IECON'01: The 27th Annual Conference of the IEEE Industrial Electronics Society

Network-Based Controlled DC motor with Fuzzy Compensation

Naif B. Almutairi nbalmut@eos.ncsu.edu
Mo-Yuen Chow chow@eos.ncsu.edu
Yodyium Tipsuwan ytipsuw@eos.ncsu.edu
Electrical and Computer Engineering Department
Advanced Diagnosis and Control Laboratory
North Carolina State University
Raleigh, NC 27695-7911

Abstract— **This paper introduces a fuzzy logic compensation approach for the network-based controlled DC motor. The approach is based on modulating the control signal provided by the central controller in the network-based system with a single parameter. The inputs to the fuzzy compensator are the error signal between the reference and the actual output of the DC motor, and the output signal from the PI controller. The output from the fuzzy compensator is the modulator parameter. This parameter will modulate the output of the PI controller in order to reduce the effect of the network delays and different available bandwidth. The results show that using fuzzy compensation can be an effective approach to improve the performance of the DC motor controlled through the network.**

I. Introduction

Due to the communication delay concerns, the techniques used to control network-based systems have to maintain the performance and the stability of the systems in addition to controlling the plant. Several methods have been formulated differently based on multiple types of network behaviors and configurations in conjunction with several ways to treat delay problems in network-based control systems.

A widely used fundamental idea to formulate many network-based control methods is to use state augmentation. The main concept of state augmentation is to derive additional state-space equations to cooperate with the original state-space equations of a system. This approach has been used in discrete-time systems with constant delays for several years [3]. Several researchers have used the state augmentation approach as the first step to developing their own techniques ([4], [5], [6]-[8]). The number of additional states is typically proportional to the number of original states, inputs, and delays under consideration. This method therefore can substantially increase the system complexity, especially for high-dimensional systems, even thought the formulation is mathematically straight forward. In addition to the state augmentation approach, several other techniques have been developed to handle network delays and can be found in ([9], [10], [5], [8]).

Several assumptions have been made in the aforementioned approaches in order to develop generic network-based control methods. Some of these assumptions are: network transmissions are error-free, all data messages have the same length, the time skew is constant, the computational delay is constant and is much less than the sampling period T, network traffic cannot be overloaded, and every dimension of output measurements or control inputs is packed into one single data message.

Some of these assumptions, such as the first assumption, are obviously not practical. Hence, some control methods utilizing unrealistic assumptions may not be applicable for actual applications. Nevertheless, several important phenomena can be observed, and many ideas can be obtained from simulations and experimental setups under these assumptions. These benefits are substantially useful for future developments of network-based control systems.

Relaxation of some assumptions has been reported in some control techniques ([14], [11]). On the other hand, some techniques may require additional assumptions ([10], [5], [8], [12]) such as the sum of sensor-to-controller delay and controller-to-actuator delay is less than the sampling period.

A delay compensation method for network-based control systems with random delays was proposed in ([6], [13]). The main idea of delay compensation is to utilize an observer to estimate the plant states and a predictor to compute predictive control inputs based on past output measurements. Another predictor-based method for network-based control systems with random delays by using a probabilistic approach along with knowledge of the data lengths in a queue to improve the prediction is presented in [9].

An optimal stochastic control approach for control of a system plant over a random delay network is shown in [10]. The effects of communication delays in network-based systems were treated as a Linear-Quadratic-Gaussian (LQG) problem.

The perturbation method was introduced in ([7], [12]). A time-driven sensor, an event-driven controller, and an event-driven actuator are utilized as the basic system components in this method. Only data messages from a sensor can be transmitted through a network. A control loop in this method consists of a nonlinear controller and a nonlinear plant. However, the similar analysis and derivations used in this approach can be applied on linear systems as described in [12].

The sampling time scheduling method was introduced in [5]. The fundamental concept of this approach is to appropriately select a long enough sampling period for a discrete-time network-based system such that communication delays do not affect the control performance, and the system remains stable. In this case, the control delay in a discrete-time control loop must be assumed to be less than the sampling period T of the loop. The control loop consists of a time-driven sensor and controller, and an event-driven actuator. This method can be applied on cyclic service networks, in which all system component connections on the network are known in advance. This approach does not only solve the periodic delay problems, but can also increase network utilization in the case where there are multiple control loops on the same network.

This paper proposes the use of fuzzy logic ([15], [2]) to modulate the system control gain in order to compensate for the time delay problems in the network-based control systems. A network-based controlled DC motor, which is one of the popular devices for actuation and propulsion systems, is used in this paper to illustrate the effectiveness of the proposed scheme.

II. Problem Formulation

A general block diagram of the network distributed control system is shown in figure(1). The network distributed

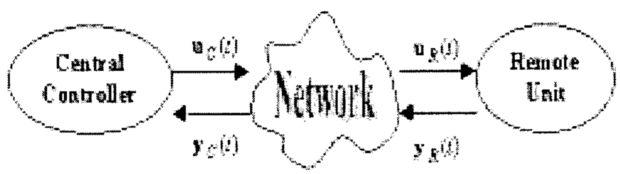

Fig. 1. A real-time network-based distributed control system.

control system can be categorized into three units: the Distributed Remote Unit, the Central controller, and the communication network.

• Remote System and Remote Controller

The remote controller can be assumed to have enough computing power to do relatively simple pre-programmed control, such as converting the control signal received from the central controller via network into a PWM signal to derive the remote process. The remote controller can send local measurements, such as motor current, speed, temperature, and local environment information back to the central controller via a network. The remote process has it own system dynamics that can be described by the state space description shown in Eq. (1), where the state vector $\mathbf{x_R} = [\mathbf{x_{R1}}, \ldots, \mathbf{x_{Rn}}]^\mathbf{T} \in \mathbf{X^n}$, the state space; $\mathbf{p_R} = [\mathbf{p_{R1}}, \ldots, \mathbf{p_{Rq}}]^\mathbf{T} \in \Re^\mathbf{q}$ are the system parameters; the input vector $\mathbf{u_R} = [\mathbf{u_{R1}}, \ldots, \mathbf{u_{Rr}}]^\mathbf{T} \in \mathbf{U^r}$, the input space; $t \in \Re^+$ is the time parameter; and $\mathbf{f_R} \in \Re^\mathbf{n}$ is the state transfer function of the remote part:

$$\dot{\mathbf{x}}_\mathbf{R} = \mathbf{f_R}(\mathbf{x_R}, \mathbf{p_R}, \mathbf{u_R}, \mathbf{t}). \tag{1}$$

Depending on the design of the network-based distributed control system, the remote controller, C_R, performs a certain task, such as regulating the performance of the plant P_R, as described by Eq. (2):

$$\mathbf{u_R} = \mathbf{g_R}(\alpha_\mathbf{R}, \cdot). \tag{2}$$

where $\alpha_\mathbf{R} = [\alpha_\mathbf{R1}, \ldots, \alpha_\mathbf{Ra}]^\mathbf{T}$ is the adjustable controller parameter vector and $(\cdot)$ represents other appropriate information. The combination of the remote controller and the remote plant can be viewed as a remote system, S_R, which has its own system dynamics that can be described by a set of differential equations:

$$\dot{\mathbf{x}}_\mathbf{R} = \mathbf{f_R}(\mathbf{x_R}, \mathbf{p_R}, \mathbf{g_R}(\alpha_\mathbf{R}, \cdot), \mathbf{t}). \tag{3}$$

• Central Controller

The central controller can be a highly sophisticated controller that requires lots of computing power and memory that is not suitable to be installed at the remote unit. The central controller is powerful and can provide advanced real-time control laws to the remote unit. The central controller will provide the control signal $\mathbf{u_C(t)}$ to the remote system via a network. Let $z^{-\tau}$ be a time delay operator, and let $QoS(t)$ be the current QoS provided by the network. We define:

$$\mathbf{u_R(t)} = \mathbf{u_C(z^{-\tau_R}, QoS(t))} \tag{4}$$
$$\mathbf{y_C(t)} = \mathbf{y_R(z^{-\tau_C}, QoS(t))} \tag{5}$$

where τ_R is the time delay in transmitting a signal from the central site to the remote site, and τ_C is the time delay in transmitting a signal from the remote site to the central site.

III. System Description

A. Remote Unit: Process and Controller

This section briefly describes the dynamics of the remote process: a DC motor with a driven load. The loop equation for the electrical circuit is:

$$u(t) = e_a = L\frac{di_a}{dt} + Ri_a + e_b \tag{6}$$

The mechanical torque balance based on Newton's law is:

$$J\frac{d\omega}{dt} + B\omega + T_l = T_e = Ki_a \tag{7}$$

where $u = e_a$ is the armature winding input voltage; L is the armature winding inductance; i_a is the armature winding current; R is the armature winding resistance; J is the system moment of inertia; B is the system damping coefficient; K and K_b are the torque constant and the back emf constant, respectively; T_l is the load torque; and ω is the rotor angular speed.

By letting $x_1 = i_a$ and $x_2 = \omega$, the electro-mechanical dynamics of the DC motor can be described by the following state-space description [1]:

$$\dot{\mathbf{x}} = \begin{bmatrix} -\frac{R}{L} & -\frac{K_b}{L} \\ \frac{K}{J} & -\frac{B}{J} \end{bmatrix}\mathbf{x} + \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix}\mathbf{u} \tag{8}$$
$$y = \begin{bmatrix} 0 & 1 \end{bmatrix}\mathbf{x} \tag{9}$$

assuming $T_l = 0$. In general form:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{Bu} & (10) \\ y &= \mathbf{Cx} & (11) \end{aligned}$$

The remote controller is used to simply convert the control voltage signal sent from the central controller into a PWM signal to drive the DC motor. The remote control value u_R can mathematically expressed as:

$$u_R(t) = u_C(t - \tau_R) \quad (12)$$

where τ_R is the delay time to transmit the control signal u_C from the central controller to the remote controller. The remote controller also sends the monitored signals $y_R(t)$ of the remote system back to the central controller, $y_C(t)$, and these two signals are related as:

$$y_C(t) = y_R(t - \tau_C) \quad (13)$$

where τ_C is the delay time to transmit the measured signal from the remote controller to the central controller.

B. Central Controller

The central controller uses a PI control algorithm to compute the control to the remote system for step tracking based on the monitored system signal sent from the remote system via the network. The PI controller used has the form:

$$u(t) = K_p e(t) + K_i \int e(t)dt \quad (14)$$

where K_p is the proportional gain; K_i is the integral gain; $r(t)$ is the reference signal for the system to track; $y(t)$ is the system output; and $e(t) = r(t) - y(t)$ is the error function. In our case, $y = \omega$, the motor speed, and $u(t)$ is the input voltage to the motor system.

The step response of the DC motor subject to tracking a 50 (rad/sec) reference signal with a PI control, $K_p = 1.3$ and $K_i = 38$, with no network effect, is shown in figure(2).

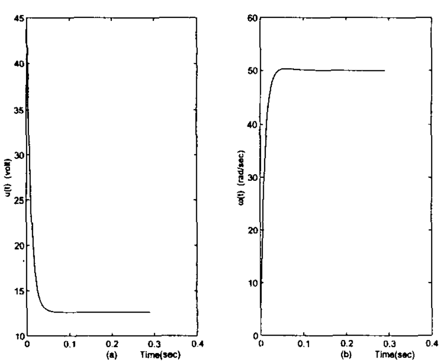

Fig. 2. (a): Control signal provided by the PI controller, (b): Step response of the DC motor, no network effect, $r(t) = 50$ (rad/sec)

The effect of the network delay and bandwidth on the step response of the DC motor with PI control is shown in figure(3), where h is the periodic sampling time. The following section will explain the network effect on the motor performance.

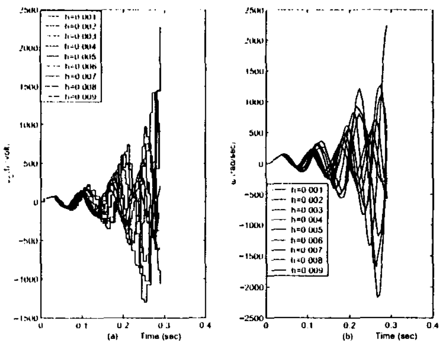

Fig. 3. (a): Control signal provided by the central controller, (b): Step response of the network-based controlled DC motor, with different bandwidth and random time delay τ_C and τ_R.

IV. Fuzzy Compensation and Performance

A. Effect of the network delays on the output of remote process.

The differential equation of the remote process,

$$\dot{\mathbf{x}}_{\mathbf{R}} = \mathbf{f}_{\mathbf{R}}(\mathbf{x}_{\mathbf{R}}, \mathbf{p}_{\mathbf{R}}, \mathbf{g}_{\mathbf{R}}(\alpha_{\mathbf{R}}, \cdot), \mathbf{t}), \quad (15)$$

is used with the state-space description in Eq. (8,9) to give

$$\begin{aligned} \dot{\mathbf{x}}_{\mathbf{R}} &= \mathbf{A}\mathbf{x}_{\mathbf{R}} + \mathbf{B}\mathbf{u}_{\mathbf{C}}(\mathbf{t} - \tau_{\mathbf{R}}) & (16) \\ y &= \mathbf{C}\mathbf{x}_{\mathbf{R}} & (17) \end{aligned}$$

where,

$$\begin{aligned} e(t) &= r(t) - y_R(t - \tau_C) & (18) \\ u_C(t) &= K_p e(t) + K_i \int e(t)dt & (19) \end{aligned}$$

solving for the state trajectory, we get

$$\mathbf{x}_{\mathbf{R}}(\mathbf{t}) = \int_0^t e^{\mathbf{A}(\mathbf{t}-\mathbf{s})} \mathbf{B}\mathbf{u}_{\mathbf{C}}(\mathbf{s} - \tau_{\mathbf{R}}) \mathbf{ds} \quad (20)$$

assuming that the initial condition $\mathbf{x}_{\mathbf{R}}(\mathbf{0}) = \mathbf{0}$. Eqns. (18-20) show mathematically how the network delays τ_C and τ_R affect the state trajectory and thus the output of the DC motor $y(t)$. Figure(4) shows the timing diagram of delay τ_R and τ_C, assuming that the computation time in the central controller and remote controller is very small compared with the delays τ_R and τ_C, so it can be ignored.

The control signal provided by the central controller and the delay time τ_C, time to transmit the measured signal from the remote controller to the central controller, can be related via the following equation:

$$u_C(t) = K_p \left[r(t) - y_R(t - \tau_C)\right] + K_i \int \left[r(t) - y_R(t - \tau_C)\right] dt \quad (21)$$

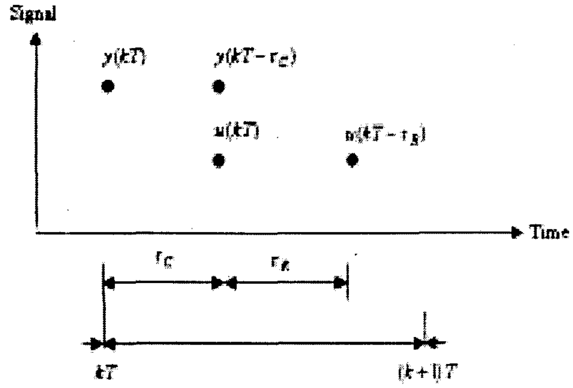

Fig. 4. Timing diagram of delay generation.

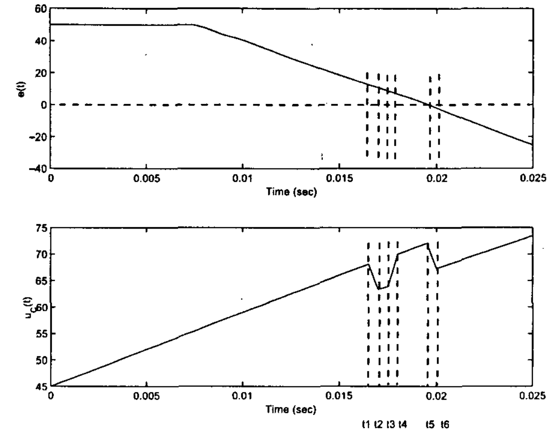

Fig. 6. Effect of the network delays on the system performance.

Figure(5) shows the norm of the control signal $\|u_C(t)\|$ verses different delay time τ_C.

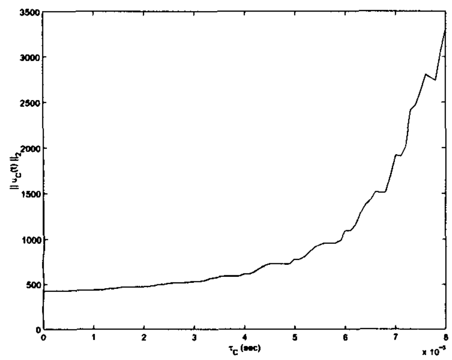

Fig. 5. The relation between τ_C and control signal $u_C(t)$.

As shown in figure (5), the network delay τ_C, affect the network-based system by increasing the value of the control signal $u_C(t)$ provided by the central controller. The control signal $u_C(t)$ is dependent on the error signal $e(t)$, which is dependent on the measured output $y_R(t)$ coming from the remote site via the network. So, at the beginning, $e(t)$ will be large, which lead to a large control signal $u_C(t)$. However, as $y(t)$ approaches $r(t)$, i.e., $e(t) \approx 0$, there is a time delay τ_C that the central controller realize the control signal should be decreases. Within this delay time period, $u_C(t)$ will keep increasing, which will lead $y(t)$ increases to a higher value. Thus, $e(t)$ will be higher than before which leads $u_C(t)$ to keep increasing. This time-delay effect can lead to a unstable closed-loop system response. The stated case is shown in figure (6), with $r(t) = 50$ (rad/sec).

Figure(6) shows that as the error signal $e(t)$ approaches zero (i.e., $\omega(t)$ approaches $r(t)$), the control signal will decrease its value in the interval $t \in [t_1, t_2]$. But the DC motor will receive this decreased value delayed by τ_R seconds, which leads $e(t)$ to keep increasing in the interval $t \in [t_2, t_3]$. This increasing in $e(t)$ will let $u_C(t)$ to resume increasing with $t \in [t_3, t_5]$. As $e(t)$ instantaneously become zero at $t = t_5$, the control signal $u_C(t)$ will again decrease its value upon receiving this error signal, but again it will resume increasing since $e(t)$ will keep increasing.

B. Network Control Delay Fuzzy Compensation

Our objective is to compensate for the effect of the network delays on the controlled DC motor without completely redesigning the controller. The approach taken is to use fuzzy logic to modify the PI controller output. A block diagram of the modified system is shown in figure(7).

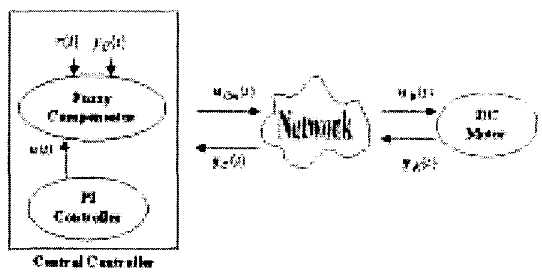

Fig. 7. A real-time network-based PI control system with fuzzy compensator.

In order to compensate and improve the performance of the PI control under different network delays and bandwidth, we introduce a parameter β such that the new control signal provided by the central controller is

$$u_{Cm}(t) = \beta u_C(t) \tag{22}$$

where $u_{Cm}(t)$ is the modulated version of the control signal provided by the central controller with parameter β. In this paper a novel methodology based on fuzzification of the new parameter β is presented in order to improve the performance of the network-based PI control. In this case, the state trajectory of the DC motor becomes:

$$\mathbf{x_R(t)} = \int_0^t e^{\mathbf{A(t-s)}}\mathbf{B}\mathbf{u_{Cm}}(\mathbf{s}-\tau_\mathbf{R})\mathbf{ds} \tag{23}$$

$$= \beta \int_0^t e^{\mathbf{A(t-s)}}\mathbf{B}\mathbf{u_C}(\mathbf{s}-\tau_\mathbf{R})\mathbf{ds} \tag{24}$$

From Eq. (24), we see that β has a direct relation to the state trajectory of the DC motor.

The compensation parameter β is given by:

$$\beta = \mathbf{h_{fz}}(\mathbf{e(t)}, \mathbf{u(t)}) \tag{25}$$

where $\mathbf{h_{fz}}$ is a nonlinear function that describes the input/output relation of the fuzzy compensator, and

$$\begin{aligned} e(t) &= r(t) - y_R(t - \tau_C) && (26)\\ &= r(t) - y_C(t) && (27) \end{aligned}$$

Based on the observation on the motor performance w.r.t. the delays described in the earlier parts of this section, the fuzzy compensator is composed of the following rules:

$$\text{if } e \text{ is } \textbf{Small} \text{ and } u \text{ is } \textbf{Large}\text{ , then } \beta = \beta_1 \tag{28}$$

$$\text{if } e \text{ is } \textbf{Large} \text{ and } u \text{ is } \textbf{Large}\text{ , then } \beta = \beta_2 \tag{29}$$

In the first rule, where e is **Small** and u is **Large**, we need to modulate the control signal provided by the central controller $u_C(t)$ by parameter $\beta = \beta_1$ in order to compensate for the network effect. In the second rule, where both e and u are **Large**, in this case, the modulator parameter $\beta = \beta_2$ will not be the same strength as in the first rule since e is **Large** in this case and the system needs higher $u_C(t)$ to reduce the error signal and the same time compensate for the network effects.

In order to acheive this heuristics we set the following condition:

$$\beta_1 < \beta_2 < 1 \tag{30}$$

where $\beta_i < 1$, $i = 1, 2$, are the consequent parameters corresponding to the compensation parameter β.

Two membership functions associated with the first input variable to the fuzzy compensator are used, namely **Small**, and **Large**, while one membership function associated with the second input variable to the fuzzy compensator is used, namely **Large**.

Membership function parameters are tuned off-line. The membership functions for the error signal are defined over the universe of discourse $e \in [0, e_{max}]$, where e_{max} is the maximum value of the error signal sequences. The initial setting for the membership functions **Small** and **Large** to describe e are defined as follows:

$$\mu_{\mathbf{Small}}(e) = \begin{cases} 0 & e > 0.6e_{max} \text{ or } e < 0 \\ 1 & 0 \leq e \leq 0.4e_{max} \\ \left(\frac{0.6e_{max} - e}{0.2e_{max}}\right) & 0.4e_{max} < e \leq 0.6e_{max} \end{cases} \tag{31}$$

$$\mu_{\mathbf{Large}}(e) = \begin{cases} 0 & 0 \leq e \leq 0.4e_{max} \\ \left(\frac{e - 0.4e_{max}}{0.2e_{max}}\right) & 0.4e_{max} \leq e \leq 0.6e_{max} \\ 1 & 0.6e_{max} \leq e \end{cases} \tag{32}$$

The membership function **Large** to describe u is defined over the universe of discourse $u \in [0.4u_s, u_s]$,

$$\mu_{\mathbf{Large}}(u) = \begin{cases} 0 & 0 \leq u \leq 0.4u_s \\ \left(\frac{u - 0.4u_s}{0.2u_s}\right) & 0.4u_s \leq u \leq 0.6u_s \\ 1 & 0.6u_s \leq u \end{cases} \tag{33}$$

where u_s is the saturated value of the DC motor actuator:

$$u = \begin{cases} u & 0 \leq u \leq u_s \\ u_s & u \geq u_s \end{cases} \tag{34}$$

The initial setting for the membership functions used for the input variables e, u to the fuzzy compensator are shown in figure(8).

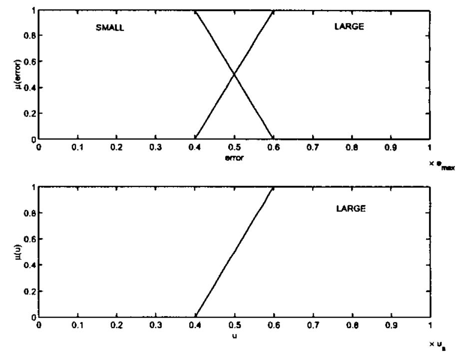

Fig. 8. Initial setting for the membership functions.

Finally, the output of the fuzzy compensator β is given by central deffuzification rule [15]:

$$\beta = \frac{\sum_{i=1}^{2} \omega^i \beta_i}{\sum_{i=1}^{2} \omega^i}. \tag{35}$$

where

$$\omega^i = \prod_{j=1}^{2} \tilde{A}_j^i(z_j). \tag{36}$$

is the rule firing strength, $z = [e(t), u(t)]^T$ is the input vector to the fuzzy compensator and $\tilde{A}_j^i$ is the membership function associated with the j^{th} input to the fuzzy compensator in the i^{th} rule.

To demonstrate the effectiveness of the proposed approach, we choose $\beta_1 = 0.10$ and $\beta_2 = 0.12$. With sufficient tuning on the membership functions, the final membership functions associated with the input variables to the fuzzy compensator is shown in figure(9).

A plot of the control signal provided by the central controller and the step response subject to tracking a 50 (rad/sec) reference signal with a PI control, $K_p = 1.3$ and $K_i = 38$, of the network-based DC motor with fuzzy compensator is shown in figure(10). By comparing figure(3) and figure(10), we see how the proposed fuzzy compensator improves the performance and tends the stability of the network-based controlled DC motor.

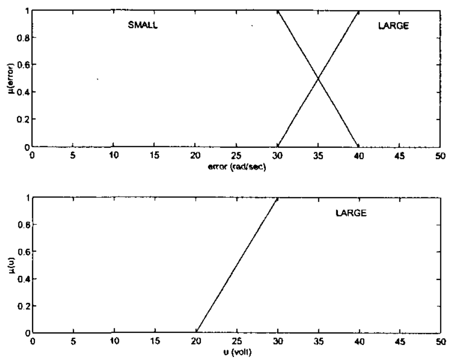

Fig. 9. Final setting of the membership functions.

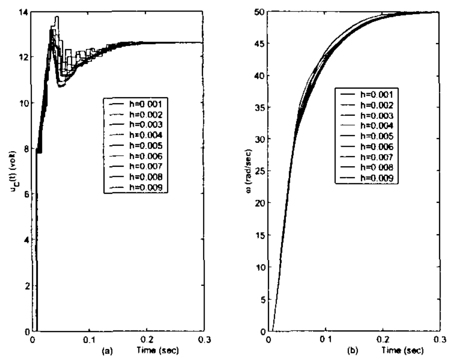

Fig. 10. (a): Control signal provided by the central controller, (b): Step response of the network-based controlled DC motor with fuzzy compensator, with different bandwidth and random time delay τ_C and τ_R.

V. Discussion and Conclusions

Adverse effect of the network on the system peformance lead researchers seek for new techiques to compensate for this effect. It is noticeable that these techniques can have their own advantages and disadvantages. In order to apply one of these techniques, designers have to consider both the network and the control algorithms. Nevertheless, none of these techniques is perfect because many unrealistic assumptions are extensively used.

Current available control techniques generally focus on conventional control techniques. In order to consider the uncertainty of communication delays, computational intelligent approaches such as fuzzy logic can be used. Fuzzy controllers are supposed to work in situations where there is a large uncertainty and/or unknown variation in plant parameters and structure which is the case in the network-based controlled DC motor problem.

In this paper, a fuzzy logic compensator is used to compensate for the network effect on the DC motor. In this approach we incorporate many advantages of using fuzzy logic such as the incorporation of heuristic knowledge, ease of implementation, and without the need for an accurate mathematical model. The fuzzy compensator is composed of two fuzzy rules derived by observing the effect of the network delay on the system performance. The key idea is to modulate the control signal provided by the central controller by a single parameter, which is the output of the fuzzy compensator. By this modulation we both prevent the system output from going to the instability region and obtained an acceptable step response. The results presented here support the new view that fuzzy logic has a promising future in the design and implementation of the network-based control system.

References

[1] Jason T. Teeter, Mo-Yuen Chow, and James J. Brickley Jr, "A novel fuzzy friction compensation approach to improve the performance of a DC motor control system," *IEEE Trans. Ind. Electron*, vol. 43, no. 1, pp. 113–120, 1996.

[2] Mo-Yuen Chow, *Chapter 39: Fuzzy Logic-Based Control,* The Industrial Electronics Handbook, 1996.

[3] K. J. Åström and B. Wittenmark, *Computer-controlled systems: Theory and Design, 2 ed.* Englewood Cliffs, Prentice-Hall, Inc. NJ, 1990.

[4] Y. Halevi and A. Ray, "Integrated communication and control systems: Part I - Analysis," *Journal of Dynamic Systems, Measurement, and Control*, vol. 110, pp. 367–373, 1988.

[5] S. H. Hong, "Scheduling algorithm of data sampling times in the integrated communication and control systems," *IEEE Transactions on Control Systems Technology*, vol. 3, pp. 225–230, 1995.

[6] R. Luck and A. Ray, "An observer-based compensator for distributed delays," *Automatica*, vol. 26, pp. 903–908, 1990.

[7] G. C. Walsh, O. Beldiman, and L. Bushnell, "Asymptotic behavior of networked control systems," *presented at IEEE International Conference on Control Applications*, 1999.

[8] Y. H. Kim, H. S. Park, and W. H. Kwon, "Stability and a scheduling method for network-based control systems," *presented at IEEE IECON 22nd International Conference on Industrial Electronics, Control, and Instrumentation*, 1996.

[9] H. Chan and U. Özgüner, "Closed-loop control of systems over a communication network with queues," *International Journal of Control*, vol. 62, pp. 493–510, 1995.

[10] J. Nilsson, B. Bernhardsson, and B. Wittenmark, "Stochastic analysis and control of real-time systems with random time delays," *Automatica*, vol. 34, pp. 57–64, 1998.

[11] J. Nilsson, "Real-time control systems with delays," 1998 Ph.D. thesis, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

[12] G. C. Walsh, H. Ye, and L. Bushnell, "Stability analysis of networked control systems," *presented at American Control Conference*, 1999.

[13] R. Luck and A. Ray, "Experimental Verification of a delay compensation algorithm for integrated communication and control systems," *International Journal of Control*, vol. 59, pp. 1357–1372, 1994.

[14] A. Ray and Y. Halevi, "Integrated communication and control systems: Part II - Design considerations," *Journal of Dynamic Systems, Measurement, and Control*, vol. 110, pp. 374–381, 1988.

[15] Li-Xin Wang, *A Course in Fuzzy Systems and Control,* NJ, Prentice-Hall, 1996.