Development of an Attitude Control System of a Heavy-lift Hexacopter using Elman Recurrent Neural Networks

Benyamin Kusumoputro, Herwin Suprijono, Muhammad Ary Heryanto, Bhakti Yudho Suprapto
Dept. of Electrical Engineering, Faculty of Engineering
Universitas Indonesia,
Kampus Baru UI, Depok, Indonesia
kusumo@ee.ui.ac.id

***Abstract*—Hexacopter is a type of multicopter that can be used to lift a heavy load, hence very convinient to be utilised in agricultural fields. As the consequence, however, the attitude control of this hexacopter is rather difficult compare with that of a quadcopter with four motors, due to gyroscopic effet of the additional motors and in its combination with the heavy loads. In this paper, we have developed a direct inverse controller system using an Elman neural networks for the attitude and altitude control of the hexacopter. Experiments are conductet using a flight data taken from a test-bed system. Results show that the attitude characteritics of the heavy-lift hexacopter can be controlled successfully, especially when an optimized Elman neural networks as the direct inverse controller system is utilized.**

Keywords—heavy-lift hexacopter; direct inverse control scheme; artificial neural networks; DIC optimization

I. INTRODUCTION

The development of an unmanned aerial vehicles (UAV) system and its performance improvement has been rapidly studied in this decade, due to its application in surveillance, search and rescue mission, law enforcement, and others. Today, the design and control of multi-copters with higher than four rotors, i.e., hexacopter and octocopter [1-5], is a chalenging research topics because of using more motors the total payload [2] and the additional flying time or the distance coverage could be extended. However, as controlling the hexacopter is not as easy the ussualy used quadcopter, the attitude control of the dynamical movement of the hexacopter is more difficult to develop.

Various type of control approaches are widely studied and utlized, including PID [3], PID-LQR [4], input-output feedback linearization [5], and Back Stepping control algorithms [6]. However, these controller system show limitation when it is applied to a nonlinear system such as the hexacopter, due to its difficulty to response the dynamic change of the flight environment. Recently, there has been a significant increase in the number of control system methods that are developed based on nonlinear concepts. The nonlinear inverse model based control is one of availability of the inverse of the plant model. As the neural networks has the ability to model any nonlinear system, including their inverse, neural networks based direct inverse controller system the such methods, which is depended on the has been applied in various flight control applications [7,8].

A neural networks is a parallel distributed processor made up of a simple neuron, with the main function is to memorize the knowledge of the system by mimicking the mathematical model of the nonlinear system after training. A simple but powerful neural networks is a multi-layer perceptron (MLP) with one hidden layer, trined by a back-propagation learning methodfor updating the neural networks parameters. However, back-propagation neural networks (BPNN) has shown disadvantages such as lower speed of convergence, easy to fall in local minima, which degrades the performance accuracy of the memorized mathematical model of the system.

In order to improve the characteristics of the usually used BPNN based control system for a hexacopter's manueverability, in this paper, an Elman reccurent neural networks (ERNN) [9] based direct inverse controller system is developed. Theoretically, the ERNN is more appropiate to model any complex models and nonlinear system [10], as in the ERNN, a delay operator is structurally added to the back propagation learning mechanism, which increasing the dynamic time-varying capability and global stability of the networks.

This paper is organized as follows. Section 2 describes the platform of the hexacopter dynamic and the kinematics model. Section 3 describes the direct inverse control neural networks using the Elman recurrent learning algorithm. Experiments on our developed hexacopter is conducted and presented in Section 4, followed by analysis of the ERNN based control system. Finally, the summary is presented in the last section.

II. HEAVY-LIFT HEXACOPTER AND ITS DYNAMIC MODELLING AND CONTROL SYSTEM

The hexacopter is an under-actuated and unstable system that many works used a simplified model in order to ignore the nonlinear effect of the system [11]. Hexacopter has two types of frame configurations, Plus (+) and X configuration, and moving within six degree of freedom (6 DOF) system. In its movement, hexacopter utilized the thrust generated by the combination of the rotation of the six-propellers at each motors in the end of the axes. As different frame configuration has different dynamic movement model, in this research, we developed a heavy-lift hexacopter with a Plus (+) frame configuration and derive the dynamic movement of the model accordingly.

The hexacopter forces and moments are generated by the angular velocity of each rotor. Rotation of the rotor generated three kinds of forces and moments which are the thrust force, the horizontal force and the moment drag. As the hexacopter modeling poses gyroscopic effect generated by the rigid body rotation, eliminating this effect is conducted by providing an opposite rotation on each pair of the rotors.

Describing the dynamic modelling of a hexacopter, two reference frame systems are necessary, as shown in Fig. 1. Reference E frame (E, X_E, Y_E, Z_E) defines the position of the linear and angular position of the hexacopter with respect to the earth, while reference B frame (B, X_B,Y_B, Z_B) is lied in the axis of the hexacopter body. The angular position of the B frame with respect to the E frame is usually defined by Euler angles, i.e., the roll *(ϕ)*, the pitch *(ϕ)*, and the yaw *(ψ)*; and the transformation from the B frame to the E frame is realized by a rotation orthogonal matrix R as follows:

$$R=\begin{bmatrix} \cos\theta\cos\psi & \cos\psi\sin\theta\sin\phi-\cos\phi\sin\psi & \cos\phi\cos\psi\sin\theta+\sin\phi\sin\psi \\ \cos\theta\sin\psi & \cos\phi\cos\psi+\sin\theta\sin\phi\sin\psi & \cos\phi\sin\psi\sin\theta-\cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \quad (1)$$

The dynamic basic movement equations of the hexacopter in E frame can be written as:

$$\begin{cases} \ddot{X} = -(\sin\psi\sin\phi+\cos\psi\sin\theta\cos\phi)\dfrac{u_1}{m} \\ \ddot{Y} = (-\cos\phi\sin\theta\sin\psi-\cos\psi\sin\phi)\dfrac{u_1}{m} \\ \ddot{Z} = -g+(\cos\theta\cos\phi)\dfrac{u_1}{m} \\ \ddot{\phi} = (\dot{\theta}\dot{\psi}(I_{yy}-I_{zz})+\tau_x)/I_{xx} \\ \ddot{\theta} = (\dot{\phi}\dot{\psi}(I_{zz}-I_{xx})+\tau_{yy})/I_{yy} \\ \ddot{\psi} = (\dot{\phi}\dot{\theta}(I_{xx}-I_{yy})+\tau_z)/I_{zz} \end{cases} \quad (2)$$

while the relations of those basic movements to the squared of the propeller velocity can be written as:

$$\begin{cases} U_1 = b\left(\omega_1^2+\omega_2^2+\omega_3^2+\omega_4^2+\omega_5^2+\omega_6^2\right) \\ U_2 = l\,b\left(-\omega_2^2-\omega_3^2+\omega_5^2+\omega_6^2\right) \\ U_3 = l\,b\left(-\omega_1^2-\omega_2^2-\omega_6^2+\omega_3^2+\omega_4^2+\omega_5^2\right) \\ U_4 = d\left(-\omega_1^2+\omega_2^2-\omega_3^2+\omega_4^2-\omega_5^2+\omega_6^2\right) \\ \omega = -\omega_1+\omega_2-\omega_3+\omega_4-\omega_5+\omega_6 \end{cases} \quad (3)$$

where $\ddot{X}, \ddot{Y}, \ddot{Z}$ the hexacopter linear acceleration with respect to the E frame, $\ddot{\phi}, \ddot{\theta}, \ddot{\psi}$ the hexacopter angular acceleration around with respect to the B frame, m the mass of the hexacopter, g the acceleration due to gravity, I_{XX}, I_{YY}, I_{ZZ} the body moment of inertia around the *xyz*-axis, *p, q, r* the hexacopter angular velocity with respect to the B frame, U_1 the vertical thrust, U_2 the roll torque, U_3 the pitch torque, U_4 the yaw torque, Ω_n the propeller speed, b the thrust factor, d the drag factor, and l the distance between the center of hexacopter and the center of the propeller.

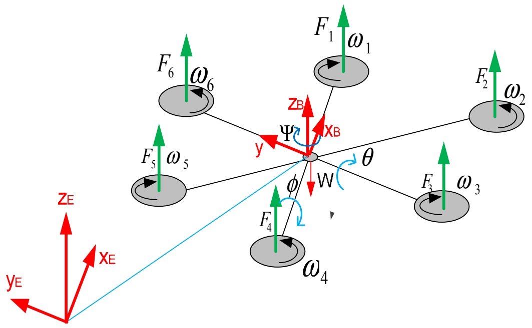


Fig. 1. Relationship of two frames reference

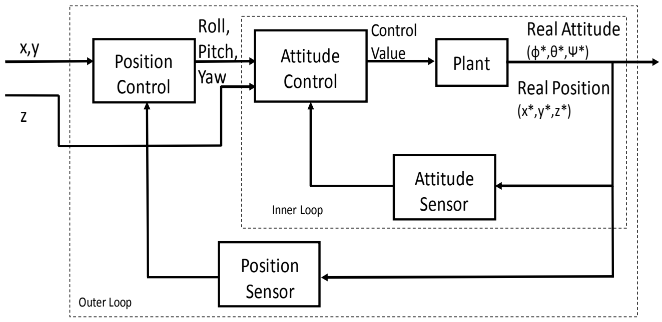


Fig. 2. Block Diagram Block of the Hexacopter Control system

Block diagram of the control system for a hexacopter in general is depicted in Fig. 2. The attitude control, which the main function is to control the roll *(ϕ)*, pitch *(θ)* and yaw *(ψ)* of the hexacopter to be as close as the input reference, is called the *inner loop control*. While the position control, which action is to move the hexacopter in the vertical plane (attitude control) and in the horizontal plane (maneuvering control), is then called the *outer loop control*.

III. Neural NEtworks Based Direct Inverse Control System

The nonlinear inverse model based control strategy is one of the promising methods within various nonlinear conceps that are being developed recently [12], and an artificial neural networks has been studied as one of the most accurate controller system for a nonlinear dynamical system [13, 14]. The controller system is implemented by training the inverse of the of the nonlinear plant using a neural network, and by cascading this neural network inverse model with the hexacopter, an identity mapping between the signal reference of the system and the output of the plant is obtained. Block diagram of the neural networks based direct inverse controller scheme (NN-DIC) is presented in Fig. 3.

As can be seen in this figure, NN-DIC scheme can be experimentally simulated by using a system identification and an inverse model. For the system identification of the hexacopter, a neural networks with backpropagation learning method is utilized through an identification learning mechanism as depicted in Fig. 4a. While for the inverse control system, an Elman recurrent neural networks is utilized through an inverse model learning mechanism as depicted in Fig. 4b. The output of inverse model are PWM signals that used to control the speed of BLDC motors in the hexacopter.

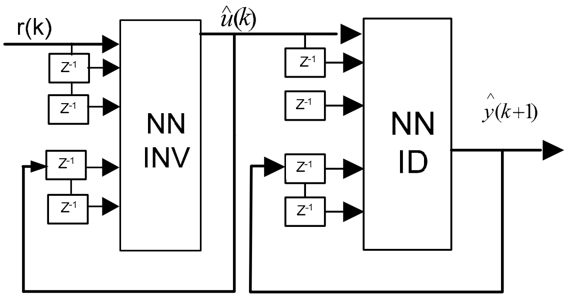
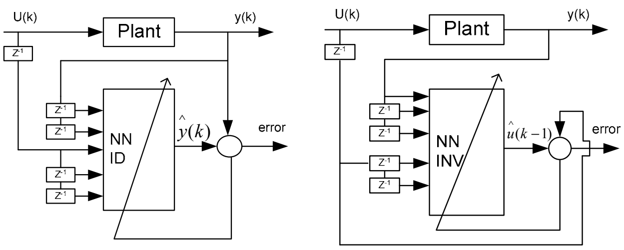


Fig. 3. Block diagram of DIC



(a) (b)

Fig. 4. The block diagram of a) identification training, b) Inverse model

A. Elman Recurrent Neural Networks based Inverse Model

Elman Recurrent Neural Network (ERNN) [8,9,15] is a class of a neural network that consists of four layers, i.e., an input layer and a context layer, a hidden layer and an output layer. The neural networks system, usually Backpropagation based learning mechanism, has been widely researched for the purpose of system identification, predicting, fault diagnosis and forecasting [10,13,16]. In this paper, however, we have used an ERNN system, which used a learning algorithm that is similar to the Back Propagation learning mechanism.

Fig. 5 shows the neural architectural of the ERNN, which emplyos a context layer (*C(t-1)*), in order to memorize the previous activations of the hidden neuron, which can be considered as a one-step time delays function. During learning, the feed forward connections weight, i.e., the hidden layer and the output layer, are updated until converge, while the reccurent weights are fixed.

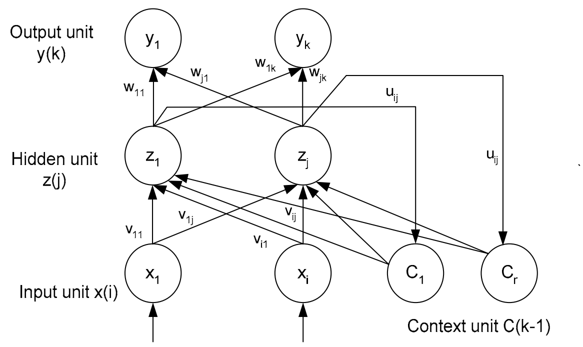


Fig.5. Elman Recurrent Neural Network Architecture

At a specific time *k*, the previous activations of the hidden neurons (at time *t-1*) and the current input are used as the inputs to the network. The activation of the hidden neurons is propagated through forward calculation to produce the output of the network. The standard back-propagation learning rule is then employed to train the network by updating the neuron weights.

After updating all of the neuron weights, the activation of the hidden neurons at time *k* are sent back through the recurrent links to the context units and saved there for the next training step (time *t+1*). The learning process of the ERNN as an inverse system is the same with that written in [14]. Suppose, *X(k)* is the input vector, and $C_j(k)$ represents the activation of the context neuron *j*, *y(k)* denoted the output vector, the *i*-th hidden neuron activation and the activation of the output neuron *y(k)* can be calculated forward through:

$$z_j(t) = f((\sum u_{ij}(t-1)C_j(t) + v_{ij}(t-1)x(t)) + b1) \quad (4)$$

$$C_i(t) = z_j(t-1) \quad (5)$$

$$y_k(t) = g((\sum w_{ij}(t-1)z_j(t)) + b2) \quad (6)$$

where w_{ij} denoted the weights of the hidden layer to the output layer, u_{ij} the weights of the context layer to the hidden layer, and v_{ij} the weights of the input layer to the hidden layer, Z_j denoted the activations of the hidden layer, and C_j is the activations of the context layer, respectively. Noted that *f* (.) is a transfer function of the neuron of hidden layer, and *g*(.) is a transfer function of output layer, both are using a Sigmoid function, and b_1 and b_2 are the bias of the input layer and the output layer, respectively.

Elman neural network used an optimization gradient-falling algorithm, which is actually a backpropagation algorithm with an adaptive learning speed and momentum gradient-falling that improve the training speed and effectively inhibit the network to be traped into a local minima. The learning algorithm is used to modify the weights and biases of the neural networks by making a minimization of the error sum of squares of the difference between actual output and the output sample.

Suppose the actual output vector at a *t* step is Y_d *(t),* then the error function is defined as follows:

$$E(t) = \frac{1}{2}(Y_d(t) - Y(t))^2 \tag{7}$$

and by using a partial derivative of this error for w_{jk} ,v_{ij}, the weights of the networks, a modifying formula for the weight update are:

$$\Delta w_{jk}(t) = (1-\mu)\eta((Y_d(t) - Y(t))f^{'}(z_j(t) + \mu\,\Delta w_{jk}(t) \tag{8}$$

$$\Delta v_{ij}(t) = (1-\mu)\eta((Y_d(t) - Y(t))w_{jk}(t-1)g^{'}(x(t) + \mu\Delta v_{ij}(t)) \tag{9}$$

where η the learning speed, μ momentum factor which is determined to be *0.9*. Theoretically, by using the momentum term, the weights are not only updating by the gradient direction at current moment only, but using also the gradient direction at previous moment, which inducd a higher speed of convergence.

IV. Neural Networks Based Direct Inverse Control System

In this research, experiments are done by using a real data flight of a heavy-lift hexacopter in a stationary test-bed system for training the Elman recurrent neural networks based inverse (ERNN-INV) system. The characteristics of the heavy-lift hexacopter used in this research can be explained as follows. The total span of the heavy-lift hexacopter is *1.7* m with an each motor on the corner of the structure provides a total thrust of *12.34* kg. The main components of the hexacopter are: six Flame of 80A ESC, six T-motor U11 100 KV-BLDC motors, six Carbon Fiber propellers *15"*x *5.5*", a microcontroller, a compass and GPS sensor, an Inertial Measurement Unit (IMU) sensor that consist of gyroscope, accelerometer and barometer, a radio control, a voltage regulator, and a Li-Po battery, respectively.

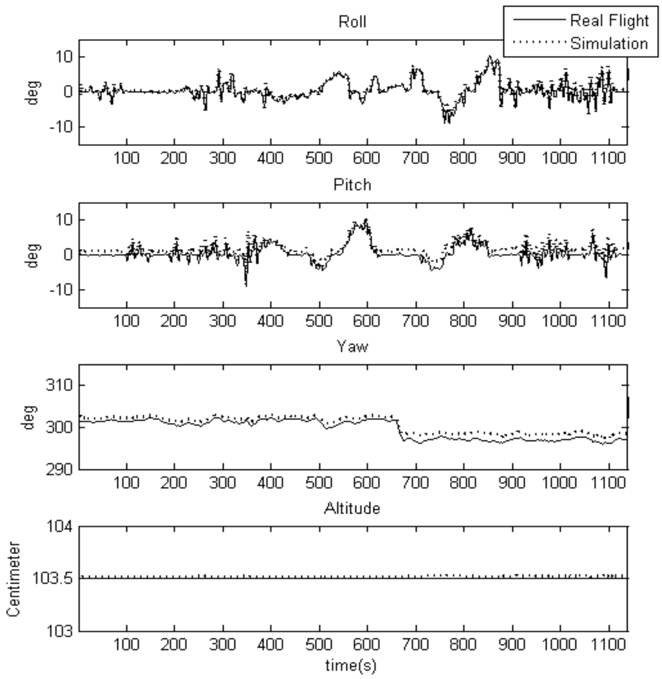


Fig. 6. Simulation result of the attitude and altitude control of the hexacopter using ERNN-INV system

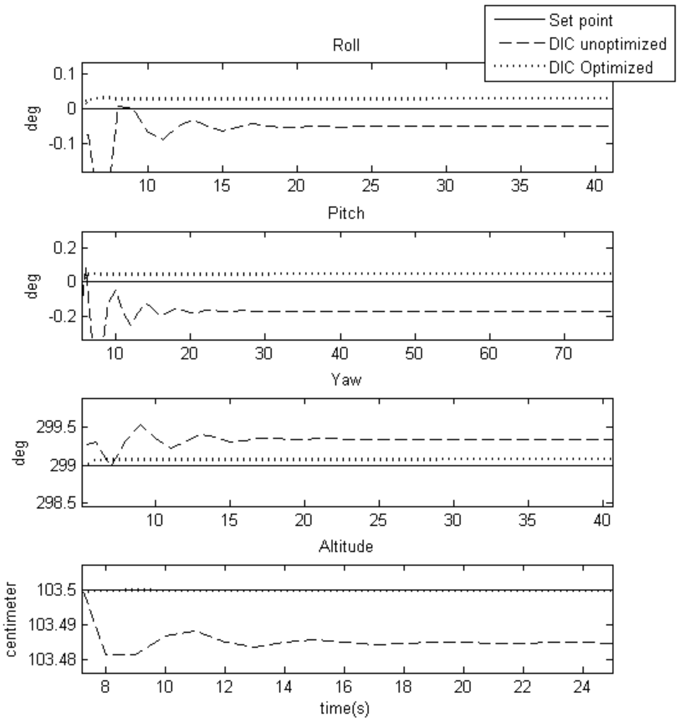


Fig. 7. Comparison of the attitude and altitude control result using ERNN-INV system and the Optimized ERNN-INV system.

Figure 6 shows the simulation results of the hexacopter's attitude parameters of the ERNN-INV system. As can be clearly seen in this figure, the response of the ERNN-INV system follows the data set with a very small mean sum square error (MSSE) of 0.0023. In order to increase the accuracy of the controller system by means of decreasing the total MSSE error, especially the settling time error, an optimization of the ERNN-INV system is proposed.

Optimization procedure is conducted by firstly determined the input set point for all of the attitude parameter, i.e., the roll = 0^0, the pitch = 0^0, the yaw = 299^0, and the altitute = 1.035 m. Using this condition as the reference input, a new training datasets are generated and used to re-train the ERNN-INV system. The new ERNN-INV system is then again utilized to follow the test dataset that has been used as in the same experiments such as in Fig. 6. The MSSE of the Optimized ERNN-INV is found to be 0.0017, which is much smaller than that of the unoptimized ERNN-INV system.

Figure 7 shows the comparison results of the attitude profile of the heavy-lift hexacopter under the test-bed system when using the ERNN-INV system, and the Optimized ERNN-INV system, as the the controller system in a direct inverse control (DIC) scheme. In order to highlight the difference of both the ERNN-INV system responses, especially the settling time period, we have only presented a part of the time sequence of the hexacopter movement.

As can be seen in this figure, the attitude error of the Optimized ERNN-DIC system is lower compare with that of the unoptimized ERNN-INV system. It is also clearly shown that the optimized ERNN-INV system produces a roll error of 0.03^0 in average, a pitch error of 0.05^0, and a yaw error of 0.01^0, while the attitude error is nearly zero. The MSSE for the Optimized ERNN-INV system is 8.6 10^{-5}, compare with that of 7.2 10^{-4} when using the unoptimized ERNN-INV system. This figure also shown that the settling time for all of the attitude parameters is very small, with an average of 6.7 second, compare with that of 20 second when using the unoptimized ERNN-INV system. The characteristics comparison of those controller systems are shown in Table 1.

TABLE I. COMPARISON OF THE HEXACOPTER CONTROL CHARACTERISTICS USING AN ERNN-INV SYSTEM AND THE OPTIMIZED ERNN-INV SYSTEM

Controller System	**Mean Sum Square Error**			**Settling Time** *(second)*
	Roll	***Pitch***	***Yaw***	
Unoptimized ERNN-INV	0.055	0.134	0.480	20
Optimized ERNN-Inv	0.010	0.054	0.100	6.7

V. CONCLUSIONS

In this paper, a neural networks based inverse controller system is developed for controlling the attitute characteristics of a heavy-lift hexacopter. Instead of usually used Backpropagation as the neural networks learning mechanism, an Elman recurrent neural networks inverse (ERNN-INV) learning system is utilized, where a delay operator is embedded in the neural structure. Optimization of the ERNN-INV system is further conducted for increasing the accuracy of the hexacopter characteristics, i.e., attitude and altitude parameters, by re-training the controller with a defined attitude condition. Experiments are conducted for comparing the performance characteristics of the hexacopter using those controller systems. Results show that the unoptimized ERNN-INV system has lower altitude and latitude error compare with that of the unoptimized ERNN-INV system. Further experiments are conducted in order to investigate, compare and analyze the control parameters characteristics when a Backpropagation learning mechanism is utilized.

ACKNOWLEDGMENT

The Authors would like to gratefully acknowledge the Ministry of Research, Technology and Higher Education through the Universitas Indonesia Research Funding 2016.

REFERENCES

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

[2] A. Alaimo, V. Artale, C. Milazzo, A. Ricciardello, and L. Trefiletti, "Mathematical modeling and control of a hexacopter," in *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*, 2013, pp. 1043-1050.

[3] 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.

[4] 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.

[5] R. Baranek and F. Šolc, "Modelling and control of a hexa-copter," in *Carpathian Control Conference (ICCC), 2012 13th International*, 2012, pp. 19-23.

[6] C. 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.

[7] V. Artale, M. Collotta, G. Pau, and A. Ricciardello, "Hexacopter trajectory control using a neural network," in AIP Conference Proceeding, International Conference of Numerical Analysis and Applied Mathematics, 2013.

[8] M. Collotta, G. Pau, and R. Caponetto, "A real-time system based on a neural network model to control hexacopter trajectories," in *Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM), 2014 International Symposium on*, 2014, pp. 222-227.

[9] S. Ding, Y. Zhang, J. Chen, and W. Jia, "Research on using genetic algorithms to optimize Elman neural networks," *Neural Computing and Applications,* vol. 23, pp. 293-297, 2013.

[10] R. Lina, L. Yanxin, R. Zhiyuan, L. Haiyan, and F. Ruicheng, "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.

[11] S. Bouabdallah, M. Becker, and R. Siegwart, "Autonomous miniature flying robots: Coming soon!", IEEE Robotics & Automation Magazine, vol. 14., no. 3, pp. 88-98, 2007.

[12] D. Pham and X. Liu, "Training of Elman networks and dynamic system modelling," *International Journal of Systems Science,* vol. 27, pp. 221-226, 1996.

[13] K.S. Narendra, and K. Parthasarathy,"Identification and control of dynamical systems using neural networks," *IEEE Trans, Neural Netw.*, 1(1), pp. 4-27, 1990.

[14] Y.-C. Cheng, W.-M. Qi, and J. Zhao, "A new Elman neural network and its dynamic properties," in *Cybernetics and Intelligent Systems, 2008 IEEE Conference on*, 2008, pp. 971-975.

[15] 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.

[16] B. Kusumoputro, K. Priandana, W. Wahab, "System Identification and Control of Pressure Process RIG® System Using Back Propagation Neural Networks," *ARPN Journal of Engineering and Applied Sciences*, vol. 10, no. 16, pp. 7190 – 7195, September 2015.