

Radial Basis Function Network Based Self-Adaptive PID Controller for Quadcopter: Through Diverse Conditions

Nur Hayati Sahrir ^{a,1}, Mohd Ariffanan Mohd Basri ^{a,2,*}

^a Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia

¹ nurhayatisahrir1998@graduate.utm.my; ² ariffanan@fke.utm.my

* Corresponding Author

ARTICLE INFO

Article history

Received December 18, 2023

Revised January 17, 2024

Accepted March 06, 2024

Keywords

Neural Network;

PID Controller;

Quadcopter UAV;

Radial Basis Function

ABSTRACT

A quadcopter is an underactuated and nonlinear system which requires a robust controller to aid in maneuvering the quadcopter during flight. A Proportional-Integral-Derivative (PID) controller is easy and suitable to implement, and its efficiency is proved in quadcopter control. However, a PID controller with fixed parameters is inadequate enough to control a quadcopter system with different inputs or perturbations. This paper proposes the development of a self-adaptive PID controller assisted by Radial Basis Function (RBF) Network, to improve the function of the PID controller and help a quadcopter to better adapt towards different inputs and situations, independently. This work contributes to introducing RBF-PID controller to adaptively fly the underactuated quadcopter through different trajectory and perturbations using simulation. By using the hidden Gaussian function to train the current input, estimate the suitable output and update the Jacobian Information during system control, the PID gains can change adaptively during flight, additionally with the help of Gradient Descent Method (GDM). The proposed method is compared to the traditional PID controller tuned using the PID Tuner App in Simulink. Different inputs are given to test the altitude, attitudes, and position tracking such as step, multistep, sine wave, circular and lemniscate trajectory. The simulated results proved the robustness of RBF-PID in enhancing the disturbance rejection capacity by 13% to 25% in the presence of perturbations (sine wave and wind gust) compared to PID controller. The proposed controller can ensure quadcopter's flight stability through perturbations that is within the quadcopter's limitations.

This is an open-access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



1. Introduction

Quadcopters' technology has manifested a wide area of application which makes many researchers take part in optimizing the technology over the years. The current famous commercial quadcopter in the market is the DJI Mavic model which can be maneuvered both manual and auto-pilot. By being able to do aerial photography [1] and videography [2], aerial mapping [3], aerial inspection [4], anticipate in weather forecasting [5] activities and more to mention, it is undeniable that quadcopters have become the center of attention by both individual and industries. Different applications require different additional models to design with regards to the problem at hand. In this paper, the specific quadcopter to be researched is the quadcopter, which is known for its four rotors and usually being utilized for personal use. This type of quadcopter has its own root problem

maneuvering before jumping into any commercial application. As an underactuated system with coupling dynamics, highly nonlinear characteristics and naturally unstable [6], a controller is needed to correct the states error of the closed-loop system [7] to avoid states deviation and thus, provide stability.

Currently, Proportional-Integral-Derivative (PID) controller [8], [9] is widely used on various systems and that includes quadcopter for its simple design. The problem here is the control method alone has a high probability of being affected by the unexpected perturbations that might cause undesired impact on the quadcopter itself. There are various other kinds of controllers being researched nowadays to stabilize the system propagation especially during trajectory tracking such as backstepping control (BC), fuzzy adaptive PID (F-PID), sliding mode control (SMC), and model predictive control (MPC). The BC method is one of the most researched control methods for quadcopters for its unique recursive design philosophy following the Lyapunov Theorem which is robust towards parametric changes and provides stability [10]. However, the over-parameterization of BC makes it difficult to find out accurately for satisfying performance [11]. To overcome the problem, the work [12] proposes an Optimal Model Free BC (OMFBC) method combined with Cuckoo Search Algorithm for quadcopter to deal with the unknown nonlinear system dynamics and external disturbances. The ideal parameters of BC are obtained offline using gravitational search algorithm (GSA) in [13]. The fuzzy logic control is quite useful in some ways. It was built as an compensator to reduce the chattering effect of SMC in [14]. As part of main controller in a system, F-PID can adjust PID parameters using the rules of fuzzy reasoning [15]. Fuzzy control can function properly when dealing with underactuated systems if the designer has good experiences in linguistic control rule while analyzing system stability. Otherwise, difficulties may occur due to approximation errors and the presence of unknown nonlinear function [11]. Besides BC, SMC is also well-known for its ability in nonlinear control strategies. An adaptive PID (APID) controller was introduced in [16] and the robustness during trajectory tracking using a real hardware was proven. In [17], SMC is combined with PID controller to withstand external disturbances and uncertainties to robustly pursue a desired time-varying trajectory with exponential convergence. Moreover, a tracking output-control strategy that is made of finite-time sliding-mode observer to detect some kinds of disturbances and determine a full state from the quantifiable output is also included. Although SMC has a major drawback of chattering effects, there are several methods introduced to eliminate it but the problem of many parameters to be tuned in the control law remains the same [18]. One type of optimization control technique is Model Predictive Control (MPC). Predicting the system's future behavior and optimizing control inputs on a receding horizon is the main idea [19]. In [20], a cascaded control approach is utilized by putting the MPC on the outer-loop to provide virtual accelerations as attitude references into the inner-loop. An acceleration reference signal that is both continuous and differentiable is produced by integrating the MPC input (jerk) with respect to time. Attitude control using fuzzy MPC in [21] is based on disturbance observer to compensate the disturbance effect. As MPC can handle input with constraints, it is derived from the Takagi-Sugeno fuzzy model to produce a more accurate model. It results in a more stable adjustment process and it can perform well in a wide operating range.

All mentioned controllers have shown excellent results especially after controller or system modifications. However, an excellent outcome requires complex computations and complicated structures. Moreover, there is an easier, simpler yet functioning controller that has a large space for improvement. A PID controller is a linear controller which is basically unfavorable when it comes to nonlinear systems such as quadcopter. Traditional tuning method like Ziegler-Nichol's does not provide optimal control and may cause the system's instability and damage [6]. Since PID controller is flexible to be improved in this paper, there are works on improving a PID controller for quadcopter system such as using offline gains optimization method [6] and online gains optimization method [12]. This is to reduce the disadvantages of the acting-alone PID controllers, in search of more robust PID controllers that can handle nonlinearities under perturbations and uncertainties during flight [11]. One of better ways for PID gains tuning is by using the metaheuristic search algorithms. Particle Swarm Optimization (PSO) was applied in [22] to optimally tune the PID controller in phi axes. By using multi-objective cost function, the

performance can be altered until optimal performance is achieved. In [23], Genetic Algorithm (GA) with PID controller is adapted on UAV transfer function and all the tedious trial-and-error method can be avoided. Different selection types of GA are tested to find the best selection type to produce the best step response.

However, for a quadcopter to have a robust flight through different perturbations, an online and continuous PID tuning is preferred. The use of artificial neural network (ANN) has greatly contributed to flight technology such as quadcopters. For example, adapting an ANN into the quadcopter system to continuously adjust PID parameters with respect to minimizing the tracking error was also done in [24]. As part of the main controller in quadcopter system, the auto-tuning of PID parameters are based on gradient descent technique [14], [25]. A sigma-pi neural network (SPNN)-augmented dynamic inversion control system is developed as a compensator to reduce model errors and implemented in a quadcopter model simulation with PID controller in [26] to improve quadcopter's performance. Position observer and attitude observer are also added, and an optimal learning rate is obtained using the Hardware-in-the-loop (HITL) test. The control law of neural network based PID controller for quadcopter in [27] is not model-based, where there is no need for system linearization. This means that the proposed controller can handle a complex nonlinear system such as quadcopter. There is also research on using offline method as the training phase for the online method to function better [28].

There is a few research utilizing Radial Basis Function (RBF) with PID controller to compensate the approximated uncertainties. To obtain Jacobian information for system identification, RBF neural networks are employed in [29] for Stewart platform. GDM-based RBFNN is utilized to dynamically adjust the PID parameters. The simulation results reveal that the proposed method enhances control precision, overcomes theoretical model errors, and significantly improves tracking performance. In [30], an energy storage system uses RBF-PID controller to stabilize the output power of photovoltaic (PV) especially during changes of external light and temperature. The response speed and anti-interference ability have improved after the implementation. An RBF-PID controller is also implemented as the control system for battery charging and discharging [31]. The research claims that RBF neural network can compensate for the shortcomings of back-propagation (BP) neural network in terms of slow convergence. Next, Gaussian function is usually being used as the activation function in the hidden layer of RBF network however, in [32], a Flat-Top Window (FTW) weighted function is being used in comparison to "traditional" FT weighted function windows. The work is applied on Quanser helicopter. The latest work of using RBF-PID on quadcopter was realized in [33] where a three-phase learning of RBF network is adopted for PID gains adjustment. The effectiveness of the work also being realized using a QAV 250 quadcopter through outdoor experiments. As can be seen from the previous works, an intelligent controller can help real-time PID tuning based on ANN. Furthermore, works related to RBF network produces excellent results with PID controller. Gradient descent method (GDM) is one of the strategies that can adjust the PID gains according to changes in system dynamics. To improve the control accuracy, the network must identify the system dynamics while updating the center point, node width, and the weights between layers. Then, the PID parameters are updated accordingly based on GDM. Unfortunately, there are not many papers let alone recent ones that describe the use of RBF-PID for quadcopters.

In this paper, a Radial Basis Function (RBF) network is chosen to online-tune PID controller in a quadcopter. It was found that an RBF network has simpler architecture with fixed three-layer network compared to Multi-Layer-Perceptron (MLP) network and it works faster than MLP. The RBF network also works on local approximation where the output is determined based on information from specific hidden units in specific local receptive fields. The initial parameters of RBF network must be essentially corrected until a good output is formed. The main contributions of this work are as follows:

1. Development of RBF networks to adaptively modify PID parameters in response to different inputs and real-time feedback with constant parameterization.

2. Implementation of RBF-PID controller to the nonlinear dynamic of quadcopter using various different tracking conditions to prove its parameters adaptation online.
3. Performance comparison of RBF-PID controller to the traditionally tuned PID approach in response speed and tracking precision.

This paper is organized as the following structure: [Section 2](#) combines the introduction to quadcopter physical structure and dynamics based on Newton-Euler formulation, and the presentation of the proposed controller approach where RBF-PID is designed using Gaussian function and GDM is detailed. [Section 3](#) presents the trajectory tracking results from altitude, attitude and position that are obtained from MATLAB simulations, in which discussions are also included. Lastly, a final conclusion from this work and future work recommendation is made in [Section 4](#) to provide enhancement for further improvement.

2. Research Methodology

2.1. Quadcopter Structure and Dynamics

As the name implies, ‘quad’ in quadcopter means it consists of four rotors that assist in flying. Two diagonal rotors have same way of spinning (clockwise/counter-clockwise), which is different from the other two diagonal rotors (counter-clockwise/ clockwise) during flight and any movement, to ensure a successful flight based on Newton’s Third Law Physics concept.

A quadcopter has three rotational motions and three translational motions that move along each axis of dimension, (X, Y, Z) . Among all six motions, there are four motions (ϕ, θ, ψ, Z) that can be actuated directly from rotors speed changing and another two motions (X, Y) that depend on the other motions to be actuated. The dependent motions are what we call as coupling dynamics in the previous section. [Fig. 1](#) visualize the (ϕ, θ, ψ, Z) motions, each with different speed allocation for each rotor, to provide different movements as stated. For (X, Y) motions, rolling and pitching of quadcopter will produce translational motions of quadcopter along Y and X axis of dimension. The quadcopter will move from one place to another respective to the earth frame.

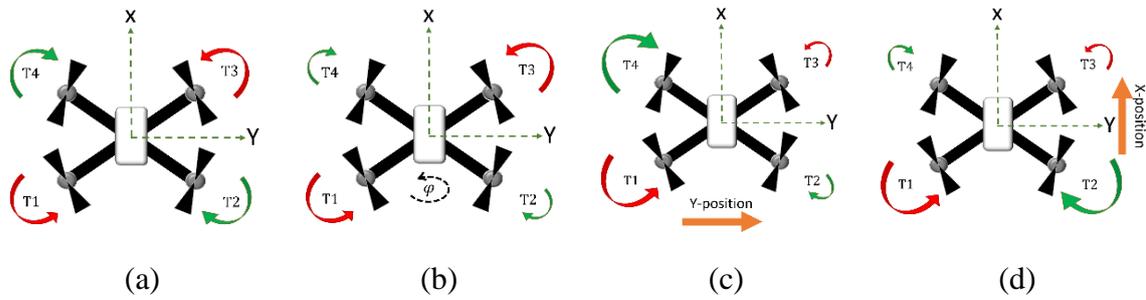


Fig. 1. Quadcopter motions (a) hover, z (b) yaw, ψ (c) roll, ϕ (d) pitch, θ

A free body diagram of quadcopter is visualized in [Fig. 2](#), which shows basic forces that acted on a quadcopter during flight. The yaw angle (ψ), pitch angle (θ), and roll angle (ϕ), which together constitute vector $\Omega^T = (\phi, \theta, \psi)$, determine the quadrotor’s orientation while vector $r^T = (x, y, z)$ provides the vehicle’s position in the earth frame. The rotation matrix R provides the vector transformation from the body fixed frame to the earth frame and is given by equation (1).

Where C and S represent cosine and sine respectively. We derive a first set of differential equations that characterize the acceleration of the quadrotor since the thrust force produced by rotor i , $i = 1, 2, 3, 4$ is $F_i = K_T \cdot w_i^2$, where K_T is the thrust factor and w_i is the speed of rotor i .

$$R_b^e = \begin{pmatrix} C\theta C\psi & S\phi S\theta C\psi - S\psi C\phi & S\theta C\phi C\psi + S\phi S\psi \\ S\psi C\theta & S\phi S\theta S\psi + C\phi C\psi & S\theta S\psi C\phi - S\phi C\psi \\ -S\theta & S\phi C\theta & C\phi C\theta \end{pmatrix} \quad (1)$$

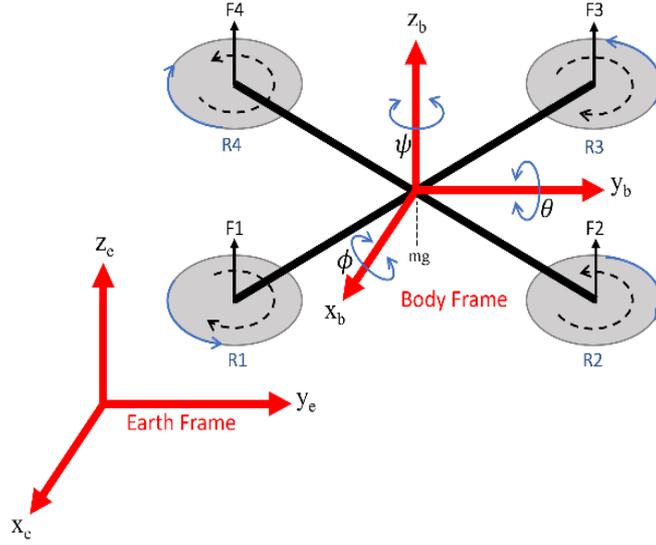


Fig. 2. Quadcopter free body diagram

$$\ddot{r} = g \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} - R \cdot b/m \sum_{i=1}^4 w_i^2 \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \quad (2)$$

A second set of differential equations is obtained using the inertia matrix I (which is a diagonal matrix with the inertias I_x , I_y , and I_z , on the main diagonal), the rotor inertia J_r , the vector M that describes the torque given to the vehicle's body, and the vector M_G of the gyroscopic torques.

$$I\dot{\omega} = -\omega \times I\omega - M_G + M \quad (3)$$

$$M = \begin{pmatrix} \tau_\phi \\ \tau_\theta \\ \tau_\psi \end{pmatrix} = \begin{pmatrix} L \cdot K_T (w_1^2 - w_2^2 - w_3^2 + w_4^2) \\ L \cdot K_T (w_1^2 + w_2^2 - w_3^2 - w_4^2) \\ K_d (-w_1^2 + w_2^2 - w_3^2 + w_4^2) \end{pmatrix} \quad (4)$$

$$M_G = J_r \left(\omega \times \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right) \Omega_r \quad (5)$$

Where K_d is the drag factor and L is the distance of each rotor from the quadcopter center. The input variables of the real vehicle are the four rotational velocities (w_i) of the rotors; nevertheless, a transformation of the inputs is appropriate for the derived model. So, the control inputs are defined in (6).

$$\begin{aligned} U_1 &= K_T (w_1^2 + w_2^2 + w_3^2 + w_4^2) \\ U_2 &= K_T (w_1^2 - w_2^2 - w_3^2 + w_4^2) \\ U_3 &= K_T (-w_1^2 + w_2^2 + w_3^2 - w_4^2) \\ U_4 &= K_d (w_1^2 - w_2^2 + w_3^2 - w_4^2) \end{aligned} \quad (6)$$

Based on the equations, the overall dynamic model of quadcopter for both translational and rotational Equations are represented in (7). The mathematical modeling of quadcopter is presented in reference to [34] and the constant parameters for the quadcopter model are set accordingly in this work as shown in Table 1. To complete the quadcopter system, controllers are needed to handle the underactuated and nonlinearity properties of the quadcopter model. The controllers are designed in the next sub-section.

$$\begin{aligned}
\dot{x} &= \frac{1}{m}(S\theta C\phi C\psi + S\phi S\psi)U_1 & \ddot{\phi} &= \frac{1}{I_{xx}}[(I_{yy} - I_{zz})\dot{\theta}\dot{\psi} - J_r\dot{\theta}\Omega_r + lU_2] \\
\dot{y} &= \frac{1}{m}(S\theta\psi C\phi + S\phi S\psi)U_1 & \ddot{\theta} &= \frac{1}{I_{yy}}[(I_{zz} - I_{xx})\dot{\phi}\dot{\psi} + J_r\dot{\phi}\Omega_r + lU_3] \\
\dot{z} &= \frac{1}{m}(C\phi C\theta)U_1 - g & \ddot{\psi} &= \frac{1}{I_{zz}}[(I_{xx} - I_{yy})\dot{\theta}\dot{\phi} + U_4]
\end{aligned} \quad (7)$$

Note: $\Omega_r = (w_1 - w_2 + w_3 - w_4)$

Table 1. Parameters associated with quadcopter model

Parameter	Value
g	9.81 m.s ²
m	0.5 kg
l	0.2 m
$J_x = J_y$	4.85×10^{-3} kg.m ²
J_z	8.81×10^{-3} kg.m ²
J_r	3.36×10^{-5} kg.m ²
K_T	2.92×10^{-6} kg.m
K_d	1.12×10^{-7} kg.m ²

2.2. Quadcopter Controller Design

A quadcopter is naturally unstable without a controller. This section details the proposed controller and methodology being used in this work with the improvements made to produce a more robust result. This work implements a PID controller with the assistance of RBF networks to online-tune PID controller during flight. Fig. 3 represents the block diagram of overall control structure of quadcopter system in general.

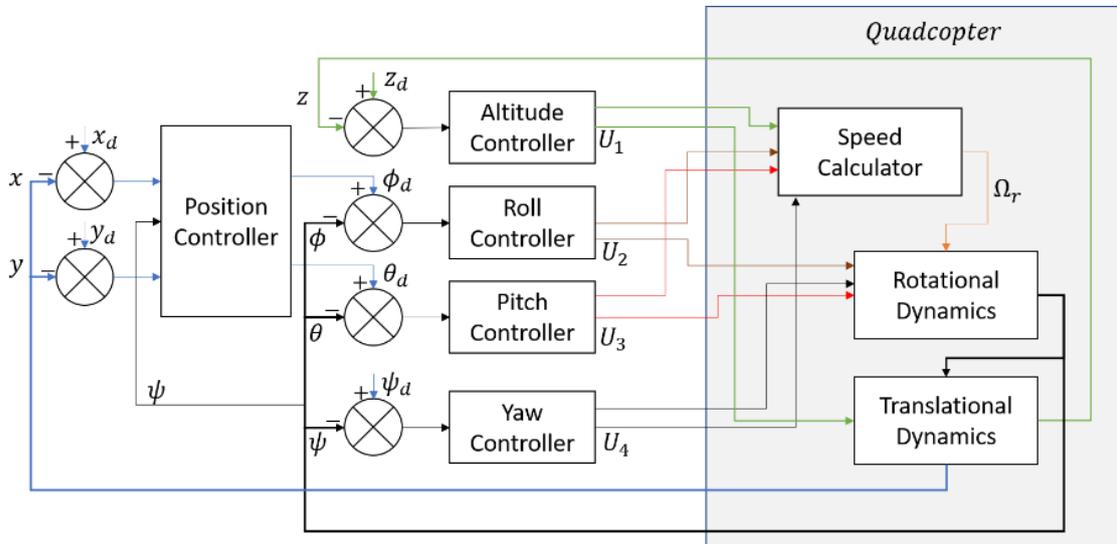


Fig. 3. Quadcopter system in MATLAB/Simulink

2.2.1. Altitude Control

Hovering is another way of saying altitude control. To ensure a stable hovering of quadcopter, the error taken from the flight output data must be corrected with the aid of altitude controller, to produce an adjusted control input, U_1 , as shown in Fig. 4, for the quadcopter to reach the desired altitude. During hovering, the nonlinear quadcopter system will become linear with condition of zero Euler angles are set. Other than altitude, U_1 also contributes to the control of X and Y positions. (refer to (7)).

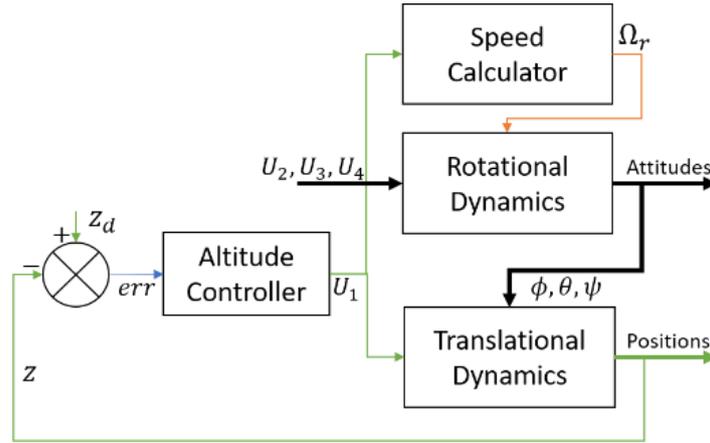


Fig. 4. Altitude controller for quadcopter

2.2.2. Attitude Control

This part controls the three Euler angles (ϕ, θ, ψ), which make the quadcopter to become nonlinear with the controllable rotational motions. The controller operation for each attitude is about the same as altitude controller where the errors from output flight data are required to be corrected with respect to the input reference. By controlling the attitudes, rolling, and pitching will translate the quadcopter position in y-direction and x-direction respectively, and yawing will change the heading of quadcopter. Roll, pitch, and yaw controller each produce an adjusted output of U_2, U_3 and U_4 . Fig. 5 visualized the attitude control structure in general.

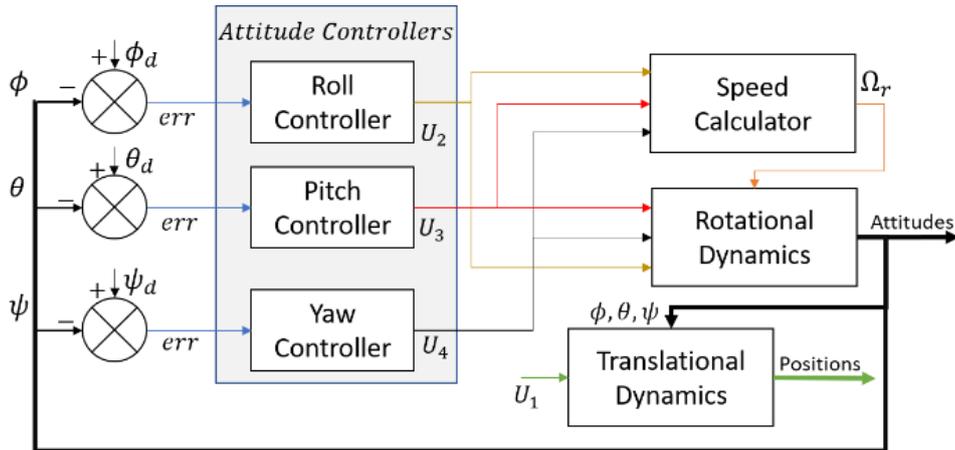


Fig. 5. Attitude controller for quadcopter

2.2.3. X-Y Position Control

The quadcopter’s translational motion in X and Y position is achieved through pitching and rolling of quadcopter. This is also called coupling dynamic and a good controller is needed. With the presence of position controllers, a cascaded controller structure in a quadcopter will naturally exist as shown in Fig. 6. Other than desired and actual position input, the position controller also receives input from the actual yaw angle. This is to translate from XY position error that is relative to the Earth’s frame, to the body frame so that the quadcopter can move to a certain place freely because pitch does not always move the body in the Earth’s X-direction, and roll does not always move the body in the Earth’s Y-direction. It will rely on how it is rotated or its yaw angle, and it must determine whether roll, pitch, or a combination of the two will be required.

$$\begin{aligned}
 x^B &= x^E \cos \psi + y^E \sin \psi \\
 y^B &= x^E \sin \psi - y^E \cos \psi
 \end{aligned}
 \tag{8}$$

2.2.4. PID Control

PID controllers are being used to control each quadcopter's motion, both rotational and translational, and the total is six. The input command is given directly to the three position controllers (X, Y, Z) and yaw controller while the input command for pitch and roll angles came from the position controllers. A PID controller takes in the error (difference between reference and actual output) and readjusts the control input for the quadcopter to follow the command successfully. When comparing the desired and actual values, the proportional term only multiplies the error by a constant. The steady state error is eliminated by the integral term, which integrates the error value over time until it approaches zero. The derivative term eliminates and estimates future errors. It multiplies the error estimate by the derivative constant and measures the error's rate of change over time [35]. The idea of improving the PID controller in this work will be detailed in the next sub-section.

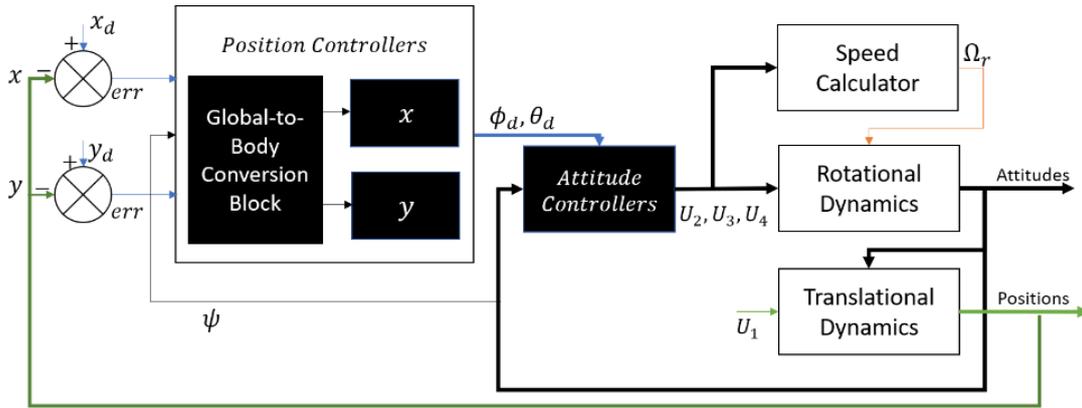


Fig. 6. Position controller of quadcopter

$$U(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_D \frac{de(t)}{dt} \quad (9)$$

Note: $e(t) = r(t) - y(t)$

While for the special case of coupling dynamics, the resulting PID controller of position controllers are as follows:

$$\begin{aligned} \phi_d &= K_p e_{y_b}(t) + K_i \int_0^t e_{y_b}(t) dt + K_d \frac{de_{y_b}(t)}{dt} \\ \theta_d &= K_p e_{x_b}(t) + K_i \int_0^t e_{x_b}(t) dt + K_d \frac{de_{x_b}(t)}{dt} \end{aligned} \quad (10)$$

2.2.5. Radial Basis Function Network

As mentioned before, Radial Basis Function Network is a three-layer-fixed structure network namely input layer, hidden layer, and output layer, and it is chosen in this work for its simple design and adaptivity with PID controller. Each neuron in the input layer acts like a special tunnel for each element in the input vector and directs the input linearly to each of the neurons in the hidden layer. The input vector is as shown in (11).

$$X = [x_1, x_2, \dots, x_i] \quad (11)$$

The RBF network structure is generally portrayed as in Fig. 7. In this work, the weights that are associated with one neuron of the hidden layer are the same and different from the other neuron, $W = [w_1, w_2, \dots, w_j]$. However, there are no weights associated between the input and hidden layer

as they are linearly connected. The output layer is notated as $O(t)$ which directs the network output to the next line in RBF-PID algorithm.

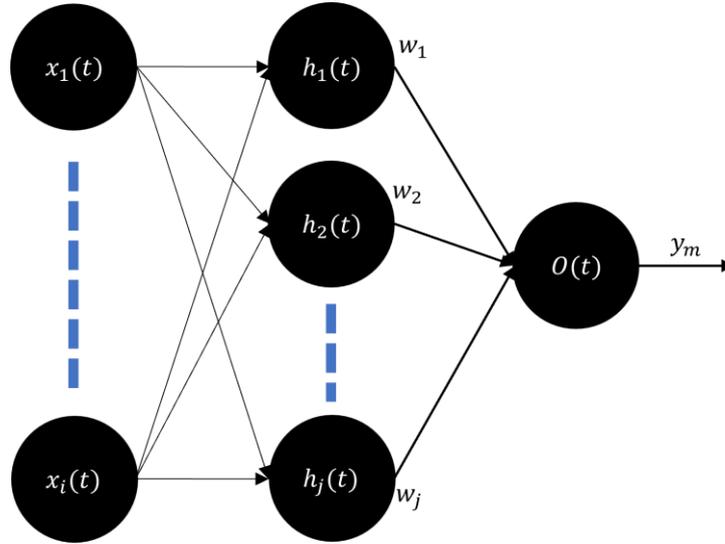


Fig. 7. RBF network with customizable neurons in hidden layer

In RBF network, there are weights, centers and widths that need to be initialized before running the network iteratively. The word ‘Radial Basis’ shows the distance value between the input vector and the center, $\varphi = \|X - C_j\|$, and the width of the hidden units, that if the distance is zero, the output will be 1 and if the distance increases, the output will converge to zero. The distance is deemed as the independent variable of the function [36]. This property greatly helps in classification and pattern recognition. All equations are in reference to [25]. The vector of hidden layer and radial basis vector that is based on Gaussian Function is shown:

$$H = [h_1, h_2, \dots, h_j] \quad (12)$$

$$h_j = \exp\left(-\frac{\|X - C_j\|^2}{2b_j^2}\right) \quad (13)$$

$$C_j = [c_{j1}, c_{j2}, \dots, c_{ji}] \quad (14)$$

$$B = [b_1, b_2, \dots, b_j] \quad (15)$$

Where C_j and B are the central vector and width vector of hidden neurons respectively. The central vector of each hidden node is a parameter vector with the same dimension as the input vector, which answers the notation C_{ji} [37]. Both parameters must be initialized before running the system. At the second realm of the network, exist the weights that are modified iteratively to account for variations between desired and actual network characteristics.

$$W = [w_1, w_2, \dots, w_j] \quad (16)$$

At the output, the estimated results are as follows and from that, the performance of the network is evaluated:

$$y_m = [w_1 h_1, w_2 h_2, \dots, w_j h_j] \quad (17)$$

$$\Delta_{ym} = y(t) - y_m(t)$$

$$E_{RBF} = 1/2 [\Delta_{ym}]^2 \quad (18)$$

By using Gradient Descent Method (GDM), all fundamental parameters (C, B, W) can be essentially corrected through iterative calculations while running simulation [38], [39]. GDM learning method is largely being used in neural network research. GDM for weights calculation is as (19).

$$\begin{aligned}w_{12} &= w_j(t-1) - w_j(t-2) \\w_{23} &= w_j(t-2) - w_j(t-3)\end{aligned}\quad (19)$$

$$w_j(t) = w_j(t-1) + \eta(\Delta_{ym})h_j + \alpha(w_{12}) + \beta(w_{23})$$

GDM for widths vector calculation is as (20).

$$\begin{aligned}\Delta b_j &= (\Delta_{ym})w_j h_j \left(\frac{\|X - C_j\|^2}{b_j^3} \right) \\b_{12} &= b_j(t-1) - b_j(t-2) \\b_{23} &= b_j(t-2) - b_j(t-3)\end{aligned}\quad (20)$$

$$b_j(t) = b_j(t-1) + \eta\Delta b_j + \alpha(b_{12}) + \beta(b_{23})$$

GDM for central vector calculation is as (21).

$$\begin{aligned}\Delta c_{ji} &= (\Delta_{ym})w_j \left(\frac{X_j - C_{ji}}{b_j^2} \right) \\c_{12} &= c_{ji}(t-1) - c_{ji}(t-2) \\c_{23} &= c_{ji}(k-2) - c_{ji}(k-3)\end{aligned}\quad (21)$$

$$c_{ji}(t) = c_{ji}(t-1) + \eta\Delta c_{ji} + \alpha(c_{12}) + \beta(c_{23})$$

2.2.6. Self-Tuned PID Controller with RBF Network

A simple visualization of the whole working system is portrayed in Fig. 8, which is directly related to the equations. In this sub-section, an online controlled quadcopter is identified using an RBF neural network, which also modifies Jacobian information. The information collected is then used by the PID controller to self-tune the PID parameters and achieve effective control by allowing real-time adjustments to the control parameters [40].

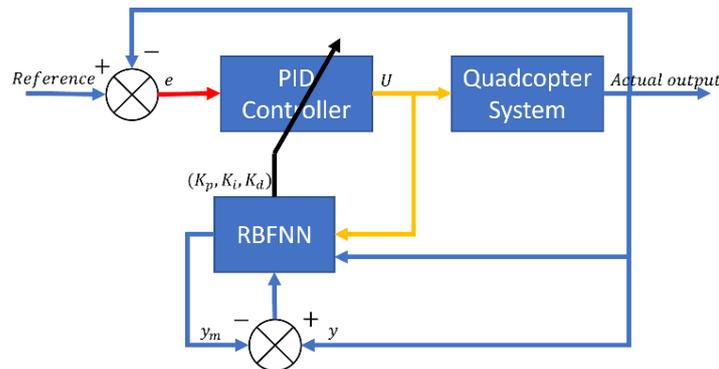


Fig. 8. Block diagram of quadcopter system with RBFPID controller

$$k_i(t+1) = k_i + \Delta k_i$$

$$\Delta k_i = -\eta \frac{\partial E}{\partial k_i} = -\eta \frac{\partial E}{\partial y} \frac{\partial y}{\partial u} \frac{\partial u}{\partial k_i} = -\eta e(t) \frac{\partial y}{\partial u} E_i \quad (28)$$

$$k_d(t+1) = k_d + \Delta k_d$$

$$\Delta k_d = -\eta \frac{\partial E}{\partial k_d} = -\eta \frac{\partial E}{\partial y} \frac{\partial y}{\partial u} \frac{\partial u}{\partial k_d} = -\eta e(t) \frac{\partial y}{\partial u} E_d \quad (29)$$

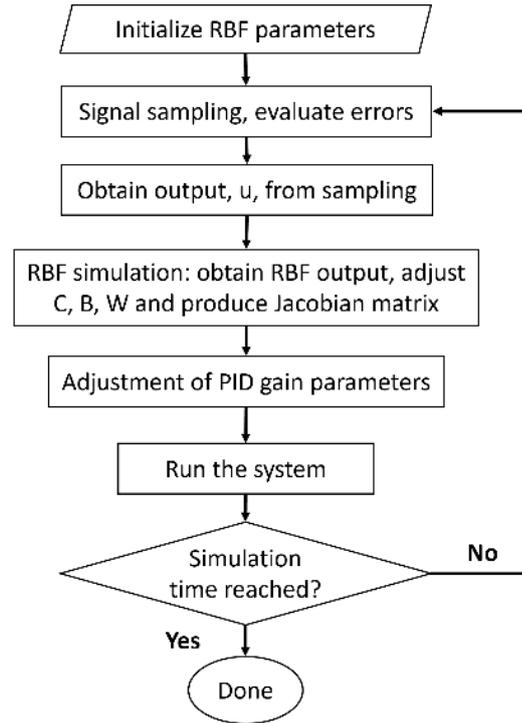


Fig. 10. The RBF algorithm

Based on the equations, Fig. 9, and Fig. 10, the step-by-step of how the RBF algorithm adaptively tunes PID controller in a quadcopter is laid out as:

- Step 1: Setting up the network's initial parameters, such as the base width vector, weight vector, learning rate, inertia coefficient, and number of nodes in the input and hidden layers.
- Step 2: Sampling of input, r and output, y to calculate error, $e(t)$ using (23)
- Step 3: Use (24) to calculate the controller's output, u .
- Step 4: To acquire network identification information, calculate the network output and modify the center vector, base width vector, weight vector, and the Jacobian matrix based on (17) to (22).
- Step 5: Modifying the PID parameters using (27) to (29).
- Step 6: Restart Step 2 and carry out the remaining stages until the simulation time runs out.

3. Results and Discussions

This work is simulated using MATLAB/Simulink environment. S-function is being used to nest the RBFPID algorithm inside Simulink environment. Several parameters are being initialized in the

algorithm using trial and error method. The parameters are sampling time, number of hidden neurons, initial PID gains, learning rates for each PID gain, RBF learning rate (η), momentum factors (α, β), initial weights, centers, and widths. In this work, inputs to the RBF network are $y(t), y(t-1)$ and $u(t-1)$ and the number of hidden neurons is initialized to six. By following the algorithm and methodology mentioned in previous section, results are produced as presented in the next sub-sections. Every result is compared to the traditionally tuned PID controllers using the PID Tuner App in Simulink. The traditional PID controller is tuned based on step input only. The initial PID gains of RBFPID and the tuned traditional PID controllers are maintained throughout this simulation and is shown in Table 2. The traditional PID controllers were tuned using the PID Tuner feature in Simulink. The quadcopter is an underactuated system with four input forces and six output states ($X, Y, Z, \phi, \theta, \gamma$). Therefore, the performance of each controller is being assessed in this section, whether they can tackle the downsides of quadcopter dynamics. Fig. 11 shows the quadcopter model built in MATLAB/Simulink.

Table 2. PID gains value for each controller

States	PID Tuner App	RBF-PID
Altitude (Z)	[9.2840, 4.3463, 4.8696]	[60,10,8]
Roll (ϕ)	[0.0546, 0.0063, 0.0752]	[40,2,3]
Pitch (θ)	[0.0508, 0.0084, 0.0756]	[18,4,1]
Yaw (γ)	[0.0198, 0.0023, 0.0273]	[30,3,2]

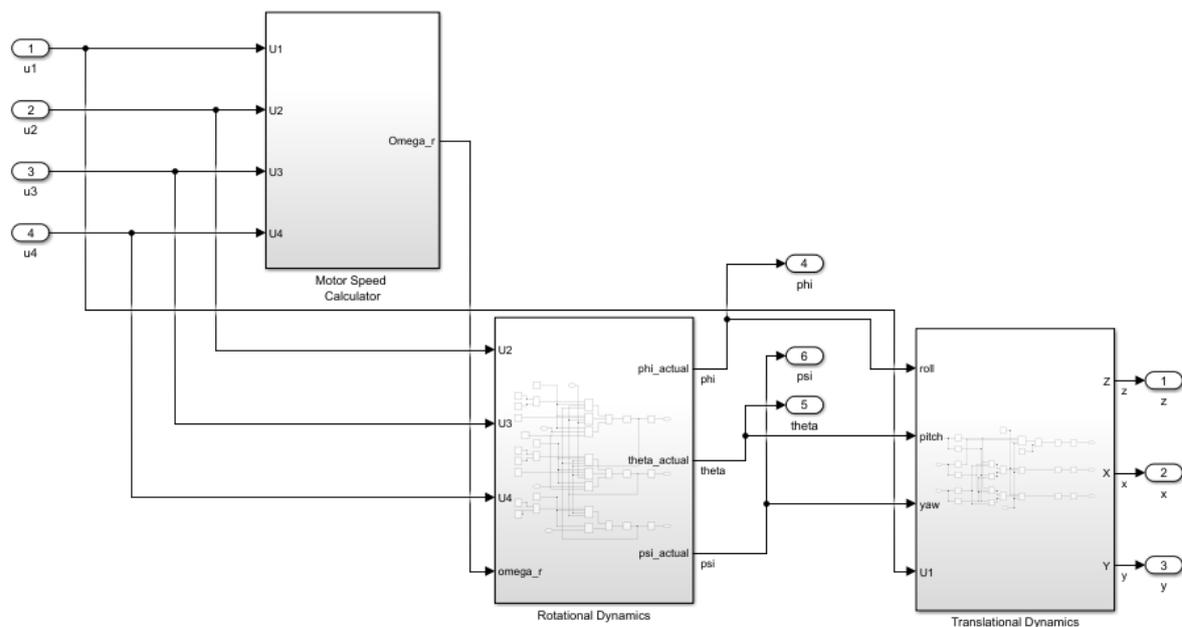
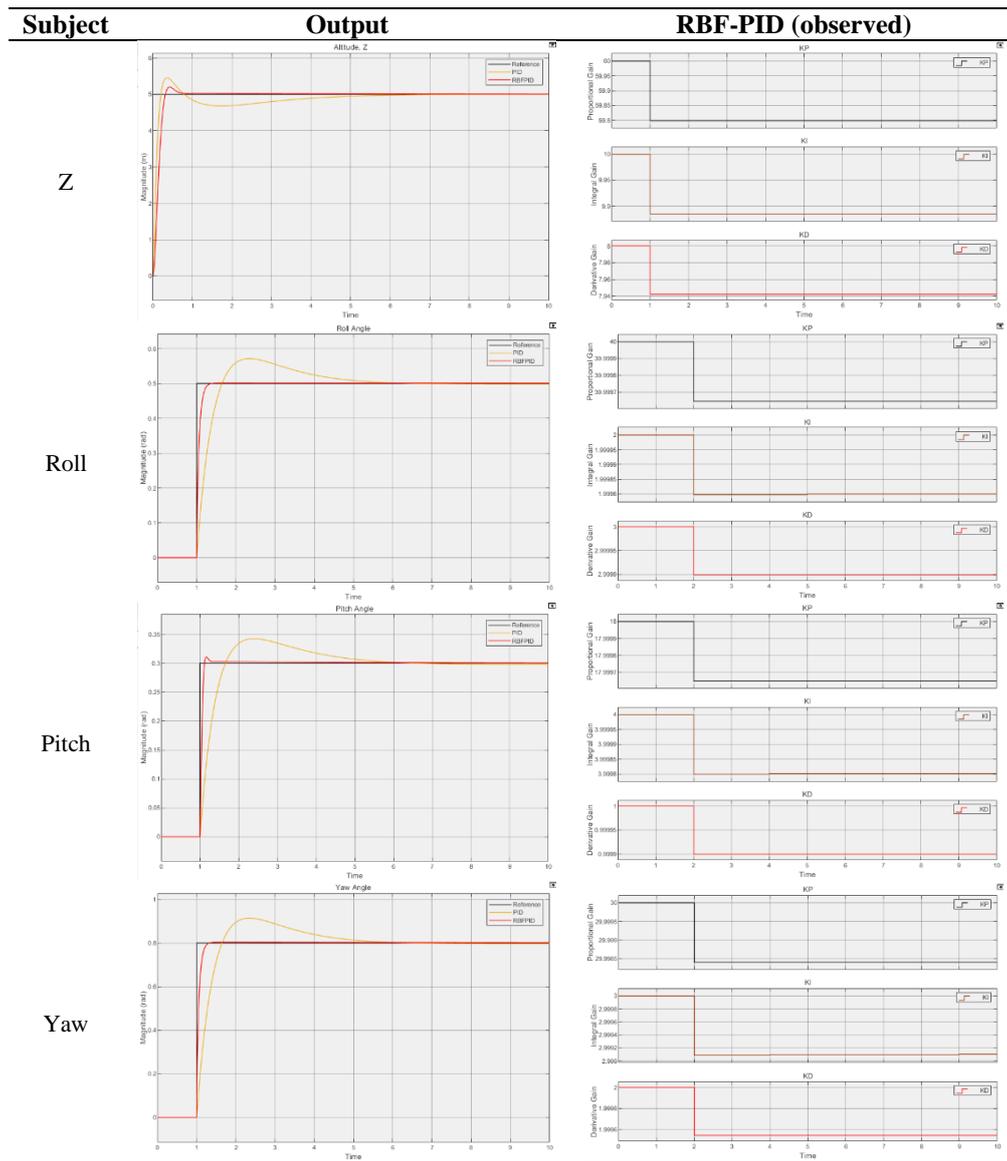


Fig. 11. Quadcopter model in MATLAB/Simulink

3.1. Altitude and Attitude Stabilization (Step Input)

Testing a quadcopter performance using step input is the most basic test and does not require much PID tuning. In several previous research papers, using a traditional PID tuning is sufficient for a quadcopter system with step input only. Table 3 listed the results for all quadcopter motions with step input. It was observed that there are small changes in PID values that occur a second later from the step time of each motion. When compared to traditional PID, RBFPID results produce less-to-none overshoot and the transient responses are well behaved. Step input simulation shows RBF-PID produces settling time less than 1 second compared to PID tuner app with around 5 seconds. The PID adaptation based on (27) until (29) is also shown in the table with respect to each state controller. The step input given to each state is $Z = 5$ meters, roll = 0.5 radian, pitch = 0.3 radian, and yaw = 0.8 radian.

Table 3. Quadcopter's motion performance from step input



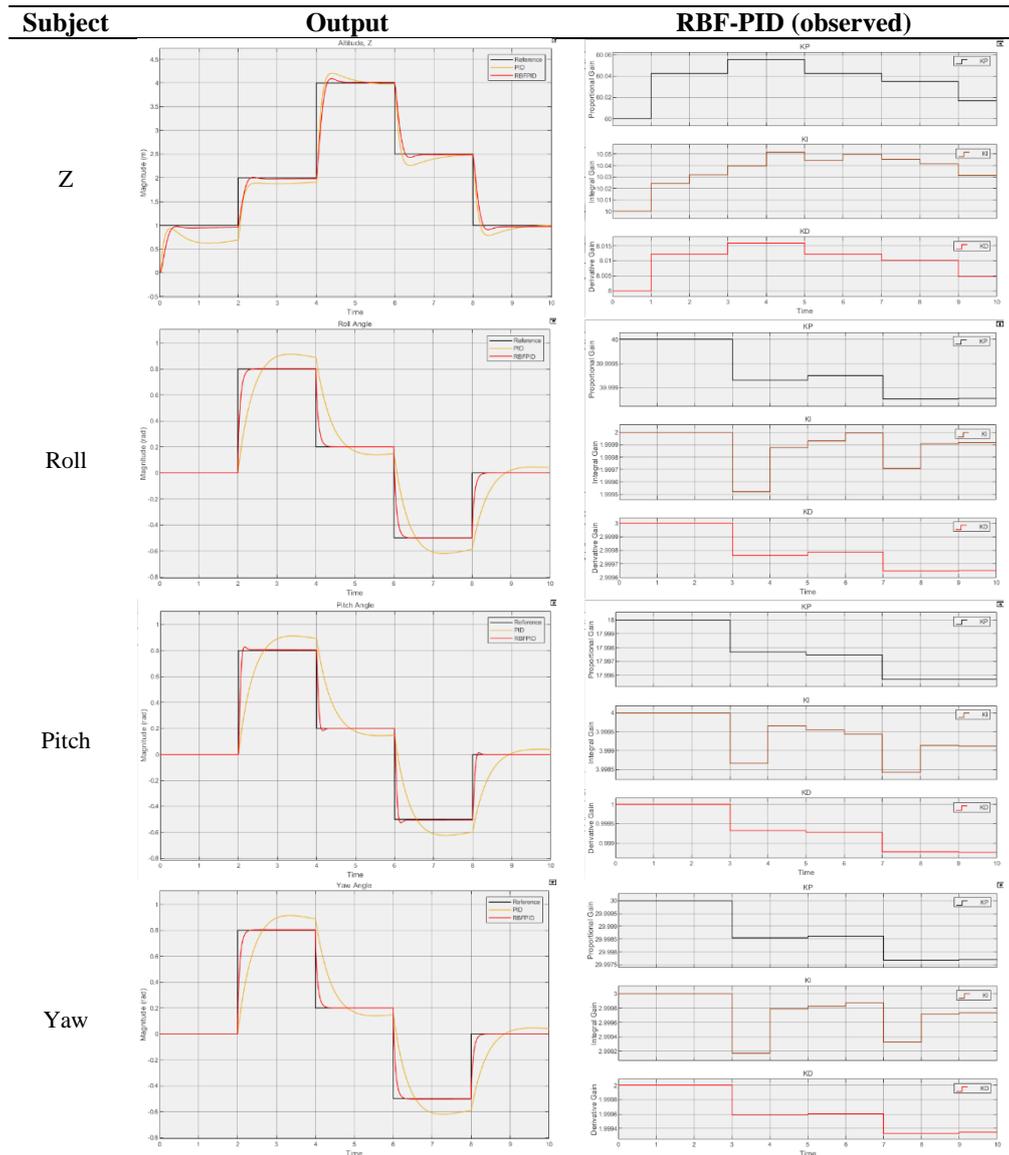
3.2. Altitude and Attitude Stabilization (Multistep Input)

Motions test by using multistep input is also important to evaluate the robustness of the quadcopter during receiving different inputs for every two seconds. This test has proved the reliability of RBFPID controller to adapt to changes that helps to maneuver the quadcopter according to the received inputs, as shown in Table 4 with the PID gains changes throughout the simulation. The quadcopter has successfully followed the given multistep input for all states for both controllers but with RBF-PID performs better. The multistep input given to each state is $Z = (1, 2, 4, 2.5, 1)$ meter, roll = pitch = $(0, 0.8, 0.2, -0.5, 0)$ radian.

3.3. Altitude and Attitude Stabilization (Sine Wave Input)

Going to the next input, the sine wave input, which further tests the robustness of quadcopter motions using RBFPID controller. Based on Table 5, RBFPID controllers have successfully led the quadcopter movement to follow the input sine wave motions. However, traditional PID controller tuned using PID Tuner App can barely lead the system. The sine wave input with amplitude 2 and bias 2 is given to Z and sine wave input with amplitude 1 is given to roll, pitch and yaw.

Table 4. Quadcopter's motion performance from multistep input



It is observed from altitude and attitudes tracking simulations that both traditional PID and RBF-PID controllers can follow the input trajectory well with RBF-PID performs better during simulations. As another means of proving results, a numerical evaluation is being performed using transient response in Table 6 and Root Mean Square Error (RMSE) in Table 7. Since it indicates how distant a line is fitted from its real value, RMSE is a useful tool for estimating errors.

Table 5. Step input transient performance

Attributes	PID			RBF-PID		
	Rise Time (s)	Settling Time (s)	Overshoot (%)	Rise Time (s)	Settling Time (s)	Overshoot (%)
Z	0.1362	4.1489	9.0979	0.2055	0.5335	2.8766
Roll	0.4705	4.8808	14.3046	0.1387	1.2430	0.2885
Pitch	0.4758	5.1078	14.1126	0.0787	1.2263	3.6250
Yaw	0.4705	4.8808	14.3046	0.1289	1.2210	0.5102

The overall observation on the rise time and settling time of RBF-PID shows a shorter time taken when compared to PID controller alone. However, RBF-PID produces a slower rise time in

altitude to compensate for a lower overshoot during take-off. Based on RMSE evaluation, RBF-PID also produces an overall better result than traditional PID controller with lower RMSE values. Low RMSE values show that the model has more accurate predictions and matches the data well while higher values indicate greater mistakes and less accurate predictions.

Table 6. Quadcopter's motion performance from sine wave input

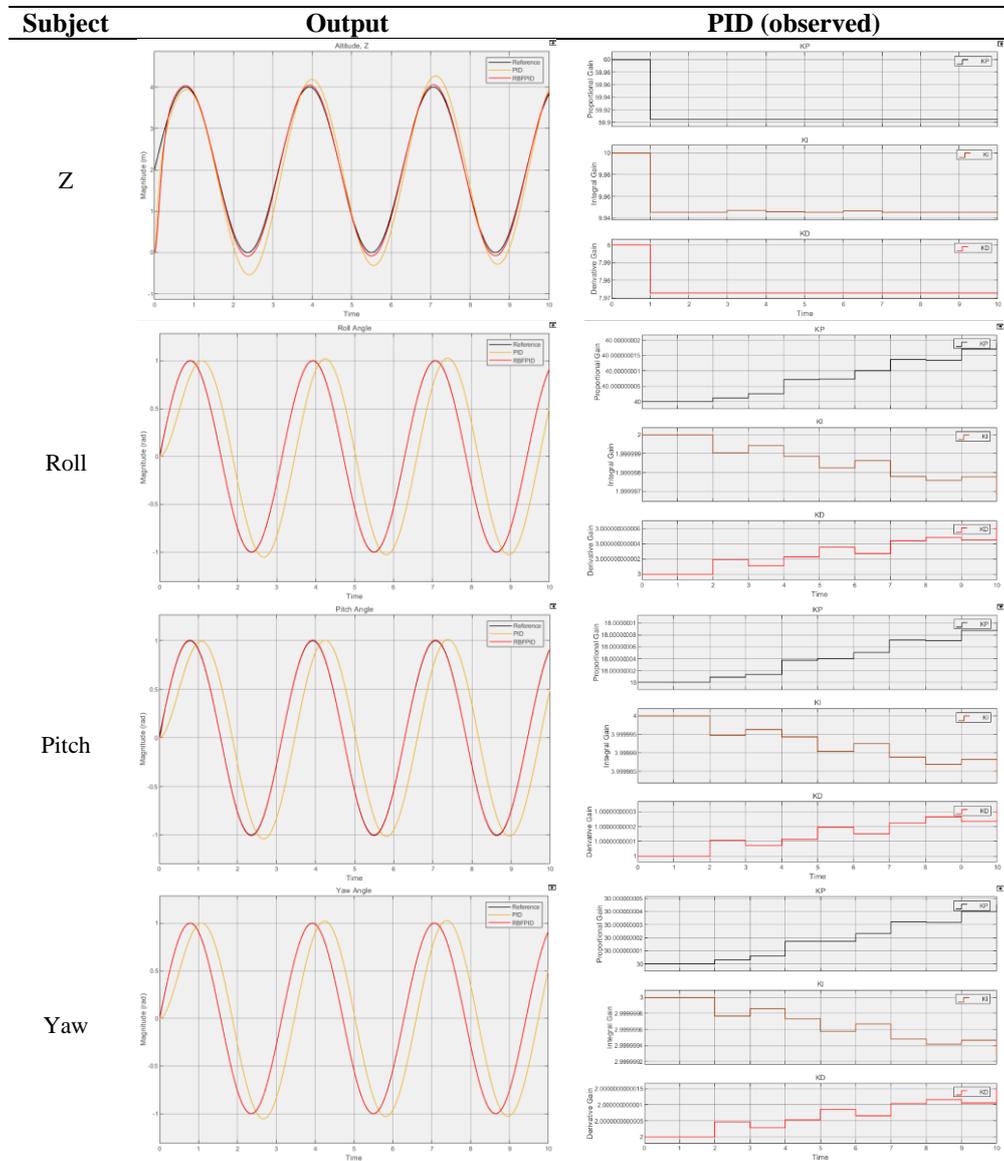


Table 7. Altitude and attitude tracking performance evaluation based on RMSE

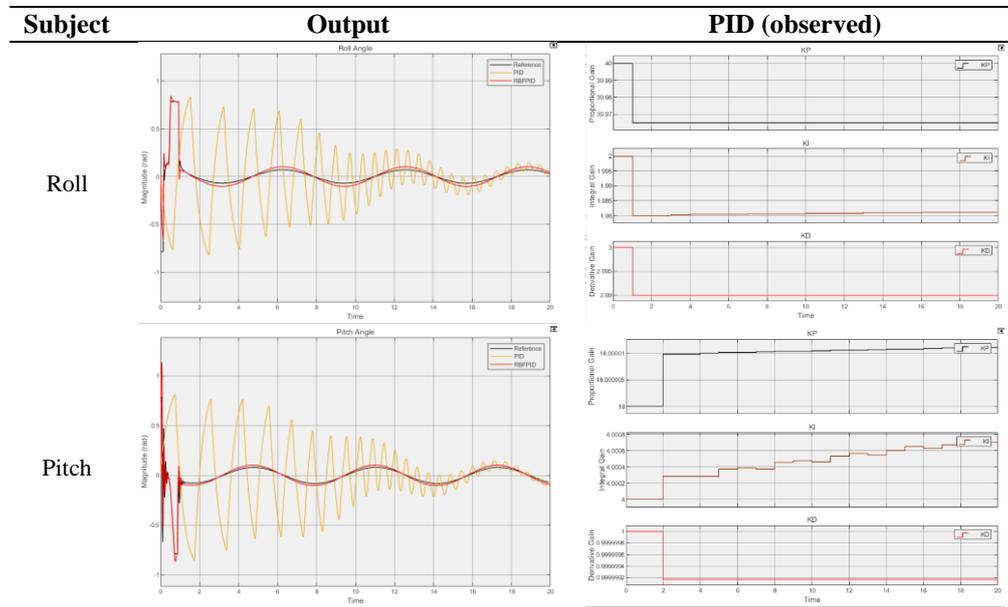
Attributes	PID			RBF-PID		
	Step	Multistep	Sine	Step	Multistep	Sine
Z	0.2850	0.2164	0.2990	0.3518	0.2020	0.1616
Roll	0.0460	0.1253	0.4375	0.0187	0.0494	0.0018
Pitch	0.0277	0.1264	0.4352	0.0116	0.0509	0.0042
Yaw	0.0736	0.1253	0.4375	0.0278	0.0460	0.0009

3.4. Position Tracking (Circular Motion)

Moving into position tracking test of quadcopter using circular trajectory. It can be seen in Table 8, the RBF-PID controllers for roll angle and pitch angle, again outshine traditionally tuned

PID controllers. The initial PID gains for RBFPID and the tuned PID gains are still maintained here. This shows that if by using traditionally tuned PID controllers with fixed gains, the system cannot survive in different situations, not to mention perturbations too. The circular trajectory is intended to be completed at an altitude of 1 meter and a radius of 0.5 meter, beginning at coordinate (0,0,0).

Table 8. Roll and pitch output for circular motion tracking



Sine waves are produced in the roll and pitch channels of a quadcopter during circular motion because the roll and pitch angles of the quadcopter vary sinusoidally with time as it moves around the circle. From Table 8, the X and Y position outputs produced are as shown in Fig. 12 (a)-(b) and the final circular pattern is as shown in Fig. 12 (c). The position controllers (X and Y) are manually tuned using traditional PID controller with gains [6,0,4].

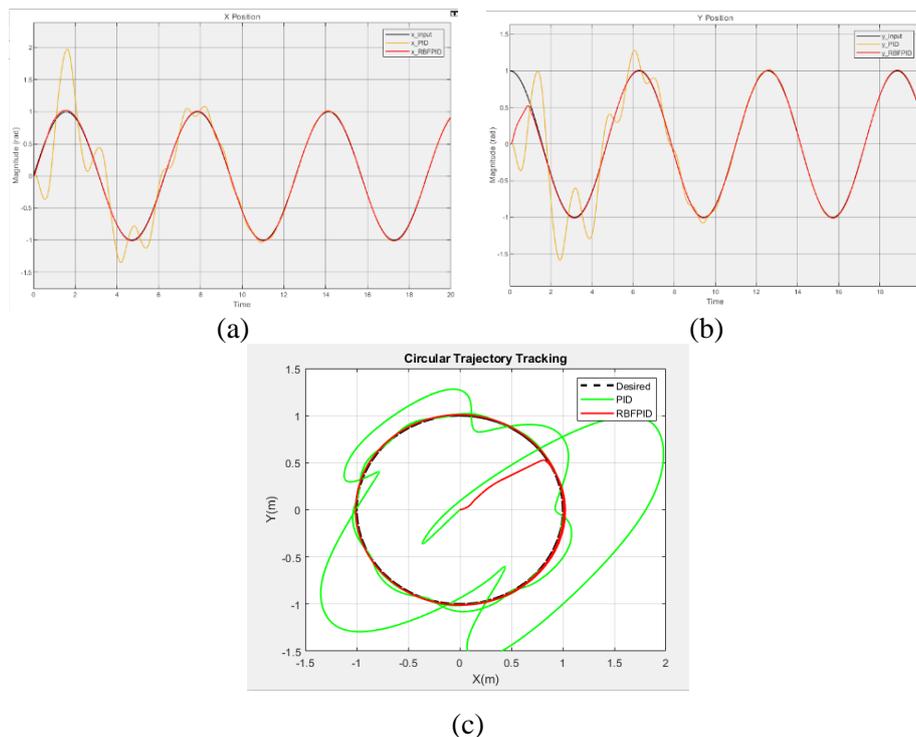
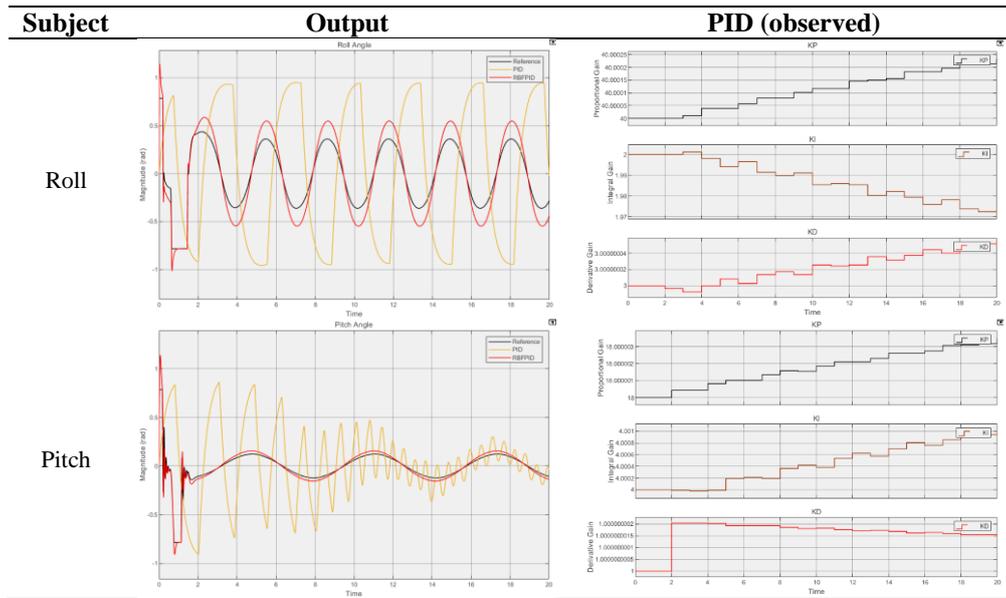


Fig. 12. X-Y position and circular output

3.5. Position Tracking (Lemniscate Motion)

For the tracking of lemniscate pattern, RBFPID performed nearly perfect for roll angle, unlike previous tests but the tracking is still in phase if compared to traditional PID controller, which totally out of phase. Table 9 shows the comparison of roll angle and pitch angle of both controllers and the changes that occurred in PID gains using RBFPID algorithm. Beginning at coordinate (0,0,0), the trajectory ascends to an altitude of 1 meter, utilizing a coordinate scale of x (-1.5 1.5) and y (-1.5 1.5).

Table 9. Roll and pitch output for lemniscate motion tracking



From Table 9, the X and Y position tracking results produced are as shown in Fig. 13 (a)-(b) and the final lemniscate pattern produced are as shown in Fig. 13 (c). The traditionally tuned PID controllers produced unsatisfying results for lemniscate since it cannot adapt well with the complex pattern using its fixed and poorly tuned PID parameters.

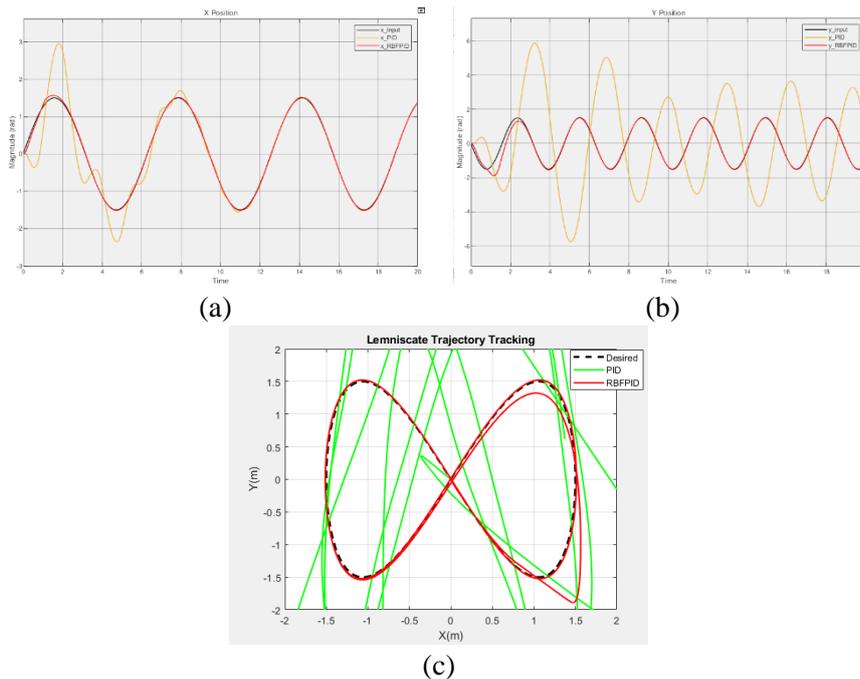


Fig. 13. X-Y position and lemniscate pattern output

From both patterns, it can be seen that RBF-PID controller can follow the trajectory excellently, but traditional PID can hardly follow the trajectory during simulation. As another means of proving results, a numerical evaluation is being performed using RMSE in Table 10. It is observed that again, RBF-PID produces an overall lower RMSE values for all states for trajectory tracking of both circular and lemniscate patterns. PID controller tuned with PID Tuner App cannot settle during trajectory tracking and produces steady-state error.

Table 10. Position tracking performance evaluation based on RMSE

Attributes	PID		RBF-PID	
	Circular	Lemniscate	Circular	Lemniscate
X	0.2538	0.4057	0.0638	0.0532
Y	0.3092	3.5874	0.1647	0.2727
Z	0.1731	0.2172	0.0783	0.0849
Roll	0.5522	0.5566	0.0446	0.1326
Pitch	0.5435	0.5569	0.0604	0.0703
Yaw	0	0	0	0

3.6. Disturbance Rejection in Altitude

The quadcopter system was subjected to external disturbances in order to assess the effectiveness of the control plans. Fig. 14 illustrates the application of two different forms of disturbances: sine wave [42] and wind gust [34] disturbances.

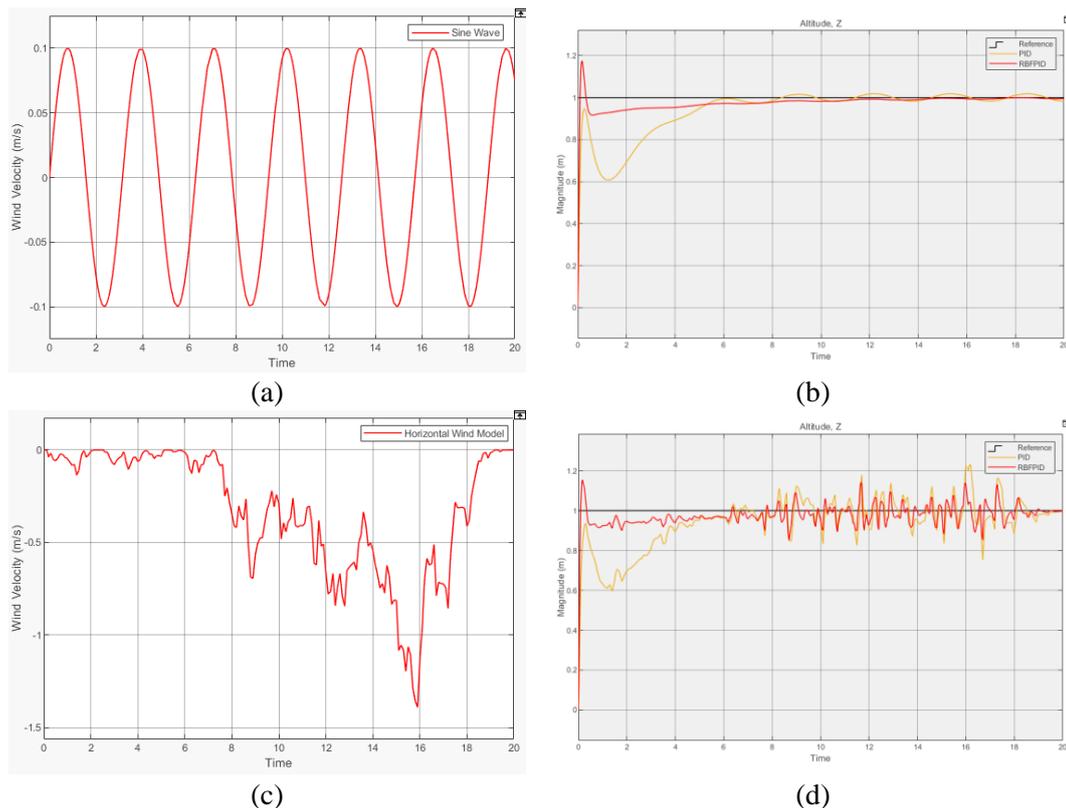


Fig. 14. Disturbances added with respective altitude responses

Fig. 14 (a) shows sine wave input with frequency and amplitude of 2 rad/s and 0.1 meter, respectively. Fig. 14 (b) shows the altitude response towards sine wave input and a better adaptation of RBF-PID compared to traditional PID is observed. Fig. 14 (c) shows the wind gust of 15 m/s speed produced from band-limited white noise that is turned into a wind gust model. Fig. 14 (d) graphically proves the altitude response towards wind gust using RBF-PID has better adaptation than PID. Table 11 tabulated the RMSE results of altitude during perturbations. RBF-PID controller

produces a 13.6% and 25.7% better results for both sine wave and wind gust disturbances respectively, than traditionally tuned PID controller. The overall results show a better PID adaptation from RBF-PID algorithm than traditional PID controller from both graphical and numerical presentations. By following the stated algorithm and related equations, PID gains have adaptively being tuned during simulations. In this work, different inputs and perturbations are included to further enhance the RBF-PID efficiency in comparison to [33] and the results are excellent.

Table 11. Altitude responses evaluation based on RMSE during perturbations

PID		RBF-PID	
Sine Wave	Wind Gust	Sine Wave	Wind Gust
0.3417	0.1786	0.2951	0.1327

4. Conclusion

This paper has presented the designed RBF network based self-adaptive PID Controller for quadcopter. Based on the methodology and results presented, it was clearly seen that the RBFPID controller can help much greater in developing a self-adaptive PID quadcopter control system than using a traditional PID controller that is fixed-tuned using PID Tuner App. A good performance can be achieved with the correct initialization of RBF parameters which are the sampling time, number of hidden neurons, initial PID gains, learning rates for each PID gain, RBF learning rate (η), momentum factors (α, β), initial weights, centers, and widths. The parameterization was done using trial and error method, which quite a time-consuming to determine the suitable ones. The sampling time was set to be 1 and the results are excellent. By using different trajectory and perturbations, this work contributes to adaptively simulate the flight of underactuated quadcopter using the introduced RBF-PID controller which was found helpful in regulating the quadcopter in different situations, with maintained initial parameters, unlike fixed-tuned PID controller. From altitude and attitude tracking simulations, both controllers seem to follow the trajectory very well with accepted RMSE values even though traditional PID controllers that are tuned using PID Tuner App suits a simpler and linear system. However, during position tracking, traditional PID controller can barely follow both circular and lemniscate patterns, leaving a higher RMSE values on each state compared to RBF-PID controller. This is because the traditional PID is tuned based on step input only. Also, RBF-PID performs better than PID controller when the perturbations are introduced with 13% to 25% improvements. However, RBF-PID cannot handle any frequency and amplitude of inputs in this research. Moreover, these excellent results can be further proved by using the same PID gains for initialization. For future work recommendations, an optimal RBF parameterization should be done using certain optimization techniques such as metaheuristic search algorithm. Moreover, by using a hardware quadcopter model such as Parrot Minidrone to further prove the efficiency of RBF-PID algorithm for it to experience more internal and external uncertainties will be impressive research in the future. Furthermore, an adaptive PID controller based on RBF algorithm will be a huge contribution in the field with extreme conditions that are unfavorable for humans to work in.

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

Acknowledgment: The authors would like to thank Universiti Teknologi Malaysia through UTMFR (Q.J130000.3823.22H67) for supporting this research.

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

References

- [1] B. Chamberlain and W. Sheikh, "Design and Implementation of a Quadcopter Drone Control System for Photography Applications," 2022 *Intermountain Engineering, Technology and Computing (IETC)*, pp. 1-7, 2022, <https://doi.org/10.1109/IETC54973.2022.9796735>.

-
- [2] S. A. Patil, P. R. Shinge, A. N. Kale, S. R. Antad, R. A. Hatgine, "Photography and Videography Drone," *International Research Journal of Modernization in Engineering Technology and Science*, vol. 5, no. 5, pp. 2416–2419, 2023, <https://www.doi.org/10.56726/IRJMETS34742>.
- [3] S. A. J. H. N., S. K. Da, "Quadcopter Unmanned Aerial Vehicle (UAV) for Mapping, Spraying and Crop Dusting," *International Research Journal of Engineering and Technology*, vol. 7, no. 12, pp. 427–435, 2020, <https://www.irjet.net/archives/V7/i12/IRJET-V7I1274.pdf>.
- [4] Y. Lee and J. Juang, "Color identification for quadcopter flight control and object inspection," *Advances in Mechanical Engineering*, vol. 11, no. 2, pp. 1–16, 2019, <https://doi.org/10.1177/1687814018822559>.
- [5] Shasyasyam, Shreeyansh, A. Jaishwal and V. M. Lakshe, "Weather Station Quadcopter Using Arduino with NRF24L01 and GPS Module," *International Research Journal of Engineering and Technology*, vol. 6, no. 3, pp. 4690–4691, 2019, <https://www.irjet.net/archives/V6/i3/IRJET-V6I31202.pdf>.
- [6] A. Sheta, M. Braik, D. R. Maddi, A. Mahdy, S. Aljahdali, and H. Turabieh, "Optimization of PID Controller to Stabilize Quadcopter Movements using Meta-Heuristic Search Algorithms," *Applied Science*, vol. 11, no. 14, p. 6492, 2021, <https://doi.org/10.3390/app11146492>.
- [7] S. Khatoun, M. Shahid, Ibraheem and H. Chaudhary, "Dynamic modeling and stabilization of quadrotor using PID controller," *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 746-750, 2014, <https://doi.org/10.1109/ICACCI.2014.6968383>.
- [8] A. Iyer and H. O. Bansal, "Modelling, Simulation, and Implementation of PID Controller on Quadrotors," *2021 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1-7, 2021, <https://doi.org/10.1109/ICCCI50826.2021.9402301>.
- [9] S. Abdelhay and A. Zakriti, "Modeling of a Quadcopter Trajectory Tracking System Using PID Controller," *Procedia Manufacturing*, vol. 32, pp. 564–571, 2019, <https://doi.org/10.1016/j.promfg.2019.02.253>.
- [10] A. Saibi, R. Boushaki, and H. Belaidi, "Backstepping Control of Drone †," *Engineering Proceedings*, vol. 14, no. 1, p. 4, 2022, <https://doi.org/10.3390/engproc2022014004>.
- [11] R. Roy, M. Islam, N. Sadman, M. A. P. Mahmud, K. D. Gupta, and M. M. Ahsan, "A Review on Comparative Remarks, Performance Evaluation and Improvement Strategies of Quadrotor Controllers," *Technologies*, vol. 9, no. 2, p. 37, 2021, <https://doi.org/10.3390/technologies9020037>.
- [12] H. E. Glida, L. Abdou, A. Chelihi, C. Sentouh, and S. E. I. Hasseni, "Optimal model-free backstepping control for a quadrotor helicopter," *Nonlinear Dynamics*, vol. 100, pp. 3449–3468, 2020, <https://doi.org/10.1007/s11071-020-05671-x>.
- [13] M. A. M. Basri and A. Noordin, "Optimal backstepping control of quadrotor uav using gravitational search optimization algorithm," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 5, pp. 1819–1826, 2020, <https://doi.org/10.11591/eei.v9i5.2159>.
- [14] A. Noordin, M. A. Mohd Basri, Z. Mohamed, and I. Mat Lazim, "Adaptive PID Controller Using Sliding Mode Control Approaches for Quadrotor UAV Attitude and Position Stabilization," *Arabian Journal for Science and Engineering*, vol. 46, pp. 963–981, 2021, <https://doi.org/10.1007/s13369-020-04742-w>.
- [15] X. Hu and J. Liu, "Research on UAV Balance Control Based on Expert-fuzzy Adaptive PID," *2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications (AEECA)*, pp. 787-789, 2020, <https://doi.org/10.1109/AEECA49918.2020.9213511>.
- [16] A. Noordin, M. A. Mohd Basri, and Z. Mohamed, "Adaptive PID Control via Sliding Mode for Position Tracking of Quadrotor MAV: Simulation and Real-Time Experiment Evaluation," *Aerospace*, vol. 10, no. 6, p. 512, 2023, <https://doi.org/10.3390/aerospace10060512>.
- [17] H. Ríos, R. Falcón, O. A. González and A. Dzul, "Continuous Sliding-Mode Control Strategies for Quadrotor Robust Tracking: Real-Time Application," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 2, pp. 1264-1272, 2019, <https://doi.org/10.1109/TIE.2018.2831191>.
- [18] A. Eltayeb, M. F. Rahmat, M. A. M. Basri, M. A. M. Eltoum and S. El-Ferik, "An Improved Design of an Adaptive Sliding Mode Controller for Chattering Attenuation and Trajectory Tracking of the Quadcopter UAV," *IEEE Access*, vol. 8, pp. 205968-205979, 2020,
-

- <https://doi.org/10.1109/ACCESS.2020.3037557>.
- [19] Z. Cai, S. Zhang and X. Jing, "Model Predictive Controller for Quadcopter Trajectory Tracking Based on Feedback Linearization," *IEEE Access*, vol. 9, pp. 162909-162918, 2021, <https://doi.org/10.1109/ACCESS.2021.3134009>.
- [20] A. Andriën, D. Kremers, D. Kooijman and D. Antunes, "Model Predictive Tracking Controller for Quadcopters with Setpoint Convergence Guarantees," *2020 American Control Conference (ACC)*, pp. 3205-3210, 2020, <https://doi.org/10.23919/ACC45564.2020.9147947>.
- [21] B. Li and Y. Wang, "An Enhanced Model Predictive Controller for Quadrotor Attitude Quick Adjustment with Input Constraints and Disturbances," *International Journal of Control, Automation and Systems*, vol. 20, pp. 648–659, 2022, <https://doi.org/10.1007/s12555-020-0815-9>.
- [22] J. A. Cárdenas, U. E. Carrero, E. C. Camacho, and J. M. Calderón, "Optimal PID ϕ 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>.
- [23] M. F. Q. Say, E. Sybingco, A. A. Bandala, R. R. P. Vicerra and A. Y. Chua, "A Genetic Algorithm Approach to PID Tuning of a Quadcopter UAV Model," *2021 IEEE/SICE International Symposium on System Integration (SII)*, pp. 675-678, 2021, <https://doi.org/10.1109/IEEECONF49454.2021.9382697>.
- [24] S. Bari, S. S. Zehra Hamdani, H. U. Khan, M. u. Rehman and H. Khan, "Artificial Neural Network Based Self-Tuned PID Controller for Flight Control of Quadcopter," *2019 International Conference on Engineering and Emerging Technologies (ICEET)*, pp. 1-5, 2019, <https://doi.org/10.1109/CEET1.2019.8711864>.
- [25] X. Du, J. Wang, V. Jegatheesan, and G. Shi, "Dissolved oxygen control in activated sludge process using a neural network-based adaptive PID algorithm," *Applied Sciences*, vol. 8, no. 2, p. 261, 2018, <https://doi.org/10.3390/app8020261>.
- [26] F. Jiang, F. Pourpanah and Q. Hao, "Design, Implementation, and Evaluation of a Neural-Network-Based Quadcopter UAV System," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 3, pp. 2076-2085, 2020, <https://doi.org/10.1109/TIE.2019.2905808>.
- [27] J. Gómez-Avila, C. López-Franco, A. Y. Alanis and N. Arana-Daniel, "Control of Quadrotor using a Neural Network based PID," *2018 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pp. 1-6, 2018, <https://doi.org/10.1109/LA-CCI.2018.8625222>.
- [28] O. Rodríguez-Abreo, J. Rodríguez-Reséndiz, C. Fuentes-Silva, R. Hernández-Alvarado and M. D. C. P. T. Falcón, "Self-Tuning Neural Network PID With Dynamic Response Control," *IEEE Access*, vol. 9, pp. 65206-65215, 2021, <https://doi.org/10.1109/ACCESS.2021.3075452>.
- [29] T. Huang, L. Shan, J. Li, Z. Yu and C. Liu, "Closed-Loop RBF-PID Control Method for Position and Attitude Control of Stewart Platform," *2019 Chinese Control Conference (CCC)*, pp. 3096-3101, 2019, <https://doi.org/10.23919/ChiCC.2019.8865860>.
- [30] Y. Dai, C. S. Yang, X. Huang and D. Xu, "RBF Neural Network Adaptive PID Control for Energy Storage System in Grid-Connected Photovoltaic Microgrid," *2018 5th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS)*, pp. 212-217, 2018, <https://doi.org/10.1109/ICCSS.2018.8572478>.
- [31] F. Che, Z. Wang and B. Wu, "Research on battery charging and discharging control system based on RBF-PID," *2022 IEEE 5th International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, pp. 648-651, 2022, <https://doi.org/10.1109/AUTEEE56487.2022.9994280>.
- [32] Q. Jiang, L. Zhu, C. Shu, V. Sekar, "An efficient multilayer RBF neural network and its application to regression problems," *Neural Computing and Applications*, vol. 34, pp. 4133–4150, 2022, <https://doi.org/10.1007/s00521-021-06373-0>.
- [33] S. Furukawa, S. Kondo, A. Takanishi and H. -o. Lim, "Radial basis function neural network based PID control for quad-rotor flying robot," *2017 17th International Conference on Control, Automation and Systems (ICCAS)*, pp. 580-584, 2017, <https://doi.org/10.23919/ICCAS.2017.8204300>.
- [34] N. H. Sahrir and M. A. Mohd Basri, "PSO – PID Controller for Quadcopter UAV: Index Performance Comparison," *Arabian Journal for Science and Engineering*, vol. 48, pp. 15241–15255, 2023, <https://doi.org/10.1007/s13369-023-08088-x3-08088-x>.

- [35] M. F. Shehzad, A. Bilal and H. Ahmad, "Position & Attitude Control of an Aerial Robot (Quadrotor) With Intelligent PID and State feedback LQR Controller: A Comparative Approach," *2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, pp. 340-346, 2019, <https://doi.org/10.1109/IBCAST.2019.8667170>.
- [36] S. Ma *et al.*, "RBF-Network-Based Predictive Ship Course Control," *2020 Chinese Control And Decision Conference (CCDC)*, pp. 3506-3511, 2020, <https://doi.org/10.1109/CCDC49329.2020.9164344>.
- [37] D. S. Soper, "Using an Opportunity Matrix to Select Centers for RBF Neural Networks," *Algorithms*, vol. 16, no. 10, p. 455, 2023, <https://doi.org/10.3390/a16100455>.
- [38] Y. Xie, X. Wang, H. Yu and R. Zhang, "Research on Adaptive Control of Down Pressure Based on RBF Neural Network," *2021 36th Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pp. 402-405, 2021, <https://doi.org/10.1109/YAC53711.2021.9486452>.
- [39] X. Meng-han and S. Wen-sheng, "RBF neural network PID trajectory tracking based on 6-PSS parallel robot," *2019 Chinese Automation Congress (CAC)*, pp. 5674-5678, 2019, <https://doi.org/10.1109/CAC48633.2019.8996255>.
- [40] W. Zhou, Y. He and S. Jin, "Research of feedforward +RBF neural network for permanent magnet linear Motor in XY motion platform," *2022 23rd International Conference on Electronic Packaging Technology (ICEPT)*, pp. 1-5, 2022, <https://doi.org/10.1109/ICEPT56209.2022.9873171>.
- [41] X. Shi, H. Zhao, and Z. Fan, "Parameter optimization of nonlinear PID controller using RBF neural network for continuous stirred tank reactor," *Measurement and Control*, vol. 56, no. 9-10, pp. 1835–1843, 2023, <https://doi.org/10.1177/00202940231189307>.
- [42] S. I. Abdelmaksoud, M. Mailah and A. M. Abdallah, "Improving Disturbance Rejection Capability for a Quadcopter UAV System Using Self-Regulating Fuzzy PID Controller," *2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE)*, pp. 1-6, 2021, <https://doi.org/10.1109/ICCCEEE49695.2021.9429661>.