International Journal of Information Science and Computing: 4(2): December 2017: p. 41-56

DOI: 10.5958/2454-9533.2017.00005.9

Neural Optimized Autotuning Fuzzy Logic Controller for Spherical Tank Process

Boo. Poonguzhali *and R. Vinodha

Department of Electronics and Instrumentation Engineering, Annamalai University, Annamalainagar, India

*Corresponding author: bookuzhali@gmail.com

ABSTRACT

In any real time control application, the controller is expected to give optimal results irrespective of plant non-linearity, operating point change, component non-linearity and saturation. Fuzzy Logic Control (FLC) meets the control objective as it crafts the human experience in the form of If-Then rules. As the gain factors or scaling factors are constant in conventional FLC, Autotuning FLC with tuned scaling factors take over the charge. Further, in real time controllers, the memory constraint is dominant. As a solution, reduction of existing fuzzy rule base by subtractive clustering and optimization of the reduced rule set via neural network is proposed. Autotuning of input scale factors clubbed with reduction and optimization of a controller rule set is proposed to meet out the dynamic control of real time spherical tank process. The efficacy of the proposed approach is compared with real time servo and regulatory results of conventional FLC and AFLC of the spherical tank process

Keywords: Autotuning, neural, optimization, reduced rules

The need for simple advanced techniques is essential in process industries where real time processes are generally complex and non-linear[1]. PID is the best known controller in industrial processes, however limited due to non-linearities[2, 3]. Fuzzy controllers are then successfully employed in industries as they handle non-linear systems, because of their knowledge based control decisions. The major drawback in Fuzzy controllers is the lack of systematic approach in tuning. The design of fuzzy controller involves three steps, namely construction of rule base from the experience of the skilled human operator, selection of relevant membership functions and final appropriate selection of scaling gains of the controller[4]. To achieve required performance, the above said factors are adjusted suitably.

A standard Fuzzy Logic Controller (FLC) is usually defined by a set of fuzzy parameters which specifies which control action to be taken for a given process state. It has been proved that the FLC can be used

for nonlinear control with proper choice of the parameters[5, 6]. Hence, the key issue for designing an FLC is to well define its parameters such as the knowledge base and scaling factors. However, the optimal setting of parameters cannot cope up with varying working conditions as the fixed parameters will not guarantee the ideal performance. A solution to this problem is addressed by tuning the FLC parameters adapting to varying conditions.

Different types of adaptive mechanisms have been developed in FLCs[14-16] and implemented for various practical processes. Of the various tunable parameters, scaling factor's (SF) have the highest priority due to their global effect on the control performance.

Neural control schemes are preferred for nonlinear time varying processes[7]. ANFIS is a kind of neural network based on Takagi-Sugeno Fuzzy Inference System. Neuro-fuzzy integrates both the principles of fuzzy and neural networks and also captures the dual benefits in a single framework resulting in a hybrid system. It is capable of reasoning and learning in an uncertain and imprecise environment. The system will be more precise when neuro-fuzzy is used, because it contains human like reasoning style and learning with connection structure also.

ANFIS uses 2 phases of identification, namely structure identification and parameter identification. Structure identification is used to determine the number of rules and membership functions for modeling the input and output variables. Parameter identification is done for the determination of shape and position of each rule[12]. The hybrid training that combines the least square approach and back propagation gradient descent method is a successful methodology[13].

The proposed method available is based on an approach used by Seema Chopra[10] for tuning rule base of the controller and input scaling factors tested successfully on a transfer function. The subtractive clustering technique is used for determining the grouping of data, which reduces the fuzzy set of rules[11]. The data given is partitioned and form a set of clusters automatically. For each & every cluster, a fuzzy IF-THEN rule base is framed. To utilize the autotuning effect and the optimized ruleset advantage completely, the proposed work Neural optimized Autotuning FLC (N-FLC) is carried out in real time spherical tank process.

This paper is organized as follows. In the present section short introduction along with literature survey is done. In section 2, the spherical tank process is explained. In section 3 conventional Fuzzy logic controller is discussed. Autotuning Fuzzy Logic Controller (AFLC) through tuning of input scaling factors is discussed in section 4. In section 5, N-AFLC is stated. In section 6 results are compared and conclusions are given to show the impact of N-AFLC on level control of real time spherical tank process.

PROCESS DESCRIPTION

Spherical tank process

A real time spherical tank process is utilized to compare the results of various Fuzzy logic based controllers. The experimental setup of spherical tank[24] with detailed specifications [Refer Table. 1] is shown in Fig. 1.

Table 1: Technical specifications of experimental setup

Part Name	Details		
Spherical tank	Material	:	Stainless Steel
	Diameter	:	50 cm
	Volume	:	102 litres
Storage tank	Material	:	Stainless Steel
	Volume	:	48 litres
Differential pressure transmitter	Type	:	Capacitance
	Range	:	2.5-250 mbar
	Span limit	:	0.65-65 kpscal
	Output	:	4-20 mA
	Make	:	ABB
Pump	Centrifugal	:	0.5 HP
Control valve	Size	:	1/4"
	Type	:	Air to open
	Input	:	3-15 psi
Rotameter	Range	:	0 - 1000 lph
Air regulator	Size	:	1/4" BSP
	Range	:	0-2.2 bar
I/P converter	Input	:	4-20 mA
	Output	:	0.2 - 1 bar
Pressure gauge	Range (G_1)	:	0-150 psi
	Range(G_2)	:	0-30 psi

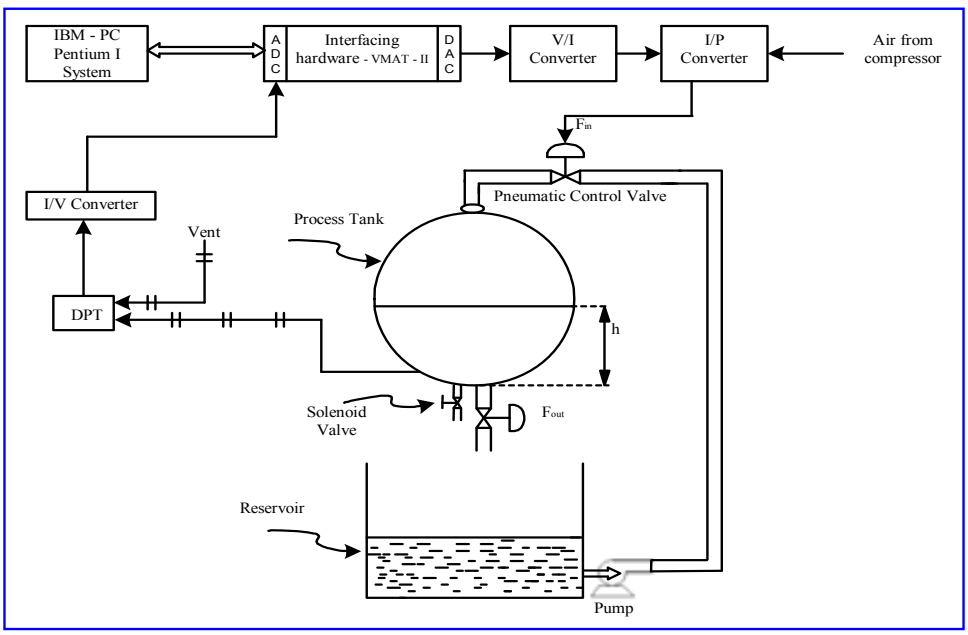

Fig. 1: Experimental setup for level control of a spherical tank

Black box modelling

The open loop parameters like process gain k, process time constant τ and time delay θ are determined from the response (level) of the process for a sudden step change in flow[25]. Fuzzy logic controller gains (G_e, G_{ce} and G_u) are associated with PI controller parameters. Using the parameters obtained from the open loop response of the process, the PI controller parameters like controller gain K_c and intergral time T_I are calculated[26].

Controller gain $K_c = \dfrac{0.9\tau}{k.\theta}$ …(1)

Integral time $T_I = 3.3\ \theta$ …(2)

The open loop model around 10% operating level and associated PI parameter values are given.

$$G(s) = \frac{4.5e^{-120s}}{440s+1}$$

$K_c = 0.73$; $T_I = 399.6s$

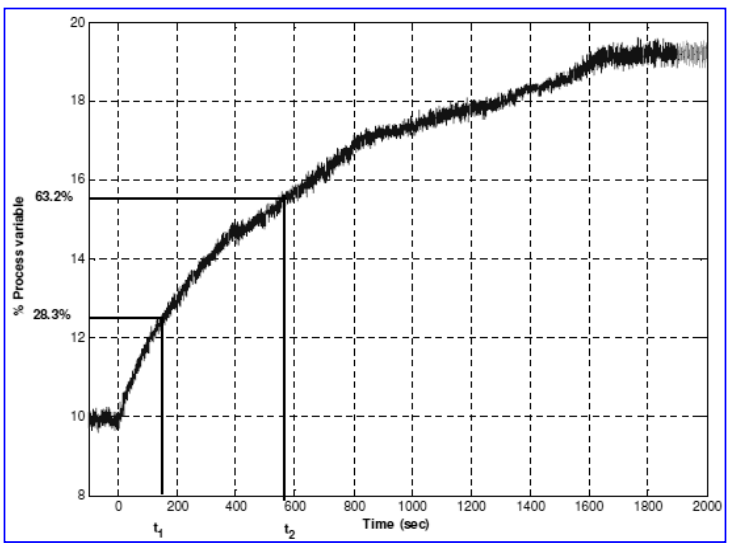

Fig. 2: Open loop response of spherical tank process around 10% operating level

CONTROLLER DESIGN

Fuzzy Logic Controller

The block diagram of level control in spherical tank process using FLC is shown in Fig. 3. FLC has two inputs as error (e) (difference between set value of level and actual level) and change in error (ce) and

change in output (cu), which represents the amount of control valve opening to maintain the constant level in the tank.

$$e(k) = y_{sp} - y(k) \quad \ldots(3)$$

$$ce(k) = e(k) - e(k-1) \quad \ldots(4)$$

$$cu(k) = u(k) - u(k-1) \quad \ldots(5)$$

The constant gain blocks called scaling factors are used to normalize inputs and outputs of FLC. [Refer Fig. 3]

$$E(k) = G_e.e(k) \quad \ldots(6)$$

$$CE(k) = G_{ce}.ce(k) \quad \ldots(7)$$

$$CU(k) = G_u.cu(k) \quad \ldots(8)$$

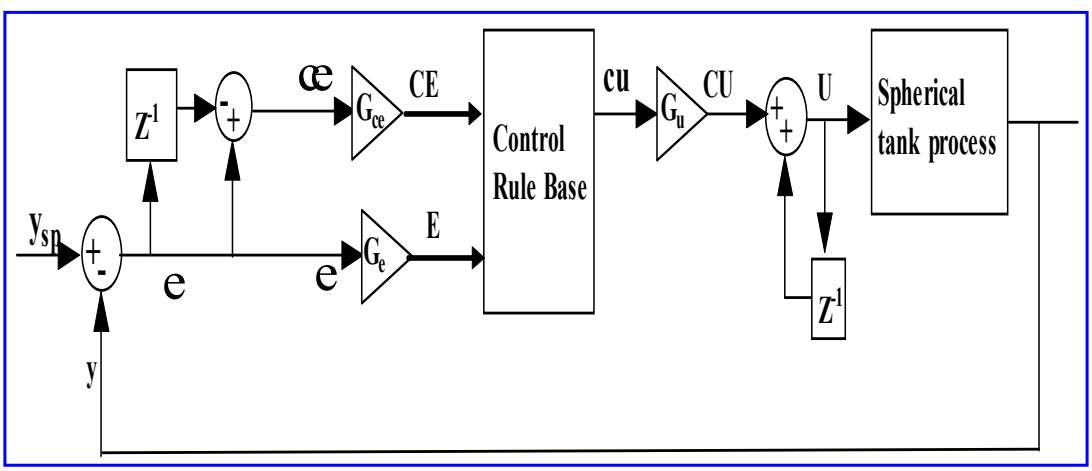

Fig. 3: Block diagram of Fuzzy Logic Controller (FLC)

The equation (9) determines the change in output of FLC in terms of PI controller parameters.

$$CU(k) = \frac{k_c MT}{T_I} D\left\{F\left(\frac{T_I M}{} ce(k) + \frac{1}{M} e(k)\right)\right\} \quad \ldots(9)$$

where, T is the sampling instant, K_c is the proportional gain, is the integral time and M is the scale factor which is greater than zero. From the equation (9), the input scaling factor G_e and G_{ce}, and the output scaling factor G_u of the controller are formed as in equation (10).

$$G_e = \frac{1}{M}; G_{ce} = \frac{T_I}{M}; G_u = \frac{K_c MT}{T_I} \quad \ldots(10)$$

FLC involves three stages of execution, namely fuzzification, rule execution and defuzzification. As the first stage, the crisp values e(k) and ce(k) are converted into fuzzy variables E(k) and CE(k) with membership function shown in Fig. 4. Each universe of discourse includes 7 fuzzy sets (NB, NM, NS, ZE, PS, PM, PB) in the range [-1 1]. Second stage of FLC execution involves processing of Fuzzy variables E and CE by an inference engine with a framed set of 7×7 rules[27] shown in Table 2.

Each rule is executed in the form:

If e is E and ce is CE then cu is cu

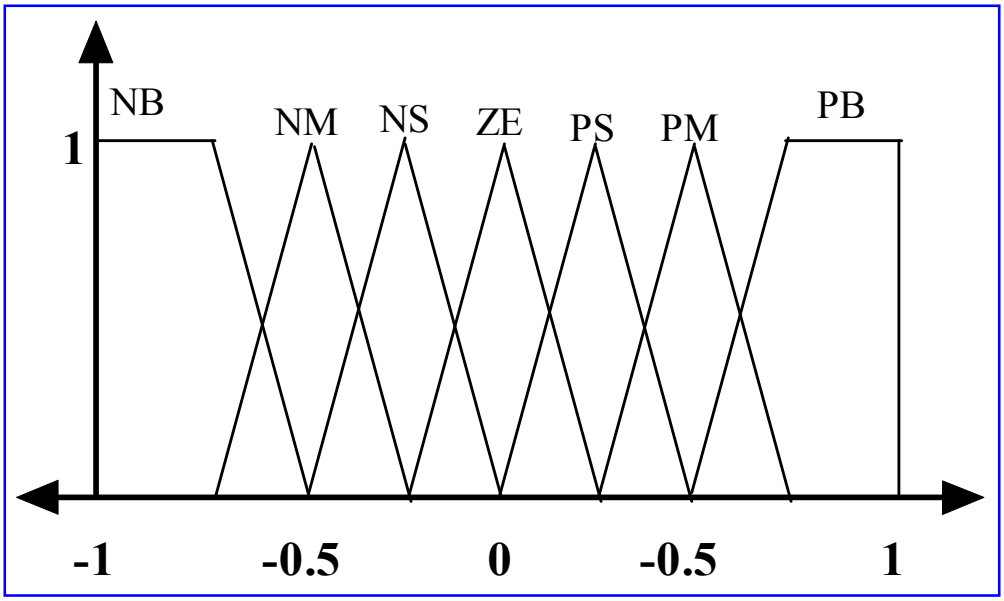

Fig. 4: Membership function of e, ce, cu

The inference engine uses max-min inference algorithm to produce the fuzzy set value. Hence the membership degree of the fuzzy set cu is the maximum of the product of E and CE membership degree. This fuzzy value is converted into crisp value in the defuzzification stage. The simplest centroid defuzzification method is used in the calculation of crisp output cu. As the universe of discourse is normalized between [-1 1], the gain factors or scaling factors G_e, G_{ce} and G_u play a vital role in FLC.

Table 2: Rule table for Δu

		e						
		NB	NM	NS	**ZE**	**PS**	**PM**	**PB**
ce	**NB**	NB	NB	NB	NM	NS	NS	ZE
	NM	NB	NM	NM	NM	NS	ZE	PS
	NS	NB	NM	NS	NS	ZE	PS	PM
	ZE	NB	NM	NS	ZE	PS	PM	PB
	PS	NM	NS	ZE	PS	PS	PM	PB
	PM	NS	ZE	PS	PM	PM	PM	PB
	PB	ZE	PS	PS	PM	PB	PB	PB

Autotuning FLC

In conventional FLC, the scaling factors are constant. Hence the change in operating conditions of the process will make the FLC not suitable to meet out the control needs. The solution to this problem is addressed by adaptation mechanism, where either input-output scaling factors or rule base of the FLC is continuously tuned[17]. In the proposed auto-tuning work, the adaptation mechanism[19-22] is applied for input scaling factors G_e and G_{ce}. The selected method is based on the work proposed by Patil *et al.*[18] for speed control of DC motor. In the proposed work [Refer Fig. 5], auto-tuning refers tuning of input scaling factors based on E and CE. Hence the normalized inputs for FLC are given as,

$$E = \alpha.G_e.e \quad \ldots(11)$$

$$CE = \beta.G_{ce}.ce \quad \ldots(12)$$

where, α and β are gain updating factors for incremental change in e and ce respectively [28]. Therefore the input gains of the FLC are not constant and updated at each sample of input e and ce.

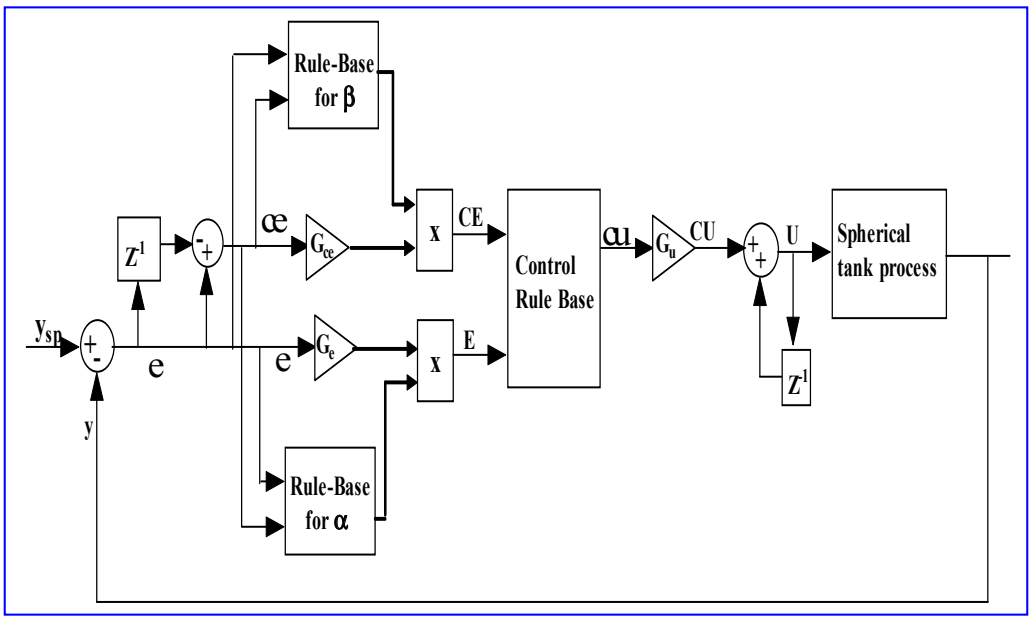

Fig. 5: Block diagram of Autotuning Fuzzy Logic Controller (FLC)

Membership Functions and Rule base

The controller of auto-tuning FLC assumes same membership functions and rule base for inputs e, ce and output cu as defined in Fig. 3 and Table 2 respectively. The membership functions for gain updating factors α fall in the range [-0.5 1.5] with two fuzzy sets big and small [Refer Fig. 6(a)]. The MFs for another gain updating factor β assume the range[15] with five singleton sets as in Fig. 6(b). The rule bases for α and β are given in Table 3 and Table 4 respectively. The rules for α and β are of the form,

If e is e and ce is ce then α *is* α

If e is e and ce is ce then β *is* β

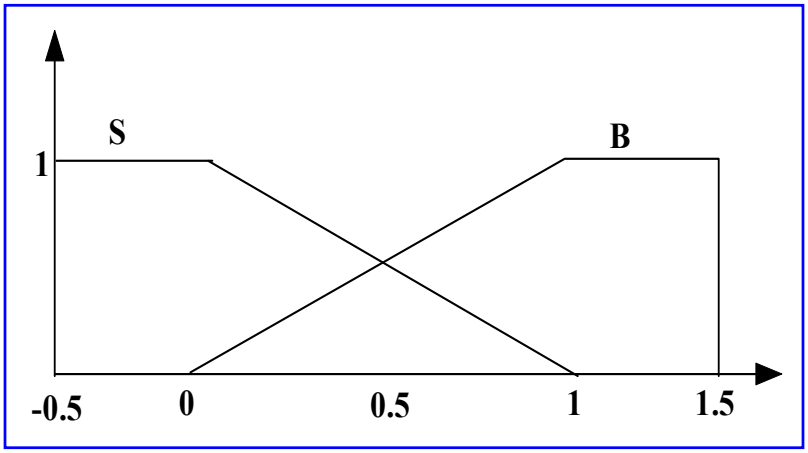

Fig. 6(a): Membership function of α

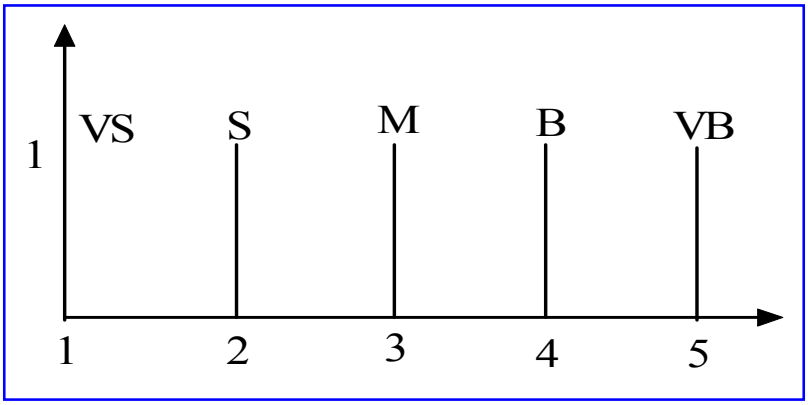

Fig. 6(b): Membership function of β

Table 3: Rule table for α

		e						
		NB	NM	NS	ZE	PS	PM	PB
ce	NB	B	B	B	B	B	B	B
	NM	S	S	B	B	B	S	S
	NS	S	S	B	B	B	S	S
	ZE	S	S	S	B	S	S	S
	PS	S	S	B	B	B	S	S
	PM	S	B	B	B	B	B	S
	PB	B	B	B	B	B	B	B

Table 4: Rule table for β

		e						
		NB	NM	NS	ZE	PS	PM	PB
ce	NB	S	S	S	S	S	S	S
	NM	M	M	S	S	S	M	M
	NS	B	M	M	S	M	M	B
	ZE	VB	B	M	M	M	B	VB
	PS	B	M	M	S	M	M	B
	PM	M	M	S	S	S	M	M
	PB	S	S	S	S	S	S	S

Neural Optimized Autotuning FLC Rule extraction and optimization

Changes in the operating conditions of the process cannot be predicted and taken into account by a conventional FLC. Hence, autotuning method based on updating input scaling factors is desired to adapt variations in the process dynamics. As autotuning concept needs additional fuzzy reasoning blocks for generating α and β, the computational time and memory used are comparatively higher. To overcome this limitation, controller rule base tends to be utilized with less number of effective rules[12, 8, 9] by clustering cum optimization approaches. Fuzzy subtractive clustering is used in the work to reduce the number of rules followed by optimization of rules using back propagation algorithm.

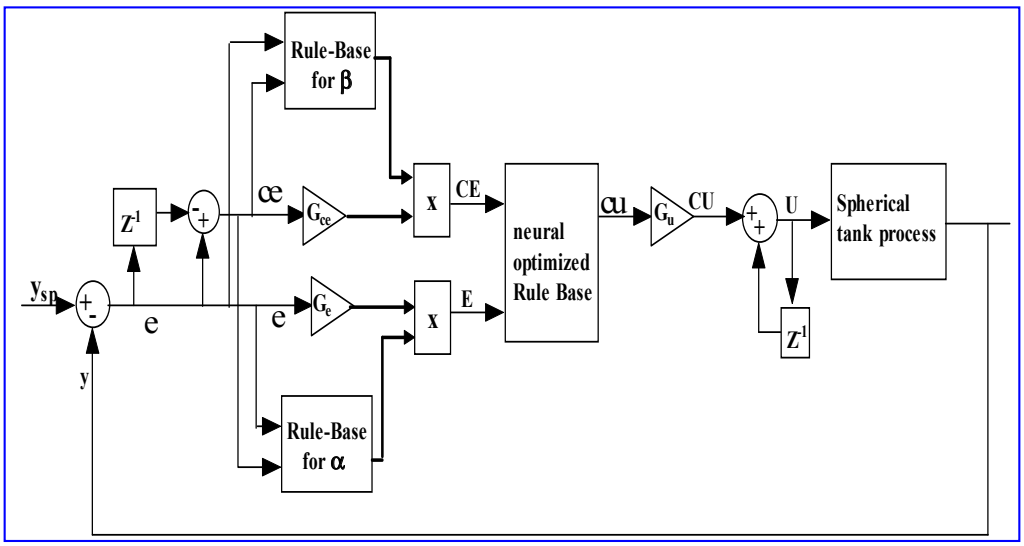

Fig. 7: Block diagram of Neural optimized Autotuning Fuzzy Logic Controller (N-AFLC)

Neuro-fuzzy learning

This Neuro fuzzy learning scheme combines the advantage of neural networks and fuzzy theory to design a hybrid model that uses a fuzzy theory to represent knowledge in an interpretable manner and the learning ability of a neural network to optimize its parameters.

First, the structure of the network is identified, followed by parameter adjustment. To initiate training, input-output data of order nx1 are provided. To extract the set of initial fuzzy rules, data is separated into various groups with respective classes. Subtractive clustering is utilized to identify cluster centre and each cluster centre may be translated into a fuzzy rule for identifying the class. The data point with the highest potential is considered as the first cluster centre x_1^* and its potential value is considered as P_1^*. The potential of each data point x_1 is revised by the equation (13).

$$P_{i+1} \Leftarrow P_i - P_1^* e^{-\lambda \left\| x_i - x_1^* \right\|_2} \quad \ldots(13)$$

where, $\lambda = \dfrac{4}{r_b^2}$ and $r_b = 1.25\ r_a$.

The data with the highest remaining potential is selected as a second cluster centre x_2^* and its potential value is considered as P_2^*. The process is then continued to find P_k^* based on the following criteria to make k clusters of fuzzy rules:

if $P_k^* > \bar{\varepsilon}\, P_1^*$

Accept x_k^* as a cluster centre and continue.

else $P_k^* < \underline{\varepsilon}\, P_1^*$

Reject x_k^* and end the clustering process.

else

Find $d_{\min}$ as the shortest of the distances between x_k^* and all previously found cluster centres.

if $\dfrac{d_{\min}}{r_a} + \dfrac{P_k^*}{P_1^*} \geq 1$

Accept x_k^* as a cluster and continue.

else

Reject x_k^* and set the potential at x_k^* to select the data point with the next highest potential as new x_k^* and re-test.

end

end

end

Here $\bar{\varepsilon} = 0.5$ specifies the threshold for the potential above which the data point is accepted as a cluster centre and $\bar{\varepsilon} = 0.15$ specifies the threshold for the potential under which the data point is rejected as a cluster centre. Let X_j be the jth input feature and μ_{ij} be the Gaussian membership function in the ith rule associated with the jth input feature. The membership function μ_{ij} is given in equation (14).

$$\mu_{ij} = e\left[-\left(\frac{x_j - m_{ij}}{\sigma_{ij}}\right)^2\right] \quad \ldots(14)$$

where m_{ij} and σ_{ij} be the mean and standard deviation.

The parameters associated with the membership functions change through the learning process. The computation of these parameters (or their adjustment) is facilitated by a gradient vector. This gradient vector provides a measure of how well the fuzzy inference system is equalizing the input/output data for a given set of parameters. The learning process is achieved in three layers.

Layer 1:

Each unit in this layer receives input from the corresponding input variable. Nodes are arranged as j groups and output of this layer $O_{ij}^{(1)}$ is calculated using the corresponding Gaussian membership function.

$$O_{ij}^{(1)} = \mu_{ij}\left(X_j\right) \quad \ldots(15)$$

Layer 2:

Each node of this layer represents a fuzzy rule. The output of this layer $O_j^{(2)}$ represents the product of all its input from layer 1.

$$O_j^{(2)} = \prod_{i=1}^{n} O_{ij}^{(1)} \quad \ldots(16)$$

Layer 3:

This single node layer represents the output computed by a centroid defuzzification method.

$$O^{(3)} = \frac{\sum_{j=1}^{j} O_j^{(2)} k_j}{\sum_{j=1}^{j} O_j^{(2)}} \quad \ldots(17)$$

Here K_j is the number of clusters or fuzzy singleton sets on the output side. The hybrid learning algorithm uses a 2 stroke learning cycle as forward stroke and backward stroke.

In forward stroke, with fixed premise parameters the least squared error estimate approach is employed to update the consequent parameters and to pass the errors to the backward stoke. In a backward stroke of learning, the consequent parameters are fixed and the steepest descent method is applied to update the premise parameters. The premise and consequent parameters will be identified for membership functions and fuzzy inference system by repeating the forward and backward strokes.

Rule extraction and optimization

For the construction of fuzzy rules to determine the change in output Δu, 441 data points $\{x_1, x_2, \ldots, x_n\}$ are collected in the form of vector of e, Δe and Δu at the step size of 0.1 from the existing rule base of 49 rules. Then subtractive clustering is applied to group the equivalent and that in turn reduces the number of rules. Here, subtractive clustering results in 9 clusters or rules. The resulting 9 rules need optimization to achieve the required performance.

A hybrid learning algorithm which combines least square estimation and back propagation gradient descent method is used to modify the membership parameters of 9 clusters obtained by subtractive clustering to minimize the output error measure. The training process stops whenever the maximum epoch number is reached or the training error goal is achieved. The membership functions before and after optimization for e and ce are shown in Fig. 8(a) & 8(b) Change in error variable shows considerable change in its shape of membership function after optimization as observed in Fig. 8(c) and Fig. 8(d). The 'i' number of rules is framed as

Rule i: If e is mf_i & ce is mf_i, then c_i, where i = 1 to 9.

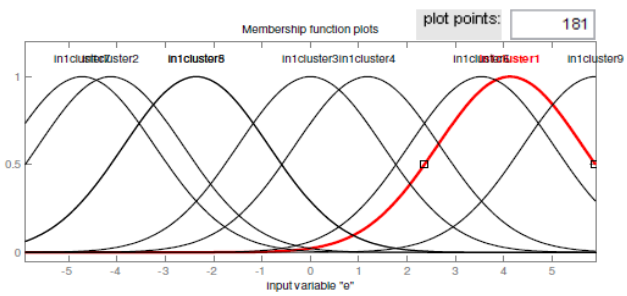

Fig. 8(a): Membership function-error (e) before optimization

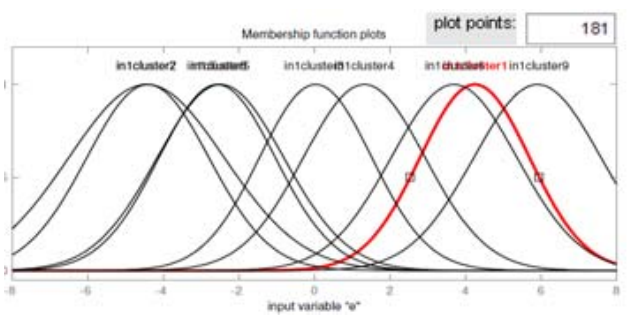

Fig. 8(b): Membership function -error (e) after optimization

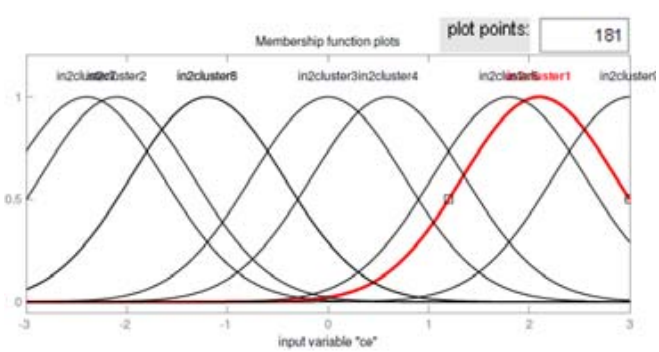

Fig. 8(c): Membership function-change in error (ce) before optimization

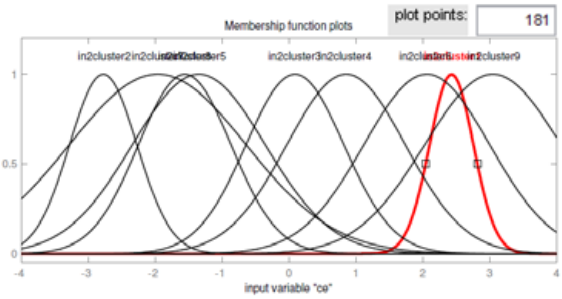

Fig. 8(d): Membership function -change in error (ce) after optimization

RESULTS AND DISCUSSION

The proposed controller (N-AFLC) has been implemented on the chosen real time spherical tank process [Refer Fig.9]. As a proof for setpoint tracking and load rejection capability, the proposed controller is compared with conventional FLC and AFLC implemented in the same process. Fig. 10(a) gives the servo response of three controllers with the respective controller performance in Fig. 10(b). The servo response visualizes the tracking capability of N-AFLC is faster than other two controllers (FLC and AFLC). To observe the disturbance rejection capability of the proposed controller, a momentary opening of solenoid valve at the bottom of the spherical tank (Refer Fig. 10(a)) has been done. In spite of this disturbance, all the three controllers are able to bring the level back to the previous position of 16% as in Fig. 11(a). The respective controller variation is shown in Fig. 11(b). Comparing the three controllers N-AFLC rejects the disturbance faster than the other two methods (FLC and AFLC) mainly because of the usage of optimized rule set in controller rule base and auto updated input scaling factors. The performance indices for load disturbance rejection are given in Table. 5. It implies that N-AFLC shows good improvement in rise time, settling time and ISE.

Table 5: Performance Indices

Controller type	Rise time (s)	Settling time (s)	ISE	
			Servo	Regulatory
FLC	250	5500	1.3317E+05	2.0688E+05
AFLC	250	2200	9.7531E+04	1.3676E+05
N-AFLC	20	500	7.0745E+04	9.5854E+04

Fig. 9: Photograph of spherical tank process

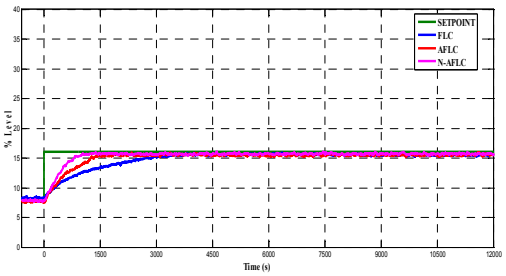

Fig. 10(a): Servo response of spherical tank process

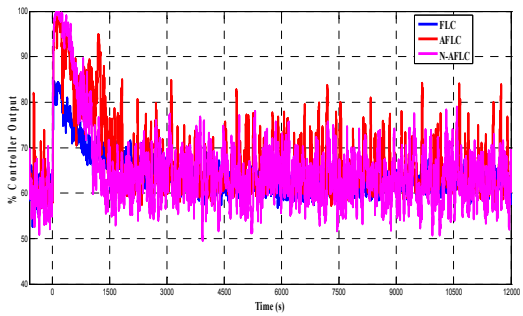

Fig. 10(b): Controller output (Servo) of spherical tank process

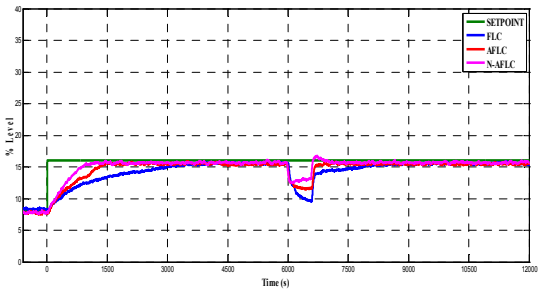

Fig. 11(a): Servo regulatory response of spherical tank process

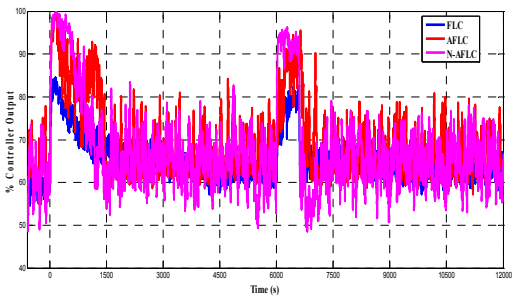

Fig. 11(b): Controller Output (Servo regulatory) of spherical tank process

CONCLUSION

In the paper, autotuning of input scaling factors to adapt the FLC under varying operating conditions in a void of compromising the memory requirement with a controller rule base of the minimal optimized rule set has been presented. The method has proved to be very efficient in both setpoint tracking and disturbance rejection, when implemented in real time spherical tank process. The results have proven that N-FLC gives better performance than conventional FLC and AFLC.

REFERENCES

1. Vinodha, R., Natarajan, M. and Prakash, J. 2008. "Simple Adaptive Control Schemes for a Spherical Tank Process", *Journal of System Science and Engineering, A Publication of Systems Society of India,* **17**(2): 55-60, 2008.

2. Bennett, S. 1993. "Development of the PID Controller," *IEEE Contr. Syst. Mag.,* **l**(13): 58–65.

3. Chen, G. 1996. "Conventional and Fuzzy PID Controllers: an overview", *International Journal of Intelligent and Control Systems,* **1**: 235-246.

4. Lee, C.C. 1990. "Fuzzy Logic in Control Systems: Fuzzy Logic Controller – Part I," *IEEE Trans. Syst. Man Cybern.,* **20**(2): 404-418.

5. Lee, C.C. 1990. "Fuzzy Logic in Control Systems: Fuzzy Logic Controller – Part II," *IEEE Trans. Syst. Man Cybern.,* **20**(2): 419-435.

6. Hu, B.G., Mann, G.K.I. and Gosine, R.G. 2001. "A Systematic Study of Fuzzy PID Controllers-Function-Based Evaluation Approach", *IEEE Transactions on Fuzzy Systems,* **9**(5): 699 -712.

7. Chaohua Ao and Jianchao Bi, 2012. "Parameter Auto-tuning Method Based on Self learning Algorithm", *Journal of Computers,* **7**(9).

8. Bassi, S.J., Mishra, M.K. and Omizegba, E.E. 2011. "Automatic Tuning of Proportional–Integral–Derivative (PID) Controller using Particle Swarm Optimization (PSO) Algorithm", *International Journal of Artificial Intelligence & Applications,* **2**(4).

9. Jin-Sung Kim, Jin-Hwan Kim, Ji-Mo Park, Sung-Man Park, Won-Yong Choe and Hoon Heo, 2008. "Auto Tuning PID Controller based on Improved Genetic Algorithm for Reverse Osmosis Plant", International *Journal of Computer, Electrical, Automation, Control and Information Engineering,* **2**(11): 3707-3712.

10. Srivignesh, N., Sowmya, P., Ramkumar, K. and Balasubramanian, G. 2012. "Design of Neural Based PID Controller for Nonlinear Process", *International Conference on Modeling Optimization and Computing, Procedia Engineering,* **38**: 3283-3291.

11. Yang, J.R. 1993. "ANFIS: Adaptive-Network-Based Fuzzy Inference System", *IEEE Trans. on Systems, Man, and Cybernetic,* **23**(3): 665-685.

12. Seema Chopra, R. Mitra, and Vijay Kumar, 2006. "Analysis of Fuzzy PI and PD Type Controllers using Subtractive Clustering", *International Journal of Computational Cognition,* **4920**: 30-34.

13. Dhanalakshmi, R., Vinodha, R. and Abraham Lincoln, S. 2014. "A Real-time Performance Evaluation of the Fuzzy PI Controller using Subtractive Clustering for Spherical Tank Process", *International Journal of Applied Engineering Research,* **9(**26): 9151-9155.

14. Boo.Poonguzhali and Vinodha, R. 2014. "Implementation of Anti-Reset Windup Scheme in PI Controller for Spherical Tank Process", *International Journal of Modelling and Simulation,* **34(**4): 1-15.

15. Sundaresan, K.R. and Krishnaswamy, R.R. 1978. "Estimation of Time Delay, Time Constant Parameters in Time, Frequency and Laplace Domains", *Canadian Journal of Chemical Engineering,* **56**(2): 257-262.

16. Ziegler, G. and Nichols, N.B. 1942. "Optimum Settings for Automatic Controllers", *Transcation ASME,* **64(**11): 759-768.

17. Takagi, T. and Sugeno, M. 1985. "Fuzzy Identification of Systems and its Application to Modelling and Control", *IEEE Trans. on Systems, Man, and Cybernetics,* **15**: 116-132.

18. Patil, N.J., Chile, R.H. and Waghmare, L.M. 2010. "Fuzzy Adaptive Controllers for Speed Control of PMSM Drive", *International Journal of Computer Applications,* **1**: 84-91.

19. Farhad Aslam, Mohd. Zeeshan Haider, 2011. "An Implementation and Comparative Analysis of PID Controller and their Auto Tuning Method for Three Tank Liquid Level Control", *International Journal of Computer Applications,* **21**(8): 0975-8887.

20. Teng, T.K., Shieh, J.S. and Chen, C.S. 2003. "Genetic Algorithms Applied in Online Autotuning PID Parameters of a Liquid-level Control System", *Transaction of the Institute of Measurement and Control,* **25**(5): 433-450.

21. Baljinder Singh and Vijay Kumar, 2015. "Design and Simulation of Auto Tuning of PID Controller using MRAC Technique for Coupled Tanks System", *International Journal of Science and Research,* **4**(5): 3037-3040.

22. Rajani K. Mudi, Chanchal Dey and Tsu – Tian Lee, 2008. "An Improved Auto – tuning Scheme for PI Controllers", *Journal of science Direct ISA Transactions,* **47**: 45-52.

23. Hang, C.C., Astrom, K.J. and Ho, W.K. 1991. "Refinements of Ziegler-Nichols tuning formula", *IEEE Proceedings – D,* **138**(2): 111-118.

24. Panda, S.K., Lim, J.M.S., Dash, P.K. and Lock, K.S. 1997. "Gain Scheduled PI Speed Controller for PMSM Drive", *Proceedings IEEE Industrial Electronics Society International Conference,* **2**: 925–30.

25. Mamdani, E.H. and Assilian, S. 1975. "An experiment in Linguistic Synthesis with a Fuzzy Logic Controller", *International Journal Man-Mach. Stud.,* **7**(1).

26. Chiu, S.L. 1994. "Fuzzy Model Identification Based on Cluster Estimation", *Journal of Intelligent and Fuzzy System,* **2**: 267-278.

27. Phaneendar Kumar, G.V., Srinu, I. and Satyanarayana, K. 2014. "Design and Performance Analysis of Ann Based Hybrid Pid Controller for Liquid Level Control System", *International Journal of Scientific Research,* **3**(9): 160-165.

28. Shahgholian, G. and Movahedi, A. 2011. "Modeling and Controller Design using ANFIS Method for Non-Linear Liquid Level System", *International Journal of Information and Electronics Engineering,* **1**(3): 271-275.