Adaptive Neural Network-based Synchronization Control for Dual-drive Servo System

Suprapto[1]
[1]Graduate School of Engineering Science & Technology,
National Yunlin University of Science and Technology,
Doulio, Yunlin, Taiwan, R.O.C.
e-mail: d10210035@yuntech.edu.tw

Wei-Lung Mao[2]
[2]Graduate School of Engineering Science & Technology
and Department of Electrical Engineering,
National Yunlin University of Science and Technology,
Douliou, Yunlin, Taiwan, R.O.C.
e-mail: wlmao@yuntech.edu.tw

***Abstract*—In this paper, the neural network control is proposed for master-slave method of dual-drive servo system application. The architecture of control system includes traditional PID, back propagation neural network (BPNN) and radial basis function neural network (RBFNN). The BPNN can adjust three parameters of traditional PID automatically. The RBFNN approximation can determine the characteristics of servo system from given input and output sets. By combining PID, BPNN and RBFNN structure, the adaptive neural network-based method can achieve accurate control of nonlinear systems in synchronization for dual-drive servo. It is shown that the system performance of synchronization control including the speed output, the accuracy and the robustness works well with better dynamic and static characteristics.**

***Keywords* — Back Propagation Neural network (BPNN); Radial Basis function Neural Network (RBFNN); Dual-drive Servo; Synchronization Control.**

I. INTRODUCTION

Synchronization control of dual-drive servo system has growth rapidly and plays a very important role in many industrial fields. Based on the connection, synchronization for dual-drive servo system control has two methods. They are an independent and a dependent synchronization [1]. Independent is the connection without physical link, whereas dependent needs physical link or mechanical couplings, such as using gears, shafts, etc. In industrial application, selection of the synchronization servo control is one of a very important factor to achieve an effectiveness and efficiency of the production as well as the maintenance cost.

Due to the some advantages, many industries have changed using synchronization control method without physical link. In [2] the synchronous control realized without physical links will improve tracking performance with software algorithms. Using the method has various benefits such as good flexibility, easy for maintenance, low noise, and low vibration. In addition to those benefits, using software algorithm will have satisfied result.

One of the popular control methods in industrial fields is traditional PID. It has some advantages such as simple structure, good stability and convenient adjustment [3]. In addition the advantages, it also has the disadvantages such as nonlinear control, low precision and static error [4]. The solution to solve the weakness of traditional PID, it must be combined with other method. This paper introduces using Neural Network to adjust three parameters of traditional PID. Combining between its, it will has properties such as ability to uniform approximate arbitrary input-output linear or nonlinear mapping, stability and robust.

Intelligent neural network has properties like ability to uniform approximate arbitrary input-output linear or nonlinear mapping. It is also popular known for its strong capacities of self-learning, self-adapting and self-organization, and it is outstanding for the control of nonlinear systems [5]. In this paper, the control design applied neural network to adjust three parameters of traditional PID control so it gets optimal synchronization control of dual-drive servo. The design of synchronization dual-drive servo control using neural network is comprised two algorithms, one is a back propagation neural network (BPNN) and the other is a radial basis function neural network (RBFNN). The function of BPNN is used for adjusting of three parameters of PID and RBFNN is used for identification between input and output of plant.

According to [6, 7], there are several synchronous control strategies in recent year, master-slave control, cross coupling control, bi-axial cross couple control, electronics line-shafting and relative coupling strategy. Master slave synchronous control method is very simple to be realized, so we apply a algorithm of dual-drive servo method using master slave control. Synchronous control based on BPNN and RBFNN applied on dual-drive servo method using master slave control has faster dynamic response and higher steady precision.

II. DUAL-DRIVE SERVO CONTROL

The synchronous control method comprises of several motors at least two servo dc motors. Connection more than one servo on synchronization control method needs a good strategy of control, which it can be achieved by closed loop control using encoder. In this paper, the design applies two servo dc motors connected in master-slave mode. According to [8], the main characteristic of master-slave mode is revolved speed output. The speed output of the master will become the reference value of the slave. The structure of the master-slave of dual servo is shown in Fig. 1

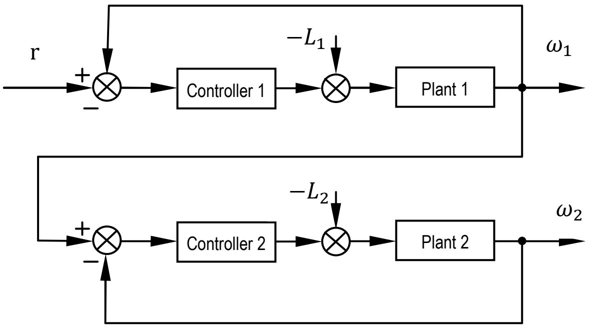

Fig. 1. Structure of Master-Slave of Dual Motor

In master-slave control mode, all instruction from master side transmitted to slave and then slave side will follow the instructions of master side. Of course, if the master side has any disturbance, the slave will reflect and follow it. On the contrary, the master will be never affected the disturbance from the slave.

III. PID CONTROLLER USING BP-NN METHOD

According to [9], there are so many techniques have been used for controller servo motor design. They are PID control, predictive control, optimal control, adaptive control, artificial intelligence control, feedback and feed forward technique. PID controller is widely used in industrial, because of the simplicity and robustness. The problem of traditional PID control can not adapt to time varying of characteristics in wide range.

Designing of combination with traditional PID and artificial intelligence control can improve control performance. Artificial intelligence neural network is choosen to adjust three parameters which the traditional PID has. This paper proposes the adaptive PID control based on BPNN. The equation for traditional PID can be written as eq.1:

$$\begin{aligned}u(k) &= u(k-1) + \Delta u(k)\\ &= u(k-1) + K_P[e(k) - e(k-1)] + K_I e(k)\\ &\quad + K_D[e(k) - 2e(k-1) + e(k-2)]\end{aligned} \quad (1)$$

where $e(k) = r(k) - y(k)$ is error, K_P is proportional gain, K_I is integral gain, K_D is differential gain and $u(k)$ is the controller output. Using the BPNN, three adjustable parameters can be adjusted in real time according to the condition of the system. The input of BPNN is decided by desire input, error, plant and the outputs are three adjustable parameters of PID controller.

In Fig. 2, suppose a stucture of BPNN has three layers neural network. They have input nodes, hidden nodes and output nodes. The input can be defined as $\mathbf{X} = [r(k), y(k), e(k), 1]$. The outputs of three parameters of PID controller are $K_{P,}$ K_I and K_D. The state variables can be written as:

$$x_1 = e(k) - e(k-1) \quad (2)$$
$$x_2 = e(k) \quad (3)$$
$$x_3 = e(k) - 2e(k-1) + e(k-2) \quad (4)$$
$$x_4 = u(k-1) \quad (5)$$

where x_1, x_2, x_3, x_3 are state variable of **X**, e(k) is error and u(k) is controller output. The relation between input and output of hidden layer as follows:

$$net_i^{(2)}(k) = \sum_{j=0}^{M} W_{ij}^{(2)}\, 0_j^{(1)} \quad (6)$$

$$0_i^{(2)}(k) = f\left(net_i^{(2)}(k)\right), (i = 1,2,..,8) \quad (7)$$

where $W_{ij}^{(2)}$ is the weight of hidden layer, $0_j^{(1)}$ is the output of input layer, $0_i^{(2)}(k)$ is the output of hiden layer and the superscripts of $(1), (2), (3)$ are three layers (input, hidden and output layer). The BPNN structure can be shown in Fig. 2.

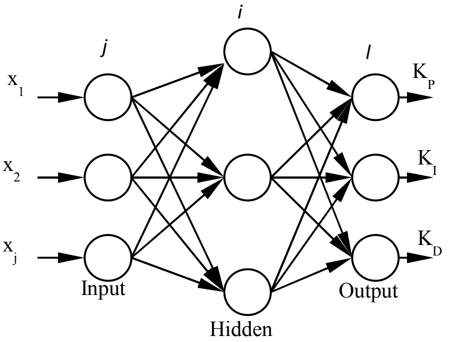

Fig. 2. Back Propagation Neural Network (BPNN)

The weight coefficient of the input layer and output layer is:

$$net_1^{(3)}(k) = \sum_{i=0}^{Q} W_{li}^{(3)} 0_i^{(2)}(k) \quad (8)$$

$$0_l^{(3)}(k) = g\left(net_l^{(3)}(k)\right), (l = 1,2,3) \quad (9)$$

$$0_1^{(3)}(k) = k_P \quad (10)$$
$$0_2^{(3)}(k) = k_I \quad (11)$$
$$0_3^{(3)}(k) = k_D \quad (12)$$

Where $0_k^{(3)}(k)$ is the output. The output K_P, K_I, and K_D are used auto-adjust traditional PID. The error can be shown as:

$$E_k = \frac{1}{2}\left(d_k - 0_k^{(3)}\right)^2 \quad (13)$$

where E_k is output error, d_k is desire input and $0_k^{(3)}$ is the output. When the different between desire input and output is bigger, the result is worse. The total error output can be known by this equation:

$$E = \frac{1}{2}\sum_{k-1}^{3}\left(d_k - 0_k^{(3)}\right)^2 \quad (14)$$

where d_k is desire input and $O_k^{(3)}$ is the output. By the gradient descent to adjust them, the equation can be calculated using a partial derivative respectiveIy as follow:

$$\Delta W_{li}^{(3)}(k) = -\eta \frac{\partial E(k)}{\partial w_{li}^{(3)}} + \alpha \Delta w_{li}^{(3)}(k-1) \quad (15)$$

where, η is the learning rate, α is the inertia coefficient,

$$\frac{\partial E(k)}{\partial w_{li}^{(3)}} = \frac{\partial E(k)}{\partial (k)} \cdot \frac{\partial y(k)}{\partial \Delta u(k)} \cdot \frac{\partial \Delta u(k)}{\partial O_l^{(3)}(k)} \cdot \frac{\partial O_l^{(3)}(k)}{\partial net_l^{(3)}} \cdot \frac{\partial net_l^{(3)}}{\partial w_{li}^{(3)}} \quad (16)$$

and while

$$\frac{\partial net_l^{(3)}}{\partial w_{li}^{(3)}} = O_i^{(2)}(k) \quad (17)$$

So

$$\frac{\partial \Delta u(k)}{\partial O_1^{(3)}(k)} = x_1 = e(k) - e(k-1) \quad (18)$$

$$\frac{\partial \Delta u(k)}{\partial O_2^{(3)}(k)} = x_2 = e(k) \quad (19)$$

$$\frac{\partial \Delta u(k)}{\partial O_3^{(3)}(k)} = x_3 = e(k) - 2e(k-1) + e(k-2) \quad (20)$$

In (16), $\frac{\partial y(k)}{\partial \Delta u(k)}$ is unknown, so it can be replaced by $\mathrm{sgn}\left(\frac{\partial y(k)}{\partial \Delta u(k)}\right)$, thus

$$\frac{\partial y(k)}{\partial \Delta u(k)} = \mathrm{sgn}\left(\frac{\partial y(k)}{\partial \Delta u(k)}\right) \quad (21)$$

Thus, the weights connection from hidden-layer to output-layer can be written as:

$$\Delta w_{li}^{(3)}(k) = \alpha \Delta w_{li}^{(3)}(k-1) + \eta \delta_i^{(3)} O_i^{(2)}(k) \quad (22)$$

$$\delta_i^{(3)} = e(k)\mathrm{sgn}\left(\frac{\partial y(k)}{\partial \Delta u(k)}\right) x \frac{\partial \Delta u(k)}{\partial O_l^{(3)}(k)} \; x \, g'\left(net_l^{(3)}(k)\right) \quad \text{for}(l = 1,2,3) \quad (23)$$

Likewise, the weights connection from input layer to hidden layer can be gained:

$$\Delta w_{li}^{(2)}(k) = \alpha \Delta w_{li}^{(2)}(k-1) + \eta \delta_i^{(2)} O_i^{(1)}(k) \quad (24)$$

$$\delta_i^{(2)} = f'\left(net_l^{(2)}(k)\right) \sum_{l=1}^{3} \delta_l^{(3)} W_{li}^{(3)}(k), \quad (i = 1, .., n-1) \quad (25)$$

Where

$$g'(\bullet) = g(x)[1 - g(x)], \;\; f'(\bullet) = \frac{[1 - f^2(x)]}{2} \quad (26)$$

IV. DYNAMIC RBFNN IDENTIFICATION NETWORK

According to [10], RBFNN was presented by J. Moody and C. Darken at the end of 1980s. The Radial functions are a special class of functions. The characteristic feature of RBFNN is decrease, increase, or monotonically with distance from a center point [11, 12]. RBFNN has three inputs and three hidden layers. The output is the weight sum of hidden nodes outputs. In Fig. 3, $\mathbf{X} = [x_1, x_2, \ldots, x_k]^T$ is NN's input.

In RBFNN, $y(k)$ as input $\mathbf{H} = [h_1, h_2, \ldots, h_k]^T$is the Radial Basis vector of RBF Network and h_j is Gaussian function.

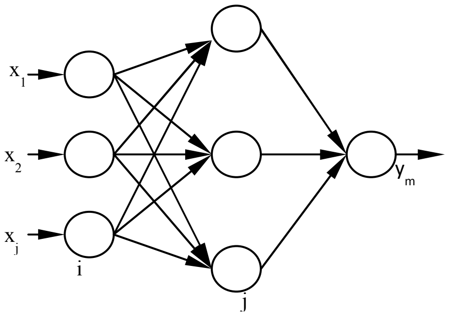

Fig. 3. RBF Neural Network

The hidden layer's output of RBFNN can be written as:

$$h_j = \exp\left(-\frac{\|y(k) - c_j\|}{2b_j^2}\right) \quad (27)$$

Where $i = 1;\; j = 1, \ldots, m;\; b_j > 0;\; \mathbf{C}_j = [c_1, \ldots, c_m];$ and $\mathbf{B} = [b_1, b_2, \ldots, b_m]^T$. The weight vector of RBF NN is expressed as:

$$\mathbf{W} = [w_1, w_2, \ldots, w_m]^T \quad (28)$$

The Output of identifying can be written

$$\hat{y}_n(k) = \sum_{n=1}^{m} w_n h_n \quad (29)$$

$$y_n(k) = [w_1 h_1 + w_2 h_2 + \ldots + w_n h_n]^T \quad (30)$$

Where n is the number of hidden layer. The criterion function of identifier can be written

$$J_1 = \frac{1}{2}[y(k) - \hat{y}(k)]^2 = \frac{1}{2} e_1^2(k)^2 \quad (31)$$

According to eq. (31), the iterative algorithms of w, b and c are shown as below:

$$w_j(k) = w_j(k-1) + \eta'(k)[y(k) - \hat{y}(k)]h_j + \alpha'[w_j(k-1) - w_j(k-2)] \quad (32)$$

$$b_j(k) = b_j(k-1) + \eta'(k)[y(k) - \hat{y}(k)]w_j h_j \frac{\|X - C_j\|^2}{b_j^3} + \alpha'[w_j(k-1) - w_j(k-2)] \quad (33)$$

$$c_{ij}(k) = c_{ij}(k-1) + \eta'(k)[y(k) - \hat{y}(k)]w_j h_j \frac{(x_i - c_{ji})}{b_j^2} + \alpha'[c_{ji}(k-1) - c_{ji}(k-2)] \qquad (34)$$

where $i = 1,2,3, j = 1,2,3, \eta'(k)$ is the learning rate, and α' is the momentum factor.The gradient parameter $\partial\hat{y}(k)/\partial\Delta u(k)$ as:

$$\frac{\partial y(k)}{\partial \Delta u(k)} \approx \frac{\partial \hat{y}(k)}{\partial \Delta u(k)} = \sum_{j=1}^{3} w_j h_j \frac{c_{j1} - \Delta u(k)}{b_j^2} \qquad (35)$$

V. SIMULATION RESULTS

The design of synchonization of dual-drive servo system based on adaptive NN can be depicted as Fig. 1. In Fig. 4., the one side blok diagram of synchornization dual-drive servo control comprises a controller using BPNN and an identification using RBFNN. The controller is used to adjust three parameters of PID. The indentification network is used to determine the characteristics of dual-drive servo control using a transfer function as controlled object.

The complete blok diagram of synchronization dual-drive servo devided into master and slave side. The master side is given a input signal. The output of master side become the input of slave side. From the properties of the master-slave synchronization dual-drive servo control, the output of slave will be decided by master side, on contrary the master side is not decided by the slave side.

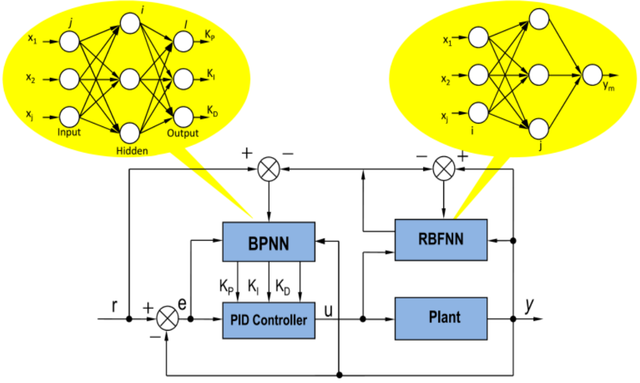

Fig. 4. Auto-Tuning PID Controller diagram

To know the ability of design, we assume two servos as controlled object for simulation. The transfer function of the controlled object is expressed as follows:

$$G(s)_1 = \frac{1.0025129}{1.28295s + 1}, G(s)_2 = \frac{1.002362}{1.28295s + 1} \qquad (36)$$

The simulation of design, a step math function signal is given on the input of master, a random value is used for initialized the weights of neural network, the learning rate η is taken 0.1, the inertia coefficient α is given = 0.05 and time sample is 0.02s.

By the given value, the output of three parameters adjusted traditional PID of master will be depicted in Fig. 5. In addition to output of K_P, K_I and K_D, the output signal of auto-tuning PID controller in Fig. 6. The outputs of them is given after the system physical paramaters are selected.

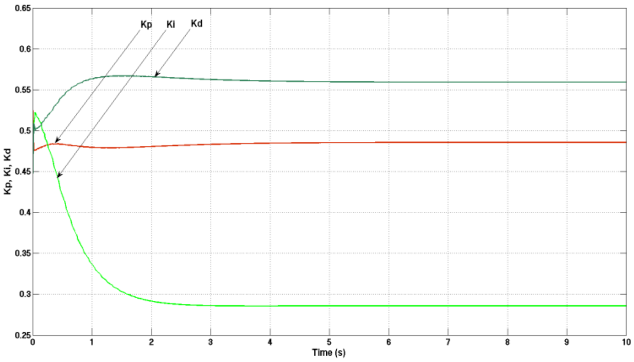

Fig. 5. Output of three parameters adjusted traditional PID

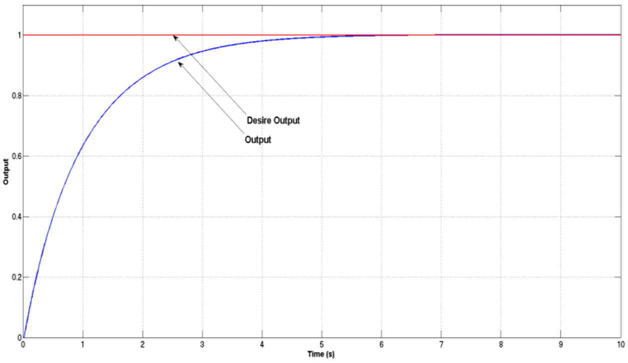

Fig. 6. Output of auto-tuning PID controller on master side

In Fig. 7, the output of synchronization of dual-drive servo in master-slave method is shown. The output has three lines, one is green color, the others are red and and blue color. Green line is the desire output, red line is output signal of master side and blue line is output signal of slave side. Both red line and blue line are different value at starting point, but after some seconds, they will same at the track.

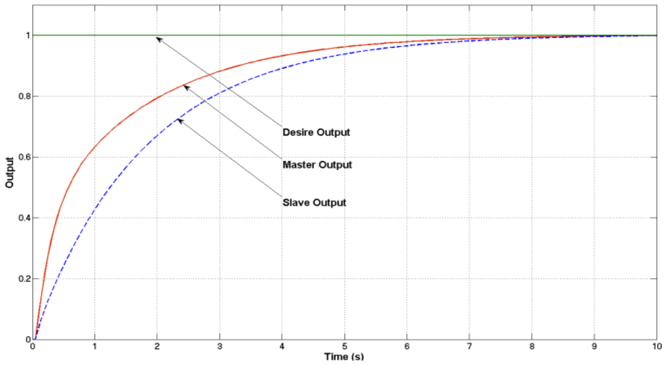

Fig. 7. Output of synchronization dual motor control.

From the output as Fig. 7. the quality of control can be calculated using Integral Absolute Error (IAE) which is mathematically given by:

$$I_{IAE} = \int_{0}^{\infty} |e(t)|dt \qquad (37)$$

where t is the time and e(t) is the error. The Error e(t) is calculated as the difference between the desire output and the output either master side or slave side. The average of calculation of IAE on master side is 1.1312 and slave side is 1.6188, whereas the average of IAE of master output and slave output have error 0.4878. The result of IAE calculation can be shown in Fig. 8.

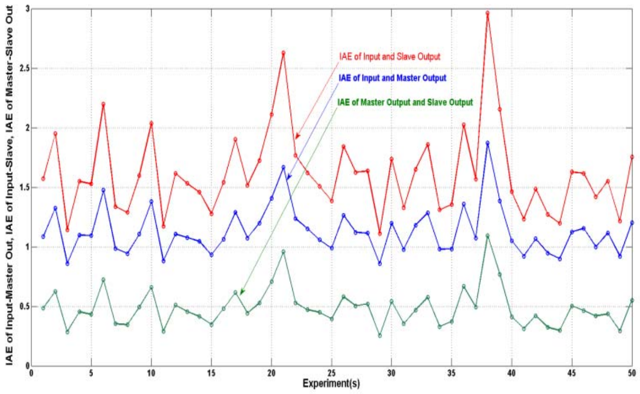

Fig. 8. The calculation of IAEs

From the calculation of IAEs, errors of the master and slave are very small. It indicates that the design has good caracteritics, robust and stable in speed and acceleration. Comparing with traditional PID, adaptive neural network using both BPNN and RBFNN adjusted PID is better because it has caracteristic to uniform approximate and identify arbitrary input-output. The output of synchronization control at master side will guide at slave side, otherwise the slave side will not influence at master side. The slave folllows of master side, either the signal control or disturbance.

VI. CONCLUSIONS

The experiment result proved using MATLAB Simulation tool, adaptive neural network could show the signal output of synchronization control for dual-drive servo. The control comprises BPNN to adjust three parameters of traditional PID and RBFNN to identify base of signal input and output given. Simulation result indicates that the system performance of synchronization for dual-drive control including the speed output, the accuracy and the robustness works well with better dynamic and static characteristics.

VII. ACKNOWLEDGEMENT

This project supported by Signal and Control Laboratory, National Yunlin University of Science and Technology, Taiwan, ROC. The first author would like to thanks for scholarship supported by the Indonesian Directorate General of Higher Education (DIKTI) and Yogyakarta State University, Indonesia.

VIII. REFERENCES

[1] S. L. Che Yambo, K.W. Eric Cheng, "Variable Gain Intelligent Control of Multi-Motor Syncronization System," *2^{nd} International Conference on Power Electronics Systems and Applications,* pp. 68-72, 2006.

[2] S.-S. Y. Seok-Kwon Jeong, "Precise position synchronous control of multi-axis servo system," *Mechatronics,* vol. 18, pp. 129 - 140, 2008.

[3] M. I. H. O. Nour, J.; Chan, K.Y., "Fuzzy logic control vs conventional PID control of an inverted pendulum robot 2007," *International Conference on Intelligent and Advanced Systems, ICIAS,* pp. 209-214 2007.

[4] K. S. K. F. M. G. Tang, Chen; Kwong, S, "An optimal fuzzy PID controller," *controllerIEEE Transactions on Industrial Electronics,* vol. 48, pp. 757-765, 2001.

[5] W. Wei, "Neural Network Computing," *Higher Education Press,* 2003.

[6] C. N. Francisco J. Perez-Pinal, Ricardo Alvarez, Ilse Cervantes, "Comparison of Multi-motor Synchronization Tehniques," *The 30th Annual Conference of the IEEE Industrial Electronics Society* p. 3, November 2-6 2004.

[7] H. Z. Biao YU, Chi Xue, "Research on Adaptive Fuzzy PID Synchronous Control Strategy of Double Motor," *I. J. Intelligent System and Applications,* vol. 5, p. 6, 2011.

[8] H. Z. Biao Yu, Chi Xue, "Research on Adaptive Fuzzy PID Synchronous Control Strategy of Double-Motor " *Intelligent Systems and Applications,* vol. 5, pp. 28-33, 2011.

[9] V. P. P. Vismay Chauhan, "Multi-motor Synchronization Techniques," *International Journal of Science, Engineering and Technology Research (IJSETR)* vol. 3, pp. 319 - 322, 2014.

[10] Y. C. S. Suni V. T. Elanayar, "Radial basis unction neural network for approximation and estimation of nonlinear stochastic dynamic systems," *IEEE Transaction on Neural Network,* vol. 5, pp. 584-603, April. 1994.

[11] F. P. Bianchini M, Gori M., "Learning without local minima in radial basis function networks," *IEEE Trans Neural Networks,* vol. 6, pp. 749-756, 1995.

[12] S. I. Park J, "Universal approximation using radial basis functions network," *Neural Comput,* vol. 3, pp. 246-257, 1991.