Engineering and Technology Journal e-ISSN: 2456-3358

Volume 11 Issue 05 May-2026, Page No.-9758-9772

DOI: 10.47191/etj/v11i05.12, I.F. – 8.595

Intelligent Drone Dynamics Simulation

Nickolay Zosimovych

Professor, Aerospace Faculty, State University "Kyiv Aviation Institute", Liubomyra Guzara Ave., 1, KAI, kor 7, 03058, Kyiv, Ukraine

ABSTRACT: During a mission, if a drone carries out machine learning procedures to all the data collected by its numerous sensors, it might make moment-to-moment assessments around how to respond to its position. Proposed paper will use the Newtonian method to build the drone dynamics model. During problem study we require a treating idea that estimates atmospheric density amounts grounded on weather conditions and other physical numbers. So, the model can execute stable flight under the impact of environmental reasons, and can clearly prove the variation in the transfer of the drone under the impact of environmental influences. As a basic was studied a proportional integral derivative (PID) controller depend on of a proportional unit, integral unit, and derivative unit. Through the simulation procedure, when a drone flight command is read, it is treated by the order interactive module first and is sent to the physical model and control model, which will update the drone's motion condition based on the command information and environment condition.

In proposed research, the newly developed neural network must be trained first, as how an individual child being skilled to execute activities.

The MATLAB/Simulink environment was employed to perform the simulations. In this part, solutions are performed through altitude and attitude balance using step input and multi-level tracking, and position tracking.

This research has positively submitted the design and implementation of the auto-tuned PID controller using neural network in evaluation to manual tuned PID controller.

KEYWORDS: Unmanned aerial vehicle (UAV), Unmanned aerial system (UAS), modeling, drone, intelligent, simulation

1. INTRODUCTION

UAVs are disposed of convinced climate conditions when flying outdoor [1], so it is vital to study weather conditions when establishing a simulation constraint for UAS. We delivered out the force study for the flying drone. We designed a physical model built on Newtonian mechanics and Euler angles. Thus, a PID-based control model is created according to the physical model. We perform intelligent control on the drone in a physical limitation to general the complete simulation.

In the proposed paper, the weather-influence features are announced into the physical model of a hexacopter over an environmental calculation model, and a technique to evaluate the environmental limitations grounded on the weather is presented, therefore entirely adding the weather prompting issues.

2. LITERATURE ANALYSIS

Corresponding to the rotor position, the shape of quadrotors is split into two main categories: a cross shape and X-shape. The cross-shaped drone usually is cross-symmetrical, and its basic all rotors are allocated in four directions: front, rear, left, and right. The X-shape drone is X-shaped symmetrical, with the main part in the center of the drone, and the six rotors allocated on the four corners.

During the last decades, there have been numerous studies about the dynamic model of drones (as hexacopter, as octocopter too). Some of these publications [2–7] use the Newton–Euler method, which analyses dynamics and motion equations based on Newton's second and Euler laws, to establish a model. Other publications [7–9] use the Euler–Lagrange method, which determines a dynamics model using Lagrangian mechanics to explicit the drone as a physical system with six degrees of freedom.

Investigation has been acted in relation to the imitation of drones under wind conditions. Khan and Nahon proposed a dynamic model of the rotor moving in wind [10]. The model gets the differential thrust model though Momentum Theory (MT) and Blade Element Theory (BET) to compute generation velocity in real time. Authors then separates the thrust of the blade air foil piece and combines to find the total thrust.

Fernando and his co-authors [11] use the Newton–Euler method to model drone dynamics. Tabassum and his group [12], and Massé and others [13] proposed a look-up table method for calculating the total force of the rotor and wind relations based on Khan's model. Refs. [2, 9, 13] used a method based on a PID controller. The work of [3] adopted an adaptive inverse control method. Ref. [6] used the ADRC (Active Disturbance Rejection Control) algorithm to build models. But for another drone type, like hexacopter and octocopter very seldom you may find real researching.

3. METHODOLOGY

AI has repeatedly depended on big data sets to build effective algorithms. IoT might use active data to AI to build further complicated algorithms and use "sense" to real-time data. AI could be used to convert IoT data into valuable data for improved decision-making processes.

Importance, AIoT is evenly essential for both tools. During a mission, if a drone carries out machine learning techniques to all the data collected by its various sensors, it might make moment-to-moment assessments around how to respond to its position. For example, in respond to wildfire, drones could be forwarded to give up fire retardant.

Focused on the locates and datasets surrounded by their IoT-enabled sensors, they might determine individually on the greatest place to drop the material. These terms would include the wind direction and speed; existing temperature; and the percentage of fire inserted [4].

Experimental simulation hexacopter model will be executed with Unity [1]. As Unity's coordinate system model will be employed, which has a left-handed coordinate system with the x-axis to the right, the y-axis up, and the z-axis onward.

Thus, defined by the above lefthanded coordinate system, the Euler angles are distinct as follows [3]:

$$\theta = (\theta, \psi, \varphi)^T, \tag{1}$$

where θ represents to the pitch angle of the drone (down direction from the drone's front perspective is positive), ψ denotes to the yaw angle (clockwise direction from above is positive), and φ denotes to the roll angle (down direction from the left-side viewpoint is positive).

Computed by triangular alteration, the relation between the angular velocity $\omega = (\omega_x, \omega_y, \omega_z)^T$ and Euler angle's rate of change $\dot{\theta}$ is [3, 9]:

$$\begin{pmatrix}\omega_x\\ \omega_y\\ \omega_z\end{pmatrix} = \begin{pmatrix}1 & 0 & -sin\theta\\ 0 & cos\psi & sin\psi cos\theta\\ 0 & -sin\varphi & cos\varphi cos\theta\end{pmatrix}\begin{pmatrix}\dot{\theta}\\ \dot{\psi}\\ \dot{\varphi}\end{pmatrix}. \tag{2}$$

It can be calculated throughout triangular alteration, specified by the Euler angle, the vector $(x^/, y^/, z^/)^T$ in the body coordinate system can be transformed to the vector $(x, y, z)^T$ in the geodesic coordinate system [3]:

$$\begin{pmatrix}x\\ y\\ z\end{pmatrix} = \begin{pmatrix}sin\psi sin\varphi sin\theta + cos\psi cos\varphi & -sin\psi cos\psi + sin\psi sin\theta cos\varphi & sin\psi cos\theta\\ sin\psi cos\theta & cos\varphi cos\theta & -sin\theta\\ sin\varphi sin\theta cos\varphi - sin\psi cos\varphi & sin\varphi sin\psi + sin\theta cos\varphi cos\psi & cos\psi cos\theta\end{pmatrix}\begin{pmatrix}x^/\\ y^/\\ z^/\end{pmatrix}. \tag{3}$$

3.1. Physical Modeling

Proposed paper will use the Newtonian method to build the drone dynamics model. According to that, the hexacopter drone force analysis in the running state is shown in Fig. 1 [21].

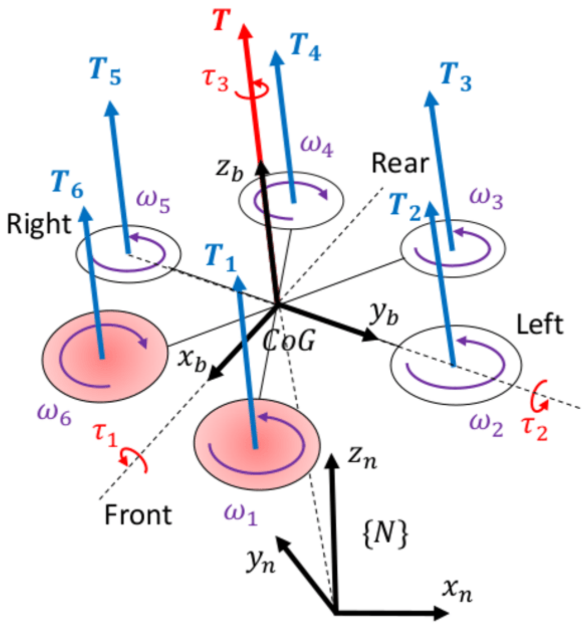

Fig. 1. Drone diagram of the force [21]

Corresponding to the aerodynamic model, when a single rotor drives, it keeps an upward lift relation to the square of the rotor's angular velocity in all types of drones (as hexacopter, as octocopter). The total dynamics of a drone are modeled as a set of nonlinear differential equations, which is split into the kinematics (1st two equations) and the system dynamics (the last one) [14, 15]:

$$[\dot{s}^S, \dot{y}^S, \dot{z}^S]^T = R^{B\to S}[\dot{x}^B, \dot{y}^B, \dot{z}^B]^T, \quad (4)$$

$$\frac{d}{dt}\begin{bmatrix}\phi\\ \Theta\\ \psi\end{bmatrix} = \begin{bmatrix}1 & sin\phi\tan\Theta & cos\phi\tan\Theta\\ 0 & cos\phi & -sin\phi\\ 0 & sin\phi\cos\Theta & cos\phi\cos\Theta\end{bmatrix}, \quad (5)$$

$$\dot{s}^D(t) = f_c(x^D(t), u(t)), \quad (6)$$

where

$$x = [x^K, x^D]^T \in R^{n_x},$$
$$x^K = [x^S, y^S, z^S, \phi, \Theta, \psi]^T$$
$$x^D = \left[u, v, w, p, q, r, a_{1s}, b_{1s}, r_{fb}\right]^T,$$
$$u = \left[u_{a1s}, u_{b1s}, u_{\Theta M}, u_{r_{ref}}\right]^T \in R^{n_x}.$$

At this point S and B indicates 3-D and body coordinates. $\dot{x}^B, \dot{y}^B$, and $\dot{z}^B$ u, v and w correspondingly, will be treated for notational ease) designate velocity regarding the body-coordinate framework. ϕ, Θ, and ψ mean roll, pitch, and yaw, and p, q and r are their rates, correspondingly [14].

Grounded on the beyond mechanical model, the dynamics model of the drone can be additionally calculated. Therefore, the following force and moment of the drone can be computed. Furthermore, the acceleration and angular acceleration of the drone can be estimated too.

3.2. Wind Influence

The resistance produced by air to the drone is relational to the speed of the air comparative to the UAV [1].

The speed of the drone comparative to the air can be given by removing the speed of the wind relative to the ground after the speed of the UAV relative to the ground.

Furthermore, all rotors are additionally disturbed by wind. Ref. [11] built a look-up table reasoning matching to the distinction cognitive model of the extra force and moment that [12] applies on the wind rotor. The technique can be worked to estimate the lateral thrust of a hexacopter, additional lift, and further moment of rotors with different speeds beneath diverse winds (Table 1) [14].

Table 1: Rotor’s Thrust in open Wind Conditions

Rotating Speed, rpm	Wind Speed, m/s				
	0-1	1-2	2-3	3-4	4-5
	Drone’s Thrust, N				
2000	0,011	0,021	0,032	0,043	0,055
2500	0,014	0,026	0,038	0,052	0,065
3000	0,016	0,030	0,045	0,060	0,075
3500	0,018	0,035	0,051	0,068	0,085
4000	0,020	0,039	0,057	0,077	0,096

3.3. Model Parameters Determination and Fuzzy Logic Technique

Lots of model factors are included in the drone’s physical model. As for these factors, we used the initial coefficients of the drone Thunder S550 Hexacopter Frame Kit W/ PCB required by Ref. [13] (Fig. 3).

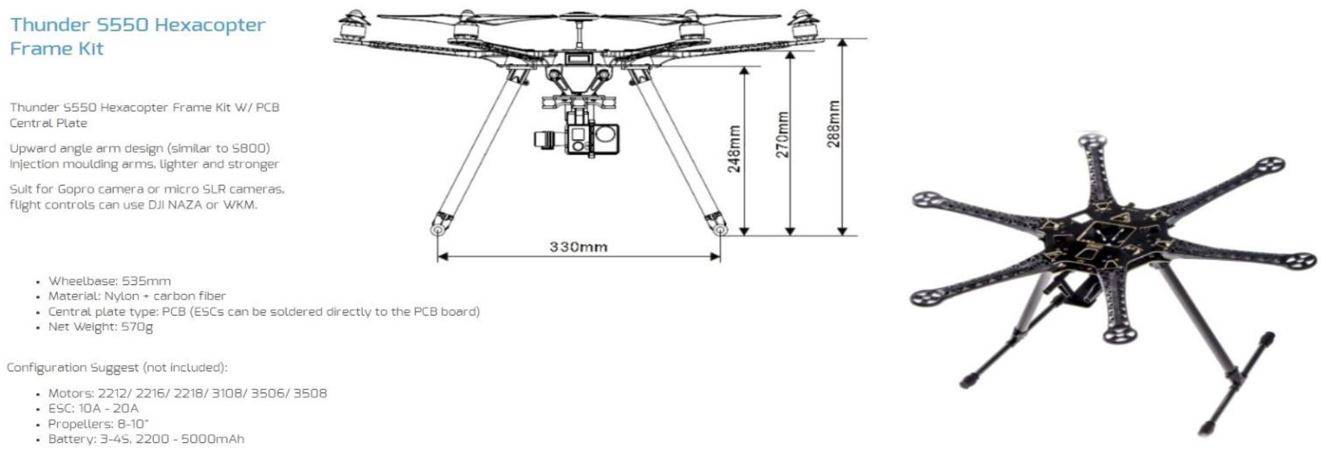

Fig. 3. Thunder S550 Hexacopter Frame Kit W/ PCB

Decompose the above forces and moments into the world coordinate system. Merge individually in three dimensions and convert the force outlined in the drone coordinate system using Eq. (3), we can get the quantity of the merged force in three dimensions in the world coordinate system [1]:

$$F_x = \frac{\rho}{\rho_0} C_{drag}(v_{wind,x} - v_x) + \frac{\rho}{\rho_0} K_T \omega_i^2 (-sin\varphi cos\psi + sin\psi sin\theta cos\varphi), \quad (7)$$

$$F_y = \frac{\rho}{\rho_0} C_{drag}(v_{wind,y} - v_y) + \frac{\rho}{\rho_0} K_T \omega_i^2 cos\varphi cos\theta - mg, \quad (8)$$

$$F_z = \frac{\rho}{\rho_0} C_{drag}(v_{wind,z} - v_z) + \frac{\rho}{\rho_0} K_T \omega_i^2 (sin\varphi sin\psi + sin\theta cos\varphi cos\psi). \quad (9)$$

During the achievement of the steps in Ref. [4], the subsequent design was done. The choice to use Python coding appeared due to a condition for the AI cluster. This coding language turns out to be very useful for this type of research. The next chart is given for the following approach [6]:

1.The initial effort to resolve the problem was to investigate the past and current activities of a drone flight control. This research was made by literature analysis.

2.Investigation was made as soon as sharing the choice for elevator control to save cruise altitude. An effort was done to get a fuzzy logic platform or applet written in Python.

3.The technique for the problem was to realize basic Python programming procedures for doing virtual drone flight files, converting these files if necessary, and operating or adapting a Fuzzy Logic application.

The methodologies executed in most of the review covered Expert Systems, Neural Networks, Fuzzy Logic, and Hybrid models of these and PID control systems. The AI methods chosen for use were Fuzzy Logic, Expert System, and next a series of Neural / Fuzzy for adaptive competence.

An experiment model was commanded to state a restricted treatment of an accepted Fuzzy control system. The needed altitude hang up was activated with an additional input for acceleration control. The acceleration control happened for advancing work covering degrees of power contained during revised mission sets (take-off, level flight, disturbed flight, and approach).

All devices can operate crisp or fixed data such as either “true” or “false”. To modify tools to carry out hazy linguistic input such as “somehow satisfied”, the quick input and output ought to be converted to linguistic variables with fuzzy models, for example: “High”, “Medium” and “Low”, or “Fast”, “Medium” and “Slow”. Varied with typical control performances, like a static PID, a fuzzy controller is able of adjusting itself to the dynamics of the plant. If the transfer function of the plant transforms over time, it is assumed to get procedure with a PID controller, even though fuzzy logic results remain constant.

The combination of this control procedure in Veronte Autopilot, makes feasible control of every dynamic system with the pros of fuzzy laws. Used to drone control, the Fuzzy Logic Controller (FLC) lets us to get attuned controllers robust to mission alterations (motor failures, control actuator block, cargo release and others) [6].

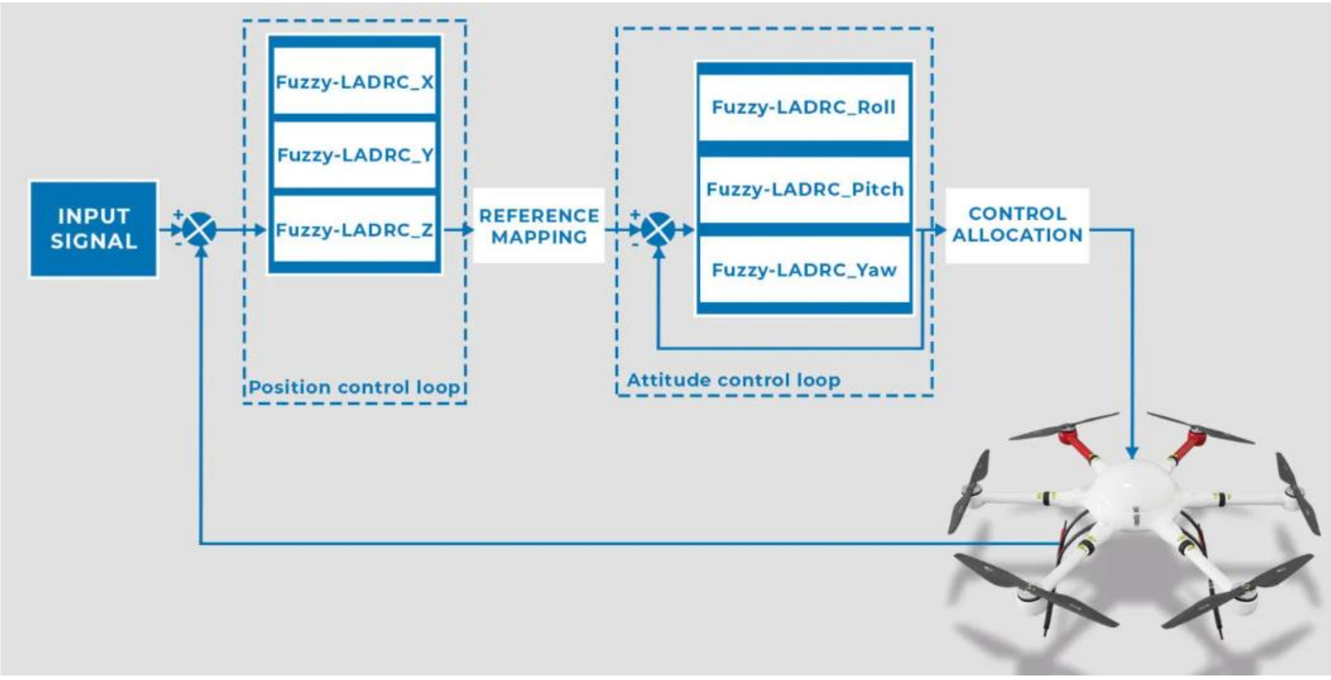

Fig. 4. Fuzzy controller principle in Veronte Autopilot [6]

The model for the fuzzy controller is drawn in Fig. 4. The first system only includes of altitude as an input and elevator as the output. It proved to be a simple and stable control strategy. Acceleration was involved to give outgoing levels of control for a throttle efficiency to reply to levels of dynamic regulation in terms of the flight envelope.

4. RESULTS AND DISCUSSION

The control model revealed by this component for the drone is usually made on the model given by [1]. The hypothesis of the control model will be examined and clarified in this part.

4.1. Proportional Integral Derivative Controller

As an introductory was examined a PID (Proportional Integral Derivative) controller depend on of a proportional unit, integral unit, and derivative unit.

The input of the PID controller is the error rate of the regulated capacity e(t), and the output is u(t). They are all services of time t. A standard PID controller can be specified by the following representation [1]:

$$u(t) = K_p e(t) + K_i C_{drag} \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t). \quad (10)$$

where K_p, K_i, and K_d, all non-negative, indicate the factors for the proportional, integral, and derivative conditions, respectively.

The proportional element is the primary part that responds to control deviations. When there is an alteration, the proportional part can change directly to control the exact quantity, so that the controlled magnitude is consistent.

The integral part is to prevent the collected error when the control gets stability. When there is a deviation, if there is a cumulative error in the value of the controlled amount, only using the proportional influence will make the controlled amount reach a fixed within a specific developed error. After adding the integral part, the controlled quantity can reach the assumed amount.

The differential part has the point of avoiding overshoot. Sometimes no integral part or differential section is mandatory, and the controller can complete the control on its own. A controller involving only a proportional group is labelled a P controller, and similarly there are PI controls and PD controls.

For discrete time controllers, constant integration of the integral part and direct differentiation of the derivative part are unnecessary. Thus, this type of controller generally uses summation instead of integral and difference as an alternative of differentiation [1]:

$$u(t) = K_p e(t) + K_i \sum_{\tau=0}^{t} e(\tau) \ + K_d(e(t) - e(t-1)). \quad (11)$$

This expression can be got straight in the computation. Deduct $u(t-1)$ from $u(t)$ to get next term [1, 15]:

$$u(t) - u(t-1) = K_p\big(e(t) - e(t-1)\big) + K_i\big(e(t) - 2e(t-1) + e(t-2)\big). \quad (12)$$

Thus, $u(t)$ come to be an amount only related to $u(t-1)$, $e(t)$, $e(t-1)$, and $e(t-2)$. The content that demands to be stored while computing is seriously cut down.

In the next, we concern the character P with varied subscripts to indicate to distinctive PID controller functions that modification through time, which can be written as $u(t) = Pe(t)$.

From fly dynamics we know that a drone can only functioning as normal when the tilt angle is particularly small. If the Euler angle is large enough, the motion condition of the drone will be very unstable. If the drone is turned upside down for a long time, it is no longer possible to control the rotor to climb upward since the lift’s direction is the same as the gravities. So, the drone will surely enter a state of falling with positive feedback.

Thus, in the arising derivation of the control model, we believe that during the mission of the drone, the pitch and roll angles are both small angels, which fits:

$$sin\phi \approx \phi, cos\phi \approx 1, sin\theta \approx \theta, \cos\theta \approx 1. \quad (13)$$

In the control of the drone, this part should also try to avoid a restriction in which the small angle assumption is damaged, or the drone is out of control. This will be done by mixing saturation to the control constraints.

For the PID control model, such a small change ratio rate creates it sufficient to concern the air resistance part of a constant with disturbance, which is settled by the integral unit in the PID control [1].

The trigonometric function rate of the attitude angle can be known as f, q itself the acceleration value of the drone in the horizontal direction can be seen with the attitude. The angle value itself is completely linked. Due to the proportional characteristics of PID, the attitude angle can be clearly equal to the horizontal force to add to the control. Hence, we need keep on from the limitations and check what kind of commands the component needs to hold. The components that this part needs to support are shown in Table 2.

Table 2: Commands of the drone

No	Command	Meaning	Return Value
1	take-off	let the drone take off	ok/error
2	land	let the drone land	ok/error
3	emergency	all motors will stop immediately	ok/error
4	up x	fly upward x m, 0.5 – 5.0	ok/error
5	down x	fly downward x m, 0.5 – 5.0	ok/error
6	forward x	fly forward x m, 0.5 – 5.0	ok/error
7	back x	fly backward x m, 0.5 – 5.0	ok/error
8	“cw” x	rotate clockwise x°, 1 - 360	ok/error
9	“ccw” x	rotate anticlockwise x°, 1 - 360	ok/error
10	go to x y z	fly to coordinate x, y, z: m, 0.5 – 5.0	ok/error
11	speed x	set the current speed to x m/s, 0.2 – 10.0	ok/error

4.1.1. Control Distribution and PID Parameters

Now, we need to transfer the four mechanical parameters to the same quantity's rotors of the drone.

Fortunately, the four control variables are the outcome of the PID controller. If the three coefficients of K_p, K_i, and K_d are multiplied by a coefficient value, and then the output result of the PID controller is divided by this control value, the outcome is the same. In other words, the proportional coefficient of the PID controller itself can be used to reduce the difference in factors in the control distributor.

So, if the proportional factors between the square of the rotor speed and the mechanical quantities are integrated into the three control quantities of the controller, the four control quantities approved into the control supplier can be regarded as virtual values f_{tr}, f_{xr}, f_{yr}, and f_{zr}.

In this way, the power factor of the identical rotor on different control variables can be ignored.

Moreover, our control model only wishes to guarantee that the relative parts of different rotors to the same control capability stay unchanged. By exploring the influence of each rotor to the different control variables, we get the control distribution model from the speed to the virtual values [1]:

$$\omega_1^2 = f_{tr} - \tau_{xr} + \tau_{zr} - \tau_{yr}, \tag{14}$$

$$\omega_2^2 = f_{tr} + \tau_{xr} + \tau_{zr} + \tau_{yr}, \tag{15}$$

$$\omega_3^2 = f_{tr} + \tau_{xr} - \tau_{zr} - \tau_{yr}, \tag{16}$$

$$\omega_4^2 = f_{tr} - \tau_{xr} - \tau_{zr} + \tau_{yr}. \tag{17}$$

In a real drone, each rotor in general is operated by an electric motor. The power of the motor and the speed under load obviously have an upper limit. Hence, the rotor speed is saturated between 0 and the maximum speed (that is, w_{max} in Table 2). If we let a certain control quantity to expand or contract, it will produce all the control values to be held at the maximum value or drop below 0, initiating the control of other control quantities to be ineffective. Therefore, the four control variables are also drowned before input.

When reaching the model of the PID control, it is required to accept the control factors in the PID model. We apply Python to create a physical model and a control model [15, 17], to more simply study the quality of PID parameters.

The PID control factors gotten in Ref. [21] are shown in Table 3.

Table 3: PID control parameters

Controller	K_p	K_i	K_d
P_{hv}	0.24	0	0
P_{hf}	3.81	0.049	0.00423
P_{vv}	1.33	0	0
P_{vf}	94	1	0.36
$P_{h\omega}$	1.52	0	0
$P_{h\tau}$	0.66	0.02	0.012
$P_{v\omega}$	1.31	0	0
$P_{v\tau}$	14	3	0.151

Relating to the prior model, the environmental effects that mostly impact drone flight involve wind and atmospheric density.

But, compared to further weather sets such as temperature, air pressure, atmospheric density and other physical attribute.

To adjust the usability of the system, so that investigator can simply add weather influence factors and present an improving atmospheric density model, which can be used to estimate the atmospheric density over other environmental information.

Complete the appropriate conditions constraints (Temperature $T = 20^0/293.15\,K$; Pressure $p = 101.32\,kPa$; Density $\rho = 1.18\,kg/m^3$), the amount of R_{spec} can be estimated to complete the calculation of atmospheric density.

4.2. Modeling and Simulation

4.2.1. Architecture Configuration

The drone simulation structure is embedded in a web page as a WebGL page environment (short for Web Graphics Library). This is a JavaScript API for giving interactive 2D and 3D graphics. We use that unit to build the model, and set it as a WebGL module, which is integrated with other parts of the system to run on the web [1].

The initial model of MVC (Model View Controller) is a normally used software architecture model. Normal MVC software architecture is displayed in Ref. [1].

The software system consuming this architecture holds of the next three key components [1]:

1. *Model*: In a conventional software system, the model part comprises algorithms for design business logic and database management for storing data, which is the core logic module of the system. In this module, the model part is equivalent next several models, such as a weather model, control model, physical model, and continually updated runtime statistics.

2. *View*: In basic software, the view part is the graphic interface, which includes all the data info displayed to the user. For that unit, the view port is equally to the simulation interface fulfilled by Unity3D and the text content on the user interface.

3. *Controller*: The controller module is in charge for posting and managing the requests. For this module, the controller joins the part that transfers and collects commands from the main system and the part of the web page outside the module where the user inputs the command data on the web page.

4.2.2. Environmental Module

Throughout problem analysis we need a handling idea that estimates atmospheric density amounts founded on weather environment and other physical information. When all environment data amounts are ignored, the workflow of the environmental model is displayed in Fig. 7 [1, 19].

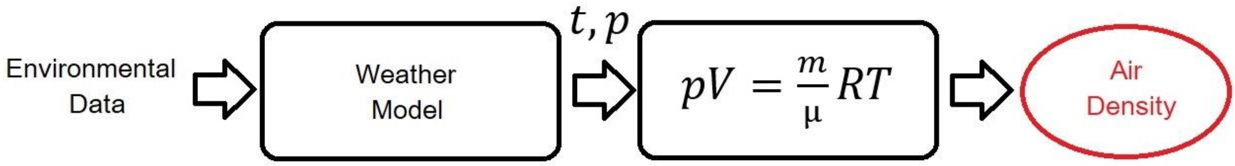

Fig. 7. Workflow of the environmental model [19]

4.2.3. Test and Validation of the Model

According to the physical and control models accepted beyond, was test and confirm the simulation model not concerning a weather state and under altered weather conditions. When the drone is working the instruction "up 20" (Table 2), the altitude modify appeared in Fig. 8. Diagram indicates that the drone can rise steadily and efficiently [19].

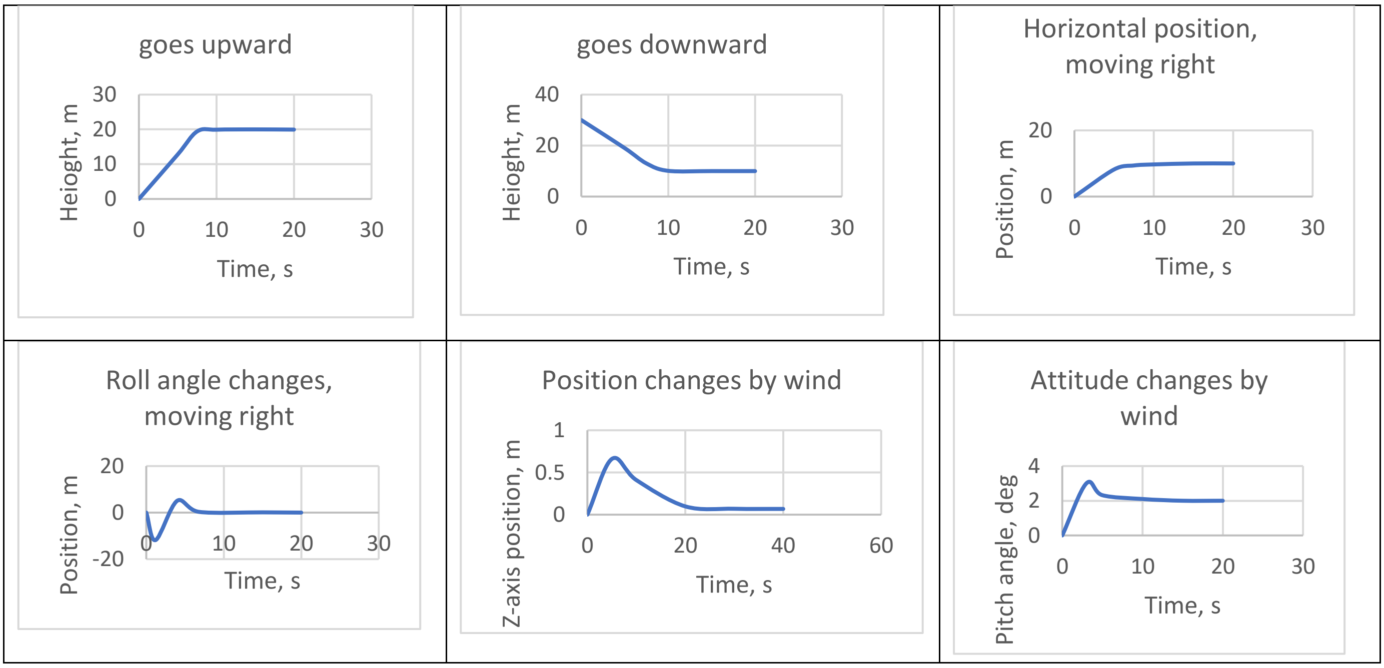

Fig. 8. The simulation model without a weather circumstance [19]

Also, the instruction "down 20" is performed at a height of 30, and the height change is as shown in Fig. 8. However there is a small shock, the land of the drone is still completely stable.

Horizontal motion has similar control effects. Assume that the drone is flying to the right, operating "right 10". The drone's X-axis attitude changes, and the attitude angle changes with time, as indicated in Fig. 8.

Throughout the accurate mission, the drone first tilts to the right to accelerate, and then turns left to slow down and stop. For a mission, the forward, left, and rear paths are all similar to the right, ergo there is no restrict to go into detail. It can be noted that the drone can move fine under control without weather requirement and can scope the target location.

Now let us test model control outcome in weather creating effects if the drone will steady in a showed attitude in the case of a sudden wind. Suppose when the drone is inactive in a state of wind missing, wind from the rear with a speed of m/s balance. Moved by the gust, the drone's position change as pointed in Fig. 8.

So, we test that if the drone will develop steady in a indicated position in the case of a unexpected wind. Consider that when the drone is stationary in a term of no wind, breeze from the rear with a speed of 8 m/s take place. Induced by the wind, the drone's position shift as demonstrated in Fig. 8.

It can be seen that the position of the drone coming back to its previous location under a small disruption. In terms of attitude, the pitch angle of the drone finally calms at a positive value following the presence and recall to contest the thrust produced by the wind. The greatest attitude angle denotes in Fig. 8 must exist, otherwise the drone cannot shrink the cumulative error in the attitude.

As a assumption, the model can execute stable flight under the influence of environmental influences, and can clearly prove the variation in the transfer of the drone under the effect of environmental influences.

4.3. Neural Network Auto-Tuner PID Controller

A feedforward neural network is built up and described in this part. An artificial neural network (ANN) is a simulated of works of anthropoidal brain cells which contains axons and dendrites that behave data transferring and explaining, using responses achieved from the human's five senses. In proposed study, the newly grown neural network must be trained first, as how an individual child being skilled to perform activities. The datasets generated will be used to train the network. Each dataset created has seven inputs $(ref, u, e, T_r, T_s, T_o, O_p)$ and 3 aimed outputs (K_p, K_i, K_d) for the training [22].

Fig. 9 temporarily visualizes the action of network creation starting from the use of datasets for network training.

Fig. 10 show up the network configuration, where x denotes the 7 inputs into the network and $\Delta K_{p,i,d}$ denotes the 3 calculated outputs of the network i, j and k denote the neurons of input, hidden and output layer respectively. The weight matrices of input layer to hidden layer and from hidden layer to output layer are represented by $w_{i,j}$ and $v_{j,k}$ respectively. Output from each layer is denoted by O_i, O_j and O_k. Next series of equations below are the mathematical model of feedforward ANN [23].

Input to the hidden layer:

$$O_i = \sum_{i=1}^{16} w_{i,j} x_i. \quad (14)$$

Output of hidden layer:

$$O_j = \frac{1}{1+e^{-O_i}}. \quad (15)$$

Input to the output layer:

$$r = \sum_{j=1}^{n} v_{j,k} x O_j. \quad (16)$$

Output of output layer:

$$\Delta K_G = \frac{1}{1+e^{-r}}. \quad (17)$$

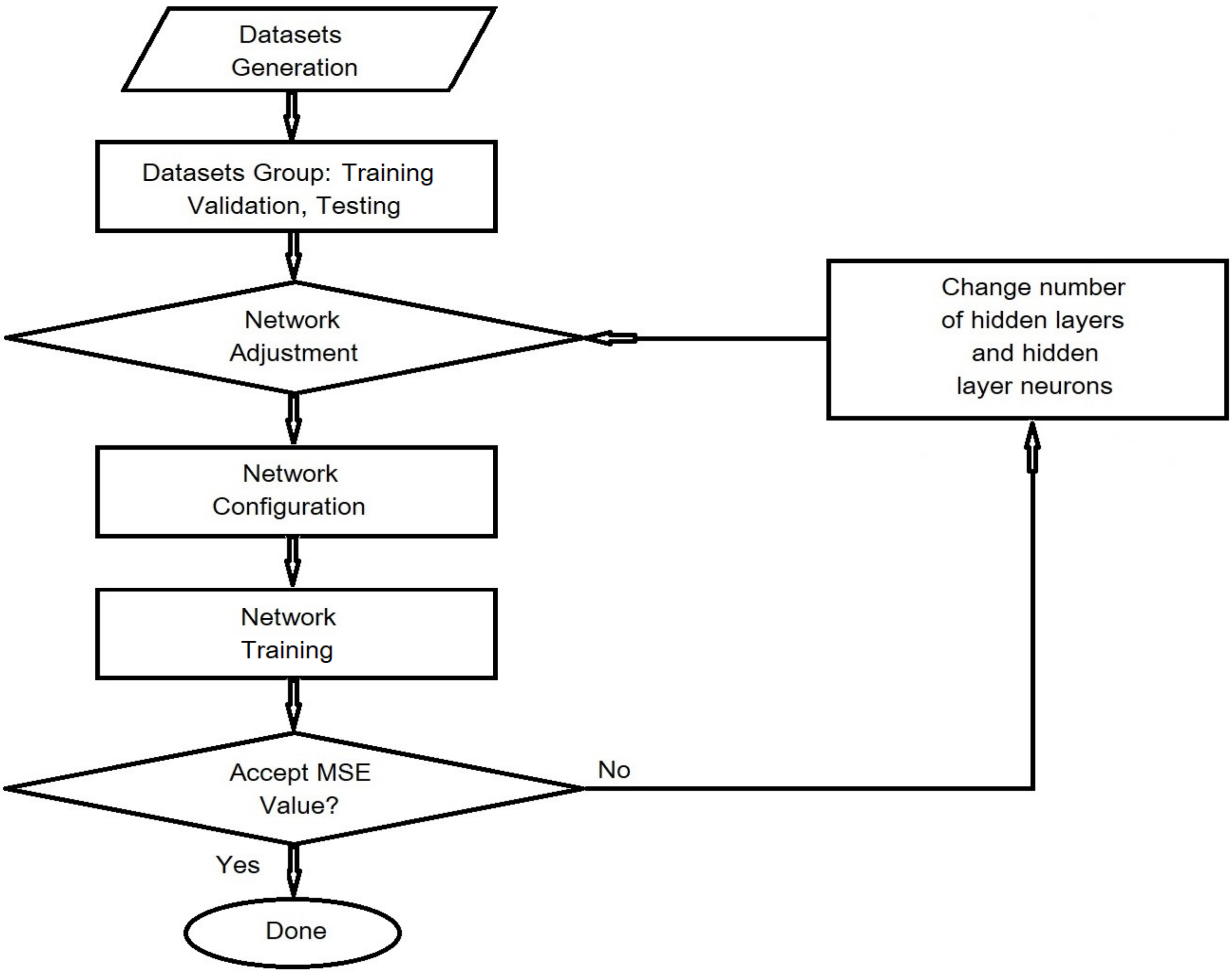

Fig. 9. Flowchart of the Neural Network Processing

Training error:

$e = \Delta K_G - T.$ (18)

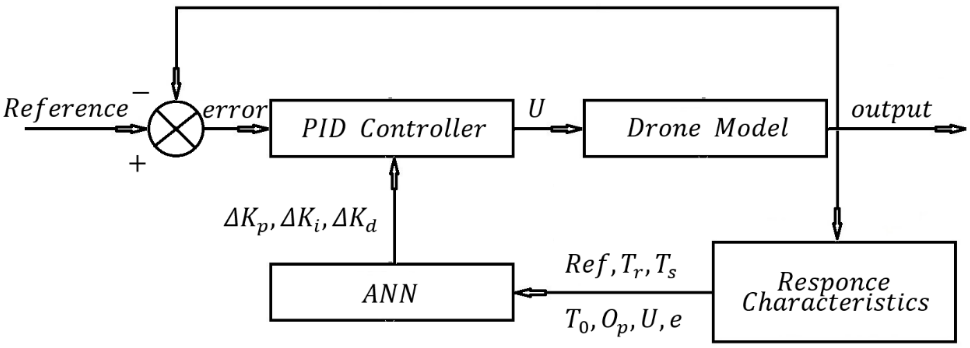

Fig. 10. Block diagram of PID-ANN for drone system

The ANN used in this research is also judged as a controlled learning process since targeted outputs are used as objects for the ANN to achieve during training. The real output from ANN training is linked with the trained output to get the errors and the training performances is estimated using MSE of the cumulative errors and the regression values [22]. Fig. 10 shows the block diagram of the whole system.

Table 4: ANN training cases

Cases	Hidden Layers		MSE Performance Values		
			Train (70 %)	Validation (15 %)	Test (15 %)
Case 1	[18]	z	1.278	2.162	2.377
		Roll	1.287	2.494	2.007
		Pitch	0.401	0.727	0.835
		Yaw	0.933	1.752	1.974
		Average	**0.975**	**1.784**	**1.798**
Case 2	[18 18]	z	0.245	2.957	3.193
		Roll	0.062	2.802	1.872
		Pitch	0.052	0.836	1.032
		Yaw	0.362	2.103	1.753
		Average	**0.180**	**2.175**	**1.963**
Case 3	[18 18 18]	z	0.022	2.682	3.131
		Roll	0.017	1.918	2.746
		Pitch	0.006	0.922	1.106
		Yaw	0.178	1.81	2.156
		Average	**0.056**	**1.833**	**2.285**
Case 4	[8 4]	z	1.511	2.128	2.521
		Roll	1.482	2.423	1.707
		Pitch	0.571	0.898	0.852
		Yaw	1.076	1.774	2.259
		Average	**1.16**	**1.806**	**1.835**
Case 5	[14 8]	z	0.81	2.483	3.118
		Roll	0.621	2.322	1.899
		Pitch	0.212	0.806	0.776
		Yaw	0.648	2.09	1.828
		Average	**0.203**	**1.305**	**1.905**
Case 6	[18 9]	z	0.672	1.992	2.546
		Roll	0.403	2.182	2.282
		Pitch	0.168	0.838	0.882
		Yaw	0.669	1.938	1.538
		Average	**0.478**	**1.738**	**1.812**

Different network conditions are being given in Table 4 to identify the best network to be used in this publication. Case 1 to Case 3 characterise the regression values acquired from networks with different statistics of hidden layers but with similar neurons. Case 4 to Case 6 symbolize the regression values gained from networks with the same number of hidden layers but different neurons [23]. Each dataset was arbitrarily split as follows: 70% of the data for the training set, 15% for a network validation set, and 15% for evaluating the network performance (test set). Splitting the dataset into these three subsets carries validate that the model studies effectively, simplifies well to new data, and carries stable performance estimations. The test set is used to measure the final performance of the trained model on unseen data and the best network will depend on MSE values from the test set.

Based on Table 4, it is found that Case 6 produces the most appropriate outcome with the lowest MSE value of the test set [22]. The trained network from Case 6 is selected to create outcome in the next section. As reasoning from this research, the ANN comprises three hidden layers and uses 300 epochs.

The Levenberg-Marquardt training method was chosen as of its common use in the field of artificial neural network (ANN) research [22]. The minimum gradient is set to be 1e-7 and the training act is estimated using MSE with a target of 1e-6. All these conditions were concluded after being altered numerous times according to the network routine.

4.3.1. Simulation Environment

The MATLAB/Simulink environment was utilized to operate the models. The simulation is acted on a MATLAB/Simulink software for drone system validation, together with the designed controller. MATLAB uses for the dataset and the ANN while Simulink simulates the whole drone system including aimed controllers.

In this part, results are performed through altitude and attitude balance using step input and multi-level tracking, and position tracking. Disturbances are also involved throughout multi-level altitude tracing to review the reliability of the indicated controller compared to manual tuned PID.

4.3.2. Altitude and Attitude Stabilization

Inspecting Fig. 11-14, the auto-tuned PID controller has seriously improved the drone performance related to the manual tuned PID controller in terms of reduction overrun, gradient time and settling time. The projected controller has determined its productivity in stabilizing the altitude and drone attitude, effectively adjusting it maintains under the influence of step inputs.

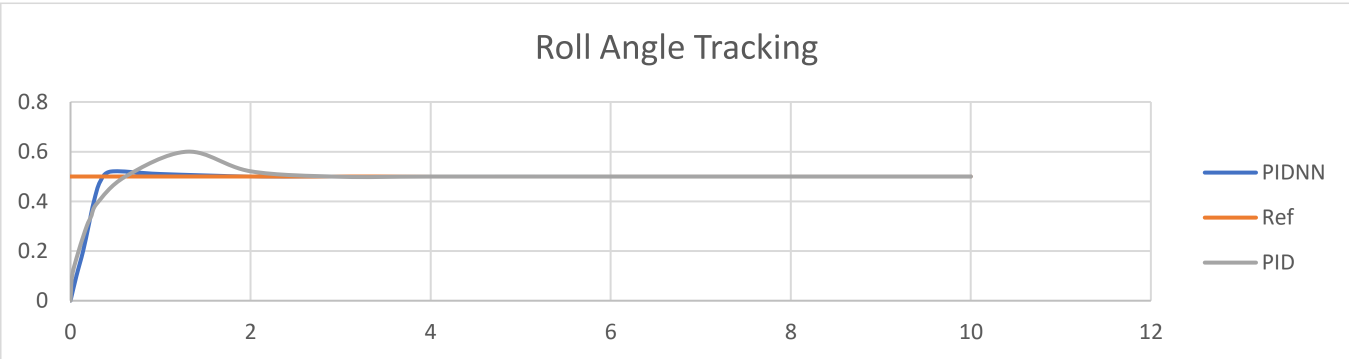

Fig. 11. Roll angle tracking

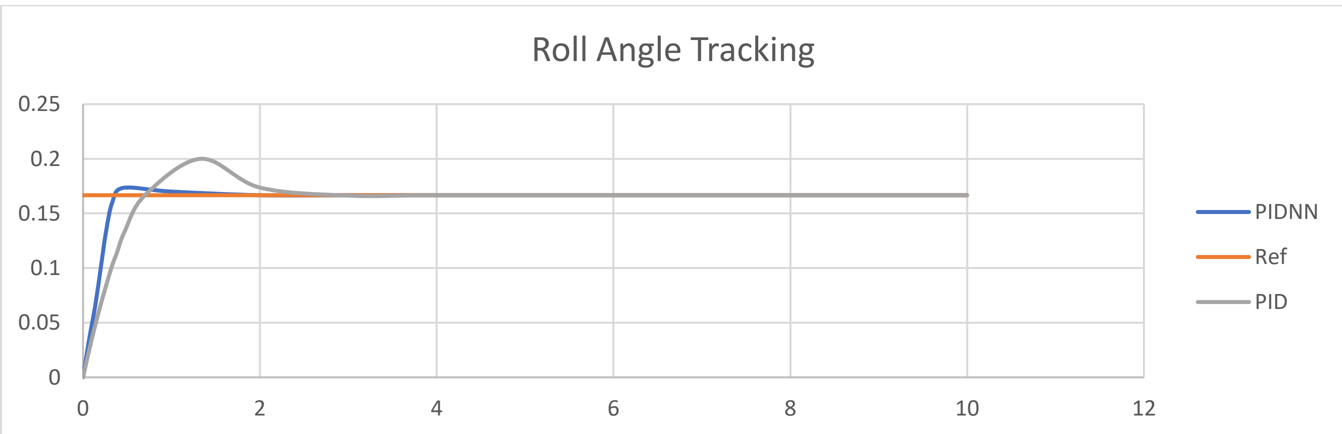

Fig. 12. Pitch angle tracking

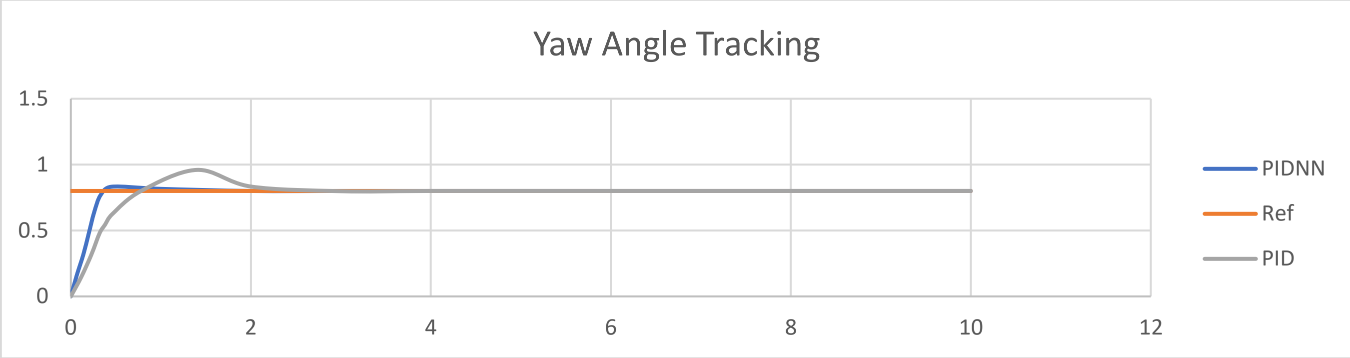

Fig. 13. Yaw angle tracking

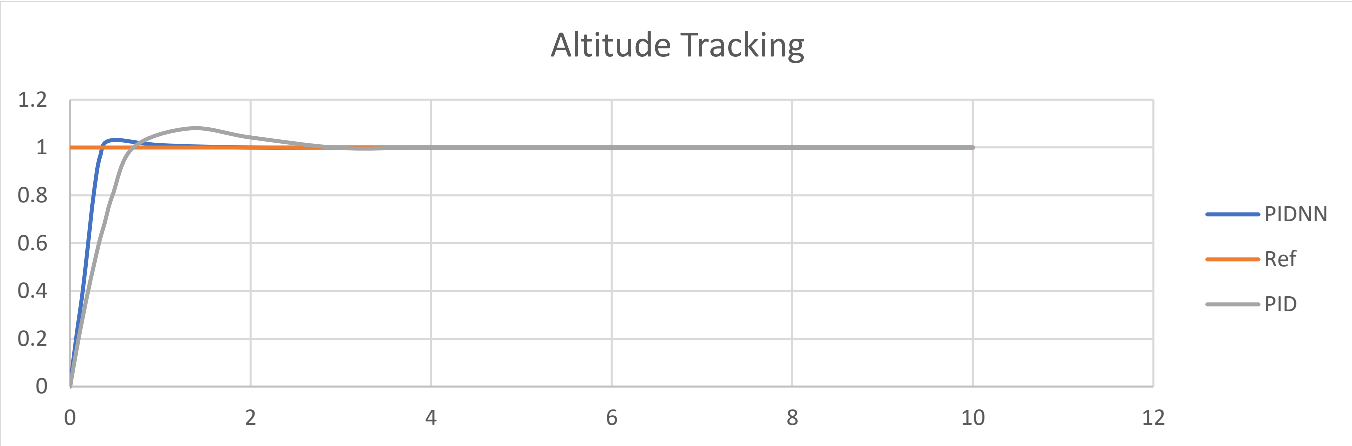

Fig. 14. Altitude tracking

4.3.3. Altitude and Attitude Multi-Level Tracking

A conventional intelligent drone model is also tested using the auto-tuned PID gains where all attitude angles are determined to zero. In Fig. 15, the outcome specifies lesser overshoot with increased settling time related to the manual tuned PID gains. Both controllers successfully track the required input reference, while differing performing features are examined.

From the tests outcomes, all multi-angle tracking of roll, pitch and yaw using the resultant gains from the designed controller can follow the input reference much better than the normal PID controller. But, it is challenging for both controllers to follow a sudden change in reference such as 1-2 seconds while a constant reference is simpler to track.

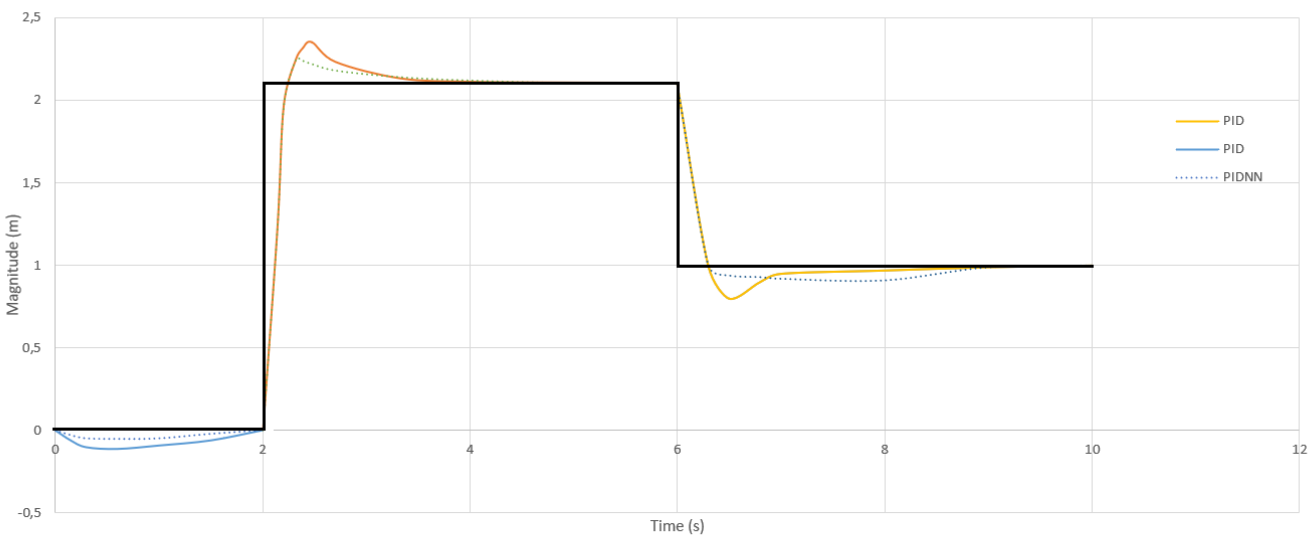

Fig. 15. Multi-level simulation of linear-model drone (altitude tracking): black line - reference

Table 5: The error execution quantities comparison of flight tracing

X-position								
PID					PIDNN			
X	IAE	ISE	ITAE	ITSE	IAE	ISE	ITAE	ITSE
Circular	0.61	0.02	18.21	0.52	0.35	0.005	10.3	0.17
Lemniscate	0.89	0.04	26.93	1.06	0.51	0.01	15.36	0.35
Y-position								
PID					PIDNN			
Y	IAE	ISE	ITAE	ITSE	IAE	ISE	ITAE	ITSE
Circular	1.08	0.42	32.33	12.68	0.78	0.27	23.84	8.06
Lemniscate	2.16	0.52	64.84	15.61	2.16	0.23	64.65	6.96
Roll angle								
PID					PIDNN			
ϕ	IAE	ISE	ITAE	ITSE	IAE	ISE	ITAE	ITSE
Circular	0.64	0.11	19.18	3.19	0.23	0.02	6.86	0.79
Lemniscate	3.19	0.43	95.66	12.96	0.91	0.04	27.18	1.26
Pitch angle								
PID					PIDNN			
θ	IAE	ISE	ITAE	ITSE	IAE	ISE	ITAE	ITSE
Circular	0.46	0.03	13.92	0.75	0.38	0.02	11.46	0.51
Lemniscate	0.65	0.03	19.34	1.03	0.55	0.02	16.42	0.73

We can arrange from results, steady-state error happens in both simulations, where it involves a longer time to settle following the desired path (the position tracking of drone in circular and lemniscate trajectory tracking for 20 seconds of each simulation). The new gains keep a adequate overshoot and smaller steady-state error in position tracking. Table 5 shows the controllers error based on index performing of IAE, ISE, ITAE and ITSE for X-position, Y-position, roll-angle, and pitch-angle, respectively [26]. Overall error confirms that the new gains (PIDNN) are linked with much lower error values if compared to the manual tuned gains.

To additional estimation the efficiency of the proposed controller, a wind gust disturbance is added into the altitude of the drone model as indicated in Fig. 16.

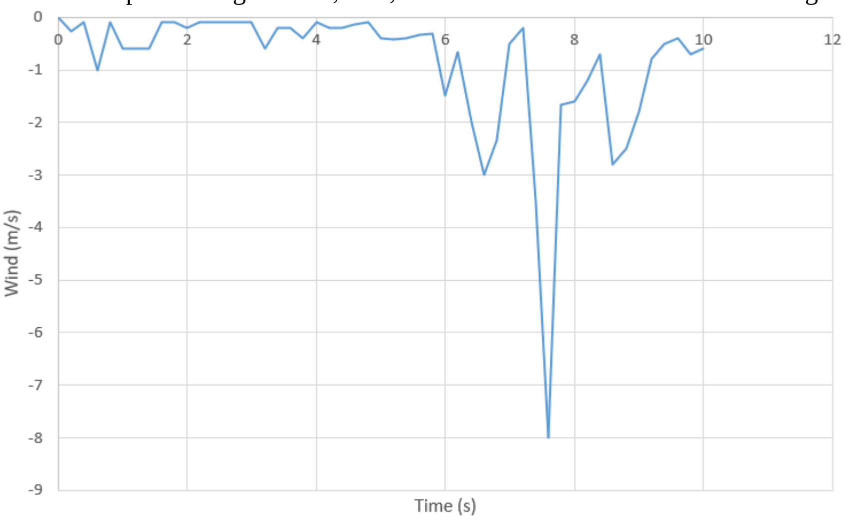

Fig. 16. Wind gust disturbance model

The demonstration of both controllers in doing perturbation is nearly the identical based on Fig. 17.

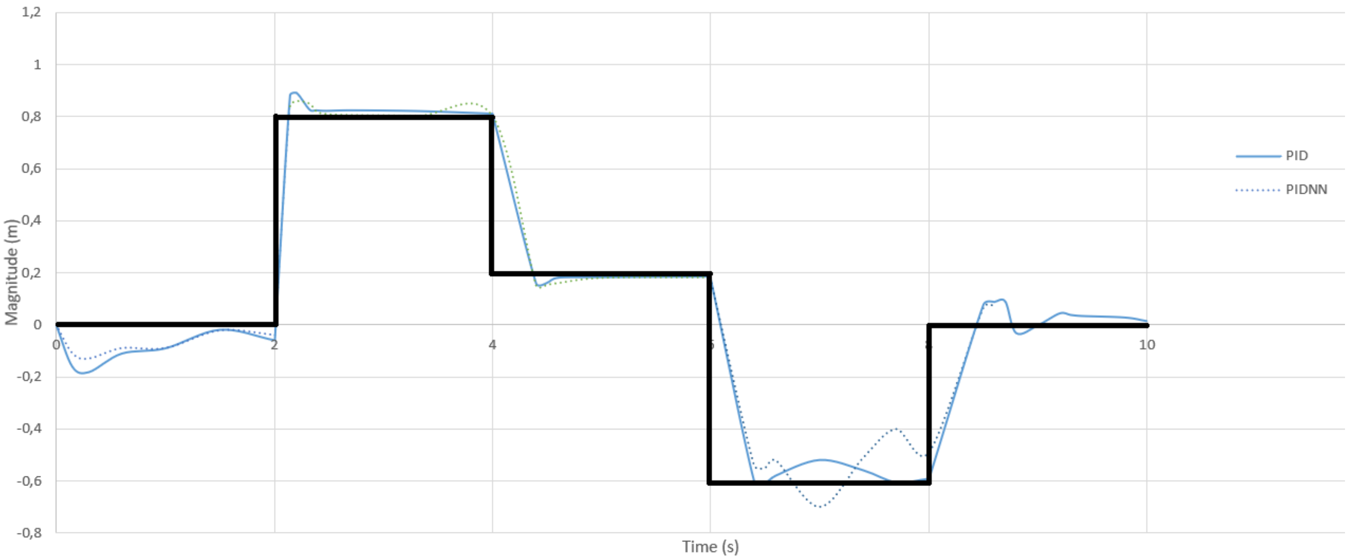

Fig. 17. Variation of altitude during perturbation (black line represent signal builder)

The auto-tuned gains (PIDNN) can hold expanding and falling edge adjusted than the manual tuned gains and have improved merging on the setpoint.

Grounded on the numerical evaluation, the auto-tuned PID controller performs 1.6 % better than the manual tuned PID controller. The steps taken from delivering the datasets to network training have generated a more adaptive PID controller gains that can consistently respond to altered reaction acts of the drone states in an offline approach.

All the results have stated small corrections being received by the designed controller if contrasted to the manual tuned one. This research has positively submitted the design and implementation of the auto-tuned PID controller using neural network in evaluation to manual tuned PID controller.

The concept of producing auto-tuned PID gains using an artificial neural network (ANN), trained through explore of numerous response points, has been effectively showed in this publication.

To improve the efficiency of the network, further datasets with varied setpoints should be made. Moreover, combining real-time feedback from the drone system could advance improve the implementation of the proposed controller.

5. CONCLUSION AND FUTURE WORK

Was proposed a 3D drone dynamic simulation module on the original simulation program. Justified on Newtonian mechanics and Euler angles, the physical model is validated. The model runs a fixed-point control for the generation. We also created weather affect effects into the model through an environmental computational model. The model worked effectively and practically simulated the drones' drive under various circumstances.

It is assumed that simulation module could be appropriate in the IoT area. A drone, used as a sensor or data collector, is an important piece in edge processing. Experimental simulation model can be employed to adapt controlling sense or stability of drones, as they are controlled by operators or as self-controlled.

For the future, it will applicably determine more environmental impacts into projected model. For example, how a drone will be disrupted in a tiny space and where the airflow made by a drone itself cannot be ignored.

In conclusion, this article has demonstrated the executing of auto-tuned neural network PID controller in controlling a smart drone, with a virtual analysis accomplished against a manually tuned PID controller. Unlike regular PID manual tuning, which involves complete analysis, the use of a neural network permits the study of a dataset having 500 simulation solutions with changing PID gains, leading to enhance autotuning of the proposed controller. The dataset was made using an initial of PID gains accomplished from physical tuning, and numerous networks with many designs were anticipated. The outcome showed that the auto-tuned PID gains provided decreased exceed and very low steady-state error found to manually tuned PID gains, though with a disadvantage in resolving time. During perturbation, the auto-tuned PID controller outrun the physical tuned PID controller by 1.6 %. Specifically, improvements were performed mainly in minor setpoints, intending the necessary for creating datasets giving to diverse setpoints to boost the flexibility of the neural network.

For future research, it is recommended to continue the study range of the neural network by supplying datasets with various setpoints.

Declarations

All stages were performed personally by author.

Funding

No funding. Personal expenses.

Conflicts of interest/Competing interests

No conflicts of interest.

Code availability

Matlab license number: 968398

Authors' contributions

All research, analysis and text were made exclusively by author.

Ethics approval

According to human's social ethics.

Consent to participate

Yes.

Consent for publication

Yes.

Availability of data and material

Yes.

REFERENCES

1) L. Xinxi, X. Zhihuan, Application of IoT Quadrotor Dynamics Simulation, *MDPI, Electronics, Vol. 11, Issue 4, 2022*. Available online: *https://www.mdpi.com/2079-9292/11/4/590* .
2) P. Borra, Gyrosucer by Keyence. Available online: *https://www.oocities.org/bourbonstreet/3220/gyrosau.html* .
3) An Overview of Cloud Computing and Leading Cloud Service Providers, *Int. Journ. of Comp. Eng. & Techn., 15(3), PP. 122-133, May 2024.* Available online: *https://www.researchgate.net/publication/381120656_An_Overview_of_Cloud_Computing_and_Leading_Cloud_Service_Providers* .
4) P.G.T., Healey, G. White, A. Eshghi, A. Light, Communication Spaces, *Computer Supported Cooperative Work (CSCW), 17(2-3), PP. 169-193, April 2008.* Available online: *https://www.researchgate.net/publication/220169070_Communication_Spaces* .
5) M. Marchese, A. Moheddine, A. Moheddine, IoT and UAV Integration in 5G Hybrid Terrestrial-Satellite Networks, *Sensors, 19(17), P. 3704, 2019.* Available online: *https://doi.org/10.3390/s19173704* .
6) Google Plans Drone Delivery Service for 2017. Available online: *https://www.bbc.com/news/technology-34704868* .
7) Z. Ren, H. Li, Y. Zhao, D. Zhang, The study of wind resistance testing equipment for unmanned helicopter, Conf., IEEE 2nd Advanced Information Technol., *Electronic and Automation Control Conference (IAEAC), March 2017*. Available online: https://www.researchgate.net/publication/320606857_The_study_of_wind_resistance_testing_equipment_for_unmanned_helicopter .
8) Y. Sen, W. Zhongsheng, Quad-rotor UAV Control Method Based on PID Control Law, *Int. Journ. of Advanced Network Monitoring and Controls, Vol. 2, No. 1, 2017*. Available online: https://intapi.sciendo.com/pdf/10.21307/iccnea.2017.24 .
9) 9. N. Zosimovych, Hierarchical Flight Control System for Tilt-Rotor Drones, *Journ. of Eng. and Applied Sc. Technol., Vol. 6(5), PP. 1-7, 2024, ISSN: 2634 – 8853*. Available online:
10) https://onlinescientificresearch.com/articles/hierarchical-flight-control-system-for-tiltrotor-drones.pdf
11) N. Zosimovych, Artificial Intelligence Techniques Application in UAS, *Transactions on Eng. and Computing Sc., Vol. 11, No. 1, Feb. 25, 2023, online, PP. 45-61*. Available online: https://journals.scholarpublishing.org/index.php/TMLAI/article/view/14012 .
12) J. de J. Rubio, J.H. Pérez-Cruz, Z.Z. Beltrán, J. Salinas, Comparison of Two Quadrotor Dynamic Models, *IEEE Latin America Transactions 12(4): 531-537, DOI: 10.1109/TLA.2014.6868851, May 2014*. Available online: https://www.researchgate.net/publication/262935100_Comparison_of_Two_Quadrotor_Dynamic_Models
13) M.N. Duc, T.N. Trong, The quadrotor MAV system using PID control, *Conf. Proc. Mechatronics and Automation (ICMA), IEEE Int. Conf., Beijing, China, DOI: 10.1109/ICMA.2015.7237537, Aug. 2015*. Available online: https://www.researchgate.net/publication/281619233_The_quadrotor_MAV_system_using_PID_control
14) C. Masse, O. Gougeon, D-T. Nguyen; D. Saussié, Modeling and Control of a Quadcopter Flying in a Wind Field: A Comparison Between LQR and Structured $\mathcal{H}\infty$ Control Techniques, *Int. Conf. on Unmanned Aircraft Systems (ICUAS), 12-15 June 2018*. Available online: https://ieeexplore.ieee.org/xpl/conhome/8428793/proceeding .
15) H. Suresh, A. Sulficar, V. Desai, Hovering control of a quadcopter using linear and nonlinear techniques, *Int. Journ. Mechatronics and Automation, Vol. 6, No. 2/3, 2018*. Available online: https://harikrishnansuresh.github.io/assets/quadpaper.pdf .
16) N. Zosimovych, Hierarchical Flight Control System for Tilt-Rotor Drones, *Transactions on Eng. and Computing Sc., 12(3), PP. 73-87, 2024, DOI:10.14738/tecs.123.17047*. Available online: https://journals.scholarpublishing.org/index.php/TMLAI/article/view/17047 .
17) W. Khan, M. Nahon, A Propeller Model for General Forward Flight Conditions, *Int. Journ. of Intelligent Unmanned Systems Vol. 3 (2/3), November 2015, DOI: 10.1108/IJIUS-06-2015-0007*. Available online: https://www.researchgate.net/publication/284070935_A_Propeller_Model_for_General_Forward_Flight_Conditions .
18) H.C.T.E. Fernando, A.D. de Silva, S.R. Munasinghe, Modelling, simulation and implementation of a

quadrotor UAV, *IEEE 8th Int. Conf. on Industrial and Information Systems, 2013*. Available online: https://www.semanticscholar.org/paper/Modelling%2C-simulation-and-implementation-of-a-UAV-Fernando-Silva/6216cd7c459c407de7bc6cc00411e54aef6784fd

19) A. Tabassum, R. Vuppala, H. Bai, K. Kara, Variance Reduction of Quadcopter Trajectory Tracking in Turbulent Wind, *IFAC-Papers OnLine Vol. 54, Issue 20, 2021, PP. 102-107*. Available online: https://www.sciencedirect.com/science/article/pii/S2405896321022047 .
20) 13. N. Zosimovych, A Flight Control System for Smart Tilt-Rotor Drone, *SADI Int. Journ. of Sc., Eng. and Technol., Vol. 12, Number 1, January-March 2025, ISSN: 2837-1941*. Available online: https://sadijournals.org/index.php/SIJSET/article/view/988/922 .
21) H.J. Kim, D.H. Shim, A flight control system for aerial robots: Algorithms and experiments. *Control Engineering Practice, 11(12), 1389-1400, 2003*. Available online: https://people.eecs.berkeley.edu/~sastry/pubs/PDFs%20of%20Pubs2000-2005/Pdfs%20of%20Misc.%20Others/Kim,H.Jin/KimFlightControlSystem2003.pdf .
22) N.K. Tran, E. Bulka, M. Nahon, Quadrotor control in a wind field. *Proc. of the Int. Con. on Unmanned Aircraft Systems, Denver, CO, USA, 9–12 June 2015.*
23) W. Khan, M. Nahon, A Propeller Model for General Forward Flight Conditions. *Int. Journ. Intell. Unmanned Syst. 2015, 3, 72–92.*
24) Thunder S550 Hexacopter Frame Kit W/PCB. Available online: https://www.hobbytech.com.au/products/rc-multi-rotors/kit-frame/thunder-s550-hexacopter-frame-kit/ .
25) Y.J. Zhai, Z.Y. Lin, Design of Software-Defined-Satellite-based PID Attitude Control Application in Python, *Int. Proc. of the 2018 IEEE 4th Inf. Technol. and Mechatronics Eng. Conf. (ITOEC), Chongqing, China, 14–16 Dec. 2018, PP. 133–137.*
26) N. Zosimovych, A. Singh, Tilt-Rotor Drone Flight Control System Modeling. In: *D.K. Singh, S. Hegde, A. Mishra, Recent Developments in Mechanics and Design. INCOME 2024. Lec. Notes in Mech. Eng. Springer, Singapore, PP. 87–97*. Available online: https://doi.org/10.1007/978-981-97-3874-8_8 .
27) N.H. Sahrir, M.A.M. Basri, Intelligent PID Controller Based on Neural Network for AI-Driven Control Quadcopter UAV. *Int. Journ. of Robotics and Control Systems, Vol. 4, No. 2, 2024, PP. 691-708*. Available online: http://pubs2.ascee.org/index.php/ijrcs .
28) N. Zosimovych, IoT Drone Controllers Simulation Platform. *XVII Int. Sc. and Eng. Conf. Proc. «ABIA-2025», April 22-24, 2025, Kyiv: National Aviation University, 2025, PP. 654-660.*
29) N. Zosimovych, A Control System for Smart Drone, *Int. Journ. of Adv. Eng. and Management Research, Vol. 11, No. 01, PP. 41-59, 2026, ISSN: 2456-3676.* Available online: https://ijaemr.com/link/781 .