International Journal of Power Electronics and Drive System (IJPEDS)
Vol. 14, No. 2, June 2023, pp. 799∼808
ISSN: 2088-8694, DOI: 10.11591/ijpeds.v14.i2.pp799-808

Feedforward neural network emulation of a PID continuous-time controller for quadcopter attitude digital control

Baba Mohammed[1], Bennaceur Said[2,3], Bounaama Fateh[2]
[1]Laboratory of Smart Grids and Renewable Energies (SRGE), University Tahri Mohammed Béchar, Béchar, Algeria
[2]Laboratory of Energy in Arid Zones (ENERGARID), University Tahri Mohammed Béchar, Béchar, Algeria
[3]Laboratory of Renewable Energies and their Applications in Saharan areas (LDREAS), University Tahri Mohammed Béchar, Béchar, Algeria

Article Info

Article history:

Received Oct 12, 2022
Revised Jan 19, 2023
Accepted Feb 13, 2023

Keywords:

Artificial intelligence
FFNN controller
Modeling
PID controller
Quadcopter system
Stabilization

ABSTRACT

Quadcopters are popular UAVs owing to their compact size and maneuverability. Quadcopters are unmanned aircraft guided by remote control, and the demand for them is increasing due to their widespread surveillance, goods delivery, aerial photography, and defense applications. Nonlinear quadcopter operation makes control system implementation very challenging. In this paper, based on artificial intelligence (AI), we train a feedforward neural network (FFNN) controller of a traditional proportional integral derivative (PID). The conventional (PID) is generally tuned to improve the quadcopter control and performance. FFNN can perform offline learning between the inputs and outputs of the controller to learn its behavior. Once the learning is complete, we replace the PID controller with the neural network controller, to get a controller that can maintain system stability, and overcome the limitations of hardware implementation problems caused by the classical PID controller.

Corresponding Author:

Baba Mohammed
Laboratory of Smart Grids and Renewable Energies (SRGE), University Tahri Mohammed Béchar
Béchar 08000, Algeria
Email: baba.mouhamed@univ-bechar.dz

1. INTRODUCTION

A quadcopter is an aircraft intended to perform certain tasks, without a pilot on board. Quadcopters are currently used in many military missions, civilian, photographing, and other applications that may endanger human lives [1], [2], the increasing use and applications of quadcopters have opened up opportunities for extensive and diverse research efforts to develop them. Environmental factors impacting Flight, which limit control time, have confused aviation engineers for ages. Many quadcopter control designs have been proposed. A collection of research offered approaches for a quadcopter controller employing linear control by making the quadcopter working mechanism linear at the working point and in the case of a quadcopter hovering [3]-[7], but these methods did not achieve stability when moving away from the working point, especially when the quadcopter is exposed to external disturbances. This prompted researchers to use nonlinear control techniques such as the backstepping method [8]-[11], and sliding mode [12]-[15], which requires system knowledge. A Gaussian process, for example, is one learning technique that has been used in certain studies to actively estimate the robot's dynamics parameters and update the prediction model in real time [16]-[18]. Errors in model and system parameters or model parameters might reduce controller performance or cause system instability.

Time-varying system characteristics produce many problems these models can't overcome. Indirect adaptive control methods like least-squares [19] fix coefficient errors based on the difference between reference and response, but not the coefficients.

This study models the quadcopter's dynamics and motion to obtain flight equations. feedforward neural network (FFNN) controls roll, pitch, yaw, and altitude based on proportional integral derivative (PID) input and output. MATLAB Simulink was used to simulate and display the controlled model.

2. QUADCOPTER CONFIGURATION

Quadcopter consists of two pairs of identical propellers, two rotating clockwise and two counter-clockwise. These propellers are connected to motors driven by electronic speed controllers. By varying the speed of each rotor, a quadcopter can generate the desired total thrust and torque. Quadcopter has a micro-processor attached to an accelerometer and a gyroscope for orientation. A receiver linked to a quadcopter's microprocessor sends instructions, forces, and moments. Figure 1 shows the configuration of different motions for the quadcopter.

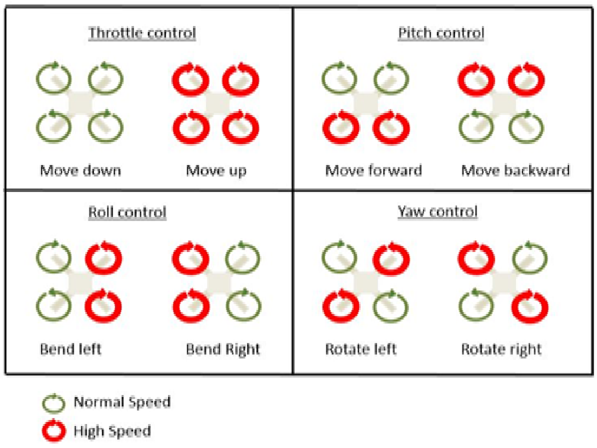

Figure 1. Different motions for the quadcopter [20]

3. QUADCOPTER MODELING

Modeling is the first and most significant step in designing an object's control strategy since it helps comprehend body specifications and constraints that limit control command application. Despite its mechanical symmetry, a quadcopter is a non-linear model, for this, the Newton-Euler method was used to model quadcopter [21].

3.1. State space variables

The position of a quadcopter around each of the three axes of rotation is determined by six functions: Euler angles $\begin{bmatrix}\phi & \theta & \psi\end{bmatrix}$ (roll, pitch, and yaw) and angular velocity $\begin{bmatrix}p & q & r\end{bmatrix}$ relative to the body frame. There are 6 other necessary functions. The position of the center of gravity $\begin{bmatrix}X & Y & Z\end{bmatrix}$ and linear velocity components $\begin{bmatrix}u & v & w\end{bmatrix}$ relative to the inertial frame. In sum, the quadcopter has 12 state variables describing 6 degrees of freedom.

$$x = \begin{bmatrix}XYZuvw\phi\theta\psi pqr\end{bmatrix}^T \tag{1}$$

3.2. Inputs variables

There are four main inputs through which we can control the quadcopter system $\begin{bmatrix}T_{\sum} & M_1 & M_2 & M_3\end{bmatrix}^T$ (altitude, roll, pitch, yaw) can be determined by delivering thrust to each rotor, the inputs variables are defined by the following equations [22]. The motor's thrust is perpendicular to its plane of revolution and proportionate to its rotational speed.

$$T_\Sigma = \sum_{i=1}^{i=N} T_i = b \sum_{i=1}^{i=N} \omega_i^2 \tag{2}$$

Only the fan speeds can directly effect the quadcopter's movement, so we must relate them to torques M_1, M_2, M_3.

$$M_1 = -\sum l(T_2 - T_4) = -bl \sum (\omega_2^2 - \omega_4^2) \tag{3}$$

$$M_2 = \sum l(T_1 - T_3) = bl \sum (\omega_1^2 - \omega_3^2) \tag{4}$$

$$M_3 = \sum (-T_1 + T_2 - T_3 + T_4) = d \sum (-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2) \tag{5}$$

Where T_i is the thrust of the rotor I where $i = \{1, 2, 3, 4\}$,$N = 4$, the (-) sign signifies lift force is opposing body frame vertical axis.

3.3. Equations of motion

The equations of motion for a quadcopter are typically stated in an inertial frame, which utilizes a rotation matrix R [23], for the following reasons:

- The inertia matrix is not related to time.
- The symmetry of the quadcopter body can be used to simplify the equations.
- Sensors' measurements and thrust forces are expressed naturally in the body frame.

3.4. Translation motion

According to Newton's second law, a quadcopter's frame motion equations are:

$$ma = \sum F \tag{6}$$

$$m\ddot{\zeta} = RT_{\sum} I_3 + mgl_3 \tag{7}$$

Where: $\ddot{\zeta}$ Is linear acceleration: $\ddot{\zeta} = [\ddot{X}\ddot{Y}\ddot{Z}]$.

$$l_3 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \tag{8}$$

R Rotation matrix allows converting linear velocity from body frame to inertial frame:

$$\dot{\zeta} = R.\eta \tag{9}$$

$$R = \begin{bmatrix} c\psi c\theta & c\psi s\theta s\phi - c\phi s\psi & c\psi s\theta c\phi + s\psi s\phi \\ s\psi c\theta & s\psi s\theta s\phi + c\psi c\phi & s\psi s\theta c\phi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\theta c\phi \end{bmatrix} \tag{10}$$

where c is $\cos$ and s is $\sin$. Substituting the relations and expressing them in the inertial frame obtains the quadcopter's linear motion equations:

$$\ddot{X} = -\frac{1}{m}(\sin\psi.\sin\phi + \cos\psi.\sin\theta.\cos\phi).T_\Sigma \tag{11}$$

$$\ddot{Y} = -\frac{1}{m}(-\cos\psi.\sin\phi + \sin\psi.\sin\theta.\cos\phi).T_\Sigma \tag{12}$$

$$\ddot{Z} = -\frac{1}{m}.\cos\theta.\cos\phi.T_\Sigma + g \tag{13}$$

3.5. Rotational motion

Newton's formalism makes it possible to express the equations of motion in the inertial frame according to the Euler angles, such as:

$$I\dot{\omega} = M - \omega \times I\omega \tag{14}$$

where M the torques applied to the quadcopter: $M = \begin{bmatrix} M_1 & M_2 & M_3 \end{bmatrix}^T$. I is the inertia matrix defined by (15).

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{15}$$

The matrix W_η^{-1} makes it possible to transform the angular velocity ω from the body frame to the inertial frame:

$$\dot{\eta} = W_\eta^{-1}.\omega \tag{16}$$

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & s\theta c\theta \\ 0 & -s\phi & c\phi c\theta \end{bmatrix} . \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \tag{17}$$

by substituting the relations and writing them in the inertial frame, we find the equations of angular motion of the quadcopter:

$$\dot{p} = \frac{qr.(I_{yy} - I_{zz})}{I_{xx}} + \frac{1}{I_{xx}}.M_1 \tag{18}$$

$$\dot{q} = \frac{qr.(I_{zz} - I_{xx})}{I_{yy}} + \frac{1}{I_{yy}}.M_2 \tag{19}$$

$$\dot{r} = \frac{qr.(I_{xx} - I_{yy})}{I_{zz}} + \frac{1}{I_{zz}}.M_3 \tag{20}$$

now we have a summary of the nonlinear quadcopter state-space model as:

$$\dot{x} = f(x, u) \tag{21}$$

where x is the state variables vector and u is the state inputs vector.

$$x = \begin{bmatrix} X\dot{X}Y\dot{Y}Z\dot{Z}\phi\theta\psi pqr \end{bmatrix}^T \tag{22}$$

So:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \\ \dot{x}_{10} \\ \dot{x}_{11} \\ \dot{x}_{12} \end{bmatrix} = \begin{bmatrix} x_2 \\ -\frac{1}{m}(sx_{11}.sx_7 + cx_{11}.sx_9.cx_7).U_1 \\ x_4 \\ -\frac{1}{m}(-cx_{11}.sx_7 + sx_{11}.sx_9.cx_7).U_1 \\ x_6 \\ -\frac{1}{m}.cx_9.cx_7.U_1 + g \\ x_{10} + sx_7.\tan x_8.x_{11} + cx_7.\tan x_8.x_{12} \\ cx_7.x_{11} - sx_7.x_{12} \\ sx_7.\sec x_8.x_{11} + cx_7.\sec x_8.x_{12} \\ \frac{x_1 0.x_1 2.(I_{yy} - I_{zz})}{I_{xx}} + \frac{1}{I_{xx}}.U_2 \\ \frac{x_{10}.x_{12}.(I_{zz} - I_{xx})}{I_{yy}} + \frac{1}{I_{yy}}.U_3 \\ \frac{x_{10}.x_{11}.(I_{xx} - I_{yy})}{I_{zz}} + \frac{1}{I_{zz}}.U_4 \end{bmatrix} \tag{23}$$

thus, we have obtained the basic equations that will be used in the simulation and design of the controller.

4. CONTROLLERS DESIGN

A process is usually controlled by a standard controller (PID and RST) [21], [22]. An identifier neural network may approximatively learn the controller's inputs and outputs offline as we see in Figure 2. The neural network controller replaces the conventional controller after learning. This architecture's purpose is not to improve the performance of the conventional controller, but to overcome hardware implementation limits.

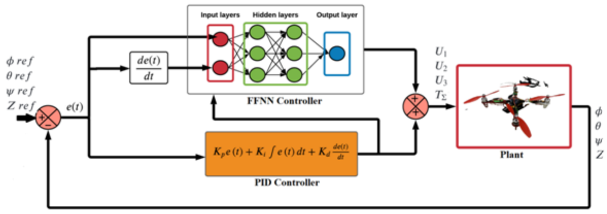

Figure 2. PID and FFNN controllers design

4.1. PID controller

In this paper, we designed PID controllers for yaw, pitch, roll, and altitude. His proportional, integrative, and differential coefficients were adjusted.

$$T_\Sigma = mg - [K_{P,\dot{Z}}(\dot{Z}^{des} - \dot{Z}) + K_{I,\dot{Z}} \int (\dot{Z}^{des} - \dot{Z}) + K_{D,\dot{Z}}(\ddot{Z}^{des} - \ddot{Z})] \tag{24}$$

$$M_1 = K_{P,\phi}(\phi^{des} - \phi) + K_{I,\phi} \int (\phi^{des} - \phi) + K_{D,\phi}(p^{des} - p) \tag{25}$$

$$M_2 = K_{P,\theta}(\theta^{des} - \theta) + K_{I,\theta} \int (\theta^{des} - \theta) + K_{D,\theta}(q^{des} - q) \tag{26}$$

$$M_3 = K_{P,\psi}(\psi^{des} - \psi) + K_{I,\psi} \int (\psi^{des} - \psi) + K_{D,\psi}(r^{des} - r) \tag{27}$$

The tuning method was applied separately to each axis over P, I, and D gains. The manual method (tweaking until a required response is achieved) is selected. Although there are several adjustment methods available. As a result, we obtained the following gains in Table 1.

Table 1. PID controller parameters

Parameters	P	I	D
Roll (Φ)	0.35	0.0	0.2
Pitch (Θ)	0.45	0.0	0.3
Yaw (Ψ)	0.8	0.0	0.3
Altitude (Z)	10.8	8.0	0.0

4.2. Neural network controller

Intelligent controllers utilizing artificial intelligence and neural networks can greatly improve automation systems [23]. The neural network acquires knowledge through training and stores it through synaptic weights, which connect neurons. Each neuron's weight is related to the following layers. Adjusting synaptic weights trains the network to perform various activities. Layered neurons have hidden layers, an input layer, and an output layer. Figure 3 shows that each neuron's output is determined by two successive arithmetic computations. The first includes determining the product of the input variables of the previous layer P R by the synaptic weights $W(1, R)$, then adding the values of displacement b to them in the second process. The output α of the neuron is determined using the result of the first process, as a variable of the neuron activation function f.

$$a = f(W_p + b) \tag{28}$$

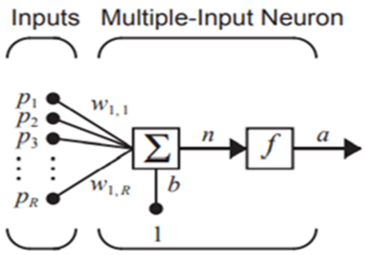

Figure 3. Neuron structure

4.2.1. Neural network controller architecture

We create an FFNN control with three layers (input, hidden, and output), and four output controls consisting of pitch, roll, yaw, and altitude control. Then we designed this later by learning the behavior of an existing PID controller. The results of the PID controller are chosen as a target in the formulation of the direct control FFNN for the quadcopter model. This approach allows us to achieve efficient control while freeing us from limitations in terms of digital PID implementation.

Hardware implementations of neural networks can be viewed as a collection of processing units that work together in parallel. These units have a basic internal structure, which may include a limited amount of local memory. There are various types of hardware implementations for neural networks [24]:

– Analogue implementation: it is possible to leverage the physical characteristics of silicon devices to operate neural networks in an analog manner, resulting in extremely fast processing speeds. to compensate for parameter variations in temperature, manufacturing conditions, etc. One approach is to implement neurons using common operational amplifiers and resistors.
– Digital implementation: the digital neural network category encompasses many subcategories including slice architectures, single instruction multiple data (SIMD) approach, systolic array devices, radial basis function (RBF) architectures, ASIC and FPGA implementations [25].
– Hybrid implementation: hybrid design attempts to combine the advantages of analog and digital techniques.
– Optical implementation: the speed of hardware implementation is determined by the technology that is available. Optical implementations, such as photonics and data storage, result in the fastest performance, while electronic digital architectures produce the slowest results [26]. This type of implementation is still in the research stage, but it has the potential to be much faster than traditional digital implementations.

4.2.2. Construction and training steps of FFNN controller

To achieve the construction of the proposed FFNN controller, we used MATLAB to program the network-building process using the neural network toolbox. This requires specifying the following:

– Entering data arranged in a matrix form of 8 rows with 6000 lengths as inputs and 4 rows with 6000 lengths as outputs, these data represent different values of network input variables, corresponding to error values, and derivative of error, in addition, Target data represent the output of the PID controller.
– The entered data was automatically divided into three groups: the training group, the validation group, and the test group, with a percentage of 0 for the training group and 15 for the two groups of verification and testing.
– Then choose the trainml function representing the Marquardt Levenberg inverse order algorithm. It is considered the most widespread and widely used in training ANN networks because it has a high speed in reaching the optimal solution corresponding to achieving an average minimum error.
– The number of nodes in the hidden layer was determined by experiments, by changing the number of neural connections in this layer and training the network to obtain the optimal performance parameters of the trained network.

Figure 4 shows FFNN controller architecture and the final model in Figure 5.

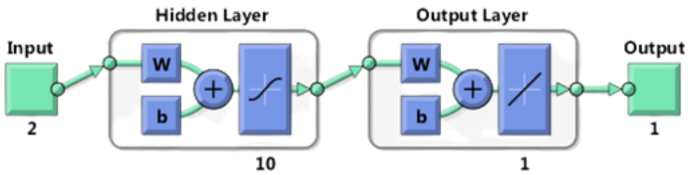

Figure 4. FFNN architecture of two inputs, one output, and ten hidden layers

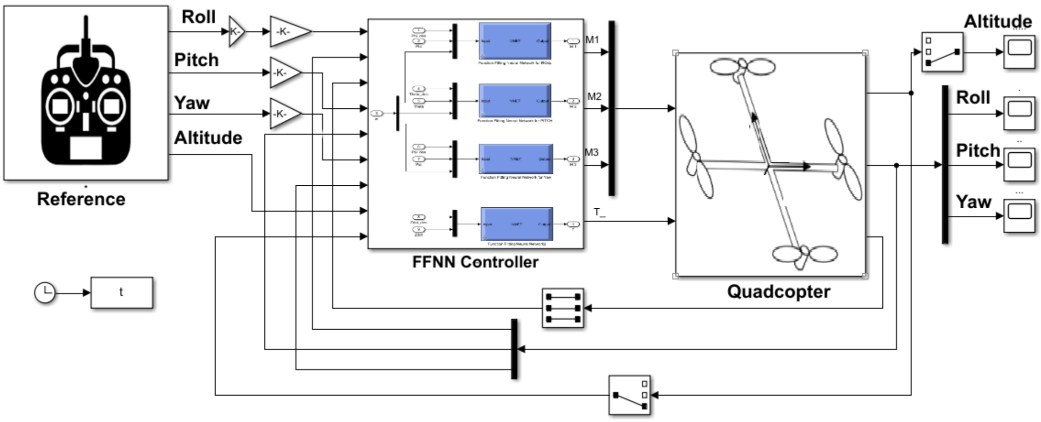

Figure 5. Structure of full model in Simulink

5. RESULTS AND DISCUSSION

A simulation program is planned to associate the steady and dynamic performances of the quadcopter control system. Simulation results are presented in the following figures. The time response of different controllers are presented in Figures 6-9, respectively. We can see there is good tracking of desired angles and altitude. The signal of the controllers (roll, pitch, yaw, and altitude) is shown in Figures 10-13, respectively. It appears clearly from the previous figures that the neural network controller results are completely similar to the results of the PID. This indicates that the neural network has learned well and, in addition, it has reduced certain overshoots that appear in the results of the PID controller. It can be said that the use of neural network controllers in our research resulted in better performance.

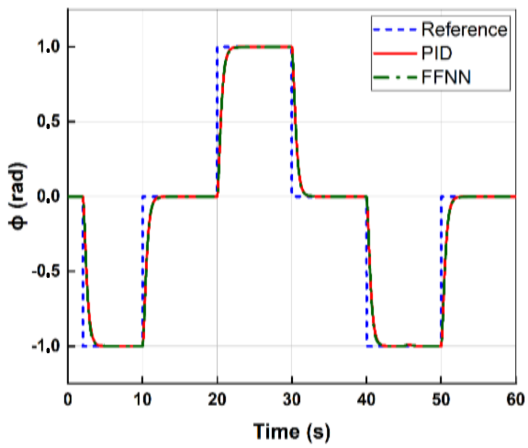

Figure 6. The roll angle response w

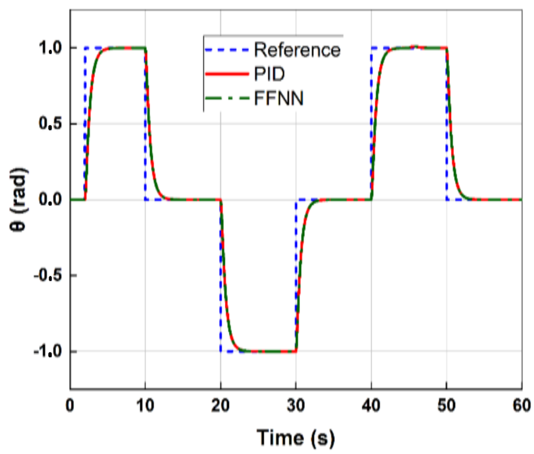

Figure 7. The pitch angle response

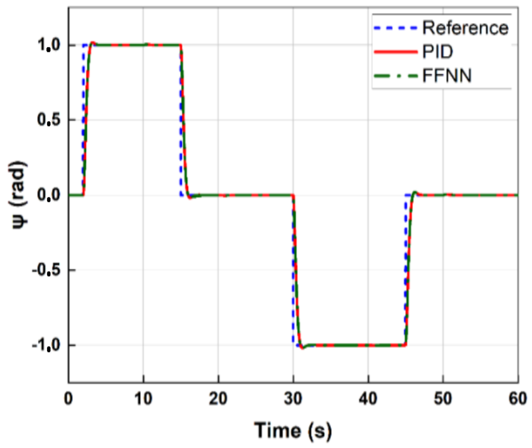

Figure 8. The yaw angle response

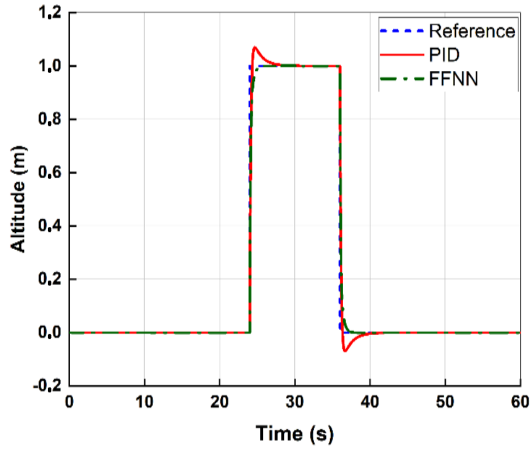

Figure 9. The altitude response

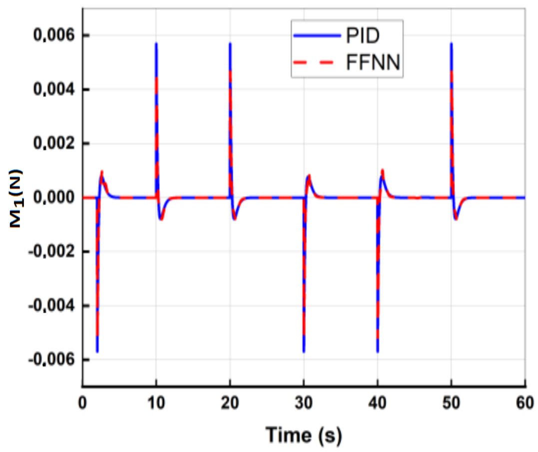

Figure 10. The control signal M_1

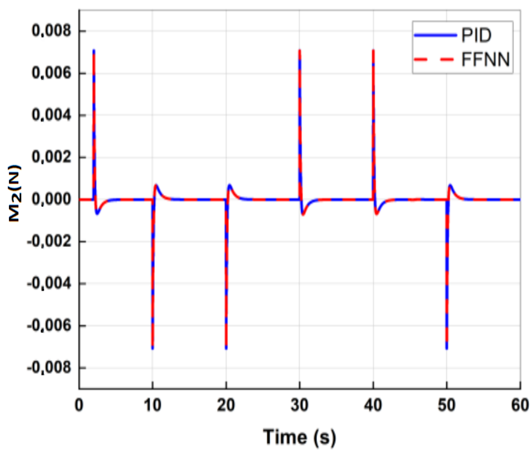

Figure 11. The control signal M_2

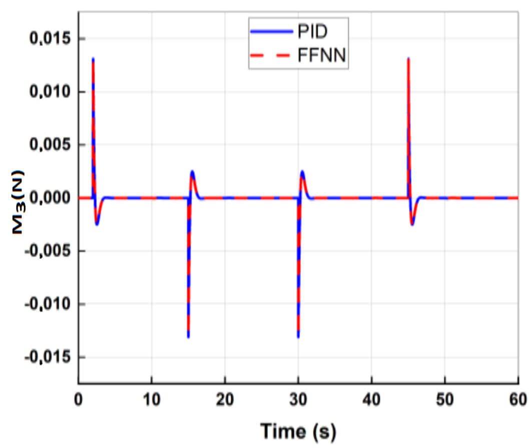

Figure 12. The control signal M_3

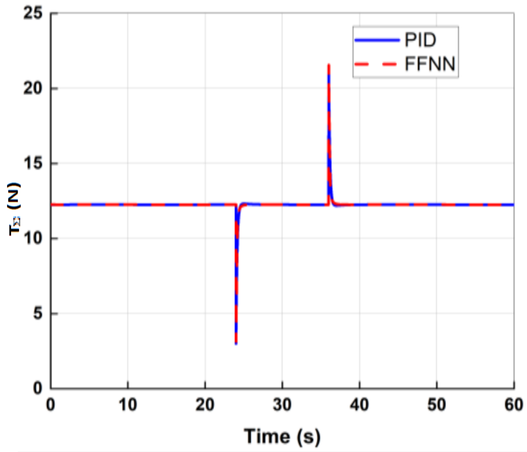

Figure 13. The control signal T_Σ

6. CONCLUSION

In conclusion, it can be said that the use of the neural network has contributed to the realization of the control of the quadcopter model, in addition, the following can be concluded: i) the design of the advanced FFNN is simple and easy. If sufficient training data is available it is possible to build easily a FFNN with the correct selection of the artificial neural network architecture; ii) the quadcopter was able to maintain a stable attitude; iii) the quadcopter was able to respond quickly and accurately; and iv) the quadcopter was able to achieve a high level of performance, with minimal overshoot or undershoot in its thrust, roll, pitch, and yaw control, indicating that the controller was able to optimize its control strategy for maximum performance. The work can be complemented using other advanced control techniques such as control fuzzy logic and adaptive neuro-fuzzy inference systems (ANFIS) control.

ACKNOWLEDGEMENT

The authors thank the Algerian research direction DGRSDT, and the Smart Grids and Renewable Energies Laboratory for supporting this research.

REFERENCES

[1] M. Hassanalian and A. Abdelkefi, "Classifications, applications, and design challenges of drones: Aa review," *Progress in Aerospace Sciences*, vol. 91, pp. 99-131, 2017, doi: 10.1016/j.paerosci.2017.04.003.

[2] A. Mairaj, A. I. Baba, and A. Y. Javaid, "Application specific drone simulators: recent advances and challenges," *Simulation Modelling Practice and Theory*, vol. 94, pp. 100-117, 2019, doi: 10.1016/j.simpat.2019.01.004.

[3] A. Noordin, M. A. M. Basri, and Z. Mohamed, "Simulation and experimental study on PID control of a quadrotor MAV with perturbation," *Bulletin of Electrical Engineering and Informatics*, vol. 9, no. 5, pp. 1811-1818, 2020, doi: 10.11591/eei.v9i5.2158.

[4] H. Suresh, A. Sulficar, and K. Desai, "Design of adaptive control algorithm for quadrotor with multirotor configuration," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, no. 7S, pp. 368-371, 2019, doi: 10.35940/ijitee.K4061.0787S19.

[5] M. N. Shauqee, P. Rajendran, and N. M. Suhadis, "An effective proportional-double derivative-linear quadratic regulator controller for quadcopter attitude and altitude control," *Automatika*, vol. 62, no. 3-4, pp. 415-433, 2021, doi: 10.1080/00051144.2021.1981527.

[6] F. Rinaldi, S. Chiesa, and F. Quagliotti, "Linear quadratic control for quadrotors UAVs dynamics and formation flight," *Journal of Intelligent and Robotic Systems*, vol. 70, pp. 203–220, 2013, doi: 10.1007/s10846-012-9708-3.

[7] D. A. Mercado, R. Castro, and R. Lozano, "Quadrotors flight formation control using a leader-follower approach," in *2013 European Control Conference (ECC)*, 2013, pp. 3858-3863, doi: 10.23919/ECC.2013.6669637.

[8] A. T. Nguyen, N. Xuan-Mung, and S. Hong, "Quadcopter adaptive trajectory tracking control: a new approach via backstepping technique," *Applied Sciences*, vol. 9, no. 18, p. 3873, 2019, doi: 10.3390/app9183873.

[9] M. Lungu, "Auto-landing of UAVs with variable centre of mass using the backstepping and dynamic inversion control," *Aerospace Science and Technology*, vol. 103, p. 105912, 2020, doi: 10.1016/j.ast.2020.105912.

[10] W. Xie, D. Cabecinhas, R. Cunha, and C. Silvestre, "Adaptive backstepping control of a quadcopter with uncertain vehicle mass, moment of inertia, and disturbances," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 1, pp. 549–559, 2022, doi: 10.1109/TIE.2021.3055181.

[11] A. K. Bhatia, J. Jiang, Z. Zhen, N. Ahmed, and A. Rohra, "Projection modification based robust adaptive backstepping control for multipurpose quadcopter UAV," *IEEE Access*, vol. 7, pp. 154121–154130, 2019, doi: 10.1109/ACCESS.2019.2946416.

[12] H. L. N. N. Thanh and S. K. Hong, "Quadcopter robust adaptive second order sliding mode control based on PID sliding surface," *IEEE Access*, vol. 6, pp. 66850–66860, 2018, doi: 10.1109/ACCESS.2018.2877795.

[13] N. P. Nguyen and S. K. Hong, "Fault-tolerant control of quadcopter uavs using robust adaptive sliding mode approach," *Energies*, vol. 12, no. 1, pp. 95, 2019, doi: 10.3390/en12010095.

[14] V. K. Tripathi, L. Behera, and N. Verma, "Design of sliding mode and backstepping controllers for a quadcopter," in *2015 39th National Systems Conference (NSC)*, 2015, pp. 1-6, doi: 10.1109/NATSYS.2015.7489097.

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

[16] M. Mehndiratta and E. Kayacan, "Gaussian process-based learning control of aerial robots for precise visualization of geological outcrops," in *2020 European Control Conference (ECC)*, 2020, pp. 10-16, doi: 10.23919/ecc51009.2020.9143655.

[17] J. Caldwell and J. A. Marshall, "Towards efficient learning-based model predictive control via feedback linearization and Gaussian process regression," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4306-4311, doi: 10.1109/IROS51168.2021.9636755

[18] K. Y. Chee, T. Z. Jiahao, and M. A. Hsieh, "KNODE-MPC: a knowledge-based data-driven predictive control framework for aerial robots," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2819-2826, Apr. 2022, doi: 10.1109/LRA.2022.3144787.

[19] M. Dhaybi and N. Daher, "Real-time estimation of the inertia tensor elements of a quadcopter hover platform," in *2019 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, 2019, pp. 1347-1352, doi: 10.1109/AIM.2019.8868641.

[20] G. P. Rible, N. A. Arriola, and M. Ramos, "Modeling and implementation of quadcopter autonomous flight based on alternative methods to determine propeller parameters," *Advances in Science, Technology and Engineering Systems*, vol. 5, no. 5, pp. 727-741, Oct. 2020, doi: 10.25046/AJ050589.

[21] A. Alkamachi and E. Ercelebi, "A proportional derivative sliding mode control for an overactuated quadcopter," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 2019, vol. 233, no. 4, pp. 1354-1363, doi: 10.1177/0954410017751739.

[22] I. Siti, M. Mjahed, H. Ayad, and A. El Kari, "New trajectory tracking approach for a quadcopter using genetic algorithm and reference model methods," *Applied Sciences*, vol. 9, no. 9, p. 1780, Apr. 2019, doi: 10.3390/app9091780.

[23] B. Aissa, T. Hamza, G. Yacine, and N. Mohamed, "Impact of sensorless neural direct torque control in a fuel cell traction system," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 4, pp. 2725–2732, Aug. 2021, doi: 10.11591/ijece.v11i4.pp2725-2732.

[24] S. J. Hammoodi, K. S. Flayyih, and A. R. Hamad, "Design and implementation speed control system of DC motor based on PID control and MATLAB simulink," *International Journal of Power Electronics and Drive Systems*, vol. 11, no. 1, pp. 127–134, 2020, doi: 10.11591/ijpeds.v11.i1.pp127-134.

[25] M. A. Ibrahim, A. K. Mahmood, and N. S. Sultan, "Optimal PID controller of a brushless DC motor using genetic algorithm," *International Journal of Power Electronics and Drive Systems*, vol. 10, no. 2, pp. 822–830, 2019, doi: 10.11591/ijpeds.v10.i2.pp822-830.

[26] H. Park and S. Kim, "Hardware accelerator systems for artificial intelligence and machine learning," *Advances in Computers*, vol. 122, pp. 51–95, Jan. 2021, doi: 10.1016/BS.ADCOM.2020.11.005.

BIOGRAPHIES OF AUTHORS

Baba Mohammed received his B.Sc. from Ahmed Draya University, Adrar, Algeria, in 2017, and M.Sc. degrees in Embedded Systems from Ferhat Abbas 1 University, Setif, Algeria, in 2019, respectively. Currently, he is a Ph.D. student at Smart Grids, and Renewable energies Laboratory, Tahri Mohamed, Béchar, Algeria. His research interests include quadcopter systems, system modeling and control, and AI control. He can be contacted at email: baba.mouhamed@univ-bechar.dz.

Bennaceur Said was born in Oran, Algeria, in 1969. He obtained a B.Sc. physics from the higher school of teachers in Saida in 1993, the Magister degree from the University of Béchar in 2006 with a specialty in energy physics and the Ph.D. from the University of Béchar in 2017 and 2021 respectively. His research areas are control, renewable energies and thermal systems. He can be contacted at email: bennaceur.said@univ-bechar.dz.

Bounaama Fatah received the Electronics Engineering degree from the University of Sciences and Technology of Oran (Algeria) in June 1988 and an M.Eng. and Ph.D. from the University of Béchar (Algeria) in 2001 and 2010 respectively. His scientific interests are simulation, identification, control, and the implementation of embedded system applications in the agriculture and horticulture environment of a greenhouse. He can be contacted at email: fbounaama2002@yahoo.fr.