Modeling and analysis of nonlinear backstepping controller for the crazyflie quadrotor trajectory tracking

Modeling and analysis of nonlinear backstepping controller for the crazyflie quadrotor trajectory tracking

O H Zekry[1], M M Ashry[1], A T Hafez[1], and T Attia[2]

[1]Electrical Engineering Department, Military Technical College, Cairo, Egypt
[2]Mechatronics Engineering Department, Military Technical College, Cairo, Egypt

E-mail: uzekry@yahoo.com

Abstract. This paper illustrates the Newton-Euler laws-based nonlinear dynamic model of a Crazyflie Quadrotor Unmanned Aerial Vehicle (CF-QUAV). It demonstrates the state-space-form equations that govern how the CF-QUAV moves around its frame (body). A Proposed nonlinear control strategy is used to regulate the height (z), position (x, y), attitude (roll and pitch), and heading (yaw) angles. This strategy is based on a Backstepping nonlinear controller defined by Lyapunov theory. The proposed controller can replace the Proportional-Integration-Derivative (PID) controller utilized in the CF-QUAV model to follow the desired trajectories gleaned from CF lab tests. In addition, the parameters of the proposed controller are tuned using the Genetic Algorithm (GA) in order to achieve the best performance and stability of the CF-QUAV model. Simulation-based experiments with statistical analysis are carried out to evaluate the performance of the proposed nonlinear controller and compare its performance and stability with the linear PID controller.

1 Introduction

Since Quadrotor Unmanned Aerial Vehicles (QUAVs) imposed their importance in our world because of their easily-perceived advantages compared to conventional aircraft. Researchers pay more attention to improve the QUAVs performance in military [1, 2] and civilian missions as (mapping, exploration, package delivery, keeping an eye on indoor or outdoor environments) [3–6]. Nevertheless, some of the features that include under-actuation, strong coupling, multivariable, and unknown nonlinearities represent significant challenges facing the QUAVs regarding vertical take-off and landing, autonomous hovering, and maneuvering [7]. As a result, a QUAV control seems to be a relatively complex and challenging operation.

It has become necessary to use nonlinear control techniques to address the drawbacks of linear controllers that cannot fully govern real-world dynamic systems. Backstepping controller is one of the nonlinear techniques that is used in QUAV. In [8] the Backstepping is merged with neural networks to follow the desired height(z) and position (x,y) while maintaining the stability of all attitude angles (roll, pitch, and yaw) at the same time. A comparison between a Backstepping controller based on fuzzy logic and another controller based on the least mean square was

introduced in [9]. Both controllers were subjected to check their ability to control the attitude angles of the QUAV, and the results showed that the first presented controller outperformed the second one in terms of overall performance. Authors in [10] revealed the precision of the transient response of the suggested optimal Backstepping-based particle swarm optimization (PSO) in tracking the required trajectories. In [11], an integral Backstepping control algorithm was designed and introduced for tracking paths, especially (x, y, and z) of the QUAV, where the QUAV was able to track the desired trajectories with the proposed controller. Similarly, the integral Backstepping controller was introduced in [12] to reduce the disturbances in the steady state mode. In [13] the Backstepping controller achieved a good performance in both transient and steady-state operations.

This paper demonstrates the shortcomings of the PID controller used in [14] while using the extracted recorded input data from the Crazyflie (CF) as the desired real-time trajectories. This is achieved by replacing the PID controller with a nonlinear Backstepping controller. Furthermore, the gains of the Backstepping controller are tuned by the GA. The results show that the proposed nonlinear controller can track the desired trajectories with high accuracy.

The remainder of this paper is structured as follows. General information about the CF-QUAV platform used in this work is shown in Section 2, along with the kinematic and dynamic models with the state-space representation of the CF-QUAV based on Newton-Euler equations. The Crazyflie control model published in [14] and the proposed control strategy for resolving the challenges introduced in this work are both described in Section 3. The simulation results of CF performance are presented in Section 4. This section also comprehensively discusses the differences in results when using the PID controller versus the proposed controller. Finally, in Section 5, the contribution of the paper is summarized together and discusses future work.

2 CF-QUAV

The Bitcraze CF-QUAV is the platform used in this work [15]. It comprises four coreless DC rotors, each one is attached to one end of a cross-like construction as shown in Figure 1. Each rotor comprises a plastic propeller and is powered individually, whereas odd-numbered rotors rotate opposite to even-numbered ones. The speed of the four rotors is stepped up or down simultaneously to provide vertical motion. The speed of the even rotors is altered to create roll rotation, while the pitch rotation is produced by changing the speeds of the odd rotors. While each propeller-pair produces a difference in the opposite torque to induce yaw rotation. The Newton-Euler equations are used to create the kinematics and dynamics models of the CF-QUAV, given the following assumptions.

Assumption 1 *The propellers and the body structure are rigid.*

Assumption 2 *The structure of CF-QUAV is symmetrical.*

Assumption 3 *The center of gravity for the CF is the same as the origin of the body frame.*

Assumption 4 *The square of the propeller speed determines the drag and the thrust.*

2.1 Kinematics and dynamics model of the CF-QUAV

Two coordinate frames are defined in order to describe the CF motion. The body frame (B), which is located at the center of the CF body with (B_x, B_y, B_z) axes, while the earth inertial frame (E) is settled at the ground level with (E_X, E_Y, E_Z) axes as shown in Figure 1. The CF-QUAV is composed of six degrees of freedom (DOF): x, y, z, roll (φ), pitch (ϑ), and yaw (ψ). The first three elements are used to express the height (z) and position (x, y), while the others are used to describe the orientation of the CF in space. The CF-QUAV attitude is determined by its roll and pitch angles, while its heading is determined by its yaw angle. The distance

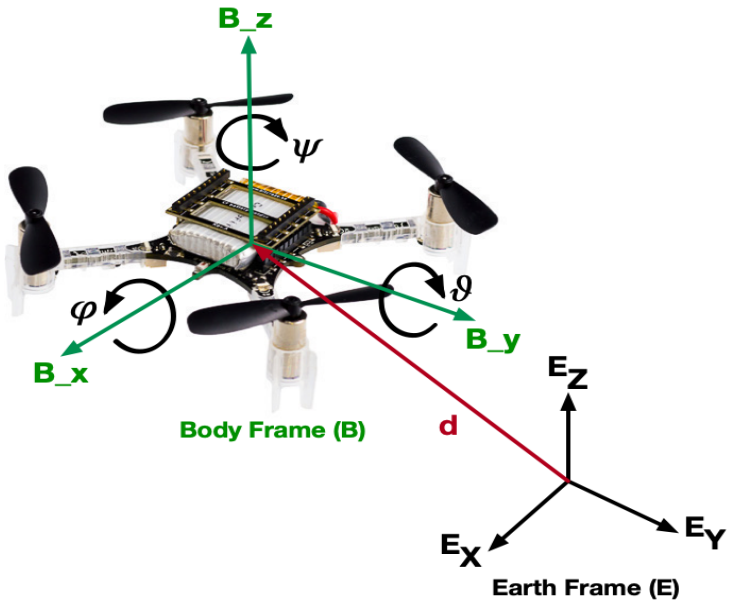


Figure 1: CF-QUAV Reference Frames.

between the earth and the body frames indicates the absolute position of the CF, $d = [x\ y\ z]^T$. In Figure 1, the orientation of the CF is indicated by roll, pitch, and yaw angles $(\varphi, \vartheta, \psi)$, which indicates rotations about the body frame axes (B_x, B_y, and B_z), respectively.

The CF-QUAV position in the earth initial frame is expressed as follows:

$$\begin{bmatrix} E_X \\ E_Y \\ E_Z \end{bmatrix} = T_B^E \begin{bmatrix} B_x \\ B_y \\ B_z \end{bmatrix}, \tag{1}$$

where T_B^E is the transformation from earth frame to body frame and has been defined as follows:

$$T_B^E = \begin{bmatrix} c\,\vartheta\ c\,\psi & s\,\varphi\ s\,\vartheta\ c\,\psi\ -c\,\varphi\ s\,\psi & c\,\varphi\ s\,\vartheta\ c\,\psi + s\,\varphi\ s\,\psi \\ c\,\vartheta\ s\,\psi & s\,\varphi\ s\,\vartheta\ s\psi + c\,\varphi\ c\,\psi & c\,\varphi\ s\,\vartheta\ s\,\psi - s\,\varphi\ c\,\psi \\ -s\,\vartheta & s\,\varphi c\,\vartheta & c\,\varphi\ c\,\vartheta \end{bmatrix}, \tag{2}$$

where c and s denote *cosine* and *sine* respectively.

2.2 Dynamics model of the CF-QUAV

The CF-QUAV motion can be divided into two subsystems: the rotational subsystem (roll, pitch, and yaw) and the translational one (z, x, and y). Figure 2 shows the forces and moments acting on the CF. Each rotor causes an upwards thrust force $\mathcal{F}_n = \beta\ w_n^2$ and generates a moment $\mathcal{T}_n = \gamma\ w_n^2$ in a direction opposite to the direction of rotation of the corresponding rotor n.

The total moments about the body frame are expressed as:

$$\mathcal{T}_b = \begin{bmatrix} \beta\ \varrho\ (-w_2^2 + w_4^2) \\ \beta\ \varrho\ (w_1^2 - w_3^2) \\ \gamma\ (w_1^2 - w_2^2 + w_3^2 - w_4^2) \end{bmatrix}, \tag{3}$$

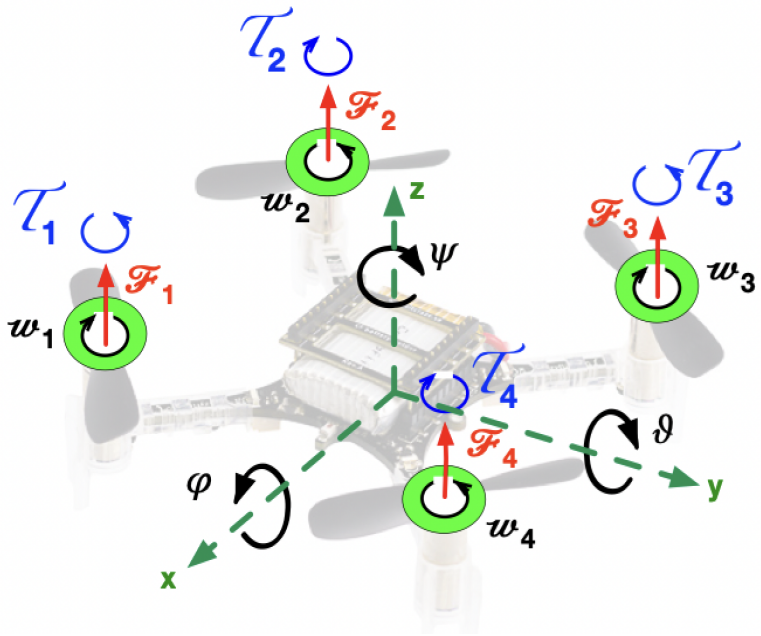


Figure 2: Acting Forces and Moments on the CF-QUAV.

where β and γ are the constants of the aerodynamic force and moment respectively, ϱ is the arm length of the CF, and w_n is the speed of each rotor. The gyroscopic moment $\mathcal{T}_g$, caused by the rotors inertia $\Im_r$ and the rotors' relative speed $w_r = -w_1 + w_2 - w_3 + w_4$, which is defined as follows:

$$\mathcal{T}_g = \omega \times \begin{bmatrix} 0 & 0 & \Im_r & w_r \end{bmatrix}^T. \tag{4}$$

The equations for rotational motion in the body frame are derived using the Newton-Euler method in the following general form:

$$\jmath\, \dot{\omega} + \omega \times \jmath\, \omega + \mathcal{T}_g = \mathcal{T}_b, \tag{5}$$

where ω is the angular body rates $[\dot{\varphi}\ \dot{\vartheta}\ \dot{\psi}]^T$ and $\jmath$ is the diagonal inertia matrix of the CF with zero off-diagonal elements due to the symmetrical Quadrotor's structure which is stated in Assumption 2, and it can be written as follows:

$$\jmath = \begin{bmatrix} I_{XX} & 0 & 0 \\ 0 & I_{YY} & 0 \\ 0 & 0 & I_{ZZ} \end{bmatrix}, \tag{6}$$

where I_{XX}, I_{YY}, and I_{ZZ} are the moment of inertia about the X, Y, and Z axes of the body frame, respectively.

The translational equations of CF motion are determined in the earth's inertial frame using Newton's second law as follows:

$$m\ddot{d} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \mathcal{F}_B, \tag{7}$$

where m is the mass of the CF and g is the acceleration of gravity. When the CF is in a horizontal orientation, the only non-gravitational force $\mathcal{F}_B$ acting on it is the thrust produced by the rotation of the propellers as follows:

$$\mathcal{F}_B = \begin{bmatrix} 0 \\ 0 \\ -\beta(w_1^2 + w_2^2 + w_3^2 + w_4^2) \end{bmatrix}. \tag{8}$$

2.3 State-space representation of the CF-QUAV

The state vector, X, which is the CF position in space, angular, and linear velocities is defined as follows:

$$X = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & x_9 & x_{10} & x_{11} & x_{12} \end{bmatrix}^T, \tag{9}$$

$$X = \begin{bmatrix} z & \dot{z} & \varphi & \dot{\varphi} & \vartheta & \dot{\vartheta} & \psi & \dot{\psi} & x & \dot{x} & y & \dot{y} \end{bmatrix}^T. \tag{10}$$

A control input vector for the CF-QUAV, $\mu = \begin{bmatrix} \mu_1 & \mu_2 & \mu_3 & \mu_4 \end{bmatrix}$ is expressed as follows:

$$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \mu_4 \end{bmatrix} = \begin{bmatrix} \beta & \beta & \beta & \beta \\ 0 & -\beta & 0 & \beta \\ \beta & 0 & -\beta & 0 \\ \gamma & -\gamma & \gamma & -\gamma \end{bmatrix} \begin{bmatrix} w_1^2 \\ w_2^2 \\ w_3^2 \\ w_4^2 \end{bmatrix}, \tag{11}$$

where μ_1 is the upward force produced by the four rotors that control the height. Due to the dissimilarity in thrust between even rotors, μ_2 is in charge of the roll rotation. μ_3 indicates the difference in thrust between odd rotors, resulting in pitch rotation. μ_4 is the torque variation between the two clockwise and counterclockwise rotors that generate yaw rotation.

Substituting (11) into (3), the total moments affecting the CF-QUAV can be written as follows:

$$\mathcal{T}_b = \begin{bmatrix} \varrho\mu_2 \\ \varrho\mu_3 \\ \mu_4 \end{bmatrix}. \tag{12}$$

Substituting (6), (4), and (12) into the (5), the rotational equation can be derived as follows:

$$\begin{bmatrix} I_{XX}\ddot{\varphi} \\ I_{YY}\ddot{\vartheta} \\ I_{ZZ}\ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\vartheta} I_{ZZ}\dot{\psi} - \dot{\vartheta} I_{YY}\dot{\psi} \\ \dot{\varphi} I_{XX}\dot{\psi} - \dot{\varphi} I_{ZZ}\dot{\psi} \\ \dot{\vartheta} I_{YY}\dot{\varphi} - \dot{\vartheta} I_{XX}\dot{\varphi} \end{bmatrix} + \begin{bmatrix} \dot{\vartheta}\Im_r w_r \\ -\dot{\varphi}\Im_r w_r \\ 0 \end{bmatrix} = \begin{bmatrix} \varrho\mu_2 \\ \varrho\mu_3 \\ \mu_4 \end{bmatrix}. \tag{13}$$

Consequently, the angular acceleration can be determined as follows:

$$\ddot{\varphi} = \Lambda_1 \ \mu_2 + \alpha_1 \ \dot{\vartheta} \ \dot{\psi} - \alpha_2 \ w_r \ \dot{\vartheta}, \tag{14}$$

$$\ddot{\vartheta} = \Lambda_2 \ \mu_3 + \alpha_3 \ \dot{\varphi} \ \dot{\psi} + \alpha_4 \ w_r \ \dot{\varphi}, \tag{15}$$

$$\ddot{\psi} = \Lambda_3 \ \mu_4 + \alpha_5 \ \dot{\varphi} \ \dot{\vartheta}. \tag{16}$$

Where,

$$\alpha_1 = \frac{I_{YY} - I_{ZZ}}{I_{XX}},\ \alpha_2 = \frac{\Im_r}{I_{XX}},\ \alpha_3 = \frac{I_{ZZ} - I_{XX}}{I_{YY}},\ \alpha_4 = \frac{\Im_r}{I_{YY}},\ \alpha_5 = \frac{I_{XX} - I_{YY}}{I_{ZZ}},$$
$$\Lambda_1 = \frac{\varrho}{I_{XX}},\ \Lambda_2 = \frac{\varrho}{I_{YY}},\ \Lambda_3 = \frac{1}{I_{ZZ}} \quad .$$

Substituting (11) into (8), the non-gravitational forces exerting on the CF is expressed as follows:

$$\mathcal{F}_B = \begin{bmatrix} 0 \\ 0 \\ -\mu_1 \end{bmatrix}. \tag{17}$$

Combining (7) and (17) to get the following:

$$m \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + \begin{bmatrix} (\cos\varphi \sin\vartheta \cos\psi + \sin\varphi \sin\psi)(-\mu_1) \\ (\cos\varphi \sin\vartheta \sin\psi - \sin\varphi \cos\psi)(-\mu_1) \\ (\cos\varphi \cos\vartheta)(-\mu_1) \end{bmatrix}. \tag{18}$$

Hence, the accelerations can be expressed by:

$$\ddot{x} = -\mu_1/m(\cos\varphi\ sin\vartheta\ \cos\psi\ + \sin\varphi\ \sin\psi), \tag{19}$$

$$\ddot{y} = -\mu_1/m(\cos\varphi\ \sin\vartheta\ \sin\psi\ - \sin\varphi\ \cos\psi), \tag{20}$$

$$\ddot{z} = -g - \mu_1/m(\cos\varphi\ \cos\vartheta). \tag{21}$$

CF-QUAV model in the state-space is represented as follows:

$$f(X,\mu) = \begin{bmatrix} x_2 \\ -g - \mu_1/m(\cos\varphi\ \cos\vartheta) \\ x_4 \\ \Lambda_1\mu_2 + \alpha_1\dot{\vartheta}\ \dot{\psi} - \alpha_2\ w_r\ \dot{\vartheta} \\ x_6 \\ \Lambda_2\ \mu_3 + \alpha_3\ \dot{\varphi}\ \dot{\psi} + \alpha_4\ w_r\ \dot{\varphi} \\ x_8 \\ \Lambda_3\ \mu_4 + \alpha_5\ \dot{\varphi}\ \dot{\vartheta} \\ x_{10} \\ -\mu_1/m(\cos\varphi\ \sin\vartheta\ \cos\psi + \sin\varphi\ \sin\psi \\ x_{12} \\ -\mu_1/m(\cos\varphi\ \sin\vartheta\ \sin\psi - \sin\varphi\ \cos\psi) \end{bmatrix}. \tag{22}$$

3 Control Design

This section illustrates the CF-QUAV control model that is introduced in [14] and the proposed controller technique, which aims to address the shortcomings and achieve the objective of this work.

3.1 Crazyflie Control Model

Wooshik proposed a model for the CF simulation using Matlab with PID controllers similar to those used in CF firmware, where the desired set points (x, y, z, and yaw) are the inputs to the controller chain [14]. The reference attitude (φ and ϑ) set points were generated from the position errors that fed the PID controllers that are assigned to position (x, y). The control vectors (μ_1 to μ_4) are generated by the controller. The state outputs from the CF model are the height, positions, attitude, and heading angles.

3.2 Backstepping Controller

Backstepping is a recursive control algorithm that defines a middle-control stage for the required state. Backstepping does not omit the system nonlinearities, leading to better flexible designs than that of the other nonlinear controller techniques because some nonlinear terms would contribute to the system stability [16].

3.2.1 Height Controller It is noted that states one and two in (22) represent the height (z) and its rate of change ($\dot{z}$) as follows:

$$\dot{x}_1 = x_2, \tag{23}$$

$$\dot{x}_2 = -g - \mu_1/m(\cos\varphi \ \cos\vartheta). \tag{24}$$

It is clear that the second state is a function of the control signal (μ_1), so the choice of the Lyapunov function to satisfy positive definite can be as follows:

$$L_1 = \frac{1}{2}\, e_1^2, \tag{25}$$

where $e_1 = z_t - z_r$ is the difference between the target and real height (z) and is expressed in terms of states as:

$$e_1 = x_{1t} - x_1. \tag{26}$$

the time derivative of the (25) is as follows:

$$\dot{L}_1 = e_1\dot{e}_1 = e_1(\dot{x}_{1t} - \dot{x}_1) = e_1(\dot{x}_{1t} - x_2). \tag{27}$$

To guarantee the system stability, (27) has to satisfy negative semi-definite by bounding the derivative Lyapunov with positive definite function $B_1(e) = c_1 e_1^2$. Hence, (27) can be written as:

$$\dot{L}_1 = e_1(\dot{x}_{1t} - x_2) \leq -c_1 e_1^2, \tag{28}$$

where $c_1 > 0$. By substituting into (28), the middle control input for the second state can be deduced as:

$$x_{2t} = \dot{x}_{1t} + c_1 e_1, \tag{29}$$

resulted in defining another deviation (e_2) between the second state (x_2) and its target (x_{2t}) as follows:

$$e_2 = x_2 - \dot{x}_{1t} - c_1 e_1. \tag{30}$$

Substituting (30) into (27), the time derivative of the Lyapunov formula can be expressed as:

$$\dot{L}_1 = -e_1 e_2 - c_1 e_1^2. \tag{31}$$

The new expression doesn't satisfy the negative semi-definite condition, which leads to defining another Lyapunov formula as follows:

$$L_2 = L_1 + \frac{1}{2}\, e_2^2. \tag{32}$$

Hence,

$$\begin{aligned}\dot{L}_2 &= \dot{L}_1 + e_2\dot{e}_2 \\ &= -e_1e_2 - c_1e_1^2 + e_2(\dot{x}_2 - \ddot{x}_{1t} - c_1\dot{e}_1).\end{aligned} \tag{33}$$

Following the same steps used in (27), the (33) can be bounded by positive definite function ($B_2(e) = -c_1e_1^2 - c_2e_2^2$) to guarantee the system stability, (33) can be formulated as:

$$\dot{L}_2 = -e_1e_2 - c_1e_1^2 + e_2(\dot{x}_2 - \ddot{x}_{1t} - c_1\dot{e}_1) \leq -c_1e_1^2 - c_2e_2^2, \tag{34}$$

where $c_2 > 0$. Substituting (24) into (34) to get the control signal for the height as follows:

$$\mu_1 = \frac{m}{cos\varphi\ cos\vartheta}(-g - \ddot{z}_t - c_1\dot{e}_1 + c_2e_2 - e_1). \tag{35}$$

3.2.2 Attitude and Heading Controllers To obtain the control signal for the attitude and heading angles, the same steps should be applied to the states (3 and 4) for roll angle, while the states (5 and 6) are for pitch angle, and the states (7 and 8) are for yaw angle as mentioned in (22). The following are the control signals for φ, ϑ, and ψ, respectively:

$$\mu_2 = \frac{1}{\Lambda_1}(-c_4e_4 + e_3 - \alpha_1\dot{\vartheta}\dot{\psi} + \alpha_2w_r\dot{\vartheta} + \ddot{\varphi}_t + c_3\dot{e}_3), \tag{36}$$

$$\mu_3 = \frac{1}{\Lambda_2}(-c_6e_6 + e_5 - \alpha_3\dot{\varphi}\dot{\psi} - \alpha_4w_r\dot{\varphi} + \ddot{\vartheta}_t + c_5\dot{e}_5), \tag{37}$$

$$\mu_4 = \frac{1}{\Lambda_3}(-c_8e_8 + e_7 - \alpha_5\dot{\varphi}\dot{\vartheta} + \ddot{\psi}_t + c_7\dot{e}_7). \tag{38}$$

Where c_3 - c_8 are positive constants, e_3, $\dot{e}_3$, e_5, $\dot{e}_5$, e_7, and $\dot{e}_7$ are the deviations between the desired and real attitude and heading angles and their rates of change, respectively. The deviations between the second and the first states that describe roll, pitch, and yaw angles are e_4, e_6, and e_8, respectively.

All the acquired gain controllers have been tuned by using the genetic optimization algorithm (GA) to have the optimum values required for achieving the best performance of the proposed controller.

4 Experimental Results

In order to evaluate the performance of the proposed Backstepping controller, we used an experimental platform as shown in Figure 3, the CF-QUAV is connected via a USB radio dongle to a MacBook with M1 processor and 16 GB RAM. Many experiments have been carried out to extract the input and output data of the CF-QUAV system. A Python code was developed to drive the CF-QUAV in a spiral path with a radius of 1.0 m at different speeds ranging from 0.1 m/s to 1.0 m/s until it reaches a height of 3.0 m before moving forward for 4.0 m in x-axis direction and then landing. The desired trajectories are used as inputs to the CF-QUAV model with PID controller and the same CF-QUAV model with Backstepping controller to evaluate the performance and response of the model with each controller.

For the PID controller, it was noted that the linear PID controller failed to track the position ($x,\ y$) and yaw trajectories when the speed increased. Furthermore, there was a steady-state

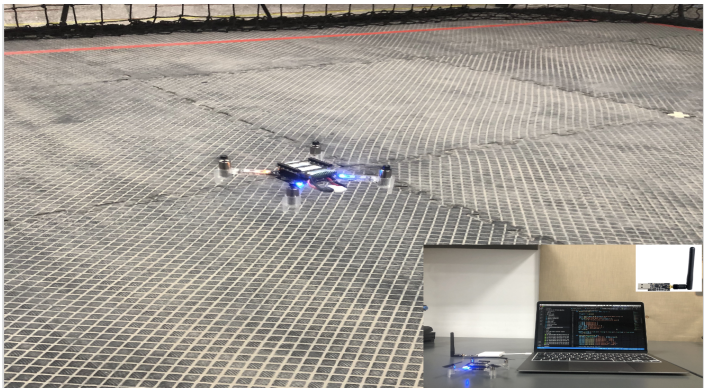


Figure 3: Crazyflie experimental setup.

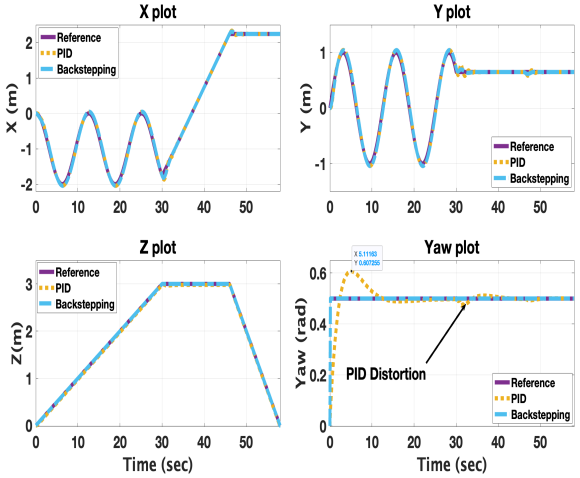


Figure 4: Crazyflie position, height, and yaw angle at a speed of 0.5 m/s.

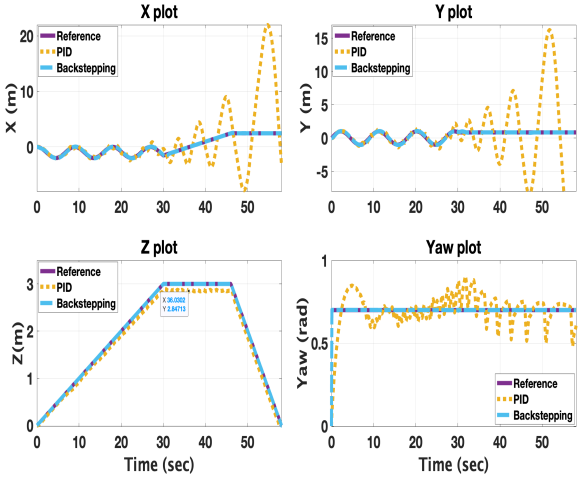


Figure 5: Crazyflie position, height, and yaw angle at a speed of 0.7 m/s.

inaccuracy at the initial speed of the yaw angle. It was worth noting that the steady-state inaccuracy for the height (z) was observed while increasing the speed.

On the other hand, the Backstepping controller improved the performance and stability of the height (z) and yaw trajectories, as well as enhanced the performance of the position (x, y) trajectory.

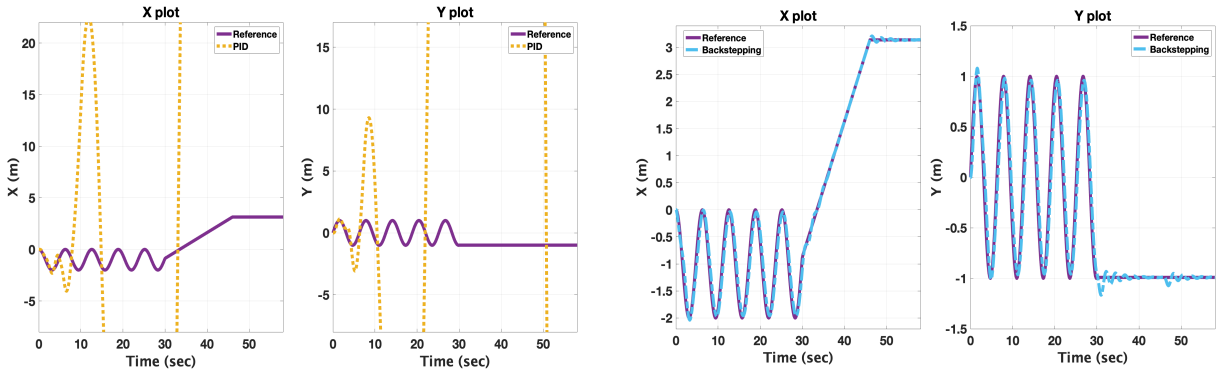
To demonstrate the differences in performance between the PID and Backstepping controllers, the simulation results of the position (x,y), height (z), and yaw angle (ψ) have been recorded at three different speeds of 0.5, 0.7, and 1.0 m/s, as shown in Figures 4 through 7, respectively.

Figure 4 demonstrates how precisely, at a speed of 0.5 m/s, the position (x and y) and height (z) for the PID and Backstepping controllers followed the desired trajectories. It should be observed that the CF-QUAV model followed the desired yaw after replacing the PID controller with the Backstepping one, with an overshoot of 21% in the yaw angle as opposed to 0% overshoot while using the Backstepping controller.

Figures 5 and 6a demonstrate that the system loses its stability with the PID controller at speeds of 0.7 and 1.0 m/s, respectively while remaining stable with the Backstepping controller for the

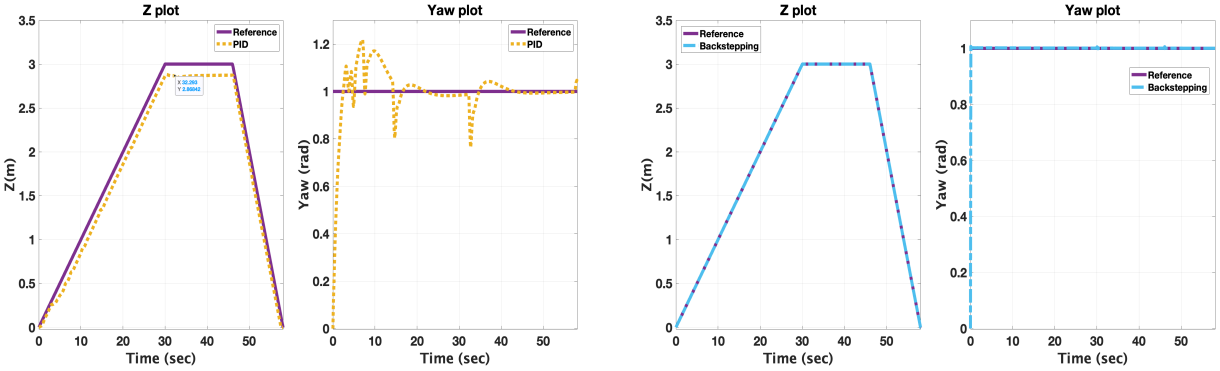
position (x and y), as illustrated in Figures 5 and 6b.

There is almost a perfect response for tracking the desired yaw trajectory with the Backstepping controller compared to the PID, as illustrated in Figures 5 and 7. Moreover, as seen in Figure 7, the PID exhibits a significant variation compared to the Backstepping controller, which accurately tracks the height trajectory while increasing the speed to 1.0 m/s.



(a) PID. (b) Backstepping.

Figure 6: Crazyflie position at speed of 1.0 m/s.

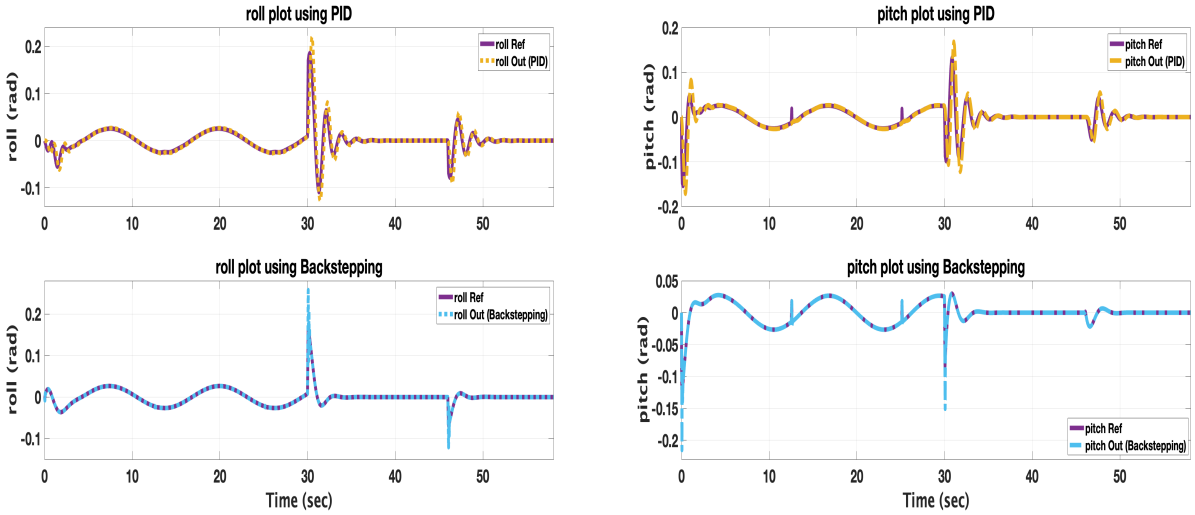


(a) PID. (b) Backstepping.

Figure 7: Crazyflie height and yaw angle at speed of 1.0 m/s.

Additionally, Figures 8 and 9 provide a comparison between the roll (φ) and pitch (ϑ) angles for the PID and Backstepping controllers at two different speeds of 0.5 m/s and 1.0 m/s. The reference for each roll and pitch varied when using the PID or Backstepping controllers due to their being generated from the error position controllers. It was evident that when the Backstepping controller was utilized instead of the PID one, the attitude angles' transient behaviors and steady-state properties were enhanced.

An analytical comparison between the PID and the Backstepping controllers has been performed to demonstrate the effectiveness of the proposed controller over the PID used in the CF model. One of the standard ways in the statistical analysis, Root-Mean-Square-Error (RMSE), is used as the evaluation criterion to measure the error between the desired trajectories and the output of the CF model when using the PID or the Backstepping controllers. The apparent discrepancy between the much less RMSE readings of the backstepping controller in comparison to that of

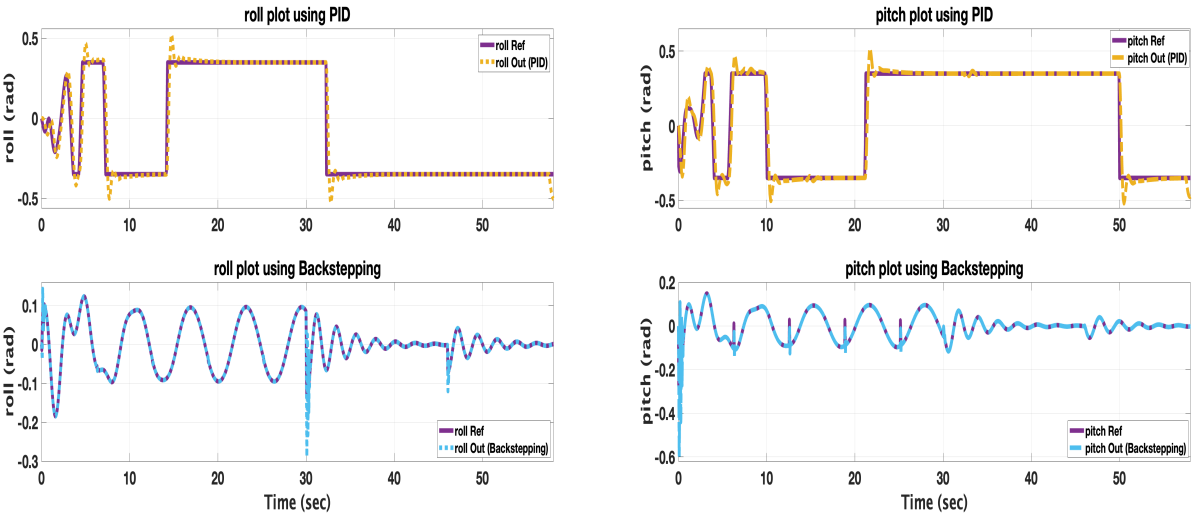


(a) PID. (b) Backstepping.

Figure 8: Attitude (roll, pitch) angles at speed of 0.5 m/s.

the PID at the three different speeds (0.5, 0.7, and 1.0 m/s) confirms the efficiency of using the proposed nonlinear controller instead of the linear one, as shown in Figures 10 - 12, respectively.

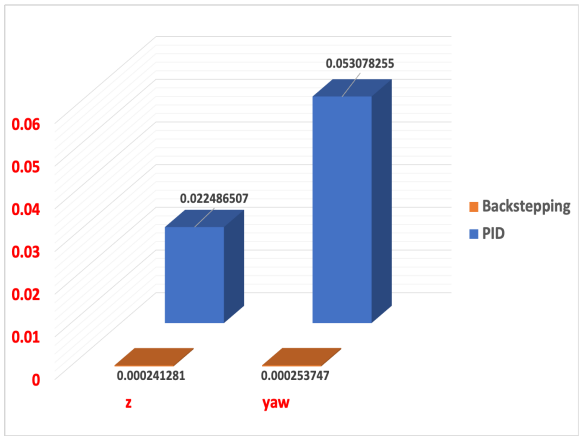
Another comparison is demonstrated in Figures 13 - 15 to show the effectiveness of the proposed Backstepping controller over the PID; the results show the error between the desired trajectories and the controllers' outputs. The results show a small error when using the Backstepping controller, especially at high speed, which confirms the effectiveness of the proposed controller over the conventional PID controller.



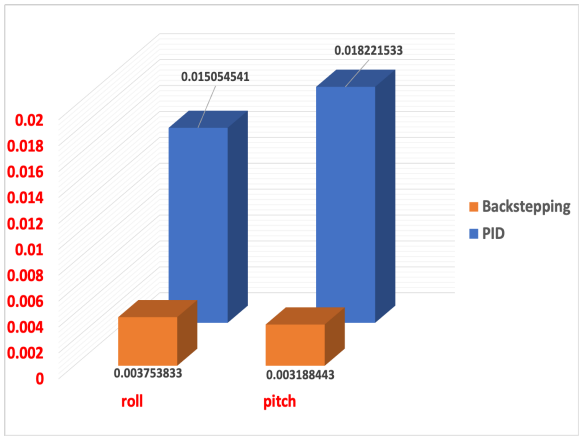
(a) PID. (b) Backstepping.

Figure 9: Attitude (roll, pitch) angles at speed of 1.0 m/s.

Finally, the desired trajectory has been simulated and compared with the outcomes of the PID and the Backstepping at three different speeds of 0.5, 0.7, and 1.0 m/s, as depicted in Figures 16

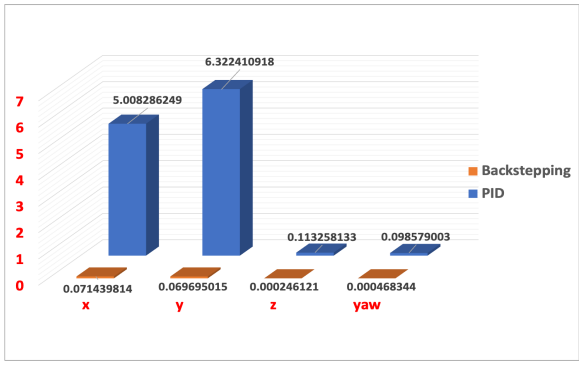


(a) Height & Heading angle.

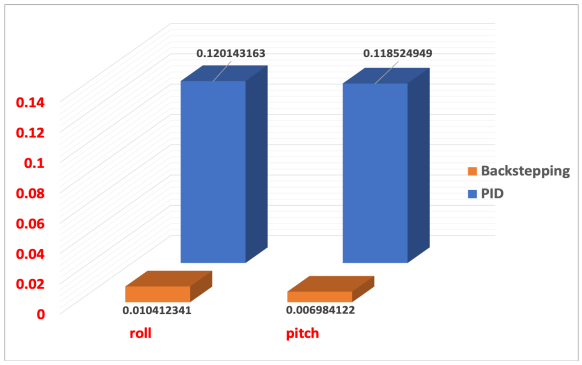


(b) Roll & Pitch angles.

Figure 10: Root-Mean-Square Error (RMSE) at speed of 0.5 m/s.

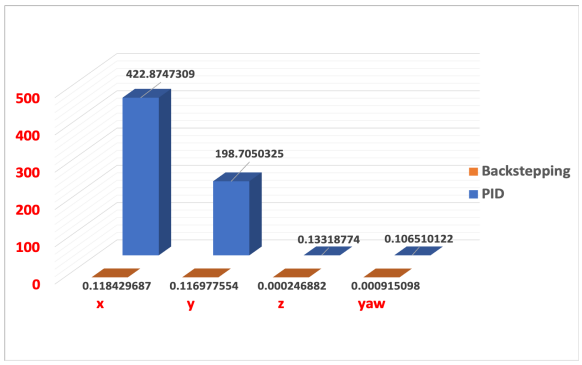


(a) Position (x, y) & Height & Heading angle.

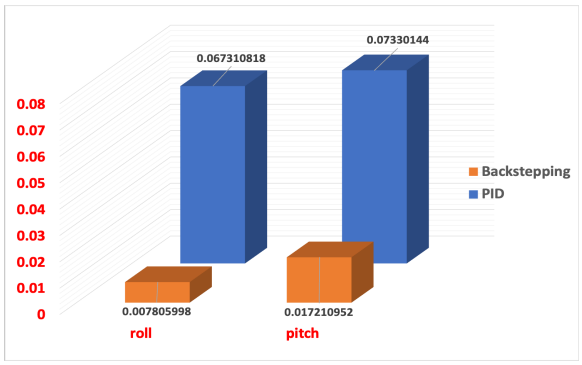


(b) Roll & Pitch angles.

Figure 11: Root-Mean-Square Error (RMSE) at speed of 0.7 m/s.



(a) Position (x, y) & Height & Heading angle.



(b) Roll & Pitch angles.

Figure 12: Root-Mean-Square Error (RMSE) at speed of 1.0 m/s..

- 18 respectively. The linear PID failed to track the desired trajectory as the speed increased, but the Backstepping created a form that nearly resembled the desired trajectory.

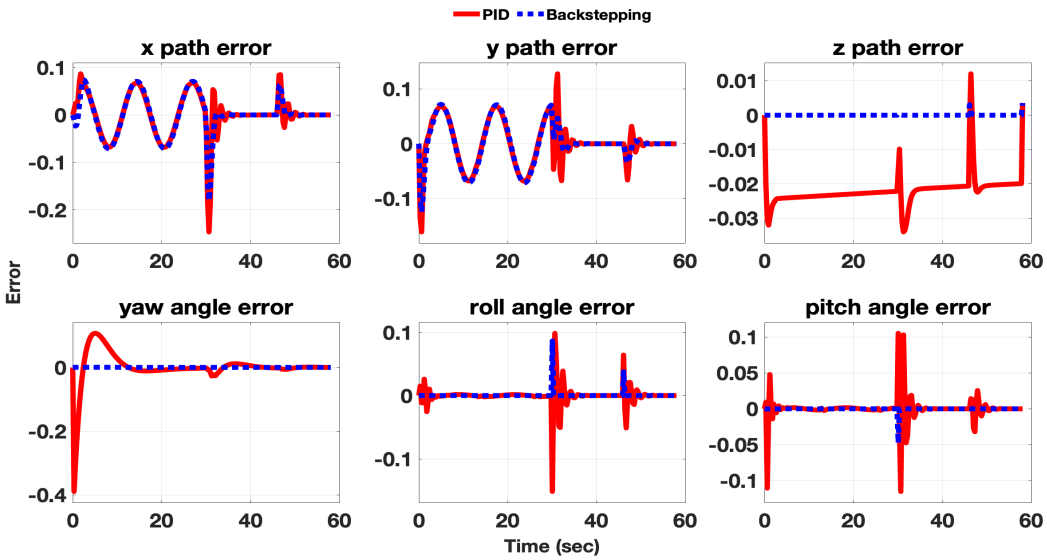


Figure 13: Error between desired and CF output trajectories at speed of 0.5 m/s

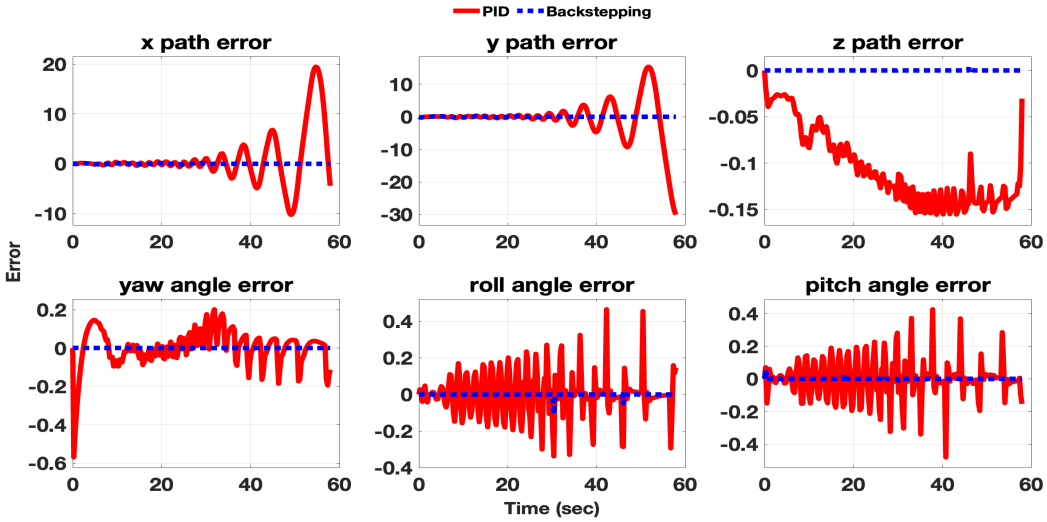


Figure 14: Error between desired and CF output trajectories at speed of 0.7 m/s

5 Conclusion

In this work, the Backstepping control technique for the CF-QUAV was introduced, and its performance and stability were evaluated using simulation-based experimental data with statistical analysis. In order to address and resolve the instability problems that occurred when applying linear PID control at real-time desired trajectories, a magnificent improvement in replacing the PID control model has been produced and confirmed using the Matlab simulation environment. All control gains are optimized using the GA in order to achieve the maximum response of the proposed technique with the shortest settling time. The simulation results showed that the proposed controller successfully improved the CF-QUAV performance in height and heading angle at all speeds. It also improved the tracking of the desired position (x, y) trajectory by utilizing the Backstepping controller up to 1.0 m/s.

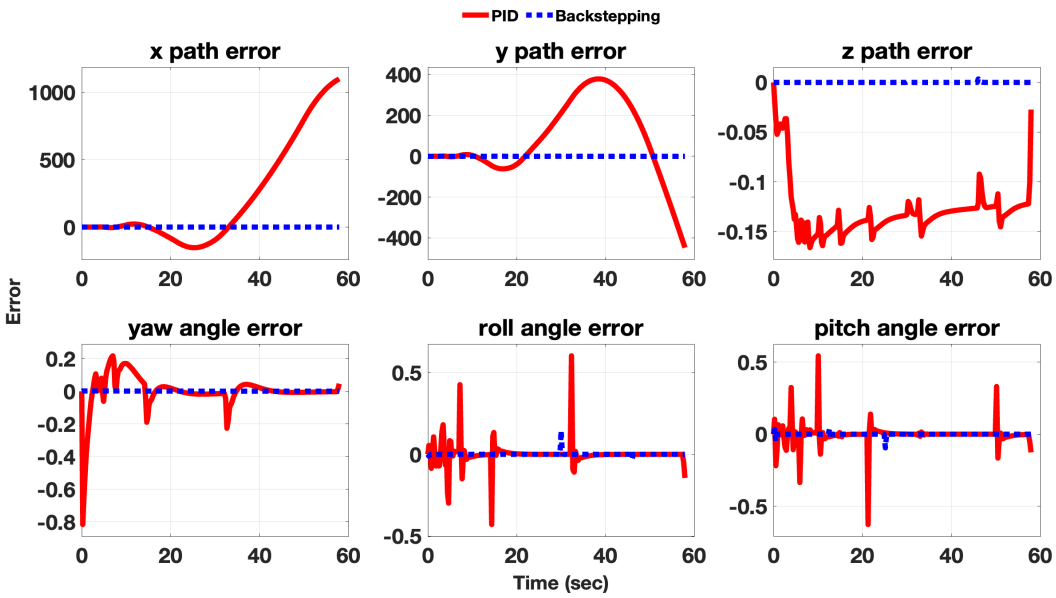


Figure 15: Error between desired and CF output trajectories at speed of 1.0 m/s

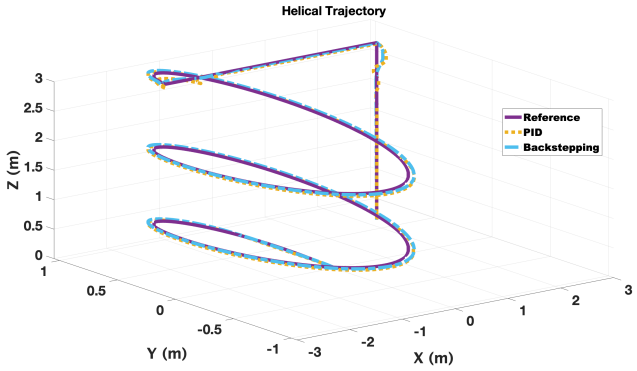
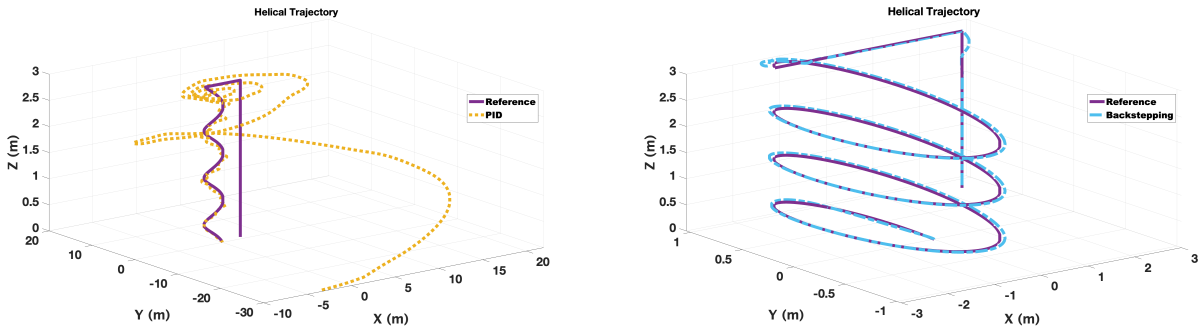
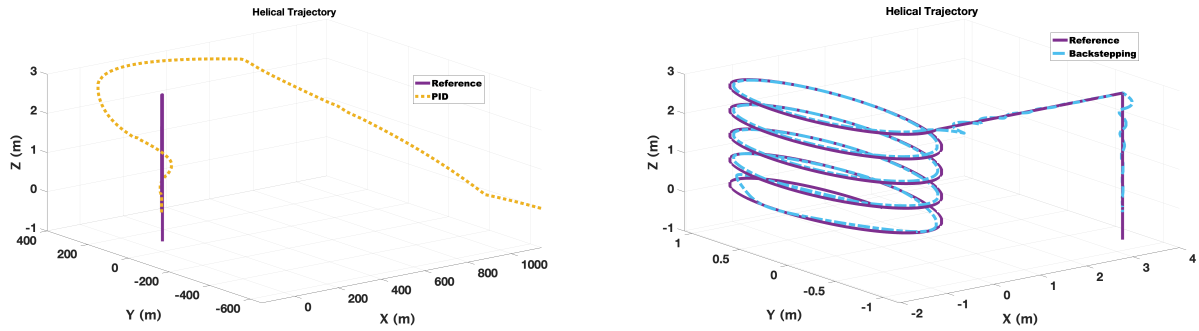


Figure 16: Crazyflie trajectory at speed of 0.5 m/s



(a) PID. (b) Backstepping.

Figure 17: Crazyflie trajectory at speed of 0.7 m/s



(a) PID. (b) Backstepping.

Figure 18: Crazyflie trajectory at speed of 1.0 m/s

Future work will focus on developing a CF-QUAV model using system identification so that it can be compared to the one used in this work.

References

[1] Hou Z, Wang W, Zhang G and Han C 2017 A survey on the formation control of multiple quadrotors *14th Int. Conf. on Ubiquitous Robots and Ambient Intelligence (URAI)* 219-225
[2] Glade D 2000 *Unmanned aerial vehicles: Implications for military operations* **No 16** Center for Strategy and Technology, Air War College, Air University
[3] Ravankar A, Ravankar A, Kobayashi Y and Emaru T 2018 Autonomous mapping and exploration with unmanned aerial vehicles using low cost sensors *Multidisciplinary digital publishing institute proceedings* **4** 44
[4] Anbaroğlu B 2017 Parcel delivery in an urban environment using unmanned aerial systems: A Vision Paper *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **IV-4/W4** 73–9
[5] Finn R L and Wright D 2012 Unmanned aircraft systems: Surveillance, ethics and privacy in civil applications *Computer Law & Security Review* **28** 184-94
[6] Li Z, Liu Y, Hayward R, Zhang J and Cai J 2008 Knowledge-based power line detection for UAV surveillance and inspection systems *23rd Int. Conf. Image and Vision Computing New Zealand* 1-6
[7] Mahfouz M, Hafez A T, Ashry M M and Elnashar G 2020 Formation reconfiguration based on backstepping-PID controller for collaborative quadrotors *12th Int. Conf. on Electr. Eng. (ICEENG)* 396-401
[8] Madani T and Benallegue A 2008 Adaptive control via backstepping technique and neural networks of a quadrotor helicopter *IFAC Proceedings Volumes* **41** 6513–8
[9] Al-Younes Y and Jarrah M A 2008 Attitude stabilization of quadrotor UAV using backstepping fuzzy logic & backstepping least-mean-square controllers *5th Int. Symposium on Mechatronics and Its Applications* 1-11
[10] Mohd Basri M A, Danapalasingam K A and Husain A R 2014 Design and optimization of backstepping controller for an underactuated autonomous quadrotor unmanned aerial vehicle *Transactions of FAMENA* **38** 27-44
[11] Jasim W and Gu D 2015 Integral backstepping controller for quadrotor path tracking *Int. Conf. on Advanced Robotics (ICAR)* 593-598
[12] Saud L J and Hasan A F 2018 Design of an optimal integral backstepping controller for a quadcopter *Journal of Eng.* **24** 46-65
[13] Saibi A, Boushaki R and Belaidi H 2022 Backstepping control of drone *Engineering Proceedings* **14** 4
[14] Wooshik Kim Crazyflie quadcopter simulation using Simmechanics *MathWorks* Available at: https://www.mathworks.com/matlabcentral/fileexchange/65469-crazyflie-quadcopter-simulation-using-simmechanics?s_tid=srchtitle
[15] Bitcraze AB *Datasheet crazyflie 2.1 - rev 3 - home — bitcraze* Available at: https://store.bitcraze.io/products/crazyflie-2-1
[16] Krstic M, Kanellakopoulos I and Kokotovic P V 1995 *Nonlinear and adaptive control design* (New York: John Wiley & Sons)