Online Adaptive Fuzzy Logic Controller Using Genetic Algorithm and Neural Network for Networked Control Systems

Pooya Hajebi*, Seyed Mohammad Taghi AlModarresi*

*Department of Electrical and Computer Engineering, Yazd University, Yazd, Iran

Hajebi@stu.yazd.ac.ir, smta@yazd.ac.ir

***Abstract*—Networked Control Systems are used for controlling remote plants via shared data communication networks such as Ethernet. These systems have found many applications in industrial, medical and space sciences fields. However there are some drawbacks in these systems, which make them challenging to design. One of the most common problems in these systems is the stochastic time delay. Packet switching in internet brings about the randomly varying time delay and consequently makes these systems instable. Convenient controllers such as PID and PI type controllers which are just matched with a constant time delay could not be a solution for these systems. Fuzzy logic controllers due to their nonlinear characteristic which is compatible with these systems are potentially a wise option for their control purpose. Fuzzy logic controller could become adaptive by means of neural networks and beneficial to deal with the varying time delay problem. Further, they do have more capabilities to tackle packet dropouts and dynamically system variables. This paper introduces a novel control method which addresses the varying time delay problem effectively. This novel method suggests an online adaptive fuzzy logic controller which has been controlled and adapted through the neural network. This method takes the advantage of the genetic algorithm to optimize the membership functions for its fuzzy logic controller. This designed controller is applied to an AC 400 W servo motor as a remote plant in order to control its position via Ethernet. The measurement of round-trip time (RTT) is used to estimate the online time delay as a parameter in online adaptive fuzzy logic controller. The rule-based table of designed fuzzy logic controller rotates in relation to this estimated time delay. The value of rotating is obtained from a trained neural network. Comparison of simulation results for different controllers indicates that this novel designed controller provides a better performance over the varying time delay. The proposed method follows the input easily, despite classical methods which result in an unstable system especially over the large time delays as large as 600 ms. Results get even more improved when genetic algorithm is applied to fuzzy logic controller.**

***Index Terms*—Data Communication Networks, Genetic Algorithm, Networked Control Systems, Neural Networks, Online Adaptive Optimized Fuzzy Logic Controller, Rules-Table Rotation.**

Manuscript received June 29, 2012. This work was supported in part by the ICT Research Institute (ITRC), Iran, under Grant T-500-12192.

P. Hajebi is with the Department of Electrical and Computer Engineering, Yazd University, Yazd, Iran (phone: +98-351-8122377; fax: +98-351-8200144; e-mail: Hajebi@stu.yazd.ac.ir).

S.M.T. AlModarresi is with the Department of Electrical and Computer Engineering, Yazd University, Yazd, Iran (e-mail: smta@yazd.ac.ir).

I. Introduction

NETWORKED control systems (NCSs) are spatially distributed systems in which the communication between sensors, actuators and controllers occurs through a shared band-limited digital communication network [1], [2]. This multipurpose shared network connecting, spatially distributed elements, creates a flexible architecture which generally reduces installation and maintenance costs. NCSs have been finding application in a broad range of areas such as mobile sensor networks [3], remote surgery [4], haptics collaboration over the Internet [5]–[7], and automated highway systems and unmanned aerial vehicles [8], [9]. Murry et al. in [10] have identified control over networks as one of the key future directions for control. However, application of a shared network versus several dedicated independent connections, introduces new challenges. Drops and variable delays in NCSs are two major problematic issues that were addressed in [11], [12]. Packet dropouts and finite level quantization make NCSs unstable [12]. When the delay time is less than the sampling time of NCSs, results show that the time delay has insignificant effect on control system. However, delay time greater than the sampling time degrade the performance of the NCSs [13]. Many controllers such as conventional PID and fuzzy logic controllers are utilized to stabilize the NCS closed loop feedback and to reduce the error. Classical Smith predictor is one of the controllers which are efficient for time delay processes [13], [14]. Lai and Hsu proposed an adaptive Smith predictor as a controller for NCS in [14]. Despite showing relatively a good performance, there are some drawbacks in these controllers. For instance, the accuracy of the model depends on plant transfer function estimation. Moreover, each new plant requires changing the controller design. Practically estimation of plant transfer function is not exact. Recently Pan et al. in [15] and Zhao et al. in [16] have shown that fuzzy logic controllers offer a better performance in tackling packet dropouts and varying time delay, at the same time are more compatible with nonlinear processes. W. Du and F. Du proposed a Smith predictor integrated with fuzzy adaptive PID

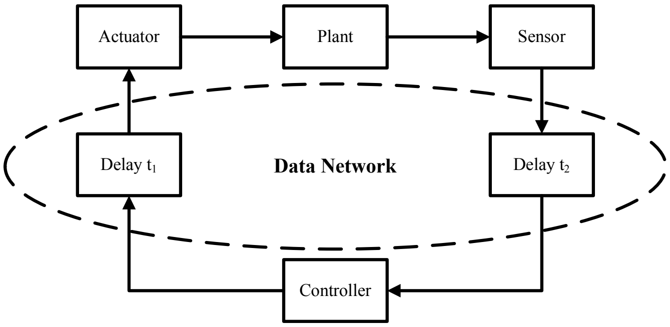

Fig. 1. A SISO networked control system structure.

controller for the NCSs in [17]. However they did not measure the network delay online. They applied fuzzy logic controller for tuning the coefficients of PID controller. This paper first, suggests a fuzzy logic controller (without PID controller) to control the position of an AC 400 W servo motor via Ethernet. At the next step, it proposes a novel control method which is an online adaptive fuzzy logic controller for the similar application. This research provides the advantages of no PID controllers application while offers an adaptive controller which its fuzzy logic rules are rotating during the plant control. The round-trip time (RTT) is measured online and this value is utilized as, t_m, time delay parameter. Then, this time delay value is mapped to an angle by means of trained neural network. This neural network has been already trained by different time delays in adaptive fuzzy logic controller. Results verify the better performance of this novel design which its fuzzy logic controller rules-table rotates through a trained neural network. The fact is that in communication networks time delay could exceed 200 ms (vs. 400 ms, 600 ms). However, results from [17], [18] show that the response would be degraded in these systems for time delays over 200 ms despite the application of designed offline controllers. This paper's proposed method has shown an improved response especially in the case of time delays over 200 ms. Even with time delay of 600 ms, there is no degradation in step response. Genetic algorithm could be implied in order to optimize a suitable objective function for tuning the fuzzy logic controller membership functions. The results indicate that this optimized controller shows better performance in comparison with a non-optimized fuzzy logic controller.

This paper includes the following sections; In Section II, NCSs, stochastic time delay and packet dropouts are introduced. Section III, first describes, designing a fuzzy logic controller in order to control the position of an AC 400 W servo motor and next introduces a novel adaptive fuzzy logic controller with a rotating rules-table by means of trained neural network. Section IV introduces the genetic algorithm and describes its application in optimization of membership functions of the designed fuzzy controller. Section V contains the related simulations and equations. This paper ends with conclusion in section VI.

II. Networked Control Systems

Due to quantum leaps in communication systems, in recent years, it has become more common to apply a shared

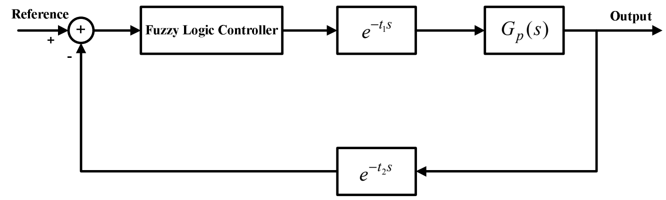

Fig. 2. Fuzzy logic controller.

communication channel such as Ethernet or Controller Area Network (CAN) bus etc. for transmission of the control signal and the measured output. This method helps reducing the wiring costs as well as eliminates the necessity for maintaining dedicated communication channels for each control parameter [15]. However, this type of networked control system is not a perfect solution and own its various unsolved issues such as transmission delays and packet dropouts [12], [15] which can degrade control performance. The SISO (single input-single output) NCS structure in the closed loop model is shown in Fig. 1. As illustrated in this figure, t_1 and t_2 indicate, time-delays induced in the network structure for the controller-to-actuator direction and the sensor-to-controller direction, respectively. Basically, the induced network delay varies according to the network load, scheduling policies, number of nodes, and different protocols. Time-varying characteristic of these NCSs makes the design and modeling of them more complicated. The total time-delay can be categorized into three classes, based on the parts where they occur, namely, the server node, the network channel, and the client node [19]. In addition, the round-trip time (RTT) measurement is crucial as it provides of accurate delay measurements periodically [19]–[21]. RTT is defined as the total time delay in SISO NCSs. Obviously the longer distances increase, the time delay of a network since more nodes are involved and consequently results in a larger RTT. In a classical Smith predictor design, the value of t_m is constant and usually equals to average approximation of time delay between two nodes in the network. The value of RTT could be applied to fuzzy logic controller for compensating of variable delay. Normally in a fuzzy logic controller, rule-based table is constant during the control process action. In this paper's suggested method, RTT applied by neural network mapping, generates a rotating rules-table.

III. Fuzzy Logic Controller Using Rules-Table Rotation

As it has been already mentioned, an online adaptive fuzzy logic controller could be a solution for tackling the stochastic time delay problem in NCSs. However it controls with simple PID or PI controllers which shows limited potentials, especially in nonlinearity processes. Recently it is proved that fuzzy logic controller is the best option for controlling nonlinear processes while make the system more robust against the varying time delay [15], [16]. In the following parts, first a fuzzy logic controller is designed then a classical Smith predictor would be integrated with this designed fuzzy logic controller based on our plant. Finally a novel rotating rules-table online adaptive fuzzy logic controller is described.

A. Designing Fuzzy Logic Controller

To implement a NCS controller, first the output of plant is measured and then it would be compared with a reference signal. This comparison generates the error signal. The error signal and its derivative are both inputs for the fuzzy logic controller shown in Fig. 2. Here in this paper, the plant is an AC 400 W servo motor which its position as an output is measured with an encoder with gain 10^4 P/R. The coefficients of the equivalent PI controller for this plant are K_p=0.0001 and K_i= 0.00000001, [14]. The open loop position control is obtained from (1). Equations (2) and (3) represent the continuous-state space form of transfer function described in (1). In Fig. 3, seven triangular membership functions have been devoted to either, input (error and derivative of error) and output. In Fig. 3, the fuzzy linguistic variables (“NB”, “NM”, “NS”, “ZE”, “PS”, “PM”, “PB”) represent (Negative Big, Negative Medium, Negative Small, Zero, Positive Small, Positive Medium and Positive Big) respectively. In position control, the output follows the input. Therefore, at first they are assumed to have similar membership functions. However in the following section, output memberships would be optimized using an objective function. Here are provided some design specifications, applied in this fuzzy logic controller: 1) The inference, used in this design is Mamdani-type [22], 2) Fuzzy logic “and operator” was implemented by “min” method while the fuzzy logic implication is based on the “min” method as well and rules are aggregated using fuzzy “max” operator, 3) The fuzzy logic output has been determined through the center of gravity method by means of defuzzification, 4) Fuzzy rules are opted based on Table I which contains 49 rules, 5) Due to

$$G_P(s) = \frac{10^4(0.058\,s + 3.221)}{s\,(0.0001\,s^2 + 0.019\,s + 1)} \tag{1}$$

$$\dot{x} = \begin{bmatrix} -190 & -78.125 & 0 \\ 128 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} x + \begin{bmatrix} 2048 \\ 0 \\ 0 \end{bmatrix} U \tag{2}$$

$$y = \begin{bmatrix} 0 & 22.13 & 1228.7 \end{bmatrix} x \tag{3}$$

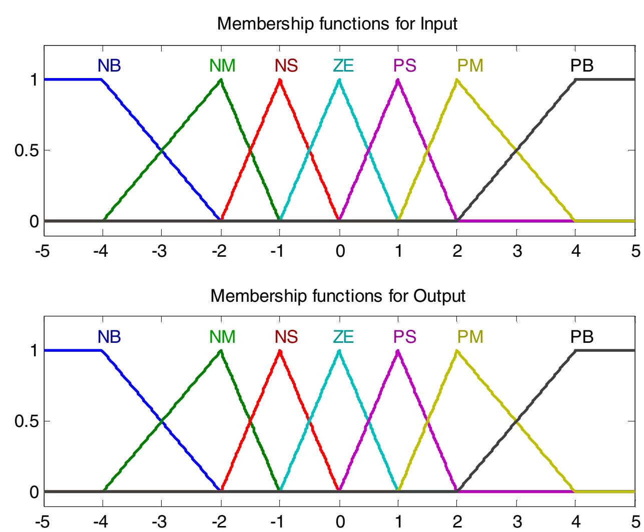

Fig. 3. Membership functions for input and output.

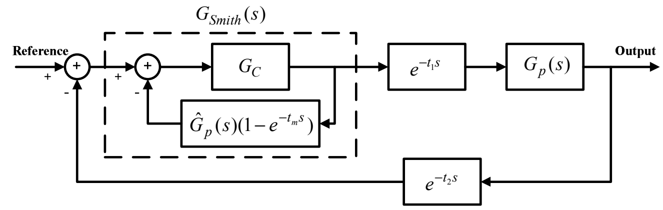

Fig. 4. A control structure of Smith predictor.

high gain of encoder the scaling factor value selected for fuzzy logic controller output is ,10^{-4}.

B. Classical Smith Predictor with Fuzzy Logic Controller

Classical Smith predictor is one of the controllers which are efficient for time delay process [13], [14]. Here a classical Smith predictor is designed for comparing the results. In this classical Smith predictor which is shown in Fig. 4, G_C is the designed fuzzy logic controller described in section III. A., G_P is the transfer function of the plant while $\hat{G}_P$ is the estimation of plant transfer function. Usually t_m is the approximation of total time delay from controller to plant and plant to controller. If t_m is the appropriate estimation of overall time delay the performance of system will be reasonable. t_m is assumed 200 ms in the simulation. $\hat{G}_P$, is the estimation of G_P, and practically difference between these two transfer functions results in instabilities and increases of the error of response. This is the main problem for classical Smith predictor and online adaptive Smith predictor. In this paper classical Smith predictor with fuzzy logic controller was assumed ideal, thus the G_P and $\hat{G}_P$ are equal in the simulations. The fuzzy rules are selected based on Table I.

C. Designing Rules-Table Rotation of Online Adaptive Fuzzy Logic Controller Using Neural Network

RTT is estimated in network [19], [20] and then this measurement would be applied to online fuzzy logic controller. In this stage the designed fuzzy logic controller in part A would be integrated with an online neural network. The measurement of round-trip time (RTT) is applied for estimating of online time delay which in turn provides the value for rotation angle of fuzzy rules-table. As already mentioned, the controller in this paper has not included any PID or PI method.

TABLE I
RULE BASE FOR ERROR, ERROR DERIVATIVE AND FLC OUTPUT (WITHOUT ROTATION).

de \ e	NB	NM	NS	ZE	PS	PM	PB
NB	NB	NB	NB	NB	NM	NS	ZE
NM	NB	NB	NB	NM	NS	ZE	PS
NS	NB	NB	NM	NS	ZE	PS	PM
ZE	NB	NM	NS	ZE	PS	PM	PB
PS	NM	NS	ZE	PS	PM	PB	PB
PM	NS	ZE	PS	PM	PB	PB	PB
PB	ZE	PS	PM	PB	PB	PB	PB

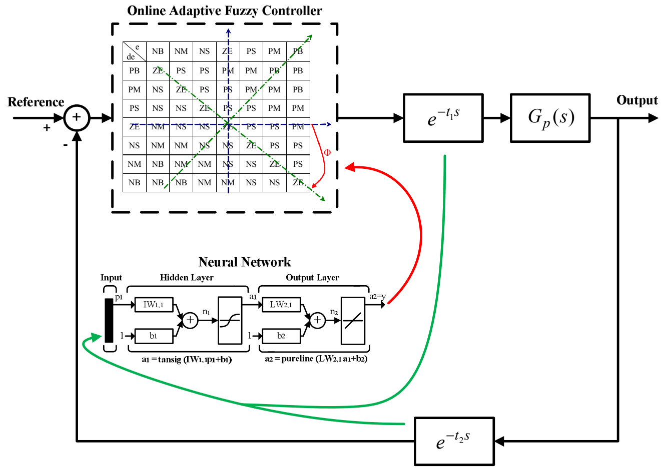

Fig. 5. Structure of online adaptive fuzzy logic controller using neural network.

The nonlinear fuzzy logic controller does have the potential to control the complicated and nonlinear processes while is more robust against the dynamically system variables specially occurs at the beginning of the process. In a control process with no delay, the error and the derivative of the error change periodically. However, these changes suggests nonlinear function pattern. During the control process and based on the taken time, the error and derivative of the error move on the fuzzy rules-table in a circular path. While delays caused the error and derivative of error do not have desired time values, application of this suggested rotation method could overcome problem of delays. Thus, this paper has suggested a control method which integrated fuzzy logic controller with a neural network. Fig. 5, shows the structure of this proposed controller. Here in this figure, the value of RTT is mapped to an angle by neural network. The structure of neural network has two-layer feedforward. First this neural network is trained by several set point time delays. It means the value of rotation for several time delays is obtained manually then these values will be applied for training the neural network.

The value of angle for rotating rules-table in online adaptive fuzzy logic controller, changes periodically based on the RTT value. Fuzzy rules are opted based on Table II, but other parameters (membership functions, fuzzy logic operators and fuzzy logic method) are similar to data in section III. A. Equation (4) shows the mapping relation of the error and variation of the error in new coordinate. Matrix A, in (5) is the rotation transform matrix which rotates coordinates by the angle

TABLE II
RULE BASE FOR ERROR, ERROR DERIVATIVE AND FLC OUTPUT (ROTATION TABLE).

de \ e	NB	NM	NS	ZE	PS	PM	PB
PB	ZE	PS	PS	PM	PM	PB	PB
PM	NS	ZE	PS	PS	PM	PM	PB
PS	NS	NS	ZE	PS	PS	PM	PM
ZE	NM	NS	NS	ZE	PS	PS	PM
NS	NM	NM	NS	NS	ZE	PS	PS
NM	NB	NM	NM	NS	NS	ZE	PS
NB	NB	NB	NM	NM	NS	NS	ZE

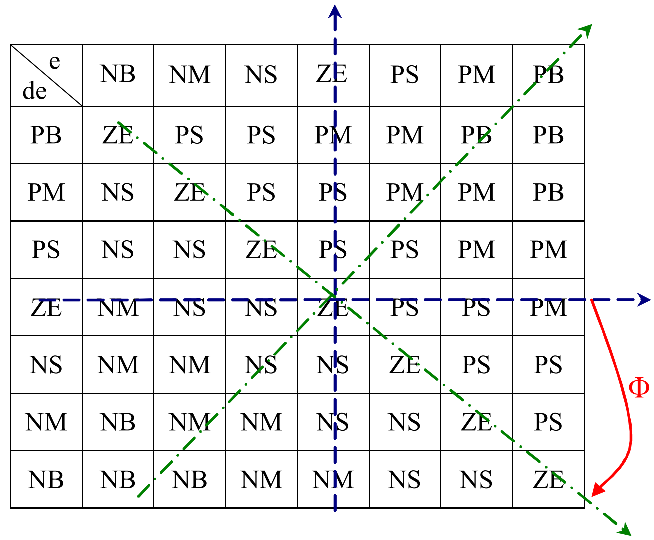

Fig. 6. Rotating of fuzzy rules-table.

of φ radian. The rules-table rotation structure of fuzzy logic controller and trend of rotation are shown in Fig. 6.

$$\begin{bmatrix} e_{new} \\ \dot{e}_{new} \end{bmatrix} = A \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \tag{4}$$

$$A = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \tag{5}$$

IV. OPTIMIZATION OF FUZZY LOGIC CONTROLLER USING GENETIC ALGORITHM

To improve the proposed fuzzy logic controller, genetic algorithm is used to find the optimal membership functions [15]. Here, firstly the genetic algorithm is explained, and then the optimization of fuzzy logic controller by means of genetic algorithm is discussed.

A. Genetic Algorithm

Genetic algorithm was firstly introduced by John Holland and developed by him, his student and colleagues [23]. These algorithms are heuristic optimization process inspired by natural evolution and could be used to minimize a suitable objective function or fitness function for tuning the fuzzy logic controller parameters. It is more effective at avoiding local minima than differentiation based methods.

The genetic algorithm will generally include three fundamental genetic operations of selection, crossover and mutation. These operations are used to modify the chosen solutions and select the most appropriate offspring to succeeding generations [23].

In a genetic algorithm, a population of strings (called chromosomes), which encode candidate solutions (called individuals) to an optimization problem, evolves toward better solutions. Traditionally, solutions represented in binary as strings of "0"s and "1"s. The evolution usually starts from a population of randomly generated individuals and happens in generations. In each generation, the fitness of every individual in the population is evaluated, multiple individuals are stochastically selected from current population (based on their fitness), and modified (recombined and possibly randomly mutated) to form a new population. The new population is then used in the next iteration of the algorithm. Commonly the algorithm terminates when either a maximum number of generations has been produced, or satisfactory fitness level has been reached for the population.

The basic iterations of genetic algorithm can be summarized as follows [23]–[25]:

1) Genetic representation: encoding the variables. Genetic algorithm often encodes solutions as fixed length "bitstrings" (e.g. 101110, 111111, and 000101).

2) A method for generating the initial population: population may be generated randomly or problem specific knowledge can be used to construct the chromosomes with the populations.

3) An evaluation function, which assign a real number to measure the fitness of each chromosome.

4) A reproduction selection scheme, which is used to select chromosome to be exposed to genetic operations. One of the most famous approaches is "roulette wheel", which selects chromosomes proportional to their fitness values.

5) Genetic operators: crossover and mutation are two main operators in genetic algorithm. These operators are applied to modify the chosen chromosomes (parents) and select the most appropriate offspring to pass on to succeed generations. The basic mechanism of genetic algorithm, crossover of the parental and mutation illustrated in Fig. 7. Crossover is done by selecting two parents during reproduction and combining genes to produce offspring. Two selected parents are combined with some probability (crossover rate); therefore two new offspring will be born. One gene or several genes of each offspring may then change randomly (mutation) with some probability (mutation rate). Usually the crossover has high probability (typically values are between 0.8 and 0.95) and mutation has small probability (typically values are between 0.1 and 0.001).

6) Termination: The cycle of genetic algorithm will be continued until the genetic algorithm reaches to stopping criteria. There are several approaches to terminate the genetic algorithm. A common approach is to terminate genetic algorithm when the number of generations reaches to specific value. The genetic algorithm process may also run just for limited time duration. It is also possible to terminate a genetic

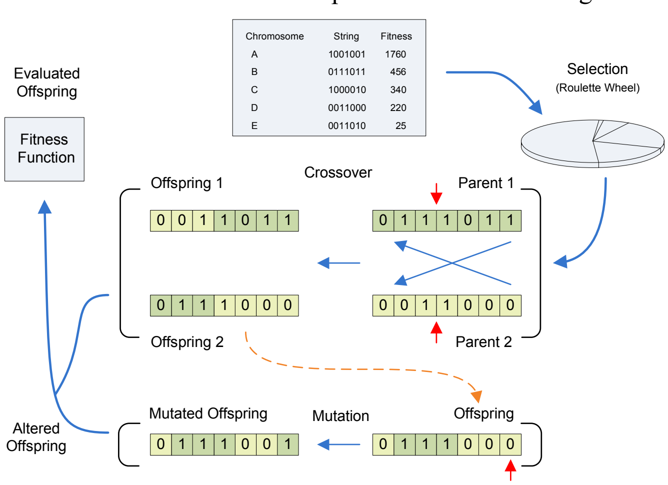

Fig. 7. Mechanism of genetic algorithm.

algorithm when the objective function of the best chromosome has not improved in the several generations.

B. Using Genetic Algorithm for Optimizing of Fuzzy Logic Controller Membership functions

In section III. A., input and output memberships of fuzzy logic controller are assumed similar. Now, with taking to consider the values from the last input membership functions, the output membership functions could be optimized. The shapes of membership functions in fuzzy logic controller are triangular. Therefore in the case of triangular fuzzy sets, three characteristic points (center and two widths) are used as the parameters should be optimized. Here, the number of triangular membership functions is seven ("NB", "NM", "NS", "ZE", "PS", "PM", "PB"). To design symmetric controller for positive and negative input pulses, the membership functions are assumed symmetric to Y axis. Also the center of ZE is assumed zero. Therefore there are nine points or variables for optimizing the output membership functions; two points for "NB", three points for "NM", three points for "NS" and one point for "ZE". The range of changes in these variables is between 0.1 and 4. Moreover, there are constraints in optimization which guarantee that the membership functions are ordered according to their values (e.g. NB<NM<NS<ZE<PS<PM<PB); for instance, the center of "NB" must be less than the center of "NM". These constraints are considered to optimize the membership functions. The number of initial population is assumed 100. Minimizing integral of time-weighted absolute error (ITAE) is commonly referred to as good performance index in designing PID controllers. Thus the ITAE assumed as objective function. Equation (6) shows the mathematical formula of ITAE. Where t is the time and e is the different between output and reference in control process.

$$ITAE = \int_0^\infty t\left|e(t)\right| dt \qquad (6)$$

Selection of parents is based on "roulette wheel". The rate of crossover which applied to parents is 0.8. Crossover trends to make the chromosomes within the population more similar, whereas mutation trends them more divers and usually has low rate. Here, because of existing constraints for membership functions, the mutation is not applied. Also here the genetic algorithm is terminated when 32 generations have been produced.

V. Simulation Results

A closed loop NCS unit in this paper includes these sections: online adaptive fuzzy logic controller, neural network, plant, data communication network. In order to analyse the whole unit each section should be analysed separately. This paper has applied the state equations to plot the step response of this NCS. At this first stage, transfer functions of plant and controller are converted to state equations. Since the data transmitted over the network is digital these equation states need to be discretized state space equation to be able to simulate the processes. Equation (7) shows the discrete state-space form of process. While A, B, C and D are the continuous state space matrices then their equal discrete state space matrices (A_d, B_d, C_d, D_d) would be obtained from (8), (9), (10), (11). By substituting the (2), (3) in to (7)-(11), discrete state space form of plant is obtained and represented in (12), (13).

$$\begin{cases} x[k+1] = A_d x[k] + B_d u[k] \\ y[k] = C_d x[k] + D_d u[k] \end{cases} \tag{7}$$

$$A_d = e^{AT} \tag{8}$$

$$B_d = \left(\int_0^T e^{A\tau} d\tau \right) B \tag{9}$$

$$C_d = C \tag{10}$$

$$D_d = D \tag{11}$$

$$x[k+1] = \begin{bmatrix} 0.0066 & -0.2973 & 0 \\ 0.4870 & 0.7295 & 0 \\ 0.0035 & 0.0089 & 1 \end{bmatrix} x[k] + \begin{bmatrix} 7.7924 \\ 7.0909 \\ 0.0277 \end{bmatrix} U[k] \tag{12}$$

$$y[k] = \begin{bmatrix} 0 & 22.13 & 1228.7 \end{bmatrix} x[k] \tag{13}$$

In these simulations a random time delay is provided to analyse the performance of NCS. The total of command delay, t_1, and feedback delay, t_2, generates the total time delay (RTT) which is shown in Fig. 8. After training of neural network, the biases and weights values are obtained shown in Table III. Neural network transfer functions for layer 1 (hidden layer) and layer 2 (output layer) are "tansig" and "purelin" respectively. Equations (14) and (15) show these two transfer functions in mathematical forms, while figures of these transfer functions are plotted in Fig. 9.

$$tansig\,(n) = \frac{2}{1+e^{-2n}} - 1 \tag{14}$$

$$purelin\,(n) = n \tag{15}$$

The sampling time is assumed 0.01 second and the model of NCS is based on the model described in [26]. Here in this paper the simulations and comparisons of the step response are provided among three controller types: 1) Online adaptive fuzzy logic controller, 2) Classical Smith predictor with fuzzy logic controller and 3) Pure fuzzy logic controller. The results are illustrated in Fig. 12. Results show that the online adaptive fuzzy logic controller offer a better performance compared to other two controllers.

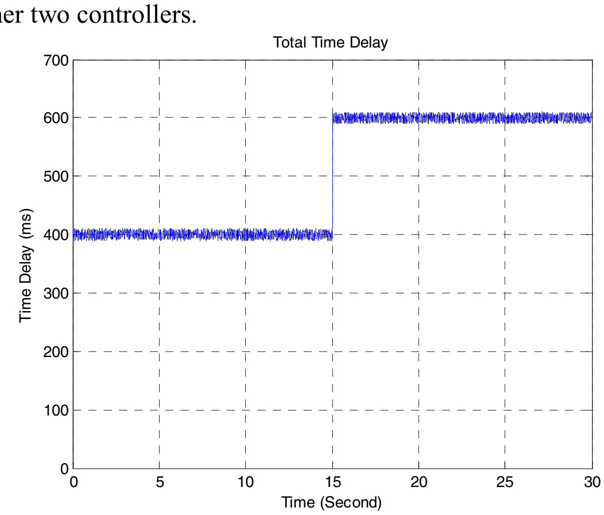

Fig. 8. A sample of simulated varying time delay in NCS.

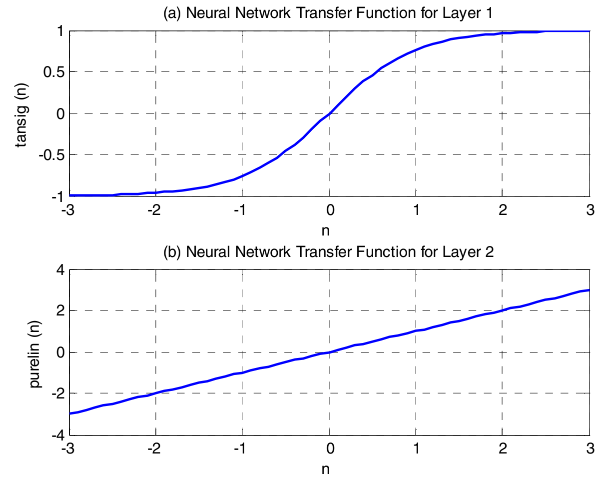

Fig. 9. Neural networks transfer functions; a) tansig; b) purelin.

As can be seen the output signal of online adaptive fuzzy logic method does have small overshoot and fast response. Therefore this controller is recommended for networked control systems purposes. W. Du and F. Du have suggested a Smith predictor integrated with adaptive fuzzy-PID controller for the NCSs in [17]. However they did not measure the network delay online. They applied a fuzzy logic controller just for tuning the coefficients of PID controller, which means that their suggested controller works offline. They also have designed in [18] a RBF neural network control with Smith predictor for NCSs which works offline as well. In communication networks time delay could exceeds 200 ms (vs. 400 ms, 600 ms). The results from [17], [18] show that response would degrade with time delays over 200 ms. In [17], [18], it was assumed that the maximum of burst time delay is about 200 ms while this time delay was applied discretely. Our proposed method has more improved response especially when the time delay is over 200 ms even with the time delay of 600 ms there is no degradation in step response. In the spite of applying this large value of time delay continuously, results in Fig. 13 show that this does not have any more effects on the step response as well.

As described in section IV, This paper takes the advantage of genetic algorithm method to optimize the fuzzy controller membership functions and consequently improve the overall results. For this propose, it starts with assumption of a no delay system while optimizes the fuzzy logic controller. The applied objective function type in this paper is ITAE. In this case, which is a position control example, the goal is that the output follows the input more closely. The closer output follows the input the more accurate is the performance. Tuning of an online adaptive fuzzy logic controller reduces the error value even with the presence of time delay in the system. Hence genetic algorithm is applied for some input pulses, and as a result, the output membership functions become tuned. The values of objective functions in each generation are shown in Fig. 10. As could be perceived, the objective function values in each generation become smaller compared to their previous generation which is inherently a genetic algorithm characteristic. Another point is that objective functions do not show significant changes in last four generations, which indicates that, this genetic algorithm roughly obtains its best answer. The best value of ITAE in 32th generation is 0.05292. After applying the genetic algorithm to fuzzy logic controller, tuned output membership function shapes could be illustrated in Fig. 11. These membership functions are used for last three types of controller. For online adaptive fuzzy logic controller, the neural network is trained based on new optimized membership functions. Obtained biases and weights values are shown in Table IV. Rotation values are obtained according to different delays in experiments. Then the neural network is trained by these values. Experiments in Fig. 12 and Fig. 13 are repeated and are shown in Fig. 14 and Fig. 15. Results of online adaptive fuzzy logic controller are better compared to the other two controllers. Since the pure fuzzy logic controller is tuned in conditions with no delay, changes in membership make it unstable with the presence of time delay. Moreover, classical Smith predictor as shown in Fig .15 becomes unstable for delay times over 200 ms similar to pure fuzzy logic controller. It is even worse when the time it is not tuned. Table V shows the comparison of results. Two main evaluation indexes for comparison of results are the rise time (T_r) and the Percent Overshoot (*P.O.*). Rise time is that time taken for the output of plant to rise from 10% to 90% of its final value when simulated by step input. The Percent Overshoot is defined as (16). For a unit step input, where M_{Pt}, is the first peak value of the time response, and f_v is the final value of the response. Normally, f_v is the magnitude of the input.

$$P.O. = \frac{M_{Pt} - f_v}{f_v} \times 100\,\% \qquad (16)$$

Peak time (T_p) is the time that takes for output of plant to reaches the peak value. The smaller the *P.O.* and M_{Pt} values, the better is the controller performance. The results from Table V show that our proposed method has no overshoot. Furthermore, when the membership functions are optimized with genetic algorithm, the T_r is more smaller compared to the non-optimized case. Since the Smith predictor controller is tuned for 200 ms delay, when optimization is applied, this controller shows better results with the time delay of 200 ms. However, for other induced delays, its performance will decrease. Pure fuzzy logic controller tuned with no time delay and shows poor performance result compared to other two controllers.

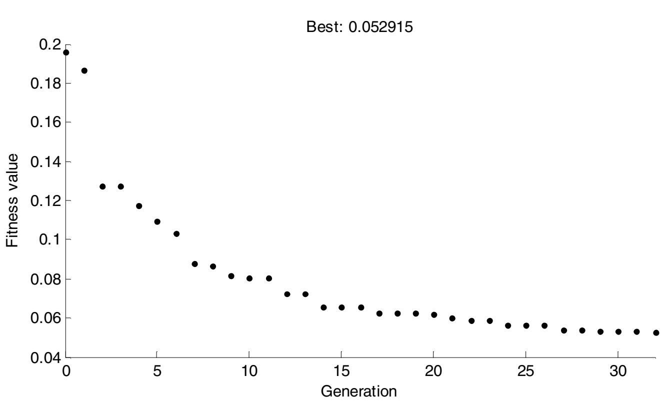

Fig. 10. Objective functions values for different generations of genetic algorithm.

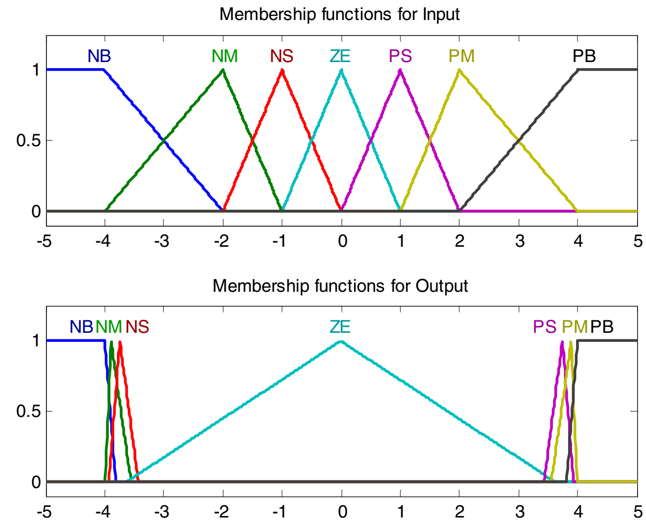

Fig. 11. Optimal fuzzy logic membership functions.

VI. Conclusion

NCSs have found widely application in various fields recently. However there are some drawbacks in their structures such as varying time delay and packet dropouts, which makes the control design of these systems challenging. Conventional PID and fuzzy logic controllers are mostly designed to address the instability problems in NCSs. Fuzzy logic controllers with the great potential in tackling the nonlinear processes and making NCSs more robust against the dynamically variable parameters could be a more reasonable option for NCSs. Moreover, a rotating rules-table fuzzy logic controller is a great solution for stochastic time delay. Thus according to above mentioned characteristic of both fuzzy logic controller and rules-table rotation, this paper has proposed a novel controller for NCSs. This novel design has integrated a rotating rules-table fuzzy logic controller with a neural network to control the position of an AC 400 W servo motor as a remote plant via Ethernet. Simulation results and their comparison for three different methods of controlling over this plant verified that this novel controller design is more beneficial especially over the big value of delay times as large as 600 ms. The proposed method shows better performance when the fuzzy logic membership functions are optimized by means of genetic algorithm.

TABLE III
Weights and Biases Values for Neural Network.

$IW_{1,1} = \begin{bmatrix} -0.0057 \\ -0.2547 \end{bmatrix}$	$LW_{2,1} = \begin{bmatrix} 0.3566 & 0.8011 \end{bmatrix}$
$b_1 = \begin{bmatrix} 1.1243 \\ 24.6125 \end{bmatrix}$	$b_2 = 0.5684$

TABLE IV
WEIGHTS AND BIASES VALUES FOR NEURAL NETWORK.
(FOR OPTIMIZED FUZZY LOGIC CONTROLLER)

$IW_{1,1} = \begin{bmatrix} 0.0036 \\ 0.0154 \end{bmatrix}$	$LW_{2,1} = \begin{bmatrix} -2.9954 & -15.2386 \end{bmatrix}$
$b_1 = \begin{bmatrix} 1.286 \\ 1.4904 \end{bmatrix}$	$b_2 = 17.5069$

REFERENCES

[1] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, “A survey of recent results in networked control systems,” *Proc. IEEE*, vol. 95, no. 1, pp. 138–161, Jan. 2007.

[2] F. L. Lian, J. R. Moyne, and D. M. Tilbury, “Performance evaluation of control networks: Ethernet control net and device net,” *IEEE Control Syst. Mag.*, pp. 66–83, Feb. 2001.

[3] P. Ogren, E. Fiorelli, and N. E. Leonard, “Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment,” *IEEE Trans. Automat. Contr.*, vol. 49, no. 8, pp. 1292–1302, Aug. 2004.

[4] C. Meng, T. Wang, W. Chou, S. Luan, Y. Zhang, and Z. Tian, “Remote surgery case: Robot-assisted teleneurosurgery,” in *IEEE Int. Conf. Robot. and Auto. (ICRA’04)*, Apr. 2004, vol. 1, pp. 819–823.

[5] J. P. Hespanha, M. L. McLaughlin, and G. Sukhatme, “Haptic collaboration over the Internet,” in *Proc. 5th Phantom Users Group Workshop*, Oct. 2000.

[6] K. Hikichi, H. Morino, I. Arimoto, K. Sezaki, and Y. Yasuda, “The evaluation of delay jitter for haptics collaboration over the Internet,” in *Proc. IEEE Global Telecomm. Conf. (GLOBECOM)*, Nov. 2002, vol. 2, pp. 1492–1496.

[7] S. Shirmohammadi and N. H. Woo, “Evaluating decorators for haptic collaboration over the Internet,” in *Proc. 3rd IEEE Int. Workshop Haptic, Audio and Visual Env. Applic.*, Oct. 2004, pp. 105–109.

[8] P. Seiler and R. Sengupta, “Analysis of communication losses in vehicle control problems,” in *Proc. 2001 Amer. Contr. Conf.*, Jun. 2001, vol. 2, pp. 1491–1496.

[9] P. Seiler and R. Sengupta, “An H_∞ approach to networked control,” *IEEE Trans. Automat. Contr.*, vol. 50, no. 3, pp. 356–364, Mar. 2005.

[10] R. M. Murray, K. J. Astrom, S. P. Boyd, R. W. Brockett, and G. Stein, “Control in an information rich world, ” *IEEE Contr. Syst. Mag.*, vol. 23, no. 2, pp. 20–33, Apr. 2003.

[11] M. G. Rivera and A. Barreiro, “Analysis of networked control systems with drops and variable delays,” *Automatica*, vol. 43, no. 12, pp. 2054–2059, Dec. 2007.

[12] Y. Ishido, K. Takaba, and D. E. Quevedo, “Stability analysis of networked control systems subject to packet-dropouts and finite-level quantization,” *Syst. & Control Lett.*, vol. 60, no. 5, pp. 325–332, May 2011.

[13] C. W. Cheng, C. L. Lai, B. C. Wang, and P. L. Hsu, “The time-delay effect of multiple-networks systems in NCS,” in *Proc. SICE Annu. Conf.*, Sep. 2007, pp. 929–934.

[14] C. L. Lai and P. L. Hsu, “Design the remote control system with the time-delay estimator and the adaptive smith predictor,” *IEEE Trans. Ind. Inform.* , vol. 6, no. 1, pp. 73–80, Feb. 2010.

[15] I. Pan, S. Das, and A. Gupta, “Tuning of an optimal fuzzy PID controller with stochastic algorithms for networked control systems with random time delay,” *ISA Trans.*, vol. 50, no. 1 , pp. 28–36, Jan. 2011.

[16] D. Zhao, C. Li, and J. Ren, “Fuzzy speed control and stability analysis of a networked induction motor system with time delays and packet dropouts,” *Nonlinear Anal.: Real World Applicat.*, vol. 12, no. 1, pp. 273–287, Feb. 2011.

[17] W. Du and F. Du, “Novel smith predictor and fuzzy control for networked control systems,” in *Proc. Asia-Pacific Conf. Inform. Process. (APCIP)*, July 2009, pp. 75–78.

[18] F. Du and W. Du, “RBF neural network control and novel smith predictor for networked control systems,” in *Proc. Asia-Pacific Conf. Inform. Process. (APCIP)*, July 2009, pp. 67–70.

[19] C. L. Lai, P. L. Hsu, and B. C. Wang, “Design of the adaptive smith predictor for time-varying network control system,” in *Proc. SICE Annu. Conf.*, Aug. 2008, pp. 2933–2938.

[20] C. L. Lai and P. L. Hsu, “Realization of networked control systems on Ethernet with varied time delay,” in *Proc. IEEE Int. Conf. Syst., Man, Cybern. (SMC)*, Oct. 2010, pp. 66–73.

[21] N. Vatanski, J. P. Georges, C. Aubrun, E. Rondeau, and S. L. J. Jounela, “Networked control with delay measurement and estimation,” *Control Eng. Practice*, vol. 17, no. 2, pp. 231–244, Feb. 2009.

[22] T. J. Ross, *Fuzzy Logic with Engineering Applications.* 2nd ed. John Wiley & Sons, Ltd., 2004, pp. 151–161.

[23] A. Mellit and S.A. Kalogirou, “Artificial intelligence techniques for photovoltaic applications: A review,” *Progress in Energy and Combustion Sci.*, vol. 34, no. 5, pp. 574–632, Oct. 2008.

[24] C. Hicks, “A genetic algorithm tool for optimising cellular or functional layouts in the capital goods industry,” *Int. J. Prod. Econ.*, vol. 104, no. 2, pp. 598–614, Dec. 2006.

[25] A. Messai, A. Mellit, A. Guessoum, and S.A. Kalogirou, “Maximum power point tracking using a GA optimized fuzzy logic controller and its FPGA implementation,” *Solar Energy*, vol. 85, no. 2, pp. 265–277, Feb. 2011.

[26] M. Y. Chow and Y. Tipsuwan, “Network-based control systems: A tutorial,” in *Proc. 27th Annu. Conf. IEEE Ind. Electro. Soc.*, 2001, pp. 1593–1602.

Pooya Hajebi (S’10) was born in Isfahan, Iran, in 1981. He received the B.S. degree in electrical engineering (Electronics) and the M.S. degree in electrical engineering (Communication Systems), both from Yazd University, Yazd, Iran, in 2005 and 2009, respectively. Currently, He is working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, Yazd University, Yazd, Iran. His research interests include Networked Control Systems, Fuzzy Systems, Neural Networks, Data Communication Networks, Digital Signal Processing, Biological Signal Processing, Digital Image Processing, Cellular Networks, Optimizations, Time Delay Systems and Real Time Systems.

Seyed Mohammad Taghi AlModarresi obtained his B.S. degree in Electronics Engineering and M.S. degree in Communication Systems, both from the Isfahan University of Technology, Isfahan, Iran. He also holds Ph.D. in Electronics (Intelligent Signal Processing) from University of Southampton, UK (Department of Electrical and Computer Science: ECS). He works at the Department of Electrical and Computer Engineering in Yazd University where he pursues his research interests in: (i) Networked Control Systems (NCS) (ii) Neuro-Fuzzy Networks (iii) Wireless Networks.

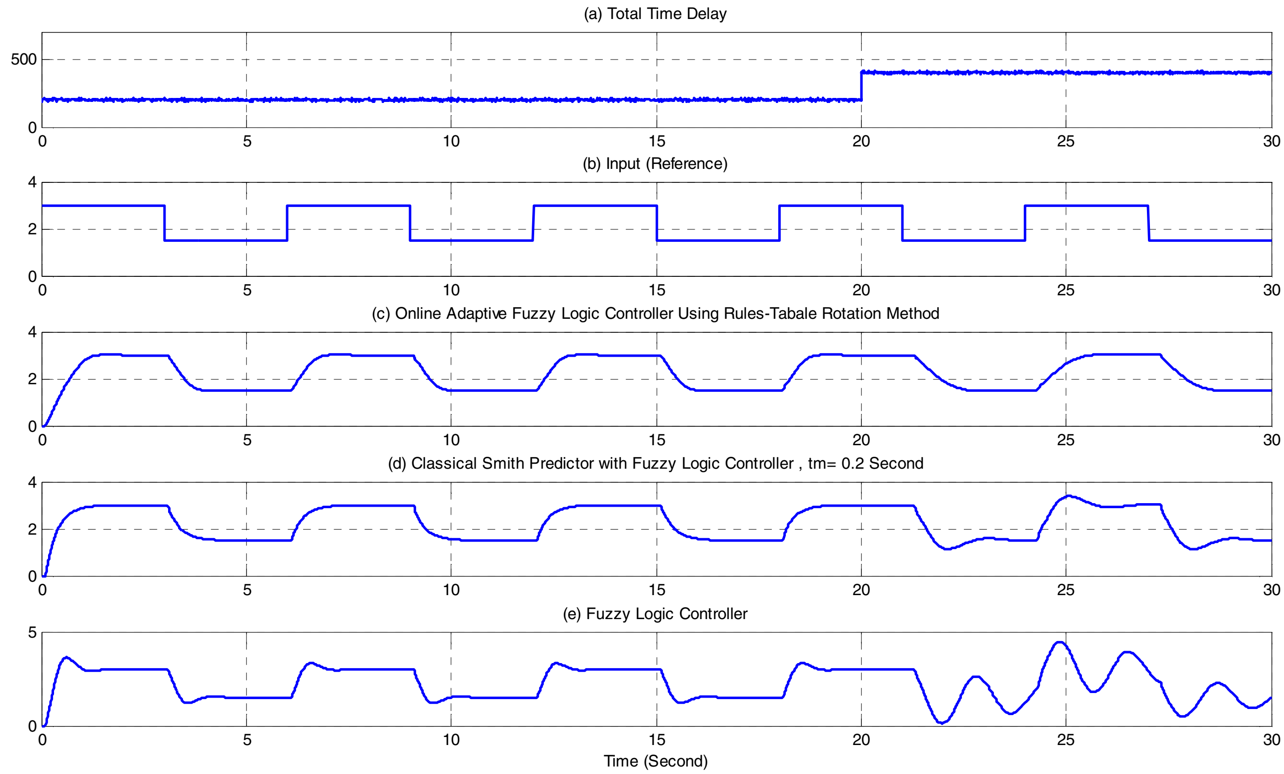

Fig. 12. Simulation results (The maximum time delay is about 400 ms); a) Time delay; b) Reference signal; c) Position (revolution of the motor shaft) for online adaptive fuzzy logic controller using rules-table rotation; d) Position (revolution of the motor shaft) for classical Smith predictor with fuzzy logic controller; e) Position (revolution of the motor shaft) for fuzzy logic controller.

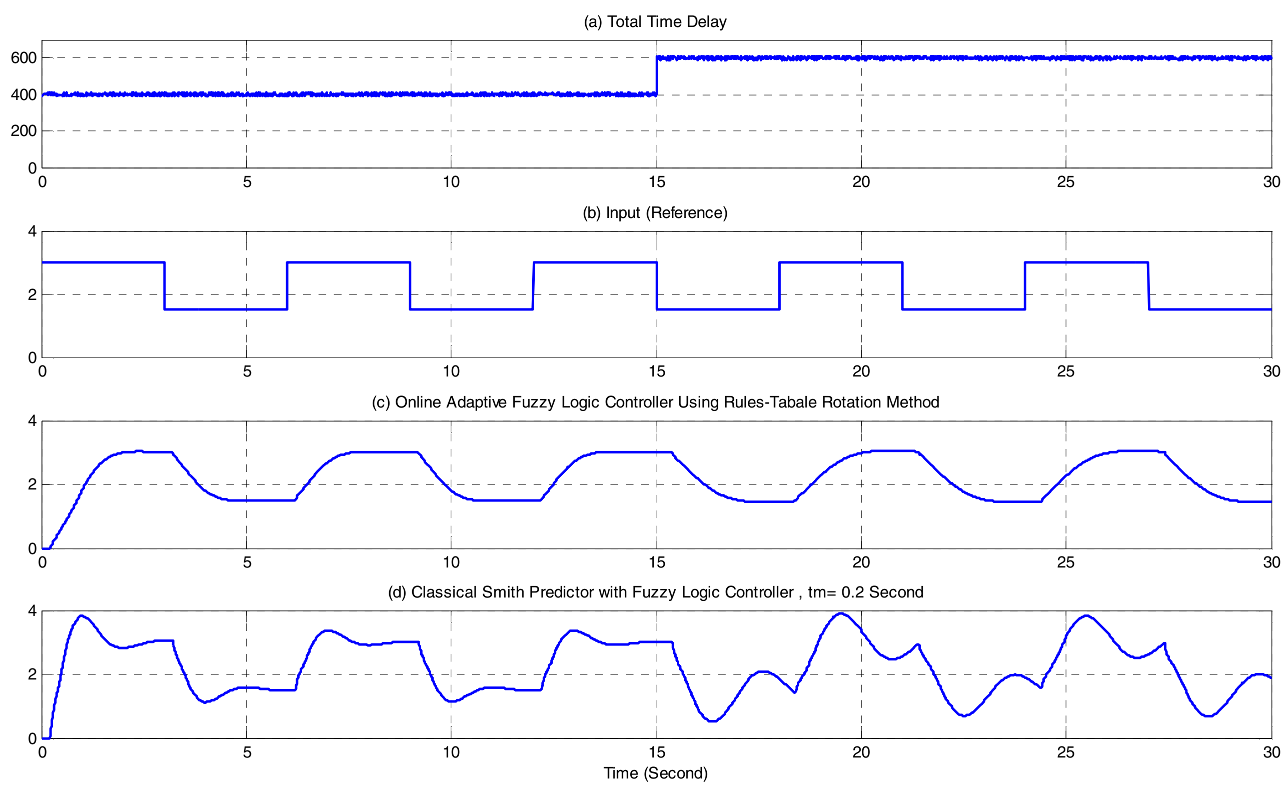

Fig. 13. Simulation results (The maximum time delay is about 600 ms); a) Time delay; b) Reference signal; c) Position (revolution of the motor shaft) for online adaptive fuzzy logic controller using rules-table rotation; d) Position (revolution of the motor shaft) for classical Smith predictor with fuzzy logic controller.

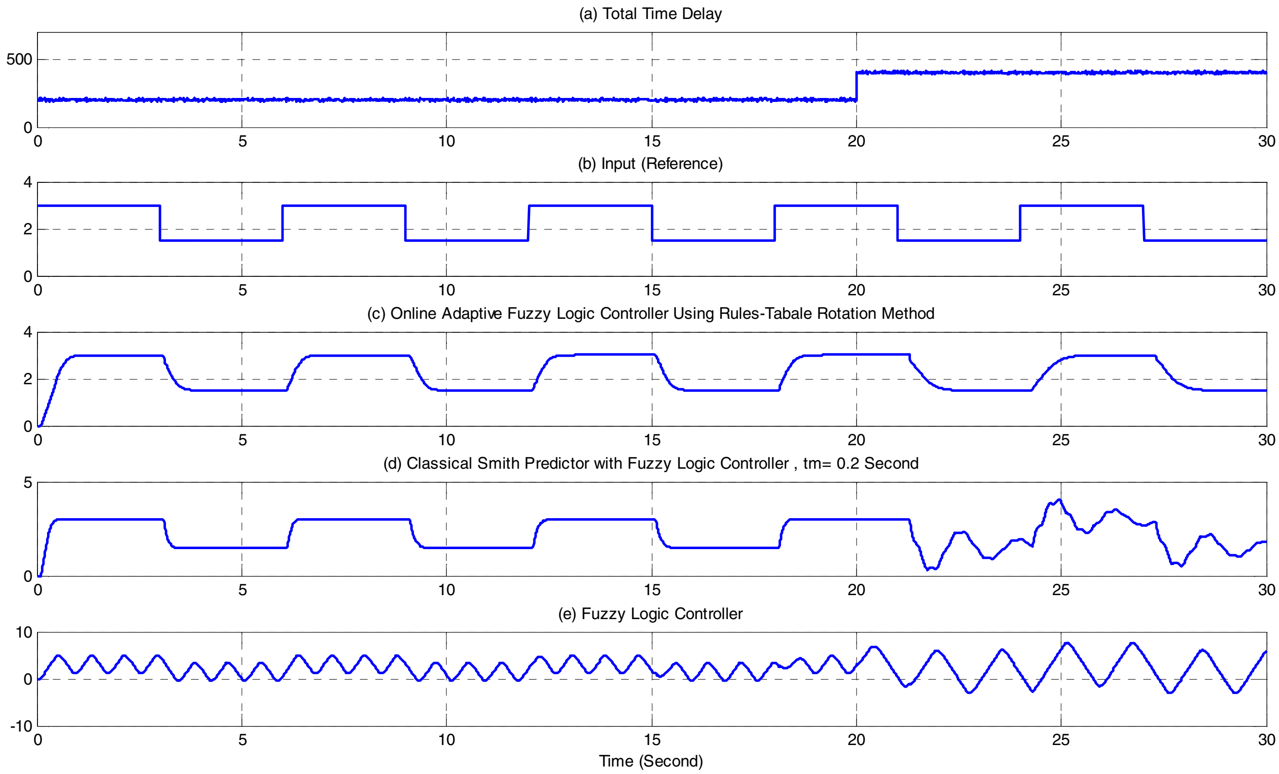

Fig. 14. Simulation results using genetic algorithm (The maximum time delay is about 400 ms); a) Time delay; b) Reference signal; c) Position (revolution of the motor shaft) for online adaptive optimized fuzzy logic controller using rules-table rotation; d) Position (revolution of the motor shaft) for Classical Smith predictor with optimized fuzzy logic controller; e) Position (revolution of the motor shaft) for optimized fuzzy logic controller.

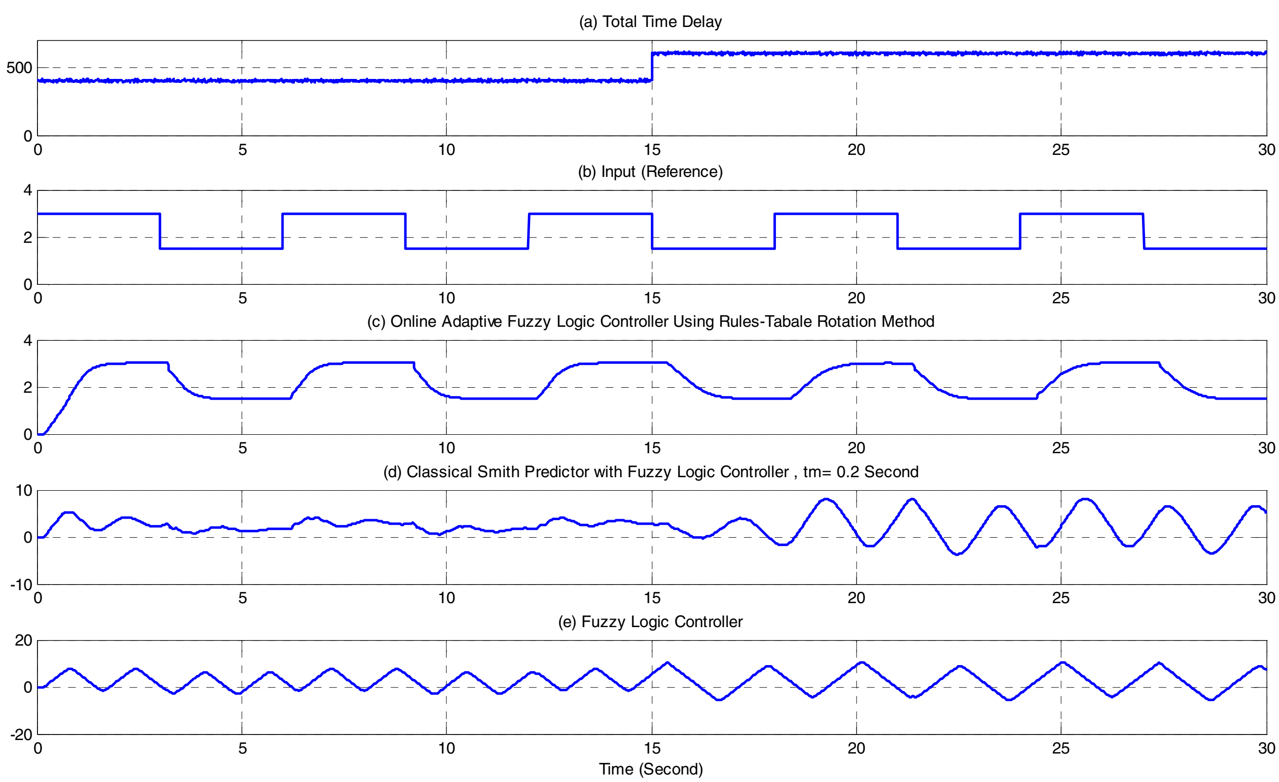

Fig. 15. Simulation results using genetic algorithm (The maximum time delay is about 600 ms); a) Time delay; b) Reference signal; c) Position (revolution of the motor shaft) for online adaptive optimized fuzzy logic controller using rules-table rotation; d) Position (revolution of the motor shaft) for classical Smith predictor with optimized fuzzy logic controller. e) Position (revolution of the motor shaft) for optimized fuzzy logic controller.

TABLE V
COMPARISON OF RESULTS

Approximate value of delay	Method	Optimization	T_r (s)	$P.O.$ (%)	ITAE	T_p (s)
20 ms	Online Adaptive FLC	Optimized	**0.1990**	**0**	**0.0429**	-
		Non-Optimized	0.8289	0	0.4959	-
	Classical Smith Predictor	Optimized	0.6863	0	0.2858	-
		Non-Optimized	1.0869	0	0.7152	-
	FLC	Optimized	0.2075	0	0.0436	-
		Non-Optimized	0.5899	0	0.2355	-
200 ms	Online Adaptive FLC	Optimized	**0.5062**	**0**	**0.3034**	-
		Non-Optimized	0.7876	0	0.5937	-
	Classical Smith Predictor	Optimized	0.2195	2.04	0.0972	0.5600
		Non-Optimized	0.6026	0	0.3355	-
	FLC	Optimized	0.2439	24.45	4.7879	0.5400
		Non-Optimized	0.2432	21.77	0.3322	0.6300
400 ms	Online Adaptive FLC	Optimized	**1.0146**	**0**	**1.2863**	-
		Non-Optimized	1.2394	0	1.6309	-
	Classical Smith Predictor	Optimized	0.1388	144.38	3.8189	0.8600
		Non-Optimized	0.3538	26.91	1.0856	1
	FLC	Optimized	0.1506	202.66	10.6089	0.8400
		Non-Optimized	0.2941	52.25	6.4426	0.9100
600 ms	Online Adaptive FLC	Optimized	**1.2878**	**0**	**2.2986**	-
		Non-Optimized	1.5202	0	2.6264	-
	Classical Smith Predictor	Optimized	0.3932	36.03	11.2648	1.1600
		Non-Optimized	0.2371	144.27	5.0326	1.3000
	FLC	Optimized	0.1379	343.67	16.8155	1.1400
		Non-Optimized	-	-	-	-