Fuzzy Controller Design in Network Control Systems Based on Double Inverted Pendulum Control

Mojtaba Ghodrati[1] and Alireza Sahab[2],

[1]Garmsar branch Islamic Azad University
[2]Lahijan branch Islamic Azad University

Summary
Network control systems (NCSs) are time-dependent systems which connect to each other through a communication link like sensors, controllers and actuators. Therefore, numerous challenges arise for the design process of NCS. Generally, the feedback control systems that use a network system in their loops are called the network control systems (NCSs). NCSs are divided into two main types of local NCSs and wide NCSs. In local NCSs, components are connected to each other by local connections. This study presents various methods for restoring the inverted pendulum from unbalanced state to balance state and maintaining this state. Initially before setting up the inverted pendulum system, the system has been simulated in MATLAB software using its mechanical equations in order to reveal its general schematics. Normally, it is possible to determine the network effective parameters on the control system, such as delay and data loss in these systems. There are wide NCSs in contrast to these systems; these systems are distributed and the components of the system are interconnected by the local network and the Internet. The delay plays an important role in the implementation of the real-time NCS. This can make the NCS system unstable and negatively impact the realization and stability of the system. Communication networks inevitably cause delay in NCSs. Therefore, the delay is necessary to be compensated. The time used by the system depends on network characteristics such as topology and routing algorithms used in the network. In the present study, the objective was to design and simulate a fuzzy control for NCSs, and TrueTime and LMI toolboxes of the MATLAB software were used to simulate a fixed delay network. The Takagi-Sugeno (T-S) fuzzy model method was used to design the fuzzy controller.

Key words:
Fuzzy controller, Double inverted pendulum, TrueTime, NCS network control.

1. Introduction

For many years, data networking technologies have been applied to industrial and military control applications [1]. These applications include factory plants, automobiles, and spacecraft. Connection of components of a control system such as sensors, controllers, and actuators through the network can effectively reduce the complexity of the system with an effective economic development. In addition, network controllers allow data to be effectively shared. More importantly, cyberspace is connected to physical space in this system and remote implementation of tasks has become possible (a remote presence model). These systems are now very understandable and have many potential applications [2,3,4] including spatial explorations, soil research, factory automation, remote sensing and troubleshooting, hazardous environments, practical flexibilities, domestic robots, automobiles, spacecraft, factory plant monitoring, home or hospital nursing, remote robots and remote performance are just some of these applications [5].

Network control systems (NCSs) are time dependent systems that connect to each other through a communication link, like sensors, controllers, and actuators [6,7]. Network as a medium of interconnection can be wireless or wired.

NCSs act over the networks. Therefore, many challenges arise when designing the NCS process. Network reliability is one of the important factors. In the presence of the network in the NCSs, important issues arise, including delay, collapse and loss of data packets, etc. [7]. The time taken by the system to measure and send the control signal to the actuator and other components in NCS, depends on the network specifications, topology, and routing algorithms used in the network [2].

The NCS is a distributed real-time system [8]. A NCS can be a multi-sensor or multi-actuator system. Therefore, the bandwidth allocated to NCS should be taken into consideration. Timing techniques and bandwidth assignment strategies should be applied in optimal and efficient circumstances [9].

As a start, delay is a challenging issue in NCSs. All kinds of delays make serious problems for the realization of the system [2,6]. Generally, delay includes the total calculation time, measurement time, process time, and time spent by the controller or the signal transmitted from the network. NCS design is complicated when the network involves uncertainty in the NCS components (actuators, sensors, and controllers).

The first step in a computer network is to introduce the remote control concepts of a system with the advent of NCSs [2,5,10]. A NCS can be defined as a feedback

Manuscript received September 5, 2018
Manuscript revised September 20, 2018

control system in which the control loop is closed through a real-time network [1,2].
Therefore, random delays are a major challenge in NCS and are generally affect the random factors (e.g., network load, comparison points, and network congestion). There are two main types of random delay in NCSs [11]: sensor-to-controller delay (S-C delay) in the backward network channel and controller-to-actuator delay (C-A delay) in the forward path of the communication channel [4,11,12]. NCSs are at the intersection of the control and false communication of theories. Traditionally, the control theory focuses on the study of interconnected dynamical systems through bideal channels, while communication theory investigates data transfer on imperfect channels. A combination of these two frameworks is required by the NCSs model [2,13].
Many researchers discussed the stability analysis and controller design for NCS along with a linear control plant (i.e. linear NCS) in the presence of network delays and the loss of data packets, [11, 14,15]. Compared to the maximum allowed delay band (MADB) results, [14] the linear NCS stability problem has been developed with the sampled data control method [11]. Additionally, NCS along with a nonlinear control plant (i.e. nonlinear NCS) has been less considered [16].
Walsh et al. [11] discussed the exponential stability of nonlinear NCS under certain assumptions. However, they did not consider the Lyapunov function structure to meet the conditions and did not provide a control design method. The network was only available between the sensor and the controller.
In [14], a group of nonlinear NCSs was considered in which the controlled plant was a continuous-time linear plant with uncertainty. However, the results would be conservative because the banding method was used for mutual conditions and the network included delays over the buffer-generated level. Meanwhile, this method was not applicable for a non-linear plant.
Recently, multi-fractal natural delays have been obtained for network in the real-time NCS [16]. In the present study, a real-time hierarchical structure has been presented in order to obtain a predictable time behavior of network control [16,17], in which the packet loss in the actuator is compensated by the simple extrapolation of the previous control signals. Estimation of the loss of control packets from past control signals is similar to the dynamic voltage scheduling of past voltage settings by Varma et al. [18].
In [15], functional analysis of the simple packet loss method was obtained using some conventional methods, and thus an adaptive linear extrapolation was proposed for estimating the time control which was not received by the actuator [16].
Maintaining the balance of an inverted pendulum mounted on a cart moving in a horizontal direction is a classic issue in control systems. In the present study, various methods have been presented to restore the inverted pendulum from unbalanced to balanced state and to maintain this state.
In this system, an inverted pendulum is connected to a cart driven by a motor in the direction of the horizontal axis. A certain speed and position are given to the cart by means of the motor, and the rail path limits the cart motion into a single direction. Sensors have been placed in the system for measuring the pendulum deviation angle and the speed and position of the cart, and the necessary measurements as well as the motor control signals are performed by a control board, which is in fact the interface between the computer and the system, in addition, the commands necessary to control and results have been analyzed using MATLAB software.
The inverted pendulum system has two inherent balance points: one is stable and the other unstable. The stable balance point is where the pendulum is positioned downwards, and the system normally goes into this state without the need for any controller. The unstable balance point is related to the situation where the pendulum is positioned exactly upwards and therefore requires a controller to maintain its balance. The main objective in the inverted pendulum problem is to maintain balance at an unstable point, and the control object in the present project is to cause to reach the pendulum to a balance point from an unbalanced state.
Initially before setting up the inverted pendulum system, the system has been simulated in MATLAB software using its mechanical equations in order to reveal its general schematics.

2. NCSs

NCS is a system distribution space, in which communication between sensors, actuators, and controls is performed through a common limited band.
Typical digital communication networks are shown in figure 1 [8].

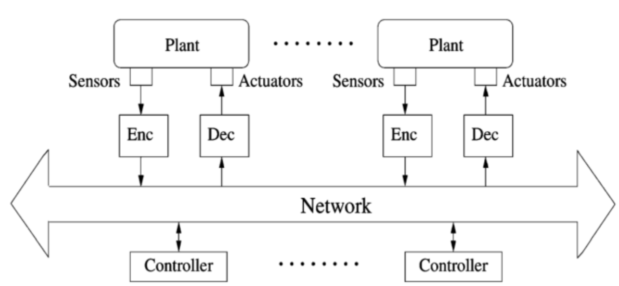

Fig. 1 Network control system (NCS) general architecture

The use of a multipurpose shared network for spatially distributed elements in addition to flexible architectural results, generally reduces installation and start and maintenance costs. Therefore, NCSs are applied in a wide variety of areas, including cellphone sensor networks, remote surgery, haptics collaboration over the Internet, automated highway system, and unmanned aerial vehicles (UAVs) [1,2,3,6,13]. However, the use of a shared network against multiple conventional communications has introduced a new challenge against Murray et al. [13]. Identification of network control is one of future key directions for control.

2.1 Modeling of NCS

2.1.1 Linear NCS

In general, a NCS consists of three main parts. According to figure 2, various NCS parts are shown in the following figure in order to model this system [19].

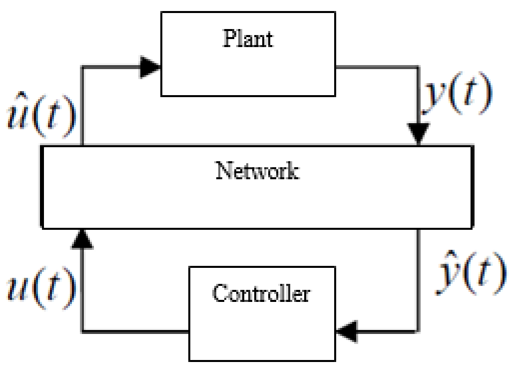

Fig. 2 Network control system (NCS) parts

The plant is considered as follows:

$$\dot{x}_p(t) = A_p x_p + B_p \hat{u}(t)$$
$$y(t) = C_p x_p(t)$$

Where, xp, u , and y are the states, input, and output of the plant, respectively.

$$\dot{x}_c(t) = A_C x_C(t) + B_C \hat{y}(t)$$
$$u(t) = C_C x_C(t) + D_C \hat{y}(t)$$

Similarly, xc, $\hat{y}$, and u are the states, input, and output of the controller, respectively.
Network is modeled by defining the error between input and output signals of the network. e1, e2 are considered as network delay errors (delay = τ) and are defined as follows:

$$e_1 = u - \hat{\hat{u}} \qquad , \hat{\hat{u}}(t) = u(t-\tau)$$
$$e_2 = y - \hat{\hat{y}} \qquad , \hat{\hat{y}}(t) = y(t-\tau)$$

The matrices of control signal loss and plant output loss are defined in order to model the effect of data loss. The control signal loss matrix is:

$$\theta = diag\{\theta_1, \theta_2, \ldots, \theta_m\}$$

Where, θ_i is the probability of passing the control signal $\hat{\hat{u}}_i$, so that $\theta_i = 1$ and $\theta_i = 0$ are considered for the signal passage and loss, respectively. Having this definition, the plant input control signal is:

$$\hat{u} = \theta . \hat{\hat{u}}$$

The plant output loss matrix is:

$$\varphi = diag\{\varphi_1, \varphi_2, \ldots, \varphi_w\}$$

Where, φ_i is the probability of passing the output $\hat{\hat{y}}_i$, so that $\varphi_i = 1$ and $\varphi_i = 0$ are considered for the passage and loss, respectively. Having this definition, the controller input signal is:

$$\hat{y} = \varphi . \hat{\hat{y}}$$

Now, if the state variables of the NCS are considered as $z = [x_p, x_c, e_1, e_2]^T$, the system state space equation is expressed as follows:

$$\dot{z}(t) = \Lambda z(t) \qquad \& \quad \Lambda = \begin{bmatrix} \Lambda_{11} & \Lambda_{12} \\ \Lambda_{21} & \Lambda_{22} \end{bmatrix}$$

Where,

$$\Lambda_{11} = \begin{bmatrix} A_P + B_P.\theta.D_C.\varphi.C_P & B_P.\theta.C_C \\ B_C.\varphi.C_P & A_C \end{bmatrix}$$

$$\Lambda_{12} = \begin{bmatrix} B_P.\theta & B_P.\theta.D_c.\varphi \\ 0 & -B_c.\varphi \end{bmatrix}$$

$$\Lambda_{21} = \begin{bmatrix} C_c.B_c.\varphi.C_p & Cc.Ac \\ C_p.A_p + B_p.\theta.D_c.\varphi.C_p & B_p.\theta.C_c \end{bmatrix}$$

$$\Lambda_{22} = \begin{bmatrix} 0 & -C_c.B_c.\varphi \\ -B_p.\theta & -B_p.\theta.D_c.\varphi \end{bmatrix}$$

The presence of controller representation matrices (D_c, C_c, B_c, A_c) in the elements of $\Lambda_{22}, \Lambda_{21}, \Lambda_{12}$ indicates the effect of the controller structure on the performance of the NCS. Moreover, if the network lacks any delay, that is:

$$\hat{\hat{y}}(t) = y(t) \quad , \quad \hat{\hat{u}}(t) = u(t)$$

Then, $e_1 = e_2 = 0$ and NCS is expressed as follows:

$$\begin{bmatrix} \dot{x}_p \\ \dot{x}_c \end{bmatrix} = \Lambda_{11} \begin{bmatrix} x_p \\ x_c \end{bmatrix}$$

The presence of matrices (D_c, C_c, B_c, A_c) in the elements of Λ_{11} indicates the effect of controller structure on the performance of the control system in the presence of data loss in the network. The above results indicate the need for choosing a controller with an appropriate behavior with network effects on system performance [19].

2.2 Simulation of NCSs

TRUETIME is a MATLAB/SIMULINK software package written by Henryxon, which simulates the temporal behavior of multi-task and real-time kernels containing control programs.

The TRUETIME simulation environment provides two simulation blocks. A computer block and a network block (both are event_drive); the computer block simulates the activity of a computer, including executing programs and switching for user defined fields and manual interrupts. Network block on the other hand, simulates the network dynamics based on the parameters entered by the user, like the message structure and a priority function used to determine the traffic priority. Figure 3 demonstrates the two blocks in the TRUETIME simulation environment. The network block has a suitable flexibility to simulate different types of networks, including CAN and internet-based networks. However in general, the TRUETIME network block is not adequate for simulation of the Internet arbitrary and complicated networks as well as IP-based networks.

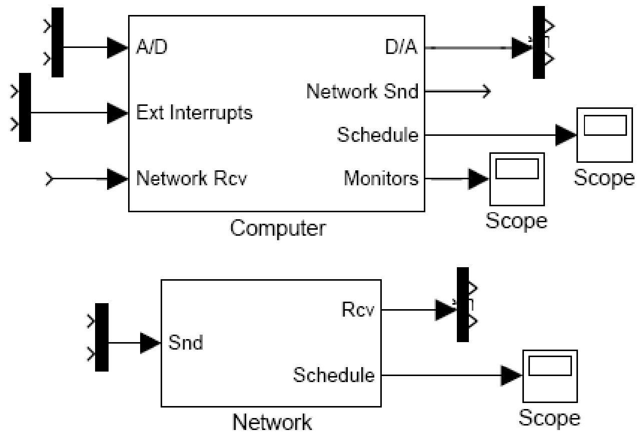

Fig. 3 Blocks of the TRUETIME simulation environment

2.3. Estimation of delay in NCSs

Delay in NCSs can be either deterministic or random, time-variant or time-invariant [20,21]. For local NCSs, the delay analysis and then the controller design are possible for control purposes, however, the delay is very complicated and usually random and time-variant in wide NCSs. Therefore, one of the important assumptions in the investigation of the NCSs is the consideration of an undetermined but bounded delay. Accordingly, the way and amount of data loss in the wide NCSs cannot be determined, unlike local NCSs. However, delay and data loss with a certain amount are considered in the analysis of NCSs, but this statement is not correct given the wide network structure, which indicates the necessity of considering them as uncertain values and their proper estimation is possible with the information available at any given moment. In this regard, it is first necessary to model the delay effect in the NCS equations properly, then obtain an instantaneous estimate using these equations [23]. This estimate, which can contain information such as mean and variance, can be used in the following cases:

Assuming that the transfer function of the main system is G(s), if the system has a delay τ, it is considered as exp (-τ s), the system transfer function along with delay is expressed as exp (−τ s)×G(s) [16,17]. Now, the delay is defined as a state variable in the state equations of this new system and is estimated using the developed Kalman filter. The system equations are as follows:

$$\begin{cases} X(k+1) = f(\tau, X(k)) + w(k) + u(k) \\ y(k+1) = X(k+1) + v(k+1) \\ w(k) \sim N(0,q) \\ v(k) \sim N(0,r) \\ x(0) \sim N(0,P_0) \end{cases}$$

T is the unknown delay which shall be estimated along with the state variables X. In order to use the Kalman filter to estimate the parameters, the delay must be added as a state variable to the system equations. Therefore, the new system equations are obtained as follows:

$$\begin{cases} X(k+1) = f(\tau(k), x(k)) + w(k) + u(k) \\ \tau(k+1) = \tau(k) + w_\tau(k) \\ y(k+1) = X(k+1) + v(k+1) \\ w_\tau(k) \sim N(0, q_\tau) \end{cases}$$

wτ (k), w (k), and v (k) is independent of each other and u (k) is a certain input signal. Defining:

$$X(k) = \begin{bmatrix} X(k) \\ \tau(k) \end{bmatrix}, \; f(X(k)) = \begin{bmatrix} f(\tau(k), X(k)) \\ \tau(k) \end{bmatrix},$$
$$W(k) = \begin{bmatrix} w(k) \\ w_\tau(k) \end{bmatrix} \sim N(0,Q),$$
$$Q = \begin{bmatrix} q & 0 \\ 0 & q_\tau \end{bmatrix}, \; U(k) = \begin{bmatrix} u(k) \\ 0 \end{bmatrix}$$

So:

$$\begin{cases} X(k+1) = f(X(k)) + W(k) + U(k) \\ y(k+1) = X(k+1) + v(k+1) \end{cases}$$

In the developed Kalman filter algorithm, the method is that at first, the nonlinear equations of the system are linearized around the working point, then the states are estimated using the standard Kalman filter. Linearization of f (X (k)) around X* (k) yields:

$$f(X(k)) \cong f(X^*(k)) + f_X(X^*(k)).(X(k) - X^*(k))$$

Where

$$f_X(X^*(k)) = \begin{bmatrix} \tau^*(k) & x^*(k) \\ 0 & 1 \end{bmatrix}$$

Given the above two relations, the state equation is as follows

$$X(k+1) \cong f(X^*(k)) + f_X(X^*(k)). \\ (X(k) - X^*(k)) + W(k) + U(k)$$

By applying the operator E{*Y(k)} to the sides of the above-mentioned relation, the following equation is obtained:

$$\hat{X}(k+1|k) \cong f(X^*(k)) + f_X(X^*(k)). \\ (\hat{X}(k|k) - X^*(k)) + U(k)$$

If X* (k) is defined as Xˆ(k k), then:

$$\hat{X}(k+1|k) \cong f(\hat{X}(k|k)) + U(k)$$
$$\hat{y}(k+1|k) \cong \hat{X}(k+1|k)$$

With the following definitions:

$$\begin{cases} \phi(k) = f_X(\hat{X}(k|k)) \\ m(k) = f(\hat{X}(k|k)) - f_X(\hat{X}(k|k))\hat{X}(k|k) + U(k) \\ H(k+1) = h_X(\hat{X}(k+1|k)) = I \\ n(k+1) = h(\hat{X}(k+1|k)) - h_X(\hat{X}(k+1|k))\hat{X}(k+1|k) \end{cases}$$

The following equations are obtained

$$\begin{cases} X(k+1) = \phi(k)X(k) + m(k) + W(k) \\ y(k+1) = H(k+1)X(k+1) + n(k+1) + v(k+1) \end{cases}$$

Assuming Xˆ(k k) to be known at moment k, Xˆ(k + 1 k) can also be calculated, hence, n (k) and m (k) are certain signals. It can be seen that the above equations are converted into a standard linear form for the Kalman filter and X(k + 1 k + 1) can be calculated as follows:

$$
\begin{cases}
\hat{X}(k+1|k+1) = \hat{X}(k+1|k) + K(k+1) \\
\qquad \times \left(y(k+1) - \hat{y}(k+1|k) \right) \\
K(k+1) = P(k+1|k) H'(k+1) \\
\qquad \times \left[H(k+1) P(k+1|k) H'(k+1) + r \right]^{-1} \\
P(k+1|k) = \phi(k) P(k|k) \phi'(k) + Q \\
P(k+1|k+1) = [I - K(k+1) H(k+1)] \\
\qquad P(k+1|k) \\
\hat{X}(0|0) = 0 \\
P(0|0) = P_0
\end{cases}
$$

3. Fuzzy controllers

Recently, fuzzy logic has emerged as an attractive issue in the control research field. The most important principle in fuzzy logic is the structure of fuzzy controllers using experts' linguistic knowledge.
As shown in figure 4, a fuzzy controller consists of 4 parts, 2 of which performing the conversion operation:

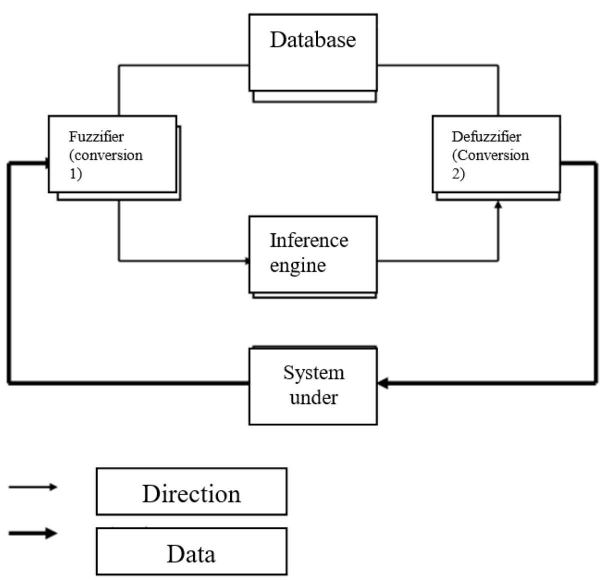

Fig. 4 Block diagram of a control system including fuzzy controller

The fuzzifier fuzzifies the input variables (real signals). Therefore, all the input signals become fuzzy. In simpler words, the fuzzifier converts numeric variables to fuzzy variables, that is, to linguistic variables. This conversion is carried out by membership functions.
The database contains basic data and linguistic rules. The basic data provide the information required in determining the rules. Database (certified rules) meets the main purpose of control by a set of linguistic control rules.
In other words, the database contains the rules provided by the experts. The fuzzy logic controller converts the input signals into proper output signals by the certified rules. The database contains a set of IF-THEN rules.
When a classical controller (PI or PID) is used, the error signal is the input of the controller. For instance, in a PI controller, the input is a speed error, which is the difference of the optimal and real speeds.
However, when a fuzzy logic controller is used, the controller will have more than one input. In most cases, the controller has two inputs, error (E) and changes (CE):
The fuzzy controller used here also has two errors: speed error and error changes. Of course, fuzzy controllers with higher inputs are also possible. The objective of the fuzzy controller is to obtain the proper output signal (CU) according to E and CE. Then, the total output of the output changes is obtained using the following formula.
As it can be observed here, the output of the fuzzy controller is the output changes that is summed up at any moment with the previous instantaneous output, however, it is also possible to use another fuzzy controller that directly returns the output itself rather than obtaining the output changes. The database is considered as the heart of a fuzzy controller, and includes rules for obtaining the optimal results. Generally, the IF-THEN linguistic rules are used in a database.
IF (E is A and CE is B) THEN (CU is C)
Where, A, B, and C are fuzzy sets for error, error changes, and output variations, respectively.

4. Simulations and numerical studies

4.1 System modeling (Inverted pendulum and motor):

The inverted pendulum system defined here is shown in figure 5, which consists of a cart, a pendulum, and a rail path to restrict the movement of the cart to a direction, as well as to determine its position. The pendulum connects from one end to the center of the upper surface of the cart and is free on the other end, so it can freely move on a plane containing the rail path. The cart, with its limited movement on the rail track, is driven by a motor, while the position of the cart from the middle of the path, and the pendulum deviation angle from the balance point, are measured by the sensors. The effect of friction has been neglected when obtaining the equations of the system and during simulations to simplify the equations, but this effect has been taken into account in practice.
The following inverted pendulum system has been considered. This system is excited by an impact force F. Then, its dynamic equations are obtained around the linear

balance point (it is assumed that the pendulum deviates only a few degrees from the balance point).

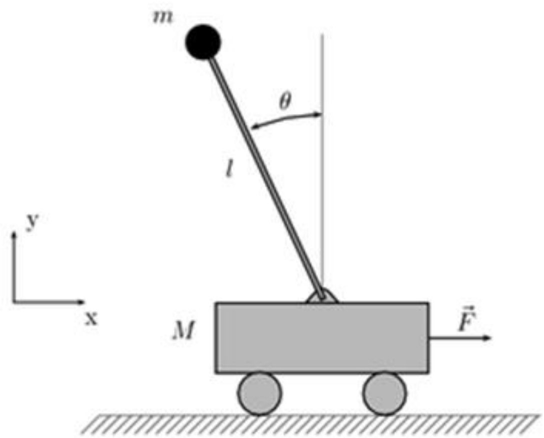

Fig. 5 Inverted pendulum free body diagram (FBD)

Parameters:

M	Cart mass	0.5 kg
m	Pendulum mass	0.05 kg
b	Cart friction	0.1 N/m/sec
l	Distance of center of mass of the pendulum	0.3 m
I	Pendulum moment of inertia	0.006 kg.m2
F	Force applied to the cart	
x	Cart position	
theta	Angle of pendulum relative to the vertical direction	

In control of locus of roots and frequency response by the PID method, the pendulum angle can only be maintained to reach balance, since these methods are suitable for controlling single input-single output systems. In this section, it is assumed that the pendulum is initially in balance and then faces a 1 N impact force. The design criterion in this simulation is that the pendulum never deviates from the balance more than 0.1 radians and returns to balance in less than 20 seconds. If the state space equations are used for controlling the system, the pendulum angle and position of the cart can be easily controlled simultaneously. As, the state space is more suitable for multi-output systems.

4.2 Simulation of fuzzy controller and PI on inverted pendulum using TRUETIME

Inverted pendulum modeling:
According to the figure 5, an inverted pendulum with a mass m and a length l m is placed on a cart with a mass M; this cart moves horizontally. The control force F drives the M and hence, causes the balance of the inverted pendulum. In this system, the friction is assumed to have a direct relationship with speed.
Modeling method based on Lagrange equations:
Generalized variables fully defining the motion of the system are x, θ
The total energy of the system is:

$$T = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m\left[\left\{\frac{d}{dt}\left(x + \frac{\ell}{2}\sin\theta\right)\right\}^2 + \left\{\frac{d}{dt}\left(\frac{\ell}{2}\cos\theta\right)\right\}^2\right] + \frac{1}{2}\mathrm{I}\dot{\theta}^2$$

$$T = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m\left[\left(\dot{x} + \frac{\ell}{2}\dot{\theta}\cos\theta\right)^2 + \left(-\frac{1}{2}\dot{\theta}\sin\theta\right)^2\right] + \frac{1}{2}\mathrm{I}\dot{\theta}^2$$

The potential energy of the system is equal to:

$$U = mg\frac{\ell}{2}\cos\theta$$

$$L = T - U = \frac{1}{2}M\dot{x}^2 + \frac{1}{2}m\left[\left(\dot{x} + \frac{\ell}{2}\dot{\theta}\cos\theta\right)^2 + \left(-\frac{1}{2}\dot{\theta}\sin\theta\right)^2\right] + \frac{1}{2}\mathrm{I}\dot{\theta}^2 - mg\frac{\ell}{2}\cos\theta$$

Solving the Lagrange equation along x, θ we have:

$$(I): \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = F$$

$$\Rightarrow (M + m)\ddot{x} + b\dot{x} + m\ell\ddot{\theta}\cos\theta - m\ell\dot{\theta}^2\sin\theta = F$$

$$(II): \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{\theta}}\right) - \frac{\partial L}{\partial \theta} = 0$$

$$\Rightarrow (\mathrm{I} + m\ell^2)\ddot{\theta} - mg\ell\sin\theta = m\ell\ddot{x}\cos\theta$$

Linearizing around the working point yields:

$$\begin{cases}(\mathrm{I} + m\ell^2)\ddot{\theta} - mg\ell\theta = m\ell\ddot{x} \\ (M + m)\ddot{x} + b\dot{x} + m\ell\ddot{\theta} = \mathrm{F}\end{cases}$$

Pendulum transfer function:
By taking Laplace transformation of equation (1) we have:

$$\begin{cases}(\mathrm{I} + m\ell^2)s^2\theta(s) - mg\ell\theta(s) = m\ell s^2 X(s) \\ (M + m)s^2 X(s) + bsX(s) + m\ell s^2\theta(s) = \mathrm{F}\end{cases}$$

Since the desired output is the pendulum angle, removing x from the above equations, we will have:

$$\frac{\theta(s)}{F(s)} = \frac{\frac{m\ell}{q}s^2}{s^4 + \frac{b(I+m\ell^2)}{q}s^3 - \frac{(M+m)mg\ell}{q}s^2 - \frac{bmg\ell}{q}s}$$

$$q = [(M+m)(I+m\ell^2) - (m\ell)^2]$$

Simplifying this yields:

$$\frac{X(s)}{F(s)} = \frac{\frac{I+m\ell^2}{q}s^2 - \frac{mg\ell}{q}}{s^3 + \frac{b(I+m\ell^2)}{q}s^2 - \frac{(M+m)mg\ell}{q}s - \frac{bmg\ell}{q}}$$

Takagi-Sugeno (T-S) fuzzy control:
A first order T-S fuzzy controller is used to control the inverted pendulum system. The system under control has four state variables, including the horizontal distance x, horizontal velocity $\dot{x}$, deviation angle θ, and angular velocity $\dot{\theta}$, the range of these variables is as follows:

$$x \in [-4,4]$$
$$\dot{x} \in [-3,3]$$
$$\theta \in [-80°,80°]$$
$$\dot{\theta} \in [-2,2]$$

The i-th rule in the T-S fuzzy control is as follows:

Rule i:
if $(x \text{ is } MF_p^1)$ *AND* $(\dot{x} \text{ is } MF_q^2)$ *AND* $(\theta \text{ is } MF_r^3)$ *AND* $(\dot{\theta} \text{ is } MF_s^4)$
Then $F = a_i x + b_i \dot{x} + c_i \theta + d_i \dot{\theta} + e_i$

Where, $p,q,r,s = 1,2$ and $i = 1,2,\cdots,16$. To avoid the complexity of the problem for each state variable, two membership functions are considered within the variable definition range. Therefore, the base of the rules of the fuzzy control system includes 16 fuzzy rules. The membership functions of the four state variables are shown in figures 6, 7, 8, and 9.

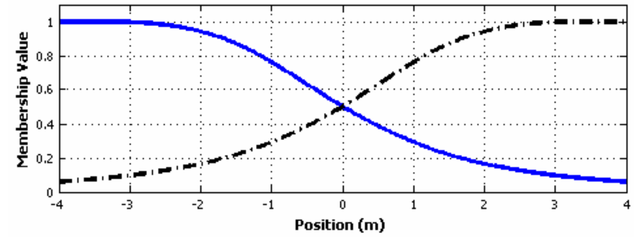

Fig. 6 Position variable membership functions (MF1, MF2)

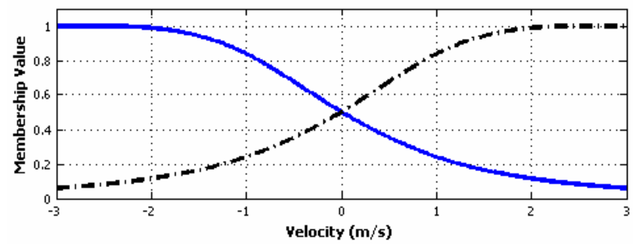

Fig. 7 Horizontal velocity variable membership functions

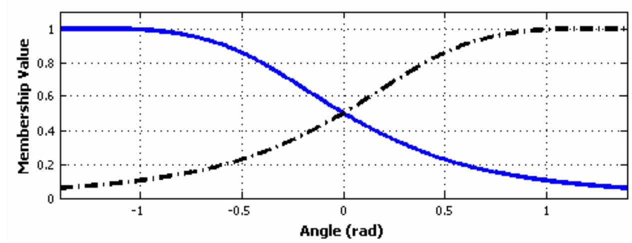

Fig. 8 Angle variable membership functions

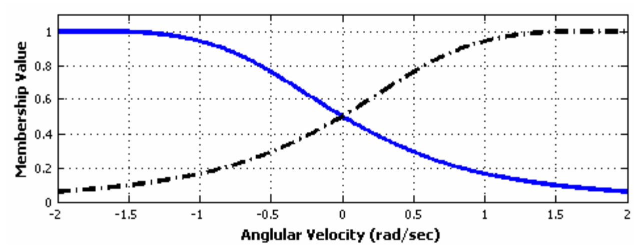

Fig. 9 Angular velocity variable membership functions

TRUETIME is a MATLAB/SIMULINK software package written by Henryxon simulating the temporal behavior of multi-task and real-time kernels containing control programs. TRUETIME simulation environment provides two simulation blocks (a computer block and a network block, both of which are event_drive); the computer block simulates the activity of a control computer, including executing programs and switching for user defined fields and manual interrupts. Network block on the other hand, simulates the network dynamics based on the parameters entered by the user, like the message structure and a priority function used to determine the traffic priority.

Figure 10 demonstrates the two blocks in the TRUETIME simulation environment. The network block has a suitable flexibility to simulate different types of networks, including CAN and Internet-based networks. However in general, the TRUETIME network block is not adequate for simulation of the Internet arbitrary and complicated networks as well as IP-based networks.

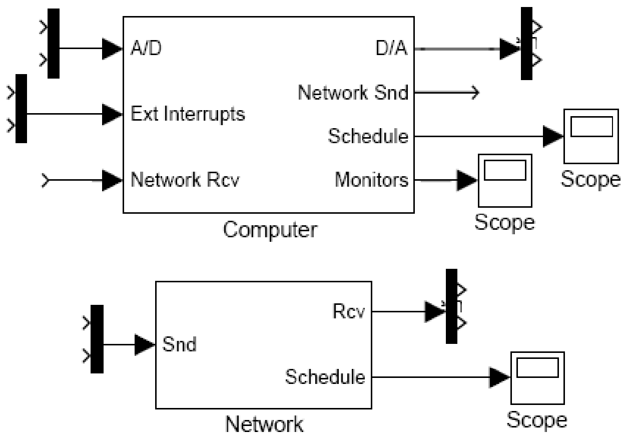

Fig. 10 Blocks of the TRUETIME simulation environment blocks

The kernel of the TRUETIME blocks simulates a computer problem in real time. Network blocks distribute messages based on the selected network model. The blocks are connected to the real-time control system via continuous-time Simulink blocks.
First, the NCS has been designed for the inverted pendulum using the PID controller and the TRUTIME toolbox in the Simulink environment as figure 11.

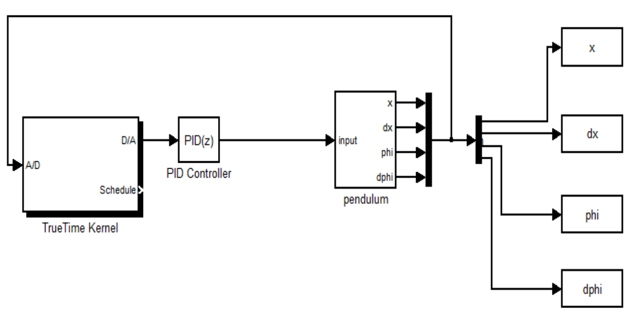

Fig. 11 Design of the inverted pendulum using the PID controller and the TRUTIME toolbox in the Simulink environment

In this section, the simulation is used with a T-S fuzzy controller. In this system, as shown in figure (4.8), the controller contains two inputs of d and de, respectively, of position and derivative of position of the pendulum, and its conditions are as table ... and the output includes three functions mf1, mf2, and mf3.

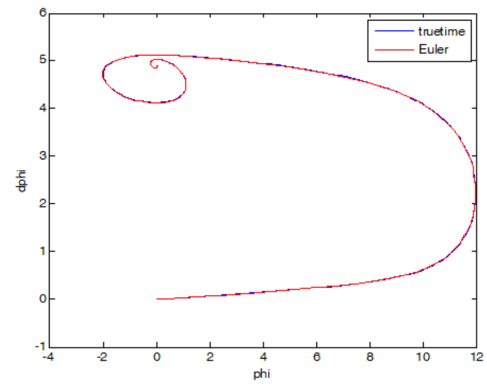

Fig. 12 Output of the PHI derivative in terms of PHI using the Euler and TRUETIME calculator

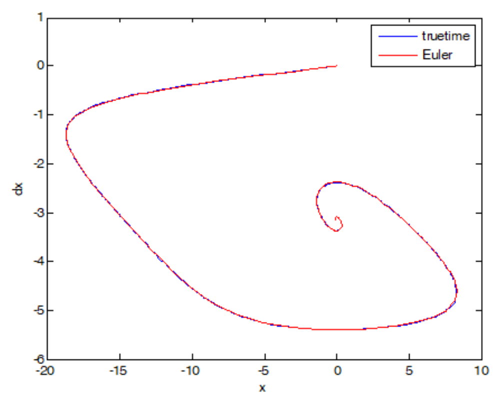

Fig. 13 Output of the derivative of x in terms of x using the Euler and TRUETIME calculator

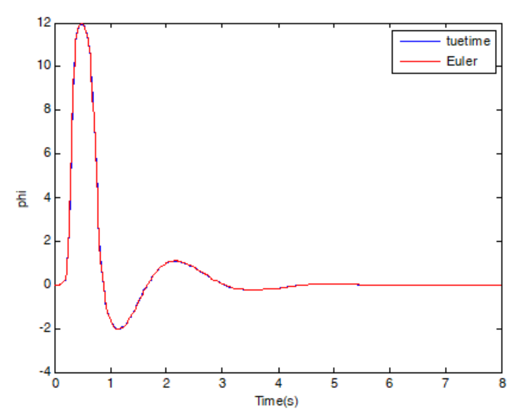

Fig. 14 Output of phi in terms of time using the Euler and TRUETIME calculator

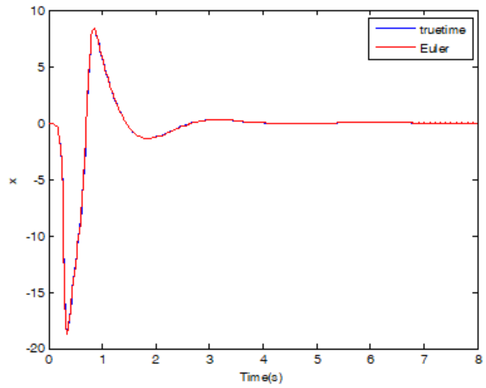

Fig. 15 Output of x in terms of time using the Euler and TRUETIME calculator

It can be seen that using TRUETIME, a small time difference is observed over the Euler method and it is implemented earlier than the Euler method. By designing the fuzzy controller as figure 16, the results are analyzed using TRUETIME and the results are as seen in figure 16.

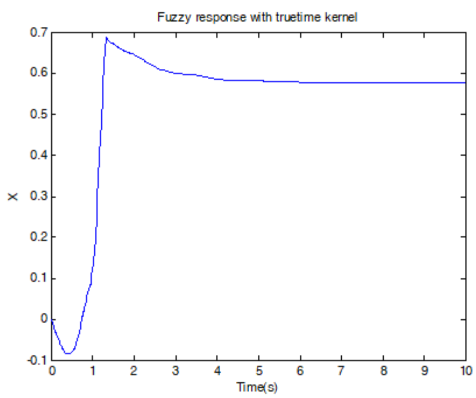

Fig. 16 Output of x in terms of time using the TRUETIME calculator and fuzzy logic

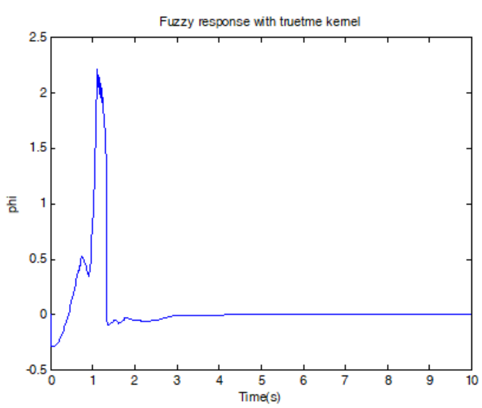

Fig. 17 Output of phi in terms of time using the TRUETIME calculator and fuzzy logic

Two important points in this design are the earlier stability compared to the PID controller in the TRUETIME toolbox in the previous simulation, and also the increased runtime in comparison to the previous case.

The empirical results of the PID controller without the time delay compensator are shown in figure ….. In this response, ypm cannot overlap (follow) yp and causes the control signal to change towards infinity. Using the fuzzy controller shown in figures ... and ..., the ym response follows yp and the output yp tracks the reference input r. Full NCS stability conditions have been achieved even under high probability of loss of NCS data packets, i.e. 0.3. Thus, the simulation results indicate that the proposed solution completely yields the desired results for the NCS system.

Simulation results for delay t = 1.049:

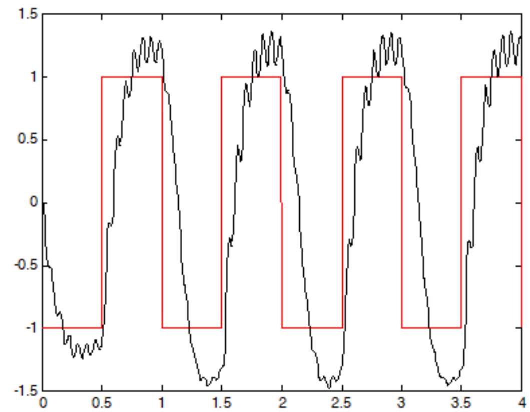

Fig. 18 Response to the system step input with a delay of 1.049

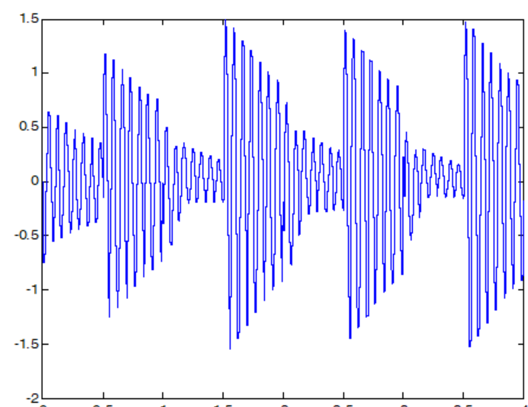

Fig. 19 Error between output and reference

Simulation results for delay t = 0.089:

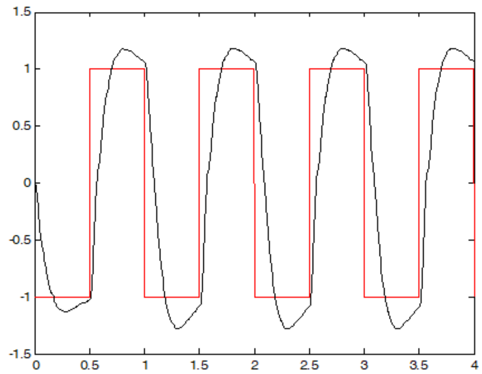

Fig. 20 Response to the system step input with a delay of 0.089

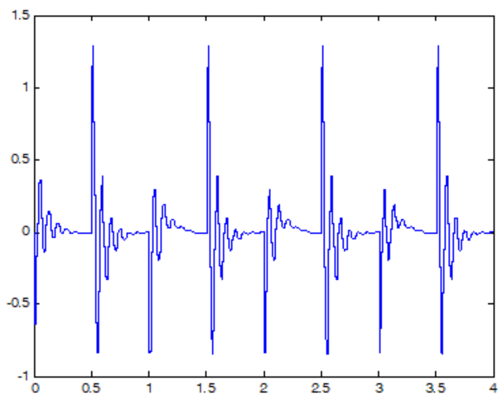

Fig. 21 Error between output and reference

Simulation results for delay t = 0.049:

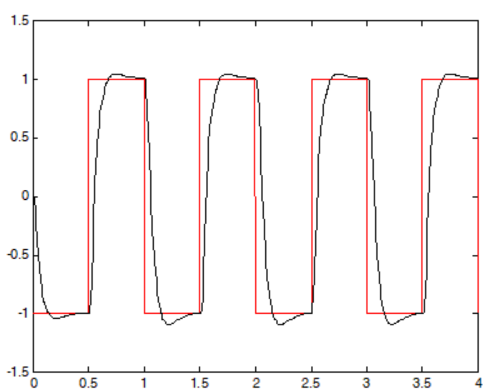

Fig. 22 Response to the system step input with a delay of 0.049

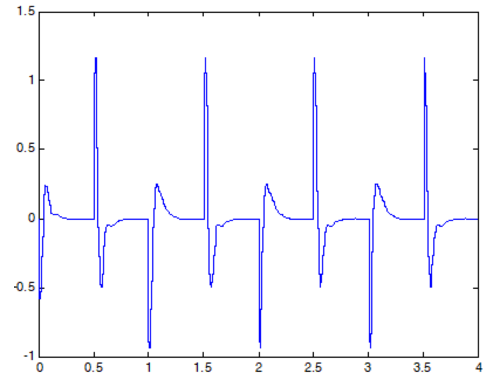

Fig. 23 Error between output and reference

It is observed that when the delay decreases, the input and output results approach each other, thus increasing the delay causes overshoot and increased response time.

5. Conclusion

In this study, in order for feasibility investigation, a NCS has been implemented to control the double inverted pendulum in a network environment. The NCS can be used in a direct or hierarchical structure, depending on the application requirements and designer preferences. Regardless of the structure used, the realization of the NCS system degrades due to network delays in the control loop. In the worst case, network delays can unstable the NCS by reducing the stability range. Coping with random network delays in the loop is much more difficult than random or periodic delays because there are no criteria to ensure the stability of the NCS. Stability criteria for NCS are usually the subject of specific methods and network protocols. Therefore, for designing a NCS with a specific network control methodology, the designer should be able to recognize the applications clearly, practically, acceptably, and reliably to control with the selected network protocol method. For example, if a plant is linear in NCS, all methods are applicable. However, if the plant is non-linear, only a disturbance method, a robust control method, and an event-based control method are useful for this stage. Hierarchical method should not be used in ring service network, as long as the method is not useful in long delays. The end-user adaptive control method is preferred when the QOS network can be supplied and monitored ... It should been investigated how to simulate a plant in a network environment. The experimental results indicate that fuzzy control is an effective method for NCS because of its robustness against parameter uncertainty. Due to the inherent robustness, the fuzzy controller system designed for a directly connected control system may be used for a NCS without the need for redesign.

References

[1] Krtolica, R., O. zgu.ner, U. ., Chan, H., Go.ktas, H., Winkelman, J., & Liubakka, M. (1994). Stability of linear feedback systems with random communication delays. International Journal of Control, 59(4), 925–953.

[2] Rachana A. Gupta & Mo-Yuen Chow.2007. Overview of Networked Control Systems. North Carolina State University, Raleigh, NC 27695, USA.

[3] Joap & Hespanha & Payam Naghshtabrizi & Yonggang Xu.2008. A survey of recent results in NCSs. Department of Electrical and Computer Engineering,Univ. of California, Santa Barbara, CA 93106-9560.

[4] P.Q. Reyes, J.O. Arjona, E.M. Monroy, H.B. Pérez , A.D. Chave sti.2012. Fuzzy Control Design for a Class of Nonlinear NCS: Helicopter Case Study . INT J Comput Commun, ISSN 1841-9836 Vol.7 (2012), No. 2 (June), pp. 365-376.

[5] Yook, J.K., Tilbury, D.M., &Soparkar, N.R., 2000. A design methodology for distributed control systems to optimize performance in the presence of time delays. Proceedings of the 2000 American control conference, Vol. 3 (pp. 1959–1964). Chicago, IL.

[6] Sumit Vardhan, Student Member, IEEE, and Rajiv Kumar, Member, IEEE ,2011. Simulations for Time Delay Compensation in Networked Control Systems. Multidisciplinary Journals in Science and Technology, Journal of Selected Areas in Telecommunications (JSAT), June Edition, 2011 .

[7] G.oktas, F., 2000. Distributed control of systems over communication networks. Ph.D. dissertation, University of Pennsylvania.

[8] Ray, A., &Halevi, Y. (1988).Intergrated communication and control systems: Part II—design considerations. Journal of Dynamic Systems, Measurement, and Control, 110, 374–381.

[9] Kim, Y.H., Park, H.S., & Kwon, W.H., 1996. Stability and ascheduling method for network-based control systems. The 22nd annual conference of the IEEE industrial electronics society (IECON96), Vol. 2 (pp. 934–939). Taipei, Taiwan.

[10] Lian, F.-L., Moyne, J. R., &Tilbury, D. M. (2001). Performance evaluation of control networks: Ethernet, Controlnet, and Device- Net. IEEE Control Systems Magazine, 21(1), 66–83.

[11] Yuan Ge, Qigong Chen, Ing Jiang and Yiqing H uang .2013. Modeling of Random Delays in Networked Control Systems . Hindawi Publishing Corporation Journal of Control Science and Engineering Volume 2013, Article ID 383415, 9 pages.

[12] Huajing Fang, Hong Zhang, Yiwei Fang, Fang Yang .2006. Quasi T-S Fuzzy Models and Stable Controllers for Networked Control Systems. Proceedings of the 6th World Congress on Intelligent Control and Automation, June 21 - 23, 2006, Dalian, China.

[13] Almutairi, N.B., Chow, M.-Y., &Tipsuwan, Y., 2001. Network-based controlled DC motor with fuzzy compensation. The 27th annual conference of the IEEE industrial electronics society (IECON 01),Vol. 3 (pp. 1844–1849). Denver, CO.

[14] Dong-Sung Kima , Young Sam Leea , Wook Hyun Kwona, Hong Seong Park.2003. Maximum allowable delay bounds of networked control systems.Science Direct , Control Engineering Practice 11 (2003) 1301–1313.

[15] Yodyium Tipsuwan, Mo-Yuen Chow . 2003 . Control Methodologies in Networked Control Systems . Science Direct, Control Engineering Practice 11 (2003) 1099–1111.

[16] Yu-Chu Tian , David levy . 2007 . Compensation For Control Packet Dropout in Networked Control Systems . Science Direct,INS 7818.No of Pages 16, Model 3+ , 14 November 2007.

[17] Amir Ali Mohaghpour, Ali Karimpour, Naser Paziz. 2007. Removal of effects of data transfer networks on performance of control systems designing a proper controller. Scientific-specialized electrical journal of Majlisi Azad University, First year / No 2, fall 2007

[18] Hong, S. H., & Kim, W.-H. (2000). Bandwidth allocation scheme in CAN protocol. IEE Proceedings—Control Theory and Applications, 147(1), 37–44.- Kaplan, G. (2001). Ethernet’s winning ways. IEEE Spectrum, 38(1),113–115.

[19] Amir Ali Mohaghpour, Ali Karimpour, Naser Pariz. 2007. Estimation of network delay in control systems by designing appropriate controller. Scientific-specialized Journal of Electrical Engineering, Majlisi Azad University, First Year / First issue, Summer 2007

[20] Liou, L.-W., & Ray, A. (1990). Integrated communication and control systems: Part III—nonidentical sensor and controller sampling. Journal of Dynamic Systems, Measurement, and Control, 112,357–364.

[21] Walsh, G.C., Ye, H., & Bushnell, L., 1999c. Stability analysis of networked control systems. Proceedings of the 1999 American control conference, Vol. 4 (pp. 2876–2880). San Diego, CA.
Nilsson, J., 1998. Real-time control systems with delays.Ph.D. dissertation, Lund Institute of Technology.

[22] Walsh, G.C., Beldiman, O., & Bushnell, L., 1999b. Error encoding algorithms for networked control systems. Proceedings of the 38th IEEE conference on decision and control, Vol. 5 (pp. 4933–4938). Phoenix, AZ.

[23] C., & Price, C. R. (1993). Network-based infrastructure for distributed remote operations and robotics research. IEEE Transactions on Robotics and Automation, 9(5), 702–704.