American Scientific Research Journal for Engineering, Technology, and Sciences (ASRJETS)

ISSN (Print) 2313-4410, ISSN (Online) 2313-4402

http://asrjetsjournal.org/

Neural Network Control for Quadrotors

Osman Çakir[a*], Tolga Yüksel[b]

[a]*Karşıyaka Tüpraş Vocational and Technical Anatolian High School, Başiskele, Kocaeli, Turkey*

[b]*Dept. of Electrical and Electronics Engineering, Bilecik Şeyh Edebali University, Bilecik, Turkey*

[a]*Email: cakirosman41@hotmail.com*

[b]*Email: tolga.yuksel@bilecik.edu.tr*

Abstract

While quadrotors are becoming more popular, their controllers should be improved. In this study, neural network control of quadrotors is aimed to obtain an artificial intelligence based controller. Firstly, the quadrotor is modeled according to quadrotor dynamics. Then, PD controllers for *x*, *y*, yaw and *z* control of quadrotor are implemented as classical controllers. The results for these controllers are recorded as training data of NN controllers. As the proposed controllers, NN controllers are trained according to these data and performance of these results are examined. The results verify that NN controllers achieve good trajectory tracking results.

Keywords: Quadrotors; neural networks.

1. Introduction

The quadrotor is one of the most important unmanned air vehicles (UAVs) in the field of vertical Take-Off and Landing (VTOL) that can achieve a stable hovering and flight using the forces produced by four rotors. During the last decade, the interest in the quadrotors has been increased powerfully. Therefore, the design of flight controllers performing robust control for the quadrotors is an important issue for the fully autonomous vehicles design. Pound examined the flight dynamics and the dynamic model of a quadrotor in[1]. However, accurate dynamics models of quadrotors operating at higher speeds and in outdoor environments are derived difficulty. For this reason, control techniques based upon such models are critical for precision and trajectory tracking control. As a good UAV candidate, quadrotors have become ideal experimental platforms for the design of the aerial vehicle control.

----

* Corresponding author.

Different control methods for the attitude and position control of quadrotor have been investigated in the literature. Classical PD and PID controllers were applied to achieve the autonomous flight of the quadrotors in [2,3]. A quadrotor is a MIMO system affected by various uncertainties such as parametric uncertainties, nonlinear dynamics, and external disturbances. Therefore, it is hard for classical controllers to provide robust tracking performance. There are a variety of techniques that have been developed to reduce the side effects of the uncertainties in the rotational dynamics of the quadrotor. Fuzzy control[4,5], sliding-mode control [6,7], and robust control [8] techniques were applied for the quadrotor stabilization with uncertainties. In this study, neural networks (NNs) are considered as compensators of uncertainties for control of stabilization. As mentioned in [9], NNs are artificial intelligence tools for modeling and classification and they can also model nonlinearities without analytical methods. It can used as controllers for such systems with uncertainties as in [10,11]. In this study, neural network controllers are considered and they are applied for the control of a quadrotor.

The paper is organized as follows: the nonlinear mathematical model of the quadrotor is described step by step in Section 2. The designs of the proposed NN controllers are presented in Section 3. The simulation results are shown in Section 4 with analysis of trajectory tracking performance of controller. The conclusion is given in Section 5.

2. Dynamics of Quadrotor

As the most popular UAV in the last decade, an example of a real quadrotor named as Pelican from Asc Tec Corporation is shown in Figure 1. The fields of applications include reconnaissance, agricultural application, and robotics research. While fixed wing UAVs provide lifting by a propeller or jet motor and flaps, quadrotor provides this thrust by four propellers in each side of the quadrotor.

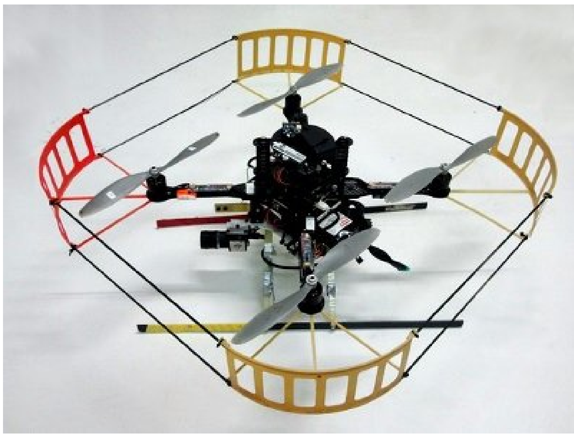


Figure 1: Pelican quadrotor

The notations of a quadrotor is shown in Figure 2 [12]. These notations include four rotors, their thrust vectors and directions of notation. It is also assumed as the fixed frame {B} of the body is attached to the quadrotor and

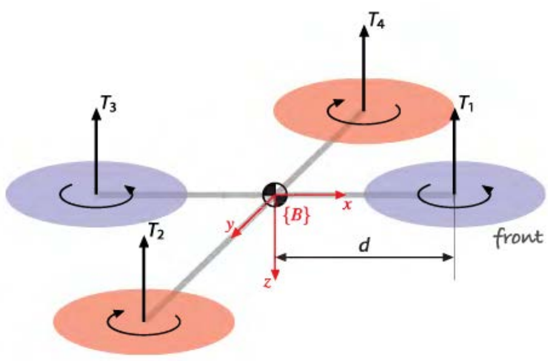


Figure 2: Notation for equations of motion of the quadrotor

has its origin at the quadrotor's center of mass. Directions of rotations of the rotors define the actions of the quadrotor. The rotors are driven by electric motors powered by electronic speed controllers. The speed of each rotor is defined asω_i and the thrust obtained from each rotor is defined using an upward vector

$$T_i = b\omega_i^2, \quad i = 1., \dots, 4. \tag{1}$$

in negative **z**-direction, where b>0 is the constant of lifting that depends on the rotor blade radius, the number of blades, the air density and the chord length of the blade.

The translational dynamics of the quadrotor is given by Newton's second law

$$m\dot{v} = \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} - {}^0R_B \begin{pmatrix} 0 \\ 0 \\ T \end{pmatrix} \tag{2}$$

where v is the quadrotor’s velocity in the world frame, g is gravitational acceleration, m is the total mass of the quadrotor and $T = \sum T_i$is the total upward thrust. The first term in (2) is the force of gravity which acts downward in the world frame and the second term is the total thrust in the quadrotor frame rotated into the world coordinate frame to define velocity in world coordinates.

The rotational acceleration is given by Euler's equation of motion

$$J\dot{\omega} = -\omega \times J\omega + \Gamma \tag{3}$$

where$J$ is the 3 x 3 inertia matrix of the quadrotor, ω is the angular velocity vector and $\Gamma = (\tau_x, \tau_y, \tau_z)^T$ is the torque applied to the airframe whereτ_x is the rolling torque, τ_y is the pitching torque and τ_z is the total reaction torque

$$\tau_x = dT_4 - dT_2 = db(\omega_4^2 - \omega_2^2) \tag{4}$$

$$\tau_y = db(\omega_1^2 - \omega_3^2) \tag{5}$$

$$\tau_z = k(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2) \quad (7)$$

By integrating the forward dynamics equations from Eq. 2 to Eq. 7, the motion of the quadrotor obtained where the forces and moments on the airframe.

$$\begin{pmatrix} T \\ \Gamma \end{pmatrix} = \begin{pmatrix} -b & -b & -b & -b \\ 0 & -db & 0 & db \\ db & 0 & -db & 0 \\ k & -k & k & -k \end{pmatrix} \begin{pmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{pmatrix} = A \begin{pmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{pmatrix} \quad (8)$$

are functions of the rotor speeds. The matrix A is of full rank and can be inverted

$$\begin{pmatrix} \omega_1^2 \\ \omega_2^2 \\ \omega_3^2 \\ \omega_4^2 \end{pmatrix} = A^{-1} \begin{pmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{pmatrix} \quad (9)$$

to give the rotor speeds required to apply a specified thrust T and moment Γ to the airframe.

3. Neural Network Controller Design for The Quadrotor

Controller designs for quadrotor control are mostly based on conventional PID controllers. In this study, we have designed NN controllers to obtain robustness against external effects. NN controllers are derived from PD controllers and simulation results verify the performance.

3.1 Classical PD controller for the quadrotor

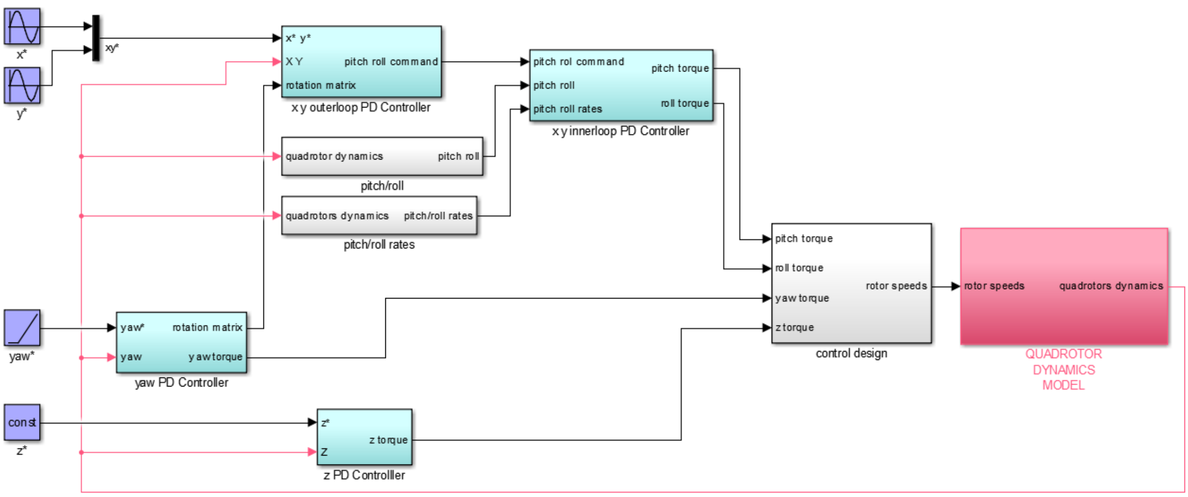


Figure 3: Block diagram of PD control of the quadrotor

In Figure 3, the block diagram of PD control of the quadrotor is given. *x**, *y**, *yaw** and *z** as reference signals of quadrotor are given externally. Four controllers calculate pitch, roll, yaw and *z* torque using PD control method. In the control design, the torques are converted to speed for four rotors. The orientation and position of the quadrotor is controlled by rotor speeds.

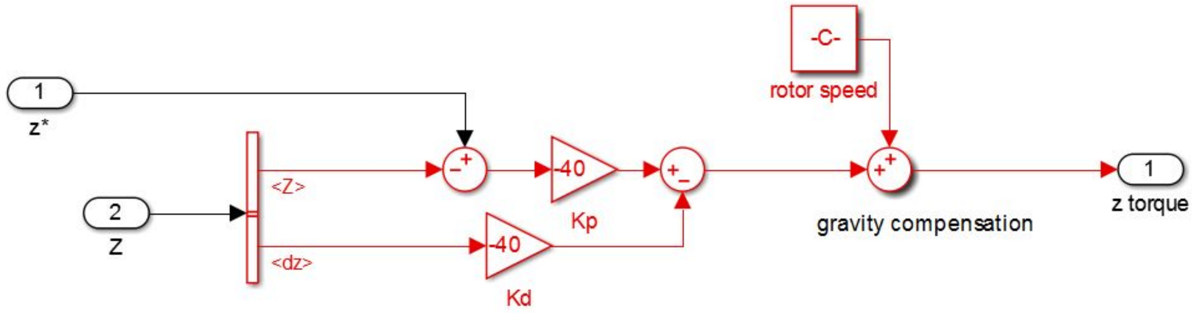


Figure 4: The inner loop of PD controller for *z*

As an example of these PD controllers, Figure 4 shows the inner loop of the quadrotor's PD controller for *z*. The error is obtained from the desired z^* and actual *z*. From the error, the error derivative term with gain multiplication is added to the PD controller. The control signal and the gravitational feed forward signal to compensate the gravity are summed, and the *z* torque value is calculated.

To define PD controllers, Figure 3 should be explained. The inner loop, shown in green, controls the attitude of the quadrotor. The actual and desired roll and pitch angles, as well as the roll and pitch angular rates are the inputs of this loop to provide damping. The outer loop controls the *xy*-position of the quadrotor by requesting changes in roll and pitch angle so as to provide a component of thrust in the direction of desired *xy*-plane motion and it is shown in gray. The PD controllers are defined as follows:

$$\tau_{roll} = \tau_x = K_{p,roll}(\theta^*_{roll} - \theta_{roll}) + K_{d,roll}(\dot{\theta}^*_{roll} - \dot{\theta}_{roll}) \quad (10)$$

$$\tau_{pitch} = \tau_y = K_{p,pitch}(\theta^*_{pitch} - \theta_{pitch}) + K_{d,pitch}(\dot{\theta}^*_{pitch} - \dot{\theta}_{pitch}) \quad (11)$$

$$\tau_{yaw} = \tau_z = K_{p,yaw}(\theta^*_{yaw} - \theta_{yaw}) + K_{d,yaw}(\dot{\theta}^*_{yaw} - \dot{\theta}_{yaw}) \quad (12)$$

$$T = K_p(z^* - z) + K_d(\dot{z}^* - \dot{z}) + \omega_0 \quad (13)$$

where$\omega_0 = \sqrt{mg/4b}$is the rotor speed necessary to generate a thrust to compensate and stabilize the weight of the quadrotor, $b > 0$ is the lift constant that depends on the radius of blades, air density and the number of blades. K_p and K_d gain constant are defined by the user to achieve good control performance.

3.2. Neural network controller for the quadrotor

Donald Hebb (1949) is known as the father of today's neural network theory. Hebb, a neurologist, has worked on how his brain learned. The studies begin by taking the nerve cell, the basic unit of the brain. He examines how the two nerve cells exhibit a correlation with each other and places the neural network theory on this basis.

By using these experiments, artificial neural network (ANN) is defined as a model of the human brain in which the nerve cells are layered and parallel with all the functions of the structure to be realized in the numerical world. The structure of ANNs are not explained here but can be found in [9] in details.

As the training phase of the NN controller instead of PD controllers, the training data is obtained from the inputs and outputs of PD controllers with trials one by one. According to the training data, each NN controller is designed for each PD controller. Subsequently, the trajectory of the quadrotor was traced by replacing the NN controller. An example of NN is shown in Figure 5 with hidden layer nonlinear function and output layer linear function.

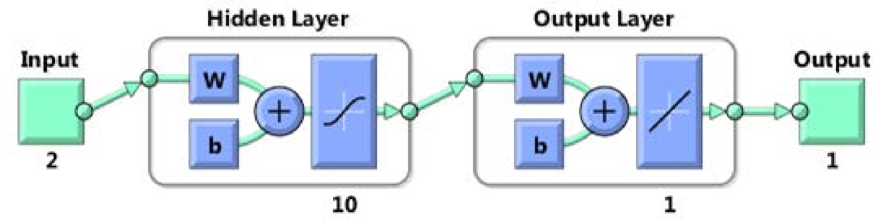


Figure 5: The internal structure of neural network

Figure 6 shows the block diagram of NN control of the quadrotor. NN controllers, quadrotor control and quadrotor dynamics blocks are shown in blue, gray and pink blocks, respectively. In Figure 7, the inner loop of the NN controller for *z* is shown. The error and error derivate of are given to the NN controller. By adding gravitational feed forward signal onto the control signal generated, a torque value is sent to the control design.

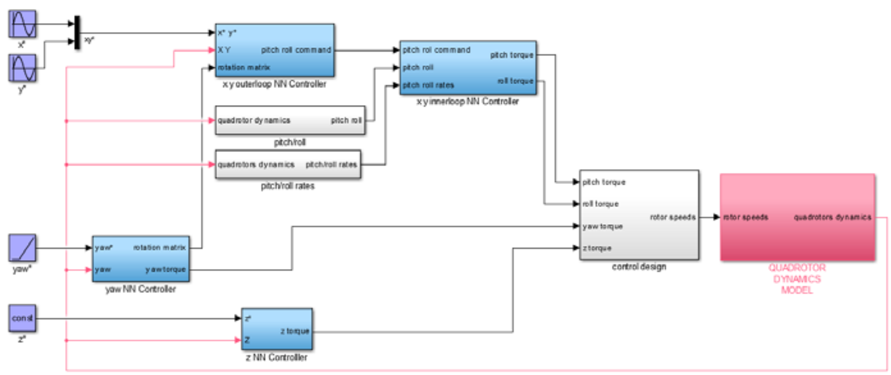


Figure 6: Block diagram of NN control of the quadrotor

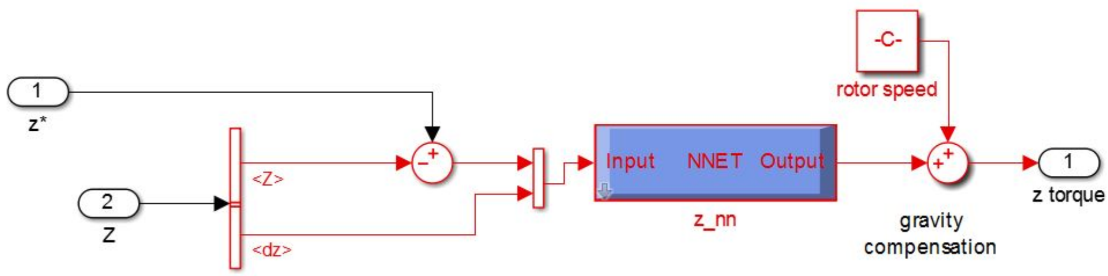


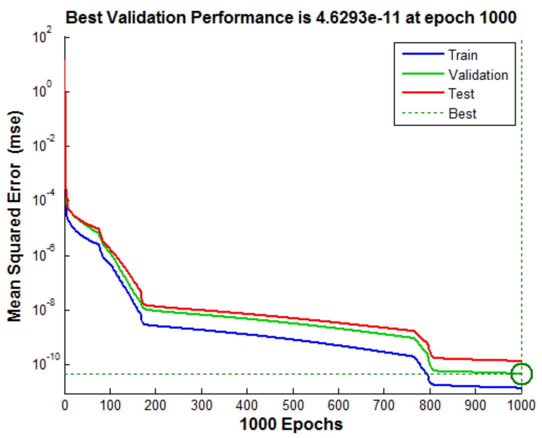
Figure 7: The inner loop of NN controller for *z*

4. Simulation Results

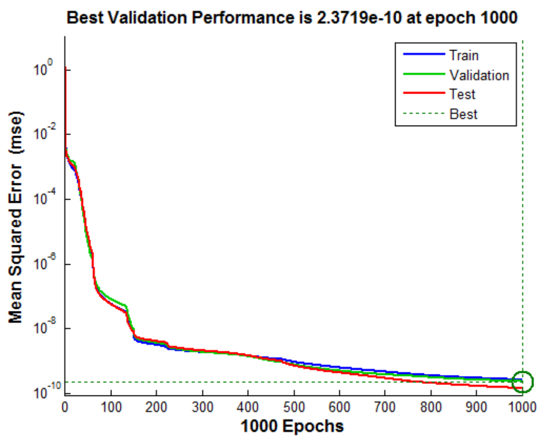
In this study, a quadrotor having 4 rotors and 4 blades is assumed. It is assumed that the gravity of the system is 9.81 m/s^2 and air density is 1,184 kg/m^3. The mass of the quadrotor is 4 kg, the wing length is 0.315 m, the rotor radius is 0.165 m, the wing width is 0.018 m and the wing mass is 0.005 kg.

All the simulations are implemented using MATLAB Simulink. Furthermore, NN controller design uses MATLAB *Neural Network Toolbox*[13]. All simulations are executed on a laptop with AMD Athlon Dual Core processor and 4 GB RAM.

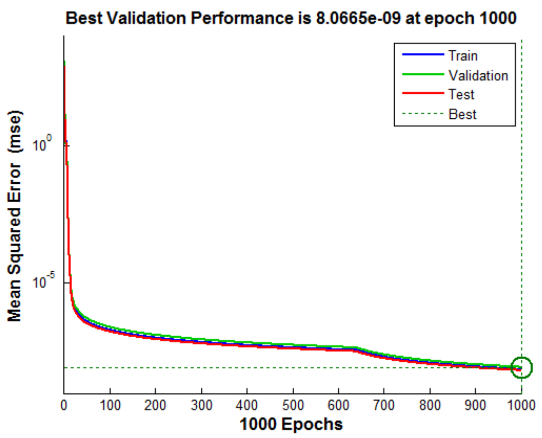
For each controller design, data are collected for different trajectories. In the following, 400x2 or 400x3 input and 400x1 output training data are obtained and used for NN trainings. All NN controllers are trained using the Levenberg-Marquardt algorithm. NN trainings are stopped with a maximum of 1000 epochs. Additionally, mean squared error (mse) type is selected.



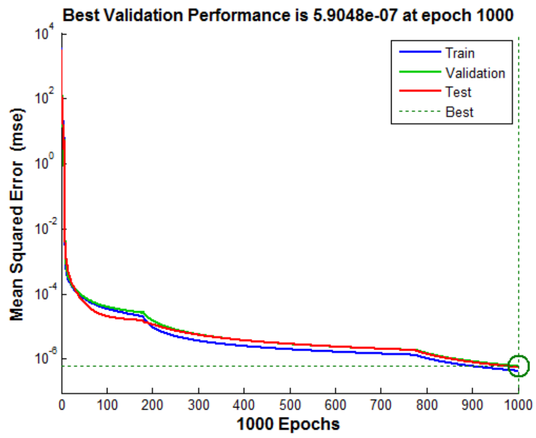
a)



b)



c)



d)

Figure 8: NN training results of controllers for a) Roll b) Pitch c) Yaw d) *z*

In Figure 8, NN training performances for roll, pitch, yaw and z controllers are shown. Pitch NN controller training is completed with $2.37.10^{-10}$ mse. Training of NN Roll controller design is completed with best validation performance of $4.63.10^{-11}$. Yaw NN controller training is completed with error $8.07.10^{-9}$mse. These results prove that trained NN controllers will provide performances close to PD controllers.

As an example of trajectory following of the proposed controller, desired x^*, y^*, yaw* and z^* are defined as follows:

$$x^*(t) = sin(2.\pi.0.125t - \pi/2) \text{ m} \quad (14)$$

$$y^*(t) = sin(2.\pi.0.125t) \text{ m} \quad (15)$$

$$yaw^*(t) = 0.2(t-2)\text{rad} \quad (16)$$

$$z^*(t) = 4 \text{ m} \quad (17)$$

e) τ_{yaw} of PD f) τ_{yaw} of NN g) T of PD h) T of NN

The torques of PD and NN controller are shown in Figure 9. It is clear that NN approximates torque values with small errors and this will result similar performances of PD controllers.

Four NN controllers are trained with very low errors, so that the quadrotor will operate with a good trajectory tracking performance. To show the performance, trajectory tracking results of PD and NN controllers are shown in Figure 10. The reference trajectory is shown in blue, the trajectory of PD controllers is shown in green and the trajectory of NN controllers are shown in red, respectively. The error in terms of root mean square error (RMSE) is 0.0696 for NN controller while it is 0.0689 for PD controller. The results are very close and this proves the performance of NN controllers. Here, it should be noted that better performance results can be obtained with other types of neural networks like adaptive neuro-fuzzy systems (ANFIS) or radial basis neural networks (RBNN).

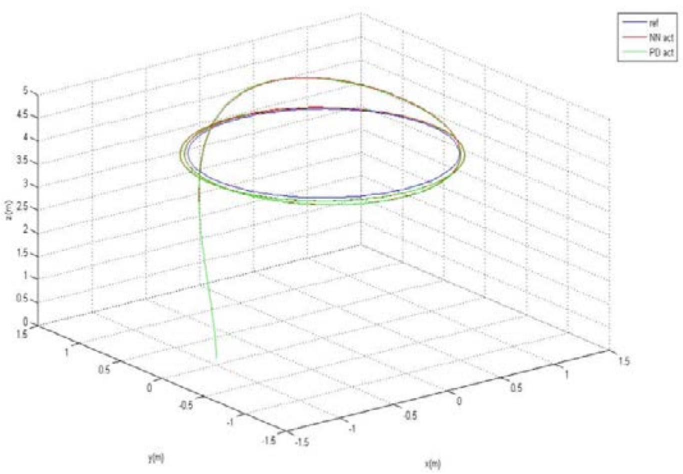


Figure 10: Trajectory tracking results of PD and NN controllers

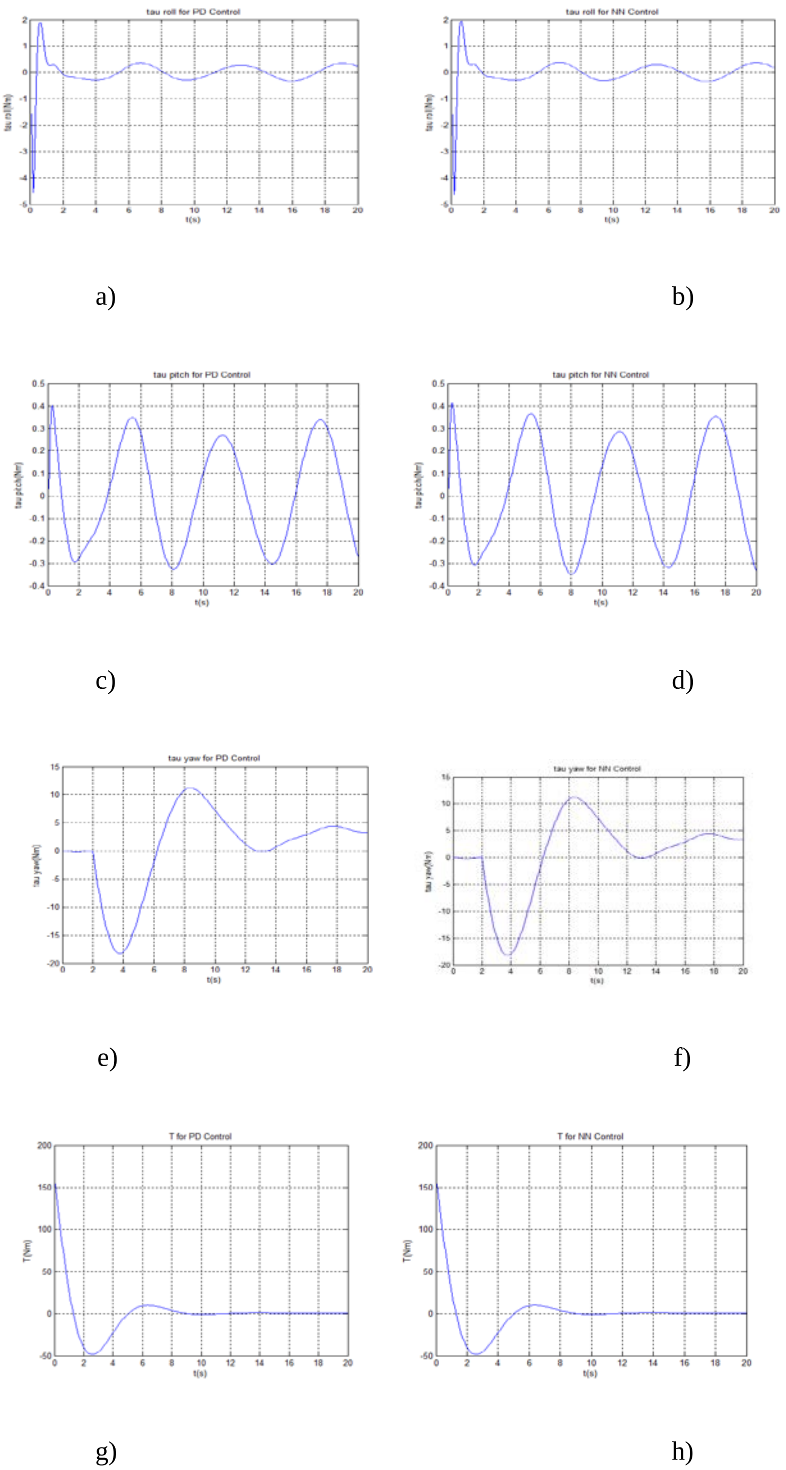


Figure 9: The torques of PD and NN controllers a) τ_{roll} of PD b) τ_{roll} of NN c) τ_{pitch} of PD d) τ_{pitch} of NN

5. Conclusion

While quadrotors are becoming more popular, their controllers should be improved. In this study, neural network control of quadrotors is aimed to obtain an artificial intelligence based controller. Firstly, the quadrotor is modeled according to quadrotor dynamics. Then, PD controllers for *x*, *y*, yaw and *z* control of quadrotor are implemented as classical controllers. The results for these controllers are recorded as training data of NN controllers. As the proposed controllers, NN controllers are trained according to these data and performance of these results are examined. The results verify that NN controllers achieve good trajectory tracking results. In the future studies, it is aimed to implement these NN controllers on a real quadrotor and the results for indoor and outdoor environments will be obtained.

References

[1] P. E. I. Pounds, "Design, Construction and Control of a Large Quadrotor Micro Air Vehicle," The Australian National University, 2007.

[2] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Precision flight control for a multi-vehicle quadrotor helicopter testbed," Control Eng. Pract., vol. 19, no. 9, pp. 1023–1036, 2011.

[3] R. Mahony, V. Kumar, and P. Corke, "Multirotor Aerial Vehicles: Modeling, Estimation, and Control of Quadrotor," IEEE Robot. Autom. Mag., vol. 19, no. 3, pp. 20–32, 2012.

[4] M. Santos, V. López, and F. Morata, "Intelligent fuzzy controller of a quadrotor," Proc. 2010 IEEE Int. Conf. Intell. Syst. Knowl. Eng. ISKE 2010, pp. 141–146, 2010.

[5] P. S. Bhatkhande, "Real time fuzzy controller for quadrotor stabilitiy control," p. 58, 2014.

[6] R. Xu and U. Ozguner, "Sliding Mode Control of a Quadrotor Helicopter," Proc. 45th IEEE Conf. Decis. Control, pp. 4957–4962, 2006.

[7] M. Herrera, W. Chamorro, A. P. Gomez, and O. Camacho, "Sliding Mode Control: An Approach to Control a Quadrotor," 2015 Asia-Pacific Conf. Comput. Aided Syst. Eng., pp. 314–319, 2015.

[8] Y. Bai, H. Liu, Z. Shi, and Y. Zhong, "Robust control of quadrotor unmanned air vehicles," Proc. 31th Chinese Control Conf., pp. 4462–4467, 2012.

[9] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.

[10] J. P. Antsaklis, "Neural Networks for Control Systems," vol. 1, no. 2, pp. 242–244, 1990.

[11] F. Lewis and S. Ge, "Neural networks in feedback control systems," Mech. Eng. Handbook. Wiley …, pp. 1–28, 2005.

[12] P. I. Corke, Robotics, Vision & Control: Fundamental Algorithms in Matlab. Springer US, 2011.

[13] MathWorks, "Neural Network Toolbox." [Online]. Available: https://www.mathworks.com/help/nnet/.