International Journal of Robotics and Control Systems
Vol. 3, No. 2, 2023, pp. 171-186
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

Improving the Size of the Propellers of the Parrot Mini-Drone and an Impact Study on its Flight Controller System

Esraa H. Kadhim [a,1], Ahmad T. Abdulsadda [b,2]

[a] Department of Electrical and Communications Engineering, University of Kufa, Iraq
[b] Department of Communication Engineering, Al-Furat Al-Awsat Technical University, Iraq
[1] engesraahadi@gmil.com; [2] coj.abdulsad@atu.edu.iq
* Corresponding Author

ARTICLE INFO

Article history
Received February 24, 2023
Revised March 17, 2023
Accepted March 22, 2023

Keywords
X-Y controller;
Quadcopter;
Matlab- Simulink;
Altitude controller;
Propellers

ABSTRACT

Unmanned Aerial Vehicles (UAVs) are widely used in transportation, delivery, surveillance and surveillance applications. The development of stable, resilient, and accurate flight based on turbulence and turbulence will likely become a key feature in the development of unique flight control systems. In this research, we studied the control system of a small Parrot mini drone, the Mambo drone, which was designed using the MATLAB program, while we added turbulence to the drone by changing the weight of the original plane in the design, where we increased the weight and calculated the vertical projection area of the propellers of the plane several times until we got the best space for the propellers able to carry more extra weight. We imposed an increase in the drone's weight due to bad conditions that the plane experienced during its flight, such as snow or dust falling on it. In order to make the aircraft bear these weather conditions without falling and colliding, we calculated an appropriate increase in the area of the aircraft wing, and we actually applied it in the MATLAB-R2021a Simulink program, and we got good results using simulation as well as in real-time inside the laboratory, turbulence was added in the simulation program. The new design of the propellers demonstrated the aircraft's ability to carry an additional payload of approximately one-third of the aircraft's weight, as shown in the roads chapter. In future work, we propose to use this design on larger aircraft with fixed propellers and to study the effects of other weather conditions on UAVs, such as the effect of temperature, humidity, and others.

1. Introduction

In recent years, the scientific community has shown considerable interest in airborne hovering vehicles in general and autonomous flying vehicles in particular. The most common UAV configuration uses quadrotors. Its ability to hover quietly at low hustles, quickly build up speed, and change its attitude as needed makes it an excellent choice for a small drone. Many quadrotor applications have been developed in the fields of aerial photography, shipping and delivery, gathering information, geographic mapping, search and rescue, and military surveillance. One of the key goals of quadrotor design has been to develop a reliable and perturbation-resistant flying vehicle. Because of their under-actuated mechanism and compact size, quadrotor is extra susceptible to uncertainties and turbulence than other kinds of aerial robots. Wind disturbance, air friction, and uncertain weather are just a few examples of such uncertainties. The research question is how to make the drone maintain its balance if it is exposed to bad weather conditions during its flight. The goal of the research paper is to obtain the best vertical projection area for the propellers of the aircraft, which is commensurate with the increase that can be obtained in the weight of the drone due to bad conditions.

Many researchers have used quadrotor design to study the properties and components of drones as well as to build and deploy them in various domains. One of the most well-known of these studies involves Parrot rolling spider mini-drone quadcopter elevation control system. The authors in that vehicle presented a PD change strategy using the characteristic ratio assignment (CRA) process. Based on the collection of the typical polynomial constants of the closed-loop organization that meets the given performance criteria, the CRA technique was utilized to determine the suitable controller parameters. The performance of the suggested controller design method was examined using height control simulations on a Parrot rolling spider mini-drone quadcopter system [1]. Other researchers used a Simulink Support Package for Parrot Minions to build nonlinear control for Parrot Mambo or Parrot Rolling Spider quadcopters. The full rigid body model of the flying vehicle is considered that does not make the assumption of small Euler angles. The nonlinear dynamics inversion and integrator backstepping approach was utilized to synthesize the control [2]. In another study, the PI-PD and Fuzzy PI-PD (FPI-PD) constructions were designed and deployed to tackle the Parrot Mambo Minidrone's position control problem. A nonlinear measured model of the Parrot Mambo Minidrone was constructed to acquire the mini drone's control models. First, a PI-PD control scheme for height and position control strategy was built using examples. FPI-PD controllers were then developed and used in the mini-position drone's control loop to address the coupled nonlinearities. The findings of the study provided comparative real-world experiment data to reveal that the proposed control systems outperformed the built-in control from the manufacturer.

According to the results, the FPI-PD control system outperformed the built-in PD and created PI-PD control systems [3]. Other authors demonstrated and successfully built a three-loop uncertainties compensator (TLUC) and an exponential reaching law sliding mode controller (ERSM) on a quadrotor Unmanned Airborne Vehicle (UAV). The TLUC evaluates unidentified time-varying disturbances to reduce their effect. The ERSM is integrated by utilizing the Lyapunov stability philosophy to achieve rapid response with the least amount of chattering. The goal of this research was to critically examine the PD controller project and structure tuning rules for the six-degree-of-freedom wave controller. This paper's unique feature is that the TLUC can estimate and adapt to the unknown time-varying turbulences in the three feedback loops [4]. This study proposes a modified adaptive sliding mode control for path tracking of mini-drone quadcopter unmanned aerial vehicles. Further, it aims to demonstrate the effectiveness of a nonlinear adaptive control technique for achieving the mini-drone quadcopter system's intended performance. Besides providing mathematical modeling and nonlinear dynamic characteristics of a mini-drone quadrotor, a modified adaptive descending mode algorithm is proposed using an adaptation law based on the Lyapunov constancy method. The boldness ring and elevation ring control systems allow the controller's nonlinear adaptive performance to compensate for disturbances and parameter perturbations. The effectiveness of the proposed control technique is validated using Matlab simulations and compared with the results of the previous strategy presented in [5]-[8].

1.1. Equipment

The Parrot mini drone was chosen because of its versatility, low cost, and interoperability with MATLAB and Simulink. The quadrotor has an internal control system that allows it to self-stabilize during flight, making it a dependable platform for flight tests. Fig. 1 shows the Parrot mini drone's rigid body frame, and Fig. 2 shows the simulation model of the parrot mini drone Mambo, which houses a battery, four sensors, four motors, four blades, and four bumpers. The platform supports both Wi-Fi and Bluetooth, allowing the Parrot mini drone to communicate with a variety of devices.

Fig. 1. The parrot mini drone Mambo

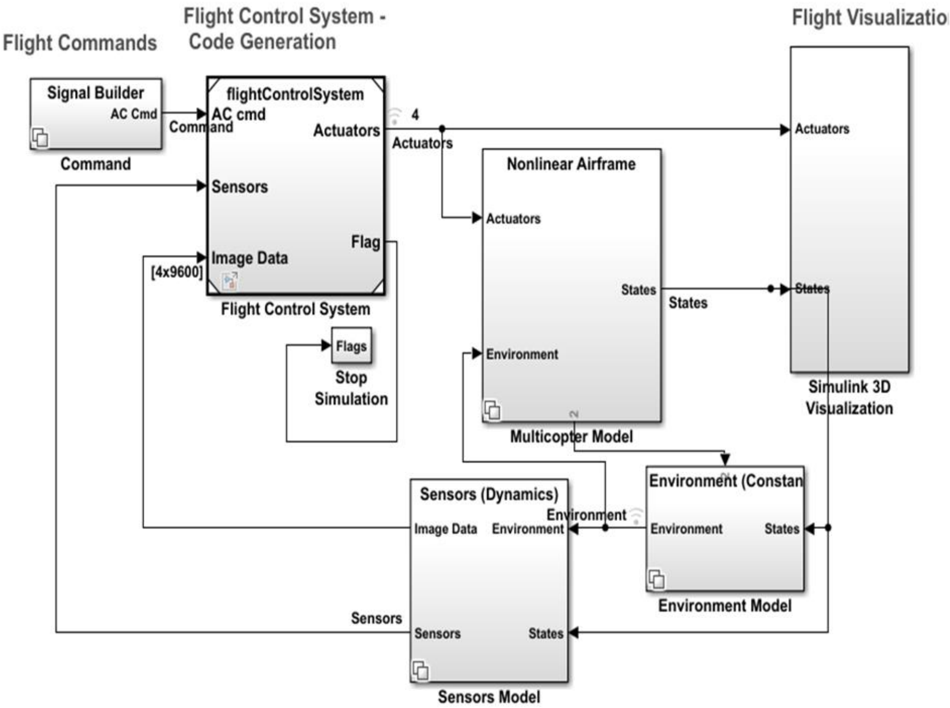

Fig. 2. Simulation model of parrot mini drone-Mambo

1.2. The architecture of the Control System (Hovering Control)

The design of the control system architecture to operate the quadcopter during the flight maneuvers required to complete the objective will be presented in this study. The initial objective is to keep the hovering under control. Given a three-dimensional reference, an X-Y location, and altitude, which are all consistent in this task, it should be verified that it is stable when hovering. The mini-drone will only have a change of reference in the X-Y location while monitoring the path, which is developed and continuously updated by the path planning algorithm [9]. If we consider that the micro drone's camera should be facing down almost orthogonally during these motions in order to see the track in the correct orientation, we can expect modest pitch and roll angles. This indicates that the route tracking task can also be accomplished using the hovering control system design. The control system is built using Douglas' technique [10]. The small drone is regarded as the plant in this architecture. It takes four motor speeds as inputs, spins the propellers, and produces forces and torques as plant outputs.

To fulfill the control target, the little drone must hover at a specified altitude. The four motors must be driven individually to obtain this output. The motor mixing algorithm (MMA) may be used to directly regulate yaw, pitch, roll, and thrust instead of using the four motor speeds. Thrust is always directed along the body frame's Z-axis. If the drone is flying at severe roll or pitch angles, there is a connection between the altitude change and the horizontal motion if the thrust is ordered. If a racing drone controller is needed, this coupling should be chosen because it can fly at higher roll and pitch angles. Low roll and pitch angles should be considered for this rudimentary hover controller. As a result, a change in thrust has a considerable impact on the altitude rate. To begin with, we should consider altering the height using a PID controller to manage the thrusts. The state to measure and provide feedback on is the drone height when compared to the reference. The altitude controller then uses the error to determine how to direct the thrust. If the micro drone hovers at a lower height, for example, the error becomes positive, and the push instruction is divided evenly among the four motors, causing the drone to climb.

However, disturbances such as wind gusts can force the drone to roll or pitch slightly, causing the push to cause the drone to move horizontally, drifting away from the reference point. As a result, the controller is insufficient, and it may be improved by keeping the roll, pitch, and yaw angles constant at zero degrees, allowing propulsion to simply affect altitude and preventing the drone from straying away. Because the thrust, roll, pitch, and yaw instructions are separated, they may be regulated individually. As a result, three more feedback controllers for yaw, pitch, and roll may be added. These angles are now used to estimate and provide feedback on plant outputs and statuses. As a result, the micro drone's height is maintained, and it faces ahead. Although this hover controller has improved, it still has significant limitations. If numerous wind gusts, for example, flow in the same direction as disturbances, the controller nevertheless permits the drone to wander away slowly [11]. The drone is not returned back to its reference point in this control system implementation. Improvements may be made by remembering that pitch and roll angles aren't always zero while hovering and that they can take on non-zero values. For example, the small drone might fly in a strong, steady breeze and maintain its reference point by leaning into the wind at a given inclination. As a result, we should create a ground position controller that detects when the quadcopter deviates and makes the necessary adjustments to return it to its original X-Y coordinates [12]. Position inaccuracy is connected with roll and pitch motion due to the coupling of forward, backward, left, and right motions with roll and pitch procedures.

To calculate the position error, the measured X-Y position should be fed back and compared to the reference signal. If there is no signal with the path-planning algorithm's series of position points to follow, the reference position may be temporarily adjusted to (0, 0). It's the same as hovering just over the take-off point. The position controller receives the position error as an input and sends the pitch and roll angles as outputs. The pitch and roll controllers use these angles as reference signals [13]. The position controller makes them in this manner. The reference signals are received by the roll and pitch controllers in the inner loop from the position controller in the external loop. They're loops that have been cascaded. It's worth noting that the position controller should additionally get the measured yaw angle. Pitch and roll angles are reported relative to the drone frame, whereas X-Y position errors are expressed relative to the environment reference frame. This implies that a movement in the X and Y world directions can only be determined by knowing the yaw angle, which is required by the position controller to convert from the drone X-Y frame to the environment reference frame. The structure of the architecture discussed previously is depicted in Fig. 3 [14]-[17].

In this research, the technique used in conducting the research was explained, the design used was explained, the mathematical model of the quadcopter was derived, the laws that were used in the theoretical calculations, as well as the results of the input and output of the quadcopter flight control system in the case of the presence and absence of disturbances and using the old and modern design.

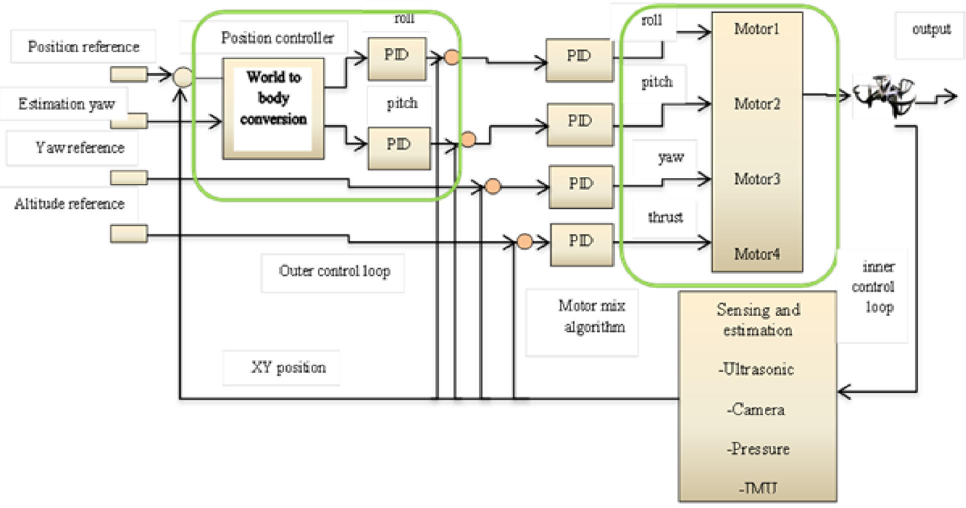

Fig. 3. Overview of the feedback control system [18]

1.3. Quadcopter Model

Many scholars have created and utilized quadrotor dynamics based on the effort of Lagrange and Newton–Euler. Fig. 4 depicts quadrotor shapes, edges, and forces [1].

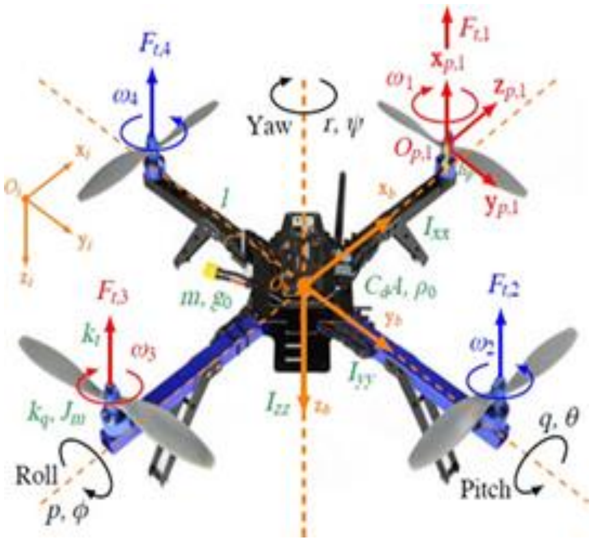

Fig. 4. Quadcopter parameter [19]

Where $X(t) = [\varphi(t), \theta(t), \Psi(t), x(t), y(t), z(t)]^T$ is expressed as (1) and (2).

$$\ddot{X}(t) = FT\left(\dot{X}(t)\right) + G(t)\big(X(t)\big)U(t) + D(t) \tag{1}$$

$$\ddot{X}(t) = F\left(\dot{X}(t)\right) + \Delta F\left(\dot{X}(t)\right) + G(t)\big(X(t)\big) + \Big(\Delta G(t)\big(X(t)\big)\Big)U(t) + D(t) \tag{2}$$

Where $F(\dot{X}(t))$ and $G(t)(X(t))$ are the nominal dynamics, which are expressed as (3) and (4).

$$F\left(\dot{X}(t)\right) = \left[\frac{iy-iz}{ix}\dot{\theta}\Psi - \frac{jr}{ix}\dot{\theta}\omega r \;\; \frac{iz-ix}{iy}\dot{\varphi}\Psi + \frac{jr}{iy}\varphi\dot{\omega}r \;\; \frac{ix-iy}{iz}\dot{\theta}\dot{\varphi}\right] \tag{3}$$

$$F\left(\dot{X}(t)\right)=\begin{bmatrix}\frac{iy-iz}{ix}\dot{\varphi}\dot{\Psi}-\frac{j\dot{r}}{ix}\dot{\theta}\omega r\\ \frac{iz-ix}{iy}\dot{\varphi}\dot{\Psi}+\frac{jr}{iy}\varphi\dot{\omega}r\\ \frac{ix-iy}{iz}\dot{\theta}\dot{\varphi}\end{bmatrix} \quad (4)$$

$G(X(t))=diag\Psi$ is explained in [20], [21], [22]. $u1$, $u2$, $u3$, and $u4$ are the attitude and altitude control inputs, whereas ux, uy are axillary control inputs meant to provide the orientation signals of the pitch and roll angles, desired roll $\emptyset d$ and desired pitch θd, respectively. The pitch and roll are subsequently controlled in $u2$, $u3$. The axillary controls and control signals are as shown (5).

$$\begin{gathered}u1=b(\omega_1^2+\omega_2^2+\omega_3^2+\omega_4^2)\\ u2=ble(\omega_1^2+\omega_4^2-\omega_3^2+\omega_2^2)\\ u3=ble(\omega_1^2+\omega_2^2-\omega_3^2-\omega_4^2)\\ ux=(sin\ sin\ \theta\ cos\ cos\ \Psi\ cos\ cos\ \emptyset\ +sin\ sin\ \emptyset\ sin\ sin\ \Psi)\\ uy=(sin\ sin\ \theta\ sin\ sin\ \Psi\ cos\ cos\ \emptyset\ -sin\ sin\ \emptyset\ sin\ sin\ \Psi)\end{gathered} \quad (5)$$

The wanted roll and pitch can be found by using the formula (6) and (7) [23], [24].

$$\emptyset d=(ux\ sin\ sin\ \Psi d-uy\ cos\ cos\ \Psi d) \quad (6)$$

$$\theta d=\left(\frac{ux\ cos\ cos\ \Psi d+uy\ sin\ sin\ \Psi d}{cos\ cos\ \emptyset d}\right) \quad (7)$$

Being aware of this is shown in Table 1.

Table 1. Description of the equation

ψ, θ and ϕ	**yaw, pitch and roll angles correspondingly [rad]**
ix, iy and iy	Moments of inertia about the body edge in x, y, and z [kg.m^ 2].
m	Mass total [kg]
g	The force of gravity [m/s^2]
jr	The inertia of the rotor [kg.m^2]
b	Coefficient of thrust [kg.m]
d	Coefficient of drag [kg.m^2]
le	Arm length at the moment [m]
$\boldsymbol{\psi}d$	The yaw angle that you want [rad]
ωi	Angular velocity ($i=1,2,3,4$) [rad/s]
$\omega r=-\omega 1+\omega 2-\omega 3+\omega 4$	Balance of rotation around the vertical axis [rad/s]

2. Method

The wings or propellers, it will pass through an airflow, separating it into two flows if it moves through the air (which has its own atmospheric pressure and speed) at a given speed with a specified upward inclination (called the angle of attack). The one that runs along the top of the profile will do so at a faster rate than the one that runs through the bottom (Venturi effect). Furthermore, according to Bernoulli's theorem, faster speed equals lower pressure. As a result, the upper wing's surface is subjected to less pressure than the inferior wings. An airfoil is normally made out of a body with a particularly designed shape to capture the majority of the force created by the fluid's speed and pressure fluctuations as it flows through an air stream [25].

In this paper, we impose a kind of disturbance on the drone by imposing an increase in its weight due to external conditions such as snowfall or quantities of dust on it, which leads to a defect in the drone system and its downfall. We have studied the impact of these disturbances on the results of the controllers. We have imposed a solution to this problem by increasing the area of the propellers by a certain amount. We have obtained good results, as will be explained further in the results section.

The reference mass of the vehicle is (0.063 kg) plus the motor (0.033 kg). We get a total mass equal (0.096 kg) [25], [26], increasing to (0.105 kg) in the Simulink block. The drone is flying and is stable if the force of lift (L) that gets from the rotation of the propellers is equal to the force of attraction of the earth to the body (F) [27], as shown in Fig. 5.

Where is the formula (8) and (9).

$$L = 0.5 \times \rho \times A \times V^2 \times C \tag{8}$$

$$\text{and } F = mg \tag{9}$$

P is the air density constant (1.3 kg/m^3), A is the area of the propeller (0.0036 m), V is the propeller speed, C is lift coefficient, m is the mass of drone, and g is a constant of gravitation (9.81 m/s^2).

Fig. 5. Balance of forces affecting a drone.

Under normal flight conditions, the vehicle mass is (0.096kg)

$$F = mg = 0.096 * 9.81 = 0.94176N = \text{lift force}$$

$$L = 0.5 \times \rho \times A \times V^2 \times C$$

$$V^2 \times C = \frac{0.94176}{0.5 \times 1.3 \times 0.0036} = 402$$

Under the influence of bad weather conditions, when we assumed an increase in vehicle weight from (0.093 to 0.105 kg), we can determine the area of the propeller as (10).

$$A = \frac{L}{0.5 \times \rho \times V^2 \times C} = \frac{0.105 \times 9.81}{0.5 \times 1.3 \times 402} = 0.004\, m^2 \tag{10}$$

Note we have changed the weight of the drone and the area of the propellers from the icons shown in Fig. 6. Table 2 shows other experiments. We repeated the same procedure to calculate the area of the new propeller in each case, and we got the same good results after applying the new results in Simulink. We discovered that the fan space could not be increased by more than 0.0045 m2 because the size of the propellers must be proportional to the size of the frame. as for the flow chart of the test to calculate the area of the new propeller in each case as shown in Fig. 7, in order to get good results after applying the new results in Simulink.

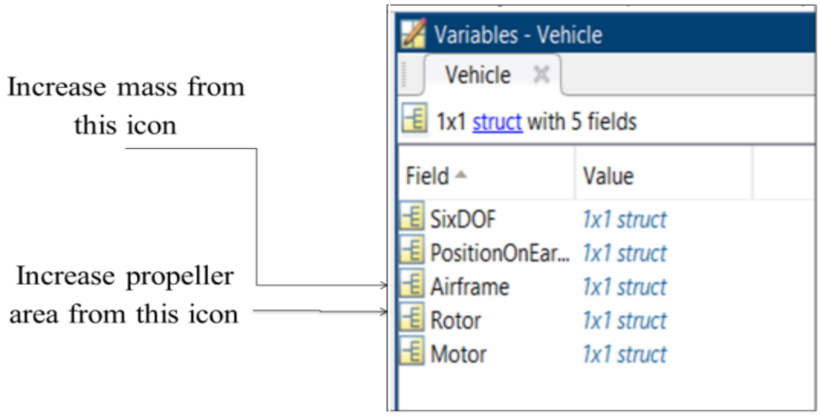

Fig. 6. Icons from workspace

Table 2. The table shows the area of the new propellers for each increase in the weight of the aircraft. We note that the best space for the propellers is ($0.0043m^2$) which fits all the weights in the table as well as the original weight of the aircraft

New weight (kg)	Calculation of propellers area (m^2)	The results
0.105	0.004	The drone flies and is stable
0.110	0.0042	The drone flies and is stable
0.115	0.0043	The drone flies and is stable
0.120	0.0045	Unstable system

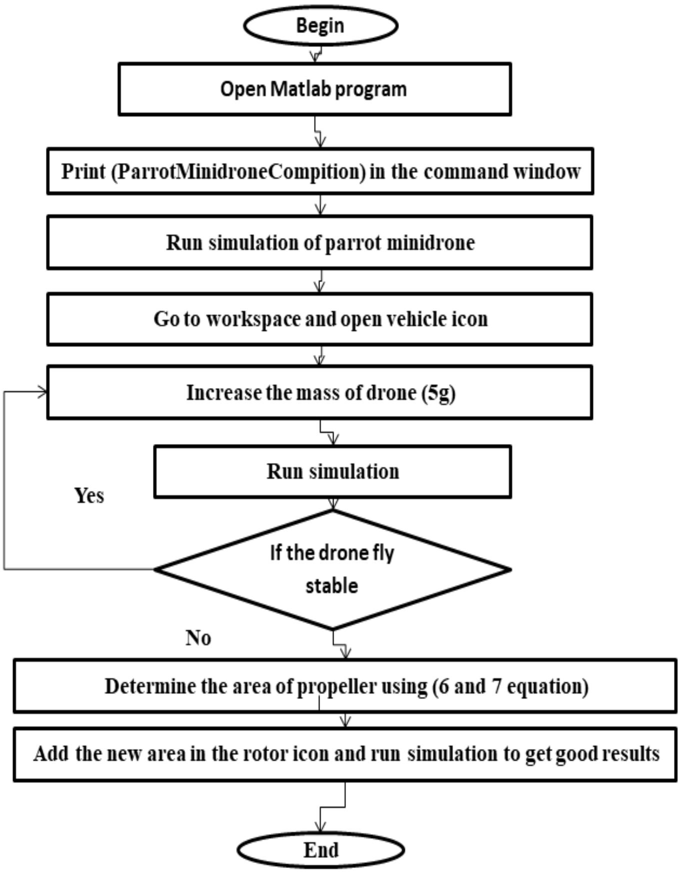

Fig. 7. The workflow algorithm

3. Results

In this section, we will draw the input and output signals for all the controllers in the flight management system in the presence and absence of turbulence. Drawing the results of turbulence after increasing the area of the propellers to make the plane able to bear some extra weight due to weather conditions. In the experiment, we used the design of a Parrot Mini Drone. Worked on changing the weight of the plane and calculating the vertical projection area through the laws mentioned in the section on methods. Applied this in the MATLAB simulation program and explained its effect on the drone flight control system.

3.1. X-Y Position Controller

The outer loop controller for X-Y positions is shown in Fig. 8. It makes the drone return to its original position and the original location (0, 0). Delivering the signals to the internal loop controller for pitch/roll (or attitude) [28]. Fig. 9 is the result of the drone in coordinate X-Y position signals without disturbance, and Fig. 10 is the result of the drone in coordinate X-Y position signals with disturbance. The results for the input and output of the XY position controller after adding the propeller area are shown in Fig. 11.

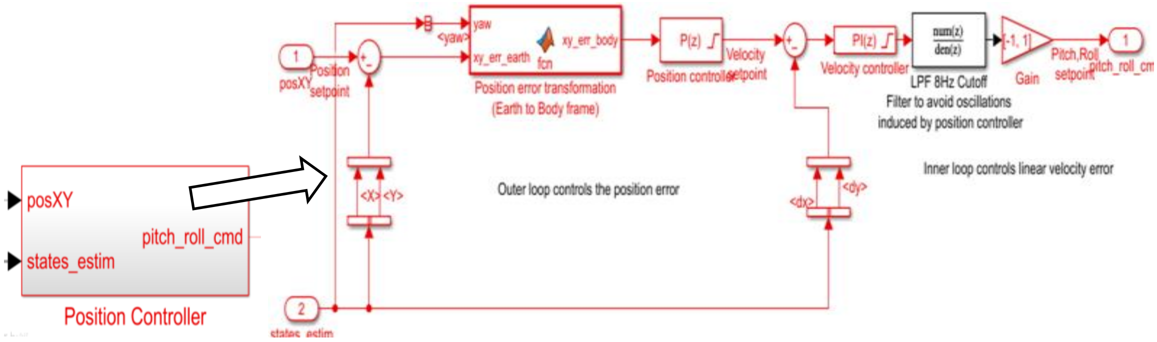

Fig. 8. X-Y position blocks and structure controller

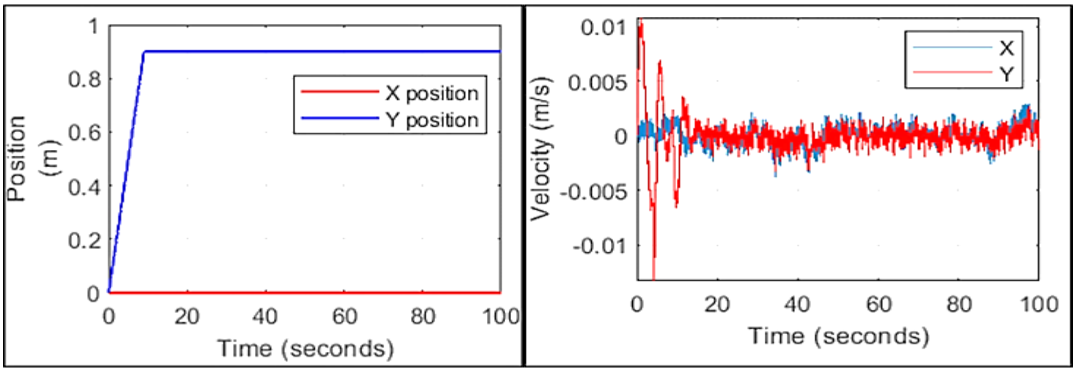

Fig. 9. The X-Y position signals without disturbance

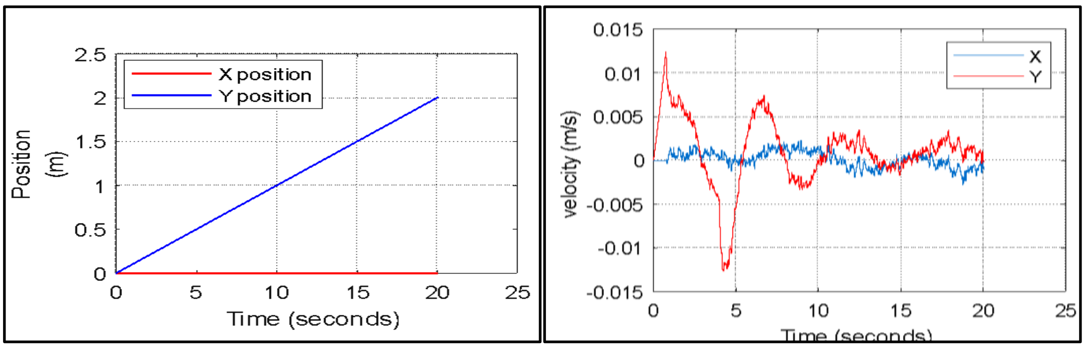

Fig. 10. The X-Y position signals with disturbance

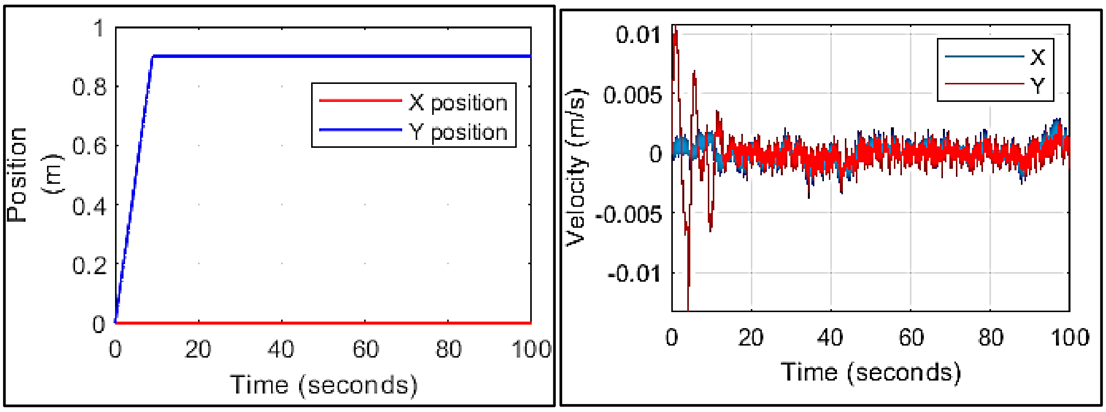

Fig. 11. The input and output of the XY position controller after increasing the area of the propeller

3.2. Attitude Position Controller

The controller for the internal loop is shown in Fig. 12. These PID controllers contain force and torque commands as outputs, which are then communicated to the (Motor Mixing Algorithm) MMA. The results of testing the altitude controller giving signals without disturbance are shown in Fig. 13, and Fig. 14 shows for results of testing the altitude controller giving signals with disturbance. Fig. 15 is the result of the Attitude controller's input and output signals after adding the propeller area.

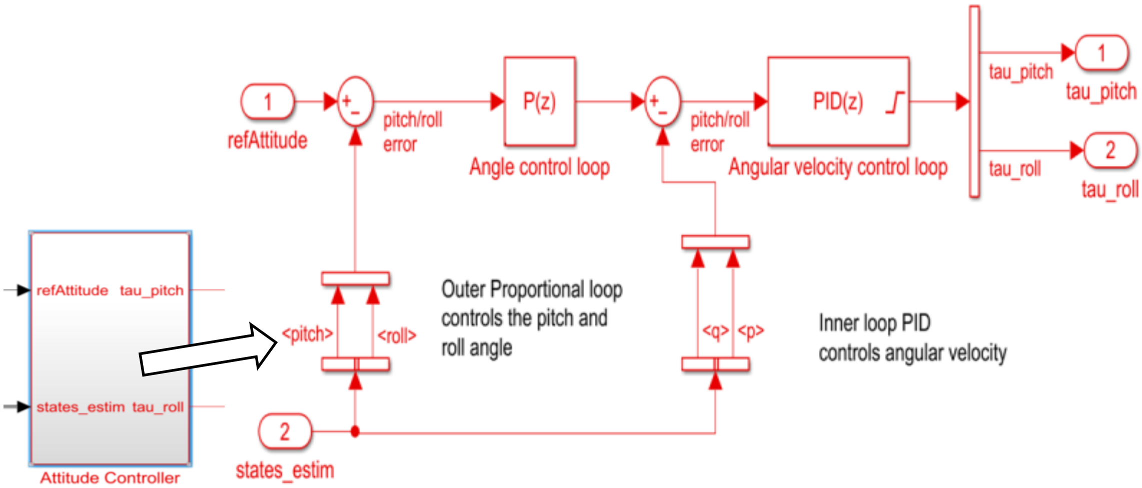

Fig. 12. Attitude blocks and structure controller

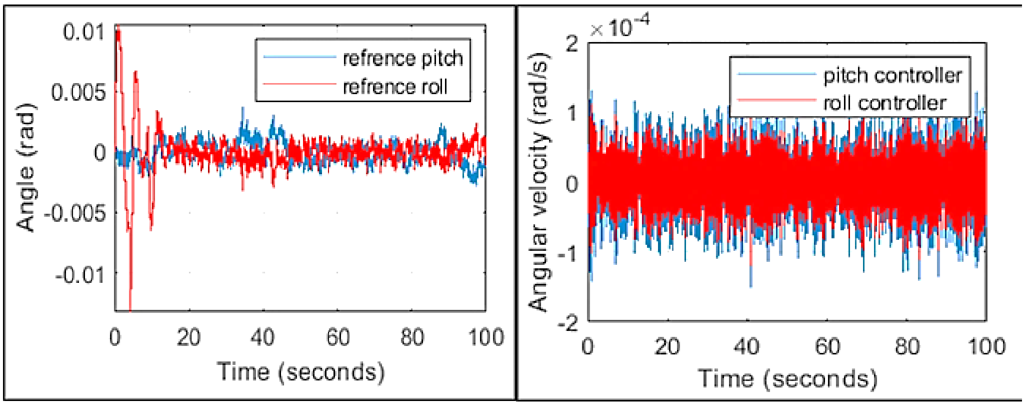

Fig. 13. The Altitude controller signals without disturbance

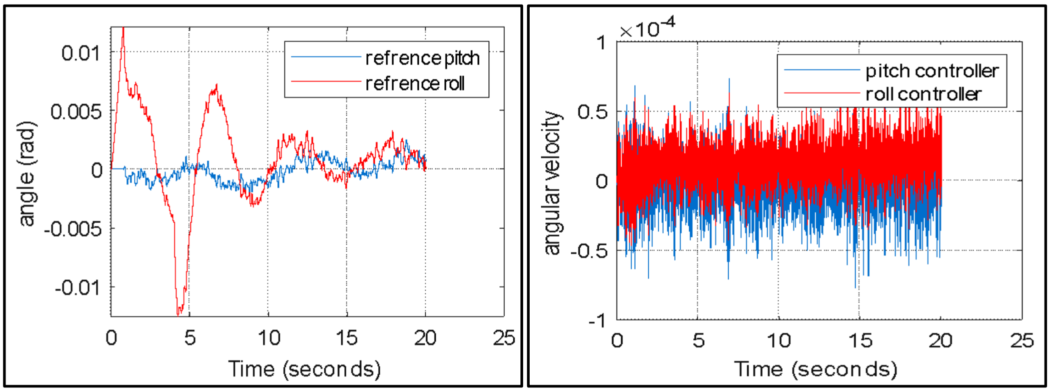

Fig. 14. Attitude controller signals with disturbance

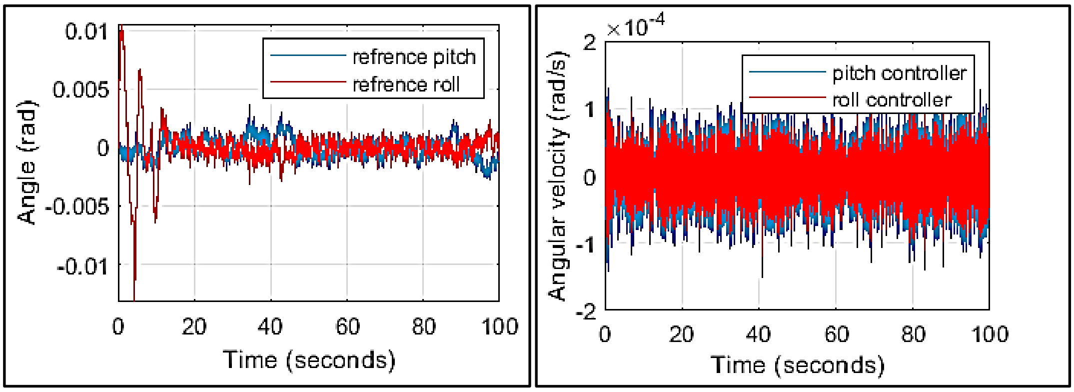

Fig. 15. The input and output of Attitude controller signals after increasing the area of the propeller

3.3. Yaw Controller

These PID controllers contain force and torque commands as outputs, which are then communicated to the (Motor Mixing Algorithm) MMA shown in Fig. 16. Testing for the yaw controller signals without disturbance can be seen in Fig. 17, and with disturbance shown in Fig. 18. Fig. 19 is the result of testing the input and output of the position control signal after increasing the propeller area.

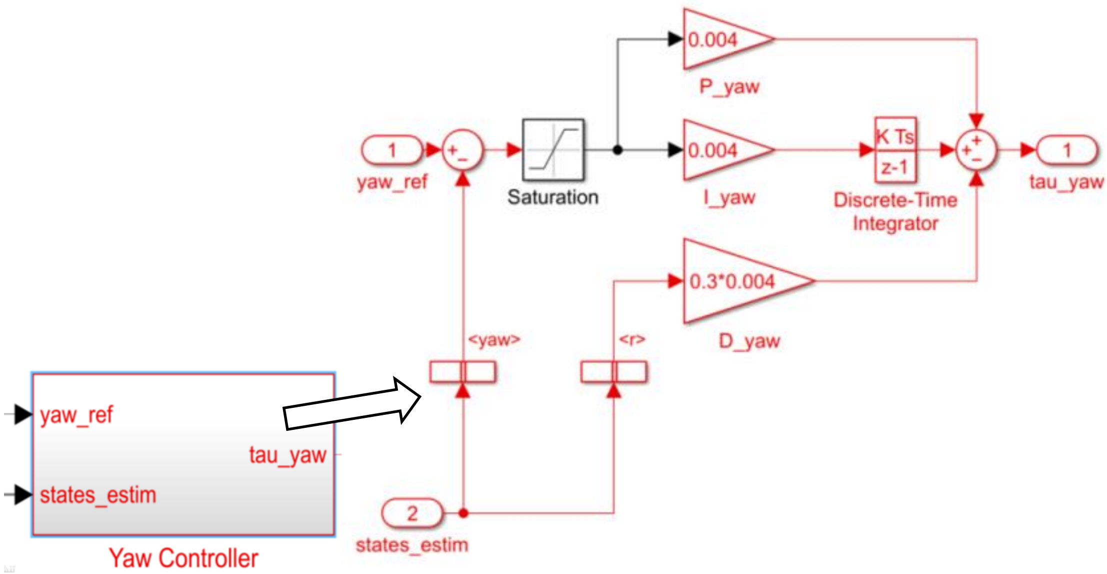

Fig. 16. Yaw blocks and structure controller

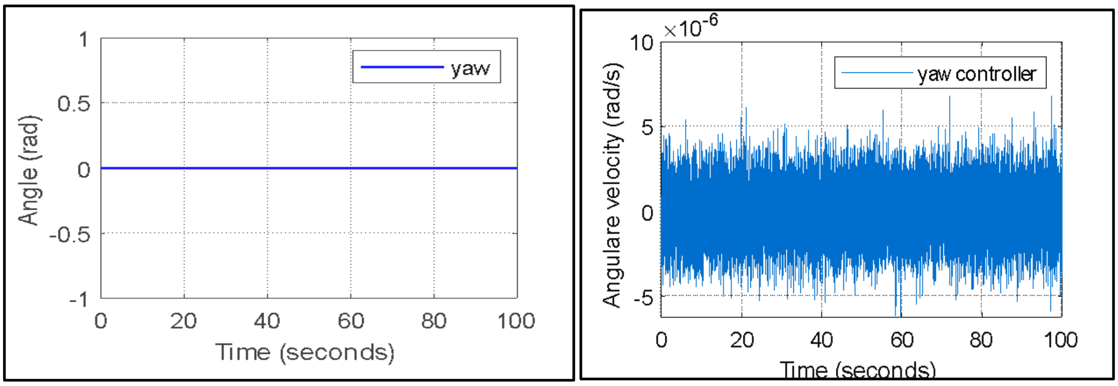

Fig. 17. Yaw controller signals without disturbance

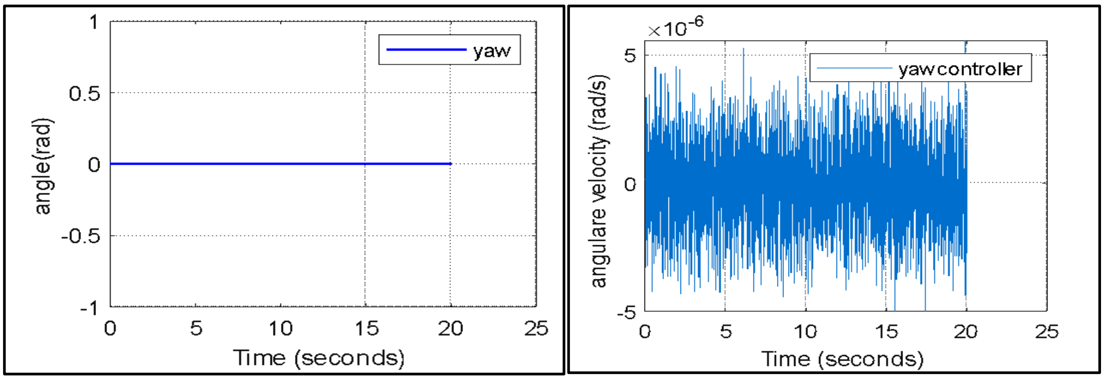

Fig. 18. Yaw controller signals with disturbance

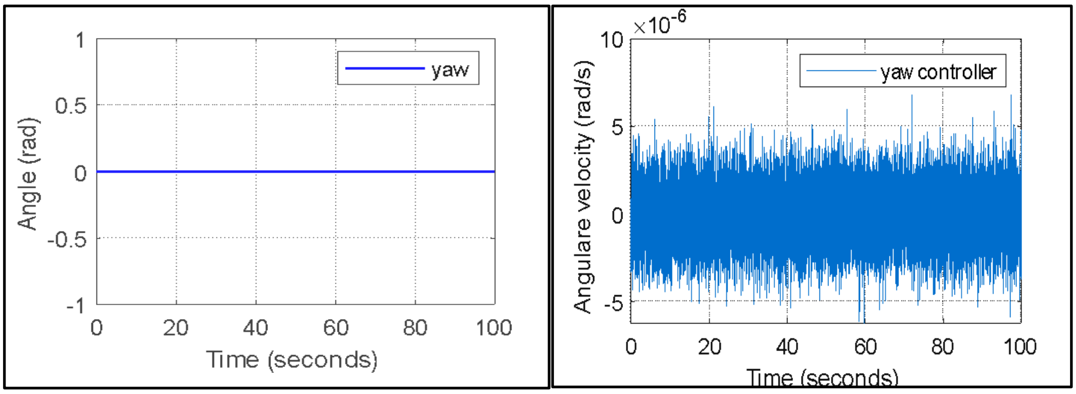

Fig. 19. The input and output of position controller signals after increasing the area of the propeller

3.4. Altitude Controller

The altitude controller is set up with a PID controller shown in Fig. 20. In this approach, the proportional gain is multiplied by the altitude error generated by the sonar sensor, and the derivative gain is multiplied by the gyroscope's altitude rate measurement, which is a less noisy signal than the ultrasound signals. It's important to note that the z-axis in the drone's coordinate system points down, so the altitude value will always have a negative sign in front of it in the control system (expressed in meters) [29], [30]. Fig. 21 is a result of testing for the drone to complete its flight for the time period specified in the simulation, which is 100 seconds without disturbance. Fig. 22 is the result of drone testing to complete its flight within a period of time and an altitude signal with a predetermined disturbance in the simulation, namely 25 seconds. Fig. 23 is a test to determine whether the flight of the drone is running well or not, namely by providing input and output positioning signals after the propeller area increases.

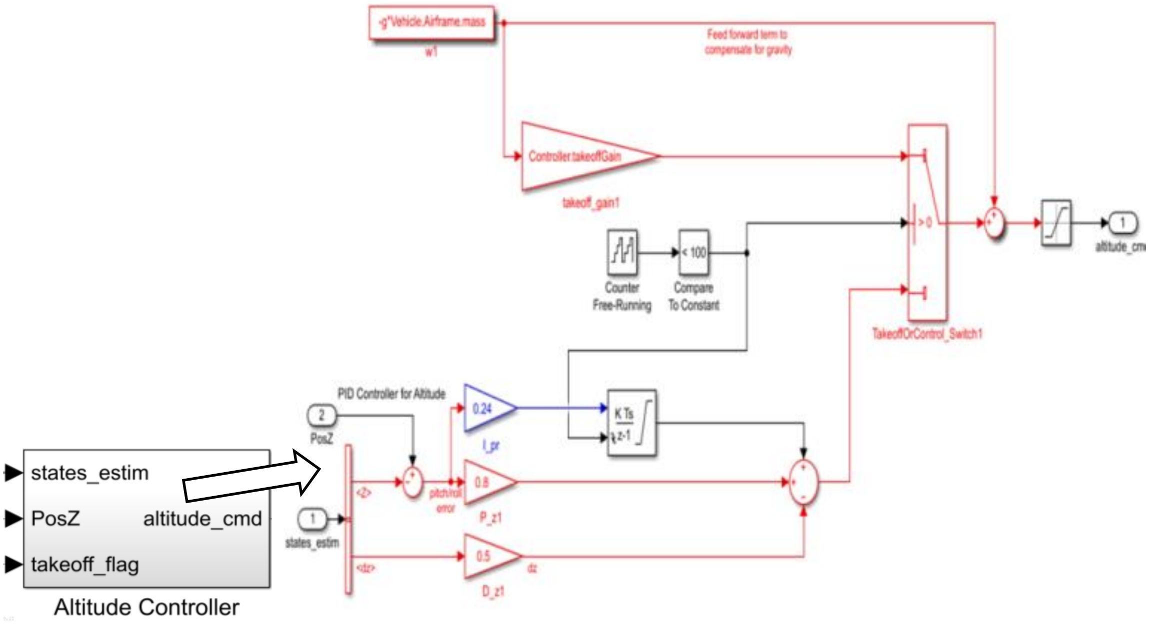

Fig. 20. Altitude blocks and structure controller

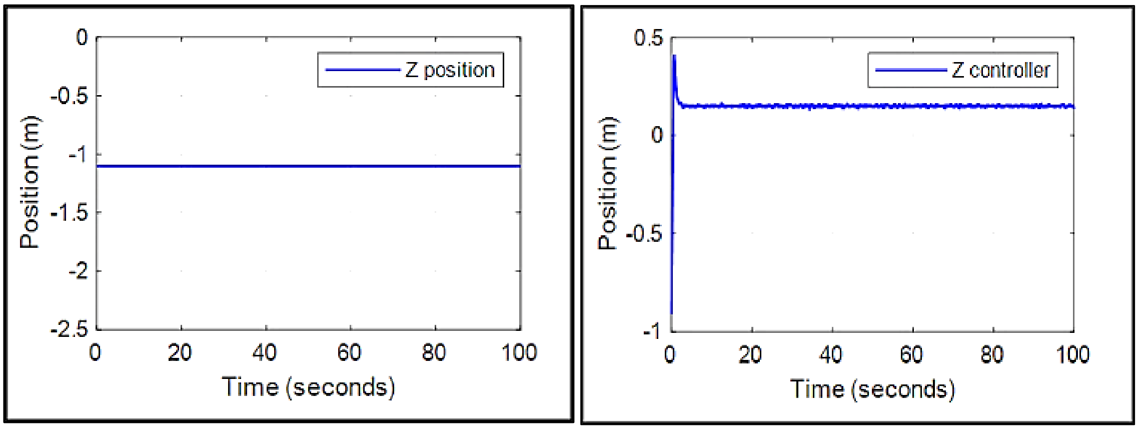

Fig. 21. Altitude controller signals without disturbance

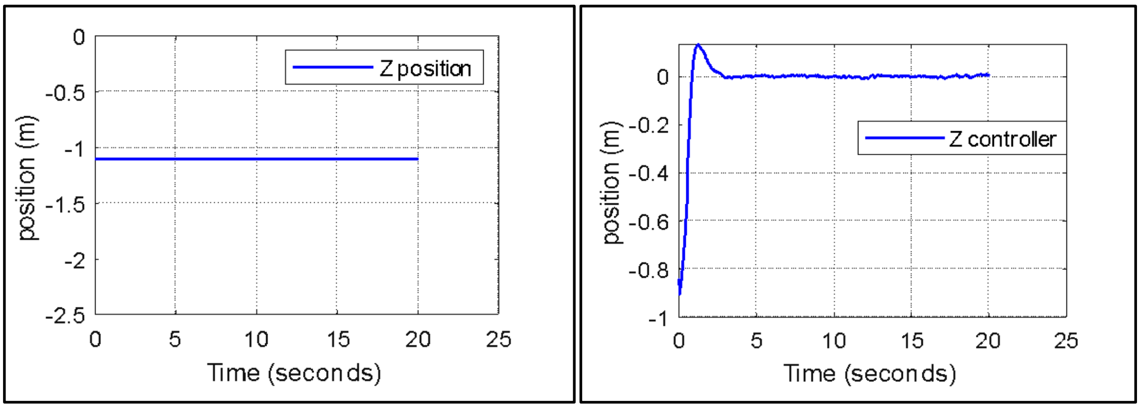

Fig. 22. Altitude controller signals with disturbance

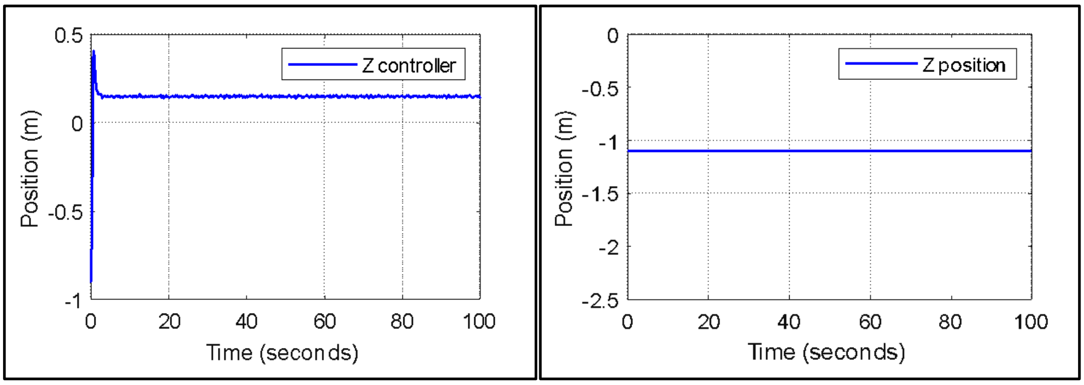

Fig. 23. The input and output of position controller signals after increasing the area of the propeller

We notice in Fig. 9, Fig. 13, Fig. 17 and Fig. 21 that the drone has completed its flight to the end of the time period specified in the simulation, which is 100 seconds without disturbances. But after increasing the weight of the drone by a certain amount, the results showed in Fig. 10, Fig. 14, Fig. 18, and Fig. 22 that the drone, after (19-20) seconds from the time of take-off, a defect occurred in the plane's system and was subjected to collision and fall, and after improving the size of the drone's propellers by a certain amount, the drone continued its flight normally as shown in Fig. 11, Fig. 15, Fig. 19 and Fig. 23.

4. Discussion

Most of the previous research focused on how to maintain the safety of the drone in various weather conditions. Some research has developed control units for drones and made them fly as stable as possible. Some research has developed and added sensors for the plane. Others are concerned with battery life. We explained this in detail at the beginning of the research paper. In this research, we have studied all types of controllers on a small Parrot mini-drone using the Matlab Simulink program. Due to bad weather conditions, we have imposed an increase in the aircraft's weight. We have extracted the results for each controller in the case of good conditions and bad conditions. We have noticed that the increase in the reference weight of the aircraft is leading to a significant defect in the flight of the aircraft, which causes its collision and crash, as illustrated in the method section in Table 2. We did the experiment using the Matlab program and got good results that were completely identical to the original results, as shown in the results section. We chose to extract the results of one trial because we got the same results in all the different cases. When the weight and lift are in opposite and equal directions, the aircraft is in equilibrium in the air; it does not rise or fall. When the lift and resistance are in opposite and equal directions, the aircraft is at a constant speed. Lift is the force generated by the area of low pressure along with the upper layer of an aircraft's wing when compared to the area of high pressure along with the lower layer of the same wing. The pressure difference between the top and bottom of the wing results in a force that pushes the wing toward the area of least pressure - that's lifting. According to the experiment that we performed in this paper, we proved that if the weight of the aircraft is increased more than the lift force, the increase in the propeller area by a certain amount has a significant impact on maintaining the stability of the aircraft and preventing it from falling and crashing.

5. Conclusions

In this work, we studied the effect of weather conditions on the operation of control devices in a small unmanned aircraft by adding some extra weight to the drone by MATLAB simulation and calculating the vertical projection area of the aircraft wings each time, as shown in the methods chapter, where we obtained the best area for the wings The plane is 0.0045 m^2, and this increase made the plane able to carry an additional weight equivalent to one-third of the original plane weight. This design is useful to make the plane more bearable in bad weather conditions. We simulated this in the MATLAB-R2021a program and obtained good results after conducting several experiments on the aircraft and comparing the results obtained before and after the design. The aircraft with the new design of the wings can withstand an increase in the weight of the aircraft up to one-third of the original weight of the aircraft. In future work, we suggest that larger aircraft with fixed propellers be used in future research, and the effects of different weather variables on UAVs, such as temperature, humidity, etc., are investigated.

Author Contribution: All authors contributed equally to the main contributor to this paper. All authors read and approved the final paper.

Funding: This research received no external funding.

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

References

[1] J. Chaoraingern, V. Tipsuwanporn, and A. Numsomran, “Mini-drone quadrotor altitude control using characteristic ratio assignment PD tuning approach,” *Lect. Notes Eng. Comput. Sci*., pp. 337–341, 2019, https://www.iaeng.org/publication/WCECS2019/.

[2] T. V. Glazkov and A. E. Golubev, “Using Simulink Support Package for Parrot Minidrones in nonlinear control education,” In *AIP Conference Proceedings*, vol. 2195, no. 1, p. 020007, 2019, https://doi.org/10.1063/1.5140107.

[3] M. R. Kaplan, A. Eraslan, A. Beke, and T. Kumbasar, "Altitude and Position Control of Parrot Mambo Minidrone with PID and Fuzzy PID Controllers," *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)*, pp. 785-789, 2019, https://doi.org/10.23919/ELECO47770.2019.8990445.

[4] W. Alqaisi, J. Ghommam, A. Alazzam, M. Saad, and V. Nerguizian, “Three-loop uncertainties compensator and sliding mode quadrotor control,” *Computers & Electrical Engineering*, vol. 81, p. 106507, 2020, https://doi.org/10.1016/j.compeleceng.2019.106507.

[5] S. Tselegkaridis and T. Sapounidis, “Simulators in educational robotics: A review,” *Education Sciences*, vol. 11, no. 1, p.11, 2021, https://doi.org/10.3390/educsci11010011.

[6] S. Waitman, H. Alwi, and C. Edwards, “Flight evaluation of simultaneous actuator/sensor fault reconstruction on a quadrotor minidrone,” *IET Control Theory & Applications*, vol. 15, no. 16, pp. 2095-2110, 2021, https://doi.org/10.1049/cth2.12180.

[7] R. Casado and A. Bermúdez, “A simulation framework for developing autonomous drone navigation systems,” *Electronics*, vol. 10, no. 1, p. 7, 2020, https://doi.org/10.3390/electronics10010007.

[8] J. Chaoraingern, V. Tipsuwanporn, and A. Numsomran, “Modified adaptive sliding mode control for trajectory tracking of mini-drone quadcopter unmanned aerial vehicle,” *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 5, pp. 145-158, 2020, https://doi.org/10.22266/ijies2020.1031.14.

[9] E. H. Kadhim and A. T. Abdulsadda, “Mini Drone Linear and Nonlinear Controller System Design and Analyzing,” *Journal of Robotics and Control (JRC)*, vol. 3, no. 2, pp. 212-218, 2022, https://doi.org/10.18196/jrc.v3i2.14180.

[10] “Drone Simulation and Control, Part 1: Setting Up the Control Problem - YouTube.” https://www.youtube.com/watch?v=hGcGPUqB67Q&list=PLy8TVa88QVfoOotVLloGWvK6PMSdjEtSb (accessed Feb. 01, 2022).

[11] C. B. Jabeur and H. Seddik, “Optimized neural networks-PID controller with wind rejection strategy for a Quad-Rotor,” *Journal of Robotics and Control (JRC)*, vol. 3, no. 1, pp. 62-72, 2022, https://doi.org/10.18196/jrc.v3i1.11660.

[12] A. E. M. Redha, R. B. Abduljabbar, and M. S. Naghmash, Drone Altitude Control Using Proportional Integral Derivative Technique and Recycled Carbon Fiber Structure, In *Proceedings of International Conference on Emerging Technologies and Intelligent Systems: ICETIS 2021 (Volume 1)*, vol. 299, pp. 55-67, 2022, https://doi.org/10.1007/978-3-030-82616-1_6.

[13] Q. H. Doan, S. Mai, Q. T. Do, and D. Thai, "A cluster-based data splitting method for small sample and class imbalance problems in impact damage classification," *Applied Soft Computing*, vol. 120, p. 108628, 2022, https://doi.org/10.1016/j.asoc.2022.108628.

[14] P. Ceppi, "Model-based Design of a Line-tracking Algorithm for a Low-cost Mini Drone through Vision-based Control," *Diss. University of Illinois at Chicago*, 2020, https://webthesis.biblio.polito.it/16018/.

[15] J. Yeom, J. Jung, A. Chang, M. Maeda, and J. Landivar, "Cotton growth modeling using unmanned aerial vehicle vegetation indices," *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pp. 5050-5052, 2017, https://doi.org/10.1109/IGARSS.2017.8128138.

[16] I. Iswanto, A. Ma'arif, O. Wahyunggoro, and A. Imam, "Artificial Potential Field Algorithm Implementation for Quadrotor Path Planning," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 8, 2019, https://doi.org/10.14569/IJACSA.2019.0100876.

[17] M. Kok, J. D. Hol and T. B. Schon, "Using Inertial Sensors for Position and Orientation Estimation," Foundations and Trends in Signal Processing, vol. 11, no. 1-2, pp 1-153, 2017, http://dx.doi.org/10.1561/2000000094.

[18] A. Shamshirgaran, H. Javidi and D. Simon, "Evolutionary Algorithms for Multi-Objective optimization of Drone Controller Parameters," *2021 IEEE Conference on Control Technology and Applications (CCTA)*, 2021, pp. 1049-1055, 2021, https://doi.org/10.1109/CCTA48906.2021.9658828.

[19] S. Karaman and F. Riether, "Getting Started with MIT's Rolling Spider MATLAB Toolbox," *An MIT Take-Home Lab for*, vol. 16, 2016, pp. 1–31, 2015, https://fast.scripts.mit.edu/dronecontrol/wp-content/uploads/2016/05/GettingStarted.pdf.

[20] M. Idrissi, M. Salami, and F. Annaz, "A review of quadrotor unmanned aerial vehicles: applications, architectural design and control algorithms," *Journal of Intelligent & Robotic Systems*, vol. 104, no. 2, p. 22, 2022, https://doi.org/10.1007/s10846-021-01527-7.

[21] M. Orsag, C. Korpela, P. Oh, and S. Bogdan, *Aerial Manipulation*. Springer Cham, 2018, https://doi.org/10.1007/978-3-319-61022-1.

[22] M. R. Kaplan, A. Eraslan, A. Beke, and T. Kumbasar, "Altitude and Position Control of Parrot Mambo Minidrone with PID and Fuzzy PID Controllers," *2019 11th International Conference on Electrical and Electronics Engineering (ELECO)*, pp. 785-789, 2019, https://doi.org/10.23919/ELECO47770.2019.8990445.

[23] M. Okasha, J. Kralev, and M. Islam, "Design and Experimental Comparison of PID, LQR and MPC Stabilizing Controllers for Parrot Mambo Mini-Drone," *Aerospace*, vol. 9, no. 6, p. 298, 2022, https://doi.org/10.3390/aerospace9060298.

[24] G. Delansnay and A. V. Wouwer, "Implementation and Tests of an INDI Control Strategy applied to the Parrot Mambo Minidrone," *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 745-752, 2022, https://doi.org/10.1109/ICUAS54217.2022.9836222.

[25] M. Dragomir, V. -M. Maer, and L. Buşoniu, "The Co4AIR Marathon – A Matlab Simulated Drone Racing Competition," *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1219-1226, 2022, https://doi.org/10.1109/ICUAS54217.2022.9836233.

[26] S. Moon, J. J. Bird, S. Borenstein, and E. W. Frew, "A Gazebo/ROS-based Communication-Realistic Simulator for Networked sUAS," *2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 1819-1827, 2020, https://doi.org/10.1109/ICUAS48674.2020.9213892.

[27] A. Dutta, J. Wang, F. Kopsaftopoulos, and F. Gandhi, "Rotor Fault Detection and Identification on Multicopter based on Statistical Data-driven Methods: Experimental Assessment via Flight Tests," 2022, https://doi.org/10.4050/F-0078-2022-17556.

[28] A. Noordin, M. A. M. Basri, and Z. Mohamed, "Position and Attitude Tracking of MAV Quadrotor Using SMC-Based Adaptive PID Controller," *Drones*, vol. 6, no. 9, p. 263, 2022, https://doi.org/10.3390/drones6090263.

[29] J. A. Cárdenas, U. E. Carrero, E. C. Camacho, and J. M. Calderón, "Optimal PID ø axis Control for UAV Quadrotor based on Multi-Objective PSO," *IFAC-PapersOnLine*, vol. 55, no. 14, pp. 101-106, 2022, https://doi.org/10.1016/j.ifacol.2022.07.590.

[30] L. Vile, H. Alwi, and C. Edwards, "Implementation of an Interpolated Model Reference Control Scheme," *2022 16th International Workshop on Variable Structure Systems (VSS)*, pp. 35-40, 2022, https://doi.org/10.1109/VSS57184.2022.9902017.