar{x} \in x\), comprises the roll, pitch, and...
yaw Euler angles and the corresponding rotational speeds \((p, q, \text{ and } r)\), as described by Equation (14), where \(\vartheta = [\phi, \theta, \psi]^T, \quad \Omega = [p, q, r]^T, \) and \(\vec{u} = [u_x, u_y, u_z]^T\) have been defined based on Equations (8–9) and the dynamic equations of the system (e.g., Equation (5)):

\[
\ddot{\vartheta} = \begin{bmatrix} \frac{\partial}{\partial \vartheta} \end{bmatrix} \frac{\partial}{\partial \vartheta} = \begin{bmatrix} \frac{\partial}{\partial \vartheta} \end{bmatrix} \frac{\partial}{\partial \vartheta} = \begin{bmatrix} \frac{\partial}{\partial \vartheta} \end{bmatrix} \frac{\partial}{\partial \vartheta}
\]

(14)

It is important to note that the lateral (sway) motion of the UAV is highly coupled with the roll angle of the body (see Table 1). The result of such coupled motions is that the UAV cannot perform pure lateral movement without rolling and vice versa. The additional coupled effects, as represented in Table 1, are manageable and can be decoupled and be precisely controlled. Therefore, for the proposes of this paper, the roll angle was not considered as a controllable state, but it was used as an input to reach a desired lateral translation. From Equation (14), one can map the state vector, \(\ddot{\vartheta}\), to the Euler angles of the UAV, \(\vartheta\), via a transformation matrix \(H\), as per Equation (15):

\[
\vartheta = H\ddot{\vartheta} = \begin{bmatrix} I_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \ddot{\vartheta}
\]

(15)

Since the first-order time derivative of the control variable vector, \(\dot{\vartheta} = [\phi, \theta, \psi]^T\), is not directly related to the control input vector, \(\vec{u}\), the relationship between the input and output variables was determined through obtaining the second-order derivative of the control vector, \(\vartheta\), as shown in Equations (16) and (17):

\[
\dot{\vartheta} = \frac{d\vartheta}{dt} = \frac{dH\ddot{\vartheta}}{dt} = H\dot{\vartheta} = H\begin{bmatrix} \omega \\ \omega \end{bmatrix} = J(\vartheta)\Omega
\]

(16)

\[
\dot{\vartheta} = \frac{d^2\vartheta}{dt^2} = \frac{d(J(\vartheta)\Omega)}{dt} = \frac{d}{d\ddot{x}} \begin{bmatrix} 1 & \tan \theta \sin \phi \\ 0 & \cos \phi \end{bmatrix} \begin{bmatrix} \tan \phi \cos \phi \\ - \sin \phi \end{bmatrix} \vartheta
\]

(17)

Considering Equation (17) and the dynamic equations of the UAV, Eqns. (5–7), one can define \(\ddot{\vartheta}\) as a function of \(\ddot{x}\) and \(\vec{u}\), as shown in Equation (18). From this formulation, and following Equation (11), the INDI controller can be formulated by performing a Taylor series expansion in the current time step, \(t\), as described in Equation (19):

\[
\ddot{\vartheta} = F_\vartheta(\ddot{x}, \vec{u})
\]

(18)

\[
\ddot{\vartheta} \approx F(\ddot{x}_0, \ddot{u}_0) + \frac{\partial F_\vartheta(\ddot{x}, \vec{u})}{\partial \ddot{x}} |_{\ddot{x}_0, \ddot{u}_0} (\ddot{x} - \ddot{x}_0) + \frac{\partial F_\vartheta(\ddot{x}, \vec{u})}{\partial \vec{u}} |_{\ddot{x}_0, \ddot{u}_0} (\ddot{u} - \ddot{u}_0)
\]

(19)

Using \(F(\ddot{x}_0, \ddot{u}_0) = \ddot{\vartheta}_0\) as the angular acceleration measurement of the UAV at the previous time step, \(t - 1\), a very short measurement sampling time (\(\Delta t\)) can be used, given that the actuator dynamics are faster (as per assumption b). Then, it is reasonable to further consider \((\ddot{x} - \ddot{x}_0) \approx 0\) when compared to the changes in the control output, \(\Delta \vec{u} = (\ddot{u} - \ddot{u}_0)\). In these circumstances, the simplified expression shown in Equation (20) can be used to obtain the control law:

\[
\ddot{\vartheta} = \ddot{\vartheta}_0 + K(\ddot{x}_0, \ddot{u}_0)\Delta \vec{u}
\]

(20)

Solving for \(\Delta \vec{u}\) from Equation (20) results in \(\Delta \vec{u} = K(\ddot{x}_0, \ddot{u}_0)^{-1}(\ddot{\vartheta} - \ddot{\vartheta}_0)\), where the matrix \(K(\ddot{x}_0, \ddot{u}_0)\) represents the partial derivative matrix of function \(F_\vartheta\) with respect to the control inputs of the system, \(\vec{u}\), resulting in:
With the above formulations, it is possible to effectively compute the control inputs (i.e., $L, M,$ and $N$) based on the vehicle’s angular acceleration measurements taken at time step $t - 1$. Thus, Equations (20) and (21) represent the attitude (orientation) INDI controller, which guides the UAV on how to attain the desired orientation at a given position $(x, y, or z)$ when following a desired path. As a result, the desired orientation behavior of the aircraft was not significantly affected when the slower dynamics of the model were ignored (which will be described in Section 6).

4.2. The Position Controller (NDI + INDI)

The controller designed to control the aircraft’s orientation (i.e., yellow boxes in Figure 3), provides the torques ($L, M,$ and $N$) required to orient the aircraft. Such torques can then be used in combination with the forces ($X$ and $Z$) needed to position the aircraft at a specific point $(x, y, or z)$ to generate the control signals needed to maneuver the UAV. As illustrated in Figure 3, a position controller must be formulated to provide such forces that will enable the UAV to track the desired path, which includes the position and orientation information. The aircraft’s translational movement control was achieved via a combination of an NDI and INDI (INDI + NDI) control strategy (blue boxes shown in Figure 3), where the NDI port is responsible for controlling the lateral motion, while the INDI port focuses on computing the $X$ and $Z$ forces that need to be applied to the aircraft through the control of the propeller’s thrust vectoring ability. Such a strategy enables to manage the coupled motions of the aircraft described in Table 1. Specifically, the proposed controller was targeted to reduce the coupled translational motions taking place in the $M − S$ (longitudinal) plane of the aircraft. That is, the position controller aimed to decouple the surge and heave motions of the aircraft. An NDI controller was employed for the lateral (sway) movement, and an INDI controller was developed for the movements in the $M − S$ directions (heave). For altitude regulation in the $x − z$ plane, the control inputs $u_4$ and $u_5$ were used. However, there was no direct control input within the vector $U$ (Equation (7)) for the side movement. Thus, a differential thrust/speed of the propellers was used to provide the required lateral motion. Therefore, finding the required roll angle, $\phi$, was required to generate the desired lateral motion in terms of the aircraft’s $y$ position and its rate of change, $\dot{y}$. In turn, the computed roll angle was provided as one of the three inputs for the attitude controller. To enable the NDI controller to cope with disturbances (in the $y$ direction), a sigma-pi neural network (NN) adaptive compensator was used to correct modeling errors and overcome the tendency of NDI controllers to be sensitive to the dynamic model’s accuracy [28,33].

The INDI position controller for the $x$ and $z$ aircraft motions was developed using a similar approach of the attitude controller (Section 4.1). For the position controller, however, a two-layer architecture was used. An INDI formulation was used to cancel the non-linearities associated with the linear dynamics of the system and generate the control signals for the $x$ and $z$ positions of the UAV. In addition, NDI was employed for the lateral (position) motion.

4.2.1. The Position INDI Controller

For the position INDI controller, a virtual control variable, $\hat{u}$, defined as per Equation (22), was used.
Considering the dynamic model of the system in the $x-z$ direction, Equation (23), and defining the states variables $x$ and $z$ as vector $\dot{x}$, the second-order time derivative of matrix $\ddot{x}$ for the INDI position controller was given as per Equation (25):

$$
\ddot{x} = \begin{bmatrix}
\begin{split}
X \\
Z
\end{split}
\end{bmatrix} = \begin{bmatrix}
0 & -g \\
\frac{1}{m}c\theta\psi & s\phi\psi + c\phi s\theta\psi & 0
\end{bmatrix} \begin{bmatrix}
X_0 \\
Z_0
\end{bmatrix}
$$

(23)

$$
\ddot{x} = \begin{bmatrix}
\begin{split}
X \\
Z
\end{split}
\end{bmatrix}
$$

(24)

$$
\ddot{x} \approx \ddot{x}_0 + \frac{\partial}{\partial \dot{u}} \left[ f_p(\ddot{x}, \dot{u}) \right]_{\ddot{x}=\ddot{x}_0, \dot{u}=\dot{u}_0} (\ddot{u} - \ddot{u}_0) = \ddot{x}_0 + \dot{G}(\ddot{x}_0, \ddot{u}_0)\Delta \ddot{u}
$$

(25)

in which $\dot{G}(\ddot{x}_0, \ddot{u}_0)$ is defined by Equation (26).

$$
\dot{G}(\ddot{x}_0, \ddot{u}_0) = \frac{\partial}{\partial \ddot{u}} \left[ f_p(\ddot{x}, \dot{u}) \right]_{\ddot{x}=\ddot{x}_0, \dot{u}=\dot{u}_0} = \begin{bmatrix}
\frac{1}{m}c\theta\psi & s\phi\psi + c\phi s\theta\psi
\end{bmatrix}
$$

(26)

$$
u = u_0 + \dot{G}(\ddot{x}_0, \ddot{u}_0)\Delta \ddot{u}
$$

(27)

4.2.2. The Position NDI Controller

As previously mentioned, due to the high coupling between the roll angle, $\phi$, and the position in the $y$ direction, it is not possible to independently control both parameters at the same time. Thus, a variable, $e$, (Equation (28)) was defined, which enables the design of an NDI controller to control the UAV’s lateral motion. The output of this controller block, $\phi_d$, was fed to the attitude control block, which will control the lateral motion of the drone.

$$
u = \sin \phi
$$

(28)

Following the process used in Section 4.1, the second-order time derivative of the side motion of the drone’s mathematical model, $y$, for the NDI position controller used the corresponding formulation in Equation (5), where the term $\sin \phi$ has been replaced with $\nu$, as shown in Equation (29):

$$
\ddot{y} = \frac{1}{m} \left( X_c \theta \psi + Z \left(-u_x \phi \psi + c\phi s\theta \psi \right) \right) = f(\theta) + g(\theta)\nu
$$

(29)

A virtual control input, $\nu$, can then be chosen as $\nu = \ddot{y}$ if $\det(\frac{\partial}{\partial \theta} g(\theta)) \neq 0$.

$$
u = g(\theta)^{-1}(\nu - f(\theta))
$$

(30)

$$
g(x, u) = -\frac{Z_0}{C\psi}
$$

(31)

An accurate description of the functions $f$ and $g$ within Equation (30) was required to cancel all model uncertainties and cross-couplings in the system. However, an exact cancelation of nonlinearities is practically impossible. Therefore, a sigma-pi neural network (NN) was used to compensate the uncertainties. Such a NN is represented as a NN adaptive compensator in Figure 3, which processes nine inputs (i.e., $x_0, y_0, z_0, x_d, Z_d, \phi, v_{adr}, X, \text{and } Z$) plus a set of three Kronecker values ($e_1, e_2, e_3$) (defined later in this document in Equation (37)), where $x_0, y_0$, and $z_0$ represent the position of
the UAV in the previous time step, and \( e = y_{rm} - y_0 \) defines the error between the reference model and the lateral position of the UAV in the previous time step. From the defined error and the Kronecker values, the sigma-pi NN generates \( v_{ad} \), which adaptively compensates the input produced via the NDI controller, which operates under the assumption that the mathematical model of the system is accurate.

As a result, the sigma-pi NN compensates the inversion error with a real-time pseudo control signal, \( v_{ad} \), where the pseudo control command was generated as per Equation (32):

\[
v = v_{rm} + v_{pd} - v_{ad}
\]

(32)

where \( v_{rm} = \ddot{y} \) is the reference model’s pseudo control, and \( v_{pd} \) is the signal generated via a PD linear regulator (see Section 4.3), which is used to regulate the dynamic response of the system as per Equation (33):

\[
v_{pd} = k_p(y_{rm} - y_0) + k_d(\dot{y}_{rm} - \dot{y}_0)
\]

(33)

As a result, the dynamics of the model tracking error can then be described via Equation (34):

\[
\dot{\varepsilon} = A\varepsilon + B(v_{ad} - \Delta)
\]

(34)

where \( \varepsilon = [\dot{y}_{rm} - \dot{y}, y_{rm} - y] \), \( A = \begin{bmatrix} 0 & 1 \\ -k_p & -k_d \end{bmatrix} \), and \( B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \).

4.2.3. The Sigma-Pi Neural Network

In comparison to other neural networks, the sigma-pi NN [33] uses summation and quadrature neurons in its hidden layers, providing the ability to preserve the network’s highly nonlinear mapping capability [34]. Figure 4 depicts the single layer sigma-pi NN used in the proposed UAV controller as having three inputs and one output.

![Figure 4. The structure of the online adaptive sigma-pi neural network [24].](image)

The neural network’s input/output map can be expressed as follows:

\[
v_{ad} = \sum_{j=1}^{k} w_j \beta_j = W^T \beta
\]

(35)

where \( W^T \) represents the weight coefficient matrix, and \( \beta \) denotes the basis function vector, which is defined as per Equation (36):

\[
\beta = kron(kron(c_1, c_2), c_3)
\]

(36)
The “\textit{kron}” denotes the Kronecker product operating on $c_1$, $c_2$, and $c_3$, which are defined as follows:

$$
\begin{align*}
  c_1 &= [0.1 \ e \ e^2]^T \\
  c_2 &= [0.01 \ x_d \ y_d \ z_d \ X \ Y \ Z]^T \\
  c_3 &= [x_0 \ y_0 \ z_0]^T
\end{align*}
$$

(37)

The corresponding weight error can then be expressed as follows:

$$
\tilde{W} = W - W^* 
$$

(38)

where $W^*$ denotes the ideal set of weights. Thus, the model tracking error (Equation (34)) was written as per Equation (39), and the update law for the NN was expressed via Equation (40).

$$
\dot{e} = Ae + B\tilde{W}^T\zeta + B(\tilde{W}^T\zeta - \Delta) 
$$

(39)

$$
\tilde{W}_j(t) = \begin{cases} 
-\Gamma_j^T(x^TP_jB)\zeta_j & \text{for } \|e_j\|_{P_j} > e_{0j} \\
0 & \text{for } \|e_j\|_{P_j} < e_{0j}
\end{cases} 
$$

(40)

The terms in Equation (40) are defined as $\|e_j\|_{P_j} = \sqrt{e_j^TP_je_j}$, $\Gamma_j$ is a positive constant adaptation gain (learning rate), and $P_j$ is a symmetric, positive definite matrix satisfying the Lyapunov equation $P_jA_j + A_j^TP_j = -I$ for the linear system error with NN weights. The Hurwitz matrix $A_j$ of Equation (37) ensures the existence of a unique $P_j = P_j^T > 0$, where the $P_j$ matrix used in the developed controller is described as per Equation (41):

$$
P_j = \begin{bmatrix}
  K_{d_j} & K_{p_j} & 1 \\
  2K_{d_j} & 2K_{d_j} + 1 & 2K_{p_j} \\
  2K_{d_j} & 2K_{d_j} & 2K_{p_j}
\end{bmatrix}
$$

(41)

4.3. The PD/PID Controller

The altitude and position controllers described above, Equations 20, 27, and 30, were designed based on the second derivative of the position and attitude states of the UAV, meaning that the controller has been developed to follow the second-order derivative of the desired trajectory (acceleration) of the aircraft. Therefore, in order to enable the UAV to follow the desired position/attitude (trajectory), three proportional derivative (PD) controllers for attitude (i.e., $PD_\phi$, $PD_\theta$, and $PD_\psi$) and three PID controllers (i.e., $PID_\phi$, $PID_\theta$, and $PID_\psi$) for position were used to map the pose error (the difference between the desired and measured poses in the previous time step) [35]. Similar to that conducted by the authors of [28], the PD/PID gains were optimized using a non-dominated, sorting-based multi-objective evolutionary algorithm (MOEA) called the non-dominated sorting genetic algorithm II (NSGA-II). Two different objectives were followed: the integral of time-weighted absolute error (ITAE) performance index, and the integral of the square of the error (ISE), in which minimizing the first objective will provide good reference tracking and good disturbance rejection, while minimizing the second objective reduces the rise time. The final gain values used in the simulations are provided in Table 2.
Table 2. Gains for the PID controllers.

<table>
<thead>
<tr>
<th>Controller</th>
<th>Gain ( x )</th>
<th>Gain ( y )</th>
<th>Gain ( z )</th>
<th>Gain ( \phi )</th>
<th>Gain ( \theta )</th>
<th>Gain ( \psi )</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>3.1</td>
<td>0.9</td>
<td>5.3</td>
<td>2.9</td>
<td>2</td>
<td>4.8</td>
</tr>
<tr>
<td>I</td>
<td>0.1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>2.7</td>
<td>1.4</td>
<td>2.9</td>
<td>16</td>
<td>13.3</td>
<td>16</td>
</tr>
</tbody>
</table>

5. Simulation Results

The final formulation for the controllers described in Section 4 (i.e., Eqs. 20, 27, and 30) were implemented and analyzed in the MATLAB/Simulink software. For this, the physical characteristics of a nine inch diameter-shrouded propeller’s version of the Navig8 UAV were used based on previous research studies, i.e., the studies published by the authors of [15,25] (Table 3). PID gains were set as per Table 2, and the sampling time was set to 0.01 s. Overall, the controller received a desired path/trajectory to follow, which would include the desired position \((x_d, y_d, or z_d)\) and orientation \((\phi_d, \theta_d, or \psi_d)\) as a function of time, defining the UAV’s maneuvers to compute the required control signals. From the physical characteristics of the UAV, the required parameters were computed.

Table 3. Simulation parameters and physical characteristics of the UAV.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
<th>Parameter</th>
<th>Value</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass of the UAV</td>
<td></td>
<td>( m )</td>
<td>5</td>
<td>kg</td>
</tr>
<tr>
<td>Gravity</td>
<td></td>
<td>( g )</td>
<td>9.81</td>
<td>m/s²</td>
</tr>
<tr>
<td>Air density</td>
<td></td>
<td>( \rho )</td>
<td>1.225</td>
<td>kg/m³</td>
</tr>
<tr>
<td>Propeller distance from the UAV’s</td>
<td>Distance to the left</td>
<td>((a, b, c))</td>
<td>(0, 0.26,</td>
<td>m</td>
</tr>
<tr>
<td>center of mass</td>
<td>propeller</td>
<td></td>
<td>0.05)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Distance to the right</td>
<td>((a, -b, c))</td>
<td>(0, -0.26</td>
<td>m</td>
</tr>
<tr>
<td></td>
<td>propeller</td>
<td></td>
<td>, 0.05)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Distance to the tail</td>
<td>((d, 0, e))</td>
<td>(-0.57, 0</td>
<td>m</td>
</tr>
<tr>
<td></td>
<td>propeller</td>
<td></td>
<td>0, 0.008)</td>
<td></td>
</tr>
<tr>
<td>Propellers</td>
<td>Trust coefficient of the</td>
<td>( C_{T1}, C_{T3} )</td>
<td>0.002</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>left/right propellers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Torque coefficient of the</td>
<td>( C_{T1}, C_{T3} )</td>
<td>0.001</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>left/right propellers</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Trust coefficient of the</td>
<td>( C_{T2} )</td>
<td>0.001</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>tail propeller</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Torque coefficient of the</td>
<td>( C_{T2} )</td>
<td>0.0005</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>tail propeller</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Footprint area of the left/right</td>
<td>( A_{1,3} )</td>
<td>0.0164</td>
<td>m²</td>
</tr>
<tr>
<td></td>
<td>propellers’ radius</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rotor disk area of the tail</td>
<td>( A_{2} )</td>
<td>0.002</td>
<td>m²</td>
</tr>
<tr>
<td></td>
<td>propeller</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Radius of the right/left</td>
<td>( r_{1,3} )</td>
<td>0.23</td>
<td>m</td>
</tr>
<tr>
<td></td>
<td>propellers’ radius</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Radius of the tail</td>
<td>( r_{2} )</td>
<td>0.098</td>
<td>m</td>
</tr>
<tr>
<td>UAV’s body</td>
<td>Inertia moment around x</td>
<td>( I_{xx} )</td>
<td>0.0667</td>
<td>kg. m²</td>
</tr>
<tr>
<td>moment of inertia</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Inertia moment around y</td>
<td>( I_{yy} )</td>
<td>0.1492</td>
<td>kg. m²</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Inertia moment around z</td>
<td>( I_{zz} )</td>
<td>0.2019</td>
<td>kg. m²</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Inertia moment around x and z</td>
<td>( I_{xz} )</td>
<td>-0.0147</td>
<td>kg. m²</td>
</tr>
</tbody>
</table>

To analyze the performance of the controller, the UAV was commanded to execute standardized maneuvers considered fundamental in determining the flight performance (worthiness) of the rotorcraft as per that conducted by the authors of [36]. Flight maneuvers included pure hover and sidestep, which the aircraft was able to perform as per flight regulations.

In what follows, however, we present the results of an untypical flight maneuver which illustrates the ability to use the developed controller to fly a highly maneuverable
rotorcraft inside confined spaces by changing its attitude according to the environmental conditions (e.g., confined space).

5.1. Test 1: Trajectory Tracking

In this first simulation, the aircraft (Navig8 UAV) was commanded to track the trajectory defined by Equation (43), which defines a set of movements within the aircraft’s flying capabilities. Although this trajectory does not exemplify a confined space, it requires the UAV to change its attitude in an atypical fashion (a maneuver that a helicopter pilot would not attempt).

\[
\begin{align*}
    x_d(t) &= 10 \sin(0.1t) \\
    y_d(t) &= -2 \cos(0.1t) + 2 \\
    z_d(t) &= \sin(0.1t) \\
    \psi_d(t) &= \sin(0.05t) \\
    \theta_d(t) &= 0.3 \sin(0.1t)
\end{align*}
\]

(43)

The UAV’s initial state, including its position, orientation and their first derivatives (as per Equation (11)), was set as \( x_0 = [0,0,0,0,0,0,0,0,0] \). It must be noted that \( x_0 \) defines the state where the UAV is in a typical hovering maneuver position at a given attitude “\( h \)” from the ground where the frame of reference that defines the trajectory is defined (set).

The drone’s position responses are shown in Figure 5, while its attitude responses are shown in Figure 6. Figure 7 depicts the position and orientation followed in a 3D space.

![Figure 5. Responses to position control in trajectory tracking.](image-url)
These results indicate that the developed trajectory tracking control approach was effective, showing a low RMS error in both position and attitude control. Although at the start there was an error of 0.53 m between the desired and the followed trajectory, the UAV was able to track the desired path for the rest of the trajectory. Such a maximum error occurs in a specific situation where the desired roll and lateral translations are somewhat in conflict. However, the controller effectively managed such conflicts and guided the UAV to complete the flight with sufficient accuracy to fly in confined spaces.

Similar results were obtained under diverse trajectories, including pure hover, pitch hover, climb, and descent.

5.2. Test 2: Disturbance Rejection

To assess/analyze the ability of the controller to deal with external disturbances, the UAV was commanded to follow the same trajectory as described in Section 5.1, where wind gusts and other disturbances were added. Here, however, we only present an illustrative example of exposing the UAV to horizontal wind gusts, having a maximum speed of 5 m/s with variable magnitude and direction (moderate breeze based on the “Beaufort Wind Scale” (see https://www.weather.gov/mfl/beaufort (accessed: 14 August 2023).

Such a varying disturbance was implemented following the work conducted by the authors of [37], where the magnitude of the wind was defined by Equation (44) and the azimuth, $\psi_w$, of the blow point of the wind (Equation (45)) defined the direction of it,
while random values were used to denote the changes in the wind’s direction. The magnitude of the wind, \( V_z \), can then be defined as a function of the altitude of the drone with respect to the sea level, \( z_0 \) (Equation (44)).

\[
V_z = V_{z0}(\frac{z}{z_0})^p
\]  

(44)

\[
\psi_w = \psi_{w0} + \text{rand}
\]  

(45)

In Equation (44), \( V_{z0} \) is the wind velocity measured at altitude \( z_0 \), \( z \) is the current altitude of the UAV, and “\( p \)” is the index of the energetic profile, and in Equation (45), “\( \psi_{w0} \)” is the azimuth angle in the previous time step, and “\( \text{rand} \)” is a random function to create a random number between zero and one (rad). Therefore, the wind force in the x and y directions of the UAV’s body frame of reference affecting the drone were calculated using Equation (46):

\[
F_{wx} = S_{ex}AV_z^2\cos(\psi_w) \\
F_{wy} = S_{ey}AV_z^2\sin(\psi_w)
\]  

(46)

where \( A = 1.6 \times 9.81 \) is the rate of converting the wind speed (\( m/s^2 \)) into pressure (\( N/m^2 \)) [38], and \( S_e \) is the influence effective area. To ease obtaining the parameters \( A \) and \( S_e \) (Equation (46)), the Navig8 was modelled as a cubic body (Figure 8), which simplifies computing the total force of the wind disturbance and its distribution over the drone’s fuselage. Although this is a very rough estimate, it allowed us to evaluate how well the controller held up to external, previously unidentified wind disturbances. From Figure 8, it was possible to compute the drone’s surface area as per Equation (47):

\[
S_{ex} = S_1 \times \sin(\theta) \times \cos(\psi) + S_f \times \cos(\theta) \times \cos(\psi) \\
S_{ey} = S_1 \times \cos(\phi) \times \cos(\psi) + S_f \times \sin(\phi) \times \cos(\psi)
\]  

(47)

where \( S_1 = \delta_1h_1h_2 \) is the drone’s lateral surface area, \( S_f = \delta_2h_1h_3 \) is the UAV’s face area, \( h_1, h_2, \) and \( h_3 \) are the cubic dimensions, as shown in Figure 8, and \( \delta_1 \) and \( \delta_2 \) are the fill features of the base cube, as shown in Figure 8.

**Figure 8.** Approximation of the Navig8 structure (\( \delta_1 = 0.1 \) and \( \delta_2 = 0.2 \), respectively).

Figure 9 shows the obtained magnitude and direction (azimuth, \( \psi_w \)) of the wind affecting the UAV over time. As shown in Figure 9a, the wind velocity has points where the wind declines (to a value of \( 2.1 \, m/s^2 \)), after which it gradually increases to a max value of \( 5 \, m/s^2 \). In turn, the direction of the wind (Figure 9b) changes at discrete intervals. Figures 10a,b shows the wind forces in the x and y directions, respectively. Figures 11 and
provide the position and attitude responses of the UAV to the wind disturbances while trying to track the desired trajectory (Equation (43)).

Figure 9. Simulated wind: (a) velocity and (b) direction.

Figure 10. Simulated wind force in the (a) x direction and (b) y direction.
Figure 11. Responses to position control in trajectory tracking under moderate wind gusts.

Figure 12. Responses to attitude control in trajectory tracking under moderate wind gusts.

Figure 13 shows the desired and reached trajectories under variable wind disturbances. The result shows the effectiveness of the developed controller in being able to perform this maneuver, and the root mean square errors (RMSEs) obtained under all position and attitude directions were considered to be small/acceptable, as presented in Table 4.

Table 4. Root mean square errors.

<table>
<thead>
<tr>
<th>RMSE_x</th>
<th>RMSE_y</th>
<th>RMSE_z</th>
<th>RMSE_θ</th>
<th>RMSE_ψ</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0281</td>
<td>0.1913</td>
<td>0.0062</td>
<td>0.0082</td>
<td>0.0082</td>
</tr>
</tbody>
</table>
Figure 13. Tracking the 3D trajectory under moderate wing gusts for 150 s. Note that a helicopter schematic has been used to better represent the change in the orientation of the UAV.

From Figures 11–13, it is observed that the controller was able to guide the UAV with high accuracy (i.e., the max error in translation was the same as the result for the simulation with no disturbances, Figure 5). However, this UAV had experienced some challenges managing the coupled lateral and roll motions as it flew (although such deviations were small).

6. Conclusions

The proposed nonlinear trajectory controller is based on the INDI controller technique, which includes an outer NDI/INDI loop controller for position and an inner INDI loop controller for attitude. The controllers are used in combination with PD/PID controllers to provide appropriate control inputs for the NDI/INDI blocks. The linear controllers’ gains are tuned manually and show satisfying results for different flight maneuvers. In order to improve the disturbance rejection capability of the NDI controller in the $y$ channel, a single-layer sigma-pi neural network is added, and the simulation result demonstrates its ability to compensate for the control law. Finally, several simulations demonstrate the ability of the controller to follow different trajectories under moderate external disturbances (wind gusts).

7. Future Work

In this paper, the pitch angles of the propeller are considered constant; that is, the variable pitch capabilities of the aircraft are not utilized. However, in order to take advantage of the full capacity of the designed UAV, investigating the relation between the blade pitch angle and the aerodynamic coefficient will be performed in future work. Furthermore, work will be conducted in enhancing the developed controller to include the necessary control allocation mechanisms to compute the required configuration of the UAV in each time step. Finally, performing flight tests under incremental levels of complexity within a controlled (engineered) environment, such as flying inside a wind tunnel, followed by flight tests under real-world scenarios are future work tasks that will be performed.

Author Contributions: Conceptualization, methodology; software, validation, formal analysis, investigation, visualization, and writing, M.T. and A.R.-S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported, by the Natural Sciences and Engineering Research Council of Canada (NSERC) through an Alliance grant and Alberta Innovates—Technology Futures.

Institutional Review Board Statement: Not applicable.
Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable—No specific data is available.

Acknowledgments: This work was supported, in part, by the Natural Sciences and Engineering Research Council of Canada (NSERC) through an Alliance grant. Support was also provided by Alberta Innovates—Technology Futures. We thank them for their gracious support.

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

References

**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.