# Model Free Fuzzy Adaptive Control For Networked Control Systems

Muhammad Bilal Kadri[1], Syed M K Raazi[2]

[1]Karachi Institute of Economics and Technology, Karachi, Pakistan.
[2]Muhammad Ali Jinnah University, Karachi, Pakistan.

Received: 10 February, 2019 / Revised: 16 April, 2019 / Accepted: 11 June, 2019

## ABSTRACT

In this work Model Free Fuzzy Adaptive Control (MFFAC) has been proposed for a networked environment. Network control architecture (NCA) has been designed which manages communication between the various components of the closed loop such as the controller, sensor(s) and actuator(s). The MFFAC communicates with the plant through the NCA which provides a seamless integration of the all the modules with the communication network. The complete network control system (NCS) has been tested on a laboratory test jig of a coupled tank system. The controller efficiently maintains the water level by adjusting the water flow rate through the output valves. In the presence of both unmeasured disturbances and network induced time delays, the controller is able to track the reference trajectory satisfactorily.

## KEYWORDS

Model free fuzzy control, adaptive control, network control system, coupled tank system.

## I. Introduction

Fuzzy logic controllers are able to control complex non-linear processes due to their global approximation property[1, 2]. If the plant parameters or the operating environment varies, controller has to be re-tuned. Adaptive fuzzy controllers have been proposed to adapt the controller parameters based on the plant operating data. The adaptation process might be offline or online[3]. Adaptive fuzzy controllers in which the controller parameters are updated offline might result in degraded control performance if the plant operating conditions changes or the plant is affected by unknown disturbances[4]. Online adaptive fuzzy controllers have the advantage of initializing the rectifying behavior once the plant output deviates from the reference trajectory. The adaptive algorithm updates the controller parameters online and the control performance improves. Online adaptive fuzzy controllers can be designed based on the direct or indirect adaptive control strategy.

Indirect control also known as model based control requires a plant model and the control law is dependent on the plant parameters[5, 6]. Control performance is heavily dependent on the availability of a good plant model. If the plant model cannot be identified the control performance is compromised. Another disadvantage with the indirect adaptive control scheme is the increase in computational load. The plant model has to be developed to guarantee acceptable control performance. Many of the complex industrial processes can be categorized as information poor systems, hence obtaining a plant model online often results in control performance deterioration[7].

The alternative strategy is the direct adaptive control or the model free control. In model free control no plant information is required a priori. The adaptive procedure directly updates the controller parameters[8, 9]. If the controlled variable deviates from the reference trajectory the adaptive algorithm updates the controller parameters based on the tracking error. Model free adaptive control schemes are best suited for information poor systems[10, 11]. Many different model free fuzzy adaptive controllers have been proposed in the control literature; some of the most cited works are by Dexter et.al [10, 12-14], Pomares et.al [15], Abonyi et.al [16, 17]and Sousaet.al [18]. All the different model free schemes build an inverse plant model of the process to be controlled[17, 19]. In the model free scheme proposed by Dexter et.al[13, 20] the inverse model is developed using feedback error learning mechanism[21-23].

Decentralized controls have received sustained interest in control research community. Architecture of decentralized systems may use a central computer acting as a controller along with sensors and actuators. The complete setup requires huge wiring connected from the sensors to computer and computer to actuators. Moreover this scheme becomes cumbersome on requirement of reconfiguring the physical setup and functionality. Diagnosis and maintenance are also difficult in such systems[24, 25]. To overcome aforementioned difficulties posed by the decentralized control, Networked Control System (NCS) has been proposed[26, 27]. With advances in control and communication technologies the implementation is conveniently realizable with the physical plant.

In NCS the controller, sensors, and actuators are geographically distributed requiring data to be transmitted from one location to another. Data from sensor, actuator and controller are transmitted over a network. A typical NCS configuration is shown in Figure 1. Sensor and actuator nodes sample data with period h, time duration required for this contributes a sampling delay of $\tau_S$ and $\tau_A$ seconds. Delay introduced by the control algorithm is $\tau_C$ and $\tau_{CA}$, $\tau_{SC}$ are the communication delays induced over a network. The combined effect of these distributed delays, is termed as sampling actuator delay $\tau$ and is represented by:

$$\tau = \tau_S + \tau_{SC} + \tau_C + \tau_{CA} + \tau_A \tag{1}$$

When a communication network is introduced in the feedback control two major problem arises;the first being that of packet dropout[28] and secondly network induced delays[29, 30]. The network induced delays i.e. the delay, in transmission of data packet from the controller to the actuator and from the sensor to the plant might prove to be hazardous for the closed loop stability[31, 32].

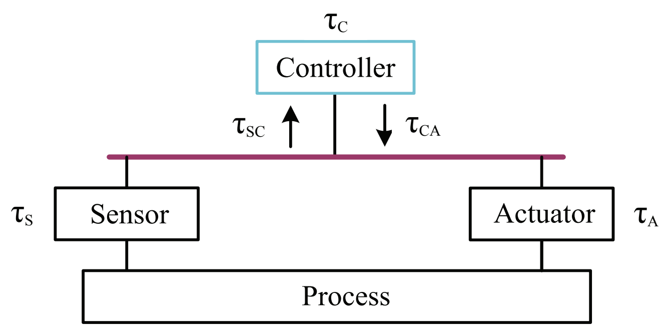


**Figure 1 General configuration of Networked Control System**

Information poor systems i.e. plants whose underlying mathematical model is difficult to obtain accurately,when connected to the controller through a network pose a challenging control problem. When information poor systems has to be controlled over a network the complexity of the control problem increases enormously due to the stochastic nature of the time delays involved in data transmission. Model free fuzzy adaptive control proposed by Dexter et.al [10, 20] has shown good control performance with information poor systems.

In this paper an extension of the model free fuzzy adaptive control (MFFAC)[12, 33] for networked environment has been proposed. The underlying network protocols remains unchanged whereas a comprehensive network control architecture (NCA) has been designed which communicates with the MFFAC, sensors and actuators. The NCA has been designed to communicate over any type of network. The NCA provides a seamless integration of the underlying network with the controller, sensor and actuator modules. In this work impact of varying the underlying network connection on the performance of the MFFAC has been investigated with a laboratory test jig of a coupled tank system (CTS). To establish the efficacy of the proposed scheme the control experiments have been conducted with various network communication channels .

The paper is organized as follows. Network control architecture is explained in section 2. Section 3 contains a detailed overview of the network based model free fuzzy adaptive controller. Control performance of the NCS based MFFAC is presented in section 4. Stability analysis of the proposed controller is given in section 5. Section 6 concludes the paper.

## II. Network Control Architecture

Network control architecture (NCA) is presented in Figure 2. System primarily consists of two main components Server and Client, which are hosted at two different machines. Client and Server are connected through a network or the internet. Sender and Receiver modules at both Client and Server carry out connection oriented data transfer between Client and Server. Apart from Sender and Receiver modules, Client has Sensor and Actuator modules, which sense and update states of the plant. Controller module, which is the main component of the server, processes sensor data and computes the control signal which is transmitted to the actuator.

All modules run in separate threads of execution. Data transfer between Client and Server is done through connection oriented TCP protocol, which maintains ordering of data. Problem of transferring data between threads at the same machine was mapped to producer-consumer problem with in-order delivery. Producer-Consumer problem is a classic problem of synchronization. The plant produces and accepts data in specific formats. Controller module checks the format of sensor data and screens out erroneous and anomalous input. Simultaneously the controller module generates the actuator data in the proper format.

Server module continuously listens for connection from any Client, which is connected to the plant to be controlled. At Client side, Sensor module gathers sensor data from the plant and sends it to the Server through Sender module at the Client side. Receiver module at the server side forwards the senor data to the Controller. Controller module at the server screens the input i.e. sensor data, processes it, and generates plant operating instructions. Plant operating instructions are sent to the connected Client through Sender module at the Server. Receiver module at the connected client forwards plant operating instructions to the Actuator module, which stimulates the actuator accordingly.

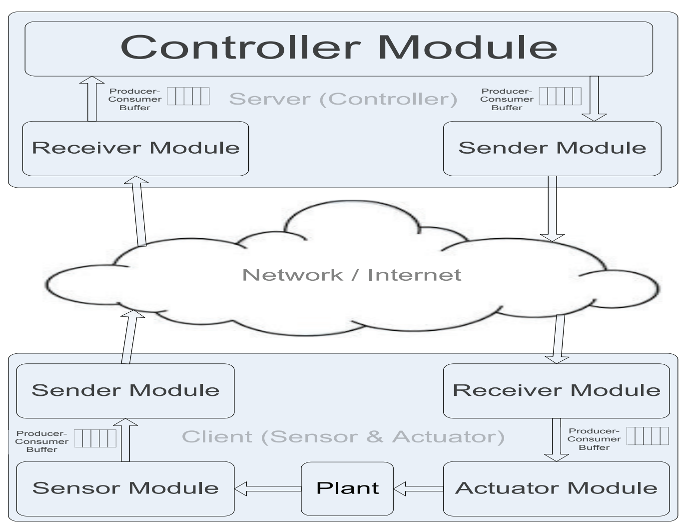


**Figure 2 Network control architecture**

Communication through network between Sender and Receiver modules are carried out using standard TCP/IP protocols, which guarantee in-order delivery of data. In order to guarantee in-order delivery of data within a component, queues (First In First Out (FIFO)

Producer-Consumer buffers) are maintained for transferring sensor and actuator data between each pair of the communicating modules. If no new data is sent to the plant from the actuator module, previous value of the data is maintained.

In the starting phase, Server (Controller) component starts accepting connections from prospective Clients. Normal working starts after all the connections are established and separate threads are created for every module. Sensor module senses a unit of plant data whenever data is available at the COM port. After sensing a unit of data, Sensor module sends it to the Sender module in the Client component through FIFO Producer-Consumer buffer using the following steps:-

1. Sensor module checks for empty space in FIFO Producer-Consumer buffer.
2. If space is available, it waits on a binary semaphore to gain control of the FIFO Producer-Consumer buffer. If space is not available, it blocks itself in this step until space is available in the FIFO Producer-Consumer buffer.
3. When it gains control of the FIFO Producer-Consumer buffer, it inserts the unit of data behind the last inserted value. Also, it increments the number of filled spaces and decrements the number of empty spaces.
4. After inserting the latest data, it releases control of the FIFO Producer-Consumer Buffer and all the related resources.

Sender component at the Client component works as follows:

1. Sender module checks for data (filled spaces) in FIFO Producer-Consumer buffer.
2. If data is available, it waits on a binary semaphore to gain control of the FIFO Producer-Consumer buffer. If data is not available, it blocks itself in this step until data is available in the FIFO Producer-Consumer buffer.
3. When it gains control of the FIFO Producer-Consumer buffer, it extracts the oldest inserted value. Also, it increments the number of empty spaces and decrements the number of filled spaces.
4. After extracting the oldest data, it releases control of the FIFO Producer-Consumer Buffer and all the related resources.

Communications between the following pairs of modules Figure 2 are done in the same way as described above: Receiver module and Controller module in the Server component, Controller module and Sender module in the Server component and Receiver module and Actuator module in Client component. The NCA is implemented as API (application programming interface) in C#.

## III. Network Based Model Free Adaptive Fuzzy Control

Model free fuzzy adaptive control (MFFAC) has been successfully tested on many applications including but not limited to cryogenic process[13, 34], cooling coil of an air handling unit [20] and coupled tank system. A comprehensive explanation of the MFFAC can be found in author's work in [14, 20].

The MFFAC is based on the model reference adaptive control (MRAC) strategy. It develops an inverse model of the plant by incorporating a special type of reinforcement leaning referred to as feedback error learning. The feedback error learning law is given in equation (2).

$$\tilde{u}_f(t-t_d) = u_f(t-t_d) + \gamma e(t) \tag{2}$$

where γ is the feedback error learning rate,

$\tilde{u}_f(t-t_d)$ is an estimate of the correct control action.

$u_f(t-t_d)$ is the incorrect control action that was produced 'td' samples ago which resulted in the tracking error e(t). A detailed block diagram of the MFFAC is shown in Figure 4. MFFAC is modeled using a $0^{th}$ order TS model. The affine form of the $0^{th}$ order TS Fuzzy model consists of rules $R_i$ with the following structure:

$$R_i \;:\; if\;\; \mathbf{x}\;\; is\;\; A_i\;\; then\;\; \mathbf{u}_i = w_i, \qquad i = 1,2,..,K \tag{3}$$

where, x is a vector of crisp inputs, Ai is a multidimensional fuzzy set, uiis the scalar output of the ith rule, wi are the controller parameters to be learnt by the adaptive algorithm and K is the number of rules in the rule base. The output of multi input single output (MISO) 0th order TS model can be described by:

$$u(\mathbf{x}) = \frac{\sum_{i=1}^{K} \mu_i(\mathbf{x}) u_i}{\sum_{i=1}^{p} \mu_i(\mathbf{x})} \tag{4}$$

where $\mu_i(\mathbf{x})$ is the degree of membership of x in the multi-dimensional fuzzy set Ai and $u(x) \in \Re^1$ is the control signal. The controller parameters are updated using a fuzzy identification scheme. The identification scheme is elaborated in the next section.

III.A. Fuzzy Identification Scheme

Fuzzy Least Mean Square (FLMS) Algorithm is used as the fuzzy identification scheme, it was proposed by Tan et.al [35]. The FLMS is based on the philosophy of Normalized Least Mean Square Algorithm (NLMS) and the recursive RSK fuzzy identification scheme [36]. The update equation for the controller parameters $\hat{w}(t)$ is given below:

$$\hat{w}(t) = \hat{w}(t-1) + \delta \frac{S(t-1)\mathrm{a}(t-t_d)}{\mathrm{a}^T(t-t_d)S(t-1)\mathrm{a}(t-t_d)} \varepsilon(t) \tag{5}$$

where $\delta$ is the update rate. The S(t) array consist of the cumulative strength a rule has been fired. The controller parameter is weighted by the element of the S(t) array i.e. sp.

$$S(t) = diag\{s_1, s_2, ......., s_i, ......., s_p\} \tag{6}$$

$$s_i = \prod_{j=1,\, j\neq i}^{p} F_j(t) \tag{7}$$

$$F_i(t) = F_i(t-1) + a_i(t) \tag{8}$$

$$a_i(x(t)) = \mu_{A_{1i_1}}(x_1(t)).\mu_{A_{2i_2}}(x_2(t)).....\mu_{A_{ni_n}}(x_n(t)) \tag{9}$$

where $a_i(x(t))$ is the strength with which the rule is fired

$$\mathrm{a}(t) = [a_1 \;\; a_2 \;\; ......a_p] \tag{10}$$

$$\varepsilon(t) = \tilde{u}_f(t) - \mathrm{a}^T(t-t_d)\hat{w}(t-1) \tag{11}$$

where, $\varepsilon(t)$ is the prediction error.

## III.B. Network based MFFAC

In this work MFFAC has been used in network control setting. The controller is placed on the server side and it accepts connection from multiple clients. The clients consist of the sensors and actuators. In this work the sensor and actuator are placed on one single client however the network model has the capability to establish communication between multiple clients and the server. The sensor and actuator threads run as independent modules. A basic block diagram of the MFFAC in a networked environment is shown in figure 3. The network introduces delays in the forward and feedback path of the closed loop system. These delays alongside with the sensor noise and measurement uncertainties might prove to be catastrophic for the control system.

The network model has been developed such that no data packet is lost. If there are unavoidable delays in communication the data packets are held in queues and they are prioritized based on their time stamp.

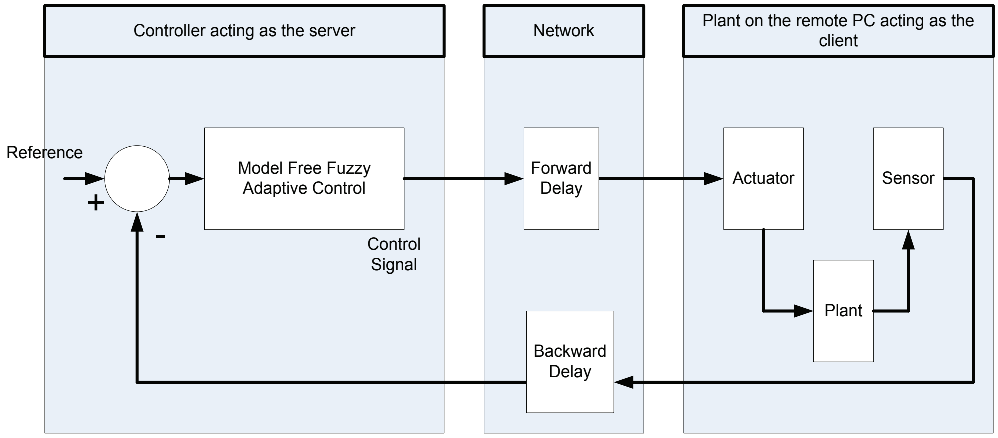


**Figure 3 MFFAC on a network**

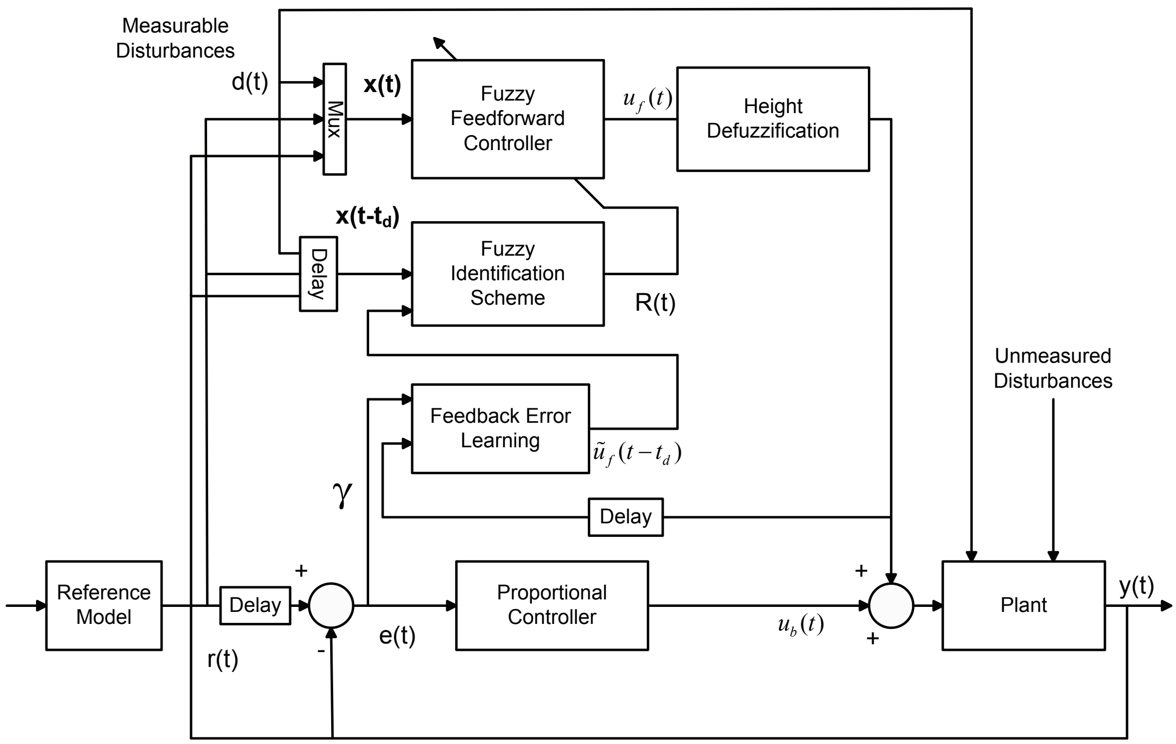


**Figure 4 Block diagram of Model Free Fuzzy Adaptive Controller**

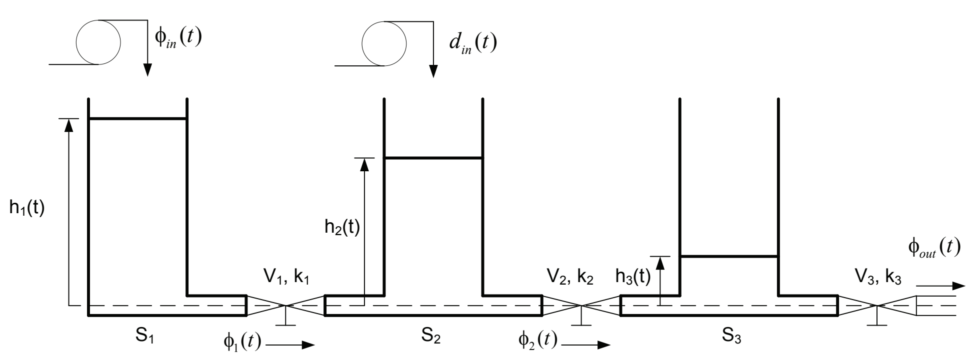


**Figure 5 Block diagram of the Coupled Tank System (CTS)**

## IV. Control Performance

In order to gauge the performance of the proposed control scheme, the controller was tested in three different scenarios on an experimental test jig of a coupled tank system.

## IV.A Experimental Setup

The input to the controller is x(t)=[r(t), y(t-td)] where y(t) is the plant output i.e. height of water in the middle tank. The block diagram of the CTS is shown in figure 5. Water flow rate $\phi_{in_1}(t)$ is the controlled input to the system whereas $h_2(t)$ i.e. height of water in the middle tank ($S_2$) is considered to be the controlled variable. , $\phi_1(t)$ and $\phi_{out}(t)$ are the drain of tank 1 ($S_1$), tank 2 ($S_2$) and tank3 ($S_3$) respectively. All of them are 50% open. The reference signal is a pulse which varies between 10 and 20 cm respectively and has a period of 2000 seconds with 50% duty cycle. The reference signal is filtered with a low pass filter having bandwidth of 10rad/sec. The setpoint r(t) and the controlled variable $h_2(t)$ are defined by five fuzzy sets distributed uniformly from 0 to 1.0. The values of update rate (δ) in FLMS and learning rate (γ) in feedback error learning are 0.01 and 0.5 respectively.The reference, control signal and the height $h_2(t)$ are normalized in the range [0,1].

## IV.B Case I: Control Performance on A LAN With Heavy Traffic Load

The client and the server PCs were on the same network and connected via a switch. The local network comprise of ten PCs all connected with the switch. The experiment was conducted during the daytime when the network experiences maximum traffic load. It can be observed from figure 6 that network based MFFAC is able to follow the reference trajectory. Initially the water level in the middle tank rises from 0 to approximately 16 cm. Since the controller is initialized with no a priori knowledge about the coupled tank system, the response overshoots during the first phase of the first cycle. The weights are initialized to 0.5, therefore the control signal starts from near 50 (half valve open) which steadily decreases. The control signal is quite smooth and does not cause excessive actuator movement. As the controller learns the behavior the control performance improves. During the second phase of the first cycle i.e. from 1000 to 2000 seconds there is a delay in the plant response which is due to the network traffic. The plant response rises above 20 cm but the overshoot is comparatively much less as compared to the previous phase. As the controller learns the behavior the tracking performance improves significantly. There are no overshoots and undershoots after 2000 seconds i.e. after one complete learning cycle. The delay in the plant response when the set point changes from 10 cm to 20 cm at 3000 seconds and 5000 seconds is due to the large time constant of the plant.

Table 1 Dimensions Parameters of the Coupled Tank System (CTS)

| | |
|---|---|
| **Dimensions** | |
| Height | 60cm |
| Width | 30cm |
| Length | 12cm |
| Dimension of each tank | 60cm x 9cm x 10.4cm |
| **Communication** | |
| Pump speed control | UART |
| Level sensors | UART |
| **Pump Speed control** | |
| Range | 0 – 100 (expressed as percentage of maximum flow rate) |
| Minimum step size for pump speed control | 1 % |
| **Level sensors** | |
| Range | 0 – 60m |
| Sensitivity | 1 mm |

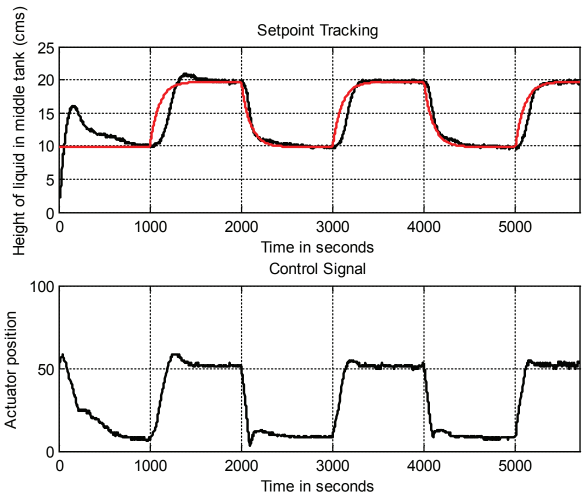


**Figure 6 Control performance over a LAN**

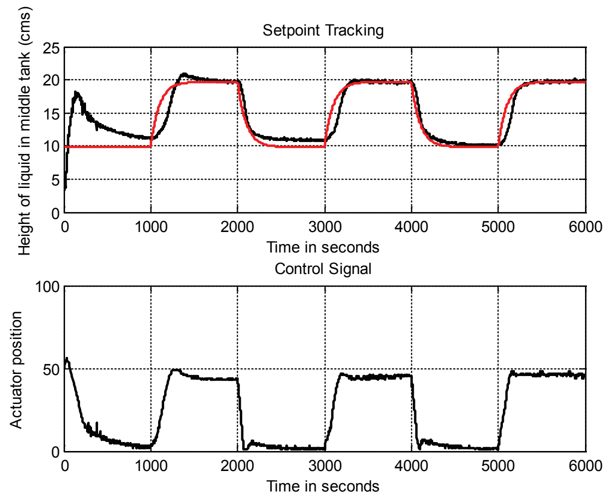


**Figure 7 Control performance with Ethernet internet connectivity**

## IV.B Case II: Control Performance Over the Internet With Ethernet Connectivity

In the second case, both the server and client PCs were assigned static IPs. All the data packets have to pass through the internet gateway. The experiment was conducted during the day time when the ISP (internet service provider) experiences heavy internet traffic load. Due to the latency in the communication between the plant (client) and the controller (server) the plant response is unable to converge to 10 cm during the initial phase. There is some high frequency noise in the plant response as well as on the control signal. When the set point changes from 10 cm to 20 cm the height of water in the middle tank overshoots the desired level but finally settles to 20 cm. During the first phase of the second cycle i.e. between 2000 and 3000 seconds the controller is unable to remove the steady state error. It can be observed from figure 7 that network based MFFAC offers tight control performance during the third cycle i.e. from 5000 seconds to 6000 seconds. There are no overshoots and undershoots and the steady state error is also eliminated. This experiment demonstrates the efficacy and robustness of the control scheme in an internet based communication.

## IV.C Case III: Control Performance Over The Internet With Wi-Fi Connectivity

In the third and final case, the server PC was assigned a static IP whereas the client was connected to the internet with a Wi-Fi connection. The data transmission between the two nodes is severely hampered by the concrete walls between the client (consisting of the sensor and actuator module) and the Wi-Fi router. Similar to the previous two cases, the water level overshoots quite a lot during the first phase of the first cycle i.e. from 0 to 1000 seconds. During the second phase of the first cycle, it can be observed from figure 8 that there is a significant delay in plant response. The water level remains at 10 cm for more than 30 seconds and then starts rising. The delay is caused by the Wi-Fi connection and packet losses. During the second cycle there are no overshoots and undershoots. Although the plant response is delayed during the second phase of the second cycle but the delay is much less as compared to the initial cycle. As evident from the response plot that MFFAC is able to follow the set point and effectively compensates for the network delay.

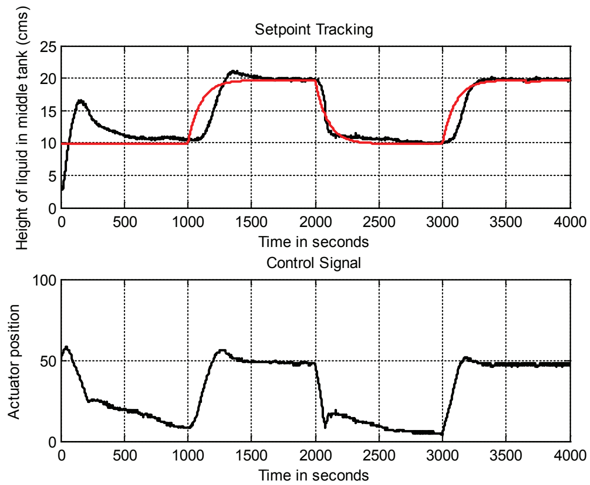


**Figure 8 Control performance with Wi-Fi internet connectivity**

## V. Stability Analysis

The detailed stability analysis of the MFFAC has been presented in [20]. When MFFAC is employed in a networked environment the stability of the closed loop might be affected due to the delays in the network. The network control architecture discussed earlier has been designed on a producer consumer model with queues at receiving end. None of the data is lost or rejected. If the controller is busy processing the delayed samples, all new samples are stored in queues. If all the data is consumed by the controller and there is none to be processed in the queues, the controller holds the control signal until new sensor data is available on the queue.

## VI. Conclusion

Model free fuzzy adaptive controller (MFFAC) has been proposed for a networked environment. A comprehensive network control model has been established which guarantees error free delivery of the data packets and ensure consistency. In this work the controller is placed on the server machine whereas the sensor and actuator modules are placed on the client side. One of the major challenges for control systems, in a networked environment is to guarantee stability of the closed loop system. Due to the communication delays and network overhead control performance deteriorates significantly. It has been successfully demonstrated that MFFAC is able to offer tight control performance with good delay compensation in a networked environment. The MFFAC is tested on a laboratory test jig of a coupled tank system. Different cases were considered, which included testing on a local area network as well as over the internet both wired and wireless medium. The experimental results from the networked based MFFAC has shown the effectiveness of the proposed controller as well as the network model to guarantee good control performance.

## References

[1] D. Driankov, H. Hellendoorn , and M. Reinfrank, An introduction to fuzzy control. Berlin: Springer-Verlag, 1993.

[2] K. M. Passino and S. Yurkovich, Fuzzy Control: Addison Wesley Longman, 1998.

[3] M. Athans, D. Stein, and L. Valavani, "Nonlinear and Adaptive Control," Massachusetts Institute of Technology, Cambridge,MA 02139June 30,1987 1987.

[4] A. B. Rad and W. F. Xie, "Fuzzy adaptive internal model control," IEEE Transaction IE, vol. 47, pp. 193-202, 2000.

[5] Y. Wu and A. L. Dexter, "Adaptive Fuzzy Model Based Predictive Control using Fuzzy Decision Making," in Proceedings of UKACC, Manchester, UK, 2008.

[6] C. G. Moore and C. J. Harris, "Indirect Adaptive Fuzzy Control," International Journal of Control, vol. 56, pp. 441-468, 1992.

[7] P. Angelov and C. Xydeas, "Fuzzy Systems Design: direct and indirect approaches," Soft Computing, vol. 10, pp. 836-849, 2006.

[8] M. V. Kumar, S. Suresh, S. N. Omkar, R. Ganguli, and P. Sampath, "A direct adaptive neural command controller design for an unstable helicopter," Engineering Applications of Artificial Intelligence, vol. 22, pp. 181-191, 2009.

[9] L. S. Coelho, M. W. Pessoa, R. R. Sumar, and A. A. R. Coelho, "Model Free adaptive control design using evolutionary-neural compensator," Expert Systmes with Applications, vol. 37, pp. 499-508, 2010.

[10] A. L. Dexter, Monitoring and Control of Information-Poor Systems. Oxford: Wiley, 2012.

[11] R. Thompson, "Neuro-Fuzzy Predictive Control of an Information-Poor System," D.Phil Thesis,University of Oxford, 2002.

[12] M. B. Kadri and A. L. Dexter, "Fuzzy Relational Control of Uncertain Systems," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 22, pp. 243-261, 2014.

[13] W. W. Tan and A. L. Dexter, "Self-learning neurofuzzy control of a liquid helium cryostat," Control Engineering Practice, vol. 7, pp. 1209-1220, 1999.

[14] M. B. Kadri, "Disturbance Rejection using Fuzzy Model Free Adaptive Control (FMFAC) with Adaptive Conditional Defuzzification Threshold," Journal of the Franklin Institute, vol. 351, pp. 3013-3031, 2014.

[15] I. Rojas, H. Pomares, J. Gonzalez, L. J. Herrera, A. Guillen, F. Rojas, et al., "Adaptive fuzzy controller: Application to the control of the temperature of a dynamic room in rael time," Fuzzy Sets and Systems, 2006.

[16] S. Molov, R. Babuska, J. Abonyi, and H. B. Verbruggen, "Effective optimization for fuzzy model predictive control," IEEE Transaction on Fuzzy Systems, vol. 12, pp. 661-675, 2004.

[17] J. Abonyi, H. C. Andersen, L. Nagy, and F. Szeifert, "Inverse fuzzy process model based direct adaptive control," Mathematics and Computers in Simulation, vol. 51, pp. 119-132, 1999.

[18] J. M. d. C. Sousa, A Fuzzy Approach to Model-Based Control. Lisbon,Portugal, 1998.

[19] R. Boukezzoula, S. Galichet, and L. Foulloy, "Nonlinear internal model control: application of inverse model based fuzzy control," IEEE Transaction on Fuzzy Systems, vol. 11, pp. 814-829, 2003.

[20] M. B. Kadri, "Disturbance rejection in information poor systems using model free neurofuzzy control," D.Phil Thesis, University of Oxford, 2009.

[21] M. B. Kadri, "Robust Model Free Fuzzy Adaptive Controller with fuzzy and crisp feedback error learning schemes," in 12th International Conference on Control, Automation and Systems (ICCAS), 2012, Jeju, South Korea, 2012, pp. 384 -388.

[22] K. Sabahi, M. Teshnehlab, and M. A. Shoorhedeli, "Recurrent fuzzy mneural network by using feedback error learning approaches for LFC in interconnected power system," Energy Conversion and Management, vol. 50, pp. 938-946, 2009.

[23] W. W. Tan and C. H. Lo, "Development of Feedback Error Learning Strategies for Training Neurofuzzy Controllers On-Line," IEEE International Fuzzy Systems Conference, pp. 1016-1021, 2001.

[24] M. Das, A. Banerjee, R. Ghosh, B. Goswami, R. Balasubramanian, A. Chandra, et al., "A study on multivariable process control using message passing across embedded controllers," ISA Transactions vol. 46, pp. 247-253, 2007.

[25] M. Das, R. Ghosh, B. Goswami, A. Chandra, R. Balasubramanian, P. Luksch, et al., "Multi-loop networked process control: a synchronized approach," ISA Transactions vol. 48, pp. 122-131, 2009.

[26] (2001) Special Section on Networks and Control. IEEE Control System Magzine.

[27] J. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," Proceedings of the IEEE 95, pp. 138-162, 2007.

[28] F.-L. Qu, Z.-H. Guan, D.-X. He, and M. Chi, "Event-triggered control for networked control systems with quantization and packet losses," Journal of the Franklin Institute, vol. 352, pp. 974-986, 2015.

[29] W. Zhang, M. S. Branicky, and S. M. Phillips, "Stability of networked control system," IEEE Control Systems Magazine vol. 21, pp. 84-99, 2001.

[30] S. Li, D. Sauter, and B. Xu, "Co-design of event-triggered image control for discrete-time linear parameter-varying systems with network-induced delays," Journal of the Franklin Institute, vol. 352, pp. 1867-1892, 2015.

[31] Z. Dezong, L. Chunwen, and R. Jun, "Fuzzy speed control and stability analysis of a networked induction motor system with time delays and packet dropouts," Nonlinear Analysis: Real World Applications, vol. 12 pp. 273-287, 2011.

[32] 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 Transactions vol. 50, pp. 28-36, 2011.

[33] W. W. Tan and A. L. Dexter, "A self-learning fuzzy controller for embedded applications," Automatica, vol. 36, pp. 1189-1198, 2000.

[34] M. Santos and A. L. Dexter, "Temperature control in liquid helium cryostat using self-learning neurofuzzy controller," IEE Proceedings Control Theory Application, vol. 148, pp. 233-238, 2001.

[35] W. W. Tan, "Self-learning neurofuzzy control of non-linear systems," D.Phil Thesis, University of Oxford, 1997.

[36] B. E. Postlethwaite, "Building a model-based fuzzy controller," Fuzzy Sets and Systems, vol. 79, pp. 3-13, 1996.