IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 23 September 2022, accepted 6 October 2022, date of publication 14 October 2022, date of current version 24 October 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3214810

RESEARCH ARTICLE

Improved Performance in Quadrotor Trajectory Tracking Using MIMO PI$^{\lambda}$-D Control

VIVEK PAWAN SHANKARAN[1], SHEIKH IZZAL AZID[1], (Member, IEEE), UTKAL MEHTA[1], (Senior Member, IEEE), AND ADRIANO FAGIOLINI[2], (Member, IEEE)

[1]School of Information Technology, Engineering, Mathematics and Physics, The University of the South Pacific, Laucala Bay, Suva, Fiji
[2]Mobile & Intelligent Robots@Panormous Laboratory (MIRPALab), Department of Engineering, University of Palermo, 90128 Palermo, Italy

Corresponding author: Adriano Fagiolini (adriano.fagiolini@unipa.it)

This work was supported by The University of the South Pacific with M.Sc. Engineering Grant.

ABSTRACT This paper aims to develop a fractional control approach for quadrotor trajectory tracking. A fractional-order integrator (PI$^{\lambda}$) with a feedback derivative scheme is designed to control each state of the MIMO system. The designed feedback controller stabilizes the initially unstable decoupled states and widens the stability, while PI$^{\lambda}$ provides precise trajectory tracking capabilities. After a successful simulation study, the new PI$^{\lambda}$-D controller is implemented in the hardware environment. The various performance and load disturbance analyses reveal the effectiveness of the proposed scheme compared with the classical PD/PID controllers. The real-time study also shows that this scheme is a simple yet robust solution.

INDEX TERMS Fractional-order control, MIMO system, quadrotor, trajectory tracking, real-time, optimization, stability boundary surface.

I. INTRODUCTION

The first generation of aerial vehicles were known to be big, heavy, expensive and manually operated only. However, these are fascinating times for aerial robotics. The recent cutting-edge research and developments in the field of robotics have produced improvements in microcontroller technology and sensor technology [1]. Moreover, significant progress has also advanced the aerodynamic and control theory [2]. Undoubtedly, these improvements have not only paved the existence of unmanned aerial vehicles (UAV) or drones but have also made them much lighter, smaller and inexpensive [1], [2], [3]. A drone in the UAV context is known as an aircraft without a pilot or remotely controlled. This dramatic improvement in drone technology has resulted in its growing application in various contrasting sectors. Today, drones are commercially available for applications such as photography, agriculture, film production, surveillance, disaster management, and monitoring [4], [5], [6], [7]. Among the UAVs, the quadrotor is most commonly adopted for the above-mentioned applications. Its reliable manoeuvrability and affordable price point due to its simple yet effective production make them an attractive choice for these applications [8].

Although quadrotors have undergone a series of significant upgrades throughout the years, researchers still confront challenges in perfecting their control. The coupled translational and rotational dynamics, underactuated, and gyroscopic and aerodynamic effects of the aircraft constantly pose an obscure hindrance to administering flawless control. For this reason, adopting and frequently improving advanced control strategies become compulsory to achieve perfect control. In [9], a complete guide for a closed loop control of a quadrotor is discussed.

Numerous control strategies have been applied to control the quadrotor autonomously. Famously known linear and nonlinear control schemes, like PD, PID, sliding mode control, LQR and Backstepping control techniques, have all been applied for quadrotor control [10]. Furthermore, the quality of the PID controller has been discussed by researchers, such as in [11]. The likes of cascaded PID with the classical PID control were also compared [12]. The study revealed that the main drawback of PID controllers was that they are susceptible to significant disturbances, which compromised their robustness. However, a nonlinear PID controller was introduced in [13] that could compensate for disturbances.

The associate editor coordinating the review of this manuscript and approving it for publication was Bidyadhar Subudhi.

On the other hand, the PD controller has been a favourite choice for control due to its simplicity [14]. Again, this controller action is very sensitive to input disturbances and parameter perturbations. It has been used several times to design hybrid control schemes [15], [16]. In [15], the PD controller is implemented with sliding mode control and in [17], a hybrid PD-fuzzy controller was developed for autopilot applications. In addition, sliding mode control proposed for quadcopters in [18] concluded that SMC has an advantage in rejecting disturbances. Moreover, [19] solved the issue of disturbances by designing a model-based controller for attitude control while the PID was used for position control with state estimations. The cuckoo algorithm tuned the controller gains, and a power reduction methodology was added to the design. The Lyapunov function shows the trajectory design procedure and a controller-gains tuning stage whose stability analysis. The performance index depends on the tracking error and the quadrotor inputs to reduce the power consumption of the quadrotor [20]. [21] designed the robust system with the unknown input observer and PD-based precise trajectory control of quadrotor in the presence of wind gusts.

The research on quadrotor control has seen a tremendous amount of advances recently. To begin with, model predictive control, also known as MPC has been applied to altitude control of the quadcopter [22]. Moreover, an adaptive nonlinear tracking controller was developed by [23] and [24] in 2019. In addition to this, improvements in sliding mode control were also done [25], [26]. Then an interesting study was published by [27] in 2019 where a neural network-based quadcopter UAV system was introduced. The study showed a decrease in trajectory tracking errors when comparing neural network-based control with PID. Note that the advanced controlling schemes are highly dependent on immense computational power and are complex to implement in real-time.

In 1994, Podlubny [28] proposed a generalised version of the PID controller. It was called the PI$^{\lambda}$D$^{\mu}$ controller having not three but five tuning parameters. Since then, the design of the PI$^{\lambda}$ controller and the PD$^{\mu}$ controller has also been studied [29], [30], [31]. Today, fractional control strategies have been applied for systems like obstacle avoidance robots, controlling flexible joints for robots and controlling electro hydraulic systems [32]. The growing interest in fractional PID controllers has been fueled by the fact that they have additional ''tuning knobs'' that allow intelligible adjustment of the dynamics of control systems, and it is a revolution towards industrialization [33], [34].

Recent studies present fractional control has been extensively linked with sliding mode control for applying quadrotor control [35], [36], [37]. In [35], an adaptive fractional order sliding mode control was designed for a quadrotor with varying loads. Similarly, in another study conducted by [38], fractional order sliding mode control was applied for altitude and attitude stabilization of quadrotors. Fractional order control was applied for a robust position and yaw angle tracking for quadrotors in [39]. Furthermore, research on a fractional PD path following control was also done by [40]. The study

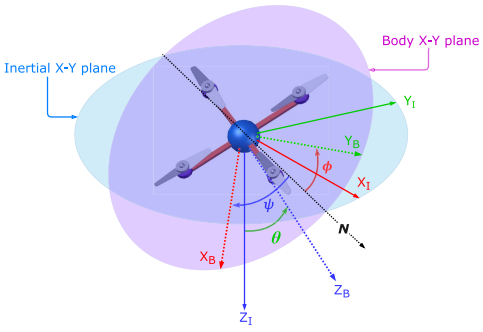


FIGURE 1. Quadrotor with reference frames and their transformation by tait–bryan angles.

compared the performance of an integer order PD controller, a model predictive control strategy with the fractional order PD controller, with the fractional PD controller exhibiting superior results. On the other hand, a fractional nonlinear PI structured control has been implemented for attitude tracking of quadrotors [41]. This study shows that the nonlinear PI structured fractional controller yielded faster responses than the classical methods. The feedback controller-based structure using fractional-order PI was presented recently for the altitude control in a quadrotor [42]. The improved performance was observed again from the fractional-order approach.

The significant contribution in this paper can be further distinguished as below.

- A simple yet robust control solution is discussed instead of a complex or nonlinear method using fractional-order theory.
- The method proposed in this article is adaptable and feasible as per the result.
- The proposed design improves the stability region of the quadrotor using explicit formulae derived. This can help to adopt any optimization routine easily to check optimal performances on the system.
- The numerical study shows much superior performance than previous controllers in the tracking and faster regulation capabilities, even when disturbances affect the system.
- A real-time implementation is examined with five way-point and helical trajectories.

The remainder of the paper is as follows. Section II devotes to deriving six decoupled state transfer functions for controller design. Section III presents the PI$^{\lambda}$-D controller design and tuning method. The simulations and hardware experimentation are provided in Sections IV and V, respectively. Finally, some essential conclusions are drawn in Section VI.

II. DYNAMICS OF THE QUADROTOR

A. MATHEMATICAL MODEL

Assuming the origins of inertial reference frame and the centre of mass of the quadrotor are concentric as shown in Fig. 1, the orientation of the quadrotor could be expressed using the so called Tait–Bryan angles [43] expressed by the

rotational matrix $\mathbf{R}$ as (1), shown at the bottom of the page, where sine is denoted as S and cosine is denoted as C. The linear dynamics of the quadrotor taking account of the forces $\boldsymbol{f_B} = \begin{bmatrix} f_x & f_y & f_z \end{bmatrix}^T \in \mathbb{R}^3$ acting on the aircraft can therefore be described as expression (2).

$$m\,\dot{\boldsymbol{\nu}} = \mathbf{R} \cdot \boldsymbol{f_B} = mg\,\hat{\mathbf{e}}_{\mathbf{z}} - f_t \mathbf{R} \cdot \hat{\mathbf{e}}_{\mathbf{3}} \tag{2}$$

where $\dot{\boldsymbol{\nu}} = \begin{bmatrix} \ddot{x} & \ddot{y} & \ddot{z} \end{bmatrix}^T \in \mathbb{R}^3$,$\hat{\mathbf{e}}_{\mathbf{z}}$ is the unit vector associated with the z-axis in the inertial frame and $\hat{\mathbf{e}}_{\mathbf{3}}$ is the unit vector associated with the z-axis in the body frame. The mass of the quadrotor is denoted by m, g the gravitational acceleration and f_t is the thrust force generated by the quadrotor. Similarly, the angular dynamics of the quadrotor taking into consideration the torques $\boldsymbol{\tau_B} = \begin{bmatrix} \tau_x & \tau_y & \tau_z \end{bmatrix}^T \in \mathbb{R}^3$ acting on the aircraft can therefore be described as expression (3).

$$\dot{\boldsymbol{\omega}}_{\boldsymbol{B}} = \mathbf{I}^{-1} \cdot \boldsymbol{\tau_B} - \boldsymbol{\omega_B} \wedge (\mathbf{I} \cdot \boldsymbol{\omega_B}), \tag{3}$$

where $\boldsymbol{\omega_B} = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T \in \mathbb{R}^3$ and $\mathbf{I}$ is the diagonal inertia matrix given as:

$$\mathbf{I} = \begin{bmatrix} I_x & 0 & 0 \\ 0 & I_y & 0 \\ 0 & 0 & I_z \end{bmatrix} \in \mathbb{R}^{3\times 3}.$$

Therefore, the total dynamical model of the quadrotor combining its linear and angular dynamics results in:

$$\begin{cases} \ddot{x} = -\dfrac{f_t}{m}[S\phi S\psi + C\phi S\theta C\psi] \\ \ddot{y} = -\dfrac{f_t}{m}[-S\phi C\psi + C\phi S\theta S\psi] \\ \ddot{z} = g - \dfrac{f_t}{m}[C\phi C\theta] \\ \ddot{\phi} = \dfrac{I_y - I_z}{I_x}\dot{\theta}\dot{\psi} + \dfrac{\tau_x}{Ix} \\ \ddot{\theta} = \dfrac{I_z - I_x}{I_y}\dot{\phi}\dot{\psi} + \dfrac{\tau_y}{Iy} \\ \ddot{\psi} = \dfrac{I_x - I_y}{I_z}\dot{\phi}\dot{\theta} + \dfrac{\tau_z}{Iz} \end{cases} \tag{4}$$

B. QUADROTOR ACTUATOR DYNAMICS

Having denoted f_t as the thrust force generated by the four rotors and defined the two vectors of rotor speeds and total torque generated by differences in the rotor speeds, respectively, as

$$\boldsymbol{\Omega} = \begin{bmatrix} \Omega_1 & \Omega_2 & \Omega_3 & \Omega_4 \end{bmatrix}^T \in \mathbb{R}^4 \tag{5}$$

$$\boldsymbol{\tau_B} = \begin{bmatrix} \tau_x & \tau_y & \tau_z \end{bmatrix}^T \in \mathbb{R}^3 \tag{6}$$

the equations describing how rotor speeds have an influence on the quadrotor can be formulated as:

$$\begin{cases} f_t = k_f(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \\ \tau_x = \dfrac{k_f l}{2}\big((\Omega_1^2 + \Omega_4^2) - (\Omega_2^2 + \Omega_3^2)\big) \\ \tau_y = \dfrac{k_f l}{2}\big((\Omega_1^2 + \Omega_2^2) - (\Omega_3^2 + \Omega_4^2)\big) \\ \tau_z = k_m(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{cases} \tag{7}$$

where k_f is motor force constant, l is the length from the centre of the quadrotor to a particular rotor and k_m is the motor torque constant of the quadrotor affecting the yawing action.

III. QUADROTOR ARCHITECTURE AND PI$^{\lambda}$-D CONTROLLERS

A. QUADROTOR CASCADED CONTROL ARCHITECTURE

From the dynamical modeling for control design in (4), it is known that the x position is coupled with the pitch angle and the y position is coupled with the roll angle. This is intuitively true since the quadrotor itself cannot move in the $x-y$ plane without either pitching or rolling, respectively. Following (4) and assuming the quadrotor is at a hovering state, concurring the fact that the thrust force is equal to the gravitational force, the x and y dynamics can be written as (8). It is also assumed that only the roll and pitch angles are small and so, the sine functions of roll and pitch are approximated to its argument and cosines are approximated to 1.

$$\begin{cases} \ddot{x} = -g[\phi S\psi + \theta C\psi] \\ \ddot{y} = -g[-\phi C\psi + \theta S\psi] \end{cases} \tag{8}$$

Rewriting in matrix form:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} -g & 0 \\ 0 & -g \end{bmatrix} \begin{bmatrix} S\psi & C\psi \\ -C\psi & S\psi \end{bmatrix} \begin{bmatrix} \phi \\ \theta \end{bmatrix} \tag{9}$$

Now, solving for ϕ and θ one gets the following matrices:

$$\begin{bmatrix} \phi \\ \theta \end{bmatrix} = \begin{bmatrix} -\frac{1}{g} & 0 \\ 0 & -\frac{1}{g} \end{bmatrix} \begin{bmatrix} S\psi & -C\psi \\ C\psi & S\psi \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{y} \end{bmatrix} \tag{10}$$

where

$$\begin{bmatrix} S\psi & -C\psi \\ C\psi & S\psi \end{bmatrix} \in \mathbb{R}^{2\times 2}$$

is the transformation matrix that transforms the x-y coordinates from the inertial reference frame to the body reference frame. Following this scheme, the $x-y$ position controls become cascaded with the roll-pitch angle controls. The altitude and the yaw angle control on the other hand can be controlled in an independent loop. Referring back to (7), the

$$\mathbf{R} = \begin{bmatrix} C(\theta)C(\psi) & S(\phi)S(\theta)C(\psi) - C(\phi)S(\psi) & C(\phi)S(\theta)C(\psi) + S(\phi)S(\psi) \\ C(\theta)S(\psi) & S(\phi)S(\theta)S(\psi) + C(\phi)C(\psi) & C(\phi)S(\theta)S(\psi) - S(\phi)C(\psi) \\ -S(\theta) & S(\phi)C(\theta) & C(\phi)C(\theta) \end{bmatrix} \tag{1}$$

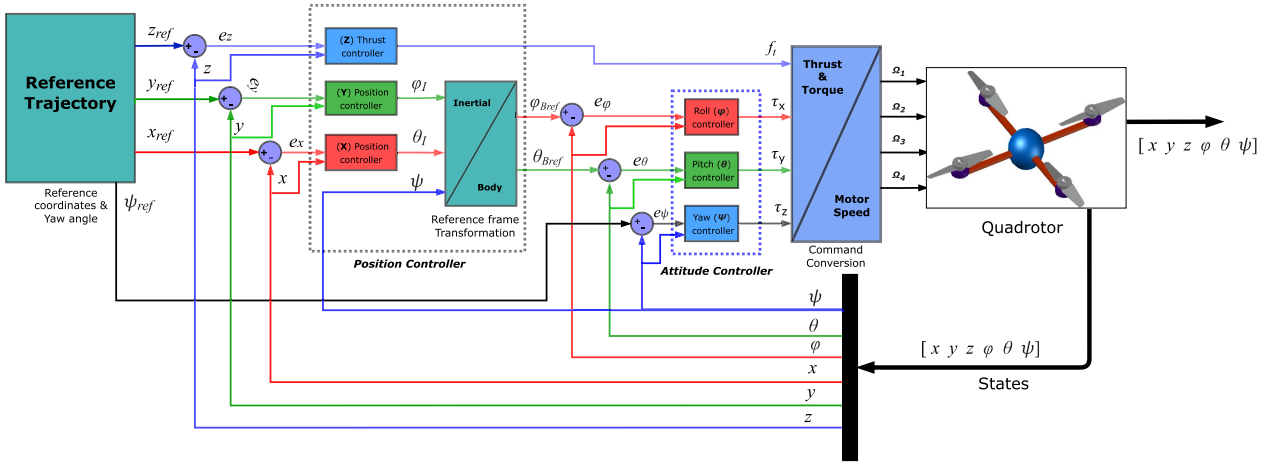


FIGURE 2. Quadrotor control scheme.

motor speeds could be mapped to the quadrotor for its inputs. Writing (7) in matrix form:

$$\begin{bmatrix} f_t \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} k_f & k_f & k_f & k_f \\ \frac{k_f l}{2} & -\frac{k_f l}{2} & -\frac{k_f l}{2} & \frac{k_f l}{2} \\ \frac{k_f l}{2} & \frac{k_f l}{2} & -\frac{k_f l}{2} & -\frac{k_f l}{2} \\ k_m & -k_m & k_m & -k_m \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{11}$$

Remark: To note that the 4×4 matrix in (11) can only be invertible if and only if values are positive. As per Table 6 of minidrone hardware parameters, it can be ensured for any quadrotor, $(k_f, l, k_m) > 0$. Furthermore (11) can be written for rotor speeds as,

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4k_f} & \frac{1}{2k_f l} & \frac{1}{2k_f l} & \frac{1}{4k_m} \\ \frac{1}{4k_f} & -\frac{1}{2k_f l} & \frac{1}{2k_f l} & -\frac{1}{4k_m} \\ \frac{1}{4k_f} & -\frac{1}{2k_f l} & -\frac{1}{2k_f l} & \frac{1}{4k_m} \\ \frac{1}{4k_f} & \frac{1}{2k_f l} & -\frac{1}{2k_f l} & -\frac{1}{4k_m} \end{bmatrix} \begin{bmatrix} f_t \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \tag{12}$$

To obtain the individual rotor speed of the quadrotor, (13) is to be applied so it follows its appropriate rotations.

$$\begin{bmatrix} \Omega_1 \\ \Omega_2 \\ \Omega_3 \\ \Omega_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} \sqrt{|\Omega_1^2|} \\ \sqrt{|\Omega_2^2|} \\ \sqrt{|\Omega_3^2|} \\ \sqrt{|\Omega_4^2|} \end{bmatrix} \tag{13}$$

B. MODEL TRANSFER FUNCTIONS FOR CONTROLLER DESIGN

Knowing the dynamic modelling of the quadrotor with six critical states for control, their transfer functions could be derived to individually control each state without affecting the other. To decouple these six critical states from each other, some assumptions were made.

1) *Assumption 1*: The quadrotor is assumed to be in a hovering state this implies that orientations ϕ and $\theta \approx 0$ in deriving the attitude transfer function. Thus, the system is fully controllable, which are 4 inputs and 4 outputs.
2) *Assumption 2*: For x and y position transfer functions, the quadrotor is also assumed to be in a hovering state and the roll and pitch angles are approximated to 0 for sine and 1 for cosine, using small angle approximations.
3) *Assumption 3*: Angular velocities $\dot{\phi}$ and $\dot{\theta} \approx 0$ when deriving position attitude functions.

Under the above assumptions, (4) can be rewritten in decoupled form as:

$$\begin{cases} \ddot{x} = \frac{f_t}{m}\theta \\ \ddot{y} = \frac{f_t}{m}\phi \\ \ddot{z} = \frac{f_t}{m} \\ \ddot{\phi} = \frac{\tau_x}{Ix} \\ \ddot{\theta} = \frac{\tau_y}{Iy} \\ \ddot{\psi} = \frac{\tau_z}{Iz} \end{cases} \tag{14}$$

Taking the Laplace transform of the second order deferential equations in (14), the transfer functions of six states with respect to their inputs and outputs are

$$\begin{cases} \frac{X}{\theta} = \frac{\frac{f_t}{m}}{s^2} \\ \frac{Y}{\phi} = \frac{\frac{f_t}{m}}{s^2} \\ \frac{Z}{ft} = \frac{\frac{1}{m}}{s^2} \\ \frac{\phi}{\tau_x} = \frac{\frac{1}{Ix}}{s^2} \\ \frac{\theta}{\tau_y} = \frac{\frac{1}{Iy}}{s^2} \\ \frac{\psi}{\tau_z} = \frac{\frac{1}{Iz}}{s^2} \end{cases} \tag{15}$$

C. PI$^{\lambda}$-D CONTROL

The dynamics of the quadrotor position and attitude control can be further modelled using the form of second order integrating model as given in (16)

$$G_p(s) = \frac{ke^{-\theta_t s}}{s^2} \tag{16}$$

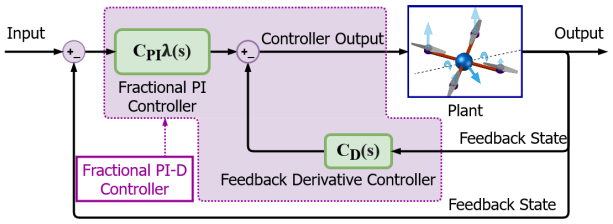


FIGURE 3. The proposed control scheme.

where $e^{-\theta_t s}$ is a small time delay term introduced to the system which has only been employed for the purpose of controller design. The PI$^{\lambda}$ and the inner loop derivative controller are defined as

$$C_{PI^{\lambda}}(s) = K_p + \frac{K_i}{s^{\lambda}} \tag{17}$$

$$C_D(s) = \frac{T_{df}\, s}{\eta T_{df}\, s + 1} \tag{18}$$

where K_p is the proportional gain, K_i is the integral gain, λ is the real power of the integrator in PI$^{\lambda}$ ($0 < \lambda < 2, \lambda \in \mathbb{R}^{+}$). Furthermore, T_{df} is the derivative time constant, and η is the derivative filter constant. The value of η, compared to T_{df}, is significantly minimal and, therefore, can be assumed to be zero for ease in the controller design. The proposed control structure can be represented in a simplified way as in Figure 3. In the proposed tuning method, the inner loop derivative controller is set up as per the guideline given below. Then, the forward PI$^{\lambda}$ is designed with specific boundary conditions.

1) INNER LOOP D-CONTROLLER GAIN

Utilizing the first order Pade approximation, (16) could be written as

$$G_p(s) = \frac{k}{s^2(1 + \theta_t s)} \tag{19}$$

Computing (19) and (18) as a closed loop system,

$$G_p(s) = \frac{k}{s^2(1 + \theta_t s) + kT_{df}\, s} \tag{20}$$

The characteristic equation from (20) gives the information that there are three poles in the system, one of which is present in the origin of the root locus plane. The locations for the other two poles can be given as

$$\begin{aligned} \mathfrak{Re}(s_{(1,2)}) &= -\frac{1}{2\theta_t} \\ \mathfrak{Im}(s_{(1,2)}) &= \pm\frac{\sqrt{1 - 4\theta_t kT_{df}}}{2\theta_t} \end{aligned} \tag{21}$$

In order to design the system for a good transient performance, an assumption is made to place the other two imaginary poles where the imaginary values of the poles do exceed its real values. Therefore,

$$\sqrt{3}\frac{1}{2\theta_t} = \pm\frac{\sqrt{1 - 4\theta_t kT_{df}}}{2\theta_t} \tag{22}$$

Solving for T_{df} using (22), a range is obtained which is written as

$$0 < T_{df} < \frac{1}{\theta_t k} \tag{23}$$

2) PI$^{\lambda}$ CONTROLLER's STABILITY BOUNDARIES

To find the stability boundaries of PI$^{\lambda}$ upon adding the inner loop controller C_D, the closed-loop transfer function together with PI^{λ} is defined as below.

$$\frac{Y(s)}{R(s)} = \frac{k(K_p + \frac{K_i}{s^{\lambda}})}{\theta_t s^3 + s^2 + kT_{df}\, s + (k)(K_p + \frac{K_i}{s^{\lambda}})} \tag{24}$$

Then, the characteristic equation of the system becomes,

$$P(s) = k\left(K_p + \frac{K_i}{s^{\lambda}}\right) + \theta_t s^3 + s^2 + kT_{df}\, s \tag{25}$$

Now, the system's stability and instability domains can be further studied by examining the boundaries [44], [45]. Let us define three stability boundaries; the real root boundary (RRB), the complex root boundary (CRB) and the infinite root boundary (IRB). The RRB can be obtained by substituting $s = 0$ in (25). Similarly, the CRB can be obtained by substituting $s = j\omega$ and the IRB by $s = j\infty$.

From the three above mentioned boundaries, the CRB is implemented to find the stability boundaries since it sweeps all possible solutions for s from the range $0 < s < \infty$. Therefore substituting $j\omega$ in place of s gives

$$P(s) = k\left(K_p + \frac{K_i}{(j\omega)^{\lambda}}\right) + \theta_t(j\omega)^3 + (j\omega)^2 + kT_{df}(j\omega) \tag{26}$$

Letting the real and imaginary values of a complex number, $(j\omega)^{\lambda}$, equal to ϵ and σ, respectively, (26) can be written as

$$P(s) = k\left(K_p + \frac{K_i}{\epsilon + j\sigma}\right) + \theta_t(j\omega)^3 + (j\omega)^2 + kT_{df}(j\omega) \tag{27}$$

Equating the real and imaginary parts of (27) to zero provides

$$\begin{cases} kK_p + \frac{kK_i\epsilon}{\epsilon^2 - \sigma^2} - (\omega)^2 = 0 \\ -\frac{kK_i\sigma}{\epsilon^2 - \sigma^2} - \theta_t(\omega)^3 + kT_{df}(\omega) = 0 \end{cases} \tag{28}$$

After solving (28), the expressions of K_p and K_i for a particular λ value can be defined as

$$\begin{cases} K_p = \frac{\omega^2\sigma + (kT_{df}\omega^3 - \theta_t\omega)\epsilon}{k\sigma} \\ K_i = \frac{\theta_t\omega^3 - (kT_{df}\omega)(\epsilon^2 - \sigma^2)}{-k\sigma} \end{cases} \tag{29}$$

Above explicit relation in (29), one can construct the CRB of any SOI plant in order to identify its stability region effectively. Changing ω from 0 to ∞ creates a stability boundary root locus in the $K_p - K_i$ plane for a particular λ value. Here, the stability region helps to obtain the controller parameters optimally together λ. In following section, the parameter vectors are calculated together with a fractional-order degree. The possible stable values of controller parameters will help to put the constraint on optimization to estimate the final optimal set.

D. CONTROLLER PARAMETER OPTIMIZATION

To propose any optimization algorithm, firstly some parameter vectors must be introduced to optimize all of the decoupled states for the MIMO system. Considering the parameter vectors,

$$\begin{cases} \vec{\rho}_1 = [K_{px}, K_{ix}, \lambda_x, T_{dfx}]^T, & \vec{\rho}_1 \in \mathscr{K} \\ \vec{\rho}_2 = [K_{py}, K_{iy}, \lambda_2, T_{dfy}]^T, & \vec{\rho}_2 \in \mathscr{K} \\ \vec{\rho}_3 = [K_{pz}, K_{iz}, \lambda_3, T_{dfz}]^T, & \vec{\rho}_3 \in \mathscr{K} \\ \vec{\rho}_4 = [K_{p\phi}, K_{i\phi}, \lambda_\phi, T_{df\phi}]^T, & \vec{\rho}_4 \in \mathscr{K} \\ \vec{\rho}_5 = [K_{p\theta}, K_{i\theta}, \lambda_\theta, T_{df\theta}]^T, & \vec{\rho}_5 \in \mathscr{K} \\ \vec{\rho}_6 = [K_{p\psi}, K_{i\psi}, \lambda_\psi, T_{df\psi}]^T, & \vec{\rho}_6 \in \mathscr{K} \end{cases} \tag{30}$$

where $\vec{\rho}_1$, $\vec{\rho}_2$ and $\vec{\rho}_3$ are parameter vectors for x,y and z controller, respectively. Similarly, $\vec{\rho}_4$, $\vec{\rho}_5$ and $\vec{\rho}_6$ are parameter vectors for ϕ, θ and ψ, respectively, and $\mathscr{K}$ is the total solution space. In this work, the fitness function that is chosen to minimize the input error of the controller is defined as the integral of time multiplied by the absolute of the error (ITAE). It can be defined as,

$$ITAE = \min_{\vec{\rho}_i \in \mathscr{K}} \sum_{k=0}^{N} |k(r(k) - y(k))| \tag{31}$$

where $y(k)$ and $r(k)$ are the response and setpoint input, respectively, at time $t = t_k$ with N data points. The function *ga* from the *Matlab*® Optimization Toolbox is utilized to solve the optimization problem of (31) and can obtain the optimal set $\vec{\rho}_i$.

Remark: Before commencing the optimization algorithm, initial parameters are required to be given to the algorithm to aid in finding an optimal solution faster. The initial parameter vector $\vec{\rho}_i = [\Phi_1, \Phi_2, \Phi_3, \Phi_4]$ is used to initiate the first set of solutions. Here Φ_1, Φ_2, Φ_3 and Φ_4 are random initial values of K_p, K_i, λ and T_{df}, respectively, defined specifically within the solution space of $\mathscr{K}$.

The design steps can be summarized as follows

Step 1: Compute the feedback derivative gain T_{df}

Step 2: From the gain range obtained from Step 1, the CRB curves are constructed using the maximum value of T_{df} so that the total solution space defined by $\mathscr{K}$ is covered. The CRB is then constructed by using (29). The boundaries obtained are used to determine the optimization constrains $\mathscr{K}$.

Step 3: The GA is set up and initialized with random values in $\vec{\rho}$ to find an optimal solution from the search space defined in $\mathscr{K}$.

E. QUADROTOR CONTROLLER TUNING

The transfer functions of the six states derived by substituting the parameters in Table 6 into (15). Before applying the proposed tuning, it is important to derive the mathematical model as per the minidrone's parameters. The time delay is associated with sampling in digital micro-controllers and processing time. From the simulation model and actual output fitting, the position control model has a time delay of 0.05s,

Algorithm 1 Psuedocode for GA

```
Initalize parameters
Set identified optimization constrains
while ITAE > MinFitness do
    Fitness calculation
    Selection
    Crossover
    Mutation
end while
Decode optimized parameters with minimum fitness
return best found solution
```

while the attitude control model has a time delay of 0.01s. Since the attitude dynamics are faster, the transfer functions used specifically for tuning are presented in (32).

$$\begin{cases} \frac{X}{\theta} = \frac{9.98e^{-0.05}}{s^2} \\ \frac{Y}{\phi} = \frac{9.98e^{-0.05}}{s^2} \\ \frac{Z}{ft} = \frac{14.39e^{-0.05}}{s^2} \\ \frac{\phi}{\tau_x} = \frac{14577e^{-0.01}}{s^2} \\ \frac{\theta}{\tau_y} = \frac{10870e^{-0.01}}{s^2} \\ \frac{\psi}{\tau_z} = \frac{7320e^{-0.01}}{s^2} \end{cases} \tag{32}$$

Then, the upper bounds for the T_{df} controllers for each state were calculated. The maximum gain for T_{df} for each state are given in (33).

$$\begin{cases} T_{dfx} = 1.6989 \\ T_{dfy} = 1.6989 \\ T_{dfz} = 1.3900 \\ T_{df\phi} = 0.0017 \\ T_{df\theta} = 0.0023 \\ T_{df\psi} = 0.0034 \end{cases} \tag{33}$$

The equations to obtain K_p and K_i values to construct the CRB curves are derived from (15), with employing the parameters in Table 6 and (32). The K_p and K_i equations for the position controllers are given in (34) to (35)

$$\begin{bmatrix} K_{px} \\ K_{py} \\ K_{pz} \end{bmatrix} = \begin{bmatrix} \omega^2 & 0 & 0 \\ 0 & \omega^2 & 0 \\ 0 & 0 & \omega^2 \end{bmatrix} \begin{bmatrix} \frac{1}{g} \\ \frac{1}{g} \\ m \end{bmatrix} + \begin{bmatrix} \frac{\omega^3\epsilon}{\sigma} & 0 & 0 \\ 0 & \frac{\omega^3\epsilon}{\sigma} & 0 \\ 0 & 0 & \frac{\omega^3\epsilon}{\sigma} \end{bmatrix} \begin{bmatrix} \frac{\theta_{tp}}{g} \\ \frac{\theta_{tp}}{g} \\ m\theta_{tp} \end{bmatrix} - \begin{bmatrix} \frac{\omega\epsilon}{\sigma} & 0 & 0 \\ 0 & \frac{\omega\epsilon}{\sigma} & 0 \\ 0 & 0 & \frac{\omega\epsilon}{\sigma} \end{bmatrix} \begin{bmatrix} \frac{T_{dfx}}{g} \\ \frac{T_{dfy}}{g} \\ mT_{dfz} \end{bmatrix} \tag{34}$$

$$\begin{bmatrix} K_{ix} \\ K_{iy} \\ K_{iz} \end{bmatrix} = \begin{bmatrix} \frac{\omega\epsilon^2}{\sigma} & 0 & 0 \\ 0 & \frac{\omega\epsilon^2}{\sigma} & \\ 0 & 0 & \frac{\omega\epsilon^2}{\sigma} \end{bmatrix} \begin{bmatrix} T_{dfx} \\ T_{dfy} \\ T_{dfz} \end{bmatrix} - \begin{bmatrix} \omega\sigma & 0 & 0 \\ 0 & \omega\sigma & 0 \\ 0 & 0 & \omega\sigma \end{bmatrix} \begin{bmatrix} T_{dfx} \\ T_{dfy} \\ T_{dfz} \end{bmatrix} - \begin{bmatrix} \frac{\omega^3}{\sigma} & 0 & 0 \\ 0 & \frac{\omega^3}{\sigma} & 0 \\ 0 & 0 & \frac{\omega^3}{\sigma} \end{bmatrix} \begin{bmatrix} \frac{\theta_{tp}}{g} \\ \frac{\theta_{tp}}{g} \\ m\theta_{tp} \end{bmatrix} \tag{35}$$

TABLE 1. Identified controller parameters constrains.

Control Component	Lower Bound				Upper Bound			
	K_p	K_i	λ	T_{df}	K_p	K_i	λ	T_{df}
x position	0	0	0	0	36.637	36.172	0.500	1.699
y position	0	0	0	0	36.637	36.172	0.500	1.699
z position	0	0	0	0	29.974	29.595	0.500	1.390
Roll angle	0	0	0	0	0.046	0.045	0.500	0.002
Pitch angle	0	0	0	0	0.062	0.060	0.500	0.002
Yaw angle	0	0	0	0	0.092	0.089	0.500	0.003

TABLE 2. Optimized controller parameters.

Controller Type	Controller Parameters			
	K_p	K_i	λ	T_{df}
z position	4.002	0.062	0.124	0.832
x position	3.959	0.055	0.120	1.225
y position	3.959	0.055	0.120	1.225
Roll angle	0.026	0.001	0.129	0.001
Pitch angle	0.039	0.001	0.121	0.002
Yaw angle	0.028	0.001	0.121	0.003

where ϵ and σ are the real and imaginary parts of $j\omega^{\lambda}$, respectively, and $\theta_{tp} = 0.05$. Similarly, the K_p and K_i equations that were derived for the attitude controllers are given in (36) to (37)

$$\begin{bmatrix} K_{p\phi} \\ K_{p\theta} \\ K_{p\psi} \end{bmatrix} = \begin{bmatrix} \omega^2 & 0 & 0 \\ 0 & \omega^2 & 0 \\ 0 & 0 & \omega^2 \end{bmatrix} \begin{bmatrix} I_x \\ I_y \\ I_z \end{bmatrix} + \begin{bmatrix} \frac{\omega^3\epsilon}{\sigma} & 0 & 0 \\ 0 & \frac{\omega^3\epsilon}{\sigma} & 0 \\ 0 & 0 & \frac{\omega^3\epsilon}{\sigma} \end{bmatrix} \begin{bmatrix} \theta_{ta}I_x \\ \theta_{ta}I_y \\ \theta_{ta}I_z \end{bmatrix} - \begin{bmatrix} \frac{\omega\epsilon}{\sigma} & 0 & 0 \\ 0 & \frac{\omega\epsilon}{\sigma} & 0 \\ 0 & 0 & \frac{\omega\epsilon}{\sigma} \end{bmatrix} \begin{bmatrix} T_{df\phi}Ix \\ T_{df\theta}Iy \\ T_{df\psi}Iz \end{bmatrix} \tag{36}$$

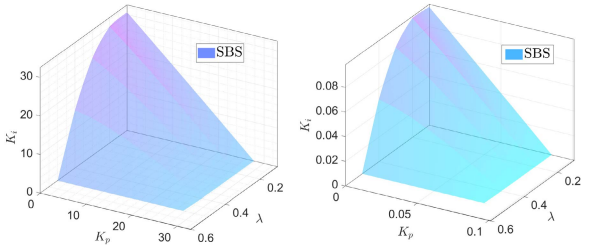
$$\begin{bmatrix} K_{i\phi} \\ K_{i\theta} \\ K_{i\psi} \end{bmatrix} = \begin{bmatrix} \frac{\omega\epsilon^2}{\sigma} & 0 & 0 \\ 0 & \frac{\omega\epsilon^2}{\sigma} & \\ 0 & 0 & \frac{\omega\epsilon^2}{\sigma} \end{bmatrix} \begin{bmatrix} T_{df\phi} \\ T_{df\theta} \\ T_{df\psi} \end{bmatrix} - \begin{bmatrix} \omega\sigma & 0 & 0 \\ 0 & \omega\sigma & 0 \\ 0 & 0 & \omega\sigma \end{bmatrix} \begin{bmatrix} T_{df\phi} \\ T_{df\theta} \\ T_{df\psi} \end{bmatrix} - \begin{bmatrix} \frac{\omega^3}{\sigma} & 0 & 0 \\ 0 & \frac{\omega^3}{\sigma} & 0 \\ 0 & 0 & \frac{\omega^3}{\sigma} \end{bmatrix} \begin{bmatrix} \theta_{ta}Ix \\ \theta_{ta}Iy \\ \theta_{ta}Iz \end{bmatrix} \tag{37}$$

From the identified stability boundary surfaces, the optimization constrains which are found to optimize the controller are given in Table 1. Now, using these constrains, the controllers were optimized and the resulting controller parameters which were obtained are given in Table 2. These tuned controller parameters were then used to perform simulation tests of the quadrotor.

IV. SOFTWARE SIMULATION RESULTS

A. SIMULINK® 3D ANIMATION VIRTUAL ENVIRONMENT

The software simulation was performed in the Matlab® Simulink® environment. The Simulink® 3D animation toolbox was used to create a virtual world in which the simulated performance of the quadrotor could be visualized in a more intuitive way. For fair comparisons with the proposed controller, integer-order PD and PID controllers have been tuned using the same performance measures and a GA optimization algorithm. Also, identical test scenarios are used to validate the effectiveness of each controller. All the input signals adhere to Assumptions 1-3.



(a) Stability Boundary Surface for z position controller (b) Stability Boundary Surface for ψ angle controller

FIGURE 4. Stability boundary surfaces (SBS) for the Z position Yaw attitude controllers.

TABLE 3. Step response result summary.

Control Strategy	% Overshoot				Settling time (s)			
	x	y	z	ψ	x	y	z	ψ
PI$^{\lambda}$-D	4.72	4.72	5.40	5.22	2.00	2.00	1.20	0.50
Integer PD	34.96	34.96	18.24	19.25	8.00	8.00	2.00	0.80
Integer PID	50.82	50.82	19.93	21.48	12.00	12.00	2.00	0.80

B. STEP RESPONSE RESULTS

A summary of the step response results are presented in Table 3 which presents the fact that the proposed controller exhibits lesser overshoot while maintaining a fast settling time in comparison to the integer order PD and PID controllers.

C. WAYPOINT TRAJECTORY RESULTS

The next test is to command the quadrotor to specific points in specific times in a 3D space. This was achieved by giving way-point commands to the quadrotor. There are a total of 5 way-points commanded to the quadrotor in this test, with the trajectory being a horizontal square at a height of 1m. A yaw angle command is also given at $t = 25$s. The displacement of the way-points in 3D space are defined in (38):

$$pos[x, y, z] = \begin{cases} [0, 0, 1], & 0 \geq t > 5 \\ [1, 0, 1], & 5 \geq t > 10 \\ [1, 1, 1], & 10 \geq t > 15 \\ [0, 1, 1], & 15 \geq t > 20 \\ [0, 0, 1], & 20 \geq t > Ts \end{cases} \tag{38}$$

where Ts is the simulation time set to 30s at a sample time of 5ms. Consequently, the test is performed with the integer order PD and PID controllers for comparison.

1) SIMULATED WAYPOINT TRAJECTORY RESULTS VISUALIZED IN 3D SPACE

Figure 5 shows the 3D visuals of the 5 waypoint trajectories. The results present that the quadrotor has a significant amount

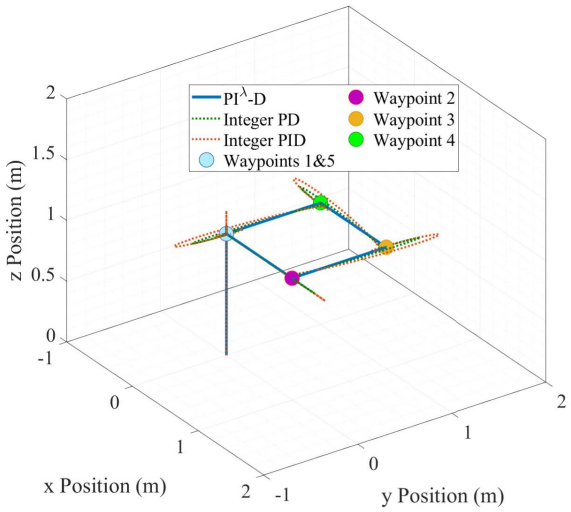


FIGURE 5. 3D visualization of the 5 way-point trajectory.

of deviation along its trajectory with integer order controllers. However, this is not the case with the proposed method. The new fractional controller exhibits only a slight deviation, providing more efficient tracking performance.

2) SIMULATED WAYPOINT TRAJECTORY RESULTS FOR DIFFERENT CONTROLLERS

From the individual graphs presented in Figure 6, it is now evident that with the proposed controller, the quadrotor is able to perform better tracking when compared to the integer order controllers at a given altitude with the sudden change in the x and y coordinates. Figure 6(d) shows that the proposed controller is superior even in yaw attitude tracking.

3) ROTOR SPEEDS FOR WAYPOINT TRAJECTORY RESULTS

The rotor speed graphs given in Figure 7 show the variations made by the controllers to account for the sudden change in position. Although the proposed controller provides lesser sudden variations, it is a bit aggressive.

4) SIMULATED TRAJECTORY ERROR RESULTS FOR WAYPOINT TRAJECTORY

The position error graphs are given in Figure 8 to evaluate the errors of the respective controllers. The position errors in Figures 8(a) to Figures 8(c) show that the proposed controller has lesser trajectory tracking errors for the waypoints. The same can be stated for the yaw attitude tracking error in Figure 8(d). The proposed controller proves to have lesser errors in yaw angle tracking than the integer order controllers.

D. HELICAL TRAJECTORY RESULTS

The helical trajectory is generated by using sine and cosine functions for the x and y position. The z position is generated by a negative ramp function with an offset of 1.5m to form the helical trajectory. The displacement functions to generate the trajectory for helical path in space-time are defined in (39):

$$\begin{cases} x(t) = \sin(\omega t) \\ y(t) = \cos(\omega t) \\ z(t) = -\frac{1}{40}t + 1.25 \\ \psi(t) = 0 \end{cases} \tag{39}$$

where $\omega = \frac{3\pi}{20}$ with the simulation time set to 35s at a sample time of 5ms. The low frequency is given so that the quadrotor moves slowly in the path and the error in sensor readings is at a minimum. Consistently, the helical path is also subjected to the integer order controllers for comparison purposes.

1) SIMULATED HELICAL TRAJECTORY RESULTS VISUALIZED IN 3D SPACE

The 3D helical trajectories given in Figure 9 displays that the proposed controller outperforms the two integer order controllers in a significant manner. The helical trajectory test is a unique test since all three co-ordinates change at every time step. However, the proposed controller manages to effectively track the x-y co-ordinates while simultaneously changing altitude.

2) SIMULATED HELICAL TRAJECTORY RESULTS FOR DIFFERENT CONTROLLERS

The individual position graphs in Figure 10 displays the deviations of the integer order controllers with the reference paths. The deviations of the proposed controller is significantly lesser when compared to the integer order controllers. Furthermore, the proposed controller maintains control over the varying altitude command while navigating in the x-y plane.

3) ROTOR SPEEDS FOR HELICAL TRAJECTORY RESULTS

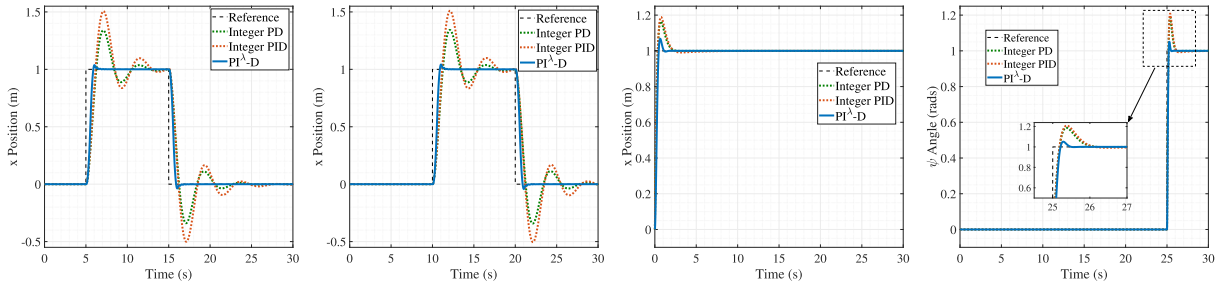
The rotor speeds shown in Figure 11 depicts that the proposed controller is a bit aggressive in control. However, consistent to the previous tests, the integer order controllers exhibit a large initial variation to the rotor speeds before stabilization.

4) SIMULATED TRAJECTORY ERROR RESULTS FOR THE HELICAL PATH

The error graphs in Figure 12 show that the proposed controller has lesser trajectory tracking errors in all the three axes, in comparison to the integer order controllers. The initial tracking error is high and then the error reduces significantly during flight. The proposed controller shows the minimum error throughout the flight when compared to other controllers. This correlates to the greater amount trajectory deviation exhibited by the integer order controllers that is distinctly depicted in the Figure.

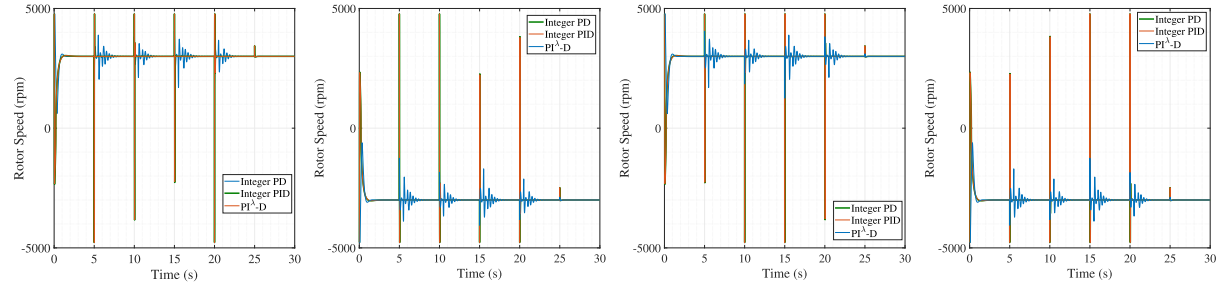
E. CIRCULAR PATH WITH DISTURBANCE

The present scheme was tested for robustness by exposing maximum wind gusts of 9m/s on the x and y direction of the quadrotor. The system was operated under wind gust disturbances to test the robustness when circulating a target point



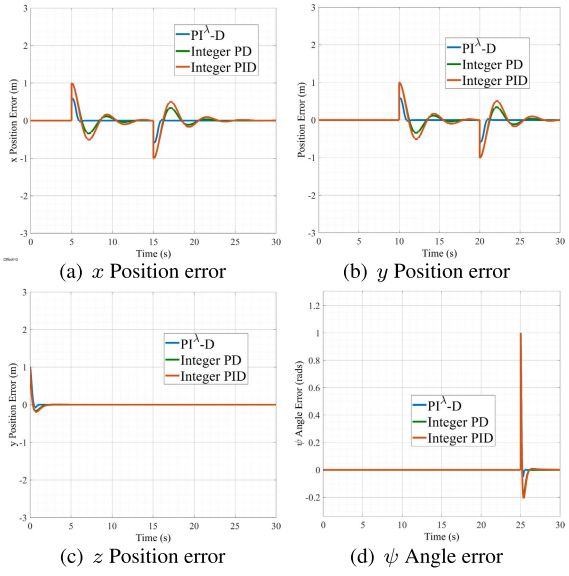
(a) x position responses (b) y position responses (c) z position responses (d) ψ angle responses

FIGURE 6. 5 way-point trajectory response results for the position controllers and ψ angle controller with different control strategies.



(a) Rotor 1 Speed (b) Rotor 2 Speed (c) Rotor 3 Speed (d) Rotor 4 Speed

FIGURE 7. Rotor speeds of waypoint trajectory for different control strategies.



(a) x Position error (b) y Position error (c) z Position error (d) ψ Angle error

FIGURE 8. Trajectory error graphs of waypoint trajectory for different control strategies.

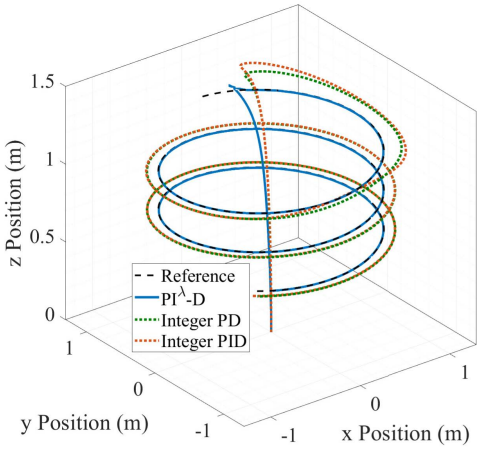
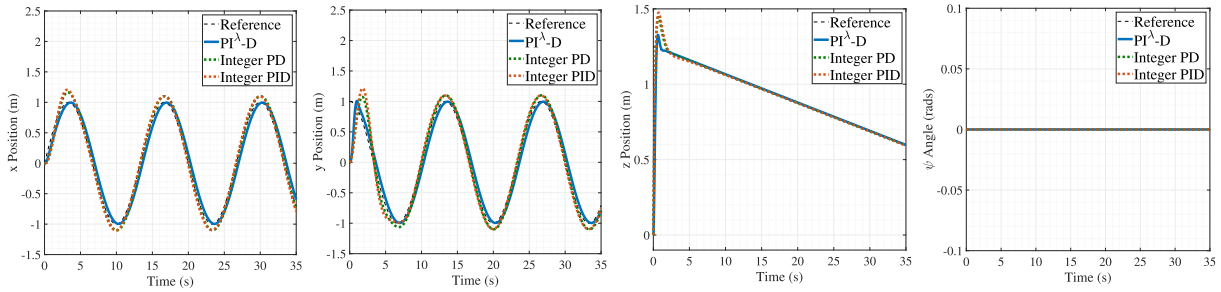


FIGURE 9. 3D visualization of the helical trajectory.

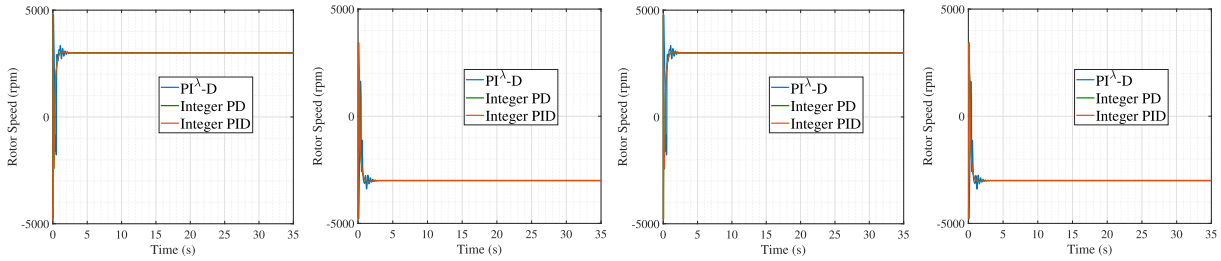
and taking off. The wind gust affects the quadrotor from x and y directions between 5sec to 8sec with a maximum gust of 9m/s. This value is quite high when compared to the average wind speed. It can be seen that the proposed controller is quick in overcoming the disturbance and reaching the desired path faster; while the integer PD and PID based systems are more disturbed and comparatively slower in rejecting the effects concerning the reference path as shown in Figure 13.

The qualitative analyses are conducted to check the numerical performance differences. When we compared the controllers with reference to performance error indices such as ITAE and Integral Square Error (ISE), it is found from Table 5 that the new method clearly outperformed than others. It is interesting to see also that the control signal variation for all four rotors' speeds is significantly less than for others. In addition, the measures of control signal variation are calculated from the absolute difference between each input sample and should be the optimal value. This measure is powerful enough to benefit edge-preserving moving parts [46]. Table 4 proved that the fractional-order scheme would provide a better result without adding more effort from the controller.



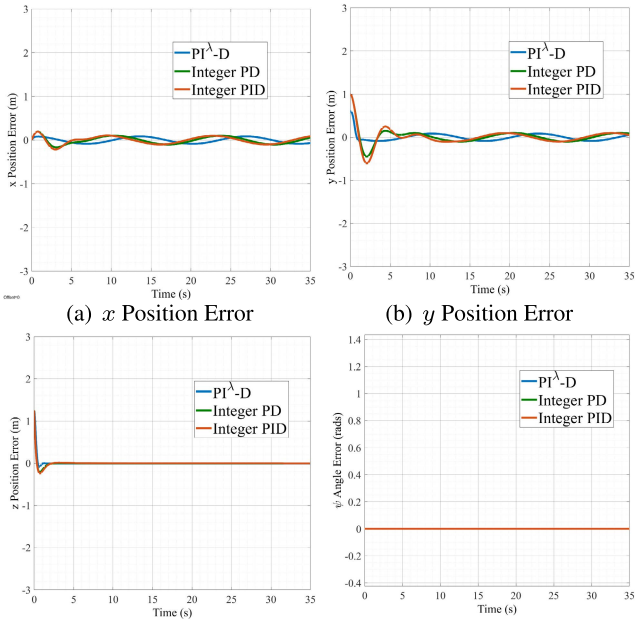
(a) x position responses (b) y position responses (c) z position responses (d) ψ angle responses

FIGURE 10. Helical trajectory response results for the position controllers and ψ angle controller with different control strategies.



(a) Rotor 1 Speed (b) Rotor 2 Speed (c) Rotor 3 Speed (d) Rotor 4 Speed

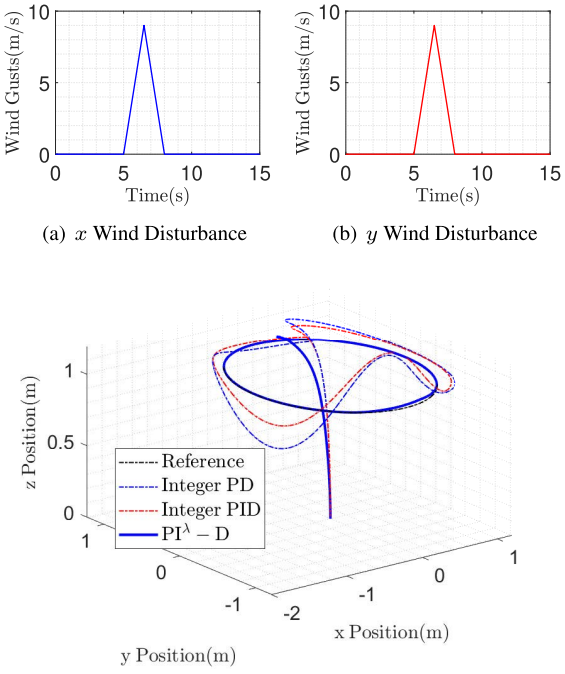
FIGURE 11. Rotor speeds of helical trajectory for different control strategies.



(a) x Position Error (b) y Position Error

(c) z Position Error (d) ψ Angle Error

FIGURE 12. Trajectory error graphs of the helical trajectory for different control strategies.



(a) x Wind Disturbance (b) y Wind Disturbance

(c) Results with wind disturbance

FIGURE 13. 3D visualization of the circular path with wind disturbance from x and y directions.

V. REAL-TIME EXPERIMENTATION RESULTS

The hardware experimentation was conducted in a closed and controlled environment as seen in Figure 14. The parrot rolling spider minidrone was utilized together with a base station that had a Laptop with Matlab® Simulink® installed. The parrot rolling spider is an indoor minidrone which is programmable. Its sensors consists of a 3 axis gyroscope for attitude sensing, 3 axis accelerometer, camera for visual navigation to sense the x-y position and ultrasonic sensors with a pressure sensor that work together to estimate its flying altitude. The hardware support package for parrot minidrone is available for Matlab® Simulink®.

TABLE 4. Control signal measures for comparison.

Rotor	PI$^{\lambda}$-D	PID	PD
Rotor Speed1	7.47	39.6×10^3	39.6×10^3
Rotor Speed2	7.47	25.2×10^3	25.2×10^3
Rotor Speed3	7.47	35.0×10^3	35.0×10^3
Rotor Speed4	4.47	29.7×10^3	29.7×10^3

TABLE 5. Performance error indices.

Position	Controller	ITAE	ISE
x	PI$^{\lambda}$-D	2.3×10^3	37.26
	PID	5.2×10^3	471.83
	PD	6.9×10^3	560.13
y	PI$^{\lambda}$-D	1.94×10^3	111.13
	PID	5.62×10^3	540.00
	PD	7.02×10^3	692.34
z	PI$^{\lambda}$-D	11.45	32.74
	PID	31.75	21.50
	PD	69.25	22.80
ψ	PI$^{\lambda}$-D	2.12×10^{-11}	2.7×10^{-27}
	PID	1.60×10^{-10}	1.48×10^{-25}
	PD	6.20×10^{-12}	9.3×10^{-27}

TABLE 6. Parrot minidrone hardware parameters.

Parameter	Value	Unit	Symbol
Weight	6.95×10^{-2}	kg	m
Arm Length	6.20×10^{-2}	m	l
Inertial moment about x	6.86×10^{-5}	$\mathrm{kg\,m^2}$	I_x
Inertial moment about y	9.20×10^{-5}	$\mathrm{kg\,m^2}$	I_y
Inertial moment about z	1.36×10^{-4}	$\mathrm{kg\,m^2}$	I_z
Motor Thrust Factor	1.00×10^{-2}	$\mathrm{N\,rad^{-2}\,s^{-2}}$	k_f
Motor Torque Factor	7.82×10^{-4}	$\mathrm{N\,m\,rad^{-2}\,s^{-2}}$	k_m
Gravitational Acceleration	9.98	$\mathrm{m^2}$	g

This enables developments to be Matlab® based on the minidrone. The hardware support package is loaded through the Matlab® add-ons tab. The Simulink models and program codes for controller design are available for the reference in [47].

A. PARROT MINIDRONE QUADROTOR PHYSICAL PARAMETERS

The hardware parameters are tabulated in Table 6, used same for the simulation study. After following the proposed tuning method, the controller parameters are obtained as given in Table 2.

Upon loading and configuring the hardware support package for the parrot minidrone, the proposed PI$^{\lambda}$-D controller was designed in a Simulink® project with the help of the FOMCON toolbox. The FOMOCON toolbox provides

FIGURE 14. Hardware experimental environment setup.

fractional blocks to be incorporated in controller design in Simulink®. A Bluetooth personal area network (PAN) is used to connect the minidrone to the workstation laptop for programming. After completing the controller design, the Flight Controller Subsystem in Simulink® is converted into C code and uploaded onto the minidrone through the Bluetooth PAN. Upon uploading the code, a command is given from Simulink® to initiate the experiment. The connection between the minidrone and the Simulink® environment are in real-time. When a flight experiment is completed, the experimental flight data, including the positions, velocities and rotor speeds, are transferred from the minidrone to the Matlab® workspace via the bluetooth PAN. Post-flight analysis are then done with the data recorded into the Matlab® workspace.

B. WAYPOINT TRAJECTORY RESULTS

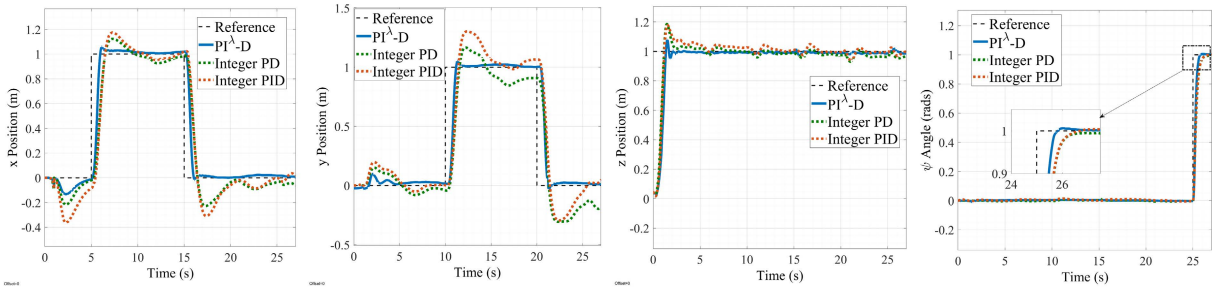
Consequently, the 5 waypoint trajectory test was also replicated in real time experimentation for practical analysis.

1) EXPERIMENTAL WAYPOINT TRAJECTORY RESULTS VISUALIZED IN 3D SPACE

The 3D plot in Figure 15 displays the 5-waypoint trajectory results for the minidrone. It can be distinctively speculated that the proposed controller delivers better performance in comparison to the integer order PD and PID controllers.

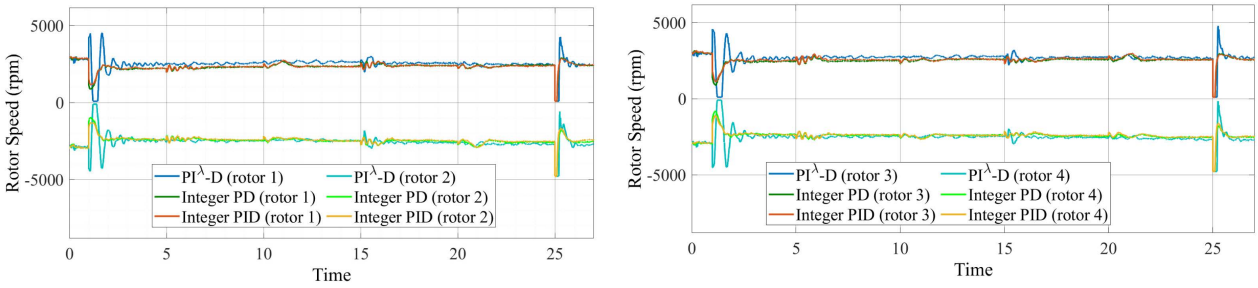
2) EXPERIMENTAL WAYPOINT TRAJECTORY RESULTS

Studying Figures 15(a)-15(b), it is observed that while the integer order controllers suffer from some bias in the steady state, the proposed controller has significantly lesser bias in its steady state. Ripples in the altitude of the minidrone



(a) x position responses (b) y position responses (c) z position responses (d) ψ angle responses

FIGURE 15. 5 way-point trajectory experimental results for the position controllers and ψ angle controller with different control strategies.



(a) Rotor 1 Speed (b) Rotor 3 Speed

FIGURE 16. Experimental rotor speeds of waypoint trajectory for different control strategies.

while changing between waypoints are lesser for the proposed controller as depicted in Figure 15(c). The yaw angle control in Figure 15(d) also suggests that the proposed controller provides better yaw angle control in comparison to the integer order controllers. The 3D visualization of the 5 way-point trajectory is presented in Figure 17.

3) EXPERIMENTAL ROTOR SPEED RESULTS FOR WAYPOINT TRAJECTORY TRACKING

The changes in the rotor speeds for different control strategies are given in Figure 16. It can be also noticed that there are spikes in rotor speeds at the time of waypoint transitions and also for when the quadrotor is commanded a yaw angle.

4) EXPERIMENTAL TRAJECTORY ERROR RESULTS FOR WAYPOINT TRAJECTORY

The experimental error graphs in Figure 18 displays that the proposed controller exhibits lesser trajectory tracking errors for the 5 waypoint trajectory when compared to the integer order controllers.

C. HELICAL TRAJECTORY RESULTS

Finally, the helical trajectory test in correlation with the other tests is again replicated in real time experimentation to understand the practicality of the proposed controller in real life applications. The responses in Figure 22 show that

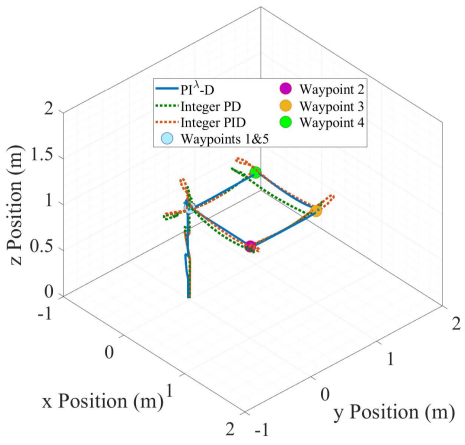
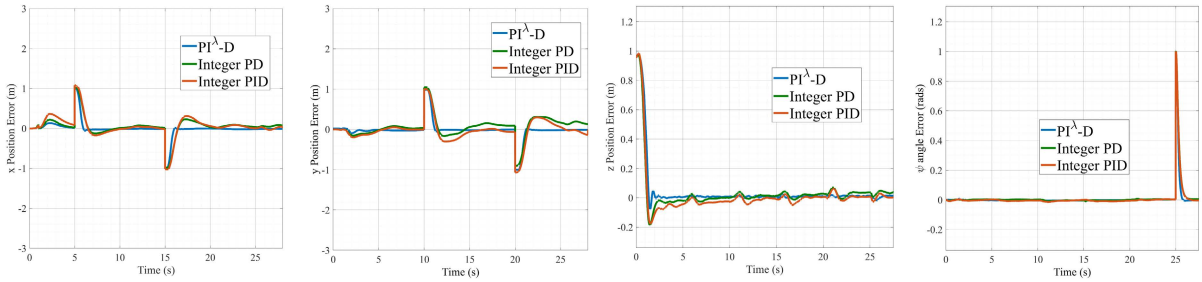


FIGURE 17. 3D visualization of the 5 way-point trajectory from experimental Results.

although there are some errors, the proposed method tracks the reference path more efficiently in comparison to PD and PID controllers.

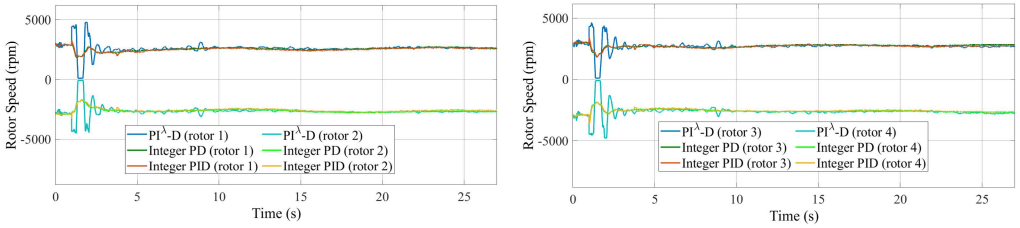
1) ROTOR SPEED EXPERIMENTAL RESULTS

Fig. 19 displays the rotor speeds of different controllers on the minidrone acting on the reference commands given to the minidrone. It is again noticeable that the proposed controller has a more aggressive rotor speed activity.



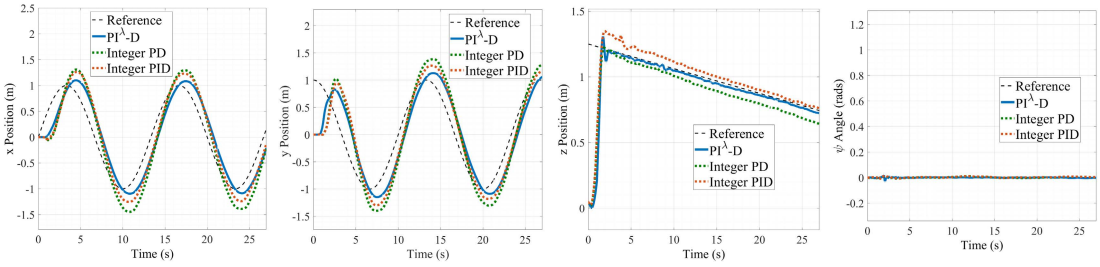
(a) x Position Error (b) y Position Error (c) z Position Error (d) ψ Angle Error

FIGURE 18. Experimental trajectory error graphs of the waypoint trajectory for different control strategies.



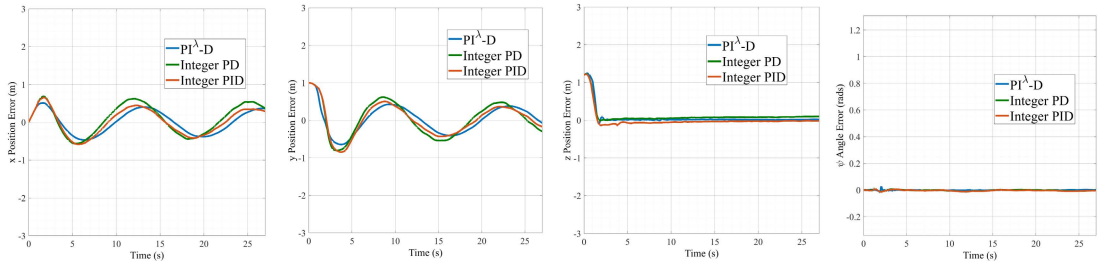
(a) Rotor 1 Speed (b) Rotor 3 Speed

FIGURE 19. Experimental rotor speeds of helical trajectory for different control strategies.



(a) x position responses (b) y position responses (c) z position responses (d) ψ angle responses

FIGURE 20. Helical trajectory experimental results for the position controllers and ψ angle controller with different control strategies.



(a) x Position Error (b) y Position Error (c) z Position Error (d) ψ Angle Error

FIGURE 21. Experimental trajectory error graphs of the helical Trajectory for different control strategies.

2) HELICAL TRAJECTORY EXPERIMENTAL RESULTS

The helical trajectory results are illustrated in Figures 20(a) to 20(b). These results again depicted that the proposed controller has lesser deviations from the desired trajectory.

3) EXPERIMENTAL TRAJECTORY ERROR RESULTS

The experiment results for the positioning and ψ angle with various approaches are studied carefully. The error graphs in Figure 21 show that the proposed scheme has less trajectory tracking errors in all the three axes.

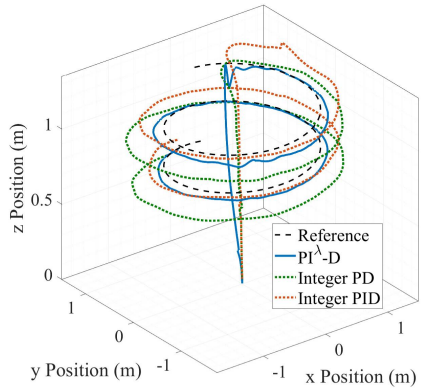


FIGURE 22. 3D visualization of the helical trajectory from experimental results.

VI. CONCLUSION

This paper has successfully developed MIMO PI$^{\lambda}$-D design for better trajectory tracking of a quadrotor system. Firstly, the cascaded control scheme was presented on a linearized quadrotor model with six states. Firstly, a practical and effective scheme was proposed using a designed feedback D-controller to stabilize the unstable states in the quadrotor. Then, a PI$^{\lambda}$ controller was designed with a constraint boundary condition. The method was evaluated for two different tests; one was waypoint tracking, and the other was helical trajectory tracking. The simulation analysis shows that the new PI$^{\lambda}$-D has better trajectory tracking results, with significantly fewer indices values compared to the integer counterparts, even with disturbances affecting the system. Importantly, real-time tests also depicted the same robust results in trajectory tracking. It would be interesting to verify further the merits of fractional controllers on multirotor UAVs such as hexacopters.

REFERENCES

[1] H. Aasen, E. Honkavaara, A. Lucieer, and P. Zarco-Tejada, "Quantitative remote sensing at ultra-high resolution with UAV spectroscopy: A review of sensor technology, measurement procedures, and data correction workflows," *Remote Sens.*, vol. 10, no. 7, p. 1091, 2018.

[2] F. Sabatino, "Quadrotor control: Modeling, nonlinearcontrol design, and simulation," M.S. thesis, KTH Elect. Eng., Stockholm, Sweden, 2015.

[3] I. A. Sheikh, "Adaptive control strategies robust to external disturbances for multirotor UAV's: Theoratical aspects and experimental validation," Ph.D. dissertation, School Inf. Technol., Eng., Math., Phys., Univ. South Pacific, Suva, Fiji, 2018.

[4] B. Rao, A. G. Gopi, and R. Maione, "The societal impact of commercial drones," *Technol. Soc.*, vol. 45, pp. 83–90, May 2016.

[5] P. Cleveland, "Drama drones: An investigation into integrating drones into real world filmmaking in New Zealand," Ph.D. dissertation, Dept. Mech. Eng., Auckland Univ. Technol., Auckland, New Zealand, 2017.

[6] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: Leveraging UAVs for disaster management," *IEEE Pervasive Comput.*, vol. 16, no. 1, pp. 24–32, Jan. 2017.

[7] S. I. Azid, K. Kumar, M. Cirrincione, and A. Fagiolini, "Wind gust estimation for precise quasi-hovering control of quadrotor aircraft," *Control Eng. Pract.*, vol. 116, Nov. 2021, Art. no. 104930.

[8] F. Riether, "Agile quadrotor maneuvering using tensor-decomposition-based globally optimal control and onboard visual-inertial estimation," M.S. thesis, Des. Creative Technol. Colab, Massachusetts Inst. Technol., Cambridge, MA, USA, 2016.

[9] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles: Modeling, estimation, and control of quadrotor," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 20–32, Sep. 2012.

[10] E.-H. Zhang, J.-J. Xiong, and J.-L. Luo, "Second order sliding mode control for a quadrotor UAV," *ISA Trans.*, vol. 53, no. 4, pp. 1350–1356, Jul. 2014.

[11] A. L. Salih, M. Moghavvemi, M. A. F. Haf, and K. Gaeid, "Flight PID controller design for a UAV quadrotor," *Sci. Res. Essays*, vol. 5, pp. 3660–3667, Dec. 2010.

[12] P. Wang, Z. Man, Z. Cao, J. Zheng, and Y. Zhao, "Dynamics modelling and linear control of quadcopter," in *Proc. Int. Conf. Adv. Mech. Syst. (ICAMechS)*, Nov. 2016, pp. 498–503.

[13] J. Moreno-Valenzuela, R. Pérez-Alcocer, M. Guerrero-Medina, and A. Dzul, "Nonlinear PID-type controller for quadrotor trajectory tracking," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 5, pp. 2436–2447, Oct. 2018.

[14] B. Erginer and E. Altug, "Modeling and PD control of a quadrotor VTOL vehicle," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2007, pp. 894–899.

[15] Z. Li, X. Ma, and Y. Li, "Robust tracking control strategy for a quadrotor using RPD-SMC and RISE," *Neurocomputing*, vol. 331, pp. 312–322, Feb. 2019.

[16] M. Islam, M. Okasha, and M. M. Idres, "Trajectory tracking in quadrotor platform by using PD controller and LQR control approach," *IOP Conf. Ser., Mater. Sci. Eng.*, vol. 260, Nov. 2017, Art. no. 012026.

[17] F. Santoso, M. A. Garratt, and S. G. Anavatti, "Hybrid PD-fuzzy and PD controllers for trajectory tracking of a quadrotor unmanned aerial vehicle: Autopilot designs and real-time flight tests," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 51, no. 3, pp. 1817–1829, Mar. 2021.

[18] L. Xulin and G. Yuying, "Fault tolerant control of a quadrotor UAV using control allocation," in *Proc. Chin. Control Decis. Conf. (CCDC)*, Jun. 2018, pp. 1818–1824.

[19] R. Miranda-Colorado and L. T. Aguilar, "Robust PID control of quadrotors with power reduction analysis," *ISA Trans.*, vol. 98, pp. 47–62, Mar. 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S001905781930391X

[20] R. Miranda-Colorado, L. T. Aguilar, and J. E. Herrero-Brito, "Reduction of power consumption on quadrotor vehicles via trajectory design and a controller-gains tuning stage," *Aerosp. Sci. Technol.*, vol. 78, pp. 280–296, Jul. 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1270963817319028

[21] S. I. Azid, K. Kumar, M. Cirrincione, and A. Fagiolini, "Robust motion control of nonlinear quadrotor model with wind disturbance observer," *IEEE Access*, vol. 9, pp. 149164–149175, 2021.

[22] P. T. Jardine, S. N. Givigi, S. Yousefi, and M. J. Korenberg, "Adaptive MPC using a dual fast orthogonal Kalman filter: Application to quadcopter altitude control," *IEEE Syst. J.*, vol. 13, no. 1, pp. 973–981, Mar. 2019.

[23] S.-K. Kim and C. K. Ahn, "Adaptive nonlinear tracking control algorithm for quadcopter applications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 1, pp. 84–94, Feb. 2020.

[24] R. Perez-Alcocer and J. Moreno-Valenzuela, "Adaptive control for quadrotor trajectory tracking with accurate parametrization," *IEEE Access*, vol. 7, pp. 53236–53247, 2019.

[25] A. Eltayeb, M. F. Rahmat, M. A. M. Basri, M. A. M. Eltoum, and S. El-Ferik, "An improved design of an adaptive sliding mode controller for chattering attenuation and trajectory tracking of the quadcopter UAV," *IEEE Access*, vol. 8, pp. 205968–205979, 2020.

[26] D. J. Almakhles, "Robust backstepping sliding mode control for a quadrotor trajectory tracking application," *IEEE Access*, vol. 8, pp. 5515–5525, 2019.

[27] F. Jiang, F. Pourpanah, and Q. Hao, "Design, implementation, and evaluation of a neural-network-based quadcopter UAV system," *IEEE Trans. Ind. Electron.*, vol. 67, no. 3, pp. 2076–2085, Mar. 2020.

[28] I. Podlubny, "Fractional-order systems and fractional-order controllers," *Inst. Experim. Phys., Slovak Acad. Sci., Kosice*, vol. 12, no. 3, pp. 1–18, 1994.

[29] C. A. Monje, Y. Chen, B. M. Vinagre, D. Xue, and V. Feliu-Batlle, *Fractional-order Systems and Controls: Fundamentals and Applications*. Cham, Switzerland: Springer, 2010.

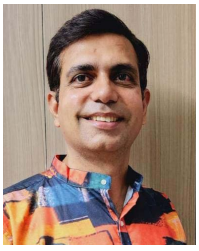
[30] U. Mehta, V. Lechappe, and O. P. Singh, "Simple FOPI tuning method for real-order time delay systems," in *Advances in Systems, Control and Automation*. Cham, Switzerland: Springer, 2018, pp. 459–468.

[31] P. Chen and Y. Luo, "A two-degree-of-freedom controller design satisfying separation principle with fractional-order PD and generalized ESO," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 1, pp. 137–148, Feb. 2022.

[32] H.-P. Ren, S.-S. Jiao, X. Wang, and O. Kaynak, "Fractional order integral sliding mode controller based on neural network: Theory and electro-hydraulic benchmark test," *IEEE/ASME Trans. Mechatronics*, vol. 27, no. 3, pp. 1457–1466, Jun. 2022.
[33] A. Tepljakov, B. B. Alagoz, C. Yeroglu, E. A. Gonzalez, S. H. Hosseinnia, E. Petlenkov, A. Ates, and M. Cech, "Towards industrialization of FOPID controllers: A survey on milestones of fractional-order control and pathways for future developments," *IEEE Access*, vol. 9, pp. 21016–21042, 2021.
[34] S. I. Azid, V. P. Shankaran, and U. Mehta, "Fractional PI controller for integrating plants," in *Proc. 16th Int. Conf. Control, Autom., Robot. Vis. (ICARCV)*, Dec. 2020, pp. 904–909.
[35] M. Vahdanipour and M. Khodabandeh, "Adaptive fractional order sliding mode control for a quadrotor with a varying load," *Aerosp. Sci. Technol.*, vol. 86, pp. 737–747, Mar. 2019.
[36] M. N. Soorki, T. V. Moghaddam, and A. Emamifard, "A new fast finite time fractional order adaptive sliding-mode control for a quadrotor," in *Proc. 7th Int. Conf. Control, Instrum. Autom. (ICCIA)*, Feb. 2021, pp. 1–5.
[37] A. Bist and S. Sondhi, "Disturbance observer based fractional order sliding mode control for quadrotor," in *Proc. IEEE 17th India Council Int. Conf. (INDICON)*, Dec. 2020, pp. 1–7.
[38] Z. Cheng, Z. Ma, G. Sun, and H. Dong, "Fractional order sliding mode control for attitude and altitude stabilization of a quadrotor UAV," in *Proc. Chin. Autom. Congr. (CAC)*, Oct. 2017, pp. 2651–2656.
[39] C. Izaguirre-Espinosa, A. J. Muñoz-Vazquez, A. Sanchez-Orta, V. Parra-Vega, and I. Fantoni, "Fractional-order control for robust position/yaw tracking of quadrotors with experiments," *IEEE Trans. Control Syst. Technol.*, vol. 27, no. 4, pp. 1645–1650, Jul. 2019.
[40] R. Cajo, C. Copot, C. M. Ionescu, R. De Keyser, and D. Plaza, "Fractional order PD path-following control of an AR. Drone quadrotor," in *Proc. IEEE 12th Int. Symp. Appl. Comput. Intell. Informat. (SACI)*, May 2018, pp. 291–296.
[41] F. Oliva-Palomo, A. J. Munoz-Vazquez, A. Sanchez-Orta, V. Parra-Vega, C. Izaguirre-Espinosa, and P. Castillo, "A fractional nonlinear PI-structure control for robust attitude tracking of quadrotors," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 6, pp. 2911–2920, Dec. 2019.
[42] V. P. Shankaran, S. I. Azid, and U. Mehta, "Fractional-order PI plus d controller for second-order integrating plants: Stabilization and tuning method," *ISA Trans.*, vol. 129, pp. 592–604, Oct. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0019057821006297
[43] P. I. Corke and O. Khatib, *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*, vol. 73. Cham, Switzerland: Springer, 2011.
[44] S. E. Hamamci, "Stabilization using fractional-order PI and PID controllers," *Nonlinear Dyn.*, vol. 51, nos. 1–2, pp. 329–343, Oct. 2007.
[45] Y. Cheng and C. Hwang, "Stabilization of unstable first-order time-delay systems using fractional-order Pd controllers," *J. Chin. Inst. Eng.*, vol. 29, no. 2, pp. 241–249, Mar. 2006.
[46] A. Ranjan and U. Mehta, "Fractional filter IMC-TDD controller design for integrating processes," *Results Control Optim.*, vol. 8, Sep. 2022, Art. no. 100155.
[47] V. P. Shankaran, "Fractional order controller design for quadrotor applications," M.S. thesis, School Inf. Technol., Eng., Math., Phys., Univ. South Pacific, Suva, Fiji, 2022.

VIVEK PAWAN SHANKARAN received the B.E. degree in electrical/electronics and the M.Sc. degree in engineering from The University of the South Pacific, Fiji, in 2020 and 2022, respectively. He is currently a Technical Services Engineer at Fiji Airways specializing in avionics systems. His current research interests include automation, control systems for robotics and UAV, computer vision, and deep learning for industrial applications. He was awarded the Gold Medal from The University of the South Pacific for having the most outstanding Master of Science Thesis for the year 2022.

SHEIKH IZZAL AZID (Member, IEEE) received the M.Sc. degree in engineering, in 2010, in the area of automation and intelligent systems. He is currently a Lecturer with the School of Engineering and Physics, The University of the South Pacific. His Ph.D. studies are in the area of aerial robotics and robust control at The University of the South Pacific in collaboration with the University of Palermo, Italy. He is also the Vice President of IEEE VTS New Zealand North. He also worked as a Consultant for several European Projects on Science, Technology, and Innovation in the Pacific Island Countries. His research interests include robotics, controls, UAVs, autonomous systems, state observers, and microprocessor applications. He was a recipient of the Erasmus/Staff Exchange Program and he was attached at the Technical University of Sofia.



UTKAL MEHTA (Senior Member, IEEE) received the Ph.D. degree in the area of system identification and process control from IIT Guwahati, India. He is currently working in electrical and electronics engineering at The University of the South Pacific, Fiji, as an Associate Professor. His current research interests include process identification, applied fractional calculus for modeling, and fractional-order filter design on re-configurable devices like FPAA and various robotics applications for medical and industrial automation.

ADRIANO FAGIOLINI (Member, IEEE) received the M.S. degree in computer science engineering and the Ph.D. degree in robotics and automation from the University of Pisa, in 2004 and 2009, respectively. He is currently an Assistant Professor at the University of Palermo, Italy. He has been a Visiting Researcher at the Department of Energy, IUT Longwy Université de Lorraine, France, in 2019, and the Department of Mechanical Engineering, University of California at Riverside, in 2015 and 2017. He enrolled in the Summer Student Programme at the European Center for Nuclear Research (CERN), Geneva, in 2002. In 2008, he led the team of the University of Pisa, during the first European Space Agency's Lunar Robotics Challenge, which resulted in a second-place prize for the team. He was one of the recipients of the IEEE ICRA Best Manipulation Paper Award, in 2005.