

Article

# Design of Combined Neural Network and Fuzzy Logic Controller for Marine Rescue Drone Trajectory-Tracking

Duc-Anh Pham  and Seung-Hun Han \*

Department of Mechanical System Engineering, Gyeongsang National University, Tongyeong 53064, Korea  
\* Correspondence: shhan@gnu.ac.kr; Tel.: +82-55-772-9105

**Abstract:** In recent years, the research on drones has increased rapidly because of its high applicability in many fields and its great development potential. In the maritime sector too, especially marine rescue, a Drone with a compact size and fast flight speed is an effective solution in search and surveillance, giving quick results and being very convenient. When operating at sea, marine rescue drones are often affected by the environment, especially wind, which leads to turbulence that causes the drone to deviate from its predetermined flight trajectory. To overcome the above problem, the author has proposed the application of a Neural-Fuzzy controller for unmanned marine rescue aircraft presented in this paper introduces a controller that combines neural networks and fuzzy controllers that enhance the efficiency of the drone's trajectory tracking. The paper presents the mathematics of a quadcopter described by the Newton-Euler equations. Presentation on stable flight control and trajectory control of marine rescue drones. In this paper, Matlab/Simulink is used to describe the operation of the Drone, and the characteristics obtained after using the simulation are used to compare, test, and analyze the system. The obtained results show that the Neural-Fuzzy controller is much more sensitive, more resistant to turbulence, and can be used on different sizes, weights, and configurations of drones without adjusting PID gain.



**Citation:** Pham, D.-A.; Han, S.-H. Design of Combined Neural Network and Fuzzy Logic Controller for Marine Rescue Drone Trajectory-Tracking. *J. Mar. Sci. Eng.* **2022**, *10*, 1716. <https://doi.org/10.3390/jmse10111716>

Academic Editor: Mihalis Golias

Received: 5 October 2022

Accepted: 4 November 2022

Published: 10 November 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** orbital tracking; search and rescue system; artificial intelligence; marine rescue drones; neural network and fuzzy-PID controller

## 1. Introduction

A drone [1] (UAV—Unmanned Aerial Vehicle) is a type of flying vehicle or unmanned aircraft, aircraft with remote control. Drones can come in many sizes, shapes, and perform many different roles. Their main components include the engine, the central processing unit, the power, the propeller, and the racks. Drones can fly high and up to tens of kilometers. It can be said that the era of drones has come, and these drones are slowly evolving to become an integral part of everyday life. Against this backdrop, industries and governments are racing to find ways to use this emerging technology.

Their main structure includes (Figure 1): an engine, central processing system, power, propeller, and racks. Drones can fly high and up to tens of kilometers.

Drones are not simply flying devices, they are also a data collection and processing tool—drones are aerial robots [2]. They are very light, agile, and especially hundreds of times cheaper than bulky ground robots. Drones can easily overcome obstacles and change geographical positions flexibly. The size of drones varies widely, from the size of an adult to those only as big as the palm of your hand. The tasks that these drones can perform are also varied and not the same. Thanks to these advantages, drones have the potential to expand into a large-scale industry, providing multi-industry solutions. Drones can also integrate self-aware technology [3]—where computers process images and convert what they see into specific behaviors to automate more routine or difficult tasks. for human.



**Figure 1.** Structure of Drones.

A wide range of industries and businesses are looking to capitalize on the unique advantages of drones, particularly their data collection and processing capabilities. One of the main benefits of drones in business is that they are simply safer than using human labor. From a safety perspective, drones help businesses reduce the need to use helicopters or human workers to climb high-rise structures and conduct inspection operations. In addition, once a fault has occurred and is determined to be due to the drone, the repair will take less time and manpower. The quality of drone models will improve over time, and artificial intelligence (AI) [4] models will ensure fault detection in minimal time. It is also worth noting that using a drone is very environmentally friendly. The device consumes less energy and emits less than a traditional helicopter, which requires thousands of liters of fuel to operate.

The most mentioned application of drone technology is search and warning monitoring which helps improve safety in people's daily lives. The trend of using drones in this industry is also increasing. Drones are a perfect complement to traditional means of transport, especially in rural or remote areas that are hard to reach. Thanks to their unique advantages, drones are also gradually playing an important role in emergency situations. They help provide information about disaster situations, crime scenes, and many other factors before anyone reaches the scene. Observers say drones can help provide quick, effective responses to disaster and medical emergencies, including delivering supplies and equipment to victims in areas where there is no such thing as an emergency safe for people to move and approach. Drones can also be deployed before police arrive at the scene to provide initial information about the case, including live footage of the scene and images of the crime. In addition, drones can also help create or check maps [5], area topography, or support high-tech integrated agriculture.

Smart Drones packed with self-protection and command compliance, precision sensing, and self-monitoring technologies are the next big revolution in the field. With the momentum of modern technology development, the drone will be integrated with many other specialized automated features, ensuring the ability to shoot in the air and on the ground to provide closed inspection solutions. Automation, machine learning, and AI will improve the quality of data collection and verification, ensuring faster drone response and action in the future. With that development prospect, drones promise to provide new opportunities in the fields of transportation, logistics, and many other commercial activities. Experts believe that drones will play a prominent role in everyday life in the not-too-distant future, as their technology becomes more sophisticated and autonomous. The rational use of drones will bring clear positive social impacts such as helping to improve access to healthcare services and equipment in rural and minority communities. This will make this type of drone more receptive to the public, thereby paving the way for many other applications of drones.

Since the 20s, fuzzy set theory and artificial neural networks have developed rapidly and are of interest. With fuzzy logic, artificial intelligence has developed strongly, creating the basis for building expert systems, which are capable of providing system control experience. Artificial intelligence is built on top of artificial neural networks. The combination of fuzzy logic and neural networks in the design of automatic control systems is a completely new trend, the direction of intelligent control system design, a system where the controller has the ability to think like a controller. The human brain can self-learn and re-adjust itself to the unpredictable changes of the subject. As known, fuzzy systems and neural networks are both capable of working in unstable, inaccurate systems and extreme environmental conditions. Fuzzy systems and neural networks have many examples to evaluate and compare them. Nowadays, designers have widely and systematically applied fuzzy logic and neural networks in the field of cybernetics. The idea is to cancel out the disadvantages and gain the advantages of both technologies, this means that the two technologies combine to maximize the strengths of each and complement the disadvantages to form one. The new system is more optimized. This unified system will have the advantages of both: Neural network (learning ability, optimization ability, structural connectivity) and a fuzzy system (human intelligence through fuzzy if-then rule, the advantage of having a close understanding of the expertise of experts).

In fact, the development of drones in the last 10 years is a tremendous development. Presence in almost all areas of life from medical [6], agriculture [7], delivery [8], entertainment [9], military [10], etc. In the maritime sector is no exception, so drones at sea [11] are also an area that needs attention. Apart from that, in the first decade of the 21st century, machine learning was strongly researched and developed, marking an important turning point that changed the research foundation of Artificial Intelligence. Machine learning involves building computer programs that can automatically acquire knowledge, improve their abilities through experiences, and study the principles of the learning process [12]. The results and technologies of machine learning are demonstrated through diverse practical applications in areas such as natural language processing, computer vision, search and recognition, robotics, data mining, etc. In all, intelligent control is one of the outstanding topics that is highly effective, and the art of modern human-drone interaction [13] increases. So, the author wants to use a smart method to control the Drone. This study presents a Neural network for supervisory control—the Fuzzy control algorithm—one of the highly effective hybrid controllers [14–16] in recent years, making machine control become more efficient, more efficient and stable towards intelligent artificial intelligence capable of reasoning like humans. This controller can control mechanical devices such as robots, automatic cars, etc., and in this article, the problem is to stabilize altitude and track the trajectory [17–19] of drones using a Neural-Fuzzy hybrid controller. The operation of Marine Rescue Drones is often unstable due to many external factors, for example, wind is one of the main confounding factors to the Drone's flight trajectory, and the implementation of altitude control and Trajectory control set plays an important role. Therefore, the author proposes an algorithm to control the altitude and track the Marine Rescue Drone's trajectory using a Neural network- Fuzzy controller [20–24] that is to use a neural network controller in combination with a fuzzy controller to update the parameters of the drone dynamic PID controller (the coefficients K<sub>p</sub>, K<sub>i</sub>, K<sub>d</sub> are updated continuously thanks to the learning of the proposed controller). The learning of the Neural network and the inference of the fuzzy controller is a combination. It is more convenient when the set does not need expert intervention in the fuzzy controller, but it will be transferring data from the neural network and vice versa. The neural network control algorithm combining fuzzy control and PID control is simulated on Matlab/Simulink. The results obtained from both controllers obtained after conducting the simulation are compared with each other so that the reader can see the difference when applying these two controllers in Marine Rescue Drone.

The controller introduced by the author in this study has the following characteristics:

1. Brings high efficiency in adjusting the Drone to follow the set trajectory.
2. Suppress the effect of noise at a high level with high efficiency.

3. Reduce the operator's responsibility thanks to the high learning nature of the controller, making it easier to control.
4. High practicality and is a premise for the development of research on actual operation models.

## 2. Drone Model

### 2.1. Structure of Drone

A Drone is a device composed of parts: control motor, central processor, propeller, antenna, bracket, battery, and camera. Users can control the Drone by remote controller or pre-programmed according to GPS-based routes and coordinates.

The Drone controller has a similar structure to the remote controls of the model aircraft, including two buttons and a collapsible antenna. They usually use 2.4 GHz radio waves. However, there are also some controllers that combine both 2.4 GHz and Wi-Fi signals, which can also be based on a control app running on mobile devices.

Conventional Drones can fly thanks to the power from the battery supplied to the propellers. However, today there are expensive high-end models that use jet engines instead of traditional propellers, which can fly up to 800 km and up to 15 km high. Some types of drones work by pre-programmed operation, others are operated by humans using a remote control panel.

A Drone has a compact and lightweight design, which is also the reason why it can be carried easily and conveniently. In addition, it also helps Drones to fly longer in the air. A Drone also has outstanding features such as high mobility, convenience, and effective operation. Smaller and more efficient microprocessors, ever-improving wireless communications, and more diverse and multi-purpose Drones have opened up opportunities for Drones. thrive with more useful applications.

### 2.2. Dynamics Model of the Drone

To learn about the kinematic model of the drone Drone, we first determine the coordinate systems that will be used. In Figure 1, we have the Oxyz coordinate system, which is a global coordinate system (earth reference frame) fixed at a location on the ground.  $O_Bx_By_Bz_B$  is a local coordinate system (body reference frame) mounted at the center of the Drone with the  $x_B$  axis toward the 1st motor (Front motor), the  $y_B$  axis towards the 4th motor (Right motor), the  $z_B$  axis pointing down land.

A Drone usually consists of movements: up/down motion (attitude), motion around the  $x$ -axis-roll movement, motion around the  $y$ -axis-pitch movement, and motion around the  $z$ -axis-yaw movement (Figure 2). By varying the speed of the engines, it is possible to create the movements of the Drone.

In this study, a mathematical model of a Drone [17,19] is developed along with its dynamics. The basic equation of motion and force of the Drone is removed and the design parameters for a given Drone are selected. The model is made non-linear for the Drone based on the equation of motion [22,23] and the force of the moment using the Newton-Euler method:

$$\frac{d}{dt} \begin{bmatrix} P_n \\ P_s \\ h \end{bmatrix} = \begin{bmatrix} c_\Psi c_\theta & s_\Phi c_\Psi s_\theta - c_\Phi s_\Psi & c_\Phi c_\Psi s_\theta + s_\Phi s_\Psi \\ s_\Psi s_\theta & s_\Phi c_\Psi s_\theta + c_\Phi s_\Psi & c_\Phi c_\Psi s_\theta + s_\Phi s_\Psi \\ -s_\theta & c_\Phi s_\Psi & c_\Phi c_\theta \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} + \begin{bmatrix} -gs_\theta \\ gc_\theta s_\Phi \\ gc_\theta c_\Phi \end{bmatrix} + \frac{1}{m} \begin{bmatrix} 0 \\ 0 \\ -F_1 - F_2 - F_3 - F_4 \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \frac{J_y - J_z}{J_x} qr \\ \frac{J_z - J_x}{J_y} pr \\ \frac{J_x - J_y}{J_z} pq \end{bmatrix} + \begin{bmatrix} \frac{1}{J_x} \zeta_\Phi \\ \frac{1}{J_y} \zeta_\theta \\ \frac{1}{J_z} \zeta_\Psi \end{bmatrix} \quad (3)$$

where:

$P_n, P_s, h$ —position at the North pole, South pole, the attitude of the Drone in the earth coordinate system;

$u, v, w$ —are the velocity of the Drone toward the  $x, y, z$  axis, respectively;

$\Phi, \theta, \Psi$ —are the rotation around the  $x$ -axis (Roll angle),  $y$ -axis (Pitch angle),  $z$ -axis (Yaw angle);

$p, q, r$ —are the rotational velocities around the  $x, y$ , and  $z$  axis, respectively;

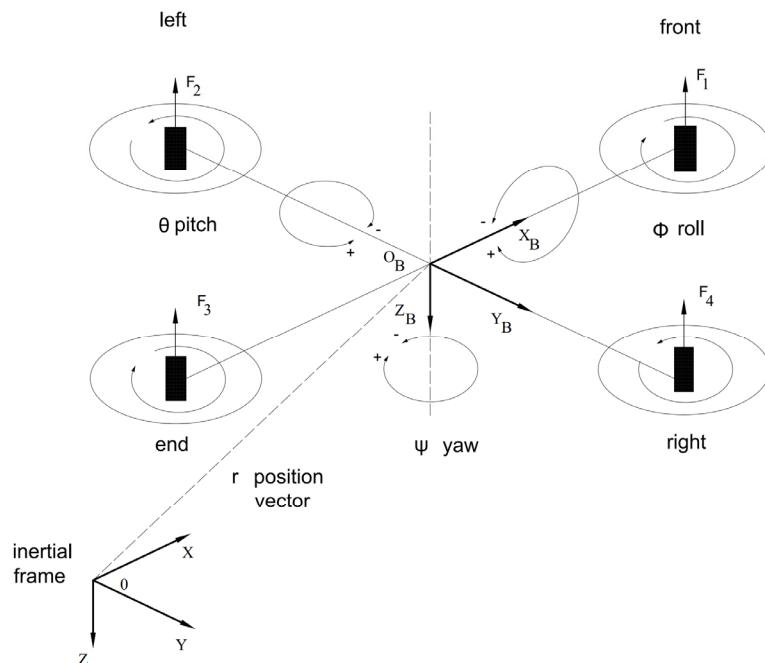
$g$ —acceleration due to gravity;

$c$ —cosine;  $s$ —sin;

$F_1, F_2, F_3, F_4$ —Corresponding thrust of 4 propellers;

$J_x, J_y, J_z$ —are the moments of inertia of the drone;

$m$ —mass of Drone.



**Figure 2.** Configuration and coordinate systems of Quadcopter.

### 2.3. Overview Neural Network–Fuzzy Logic Controller

First of all, we immediately see that both fuzzy controllers and neural network controllers are, in principle, static, nonlinear controllers. They can be designed with a given system quality to arbitrary precision and work according to the principles of human thought.

The performance of a neural network is determined by the type of neurons used and the network structure that connects those neurons together. It is completely independent of the object being controlled. Even if the designers have knowledge and understanding of the object, it will not help in the selection of neurons and the construction of the network structure. In contrast, for the fuzzy controller designer, knowledge and understanding of the object is very necessary. Although these insights are not necessarily represented in the form of a mathematical model describing the object, but can only be experiences gained in the process of approaching the object, without them, the designer is not possible to build a composition rule, can not blur the signal value (determine the membership function). The reality of fuzzy controller design shows that designers are always most confused when building functions belonging to language value descriptions. The reason is that while the association rule is built on the basis of only needing to have a qualitative understanding of the object (which is a fundamental advantage of the fuzzy control principle), the determination of the membership function for each linguistic value is not sufficient. Again, it

requires detailed quantitative knowledge (although it is limited to the input and output signal value of the object), which is the key point to distinguish fuzzy control from classical control, and also, the basic point of the neural network.

Even when it was first designed, neural networks had no knowledge. Its knowledge is formed through the learning phase through learning patterns. The better the learning sample, the more diverse, and the more cases, the closer the initial knowledge of the network is to reality. If that is not enough, the knowledge of the network can still be added, and further perfected in the process of working with the object. With the fuzzy controller, the opposite is true. When the design is complete, the fuzzy controller has a certain working mechanism immediately and this mechanism will not change and be kept fixed throughout the working period. In other words, the neural network is capable of learning and the fuzzy controller is not.

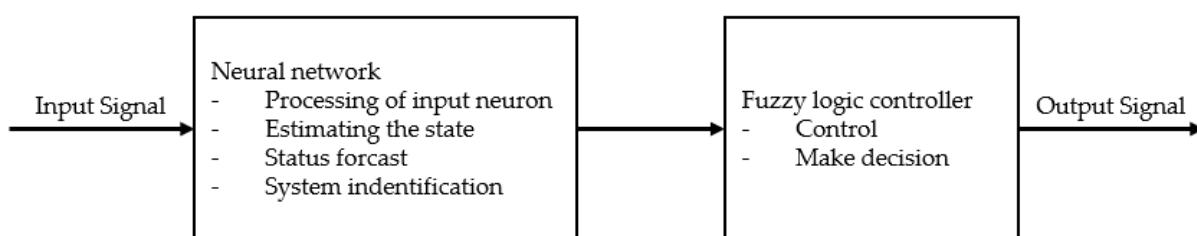
The neural network is still considered an application solution in which the designer is "blind" to the object (black-box solution). The knowledge of the network lies in the weights that are evenly distributed throughout the network. A small change in the weights is not enough to change the performance of the neural network, so it is difficult to evaluate the existing knowledge of the network without a comparison sample. Fuzzy controllers are different, a small change in the membership function, or composition rule, will immediately lead to a corresponding change that is very noticeable in its nature. Therefore, the fuzzy controller has higher real-time adaptability than the neural network.

Thus, it is clear that the advantages of neural networks are disadvantages of fuzzy controllers and vice versa. The following table (Table 1) summarizes the advantages and disadvantages of the two controllers.

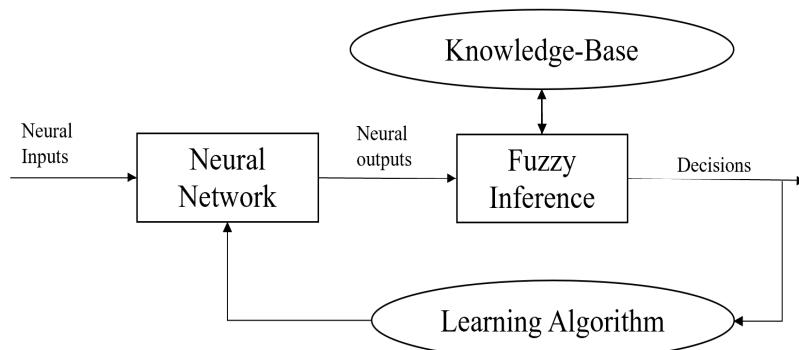
**Table 1.** Advantages and disadvantages of neural networks and fuzzy control.

Nature	Neural Network	Fuzzy Logic Control
Show knowledge	Through weights expressed hidden in the network	Expressed right in the law of composition
Source of knowledge	From learning patterns	From the experience of experts
Handling uncertain information	Quantitative	Quantitative and qualitative
Knowledge retention	In neurons and the weights of each neuron junction	In the law of composition and membership functions
Ability to update and improve knowledge	Through the learning process	Not available
Sensitivity to model changes	Low	High

To explain the task of the Neural Network and Fuzzy Logic Control in the proposed Neural-Fuzzy control system in Figures 3 and 4:



**Figure 3.** Model architecture of a neural-fuzzy system.



**Figure 4.** The block diagram of the Neural-Fuzzy system.

In Figures 3 and 4, it is easy to see a neural fuzzy system is characterized by the use of neural networks to provide fuzzy systems with a kind of automatic tuning method, but without altering their functionality. In this study, connecting the fuzzy controller and neural network is a form of serial coupling. This form of coupling is very suitable for problems where the neural network is responsible for processing input and output signals for fuzzy controllers.

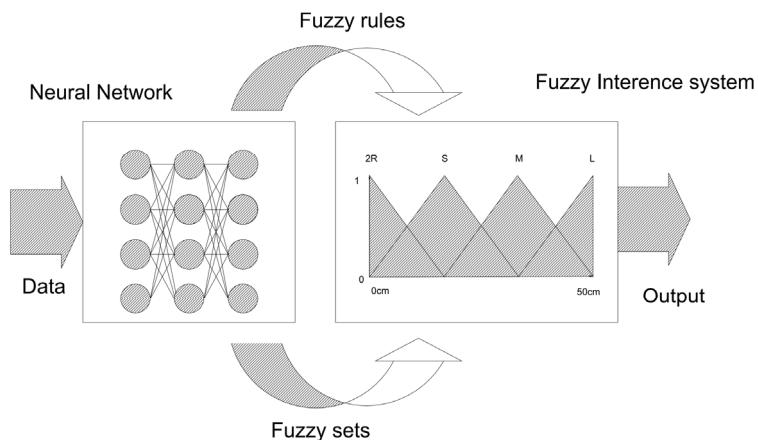
Neural networks have a major drawback of their “black box” nature. Data is fed into the network, computation occurs, and results are output. However, we don’t know exactly what happened.

Fuzzy logic is the opposite, their rule sets are very close to human knowledge. However, fuzzy logic does not have the self-learning ability of neural networks. Fuzzy rules and membership functions can only be adjusted manually.

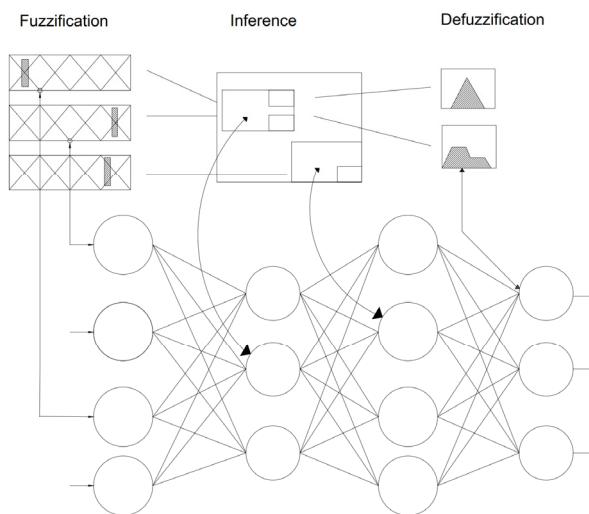
Therefore, the combination of neural networks and fuzzy control has invisibly eliminated their shortcomings, creating a neural-fuzzy controller with outstanding advantages.

The following table shows the strengths and weaknesses of Neural networks and Fuzzy Logic Control. For the reasons mentioned above, the combination of these two control methods is a perfect complement to each other. The new hybrid controller is more efficient and intelligent.

The fuzzy neural system uses the learning algorithm of a multi-layer neural network to adjust the parameters and structure of the fuzzy set (Figures 5 and 6). This means that the fuzzy set can construct itself (without needing specialized knowledge of the object it models) when given a rich data set. The finished structure can easily be “compiled” into a human-friendly legal language. This is an effective combination of soft computing technology.

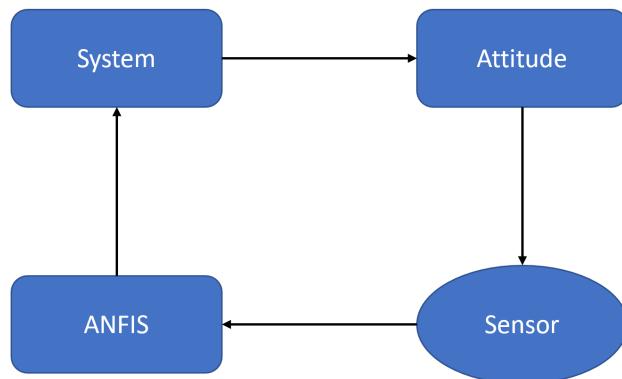


**Figure 5.** The block diagram describes the operation of the Neural-Fuzzy operating system.



**Figure 6.** The process of learning and processing information of the Neural-Fuzzy system.

The neural network simulates the processing of a fuzzy system in which the neurons of the first layer are responsible for the fuzzification process. The neurons of the second layer represent the fuzzy words used in the fuzzy rules (third layer). Finally, the neurons of the last layer are responsible for the defuzzification process. In this sense, the training of this neural network results in automatically adjusting the parameters of a fuzzy system and finding their optimal values. Figure 7 explains the connection of the ANFIS controller to the sensors and the system to control the rescue drone at sea.



**Figure 7.** Block diagram of the control system using the ANFIS integrated controller and observer of the marine rescue drone.

### 3. Drone Controller Scheme

#### 3.1. Position Controller and Altitude Controller of Marine Rescue Drone

- 6 degrees of freedom: X, Y, Z, roll, pitch, yaw
- 4 control inputs:  $U_1$  for thrust,  $U_2, U_3, U_4$  for torque on  $x_B, y_B, z_B$  axis.

$\Omega_1, \Omega_2, \Omega_3, \Omega_4$  : is the angular speed of the four motors.

g: Gravity acceleration.

b: Thrust coefficient.

The symbol for the sum of all external forces acting on the plane is:

$F = [F_x \ F_y \ F_z]$  with components  $F_x, F_y, F_z$  defined by the body coordinate system. According to Newton's 2nd law we have:

$$F = m\ddot{V} \quad (4)$$

With  $\mathbf{V} = (u, v, w)$  is the absolute velocity vector of the plane in terms of the body coordinate system.

The equations describing the horizontal motion of the Quadrotor:

$$\Delta \dot{y} = \Delta v \quad (5)$$

$$\Delta \dot{v} = g \Delta \phi \quad (6)$$

$$\Delta \dot{\phi} = \Delta p \quad (7)$$

$$\Delta \dot{p} = \frac{l}{I_{xx}} \Delta U_2 \quad (8)$$

Due to the suspended state, the roll torque  $U_{20} = 0$ , so we have:

$$U_2 = \Delta U_2 = b(\Omega_4^2 - \Omega_2^2) \quad (9)$$

$$\Omega_4 = \Omega_o + \Delta \Omega \quad (10)$$

$$\Omega_2 = \Omega_o - \Delta \Omega \quad (11)$$

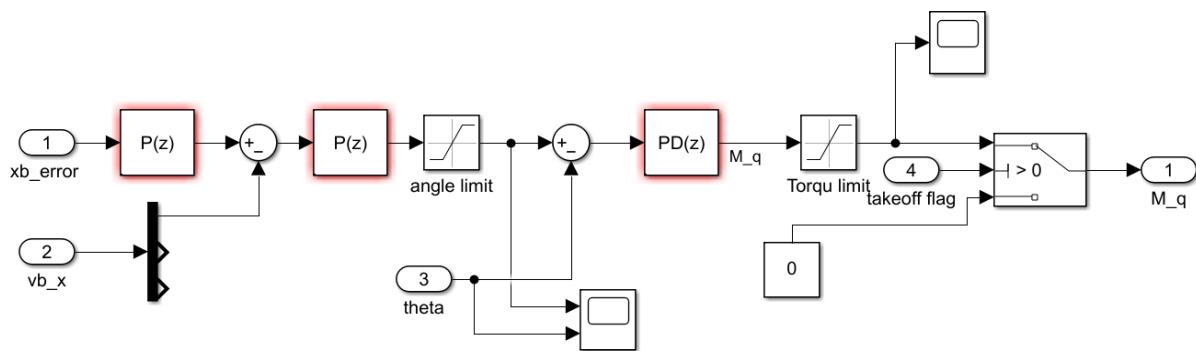
Deduce

$$\Delta U_2 = b[(\Omega_o + \Delta \Omega)^2 - (\Omega_o - \Delta \Omega)^2] = 4b\Omega_o\Delta \Omega \quad (12)$$

So, the state space model describing the horizontal motion of the plane is:

$$\begin{bmatrix} \Delta \dot{y} \\ \Delta \dot{v} \\ \Delta \dot{\phi} \\ \Delta \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta y \\ \Delta v \\ \Delta \phi \\ \Delta p \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{l}{I_{xx}} 4b\Omega_o \end{bmatrix} \Delta \Omega \quad (13)$$

From the above formula, the transfer function is obtained, thereby simulating the stable controller X shown in Figure 8.



**Figure 8.** The PID controller controls X (current pitch).

- The equations describing the vertical motion of the Quadrotor:

$$\Delta \dot{x} = \Delta u \quad (14)$$

$$\Delta \dot{u} = g \Delta \theta \quad (15)$$

$$\Delta \dot{\theta} = \Delta q \quad (16)$$

$$\Delta \dot{q} = \frac{l}{I_{yy}} \Delta U_3 \quad (17)$$

Due to suspended flight, pitch moment  $U_{30} = 0$ , we have:

$$U_3 = \Delta U_3 = b(\Omega_1^2 - \Omega_3^2) \quad (18)$$

$$\Omega_1 = \Omega_o + \Delta\Omega \quad (19)$$

$$\Omega_3 = \Omega_o - \Delta\Omega \quad (20)$$

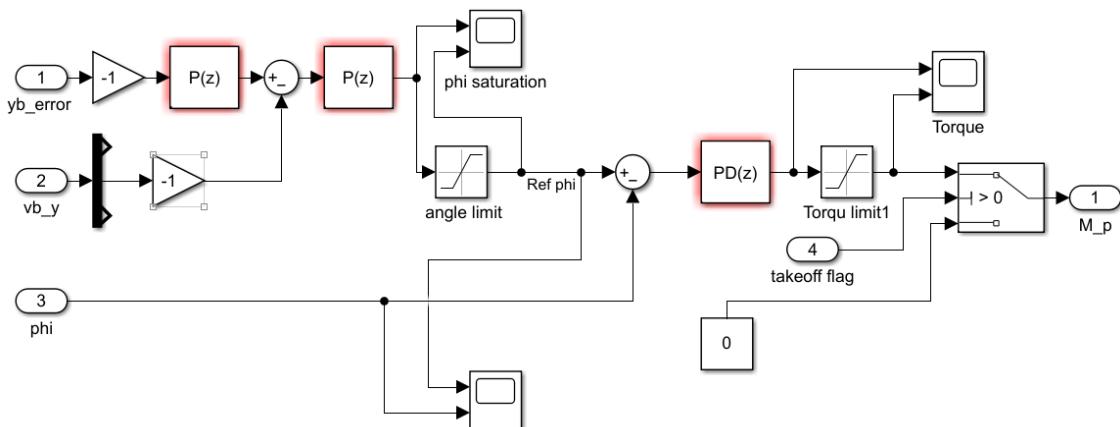
Deduce,

$$\Delta U_3 = b[(\Omega_o + \Delta\Omega)^2 - (\Omega_o - \Delta\Omega)^2] = 4b\Omega_o\Delta\Omega \quad (21)$$

So, the state space model describing the horizontal motion of the plane is:

$$\begin{bmatrix} \Delta\dot{x} \\ \Delta\dot{u} \\ \Delta\dot{\theta} \\ \Delta\dot{q} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & g & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta u \\ \Delta\theta \\ \Delta q \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{l}{J_{yy}}4b\Omega_o \end{bmatrix} \Delta\Omega \quad (22)$$

From the above formula, the transfer function is obtained, thereby simulating the stable controller Y shown in Figure 9.



**Figure 9.** The PID controller controls Y ( $\Phi$ ).

The equations describing the translational motion in the  $O_{ze}$  axis of the Quadrotor:

$$\Delta\dot{z} = \Delta w \quad (23)$$

$$\Delta\dot{w} = -\frac{\Delta U_1}{m} \quad (24)$$

We have:

$$U_1 = U_{10} + \Delta U_1 = b(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) = 4b(\Omega_o + \Delta\Omega)^2 = 4b\Omega_o^2 + 2\Omega_o\Delta\Omega + \Delta\Omega^2 \quad (25)$$

According to small noise theory, we can ignore higher order small noise quantities larger than 1:

$$\Rightarrow \Delta\Omega^2 \approx 0 \quad (26)$$

$$\Rightarrow U_1 = U_{10} + \Delta U_1 = 4b\Omega_o^2 + 2\Omega_o\Delta\Omega \quad (27)$$

Since the thrust in the suspended state of the engines is:

$$U_{10} = 4b\Omega_o^2 \quad (28)$$

$$\Rightarrow \Delta U_1 = 2\Omega_o\Delta\Omega \quad (29)$$

So, the state space model describing the horizontal motion of the plane is:

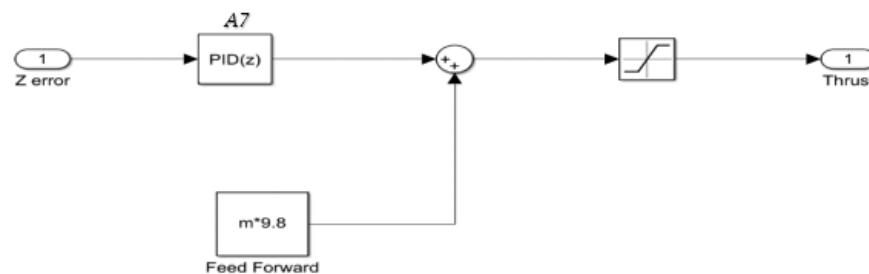
$$\begin{bmatrix} \Delta \dot{z} \\ \Delta \dot{w} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta z \\ \Delta w \end{bmatrix} + \begin{bmatrix} 0 \\ 2\Omega_o \end{bmatrix} \Delta \Omega \quad (30)$$

From the space state models infer:

The Altitude Control is the PD Controller then the Mass Resume block combines to control the altitude.

Position control consists of a circular cascaded PD loop where the outer PD control loop generates a signal output signal to the inner corner PD control loop.

From the above formula, the transfer function is obtained, thereby simulating the stable controller Z shown in Figure 10.



**Figure 10.** The PID controller controls Z.

### 3.2. Disturbance Signal

This simulation models wind turbulence. The goal is to add real-world effects. Using the concept in this model, users can further model other real-world terms, such as noise sensors and motor output fluctuations.

Turbulence is written in the [Force and Environment/Turbulence] block. Wind Level, Noise Level, and Wind Sampling Time are editable to see how the drone performs in different conditions.

In the general case, the wind speed vector W is a very complex function of space and time:

$$W = W(x_0, y_0, z_0, t)$$

The wind speed vector in the general case can have an arbitrary direction in space, so it is possible to analyze the components W, W, W (according to the ground coordinate system O). The components themselves are also spatially and temporally dependent.

$$W = W_x(x_0, y_0, z_0, t)$$

$$W = W_y(x_0, y_0, z_0, t)$$

$$W = W_z(x_0, y_0, z_0, t)$$

The wind simulation varies at various levels [25] below as shown below, Algorithm 1. is algorithm for Wind setting using Matlab/Simulink:

---

**Algorithm 1:** Wind setting algorithm

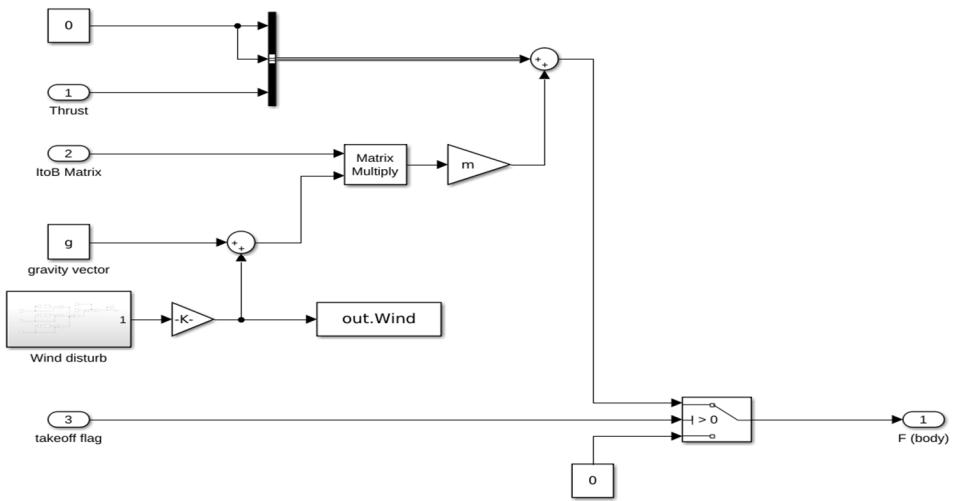
---

```
% Disturbance Signal
ts = 0.01;
WindLevel = [10];
WindSpeed = [13244];
```

---

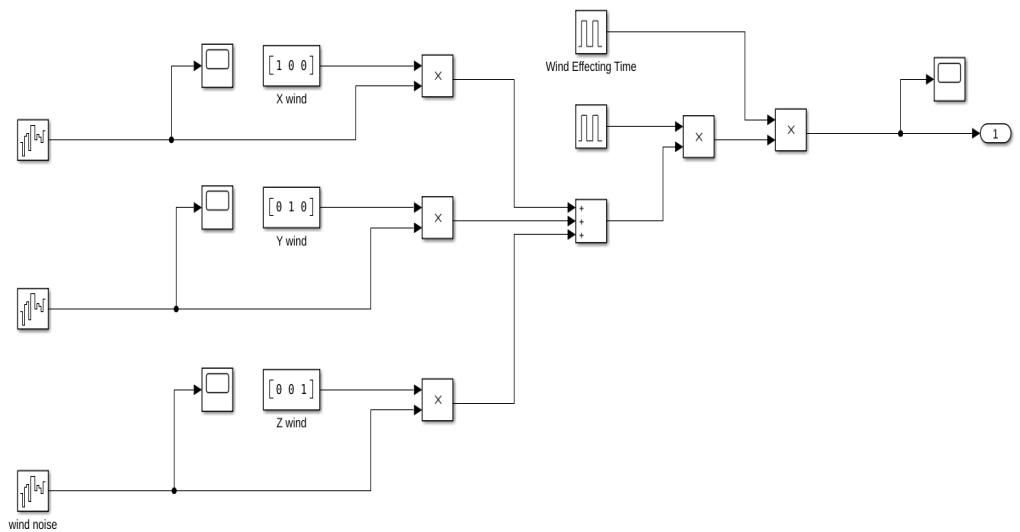
In the Wind algorithm is the set WindLevel = [x] and WinSpeed = [y] where x, y are any 2 values that are suitable for the problem. It can be customized according to actual requirements.

Figures 11 and 12 show simulation of the forces and noise signals acting on the Drone.



**Figure 11.** Forces acting on the Drone.

Inside Force and Environment subsystem:  
Inside Wind Disturb block:

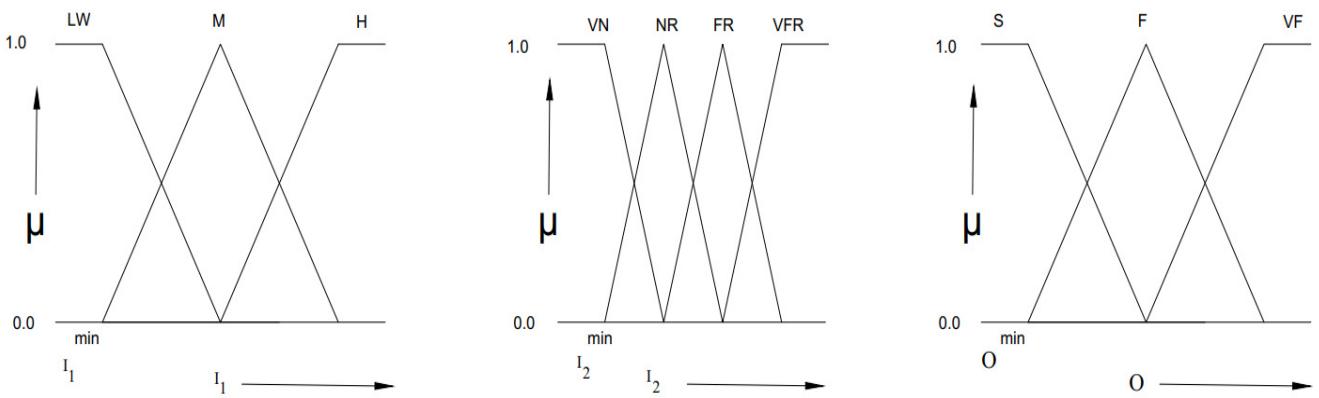


**Figure 12.** Wind noise.

### 3.3. Algorithm Neural Network–Fuzzy Controller

In this study, the proposed neural-fuzzy controller is a neural-fuzzy system based on Takagi and Sugeno's Approach. It is also an Adaptive Neuro-Fuzzy Inference System (ANFIS) [25–31]. The proposed controller has two inputs:  $I_1$  and  $I_2$  and one output:  $O$ . Membership function distributions of input and output variables of the controller to be output are shown in Figure 13.

$$O = f(I_1, I_2) \quad (31)$$



**Figure 13.** Membership function distributions of input and output variables.

The Neural-Fuzzy controller has 7 values of the fuzzy set that are related according to the Table 2.

**Table 2.** Rule Base.

	VN	NR	FR	VFR
LW	S	S	F	F
M	S	F	F	VF
H	S	F	VF	VF

Where, LM: Low; M: Medium; H: High; VN: Very Near; NR: Near; FR: Far; VFR: Very Far; S: Slow; F: Fast; VF: Very Fast.

According to first-order Takagi-Sugeno's model of Fuzzy Logic Control:

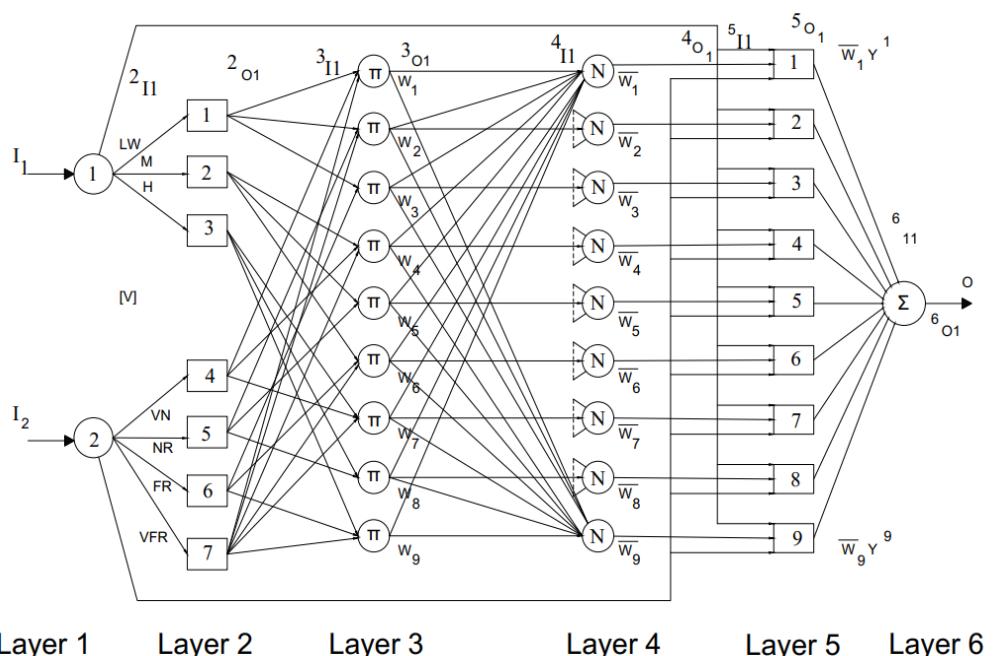
$$y^i = a_i I_1 + b_i I_2 + c_i \quad (32)$$

where,

$$I = 1, 2, \dots$$

$a_i, b_i, c_i$  are coefficients.

Figure 14 shows a diagram of the relationship between the classes of the ANFIS layer.



**Figure 14.** ANFIS architecture.

Following 4 rules are fired, corresponding to a set input  $I_1$  and  $I_2$ :

$$\text{If } I_1 \text{ is LW and } I_2 \text{ is SM the } y^1 = a_1 I_1 + b_1 I_2 + c_1 \quad (33)$$

$$\text{If } I_1 \text{ is LW and } I_2 \text{ is LR the } y^2 = a_1 I_1 + b_1 I_2 + c_1 \quad (34)$$

$$\text{If } I_1 \text{ is M and } I_2 \text{ is SM the } y^4 = a_1 I_1 + b_1 I_2 + c_1 \quad (35)$$

$$\text{If } I_1 \text{ is M and } I_2 \text{ is LR the } y^5 = a_1 I_1 + b_1 I_2 + c_1 \quad (36)$$

Layer 1:

Linear TF (output = input)

$$1_{o_1} = 1_{I_1} = I_1^* \quad (37)$$

$$1_{o_2} = 1_{I_2} = I_2^* \quad (38)$$

Layer 2:

Fuzzification ( $\mu$ )

Layer 3:

Firing strengths of the rules are calculated considering the products of  $\mu$  values:

$$w_1 = \mu_{LW}(I_1^*) \times \mu_{SM}(I_2^*) \quad (39)$$

$$w_2 = \mu_{LW}(I_1^*) \times \mu_{LR}(I_2^*) \quad (40)$$

$$w_4 = \mu_M(I_1^*) \times \mu_{SM}(I_2^*) \quad (41)$$

$$w_5 = \mu_M(I_1^*) \times \mu_{LR}(I_2^*) \quad (42)$$

Layer 4:

Calculate normalized firing strength of each rule:

$$\overline{w_1} = \frac{w_1}{w_1 + w_2 + w_4 + w_5} \quad (43)$$

$$\overline{w_2} = \frac{w_2}{w_1 + w_2 + w_4 + w_5} \quad (44)$$

$$\overline{w_4} = \frac{w_4}{w_1 + w_2 + w_4 + w_5} \quad (45)$$

$$\overline{w_5} = \frac{w_5}{w_1 + w_2 + w_4 + w_5} \quad (46)$$

Layer 5:

Output is calculated as the product of normalized firing strength and output of the corresponding fired rule  $y$

$$5_{O_1} = \overline{w_1} \times y^1 \quad (47)$$

$$5_{O_2} = \overline{w_2} \times y^2 \quad (48)$$

$$5_{O_4} = \overline{w_4} \times y^4 \quad (49)$$

$$5_{O_5} = \overline{w_5} \times y^5 \quad (50)$$

Layer 6:

Overall output:

$$6_{O_1} = \overline{w_1}y^1 + \overline{w_2}y^2 + \overline{w_4}y^4 + \overline{w_5}y^5 \quad (51)$$

The error loss function  $E$  is given by:

$$E = \frac{1}{2}(y_m - y) \quad (52)$$

where:

$y_m$  is the actual output value

$y$  is the desired output value

Algorithm 2. is algorithm for ANFIS controller using Matlab/Simulink.

---

**Algorithm 2:** Neural-Fuzzy controller setting algorithm

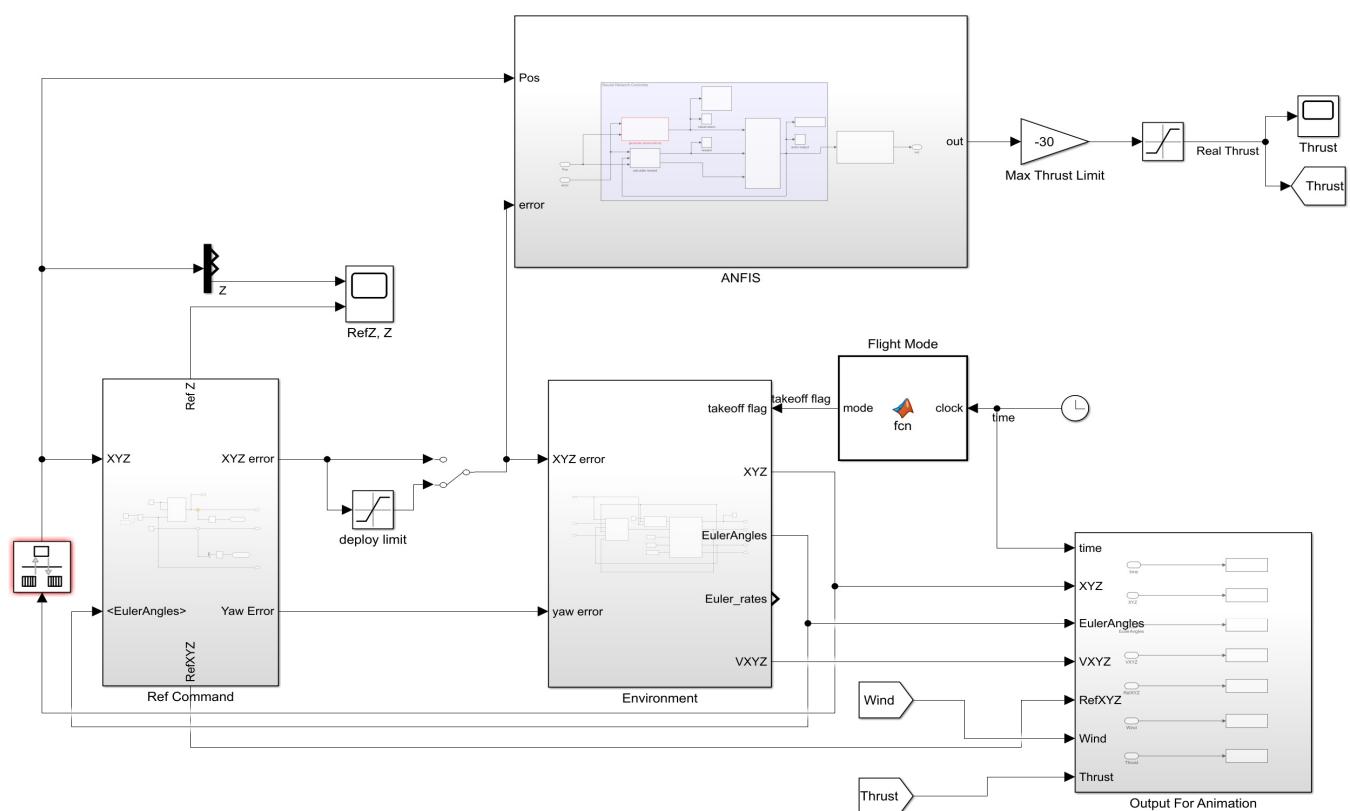
---

```
fis = anfis( Drone_NF_ControllerData );
numObs = 3;
opt = anfisOptions('InitialFIS',6,'EpochNumber',40);
[trainFIS,trainFISError,~,validationFIS,validationFISError] = anfis(NFControllerData,opt);
observationInfo = rlNumericSpec([numObs 1]);
observationInfo.Name = 'observations';
numAct = 1;
actionInfo = rlNumericSpec([numAct 1]);
actionInfo.LowerLimit = -1;
actionInfo.UpperLimit = 1;
actionInfo.Name = 'Control';
actionInfo.Description = 'Thrust';
mdl = 'Drone_NF_Controller';
load_system(mdl);
blk = [mdl,'/RL Data'];
```

---

### 3.4. Model of Neural Network–Fuzzy Logic Controller in Matlab/Simulink

Figure 15 simulates the Neural-Fuzzy controller [26,32,33] on Matlab/Simulink. Including environmental blocks (interference force of the impacting environment, control block, and simulation of 6DoF coordinate system mentioned in Section 2.1) and Neural-Fuzzy system(ANFIS), Ref command.



**Figure 15.** Model of Neural network–Fuzzy controller for marine rescue drone in Matlab/Simulink.

#### 4. Results and Discussion

The input parameters used in the simulation are shown in Tables 3 and 4.

**Table 3.** Control input values.

Value Name	Variable Value	Unit of Variable
$U_1$	$\leq 30$	N
$U_2$	$\leq 1.2$	Nm
$U_3$	$\leq 1.2$	Nm
$U_4$	$\leq 1.2$	Nm
Mass (m)	0.2	Kg
Gravity acceleration (g)	9.8	$m/s^2$
Moment of inertia (I)	$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0.08 \end{bmatrix}$	
Angles	$\leq 60$	Degree

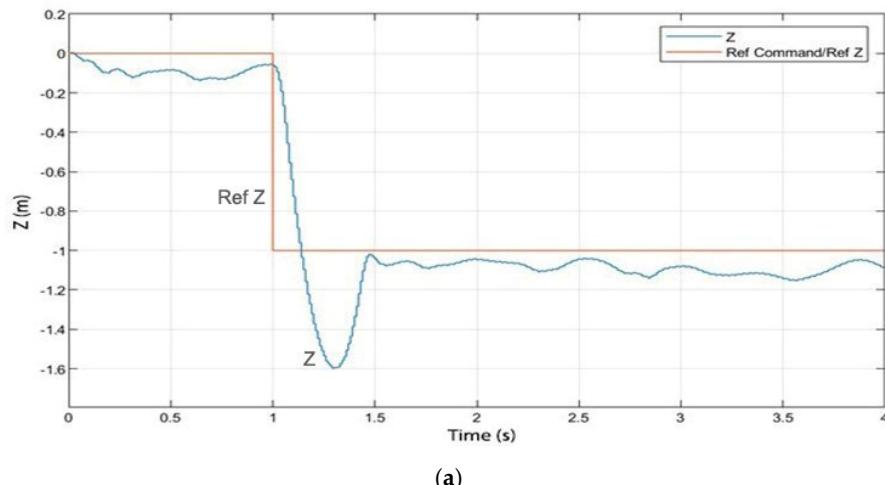
**Table 4.** Parameter for PID Control Scheme.

Part	P	I	D
$A_1$	0.95	X	X
$A_2$	0.21	X	X
$A_3$	1.1	X	1.2
$A_4$	1	X	X
$A_5$	0.2	X	X
$A_6$	1.1	X	0.45
$A_7$	0.6	X	0.4

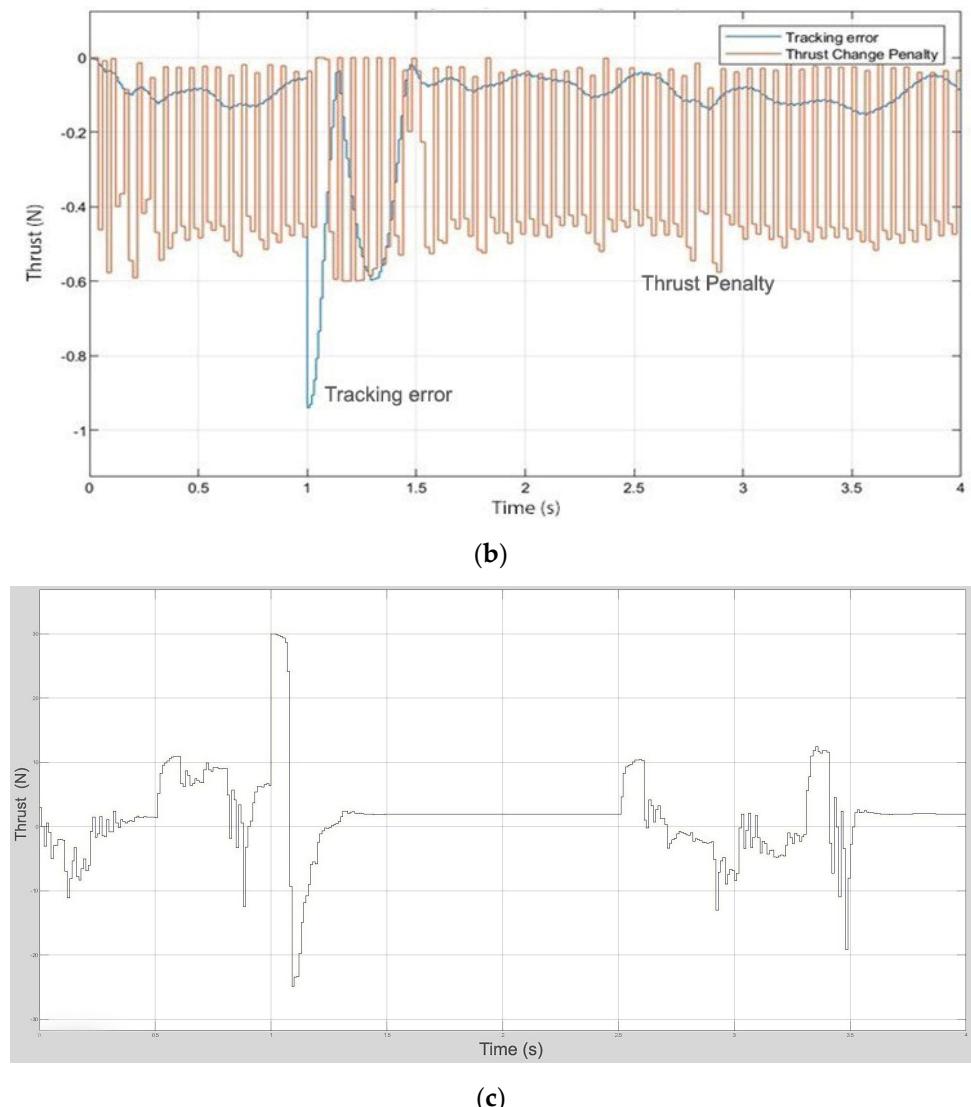
Using the manual method is performed many times testing many times, and the Tunning method of Matlab/Simulink obtains the PD coefficients of the PID controller.

The experiment was performed several times with different noise parameters and obtained the same results. There are different levels of wind, and the Drone can only work stably with a wind level below 10. At level 10 there is already but deviation in motion. When at level 11, the Drones cannot work stably. Below are the results obtained showing the features of the Neural-Fuzzy controller. Simulation on Matlab obtained the following results:

Figure 16 shows that signal Z is always selling close to the given set signal and Thrust's quick response to a tracking error signal. This shows the responsiveness of the controller so that operations are more accurate than expected.



**Figure 16.** Cont.

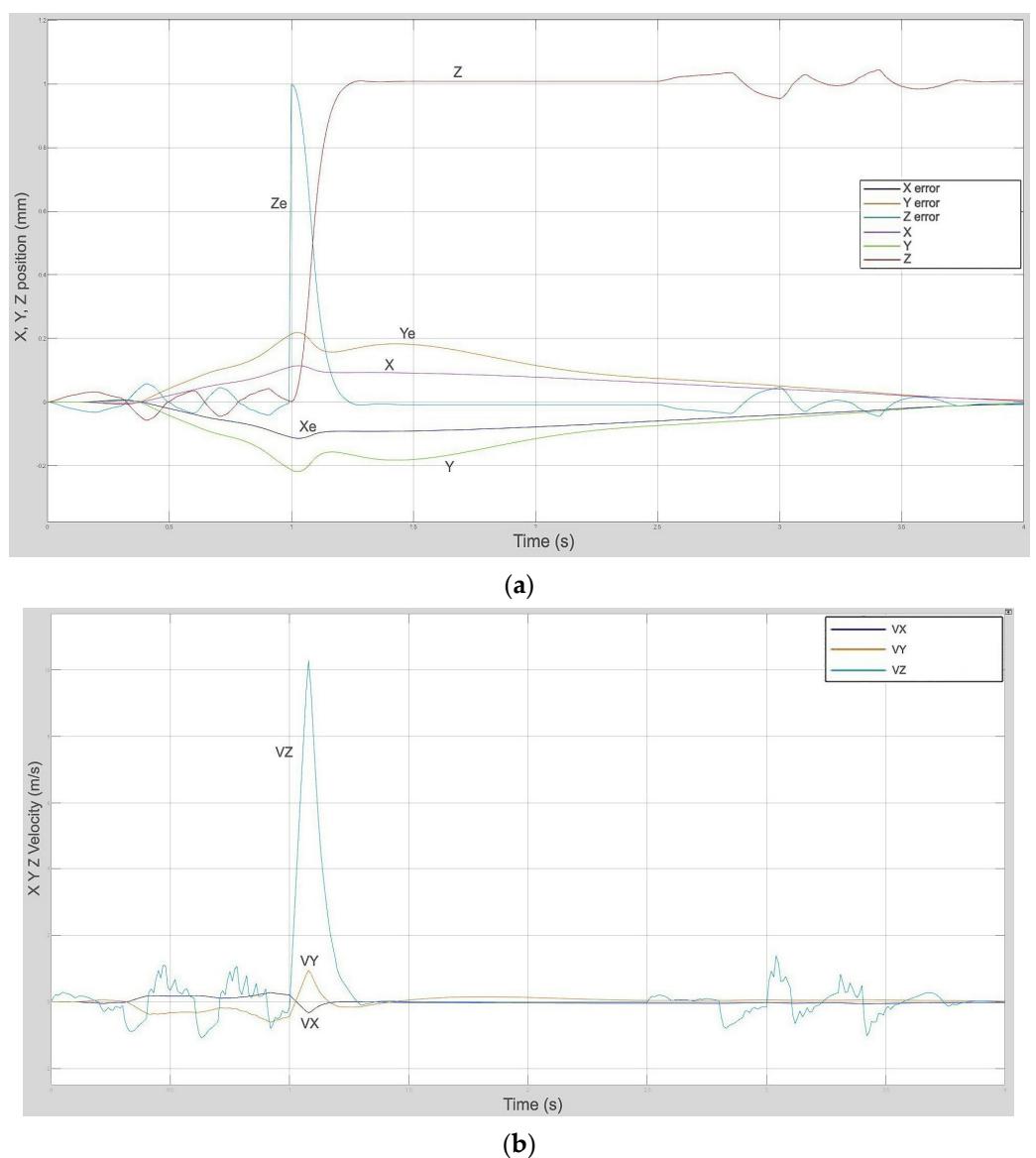


**Figure 16.** (a) Adjust the height of the Drone to follow the set signal. (b) The actual change of the drone system. (c) Change of the rudder so that the error deviates from the trajectory to zero.

Figure 17 shows the change of parameters position xyz as well as velocity Vxyz to reduce the error signal to zero. The first second due to noise oscillation makes the operation stable, and the error signal reaches the highest amplitude where the values are learned and adjusted for the error signals and bring them to zero at time  $t = 4$  s, and can still be interfered with by the impact of the ever-changing and continuous environment. Then the Neural-Fuzzy controller will always update and give a new signal to adjust and bring the error signal to zero to help the rescue drone work stably.

Figure 18 shows that both Theta and Phi can change in a timely manner to closely follow the set signal and increasingly narrow the error gap with the set signal.

Figure 19 is a simulated image of the drone's movement on the 3D xyz coordinate system, it can be seen that the orbits are close to each other and have very little deviation, the difference between the two orbits is very small, the two orbits are different each other is approximately 0.3 mm.



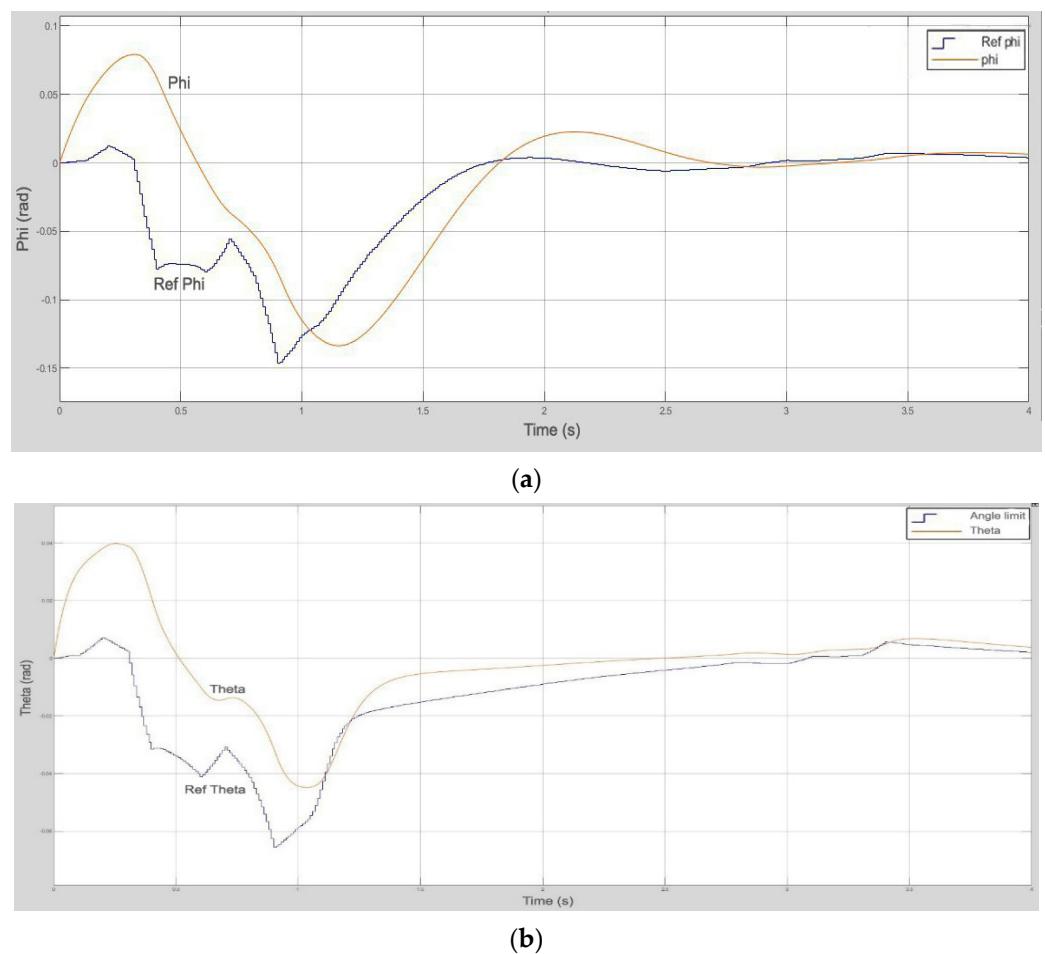
**Figure 17.** Position tracking error. (a) Position tracking error. (b) Drone velocity along coordinate axes.

To verify how to evaluate the difference, we use a Neural-Fuzzy controller to control the Z position of the drone. Compare the controller with a classical PID controller under different wind disturbance levels along the z-axis and thrust.

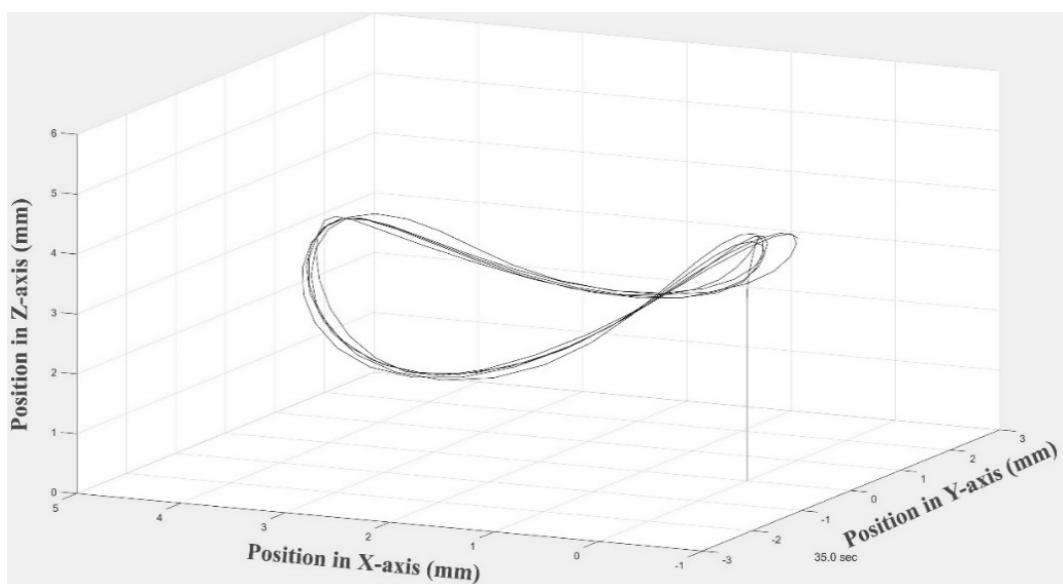
To verify increased accuracy, compare two controllers with noise and no noise conditions to be able to compare and evaluate in the most objective and accurate way.

Where there is no wind:

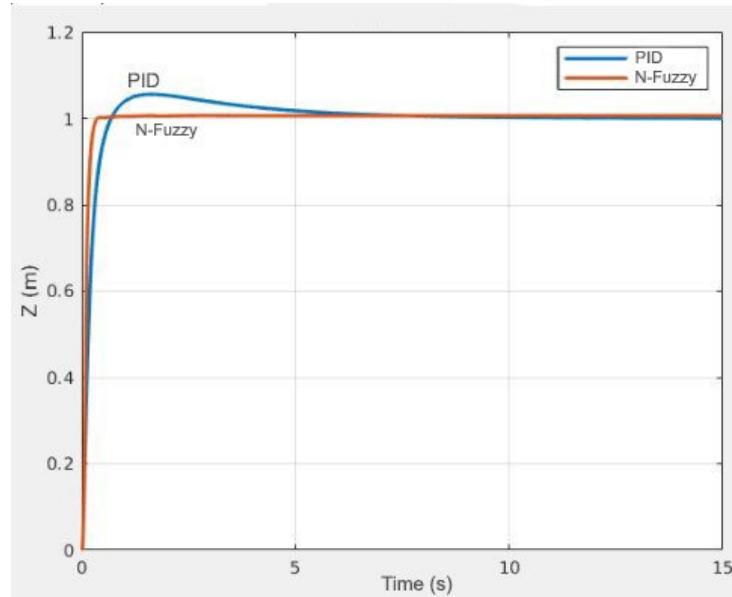
The stability and tracking of the given altitude  $Z = 1$  of the Neural-Fuzzy controller is much higher than that of the classical PID controller. From the above two simulation results (Figures 20 and 21), we can see that under the influence of different noise levels, the Neural-Fuzzy controller is more effective.



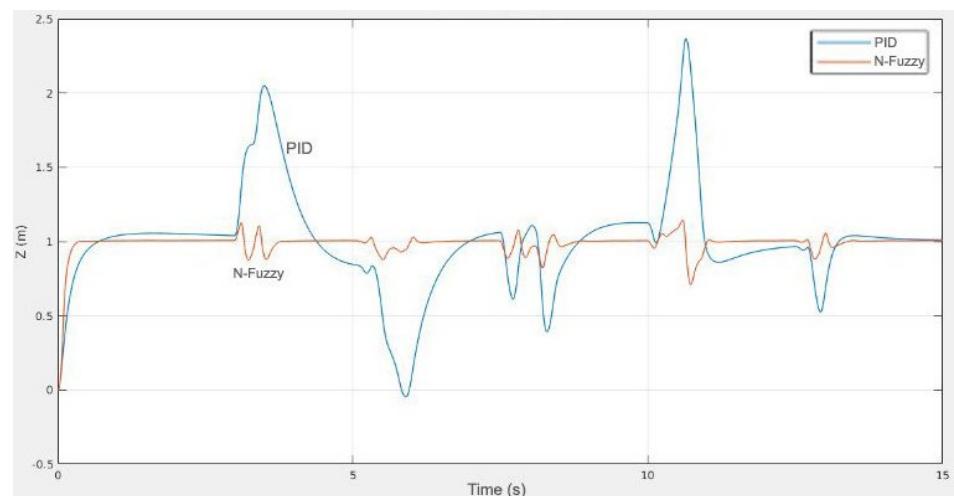
**Figure 18.** The return phi angle relative to the value set in the simulation. **(a)** The return phi angle relative to the value set in the simulation. **(b)** The return theta angle relative to the value set in the simulation.



**Figure 19.** Motion trajectory of Marine Rescue Drone.

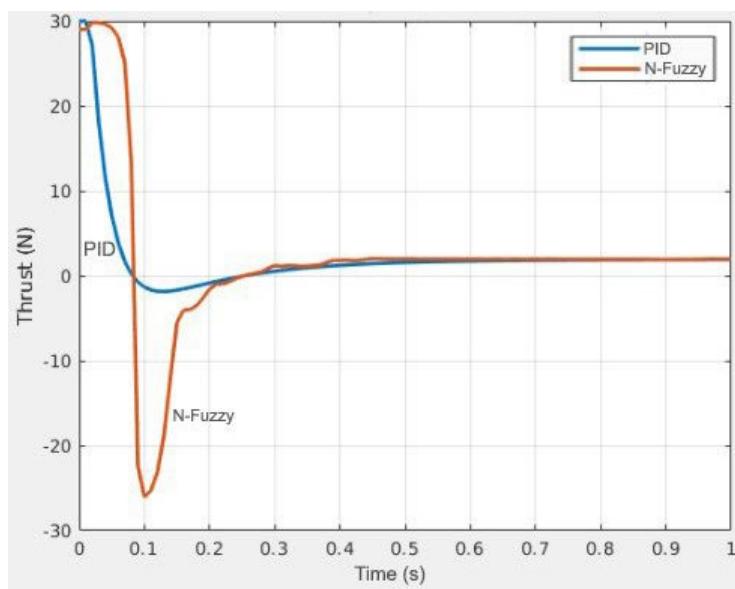


**Figure 20.** The results compare the Z position (attitude) of the Neural-Fuzzy controller and the classic PID controller (no wind).

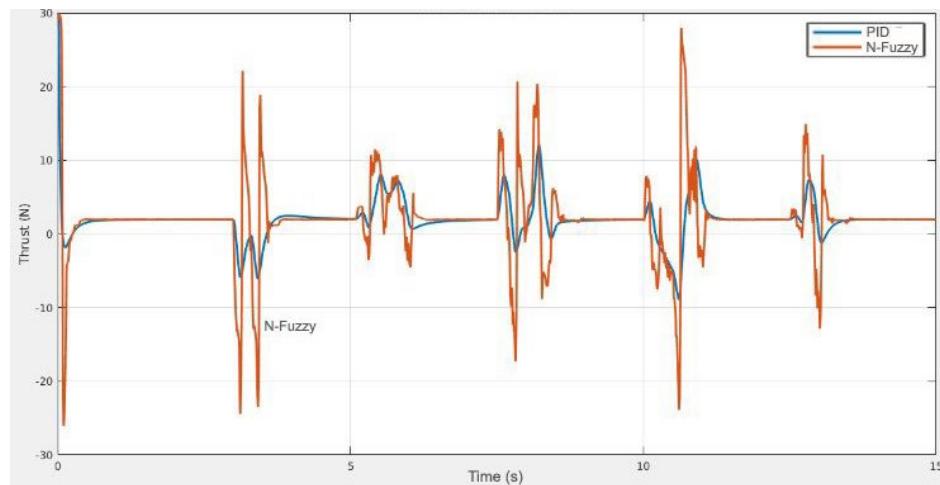


**Figure 21.** The results compare the Z position (attitude) of the Neural-Fuzzy controller and the classic PID controller (wind action).

In Figures 22 and 23, we can see that the propulsion of the control system by the Neural-Fuzzy hybrid controller is fast, strong, and increases the efficiency to help the navigation of the maritime unmanned rescue aircraft, better than the conventional PID controller. The response of the Neural-Fuzzy controller is much better than that of the PID controller.



**Figure 22.** Comparison of thrust of unmanned sea rescue aircraft using PID controller and Neural-Fuzzy hybrid controller (no wind).



**Figure 23.** Comparison of thrust of unmanned sea rescue aircraft using PID controller and Neural-Fuzzy hybrid controller (wind action).

It is easy to see that the Neural-Fuzzy controller reaches the steady state of Z very soon (it takes about 0.5 s from the start) with a Thrust force of up to 30 Newtons to quickly adjust the state. As for the PID controller, it takes about 1.5 s to reach a stable state, and the highest Thrust force can only be up to 10 Newtons, making the adjustment against noise to help the Drone stabilize the state for longer than the Neural-Fuzzy controller.

From the obtained results, we can see that the disturbances caused by the environment were quickly overcome thanks to the self-learning feature of the neural network and the inference of the fuzzy logic controller for the drone to operate efficiently. There are some features worth discussing, such as:

How to create a reward function so that this composite controller can learn just like what we want it to do. It is noticed that one of the advantages of a neural network is that it can learn on its own. However, it is still needed to monitor the training to determine if it “learns the right on path” or not.

Training time and computing resources are enormous. (For this simulation it requires 24 h memory and 6 GB for training).

The generated command output may not be appropriate in real-life cases. (This problem assumes the use of a perfect motor that can perfectly obey the thrust command. To implement the Neural-Fuzzy controller in the real world, it is necessary to model the motor at the same point in time).

## 5. Conclusions

1. The design of the Neural-Fuzzy controller will help improve the ability to control the trajectory to help search and rescue at sea become more accurate and faster. That is the main purpose of this article. When using a hybrid controller, it helps individual systems to complement and compensate for their shortcomings, and combine to promote their own strengths, helping to create an efficient system.
2. Neural-Fuzzy controller is 67% more efficient, responds approx 85% faster and has better anti-interference ability with three times larger amplitude thrust than PID controller.
3. This proposed hybrid controller can be used across different drone sizes, weights, and configurations without the need to re-adjust the PID gain.
4. Along with the strong development of science and technology; Neural systems and fuzzy networks are increasingly being applied in many fields. Based on human logic, with the advantage of simplicity and accurate handling of uncertain information; The artificial neural network and the combined fuzzy network have brought many significant effects in the field of control and automation.

**Author Contributions:** D.-A.P., Formal analysis, Software, resources, data curation, writing—original draft preparation, data collected and analyzed, visualization; S.-H.H., supervision, project administration, writing—review and editing. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the research grant of the Gyeongsang National University in 2022.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

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

## References

1. Chamayou, G. *Drone Theory*; Penguin: London, UK, 2015.
2. Park, S.; Her, J.; Kim, J.; Lee, D. Design, modeling and control of omni-directional aerial robot. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016.
3. Niu, J.; Wang, D.; Han, P.; Fang, J.; Ren, X.; Qin, Y.; Xu, Z. Image Enhancement of Low Light UAV via Global Illumination Self-aware feature Estimation. In Proceedings of the 2021 3rd International Conference on Natural Language Processing (ICNLP), Beijing, China, 26–28 March 2021.
4. Vrontisa, D.; Christofi, M.; Pereira, V.; Tarba, S.; Makrides, A.; Trichina, E. Artificial intelligence, robotics, advanced technologies and human resource management: A systematic review. *Int. J. Hum. Resour. Manag.* **2021**, *33*, 1237–1266. [[CrossRef](#)]
5. Bennett, M.K.; Younes, N.; Joyce, K. Automating Drone Image Processing to Map Coral Reef Substrates Using Google Earth Engine. *Drones* **2020**, *4*, 50. [[CrossRef](#)]
6. Ling, G.; Draghic, N. Aerial drones for blood delivery. *Transfusion* **2019**, *59*, 1608–1611. [[CrossRef](#)] [[PubMed](#)]
7. Ahmed, S.; Qiu, B.; Kong, C.; Xin, H.; Ahmad, F.; Lin, J. A Data-Driven Dynamic Obstacle Avoidance Method for Liquid-Carrying Plant Protection UAVs, Francisco Manzano-Agugliaro. *Agronomy* **2022**, *12*, 873. [[CrossRef](#)]
8. Scott, J.; Scott, C. Drone Delivery Models for Healthcare. In Proceedings of the 50th Hawaii International Conference on System Sciences, Hilton Waikoloa Village, HI, USA, 4–7 January 2017.
9. Kim, S.J.; Jeong, Y.; Park, S.; Ryu, K.; Oh, G. A Survey of Drone use for Entertainment and AVR (Augmented and Virtual Reality). In *Augmented Reality and Virtual Reality*; Springer: Cham, Switzerland, 2017; pp. 339–352.
10. Packer, J.; Reeves, J. Romancing the Drone: Military Desire and Anthropophobia from SAGE to Swarm. *Can. J. Commun.* **2013**, *38*, 309–332. [[CrossRef](#)]

11. Gray, P.C.; Fleishman, A.B.; Klein, D.J.; McKown, M.W.; Bézy, V.S.; Lohmann, K.J.; Johnston, D.W. A convolutional neural network for detecting sea turtles in drone imagery. *Methods Ecol. Evol.* **2018**, *10*, 345–355. [[CrossRef](#)]
12. Simon, P. *Too Big to Ignore: The Business Case for Big Data*; John Wiley & Sons: Hoboken, NJ, USA, 2013; Volume 72.
13. Tezza, D.; Andujar, M. The State-of-the-Art of Human–Drone Interaction: A Survey. *IEEE Access* **2019**, *7*, 167438–167454. [[CrossRef](#)]
14. Um, Y.-C.; Cho, H.-L. Integral  $\gamma$ -Sliding Mode Control for a Quadrotor with Uncertain Time-Varying Mass and External Disturbance. *J. Electr. Eng. Technol.* **2022**, *17*, 707–716. [[CrossRef](#)]
15. Wang, R.; An, A.; Wen, Y.; Song, H. Study on the Influence of Parallel Fuzzy PID Control on the Regulating System of a Bulb Tubular Turbine Generator Unit. *J. Electr. Eng. Technol.* **2021**, *16*, 1403–1414. [[CrossRef](#)]
16. Khan, S.A.; Liu, C.; Ansari, J.A. Centralized Fuzzy Logic Based Optimization of PI Controllers for VSC Control in MTDC Network. *J. Electr. Eng. Technol.* **2020**, *15*, 2577–2585. [[CrossRef](#)]
17. Abdelhay, S.; Zakriti, A. Modeling of a Quadcopter Trajectory Tracking System Using PID Controller, ScienceDirect. *Procedia Manuf.* **2019**, *32*, 564–571. [[CrossRef](#)]
18. Desa, H.; Ahmed, S.F.; Azfar, A.Z. *Adaptive Hybrid Control Algorithm Design for Attitude Stabilization of Quadrotor (Drone)*; Archives Des Sciences, University Malaysia Perlis: Arau, Malaysia, 2013; Volume 66.
19. Teppo, L. *Modelling and Control of Quadcopter*; Independent research project in applied mathematics; Aalto University: Espoo, Finland, 2011.
20. Lin, C.T.; Lee, C.S.G. *Neural Fuzzy Systems: A Neural Fuzzy Synergism to Intelligent Systems*; Prentice Hall: Hoboken, NJ, USA, 1996; ISBN 0132351692.
21. Zadeh, L.A. *Fuzzy Logic, Neural Networks, and Soft Computing*; ACM: New York, NY, USA, 1994.
22. Tiep, D.K.; Lee, K.N.; Im, D.Y.; Kwak, B.; Ryoo, Y.J. Design of Fuzzy—PID Controller for Path Tracking of Mobile Robot with Differential Drive. *Int. J. Fuzzy Log. Intell. Syst.* **2018**, *18*, 220–228. [[CrossRef](#)]
23. Lin, C.-T. A neural fuzzy control system with structure and parameter learning. *Fuzzy Sets Syst.* **1995**, *70*, 183–212. [[CrossRef](#)]
24. Lee, S.-J.; Ouyang, C.-S. A Neuro-Fuzzy System ModelingbWith Self-Constructing Rule Generation and Hybrid SVD-Based Learning. *IEEE Trans. Fuzzy Syst.* **2003**, *11*, 341–353.
25. Sorbelli, F.B.; Corò, F.; Das, S.K.; Pinotti, C.M. Energy-Constrained Delivery of Goods with Drones under Varying Wind Conditions. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 6048–6060. [[CrossRef](#)]
26. Jang, J.-S.R. ANFIS: Adaptive-network-based fuzzy inference system. *IEEE Trans. Syst. Man Cybern.* **1993**, *23*, 665–685. [[CrossRef](#)]
27. Seng, T.L.; Khalid, M.B.; Yusof, R. Tuning of a neural fuzzy controller by genetic algorithm. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **1999**, *29*, 226–236. [[CrossRef](#)] [[PubMed](#)]
28. Sandhu, G.S.; Rattan, K.S. Design of a neural fuzzy controller. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997.
29. Louzazni, M.; Aroudam, E. Control and Stabilization of Three-Phase Grid Connected Photovoltaics using PID-Fuzzy Logic. In Proceedings of the IEEE International Conference on Intelligent Energy and Power Systems (IEPS), Kyiv, Ukraine, 2–6 June 2014.
30. Gao, W.; Selmic, R.R. Neural network control of a class of nonlinear systems with actuator saturation. *IEEE Trans. Neural Net.* **2006**, *17*, 147–156. [[CrossRef](#)] [[PubMed](#)]
31. Zadeh, L.A.; Aliev, R.A. Chapter 16: Adaptive Neuro-Fuzzy Inference Systems (ANFISs). In *Fuzzy Logic Theory and Applications*; World Scientific Book: Singapore, 2018; pp. 453–493.
32. The MathWorks, Inc. *Fuzzy Logic Toolbox™ User’s Guide*; Version 2.2.24; (Release 2016); The MathWorks, Inc.: Natick, MA, USA, 2016.
33. Demuth, H.; Hagan, M.B.V.M. *Neural Network Toolbox™ User’s Guide*; Version 9.1; (Release 2016); The Math Works, Inc.: Natick, MA, USA, 2016.