Second Order Diagonal Recurrent Neural Network

Ali Kazemy†, Seyed Amin Hosseini† and Mohammad Farrokhi†,‡
† Faculty of Electrical Engineering
‡ Center of Excellence for Power System Automation and Operation
Iran University of Science and Technology
Tehran, Iran
ali_kazemy@ee.iust.ac.ir, amin_hosseini@ee.iust.ac.ir, farrokhi@iust.ac.ir

***Abstract*—In this paper a new diagonal recurrent neural network that contains two recurrent weights in hidden layer is proposed. Since diagonal recurrent neural networks have simpler structure than the fully connected recurrent neural networks, they are easier to use in real-time applications. On the other hand, all diagonal recurrent neural networks in literature use one recurrent weight in hidden neurons, while the proposed network takes advantage of two recurrent weights. It will be shown, by simulations, that the proposed network can approximate nonlinear functions better than the existing diagonal recurrent neural networks. After deriving the training algorithm, the convergence stability and adaptive learning rate will be presented. The performance of the proposed network in model identification shows the accuracy of this network against the diagonal recurrent neural networks. Moreover, this network will be applied to real-time control of an image stabilization platform.**

I. Introduction

Neural networks are effective tools for modeling and control of nonlinear systems [1]-[3]. Most researchers use feedforward neural networks (FNNs), combined with tapped delays and the backpropagation training algorithm (BP) [4] to identify dynamical systems. The problem is that FNNs provide a static mapping and without the aid of tapped delays, it cannot approximate nonlinear dynamic systems with good accuracy. Recurrent Neural Networks (RNNs), on the other hand, have dynamic ability; hence, more suitable for dynamic systems. In Fully connected Recurrent Neural Networks (FRNNs) [5],[6], where all neurons are connected to each other, it is difficult to train the network due to massive number of weights; hence, it takes more time for weights to converge. Alternatively, Diagonal Recurrent Neural Networks (DRNNs) are a special kind of recurrent networks, which have fewer weights and shorter training time as compare to FRNNs. DRNN was first established by Ku [7]-[11]. Later on, other researchers developed the performance of this network [12]. DRNNs have been widely applied in system identification and controller design [13]-[20]. In the past decade, many recurrent neural network architectures were introduced in literatures. The structures of these networks are similar except for the recurrent neurons and their synaptic weights. Higher order DRNNs [21] use different combinations of recurrent neurons. Most of these networks are tailored for special plants and generally may not be able to identify and/or control a large class of systems. Also block-diagonal recurrent neural networks (BDRNN) [22]-[26], which were introduced recently, use a combination of a pair of neurons, called block, in their network structure, along with one tapped delay in hidden neurons. The main purpose of the taped delay is to retain the history of states of neurons. However, the recurrent weights are updated only using the previous state and cannot use other states directly. In this paper, two tapped delays are used in the hidden neurons of DRNN, hence calling it Second-order Diagonal Recurrent Neural Network (SDRNN). With this structure, more history of states of neurons can be incorporated directly into the training algorithm. A generalized dynamic backpropagation (DBP) algorithm will be derived, and it will be shown that the proposed SDRNN not only provides more accurate identification, but also a shorter training time as compare to DRNN. Moreover, convergence of the proposed dynamic backpropagation is developed and the maximum learning rate that guarantees the convergence of the algorithm is derived.

This paper is organized as follows. In section II, a comparison between the numbers of weights in the above-mentioned recurrent neural networks will be given. Section III represents the training algorithm for the proposed neural network, followed by convergence and stability analysis of the algorithm in section IV. In section V a simulation example for system identification will be given. Section VI shows the ability of the proposed neural network to control a highly nonlinear system in an experimental setup. Finally, section VII concludes the paper.

II. The Number of Weights in SDRNN, FRNN and DRNN

Let $N^T=\left\{I^p,H^q,O^r\right\}$ represent a T-type neural network with p inputs $\left(I^p\right)$, q sigmoid neurons in the hidden layer $\left(H^q\right)$, and r linear neurons in the output layer $\left(O^r\right)$. N^R, N^D and N^S represent FRNN, DRNN and SDRNN, respectively.

Let G^T be the total number of weights for a T-type neural network.

Therefore, the total number of weights (including q bias weights), for the N^R, N^D and N^S neural networks are

$$G^R=\left(p+r+1\right)q+q^2, \tag{1}$$

$$G^D=\left(p+r+2\right)q, \tag{2}$$

$$G^S=\left(p+r+3\right)q. \tag{3}$$

For instance, if $p=3$, $q=10$, and $r=1$, then $G^R=150$, $G^D=60$ and $G^S=70$. Even in this small neural network, the number of weights in FRNN is far more than that of DRNN or SDRNN; but the number of weights in SDRNN is just q weights more than that of DRNN.

III. Dynamic Training Algorithm For Second Order Diagonal Recurrent Neural Network

Fig. 1 shows the structure of the proposed neural network. Mathematical model for this network is given by

$$y(k) = O(k) = \sum_j W_j^O Z_j(k) \tag{4}$$

$$Z_j(k) = \rho\big(H_j(k)\big) \tag{5}$$

$$H_j(k) = \sum_i W_{ij}^I u_i + W_j^{D1} Z_j(k-1) + W_j^{D2} Z_j(k-2), \tag{6}$$

where $\rho(\cdot)$ is the sigmoid function.

Let $y(k)$ and $y_d(k)$ be the real and desired outputs, respectively. The error cost function is defined by

$$E(k) = \frac{1}{2}\big(y_d(k) - y(k)\big)^2. \tag{7}$$

The gradient of error in equation (7), with respect to an arbitrary weight vector W, is represented by

$$\frac{\partial E}{\partial W} = -e(k).\frac{\partial y(k)}{\partial W} = -e(k).\frac{\partial O(k)}{\partial W} \tag{8}$$

where $e(k) = y_d(k) - y(k)$ is the error between the plant and the network response. The derivatives of the output with respect to the neural network weights are

$$\frac{\partial O(k)}{\partial W_j^O} = Z_j(k) \tag{9}$$

$$\frac{\partial O(k)}{\partial W_j^{D1}} = \frac{\partial O(k)}{\partial Z_j(k)}.\frac{\partial Z_j(k)}{\partial W_j^{D1}} = W_j^O.P_j(k) \tag{10}$$

$$\frac{\partial O(k)}{\partial W_j^{D2}} = \frac{\partial O(k)}{\partial Z_j(k)}.\frac{\partial Z_j(k)}{\partial W_j^{D2}} = W_j^O.G_j(k) \tag{11}$$

$$\frac{\partial O(k)}{\partial W_{ij}^{I}} = \frac{\partial O(k)}{\partial Z_j(k)}.\frac{\partial Z_j(k)}{\partial W_{ij}^{I}} = W_j^O.Q_{ij}(k) \tag{12}$$

where

$$\begin{aligned} P_j(k) &= \frac{\partial Z_j(k)}{\partial H_j(k)}.\left(\frac{\partial H_j(k)}{\partial W_j^{D1}} + \frac{\partial H_j(k)}{\partial Z_j(k-1)}.\frac{\partial Z_j(k-1)}{\partial W_j^{D1}} + \frac{\partial H_j(k)}{\partial Z_j(k-2)}.\frac{\partial Z_j(k-2)}{\partial W_j^{D1}}\right) \\ &= \rho'\big(H_j(k)\big).\big(Z_j(k-1) + W_j^{D1}.P_j(k-1) + W_j^{D2}.P_j(k-2)\big) \end{aligned} \tag{13}$$

$$\begin{aligned} G_j(k) &= \frac{\partial Z_j(k)}{\partial H_j(k)}.\left(\frac{\partial H_j(k)}{\partial W_j^{D2}} + \frac{\partial H_j(k)}{\partial Z_j(k-1)}.\frac{\partial Z_j(k-1)}{\partial W_j^{D2}} + \frac{\partial H_j(k)}{\partial Z_j(k-2)}.\frac{\partial Z_j(k-2)}{\partial W_j^{D2}}\right) \\ &= \rho'\big(H_j(k)\big).\big(Z_j(k-2) + W_j^{D1}.G_j(k-1) + W_j^{D2}.G_j(k-2)\big) \end{aligned} \tag{14}$$

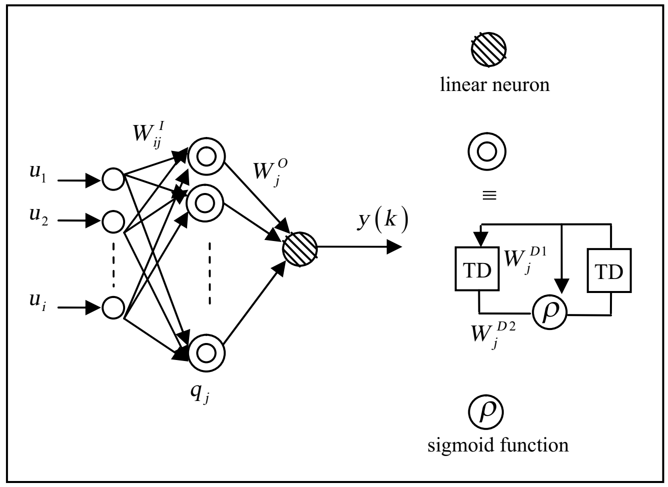

Fig. 1. Second order diagonal recurrent neural network structure

$$\begin{aligned} Q_{ij}(k) &= \frac{\partial Z_j(k)}{\partial H_j(k)}.\left(\frac{\partial H_j(k)}{\partial W_{ij}^I} + \frac{\partial H_j(k)}{\partial Z_j(k-1)}.\frac{\partial Z_j(k-1)}{\partial W_{ij}^I} + \frac{\partial H_j(k)}{\partial Z_j(k-2)}.\frac{\partial Z_j(k-2)}{\partial W_{ij}^I}\right) \\ &= \rho'\big(H_j(k)\big).\big(u_i + W_j^{D1}.Q_{ij}(k-1) + W_j^{D2}.Q_{ij}(k-2)\big) \end{aligned} \tag{15}$$

with the following initial values

$$P_j(0) = 0,\ P_j(1) = 0, \tag{16}$$

$$G_j(0) = 0,\ G_j(1) = 0, \tag{17}$$

$$Q_{ij}(0) = 0,\ Q_{ij}(1) = 0. \tag{18}$$

Therefore, the weights are adjusted by the following equation:

$$W(n+1) = W(n) + \eta\left(-\frac{\partial E}{\partial W}\right) \tag{19}$$

IV. Convergence And Stability

The following theorem is based on reference [27] with some modifications.

Theorem1: Let $g_{\max} := \max_k \|g(k)\|$, where $g(k) = \partial O(k)/\partial W$, and W is a weight vector composed of all the weight values in SDRNN and $\|\cdot\|$ is the usual Euclidean norm in R^n. Then, the convergence of the identifier is guaranteed if η_m is chosen as

$$0 < \eta_m < \frac{2}{g_{\max}^2} \tag{20}$$

Note that η_m changes adaptively during learning process of the network.

Proof: Assume there are p inputs in the input layer, q neurons in the hidden layer and one neuron in the output layer.

Given a Lyapunov function as

$$V(k) = \frac{1}{2}e^2(k) \quad (21)$$

Thus, the change of the Lyapunov function in two consecutive samples due to the training process is obtained by

$$\begin{aligned}\Delta V(k) &= V(k+1) - V(k) = \frac{1}{2}\left[e^2(k+1) - e^2(k)\right] \\ &= \Delta e(k)\left[e(k) + \frac{1}{2}\Delta e(k)\right]\end{aligned} \quad (22)$$

where $\Delta e(k)$ is defined as the difference between two consecutive error samples $\Delta e(k) = e(k+1) - e(k)$, which can be defined as

$$\Delta e(k) = \left[\frac{\partial e(k)}{\partial W}\right]^T \Delta W \quad (23)$$

Putting all weights into one vector as

$$W = \left[\left[W^I\right]^T \left[W^{D1}\right]^T \left[W^{D2}\right]^T \left[W^O\right]^T\right]^T \quad (24)$$

in which

$$W^I = \left[\left[W_1^I\right]^T \left[W_2^I\right]^T \cdots \left[W_p^I\right]^T\right]^T \quad (25)$$

$$W^{D1} = \left[\left[W_1^{D1}\right]^T \left[W_2^{D1}\right]^T \cdots \left[W_q^{D1}\right]^T\right]^T \quad (26)$$

$$W^{D2} = \left[\left[W_1^{D2}\right]^T \left[W_2^{D2}\right]^T \cdots \left[W_q^{D2}\right]^T\right]^T \quad (27)$$

and $W^O = \left[W_1^O\right]^T$. In (25)-(27) W_i^Z represents the weight vector corresponding to the *i*th neuron in the z layer ($W^I \in R^{pq}, (W^{D1}, W^{D2}, W^O) \in R^q$). Also, let

$$\eta = \begin{bmatrix} \eta^I & & & \\ & \eta^{D1} & & \\ & & \eta^{D2} & \\ & & & \eta^O \end{bmatrix} \quad (28)$$

where η^I, η^{D1}, η^{D2} and η^O represent the learning rate matrix corresponding to W^I, W^{D1}, W^{D2} and W^O, respectively, and $\eta^I = \eta_1 I_I$, $\eta^{D1} = \eta_2 I_{D1}$, $\eta^{D2} = \eta_3 I_{D2}$, $\eta^O = \eta_4 I_O$. Moreover, η_i (i=1, …, 4) is a positive constant, and I_Z is the identity matrix with z representing *I*, *D1*, *D2*, and *O*, respectively. Then

$$\begin{aligned}\Delta W &= -\eta\frac{\partial V(k)}{\partial W} = -\eta\frac{\partial e(k)}{\partial W}e(k) = -e(k).\begin{bmatrix} \eta^I & & & \\ & \eta^{D1} & & \\ & & \eta^{D2} & \\ & & & \eta^O \end{bmatrix} \\ &.\left[\left[\frac{\partial e(k)}{\partial W^I}\right]^T \left[\frac{\partial e(k)}{\partial W^{D1}}\right]^T \left[\frac{\partial e(k)}{\partial W^{D2}}\right]^T \left[\frac{\partial e(k)}{\partial W^O}\right]^T\right]^T\end{aligned} \quad (29)$$

$$\begin{aligned}\Delta e(k) &= \left[\frac{\partial e(k)}{\partial W}\right]^T \Delta W = -e(k) \\ &.\left(\eta_1\left\|\frac{\partial e(k)}{\partial W^I}\right\|^2 + \eta_2\left\|\frac{\partial e(k)}{\partial W^{D1}}\right\|^2 + \eta_3\left\|\frac{\partial e(k)}{\partial W^{D2}}\right\|^2 + \eta_4\left\|\frac{\partial e(k)}{\partial W^O}\right\|^2\right)\end{aligned} \quad (30)$$

Let

$$\lambda = \eta_1\left\|\frac{\partial e(k)}{\partial W^I}\right\|^2 + \eta_2\left\|\frac{\partial e(k)}{\partial W^{D1}}\right\|^2 + \eta_3\left\|\frac{\partial e(k)}{\partial W^{D2}}\right\|^2 + \eta_4\left\|\frac{\partial e(k)}{\partial W^O}\right\|^2 \quad (31)$$

then

$$\Delta V(k) = -\frac{1}{2}e^2(k)\left(2\lambda - \lambda^2\right). \quad (32)$$

According to the Lyapunov stability theory, if convergence must be guaranteed, then $\Delta V(k) < 0$, thus $0 < \lambda < 2$, that is

$$0 < \eta_1\left\|\frac{\partial e(k)}{\partial W^I}\right\|^2 + \eta_2\left\|\frac{\partial e(k)}{\partial W^{D1}}\right\|^2 + \eta_3\left\|\frac{\partial e(k)}{\partial W^{D2}}\right\|^2 + \eta_4\left\|\frac{\partial e(k)}{\partial W^O}\right\|^2 < 2 \quad (33)$$

Let $\eta_m = \max_{i=1}^{4}\{\eta_i\}$; thus, as long as

$$\eta_1\left\|\frac{\partial e(k)}{\partial W^I}\right\|^2 + \eta_2\left\|\frac{\partial e(k)}{\partial W^{D1}}\right\|^2 + \eta_3\left\|\frac{\partial e(k)}{\partial W^{D2}}\right\|^2 + \eta_4\left\|\frac{\partial e(k)}{\partial W^O}\right\|^2 < 2 \quad (34)$$

Equation (33) must be satisfied. So the convergence condition can be written as

$$0 < \eta_m < \frac{2}{\left\|\frac{\partial e(k)}{\partial W^I}\right\|^2 + \left\|\frac{\partial e(k)}{\partial W^{D1}}\right\|^2 + \left\|\frac{\partial e(k)}{\partial W^{D2}}\right\|^2 + \left\|\frac{\partial e(k)}{\partial W^O}\right\|^2} \quad (35)$$

Note that $\|\cdot\|$ is the Euclidean norm, therefore

$$\left\|\frac{\partial e(k)}{\partial W^I}\right\|^2 + \left\|\frac{\partial e(k)}{\partial W^{D1}}\right\|^2 + \left\|\frac{\partial e(k)}{\partial W^{D2}}\right\|^2 + \left\|\frac{\partial e(k)}{\partial W^O}\right\|^2 = \left\|\frac{\partial e(k)}{\partial W}\right\|^2 \quad (36)$$

Now let

$$g(k) = \frac{\partial e(k)}{\partial W} = \frac{\partial O(k)}{\partial W} \quad (37)$$

and let $g_{\max} = \max_k \|g(k)\|$, then (20) follows. o

*Theorem2:*Let $g_{\max} := \max_k \|g(k)\|$ and $S_{\max} := \max_k \|S(k)\|$ where $g(k) = \partial O(k)/\partial W$ and $S(k) = \partial y(k)/\partial u(k) = y_u(k)$, and W is a weight vector composed of all the weight in the SDRNN, and $\|\cdot\|$ is the Euclidean norm in R^n. Then, the convergence of the controller is guaranteed if η_{mc} is chosen as

$$0 < \eta_{mc} < \frac{2}{S_{\max}^2 g_{\max}^2} \quad (38)$$

Note that $S(k)$ is the sensitivity of the plant output with respect to its input.

Proof: Same as in Theorem 1, it can be written

$$\Delta W = -\eta y_u(k)\frac{\partial e(k)}{\partial W}e(k) \quad (39)$$

and the rest of proof is straight forward. o

V. Comparison Between SDRNN And DRNN On System Identification

Consider the following plant model

$$\begin{aligned} y(k+1) &= 0.2y(k) + 0.2y(k-1) + 0.2y(k-2) \\ &+ \sin\left[0.5\left(y(k) + y(k-1) + y(k-2)\right)\right] \\ &\cdot \cos\left[0.5\left(y(k) + y(k-1) + y(k-2)\right)\right] \end{aligned} \quad (40)$$

The proposed neural network in this paper and the DRNN are employed to identify the system in (40). There are one input, 10 neurons in the hidden layer and one output for both networks. Fig. 2 shows the sum of squared error for both networks.

Fig. 3 and Fig. 4 show the comparison on the model identification and the adaptive learning rate (η_m), respectively.

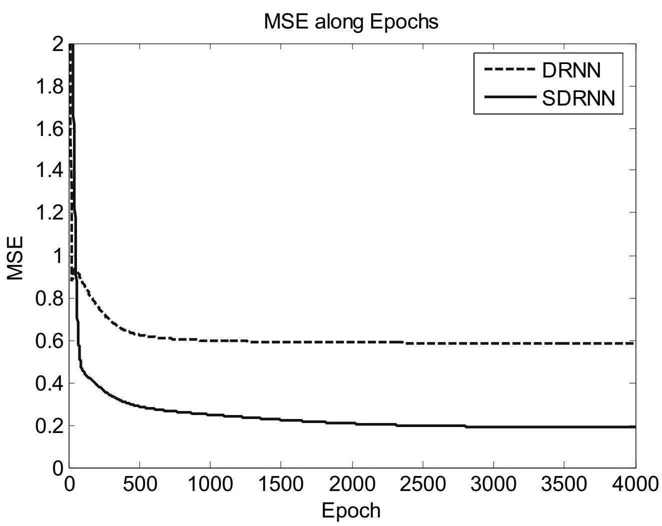

Fig. 2. MSE along learning

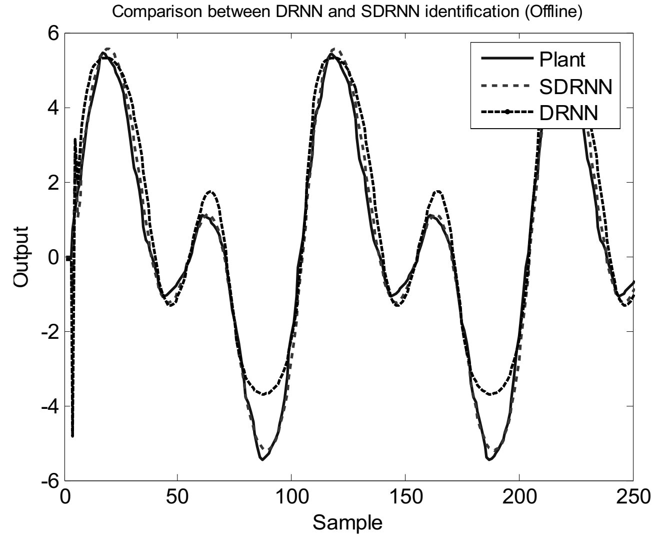

Fig. 3. Comparison of SDRNN and DRNN in system identification

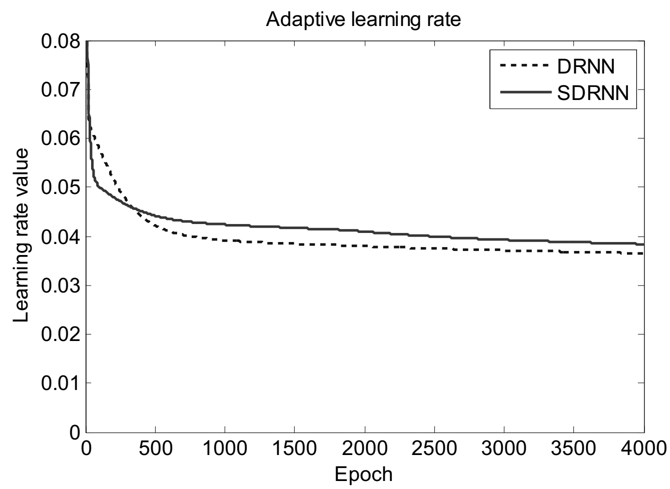

Fig. 4. Adaptive learning rate

VI. Experimental Results

The proposed SDRNN is used to control a submarine periscope mirror. Fig. 5 shows the experimental setup of this system.

Fig. 6 shows the schematic diagram of a periscope, where θ_1 and θ_2 are the motions imposed from the sea waves on the submarine and the periscope, along the roll and the pitch axes, respectively. It is assumed that gyroscopes measure these angles. Image stabilization equation maps these angles to the three dimensional space. In this paper, the mirror is controlled for the line of sight (LOS) stabilization. Fig. 7 shows the proposed control block diagram. The proposed scheme is based on [12], in which the identifier is replaced by the proposed SDRNN. This identifier estimates the required $\partial y / \partial u$ in the NN controller.

Figs. 8 to 10 show the reference tracking on θ_4 and θ_5, and the error between the reference and the plant output, respectively.

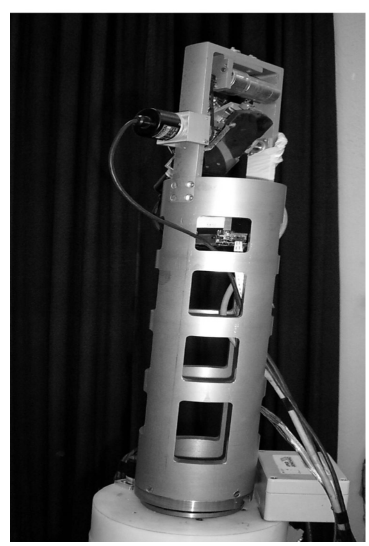
Fig. 5. Experimental setup of periscope

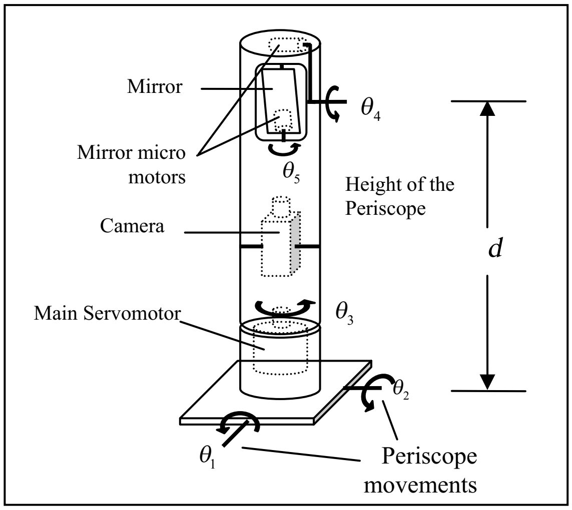

Fig. 6. Schematic of the Periscope

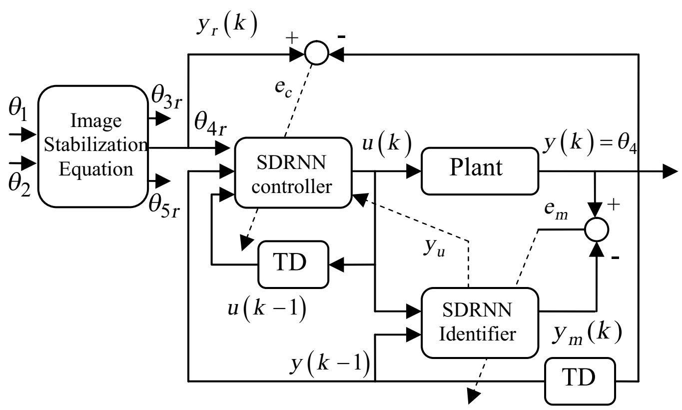

Fig. 7. Control block diagram

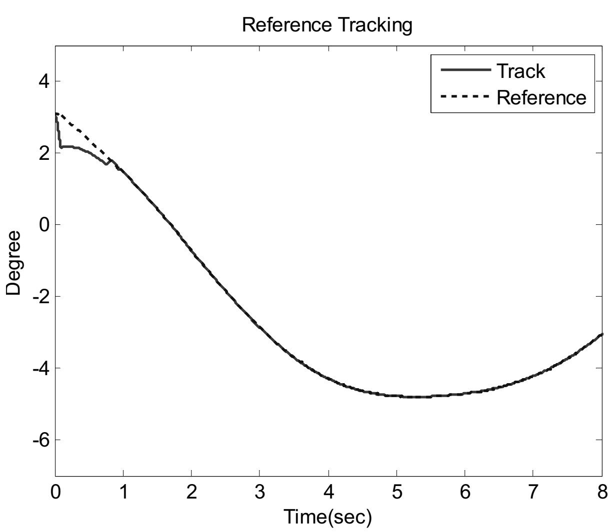

Fig. 8. Reference and model output response on θ_4

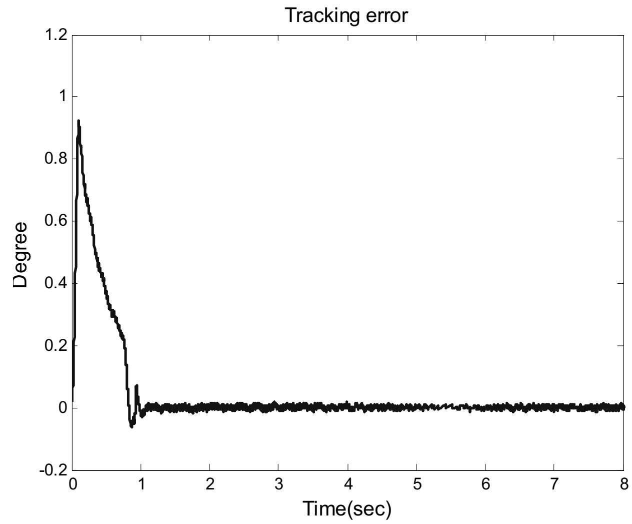

Fig. 9. Tracking Error on θ_4

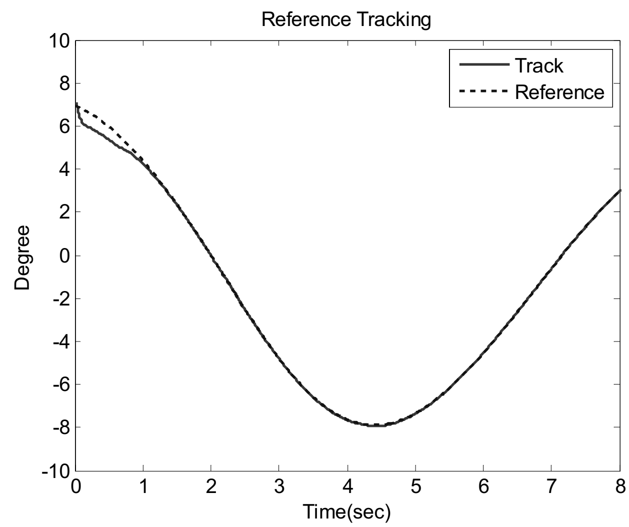

Fig. 10. Reference and model output response on θ_5

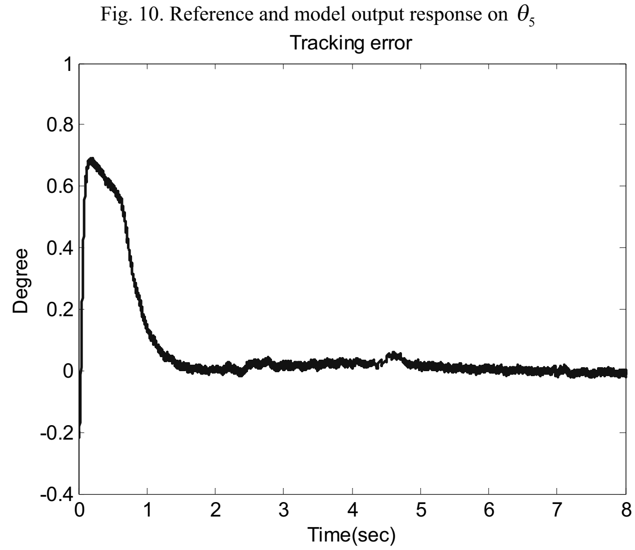

Fig. 11. Tracking Error on θ_5

VII. CONCLUSION

In this paper, a new diagonal recurrent neural network that contains two recurrent weights for every hidden neuron was proposed. It was shown that using two recurrent weights could help improve the estimation property of recurrent networks while the computation burdens are not as much. After deriving the learning law for this proposed network, convergence stability and adaptive learning rate were presented. The performance of the proposed network in model identification showed the accuracy of this network against the DRNN. At the end, this network was applied to real time control of an experimental image stabilization platform.

REFERENCE

[1] K. S. Narendra and K. Parthasarathy, "Identification and control of dynamic systems using neural networks," *IEEE Trans. on Neural networks*, vol. 1, no. 1, pp. 4-27, march 1990.

[2] Y. Zhang and J. Wang, "Obstacle avoidance for kinematically redundant manipulators using a dual neural network,"*IEEE Trans. Syst., Man and Cybernetics*, vol. 34, no. 1, pp. 752-759, 2004.

[3] S. Mukhopadhyay and K. S. Narendra, "Disturbance Regection in Nonlinear Systems Using Neural Networks," *IEEE Trans. Neural Nets.*, vol. 4, No. 1, pp. 63-72, 1993.

[4] D. Rumelhart, G.E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," *Parallel Distributet Processing*, D. Rumelhart and J. McClelland (Eds.), vol. 1, MIT Press, Cambridge, pp. 318-362, 1986.

[5] D. Luongvinh and Y. Kwon, "Behavioral modeling of power amplifiers using fully recurrent neural networks," *IEEE MTT-S International Microwave Symposium Diqest*, June 2005.

[6] G. Kechriotis and E. S. Manolakos, "Training fully recurrent neural networks with complex weights," *IEEE Trans. Circuits and Systems-II: Analog and Digital Signal Processing*, vol. 41, no. 3, pp. 235-238, March 1994.

[7] C. C. Ku and K. Y. Lee, "Nonlinear system identification using diagonal recurrent neural networks," *IEEE International Joint Conference on Neural Networks*, vol. 3, pp. 839-844, 1992.

[8] C. C. Ku and K. Y. Lee, "Diagonal recurrent neural networks for nonlinear system control," *IEEE International Joint Conference on Neural Networks*, vol. 1, pp. 315-320, 1992.

[9] C. C. Ku and K. Y. Lee, "Diagonal recurrent neural network-based control: convergence and stability," *American Contr. Conf.*, pp. 3340-3345, 1994.

[10] C. C. Ku and K. Y. Lee, "Diagonal recurrent neural networks for controller designs," *IEEE International Forum on Neural Networks to Power Systems*, pp. 87-92, 1993.

[11] C. C. Ku and K. Y. Lee, "Diagonal recurrent neural network based control using adaptive learning rates," *IEEE Conf. Decision and Contr.*, pp. 3485-3490, Tucson, Arizona, USA, 1992.

[12] C. C. Ku and K. Y. Lee, "Diagonal recurrent neural networks for dynamic systems control," *IEEE Trans. Neural Nets.*, vol. 6, no. 1, pp. 144-156, 1995.

[13] S. Liu and Y. Du, "Sonar array servo system based on diagonal recurrent neural network," *IEEE Intern. Conf. Mechatronics & Automat.*, pp. 1912-1917, Niagara Falls, Ontario, Canada, 2005.

[14] X. Wang and G. Peng, "Modeling and control for pneumatic manipulator based on dynamic neural network," *IEEE International Conference on Systems, Man and Cybernetics*, vol. 3, pp.2231-2236, 2003.

[15] X. Wang, G. Peng and Y. Xue, "Internal model controller with diagonal recurrent neural network for pneumatic robot servo system," *IEEE Intern. Symp. Comput. Intelligence in Robotics and Automat.*, pp. 1064-1069, Japan, 2003.

[16] T. G. Barbounis, J. B. Theocharis, M. C. Alexiadis and P. S. Dokopoulos, "Long-term wind speed and power forecasting using local recurrent neural network models," *IEEE Trans. Energy Conversion*, vol. 21, no. 1, pp. 273-284, 2006.

[17] F. Shaosheng and X. Hui, "Diagonal recurrent neural network based predictive control for active power filter," *Intern. Conf. Power Syst. Tech. – POWERCON*, pp. 759-762, Singapore, 2004.

[18] R. T. Bambang, R. R. Yacoub and K. Uchida, "Identification of secondary path in ANV using diagonal recurrent neural networks with EKF algorithm," *5th Asian Contr. Conf.*, pp. 665-673, Melbourne, Australia, 2004.

[19]C. C. Ku, K. Y. Lee and R. M. Edwards, "Improved nuclear reactor temperature control using diagonal recurrent neural nerworks," *IEEE Trans. Nuclear Science*, vol. 39, no. 6, pp. 2298-2308, 1992.

[20] B. Jayawardhana, L. Xie and S. Yuan, "Active control of sound based on diagonal recurrent neural network," *Conference of the Society of Instrument and Control Engineers (SICE)*, pp. 2666-2671, Osaka, 2002.

[21] J. S. Cho, Y. W. Kim and D. J. Park, "Identification of nonlinear dynamic systems using higher order diagonal recurrent neural network," *IEE Electron.Letters Online*, vol. 33, no. 25, pp. 2133-2135, 1997.

[22] P. A. Mastorocostas and J. B. Theocharis, "On stable learning of block-diagonal recurrent neural networks, part I: the RENNCOM algorithm," *IEEE International Joint Conference on Neural Networks*, vol. 2, pp. 815-820, 2004.

[23] P. A. Mastorocostas and J. B. Theocharis, "On stable learning of block-diagonal recurrent neural networks, part II: application to the analysis of lung sounds," *IEEE International Joint Conference on Neural Networks*, vol. 2, pp. 821-826, 2004.

[24] P. A. Mastorocostas and J. B. Theocharis, "A stable learning algorithm for block-diagonal recurrent neural networks: application to the analysis of lung sounds," *IEEE Trans. Syst., Man, Cybern. – Part B: Cybernetics*, vol. 36. no. 2, pp. 242-254, 2006.

[25] S. C. Sivakumar, W. Robertson and W. J. Philips, "On-line stabilization of block-diagonal recurrent neural networks," *IEEE Trans. Neural Nets.*, vol. 10, no. 1, pp. 167-175, 1999.

[26] P. A. Mastorocostas, D. Varsamis, C. Mastorocostas and I. Rekanos, "An accelerating learning algorithm for block-diagonal recurrent neural networks," *Intern. Conf. Comput. Intelligence for Modelling, Contr. and Automat.*, Vienna, AUSTRIA, 2005.

[27] P. Wang, Y. Li, S. Feng and W. Wei, "Convergence of diagonal recurrent neural networks' learning," *4th Congress on Intelligent Contr. and Automat.*, pp. 2365-2369, Shanghai, China, 2002.