Design and Simulation of Fuzzy Control in Network Control Systems

Mojtaba Ghodrati[1] and Alireza Sahab[2]

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

Summary
Generally, the feedback control systems using a network system in their loops are called the network control system (NCS). NCSs are divided into two main types: local NCSs and wide NCSs. In local NCSs, components are connected to each other by local connections. Normally, it is possible to determine the network effective parameters on the control system, such as delay and information 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 time delay plays an important role in the implementation of the real-time NCS. This can make the NCS system unstable and make its realization and stability harder to achieve. 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 fuzzy control of NCSs, and TrueTime and LMI toolboxes 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 control, Network control systems, simulation, T-S fuzzy model

1. Introduction

The development of the Internet has provided a huge base for millions of home-based, academic, commercial, and social network users, exchanging information and services, including email, online chat, file uploads, internal connection of web pages, and other World Wide Web documents [1-3]. In recent years, there has been a dramatic increase in the use of the wireless systems, leading to the development of the distributed network control system (NCS) [4]. The potential strength in various applications has resulted in the growth of NCS and this often creates many challenges for researchers to achieve an effective and reliable control. Therefore, numerous investigations have been carried out in the field of NCS for decades and it has become an important topic for research. A large branch in the literature focuses on the various strategies of control and kinematics of stimuli and appropriate tools for NCS [4,5]. Another important area of research relevant to NCS is the study of the network structure required to provide a safe and reliable communication channel with sufficient bandwidth and the development of communication protocols for control systems [6,7,8]. Collecting real-time information through the network using distributed sensors and sensor information processing in effective mode is an effective research area for NCS support [4]. Therefore, NCS is not only a disciplinary area related to the computer and communication networks, signal analysis, robotics, information technology (IT), and control theory, but also combines them very beautifully to obtain a system so that it can efficiently work through the network. For example, a robot located in the eastern part of the world can be controlled by a person in a place in the United States [8].
The applications of the NCS can be divided into two main categories: delay-sensitive applications [7] and delay-insensitive applications. Time delay causes instability in the delay-sensitive systems [9,101,11]. This factor plays an important role here. If the delay exceeds the endurable limit, the realization of the system will be difficult [12]. This can cause damage to the system. Remote operation via the network is an example of the delay-sensitive applications which can be exploited in submarine operations, fire extinguishing, surgery operations, and highway traffic control. Moreover, in the delay-insensitive applications, speed is not critical. An example of the performance of these systems has been presented in reference [13].
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. [14]. The time taken by the system to measure and send the control signal to the drive and other components in NCS, depends on the network specifications, topology, and routing algorithms used in the network [15].
The NCS is a distributed real-time system [15]. A NCS can be a multi-sensor or multi-drive 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 [17].

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

As a start, time delay is a challenge in NCSs [18]. All kinds of delays make serious problems for the realization of the system [15,19]. Generally, time 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). When the conditions are provided for the data loss, the realization of the system becomes more difficult. Time delay in NCS will make the system unstable, and no proper response is received. It is therefore necessary to design a controller to compensate for the time delay for the NCS operational and stability environments.

Compared with the old point-to-point control, NCSs decrease costs and wirings, facilitate identification and maintenance of the system, and increase the flexibility and reliability of the system [20]. Therefore, NCSs have been widely considered in many applications including transportation, spacecraft, factories, and remote surgery [21-24]. However, limited bandwidth and servicing cause new challenges such as random time delay, data packet loss, multiple sending of data packets, and interruption in communication packets [15,18,25].

Ling and Lemmon [26] considered a compensator for feedback measurements in NCS in the specified framework of the problem. They found that the optimal compensator was more efficient than previous compensation methods. Their compensator schematics were designed to measure packets and were directly related to the process model and controller design.

Schenato [17] studied optimal state estimator for NCS for random delay and packet loss, and used the estimation in control design. This works includes special mathematical behavior. However, like Lock and Ray [27] and Shan and Ezgner [28], this work is closely related to the accuracy of the process model due to the problems in the realization of the general NCS, and is often sensitive to computations.

Therefore, the present study was carried out aiming to design and simulate fuzzy control for NCSs, and TrueTime and LMI toolboxes were used to simulate a fixed delay network. In addition, the Takagi-Sugeno (T-S) fuzzy model method was used to design a fuzzy controller.

2. NCSs

Today, networks are used in control systems of many systems, including planes, robots, and factories, to exchange control signals between system components. The development of microcontrollers has increased the efficiency of the use of computers in the control systems, so that the use of application-specific integrated circuit (ASIC) chips has allowed the implementation of sensors and operators as network nodes, leading to the emergence of NCSs. Figure 1 demonstrates a sample NCS.

Generally, NCS is referred to as a feedback control system using a network system in its control loops. The analysis of these systems is complicated due to the existence of a network in a feedback loop. The NCS has various applications. Figure 2 shows an example of these applications, in which devices used for various purposes are connected to a control system by the network.

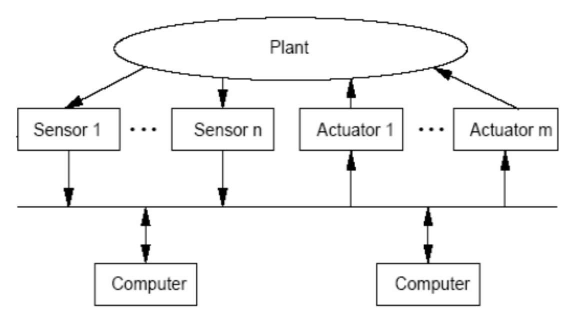


Fig. 1 Network control system (NCS)

2.1 Modeling of NCS:

2.1.1 Linear NCS

In general, a NCS consists of 3 main parts. According to figure 2, in order to model the NCS, its various parts are expressed in the following form [29].

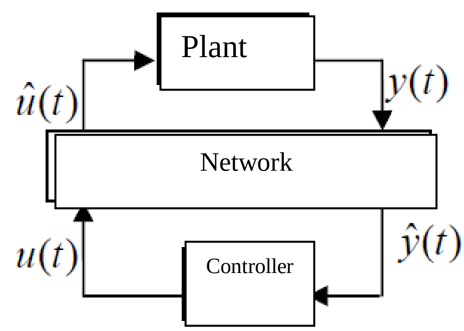


Fig. 2 Network control system (NCS) parts

The plan 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 at the time 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 [29].

2.2 Simulation of NCSs:

2.2.1 TRUETIME

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 depicts the two blocks in the TRUETIME simulation environment. This 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.

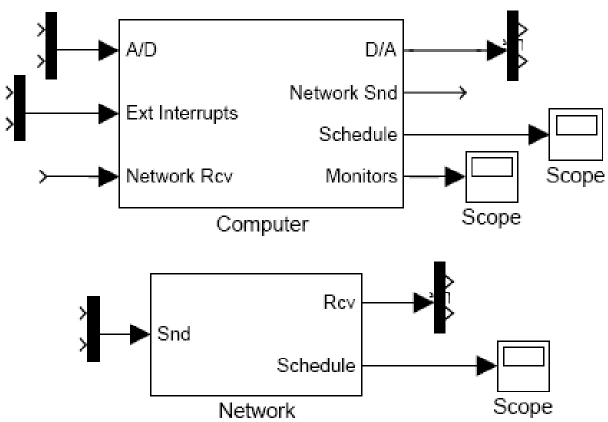


Fig. 3 Blocks of the TRUETIME simulation environment

2.3 Fuzzy logic

In Boolean logic, the function of the Boolean operators AND, OR, and INVERT is well specified. To use the OR operator, the values of 2 variables can be obtained which, the following results are obtained in different states:

1 and 1 → 1
1 and 0 → 1
0 and 1 → 1
0 and 0 → 0

However, these values are not definite in fuzzy logic, and their fuzzy distribution is described by the membership function. In this case, if the two input variables are fuzzy numbers, what will their output be? The answer to this question is determined by various fuzzy logics. In simple words, if we consider the union operator (which is equivalent to OR), then the result is equal to the maximum value of the input variables. For example, if C = A OR B then C = max (A, B). Now, if we consider the intersection operator, which is equivalent to AND, the result of this operator is equal to the lowest value of the input variables, that is, if C = A AND B, then C = min (A, B). In addition, if we consider the complement operator, which is equivalent to NOT, for any variable, its result is the complement of that variable or number. In other words, $\bar{x} = 1 - x$ represents the complement of the variable x.

2.4 Time dependent fuzzy logic

2.4.1 Fuzzy logics

As explained earlier, fuzzy logic may be a sequential or combined logic. However, the passage of time is not necessarily an important parameter. In fuzzy control, the method "if ..... then ....., otherwise ......" is followed, which in this method, the passage of time is not important.

The sequential and combinational processes are fuzzy and real-time processes in which, temporary reasoning is important. However, the existing fuzzy control methods are not entirely dependent on the specific result and are easily programmable and do not reflect the real-time evaluation of the control problems. Various approaches have been proposed to overcome this problem, but time has not been explicitly involved in these methods. In the next section, fuzzy logic is expanded to include time-dependent membership functions. This logic develops as a time-dependent membership logic. This logic is called the temporary fuzzy logic (TFL). Visually, TFL logic can be used to fill the logic matrix (Figure 4). In this figure, BL, TL, and FL are the Boolean multi-value logic, temporary logic, and fuzzy logic, respectively.

	Static	Dyna
Deterministic	B_L	T_L
Fuzzy	F_L	TFL

Fig. 4 Logic matrix table

2.5 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 5, a fuzzy controller consists of 4 parts, 2 of which performing the conversion operation:

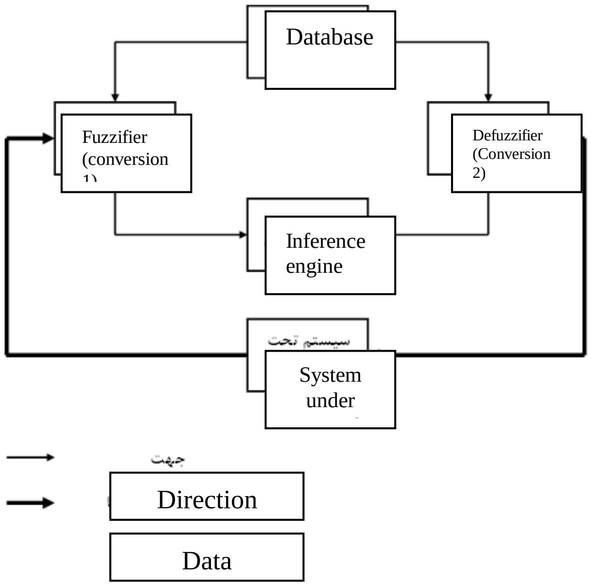


Fig. 5 Block diagram of a control system including fuzzy controller

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.

For example, if the input signal is small but positive, this signal belongs to a small positive fuzzy set, and if it is small but negative, it belongs to a small negative fuzzy set, and so the other fuzzy sets can be available as average positive, large positive, etc. In a typical fuzzy controller, the number of membership functions and their shape are initially determined by the user. The membership functions have values between 0 and 1 and determine the degree of belonging of a quantity to a fuzzy set. If belonging of a quantity to a fuzzy set is determined definitely, then its degree of belonging to this fuzzy set is one (in other words, this quantity belongs 100% to this fuzzy set). In contrast, if a quantity does not belong to a fuzzy set at all, its degree of belonging to this fuzzy set is zero.

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.

The inference engine is the brain of a fuzzy logic controller and is capable of simulating human decision making based on fuzzy idea and also concluding fuzzy control function by applying fuzzy logic rules. In other words, all input variables are converted by the fuzzifier into their own linguistic variables and the inference engine evaluates the set of IF-THEN rules in the database. In the next step, the result obtained from this evaluation, which is a linguistic value, is converted into a real output by the defuzzifier.

The second conversion, which is performed by the defuzzifier, transforms the fuzzy value of the output of the inference engine into real and numerical values by membership functions. There are several techniques for defuzzification, but the average method is used because of its simplicity of application and the simpler algorithm.

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 same as the difference of optimal speed and the real speed.

$$E(K) = \omega_{ref}(k) - \omega_r(K)$$

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):

$$E(K) = \omega_{ref}(k) - \omega_r(K)$$

$$CE(K) = E(k) - E(K-1)$$

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.

$$\cap(K) = \cap(k-1) + C\cap(K)$$

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. As mentioned earlier, the database is 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.

It should be noted that a classical controller with constant coefficients (such as the PID controller) cannot meet the three main goals mentioned for the fuzzy controller. The classical PI and PID controllers used in the ac drive system are usually adjusted by trial and error method. There are, of course, several techniques for initial adjustment of the controller parameters, the most common of which is based on the Ziegler-Nichols method. However, these techniques are often time-consuming and the controllers with constant coefficients cannot have acceptable dynamic performance in working conditions. The performance is often disrupted due to the non-linearity of the machine and variation in the parameters. Adaptive controllers can be used to resolve this problem, but convergence problems arise in these controllers. The main goal of optimizing the drive system is to reach the lowest overshoot, the minimum rise time and settling time. Of course, it is generally not possible to meet all these factors simultaneously, but it is possible to achieve the best state using some criteria.

3. Simulation and numerical studies

3.1 Simulation of fuzzy and PI controllers on inverted pendulum using TRUETIME:

Inverted pendulum modeling:
According to the figure, an inverted pendulum with a mass m and a length l m is 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:

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

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

The potential energy of the system is equal to:

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

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

Solving the Lagrange equation along x, θ we have:

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

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

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

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

Linearizing around the working point yields:

$$\begin{cases}(I + m\ell^2)\ddot{\theta} - mg\ell\theta = m\ell\ddot{x}\\ (M + m)\ddot{x} + b\dot{x} + m\ell\ddot{\theta} = F\end{cases} \quad (1)$$

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

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

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

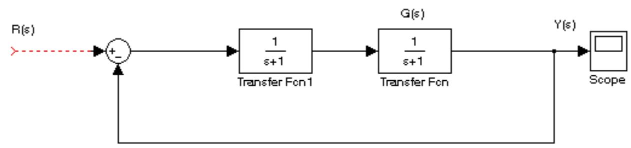
$$\frac{\theta(s)}{F(s)} = \frac{\frac{m\ell}{q}s^2}{s^4 + \frac{b(I + m\ell^2)}{q}s^3 - \frac{(M + m)mg\ell}{q}s^2 - \frac{bmg\ell}{q}s}$$

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

Simplifying this yields:

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

Proportional Controller:
K(s) = k1/s



For the step input:

$$e_{ss} = \frac{A}{1 + G(0)} = \frac{1}{1 + k_1k_e}$$

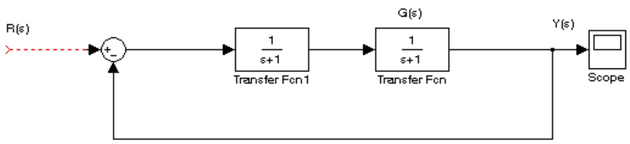
Therefore, if k 1 increases, the steady-state error decreases.
Compensator integrator:

$$K(s) = \frac{k_2}{s}$$

For the step input, since the system type is 1, ess = 0.

$$e_{ss} = \frac{1}{\hat{G}(0)} = \frac{1}{k_ek_2}$$

For the step input:
Therefore, if k 2 increases, the steady-state error decreases.
Derivative Compensator:
K = k3S-3



$$\frac{Y(s)}{R(s)}=\frac{k_3k_eS}{k_3k_eS+1+T_eS}=\frac{k_3k_eS}{1+(T_e+k_3k_e)S}$$

The derivative compensator increases the system response speed.

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:

$Rule\ i:$

$if\ (x\ is\ MF_p^1)\ AND\ (\dot{x}\ is\ MF_q^2)\ AND\ (\theta\ is\ MF_r^3)\ AND\ (\dot{\theta}\ is\ MF_s^4)$

$Then\ \ F=a_ix+b_i\dot{x}+c_i\theta+d_i\dot{\theta}+e_i$

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.

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 6 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.

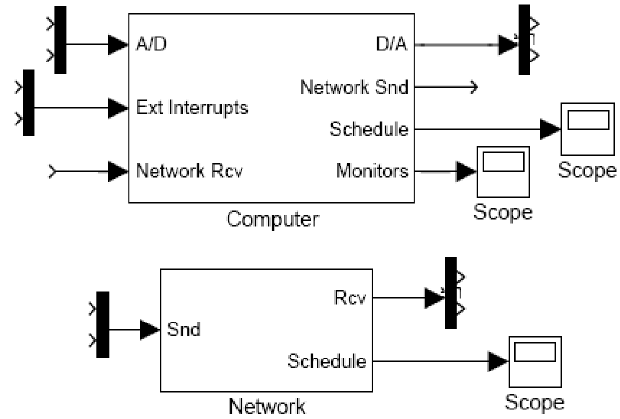


Fig. 6 Blocks of the TRUETIME simulation environment blocks

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 7.

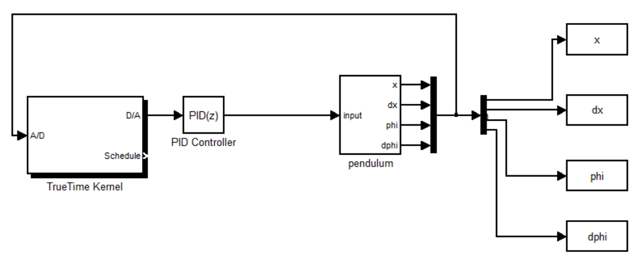


Fig. 7 Design of the inverted pendulum using the PID controller and the TRUTIME toolbox in the Simulink environment

In this section, the simulation is used with a T-S fuzzy controller. In this system, as shown in figure 8, the controller contains two inputs of d and de, respectively, of position and derivative of the position of the pendulum, and its conditions are as table ... and the output is three functions mf1, mf2, and mf3.

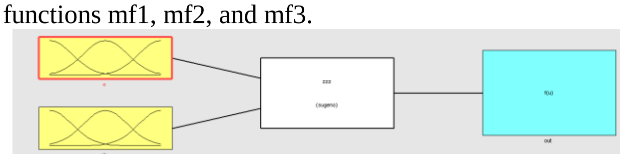
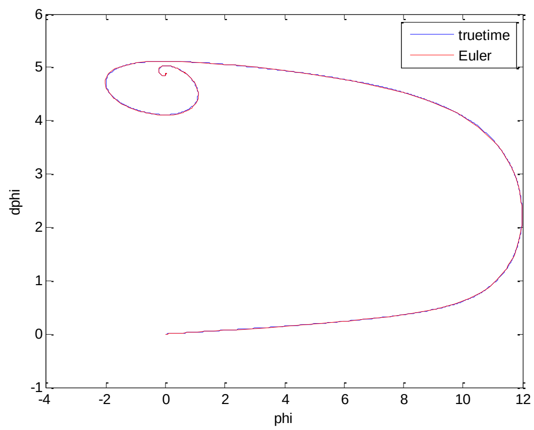
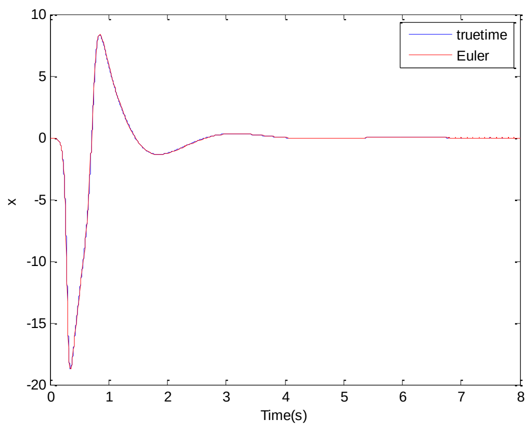
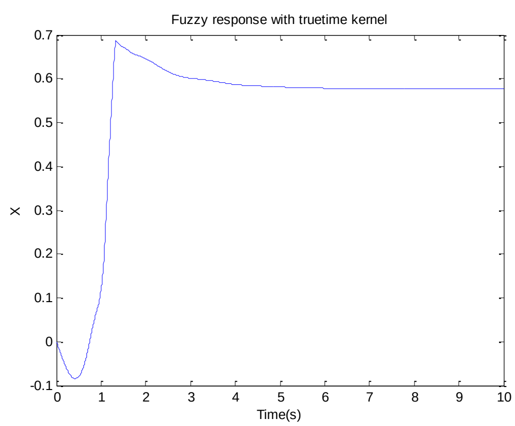


Fig. 8 Tagaki Sugeno fuzzy controller





It can be seen that using TRUETIME, a small time difference is observed relative to the Euler method and is executed earlier than this method. By designing the fuzzy controller as figures 4-8, the results were analyzed using TRUETIME and can be observed in figure 9.



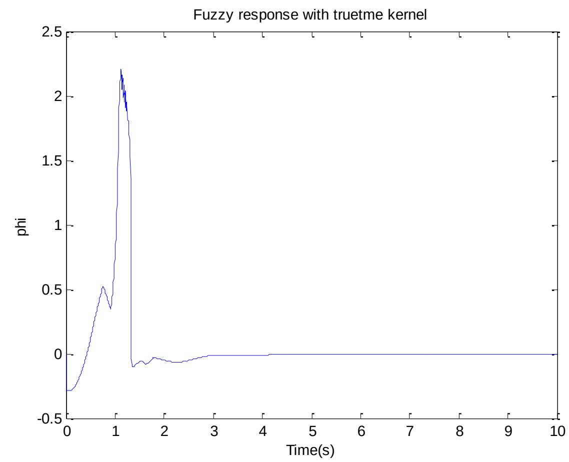


Fig. 9.

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.

3.2 Simulation on DC motor system:

Simulation on the DC servo motor system with fuzzy logic and TRUETIME toolbox:

In this simulation, the PID controller coefficients have been calculated using its error and derivate by the fuzzy logic. The DC motor model is as follows:

$$\dot{X} = \begin{bmatrix} -25 & 0 \\ 1 & 0 \end{bmatrix} x + \begin{bmatrix} 1 \\ 0 \end{bmatrix} u$$

$$y = \begin{bmatrix} 0 & 150 \end{bmatrix}$$

The variables x, u, and y are output, position, and control, respectively. $\omega_n^2 = 150$, $2\xi\omega_n = 25$ is the natural frequency $\omega_n = 12.25$ for the damping ratio $\xi = 1.02.$ (Figure 10).

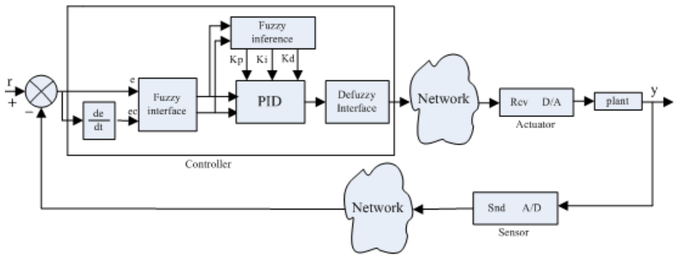


Fig. 10 Simulation on the DC servo motor system with fuzzy logic and TRUETIME toolbox

In addition, the simulation structure of the system in the Simulink environment is as follows (Figure 11):

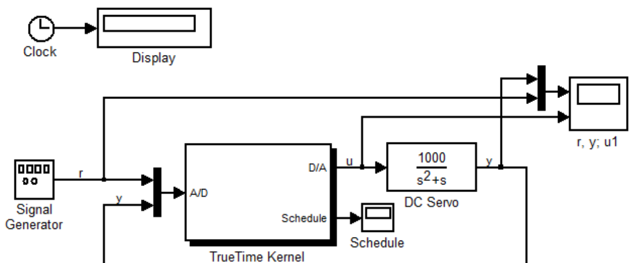
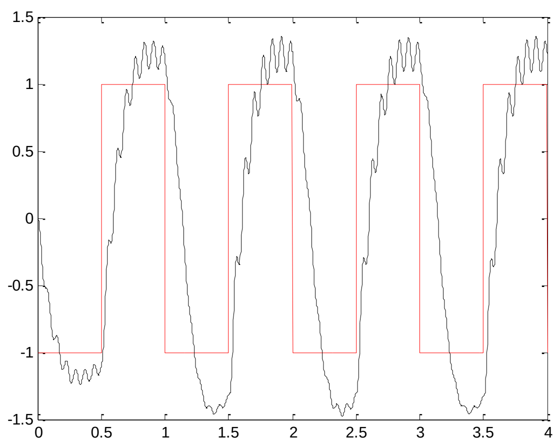
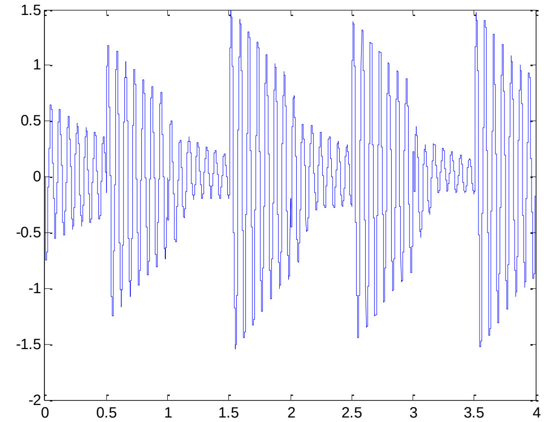


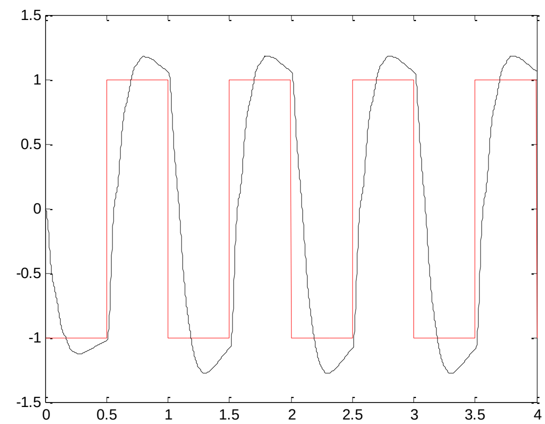
Fig.11 Simulation of the DC servo motor system in the Simulink environment

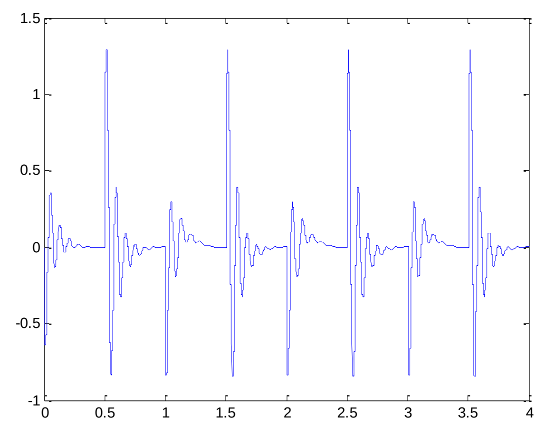
Simulation results for delay t = 1.049:



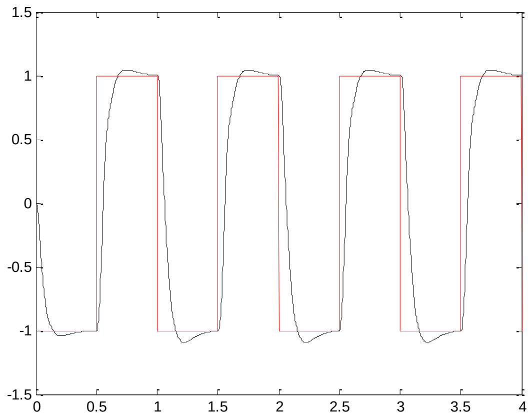


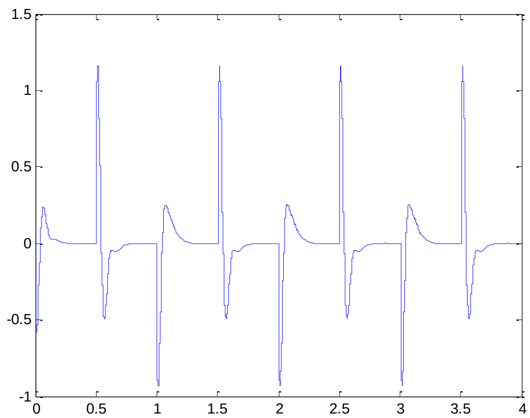
Simulation results for delay t = 0.089:





Simulation results for delay t = 0.049:





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.

4. Conclusion

In this study, a fuzzy control method was introduced for the NCS. Due to network characteristic changes- at almost all times- fuzzy control has a better performance compared to other methods which require a prior knowledge of the plant or network. To verify the feasibility, the NCS has been implemented to control the double inverted pendulum in the 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. Facing random network delays in the loop is much more difficult than random or periodic delays, since there are no criteria to ensure the stability of the NCS. NCS stability criteria 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. Most of the additional factors include concerns on network protocol prices, size, and usage intervals. The control methods presented in the present study cover various systems and protocols. 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. If network delays are limited and the system's final time are not the criteria, an event-based approach is preferred for system stability. 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] 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.

[2] 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.

[3] Luck, R., & Ray, A. (1994). Experimental verification of a delay compensation algorithm for integrated communication and control systems. International Journal of Control, 59(6), 1357–1372.

[4] 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.

[5] 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 .

[6] Ray, A. (1987). Performance evaluation of medium access control protocols for distributed digital avionics. Journal of Dynamic systems, Measurement, and Control, 109, 370–377.

[7] Luck, R., & Ray, A. (1994). Experimental verification of a delay compensation algorithm for integrated communication and control systems. International Journal of Control, 59(6), 1357–1372.

[8] 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.
[9] Nilsson, J., 1998. Real-time control systems with delays.Ph.D. dissertation, Lund Institute of Technology.
[10] O. zgu.ner, U. ., Go.ktas, H., Chan, H., Winkelman, J., Liubakka, M.,& Krotolica, R., 1992. Automotive suspension control through a computer communication network. Proceedings of the 1992 IEEE conference on control applications, Vol. 2 (pp. 895–900). Dayton,OH.
[11] Chan, H., & O. zgu.ner, U. . (1995). Closed-loopcontrol of systems over a communications network with queues. International Journal of Control, 62(3), 493–510.
[12] Kuo, B. C. (1987). Automatic control systems. Englewood Cliffs, NJ: Prentice-Hall.Li, Q., &Mills, D. L. (2001). Jitter-based delay-boundary prediction of wide-area networks. IEEE/ACM Transactions on Networking, 9(5),578–590.
[13] 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.
[14] G.oktas, F., 2000. Distributed control of systems over communication networks. Ph.D. dissertation, University of Pennsylvania.
[15] Rachana A. Gupta & Mo-Yuen Chow.2007. Overview of Networked Control Systems. North Carolina State University, Raleigh, NC 27695, USA.
[16] Ray, A., &Halevi, Y. (1988).Intergrated communication and control systems: Part II—design considerations. Journal of Dynamic Systems, Measurement, and Control, 110, 374–381.
[17] 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.
[18] Zhang, W., Branicky, M. S., & Phillips, S. M. (2001). Stability of networked control systems. IEEE Control Systems Magazine, 21(1), 84–99.
[19] 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.
[20] Amir Ali Mohaghpour, Ali Karimpour, Naser Paziz. 2008. Reviewing and comparing different methods of simulation of NCSs. Scientific-specialized electrical journal of Majlisi Azad University, First year / No 4, Spring 2008
[21] 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.
[22] Wen-juan zhang & Yu-fendeng & Lian-ming wang & Hong-wei zhang .2008. Fuzzy PID Control Method For Networked Control System With Constant Delays .2008. Proceedings of the Seventh International Conference on Machine Learning and Cybernetics, Kunming, 12-15 July 2008
[23] 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.
[24] 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.
[25] Tipsuwan, Y., & Chow, M.-Y., 1999. Fuzzy logic microcontroller implementation for DC motor speed control. The 25th annual conference of the IEEE industrial electronics society (IECON 99), Vol. 3 (pp. 1271–1276). San Jose, CA.
[26] 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.
[27] Yodyium Tipsuwan, Mo-Yuen Chow . 2003 . Control Methodologies in Networked Control Systems . Science Direct, Control Engineering Practice 11 (2003) 1099–1111.
[28] Abouelabbas Ghanaim & Georg Frey.2011. Modeling and Control of Closed-Loop Networked PLC-systems . 2011 American Control Conference on O'Farrell Street, San Francisco, CA, USA June 29 - July 01, 2011.
[29] 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
[30] 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.