American J. of Engineering and Applied Sciences 2 (4): 669-675, 2009
ISSN 1941-7020

Improved Coupled Tank Liquid Levels System Based on Swarm Adaptive Tuning of Hybrid Proportional-Integral Neural Network Controller

M.S. Ramli, RM.T. Raja Ismail, M.A. Ahmad, S. Mohamad Nawi and M.A. Mat Hussin
Faculty of Electrical and Electronics Engineering, University Malaysia Pahang, Malaysia

Abstract: Problem statement: Accuracy and stability of many systems in chemical and process industries which has Two-Input Two-Output (TITO) is one of the key factors of process which have cross coupling between process input and output. **Approach:** Unlike traditional neural network weight adaptation using gradient descent method, Particles Swarm Optimization (PSO) technique was utilized for adaptive tuning of neural network weights adjustment and fine tuning the controller's parameters. Design approach for controlling liquid levels of Coupled Tank TITO system by using hybrid PI-Neural Network (hybrid PI-NN) controllers. **Results:** Tuning method for parameters of improved hybrid PI-NN controller was also discussed. **Conclusion:** Performances of proposed method also compared with PID-NN controllers, it was shown that hybrid PI-NN controller exhibited better performance in terms of transient response analysis.

Key words: NN, PSO, level control, water tank

INTRODUCTION

Liquid tank systems play important role in industrial application such as in food processing, beverage, dairy, filtration, effluent treatment, pharmaceutical industry, water purification system, industrial chemical processing and spray coating. A typical situation is one that requires fluid to be supplied to a chemical reactor at a constant rate. An upper tank can be used for filtering the variations in the upstream supply flow. Many times the liquid will be processed by chemical or mixing treatment in the tanks, but always the level of the fluid in the tanks must be controlled. Vital industries where liquid level and flow control are essential include petrochemical industries, paper making industries, water treatment industries[1].

In order to achieve high performance, feedback control system is adopted. Classical PID controller is widely applied in industry control such as temperature control, speed control, position control, but it is difficult for PID regulation to reach the aim of high speed and short transition time and small overshoot[2]. Advanced control methods also have been proposed by several researchers such as sliding mode control[3] and nonlinear back stepping control[4], tuning methods based on optimization approaches with the aim of ensuring good stability robustness have received attention in the literature[5,6].

Accurate model and its parameters which capture the characteristic of the coupled tank system is required for designing its controller for achieving a good performance. The specific point tackled in the study is about the advantages of using a new hybrid PI-NN instead of a PID-NN controller, consisting of PSO, Neural Network (NN) and PI controller. From a generic tuning rule the optimum settings from an Integral Squared Error criterion point of view are derived. The validation result shows that hybrid PI-NN controller much faster than PID-NN and also good robustness and small overshoot.

Process plant description: The schematic diagram of the coupled tank system considered in this study is shown in Fig. 1 where $Q_i = \{Q_{i1}, Q_{i2}\}$ are the inlet flow rate to tank 1 and 2, Q_{12} is the liquid flow rate from tank 1 to tank 2 through orifice, $Q_o = \{Q_{o1}, Q_{o2}\}$ are the outlet flow rate of tank 1 and 2 and $h = \{h_1, h_2\}$ denotes the liquid level of tank 1 and 2, respectively. In this simulation, the target is to control the level in two tanks by the inlet liquid flow from two pumps. The process input are $u = \{u_1(t), u_2(t)\}$ (voltage input to pumps) and the output are $h = \{h_1(t), h_2(t)\}$ liquid level in tank 1 and 2 respectively.

Corresponding Author: M.S. Ramli, Faculty of Electrical and Electronics Engineering, University Malaysia Pahang, Malaysia

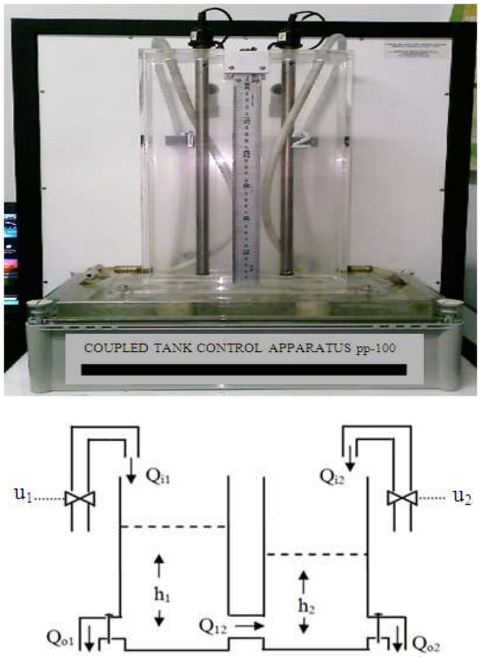

Fig. 1: Schematic of coupled tank process

The nonlinear plant equations can be obtained by mass balance equations and Bernoulli's law. After linearization process, the linear plant equations can be obtained as:

$$\begin{aligned}\dot{h}_1(t) &= \frac{k_1}{A}U_1(t) - \frac{\beta_1 a}{A}\sqrt{\frac{g}{2\bar{h}_1}H_1(t)} + \frac{\beta_x a}{A}\sqrt{\frac{g}{2\left|\bar{h}_2 - \bar{h}_1\right|[H_2(t) - H_1(t)]}} \\ \dot{h}_2(t) &= \frac{k_2}{A}U_2(t) - \frac{\beta_2 a}{A}\sqrt{\frac{g}{2\bar{h}_2}H_2(t)} + \frac{\beta_x a}{A}\sqrt{\frac{g}{2\left|\bar{h}_2 - \bar{h}_1\right|[H_2(t) - H_1(t)]}}\end{aligned} \quad (1)$$

Where:

A = The cross sectional area of tank 1 and 2 (cm^2)
a = The cross sectional area of outlet hole of tank 1, 2 and the cross sectional area of jointed opening between tank 1 and 2 (cm^2)
β_1 = The valve ratio at the outlet of tank 1
β_2 = The valve ratio at the outlet of tank 2
β_x = The valve ratio between tank 1 and 2, $\bar{h}_1$, $\bar{h}_2$ are the steady-state water level of tank 1 and 2
g = The gravity ($cm^2\,sec^{-}1$)
k_1, k_2 = The gain of pump 1 and pump 2 ($cm^3\,Vsec^{-1}$), respectively

From the linear plant Eq. 1, it can be transformed to yield a nominal block transfer function of the form (2):

$$\begin{bmatrix} h_1(s) \\ h_2(s) \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix} \quad (2)$$

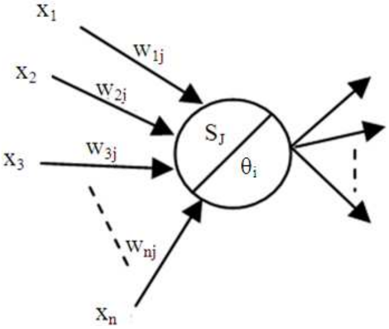

Fig. 2: Neuron form

Through simple algebraic manipulation, the transfer matrix $G_{ij}(s)$ yields to:

$$\left.\begin{aligned} G_{11}(s) &= \frac{\frac{k_1}{A}\left(s + \frac{T_x + T_2}{T_2 T_x}\right)}{\Delta} \quad G_{12}(s) = \frac{\frac{k_2}{A}\left(\frac{1}{T_x}\right)}{\Delta} \\ G_{21}(s) &= \frac{\frac{k_1}{A}\left(\frac{1}{T_x}\right)}{\Delta} \quad G_{22}(s) = \frac{\frac{k_2}{A}\left(s + \frac{T_x + T_2}{T_1 T_x}\right)}{\Delta} \end{aligned}\right\} \quad (3)$$

$$\Delta = s^2 + \left(\frac{T_1T_x + T_2T_x + 2T_1T_x}{T_1T_2T_x}\right)s + \left(\frac{1}{T_1T_2} + \frac{1}{T_1T_x} + \frac{1}{T_2T_x}\right)$$

provided that T_1 is the time constant of tank 1, T_2 is the time constant of tank 2 and T_x is the time constant interaction between tank 1 and 2.

According to transfer matrix $G_{ij}(s)$ in (2) and (3), the transfer functions of coupled-tank process are second order form which have cross coupling between process input and outputs. The decoupling controllers are required for minimizing the effects from cross coupling and transform TITO plant transfer function into SISO form. This is where neural network structure is introduced at which can be functioning as the de-coupler controller.

PID neural network controller: A control structure for controlling the liquid level tank using PID neural network controller as shown in Fig. 2 where it has an input s_j and an output θ_j. The property of a neuron is decided by the input-output activation function (f) whereby the P-neuron, I-neuron and D-neuron are representing the Proportional (P) function, Integral (I) function and Derivatives (D) function, respectively.

For any neuron (namely the jth neuron) in the network which has n-1 inputs, at any t time, the input of the neuron is given by:

$$s_j(t) = \sum_{i=1}^{n-1} w_{ij} x_i(t) \tag{4}$$

Where:

$x_i(t)$ = The outputs of n-1 connected neurons in foregoing layer and

w_{ij} = The connected weights

The output of this neuron will depend on its activation function which can be Proportional (P), Integral (I) or Derivative (D) functions. The neuron's output for each function is shown in Table 1.

As a rule, a basic PID-NN consists of two input neurons and one output neurons whereby the hidden layer of this network structure is made of three neurons which each representing P, I and D activation function respectively. The basic PID-NN is as shown in Fig. 3.

Through connective weight adaptation between layers, the PID-NN is actually acting as a conventional PID controller. Since PID controllers have been widely used in industry, that is to say there are much experience to choose P, I and D parameters in order to suit the system's stability without changing one's plant.

The control system for controlling the coupled tank level system consists of several basic PID-NN whereby every basic PID-NN is a sub-net. The multi PID-NN control system is shown in Fig. 4.

The structure of multi PID-NN is special. If suitable connective weights are obtained, each sub-net of PID-NN is comparatively equal to a PID controller. By referring to Fig. 3 of the basic PID-NN, let say that:

$w_{1j} = +1$, $w_{2j} = -1$, $w'_{1o} = K_P$, $w'_{2o} = K_I$, $w'_{3o} = K_D$

Table 1: Activation function for each type of neuron

Type of neuron	Output, θ_j (t)
P	$s_j(t)$
I	$\int_0^t s_j(t)dt$
D	$\frac{ds_j(t)}{dt}$

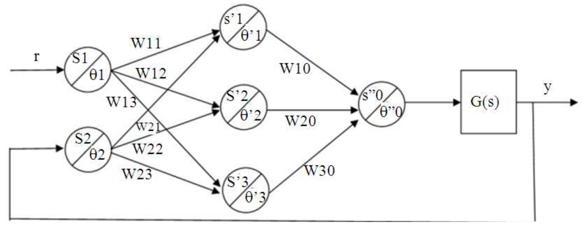

Fig. 3: Basic PID-NN

Then, the input to the network structure will be:

$$s'_j = \sum_i w_{ij} x_i = r - y = e \tag{5}$$

Meanwhile, the network output (depending on the type of neuron) of the hidden layer for each neuron is obtained as:

$$\theta'_j = f(s'_j) \tag{6}$$

Therefore, we derived the total output for the basic PID-NN as shown in Fig. 3 as:

$$\theta''_o = \sum w'_{jo} x'_j = \left[K_P e + K_I \int edt + K_D \, {}^{de}\!/_{dt}\right] \tag{7}$$

At any rate, the manipulated variable signals injected into the plant as shown in Fig. 4 is obtained as:

$$U_k = \theta''_k \tag{8}$$

Hybrid PI-neural network controller: A combinational PI controller with neural network structure for controlling the liquid level system of the coupled tank as follows.

Proportional-Integral (PI) controller is a feedback controller which drives the plant to be controlled with a weighted sum of error (difference between output and desired response) and the integral of that value. The general model for a PI controller is given in Eq. 9:

$$G_{PI_k}(s) = \frac{H_k(s)}{E_k(s)} = \frac{sK_{P_k} + K_{I_k}}{s} \tag{9}$$

Where:

H_k = The process variable

E_k = The difference between the output and the desired response

K_{Pk} and K_{Ik} = The proportional and integral gains respectively

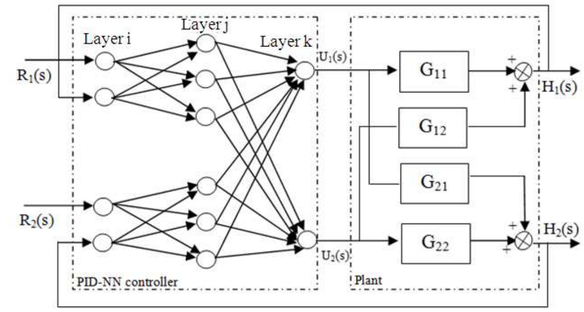

Fig. 4: (TITO) process with multi PID-NN control system

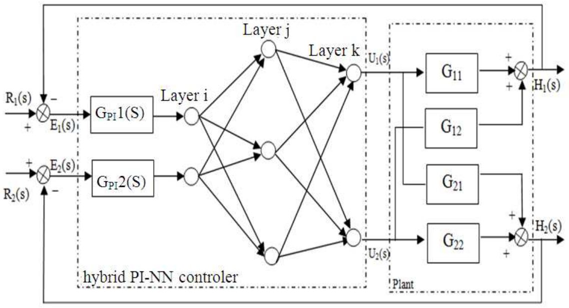

Fig. 5: (TITO) process with hybrid PI-NN control system

The hybrid PI-NN is constructed by series cascading the PI controllers with a neural network structure as shown in Fig. 5 Throughout the network, the linear activation function is used in all neurons.

Figure 5 shows the plant transfer function $G_{ij}(s)$ that has the cross coupling between process inputs and outputs. Because of interaction between processes, the neural network structures will basically act as a de-coupler controller for minimizing the cross coupling effects via its connective weight adaptation.

In contrast to the PID-NN, the manipulated variable signals injected into the plant for the hybrid PI-NN is obtained as:

$$U_k = O_k \tag{10}$$

where we have:

$$O_k = \sum_j W_{kj} O_j$$

and

$$O_j = \sum_i W_{ji} O_i$$

The net output O_i on the other hand, comes from Eq. 9 which yield to $O_i = G_{PI_k}(s)E_k(s)$

Particles swarm optimization:

Overview of the PSO: PSO is a method for optimizing hard numerical function on metaphor of social behavior of flocks of birds and schools of fish. This technique has been widely used in across wide range of application such as in communication, biomedical[ ]. It also has, very recently, emerges as an important combinatorial metaheuristic technique for both continuous-time and discrete-time optimization. In past several years, PSO algorithms have been successfully applied in many research and application areas. It has been demonstrated that PSO gets better results in a faster and cheaper way as compared with other methods. First introduced by Eberhart and Kennedy[8], PSO, like other evolutionary computations, can typically initialize a pool of particles with random bird positions (called agent) in two-dimensional space[9] where each is represented by a point in the X-Y coordinates and the velocity is similarly defined. Bird flocking is assumed to optimize a certain fitness function. Each agent knows its best value so far (pbest) and its current position. This information is an analogy of personal experience of an agent. Each agent tries to modify its position using the concept of velocity. The velocity of each agent can be updated by the following equation:

$$\psi_i^{k+1} = \omega\psi_i^k + \eta_1\Gamma_1(\text{pbest}_i - \Omega_i^k) + \eta_2\Gamma_2(\text{gbest} - \Omega_i^k) \tag{11}$$

Where:

Ψ_i^k = The velocity of agent i at iteration k

ω = Weighting function

η_1 and η_2 = Weighting factors

Γ_1 and Γ_2 = The cognitive and social learning parameters which generated randomly between 0 and 1

Ω_i^k = The current position of agent i at iteration k

p_{besti} = The p_{best} of agent i

g_{best} = The best value so far in the group among the p_{bests} of all agents

The following weighting function is normally used in Eq. 11:

$$\omega = \omega_{max} - \left(\frac{\omega_{max} - \omega_{min}}{iter_{max}}\right) \times iter \tag{12}$$

Where:

ω_{max} = The initial weight

ω_{min} = The final weight

$iter_{max}$ = The maximum iteration number

iter = The current iteration number

Using the previous equation, a certain velocity, which gradually brings the agent close to pbest and gbest, can be calculated. The current position (search point in the solution space) can be modified by the following equation:

$$\Omega_i^{k+i} = \Omega_i^k + \Psi_i^{k+i} \tag{13}$$

At some iteration, the position of the agent based on Eq. 13 might be flying-off from the initial limit. Hence, a fly-back algorithm is implemented to bring back the agent to within the limit. The fly-back pseudo-code used in the program is presented below:

```
If    $\Omega_i^{k+1}$ less than $\Omega_{min}$
          $\Omega_i^{k+1} = \Omega_{min}+(\Omega_{max}-\Omega_{min)X}$ r and
else if $\Omega_i^{k+1}$ more than $\Omega_{max}$
          $\Omega_i^{k+1} = \Omega_{min}+(\Omega_{max}-\Omega_{min})$X r and
end
```

Model reference adaptive tuning using PSO: In both PID-NN and hybrid PI-NN control systems; the aim of the controllers' algorithm is to minimize the following fitness function (f_t):

$$f_t = \sum_{k=1}^{n} E_k = \frac{1}{m}\sum_{k=1}^{n}\sum_{q=1}^{m}\left[\{G_{ref}R_k\}(q) - H_k(q)\right]^2 \tag{14}$$

Where:
R_k = The desired set-points and
H_k = The outputs of the system as shown in Fig. 4 and 5

Meanwhile, q (=1,2,…,n) is the serial number. G_{ref} is the first order model reference transfer function and is represented as:

$$G_{ref}(s) = \frac{1}{\tau s + 1} \tag{15}$$

where, τ the time constant for shaping the output transient responses to be as desired.

The connective weights of PID-NN and hybrid PI-NN as well as the PI parameters are changed and optimized on each iteration k of the PSO. Before beginning the optimization, a population size (i.e., number of particles) N and a maximum number of iterations $iter_{max}$ are chosen. The computation flow of PSO technique can be described in the following steps:

Step 1: Randomly initialize the population: select the (normalized) particle positions $\Omega^i = \left[\Omega_1^i, \Omega_2^i, \cdots, \Omega_N^i\right]$ and velocities $\Psi^i = \left[\Psi_1^i, \Psi_2^i, \cdots, \Psi_N^i\right]$ (i = 1,2,…,N) from uniform distributions with $\Omega_i^j \in \{\Omega_{min}, \Omega_{max}\}$ and $\Psi_i^j \in \{0, 0.1(\Omega_{max} - \Omega_{min})\}$, j = 1,2,…,N.

Step 2: Evaluate the fitness function values by f_t (Ω^i) assigning each Ω^i as the neural network weights and the controller's parameters.

Step 3: Assign the global and local best positions: Set the local best position for each particle using $pbest^i = \Omega^i$ and compare the evaluated fitness values and find the global best position $gbest^i = \Omega^i$, for some 1≤ J ≤ N, such that $f_t(\Omega^j) \le f_t(\Omega^i)$ for I = 1, 2… N.

Step 4: Search for minimum value of f_t:

- Update the particle velocities Ψ^I according to Eq. 11
- Update all positions Ω^i using formula (13). Check all positions to ensure that $\Omega_{min} \le \Omega_j^i \le \Omega_{max}$. If any of the components of the position vectors go out of bounds, they can be called back using the fly-back algorithm
- Evaluate $f_t(\Omega^i)$ (I = 1,2,…,N)
- Update the local best position: if $f_t(\Omega^i) < f_t(pbest^i)$ Then $pbest^i = \Omega^i$
- Update the global best position gbest, by letting $gbest^i$ = Ωi, for some 1 ≤ J ≤ N such that $f_t(\Omega^j) < f_t(\Omega^i)$ for (I = 1,2,…,N)

Step 5: Repeat Step. 4 until a goal is reached or the number of iterations is surpassed.

RESULTS AND DISCUSSION

The parameters of the coupled tank system are taken as follows:

Cross sectional area of tank 1and 2, A	= 66.25 (cm^2)
Height of each tank H	= 18.5 (cm)
Area of the coupling orifice, a	= 0.1963 (cm^2)
Valve ratio at the outlet of tank 1, β_1	= 0.35903
Valve ratio at the outlet of tank 2, β_2	= 0.345848
Valve ration of the outlet between tank 1 and 2, β_x	= 0.38705
Gravitational rate g	= 981 cm sec^{-12}

The liquid levels of the coupled tank system are required to follow step responses within the range of 0~300 cm (0-100%). System responses namely the liquid level for both tank 1 and 2 are observed. The minimum and maximum values of the controlled manipulated variables are capped to u_{min} = 0 volt and u_{max} = 5 V.

In the training stage, initialize the parameters of PSO as following.

For hybrid PI-NN, there are additional $K_{P's}$ and $K_{I's}$ for the PI controllers. Population size = 20, inertia weight factor ω is set according to (20) where $\omega_{max} = 0.9$ and $\omega_{min} = 0.1$. Cognitive and social learning constants are $\Gamma_1 = \Gamma_2 = 1.4$. The value in every position can be clamped to the range $[\Omega_{min}, \Omega_{max}]$ using fly-back algorithm to reduce the likelihood of particles leaving the search space. The number of iterations is $iter_{max}$ = 200. The time constant for the model reference is chosen as τ = 20s.

Figure 6 and 7 shows the liquid level responses of coupled tank system using PID-NN and hybrid PI-NN controllers for tank 1 and 2, respectively. It is noted that both controllers can track the step responses of 150 cm. However, the hybrid PI-NN shows a better performance in terms of time response specifications and integral square error as compared to the PID-NN controller. For the time response performance of the liquid level in tank 1, the PID-NN produces settling time and rise time of 207 and 15.7 sec, whereas the hybrid PI-NN produces settling time and rise time of 28.3 and 22 sec. In the mean time, PID-NN produces settling time and rise of 209 and 24 sec whereas the hybrid PI-NN produces settling time and rise time of 39.7 and 22.2 sec for the liquid level in tank 2. It shows that the PID-NN results in a slower response as compared to hybrid PI-NN. It can be said that with higher number of connective weights of neural network structure, the complexity to compute the required manipulated variables will increase and affect the speed of the response.

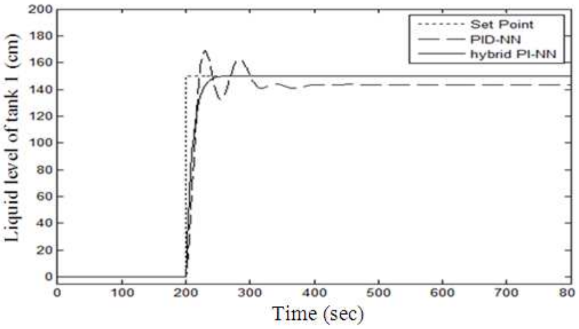

Fig. 6: Simulated response of the liquid level in tank 1

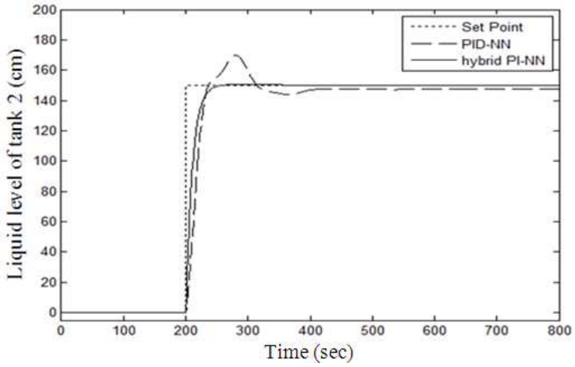

Fig. 7: Simulated response of the liquid level in tank 2

In term of integral square error for liquid level in tank 1, the hybrid PI-NN results in twice less of ISE as compared to the PID-NN with the value of 1.145×10^5 and 2.567×10^5 respectively. Similarly for liquid level in tank 2, the ISE of hybrid PI-NN and PID-NN were obtained as 1.209×10^5 and 2.567×10^5, respectively. The comparative assessment of both controllers is shown in Table 2

Figure 8 and 9 shows the liquid level responses in tank 1 and 2, respectively, with a step disturbance of 40 cm injected into the process variable of tank 1 during the steady-state response. Noted that the PID-NN controller produced maximum percentage overshoots of 35 and 41% for liquid level in tank 1 and 2 whereas hybrid PI-NN produces 27 and 18%, respectively. Furthermore, it could be seen that hybrid PI-NN produce minimum oscillation as compared to PID-NN in response to disturbance injection. It is proven that the hybrid PI-NN controller results in faster settling time and minimum overshoot. Besides that hybrid PI-NN also exhibits good robustness in minimizing the cross-coupling effect between two tanks.

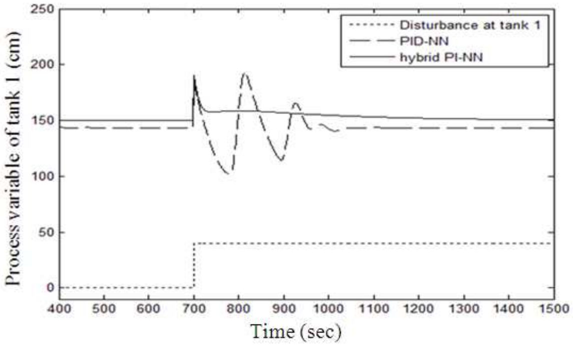

Fig. 8: Simulated response of the liquid level in tank 1 with disturbance

Table 2: Performance comparison of liquid level in tank 1 and 2

	Settling time (sec)		Rise time (sec)		Overshoot (%)		ISE ($\times10^5$)	
Controller	Tank 1	Tank 2	Tank 1	Tank 2	Tank 1	Tank 2	Tank 1	Tank 2
PID-NN	207	209.0	15.7	24.0	17.78	15.41	2.391	2.567
Hybrid PI-NN	383	39.7	22.0	22.2	0.00	0.067	1.145	1.209

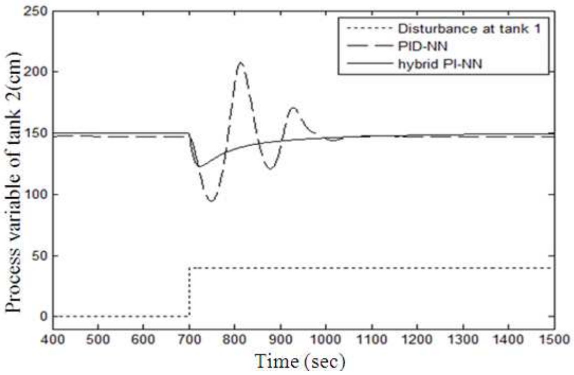

Fig. 9: Simulated response of the liquid level in tank 2 with disturbance at tank 1

CONCLUSION

This study introduces an improved hybrid PI-NN controller for the coupled system. The NN weights connective and controller parameters are optimized by utilizing the PSO algorithm via model reference adaptation. The proposed method provides a better performance with respect to PID-NN controller even under disturbance injection. The simulations for both PID-NN and hybrid PI-NN controllers are also performed and compared. Based on the results, it can be concluded that hybrid PI-NN is more robust and can provide more stable responses than PID-NN.

ACKNOWLEDGEMENT

This study was supported by Faculty of Electrical and Electronics Engineering, University Malaysia Pahang, under Control and Instrumentation (COINS) Research Group.

REFERENCES

1. Visioli, 2004. A new design for a PID plus feedforward controller. J. Process Control, 14: 457-463. DOI: org/10.1016/j.jprocont.2003.09.003\%20
2. Tan, K.K., S. Huang and R. Ferdous, 2002. Robust self tuning PID controller for nonlinear systems. J. Process Control, 12: 753-761. DOI: 10.1016/S0959-1524(02)00005-7
3. Almutairi, N.B. and M. Zribi, 2006. Sliding mode control of coupled tanks. Mechatronics, 16: 427-441. http://cat.inist.fr/?aModele=afficheN&cpsidt=17958798
4. Pan, H., H. Wong, V. Kapila and M.S. De Queiroz, 2005. Experimental validation of a nonlinear back stepping liquid level controller for a state coupled two tank system. Cont. Eng. Pract., 13: 27-40. http://cat.inist.fr/?aModele=afficheN&cpsidt=16207892
5. Xu, Z. and Q. Zhao, 2002. A novel approach to fault detection and isolation based on wavelet analysis and neural network. Proceedings of the 2002 IEEE Canadian Conference on Electrical and Computer Engineering, pp: 572-577. DOI: 10.1109/CCECE.2002.1015290
6. Aminian, M. and F. Aminian, 2007. A modular fault-diagnostic system for analog electronic circuits using neural networks with wavelet transform as a preprocessor. IEEE Trans. Circ. Syst. Instrument. Measur., 56: 1546-1554. DOI: 10.1109/TIM.2007.904549
7. Poli, R., 2008. Analysis of the publications on the applications of particles swarm optimization. J. Artifi. Evolut. Appli., 1-10. DOI: 10.1155/2008/685175
8. Eberhart, R.C. and J. Kennedy, 1995. A new optimizer employing particles swarm theory. Proceedings of 6th International Symposium Micro Machine and Human Science, Oct. 4-6, Nagoya, Japan, Piscataway, pp: 39-43. DOI: 10.1109/MHS.1995.494215
9. Reynolds, C., 1987. Flocks, herds and schools: A distributed behavioral model. Comput. Graph., 21: 25-34.