Self-tuning Neuro-fuzzy Controller
by Genetic Algorithm with Application to a
Coupled-Tank Level Control System

Application Note Ref: CT-105
Date: March 22, 1997
Authors: LS Teo, UTM
         K Marzuki, UTM
         Y Rubiyah, UTM
Self-tuning Neuro-fuzzy Controller
by Genetic Algorithm with Application to a
Coupled-Tank Level Control System

IMPORTANT NOTICES

This is a reprint with permission from the authors. It reflects the authors’ opinions and are reproduced without change, in the interest of information dissemination. KentRidge Instruments Pte. Ltd. assumes no responsibility for any errors or consequential damages that may result from the use or misinterpretation of any information contained in this article. Inclusion of this article also does not necessarily constitute endorsement by KentRidge Instruments Pte. Ltd.

Acknowledgements

The KRi Coupled-Tank Control Apparatus PP-100 is designed and manufactured by KentRidge Instruments Pte. Ltd. in collaboration with Hyatech, Australia.

Copyright and Reprint Permissions

Please write to authors at address: CAIRO, University Teknologi Malaysia, Jalan Semarak, 54100 Kuala Lumpur, Malaysia. Email address: marzuki@kired.utm.my or fax: +60 3 390 4892
SELF-TUNING NEURO-FUZZY CONTROLLER BY GENETIC ALGORITHM
WITH APPLICATION TO A COUPLED-TANK LEVEL CONTROL SYSTEM

Teo Lian Seng, Marzuki Khalid*, and Rubiyah Yusof

CAIRO, University Teknologi Malaysia,
Jalan Semarak, 54100 Kuala Lumpur, Malaysia.
Email address: marzuki@klred.utm.my
Fax number: 603-2904892
(All correspondence should be sent to *)

ABSTRACT

This paper describes a self-tuning Neuro-fuzzy Controller by Genetic Algorithm (NFCGA) applied to a coupled-tank fluid level control. The controller use a simplified fuzzy algorithm which based on the Radial Basis Function neural network (RBF). Genetic Algorithm (GA) is used to tune simultaneously all the parameters of controller from a random state. A plant model is used initially to tune the parameters of the neuro-fuzzy controller. A coupled-tank system for fluid level control is used as a test bed. Nonlinearity is inherent in the plant due to the pumps, valves and sensors. The NFCGA is compared with a manually tuned conventional fuzzy logic controller (CFLC) and a PID controller in terms of setpoint tracking, load disturbance rejection and changes in the plant dynamics. It is found that the NFCGA copes well over the complexities in the plant, and has several advantages over the other two controller.

1. INTRODUCTION

Fuzzy logic and neural networks are two AI techniques that have been recently applied to many engineering problems. However, the ‘implicit knowledge’ distributed in the neural network structure is difficult to extract, and also, learning is not easy to be implemented in fuzzy logic[1]. Therefore, in recent years, the hybrid paradigms of fuzzy logic and neural network, namely, the neuro-fuzzy techniques are being viewed as a promising approach to integrate the advantages of both techniques while compensating the disadvantages of each other [2-4]. One of the more popular neuro-fuzzy paradigms is the RBF neural network based fuzzy system due to their close structural and computational similarity [5-8].

Genetic algorithm (GA) is the robust search algorithm based on the Darwinian survival of the fittest in natural evolution. It is proven to be an effective optimization mechanism in complex search spaces [9-10]. In control engineering, GA is usually used to overcome the difficulty and complexity in the manual tuning of the fuzzy system parameters, for example, scaling factors, membership functions and control rules configuration [11-12]. The generic method of tuning the FLC parameters by GA has normally been implemented as partial or sequential tuning. However, in the design of the FLC, its parameters are not independent from each other. Thus, it is important to consider them simultaneously when tuning of these parameters by GA [13,14].

A computer model can be used to simulate the controllability and performance of a fuzzy control system. The fitness of the corresponding FLC is then formulated based on the plant model’s response through a predefined performance function[13-14]. Although there has been a substantial amount of research in using GA to tune the FLCs [11-12,15-17], applications to real time process control has not much been investigated. This paper attempts to use an off-line GA tuned neuro-fuzzy controller (referred to as NFCGA) for real-time fluid level control of a coupled-tank. The neuro-fuzzy controller is based on the RBF neural network. All of the parameters of the controller are tuned by GA simultaneously from an initial random state. By using a mathematical plant model, the NFCGA parameters are then encoded and tuned by GA based on a predefined performance criteria. After convergence, the NFCGA is used as the controller for the fluid level control in real-time. The results are compared to a manually tuned conventional FLC and a PID controller.

2. DESCRIPTION OF THE COUPLED-TANK PLANT

A laboratory scale coupled-tank system developed by KentRidge Instruments, Singapore is used as a test bed for the proposed NFCGA. It consists of two tower-type tanks with an internal baffle in between as shown in Fig. 1. The baffle can be raised to control the leakage from one tank to another in a two-way manner, which makes the dynamics different from the regular two-tank system[18]. To measure the level of fluid in each tank, a capacitive-type probe sensor is used. It is observed that the raw feedback signal is rather noisy, and this system is not equipped with any advanced signal conditioning circuit, thus, providing a challenging control problem. The system is configured as a SISO control problem by slightly raising the baffle, where the plant dynamics can be simplified and linearised to a second order transfer function. The control objective is to control the fluid level in Tank #2 by manipulating the incoming voltage of Pump #1.
By considering the mass balance of the fluid, the Bernoulli’s equations for steady, non-vicious, incompressible fluid principles, and a few assumptions on the derivation, the non-linear plant dynamic can be simplified to a linearised perturbation model as follow [19]:

\[
A_1 S h_1(s) = q_1(s) - \left[ \frac{a_1}{2\sqrt{H_1}} + \frac{a_3}{2\sqrt{H_1 - H_2}} \right] h_1(s) + \frac{a_3}{2\sqrt{H_1 - H_2}} h_2(s)
\]

\[
A_2 S h_2(s) = q_2(s) - \left[ \frac{a_2}{2\sqrt{H_2}} + \frac{a_3}{2\sqrt{H_1 - H_2}} \right] h_2(s) + \frac{a_3}{2\sqrt{H_1 - H_2}} h_1(s)
\]

where \( Q_1 \) and \( Q_2 \) are the volumetric flow rate of Pump #1 and Pump #2 respectively. \( a_1, a_2 \) and \( a_3 \) are proportionality constants of the corresponding \( \sqrt{H_1}, \sqrt{H_2} \) and \( \sqrt{H_1 - H_2} \) terms. \( q_1 \) and \( q_2 \) are small variations in \( Q_1 \) and \( Q_2 \) respectively, \( h_1 \) and \( h_2 \) is the resulting perturbation resulted by \( q_1 \) and \( q_2 \) respectively.

The objective of the system is to control the fluid level in Tank #2 by controlling the flow rate of the fluid into Tank #1. The time delay of the system is influenced by the transferability of the fluid by the tube, and also the opening of the baffle. Besides the non-linear characteristic of the plant dynamics, the volumetric flow rates of the pump given by the operating voltage also exhibit a non-linear behavior.

3. FORMULATION OF THE SELF-TUNING FUZZY LOGIC CONTROLLER

Basically, fuzzy logic control involves three main stages: fuzzification, inferencing, and defuzzification. The first and last stages are needed to convert and re-convert real world crisp signals into fuzzy values and vice-versa. The inference or reasoning mechanism can be described as follows, the first is to determine the matching degree of the current fuzzy input (class) with respect to each rule, i.e. the IF part. The mechanism then decides which rules are to be fired according to the input field. Finally, the fired rules are combined to form the control actions, i.e., the THEN part or consequents of the fuzzy control rules. The above procedure can be further simplified to as only pattern matching and weights averaging, thereby, eliminating the procedure of fuzzification and defuzzification [6-7,20-21].

The first operation of the algorithm deals with the IF part of the fuzzy control rules; it determines the matching degree of the current input to the condition of each of the fuzzy control rules. The matching degree process is simply an operation that returns the matching level, \( h_i \in [0,1] \) between the inputs and the rule pattern for the \( i^{th} \) rule. The matching formula can be written as follows:

\[
h_i = \exp \left( - \frac{\left\| C_{x,n} \cdot x_n - D_{x,n} \right\|^2}{2} \right)
\]

for \( i = 1 \) to \( T \)

where \( T \) is the total number of fuzzy rules, and \( \| \cdot \| \) is the norm operator presented as either Euclidean, Hamming, Maximum, etc.. In this paper, the following is used:

\[
\| \cdot \| = \left( \sum_{n=1}^{N} \left( \cdot_n \right)^2 \right)^{\frac{1}{2}}
\]

Then averaging of the weights is applied to obtain the control action of each output variable, \( y_m \) which corresponds to the input vector \( x \), where the control action \( y_m \) is given by:
This weight averaging method uses only the center of the *THEN* part of the rule \((c_{y,m}^i)\), which is defined as a singleton output variable. The algorithm, therefore can be understood as a modification of the maximum membership decision scheme, where the global center is calculated by the center of gravity algorithm.

The computational procedures of the above fuzzy algorithm is closely similar to the implementation of a normalized version of radial basis function neural network (RBF). However, the center and the width of the radial basis nodes in the RBF are determined differently, i.e., by clustering algorithms and ‘root mean square’ method respectively[21-22]. The radial function of the RBF can be treated as the fuzzy membership functions which is characterize by two parameters, i.e., the center and the width. By appropriately choosing the center and width of the radial basis units, which forms the fuzzy membership functions, the network can be used to represent the rule-base fuzzy knowledge. Each of the fuzzy rule inference mechanism is being process in a radial basis node, thus, the \(i^{th}\) rule in (2.2) can be written as:

\[
\text{IF } (c_{x,1}^i, D_{x,1}^i) \text{ and } \ldots \text{ (}c_{x,n}^i, D_{x,n}^i\text{) and } \ldots (c_{x,N}^i, D_{x,N}^i) \\
\text{THEN } (w_{i,1}) \text{ and } \ldots (w_{i,m}) \ldots \text{ and } (w_{i,M})
\]

where the control action is determined by the weights connection \(w_{im}\), a singleton real number value.

A general block diagram of a multi-input and multi-output RBF based FLC is shown as in Fig. 2. The input layer accepts the system state feedback \((x_1, x_2, ..., x_n, ...., x_N)\) (input vector) , and the fuzzy inferencing is processed at the hidden layer. The output layer implements the normalization operation to produce the control signals \((y_1, y_2, ..., y_m, ...., y_M)\).

In order to further visualize this concept, consider a fuzzy logic control system where the FLC has two input variables, namely, the error \(e\) and the change of the error \(\Delta e\). Each of these variables take five Gaussian type of fuzzy membership functions that are labeled as PB, PS, Z, NS and NB. Each of the membership functions has two parameters, i.e., the center and width of the Gaussian functions. The multi-variate Gaussian can also be viewed as the product of a single-variate Gaussian function. It performs conjunctive operation in the ‘premise’ part of the fuzzy rules in the hidden layer. Figure 3 shows the rule base matrix of the corresponding fuzzy basis units at the hidden layer of the controller. Each of the kernel squares represents one control rule. Thus, the number of the hidden nodes for this network is exactly equal to the number of fuzzy control rules. The output of these units is the matching degree or inferred result of the particular fuzzy control rules.

\[
y_m = \frac{\sum_{i=1}^{p} (h_i \cdot c_{y,m}^i)}{\sum_{i=1}^{p} h_i}
\]

The strength of the controller output depends on the interconnected weights between the hidden layer and the output. The output is computed by normalizing the weights as follows:

\[
y_m = \frac{\sum_{i=1}^{p} (h_i \cdot w_{im})}{\sum_{i=1}^{p} h_i}
\]

where \(P\) is the number of hidden units (rules), and \(h_i \in [0,1] \in \mathbb{R}\) is the output of the fuzzy basis function of each rule, \(w_{im}\) is the weight that connects the \(i^{th}\) local unit to the \(m^{th}\) output node. Figure 4 shows graphically of how such computation is carried out. It can be viewed as the modified center of gravity defuzzification strategy. The controller
output $y_1$ is a crisp value that can be readily applied to the system. GAs are then implemented as an optimization algorithm to tune all the parameters of this RBF-based FLC, which is discussed in the next section.

4 IMPLEMENTATION OF THE SELF-TUNING FLC BY GENETIC ALGORITHM

This section discusses how the proposed self-tuning FLC is formulated by using the GA approach, where all the parameters of the FLC are initially randomized, then being tuned and optimized simultaneously by GA. The basic GA concept is first discuss briefly.

4.1 Genetic Algorithm

Genetic Algorithms (GAs) are random search algorithm that imitates natural evolution with Darwinian survival of the fittest approach. The coding method allows GAs to handle multi-parameters or multi-model type of optimization problems easily, which is rather difficult or impossible to be treated by classical optimization methods. The population strategy enables GAs to search the near optimal solutions from various parts and directions within a search space simultaneously. GAs process each chromosome independently and make it highly adaptable for parallel processing. It needs no more than only the relative fitness of the chromosomes, thus, it is rather suitable to be applied to systems that are ill-defined. GAs can also work well for non-deterministic systems or systems that can only be partially modeled. GAs use random choice and probabilistic decision to guide the search, where the population improves toward near optimal points from generation to generation. The fundamental of GAs consist of three basic operations: reproduction, crossover, and mutation. Further discussion on GAs can be obtained in [13-14].

4.2 Coding strategy of the FLC parameters

In this paper, the NFCGA as shown in Fig.2 is configured to have two inputs ($X_1$, $X_2$) and one output ($y$), which is the controlled variable. The Gaussian membership functions has the center $C_{x_1}^i$ ($C_{x_2}^i$) and the width $D_{x_1}^i$ ($D_{x_2}^i$) for the controller’s input $X_1$ and $X_2$ respectively. In the following experiments, each of the input fuzzy variables is quantified into five membership functions, therefore, resulting in 20 parameters. Our initial investigations showed that increasing the number of membership functions does not significantly improve the experimental results. Furthermore, it increases the complexity of GA searching process. On the other hand, reducing the number of membership functions, however, does have an effect on the accuracy of the problem. With these fuzzy input membership functions, there are 25 fuzzy radial units (5x5) at the hidden layer, thus, 25 weights ($w_{ij}$) are needed to connect the hidden units to the output node, given as $w_{ij} = [w_{11}, w_{12}, w_{13}, \ldots, w_{21}, w_{22}, w_{23}, \ldots, w_{55}, w_{56}]$. Thus, a total of 45 parameters ($5 \times \text{membership functions} \times 2 \times \text{parameters} + 25 \times \text{weights}$) are needed to be tuned by the GA, which is much less than the parameters of a conventional FLC and also the Takagi-Sugeno FLC when configured in the same manner. The Linear Mapping Method [20,27] is used to encode the NFCGA parameters, which can be expressed as follows:

$$g_q = G_{q_{\text{min}}} + (G_{q_{\text{max}}} - G_{q_{\text{min}}}) \frac{A_q}{2^{N} - 1} \quad \text{(4.1)}$$

where $g_q$ is the actual value of the $q^{th}$ parameter, $A_q$ is the integer represented by a $N$-bit string gene. $G_{q_{\text{max}}}$ and $G_{q_{\text{min}}}$ are user-defined upper and lower limits of the gene. The encoded genes are concatenated to form a complete chromosome. Each of the parameters is encoded into 8-bit strings, resulting in a complete chromosome of 360 bits. The coded parameters of the FLC are arranged as shown in the following equation to form the chromosome of the population:

<table>
<thead>
<tr>
<th>gene</th>
<th>1</th>
<th>2</th>
<th>……</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>……</th>
<th>19</th>
<th>20</th>
<th>21</th>
<th>22</th>
<th>……</th>
<th>44</th>
<th>45</th>
</tr>
</thead>
<tbody>
<tr>
<td>chromosome</td>
<td>sub-chromosome of $X_1$</td>
<td>sub-chromosome of $X_2$</td>
<td>sub-chromosome of weights</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>parameter</td>
<td>…… ($C_{x_1}^i$, $D_{x_1}^i$)</td>
<td>…… ($C_{x_2}^i$, $D_{x_2}^i$)</td>
<td>$w_{11}$</td>
<td>$w_{12}$</td>
<td>$w_{13}$</td>
<td>$w_{14}$</td>
<td>$w_{55}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

It can be observed that Gene 1 to Gene 10 is allocated to the sub-chromosome of the first controller input ($X_1$) with the centers of membership functions (Gene 1,3,5,7,9) and the corresponding membership widths (Gene 2,4,6,8,10) at the antecedent position. Gene 11 to Gene 20 are assigned in a similar way for the second controller input ($X_2$).

In this paper, the flexible position coding strategy is applied to improve the diversity of possible input spaces partition. The order of the center-width pair genes in the sub-chromosome is not prefixed as the order of the correspondence membership function in the universe of discourse. Each time when the genes are decoded for fitness evaluation, the fuzzy memberships are rearranged in an ascending manner in its universe of discourse based on the center of the memberships. The respective weight ($w_{ij}$) is then assigned to the respective control rule condition of “IF ($C_{x_1}^i$, $D_{x_1}^i$) AND ($C_{x_2}^i$, $D_{x_2}^i$)”, as given in (3.4).
4.3 GA optimization

To describe the GA optimization process, consider the functional block diagram as shown in Fig. 5. At the beginning of the process, the initial populations comprise a set of chromosomes scattered all over the search space. In all our experiments, the population consists of 200 chromosomes which are all randomized initially. Thus, the use of heuristic knowledge of the controller is minimized.

After each of the chromosomes is evaluated and associated with a fitness, the current population undergoes reproduction process to create the next generation of population using the ‘Roulette wheel’ selection scheme [12]. After the new group of population is built, the mating pool is formed and the crossover is carried out. This is then followed by the mutation operation. Generally, after these three operations, the overall fitness of the population improves. Each of the population generated then goes through a series of evaluation, reproduction, crossover and mutation, the procedure is repeated until the termination condition is reached. After the evolution process, the final generation of population consists of highly fit strings that provide optimal or near optimal solutions.

In the evaluation routine for control purposes, one chromosome is taken and decoded to the actual value of the parameters. These sets of controller parameters is then used to control the system where it undergoes a series of tracking response of multi-step reference setpoints. The use of multi-step reference signal is to excite the different states of the system, to enable the evaluation to cover a wider system operating range. Based on the various states of the control system, the performance of the controller is calculated by using a predefined cost function. GA is then used to tune the controller parameters to minimize the accumulative cost function.

4.4 Initialization of the GA parameters

Dynamic crossover and mutation probability rates, are used in the GA operation, as they provide faster convergence when compared to constant probability rates [23]. The crossover and mutation probability rate are given by the following formula:

\[ \text{Crossover rate} = \exp(- \frac{\text{current\_generation}}{\text{the\_maximum\_generation}}) \]

\[ \text{Mutation rate} = \exp(0.05 \times \frac{\text{current\_generation}}{\text{the\_maximum\_generation}}) - 1 \]  

The proposed tuning of the NFCGA involves 200 chromosomes which are all initially randomized. The Gray-Code transformation method and elitist selection strategy are employed in the GA process. In addition, a generation gap of 0.9 is used during the reproduction operation which means 90% of the members in the new population are determined by the selection scheme employed, and the remaining 10% is selected uniformly from the old generation. This strategy helps to prevent premature convergence of the population. Two-point crossover is applied in exchanging the gene information.

4.5 The GA Evaluation Configuration

The RBF based FLC is first tuned by GA before being applied to the real time fluid level control of the coupled-tank. The fitness of the GA chromosome is evaluated by using a predefined performance index based on an estimated plant model. This approach is used because it is impractical to carry out the GA evaluation on the real plant, as the untuned controller may affect the stability of the system.

The sampling time applied for the control system is 0.5s. From the actual input-output data of the coupled-tank, a second order plant model is identified using the Least Squares Estimation method [24]. It is observed that at the steady state, the maximum rate of change of error per sampling instant cause by the noise of the sensor is slightly
higher than the actual maximum rate of change of error per sampling instant at the transient state. Thus, it is difficult to determine the real system states based on the change of error per sample. As the NFCGA takes the change of error as one of its inputs, the control scheme may seriously deteriorated, because its control action is calculated based on the state of the error and change of error. Therefore, the NFCGA’s inputs, i.e., error (e) and change of error (Δe) are defined differently from normal practice. Thus, for each sampling instant k, e(k) and Δe(k) are defined as follows:

\[
e(k) = \frac{e(k) + e(k-1)}{2}
\]

\[
Δe(k) = \frac{e(k) - e(k-4) + e(k-1) - e(k-5)}{2}
\]  

(4.4)

where \(e(k)\) is the error between the setpoint and the filtered feedback fluid level signal at the k\textsuperscript{th} sampling instant. Although this setup involves some time delay in the controller, it is useful in overcoming the above problem caused by the noise.

The task of defining a fitness function is usually application specific, such that it is formulated to achieve the goal of the controller[28]. Since the central objective of the control system is to minimize the error between the actual plant response and the set-point, a simple performance index, \(F\) is chosen as follows:

\[
F = \sum_{i=1}^{L} \sum_{k} e^i(k) \cdot k^3
\]  

(4.5)

The performance index (\(F\)) is related to fitness (\(f\)) using the following relationship:

\[
f = A / (1 + F)^g
\]  

(4.6)

where \(f\) is the fitness for the parameter set, \(F\) is the performance index, and \(g\) is a constant that affects the performance curve. \(A\) is a non-negative constant and appropriately chosen so that \(f\) will not be too small, i.e., becomes insignificant due to the large value of \(F\).

5. EXPERIMENTS AND DISCUSSIONS

Experiments were conducted to control the fluid level of the coupled-tank by NFCGA. The fluid level in Tank #2 is manipulated by controlling the voltage to Pump #1. For comparison purposes, the performances of a conventional FLC and PID controller were also investigated on the same system. The inputs of the conventional FLC was configured similarly to the RBF based FLC, and it was tuned manually for its best performance. The PID controller is initially tuned by the Ziegler-Nichols method [25] and followed by manual fine-tuning. The PID parameters for proportional, integral, and derivative gains were obtained as follows: 2.893, 0.01 and 1.239 respectively. Both of these controllers have been tuned and adjusted to give the best transient response.

The fuzzy membership functions and the weights (control action) of the NFCGA, which is tuned by GA are shown in Fig. 6(a) and Fig. 6(b).

![Fuzzy membership functions of error (e)](image)

<table>
<thead>
<tr>
<th>label</th>
<th>T1</th>
<th>T2</th>
<th>T3</th>
<th>T4</th>
<th>T5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-0.491</td>
<td>-0.255</td>
<td>0.000</td>
<td>0.202</td>
<td>0.484</td>
</tr>
<tr>
<td>center</td>
<td>width</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.081</td>
<td>0.100</td>
<td>0.041</td>
<td>0.083</td>
<td>0.146</td>
<td></td>
</tr>
</tbody>
</table>

Fig.6(a) The fuzzy inputs membership functions for the NFCGA.

![Fuzzy membership functions for change of error (Δe)](image)

<table>
<thead>
<tr>
<th>label</th>
<th>R1</th>
<th>R2</th>
<th>R3</th>
<th>R4</th>
<th>R5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-0.098</td>
<td>-0.058</td>
<td>0.000</td>
<td>0.054</td>
<td>0.106</td>
</tr>
</tbody>
</table>

Fig.6(b) The weights of the FLC for the NFCGA.
The step response of all the three controllers are shown in Fig.7. Generally, it can be seen that the three controllers are aggressive and their performances are almost similar. However, it can be observed that the control signal output from the NFCGA is smoother than other two controllers and this can prolong the actuator’s life time.

The NFCGA was also tested for load disturbance rejection capability. The disturbances were applied during the steady state, where load is added into the system by allowing Pump #2 to operate at 20% of the maximum control signal. After a specific length of time, the load from Tank #2 was removed. Figure 8 shows how the three controllers responded in these circumstances. The fluid level in Tank #2 experienced a sudden increase when the load is applied. The NFCGA managed to bring the system response back to the setpoint much quicker than the conventional FLC and the PID controller. The result is more obvious during the removal of the load. As expected, the PID controller could not cope with unexpected load disturbances due to its linear and slower algorithm. Both the FLCs on the other hand, are nonlinear controllers and therefore can cope with such problem.

In another set of experiments, the system was tested with changes in plant dynamics. This is done by totally clamping the outlet of Tank #2 gradually, i.e., no fluid flow out from this outlet. After a certain period, the clamp is released again. The changes in the plant dynamics resulted in a variation in the system response during the steady state as shown in Fig.9. The experiments showed that both NFCGA and the conventional FLC were able to cope better than the PID controller.

6. CONCLUSION

This paper has proposed a self-tuning FLC based on the RBF neural network (NFCGA), where all of its parameters are simultaneously tuned by GA. The NFCGA is then applied to a noisy coupled-tank fluid level control with non-linear dynamics in real-time. The performance of the NFCGA has been compared with a conventional FLC and PID controller in terms of transient response, load disturbance and changes in plant dynamics. It was observed that the NFCGA is more superior than the other two controllers.
The presented methodology can eliminate laborious manual tuning of the FLC parameters, such as the fuzzy membership functions, scaling factors and control rules, where GA is used to tune and provide these parameters. Although it can be argued that GA also has to be formulated in prior, its encoding technique is more systematic and less laborious. For the control problem where its actual control environment can be appropriately formulated in the computer, the experimental results also showed that the proposed methodology can be a considerable approach for designing of FLCs for real time control problems.

References