Eng Int Syst (2005) 2: 113–118
© 2005 CRL Publishing Ltd

Engineering
Intelligent Systems

Application of self-organized neuro-fuzzy identifier in intelligent predictive control of power plant

Hamid Ghezelayagh and Kwang Y Lee

Department of Electrical Engineering , The Pennsylvania State University, University Park, PA16802, USA

In an intelligent predictive controller, a neuro-fuzzy identifier predicts the response of the plant in future time interval, and provides a non-model based control approach. This identifier generates fuzzy rules and tunes membership functions automatically. A Genetic Algorithm (GA) method is implemented to encode and alter the fuzzy rules to minimize the identifier response error to the plant's logged data. Furthermore, the membership functions are tuned by error back-propagation method. These training methods dedicate self-organization to the neuro-fuzzy identifier. The identifier is used to model a power plant in predictive control approach. The experiments are conducted to achieve a reliable training data set.

Keywords: fuzzy neural network, power plant control, predictive control, identification, genetic algorithms, back-propagation

1. INTRODUCTION

A power plant system is a Multi-Input Multi-Output (MIMO) system with a wide range of nonlinear operation. This makes the linear model-based controllers not to be effective in all operating points. Model reference controller is an intelligent system [1] that has a complicated implementation for MIMO plants. A non-model based intelligent controller avoids limitations of plant's linear model. Among them, fuzzy control approach is one solution to overcome the practical limitations of controller design [2]. Predictive control has been applied in power plant and process control extensively [3–4]. Moreover, the training methods, such as Genetic Algorithm [5] and Evolutionary Programming (EP) [6], give the adaptation capability to the control system to optimize the system response.

The implementation of an intelligent predictive control system for a boiler/turbine unit is introduced in this paper. A self-organized neuro-fuzzy identifier performs as a model of plant to anticipate the output response in a prediction time interval. The GA and error back-propagation training methods perform fuzzy rule extraction and membership function tuning, respectively. This predictive control scheme uses EP in optimization of control inputs to minimize errors of the predicted outputs and reference set-points. This paper is organized to describe the structure of intelligent predictive controller in power plant application. Neuro-fuzzy identifier and its training are explained next. At last, the performance of the identifier with separate training data is reviewed.

2. INTELLIGENT PREDICTIVE CONTROL

A predictive controller anticipates the plant response for a sequence of control actions in future time interval which is known as *prediction horizon* [7]. The control action in this prediction horizon should be determined by an optimization method to minimize the difference between set-point and predicted response. Model-based Predictive Control (MPC) has been extended to a limited class of nonlinear systems [8]. An intelligent system may replace the empirical model of the plant in predictive control methodology. The structure of an intelligent predictive control system is shown in Figure 1. A non-model based identifier predicts response of the actual plant in prediction horizon. This identifier is a neuro-fuzzy system that is trained with an appropriate training data to estimate a nonlinear power plant. The optimization block finds the sequence of inputs to minimize a cost function for future time, but only the first value of this sequence is applied to the plant. The identifier does not depend on the mathematical model of the plant. Therefore, the optimization can not be implemented by conventional methods as in MPC. A search engine, based on Evolutionary Programming (EP) is used to determine the optimized control variables for a finite future time interval. The EP performs a competition search in a population and its mutated offspring. The members of each population are the input vector deviations that are initialized randomly. The mutation and competition continue making new generations to minimize value of a cost function. The output of the optimizer block is the control valve deviations that are integrated and applied to the identifier and power unit. The EP population consists of the individuals to present the deviation of control inputs.

The number of steps in discrete-time horizon for the power unit input estimation is defined by

$$n_u = N_u - N_1 \tag{1}$$

where N_1 is the start time of prediction horizon and N_u is the end time of the input prediction. The individuals of input deviation vector belong to a limited range of real numbers. In the beginning of EP algorithm, population is initialized with randomly chosen individuals. Each initial individual is selected with uniform distribution from the range of corresponding input.

The EP with adaptive mutation scale has shown a good performance in locating the global minima. Therefore, this method is used as it is formulated in [9]. The fitness value of each population is determined with a cost function to consider the error of predicted input and output in prediction time window. The cost function of the i^{th} individual in a population is defined by

$$f_{i,n} = \sum_{k=1}^{n_y} \|\underline{r}(t+k) - \hat{\underline{y}}_{i,n}(t+k)\|_R^2 + \sum_{k=1}^{n_u} \|\Delta\underline{U}_{i,n}(k)\|_Q^2 + \|\underline{r}(t) - \underline{y}(t)\|_R^2 \tag{2}$$

where $\underline{r}(t+k)$ is the desired reference set-points at sampled time of $t+k$, and $\hat{\underline{y}}_{i,n}(t+k)$ is the discrete predicted plant output vector which is determined by applying $\Delta\underline{U}_{i,n}(k)$ into the neuro-fuzzy identifier for time horizon of $n_y = N_2 - N_1$. The $\Delta\underline{U}_{i,n}(k)$ in (2) is the k^{th} row of the i^{th} individual in the n^{th} generation. The input deviation vectors is determined in a smaller time window of n_u as in (1) such that $n_u \le n_y$. The inputs of the identifier stay constant after $t + n_u$.

The fuzzy rules and membership functions of the identifier is trained off-line by the actual measured data of power unit. Then, the tuning process continues on-line to adapt the identifier with the plant's output while the control process performs. This training process improves the performance of the control system in prediction interval.

The power plant simulation is performed by a model that is developed by Bell and Åström [10]. This is a 3rd order nonlinear model, derived by physical and empirical methods, as in the following:

$$dP/dt = 0.0018u_2 P^{9/8} + 0.9u_1 - 0.15u_3 \tag{3}$$

$$dE/dt = ((0.73u_2 - 0.16)P^{9/8} - E)/10 \tag{4}$$

$$d\rho_f/dt = (141u_3 - w_s)/85 \tag{5}$$

$$w_s = (1.1u_2 - 0.19)P \tag{6}$$

$$L = 0.05(0.13073\rho_f + 60\alpha_{cs} + 0.11q_e - 67.975) \tag{7}$$

$$\alpha_{cs} = \frac{(1 - 0.001538\rho_f)\ (0.8p - 25.6)}{\rho_f(1.0394 - 0.0012304p)} \tag{8}$$

$$q_e = (0.85u_2 - 0.147)\,p + 45.6u_1 - 2.5u_3 - 2.1 \tag{9}$$

where p is drum steam pressure (Kg/cm^2), E is electrical power (MW), w_s is steam mass flow rate (Kg/s), L is water level deviation about mean (m), ρ_f is fluid density (Kg/m^3), u_1, u_2 and u_3 are normalized fuel, steam, and feedwater valve positions, as is steam quality (mass ratio), and q_e is evaporation rate (Kg/s). In addition, the actuator dynamics of control valves are constrained by

$$|du_1/dt| \le 0.007 \quad (s^{-1}) \quad 0.0 \le u_1 \le 1.0 \tag{10}$$

$$-2.0 \le du_2/dt \le 0.02 \quad (s^{-1}) \quad 0.0 \le u_2 \le 1.0 \tag{11}$$

$$|du_3/dt| \le 0.05 \quad (s^{-1}) \quad 0.0 \le u_3 \le 1.0 \tag{12}$$

to limit the rate of change in valve positions.

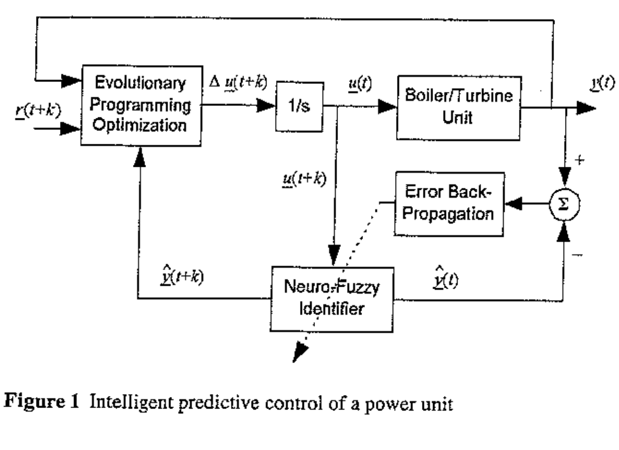

Figure 1 Intelligent predictive control of a power unit

3. SELF-ORGANIZED NEURO-FUZZY IDENTIFIER

Identification process needs to incorporate dynamics of the input/output interaction properly. This means that the past states of outputs provide information that may be used in conjunction with the inputs. Figure 2 depicts a structure to determine the present output from the past outputs and present inputs. Therefore, the plant is written in a nonlinear discrete form:

$$\hat{\underline{y}}(k+1) = \hat{f}(\underline{y}(k), \ldots, \underline{y}(k-i); \underline{u}(k), \ldots, \underline{u}(k-j)) \tag{13}$$

such that $\hat{y}(k)$ is the estimated output at time step k, and $\hat{f}(\cdot)$ is identifier function, $\underline{u}(k)$ and $\underline{y}(k)$ are plant input and output, respectively, at time step k. This identifier is known as series-parallel configuration [11]. In this structure, identifier uses the past and present plant inputs and outputs. Minimum number of inputs should be added to avoid implementation complexity and calculation time.

Neuro-fuzzy identifier technique is chosen for identification in predictive control loop [12]. The structure of a MIMO neuro-fuzzy system, with m inputs and n outputs, is shown in Figure 3. The i^{th} input and the j^{th} output retain p_i and q_j bell-shaped membership functions. Therefore, the number of possible rules, with *IF-THEN* format, is:

$$\eta = \prod_{i=1}^{m} p_i$$

The weights and biases in the second layer represent the widths and means of input membership functions, respectively. Using exponential activation function, the outputs

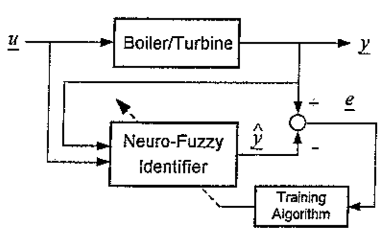

Figure 2 Series-parallel identification

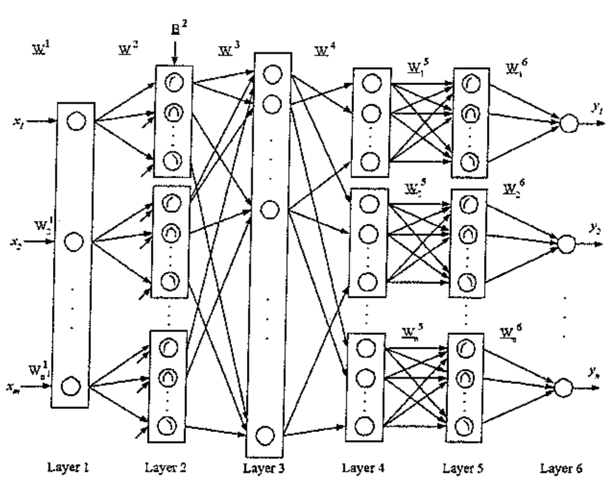

Figure 3 Structure of a neuro-fuzzy system

of the second layer neurons are the fuzzified system inputs. The third layer has η neurons. Weighting matrix of the third layer input represents antecedent parts of rules. The fourth layer consists of separate sections for every system output. Each section represents consequent parts of rules for an output. The sixth layer completes defuzzification, and provides crisp outputs.

Training algorithms enable identifier to configure fuzzy rules and adjust membership functions to model the plant with certain error penalty. Training determines fuzzy rules automatically based on available data from the plant operation. For this purpose Genetic Algorithm (GA) training is chosen, because of specific structure of the neuro-fuzzy identifier. Fuzzy membership functions are also adjusted by error back-propagation method.

In the start of training, the identifier is initialized with default input/output membership functions and fuzzy rules. Positions of '1's in weighting matrices of the third and fourth layers define fuzzy rules. These matrices are encoded in the form of GA chromosome that has a compound structure with n sections as the number of power unit outputs. The number of genes in each section is equal to the rule number η. Each section encodes the weighting matrices with an integer value of *allele*, θ_j^i, that belongs to the set of $\theta_j^i \in A = \{0, 1, \ldots, q_i\}$, for $j = 1, 2, \ldots, \eta$. Therefore, GA with non-binary alphabet size will be the training method [13]. Having a set of experimental input/output plant data points, GA can be applied to find optimal set of fuzzy rules. The fitness function, based on the least squares principle, provides evaluation of population individuals. To complete the GA iteration, it is necessary to prepare the next generation of population with applying three GA operators: selection, crossover and mutation. The weighted roulette wheel is used as selection operator that assigns a weighted slot to each individual. Crossover operator generates two offspring strings from each pair of parent strings. Crossover takes place in every sub-chromosome of parents. Crossover points are determined randomly with uniform distribution. Mutation operator changes value of a gene position with a frequency equal to mutation rate.

Tuning the parameters of fuzzy membership functions completes training of the neuro-fuzzy system. Adjusting the membership function increases the accuracy of the identifier. Error back-propagation is used for training self-organized neuro-fuzzy system. Let $D_k = \{\underline{\beta}(k), \underline{\gamma}(k)\}$ be a set of given pair of desired system input/output, such that $\underline{\beta}(k) \subset \Re^n$, $\underline{\gamma}(k) \subset \Re^m$. If $\hat{\underline{y}}(k)$ is the output of the neuro-fuzzy system in response to the input $\underline{\beta}(k)$, the squared error is defined as the following:

$$E = \frac{1}{2}[\underline{\gamma}(k) - \hat{\underline{y}}(k)]^t\,[\underline{\gamma}(k) - \hat{\underline{y}}(k)] \tag{14}$$

The training set contains the mean and width of the input-output membership functions. The sixth layer's weighting factor $w_{i,j}^6$ is updated by

$$w_{i,j}^6(k+1) = w_{i,j}^6(k) - \nu\,(\partial E/\partial w_{i,j}^6)|_k \tag{15}$$

where ν is the learning rate. The error rate is derived from

(14) as in the following:

$$\partial E/\partial w_{i,j}^{6} = -[y_i(k) - \hat{y}_i(k)](\partial \hat{y}_i/\partial w_{i,j}^{6})|_k \tag{16}$$

such that rate of the neuro-fuzzy output is derived by

$$\partial \hat{y}_i/\partial w_{i,j}^{6} = I_{i,j}^{6} / \sum_{l=1}^{n_i^5} I_{i,l}^{6} \tag{17}$$

In a way similar to the derivation of (15) to (17), the width and mean of membership functions in all other layers can be derived. The learning rates of error back-propagation should be chosen appropriately. Training ends after achieving specified error or reaching the maximum iteration number.

In power plant identification problem, the plant model is defined by equations (3) to (9). This model shows that input set of fuel, steam and feedwater control valve positions (u_1, u_2 and u_3) can be augmented with drum steam pressure (P) that has the most effect in the selected outputs. The other variable that is strongly dependent on the other variables is drum water level deviation (L). The specified boiler outputs, P and L, are used as extra inputs to identifier with one time step delay. If plant's model is not available, experiments and data gathering from the plant helps the experts to define the augmented input variables.

Neuro-fuzzy identifier is implemented in Object Oriented Programming (OOP) with C++ language. C++ provides a powerful tool called object. The neuron layers are encapsulated in class objects. Each layer has its own number of neurons to be configured. Objects of layers have dedicated parameters and functions (methods) that should be assigned or operate in run time. All these layer belongings are declared in classes of layer. Therefore, it is convenient to change or modify only class definitions without disturbing function type, activation function, training algorithm, etc.

4. SIMULATION AND TRAINING RESULTS

The identifier is trained and initialized before the control action in predictive control process starts. Training algorithm uses actual plant data to reduce the error that appears in identifier outputs. To simplify our evaluation, this training data is obtained from simulation of the power unit model, rather than the real plant. Experimental data should be complete enough to represent dynamics of the plant. This can be obtained by defining a set of necessary inputs to excite the plant such that plant response shows the nonlinearity of the model. Therefore, input conditions must be determined to incorporate process dynamics in the experiments. Input conditions for linear identifiers are investigated in several works that lead to theorems such as *Persistent Excitation Inputs.* In nonlinear systems, random input signals do not define any signal conditions. Hence, there is no guarantee that the inputs can excite the plant to show its dynamics.

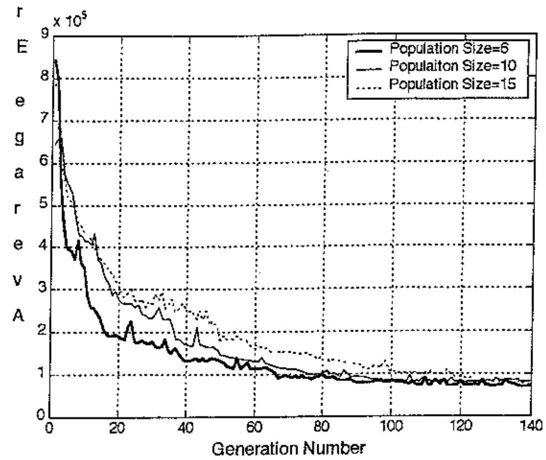

Figure 4 Convergence of GA training of identifier with staircase training data, input/output membership function number = 5

To gather training data, one should consider two goals of output transient and steady state. If database obtains a qualitative behavior of the plant during transient, trained identifier will have a good performance in transient responses. In addition, steady state accuracy in identifier will be achieved, if the training data includes the steady state operating conditions of the plant. For this purpose, a data log experiment is chosen based on series of step changes in plant's equilibrium operating points. The order of staircase step changes in this experiment is shown in Table 1. The staircase in valve positions is selected to cover a wide range of electrical power output.

Experiment starts with the lowest electrical power of 10 MW and steam pressure of 65 Kg/cm². Power and pressure increases in time with step changes in valve inputs. In a peak electrical power of 128.9 MW, the valve input changes to decrease power and pressure back to low values with another set of equilibrium points. Afterward, inputs are chosen to increase the power to the highest

Table 1 Staircase equilibrium points in data log experiment on power unit

Time (s)	E (MW)	P (Kg/cm²)	u_1 (pu)	u_2 (pu)	u_3 (pu)	w_s (Kg/s)	ρ_f (Kg/m³)	α_s (-)	L (m)
0	**10**	**65**	0.09	0.34	0.09	12.3	484.77	0.014	0
500	**20**	**70**	0.13	0.45	0.15	21.3	475.68	0.018	0
900	**40**	**80**	0.22	0.62	0.28	38.9	457.14	0.027	0
1300	**60**	**90**	0.30	0.74	0.40	56.1	437.89	0.037	0
1700	**80**	**100**	0.38	0.84	0.52	72.9	417.60	0.051	0
2100	**100**	**110**	0.47	0.91	0.63	89.4	395.80	0.068	0
2500	**120**	**120**	0.55	0.97	0.75	105.5	371.74	0.091	0
2900	**128.9**	**140.4**	0.60	0.90	0.79	111.9	344.92	0.136	0
3300	**105.8**	**129.6**	0.50	0.83	0.66	93.4	377.86	0.098	0
3700	**85.06**	**118.8**	0.42	0.76	0.54	76.6	403.68	0.073	0
4100	**66.65**	**108**	0.34	0.69	0.44	61.5	424.87	0.055	0
4500	**50.52**	**97.2**	0.27	0.62	0.34	47.9	442.63	0.041	0
4900	**36.65**	**86.4**	0.21	0.55	0.26	36.0	457.59	0.030	0
5300	**15.26**	**75.6**	0.12	0.38	0.12	17.3	477.65	0.020	0
5700	**10**	**140**	0.16	0.27	0.11	15.3	457.01	0.065	0
6100	**20**	**140**	0.20	0.32	0.17	23.4	448.42	0.069	0
6500	**40**	**140**	0.27	0.43	0.28	39.7	430.95	0.078	0
6900	**60**	**140**	0.34	0.54	0.40	55.9	412.99	0.088	0
7300	**80**	**140**	0.42	0.64	0.51	72.2	394.45	0.099	0
7700	**100**	**140**	0.49	0.75	0.63	88.4	375.15	0.112	0
8100	**120**	**140**	0.57	0.85	0.74	104.6	354.82	0.128	0
8500	**140**	**140**	0.64	0.96	0.86	120.9	333.07	0.146	0

value of 140 MW while pressure is kept constant in 140 Kg/cm^2. The input steps must be long enough to let the outputs to reach steady states. This experiment is only with a small subset of power nonlinearity data that helps us to have experiments in neuro-fuzzy identifier in a shorter training time. Figure 4 depicts average error of GA training for various population sizes. The crossover and mutation rates are $P_c = 0.7$, $P_m = 0.005$. The scaling weight vector is $W = [0.05, 0.01, 0.05]$. Inputs and outputs have five membership functions that lead to 3125 fuzzy rules. The convergence time rises as population size increases.

In a small range of operating point, a nonlinear plant performs linear behavior. This linearity of the power plant unit is tested in another data log experiment. The valve inputs are determined with sinusoidal perturbation around an operating point in medium power of 66.65 MW and steam pressure of 108 Kg/cm^2. Frequencies of these sinusoidal perturbations are swept in 3 steps for each individual valve position and any combination of two valves. Experiment longs 200 minutes and data is gathered every second.

After extraction of fuzzy rules, tuning phase starts to modify the membership function's mean and width. The range of the normalized input/output is limited to [−1, 1]. Training restricts the mean of membership function not to exceed this range. The widths are also limited to avoid dividing by zero error in weighting matrix of layer 2. Figure 5 shows an example of final input membership functions.

This training is repeated to include effect the new set of membership functions. Figures 6–8 illustrate transient responses of the identifier that is trained with sinusoidal data. The GA has population size of 10. Pressure, electrical power and level deviation are compared with part of the sinusoidal training data. It can be seen that the identifier follows the frequency and amplitude of the actual plant

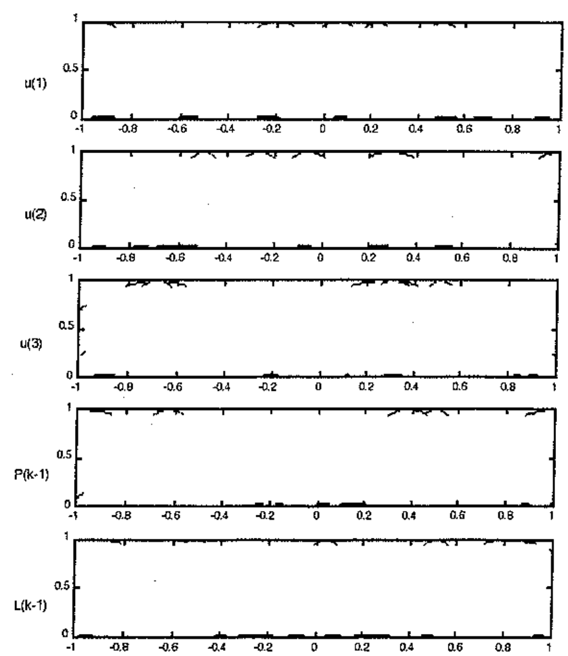

Figure 5 Input membership functions after training with sinusoidal data

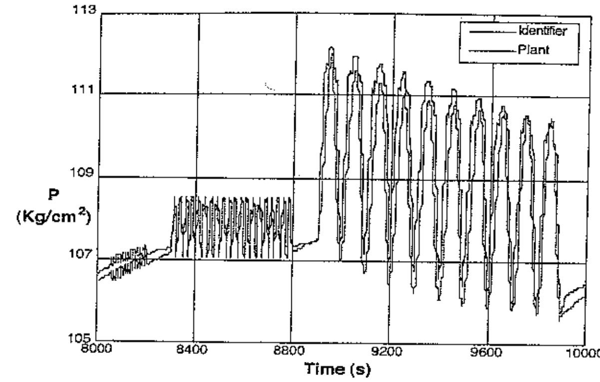

Figure 6 Drum pressure transient in sinusoidal test

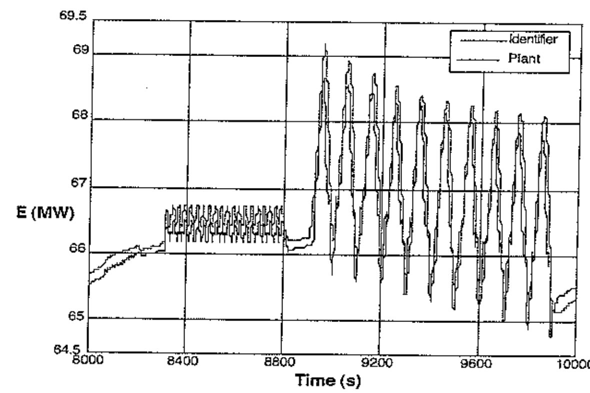

Figure 7 Electrical power transient in sinusoidal test.

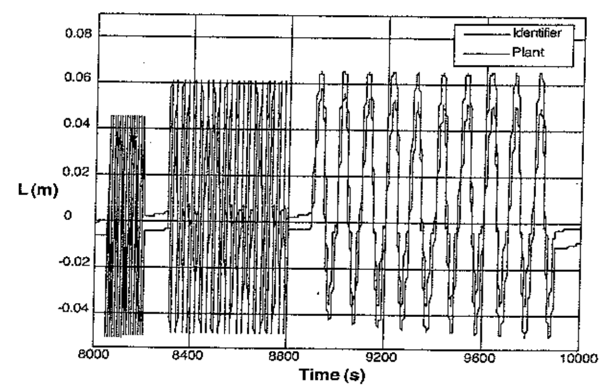

Figure 8 Drum level transient in sinusoidal test

data with good accuracy. However, the sinusoidal waveform is distorted in lower half of the sinus. Increasing training iteration improves this distortion. In addition, the outputs of identifier have offset relative to the actual plant data. The sinusoidal training data does not cover low frequencies. This makes the identifier not to be trained enough for the equilibrium points.

Transient response of the identifier that is trained by staircase data is shown in Figures 9–11. This test has a wide range from low power to high power. GA training is able to extract appropriate set of rules from random initial rules. Neuro-fuzzy identifier is able to track the plant out-

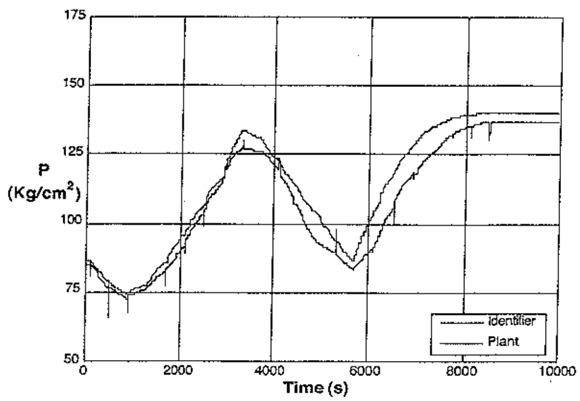

Figure 9 Drum pressure in staircase test

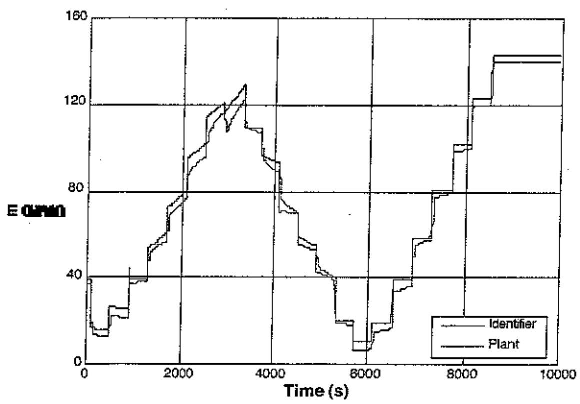

Figure 10 Electrical power transient in staircase test

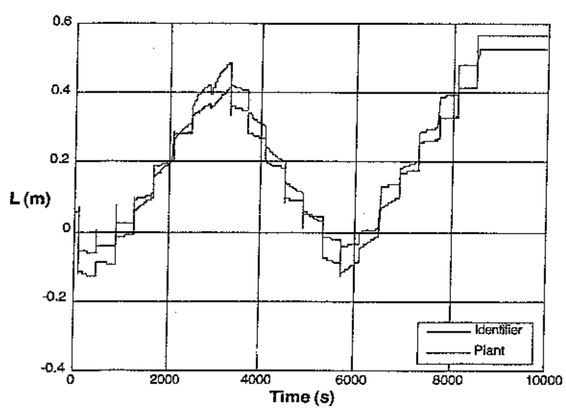

Figure 11 Drum level in staircase test

put changes. Output transient response shows small amount of offset and scaling. Each step of the training data does not last long enough to let the outputs to reach its steady state. Hence, the training does not have sufficient information of equilibrium points. The identification error is improved by more membership functions, longer training phases, and expanded training data. We should remember that the staircase test covers a wide range of power variation. Considering this, the fact that identifier is trained from completely random initial values confirms that both training phases successfully were able to obtain the power plant dynamics. The rule generation with this training method needs a long processing time, but we should note that writing large number of fuzzy rules manually is either impossible or takes more effort and time.

5. CONCLUSION

Implementation of an intelligent predictive control is introduced for a boiler/turbine power unit. This controller uses a neuro-fuzzy identifier to predict the plant outputs in response to a sequence of valve positions in future time window. The GA and error back-propagation training of the neuro-fuzzy identifier is also formulated for automatic rule generation and membership function tuning. The obtained identifier is trained for local and wide range with sinusoidal and staircase training data. The achieved self-organized neuro-fuzzy identifier is capable of modeling a MIMO system such as power plant, where the mathematical model is complicated to obtain in a wide range of operation.

REFERENCES

1 K.J. Åstrom and B. Wittenmark: *Adaptive Control.* Reading, MA, Addison-Wesley, 1989.
2 R. Garduno-Ramirez and K.Y. Lee, "Wide range operation of a power unit via feedforward fuzzy control," *IEEE Trans. Energy Conversion*, vol. 15, no. 4, pp. 421–426. 2000.
3 H. Tolle and E. Ersu: *Neurocontrol; Learning Control Systems Inspired by Neuronal Architectures and Human Problem Solving.* Lecture notes in control and information Sciences, Berlin, Springer-Verlag, 1992.
4 J. Saint-Donat, N. Bhat and T.J. McAvoy: Neural-Net based model predictive control. *Advances in Intelligent Control*, C. J. Harris (Ed.), London, Taylor and Francis, 1994.
5 D.E. Goldberg: *Genetic Algorithm in Search, Optimization, and Machine Learning.* Reading, MA, Addison-Wesley, 1989.
6 L.J. Fogel: The future of evolutionary programming. *Proceedings of the 24th Asilomar Conference on Signals, Systems, and Computers.* Pacific Grove, CA, 1991.
7 D. Clarke: *Advances in model-based prediction control.* Oxford University Press, New York, 1994.
8 J.B. Rawlings, E.S. Meadows, K.R. Muske, "Nonlinear model predictive control: A tutorial and survey," *Proceedings on ADCHEM '94*, Kyoto, Japan, 1994.
9 L.L. Lai: *Intelligent system application in power engineering; Evolutionary programming and neural networks.* John Wiley & Sons Inc., NY, USA, 1998.
10 K.J. Åstrom, K.J. and R.D. Bell: Dynamic models for boiler-turbine-alternator units: Data logs and parameter estimation for a 160 MW unit. In: *Report LUTFD2/(TFRT-3192)/1-137/(1987)*, Department of Automatic Control, Lung Institute of Technology, Lund, Sweden, 1987.
11 K.S. Narandra and K. Parthasarathy, "Identification and control of dynamical systems using neural networks," *IEEE Trans. Neural Networks*, vol. 1, no. 1, pp. 4–27, 1990.
12 H. Ghezelayagh and K.Y. Lee, "Neuro-Fuzzy Identifier of a Boiler System," *Engineering Intelligent Systems*, vol. 4, pp. 227–231, 1999.
13 A.J. Mason, "Partition coefficients, static deception and deceptive problems for non-binary alphabets," *Proceedings of the 4th International Conference on Genetic Algorithms*, pp. 210–214, 1991.