A Comparison of Back Propagation Neural Network and Elman Recurrent Neural Network Algorithms on Altitude Control of Heavy-lift Hexacopter Based on Direct Inverse Control

Bhakti Yudho Suprapto
Department of Electrical Engineering
Universitas Sriwijaya
Inderalaya, South of Sumatera, Indonesia
bhakti@ft.unsri.ac.id

Benyamin Kusumoputro
Department of Electrical Engineering
Universitas Indonesia
Depok, Indonesia
kusumo@ee.ui.ac.id

Abstract— **The altitude control is one of the important factors in controlling the heavy-lift hexacopter. This altitude control needs precise control since the heavy-lift hexacopter moves based on the speed of its driving motors. This paper uses a control based on Direct Inverse Controller with Neural Network Algorithm. The algorithm which used in this paper is Elman Recurrent Neural Network compared with Backpropagation Neural Network. The Backpropagation is the most used algorithm in Neural Network. In the testing result, The Elman Recurrent Neural Network algorithm result in smaller MSE value and capable to keep up with given data test compared with Backpropagation Neural Network algorithm.**

Keywords—Altitude Control, Backpropagation Neural Network, Direct Inverse Controller, Elman Recurrent Neural Network, Heavy-lift Hexacopter

I. INTRODUCTION

Hexacopter is one part of Unmanned Aerial Vehicle group. It mostly used for many needs such mapping, mapping on mining exploration, natural disaster area, and monitoring on environment change [1], monitoring and inspection of infrastructure i.e. bridges, power lines [2]. The heavy-lift hexacopter is a huge body hexacopter and it is capable on lift heavy things. To accomplish its duty, the heavy-lift hexacopter needs a reliable controller. One of the controlling on heavy-lift hexacopter is altitude control. This topic is so challenging [3] and it has been widely studied. Altitude control of the heavy-lift hexacopter needs special attention since it really depends on driving motors speed combination to against gravitation. Thus, to reach or keep particular height, it needs a precise controller. This altitude controller has been widely studied, some of them use Proportional Integral Derivative(PID) [3][4], and Linear Quadratic Regulator (LQR)[5]. Several problems to control heavy-lift hexacopter such as the difficulty in obtaining parameter of hexacopter, non-linear characteristics of the hexacopter [6] that causes many researchers develops another controlling methods such as Fuzzy Logic [7], Neural Network [8], and Elman Recurrent Neural Network[9]. These methods are also known as computational intelligent. It is capable to work in a system with high non-linear characteristic. Especially on Neural Network, that becomes a concern against along with the increasing of process speed on computer and hardware as a result of electronic and digital development.

The most uses algorithm on Neural Network is Backpropagation Neural Network (BPNN). BPNN algorithm is so popular because of its capability in studying complicated multidimensional mapping on non-linear system. It is usually called as “beyond regression” [10]. Besides, it has simple structure design so many researchers use this Backpropagation to solve their problem. There are many BPNN use such as attitude and altitude controller of the quadcopter [11], altitude control of the helicopter [12], as predictor [13], image processing[14] and etc. This paper also uses algorithm of Elman Recurrent Neural Network (ERNN). ERNN is part of Neural Network that uses context layer as storage of output activation function from hidden layer that will be used in the next data. By using network training that use Backpropagation, ERNN is appropriate to be used for high-order non-linear system [15][16]. The ERNN algorithm has been used as altitude and attitude controller of the heavy-lift hexacopter [17][9], load forecasting [18] and etc. ERNN is used to overcome BPNN problems which are often trapped on overfitting, easily fall on local minimum, inconsistent number of neuron hidden layer which makes network training failure [19]. This ERNN algorithm will be compared performance i.e. MSE, and attitude response with BPNN to control altitude of the heavy-lift hexacopter.

II. HEAVY-LIFT HEXACOPTER MODELLING

The heavy-lift hexacopter moves depend on the combination of driving motors. Hence, all of its move will be influenced by rotational speed of the rotor. This Rotor is a combination of motor as driving and propeller. The roll movement is combination of rotor 1,2 and 3 or rotor 4,5 and 6 that its speed change. Pitch movement is influenced by rotor movement 1 and 6 or 3 and 4 meanwhile, yaw movement is affected by rotor speed 1,3,5 or 2,4,6. This combination is caused by heavy-lift hexacopter uses plus structure (+) of the frame. This rotor speed combination also produces amount of thrust that comes from (1) below

978-1-5386-5721-8/18/$31.00 ©2018 IEEE

$$T_i = k_f \omega_i^2 \quad (1)$$

where i = 1,2….6 number of motors, constant k_f is lift constant which obtained from static thrust experiment. With neglect of propeller, so all strength and moment on rotor will be obtained (2) as follows.

$$F_i = C_T \rho A R^2 \omega_i^2 \quad (2)$$

The ρ is the air density constant, C_T is thrust coefficient that can be obtained from experiment and A, R each of them is variable area of rotor cross section and rotor radius that can be obtained from measurement. The Effect of dynamics on the heavy-lift hexacopter system refers to frame body while its position refers to Earth frame. The heavy-lift hexacopter dynamic based on those two frames can be elaborated with linear equation that combines both of frames that can be seen in Fig. 1. Therefore, equation of heavy-lift hexacopter speed from basic movement can be obtained through (3) below:

$$\begin{aligned}
\ddot{x} &= (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi)\frac{1}{m}U_1 \\
\ddot{y} &= (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi)\frac{1}{m}U_1 \\
\ddot{z} &= -g + (\cos\phi \cos\theta)\frac{1}{m}U_1 \\
p &= qr\left(\frac{I_y - I_z}{I_x}\right) - \frac{J_R}{I_x}\dot{\theta}\,\omega + \frac{d}{I_x}U_2 \\
q &= pr\left(\frac{I_z - I_x}{I_y}\right) - \frac{J_R}{I_y}\dot{\phi}\,\omega + \frac{d}{I_y}U_3 \\
r &= pq\left(\frac{I_x - I_y}{I_z}\right) - \frac{1}{I_z}U_4
\end{aligned} \quad (3)$$

Meanwhile, equation for basic movement from rotor angle speed change is showed by (4) below

$$\begin{aligned}
U_1 &= b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2 + \omega_5^2 + \omega_6^2) \\
U_2 &= bl\frac{\sqrt{3}}{2}(\omega_2^2 + \omega_3^2 - \omega_5^2 - \omega_6^2) \\
U_3 &= bl\frac{1}{2}(\omega_1^2 + \omega_2^2 - \omega_3^2 - \omega_4^2 - \omega_5^2 + \omega_6^2) \\
U_4 &= d(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2 - \omega_5^2 + \omega_6^2) \\
\omega &= -\omega_1 + \omega_2 - \omega_3 + \omega_4 - \omega_5 + \omega_6
\end{aligned} \quad (4)$$

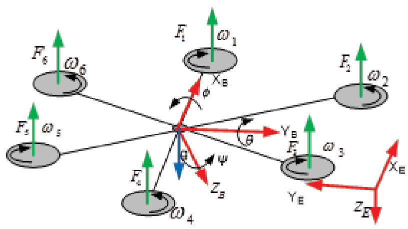


Fig. 1. Frame of heavy-lift hexacopter and axis work

Where $\ddot{x}, \ddot{y}, \ddot{z}$ is heavy-lift hexacopter linear acceleration on axis X_E, Y_E, Z_E, that refer to axis EF, Moment of inertia on axis X, Y,

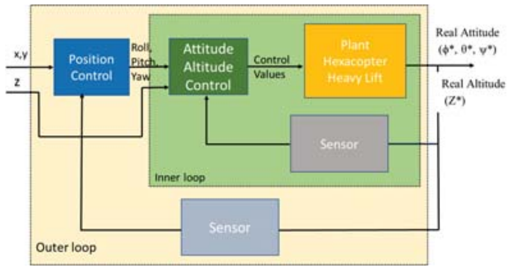


Fig. 2. Block Diagram of Heavy-lift Hexacopter Control System

Z which are I_{xx}, I_{yy}, I_{zz}, m and g is heavy-lift hexacopter mass and gravitation speed, U_1, U_2, U_3 and U_4 each of them is, torsi roll, torsi pitch and torsi yaw. By seeing at equation (3), heavy-lift hexacopter has 6 DOF on three rotation axis roll, pitch, yaw, and three axis translation x, y, z. Hence, in controlling, data such as roll, pitch, yaw movement and altitude of flying are needed. Moreover, to move this heavy-lift hexacopter in rolling (ϕ), pitching (θ), yawing (ψ), and height (z) movement needs motor speed. Thus, the next data which will be used is motor speed. These data will be used to train and test on Neural Network. The heavy-lift hexacopter controlling can be seen on Fig. 2.

It can be seen on Fig. 2, there are two loop controllers in the block diagram system, namely: inner loop, and outer loop. In the inner loop there is attitude and altitude controller where the target is how heavy-lift hexacopter moves according to basic movement on axis x and y, i.e. roll, pitch and yaw also reach particular altitude. This control becomes the basic of heavy-lift hexacopter control since it causes the heavy-lift hexacopter movement. On this attitude controller, it is also altitude control heavy-lift hexacopter where the target is how heavy-lift hexacopter moves on axis z, heavy-lift hexacopter altitude flying is determined on how big throttle given.

III. Direct Inverse Controller

A method to control non-linear system used dynamic inverse on its input to produce an expected output [20]. To reduce dynamics on system, inverse from non-linear system is trained until reach convergent [21]. Thus, basic principle of DIC system is by placing $r_{(t)}$ as input/reference, $f_{()}$ as system plant function and $f_{()}^{-1}$ as inverse of system plant function also $y_{(t)}$ as system output that can be seen from (5) and (6) below:

$$y_{(t)} = r_{(t)} \cdot f_{()}^{-1} \cdot f_{()} \quad (5)$$

$$y_{(t)} \cong r_{(t)} \quad (6)$$

DIC scheme on Fig. 3 is used to control altitude heavy-lift hexacopter. In this paper, DIC will combine with Neural Network algorithm to form a system that can adapt with dynamic system change. In this system, there is identity to map reference signal to output from controlled system. Hence, plant will be controlled by NN to reach output response expected. DIC method is chosen because it is easy to implement and can be maximized on particular mission. The DIC system consists of two blocks that is inverse model and identification system that implement multilayer perceptron consisting of input layer, hidden layer and output layer.

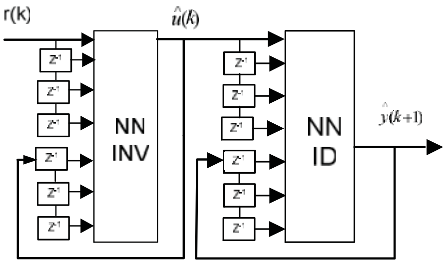


Fig. 3. Block diagram of Direct Inverse Control Neural Network

In this paper, DIC system uses both of BPNN and ERNN algorithm. Yet, both of these algorithms differ in inversing model while for identification system both of them use BPNN algorithm.

A. Identification System

On DIC-NN scheme, modeling heavy-lift hexacopter uses identification system. Since to be modeled mathematically needs a parameter which difficult to reach. Hence, this paper uses identification system from flight data. NN implementation on this identification system is output from system, which as function result from input process with NN activation function to the optimal weight combination. The weight iteratively comes to minimize error. The error is calculated based on the distinction of system output model with the output expected which symbolize with mean squared error (MSE). Non-linear dynamic system with input x and output y can be modeled (7) as follow [22]

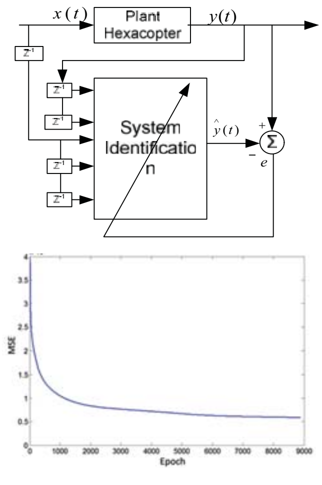
$$y(k) = f(\Phi(k), \Theta) \tag{7}$$

where y(k) as output model, Φ(k) is regression vector and parameter vector that represent as Θ(k). By structure model of Nonlinear Auto Regressive with eXogenous input (NARX), regression vector can be calculated with (8) as follow [22]

$$\Phi(k) = (x(k-1)..x(k-N_x), y(k-1)..y(k-N_y)) \tag{8}$$

Where N_x is maximum lag input and N_y is maximum lag output. Dynamics from heavy-lift hexacopter is determined with input and output, N_x = 2, N_y=2. Fig. 4 is a block diagram of identification system where the input is motor speed in form of pulse width modulation (PWM). Meanwhile, the output is heavy-lift hexacopter movements namely roll, pitch, yaw and altitude. Furthermore, this system identification is trained with 26 neuron of input layer, 35 neuron of hidden layer, and 4 neuron of output layer while the learning rate (α) is 0.2. The training is stopped since the decreasing of *Mean Square Error* (MSE) is not significant and the training reaches MSE value in amount of 5.88 x 10^{-5} on iteration 9,000. After testing for identification system is done, it can be seen at Fig. 5.

On Fig. 5 shows a good testing result that is proven by small MSE value and the response can adapt with MSE data value in amount 8.4811 x 10^{-4}



a) b)

Fig. 4. a) Block diagram of identification system b) MSE research result

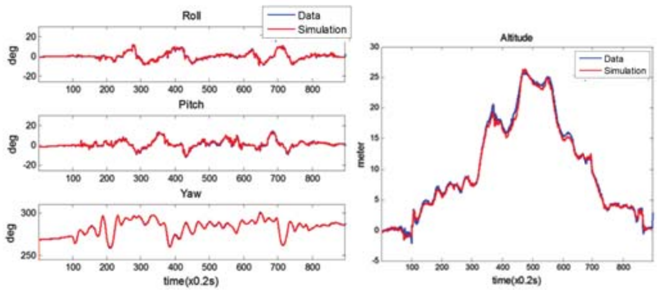


Fig. 5. The result of Identification test response

B. Inverse Model

On DIC scheme, inverse model is a controller that acts as inverse from plant so it's expected output plant can be used as input controller. The equation for inverse is:

$$x(k) = f(x[k-2], \dots, x[k-n_x-1], y[k], \dots, y[k-n_y]) \tag{9}$$

Where y is output plant, x is input plant, and n_y and n_x is the number of lag or operator delay for each output and input plant. Simulation scheme which used is the same with identification plant scheme, with the switching on input and output likes in Fig. 6. Inverse control training by using 24 of neuron input (data roll, pitch, yaw and altitude), 35 neuron hidden layer and 6 of neuron output (six PWM motor) with learning rate = 0.1. In this inverse model BPNN and ERNN will be used. Training for ERNN algorithm is conducted until 45,000 epochs and stopped because of MSE value does not significantly change. Fig. 7(a) shows response result of inverse training using ERNN with the MSE value in amount of 0.0212. Training result using BPNN algorithm can be seen in Fig. 7(b) with MSE value in amount of 0.0069 on epoch 140,000.

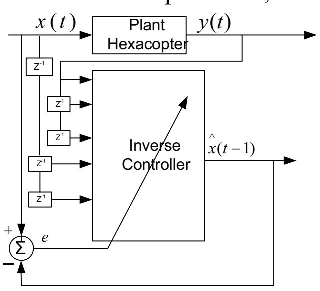
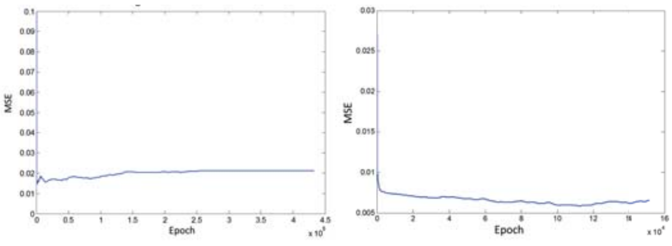
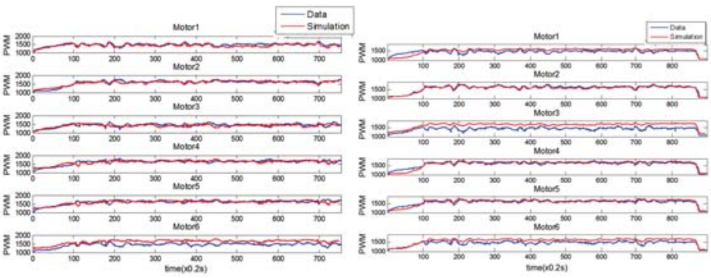


Fig. 6. Block diagram of inverse system



a) b)

Fig. 7. Inverse training result for control of attitude and altitude (a) ERNN and (b) BPNN



a) *b)*

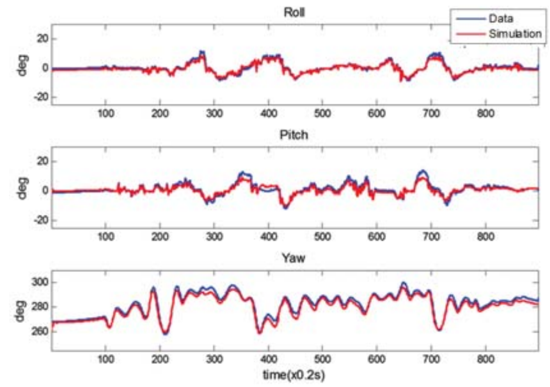
Fig. 8. Inverse testing result for altitude control with (a) ERNN and (b) BPNN

After optimum weight is obtained, testing data which shows in Fig. 8 is conducted. The MSE value is 0.1284 for ERNN algorithm and MSE value in amount of 0.158 for BPNN algorithm. By seeing at Fig. 8(a) response result considered as good but there is big error on motor 6, while on Fig 8(b) error occurs on motor 1, motor 3 and motor 6 with big enough MSE value. If it is compared with response result on ERNN, there is bigger error in result from BPNN algorithm

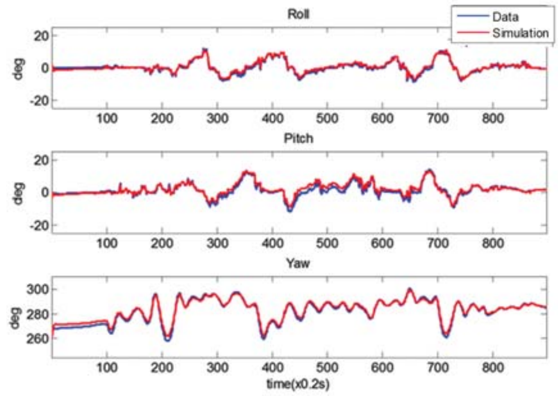
IV. Experiment Result

After training is conducted on identification system and inverse model, optimum weight value is obtained. Then, testing on both of algorithms are conducted by using different test data with training data and never trained before. Based on testing with ERNN algorithm, response result is obtained that shows on Fig. 9(a) where MSE value in amount of 0.0086. Meanwhile on Fig. 9(a) and 10(a), it can be seen that DIC controller with ERNN algorithm is generally able to adapt test data with small error. Even though on pitch, roll and yaw movements have error, but on altitude test there is error on the beginning and ending of test in amount of 1.6 meters and 1.5 degree. It happens because of measurement mistake of sensor reading. But, above 5 meters, response controller can follow data until it reaches 26 meters and decreases again until landing. On second 116 until 134 there is mistake until reach error in 1.6 meters. It is caused by motor speed decreasing and various slop change. After that control response can follow data until landing.

Meanwhile, on BPNN algorithm, testing response can be seen on Fig. 9(b), MSE value is 0.0114. On Fig. 9(b) and 10(b) can be seen that testing response on BPNN algorithm can follow test data that is given. Yet, there is big error on beginning and ending of testing in amount of 2.17 meter and 2.5 meter that causes by sensor reading mistake. After heavy-lift hexacopter above 5 meters, the mistake is handled. However, at its peak there is error in mount of 1.25 meters, so BPNN algorithm cannot reach maximum altitude as on test data. This test is done in the morning in order to avoid environmental disturbances such as wind due to testing with disturbance done in further research.

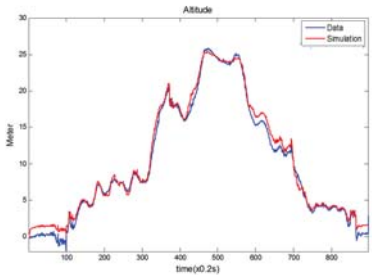


a) ERNN Algorithm

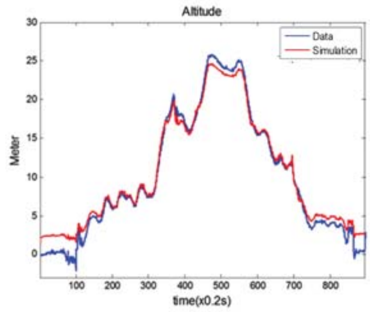


b) BPNN Algorithm

Fig. 9. Response of attitude on DIC Controller



a). ERNN Algorithm



b). BPNN Algorithm

Fig. 10. Response of altitude on DIC Controller

From this testing, ERNN algorithm has smaller MSE value than BPNN algorithm. As well as response chart, where ERNN algorithm is more capable in following test data than BPNN algorithm. It happens because on ERNN algorithm has context layer that save output result from hidden layer. After that, weight on context layer updated until it is more adaptive to the system dynamic. In ERNN algorithm, training of epoch number is smaller to reach convergent BPNN, but for training time BPNN faster than ERNN in obtaining convergent training. In both of algorithm, learning rate made the same, as well as activation function, number of neuron in input layer, hidden layer, and output layer. It is done to compare both of algorithms in running the same mission.

V. CONCLUSIONS

Based on the testing, both of algorithms can perform their mission as altitude control on heavy-lift hexacopter with small MSE value and attitude response that can follow test data with small error. Nevertheless, ERNN algorithm is better in controlling altitude because it has smaller MSE value and error than BPNN.

In achieving maximum altitude, ERNN is more capable in reaching the altitude than BPNN. Even though both of algorithms experienced error in the beginning and ending of testing which caused by reading sensor mistake.

Context layer tethering gives influence to controller in facing system dynamics and non-linear characteristic on the system. Future works of this research will increase ERNN performance by modifying context layer and adding self-feedback on the context layer.

REFERENCES

[1] P. Liu *et al.*, "A review of rotorcraft Unmanned Aerial Vehicle (UAV) developments and applications in civil engineering," *Smart Struct. Syst.*, vol. 13, no. 6, pp. 1065–1094, 2014.

[2] S. Darma, J. L. Buessler, G. Hermann, J. P. Urban, and B. Kusumoputro, "Visual servoing quadrotor control in autonomous target search," in *Proceedings - 2013 IEEE 3rd International Conference on System Engineering and Technology, ICSET 2013*, 2013, pp. 319–324.

[3] M. Fatan, B. L. Sefidgari, and A. V. Barenji, "An adaptive neuro PID for controlling the altitude of quadcopter robot," *2013 18th Int. Conf. Methods Model. Autom. Robot.*, pp. 662–665, 2013.

[4] Z. Mustapa, S. Saat, A. M. Darsono, and H. H. Yusof, "Experimental validation of an Altitude control for quadcopter," *ARPN J. Eng. Appl. Sci.*, vol. 11, no. 6, pp. 3789–3795, 2016.

[5] E. C. Suicmez and A. T. Kutay, "Attitude and altitude tracking of hexacopter via LQR with integral action," in *2017 International Conference on Unmanned Aircraft Systems, ICUAS 2017*, 2017, pp. 150–159.

[6] A. S. Sanca, P. J. Alsina, and J. De Jesus F. Cerqueira, "Dynamic modeling with nonlinear inputs and backstepping control for a hexarotor micro-aerial vehicle," in *Proceedings - 2010 Latin American Robotics Symposium and Intelligent Robotics Meeting, LARS 2010*, 2010, pp. 36–42.

[7] A. Bulgakov, S. Emelianov, T. Bock, and D. Sayfeddine, "Control of hovering altitude of a quadrotor with shifted centre of gravity for inspection of high-rise structures," in *31st International Symposium on Automation and Robotics in Construction and Mining, ISARC 2014 - Proceedings*, 2014.

[8] J. Muliadi and B. Kusumoputro, "Neural Network Control System of UAV Altitude Dynamics and Its Comparison with the PID Control System," *J. Adv. Transp.*, vol. 2018, 2018.

[9] B. Y. Suprapto, M. A. Heryanto, H. Suprijono, and B. Kusumoputro, "Altitude Control of Heavy-Lift Hexacopter using Direct Inverse Control Based on Elman Recurrent Neural Network," in *Proceedings of the 8th International Conference on Computer Modeling and Simulation - ICCMS '17*, 2017, pp. 135–140.

[10] R. Hecht-Nielsen, "Theory of the Backpropagation Neural Network," in *Neural Networks for Perception*, 1992, pp. 65–93.

[11] M. A. Heryanto, H. Suprijono, B. Y. Suprapto, and B. Kusumoputro, "Attitude and altitude control of a quadcopter using neural network based direct inverse control scheme," *Adv. Sci. Lett.*, vol. 23, no. 5, 2017.

[12] H. Suprijono and B. Kusumoputro, "Direct inverse control based on neural network for unmanned small helicopter attitude and altitude control," *J. Telecommun. Electron. Comput. Eng.*, vol. 9, no. 2–2, 2017.

[13] F. Gaxiola, P. Melin, F. Valdez, and O. Castillo, "Interval type-2 fuzzy weight adjustment for backpropagation neural networks with application in time series prediction," *Inf. Sci. (Ny).*, vol. 260, pp. 1–14, 2014.

[14] A. Nagathan and M. I. Manimozhi, "Content-Based Image Retrieval System Using Feed-Forward Backpropagation Neural Network," *Int. J. Comput. Sci. Eng.*, vol. 2, no. 4, pp. 143–151, 2013.

[15] Y.-C. Cheng, W.-M. Qi, and J. Zhao, "A new Elman neural network and its dynamic properties," in *2008 IEEE International Conference on Cybernetics and Intelligent Systems, CIS 2008*, 2008.

[16] D. T. Pham and X. Liu, "Training of Elman networks and dynamic system modelling," *Int. J. Syst. Sci.*, vol. 27, no. 2, pp. 221–226, 1996.

[17] B. Y. Suprapto and B. Kusumoputro, "Optimized neural network-direct inverse control for attitude control of heavy-lift hexacopter," *J. Telecommun. Electron. Comput. Eng.*, vol. 9, no. 2–5, pp. 103–107, 2017.

[18] L. Ren, Y. Liu, Z. Rui, H. Li, and R. Feng, "Application of Elman neural network and MATLAB to load forecasting," in *Proceedings - 2009 International Conference on Information Technology and Computer Science, ITCS 2009*, 2009, vol. 1, pp. 55–59.

[19] S. Ding, Y. Zhang, J. Chen, and W. Jia, "Research on using genetic algorithms to optimize Elman neural networks," *Neural Comput. Appl.*, vol. 23, no. 2, pp. 293–297, 2013.

[20] B. Widrow and M. Bilello, "Adaptive inverse control," *Proc. 8th IEEE Int. Symp. Intell. Control*, pp. 1–6, 1993.

[21] M. Amir and V. Abbas, "Modeling and Neural Control of Quad rotor Helicopter," *Yanbu J. Eng. Sci.*, vol. 2, pp. 35–51, 2011.

[22] B. Kusumoputro, K. Priandana, and W. Wahab, "System identification and control of pressure process rig?? system using backpropagation neural networks," *ARPN J. Eng. Appl. Sci.*, vol. 10, no. 16, pp. 7190–7195, 2015.