PID and Intelligent Controllers for Optimal Timing Performances of Industrial Actuators

Danilo Pelusi
Dipartimento di Scienze della Comunicazione
University of Teramo
Teramo, Italy
dpelusi@unite.it

***Abstract* — PID controllers are widely used in process industries due to its simplicity and robustness. The main problem sometime is tuning the PID parameters in order to improve the settling time, the rise time and the overshoot. In literature, there are procedures to obtain the PID settings which gives the better performance and robustness. Some experiments on this research line show that the controller gain is only a function of the overshoot observed in the setpoint experiment. The challenge is to improve the timing parameters to achieve optimal control performances. Remarkable findings are obtained through the use of Artificial Intelligence techniques as Fuzzy Logic, Genetic Algorithms and Neural Network. The first theory is good for decisional problems, the second one can be used in search algorithms and the Neural Networks have the capability to learn from data. The combination of these approaches can give good results in terms of settling time, rise time and overshoot. In this paper, we propose the design of suitable controllers which target is the improvement of timing performance of industrial actuators. The designed controllers are PID controller, genetic-fuzzy controller and neuro-fuzzy controller. The results show that the PID controller has good overshoot values and shows optimal robustness. The genetic-fuzzy controller gives a good value of settling time and a very good overshoot value. The neural-fuzzy controller gives the best timing parameters improving the control performances of the others two approaches.**

Keywords - PID controllers, fuzzy logic, Genetic Algorithms, control systems, Neural Networks

I. INTRODUCTION

In industrial applications, it is very important designing controllers able to improve the performances of electro-hydraulic actuator. The range of applications for electro-hydraulic actuators includes materials test machines, robotics, flight simulation and manufacturing systems. Such applications need of good performances of the electro-hydraulic actuators on position and pressure. To improve the performances of industrial actuators, some suitable controllers are required.

The goodness of control systems depends on timing parameters such as settling time, rise time and overshoot. If these parameters are suitably small, the system gives a good control performance. The main challenge is to reduce the timing values. In this way, some problems as undesirable overshoot, longer settling times and vibrations during the switching from a state to another one, can be avoided. In literature, there are different approaches to solve these kinds of problem. A classical approach is the use of Proportional Integral Derivative (PID) controllers. Such controllers are based on a control method extensively used in industrial process control application. Vaishnav and Khan [24] designed a PID controller using Ziegler-Nichols technique for higher order systems. A simple method for PID controller tuning of unidentified process using closed-loop experiments has been developed [22]. Such method requires one closed-loop step setpoint response experiment similar to the classical Ziegler-Nichols experiment. However, in complex systems characterized by nonlinearity, large delay and time-variance, the PID's are of no effect [11]. The design of a PID controller is generally based on the assumption of exact knowledge about the system. Sometime such knowledge is not available for the majority of the systems. In order to overcome these limits, many advanced control methods have been introduced. Such methods make use of the fuzzy logic to control the considered system. The application of such control techniques simplifies the control designing for difficult system models. Kumar and Garg [14] designed a fuzzy controller to control a single link manipulator robot. A gain tuning fuzzy controller has been designed to monitor the track seeking in optical disks [23] . In order to improve the control precision of a ball mill circuit, a fuzzy interpolation algorithm is presented [11]. Moreover, PID fuzzy controllers can be designed as power system stabilizer [7].

The design of a traditional fuzzy controller depends on input-output membership functions number and input-output membership functions shape. Moreover, it is very difficult to examine all the input-output data from a complex system to find the optimal membership function for a fuzzy system. A time-consuming adjusting process is required to achieve the membership functions which improve the control system performances. Because a natural choice through trial and error procedure is impossible to obtain, Genetic Algorithms (GA) are applied to fuzzy controllers with good results [13], [14], [5], [17]. Genetic Algorithms are useful approaches to

DOI 10.5013/IJSSST.a.13.02.07

ISSN: 1473-804x online, 1473-8031 print

problems requiring effective and efficient searching. In [13] the membership functions and the fuzzy logic rules were optimized by the genetic algorithms technique for a temperature control system. Moreover, with the application of GA, Chegeni et al. [5] eliminated the limitation on symmetric membership functions and symmetric rules base. Another approach to achieve a good control system consists of neuro-fuzzy techniques. These two theories combine the capability to model a problem domain using a linguistic model instead of complex mathematical models and the capability to learn from data. In other words, a neural-fuzzy network can self-adjust the parameters of the fuzzy rules using neural-network-based learning algorithms. A datadriven adaptive neuro-fuzzy controller has been designed for the water-level control of U-tube steam generators in nuclear power plants [16]. In [1] a neuro-fuzzy controller has been designed to control the DC motor speed.

In this paper, we attempt to achieve an optimal control performance of industrial actuators designing suitable controllers. We consider three different approaches. The first one regards the design of a PID controller based on Ziegler-Nichols tuning formula [9] . The relative rules for tuning PID controllers have been very influential. Ziegler and Nichols presented two methods: the step response method and the frequency response method. The first method is presented in this work. The second approach attempts to improve the PID results designing a fuzzy controller optimized through GA techniques. The design of a suitable neuro-fuzzy controller which improves the performances of genetic-fuzzy controller is the third approach of our model. The target of these different approaches is also that one of improving the simulation results of [13]. Recent studies [18], [19], [20] propose genetic-neuro-fuzzy techniques able to improve the timing performances of second order control systems.

II. TUNING PARAMETERS OF PID CONTROLLER

The PID controllers have a wide range of applications in industrial control because of their simple control structure. An important feature of PID controllers is that they need of less plant information than a complete mathematical model. Thus the controller parameters can be adjusted with a minimum of effort. Moreover, one survey of Desborough and Miller [8] indicates that more than 97% of regulatory controllers utilize the PID algorithm.

There are may versions of a PID controller. In this paper, we consider a controller described by

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \tag{1}$$

where u(t) is the input signal sent to the plant model, e(t)=r(t)-y(t) the error, y(t) the output and r(t) is the reference input signal. The parameters Kp, Ti and Td are the tuning parameters. There are more ways to obtain the tuning values of Kp, Ti and Td: our PID controller uses the Ziegler-Nichols tuning formula. The tuning formula is obtained when the plant model is given by a first-order plus dead time which can be expressed by

$$G(s) = \frac{k}{1+sT} e^{-sL} \tag{2}$$

In real-time process control systems, a large variety of plants can be approximately modeled by (2). If the system model cannot be physically derived, experiments can be performed to extract the parameters for the approximate model (2). For instance, if the step response of the plant model can be measured through an experiment, the output signal can be recorded and the parameters of k, L, and T (or a, where a = kL/T) can be extracted [9] .

The proposed PID controller is designed to control some second order control systems. In order to verify the robustness of the model, we consider three systems with different transfer function. The first one is typically used to approximate the working of DC motors [13] and has the form:

$$G(s) = \frac{2}{s^2 + 12s + 24} \tag{3}$$

The second transfer function (defined in (4)) is used for processes with first order dynamics with time delay [2].

$$G(s) = \frac{1}{(1+s)(1+5s)} \tag{4}$$

The third transfer function choice (see equation (5)) is joined to the attempts of many researches of improving the tuning parameters through intelligent techniques [15]

$$G(s) = \frac{400}{s^2 + 50s} \tag{5}$$

The Fig. 1 shows the block diagram of PID controller. The difference between the step and the output feedback is passed as input into PID controller block. The PID controller block contains some MATLAB functions which implement the Ziegler-Nichols tuning formulas [9] . The output of the PID controller block serves as an input to the transfer function block. We consider the PID controller behavioral for different plants defined by (3), (4) and (5) whereas the genetic-fuzzy and neuro-fuzzy controllers are designed only for second order systems with transfer function (3).

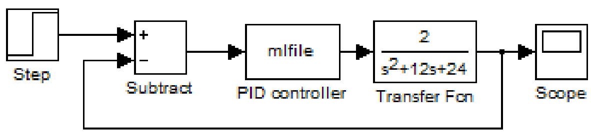

Figure 1. PID controller blocks diagram

DOI 10.5013/IJSSST.a.13.02.07

ISSN: 1473-804x online, 1473-8031 print

III. DESIGN OF GENETIC-FUZZY CONTROLLER

In order to improve the timing performances of designed PID controller, suitable genetic procedures are used to design the fuzzy logic controller. To do this, we use the Genetic Algorithms techniques to optimize the control strategy.

The first step to design the fuzzy controller is the choice of the number of input/output membership functions. Assuming all possible rules are used (which is often the case), if the inputs to the fuzzy controller are increased with the membership functions for each input, then the number of rules increases exponentially. It is necessary to avoid this situation because it is very important to try to minimize the time to compute the fuzzy controller outputs given some inputs. Some studies [6] deal with the design of fuzzy logic controllers with less number of rules leading to a smaller amount of computational time. The fuzzy controller has two input: the error e, that is the difference between the reference value and the output of controller and the change in error de, that is the difference between the error at time t and that one at $t - 1$. For these inputs, we choose seven membership functions: NB (Negative Big), NM (Negative Medium), NS (Negative Small), ZE (Zero Error), PS (Positive Small), PM (Positive Medium), PB (Positive Big). The fuzzy output has the same membership functions of fuzzy inputs. Analyzing the findings of [6], we define the rules of Table I. The rules have the following structure: if the error e is NB and the change in error de is NB, than the output of controller is NB (as an example). During the rules designing process, we have discovered that increasing the rules and number of membership functions beyond 49 rules is a futile procedure. In fact, this procedure increases the complexity of fuzzy logic controller and has no effect on output response of the system.

The Fig. 2 shows the block diagram of fuzzy controller. The difference between the step and the output feedback is passed as an input into fuzzy logic controller together to the change in error. The output of the fuzzy logic controller serves as an input to the transfer function block. Once chosen the shapes of membership functions, it needs to tune the parameters which define the slope of the curves. We choose trapezoidal and triangular membership functions and optimize their definition parameters through a search algorithm based on Genetic Algorithms. This technique assures that at least a good local optimum can be discovered. Genetic Algorithms are based on the survival principle of the fittest. Therefore, we must establish a fitness function which provides a performance measure of tuning parameters. Such function can be expressed as

$$f(x) = \exp(-x) \tag{6}$$

where

$$x = \sum_{i=1}^{n} e^2(i) \tag{7}$$

and n is the number of iterations during simulation. In this way, the error e is reduced at minimum.

The variables to optimize are four for the first and seventh membership function (trapezoidal functions) and three for the others five membership functions (triangular functions). Because there are two inputs and one output with seven membership functions, the number of variables to optimize is 69.

The optimization algorithm works as follows.

Step 1. Initialize the variables to optimize.

Step 2. Compute randomly the slope parameters and establish the termination criteria.

Step 3. When it is achieved the termination criteria, the genetic procedure is stopped and go to Step 6.

Step 4. Implement the genetic operations as crossover, mutation and selection [12].

Step 5. Repeat the steps 3-4.

Step 6. Print the optimal values of slope parameters. After 20 generations, the optimal fuzzy sets of Fig. 3, 4 and 5 are obtained. The optimized fuzzy controller uses the Mamdani inference method and the centroid defuzzification technique. Moreover, MATLAB fuzzy logic toolbox is used to design the fuzzy inference system.

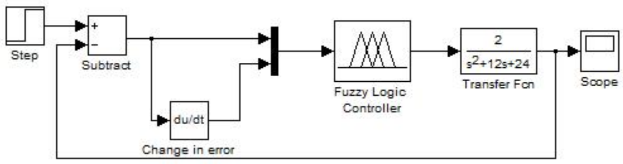

Figure 2. Fuzzy controller blocks diagram

TABLE I. FUZZY RULES

de\e	NB	NM	NS	ZE	PS	PM	PB
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

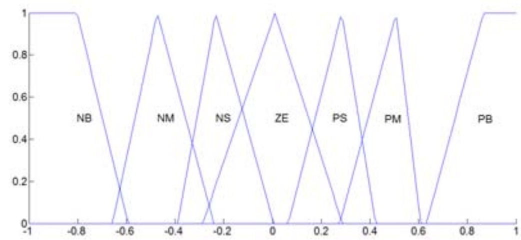

Figure 3. Optimized membership functions of input e

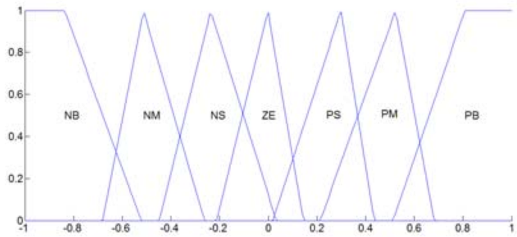

Figure 4. Optimized membership functions of input de

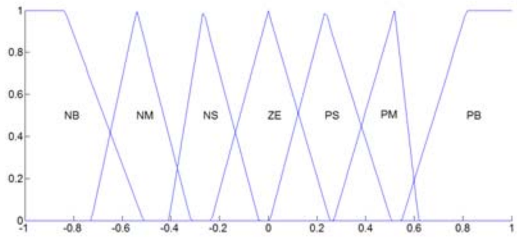

Figure 5. Optimized membership functions of output

IV. NEURAL NETWORKS APPLICATIONS

The genetic-fuzzy controller is characterized by optimal membership functions which give good control system performances. A further improvement can be searched on the tuning of rules base, in particular on the weights of fuzzy rules. In order to tune these parameters, we consider a datadriven intelligent controller, that is an adaptive neuro-fuzzy controller. In fact, a neural-fuzzy network can self-adjust the parameters of the fuzzy rules using neural-based learning algorithms. Our idea is to tune the rules weights giving more relevance to the rules that give good timing performances. We tune the rules weights of optimized genetic-fuzzy controller with the constraint to achieve small values of settling and rise time. Our control system has self-tuning capabilities and requires an initial rule base (see Table I) to be specified prior to training.

The design of neural networks for specific applications is often a test and error process. This process sometime depends mainly on previous experience in similar applications. Moreover, the performances and the cost of a neural network are joined to the choice of the neurons number, net architecture and learning algorithms. Some works [10] , [4] are focused in the development of methods for the evolutionary design of architectures to search optimal configurations of neural networks.

A powerful training technique which can be applied to networks is the back-propagation algorithm. This training procedure is used in many applications [3], [21] . Backpropagation involves minimization of an error function which is accomplished by performing gradient descent search on the error surface. The term back-propagation refers to the process by which derivatives of network error, with respect to network weights and biases, can be computed.

Because there are not fix rules to establish the network designing parameters, we use trial and error procedures to define the layers number and the neurons number for each layer. The designed neural network has three layers: the first one has 2 neurons (equal to inputs number), the hidden layer has 7 neurons and the output layer has 49 neurons (see Fig. 6). Moreover, as training technique, we choice the backpropagation algorithm. MATLAB Neural Network Toolbox is used in order to construct the neural network.

Once defined the network architecture, the next step is to define a suitable training set. The training sample of our neural network is characterized by the two inputs e and de and 49 rules weights values. The training data are obtained as follows. The error and change in error are randomly generated and sent to the genetic-fuzzy controller. The weights values where settling and rise time are less than the best timing values of genetic-fuzzy controller are extracted. In others terms, we impose

$$t_s < t_{s_{best}} \tag{8}$$

and

$$t_r < t_{r_{best}} \tag{9}$$

where t_s is the settling time of neural-fuzzy controller and t_{sbest} is the best settling time of genetic-fuzzy controller, whereas t_r is the rise time of neural-fuzzy controller and t_{rbest} is the best rise time of genetic-fuzzy controller. The obtained training sample of 1050 patterns is applied to the neural network.

One epoch of training is defined as a single presentation of all input vectors to the network. The network is then updated according to the results of all those presentations. Training occurs until a maximum number of epochs occurs or the performance goal is met. After 150 epochs and with a goal of 0.05, the performance of neural network is 0.0820087 (see Fig. 8).

Once that the neural network has been trained, the neuro-fuzzy controller can be designed. The block diagram is showed in Fig. 7. The inputs e and de are sent to the neural network which gives the optimal weights for the 49 fuzzy rules. Such tuning parameters are passed to the fuzzy controller together with the error signal e and the change in error de. The output of fuzzy controller tunes the second order plant. The difference between the signal reference and the output feedback is passed as an input to the neural network and fuzzy controller. Subsequently, the process restarts with the calculation of new values of the error and change in error.

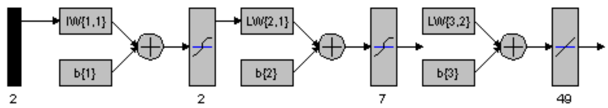

Figure 6. Neural Network architecture

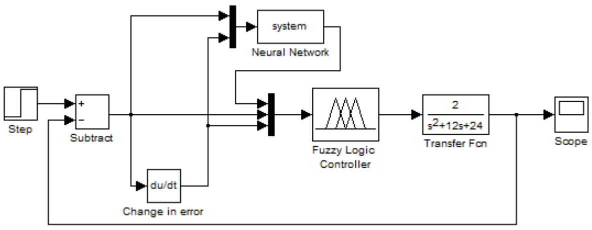

Figure 7. Neuro-fuzzy controller blocks diagram

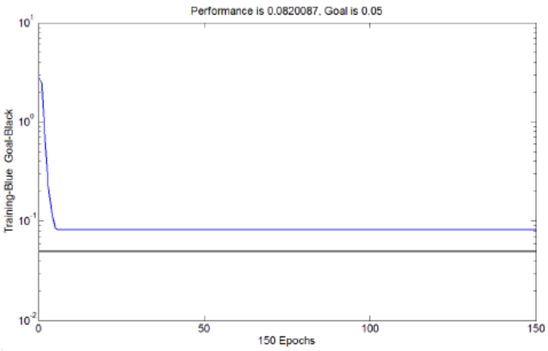

Figure 8. Training results of Neural Network

V. SIMULATION RESULTS

The designed controllers are simulated through the MATLAB. The simulation results of our controllers are showed in Table II. Our PID controller improves the rise time and the overshoot of [13]. In fact, the rise time of conventional PID controller in [13] is 0.371s, whereas the PID controller here designed has rise time equal to 0.118s. Moreover, the overshoot of [13] is 0.6748, whereas our PID controller gives a value of 0.223. The step response of PID controller with plant defined by (3) is showed in Fig. 9a. Moreover, the controller has shown good robustness performances changing the plant parameters. In Table II are presented the settling time and rise time of control system for the three different plants. The step response are shown in Fig. 9a, 9b, 9c.

The genetic-fuzzy controller has better ts and tr than fuzzy logic PD controller [13]. In our work, the settling time is 0.699s versus a value of 0.8735s of fuzzy logic PD controller designed in [13]. Comparing the results, we can note that the rise time has an improvement of above 54 percent respect to [13]. The genetic-fuzzy controller also gives a settling time better than our PID controller and zero overshoot (see Table III). The improvements can be deduced observing PID and genetic-fuzzy controllers step response (see Fig. 9 and 10).

Using the constraints (8) and (9) in the neuro-fuzzy controller with t_{sbest} = 0.699s and t_{rbest} = 0.385s, the genetic-fuzzy controller results are improved. In fact, the adaptive neuro-fuzzy approach gives a settling time of 0.423s and a rise time of 0.234s. The step response of neural-fuzzy controller is shown in Fig. 11.

Finally, by comparing the results of control systems performances, it can be noted that the neuro-fuzzy controller produces a more desirable performance when compared with PID and genetic-fuzzy controllers.

TABLE II. SETTLING TIME AND RISE TIME OF PID CONTROLLER FOR DIFFERENT PLANTS

	$G(s)=\frac{2}{s^2+12s+24}$	$G(s)=\frac{1}{(1+s)(1+5s)}$	$G(s)=\frac{400}{s^2+50s}$
Settling time [s]	0.846	8.52	0.194
Rise time [s]	0.118	1.11	0.022
Overshoot	0.223	0.251	0.369

TABLE III. TIMING PERFORMCES OF PID CONTROLLER, GENETIC-FUZZY CONTROLLER AND NEURO-FUZZY CONTROLLER

	PID controller	Genetic-Fuzzy controller	Neuro-fuzzy controller
Settling time [s]	0.846	0.699	0.423
Rise time [s]	0.118	0.385	0.234
Overshoot	0.223	0.000	0.000

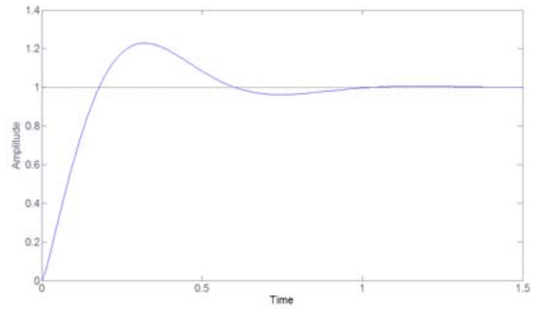

Figure 9a. Step response of PID controller for transfer function defined in (3)

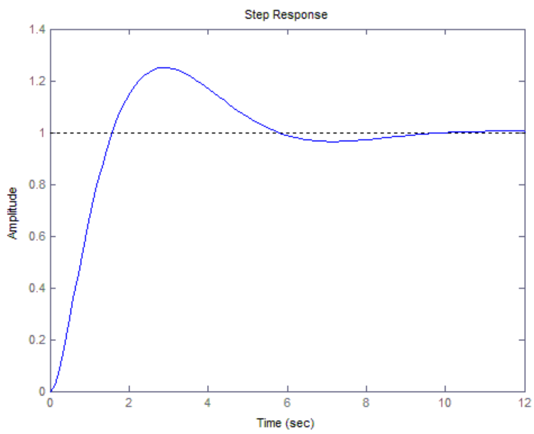

Figure 9b. Step response of PID controller for transfer function defined in (4)

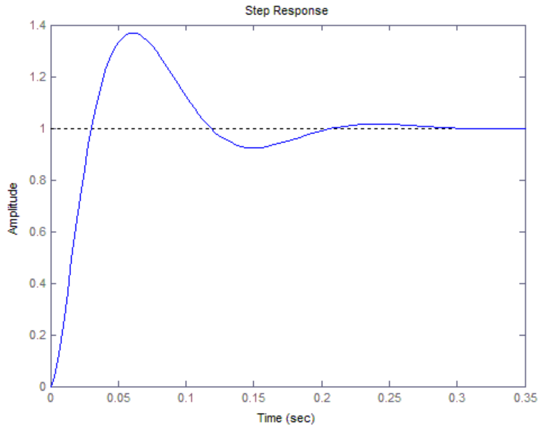

Figure 9c. Step response of PID controller for transfer function defined in (5)

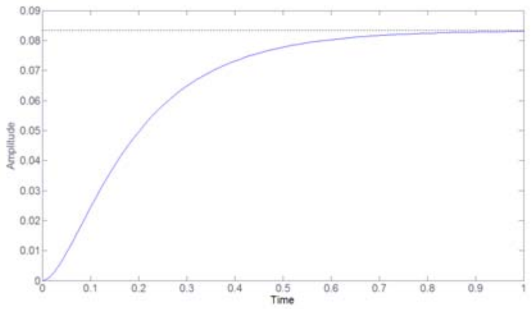

Figure 10. Step response of genetic-fuzzy controller

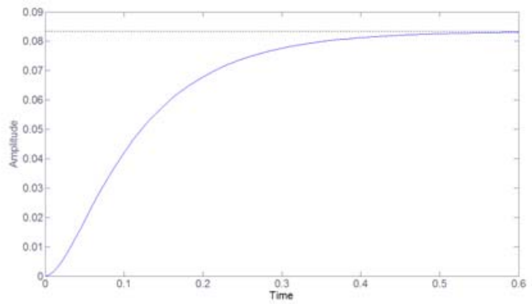

Figure 11. Step response of neuro-fuzzy controller

VI. CONCLUSIONS

Three different approaches has been presented for the control of second order plants. The first approach employs a PID controller based on Ziegler-Nichols tuning technique to control second order systems. The designed PID controller gives good results in terms of rise time. In fact, the simulations shown a value of tr=0.118s. Moreover the controller has shown a good robustness considering the plant parameter change. In order to improve the overshoot and the settling time, we consider a fuzzy controller with optimized membership functions. This optimization is accomplished through the application of Genetic Algorithms. The genetic-fuzzy controller gives a good value of settling time and a very good overshoot value. The third approach is based on the construction of data-driven intelligent controllers able to adjust the weights of fuzzy rules. This task is accomplished through the neural networks. The neural-fuzzy controller gives the best timing parameters improving the control performances of the others two approaches.

The simulation results show that the presented approaches improve the control performance of conventional PID controllers and fuzzy logic PD controllers.

For further works, the fuzzy rules with low weight will be identified and removed to make fuzzy controller more compact and transparent. Once obtained the rules base, the Genetic Algorithms will optimize the fuzzy rules. Another task is to design a suitable training sample to improve the training phase of the neural network. A further improvement will consist to optimize the weights of the neural networks through Genetic Algorithms. Our attempt will be to improve the training phase to achieve optimal neural networks. The application of suitable constraints on overshoot will improve the Ziegler-Nichols PID controller performances. The genetic-fuzzy and neuro-fuzzy controllers will also be used on plants with transfer functions (4) and (5).

REFERENCES

[1] B. Allaoua, A Laoufi, B. Gasbaoui, A. Abderrahmani, Neuro- Fuzzy DC Motor Speed Control Using Particle Swarm Optimization, Leonardo Electronic Journal of Practices and Technologies ISSN 1583-1078, Issue 15, July-December 2009

[2] K.J. Astrom, T. Hagglund, Revisiting the Ziegler–Nichols step response method for PID control, Journal of Process Control 14, pp. 635-650, 2004

[3] J. Amini, Optimum Learning Rate in Back-Propagation Neural Network for Classification of Satellite Images (IRS-1D), Scientia Iranica, Vol. 15, No. 6, pp. 558-567, Sharif University of Technology, December 2008

[4] A. Chad, Genetically Evolving Optimal Neural Networks, Neural Networks and Expert Systems, The Institute of Chartered Financial Analysts of India (ICFAI), 2007

[5] A. Chegeni, A. Khoei, K. Hadidi, Improved Genetic Algorithm-Based Optimization of Fuzzy Logic Controllers, First Joint Congress on Fuzzy and Intelligent Systems, Ferdowsi University of Mashhad, Iran, 2007

[6] S. Chopra, R. Mitra and V. Kumar, Fuzzy Controller: Choosing an Appropriate and Smallest Rule Set, International Journal of Computational Cognition, Vol. 3, No. 4, December 2005

[7] J. I. Corcau, E. Stoenescu, Fuzzy logic controller as a power system stabilizer, International Journal of Circuits Systems and Signal Processing, Vol. 1, 2007

[8] Desborough, L. D., Miller, R. M., Increasing customer value of industrial control performance monitoring: Honeywell's experience. Chemical Process Control VI, Vol. 98, AIChE Symposium Series No. 326, USA, 2002

[9] Dingyu Xue, YangQuan Chen, Derek P. Atherton, Linear Feedback Control: Analysis and Design with MATLAB, Chapter 6, Siam, 2007

[10] A. Fiszelew, P. Britos, A. Ochoa, H. Merlino, E. Fernandez, R. Garca-Martnez, Finding Optimal Neural Network Architecture Using

Genetic Algorithms, Research in Computing Science, pp. 15-24, 27 2007

[11] Hui Cao, Gangquan Si, Yanbin Zhang,Xikui Ma, Jingcheng Wang, Load control of ball mill by a high precision sampling fuzzy logic controller with self-optimizing, Asian Journal of Control, Vol. 10, N. 6, pp. 621-631, 2008

[12] J. Ivakpour, Genetic Algorithm Performance, Matlab Central, MathWorks, 2006

[13] S. Khan, S. F. Abdulazeez, L. W. Adetunji, AHM Zahirul Alam, Momoh Jimoh E. Salami, Shihab Ahmed Hameed, Aisha Hasan Abdalla and Mohd Rafiqul Islam, Design and Implementation of an Optimal Fuzzy Logic Controller Using Genetic Algorithm, Journal of Computer Science 4 (10): 799-806, ISSN 1549-3636, 2008

[14] M. Kumar, D.P. Garg, Intelligent learning of fuzzy logic controllers via neural network and Genetic Algorithms, Japan-USA Symposium on Flexible Automation, Denver, Colorado, 2004

[15] Meza, J.L., Soto, R., Arriaga, J., An Optimal Fuzzy Self-Tuning PID Controller for Robot Manipulators via Genetic Algorithm, Eighth Mexican International Conference on Artificial Intelligence, pp. 21-26, 2009.

[16] S. R. Munasinghe, Min-Soeng Kim,Ju-Jang Lee, Adaptive Neurofuzzy Controller to Regulate UTSG Water Level in Nuclear Power Plants, IEEE Transactions on Nuclear Science, Vol. 52, No. 1, 2005

[17] D. Pelusi, Optimization of a Fuzzy Logic Controller using Genetic Algorithms, IEEE International Conference on Intelligent Human-Machine Systems and Cybernetics, ISBN: 978-1-4577-0676-9, Vol. 2, pp. 143-146, Hangzhou, 2011

[18] D. Pelusi, Genetic-Neuro-Fuzzy Controllers for second order control systems, IEEE European Modelling Symposium on Mathematical Modelling and Computer Simulation, ISBN: 978-1-4673-0060-5, pp. 12-17, Madrid, 2011

[19] D. Pelusi, On designing optimal control systems through genetic and neuro-fuzzy techniques, IEEE International Symposium on Signal Processing and Information Technology, ISBN: 978-1-4673-0753-6, pp. 134-139, Bilbao, 2011

[20] D. Pelusi, Improving Settling and Rise Times of Controllers Via Intelligent Algorithms, IEEE International Conference on Modelling and Simulation, ISBN:978-0-7695-4682-7/12, pp. 187-192, Cambridge, 2012

[21] Ping Chang, Jeng-Shong Shih, The Application of Back Propagation Neural Network of Multi-channel Piezoelectric Quartz Crystal Sensor for Mixed Organic Vapours, Tamkang Journal of Science and Engineering, Vol. 5, No. 4, pp. 209-217, 2002

[22] M. Shamusuzzoha, S. Skogestad, The setpoint overshoot method: A simple and fast closed-loop approach for PID tuning, Journal of Process Control 20, 1220-1234, 2010

[23] Shiuh-Jer Huang, Ming-Tien Su, Gain Tuning Fuzzy Controller for an Optical Disk Drive, World Academy of Science, Engineering and Technology, 36, 2007

[24] S.R. Vaishnav, Z.J. Khan, Design and Performance of PID and Fuzzy Logic Controller with Smaller Rule Set for Higher Order System, Proceedings of the Word Congress on Engineering and Computer Science, San Francisco, USA, 2007