Robust Trajectory-Tracking for a Bi-Copter Drone Using INDI: A Gain Tuning Multi-Objective Approach

Maryam Taherinezhad 1, Alejandro Ramirez-Serrano 1, 2,* and Arian Abedini 2, ©

1 Department of Mechanical Engineering, University of Calgary, Calgary, AB T2N 1N4, Canada
2 Department of Aerospace Engineering, K. N. Toosi University of Technology, Tehran 16569-83911, Iran
* Correspondence: aramirez@ucalgary.ca

Abstract: This paper presents an optimized robust trajectory control system for an autonomous tiltrotor bi-copter based on an incremental nonlinear dynamic inversion (INDI) strategy combined with a set of PID/PD controllers. The methodology includes a lower level, fast attitude control action using an incremental nonlinear dynamic inversion (INDI) strategy, which is driven by a higher level, slow trajectory control action that uses nonlinear dynamic inversion (NDI). The nonlinear dynamic model of the drone is derived, and the basis of the motion and the design of the attitude and position stabilizing controllers are discussed. To develop and test the suggested controller, a circle-shaped flight profile is simulated. The linear control providing inputs to the NDI and INDI controllers is tuned via a novel multi-objective optimization auto-tuning method using the non-dominated sorting genetic algorithm II (NSGA-II). The tracking and disturbance rejection optimization is achieved via the use of the integral of time multiplied by the absolute error (ITAE) and the integral of the square of the error (ISE) objective functions, which are optimized concurrently. The simulation results reveal that the proposed control design outperforms the traditional dynamic inversion controller design and demonstrate that the developed INDI + PID/PD controller possesses exceptional accuracy and performance, enabling the tiltrotor bi-copter to track the given trajectory. Furthermore, the paper shows that the proposed controller produces 40% lower overshoot and settling time as measured with respect to previous backstepping controllers reported in the literature. The robustness of the controller is validated through diverse tests where the aircraft is subjected to external (wind gust) disturbances.

Keywords: tiltrotor bi-copter drone; INDI; trajectory tracking; optimization; NSGA-II; sigma-pi neural network; adaptive control

1. Introduction

Multi-rotor drones and their novel structures and designs have been an attractive topic for researchers and industries in recent years. Typical multi-rotor drones have four, six, and eight rotors and can be controlled to perform different maneuvers by changing their propellers’ speed. They have been used in a wide variety of applications such as search and rescue [1,2], agriculture [3,4], data collection and monitoring [5,6], health assistance missions (for COVID-19) [7], and many other tasks. However, due to their high power requirements, their nonlinearities, and coupled flight effects, multi-rotor drones usually have short mission times and are difficult to control, especially under the presence of internal (e.g., failures) and external (e.g., wind) disturbances. Therefore, a new configuration of multi-rotor drones with two motors/propellers that can reduce the unit time power demand was introduced in [8]. By reducing the number of propellers, bi-copter airframes not only provide a less power-hungry eVTOL (Electric Vertical Take-off and Landing) aircraft but the aerodynamic interaction of the propellers with the fuselage and the ground (aerodynamic wall and ground effects) decreases, thus reducing the typical adverse effects during flight. Similar drones have enhanced flying acrobatic capabilities not seen in typical eVTOL systems (e.g., pitched hover); they have also been developed to fly inside confined spaces.
using diverse control schemes [9–11]. Although such aircraft are complex, they provide the characteristics for developing and testing new control strategies that can be extended to control other VTOL vehicles having advanced flight capabilities, such as the highly maneuverable drones for confined spaces developed at the University of Calgary [12]. Although the aircraft introduced in [8] is relatively simple when compared to other tiltrotor aircraft (e.g., [12] and V-22 Osprey), it is a nonlinear system [13], and has the ability, at least in theory, to maneuver through small spaces providing a suitable testbed platform to develop and test concept (enhanced) controllers for sophisticated eVTOL aircraft systems.

Several works have developed traditional PID controllers for position and attitude tracking of bi-copter drones [8,14–16]. Although somewhat effective, such techniques have a number of drawbacks. For starters, a linearized model of the aircraft is typically used, which provides an approximation of the behavior and flight capabilities of such systems. To really exploit the capabilities of such systems, nonlinear models should be used that describe the motion-coupling and aerodynamic effects inherited by the flying mechanisms. Second, any linearized model is only viable within minor deviations from the equilibrium or trim point. Thus, performance can suffer significantly if the aircraft drifts away from its intended design/flight trim point. As a result, gain scheduling is frequently used to achieve somewhat acceptable performance across the whole flight envelope.

A variety of nonlinear flight control systems have been proposed and developed in the hopes of overcoming some of the theoretical constraints and downsides of linear controller design. Examples of nonlinear control methods to control the motion of flying robots include feedback linearization [17], backstepping [10,13,18,19], predictive [9,20,21], and sliding mode control [22–24]. Although these methods provide the ability to control drones in different conditions, they are highly dependent on the accuracy of the vehicle’s dynamic model. Furthermore, such controllers also have limited capability to cope with unknown internal or external disturbances affecting the aircraft flight and the uncertainties (accuracy) of the mathematical model.

Among such controllers, the feedback linearization technique has attracted the most attention as it has proven to have the most potential to control diverse Unmanned Aerial Vehicles (UAVs) [25]. The most important feature of the associated synthesis process for this type of controller is that gain scheduling is no longer required. A characteristic that other controllers share.

Nonlinear dynamic inversion (NDI) is a feedback linearization candidate methodology that aims to remove gain scheduling by inverting and canceling the inherent dynamics and replacing them with a set of user-selected desirable dynamics [26]. Because of these factors, NDI is a promising control design method extending the capabilities of other control techniques, especially for aircraft having large flight maneuvering envelopes. However, the main challenge in using feedback linearization for air vehicle control is that it necessitates extensive knowledge of the nonlinear dynamic model of the system and, therefore, cannot deal with the uncertainties [27]. Because these are approaches based on system knowledge, they are referred to as model-based solutions. However, the model accuracy requirement can be costly and time-consuming. In the case of aircraft control, where there are a number of uncertainties, such as aerodynamic coefficients (coefficients that change based on the flight maneuver being performed), extensive wind tunnel tests as well as flight and CFD studies are required to be able to properly generate the needed aircraft’s mathematical model.

Because of the general advancement of micro-electromechanical systems (MEMS), smaller, more accurate, and more accessible sensors are now available. This enables sensor-based control approaches in which model knowledge can be replaced by variable measurements to be developed. Incremental nonlinear dynamics inversion (INDI) is one such sensor-based control approach [28–31] in which the incremental control input replaces the total control input used in NDI. For this, the incremental control input is calculated based on the performance of the system in the previous time step via sensor measurements. Therefore, INDI is less dependent on the dynamic model of the system and
model uncertainties. However, this technique is required to be executed at a high-frequency using accurate sensor data and/or accurate variable estimation.

In this paper, a nonlinear mathematical model of the bi-copter presented in [13] and shown in Figure 1 is developed.

Figure 1. Bi-copter prototype.

The control architecture includes the development of a novel multi-loop trajectory controller designed based on INDI and PID controller formulations. First, an inner-loop controller is designed for high-performance attitude tracking, followed by the development of an outer loop that tracks position variables via an NDI controller. The outer loop generates attitude commands that appropriately orient the tilting rotor forces to generate the required translational accelerations. This two-loop control approach generally requires timescale separation where the outer-loop bandwidth must be significantly lower than the inner-loop bandwidth. Following the typical system linearization performed in INDI (or NDI), it is now possible to use the linear control to stabilize each state of the UAV (position and orientation) separately.

PID controllers have been developed and used in diverse fields such as industrial control, energy, as well as in the control of unmanned aircraft. However, optimizing the three parameters required in any PID controller is a difficult task, especially when several performance indices are to be met at the same time. There exist several common methodologies, including Pole placement and Lyapunov theory, to obtain linear control parameters for a given system, including the control gains to use, but there is not an accurate method of tuning these parameters [32].

In this paper, a bi-copter drone similar to the one presented in [21] is used to develop diverse methods have been developed to obtain optimal parameters (e.g., gains) for numerous aircraft. Examples of such techniques include Min–Max parameter optimization, where researchers have used DI airplane flight controller (e.g., [32]), and fuzzy proportional/integral/derivative (fuzzy-PID) design [33].

In this paper, the NSGA-II optimization algorithm is used to optimize the PID parameters of the bi-copter’s INDI controller. According to [34], NSGA-II can find a much wider range of solutions and converge closer to the true optimal solution when compared to other evolutionary algorithms (EA).

In this paper, a bi-copter drone similar to the one presented in [21] is used to develop a novel INDI controller for autonomous operation. Due to the complexities associated with the control of such a drone concept (e.g., coupled dynamics), prior work only considered the development of a backstepping (BKS) controller to operate the aircraft under benign flight conditions [21] and tracking relatively simple trajectories (defined as a set of points in 3D space) without explicit system (drone) orientation. Since BKS requires explicit system model information, such a solution was highly sensitive to model uncertainties and prone to poor disturbance rejection. This paper provides an improved control solution based on the combination of the incremental nonlinear dynamic inversion (INDI) strategy
with simple PID controllers, which reduces model dependency and enhances disturbance rejection. The proposed controller is a novel architecture that can be easily applied to diverse systems beyond autonomous aircraft, especially those with complex dynamics operating in the presence of inner and/or outer disturbances, such as wind gusts and system failures. The proposed control architecture employs an offline optimization technique to tune the required PID control gains, which provides effective results even with minimal tuning efforts.

The paper is divided into four sections: Section 2 extracts the bi-copter’s nonlinear mathematical model; Section 3 describes the proposed dynamic inversion method and Section 4 describes the NSGA-II optimization algorithm. Numerical simulations are presented in Section 5, followed by the conclusions in Section 6.

2. Dynamic Model: Tiltrotor Bi-Copter

2.1. Flight Mechanism

The drone that is the subject of this paper is a six degrees-of-freedom (DOF) dual rotor drone, bi-copter, with tilting rotors (Figure 1). The aircraft has four control inputs, \( w_1, w_2, t_1, \) and \( t_2 \), which correspond to the left/right propellers’ speed and their corresponding tilt angles around the \( y \)-axis of the vehicle’s body frame (Figure 2).

![Figure 2. Body and inertia reference frames of the bi-copter shown in Figure 1.](image)

The aircraft, having a mass “\( m \)”, is able to fly sideways as well as forward and backward by the control of the four control inputs. The motion sideways (in the “\( y \)”-direction, see Figure 2) is obtained by generating a differential thrust/speed of the propellers. By changing the tilt angles, \( t \), of the propellers in equal amounts fore and aft, motion is achieved (i.e., motion in “\( x \)”-direction). Yaw motion (i.e., rotation about the vehicle’s “\( z \)”-axis) is achieved by the differential tilt angle of the propellers. The described motions are, however, coupled. For example, during sideways motion, the differential rotational speeds of the propellers will also generate roll and yaw motions. In other cases, the gyroscopic effects of the tilting propellers will also cause a coupled pitch motion of the UAV.

2.2. Dynamic Model

In order to derive a dynamic model of the system, the Newton–Euler formulation is used. For this, two frames of reference are used: An inertial reference frame, \( X_IY_IZ_I \), and a body reference frame, \( X_BY_BZ_B \), as shown in Figure 2 with the transformation matrix from the inertial reference frame to the body reference frame denoted as \( R_{B}^{I} \) (Equation (1)). Furthermore, for the modeling of the aircraft, the Euler angles (\( \psi, \theta, \phi \Rightarrow \text{roll, pitch, yaw} \)) used to denote and represent the motion of the aircraft with respect to the inertial frame of reference were used. Such angles denote the rotational speeds and tilt angles of the left and right motors, respectively. In what follows, the subscripts 1 and 2 are used to represent the
left and right propellers of the aircraft, respectively, as well as the corresponding parameters (e.g., \( \omega_i \) for the rotational speed of propeller “\( i \)”).

\[
R_B = \begin{bmatrix}
\cos \theta \cos \phi & \sin \phi & -\sin \theta \\
\cos \theta \sin \phi + \sin \phi \sin \theta \cos \psi & \cos \phi \sin \theta - \sin \phi \sin \theta \cos \psi & \cos \phi \cos \theta \\
\sin \phi \sin \theta - \cos \phi \sin \theta \cos \psi & \cos \phi \cos \theta + \sin \phi \sin \theta \cos \psi & \sin \phi \cos \theta
\end{bmatrix}
\] (1)

The relationship between the angular velocity of the aircraft with respect to the inertial frame \((p, q, r)\), and the first derivative of the Euler angles is:

\[
\begin{bmatrix}
\dot{\phi} \\
\dot{\theta} \\
\dot{\psi}
\end{bmatrix} = \begin{bmatrix}
1 & \tan \theta \sin \phi & \tan \theta \cos \phi \\
0 & \cos \phi & -\sin \phi \\
0 & \sin \phi / \cos \theta & \cos \phi / \cos \theta
\end{bmatrix} \begin{bmatrix}
p \\
q \\
r
\end{bmatrix}
\] (2)

In order to simplify the equations of motions, the following four assumptions are employed: (i) the center of the mass of the UAV and the body reference frame coincide, (ii) the structure of the drone is symmetric to the XZ (longitudinal) plane so that \( I_{XY} = 0 \) and \( I_{YY} = 0 \), (iii) the aerodynamic coefficients such as drag do not change, and (iv) the forces created by the propellers are proportional to the square of their rotation speed. Furthermore, the drag forces of propellers are considered negligible.

Based on the above-described aspects, the equation of the motion for the bi-copter is formulated as described in Equation (3):

\[
\begin{aligned}
\dot{x} & = -\sin \phi \sin \theta + \cos \phi \cos \theta \omega_1^2 \cos \theta_1 + \omega_2^2 \cos \theta_2 \\
\dot{y} & = \cos \phi \sin \theta + \sin \phi \cos \theta \omega_1^2 \sin \theta_1 + \omega_2^2 \sin \theta_2 \\
\dot{z} & = -\frac{m}{C_T} \omega_1^2 \sin \theta_1 \omega_2^2 \sin \theta_2 + g \\
\dot{p} & = \frac{I_{YX}}{I_{XX}} (r + p q) + \frac{I_{YY} - I_{YX}}{I_{XX}} q r + \frac{l_{CY}(2 \omega^2 \cos \theta_1 - \omega_2^2 \cos \theta_2)}{l_{CY}} \\
\dot{q} & = \frac{I_{ZX}}{I_{YY}} (r^2 - p^2) + \frac{2 \omega_1 \omega_2 \sin \theta_1 + \omega_2^2 \sin \theta_2}{I_{YY}} p r + \frac{h_C \omega^2 \sin \theta_1 + \omega_2^2 \sin \theta_2}{l_{CY}} \\
\dot{r} & = \frac{I_{ZX}}{I_{ZZ}} (p - q r) + \frac{I_{XXX} - I_{XYZ} p q}{I_{XX}} + \frac{l_{CR}(\omega_1^2 \sin \theta_1 + \omega_2^2 \sin \theta_2)}{l_{ZZ}}
\end{aligned}
\] (3)

where \( C_T \) is the thrust coefficient of the motors, \( \omega_1 \) and \( \omega_2 \) are the rotational speeds of the left and right propellers, respectively, \( l \) and \( h \) are the horizontal and vertical distances of the rotors to the center of the mass of the UAV, respectively, and, \( t_{ij} \), represents the tilt angle of the propellers “\( i \)” (see Figures 1 and 2) [8]. In order to simplify these coupled sets of equations (Equation (3)), the four control variables \( U_1, U_2, U_3, U_4 \) are defined as shown in Equation (4). Then Equation (3) can be rewritten as Equation (5).

\[
U = \begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} C_T(\omega_1^2 \cos \theta_1 + \omega_2^2 \cos \theta_2) \\ C_T(\omega_1^2 \cos \theta_2 - \omega_2^2 \cos \theta_2) \\ C_T(\omega_1^2 \sin \theta_1 + \omega_2^2 \sin \theta_2) \\ C_T(\omega_1^2 \sin \theta_1 - \omega_2^2 \sin \theta_2) \end{bmatrix}
\] (4)
where

\[- \frac{L_t}{m} (\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi) - \frac{\cos \phi}{m} U_3 \]

\[- \frac{L_t}{m} (\sin \phi \cos \psi - \cos \phi \sin \theta \sin \psi) + \frac{\cos \phi}{m} U_3 \]

\[- \frac{\cos \psi}{m} \frac{L_t}{m} - \sin \phi U_3 + g \]

\[p = \frac{I_{yy} - I_{zz}}{I_{xx}} (\dot{r} + pq) + \frac{I_{yy} - I_{zz}}{I_{xx}} qr + \frac{I_{yy} - I_{zz}}{I_{xx}} pr + \frac{I_{yy} - I_{zz}}{I_{xx}} \]

\[q = \frac{I_{yy} - I_{zz}}{I_{xx}} (r^2 - p^2) + \frac{I_{yy} - I_{zz}}{I_{xx}} qr + \frac{I_{yy} - I_{zz}}{I_{xx}} pr + \frac{I_{yy} - I_{zz}}{I_{xx}} \]

\[r = \frac{I_{yy} - I_{zz}}{I_{xx}} (p - qr) + \frac{I_{yy} - I_{zz}}{I_{xx}} pq + \frac{I_{yy} - I_{zz}}{I_{xx}} \]

The kinetic model, expressed by Equation (5), is then used to develop the corresponding controller, which is then implemented in the MATLAB/Simulink simulation environment where testing and analyzing of the flight controller is performed.

3. Controller Design

Aircraft do not always behave as linear systems. This is especially true when performing aggressive maneuvers or experiencing internal or external disturbances or when the aircraft’s design is non-traditional such as bi-rotor and tiltrotor aircraft. Thus, in some flight regimes, including failure scenarios, they behave in a nonlinear fashion. To control them effectively and efficiently, one must use nonlinear controllers. Such controllers must also be robust, as the mathematical models used to derive the corresponding controller never exactly matches the real-world system.

In this paper, nonlinear dynamic inversion (NDI) and incremental NDI (INDI) are considered. The reason is that such controllers have been proven to perform much better than linear controllers (e.g., PID) in terms of disturbance rejection [28,29] (as one example) and are easier to use for controlling nonlinear systems. Moreover, such controllers offer the possibility for very robust control.

3.1. Controller Formulation

3.1.1. Nonlinear Dynamics Inversion

Considering the general equation of the motion for a Multi Input Multi Output (MIMO) system [35], the equation of the motion of the bi-copter tiltrotor UAV has the following form (Equations (6) and (7)):

\[
\dot{x} = f(x) + G(x)u
\]

\[
y = h(x)
\]

where \(x \in \mathbb{R}^n\) is the system’s state vector, \(u \in \mathbb{R}^m\) is the control input vector, \(f(x) \in \mathbb{R}^n\) and \(h(x) \in \mathbb{R}^m\) are smooth nonlinear vector fields (functions), and \(G(x) \in \mathbb{R}^{n \times m}\) is one of the system’s model matrix where each column contains values defined as \(g_i \in \mathbb{R}^n\).

By differentiating Equation (7) with respect to time, Equation (8) is obtained:

\[
\dot{y} = \frac{dh(x)}{dt} = \frac{\partial h}{\partial x} \dot{x} = \frac{\partial h}{\partial x} f(x) + \frac{\partial h}{\partial x} G(x)u = L_f h(x) + L_G h(x)u
\]

where \(L_f h(x)\) and \(L_G h(x)\) are the Lie Derivatives of \(h(x)\) with respect to \(f\) and \(G\), respectively. Assuming \(L_G h(x)\) is invertible, it is possible to isolate the (virtual) control action \(u\) as shown in Equation (9).

\[
u = (L_G h(x))^{-1}(\dot{y} - L_f h(x))
\]

Equation (9) uses a reasonable assumption, which guarantees that the function \(L_G h(x)\) satisfies the zero typical conditions of the \(\det(L_G h(x))\) while avoiding infeasible system configurations. This is possible by defining limitations to the equations describing the aircraft’s behavior. In turn, such determinants can be easily obtained via a pseudo-inverse computation to avoid any singularity of the controller. Applying this control action to (6) results in a closed-loop linearized system of the form \(\dot{x} = \ddot{x}_d\). A block diagram describing such an approach is presented in Figure 3.
where \( x \in \mathbb{R}^{n} \) is the system’s state vector, \( u \in \mathbb{R}^{m} \) is the control input vector, \( f(x,u) \) is a nonlinear function: 
\[
\dot{x} = f(x,u)
\]  
(10)

If the model described by Equation (10) is used to develop a controller running with a sampling time, \( \Delta T \), it is possible to linearize Equation (10) around the previous time step, \( t_0 = t - \Delta T \), where the system’s state was defined by \( (x_0, u_0) \), as represented by Equation (11). 
\[
\dot{x} = \dot{x}_0 + \frac{\partial F}{\partial x}|_{x_0,u_0}(x - x_0) + \frac{\partial F}{\partial u}|_{x_0,u_0}(u - u_0)
\]  
(11)

where \( \dot{x}_0, x_0, \) and \( u_0 \) denote the state derivative, state, and control input parameters evaluated/obtained at the previous time step, and \( \frac{\partial F}{\partial x}|_{x_0,u_0} \) and \( \frac{\partial F}{\partial u}|_{x_0,u_0} \) are the partial derivatives of \( F(x_0,u_0) \) with respect to \( x_0 \) and \( u_0 \) at the previous time step. Assuming a small enough sampling time, \( \Delta T \) is used; considering the fast actuators’ dynamic of the UAV, the state variation between samples can be considered negligible (\( \dot{x} \approx x_0 \)). Thus, it is possible to further simplify (11) via Taylor series, resulting in Equation (12). 
\[
\dot{x} \approx \dot{x}_0 + \frac{\partial F}{\partial u}|_{x_0,u_0}(u - u_0)
\]  
(12)

Setting the desired dynamics as \( \dot{x} = \dot{x}_d \) the following control input is obtained: 
\[
u = u_0 + (F_u)^{-1}(\dot{x}_d - \dot{x}_0)
\]  
(13)

where \( F_u = (\frac{\partial F}{\partial u}|_{x_0,u_0}) \) must be invertible and \( \dot{x}_0, x_0, \) and \( u_0 \) must be observable. To make sure \( F_u \) is invertible in all time steps, \( F_u \) must be a square matrix with a non-zero determinant. In cases when \( F_u \) is not square, the pseudo inverse must be used to avoid any singularity of the controller. A block diagram of the proposed closed-loop system, as described by Equations (11)–(13), is presented in Figure 4.

**Figure 3.** NDI control structure.
where faster and more complex dynamics are involved.

(assumed to be immediately achieved) when needed by the slow dynamics of the outer degrees of freedom) provides input to the inner loop (designed to control the rotational equations of the targeted UAVs and INDI provides the opportunity to use unknown/partially known dynamic equations of the aircraft in addition to being capable of dealing with other sources of uncertainties and disturbances.

Due to the under-actuated kinematics and dynamics of the bi-copter as defined by its mathematical model (Figure 1), the timescale separation principle must be applied, dividing the control law(s) into two loops: The outer loop (designed to control the translational degrees of freedom) provides input to the inner loop (designed to control the rotational degrees of freedom). This principle is used here to ensure that the dynamics of the inner loop are fast enough so that the vehicle’s rotational motion (i.e., attitude angles and rates of change of the rotation) required by the outer loop in its computation are readily available (assumed to be immediately achieved) when needed by the slow dynamics of the outer loop. To achieve this, a nonlinear controller that combines an NDI and an INDI controller is proposed (Figure 5). The reason for this control selection is that both NDI and INDI control structures have a good ability to deal with highly coupled complex dynamic equations of the targeted UAVs and INDI provides the opportunity to use unknown/partially known dynamic equations of the aircraft in addition to being capable of dealing with other sources of uncertainties and disturbances.

Since the outer loop control has slow dynamics, an NDI controller is used for position (x, y, z) control, while an INDI is used to control the aircraft’s angular rotation, (φ, θ, ψ), where faster and more complex dynamics are involved.

3.2. Attitude Controller

For the INDI attitude controller, the state of the drone is defined to include the Euler angles for roll, pitch, and yaw and the corresponding rotational speeds of the drone with respect to the aircraft’s body frame of reference (p, q, r), as described by Equation (14).

\[
X = \begin{bmatrix} \theta \\ \omega \end{bmatrix} = \begin{bmatrix} \phi \\ \theta \\ \psi \\ p \\ q \\ r \end{bmatrix} \quad u = \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix}
\]  

(14)
where $\theta = (\phi, \theta, \psi)$ and $\omega = (p, q, r)$. It should be noted that the mapping between $\theta$ and $\omega$ is defined by the corresponding transformation matrix. Since the control variables are the Euler angles, we can define $\phi$, $\theta$, and $\psi$ as the output variables and later as the control variables per Equation (15):

$$\vartheta = Hx = [I_{3 \times 3}]_x$$  \hspace{1cm} (15)

Since the first-order time derivative of the control variable vector, $y$, does not contain the control input vector, $u$, as described in Equations (16) and (17), the direct relationship between the input and output variables was determined by executing the second-order derivative of the control output,

$$\frac{d^2 \theta}{dt^2} = \frac{d(\vartheta)}{dt} = \frac{d}{dx} \left( \begin{array}{c} \tan \theta \sin \phi \\ \cos \phi \\ -\sin \vartheta \end{array} \right) \left( \begin{array}{c} p \\ q \\ r \end{array} \right) = J_x$$  \hspace{1cm} (17)

It is then clear that the input explicitly appears in the obtained expression. From this formulation, it is then possible to deduce the INDI controller for the complete system by performing a Taylor expansion in the current time step, $t$, as described in Equations (18) and (19). This expansion provides the linearized relationship between the control inputs and the Euler angles.

$$\dot{\theta} = K_\theta(x, u) = J(x)[K(x, u)]$$  \hspace{1cm} (18)

$$\ddot{\theta} \approx K(x_0, u_0) + \frac{\partial K_\theta(x, u)}{\partial x} |_{x_0, u_0} (x - x_0) + \frac{\partial K_\theta(x, u)}{\partial u} |_{x_0, u_0} (u - u_0)$$  \hspace{1cm} (19)

The linearization shown in Equation (19) assumes $K(x_0, u_0)$ to be the angular acceleration measurement taken at the previous time step, $t-1$. With a very small sampling time and considering that the actuator dynamics are faster than the dynamics of the other system dynamics, it is possible to assume that $(x - x_0) \approx 0$ when compared to the changes of control output, $\Delta u = (u - u_0)$. When considering the high dynamics of the tiltrotor bi-copter, this assumption is quite bold. However, the dynamics of the tiltrotor bi-copter are highly dependent on the actuators’ dynamics. Even though one might think that the above-mentioned assumption (i.e., $\Delta T$) leading to the assumption that the variation between samples can be considered negligible, is restrictive, in practice, such conditions can be easily met as the state changes are negligible when compared to the typical fast actuator’s dynamics available in the market, under these conditions, a simplified expression that will be later used to obtain the control law can be expressed, as shown in Equation (20):

$$\ddot{\theta} \approx \ddot{\vartheta}_0 + K(x_0, u_0) \Delta u$$  \hspace{1cm} (20)

where matrix $K(x_0, u_0)$ is the derivative of the control inputs as shown in Equation (21).

$$K(x_0, u_0) = J(x_0) \left[ \begin{array}{c} I_{3 \times 3} \\ 0 \\ 0 \end{array} \right]$$  \hspace{1cm} (21)

$$\Delta u = K(x_0, u_0)^{-1} (\ddot{\vartheta} - \ddot{\vartheta}_0)$$  \hspace{1cm} (22)

Using Equation (20), as the control law, one can then obtain the input for the system by considering a simple relationship $\Delta u = (u - u_0)$, where $u_0$ is the previous input, and solving Equation (22) to reach the control input $u$. The formulation $K(x_0, u_0)^{-1}$ can then be used to calculate the needed control input at each time step, where, based on the previous assumptions, the system will not enter singularity points in the cases where K is not
invertible (as the pseudo inverse can be computed in such cases, enabling the controller to pass such conditions).

The nonlinear controller obtained in this form is slightly less dependent on the system’s mathematical model, thus enabling it to effectively obtain the control input base on the angular acceleration measurements, which, otherwise, would be challenging to obtain. Even though the controller does not depend on part of the model, it is verifiable that the part of the controller that is neglected only slightly influences the bi-copter’s behavior. Therefore, disregarding the slower dynamics of the model when the controller is designed does not significantly affect the bi-copter’s behavior.

3.3. Position Controller

As described above, to control the position of the UAV, an NDI controller is used comprising the outer loop within the proposed controller (Figure 5).

The proposed position controller follows a similar approach as the one developed for attitude control. The first control layer uses the INDI technique to cancel the nonlinearities associated with the linear dynamics of the system. Since, in the approach illustrated in Figure 5, the outer loop is less dependent on the model uncertainty, NDI can be effectively used to design the corresponding position controller. The virtual control variables, $u$, chosen for this controller are defined as per Equation (23).

$$u = \begin{bmatrix} U_x \\ U_y \\ U_z \end{bmatrix} = \begin{bmatrix} U_1 (\sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi) \\ U_1 (-\sin \phi \cos \psi + \cos \phi \sin \psi \sin \theta) \\ U_1 \cos \phi \cos \theta \end{bmatrix}$$ (23)

Therefore, based on the equations presented in Section 3.1, the first-order time derivative of $y$ for the NDI position controller is given per Equation (24).

$$\dot{X} = f(x) + G(x, u)u$$ (24)

Since an explicit dependence on $u$ was already found (i.e., Equation (22)), the linear relationship $\nu = X$ can then be imposed when det $G(x, u) \neq 0$ by selecting the control input vector, $u$, have the form described by Equation (25).

$$u = G(x, u)^{-1}(\nu - f(x))$$ (25)

As in Section 3.2, it is common to use a PID controller to improve the response of the system. For this, however, there is a need to obtain the optimized values of the PID gains. To obtain such gains in the scope of the multi-objective goal for the bi-copter UAV, a non-dominated sorting algorithm, NSGA-II, is used (described in Section 4).

The NDI outer loop controls the slow dynamics and generates the state command to the inner INDI block following Equations (26)–(28), as is shown in Figure 5.

$$\theta_d = \tan^{-1} \left( \frac{U_x \cos \psi + U_y \sin \psi}{U_z} \right)$$ (26)

$$\phi_d = \tan^{-1} \left( \frac{\cos \theta_d (U_x \sin \psi - U_y \cos \psi)}{U_z} \right)$$ (27)

$$U_1 = \frac{U_z}{\cos \theta \cos \phi_d}$$ (28)

4. NSGA-II

In general terms, the NSGA-II technique is a well-known fast sorting and selection multi-objective genetic algorithm. This algorithm is typically used in machining to select
diverse process parameters such as cutting speed, spindle rotational speed, and feed rate [36] needed to optimize the corresponding machining operation.

In this paper, NSGA-II is used to optimize the parameters of the PID controller shown in Figure 5, which provide the inputs to the NDI and INDI controllers described in Section 3, where the number of parameters is large, and an analytical solution is difficult to obtain. More specifically, we aim to tune the linear controller gains as an essential part of obtaining an accurate controller. Therefore, in this paper, the non-dominated sorting-based multi-objective evolutionary algorithm (MOEA), called the non-dominated sorted genetic algorithm II (NSGA-II), is used. By using this method, the goal is to optimize the PID linear parameter, which has a significant role in the behavior of the desired controller. During the optimization process, two different objectives are considered: the integral of the time-weighted absolute error (ITAE) performance index and the integral of the square of the error (ISE). Whereas minimizing the first objective will provide good reference tracking and good disturbance rejection, minimizing the second reduces the rise time. The mathematical descriptions of these two objectives are given by Equations (29) and (30), respectively:

\[ Ob_1 = ITAE = \int_0^{T_s} t |e(t)| dt \]  
\[ Ob_2 = ISE = \int_0^{T_s} e(t)^2 dt \]  

where the error signal \( e(t) \) is defined as the difference between the reference input signal and the controlled variable, and \( T_s \) is the simulation time.

5. Simulation Results

In order to analyze the advantages and disadvantages of the proposed control algorithm, as well as to quantify the differences with and without the gain optimization process, two sets of simulation results are presented (Sections 5.1 and 5.2, respectively).

Table 1 provides the values of the parameters that describe the bi-copter UAV and were used to obtain the results described in Sections 5.1 and 5.2.

<table>
<thead>
<tr>
<th>Parameters of the bi-copter UAV</th>
<th>Mass (kg)</th>
<th>Length (m)</th>
<th>Height (m)</th>
<th>Propeller Tilt Angle Range</th>
<th>( t_1 ) (Rad)</th>
<th>( t_2 ) (Rad)</th>
<th>( I_{xx} ) (N·m²)</th>
<th>( I_{yy} ) (N·m²)</th>
<th>( I_{zz} ) (N·m²)</th>
<th>( I_{XZ} ) (N·m²)</th>
<th>Max ( \omega_{1&amp;2} ) (rpm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mass</td>
<td>0.87</td>
<td>0.23</td>
<td>0.1</td>
<td>( t_1 ) (Rad)</td>
<td>( t_2 ) (Rad)</td>
<td>( I_{xx} ) (N·m²)</td>
<td>( I_{yy} ) (N·m²)</td>
<td>( I_{zz} ) (N·m²)</td>
<td>( I_{XZ} ) (N·m²)</td>
<td>Max ( \omega_{1&amp;2} ) (rpm)</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>0.04</td>
<td>0.04</td>
<td>0.04</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5.1. Simulation before Optimization

As an example of the numerous tests performed when controlling the UAV without gain parameter optimization, the UAV was commanded to track the trajectory shown in Figure 6, following a circular path defined by a set of \( x, y, \) and \( z \) points represented in Equation (31). It should be mentioned that \( x, y, z, \) and \( \psi \) are the controllable states, and the desired values of \( \phi \) and \( \theta \) are computed in the control allocation block to enable the system to follow the desired trajectory.

\[ x = \sin(0.05t) \]
\[ y = -\cos(0.05t) \]
\[ z = 1 \]  

\[ \phi(t) = \int_0^t \omega(t) dt \]
\[ \theta(t) = \int_0^t \psi(t) dt \]  

where \( \phi(t) \) and \( \theta(t) \) are the states of the desired controller.

\[ x = \sin(0.05t) \]
\[ y = -\cos(0.05t) \]
\[ z = 1 \]  

\[ \phi(t) = \int_0^t \omega(t) dt \]
\[ \theta(t) = \int_0^t \psi(t) dt \]  

where \( \phi(t) \) and \( \theta(t) \) are the states of the desired controller.
At the start of the simulation, the UAV is located at the takeoff site where the UAV is positioned 1 unit distance from the height where the desired circular trajectory is to be followed (see Figure 6). At takeoff, the UAV rapidly climbs to reach the required altitude. Since there is no constraint on how the vehicle intercepts the given flight path, how the aircraft reaches the desired trajectory (e.g., speed of climb) is entirely governed by the proposed INDI + NDI + PID controller.

During the test, the PD and PID gains are represented by a $1 \times 18$ vector, the elements of which correspond to the P and D gains of the corresponding controllers for the $\phi$, $\theta$, and $\psi$ angles (nine gains, considering the corresponding I gains are equal to zero), and the P, I, and D gains for the position $x$, $y$, and $z$ (nine gains) for a total of 18 preselected values as shown in the following $k$ vector ($k = [k_P^\phi, k_I^\phi, k_D^\phi; k_P^\theta, k_I^\theta, k_D^\theta; k_P^\psi, k_I^\psi, k_D^\psi; k_P^x, k_I^x, k_D^x; k_P^y, k_I^y, k_D^y; k_P^z, k_I^z, k_D^z]$):

$$k = [2.5, 0, 10, 2.5, 0, 10, 2.5, 0, 10, 1, 0, 3, 1, 0, 3, 5, 1, 10]$$ (32)

These 18 gains were selected by trial and error, which were considered the best set of gains. Since the response of the propellers’ rotation angles and the position response of the UAV in the $z$-direction occurs very fast, the UAV reaches the desired height rapidly, and subsequently, it becomes more evident how the controller guides the UAV to follow the $x$ and $y$ values of the desired trajectory. Figures 7 and 8 show the results for the position and orientation when the UAV was under control using non-optimal gains (Equation (32)).
5.2. Simulation after Optimization

Similar to the test presented in Section 5.1, a second simulation test was performed using the aspects described in Section 5.1 (e.g., same UAV parameters, same takeoff position, and same desired trajectory) but using an optimized set of gains. The purpose is to compare and show the effects of the optimization process on the system’s behavior. To obtain the optimal set of gains, the NSGA-II optimization technique described in Section 4 was used to minimize the two objective functions, $O_1$ and $O_2$, defined by Equations (29) and (30). To perform such optimization, a population size equal to 30 was used while the optimization cycle was repeated for 100 generations. During the optimization process, crossover and mutation rates equal to 0.95 and 0.01, respectively, were used. The parameters (gains) optimized in this process included $K_1$, $K_2$, and $K_3$ for each of the $x$, $y$, $z$, $\phi$, $\theta$, and $\psi$ parameters (18 gains in total), respectively (note that in this case, the integral gains for the three angular orientations were considered as zero). The range of decision variables for $K_1$ and $K_2$ was selected to be between 0 and 15, while the range for the gains $K_3$ were selected to be between 0 and 1. This variable range selection was based on numerous simulation results used to eliminate reaching singularity points. The result of the optimization process produced the following $k$ vector.

$$ k = [10.6, 0, 13.4, 0, 5.8, 0, 8.8, 10.8, 13.0, 4.3, 1.9, 3.6, 6, 0, 3.5, 8.4, 0.2, 8] $$

Figure 6. Three-dimensional simulation of the UAV’s flight result when using non-optimal gains.

(a) $x$ position

(b) $y$ position

(c) $z$ position

Figure 7. Position response of the UAV when using non-optimal gains.

(a) roll

(b) pitch

(c) yaw

Figure 8. Orientation response when using non-optimal gains.
5.2. Simulation after Optimization

Similar to the test presented in Section 5.1, a second simulation test was performed using the aspects described in Section 5.1 (e.g., same UAV parameters, same takeoff position, and same desired trajectory) but using an optimized set of gains. The purpose is to compare and show the effects of the optimization process on the system’s behavior. To obtain the optimal set of gains, the NSGA-II optimization technique described in Section 4 was used to minimize the two objective functions, \( Ob_1 \) and \( Ob_2 \), defined by Equations (29) and (30). To perform such optimization, a population size equal to 30 was used while the optimization cycle was repeated for 100 generations. During the optimization process, crossover and mutation rates equal to 0.95 and 0.01, respectively, were used. The parameters (gains) optimized in this process included \( K_P, K_I, \) and \( K_D \) for each of the \( x, y, z, \phi, \theta \) and \( \psi \) parameters (18 gains in total), respectively (note that in this case, the integral gains for the three angular orientations were considered as zero). The range of decision variables for \( K_P, K_D \) was selected to be between 0 and 15, while the range for the gains \( K_I \) were selected to be between 0 and 1. This variable range selection was based on numerous simulation results used to eliminate reaching singularity points. The result of the optimization process produced the following \( k \) vector:

\[
k = [10.6, 0, 13.4, 0, 5.8, 0, 8.8, 10.8, 13.0, 4.3, 1.9, 3.6, 6, 0, 3.5, 8.4, 0.2, 8]
\]

Figure 9 shows the changes in the optimization cost functions before and after the optimization process for each state. Figure 9 shows the significant changes that took place in the objective functions for all states.

Figures 10–12 present the simulation result for position and orientation to track the desired trajectory and illustrate the obtained trajectory that the UAV generated. When compared to the first test (Section 5.1), the results using the optimized gains show faster trajectory tracking and, thus, better UAV performance. This is especially true in position tracking where the UAV converges to the desired path much faster and uses a smoother path.
Figure 9 shows the changes in the optimization cost functions before and after the optimization process for each state. Figure 9 shows the significant changes that took place in the objective functions for all states.

(a) Before optimization. (b) After optimization.

Figure 9. Objective values before and after optimization.

Figures 10–12 present the simulation result for position and orientation to track the desired trajectory and illustrate the obtained trajectory that the UAV generated. When compared to the first test (Section 5.1), the results using the optimized gains show faster trajectory tracking and, thus, better UAV performance. This is especially true in position tracking where the UAV converges to the desired path much faster and uses a smoother path.

(a) x position (b) y position (c) z position

Figure 10. Position response of the UAV when using optimal gains.

(a) roll (b) pitch (c) yaw

Figure 11. Orientation response of the UAV when using optimal gains.
Thus, in the tests conducted, the azimuth velocity as a function of the drone’s altitude: $\psi_0$ is the wind velocity at altitude $z$, $V_{\psi_0}$ is the specified (measured) wind velocity at altitude $z_0$, and $p$ is the index of the energetic wind profile. This formulation was used to compute the wind that the drone would experience as it changes altitude while tracking the desired trajectory. Furthermore, to better simulate the wind disturbances that the tiltrotor

Figure 12. Three-dimensional simulation result of the UAV’s path when using optimal gains.

Figures 10 and 11 illustrate the effect of the optimization on the controller results. Not only were the steady-state errors of all state variables reduced, but the average overshoot of the corresponding responses and the settling time of the system also showed significant improvements.

5.3. Wind Gust Disturbances

To show the effectiveness of the proposed controller to identify and control the aircraft under external disturbances, the drone was tested under diverse wind gust conditions showing the robustness of the proposed control mechanisms. To test such aspects, the tiltrotor bi-copter drone shown in Figure 1 was flown within an uncertain outdoor flying environment simulation. Within such a simulation environment, the aircraft was placed under moderate wind disturbances mimicking real wind conditions. Although the wind effect is typically considered a constant within the available literature (as it is very challenging to simulate the variability of real-world wind conditions), such wind effects do not reflect real flight conditions, and as a result, it leads to misleading conclusions about the performance of the tested control scheme. In contrast to such simplistic approaches, in this paper, the control scheme described in the previous sections was tested using a macro approach, which defines two main characteristics of typical wind: the magnitude and the direction of the wind velocity. To define the wind affecting the aircraft during the simulation tests, the description and wind formulations used in [37] were employed. Thus, in the tests conducted, the azimuth $\psi_w$ of the point from which the wind blows was used as a wind direction, where a random value $\Delta\psi_w$ was used to represent the random value of wind direction change. The azimuth is measured from the north point through the east. The velocity of the wind is also defined as a function of the $z$ (altitude) position of the UAV (drone), $V_z$ with respect to the sea level. Based on the results provided in [37], a moderate horizontal wind gust of 4 m/s was used to study the stability and robustness of the designed controller. The following formula was used to determine the average wind velocity as a function of the drone’s altitude:

$$V_z = V_{\psi_0} \left( \frac{z}{z_0} \right)^p$$
bi-copter drone would experience, the wind force that influences the aircraft was considered to be more representative than the actual instantaneous wind velocity. To compute such a force, the following formulation was used

$$F_w = S_e AV_z^2$$

where $S_e$ is the influence effective area, and $A = 1/16 \times 9.81 = 0.61$ is the rate of wind velocity (m/s$^2$) conversion to pressure (N/m$^2$). To compute the total force and its distribution over the drone’s fuselage, the tiltrotor bi-copter was represented as a cylindrical body. Even though this is a very simplistic approximation, it enabled us to test the robustness of the controller to external a priori unknown wind disturbances. The expression determining the bi-copter surface area was set to be equal to the sum of the lateral surface area, $S_l$, and the area of the base, $S_b$, as represented by the following mathematical expression where $r$ and $h$ are the radius and height of the cylinder representing the drone, and $a$ and $b$ are the fill factors of the cylinder area that depend on the tiltrotor bi-copter design.

$$S = S_l + S_b = a2\pi rh + b2\pi r^2$$

5.3.1. Nonlinear Dynamic Inversion Structure with Neural Network Augmentation

Because of the sensitivity of the dynamic inversion approach to the accuracy of the dynamic model, a sigma-pi neural network (NN) adaptive compensator is used to compensate for modeling errors [38]. Thus, the control scheme shown in Figure 5 is enhanced to include a NN adaptive compensator, as illustrated in Figure 13. The modeling error feeds the NN, which, in turn, generates an output to compensate for deviations due to the approximate NDI output. It is important to note that such NN is applied to each of the channels ($x$, $y$, and $z$), where each NN might have different weights.

![Figure 13. Designed controller with NN compensator.](image)

Considering the model error, the new model of the system can be defined as Equation (33):

$$\dot{x} = f(x, \dot{x}, u) + \Delta(x, \dot{x}, u)$$

(33)

where $\Delta(x, \dot{x}, u) = \dot{x} - f(x, \dot{x}, u)$ represents the model inversion error. The sigma-pi NN compensates the inversion error adaptively with a real-time pseudo control signal $v_{ad}$ where the pseudo control command is provided as per Equation (34):

$$v = v_{rm} + v_{pd} - v_{ad}$$

(34)

where $v_{rm} = \dot{x}$ is the reference model’s pseudo control, and $v_{pd}$ is the PD linear regulator used to regulate the dynamic response of the system as per Equation (35):

$$v_{pd} = k_p(x_{rm} - x) + k_d(\dot{x}_{rm} - \dot{x})$$

(35)
As a result, the dynamics of the model tracking error can then be described via Equation (36):

\[ \dot{e} = Ae + B(\nu_{ad} - \Delta) \]  

where \( e = \begin{bmatrix} x_{rm} - x \\ \dot{x}_{rm} - \dot{x} \end{bmatrix} \), 
\( A = \begin{bmatrix} 0 & 1 \\ -k_p & -k_d \end{bmatrix} \), and 
\( B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \).

5.3.2. Neural Network Design

The sigma-pi neural network (NN) acting as the adaptive compensator shown in Figure 13 can be made up of any linearly parameterized feedforward structure capable of reconstructing the inversion error, as depicted in Figure 14.

![Figure 14. The online adaptive sigma-pi neural network structure.](image)

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

\[ \nu_{ad} = \sum_{j=1}^{k} w_j \beta_j(\tilde{X}, \nu) = W^T \beta(\tilde{X}, \nu) \]  

where \( \tilde{X} \) represents the normalized states, \( W \) is a vector of variable networks weights, and \( \beta(c_1, c_2, c_3) = \text{kron}(\text{kron}(c_1, c_2), c_3) \) is a network basis function vector. The corresponding weight error can then be described as follows (Equation (38)):

\[ \tilde{W} = W - W^* \]  

where \( W^* \) denotes the ideal set of weights. Thus, the error vector can be written as per Equation (39), and the updated law for the NN can be expressed by Equation (40).

\[ \dot{e} = Ae + BW^T \beta + B(\nu_{ad} - \Delta) \]  

\[ \dot{\tilde{W}}_i(t) = \begin{cases} -\Gamma_i (e^T P_i B) \beta_i & \text{for } \| e_i \|_{P_i} > \epsilon_0, \\ 0 & \text{for } \| e_i \|_{P_i} < \epsilon_0, \end{cases} \quad i = 1, 2, \ldots, n \]  

where \( \| e_i \|_{P_i} = \sqrt{e^T_i P_i e_i} \), \( \Gamma_i \) is a positive constant adaptation gain (learning rate), and \( P_i \) is a symmetric, positive definite matrix satisfying the Lyapunov equation \( P_i A_i + A_i^T P_i = -I \) for the linear system error with NN weights. The Hurwitz matrix \( A_i \) of Equation (38) ensures the existence of a unique \( P_i = P_i^T > 0 \) where \( P_i \) is described as per Equation (41):

\[ P_i = \begin{bmatrix} K_p + \frac{1}{2K_p} & \frac{K_d}{2K_p} \\ \frac{K_d}{2K_p} & \frac{1}{K_d} \end{bmatrix} \]  

Figure 15 shows one of the many wind profiles used when testing the proposed controller. Under such wind conditions, the aircraft was commanded to take off and fly following the trajectory defined by the desired \( x, y, \) and \( z \) curves shown in Figure 16.
Figure 16 also shows the drone’s position in terms of the flight path achieved by the drone and its deviation from its intended path.

![Wind magnitude](image1)

![Wind direction](image2)

**Figure 15.** Simulated wind direction and velocity.

**Figure 16.** Simulated results under moderate wind gusts (position). During the tests, the position and orientation of the drone were recorded. Figure 17 shows the results of the drone’s orientation (i.e., roll, pitch, and yaw) compared to the desired orientation as defined by the desired flight path achieved by the drone. From Figure 17, it is clear to see that the proposed controller is effective in terms of the drone’s orientation, while the results for the x, y, and z positions are satisfactory. The results show the capability of the controller to deal with disturbances.
Figure 17. Simulated result under moderate wind gusts (orientation). To be able to compare the result of the proposed controller with previously developed backstepping controllers for the same system [13], the drone was also flown to follow the same trajectory under the same wind disturbance (Figure 15) but using the backstepping controller. Figures 18 and 19 show the obtained results. When comparing Figures 16 and 17 (proposed controller) with Figures 18 and 19 (backstepping controller), it is clear that the superiority of the new control architecture to handle disturbances even under inaccurate or partial knowledge of the system’s dynamic equations.

Figure 18. Position results of the backstepping controller under moderate wind gusts.
In this paper, a nonlinear mathematical model for a tiltrotor bi-copter UAV prototype is extracted. Then, a nonlinear trajectory controller based on nonlinear dynamic inversion, including an outer NDI loop controller for position and an inner INDI loop controller for orientation is proposed. The proposed control architecture is improved with the introduction of the NSGA-II-PID controller. Due to the significant cost in time and effort required to repetitively perform manual tuning of the linear gains using typical control tuning approaches, in this paper, a multi-objective-gain optimization process based on the NSGA-II algorithm is used to minimize the two error functions. By combining these two controllers with a multi-objective optimization process, faster and more accurate flight control is achieved. The proposed controller does not require a complete and accurate dynamic model of the system. The performance of the control system was tested over diverse flight paths, including flight tests under moderate wind gusts.

The obtained results of the proposed controller, implementing gain matrix \( k = [10.6, 0, 13.9, 5.8, 0, 8.8, 10.8, 0, 13.0, 4.3, 0, 3.6, 6, 0, 3.5, 8.4, 0, 8] \) for the PID control blocks, were compared with results obtained using a previously designed backstepping controller, using a gain matrix \( \alpha \) equal to \( [15, 0.1, 15, 0.1, 15, 0.1, 20, 1, 2, 1, 1, 1] \). From the obtained results, it is concluded that the proposed control system design performs better than previously developed controllers (e.g., the backstepping controller, which requires accurate equations of motion) and provides better results than the classic dynamic inversion type of controllers presented in the available literature. These results are anticipated to be very helpful for the development of a new concept of bi-copter aircraft using diverse mechanisms, including variable pitch propeller systems and other architectures that will enable such eVTOL aircraft to perform missions in civil and military applications and overcome aggressive disturbances where current UAV systems cannot be used.

As described in this paper, the proposed method is an offline method. However, it could be used online if the needed time window for optimization is calculated and guaranteed prior to the actual flight.

**Author Contributions:** Conceptualization, A.R.-S.; Methodology, M.T.; Validation, M.T.; Resources, A.A.; Writing, M.T. and A.R.-S.; Supervision, A.R.-S.; Funding Acquisition, A.R.-S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Science and Engineering Research Council of Canada (NSERC) under grant number [RGPIN-2021-02441].

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


