International Conference on Optimization, Simulation and Control

Compared Results Of Neural Network Models In Quadcopter Control

Bat-Erdene Gotov, MS[1]; Tengis Tserendondog, PhD[2]; Lodoiravsal Choimaa, PhD[3]; Batmunkh Amar, PhD, Prof[2]

[1]Mongolian National University of Medical Sciences, [2]Mongolian University of Science and Technology, [3]National University of Mongolia

Abstract

This article presents the results of an experiment in which quadcopter were stabilized using a neural network-based control method. To train the dynamics of unmanned aerial vehicles, a neural network model with two hidden layers of 3, 10, and 20 nodes was used. Data were collected for use in neural network training using standard PID controls. The data collected were used to train neural network flight stability and to test learning outcomes. The results of neural network training and experimental results of PID control methods are compared.

Introduction

In recent years, quadrotors (UAV, quadcopter) emerged widely in both civil and military areas. There are many applications of quadrotor UAVs, such as rescue operations, surveillance, aerial photography for mapping, inspection of power lines, traffic monitoring in urban areas, crop monitoring and spraying, border patrol, search operations for missing persons and natural disasters.

In supervised learning, each (training) example is a pair consisting of an input value (angles of quadcopter) and a desired output value (rotational speed of propeller). After learning on the training data, the system is expected to make correct predictions for future (unseen) inputs.

Methods and Materials

MATHEMATICAL MODEL OF QUADROTOR

The quadrotor has "CROSS type" flying configuration (Fig. 1), with two pairs of opposite rotors rotating clockwise and the other rotor pair rotating counter-clockwise to balance the torque. The roll, pitch, yaw and up-thrust actions are controlled by changing the thrusts of the rotors using pulse width modulation (PWM) to give the desired output. Consider a quadrotor UAV with six DOF and the motion of the quadrotor can be divided into two subsystems which include a rotational subsystem (roll, pitch and yaw) and a translational subsystem (altitude and x and y position). In order to obtain the necessary measurements quadcopter must be equipped with a suitable inertial measurement system (IMU). This IMU delivers the accelerations and angular rates that can be used to further estimate velocities and Euler angles. We consider an inertial frame and a body fixed frame whose origin is in the center of mass of the quadrotor as shown in Fig. 1

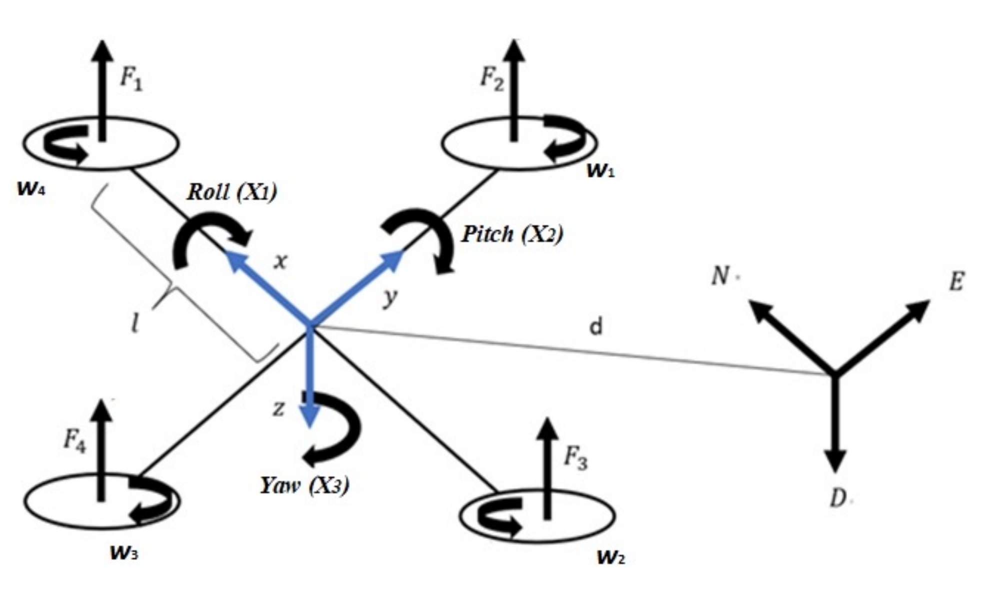

Figure 1. Quadcopter frame.

DESIGN OF PROPOSED NEURAL NETWORK MODEL

The choice to use a neural network for the main quadrotor control was based on the need to deal with large uncertainties in both parameter estimates and wind disturbances. One potential approach can be to model such dynamics using neural networks. Neural networks (NN) are known to be universal function approximators; their structure allows them to model highly nonlinear functions and unobserved states directly from the observed data, which might in general be hard to model directly. Moreover, they can learn a generalized model that can be extended beyond the observed data. However, it is not clear if the proposed NN-based model can be used to control the system, and if the learned dynamics accurately represent the system beyond the data it was trained on.

This study proposes a four-layer NN model, which consists of an input layer, a hidden layer 1, a hidden layer 2, and an output layer. The NN architecture is shown in Figure 2 and can be explained as follows. The input layer takes the current state of the system as input. Prior to the training, the weight of the input layer was randomly assigned $\theta_{i,j}^{(1)}$ and a base value of b_1 was added. The second layer has 10 neurons, and the weight is randomly selected as $\theta_{i,j}^{(2)}$ and an additional base value b_2 was added. The sigmoid function is discussed here. Each of the NN hidden units computes the previous product of $\theta_{i,j}^{(1)}$. The output layer has 4 neurons that represent the speed of each motor.

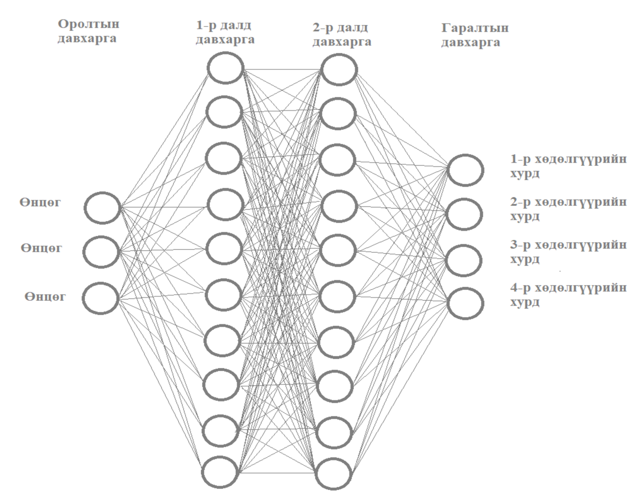

Figure 2. The neural network model we use.

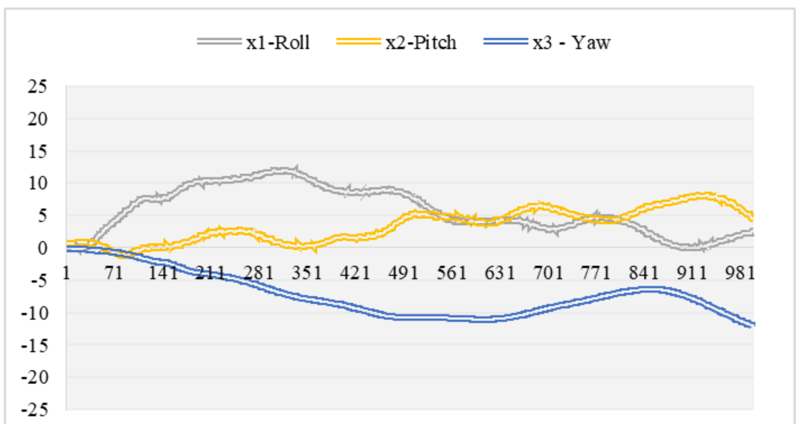

Figure 3. drone angle changes.

Experimental Results

The total data is divided into three sections: 70% for training, 20% for validation, and 10% for testing. The neural network model was trained using MATLAB. Some network parameters were adjusted as follows: learning speed (0.001), no regulatory factor, repetition (9M), number of layers, and number of neurons in the latent layer. Before training a neural network, we follow a number of steps to pre-process the data. We did not take the position of the drone as an input, but the angle of the drone as an input to the neural network. Rotor speed or PWM is given in the range of 800 to 2000. We calculated the collected outputs (rotational speed - PWM value) on a scale, each of which has a value between 0 and 1 (1).

$$y_i = \left(\frac{w_i - 700}{1000}\right) \qquad (1)$$

This is to give equal weight to the data collected by the neural network.

We measured the angles exerted on the drone by an external force, as shown in Figure 3.

Figure 4 (a, b, c, d) shows the dependence of the angular change of the motor speed adjusted by the PID control in blue and orange on the motor speed learned in the neural network model.

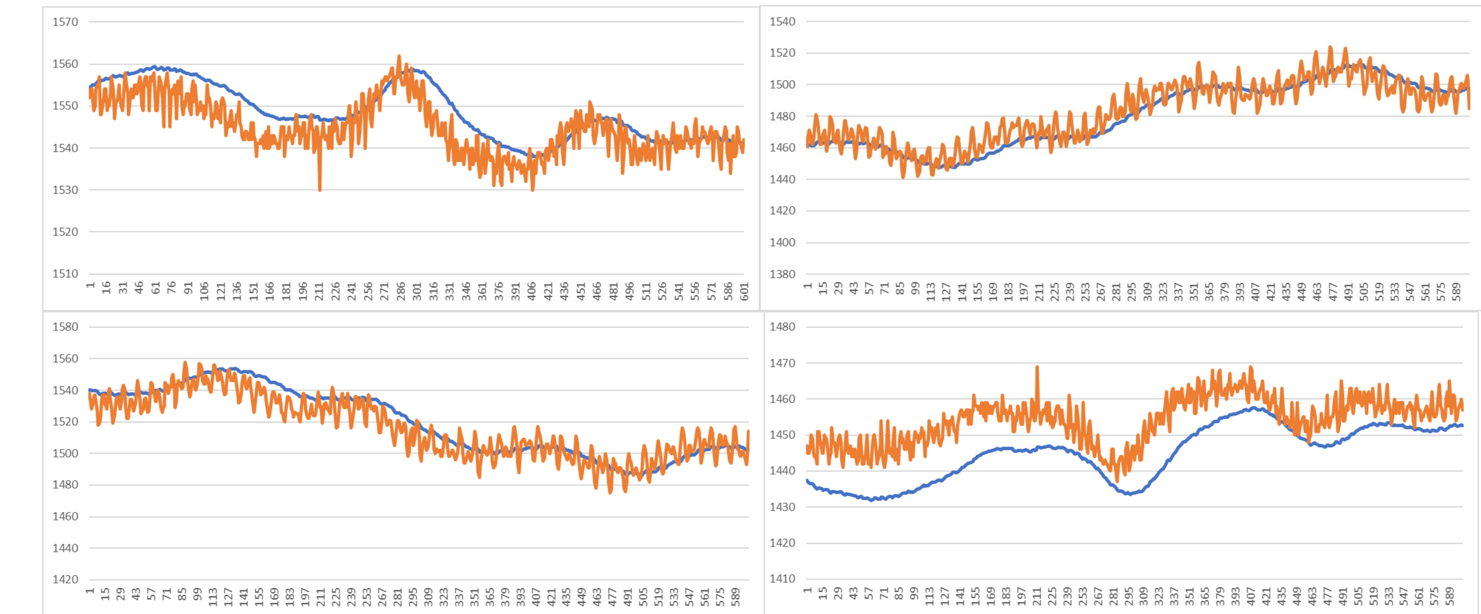

Figure 4. PID speed of motors, speed of NN training.

In the second experiment, we changed the number of nodes in the hidden layer. Depending on the number of nodes in the ground layer, the dynamic design of the drone will change. To determine the value of the cost function, we changed the nodes to 3, 10, and 20. Figure 5 shows the change in the value of the evaluation function depending on the number of nodes.

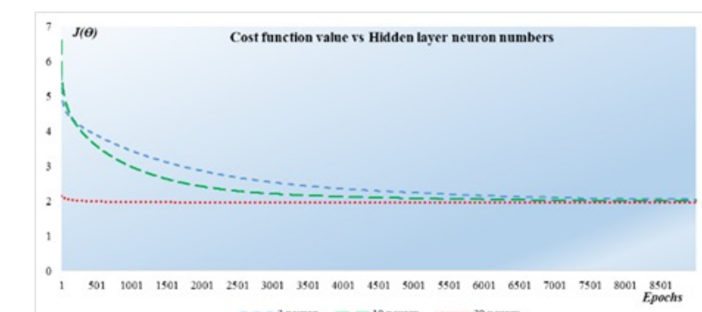

Figure 5. Correlation between the value of the evaluation function and the number of nodes.

In the third experiment, we randomly mixed the training data and compared the results of the first training with those of the first training, as shown in Figure 6 (a, b, c). To determine the value of the evaluation function, we conducted trainings on nodes 3, 10, and 20.

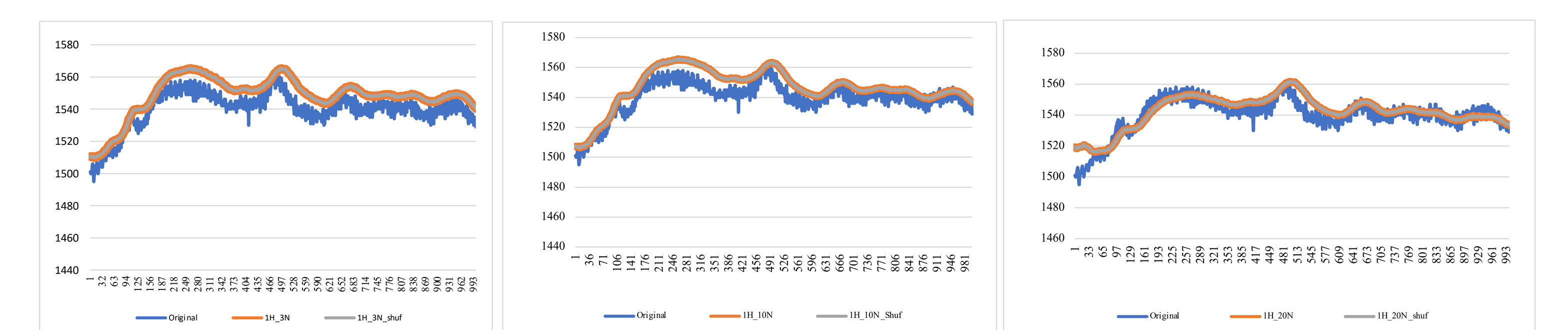

Figure 6. Results of NN model training with 3, 10, and 20 neurons with one latent layer.

In the next experiment, we conducted a training on the initial and randomly mixed data when the number of hidden layers was 2 and the number of nodes was 3, and compared the results in Figure 7.

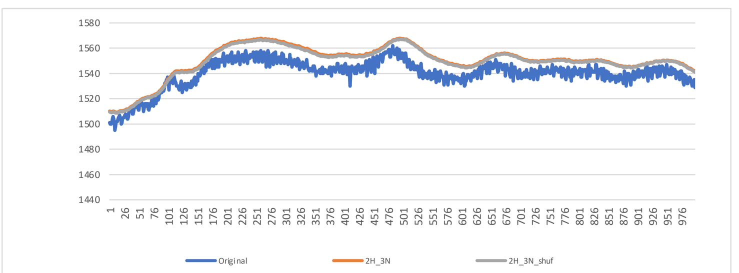

Figure76. The speed of rotation learned by NN at one and two times the number of hidden layers.

Conclusions

PID control uses 3 coefficients for each angle adjustment, and a total of 9 coefficients are required to control four propellers. In the latent layer, an NN with 3 neurons requires 28 coefficients to control four propellers. As the number of latent and latent neurons increases, so does the number of coefficients.

The results of the training using the initial training data and the randomly mixed data show that the neural network is learning the same regardless of the sequence of the training data.

There were no significant changes in the results of neural network training when the number of latent layers was trained in one or two phases.

Contact

Bat-Erdene Gotov
Mongolian National University of Medical Sciences
Bayangol district 6th khoroo, Ard Ayush street P.O.Box-188, Ulaanbaatar-26, Mongolia
Baterdene.g@mnums.edu.mn
+976 88108026

References

1. M. Lower and W. Tarnawski, "Quadrotor Navigation Using the PID and Neural Network Controller," Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
2. A. Zulu*, S. John, "A Review of Control Algorithms for Autonomous Quadrotors", Open Journal of Applied Sciences, 2014, 4, 547-556
3. H. Boudjedir , O.Bouhali and N. Rizoug, "Neural Network Control Based on Adaptive Observer for Quadrotor Helicopter", International Journal of Information Technology, Control and Automation (IJITCA) Vol.2, No.3, July 2012
4. H. Boudjedir , O.Bouhali and N. Rizoug, "Adaptive neural network control based on neural observer for quadrotor unmanned aerial vehicle", Advanced Robotics, 2014, Vol.28, No.17, 1151–1164, http://dx.doi.org/10.1080/01691864.2014.913498
5. T. Dierks and S. Jagannathan, "Neural Network Control of Quadrotor UAV Formations", 2009 American Control Conference Hyatt Regency Riverfront, St. Louis, MO, USA June 10-12, 2009
6. Holger Voos, "Nonlinear and Neural Network-based Control of a Small Four-Rotor Aerial Robot", 2007 IEEE/ASME international conference on advanced intelligent mechatronics. DOI: 10.1109/AIM.2007.4412470
7. Yan-Fei Teng, Bin Hu, Zhi-Wei Liu, Jian Huang, and Zhi-Hong Guan, "Adaptive Neural Network Control for Quadrotor Unmanned Aerial Vehicles", 2017 11th Asian Control Conference (ASCC) Gold Coast Convention Centre, Australia December 17-20
8. X Yu, Z Lv, Y Wu, XM Sun, "Neural network modeling and backstepping control for quadrotor", Chinese Automation Congress (CAC), 2018 DOI: 10.1109/CAC.2018.8623432
9. C. Nicol1,C.J.B. Macnab, A. Ramirez-Serrano, "Robust Neural Network Control Of A Quadrotor Helicopter" Canadian Conference on Electrical and Computer Engineering, 2008. DOI: 10.1109/CCECE.2008.4564736
10. Xumei Lin,Yulu Wang,Yunfei Liu, "Neural-network-based robust terminal sliding-mode control of quadrotor", Chinese Automatic Control Society and John Wiley & Sons Australia, 2020 https://doi.org/10.1002/asjc.2478