Position Control of Manipulator's Links Using Artificial Neural Network with Backpropagation Training Algorithm

Thiang, Handry Khoswanto, Tan Hendra Sutanto
Electrical Engineering Department, Petra Christian University
Siwalankerto 121-131 Surabaya, 60236 Indonesia
e-mail: thiang@petra.ac.id, handry@petra.ac.id

Abstract—**This paper describes about application of artificial neural network for controlling the position of manipulator's links. In this case, the manipulator has three degree of freedoms and the manipulator is implemented for drilling a printed circuit board. In this application, artificial neutral network was used as the control system method therefore it acts as the controller. The artificial neural network architecture used as the controller is a multilayer perceptron networks with backpropagation training algorithm. The artificial neural network has two inputs of control signal and one output of control signal and varies in number of hidden layer. The inputs of network are error signal and delta error signal. The output of network is speed of the DC motor. The experiments were done in variation of number of hidden layer, neuron per layer and learning rate. Experimental results show that architecture of the network that gives the best response is 1 hidden layer with 20 neurons per layer for the first link and 2 hidden layers with 40 neurons per layer for the second link.**

Keywords- artificial neural network; manipulator; control system; intelligent control; backpropagation

I. Introduction

Artificial neural networks (ANNs) are simplified models of the central nervous system. It is believed by many researches in the field that neural network models offer the most promising unified approach to build truly intelligent computer systems. ANNs have been shown to be effective as computational processors for various tasks including pattern recognition, associative recall, classification, data compression, modeling and forecasting, adaptive control and noise filtering[1].

Reference [2] describes one of applications of ANN in control system. ANN is applied for controlling one arm robot. In this application, ANN is used to model the second order controller so that ANN can act as the controller. Architecture of ANN used in this system is fully connected multilayer perceptron with backpropagation training algorithm. The result shows that ANN can control one arm robot successfully. Reference [3] describe about another application of ANN in control system. ANN is applied to control the speed of DC motor. In this application, ANN also acts as the controller and the result shows that ANN can control the speed of DC motor well. Felix Pasila and friends have successfully applied ANN combined with Fuzzy Logic for electrical load forecasting application [4][5][6].

This paper describes another application of ANN which is a further work of previous work [2][3]. In this case, ANN was applied to control the position of manipulator's links. The manipulator has three degree of freedom, which consists of two revolute joints and one prismatic joint. Fig. 1 shows the model of manipulator that was controlled by ANN. Because the manipulator was implemented for drilling process, there are only two links that are controlled by ANN. Therefore, there are two parallel ANNs used in this system in order to control the two links of the manipulator. The training data are created from the knowledge of the expert. ANNs learn the training data in order to model the training data automatically. The resulted model is in form of architecture of ANN including the weight and bias connection of the network.

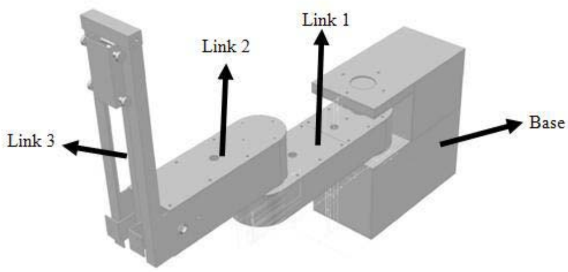

Figure 1. The manipulator with three degree of freedoms

II. Methods

A. Design of Artidficial Neural Network Architecture

Block diagram of the control system applied in this research is shown at Figure 2.

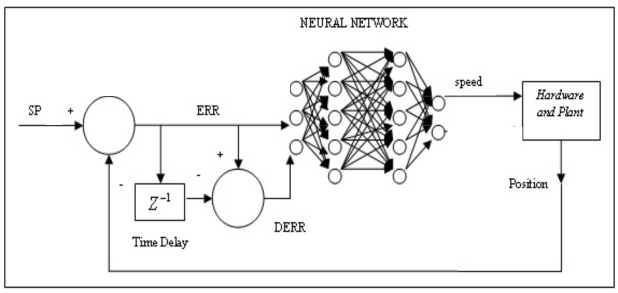

Figure 2. Block diagram of control system using artificial neural network

As shown at Fig. 2, ANN is applied as the control system method. ANN runs as the controller in order to control the plant. In this research, the plant is a manipulator with three degree of freedom. The designed ANN has two inputs and one output. The inputs are error signal (ERR) and delta error signal (DERR). ERR and DERR are determined by using the following equations:

$$ERR = SP - PV \tag{1}$$

$$DERR = ERR(n) - ERR(n-1) \tag{2}$$

where SP = Setting point

PV = Process variable

$ERR(n)$ = Current error

$ERR(n-1)$ = Previous error

The output is speed of the DC motor that is used as the actuator for moving the links of manipulator. The ANN was designed by using fully connected multilayer perceptrons architecture. Fig. 3 shows the architecture of ANN used as the controller in order to control the links of manipulator.

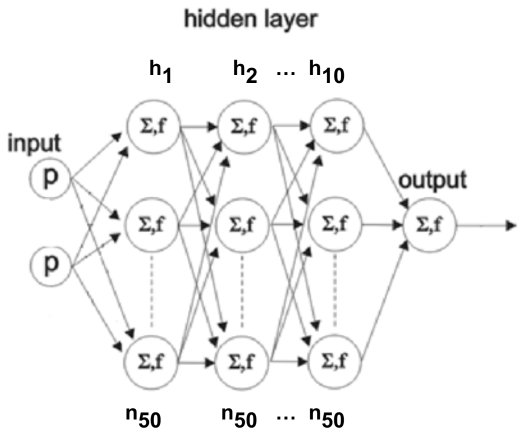

Figure 3. Architecture of ANN used as controller

Number of hidden layer varies from 1 to 10 layers. Number of neuron per hidden layer varies from 1 to 50 neurons. If the ANN has m layers and receives input of vector $\boldsymbol{p}$, then the output of the network can be calculated by using the following equation:

$$a^m = f^m(W^m f^{m-1}(W^{m-1} f^{m-2}(\cdots W^2 f^1(W^1 p + b^1) + b^2) + b^{m-1}) + b^m) \tag{3}$$

where $\boldsymbol{f}^m$ is log-sigmoid transfer function of the m[th] layer of the network that can be defined as following equation:

$$f(n) = \frac{1}{1+e^{-n}} \tag{4}$$

$\boldsymbol{W}^m$ is weight of the m[th] layer of the network, and $\boldsymbol{b}^m$ is bias of the m[th] layer of the network. Equation (3) is known as the feed forward calculation.

Because there are two links that are controlled by ANN, the system runs two ANNs system, one for each link. Both ANN have the same architecture.

Backpropagation algorithm is used as the training method of the designed artificial neural network. The backpropagation algorithm includes the following steps:

1. Initialize weights and biases to small random numbers.
2. Present a training data to neural network and calculate the output by propagating the input forward through the network using (3).
3. Propagate the sensitivities backward through the network:

$$\mathbf{s}^M = -2\dot{\mathbf{F}}^M(\mathbf{n}^M)(\mathbf{t}-\mathbf{a}) \tag{5}$$

$$\mathbf{s}^m = \dot{\mathbf{F}}^m(\mathbf{n}^m)(\mathbf{W}^{m+1})^T\mathbf{s}^{m+1}, \text{ for } m = M-1,\ldots,2,1 \tag{6}$$

where

$$\dot{\mathbf{F}}^m(\mathbf{n}^m) = \begin{bmatrix} \dot{f}^m(n_1^m) & 0 & \cdots & 0 \\ 0 & \dot{f}^m(n_2^m) & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \dot{f}^m(n_{s^m}^m) \end{bmatrix} \tag{7}$$

$$\dot{f}^m(n_j^m) = \frac{\partial f^m(n_j^m)}{\partial n_j^m} \tag{8}$$

4. Calculate weight and bias updates

$$\Delta\mathbf{W}^m(k) = -\alpha\,\mathbf{s}^m(\mathbf{a}^{m-1})^T \tag{9}$$

$$\Delta\mathbf{b}^m(k) = -\alpha\,\mathbf{s}^m \tag{10}$$

Where α is learning rate.

5. Update the weights and biases

$$\mathbf{W}^m(k+1) = \mathbf{W}^m(k) + \Delta\mathbf{W}^m(k) \tag{11}$$

$$\mathbf{b}^m(k+1) = \mathbf{b}^m(k) + \Delta\mathbf{b}^m(k) \tag{12}$$

6. Repeat step 2 – 5 until error is zero or less than a limit value.

B. *Implementation of Artificial Neural Network as Controller*

Implementation of ANN as the control system method consists of two step. First step is the designed ANNs were trained to create model of the controller so that satisfy training data. The training data are created from the knowledge of the expert. Training data are also created by considering the step response of general automatic control system, which is shown at Fig. 4.

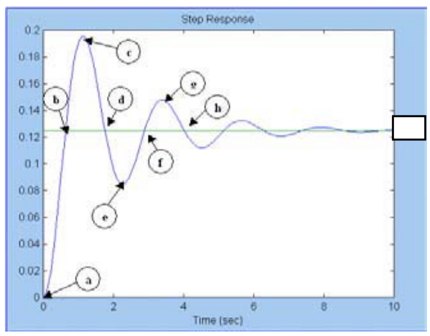

Figure 4. Step response of general automatic control system

For example, at the point a in Fig. 4, the input error and delta error can be determined. The value of error is positive and big value and the value of delta error is zero.

Thus, by considering the expert knowledge, output speed of DC motor can be determined. It was done same as creating the rule of fuzzy logic controller. Others training data were created with the same procedure using others points.

ANNs learn the training data automatically in order to model the controller equation. The resulted model is in form of architecture of ANN including the weight and bias connection of the network. There are 600 data used to train the ANN.

The second step is implementation of the trained ANNs for controlling the link position of manipulator. The ANNs system was implemented on a Personal Computer (PC), which is connected to the manipulator.

III. Experimental Results

For testing the performance of ANN to create model of controller equation and to control link position of manipulator, several experiments were done in variation of ANN architecture and training parameter, i.e. variation of hidden layer number, variation of neuron number per layer and variation of learning rate. The performance of artificial neural network is indicated by the MSE value.

A. *Experimental Results of Modeling the Controller*

This experiment was done in variation of number of hidden layer, number of neuron per layer, learning rate value. This experiment was done for testing how well ANN can model the controller equation. Number of hidden layer variation, used for this experiment are 1 hidden layer, 2 hidden layers, and 3 hidden layers. Number of neuron varies from 1 neuron, 20 neurons, and 40 neurons. Learning rate used for this experiment are 0.1, 0.5, and 0.9. The experiment was done for both links, which were controlled by ANNs.

Table I shows the summary of experimental result for the first link and Table II shows the experimental result summary for the second link.

TABLE I. Experimental Result Summary of MSE Value for First Link

Number of hidden Layer/Neurons per layer		Learning Rate		
		0.1	0.5	0.9
1 Hidden Layer	1 Neuron	0.214	0.119	0.095
	20 Neurons	0.055	0.050	0.056
	40 Neurons	0.052	0.054	0.056
2 Hidden Layers	1 Neuron	0.191	0.113	0.089
	20 Neurons	0.055	0.055	0.045
	40 Neurons	0.056	0.050	0.046
3 Hidden Layers	1 Neuron	0.225	0.132	0.101
	20 Neurons	0.055	0.054	0.048
	40 Neurons	0.056	0.049	0.049

TABLE II. Experimental Result Summary of MSE Value for Second Link

Number of hidden Layer/Neurons per layer		Learning Rate		
		0.1	0.5	0.9
1 Hidden Layer	1 Neuron	0.193	0.157	0.136
	20 Neurons	0.048	0.053	0.070
	40 Neurons	0.046	0.054	0.066
2 Hidden Layers	1 Neuron	0.195	0.179	0.153
	20 Neurons	0.050	0.047	0.047
	40 Neurons	0.049	0.046	0.047
3 Hidden Layers	1 Neuron	0.243	0.157	0.155
	20 Neurons	0.051	0.051	0.051
	40 Neurons	0.048	0.047	0.043

For the first link, the best architecture of ANN that can model the controller equation by learning the training data is ANN with two hidden layers and each layer has 20 neurons. It could achieve MSE value of 0.045. Increasing the number of hidden layer does not always give better value of MSE. In case of first link, the best result is achieved by ANN with 2 hidden layers although the difference of MSE value among 1, 2, and 3 hidden layer architectures are small especially at 20 and 40 neurons per layer. Table I also shows that increasing number of neurons per layer give a significant improvement of MSE value until it reaches a specific number. When the number is greater than that specific number, there is no significant improvement of MSE value.

For the second link, the best architecture of ANN that can model the controller equation by learning the training data is ANN with three hidden layers and each layer has 40 neurons. It could achieve MSE value of 0.043. Same as the first link, increasing the number of hidden layer does not give better MSE value. Moreover, it tends giving worse result. Table II also shows that increasing number of neurons per layer give a significant improvement of MSE value until it reaches a specific number. When the number is greater than that specific number, there is no significant improvement of MSE value.

B. *Experimental Result of Position Control Performance*

This experiment was done using the best architecture of ANN that was resulted from previous experiment. There are three architecture of ANNs used in this experiment. The first is the best architecture of ANN using 1 hidden layer, the second is the best architecture of ANN using 2 hidden layers and the third is the best architecture of ANN using 3 hidden layers. The purpose of this experiment is to test performance of the controller in order to control the link position of manipulator. This experiment was done in variation of setting point (SP) value. The parameters used to see performance of control system is rise time (t_r), settling time (t_s), and maximum overshoot (M_o).

TABLE III. EXPERIMENTAL RESULT SUMMARY OF CONTROL SYSTEM RESPONSE FOR FIRST LINK

ANN Architecture for First Link	Setting Point	t_r (second)	t_s (second)	Mo (%)
1 Hidden layer 20 Neuron	45°	3	4.5	4
	60°	3	3.5	3.3
	90°	3.5	3.5	-
	120°	3.5	3.5	-
2 Hidden layer 20 Neuron	45°	3	6	4.4
	60°	3	3.5	3.3
	90°	3	3	2.2
	120°	3.5	3.5	0.8
3 Hidden layer 40 Neuron	45°	4	4	4.4
	60°	3	4	6.6
	90°	3	3.5	2.2
	120°	3.5	3.5	0.8

TABLE IV. EXPERIMENTAL RESULT SUMMARY OF CONTROL SYSTEM RESPONSE FOR SECOND LINK

ANN Architecture For Second Link	Setting Point	t_r (second)	t_s (second)	Mo (%)
1 Hidden layer 40 Neuron	45°	3.5	5.5	8.8
	60°	4	4	3.3
	90°	4	4	6.6
	120°	5	5	-
2 Hidden layer 40 Neuron	45°	3.5	5.5	8.8
	60°	5	5	-
	90°	4	4	3.3
	120°	4.5	4.5	-
3 Hidden layer 40 Neuron	45°	3.5	7	6.6
	60°	4	5	-
	90°	3.5	4.5	2.2
	120°	4.5	5	-

Table III and IV show the experimental result summary of control system response for first link and second link respectively.

For the first link, control system using ANN has relatively the same response. But, the best response was resulted from ANN architecture with 1 hidden layer, 20 neurons per layer. This architecture resulted rise time value varies from 3 to 3.5 seconds, settling time varies from 3.5 to 4.5 seconds, and maximum overshoot was 4.4%.

From table IV, it is shown that the best response of controller for second link is resulted from ANN architecture with 2 hidden layers, 40 neurons per layer. This architecture resulted rise time value varies from 3.5 to 4.5 seconds, settling time varies from 4 to 5.5 seconds, and maximum overshoot was 4.4%.

Fig. 5 and 6 show other experiments that were also done to see the performance of the ANN as the control system method. In this experiment, the setting point was changed every an interval time in order to see whether the controller can control both links or not. Fig. 5 shows the control system response of ANN for controlling the first link. Fig. 6 shows the control system response of ANN for controlling the second link. From Fig. 4 and 5, it can be concluded that ANN can control both links position. The position of both link can move according to the changing of setting point value.

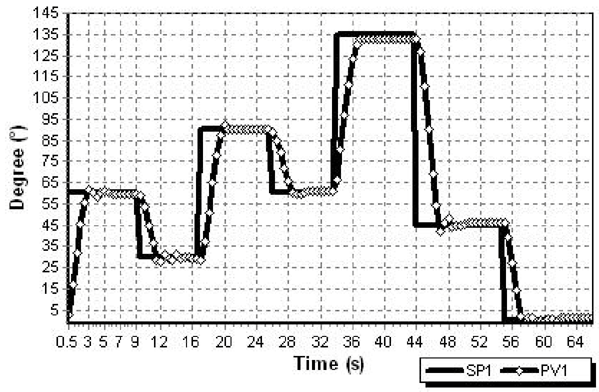

Figure 5. ANN controller response system for first link

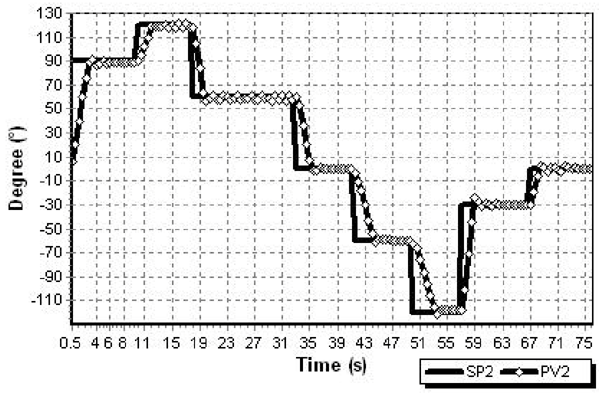

Figure 6. ANN controller response system for second link

IV. CONCLUSION

From experimental result, it can be concluded that the Artificial Neural Network (ANN) can be used as the control system method. In this case, ANN can model the controller equation well and ANN can control link position of the manipulator well. The best response control system was achieved by ANN architecture with 1 hidden layer, 20 neurons per layer for first link, and 2 hidden layers, 40 neurons for second link. In this case, the third link is not controlled by ANN because the third link only moves up and down for drilling process.

REFERENCES

[1] Dan W. Patterson. *Artificial Neural Networks, Theory and Applications*. Singapore: Prentice Hall, 1996.

[2] Thiang, Rianto Chandra, Iwan Njoto Sandjaja, "One Arm Robot Position Control Using Artificial Neural Network," in *Proceedings of National Seminar: The Application of Technology Toward Better Life*. Yogyakarta, 2005.

[3] Thiang, Indar Sugiarto, Hendrik Chandra, "DC Motor Speed Control System Using Artificial Neural Network," in *Proceedings of National Seminar of Computer Science and Information Technology,* Jakarta, 2004.

[4] Felix Pasila, "Multivariate Inputs for Electrical Load Forecasting on Hybrid Neuro-Fuzzy and Fuzzy C-Means Forecaster," in *Proceedings of International Conference on Fuzzy Systems*, Hongkong, 2008.

[5] Felix Pasila, Sautma Ronni, Thiang, Lie Hendra Wijaya, "Long-term Forecasting in Financial Stock Market using Accelerated LMA on Neuro-Fuzzy Structure and Additional Fuzzy C-Means Clustering for Optimizing the GMFs," in *Proceedings of International Joint Conference on Neural Networks*. Hongkong, 2008.

[6] Felix Pasila, "Neuro-Fuzzy Forecaster for Modeling and Forecasting Electrical Load Competition Using Multivariate Inputs on Takagi-Sugeno Networks," in *Proceedings of International Conference on Soft Computing, Intelligent System, and Information Technology*. Bali, 2007.