Development of Intelligent Controller with Virtual Sensing

Yul Y. Nazaruddin & Puji Astuti

Instrumentation and Control Research Group, Faculty of Industrial Technology,
Institut Teknologi Bandung,
Jl. Ganesa no. 10 Bandung, Telp./Fax. : +62 (22) 250 4424 / 250 6281
e-mail : yul@tf.itb.ac.id

Abstract. In many industrial plants, some key variables cannot always be measured on-line and for the purpose of control, an alternative of sensing system is required. This paper is concerned with a development of an alternative intelligent control strategy, which is an integration between the neuro-fuzzy based controller and virtual sensing system. This allows an immeasurable variable to be inferred and used for control. The virtual sensor is composed of the Diagonal Recurrent Neural Network (DRNN) for plant modeling and the Extended Kalman Filter (EKF) as the estimator with inputs from DRNN. The integration between virtual sensor and the controller enables a development of an on-line control scheme involving the immeasurable variable. The real-time implementation demonstrates the applicability and the performance of the proposed intelligent control scheme, especially in dealing with nonlinear processes.

Keywords: *Diagonal Recurrent Neural Network; extended Kalman Filter; neuro-fuzzy controller; virtual sensor; nonlinear process; real-time environment.*

1 Introduction

In many industrial control plants, some key variables are not always available for control purposes. These variables, in general, can not always be measured on-line, because it is difficult to measure, more expensive or due to the unavailability of the reliable sensors. Examples of such variable are viscosity, concentration, composition of substances, and flow index, which are measured quite often off-line, e.g. in laboratory environment. Another problem appears if the sensor performance declines, undetected disturbance occurs or even equipments degrades. These will cause the decreasing of the overall system performance. In such a case, a virtual sensing system could be used as an alternative strategy replacing the direct measurement method using sensors. Virtual sensing system or virtual sensor is an instrumentation system, which infers values of complex process variables by integrating information from easily made measurements.

Received March 18th, 2008, Revised March 15th, 2009, Accepted for publication March 20th, 2009.

In recent years many attempts have been made to combine neural network and fuzzy system methodologies in intelligent control system design. The primary concern is the integration of the strength of both methodologies in order to achieve learning and adaptation capability, and knowledge representation via fuzzy if-then rules, producing the so-called neuro-fuzzy systems. The main advantage of the neural network is that its learning and adaptation capabilities from numerical input-output data obtained from the measurement, and hence no mathematical model of the plant to be controlled is required. This advantage is combined with the ability of fuzzy system to describe a system with linguistics variable which is easier to be understood by human. This integration is very advantageous for nonlinear plants where its mathematical model is very difficult to derive and creates a powerful tool for identification process as well as for control system designs [1-6].

In this paper, an intelligent control strategy which is an integration between virtual sensing scheme and a neuro-fuzzy based control scheme is proposed. The virtual sensor consists of Diagonal Recurrent Neural Network (DRNN) for plant modeling and Extended Kalman Filter (EKF) as the estimator with inputs from DRNN. The integration between virtual sensor and a controller enables a development of an on-line control scheme involving the immeasurable variable. The selected controller is a neuro-fuzzy based controller, namely Adaptive Neuro-Fuzzy Inference Systems (ANFIS) controller with on-line learning [1,4,5,7]. ANFIS has the ability to deal with complex, nonlinear, and time varying systems with least numerical information. Thus, it is capable to handle a system which numerical model is difficult to be obtained.

The overall neuro-fuzzy controller algorithm with virtual sensing scheme has been implemented as a real-time control software developed using graphical-based programming language LabVIEW [8] and then it was tested in real-time environment to control the water level in tanks of a process mini-plant which is assumed to have a strongly inherent mechanical nonlinearities. The experiments will demonstrate the real application of the integration of the strength of neuro-fuzzy control scheme with virtual sensing in real-time environment.

2 Virtual Sensing System

The objective of using virtual sensing system in this investigation is to estimate the input variable which can not be measured on-line. In such a case, an artificial neural network technique can be applied to model the relation which is difficult to derive analyticaly. By deriving the inverse model of the process, the artificial neural network can be used to model the immeasurable variable, which is called the primary variable, from the easier to measure variable or measurable variable, which is called secondary variable. The scheme of virtual sensing system employs an artificial neural network with Diagonal Recurrent Neural Network (DRNN) structure [9,10] as model of the system and extended Kalman Filter (EKF) as estimator [11], as shown in Figure 1. The first step to be done is to identify the plant which its immeasurable variable will be estimated off-line. In this process, the immeasurable variable should be included in the process. After the model is obtained, the next step is to design an estimator based-on the Extended Kalman Filter [12,13] without inserting the immeasurable variable in the model input.

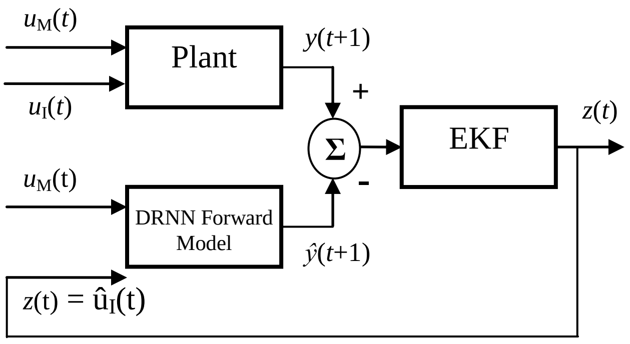

Figure 1 Virtual sensor with EKF as estimator

In developing the plant model using DRNN, during the learning phase, as input variables are the measurable input u_M (secondary variable) and immeasurable input u_I (primary variable), and as the output variable of the plant is y. The learning phase will be performed using the back-propagation method with adaptive learning rate. After the modeling process and validation, the weighting coefficients of DRNN will not be changed or in the on-line phase, the weighting should not be adaptive. The architecture of the DRNN can be seen in Figure 2 and its mathematical representation can be written as

$$S_j(n) = \sum_{k=1}^{N} W_{jk}{}^{10} I_k(n) + W_j{}^{Z} Z(n) + W_j{}^{11} X_j(n-1) + W_j{}^{1b} \tag{1}$$

$$X_j(n) = f(S_j(n)) \tag{2}$$

$$Y(n) = f\left(\sum_{j=0}^{M} W_j{}^{21} X_j(n) \right) \tag{3}$$

where at the n-th step, $I_k(n)$ is measurable input to kth neuron in input layer, $Z(n)$ is immeasurable input to neuron in input layer, $S_j(n)$ is input to jth neuron in hidden layer, $X_j(n)$ is jth output neuron in hidden layer, $Y(n)$ is network output and f is activation function, with $0 \leq k \leq N$, N is the number of neuron in input layer, and $0 \leq j \leq M$, M is the number of neuron in hidden layer.

From the above equations (1-3), the next step is to find the weighting of the network, conducted using the back-propagation algorithm. For this purposes, a cost function in the form of

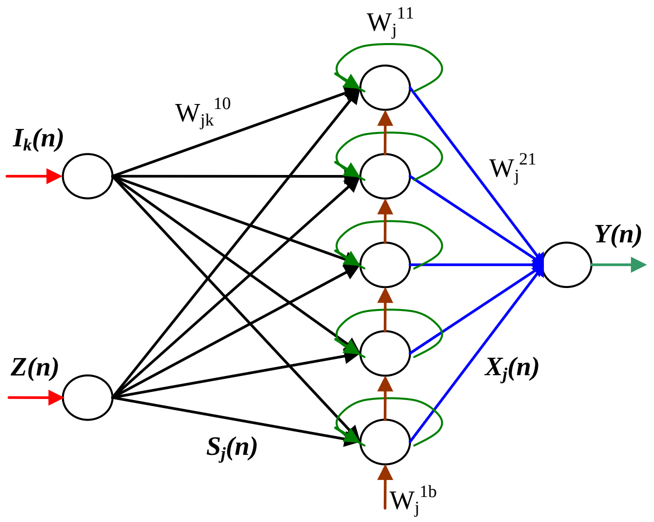

Figure 2 Structure of Diagonal Recurrent Neural Network (DRNN).

$$E = \frac{1}{2}\sum_{k=1}^{M}(y_k(n) - \hat{y}_k(n))^2 \tag{4}$$

is used, where y_k is the actual output , $\hat{y}_k$ is the output of the neural network and M is the number of output. The weight value of the neural network will be updated to minimize the error, and the correction of the weights will be performed by the partial derivation of E to the weights, or

$$W(n+1) = W(n) - \eta\frac{\partial E}{\partial W} \tag{5}$$

where η is the learning rate and $W(n)$ denotes the value of the weights at step n. If $\eta^{21}, \eta^{10}, \eta^{11}$, and η^{1b} are the learning rate of the weights of output layer, input layer, diagonal weights and bias, respectively, then equation (5) can be further elaborated as

$$W_j^{21}(n+1) = W_j^{21}(n) - \eta^{21}\frac{\partial E}{\partial W_j^{21}} \text{ : weight of output layer} \tag{6}$$

$$W_{jk}^{10}(n+1) = W_{jk}^{10}(n) - \eta^{10}\frac{\partial E}{\partial W_{jk}^{10}} \text{ : weight } I(n) \text{ of hidden layer} \tag{7}$$

$$W_j^{1Z}(n+1) = W_j^{1Z}(n) - \eta^{1Z}\frac{\partial E}{\partial W_j^{1Z}} \text{ : weight } Z(n) \text{ of hidden layer} \tag{8}$$

$$W_j^{11}(n+1) = W_j^{11}(n) - \eta^{11}\frac{\partial E}{\partial W_j^{11}} \text{ : weight of diagonal layer} \tag{9}$$

$$W_j^{1b}(n+1) = W_j^{1b}(n) - \eta^{1b}\frac{\partial E}{\partial W_j^{1b}} \text{ : weight of bias of hidden layer} \tag{10}$$

From equation (4), the following equation can be derived

$$\frac{\partial E(n)}{\partial W_{jk}^{ab}} = -e(n)\frac{\partial \hat{y}_k(n)}{\partial W_{jk}^{ab}} \tag{11}$$

where a denotes the end layer and b the beginning of layer, so that using equations (6 -10), the following expressions are obtained

$$\frac{\partial \hat{y}_k(n)}{\partial W_{jk}^{21}} = X_j(n) \tag{12}$$

$$\frac{\partial \hat{y}_k(n)}{\partial W_{jk}^{10}} = W_{jk}^{21}\frac{\partial X_j(n)}{\partial W_{jk}^{10}} = W_{jk}^{21}Q(n)$$

$$Q(n) = f'(S_j(n))\frac{\partial S_j(n)}{\partial W_{jk}^{10}} = f'(S_j(n))\left[I_k(n) + W_j^{11}Q(n-1)\right] \tag{13}$$

$$Q(0) = 0$$

$$\frac{\partial \hat{y}_k(n)}{\partial W_j^{1Z}} = W_{jk}^{\;21}\frac{\partial X_j(n)}{\partial W_j^{1Z}} = W_{jk}^{\;21}O(n)$$

$$O(n) = f'(S_j(n))\frac{\partial S_j(n)}{\partial W_j^{1Z}} = f'(S_j(n))\left[Z(n) + W_j^{1Z}O(n-1)\right] \tag{14}$$

$$O(0) = 0$$

$$\frac{\partial \hat{y}_k(n)}{\partial W_j^{11}} = W_j^{\;21}\frac{\partial X_j(n)}{\partial W_j^{11}} = W_j^{\;21}P(n)$$

$$P(n) = f'(S_j(n))\frac{\partial S_j(n)}{\partial W_j^{11}} = f'(S_j(n))\left[X_j(n-1) + W_j^{11}P(n-1)\right] \tag{15}$$

$$P(0) = 0$$

$$\frac{\partial \hat{y}_k(n)}{\partial W_j^{1b}} = W_{jk}^{\;21}\frac{\partial X_j(n)}{\partial W_j^{1b}} = W_j^{\;21}R(n)$$

$$R(n) = f'(S_j(n))\frac{\partial S_j(n)}{\partial W_j^{1b}} = f'(S_j(n))\left[1 + W_j^{1b}R(n-1)\right] \tag{16}$$

$$R(0) = 0$$

Furthermore, a parametric model of plant/process identifications is favorable to be used in industrial practice. Since most of the process models in industrial control show a strongly nonlinear behavior, a Nonlinear Auto-Regressive with eXogenous Variable (NARX) parametric model structure is widely used to represent nonlinear process. In this model, the output is a nonlinear function of previous outputs and inputs of the process, or in general it can be written as

$$y(t) = F\big(y(t-1),\cdots,y(t-n),u(t-d-1),\cdots,u(t-d-n)\big) + e(t) \tag{17}$$

where $y(t)$ and $u(t)$ are the sampled plant/process output and input at time instant t respectively, $e(t)$ is the equation error, n denotes the order of the process, d represents the process dead time as an integer number of samples and *F(.)* is an unknown nonlinear function to be identified. For the case in this investigation, the NARX model can be represented in the form as illustrated in Figure 3.

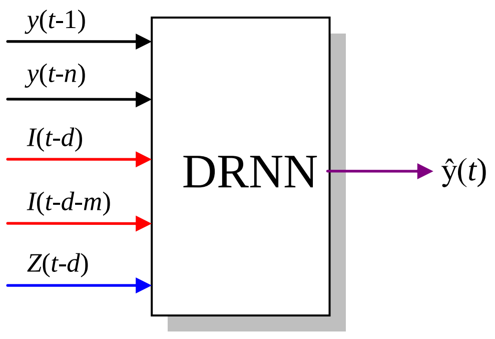

Figure 3 NARX model.

where $y(t)$ is the sampled plant/process output, $I(t)$ and $Z(t)$ is the measureable and immeasureable input, at time instant t, respectively, $\hat{y}(t)$ is output of DRNN, n the number of input to DRNN coming from system output, d is a delay time of input to DRNN coming from system input, and m is the number of input to DRNN coming from system input.

3 EKF as Estimator

Kalman Filter is well-known as an optimal estimator [12,13] which basically will give the best possible prediction of the state variables of an arbitrary system based on past observation. It will perform as an estimator of the state variables in a linear model. If the obtained model is nonlinear, then a linearization procedure should be carried out using e.g. Taylor Approximation [11]. The Kalman Filter so obtained is referred as extended Kalman Filter (EKF). In the scheme, the EKF will perform as an estimator of the state variables $(x_1(t), x_2(t))$ and the immeasurable input whereas the plant model is obtained based on the modeling using DRNN. Furthermore, the plant model of equations (1-3) can be written in time-domain form as

$$S_j(t) = \sum_{k=1}^{N} W_{jk}{}^{10} I_k(t) + W_j{}^Z Z(t) + W_j{}^{11} X_j(t-1) + W_j{}^{1b} \tag{18}$$

$$X_j(t) = f_1(S_j(t)) \tag{19}$$

$$Y(t) = f_2\left(\sum_{j=0}^{M} W_j{}^{21} X_j(t)\right) \tag{20}$$

where in every discrete-time t, $I_k(t)$ is measureable input to kth neuron in input layer, $Z(t)$ is immeasureable input to neuron in input layer, $S_j(t)$ is input to jth neuron in hidden layer, $X_j(t)$ is jth output neuron in hidden layer, $Y(t)$ is network output and f_1 and f_2 is activation function of hidden and output layer, respectively, with $0 \le k \le N$, N is the number of neuron in input layer, and $0 \le j \le M$, M is the number of neuron in hidden layer.

From the above 3 equations, f_1 and f_2 is a function of states (X_1, I, and Z), and the weights (W^{10}, W^{11}, W^{1Z}, W^{1b}, W^{21}). Assuming θ represents all parameters which can be changed during learning (such as weight and bias), then equations (18-20) can be rewritten in the following form

$$x_1(t+1) = f_1(x_1(t), Z(t+1), \theta_1(t), I(t+1)) + \xi_1(t) \tag{21}$$

$$x_2(t+1) = f_2(f_1(.), \theta_2(t)) + \xi_2(t) \tag{22}$$

$$Z(t+1) = Z(t) + \zeta(t) \tag{23}$$

$$y(t) = x_2(t) + v(t) \tag{24}$$

where the weight and bias in the hidden layer and output layer denoted as θ_1 and θ_2. Note that both vectors still have constant value during on-line process. $\{\xi_1(t)\}$and $\{\xi_2(t)\}$ is Gaussian white noise with zero mean and uncorrelated with $\{v(t)\}$, and positive definite variance, $\mathrm{var}[\zeta(t)] = S(t)$. Assuming $x(t) = [x_1(t) \quad x_2(t) \quad Z(t)]^T$ then the objective of learning is to determine $\hat{x}(t) = [\hat{x}_1(t) \quad \hat{x}_2(t) \quad \hat{Z}(t)]^T$. Further, to apply the Kalman Filter in the linear model, a linearization of equations (21-24) is necessary and for this purpose the following derivation are needed

$$\begin{aligned}
&F_{11} = \left.\frac{\partial f_1(x)}{\partial x_1}\right|_{x=\hat{x}(t)} \qquad F_{12} = \left.\frac{\partial f_1(x)}{\partial x_2}\right|_{x=\hat{x}(t)} \qquad F_{1Z} = \left.\frac{\partial f_1(x)}{\partial Z}\right|_{x=\hat{x}(t)} \\
&F_{21} = \left.\frac{\partial f_2(x)}{\partial x_1}\right|_{x=\hat{x}(t)} \qquad F_{22} = \left.\frac{\partial f_2(x)}{\partial x_2}\right|_{x=\hat{x}(t)} \qquad F_{2Z} = \left.\frac{\partial f_2(x)}{\partial Z}\right|_{x=\hat{x}(t)} \\
&F_{Z1} = \left.\frac{\partial Z(x)}{\partial x_1}\right|_{x=\hat{x}(t)} \qquad F_{Z2} = \left.\frac{\partial Z(x)}{\partial x_2}\right|_{x=\hat{x}(t)} \qquad F_{ZZ} = \left.\frac{\partial Z(x)}{\partial Z}\right|_{x=\hat{x}(t)}
\end{aligned} \tag{25}$$

so that the equations (21-24) become

$$x(t+1) = F(t)x(t) + G(t)\xi(t) \tag{26}$$

$$y(t) = H(t)x(t) + v(t) \tag{27}$$

where

$$F(t) = \begin{bmatrix} F_{11}(t) & F_{12}(t) & F_{1Z}(t) \\ F_{21}(t) & F_{22}(t) & F_{2Z}(t) \\ 0 & 0 & I \end{bmatrix} \quad ; \quad \xi(t) = \begin{bmatrix} \xi_1(t) \\ \xi_2(t) \\ \xi_3(t) \end{bmatrix} \quad ; \quad G(t) = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}$$

$$H(t) = \begin{bmatrix} 0 & I & 0 \end{bmatrix} \tag{28}$$

The initial values of the states and the covariance matrix is given by the relation

$$\begin{bmatrix} \hat{x}_1(0) \\ \hat{x}_2(0) \\ \hat{\theta}(0) \end{bmatrix} = \begin{bmatrix} E[x_1(0)] \\ E[x_2(0)] \\ \hat{\theta}(0) \end{bmatrix} \quad ; \quad P(0|0) = \begin{bmatrix} Var[x_1(0)] & 0 & 0 \\ 0 & Var[x_1(0)] & 0 \\ 0 & 0 & S(0) \end{bmatrix} \tag{29}$$

$$Q(t) = \begin{bmatrix} Q_1(t) & 0 & 0 \\ 0 & Q_2(t) & 0 \\ 0 & 0 & S(t) \end{bmatrix} \tag{30}$$

The EKF equations for this problem becomes

$$\begin{bmatrix} \hat{x}_1(t|t-1) \\ \hat{x}_2(t|t-1) \\ \hat{Z}(t|t-1) \end{bmatrix} = \begin{bmatrix} f_1(\hat{x}_1(t-1), \hat{Z}(t), \hat{\theta}_1(t), I(t)) \\ f_2(\hat{x}_1(t), \hat{\theta}_2(t)) \\ \hat{Z}(t-1) \end{bmatrix} \tag{31}$$

$$P(t|t-1) = \begin{bmatrix} F_{11}(t-1) & F_{12}(t-1) & F_{1Z}(t-1) \\ F_{21}(t-1) & F_{22}(t-1) & F_{2Z}(t-1) \\ 0 & 0 & I \end{bmatrix} P(t-1|t-1).$$

$$\begin{bmatrix} F_{11}(t-1) & F_{12}(t-1) & F_{1Z}(t-1) \\ F_{21}(t-1) & F_{22}(t-1) & F_{2Z}(t-1) \\ 0 & 0 & I \end{bmatrix}^T + \begin{bmatrix} Q_1(t-1) & 0 & 0 \\ 0 & Q2(t-1) & 0 \\ 0 & 0 & S(t-1) \end{bmatrix} \tag{32}$$

$$K(t) = P(t|t-1)H(t)^T \left[R(t) + H(t)P(t|t-1)H(t)^T \right] \tag{33}$$

$$P(t \mid t) = P(t \mid t-1) - K(t)H(t)P(t \mid t-1) \tag{34}$$

$$\begin{bmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \\ \hat{Z}(t) \end{bmatrix} = \begin{bmatrix} \hat{x}_1(t) \\ \hat{x}_2(t) \\ \hat{Z}(t) \end{bmatrix} + K(t)\left[y(t) - H(t)\hat{x}(t \mid t-1)\right] \tag{35}$$

for $t = 1, 2, \ldots.$

In equation (31), the calculation taking place is the calculation in DRNN, where $\hat{x}_2(t) = \hat{y}(t)$. The results obtained in equation (35), $\hat{x} = \left[\hat{x}_1(t) \quad \hat{x}_2(t) \quad \hat{Z}(t)\right]^T$, is used as input to equation (31). This will be continued until the value of $\hat{x} = \left[\hat{x}_1(t) \quad \hat{x}_2(t) \quad \hat{Z}(t)\right]^T$ converges to a certain value. At the time that the error, which is the different between output of system and the output of virtual sensor, less than the specified maximum error, the value of $\hat{Z}(t)$ is assumed to be similar with the immeasurable input values.

4 Neuro-fuzzy Based Control Scheme

As already mentioned above, an alternative technique which is successfully used for nonlinear plant control is based on neuro-fuzzy approach. An architecture called Adaptive Neuro-Fuzzy Inference System (ANFIS), which is an integration between neural network and fuzzy inference system has been implemented. Fuzzy inference system that is used in this paper is a first order model of Takagi-Sugeno-Kang (TSK) fuzzy system [1]. The rules of first order TSK fuzzy system with two inputs and one output can be described as

Rule i-th : If x is A_i and y is B_i then $f_i = p_i x + q_i y + r_i$ for $i = 1,2,...,m$

where m denotes the number of rules, A and B denotes the fuzzy sets and f denotes the crisp function, whereas p, q and r symbolize its consequent parameters. The architecture of neuro-fuzzy consists of five layers with different functions in every layer. Each layer is built by nodes that are connected to other nodes from different layer. The neuro-fuzzy structure which is equivalent with first order TSK fuzzy system with two rules, two inputs and one output is described in Figure 4.

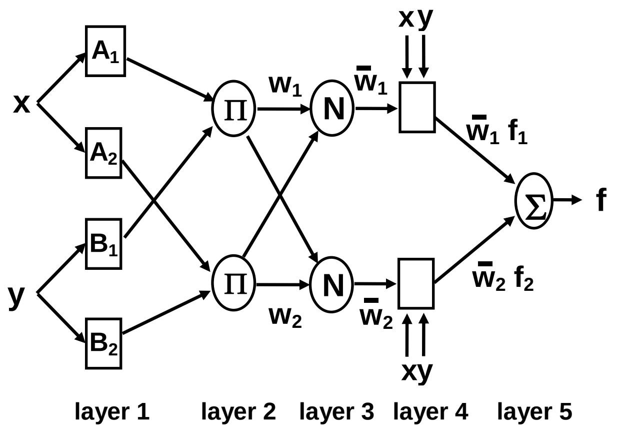

Figure 4 Adaptive Neuro-Fuzzy Inference System architecture

The network consists of two types of nodes, i.e. fixed and adaptive node. Adaptive node has a set of parameters that can be changed to minimize the error between neuro-fuzzy output and actual output. The description of each layer is described below whereas its function in every layer can be summarized as shown in Table 1:

Layer 1
Each node in this layer is adaptive node. *A* and *B* are linguistic labels (large, small, etc). Output of this layer is degree of membership which value depends on its membership function and premise parameters used.

Layer 2
Each node in this layer is fixed node and labeled Π which represents the "and" word. Output of this layer is firing strength of fuzzy sets.

Layer 3
Each node in this layer is fixed node and labeled N which mean normalization. Output of this layer is normalized firing strength.

Layer 4
Each node in this layer is adaptive node. This layer calculates the weighted function which its value depends on the consequent parameters used.

Layer 5
This layer has only one fixed node labeled Σ. Output of this layer is summation of layer 4 outputs.

Table 1 The description of each layer and its function.

Layer	Function
1	adaptive node with node function: $O_{1,i} = \mu_{A_i}(x)$ and $O_{1,i} = \mu_{B_i}(y)$
2	fixed node with firing strength of a rule: $O_{2,i} = w_i = \mu_{A_i}(x)\mu_{B_i}(y)$, i =1,2
3	fixed node with normalized firing strength: $O_{3,i} = \overline{w}_i = \dfrac{w_i}{w_1 + w_2}$, i =1,2
4	adaptive node with node function: $O_{4,i} = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i)$
5	fixed node which computes the summation of signals: $O_{5,i} = \overline{w}_1 f_1 + \overline{w}_2 f_2$

As can be seen, adaptive node is only represented in layer 1 and 4. In the 1st layer, the adaptive parameter is the parameter of membership function of input fuzzy set, which is a nonlinear function. The parameters in 4th layer are the linear function of system output. Parameters in 1st layer usually called premise parameter, while parameter in 4th layer even so called consequent parameter. In general, the structure has nonlinear premise parameter and linear consequent parameters. Nodes in 2nd layer show the number of rules that are used in neuro-fuzzy system. This layer represents the “and” word which is used in rules of TSK fuzzy system described above. In the 3rd layer, the firing strength is normalized. It is done by dividing each of firing strength by summation of all firing strengths. Moreover, the output of this network is compared to the actual output. The errors that occur are minimized by changing the premise and consequent parameter based on the learning rule. Since the consequent parameter has linear characteristic, the suitable learning method for this parameter is least-square estimator (LSE). Premise parameter has nonlinear characteristics, so the steepest descent method can be applied in the learning process. The hybrid learning rule is a combination of both least-square estimator and steepest descent learning. Using hybrid learning rule, the output of neuro-fuzzy network will converge faster and the possibillity to be trapped in local minima that usually happened in conventional back-propagation learning rule can be avoided.

4.1 Learning Process for Consequent Parameter

The output of 5th layer can be written as

$$
\begin{aligned}
f &= \overline{w}_1 f_1 + \overline{w}_2 f_2 \\
&= (\overline{w}_1 x) p_1 + (\overline{w}_1 y) q_1 + (\overline{w}_1) r_1 + (\overline{w}_2 x) p_2 + (\overline{w}_2 y) q_2 + (\overline{w}_2) r_2
\end{aligned} \tag{36}
$$

From the above equation, it can be seen that the consequent parameters are linear parameters with respect to the systems output. If *P* learning data is applied to the equation (36), it can be shown that it can be represented by $A\theta = y$, where θ is an unknown vector whose elements are the consequent parameters and *y* is the output vector whose elements are P learning data, where

$$
\begin{aligned}
\theta &= [p_1 \quad q_1 \quad r_1 \quad p_2 \quad q_2 \quad r_2]^T \\
y &= [y_1 \ldots y_n]^T
\end{aligned} \tag{37}
$$

and *A* is a matrix in the form of

$$
A = \begin{bmatrix}
(\overline{w}_1 x)_1 & (\overline{w}_1 y)_1 & (\overline{w}_1)_1 & (\overline{w}_2 x)_1 & (\overline{w}_2 y)_1 & (\overline{w}_2)_1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
(\overline{w}_1 x)_p & (\overline{w}_1 y)_p & (\overline{w}_1)_p & (\overline{w}_2 x)_p & (\overline{w}_2 y)_p & (\overline{w}_2)_p
\end{bmatrix} \tag{38}
$$

Using the least-square estimator, the best solution of this equation can be obtained using

$$
\begin{aligned}
\theta_{i+1} &= \theta_i + P_{i+1} a_{i+1} (y_{i+1}^T - a_{i+1}^T \theta_i) \\
P_{i+1} &= P_i - \frac{P_i a_{i+1} a_{i+1}^T P_i}{1 + a_{i+1}^T P_i a_{i+1}}
\end{aligned} \tag{39}
$$

with a_{i+1} is *i*+1th row vector of matrix *A*, y_{i+1} is the i+1th element of vector ***y***, and P_i is the covariance matrix. The initial conditions of $\theta_0 = 0$ and $P_0 = mI$, where *I* is identity matrix and *m* is a large number.

4.2 Learning Process for Premise Parameters

The premise parameter is an adaptive nonlinear parameter in the first layer. This parameter consists of parameters (*a,b,c*) which exist in the generalized bell function. The learning process employs the gradient descent method with back-propagation error. The ANFIS structure in Figure 4. consists of 5 layers with *p* number of data pair for learning process. The error measurement at the *p*-th training data ($1 \le p \le P$) can be written as

$$E_p = (d_p - x_5^p)^2 \tag{40}$$

with d_p is the desired *p*-th output data and x_5^p is the output of ANFIS (layer 5) for *p*-th data. The error signal $\varepsilon_{l,i}$ is the first derivative of the error measurement to the output of the i-th node at the 1st layer, or

$$\varepsilon_{l,i} = \frac{\partial E_p}{\partial x_{l,i}} \tag{41}$$

with $x_{l,i}$ is the output of 1st layer of i-th node. Then the error measurement for the output of ANFIS (layer 5) is

$$\varepsilon_5 = -2(d^p - x_5^p) \tag{42}$$

For the interior node, the error signal can be obtained using the chain rule

$$\varepsilon_{l,i} = \frac{\partial E_p}{\partial x_{l,i}} = \sum_{m=1}^{N(l+1)} \frac{\partial E_p}{\partial x_{l+1,m}} \frac{\partial f_{l+1,m}}{\partial x_{l,i}} = \sum_{m=1}^{N(l+1)} \varepsilon_{l,i} \frac{\partial f_{l+1,m}}{\partial x_{l,i}} \tag{43}$$

The notation *l,i* denotes the *l*-th layer of *i*-th node. *N*(*l*+1) represents the number of node at the *l*+1-th layer, whereas $f_{l+1,m}$ symbolizes a function at the *l*+1-th layer of *m*-th node. The error signal at the *l*-th layer can be stated as a linear combination of the error of *l*+1-th node. To calculate the error signal at the desired node and layer, the error signal is first determined at the node and 5-th layer using the equation (42), and then using equation (43), the error signal is propagated toward the desired node and layer. The 1st layer of ANFIS structure has the adaptive premise parameter. The measurement of error signal to the change of parameter is done using the following relation

$$\frac{\partial E_p}{\partial \alpha} = \frac{\partial E_p}{\partial x_{l,i}} \frac{\partial f_{l,i}}{\partial \alpha} = \varepsilon_{l,i} \frac{\partial f_{l,i}}{\partial \alpha} \tag{44}$$

with α is a set of premise parameter. The error signal $\varepsilon_{l,i}$ is calculated from the error signal of the next layer according to equation (42) and (43). $f_{l,i}$ is a function of the first layer which is the membership function of the *generalized bell* so that $\frac{\partial f_{l,i}}{\partial \alpha}$ is the derivative of the membership function of the generalized bell to the parameters *a*, *b*, and *c*.

The measurement of the error signal with respect to the change of parameter for the whole data is obtained using equation

$$\frac{\partial E}{\partial \alpha} = \sum_{p=1}^{P} \frac{\partial E_p}{\partial \alpha} \tag{45}$$

The improvement of the premise parameter is calculated iteratively using sinple gradient steepest descent method which can be formulated as follows

$$\alpha_{i+1} = \alpha_i + \Delta\alpha \; ; \; \Delta\alpha = -\eta \frac{\partial E}{\partial \alpha} \tag{46}$$

with the error signal E is the different between predicted output of neuro-fuzzy system and actual output, η is the learning rate where

$$\eta = \frac{\kappa}{\sqrt{\sum_{\alpha} \left(\frac{\partial E}{\partial \alpha} \right)^2}} \tag{47}$$

and κ is the *step-size* which can be change to speed-up the convergence.

The overall structure of the proposed intelligent control strategy using ANFIS as controller and virtual sensing scheme can be seen in Figure 5.

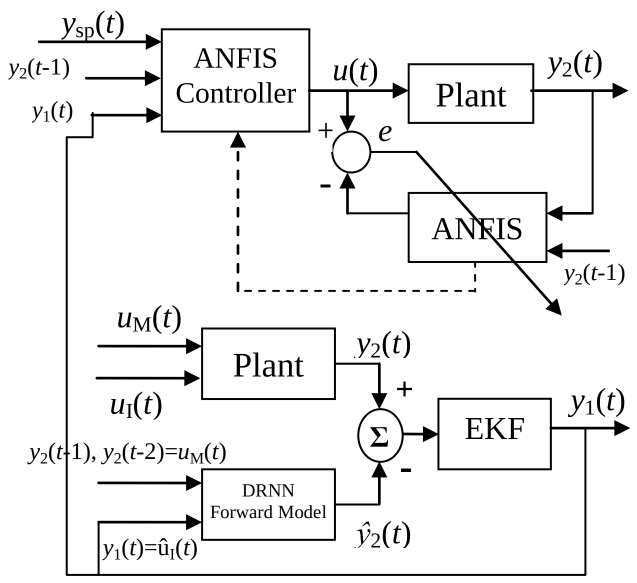

Figure 5 The overall control scheme

5 Experimental Results and Evaluation

The overall neuro-fuzzy controller algorithm with virtual sensing scheme has been implemented as a real-time control software developed using graphical-based programming language LabVIEW and runs on a personal computer. To see the capability and performance of the proposed intelligent control strategy, the scheme was tested in real-time environment to control the level of a process mini-plant which is considered to have strongly inherent mechanical nonlinearities due to its mechanical components.

5.1 Process Mini-plant Description

The process mini-plant basically consists of two tanks containing fluid which its level will be controlled, and real industrial-scaled components, such as differential pressure transmitter, control valve, I/P converter, so that it resembles almost real-plant characteristics. View of the process mini-plant is shown in Figure 6 and the control scheme configuration in Figure 7. The software was connected on-line to the process mini-plant through an AD/DA card and a signal conditioner circuit.

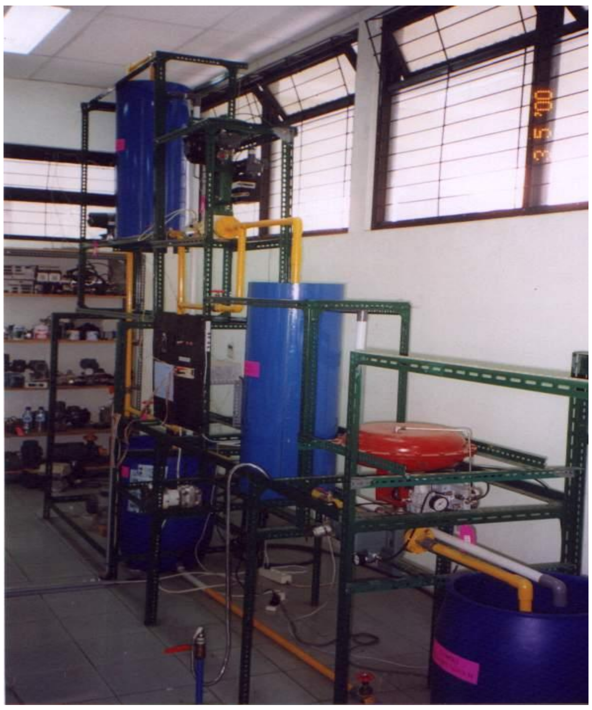

Figure 6 View of the process mini-plant.

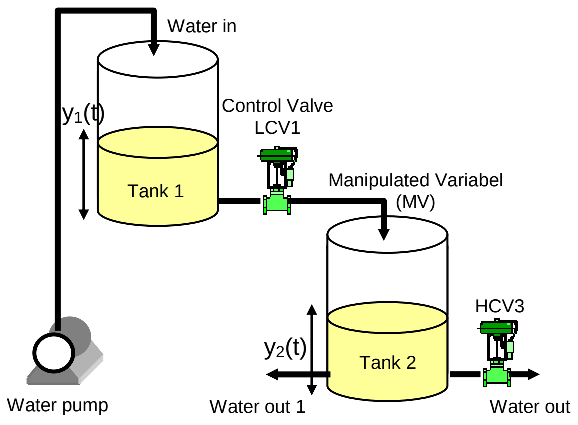

Figure 7 Control scheme configuration.

In this investigation, $y_1(t)$ is assumed as immeasurable variable, which can be measured off-line and affects to the controlled variable $y_2(t)$. Virtual sensor was used to predict y_1(t) based on variables which presumably affect it. Further, the output of virtual sensor is used as input to the ANFIS based controller with on-line learning, as can be seen in Figure 5. The control variable $u(t)$ (output of ANFIS) is the manipulated variable (MV), or the percentage opening of the valve LCV1.

5.2 Results of On-line Control

Figure 8 shows the results of on-line control of process mini-plant using the proposed intelligent control scheme with virtual sensing. It can be seen that satisfactory performance is obtained although the set-point was changed frequently. Acceptable result is also shown from the response of virtual sensing scheme, as demonstrated in Figure 9.

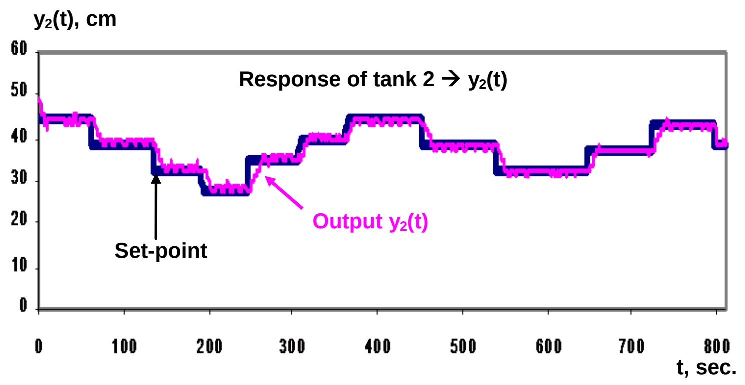

Figure 8 Response of tank 2 using the proposed intelligent control scheme with virtual sensing scheme

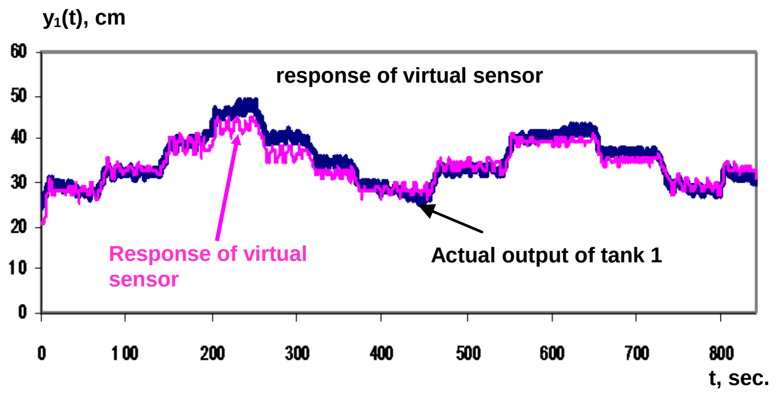

Figure 9 Results of virtual sensing scheme.

The immeasurable variable, y_1(t), could be predicted sufficiently well by the virtual sensing scheme, where it is then used for input to the ANFIS controller. Slightly decreased performance occurs in sampling instant about 200 sec. where in this time the level of tank 1, as observed during implementation, reached its highest level, i.e. 50 cm, during learning period.

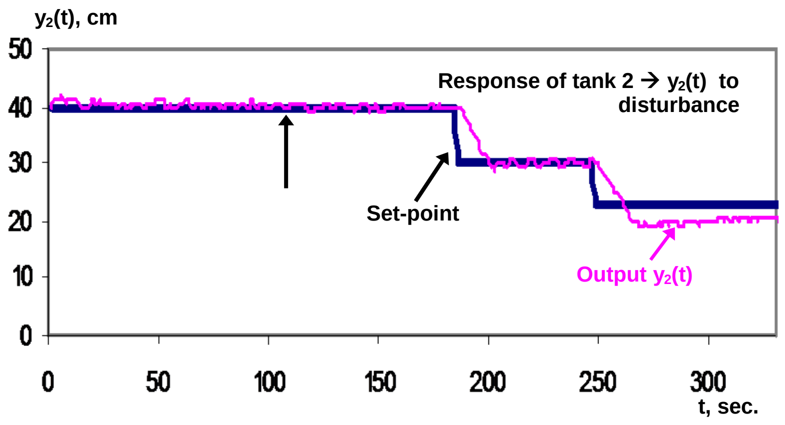

Figure 10 Response of tank 2 due to plant disturbances.

To investigate whether the learning system as well as the sensing scheme is adaptive to the disturbances and dynamics changes, the valve HCV3 was opened 35% at t = 110 sec. The results of on-line control can be seen from Figure 10.

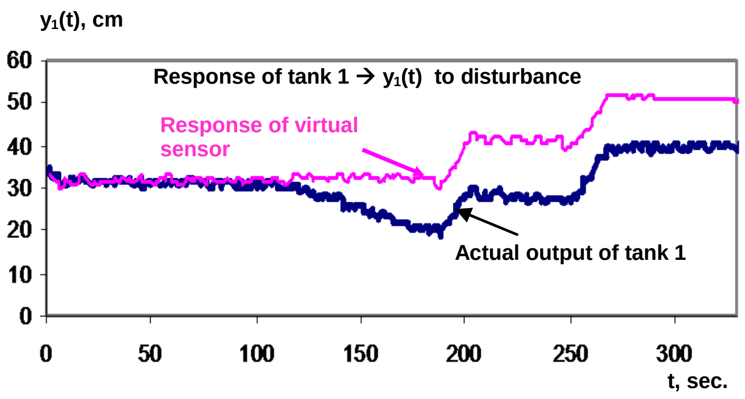

Figure 11 Results of virtual sensing scheme due to plant disturbance.

As can be observed, there is no significant response change due to disturbance. The ANFIS controller responsed to the change satisfactorily. This is due to the on-line learning scheme implemented in the control scheme. The result because of this disturbance can be observed better in Figure11, where there was a decrease of water level in tank 1 after the disturbance. Poor response of virtual sensing scheme was also demonstrated. The reason for this is that during learning phase in virtual sensing scheme, off-line method was implemented. Inadequate response of tank 2 is also shown to the set-point change subsequently.

6 Conclusions

An alternative intelligent control strategy, which is an integration between the neuro-fuzzy based controller and virtual sensing scheme has been proposed and tested on a real-time environment for on-line control of a process mini-plant. The virtual sensing scheme predicts the immeasurable variable satisfactorily based on the information from measurable or secondary variable. The experimental results also show the effectiveness and the performance of the method in controlling nonlinear systems.

References

[1] Jang, J.S.R., Sun, C.T. & Mizutani, E., *Neuro-Fuzzy and Soft Computing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1997.

[2] Nazaruddin, Y.Y. & Yamakita, M., *Neuro-Fuzzy Based Modeling of Vehicle Suspension System*, Proc. of IEEE International Conference on Control Applications (CCA’99), Kohala Coast, Island of Hawaii, Hawaii, USA, 1999.

[3] Nazaruddin, Y.Y. , Tjandrakusuma, Paula F., *On-line Adaptive Predictive Control of a Process Mini-Plant Using Neuro-Fuzzy Based Modelling*, Proc. of the IEEE International Conference on Mechatronics and Machine Vision in Practice (M^2VIP'2001), Hong Kong, August 27-29, 2001.

[4] Nazaruddin, Y.Y., Kurniawan, I. & Samsi, A., *Control of a Neutralization Process by an Adaptive Neuro-Fuzzy Controller with Genetic Algorithm*, Proc. the 3rd. International Conference on Control Theory and Application (ICCTA'01), Pretoria, South Africa, December 12-14, pp. 459-463, 2001.

[5] Nazaruddin, Y.Y., Waluyo, J. & Hadisupadmo, S., *Inverse Learning Control Using Neuro-Fuzzy Approach for a Process Mini-Plant*, Proc. of the International Conference on Physics and Control (PHYSCON'2003), Saint Petersburg, Russia, August 20-22, 2003.

[6] Nazaruddin, Y.Y., *Development of Intelligent Control Strategies for On-Line Control of a Process Mini-Plant*, Proc. of the AUN/SEED-Net Fieldwise Conference on Computer and Control Engineering Application, Vientiane, Laos, 2007.

[7] Denai, M.A., Palis, F. & Zeghbib, A., *ANFIS based Modelling and Control of Nonlinear Systems: a Tutorial*, Proc. IEEE International Conference on Systems, Man and Cybernetics, 4, pp. 3433–3438, 2004

[8] Wells, L.K, *LabVIEW : Graphical Programming for Instrumentation*, Prentice-Hall, Englewood Cliffs, New Jersey, 1995.

[9] Narendra, S. & Parthasarathy, K., *Identification and Control of Dynamical Systems Using Neural Network*, IEEE Transaction on Neural Network, Vol. 1. No. 1, 1990.

[10] Pham, D.T. & Liu, X., Neural Networks for Identification, Prediction and Control, Springer Verlag, London, 1995.

[11] Habtom, R., Dynamic System and Virtual Sensor Modeling Using Neural Networks, VDI Verlag GmbH, Düsseldorf, Germany, 1999.

[12] Kalman, R.E., *A New Approach to Linear Filtering and Prediction Problems*, J. Basic Engineering, Series D. 82, pp 35-45, 1960.

[13] Chui, C.K. & Chen, G., *Kalman Filtering with Real-Time Application*, Springer Verlag, Heidelberg, 1987.