Applied Mathematics & Information Sciences
*An International Journal*

http://dx.doi.org/10.18576/amis/100524

# Online Rotating PI Controller for NCS Over Communication Constraints

*Pooya Hajebi* and Seyed Mohammad Taghi AlModarresi*

Department of Electrical and Computer Engineering, Yazd University, P.O.Box 89195-741, Yazd, Iran

Received: 5 Jan. 2014, Revised: 25 Aug. 2014, Accepted: 27 Aug. 2014
Published online: 1 Sep. 2016

**Abstract:** A new method for controlling the plant in networked control systems (NCSs) is proposed. Network time delay and packet loss are two major drawbacks in data communication networks which make NCSs unstable. Unlike previous related research works, this new proposed rotating PI controller based method has the advantage of considering time delay and packet loss effects simultaneously. Time delay is estimated online and then used for tuning the PI controller by rotating the phase plane, while packet loss sequences are modeled by Markov chain. This novel method improves the performance compared to other methods, especially when packets are dropped consecutively and network time delays are large. In fact, the results show that with the network time delay as large as 600 ms, and packet loss occurring evenly, the index of the rotating PI controller performance will be improved by approximately two and a half times compared to the performance index of classical Smith predictor. This ratio will be improved by approximately eight and a half times compared to the performance index of PI controller. Furthermore, in the case that the packet loss occurs consecutively, the results show a ratio improvement of approximately three and ten for our suggested method in comparison to Smith predictor and PI controller, respectively.

**Keywords:** Data Communication Network, Markov Chain, Networked Control Systems, Network Time Delay, Packet Loss, Phase Plane, Rotating PI Controller.

## 1 Introduction

Networked control systems (NCSs) are spatially distributed systems in which digital communication networks are used as media to transmit control signals among sensors, actuators, and controllers [1]. NCSs could connect spatially distributed components and establish a flexible control structure which consequently provides expand ability for design and installation of control systems and easy maintenance. NCSs could be used in widespread fields such as mobile sensor networks [2], remote surgery [3], haptics collaboration over the Internet [4], and automated highway systems and unmanned aerial vehicles [5]. Murray et al. in [6] have pointed out the importance of control over networks in the future of control science. However, application of a network against several separate channels produces new constraints. Packet loss and stochastic time delay in NCSs are two major knotty challenges that were mentioned in [1], and still they are two first problems in these systems [7].

In [8], a method for improving the Smith predictor against uncertain network time delay was proposed while packet loss problem was neglected. In [9], a fuzzy inference system for determining PID controller coefficients in Smith predictor for wireless networked control system was proposed, but the range of time delay changes in network was assumed about 150 ms while again, it did not suggest any specific model for packet loss. We introduced an online adaptive fuzzy logic controller based on rules-table rotation in [10], in which the range of induced network time delay could increase to 600 ms. Optimization of the output fuzzy membership functions in the proposed controller by means of genetic algorithm, results in the system performance improvement [11]. Analysis of the packet loss problem is complicated for non-classic controllers like fuzzy logic controller thus, Wen and Li have studied on minimum packet loss sequence for classic PID controller in NCS based on Markov chain [12]. However, they have not proposed any PID controller structure and neglected the network time delay problem. In this paper, a PI controller

* Corresponding author e-mail: Hajebi@stu.yazd.ac.ir

with a rotating phase plane is introduced. The proposed controller is tested over the large time delay and high rate packet loss via Markov chain model. The proposed controller shows a robust performance not only against the network packet loss but also against the network time delay.

This paper is organized as follows: In Section 2, NCSs, modeling, stochastic time delay, and packet loss are introduced. Section 3, first describes a PI controller in order to control the position of an AC 400 W servo motor and next introduces a novel rotating PI controller by means of rotating phase plane using a trained neural network. Section 4 contains the related simulations and results. This paper ends with conclusion in Section 5.

## 2 Networked Control Systems

Due to expansion of communication networks around the world in recent years, shared commutation networks such as controller area network (CAN) and Ethernet could be used to transmit control and feedback signals. In this case, wiring costs and the demand for maintaining separate dedicated communication links for each control system are reduced. However, utilization of communication network to control systems is not a comprehensive solution and has its own constraints and challenges such as network time delay and packet loss [1,7] which can negatively affect the system performance. Figure 1 shows a common model of single-input single-output (SISO) networked control system. As shown in this figure, $t_1$ and $t_2$ display network time delays in control loop for forward (controller to actuator) and feedback (sensor to controller) directions, respectively. Measurement of round-trip time (RTT) could be significant as it should provide an accurate estimation for total time delay. As the distances between server node and client node increase, the time delay raises respectively. This may lead to more nodes to be included and hence gaining a higher RTT. To design a classical Smith predictor, the parameter, $t_m$, is set to a constant value which equals to average approximation of time delays in network loop. The RTT value could be used for rotating PI controller to compensate induced network time delay. Generally, the parameters of PI controller are constant during control process. This paper has suggested a method in which RTT is applied to a mapping neural network to calculate an angle for rotating the phase plane in a PI controller.

### 2.1 Networked Control System Model

Generally, a networked control system contains three main sections: controller, network, and plant. In Fig. 1, controller is defined by the following equations:

$$\begin{cases} \dot{x}_C(t) = A_C x_C(t) + B_C \hat{y}(t) \\ u(t) = C_C x_C(t) + D_C \hat{y}(t) \end{cases} \tag{1}$$

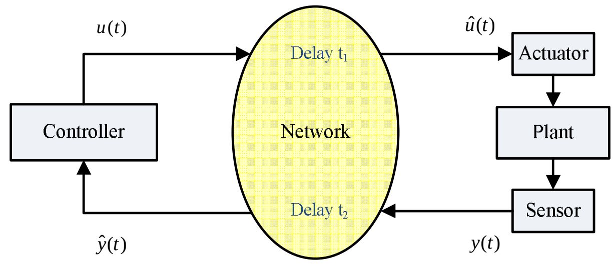


**Fig. 1:** A SISO NCS model.

where $x_C(t)$ is the state, $\hat{y}$ and $u$ are input and output of the controller, respectively. Plant is considered by the following equations:

$$\begin{cases} \dot{x}_P(t) = A_P x_P(t) + B_P \hat{u}(t) \\ y(t) = C_P x_P(t) \end{cases} \tag{2}$$

where $x_P(t)$ is the state, $\hat{u}$ and $y$ are input and output of the plant, respectively. In networked control systems, the effect of network can be modeled by two major parameters: time delay and packet loss. Network time delays are defined as the time differences between the input and output signals of the network. The following equations define $e_1$ and $e_2$ as two network error parameters:

$$\begin{aligned} e_1 &= u - \hat{u}' \,,\ \hat{u}' = u(t - t_1) \\ e_2 &= y - \hat{y}' \,,\ \hat{y}' = y(t - t_2) \end{aligned} \tag{3}$$

Packet loss phenomenon is described by two loss vectors: loss vector of control signal and loss vector of plant output. The loss vector of control signal is defined by (4).

$$\theta = [\theta_1, \theta_2, \ldots, \theta_m] \tag{4}$$

where $\theta_k$ indicates whether the control signal, $\hat{u}'_k$, is transmitted successfully ($\theta_k = 1$) or it is lost ($\theta_k = 0$). Based on this definition, the control signal of plant input, $\hat{u}_k$, is obtained by (5).

$$\hat{u}_k = \theta_k \cdot \hat{u}'_k \tag{5}$$

The loss vector of plant output is defined by (6).

$$\phi = [\phi_1, \phi_2, \ldots, \phi_m] \tag{6}$$

where $\phi_k$ indicates whether the output, $\hat{y}'_k$, is transmitted successfully ($\phi_k = 1$) or it is lost ($\phi_k = 0$). Based on this definition, the feedback signal of controller input, $\hat{y}_k$, is obtained by (7).

$$\hat{y}_k = \phi_k \cdot \hat{y}'_k \tag{7}$$

## 2.2 Network Time Delay and Network Packet Loss Models

The controller or plant cannot always access to the most updated signal of the plant or controller, respectively, because time delays and packet loss occur in a communication channel. Therefore, knowledge about network time delay or network packet loss patterns are the most important issues to be considered in NCS.

### 2.2.1 Network Time Delay Model

In a networked control system, the control commands are transmitted over the network in a packet form. Since packets can be transmitted from different ways then different time delays are resulted for two consecutive packets and causes jitter. For example in forward direction of control process, this problem causes the sensor receives the later packet earlier. Jitter may cause undesirable effects on control performance. The total communication latency of time delays in forward direction (from controller to actuator) and backward direction (from sensor to controller) is defined as network-induced time delay in NCSs. Time delays in control loop have harmful effect on system performance. Time delay in the network is a random process. Any changes in time delay would directly affect the control performance. The range of network time delay is the other important factor that affects the control performance directly. Du et al. in [9] have assumed the maximum of time delay is about 150 ms and in [13], interval of time delays is between 0 and 200 ms. However, in reality the range of time delays could exceed 500 ms in network [8]. Thus, in this paper we have used a random process with uniform distribution changing in constant steps to model network time delays. These constant steps show the modes of different delays (e.g. 200 ms, 400 ms, and 600 ms).

### 2.2.2 Packet Loss Sequence Model

In a simple model, the packet loss distribution is assumed based on a Bernoulli process [1,14] but this distribution cannot model the behavior of network in NCS preciously, nevertheless still in some recent studies, Bernoulli process has been used to model the network packet loss in NCS [15]. Packet loss over a communication link causes probabilistic behavior due to random inherent of communication channel. One of the most popular parameters used for introducing the packet loss phenomenon is packet loss rate, i.e. the average of packet loss. The packet loss rate is defined as the percentage of missing packets to all packets ratio. By this definition, only the rate of packet loss is known. Therefore, the distribution of packet loss is unknown. However, the patterns and distribution of packet loss sequences in networks play vital role in real-time applications such as control over network, voice over network, and video over network. For example, suppose that in ten packets, five packets are dropped. Thus, the network packet loss rate is 50 percent. If the packet loss happens evenly, it will not have any significant effect on the performance of NCS. If the packets are dropped consecutively, the performance of control system will reduce strongly. To model packet loss sequence with a specific packet loss rate, Imer et al. used a Bernouli random process, which looks at the most updated packet [16]. However, for controlling a plant, the controller and plant require to know the most updated packet and previous signal. When the packet loss happens consecutively, the controller or plant cannot access to feedback signals or control signals for a long period of time, respectively. This problem is more significant when the control or feedback signals change quickly. Each packet loss sequence can have different probability. This probabilistic behavior is not considered in Bernouli random process. Yuksel and Basar in [17] proposed a Markov chain model to capture the state of reliability of the network. Wen and Li in [12] considered only the packet loss sequence and did not look at network time delay problem whilst they did not propose any controller or any simulation results. The number of states in Markov chain model is derived from the transfer functions of controller and plant. When the control signals and feedback signals want to be transmitted over data communication network, they should be changed to digital signals. After receiving digital signals by actuator or controller, they usually will be transformed to analog signals. The general transfer function of the controller or plant is according to (8).

$$G(z) = \frac{C(z)}{R(z)} = \frac{\beta_0 z^{-n} + \beta_1 z^{-n+1} + \cdots + \beta_n}{\alpha_0 z^{-n} + \alpha_1 z^{-n+1} + \cdots + \alpha_n} \tag{8}$$

Usually, digital signals are transformed to analog signals using zero-order hold (ZOH) method. It means that recent digital signals are held as the value of the analog signal until the next sampling moment. It leads to one sampling interval delay. Therefore, always the power of z in the numerator of the transfer functions is lower than the dominator at least by one. Then the transfer function in (8) can be rewritten as the following form:

$$G(z) = \frac{C(z)}{R(z)} = \frac{\beta_0 z^{-n} + \beta_1 z^{-n+1} + \cdots + \beta_{n-1} z^{-1}}{\alpha_0 z^{-n} + \alpha_1 z^{-n+1} + \cdots + \alpha_n} \tag{9}$$

The (9) can be represented in (10) as difference form,

$$c[k] = \frac{1}{\alpha_n}[\beta_{n-1} r[k-1] + \cdots + \beta_0 r[k-n] - \alpha_{n-1} c[k-1] - \cdots - \alpha_0 c[k-n]] \tag{10}$$

where $n$ is the order of the transfer function. From (10), it is concluded that $n$ previous input signals and $n$ previous output signals, which affect the output ($c[k]$) directly, are needed to compute the value of $c[k]$. Lack of one of these signal packets will cause the $c[k]$ not to reach its real value. Therefore, to consider the performance of NCS, it is required to consider the $n$ previous signals [12]. The feedback signal, which is produced by sensor, is the input of controller and the control signal, which is produced by controller, is the input of actuator. Packet loss occurs in both the forward and backward paths and affects the performance of NCS. Thus, it is important to consider both controller and plant transfer functions. They can have different orders. For modeling the packet loss, however, the maximum order between controller transfer function and plant transfer function should be selected. If the maximum order between these two transfer functions is three then three consecutive packets should be selected for analyzing the packet loss phenomenon. In this case, there are $2^3 = 8$ states for either receiving or dropping these three packets. Markov chain model can be used for representing these 8 states. Markov chain model is a discrete stochastic process with $p$ states denoted as $X(p)$ for $p = 0, 1, \ldots, P-1$, and $P > 1$. The transition probability from one state to another state is the probability that whether the next packet will be received or dropped. To model packet loss using Markov chain, the state is shown by bit string. The rightmost bit indicates the situation of recent packet. When the packet is not received, this bit is "0" and when the packet is received successfully this bit is "1". Also in Markov chain model, each state can transit to only two other states [12]. Suppose the recent state is "001". This state can transit only to "010" if the next packet is dropped, or "011" if the next packet is received. Therefore, for each state, an error probability can be determined that shows the probability of transition to two other states. Thus, both the rate of packet loss and the probabilities of patterns can be determined using Markov chain model. To illustrate the process, consider the following sequence of packet situations, where "1" represents the received packet and "0" represents the dropped packet.

$1 \to 0 \to 0 \to 1 \to 0 \to 1 \to 1 \to 1 \to 1 \to 0 \to 0 \to 0 \to 0 \to 1$

This sequence can be shown via Markov chain model. The 4-states, 8-states, 16-states, and 32-states Markov chain models for this sequence are represented in Table 1. For example in 8-states, the first state is "100". This means that the first bit, the second bit, and the third bit are "1", "0", and "0", respectively. When the fourth bit is "1", the next state will be "001" as shown in the second row of 8-states in the Table 1. Figure 2 shows the 8-states Markov chain model with its error probabilities. The green continuous lines indicate that in the next transition, packet has been received successfully and the red dashed lines indicate that packet loss has occurred in the next transition. $T_8$ in (11), is the Markov state transition matrix for Fig. 2. In this matrix, $e_0$, $e_1$, $e_2$, $e_3$, $e_4$, $e_5$, $e_6$, and $e_7$ are the probabilities of packet loss patterns. They can be different if some patterns may occur more, but if there is not any knowledge about their prior probabilities, these values will be assumed to be equal.

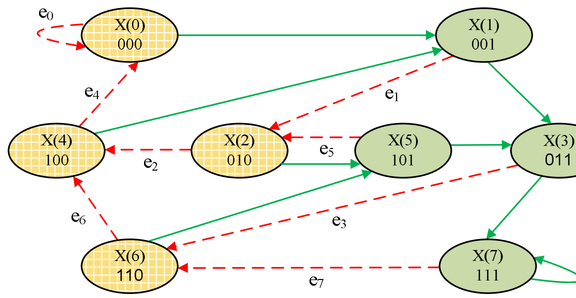


**Fig. 2:** 8-states Markov chain model for packet loss sequences.

$$T_8 = \begin{bmatrix} e_0 & 1-e_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e_1 & 1-e_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e_2 & 1-e_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e_3 & 1-e_3 \\ e_4 & 1-e_4 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & e_5 & 1-e_5 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e_6 & 1-e_6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & e_7 & 1-e_7 \end{bmatrix} \quad (11)$$

**Table 1:** 4-states, 8-states, 16-states, and 32-states Markov chain models for a sequence of packet situations.

| 4-states | 8-states | 16-states | 32-states |
|---|---|---|---|
| 10 | 100 | 1001 | 10010 |
| 00 | 001 | 0010 | 00101 |
| 01 | 010 | 0101 | 01011 |
| 10 | 101 | 1011 | 10111 |
| 01 | 011 | 0111 | 01111 |
| 11 | 111 | 1111 | 11110 |
| 11 | 111 | 1110 | 11100 |
| 11 | 110 | 1100 | 11000 |
| 10 | 100 | 1000 | 10000 |
| 00 | 000 | 0000 | 00001 |
| 00 | 000 | 0001 | |
| 00 | 001 | | |
| 01 | | | |

## 3 Rotating PI Controller for NCS

One of the most broadly applicable controllers in industrial processes is proportional-integral-derivative (PID) controller. A PID controller calculates an error value as the difference between a measured output variable and a desired set point. The controller attempts to minimize the error by adjusting control inputs of the process. More than 90 percent of industrial controllers are still designed based on PID methods. Their good performance in different functional conditions caused their widespread applications and sometimes because of their simple scheme, PID can be implemented in a simple method. An increasing research attention has recently been paid to the PID control design for NCS. In this section, firstly the PI controller and the classical Smith predictor are described. In the next step rotating PI controller is proposed, which could improve the performance of NCS.

### *3.1 PI Controller*

Proportional-integral (PI) controllers are more common to use. While derivative action is sensitive to the measurement noise, the integral term may facilitate the system to reach its target value. The transfer function of a PI controller is obtained from (12).

$$G_c = K_P + K_I \frac{1}{S} \tag{12}$$

As shown in (12), this controller has two terms: the proportional and the integral terms. Also it has two constant parameters. $K_P$ and $K_I$ are the proportional gain and the integral gain, respectively. While the proportional term depends on the present errors, the integral term depends on the accumulation of the past errors. To analyze a PI controller in a networked control system, first the output of plant is measured and then it would be compared with a reference signal. This comparison generates the error signal. The plant used for the simulation is an AC 400 W servo motor. An encoder with gain of $10^4$ P/R is considered for measuring the position of servo motor as output. The coefficients of the PI controller, $K_P$ and $K_I$, for this plant are set to be $10^{-4}$ and $10^{-8}$, respectively [8]. The transfer function of the plant is obtained from (13).

$$G_P(s) = \frac{10^4(0.058s + 3.221)}{s(0.0001s^2 + 0.019s + 1)} \tag{13}$$

### *3.2 Classical Smith Predictor*

A classical Smith predictor is utilized in this paper to compare the results. Figure 3 shows the structure of Smith predictor in which $G_C$ is the common PI controller introduced in Section 3.1, $G_P$ and $\hat{G}_P$ are the transfer functions of the plant and its estimation. The Smith tuning parameter for compensation of time delay, $t_m$, is a constant value which equates approximately to the total time delays in control process. The performance of system would be better off when calculating the exact total time delay. Here, $t_m$ is set to be 200 ms for the simulations. As mentioned $\hat{G}_P$ is the estimation of $G_P$.

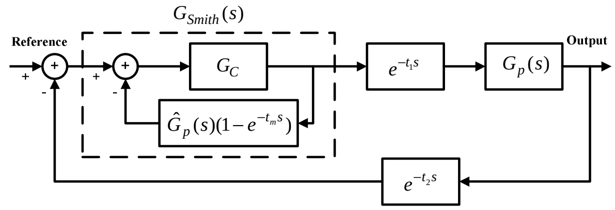


**Fig. 3:** Classical Smith predictor in networked control systems.

Hence it makes imparity among these transfer functions which in turn results in instabilities and increasing the error of system. It could be one of the important issues in classical Smith predictors. In the simulations of this paper the $G_P$ and $\hat{G}_P$ are ideally assumed to be equal.

### *3.3 Rotating PI Controller*

In the suggested rotating PI controller, firstly RTT is estimated between two nodes of network [8,11], then this estimated value would be utilized for rotating the phase plane in PI controller via a mapping neural network. To be more accurate in this method, the PI controller introduced in Section 3.1 would be integrated with an online neural network. The value of RTT would gain an estimation of online time delay which in turn provides the amount of rotation angle of PI phase plane by a mapping. As time elapses, the error and its derivative values move on the phase plane in a spiral path during the control process. Time delays prevent the application of the error and its derivative values on desired time as a requirement for producing the proper control signals. Thus, the application of this suggested method with a rotating phase plane could be a solution for these problematic time delays and improve the system performance against packet loss. This paper has suggested a control method which integrates PI controller with a two-layer multilayer perceptron (MLP) neural network shown in Fig. 4. The input and output of the neural network are RTT and rotation angle, respectively. This neural network has two neurons in the first layer and one neuron in the second layer. The “tansig” and “purelin” shown in equations (14) and (15) are applied for the transfer functions of the first and second layers in the neural network, respectively.

$$tansig(n) = \frac{2}{1 + e^{-2n}} - 1 \tag{14}$$

$$purelin(n) = n \tag{15}$$

Several time delays are used as the input of data set to train the neural network. In this case, the rotation angles of these time delays are mined manually and then these values would be applied for training the neural network.

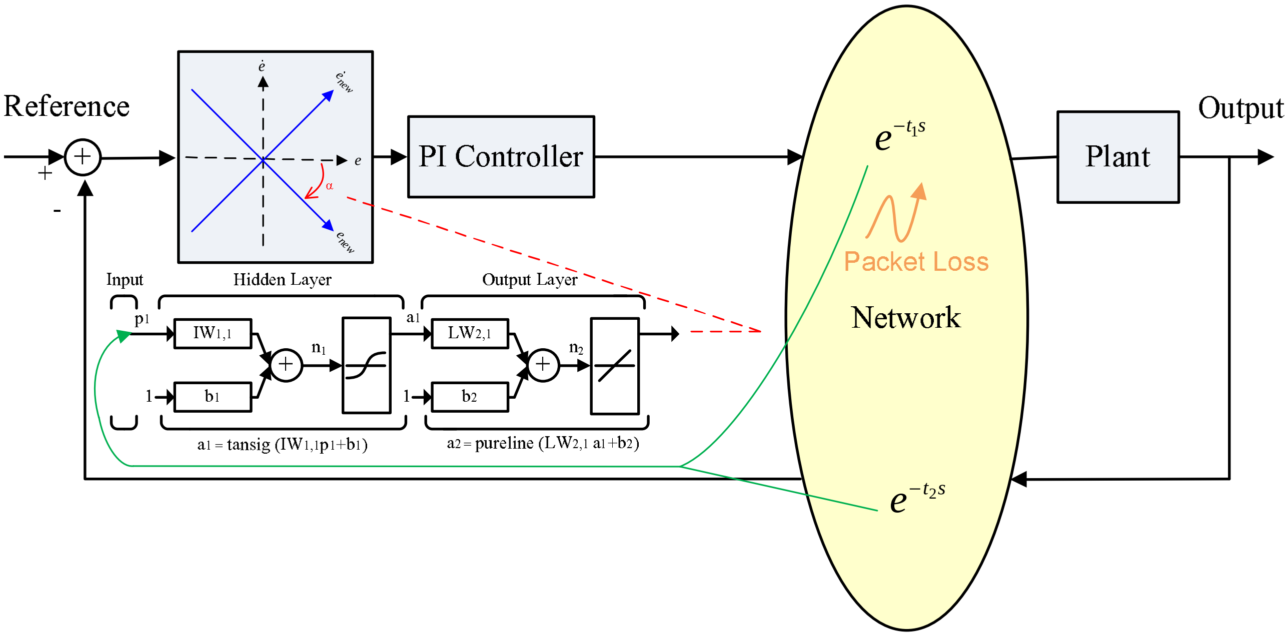


**Fig. 4:** Online rotating PI controller.

The output to input ratio of the neural network is shown in Fig. 5. The rotation angle's value which is determining for rotation of phase plane in PI controller would shift alternatively according to the value of RTT. The parameters of rotating PI controller, $K_P$ and $K_I$, are equal to the values used in Section 3.1. The phase plane rotation structure of rotating PI controller and trend of rotation are shown in Fig. 6. From this figure, the relation between the old coordinates and new coordinates is obtained. Equation (16) depicts the mapping relation of the error and its derivative values in new coordinates in which matrix A, a rotation transformation matrix, is shown in (17) and it rotates coordinates by the angle of $\alpha$ radian. Rotation of coordinates results in that the error signal and its derivative would apply on desired time based on network time delay. For example if the network time delay is about 300 ms, and there is not any time delay compensation, each control signal will be applied to the plant with 300 ms time delay. Thus, the plant will not receive the desired control signal. By rotating coordinates, control signal is estimated based on network time delay and then will be applied to the plant. Due to using PI controller, the packet loss effect in the network could be considered. Thus, this new method can compensate network time delay while the packet loss effect in network is considered.

$$\begin{bmatrix} e_{new} \\ \dot{e}_{new} \end{bmatrix} = A \begin{bmatrix} e \\ \dot{e} \end{bmatrix} \tag{16}$$

$$A = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix} \tag{17}$$

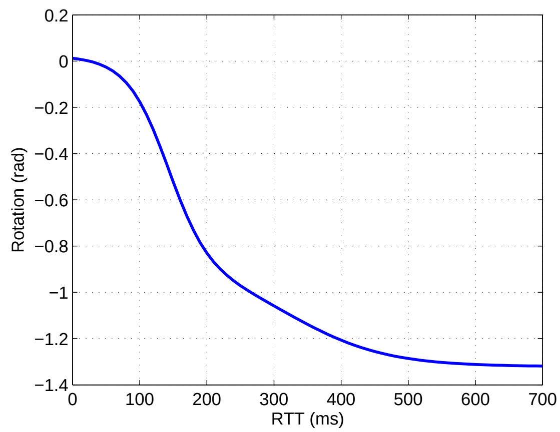


**Fig. 5:** Output vs. input of neural network.

From complexity viewpoint, although the algorithm needs some time to train neural network, this time consuming stage does not affect the system performance during the control process because the network training process does not repeat when the system is in operation mode. While the neural network was trained, since the assumed neural network has one input and one output and contains two layers, the angle of rotation is obtained by two times multiplying matrices and two times summing results and biases. The obtained value (neural network output) is a one-dimensional element used as angle of rotation. The error and its derivative are mapped to their

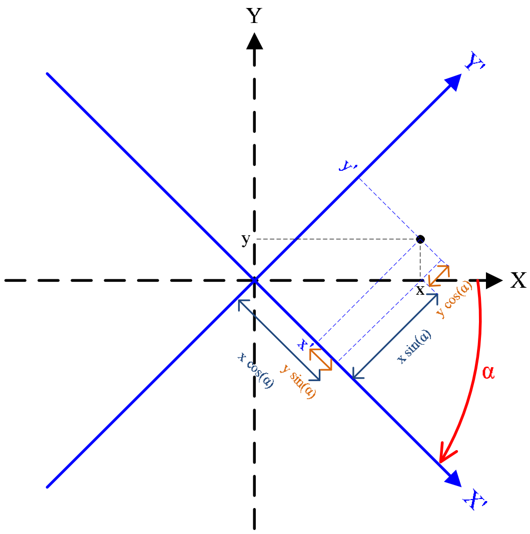


**Fig. 6:** Rotation of coordinates.

new values using rotation transformation matrix. The mapping operation is done using multiplying the error and its derivative vector (two-by-one vector) by rotation transformation matrix (two-by-two matrix). Therefore, the calculation time of proposed method compared to large values of time delays in digital network is negligible. The calculation is simple and it does not have any effect on the system performance.

## 4 Simulation Results

In this paper, the NCS consists of four main parts: rotating PI controller, neural network, plant, and communication network. In order to have an accurate analysis of NCS, each part should be considered separately. To plot the step response of system using MATLAB the state-space equations are applied. For this purpose, the transfer functions of controller and plant are converted to continuous state-space equations. Since the data transmitted through the network have digital inherent, these continuous state-space equations need to be converted to discrete state-space equations [10,11]. Discrete state-space form of the transfer function for the plant shown in (13) is represented in (18) and (19).

$$x[k+1] = \begin{bmatrix} 0.0066 & -0.2973 & 0 \\ 0.4870 & 0.7295 & 0 \\ 0.0035 & 0.0089 & 1 \end{bmatrix} x[k] + \begin{bmatrix} 7.7924 \\ 7.0909 \\ 0.0277 \end{bmatrix} u[k] \quad (18)$$

$$y[k] = \begin{bmatrix} 0 & 22.13 & 1228.7 \end{bmatrix} x[k] \quad (19)$$

Several random time delays are generated in the simulations to evaluate the performance of NCS. Total time delay (RTT) is the summation of the forward-command time delay, $t_1$, and the feedback time delay, $t_2$. By training the neural network using data set, the values of biases and weights are derived which are shown in (20)-(23).

$$IW_{1,1} = \begin{bmatrix} 0.0164 \\ 0.0070 \end{bmatrix} \quad (20)$$

$$b_1 = \begin{bmatrix} -2.3288 \\ -2.2803 \end{bmatrix} \quad (21)$$

$$LW_{2,1} = \begin{bmatrix} -0.4548 & -0.2184 \end{bmatrix} \quad (22)$$

$$b_2 = -0.6479 \quad (23)$$

In the simulations, the sampling time is considered to be 0.01 second. In this paper, the simulations and comparisons of the step responses are provided among three controller types: 1) Rotating PI controller; 2) Classical Smith predictor with PI controller; and 3) PI controller.

The order of PI controller transfer function is one and the order of plant transfer function is three. Thus, the maximum of these orders is three. Therefore, the dimensions of Markov transition matrices are 8 by 8. In networked control systems which are real-time systems, unlike data transfer applications, the information at current time is important and the past measurements become worthless. Therefore, in NCS, protocols such as user datagram protocol (UDP) are used. In UDP unlike transmission control protocol (TCP), receiving packets by destination is not verified and retransmission does not occur. In other words, there is no guaranty that packets are received successfully. Therefore, depends on distance, network traffic, number of nods, and noise, packet loss rate may reach values over 70 percent. Here, six matrices are described to generate packet loss sequences in different conditions. The packet loss rates of $T_{20\%eq}$, $T_{50\%eq}$, and $T_{70\%eq}$ are 20%, 50%, and 70%, respectively while the probabilities of packet loss patterns are equal. $T_{20\%neq}$, $T_{50\%neq}$, and $T_{70\%neq}$ are the Markov transition matrices with packet loss rates of 20%, 50%, and 70%, respectively while the probabilities of packet loss patterns are not equal. These values are according to the relevant matrix in which some special patterns occur with higher probability. The elements for all of these transition matrices are as follows: $T_{20\%eq}$, $T_{50\%eq}$, and $T_{70\%eq}$ are $T_8$ when $e_0 = e_1 = e_2 = e_3 = e_4 = e_5 = e_6 = e_7$ and they are equal to 0.2, 0.5, and 0.7, respectively and $T_{20\%neq}$, $T_{50\%neq}$, and $T_{70\%neq}$ are as follows: $T_{20\%neq}$ is $T_8$ when $e_0 = e_7 = 0.2$, $e_1 = e_2 = e_3 = e_5 = 0.1, e_4 = e_6 = 0.4$; $T_{50\%neq}$ is $T_8$ when $e_0 = 0.9, e_1 = e_5 = 0.3, e_2 = e_6 = 0.7, e_3 = 0.2, e_4 = 0.8, e_7 = 0.1$; $T_{70\%neq}$ is $T_8$ when $e_0 = e_2 = e_4 = e_5 = 0.9, e_1 = 0.5, e_3 = 0.4, e_6 = 1, e_7 = 0.1$.

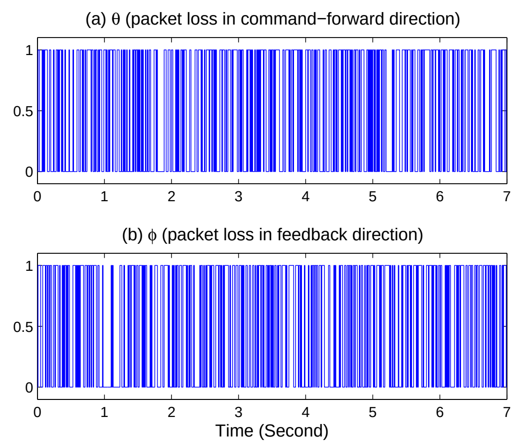


**Fig. 7:** Markov chain model (Markov transition matrix is $T_{50\%eq}$): a) Packet loss sequence in command-forward direction; b) Packet loss sequence in feedback direction.

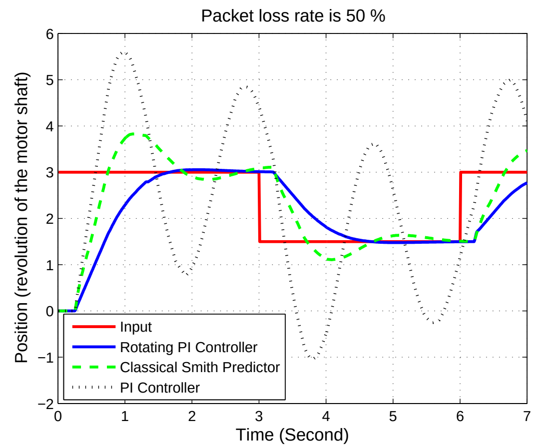


**Fig. 9:** Position (revolution of the motor shaft) control for three different controllers (Markov transition matrix is $T_{50\%eq}$).

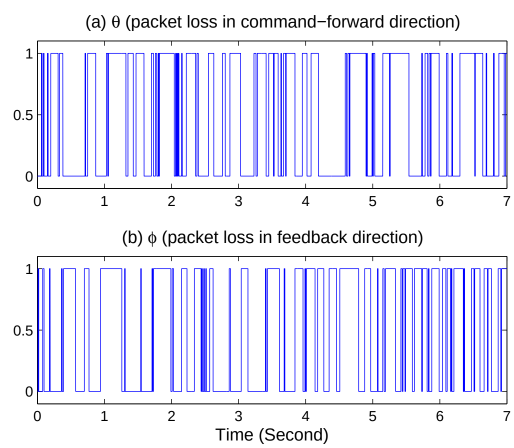


**Fig. 8:** Markov chain model (Markov transition matrix is $T_{50\%neq}$): a) Packet loss sequence in command-forward direction; b) Packet loss sequence in feedback direction.

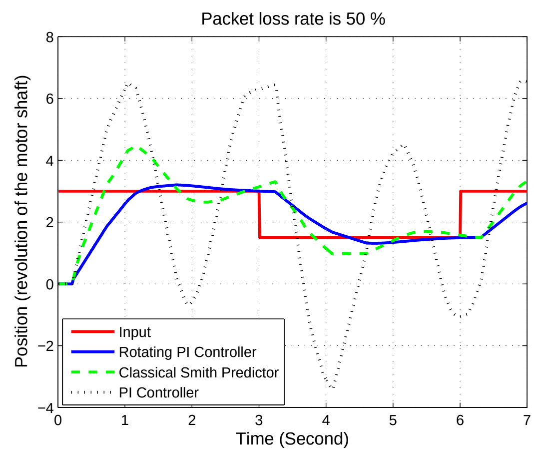


**Fig. 10:** Position (revolution of the motor shaft) control for three different controllers (Markov transition matrix is $T_{50\%neq}$).

The packet loss sequences applied for command-forward and feedback directions are according to Fig. 7 (a) and (b), respectively, while the packet loss rate is 50% and the probabilities of packet loss patterns are assumed to be equal. Figures 8 (a) and (b) show similar generated sequences with 50% packet loss rate and non-equal packet loss pattern probability. Although the packet loss rate is 50% in both figures, due to difference between probabilities of packet loss patterns in $T_{50\%neq}$ some states for this matrix may occur more and some other states may occur less. In these cases, the number of states in which packet loss occurred consecutively is more and it causes much effect on networked control systems.

Figure 9 and Fig. 10 are simulation results for three types of controllers based on packet loss sequences illustrated in Fig. 7 and Fig. 8, respectively. For these simulations, the total network time delay is considered as a uniform random process with mean of 400 ms and distribution interval between -10 ms and 10 ms. These figures show the NCS in order to control the position (revolution of the motor shaft) of plant via network. Here, the goal is to have the output follows the input more closely. When the controller or plant did not receive the packet, the last previous received signal value is replaced as current one. As illustrated in these figures for either of

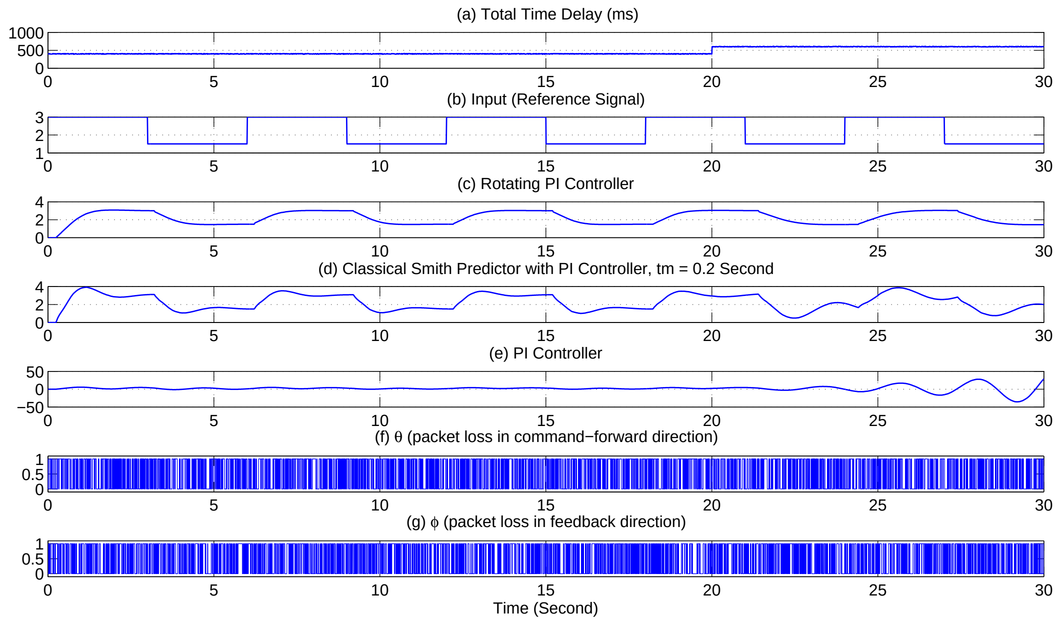


**Fig. 11:** Simulation results (The transition matrix is $T_{50\%eq}$): a) Time delay; b) Reference signal; c) Position (revolution of the motor shaft) for Rotating PI Controller; d) Position (revolution of the motor shaft) for classical Smith predictor; e) Position (revolution of the motor shaft) for PI controller; f) Packet loss sequence in command-forward direction; g) Packet loss sequence in feedback direction.

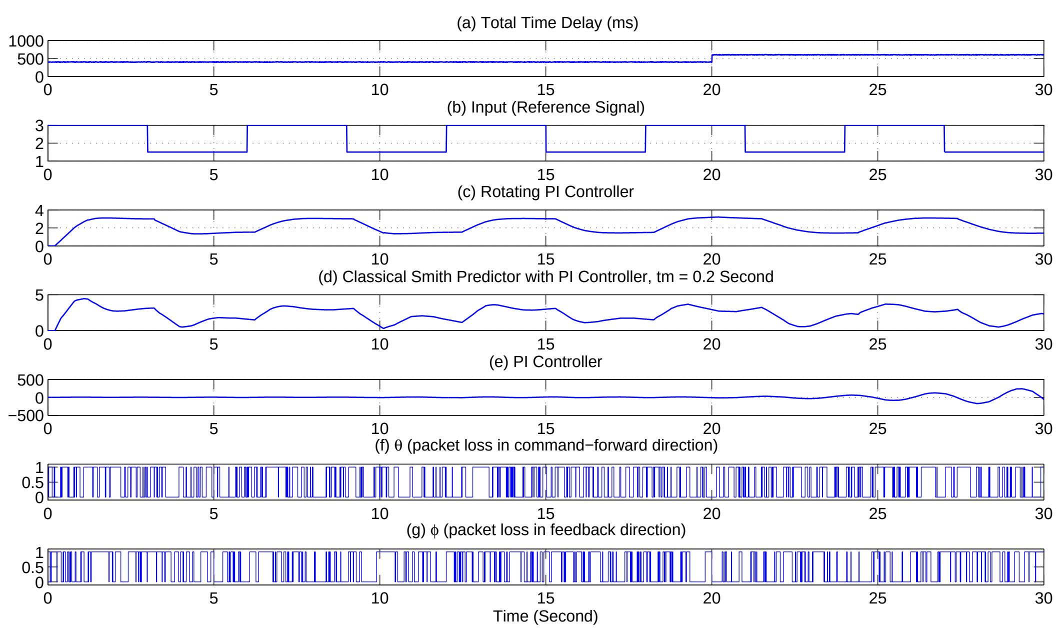


**Fig. 12:** Simulation results (The transition matrix is $T_{50\%neq}$): a) Time delay; b) Reference signal; c) Position (revolution of the motor shaft) for Rotating PI Controller; d) Position (revolution of the motor shaft) for classical Smith predictor; e) Position (revolution of the motor shaft) for PI controller; f) Packet loss sequence in command-forward direction; g) Packet loss sequence in feedback direction.

cases, the proposed controller follows the input accurately, with no significant overshoot.

Another important point is that in the case in which the probabilities of packet loss patterns are not equal (e.g. Fig. 10), the first peak of Smith predictor and the first peak of PI controller are over than 4 and 6, respectively, also the second peak of PI controller is over than 6 which is less than 5 in equal pattern probability case.

In order to conduct more precise evaluation, simulations are carried out when network time delay changes in step units, and reaches about 600 ms, as shown in Fig. 11 and Fig. 12. In these figures, the total network time delay is considered as a uniform random process with mean of 400 ms and distribution interval between -10 ms and 10 ms, in which its mean increases to 600 ms after the 20th second. To consider packet loss effect, the results are illustrated in Fig. 11 when the state transition matrix of Markov chain is $T_{50\%eq}$ and they are illustrated in Fig. 12 when the state transition matrix of Markov chain is $T_{50\%neq}$. Results show that the rotating PI controller offers a better performance compared to other controllers. When the probabilities of packet loss patterns are different, with the sudden increase of the time delay (Fig. 12), the effect of packet loss is more perceptible on classical Smith predictor and PI controller. In all of these figures, the output signal of rotating PI method does have smaller overshoot and faster response compared to other controllers. Therefore, this controller is preferable for networked control system purposes. The results in [9] indicate that system response would degrade with network time delays over 200 ms. In [9], it was assumed that the maximum of time delay is about 150 ms while this maximum value was applied separately in command and feedback directions. Despite considering the packet loss issue, our suggested method shows more improved response especially when the time delay is over 200 ms. Even in the time delays about 600 ms, deformation in step response is negligible compared to other investigated methods. In spite of this large value of time delay inducing continuously, results in Fig. 11 and Fig. 12 show that, irrespective of the probabilities of patterns in packet loss sequences, this does not have any significant effect on the step response of the proposed method. To evaluate the results analytically, a performance index should be used. In this case, which is a position control example, the aim is to have the output follows the input more closely. The closer output follows the input the more accurate is the performance. Thus, the integral of time multiplied by absolute error (ITAE) can be a proper index for comparing results. ITAE is commonly good performance index in designing PID controllers. Equation (24) shows the mathematical formula of ITAE.

$$ITAE = \int_{0}^{\infty} t|e(t)|dt \tag{24}$$

Where $t$ is the time and $e$ is the difference between output and reference signals in control process.

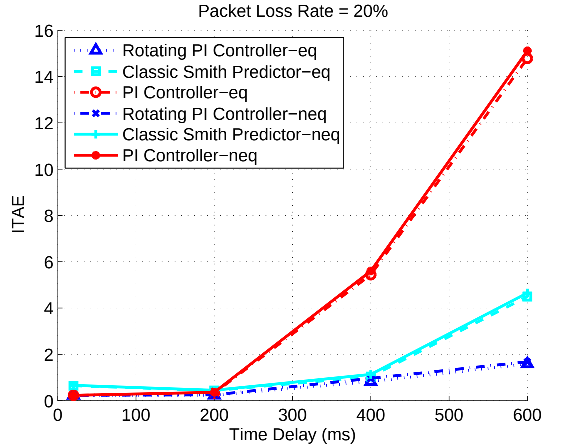


**Fig. 13:** ITAE for three controllers (Packet loss rate is 20 %).

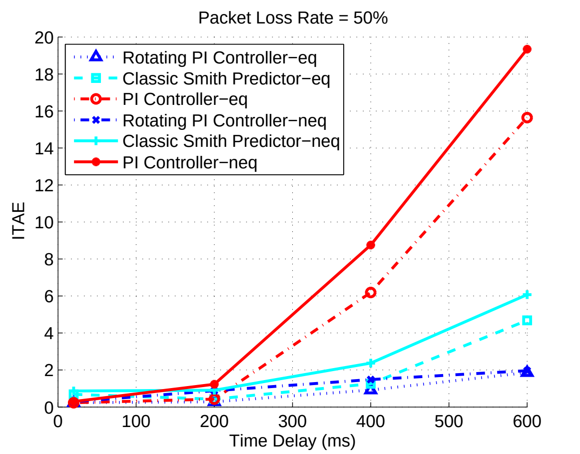


**Fig. 14:** ITAE for three controllers (Packet loss rate is 50 %).

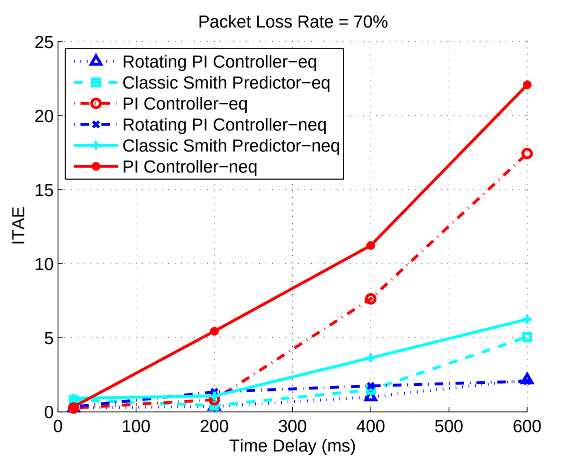


**Fig. 15:** ITAE for three controllers (Packet loss rate is 70 %).

The smaller the ITAE values, the better is the controller performance. Figures 13-15 show the ITAE values of the controllers versus network time delay when the packet loss rates are 20%, 50%, and 70%, respectively. The points over horizontal coordinates in these figures are time-varying delays which are random processes with mean of those points and uniform distribution over the interval -10 ms and 10 ms. The ITAE values of the controllers have been calculated over the first pulse of input (2.9 seconds). Based on the results from Fig. 13-15, when the probabilities of packet loss patterns are equal, in all cases the ITAE value of rotating PI controller is less than both ITAE values of classical Smith predictor and PI controller. When network time delay increases, the differences between ITAE values of rotating PI controller and two other controllers become greater. When the probabilities of packet loss patterns are not equal, the results show that the performance of rotating PI controller is better than two other controllers especially when the time delay and packet loss rate increase. Generally, when the time delay is about 600 ms and the probabilities of packet loss patterns are equal, the ITAE values of classical Smith predictor and PI controller are 2.5622 and 8.5972 times of ITAE value of rotating PI controller, respectively, and when the probabilities of packet loss patterns are not equal, the ITAE values of classical Smith predictor and PI controller are 2.9615 and 9.8395 times of ITAE value of rotating PI controller, respectively. Therefore, even when the probabilities of packet loss patterns are not equal, the results of rotating PI controller are much better than two other controllers.

## 5 Conclusion and Future Work

A new method for controlling the plant in networked control systems was proposed. The proposed method tackles the problems of time delay and packet loss, which destabilize the system, by rotating the phase plane in PI controller. Packet loss is modeled by Markov chain. Markov chain model makes it possible to investigate some situations that are more effective on networked control systems such as consecutive packet loss. The new proposed method has the advantage of considering the effects of time delay and packet loss simultaneously. This controller used the online estimation of time delay and was tuned for such time delay online by rotating the phase plane. The results show the improvement of performance in this novel method compared to other methods. In future work the values in dropped packets will be estimated and these values will be used for improvement of system performance.

## Acknowledgement

This work was partially supported by the ICT Research Institute (ITRC).

The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

## References

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

[2] P. Ogren, E. Fiorelli, and N. E. Leonard, Cooperative control of mobile sensor networks: Adaptive gradient climbing in a distributed environment, IEEE Transactions on Automatic Control, **49**, 1292-1302 (2004).

[3] C. Meng, T. Wang, W. Chou, S. Luan, Y. Zhang, and Z. Tian, Remote surgery case: Robot-assisted teleneurosurgery, in Proceedings of the IEEE International Conference on Robotics and Automation (ICRA'04), 819-823 (2004).

[4] S. Shirmohammadi and N. H. Woo, Evaluating decorators for haptic collaboration over Internet, in Proceedings of the 3rd IEEE International Workshop on Haptic, Audio and Visual Environments and Their Applications, 105-109 (2004).

[5] P. Seiler and R. Sengupta, Analysis of communication losses in vehicle control problems, in Proceedings of the American Control Conference, **2**, 1491-1496 (2001).

[6] R. M. Murray, K. J. Astrom, S. P. Boyd, R. W. Brockett, and G. Stein, Future directions in control in an information-rich world, IEEE Control Systems Magazine, **23**, 20-33 (2003).

[7] L. Zhang, H. Gao, and O. Kaynak, Network-induced constraints in networked control systems-A survey, IEEE Transactions on Industrial Informatics, **9**, 403-416 (2013).

[8] C. L. Lai and P. L. Hsu, Design the remote control system with the time-delay estimator and the adaptive Smith predictor, IEEE Transactions on Industrial Informatics, **6**, 73-80 (2010).

[9] W. Du, F. Du, C. Shen, Y. Zhou, Y. Du, S. Feng, and B. Chen, New Smith predictor and fuzzy adaptive control for wireless networked control systems, Software Engineering Research, Management and Applications, **253**, 239-249 (2009).

[10] P. Hajebi and S.M.T. AlModarresi, Online adaptive fuzzy logic controller using neural network for networked control systems, in Proceedings of the 14th International Conference on Advanced Communication Technology (ICACT), 888-893 (2012).

[11] P. Hajebi and S.M.T. AlModarresi, Online adaptive fuzzy logic controller using genetic algorithm and neural network for networked control systems, ICACT Transactions on Advanced Communications Technology (TACT), **1**, 88-98 (2012).

[12] P. Wen and Y. Li, Minimum packet drop sequences based networked control system model with embedded Markov chain, Simulation Modelling Practice and Theory, **17**, 1635-1641 (2009).

[13] I. Pan, S. Das, and A. Gupta, Tuning of an optimal fuzzy PID controller with stochastic algorithms for networked control systems with random time delay, ISA Transactions, **50**, 28-36 (2011).

[14] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. I. Jordan, and S. S. Sastry, Kalman filtering with intermittent observations, IEEE Transactions on Automatic Control, **49** 1453-1464 (2004).

[15] F. Yang and Q.-L. Han, $H_\infty$ control for networked systems with multiple packet dropouts, Information Sciences, **252**, 106-117 (2013).
[16] O. C. Imer, S. Yuksel, and T. Basar, Optimal control of LTI systems over unreliable communication links, Automatica, **42**, 1429-1439 (2006).
[17] S. Yuksel and T. Basar, State estimation and control for linear system over communication networks, in Proceedings of the IEEE Conference on Control Applications, **1**, 409-414 (2003).

**Pooya Hajebi** was born in Isfahan, Iran, in 1981. He received the B.S. degree in electrical engineering (Electronics) and the M.S. degree in electrical engineering (Communication Systems) both from Yazd University, Yazd, Iran, in 2005 and 2009, respectively. He is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering, Yazd University, Yazd, Iran. He is a student member of IEEE. His research interests include Networked Control Systems, Fuzzy Systems, Neural Networks, Data Communication Networks, Digital Signal Processing, Biological Signal Processing, Digital Image Processing, Cellular Networks, Optimization Algorithms, Time Delay Systems, and Real-Time Systems. He is referee of International Journal of Systems Science (Taylor & Francis) and several conferences.

**Seyed Mohammad Taghi AlModarresi** obtained his B.S. degree in Electronics Engineering and M.S. degree in Communication Systems, both from the Isfahan University of Technology, Isfahan, Iran. He also holds a Ph.D. in Electronics (Intelligent Signal Processing) from University of Southampton, UK (Department of Electrical and Computer Science: ECS). He works at the Department of Electrical and Computer Engineering in Yazd University where he pursues his research interests in: (i) Networked Control Systems (NCS); (ii) Neuro-Fuzzy Networks; and (iii) Wireless Networks.