Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Dynamic-Fuzzy-Neural-Networks-Based Control of an Unmanned Aerial Vehicle

Zhe Tang* , Meng Joo Er, Yi Zhou*****

* *School of Information Science and Engineering, Central South University, Changsha, 410083, China*
***Intelligent Systems Centre 50 Nanyang Drive, 7th Storey, Research Techno Plaza, BoderX Block, Nanyang Technological University, Singapore 637533 (e-mail: Emjer@ntu.edu.sg)*
****School of Electrical and Electronics Engineering Singapore Polytechnic 500 Dover Road, Singapore 139651 (e-mail: zhouyi@sp.edu.sg)*

Abstract: In this paper, hierarchical control of an UAV (unmanned aerial vehicle) is proposed. The proposed controller is made up of two sub-systems, namely inner-loop controller and outer-loop controller. The inner-loop controller is an attitude control system while the outer-loop control system is a trajectory control system. In the proposed architecture, a DFNN-(dynamic-fuzzy-neural-network-) based reference model controller is deployed. Hover motion has been implemented to demonstrate the effectiveness of the proposed controller. Both the PD controller and BP (back propagation) NN-based controller have been developed for the control of an UAV. The input of the DFNN is the position of an UAV while the output is the desired force to control the UAV. Simulation results show that the DFNN has faster convergence speed than the PD and the BPNN. Furthermore, the DFNN is able to produce the desired force to achieve hover motion at any positions in a given area.

1. INTRODUCTION

An UAV (unmanned aerial vehicle) [1] is a powered, aerial vehicle that does not carry a human operator. It uses aerodynamic forces to provide vehicle lift, can fly autonomously or piloted remotely, can be expendable or recoverable.

UAVs have many potential applications such as reconnaissance, surveillance, tracking of individuals or other objects of interests, and tracking monitoring. The control of an UAV poses many challenges due to various factors, such as parametric uncertainties (changing mass and aerodynamic characteristics), unmodeled dynamics, actuator magnitude and rate saturation.

Neural networks, which have the ability to approximate general continuous nonlinear functions, are ideal for adaptive flight control application [2]. One advantage of the NN is the ability to learn how to perform tasks based on the data given for training or initial experience. In addition, NNs require much less memory than a simple lookup table. Furthermore, NNs can function as highly nonlinear adaptive control elements and offer distinct advantages over more conventional linear parameter adaptive controllers.

R. San *etc*. [3] use supervised NN for the modelling an UAV. NN-based system identification for an UAV is more computationally efficient than conventional methods. In their work, a model was chosen in attitude and position due to the presence of noise in the velocity for slow short motions. One main disadvantage is that it requires different NNs for different flight stages. Eric and his team [4, 5, 6] use a single hidden layer NN to cancel model error arising from the approximate linearization in their proposed UAV controller. A nonlinearly parameterized NN is used to provide on-line adaptation. Gordon and Jonathan [7] use a NN as a purely reactive UAV controller. Many other NN-based flight controller are surveyed in [8].

In this paper, a more efficient NN, DFNN (dynamic fuzzy neural network), is deployed in the model reference controller for a UAV control system. The control architecture consists of two loop systems. The inner loop system is to control attitude, the outer loop is to control position. A thorough simulation experiments are conducted to prove the effectiveness of the DFNN-based UAV controller. The rest of this paper is organized as follows: The dynamics and control architecture of an UAV is described in section 2 and section 3 respectively. A DFNN based controller is proposed in Section 4. Simulation results and conclusions are given in the section 5 and final section.

2. VELCHILE DYNAMICS

The dynamics of an UAV can be described as the following equations [4]:

$$v = \dot{s} \tag{1}$$

$$\dot{v} = F(s, v, q, \omega, \delta) \tag{2}$$

$$\dot{\omega} = M(s, v, q, \omega, \delta) \tag{3}$$

where s is position vector, v is the velocity of an UAV, q is attitude quaternion, ω is angular velocity. Equ.2 represents translational dynamics and Equ.3 represents the attitude dynamics. The state vector can be defined as $x = [s^T v^T q^T \omega^T]^T$. The control vector is denoted by $\delta = [\delta_t \delta_e \delta_r \delta_a]$, where δ_t denotes the primary throttle force, δ_e denotes elevator, δ_r denotes rudder, δ_a denotes aileron.

The translational motion of the body-fixed coordinate frame is given below

$$F_b = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m(\dot{v} + \omega \times v) \tag{4}$$

$$v = \dot{s} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \ \omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{5}$$

where the subscript b means body frame, the applied forces $[F_x F_y F_z]^T$ are in the body-fixed frame, and the mass of the body m is assumed constant, p, q, and r are roll rate, yaw rate and yaw rate respectively.

The rotational dynamics of the body-fixed frame are given below

$$M_b = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = I\dot{\omega}_b + \omega \times (I\omega) \tag{6}$$

$$I = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \tag{7}$$

where the applied moments are $[L\ M\ N]^T$, and the inertia tensor I is with respect to the origin O.

3. CONTROLLER ARCHITECTURE

The architecture of the controller for the UAV is given in Fig. 1. There are two sub-systems in this controller. The inner-loop system is the attitude control system of an UAV, while the outer-loop control system is the trajectory control system. $(s_c v_c q_c \omega_c)$ is external command signal. s_r and v_r are the reference model states, q_{es} is the desired attitude that the outer-loop reference model states, q_r and ω_r are inner-loop outer-loop inversion expects for contributing towards achieving desired translational acceleration.

The reference model dynamics is given as:

$$\dot{v}_r = R_p(s_c - s_r) + R_d(v_c - v_r) \tag{8}$$

$$\dot{\omega}_r = K_p Q(q_c \oplus q_{des}, q_r) + K_d(\omega_c - \omega_r) \tag{9}$$

where R_p, R_d, K_p and K_d are the PD compensator gains for the inner loop and outer loop. Q is a function that, given two quaternions results in an error angle vector with three components. Explicit formulas of the gain values can be found through an analysis detailed in Ref [4] and are given as follows:

$$R_p = \frac{\omega_0^2 \omega_i^2}{\omega_i^2 + 4\zeta_0 \omega_0 \zeta_i \omega_i + \omega_o^2} \tag{10}$$

$$R_d = \frac{2\omega_o \omega_i (\zeta_o \omega_i + \zeta_i \omega_o)}{\omega_i^2 + 4\zeta_o \omega_o \zeta_i \omega_i + \omega_o^2} \tag{11}$$

$$K_p = \omega_i^2 + 4\zeta_o \omega_o \zeta_i \omega_i + \omega_o^2 \tag{12}$$

$$K_d = 2\zeta_i \omega_i + 2\zeta_o \omega_o \tag{13}$$

where ω is natural frequency, ζ is damping ratio, the subscript i, o represents the inner and outer loop values respectively.

4. DFNN-BASED CONTROLLER

4.1 Reference Model Control

The general idea behind Model Reference Adaptive Control (MRAC, also know as an MRAS or Model Reference Adaptive System) is to create a closed loop controller with parameters that can be updated to change the response of the system. The output of the system is compared to a desired response from a reference model. The control parameters are updated based on this error. The goal is for the parameters to converge to ideal values that cause the plant response to match the response of the reference model.

The architecture of MRAC with a neural controller is given in Fig.2. In our proposed UAV control architecture, this controller is to replace the PD controller in the above UAV control architecture. And a DFNN is used as NN controller in the MRAC.

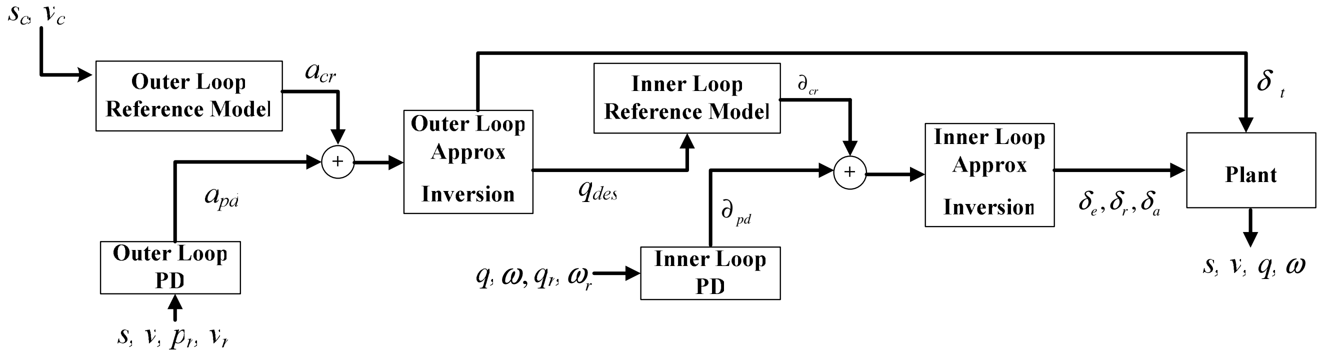

Fig. 1. UAV control architecture

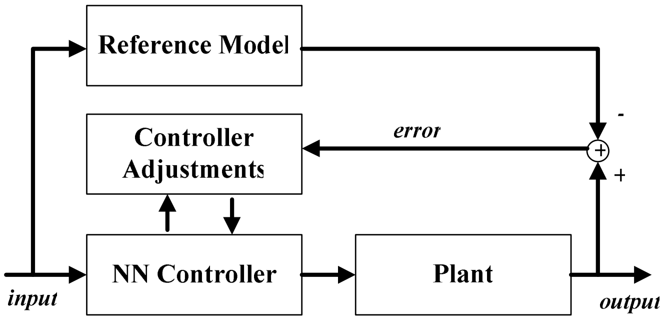

Fig. 2. Reference model controller

4.2 DFNN

DFNN with a hierarchical on-line self-organizing learning algorithm was developed in [9]. Many successful applications of DFNN have been published [10, 11]. The architecture of the DFNN is depicted in Fig.3. It consists of 5 layers.
Layer 1: Input layer. Each node represents an input linguistic variable
Layer 2: Each node represents a membership function

$$\mu_{ij} = \exp\left[-\frac{(x_i - c_{ij})^2}{\sigma_j^2}\right] \qquad i = 1,\ldots,r \qquad j = 1,\ldots,u \quad (14)$$

where c_{ij}, σ_j are the center and width of the Gaussian function respectively.
Layer 3: Each node represents a possible IF-part for fuzzy rules. The jth rule R_j is

$$R_j = \exp\left[-\frac{\sum_{i=1}^{r}(x_i - c_{ij})^2}{\sigma_j^2}\right] = \exp\left[-\frac{\| (X - C_j)^2 \|}{\sigma_j^2}\right] \quad (15)$$

$j = 1,\ldots,u$

Layer 4: This layer is to normalize the previous layer outputs.

$$N_j = \frac{R_j}{\sum_{j=1}^{u} R_j} \quad (16)$$

Layer 5: This layer is the output layer.

$$y(X) = \sum_{j=1}^{u} w_j N_j \quad (17)$$

For the TSK model, w_j can be expressed as follows:

$$w_j = k_{j0} + k_{j1}x_1 + \ldots + k_{jr}x_r \quad (18)$$

and equ.(17) can be written in the following compact form

$$Y = W_2 \psi \quad (19)$$

where

$$w_2 = [k_{10}, \cdots, k_{u0,} k_{11}, \cdots, k_{u1}, \cdots, k_{1r}, \cdots, k_{ur}]$$

$$\psi = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \vdots & \vdots \\ a_{u1} & \cdots & a_{un} \\ a_{11}x_{11} & \cdots & a_{1n}x_{1n} \\ \vdots & \vdots & \vdots \\ a_{u1}x_{11} & \cdots & a_{un}x_{1n} \\ \vdots & \vdots & \vdots \\ a_{11}x_{r1} & \cdots & a_{1n}x_{rn} \\ \vdots & \vdots & \vdots \\ a_{u1}x_{r1} & \cdots & a_{un}x_{rn} \end{bmatrix}$$

where a_{jk} is the normalized output from layer 4, $j = 1, \cdots, u, k = 1, \cdots, n$, n is the length of the processing data. The optimal coefficient vector W_2^* can be easily solved by the well-known linear least square (LLS) method

$$W_2^* = Y \cdot (\psi T \psi)^{-1} \psi^T \quad (20)$$

The main advantage of the DFNN lies in its capabilities of reconstruction and selecting fuzzy rules automatically. It is normally used for a function or system approximation. Once a sample pattern input is fed to DFNN, it may reconstruct network to store system information given from the sample data.

For the controller in the outer loop control system for an UAV, there are two inputs and two outputs of the DFNN. The input of the DFNN is a commanded position (x_c, y_c), the output of the DFNN is the desired forces (F_x, F_y) to achieve the commanded position. Given enough training samples, the DFNN is able to generate desired control forces at any positions in a given area.

Unlike the BP training algorithms, the weight adjustment of a DFNN is realized by an inversion of a matrix which includes information of all training samples. This pattern of training can be termed as radical training, because only a calculation of a matrix inversion can generate new NN weights, which has desired approximation accuracy. For BP training algorithm, the NN weight is adjusted step by step. The same training samples need to be used repeatedly to achieve desired accuracy. However, the DFNN adjustment may not be definitely better than BP training in terms of computation cost. If there are too many samples, and the sample space is not well distributed, there will be many fuzzy rules and the size of matrix is very big. Consequently, the calculation of matrix inversion is also computationally intensive. This property makes DFNN suitable for UAV controller. For UAV control, it is difficult to obtain many good training samples. For different motion, the control architecture need different NN controller, which makes sample space limited. As a result, the matrix in the DFNN is very small, and the computation cost is low.

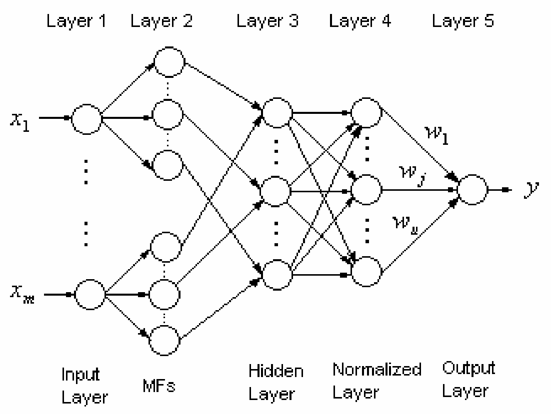

Fig. 3. DFNN

5. SIMULATION RESULTS

The UAV was commanded to perform a circular maneuver. The command trajectories are given by

$$p_c = \begin{bmatrix} \frac{V}{\omega}\cos(\omega t) \\ \frac{V}{\omega}\sin(\omega t) \\ -h \end{bmatrix}$$

$$v_c = \begin{bmatrix} -V\sin(\omega t) \\ V\cos(\omega t) \\ 0 \end{bmatrix}$$

$$\psi_c = \omega t f$$

where t is simulation time and h is a constant altitude command. V is speed of maneuver, ω is angular speed of the UAV around the local frame origin, and f is the number of pirouettes to be performed per circuit.

The inner loop frequency is given as [2.5, 2, 2.8] for the roll, pitch and yaw channels respectively, and the damping ration is 1.0. The outer loop frequency is [1, 2.2, 2.6] for the x, y, z body axis, and damping ration is 1.0. By using the above methods, the PD parameters can be obtained

$$K_d = \begin{bmatrix} 7.0 & 0 & 0 \\ 0 & 8.4 & 0 \\ 0 & 0 & 10.8 \end{bmatrix}, \quad K_p = \begin{bmatrix} 17.25 & 0 & 0 \\ 0 & 26.44 & 0 \\ 0 & 0 & 43.72 \end{bmatrix}$$

$$R_p = \begin{bmatrix} 0.3623 & 0 & 0 \\ 0 & 0.7322 & 0 \\ 0 & 0 & 1.2122 \end{bmatrix},$$

$$R_d = \begin{bmatrix} 1.0145 & 0 & 0 \\ 0 & 1.3979 & 0 \\ 0 & 0 & 1.7984 \end{bmatrix}$$

V is 3 *ft/s*, ω =1 *rad/s*. Fig.4 presents the commanded trajectory and real trajectory. The desired forces and moments to achieve commanded trajectory are given in the Fig. 5 and Fig. 6 respectively. Tracking error in the x axis for the DFNN, BPNN and PD is given in the Fig.7, while the tracking error in the y axis is given in the Fig.8. From the two figures, the DFNN based controller converges faster than both of the BPNN and PD controller. Fig.9 gives the procedure of rule generation during the DFNN training. Besides the faster convergence, another advantage of DFNN is the capability of generalization. Given enough samples, the DFNN-based controller is able to give desired control forces at any positions in a given area. Fig.10 and Fig.11 show the desired forces in a circle with the radius of 3 m.

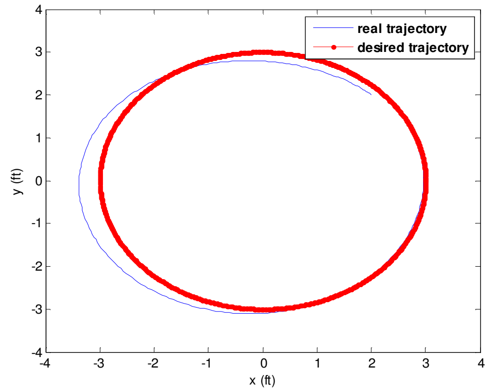

Fig. 4. Commanded trajectory and real trajectory

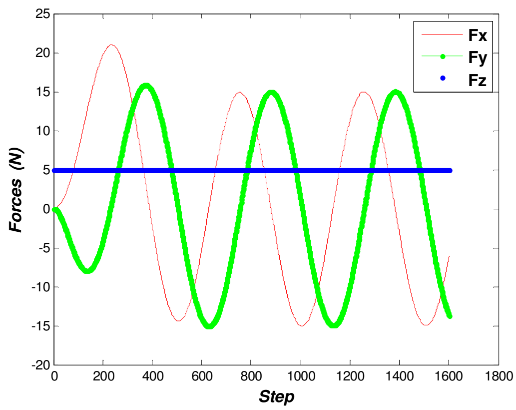

Fig. 5. Forces of the UAV

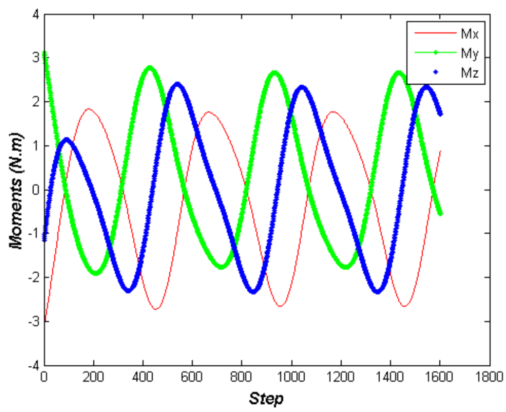

Fig. 6. Moments trajectories

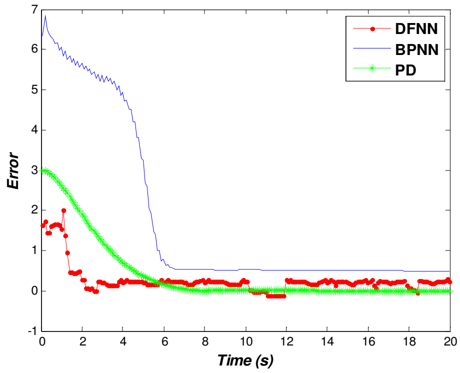

Fig. 7. Tracking error in x axis

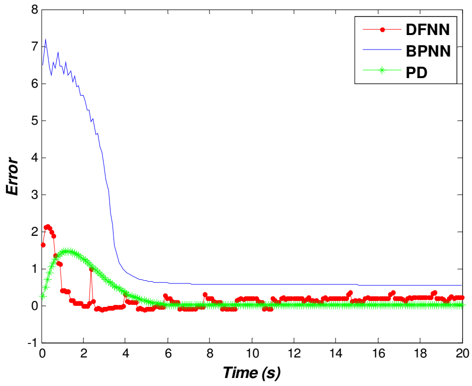

Fig. 8. Tracking error in y axis

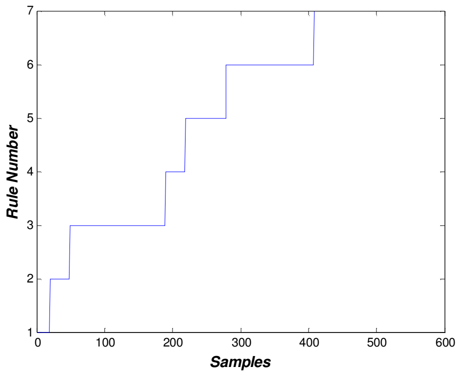

Fig. 9. Rule generation of DFNN traning

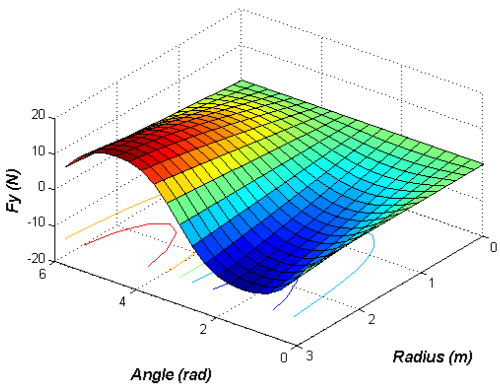

Fig. 10. F_y in a circle

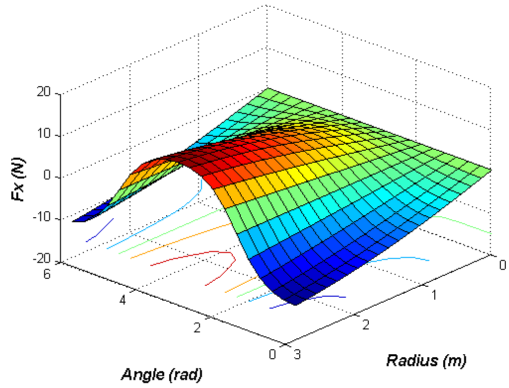

Fig. 11. F_x in a circle

6. CONCLUSIONS

The UAV control system adopted in this paper is a two loops system. The inner loop control system is to control the attitude of an UAV, while the outer loop control system is to control the position. There are two separate controllers in this control system. A DFNN based model reference controller has been adopted in the UAV control system. The simulation results of a hover motion have demonstrated the superiority to PD and BPNN controller. In our current simulation results, DFNN based model reference controller has only implemented in the outer loop control. In our future work, we will implement it in the attitude controller.

REFERENCES

[1] Office of the Secretary of Defense (2005). *Unmanned aircraft systems (UAS) roadmap, 2005 – 2030*

[2] Calise, A.J. and Rysdyk, R.T (Dec. 1998). Nonlinear adaptive flight control using neural networks. In: *IEEE Control Systems Magazine,* Vol.18, Issue 6, pp.14 - 25

[3] San Martin, R. Barrientos, A. Gutierrez, P. del Cerro, J(May 15-19, 2006). Unmanned aerial vehicle (UAV) modelling based on supervised neural networks. In: *Proceedings 2006 IEEE International Conference on Robotics and Automation*, pp. 2497 - 2502

[4] Johnson, E. N. and Kannan, S. K.(2005). Adaptive Trajectory Control for Autonomous Helicopters. *Journal of Guidance,Control, and Dynamics*. Vol. 28, No. 3, pp. 524-538.

[5] Johnson, E. N.(Dec 2000) *Limited Authority Adaptive Flight Control* , Ph.D. thesis, Georgia Institute of Technology, School of Aerospace Engineering, Atlanta, GA 30332.

[6] Johnson, E. N. and Calise, A. J.(Nov-Dec 2003). Limited Authority Adaptive Flight Control for Reusable Launch Vehicles. In: *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 26, No. 6, pp. 906-913.

[7] Wyeth G.F., Buskey G. and Roberts J. (August 30 - September 1 2000). Flight Control Using an Artificial Neural

Network. In: *Proceedings of the Australian Conference on Robotics and Automation.*, Melbourne, pp. 65 -70.

[8] P.J.Werbos (1995). Neural Networks and Flight Control: Overview of Capabilities and Emerging Applications. In: *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pp. 912-919.

[9] S. Q. Wu and M. J. Er (Apr. 2000). Dynamic fuzzy neural networks—A novel approach to function approximation. In: *IEEE Trans. Syst., Man, Cybern. B, Cybern*. Vol. 30, no. 2, pp.358–364.

[10] M. J. Er, Z. Li, H. Cai, Q. Chen (Jun. 2005). Adaptive noise cancellation using enhanced dynamic fuzzy neural networks. In*: IEEE Transaction on Fuzzy Systems*, Vo.13, NO. 3, pp.331-342.

[11] M. J. Er , C. B. Low, K. H. Nah, M. H. Lim, S. Y. Ng (2002). Real-time implementation of a dynamic fuzzy neural networks controller for a SCARA. In: *Microprocessors and Microsystems*. Vo. 26. pp.449-461