Modified Elman Recurrent Neural Network for Attitude and Altitude Control of Heavy-lift Hexacopter

Bhakti Yudho Suprapto
Department Electrical Engineering
Universitas Indonesia
Kampus baru UI Depok, Indonesia
bhakti.yudho@ui.ac.id

Amsa Mustaqim
Department Electrical Engineering
Universitas Indonesia
Kampus baru UI Depok, Indonesia
amsamustaqim@hotmail.com

Wahidin Wahab
Department Electrical Engineering
Universitas Indonesia
Kampus baru UI Depok, Indonesia
wahidin.wahab@gmail.com

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

***Abstract*— Hexacopter is a member of rotor-wing Unmanned Aerial Vehicle (UAV) which has 6 six rotors with fixed pitch blades and nonlinear characteristics that cause controlling the attitude of hexacopter is difficult. In this paper, Modified Elman Recurrent Neural Network (MERNN) is used to control attitude and altitude of Heavy-lift Hexacopter to get better performance than Elman Recurrent Neural Network (ERNN). This Modified Elman Recurrent Neural Network has a self-feedback which provides a dynamic trace of the gradients in the parameter space. In the self-feedback, the gain coefficients are trained as connection weight. This connection weight could enhance the adaptability of Elman Recurrent Neural Network to the time-varying system. The flight data are taken from a real flight experiment. Results show that the Modified Elman Recurrent Neural Network can increase performance with small error and generate a better response than Elman Recurrent Neural Network.**

Keywords*— *Direct Inverse Control; Elman Recurrent Neural Network; Heavy-lift Hexacopter; Modified Elman Recurrent Neural Network.

I. INTRODUCTION

The hexacopter has been developed and studied in this decade due to its advantages such as vertical take-off and landing (VTOL), maneuvers, and hover[1, 2]. The hexacopter has six motors as actuators with propellers which are mounted on a rigid body frame making 120 degrees of angle from each other. The Propellers of hexacopter have three sets of clockwise and counter-clockwise. Each propeller of the hexacopter produces an upward thrust by pressing air downwards. The angular velocity of the rotors can be controlled based on this propeller configuration. The other advantages of this hexacopter are the possibility of managing one or more motor failures and ability able to lift more payload[3-5]. One of the main issue related to the use of hexacopter is attitude and altitude control. However, controlling the hexacopter is not easy as it has the nonlinear system, multivariable and coupling parameters and unstable systems[6, 7]. Therefore, it is important to obtain a precise controller of a hexacopter in order to overcome the problem and perform the missions. Researchers have studied many methods to solve the problems of hexacopter control systems such as PID[8], PID-LQR[9], Back Stepping control[10] but these methods have limitations to be working on a nonlinear system. The other methods i.e. fuzzy logic[11], and Neural Network[5, 12] have been developed to solve the problem of control hexacopter. These methods especially neural network can work with the nonlinear system and adaptive to the environment. In the previous work, control of heavy-lift hexacopter using Neural Network has been conducted. The Neural Network using Elman Recurrent Neural Network (ERNN) algorithm as learning algorithm is similar to the Backpropagation learning mechanism. The Elman recurrent neural network (NN) is a subgroup of recurrent network model that has an additional layer used to memorize previous activations of the hidden neurons and feed to all the hidden neurons after the one-step time delayed. The Elman Recurrent Neural Network has been successful to control heavy-lift hexacopter and show a good response and small error on the test data given[5].

In this paper, the Modified Elman Recurrent Neural Network is used to improve the performance and the dynamic characteristics of Elman Recurrent Neural Network in controlling heavy-lift hexacopter. This modification is performed by adding a self-feedback connection with the fixed gain on the context layer. This self-feedback causes the output context layer at a time k equals to the output of hidden layer at k-1 time. Therefore, it could increase dynamic characteristic of the system, and convergence speed[13]. Modified Elman Recurrent Neural Network has been used as controller permanent magnet synchronous generator (PMSG) system[13], prediction network traffic[14], and fault diagnosis[15].

This paper is organized as follows: Section 1 describes introduction. Section 2 describes the heavy-lift hexacopter model. Section 3 describes the direct inverse control neural networks using the Modified Elman Recurrent Neural Network

(MERNN) learning algorithm. Section 4, experiments result and analysis of the developed controller of the heavy-lift hexacopter is conducted and presented. To sum up to a summary is presented in the last section.

II. Model of Hexacopter

A. Dynamic Model of Hexacopter

The dynamic characteristic briefly describes attitude hexacopter according to the geometry of hexacopter. This hexacopter consists of six rotors located orthogonally at fixed body frame shown in Fig 1. The combination of the pair rotors makes three movements of hexacopter i.e. Roll (rotation around the X axis), pitch movement (rotation around the Y axis), yaw (rotation about the Z axis). Roll movement is obtained when the balance of rotors 1, 2 and 3 (or 6, 5 and 4) is changed (speed increases or decreases). Pitch movement is obtained when the balance of the speed of the rotors 1 and 6 (or 3 and 4) is changed and yaw movement is got when by a simultaneous change of speed of the motors (1, 3, 5) or (2, 4, 6).

B. Kinematic of Hexacopter

The kinematic characteristic is a connection between fixed body frame and earth inertial frame. In Fig 1, the frame structure of hexacopter and rotation direction of the rotors are illustrated. From this figure, it is seen that hexacopter motion has two reference systems i.e. body fixed frame and earth inertial frame.

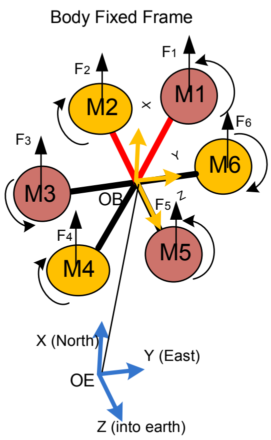


Fig. 1. The structure of hexacopter and its frame

The direction of motor rotation and the orientation of the hexacopter is presented in Fig. 1. It shows three Euler angles, namely roll angle ϕ, pitch angle θ, and yaw angle ψ that is in form of vector $\eta = [\ \phi, \theta, \psi]^T$. Vector $\xi = [x, y, z\ ]^T$ shows the position of the hexacopter in the inertial frame. Therefore, the transformation matrix of the body-fixed frame (B) to the earth-fixed frame (E) and rotation matrix is obtained by:

$$T = \begin{bmatrix} 1 & \sin\phi \tan\theta & \cos\phi \ \tan\theta \\ 0 & \cos\phi & \sin\phi \\ 0 & \sin\phi \sec\theta & \sec\theta \cos\phi \end{bmatrix} \tag{1}$$

$$R = \begin{bmatrix} \cos\theta \cos\psi & \cos\psi \sin\theta \sin\phi - \cos\phi \sin\psi & \cos\phi \cos\psi \sin\theta + \sin\phi \sin\psi \\ \cos\theta \sin\psi & \cos\phi \cos\psi + \sin\theta \sin\phi \sin\psi & \cos\phi \sin\psi \sin\theta - \cos\psi \sin\phi \\ -\sin\theta & \cos\theta \sin\phi & \cos\theta \cos\phi \end{bmatrix} \tag{2}$$

So, the equation below is the dynamic model of hexacopter

$$\begin{cases} \ddot{X} = -(\sin\psi \sin\phi + \cos\psi \sin\theta \cos\phi)\dfrac{u_1}{m} \\ \ddot{Y} = (-\cos\phi \sin\theta \sin\psi - \cos\psi \sin\phi)\dfrac{u_1}{m} \\ \ddot{Z} = -g + (\cos\theta \cos\phi)\dfrac{u_1}{m} \\ \ddot{\phi} = (\dot{\theta}\dot{\psi}(I_{yy} - I_{zz}) + \tau_x) / I_{xx} \\ \ddot{\theta} = (\dot{\phi}\dot{\psi}(I_{zz} - I_{xx}) + \tau_{yy}) / I_{yy} \\ \ddot{\psi} = (\dot{\phi}\dot{\theta}(I_{xx} - I_{yy}) + \tau_z) / I_{zz} \end{cases} \tag{3}$$

where $\ddot{\phi}, \ddot{\theta}, \ddot{\psi}$ are the hexacopter's angular accelerations in the B axis, while $\ddot{X}, \ddot{Y}, \ddot{Z}$ are the hexacopter's linear accelerations in the E axis, I_{XX}, I_{YY}, I_{ZZ} are the moments of body inertia at *xyz*-axis, *g* is the gravity speed, and *m* is the mass of the hexacopter.

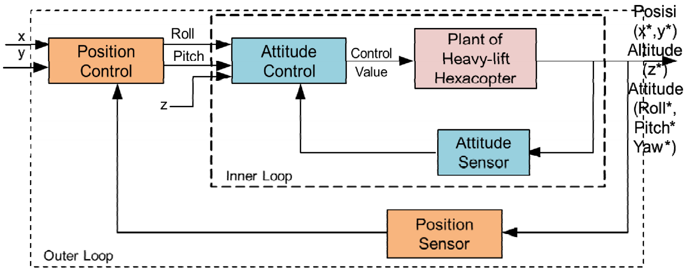


Fig. 2. Block Diagram of Heavy-lift Hexacopter Control System

Block diagram of a control system for heavy-lift hexacopter in general depicted in Fig. 2. The roll (ϕ), pitch (θ) and yaw (ψ) movement of heavy-lift hexacopter are controlled by attitude control so the output and the input reference are similar. This is called the inner loop control. But the outer loop control is used to control the x and y movement which directly depicts the real position of the heavy-lift hexacopter.

III. Direct Inverse Control Based on Elman Recurrent Neural Network

A. Direct Inverse Control (DiC)

Neural Network Direct inverse control (NN-DIC) is the simplest solution for control of a nonlinear system that consists of connecting in series the inverse model and the plant. The dynamic properties are eliminated by training on the inverse model as inverted to the plant[16, 17]. Thus, it makes the input of the inverse model is similar to the desired output plant. It caused the inverse model function as a controller that makes a similar response to the given reference signal. NN-DIC consist of system identification and an inverse model. Block diagram of the neural networks based direct inverse controller scheme (NN-DIC) is presented in Fig. 3. As seen in this figure, NN-DIC scheme can be experimentally simulated by using a system identification and an inverse model and Fig. 4. show training configuration scheme of system identification and inverse model.

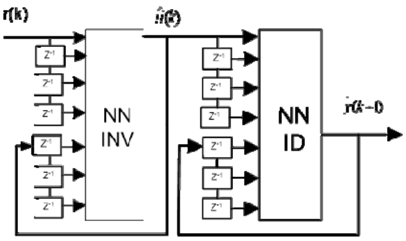
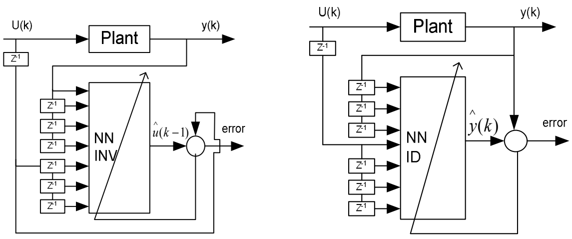


Fig. 3. Direct Inverse Control (DIC) scheme



a) *b)*

Fig. 4. Training configuration scheme a) system identification, b) Inverse model

B. Elman Recurrent Neural Network (ERNN)

Elman Recurrent Neural Network is developed by Jeffrey Elman as one kind of globally feed-forward locally recurrent network model[18]. Elman Recurrent Neural Network (ERNN) consists of four layers, i.e., an input layer, a context layer, a hidden layer, and an output layer. ERNN has been widely researched for the purpose of system identification, predicting, fault diagnosis and forecasting[15, 19]. Fig 5. shows the architecture of Elman Recurrent Neural Network.

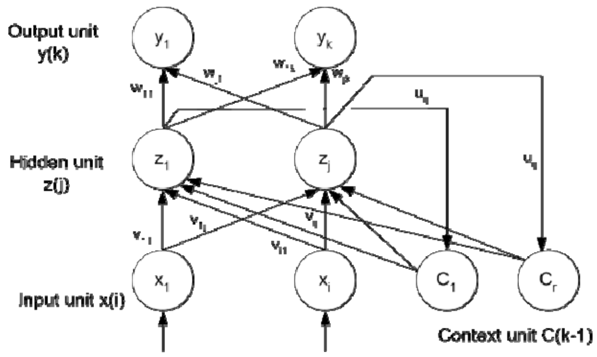


Fig. 5. Elman Recurrent Neural Network Architecture

As shown in Fig. 5., ERNN can be considered to be a special type of neural network with connections from the hidden layer to the context layer. The context layer is an additional layer that functions as a memory to memorize previous activations of the hidden layer and to feed all the hidden layer after the one-step time delay. Therefore, ERNN has a special explicit memory to save the temporal information in the context layer. Furthermore, ERNN can approximate high-order dynamic systems, and its converge speed is fast enough. Given from the Fig. 5. that the input is x(k), the output is yink(k) and the total input to the hidden layer j is zinj(k), then the equations of the architecture are:

$$z_{inj}(k)=\sum u_{ij}(k-1)C_i(k)+v_{ij}(k-1)x(k) \tag{4}$$

$$z_j(k)=f(z_{inj}) \tag{5}$$

$$C_j(k)=z_i(k-1) \tag{6}$$

$$y_{ink}(k)=\sum w_{ij}(k-1)z_j(k) \tag{7}$$

where w_{ij} is the weights of the hidden layer to the output layer, u_{ij} is the weights of the context layer to the hidden layer, v_{ij} is the weights of the input layer to the hidden layer and f is the activation function of the hidden layer. The ERNN training is similar to the backpropagation training. In learning algorithm of ERNN, the training was done iteratively by minimizing the resulting error E_k or the difference between the actual output $y_d(k)$ and the output generated by the network $y_{ink}(k)$ expressed as:

$$E_k=\frac{1}{2}(y_d(k)-y_{ink}(k))^2 \tag{8}$$

Based on the error value in equation (8), the weights of each layer can be modified by the following equations:

$$\frac{\partial E_k}{w_{ij}(k-1)}=-\left(y_d(k)-y_{ink}(k)\right)z_j(k) \tag{9}$$

$$\frac{\partial E_k}{v_{ij}(k-1)}=-\left(y_d(k)-y_{ink}(k)\right)w_i(k-1)\,f^1 z_{inj}\;x(k)$$
(10)

$$\frac{\partial E_k}{u_{ij}(k-1)}=-\left(y_d(k)-y_{ink}(k)\right)w_i(k-1)\frac{\partial z_j}{u_{ij}(k-1)} \tag{11}$$

Where $\frac{\partial z_j}{u_{ij}(k-1)}=f^1 z_{inj}\;c_j(k-1)$ (12)

The general weight modification in the gradient descent method is:

$$\Delta w=-\eta\frac{\partial E_k}{\partial w} \tag{13}$$

So,

$$\Delta w_{ij}(k)=\eta(y_d(k)-y_{ink}(k))z_j(k) \tag{14}$$

$$\Delta v_{ij}(k)=\eta\left(y_d(k)-y_{ink}(k)\right)w_i(k-1)f' z_{inj}x(k) \tag{15}$$

$$\Delta u_{i,j}(k)=\eta(y_d(k)-y_{ink}(k))w_i(k-1)fz_{inj}c_j(k-1) \tag{16}$$

where η is the learning rate value.

C. Modified Elman Recurrent Neural Network (MERNN)

The modified Elman network is a type of recurrent neural network with four layers of neurons i.e. the input layer, the hidden layer, the context layer and the output layer. The MERNN differs from the original ERNN by having self-feedback links with fixed gain in the context layer. Fig 6. depicts the modified Elman network.

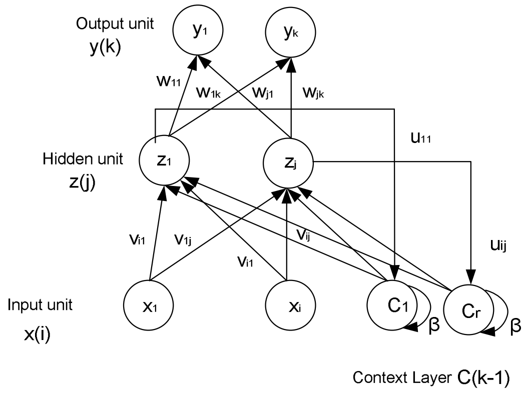


Fig. 6. Modified Elman Recurrent Neural Network Architecture

Fig. 6. shows the self-feedback (β) in the context layer that has value $0 \leq \beta < 1$. When the gain β is zero, the MERNN is identical to the original ERNN. In this paper, the value of β is *0.01*. Generally, equation MERNN is similar to ERNN, but it is different in context layer as MERNN use self-feedback (β). the equations of the architecture MERNN are:

$$z_{inj}(k) = \sum u_{ij}(k-1)C_i(k) + v_{ij}(k-1)x(k) \tag{17}$$

$$z_j(k) = f(z_{inj}) \tag{18}$$

$$C_j(k) = z_i(k-1) + \beta z_i(k-1) \tag{19}$$

$$y_{ink}(k) = \sum w_{ij}(k-1) z_j(k) \tag{20}$$

While training algorithm of MERNN

$$\Delta w_{ij}(k) = \eta(y_d(k) - y_{ink}(k)) z_j(k) \tag{21}$$

$$\Delta v_{ij}(k) = \eta\left(y_d(k) - y_{ink}(k)\right) w_i(k-1) f' z_{inj} x(k) \tag{22}$$

$$\Delta u_{i,j}(k) = \eta(y_d(k) - y_{ink}(k)) w_i(k-1) \frac{\partial z_j}{\partial u_{ij}(k-1)} \tag{23}$$

$$\frac{\partial z_j}{\partial u_{ij}(k-1)} = f^1 z_{inj} c_j(k) \tag{24}$$

Substitute (24) into (19) gives

$$\frac{\partial z_j}{\partial u_{ij}(k-1)} = f^1 z_{inj} c_j(k) + \beta \frac{\partial z_j(k-1)}{\partial u_{ij}(k-2)} \tag{25}$$

Equation (25) can provide an infinite impulse response. This is the reason why the MERNN was able to model higher-order dynamic systems.

IV. Experimental Result

In this research, experiments are done by using real data flight of a heavy-lift hexacopter that consist of one sets for training and others for testing. The training of real flight data using the Backpropagation algorithm in identification system and the ERNN in inverse model. The neural network configuration for this identification system as the plant model consists of an input layer with 26 neurons, a hidden layer with 35 neurons and an output layer with 4 neurons. While, the configuration of the inverse model consists of a single input layer, a single hidden layer, and a single output layer with 24, 35, and 6 neurons, respectively. Fig. 7. Shows the test result of identification training that reached its convergence in *45,000* epochs and the obtained Mean Square Error (MSE) for this training was 4.511×10^{-4}. On the testing stage, the obtained MSSE was *0.0033*.

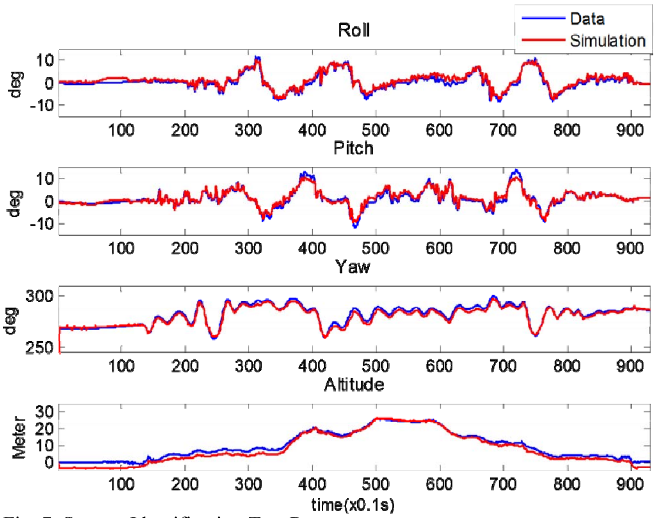


Fig. 7. System Identification Test Responses

The test result for the inverse model is shown in Figure 8. The training required *33,000* epochs to produce a training MSE of *0.0197* and MSE of testing is *0.0820*.

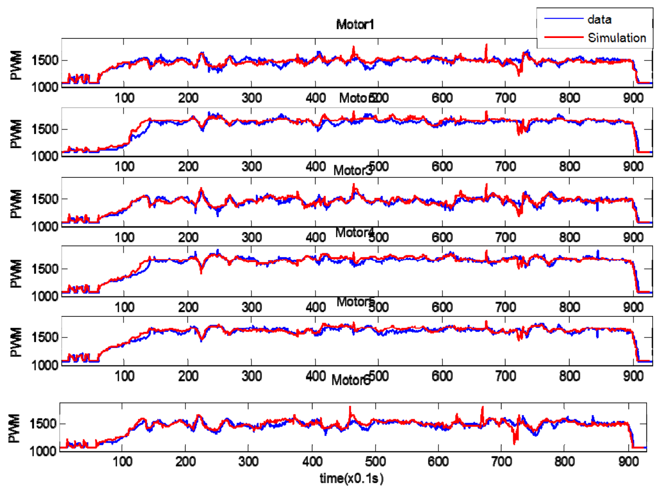


Fig. 8. Inverse Model Test Responses

Testing of NN-DIC done after the weight of the training and testing of system identification and inverse model is obtained. The result of this test is depicted in Fig. 9. and Fig. 10. The Fig. 9. reflects that the outputs of the simulated NN-DIC with ERNN algorithm shown in red curves are in good agreement with the real flight test data shown in blue curves but on a roll, pitch and yaw movement, there is a small error. The value of Mean Square Error (MSE) is 0.0256. The Error in roll, pitch and yaw movement is 3.88 degree, 5.26 degree, and 5.9 degree respectively. While Fig. 10. show result response test NN-DIC with MERNN algorithm. The output response can follow the real test data although there is a small error in roll movement. The value of MSE is 0.0099. Error in roll movement is 3.56 degree, pitch movement is 5.356 degree and yaw movement is 4.3 degree.

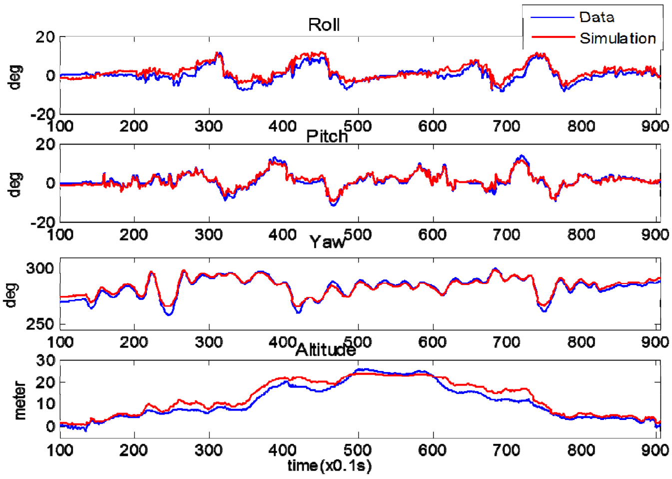


Fig. 9. NN-DIC ERNN test response

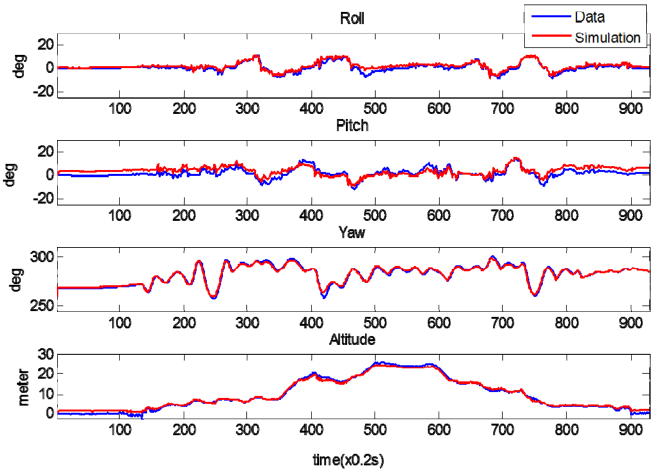


Fig. 10. NN-DIC MERNN test response

The other focus in this paper is altitude. Both algorithms show a good response and can follow the real test data. The highest position of altitude can reach by all of the algorithms although there is a small error at the beginning. It is caused by initializing when heavy-lift hexacopter starts to fly.

V. Conclusion

In this paper, controlling the attitude characteristics of a heavy-lift hexacopter using neural networks based direct inverse control system is developed. An Elman recurrent neural networks and Modified Elman recurrent neural network is utilized learning mechanism. Experiments conducted for proving the proposed algorithm could improve performance controller. Elman Recurrent Neural Network (ERNN) and Modified Elman Recurrent Neural Network (MERNN) can be utilized to control the attitude of a heavy-lift hexacopter with low error and good system response. Results show that the MERNN algorithm has lower attitude and altitude error compared with that of the ERNN algorithm. Further experiments are conducted in order to investigate, implement and analyze the MERN algorithm to control maneuvers of heavy-lift hexacopter.

Acknowledgment

This work was supported by the Ministry of Research and Higher Education of Indonesia through Penelitian Disertasi Doktor (PDD) research fund and Hibah Publikasi International Terindeks untuk Tugas Akhir (PITTA) from Direktorat Riset dan Pengabdian Masyarakat (DRPM) Universitas Indonesia.

References

[1] G. P. Falconí and F. Holzapfel, "Adaptive fault tolerant control allocation for a hexacopter system," in American Control Conference (ACC), 2016, 2016, pp. 6760-6766.

[2] F. Kendoul, Z. Yu, and K. Nonami, "Guidance and nonlinear control system for autonomous flight of minirotorcraft unmanned aerial vehicles," Journal of Field Robotics, vol. 27, pp. 311-334, 2010.

[3] V. Artale, C. L. Milazzo, C. Orlando, A. Ricciardello, T. E. Simos, and C. Tsitouras, "Genetic algorithm applied to the stabilization control of a hexarotor," in AIP Conference Proceedings, 2015, p. 780003.

[4] V. Artale, C. L. Milazzo, and A. Ricciardello, "A quaternion-based simulation of multirotor dynamics," International Journal of Modeling, Simulation, and Scientific Computing, vol. 6, p. 1550009, 2015.

[5] B. Kusumoputro, H. Suprijono, M. A. Heryanto, and B. Y. Suprapto, "Development of an attitude control system of a heavy-lift hexacopter using Elman recurrent neural networks," in Automation and Computing (ICAC), 2016 22nd International Conference on, 2016, pp. 27-31.

[6] M. Moussid, A. Sayouti, and H. Medromi, "Dynamic modeling and control of a hexarotor using linear and nonlinear methods," International Journal of Applied Information Systems, vol. 9, 2015.

[7] E. Capello, H. Park, B. Tavora, G. Guglieri, and M. Romano, "Modeling and experimental parameter identification of a multicopter via a compound pendulum test rig," in Research, Education and Development of Unmanned Aerial Systems (RED-UAS), 2015 Workshop on, 2015, pp. 308-317.

[8] A. Alaimo, V. Artale, C. L. R. Milazzo, and A. Ricciardello, "PID controller applied to hexacopter flight," Journal of Intelligent & Robotic Systems, vol. 73, pp. 261-270, 2014.

[9] N. D. Salim, D. Derawi, S. S. Abdullah, S. A. Mazlan, and H. Zamzuri, "PID plus LQR attitude control for hexarotor MAV in indoor environments," in Industrial Technology (ICIT), 2014 IEEE International Conference on, 2014, pp. 85-90.

[10] C. A. Arellano-Muro, L. F. Luque-Vega, B. Castillo-Toledo, and A. G. Loukianov, "Backstepping control with sliding mode estimation for a hexacopter," in Electrical Engineering, Computing Science and Automatic Control (CCE), 2013 10th International Conference on, 2013, pp. 31-36.

[11] E. Koslosky, M. A. Wehrmeister, J. A. Fabro, and A. S. de Oliveira, "On using fuzzy logic to control a simulated hexacopter carrying an attached pendulum," in Designing with Computational Intelligence, ed: Springer, 2017, pp. 1-32.

[12] V. Artale, M. Collotta, G. Pau, A. Ricciardello, T. Simos, G. Psihoyios, et al., "Hexacopter trajectory control using a neural network," in AIP Conference Proceedings, 2013, pp. 1216-1219.

[13] C.-H. Lin, "Novel modified Elman neural network control for PMSG system based on wind turbine emulator," Mathematical Problems in Engineering, vol. 2013, 2013.

[14] X. Wang, C. Zhang, and S. Zhang, "Modified Elman neural network and its application to network traffic prediction," in Cloud Computing and Intelligent Systems (CCIS), 2012 IEEE 2nd International Conference on, 2012, pp. 629-633.

[15] L. Hongmei, W. Shaoping, and O. Pingchao, "Fault diagnosis based on improved Elman neural network for a hydraulic servo system," in Robotics, Automation and Mechatronics, 2006 IEEE Conference on, 2006, pp. 1-6.

[16] M. T. Frye and R. S. Provence, "Direct inverse control using an artificial neural network for the autonomous hover of a helicopter," in Systems,

Man and Cybernetics (SMC), 2014 IEEE International Conference on, 2014, pp. 4121-4122.

[17] D. Xianglei, Z. Shuguang, H. Lvchang, and H. Rong, "The Neural Network Direct Inverse Control of Four-wheel Steering System," in Measuring Technology and Mechatronics Automation (ICMTMA), 2011 Third International Conference on, 2011, pp. 865-869.

[18] J. L. Elman, "Finding structure in time," Cognitive science, vol. 14, pp. 179-211, 1990.

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