A Network Based Learning Architecture for Fuzzy Logic Controllers

B. Saeed[1] and B. Mehrdadi[2]

University of Huddersfield/School of Computing and Engineering, Huddersfield, UK

[1]Email: b.saeed@hud.ac.uk

[2]Email: b.mehrdadi@hud.ac.uk

***Abstract*—Tuning the parameters of fuzzy logic controllers is one of the most important parts of the design of these controllers and it has been extensively explored by researchers. Various techniques and algorithms have been utilised to fine tune the controller parameters. Designing controllers with the ability to retain and share the tuned parameters with other controllers have potential advantages on reducing the time required in tuning process. So far, however, there has been no research on the design of networked fuzzy controllers with the ability to retain the knowledge gained in tuning process and to provide a communication facility to enable the exchange of the acquired knowledge between controllers through a network. By expanding a previous work of the authors in auto-tuning fuzzy logic controllers, this paper proposes an original architecture for designing a network of fuzzy logic controllers with the capabilities of auto-tuning and sharing parameters. To improve the performance, each controller automatically and progressively tunes its parameters and retains the acquired knowledge in its memory for the future when a similar set-point is assigned to the controller. At the same time the acquired knowledge is shared with the rest of the controllers on the network where the controllers can benefit from it in tuning their parameters. The result demonstrate that this method has a substantial impact on minimising the time required to tune the controllers.**

***Index Terms*—fuzzy logic control, auto-tuning, learning control systems, artificial intelligent, multi-agent**

I. Introduction

One of the most significant application areas of fuzzy logic [1] is the field of control engineering. Since the first application [2], there has been an increasing number of literature on the design of fuzzy logic controllers (FLC) [3-5]. These controllers have been successfully applied in industrial processes and in some cases outperform conventional proportional-integral-derivative (PID) controllers [6, 7], in particular when the controlled system is complex or non-linear, as is the case in many process control systems [8-10].

FLCs have several parameters [11-13] such as membership functions, scaling gains and rule-base. Fine tuning these parameters has a great impact on improving the performance of these controllers. Therefore, researchers have been extensively exploring various methods and developing several techniques and tools from trail-error-method to very advanced optimisation techniques such as: Genetic Algorithms (GA) [14], Ant Colony Optimization algorithm (ACO) [15], Shuffled Frog Leaping Algorithm (SFLA) [16] and Bees Algorithm (BA) [17].

Additionally, in general, controllers have a wide range of set-points, changing from one set-point to another requires the controller parameters to be re-tuned to maintain and achieve a satisfactory performance and this would be a very time-consuming task.

However, despite the tremendous efforts made in tuning the controller parameters it remains limited and local to the controller without a framework to retain the knowledge for future use and to share this knowledge with similar controllers on a network. A suitable structure enables the controllers to learn from their own past experience and those of other controllers.

In order to design a controller that can learn from its past, the controller should be able to retain new tuned parameters in its memory. The retained knowledge can be used if the same set-point is assigned to the controller at a later time [8].

Learning from experience of other controllers can be achieved by forming a network of similar controllers controlling identical processes where they can share the knowledge gained by each controller. The learning process becomes more useful as the number of controllers connected to the same network increases.

In the literature, in a different context, concepts of close relevant works have been reported [18-22]. Reference [18] has described a learning algorithm for small autonomous mobile robots, where two robots learn how to avoid obstacles and the learned knowledge is passed to each other. An investigation of a multi-agent environment was carried out in [19], where the agents perform similar tasks and exchange information between them. The results showed a performance improvement and a faster learning rate of the individual agents.

So far, there has been no research about the design of networked fuzzy controllers with the ability to retain the knowledge gained in tuning process and to provide a communication facility to enable the exchange of the acquired knowledge through the network. Such a design provides a facility to controllers on a network to learn from each other through sharing the tuning knowledge. As a result, a significant amount of time can be saved when a new set-point is assigned to a controller.

In process control, there are situations where multiple identical devices form a wired or wireless network work together towards achieving a common goal. Each individual device or node is controlled by a master controller. In order to maximise the network performance, the nodes are designed to control their local systems in an intelligent manner so that

DOI: 03.CSS.2012.3.514

they can adapt to their set-point changes and tune their local parameters. The knowledge gained by individual controllers is then made available to other controllers on the network.

Potential applications of such system could be controlling multiple motors with identical characteristics or several pneumatic valves on an oil pipeline. These devices are known for their non-linear characteristics which are not easy to control without damping or slowing down their response to new set points.

An example of such an application is depicted in Fig. 1. Processes 1-4 are identical systems and are controlled by identical FLCs 1-4. The entire system is connected via a communication medium. Based on application requirements the communication medium could be Ethernet wired/wireless, ZigBee or any other type of network [23-25].

The controllers utilise a fuzzy logic control strategy to control their systems. To improve the performance they have the ability to auto-tune their parameters [6, 26]. In each iteration cycle of the tuning process the parameters are shared on the network. When a new set-point is assigned to a controller or a controller starts to perform the tuning process, it has the ability to perform its tasks in controlling and auto-tuning independently or to use the knowledge gained by other controllers. Thus the controllers could learn from the experience of each other, consequently a significant time can be saved during the tuning stage.

In a previous work [26], the authors described an auto-tuning fuzzy logic controller, whereby a new systematic auto-tuning algorithm to fine tune fuzzy logic controller gains is proposed and applied to several second order systems.

Based on the analysis of the relationship between the closed-loop response and the controller parameters, an auto-tuning method was implemented. The results showed the effectiveness of the algorithm and produced zero overshoot with enhanced transient response. In addition, the robustness of the controller was investigated in the case of parameter changes and the results demonstrated a satisfactory performance.

This paper proposes a networked learning structure for fuzzy logic controllers. A communication capability was integrated into the previous algorithms, where in an iterative manner the controllers can tune their parameters and share their acquired knowledge with the rest of the controllers on the network. Each controller has the ability to work independently or to start from the shared knowledge.

A second order system transfer function was used for simulation. The results show that the proposed method is highly effective.

The remainder of this paper is organised as follows: section II presents the design of the proposed networked fuzzy logic controllers. The learning algorithm is illustrated in section III. Evaluation and results are shown in section IV. Finally, some conclusions are drawn in section V.

II. Learning Architecture Design

NI LabVIEW 2011 was used to realise and implement the proposed network in Fig. 1.

The design was comprised of four PC-based FLCs connected through a dedicated Ethernet switch to ensure there is no other network traffic to disturb the communication of the controllers; the structure is shown in Fig. 2.

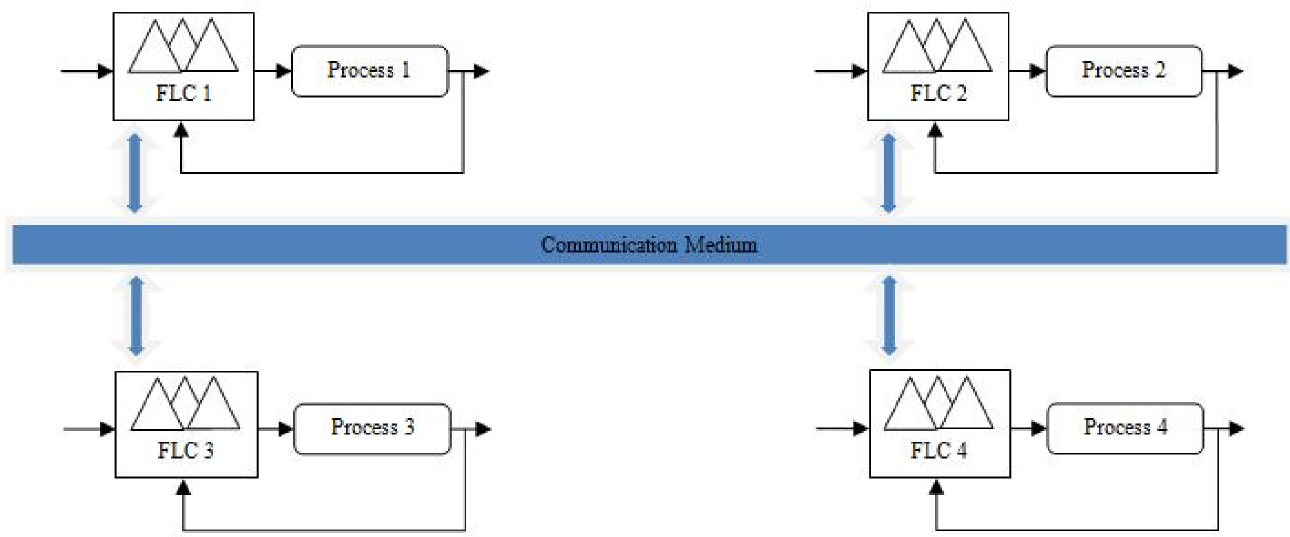

Figure 1. Structure of networked fuzzy logic controllers

The LabVIEW Control Design and Simulation module was used to design and simulate the FLCs and the learning algorithms, where the tuning process was accomplished in a flexible way. The LabVIEW "network-published shared variable" feature was used to establish the communication between the controllers, thus the complexities of communication programming were greatly minimised.

DOI: 03.CSS.2012.3.514

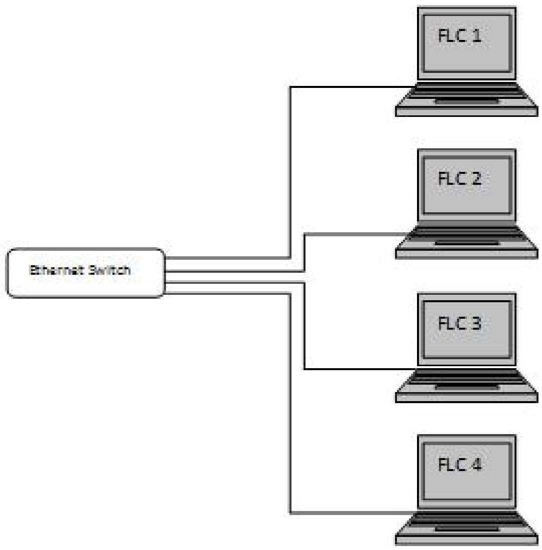

Figure 2. Learning architecture for fuzzy logic controllers

Each PC represents an independent FLC; Fig. 3 shows the simulation module of the FLC. The process transfer function can be defined in the simulation interface and the open-loop and closed-loop responses are displayed together with the response characteristics including the rise time, overshoot percentage and the settling time.

External disturbances could be introduced at the input and at the output of the process to determine the stability and robustness of the controlled system.

III. The Learning Algorithm Design

The learning algorithm was comprised of the auto-tuning algorithm in [26] with addition of new modules to retain the previous tuned parameters and to implement communication between the controllers. Thus, the new algorithms have three important capabilities of auto-tuning, memorising and sharing the acquired knowledge, which are integral parts of learning controllers [8]. Additionally, the algorithms have the ability to monitor the performance of the system and to ensure its stability.

The memorising module helps the controller to retain the tuned parameters derived for a given set-point.

The details of the above mentioned tasks can be summarised as follows: when a set-point is assigned to a controller, first it starts by checking the memory module for existence of previous tuning parameters for the given set-point. If previous tuning parameters for the set-point exist then the controller retrieves these values and continues the auto-tuning process from that stage. Otherwise the tuning process starts independently and the new tuning parameters are saved in the memory module and shared via the communication module. This sequence is repeated in every cycle.

The learning algorithm consists of several modules as shown by the flowchart in Fig. 4.

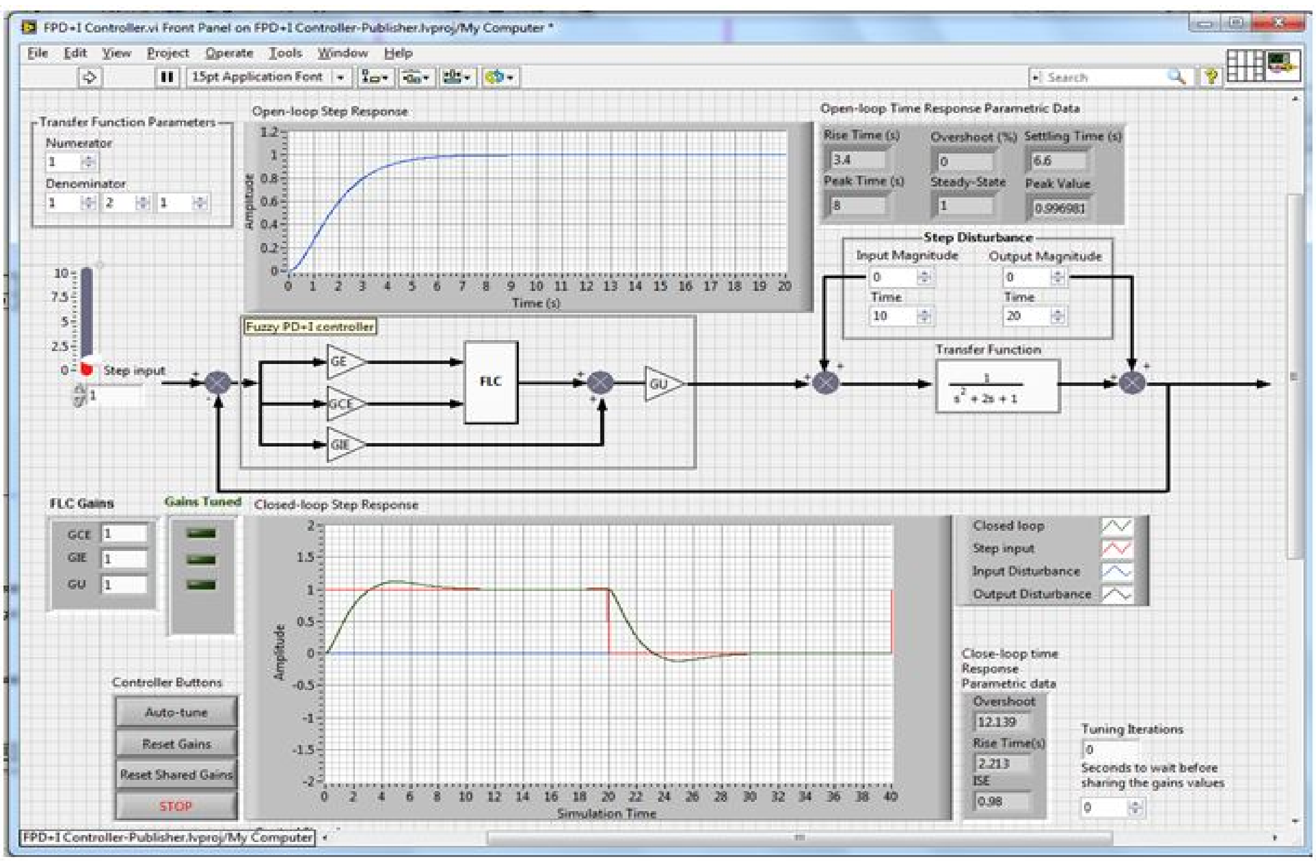

Figure 3. FLC simulation interface.

DOI: 03.CSS.2012.3.514

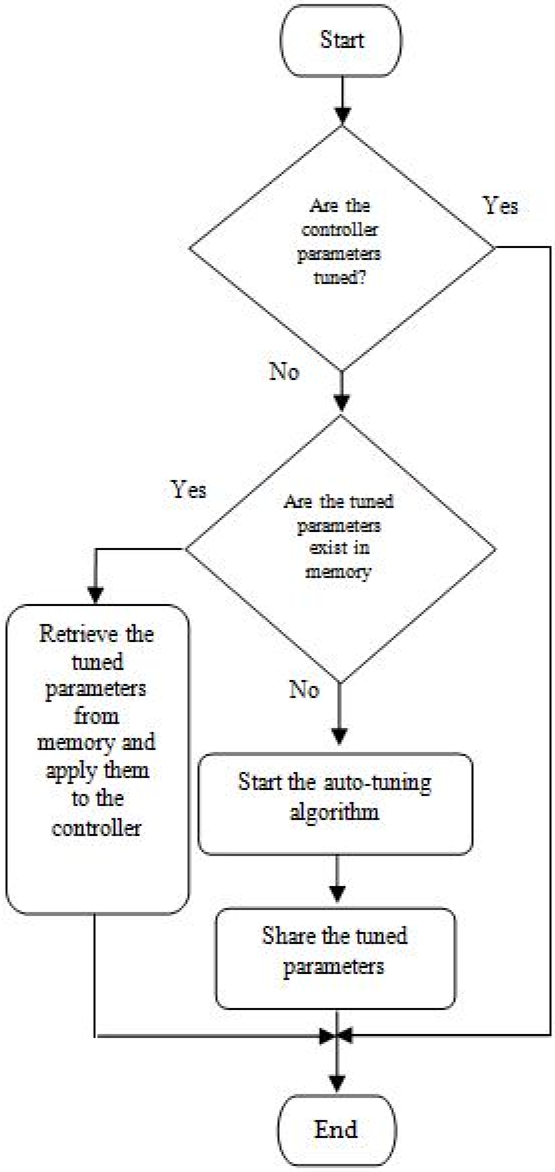

Figure 4. Flowchart of the learning algorithm

IV. Evaluation And Results

In order to test the algorithms, a standard second order system transfer function with different characteristics was used for the simulation. Due to limited space, only the response of one case (1) is presented here.

$$G(s) = \frac{1}{S^2 + 2S + 1} \quad (1)$$

The above case is a critically damped form of a second order transfer function and it is a very common low order system that could represent any two first order systems connected in series [27, 28].

All the controllers were setup to simulate the same system as in (1); the set-point of each controller was then assigned to each unit. The controllers were then initiated to operate as follows: FLC 1 at cycle 1, FLC 2 at cycle 4, FLC 3 at cycle 8 and FLC 4 at cycle 14. The closed-loop response results are shown in Fig. 5 and Fig. 6, where column 1, column 2, column 3 and column 4 show the responses of FLC 1, FLC 2, FLC 3 and FLC 4 respectively.

As can be seen from the first column of Fig. 5 and Fig. 6 the FLC 1 was able to automatically tune its parameters within 15 iterations. The overshoot was eliminated at the second iteration and the response was progressively improved. Furthermore, at each cycle, the tuning parameters were shared on the network.

For FLC 2, where it was started to operate at cycle 4, it was able to tune its parameters within 11 iterations and at its second iteration it achieved the same response as it was achieved by FLC 1. This is because it used the tuning parameters shared by FLC 1. Consequently, less cycle iterations were required to tune its parameters. This is apparent from the fifth row and the second column of Fig. 5.

As FLC 3 was started at cycle 8, it was able to tune its parameters within 8 iterations and at its second iteration the controller achieved the same performance as it was achieved by FLC 1 and FLC 2 and this is clear from the first row and the third column of Fig. 6.

Finally, FLC 4 was started at cycle 14. Therefore, it was able to tune its parameters and achieve a satisfactory response within only 2 iterations. This is shown in seventh row and fourth column of Fig. 6. Similar responses were achieved by FLC 1, FLC 2 and FLC 3 within 15 iterations, 11 iterations and 8 iterations respectively.

V. Conclusions

This paper has proposed and implemented a learning structure for fuzzy logic controllers. Algorithms with the capabilities of auto-tuning, retaining and sharing knowledge have been designed and applied to several second order systems. The controllers were able to retain and share their acquired knowledge, later on this knowledge was used by other controllers on the network. This has a great impact on reducing the time required to tune the controllers. Furthermore, over a period of time the accumulated knowledge acquired by the system enables the controllers to tune their respective parameters in a much shorter length of time.

Since the amount of the data exchanging between the controllers is small, hence embedded-microcontrollers interfaced to a ZigBee module can be utilised. Recently some microcontrollers are marketed with built-in ZigBee and sufficient internal program memory to incorporate our algorithms. These microcontrollers have several built-in Analogue to Digital Converters (ADC) and Pulse Width Modulation (PWM) outputs to realise a control loop. These microcontrollers are also physically small to be embedded in the sensor enclosures. This will provide new generation of applications that include versatile and cost-effective solutions in wireless sensor networks and control applications. The proposed structure is not limited to fuzzy controllers and can be adapted to incorporate other control strategies such as conventional PID control.

DOI: 03.CSS.2012.3.514

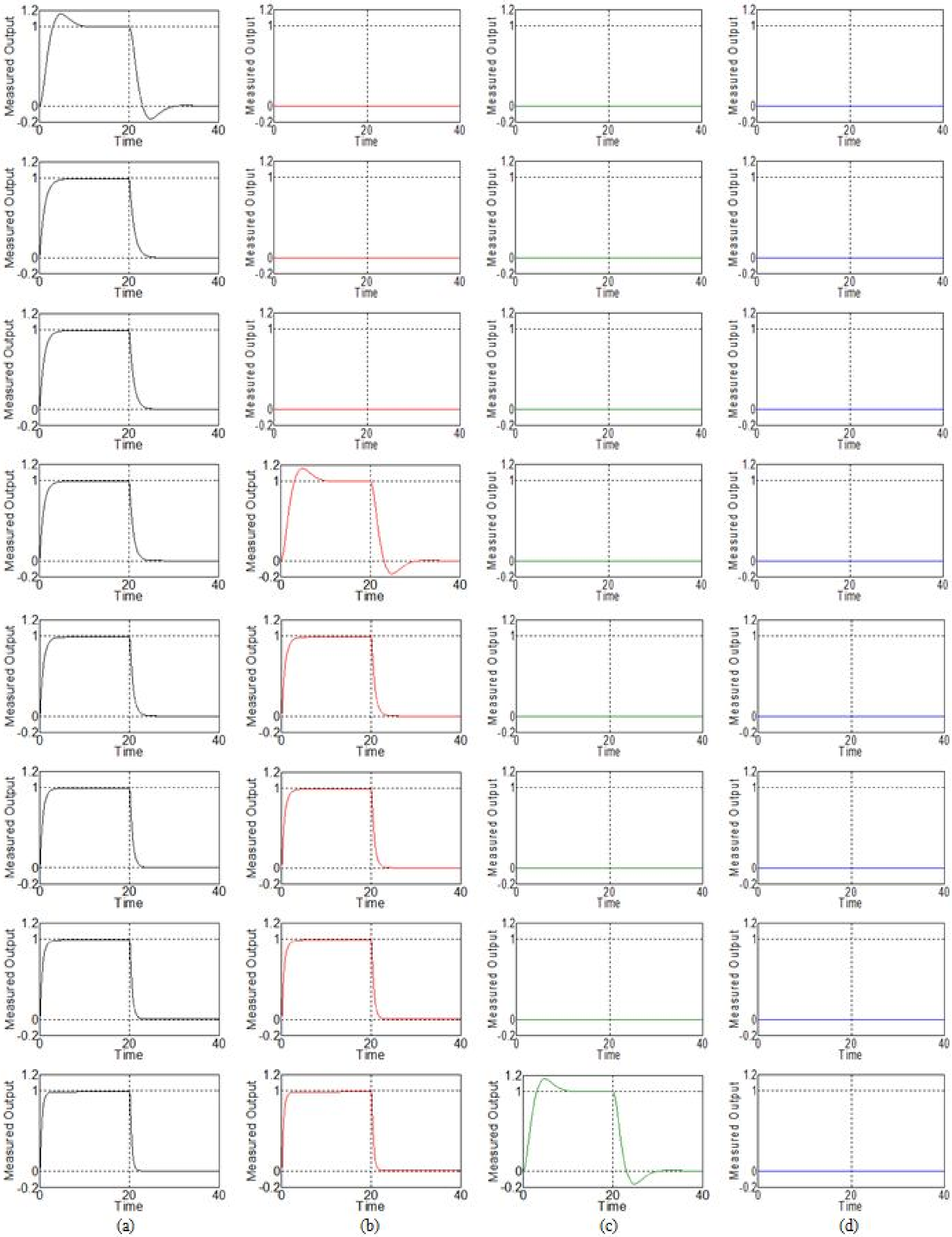

Figure 5. Iteration 1- iteration 8 of closed-loop step responses: (a) FLC 1 (b) FLC 2 (c) FLC 3 (d) FLC 4

DOI: 03.CSS.2012.3.514

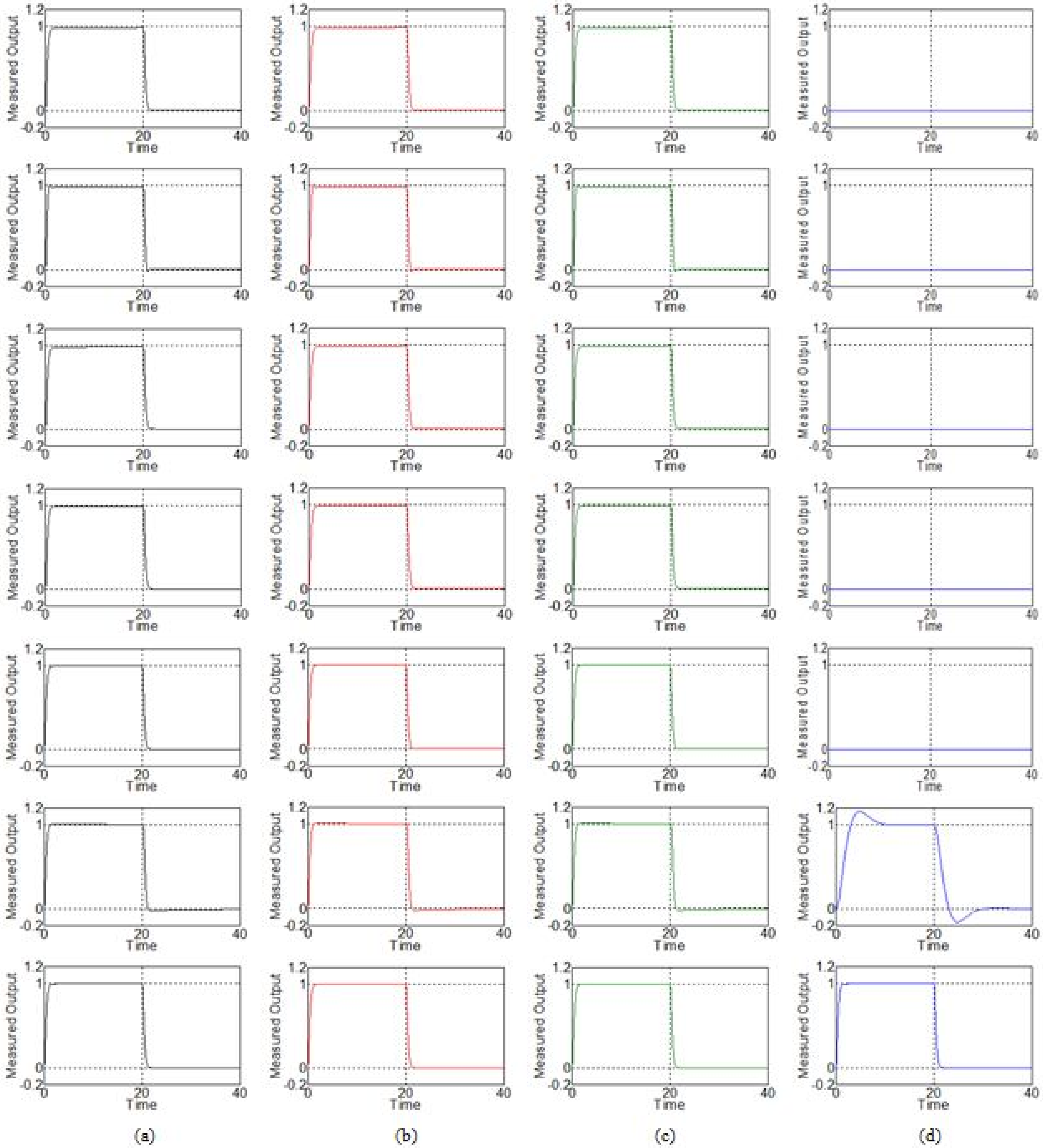

Figure 6. Iteration 9- iteration 15of closed-loop step responses: (a) FLC 1 (b) FLC 2 (c) FLC 3 (d) FLC 4

DOI: 03.CSS.2012.3.514

References

[1] L. Zadeh, "Fuzzy sets," *Information and control*. vol. 8, pp. 338-353, 1965

[2] E.H. Mamdani and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," *International Journal of Man-Machine Studies*. vol. 7, pp. 10-13, 1975

[3] V. Kumar, B. Nakra, and A. Mittal, "A Review on Classical and Fuzzy PID Controllers," *International Journal of Intelligent Control and Systems*. vol. 6, pp. 170-181, 2011

[4] Q. Shen, "Special issue on UK Fuzzy Systems Research," *International Journal of Computational Intelligence Research*. vol. 4, pp. 297–299, 2008

[5] R. Rajesh, "Fuzzy Logic Control - A Quick Review," *International Journal of Wisdom Based Computing*. vol. 1, 2011

[6] B. Saeed and B. Mehrdadi. "Zero overshoot and fast transient response using a fuzzy logic controller," *17th International Conference on Automation and Computing (ICAC)*, pp. 116-120, Huddersfield, UK, 2011

[7] S. Vaishnav and Z. Khan. "Design and performance of PID and fuzzy logic controller with smaller rule set for higher order system," *Proceedings of the World Congress on Engineering and Computer Science*, pp. 24-26, San Francisco, USA, 2007

[8] P.J. Antsaklis, *Intelligent Control*, in *Encyclopedia of Electrical and Electronics Engineering*, I. Control, Editor. 1999, John Wiley & Sons, Inc.

[9] R. Babuska and E. Mamdani (2008) *Fuzzy Control*.

[10] P. Ponce-Cruz and F. Ramírez-Figueroa, *Intelligent Control Systems with LabVIEW*. London: Springer, 2010

[11] K. Passino and S. Yurkovich, *Fuzzy Control*. Menlo Park, Calif: Addison-Wesley, 1998

[12] Z.-W. Woo, H.-Y. Chung, and J.-J. Lin, "A PID type fuzzy controller with self-tuning scaling factors," *Fuzzy Sets and Systems*. vol. 115, pp. 321-326, 2000

[13] J. Jantzen, *Foundations of Fuzzy Control*. Chichester: Wiley, 2007

[14] W.J. Tang and Q.H. Wu, "Biologically inspired optimization: a review," *Transactions of the Institute of Measurement & Control*. vol. 31, pp. 495-515, 2009

[15] C.-F. Juang and P.-H. Chang, "Designing fuzzy-rule-based systems using continuous ant-colony optimization," *IEEE Transactions on Fuzzy Systems*. vol. 18, pp. 138-149, 2010

[16] D.-H. Nguyen and T.-H. Huynh. "A SFLA-based fuzzy controller for balancing a ball and beam system," *10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 1948-1953, 2008

[17] D. Pham, A.H. Darwish, E. Eldukhr, and S. Otri. "Using the bees algorithm to tune a fuzzy logic controller for a robot gimnasta," *Innovative Production Machines and Systems Virtual Conference*, Cardiff, UK., 2007

[18] I.D. Kelly and D.A. Keating, "Faster learning of control parameters through sharing experiences of autonomous mobile robots," *International Journal of Systems Science* vol. 29, pp. 783-793, 1998

[19] A. Sch¨ollig, J. Alonso-Mora, and R. D'Andrea. "Independent vs. joint estimation in multi-agent iterative learning control," *49th IEEE Conference on Decision and Control (CDC)*, pp. 6949-6954, 2010

[20] R.G.-M. Jorge Ierache, Armando De Giusti, "Learning by collaboration in intelligent autonomous systems," *International Federation for Information Processing (IFIP) Advances in Information and Communication Technology (AICT)*. vol. 331, pp. 143-152, 2010

[21] H.-S. Ahny, Y. Chenz, and K.L. Moorex. "Multi-agent coordination by iterative learning control: centralized and decentralized strategies," *2011 IEEE International Symposium on Intelligent Control (ISIC)*, pp. 394-399, Denver, CO, USA, 2011

[22] W. Lu, Y. Zhang, and Y. Xie. "A multi-agent adaptive traffic signal control system using swarm intelligence and neuro-fuzzy reinforcement learning," *2011 IEEE Forum on Integrated and Sustainable Transportation System (FISTS)*, pp. 233-238, Vienna, Austria, 2011

[23] Y. Tipsuwan and M.-Y. Chow, "Control methodologies in networked control systems," *Control Engineering Practice*. vol. 11, pp. 1099-1111, 2003

[24] X. Yunfeng, Z. Yan, and M. Rui. "Ethernet-based network measure and control cystems cmbedded MCU," *2011 Fourth International Symposium on Computational Intelligence and Design (ISCID)*, pp. 71-73, 2011

[25] L. Yuqing and F. Huajing. "Control methodologies of large delays in networked control systems," *International Conference on Control and Automation*, pp. 1225-1230 Vol. 2, 2005

[26] B. Saeed and B. Mehrdadi. "Design of an iterative auto-tuning algorithm for a fuzzy PID controller " *25th International Congress on Condition Monitoring and Diagnostic Engineering*, Huddersfield, UK, 2012

[27] T.F. Edgar, D.E. Seborg, and D.A. Mellichamp, *Process Dynamics and Control*. Hoboken, N.J: Wiley, 2004

[28] A. Rachid and C. Scali, "Control of overshoot in the step response of chemical processes," *Computers & Chemical Engineering*. vol. 23, pp. S1003-S1006, 1999

DOI: 03.CSS.2012.3.514