

Research Article

Neural PID Control Strategy for Networked Process Control

Jianhua Zhang¹ and Junghui Chen²

¹ State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources, North China Electric Power University, Beijing 102206, China

² Department of Chemical Engineering, Chung-Yuan Christian University, Chung-Li 320, Taiwan

Correspondence should be addressed to Junghui Chen; jason@wavenet.cycu.edu.tw

Received 3 January 2013; Accepted 4 March 2013

Academic Editor: Engang Tian

Copyright © 2013 J. Zhang and J. Chen. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new method with a two-layer hierarchy is presented based on a neural proportional-integral-derivative (PID) iterative learning method over the communication network for the closed-loop automatic tuning of a PID controller. It can enhance the performance of the well-known simple PID feedback control loop in the local field when real networked process control applied to systems with uncertain factors, such as external disturbance or randomly delayed measurements. The proposed PID iterative learning method is implemented by backpropagation neural networks whose weights are updated via minimizing tracking error entropy of closed-loop systems. The convergence in the mean square sense is analysed for closed-loop networked control systems. To demonstrate the potential applications of the proposed strategies, a pressure-tank experiment is provided to show the usefulness and effectiveness of the proposed design method in network process control systems.

1. Introduction

Networked control systems (NCSs) make it convenient to control large distributed systems. Process control can integrate the controlled process and the communication network of computational devices, but sensors and actuators cannot be directly used in a conventional way because there are some inherent issues in NCS, such as delay, packet loss, quantization, and synchronization. Recently, some efforts have been made to deal with these issues about NCSs. Zhang et al. (2012) investigated the stability problem of a class of delayed neural networks with stabilizing or destabilizing time-varying impulses [1]. The controlling regions in neural networks were identified in three scales using single-objective evolutionary computation methods [2]. Tian et al. (2008) investigated the observer-based output feedback control algorithm for networked control systems with two quantizers using a set of nonlinear matrix inequalities [3]. In robust design, a robust and reliable H_∞ filter was designed for a class of nonlinear networked control systems with random sensor failure [4].

In addition, some investigations have been made to incorporate the network into process control systems in the

literature. In the work of [5], their model predictive control algorithm was extended for processes with random delays using a communication delay model, along with time-stamping and buffering technique. Two novel networked model predictive control schemes based on neighborhood optimization were presented for online optimization and control of a class of serially connected processes [6]. A two-tier control architecture was presented [7]. A lower-tier control system relying on point-to-point communication and continuous measurements was first designed to stabilize the closed-loop system. Also, an upper-tier networked control system was subsequently designed using the Lyapunov-based model predictive control theory to profit from both the continuous and the asynchronous, delayed measurements as well as from additional networked control actuators to improve the closed-loop system performance. A design methodology for fault-tolerant control systems of chemical plants with distributed interconnected processing units was presented [8]. This approach incorporated Lyapunov-based nonlinear control with the hybrid system theory. It was developed based on a hierarchical architecture that integrated lower-level feedback control of the individual units with upper-level logic-based supervisory control over communication

networks. An output feedback controller was proposed. It combined a Lyapunov-based controller with a high-gain observer for nonlinear systems subject to sensor data losses [9].

Since NCS operates over a network, data transfer between the controller and the remote system, in addition to the controlled processing delay, will inevitably introduce network delays, the controller-to-actuator delay and the sensor-to-controller delay. Random access networks, such as CAN and Ethernet, have random network delays, which may cause deterioration of the system performance [10]. The packet loss caused by communication networks also affects the control system performance; that is, process control systems under the communication environment are stochastic in nature because of the random time delays or packet loss caused by the communications on the employed networks.

The output variable of a network process is usually subjected to the network delay of the uncertain duration and/or stochastic disturbances. It can be treated as a random variable that follows a specific probability density function (PDF). Consequently, the output tracking error is also a random variable. PDF of the output tracking error can be determined if the information about the process model and PDF of both disturbances and network delays is available. The dynamic stochastic distribution control theory has been proposed [11]. It is generally true for only linear systems with Gaussian random inputs or at least symmetric PDF that the minimum variance of the output tracking error indicates optimization of controller tuning. However, for non-Gaussian stochastic systems, the control method that is focused only on mean and variance of the output tracking error is not sufficient to capture the probabilistic behavior of this stochastic system. A more general measure of uncertainty should be used to characterize the uncertainty of the output tracking error in the control design.

If the network is connected with the regulatory control level and the advanced control level is located at the remote side to cooperate with other plants, the whole control system will have a two-level hierarchy. The control structure of the regulatory control level in the plant site is unchanged. Thus, the simple and robust proportional-integral-derivative (PID) controller can be applied to the operating plant. The conventional PID controller can get the support from the networked process control systems when the operating plants encounter a new operation condition or uncertain factors. On the other hand, recently neural networks attracted a very large research interest. They have great capability of solving complex mathematical problems and they have received considerable attention in the field of chemical process control and have been applied to system identification and controller design [12]. They are effectively used in the control region for modeling nonlinear processes, especially in model-based control, such as direct and indirect neural network model based control [13], nonlinear internal model control [14], and recurrent neural network model control [15]. Although the control performances of the above methods are satisfactory, the control design still focuses on point-to-point wired communication links between the designed controllers and the controlled processes. It makes the implementation

strategy realistic only for in-site control systems. The optimal fuzzy PID controller for NCSs by minimizing the sum of the integral of time multiplied by absolute errors and the squared controller output was proposed [16]. Neural controllers for NCSs were also applied based on minimum tracking error entropy for the tuning PID controller [17], but it was not easy to implement them on practical applications because they needed a set of tracking errors at the instant time point. Ghoshtine et al. (2011) used Monte Carlo simulation to demonstrate the effectiveness of the traditional PID controller, but they did not tell how to adjust PID controller parameters [18]. Indeed, for networked process control systems, it is the random time delays and packet loss that lead to stochastic tracking errors for closed-loop systems besides random disturbances. As such, it is natural to represent networked process control systems as a stochastic control framework.

The main objective of this work is to propose a two-level hierarchical control system based on the PID-like neural network adaptive scheme for networked process control systems. The tracking error entropy of closed-loop control systems is utilized to construct the performance index so as to update the weights of the neural PID controller. The rest of this paper is organized as follows. Section 2 investigates the stochastic characteristics in networked process control systems and formulates the control problem and objective. The reason for selecting the minimum entropy of the closed-loop tracking error as a performance index is then fully described. Section 3 presents the neural PID control law for process control systems with network-induced time delays and packet dropout, and, then, the convergent condition in the mean square sense of the proposed controller is presented and proved. Section 4 shows the application of the proposed controller to a networked chemical process control system, and Section 5 concludes this paper.

2. Stochastic Characteristics of NCS in Operation Processes

A networked process control system with a two-level hierarchical control architecture shown in Figure 1, in which a process plant is controlled by a PID controller at the plant site and the controller can also be self-tuned by a neural network at the remote site, is proposed to improve the performance of the closed-loop system. In this system, the output signal $y(k)$ of the plant is synchronously measured with ideal sampler S at a sampling rate $1/T$. The digital controller uses the information of the plant transmitted through the networks to update the new set of the control parameters. The parameter values transmitted through networks and held by a zero-order holder H are then transmitted to drive the plant. $\bar{y}(k) \in R^1$ is the input to the controller. $r(k)$ is the setpoint. NCS operates over a network; data transfers between the controller and the remote system will inevitably introduce network delays; that is, the output quality $y(k)$ from the current time point k is often unavailable before the measurement is sent to the controller at the current time point due to network delays. Likewise, the control input $u(k)$ from the current time point k is also unavailable after the control action is calculated

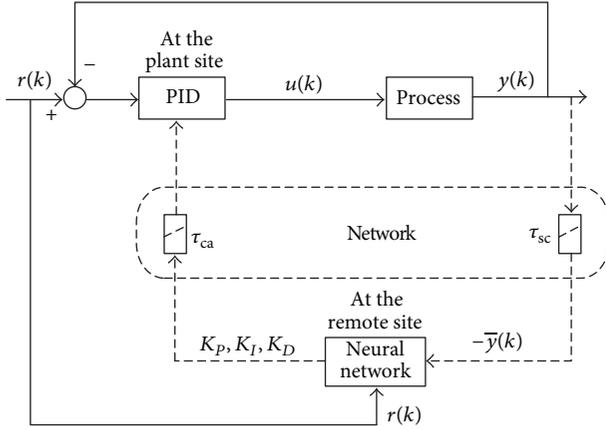


FIGURE 1: Networked process control system.

at the current time point. τ_{ca} denotes the induced delay between the controller and the actuator and τ_{sc} stands for the induced delay between the sensor and the controller.

The controlled process is nonlinear no matter whether it is a continuous or batch process. The PID controller control variable $u(k)$ at the plant site is

$$u = u_s + \left(K_p e + K_I \int edt + K_D \frac{de}{dt} \right), \quad (1)$$

where u_s is the bias value. $e(k) = r(k) - y(k)$ is the output ($y(k)$) error deviated from the setpoint. K_p , K_I , and K_D are known as the proportional gain, the integral time constant, and derivative time constant, respectively. In Figure 1, the control structure is similar to an adaptive control structure, but the parameters of the PID controller are adjusted via the backpropagation network control system. The control parameters K_p , K_I , and K_D transmitted through networks and held by a zero-order holder H are

$$\begin{aligned} K_p(t) &= K_p(kT) \\ K_I(t) &= K_I(kT) \\ K_D(t) &= K_D(kT) \end{aligned} \quad \text{for } kT \leq t < kT + T \quad (2)$$

And they are then transmitted to the controller to drive the plant.

In networked control systems, a continuous signal is sampled, encoded in digital format, transmitted over the network, and finally decoded at the receiver's side. The overall delay between sampling and eventual decoding at the receiver can be highly variable because both the network access delays and the transmission delays depend on highly variable network conditions, such as congestion and channel quality [19]. In a networked control system, message transmission delay can be divided into two parts: device delay and network delay. The device delay includes the time delay at the source and the destination nodes. The time delay at the source node includes the preprocessing time, T_{pre} , and the waiting time, T_{wait} . The time delay at the destination node is only the postprocessing time, T_{post} . The network time delay includes the total transmission time of a message, T_{tx} , and the

propagation delay of the network (T_p). The total time delay can be expressed as [20]

$$T_{delay} = T_{pre} + T_{wait} + T_{post} + T_{tx} + T_p. \quad (3)$$

The waiting time T_{wait} may be significant due to the amount of data sent by the source node, the transmission protocol, and the traffic on the network. T_{wait} is random in most cases. The postprocessing time T_{post} is negligible in a networked control system compared with other time delays. The above delays in networked control systems can be obtained by time-stamping techniques although they are normally not available [5].

In the networking area, random network delays have been modeled by using various formulations based on probability and the characteristics of sources and destinations [21]. In most cases, the network-induced time delays are random and are not Gaussian. Packet dropouts result from transmission errors in physical network links, which is far more common in wireless than in wired networks, or from buffer overflows due to congestion. Long transmission delays sometimes result in packet reordering, which essentially amounts to a packet dropout if the receiver discards "outdated" arrivals [22]. The packet loss is another random factor in networked control systems. This means that in general the system output will be very unlikely to be a Gaussian noise, leading to a non-Gaussian closed-loop tracking error when the control input is applied to the system. Thus, the randomness measure of the tracking error via the use of variance would not be sufficient in characterizing the performance of the closed-loop system. Therefore, the problems considered in this paper can be formulated to design a proper controller such that the shape of the tracking error of NCSs is made as narrow as possible.

3. Design of an NCS Controller

The output variable of the controlled processes over the communication system is subjected to the network delay of the uncertain duration and/or stochastic disturbances. It can be treated as a random variable that follows a specific PDF. To adjust the shape of PDF, the entropy is used here. Entropy has a more general meaning to arbitrary random variables than mean or variance. When entropy is minimized, all the moments of the error PDF (not only the second moment) are constrained [22]. Consequently, it can be used to measure and form a design criterion for general dynamic stochastic systems subject to arbitrary random inputs whose PDF can be of any shape. The application of entropy in control and estimation is developed in this section.

3.1. Estimation of Error PDF and Its Entropy. In practical applications, PDF of the tracking error in NCSs is often unknown a priori. A data-driven method that can get the error PDF $\hat{\gamma}_e(z)$ from the error samples e_k is employed, where z represents the current error, which is a random variable. The error data are obtained by subtracting the process output as actually received by the controller from the setpoint in a given time period. Using the Parzen method [22], the error PDF $\hat{\gamma}_e(z)$ is approximated as the weighted

sum of the K Gaussian kernel function defined as $\kappa(z, \sigma^2) = (1/\sqrt{2\pi\sigma}) \exp(-(z - e(k))^2/2\sigma^2)$ with the mean located at the error data point $e(k)$ as

$$\hat{\gamma}_e(z) = \frac{1}{K} \sum_{k=1}^K \kappa(z - e(k), \sigma^2), \quad (4)$$

where the multidimensional Gaussian function with a radially symmetric variance σ^2 is used here for simplicity. For minimizing the range of the error and maximizing the concentration of the error about mean of the error PDF at the same time, the entropy of the error should be minimized. The entropy (H) of the error PDF can be calculated using Shannon entropy

$$H_S(e) = - \int_{-\infty}^{\infty} \gamma_e(z) \log \gamma_e(z) dz. \quad (5)$$

When Shannon entropy definition is used along with this PDF estimation, the calculation of Shannon entropy becomes very complex, whereas the calculation of Rényi quadratic entropy leads to a simpler form. Rényi's entropy with order S of the tracking error is given by

$$H(e) = \frac{1}{1-S} \log \int_{-\infty}^{\infty} \gamma_e^S(e) de. \quad (6)$$

3.2. General Minimum Entropy Based Tuning Neural PID Network. Unlike the conventional deterministic controller design seeking a control input that minimizes the difference between the process output and the desired output at the next time step, the goal of the proposed controller design is to determine control parameters that minimize the shape of error PDF, for the output over NCS is stochastic. The objective is expressed as

$$\min_{K_p, K_I, K_D} H = - \min_{K_p, K_I, K_D} \log \int \gamma_e^2(z) dz = - \min_{K_p, K_I, K_D} \log V, \quad (7)$$

where the order of Rényi's entropy is set to be $S = 2$ owing to the computational efficiency. The above objective function is calculated for the present instant time point k only. V_k is information potential (IP) of quadratic Rényi's entropy at the time point k . It can be calculated in closed form from the samples using Gaussian kernels as

$$\begin{aligned} V_k &= \int \gamma_e^2(z) dz \\ &\cong \int \left(\frac{1}{N} \sum_{i=1}^N \kappa(z - e_i, \sigma^2) \right)^2 dz \\ &\cong \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N \kappa(e_i - e_j, 2\sigma^2). \end{aligned} \quad (8)$$

Although the PID controller is widely utilized because of the simple structure and the intuitive physical meanings, it is difficult to tune proper PID parameters in NCSs owing

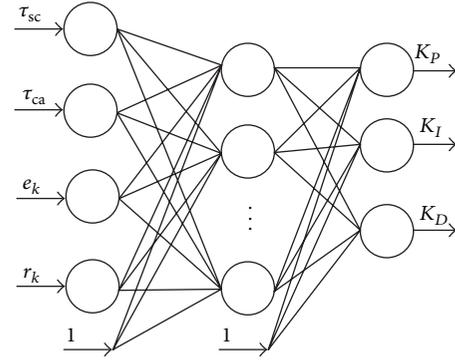


FIGURE 2: Structure of a neural network in NCS.

to induced random delays. Consequently, in this research, a self-tuning neural PID control scheme is proposed in Figure 2, where a neural network is used to tune the parameters of the conventional PID controller similar to the adjustment made by an experienced operator. Like the human operator who has accumulated past experience and knowledge on the controlled system and knows how to do the adjustment, the neural network can store such information and digest it for the final adjustment because neural networks have the function approximation abilities, learning abilities, and versatility that can adapt to unknown environments. Thus, the operator's experience and knowledge can be included into a neural network and the network is trained based on the past data history. Finally, the trained neural network can be used as means to tune the PID controller parameters online.

Using a three-layered neural network shown in Figure 2, the learning rule that finds the suitable PID parameters is realized. Thus, in the present neural network values, the setpoint $r(k)$, the current error $e(k)$, the induced delay between the controller and the actuator τ_{ca} , and the induced delay between the sensor and the controller τ_{sc} are denoted by $O_1^{(1)}$, $O_2^{(1)}$, $O_3^{(1)}$, and $O_4^{(1)}$. The outputs at the output layer are K_p , K_I , and K_D , all of which are denoted by $O_1^{(3)}$, $O_2^{(3)}$, and $O_3^{(3)}$. Then the outputs of a single hidden layer neural network can be computed via the following steps.

The input layer is

$$\begin{aligned} O_0^{(1)} &= 1, & O_1^{(1)} &= r_k, & O_2^{(1)} &= e_k, \\ O_3^{(1)} &= \tau_{ca}, & O_4^{(1)} &= \tau_{sc}. \end{aligned} \quad (9)$$

The input and the output of the hidden layer are

$$\begin{aligned} \text{net}_p^{(2)}(k) &= \sum_{q=0}^4 w_{pq}^{(2)} O_q^{(1)}(k), & p &= 1, 2, \dots, P \\ O_p^{(2)}(k) &= f(\text{net}_p^{(2)}(k)), & p &= 1, 2, \dots, P \\ O_0^{(2)}(k) &= 1, \end{aligned} \quad (10)$$

where P is the number of hidden neurons. The input and the output of the output layer are

$$\begin{aligned} \text{net}_l^{(3)}(k) &= \sum_{p=0}^P w_{lp}^{(3)} O_p^{(2)}(k), \quad l = 1, 2, 3 \\ O_l^{(3)}(k) &= g(\text{net}_l^{(3)}(k)), \quad l = 1, 2, 3 \\ K_P &= O_1^{(3)}(k), \quad K_I = O_2^{(3)}(k), \quad K_D = O_3^{(3)}(k), \end{aligned} \quad (11)$$

where w_{pq} and w_{lp} are weights trained in the neural networks. The initial values of the weight are selected from the uniform distribution across a small range $[-1 \ 1]$. $f(x) = (e^x - e^{-x})/(e^x + e^{-x})$ and $g(x) = e^x/(e^x + e^{-x})$, known as the activation functions, perform the weighted sum of the incoming signals from the input layer and the hidden layer, respectively. With the input layer and the hidden layer, the bias is introduced into the first node.

In order to derive the self-tuning algorithm of the PID controller, the cost function defined in (7) should be minimized. Based on the steepest descent approach, at the output layer, we have

$$\begin{aligned} \Delta w_{lp}(k) &= w_{lp}(k) - w_{lp}(k-1) \\ &= -\eta_1 \frac{\partial H(e(k))}{\partial w_{lp}} \iff \eta \frac{\partial V_k}{\partial w_{lp}}, \end{aligned} \quad (12)$$

Consider the following: where η_1 or η is the learning factor.

$$\begin{aligned} \frac{\partial [V(e(k))]}{\partial w_{lp}} &= \frac{1}{2K^2\sigma^2} \sum_{k_1=1}^K \sum_{k_2=1}^K (e(k_2) - e(k_1)) \kappa(e(k_1) - e(k_2), 2\sigma^2) \\ &\quad \times \left[\frac{\partial e(k_1)}{\partial w_{lp}} - \frac{\partial e(k_2)}{\partial w_{lp}} \right] \end{aligned} \quad (13)$$

$$\begin{aligned} \frac{\partial e(k)}{\partial w_{lp}} &= -\frac{\partial y(k)}{\partial w_{lp}} = -\frac{\partial y(k)}{\partial u(k)} \frac{\partial u(k)}{\partial O_l^{(3)}} \frac{\partial O_l^{(3)}}{\partial \text{net}_l^{(3)}} \frac{\partial \text{net}_l^{(3)}}{\partial w_{lp}} \\ &\quad k = 1, 2, \dots, K, \end{aligned} \quad (14)$$

where IP shown above is computed using the measurement errors of the past samples. The Jacobian information

$\partial y(k)/\partial u(k)$ can be replaced by $\text{sgn}(\partial y(k)/\partial u(k))$ or it can be calculated by the model prediction algorithm of the plant under the assumption of $u(k-i) \approx u(k-i+1)$ when the proper sampling of the control system is applied as

$$\begin{aligned} \frac{\partial y(k)}{\partial u(k)} &= \frac{\partial y(k)}{\partial u(k - \tau_{ca}^k)} \frac{\partial u(k - \tau_{ca}^k)}{\partial u(k)} = \frac{\partial y(k)}{\partial u(k - \tau_{ca}^k)} \frac{\partial u(k-1)}{\partial u(k)} \\ &\quad \times \frac{\partial u(k-2)}{\partial u(k-1)} \dots \frac{\partial u(k - \tau_{ca}^k)}{\partial u(k - \tau_{ca}^k + 1)} \approx \frac{\partial y(k)}{\partial u(k - \tau_{ca}^k)}. \end{aligned} \quad (15)$$

Also,

$$\begin{aligned} \frac{\partial u(k)}{\partial O_l^{(3)}} &= \begin{cases} e(k) - e(k-1) & l = 1 \\ e(k) & l = 2 \\ e(k) - 2e(k-1) + e(k-2) & l = 3 \end{cases} \end{aligned} \quad (16)$$

$$\begin{aligned} \frac{\partial O_l^{(3)}}{\partial \text{net}_l^{(3)}} &= g'(\text{net}_l^{(3)}(k)) = O_l^{(3)}(1 - O_l^{(3)}) \\ \frac{\partial \text{net}_l^{(3)}}{\partial w_{lp}} &= O_p^{(2)}. \end{aligned}$$

At the hidden layer,

$$\Delta w_{pq}(k) = w_{pq}(k) - w_{pq}(k-1) = \eta \frac{\partial V_k}{\partial w_{pq}} \quad (17)$$

$$\begin{aligned} \frac{\partial [V_k]}{\partial w_{pq}} &= \frac{1}{2K^2\sigma^2} \sum_{k_1=1}^K \sum_{k_2=1}^K (e(k_2) - e(k_1)) \kappa(e(k_1) - e(k_2), 2\sigma^2) \\ &\quad \times \left[\frac{\partial e(k_1)}{\partial w_{pq}} - \frac{\partial e(k_2)}{\partial w_{pq}} \right]. \end{aligned} \quad (18)$$

Using the chain rule, $\partial e(k_1)/\partial w_{pq}$ and $\partial e(k_2)/\partial w_{pq}$ can be expressed as

$$\begin{aligned} \frac{\partial e(k)}{\partial w_{pq}(k)} &= -\frac{\partial y(k)}{\partial w_{pq}(k)} = -\frac{\partial y(k)}{\partial u(k)} \frac{\partial u(k)}{\partial w_{pq}(k)} \end{aligned}$$

$$\begin{aligned}
&= -\frac{\partial y(k)}{\partial u(k)} \sum_l \frac{\partial u(k)}{\partial O_l^{(3)}(k)} \frac{\partial O_l^{(3)}(k)}{\partial \text{net}_l^{(3)}(k)} \frac{\partial \text{net}_l^{(3)}(k)}{\partial O_p^{(2)}(k)} \\
&\quad \times \frac{\partial O_p^{(2)}(k)}{\partial \text{net}_p^{(2)}(k)} \frac{\partial \text{net}_p^{(2)}(k)}{\partial w_{pq}(k)} \\
&\quad k = 1, 2, \dots, K,
\end{aligned} \tag{19}$$

where

$$\begin{aligned}
\frac{\partial \text{net}_l^{(3)}(k)}{\partial O_p^{(2)}(k)} &= w_{lp}^{(3)} \\
\frac{\partial O_p^{(2)}(k)}{\partial \text{net}_p^{(2)}(k)} &= f'(\text{net}_p^{(2)}(k)) = O_p^{(2)}(1 - O_p^{(2)}) \\
\frac{\partial \text{net}_p^{(2)}(k)}{\partial w_{pq}(k)} &= O_q^{(1)}(k).
\end{aligned} \tag{20}$$

Although IP (8) calculation can be done through the Monte Carlo method, it is preferred to find a more practical solution using sliding windowing. Supposing that the estimate $V_k(e)$ for the tracking error has been obtained at the k th run, the recursive estimation algorithm for $V_k(e)$ of the tracking error can be obtained by using the new sample $e(k+1)$ as follows:

$$\begin{aligned}
V_{k+1}(e) &\cong (1 - \lambda)V_k(e) \\
&\quad + \frac{\lambda}{L} \sum_{i=k-L+1}^k \kappa(e(i) - e(k+1), 2\sigma^2),
\end{aligned} \tag{21}$$

where $0 \leq \lambda < 1$ is the forgetting factor. L is the window length. The size of window L should be selected to contain the dynamic characteristics of the plant and network induced delays. Note that even if the sliding windowing is used to produce the information potential, the objective function is still optimized for the present instant k only.

3.3. Online Updating PID Controller Based on SPC Monitoring Chart. The conventional control parameter adjustment is triggered on the noise-free or noise-filtered data. However, it has the disadvantage of postponing the parameter adjustment until after an influent transient has subsided. Noise, on the other hand, can induce unjustified control parameter changes, which would, in turn, induce incorrect control action. Process variations in our problem may indeed be a classical electronic type of instrumentation noise and/or packet loss caused by the communications on the employed networks, but they can also be successive, small but real transient in the process output commonly because of external disturbance changes. Thus, in the past, process variations were classified into the common cause and assignable cause variations in manufacturing and service industries. A common cause variation is inherent in the process and is impossible or hard to be eliminated. It is assumed that the sample comes from a known probability distribution.

Based on Shewhart's classification, Deming (1982) argued the special cause of variations as "something special, not part of the system of common causes" [23] should be identified and removed from the root. If an error value is calculated from noisy measurements, the variable error will be a random distribution variable.

Even if controlled processes are well designed and carefully maintained, a certain amount of inherent natural variability is unavoidable. The mechanism that assesses the performance of the control loop for the current operation is the control chart. The chart summarizes the results obtained for the process conducted under the minimum variance. To detect if there is deviation from the minimum bounds in the current operations, a statistical hypothesis testing approach can be applied to the control output error at each time point. If the controlled process is at a steady state, one can conventionally calculate the error distribution function (4). Based on the traditional statistical process control (SPC) method, the upper (U) and the lower (L) control limits for the error are

$$\int_{-\infty}^L \hat{\gamma}_e(z) dz = \frac{\alpha}{2}, \quad \int_U^{+\infty} \hat{\gamma}_e(z) dz = \frac{\alpha}{2}, \tag{22}$$

where α represents a probability of falsely rejecting the hypothesis when, in fact, it should be accepted. The selected area α under the normal operation is quite small. In this study, α is 0.05. L and U limits of the integral indicate that the desired value of the probability density function will be between L and U with the integral value ($\int_L^U \hat{\gamma}_e(z) dz$) is equal to $1 - \alpha$ (22). Thus, whenever the process is around the setpoint, it is highly unlikely that the discrepancy between the observed and the setpoint values will be out of the lower and the upper limits. One aspect of the SPC philosophy is to accept the normal process variability without reporting any change in value unless the cause is identified [24, 25]. Equation (22) is numerically tractable using the probability model to calculate the thresholds (L and U) and detect any departure of the process from its standard behavior. Applying the above control limits to the measured error variable, one would retain the PID controller parameters in the local plant until there is statistically sufficient evidence that the error value is changed. This means that an SPC control chart is superimposed to monitor when the controlled system departs from the target so that the value of the NN-PID controller parameters can be regulated.

Therefore, the proposed two-level hierarchical control architecture can be summarized by the following steps.

Step 1. Calculate the tracking error in NCSs $\{e(k_1), e(k_2), \dots, e(k_N)\}$ and IP based on the sliding windowing using (21).

Step 2. Train neural networks offline based on the historical data and the conventional turning method, like ITAE (integral of the time-weighted absolute error). Of course, the historical data should cover the desired operation region. A good initial model can be achieved. Also, use (21) to construct L and U control limits. Then the weights of the neural networks are updated by (12) and (17).

Step 3. Check whether the controlled variable is “in statistical control” or not. If the process that is operating with only chance causes of variation present is “in statistical control,” keep the same controller parameters; otherwise, update K_P , K_I , and K_D obtained via neural networks.

Step 4. Formulate the control input $u(k)$ by (1) and apply the control input $u(k)$ to actuate the plant, set $k \leftarrow k + 1$, and repeat Step 1.

Note 1. Neural network adjusts the parameters of the controller in terms of the tracking error. The changes of the tracking error cannot be obtained soon when the operating point changes or the disturbance occurs. Nevertheless, the changes of the tracking error can be obtained eventually after a period of transmitting time due to the closed-loop feedback.

Note 2. The kernel size (the window width of the Parzen estimator) can be set experimentally after a preliminary analysis of the dynamic range of the tracking error. Some clustering analysis techniques can be applied to get the proper kernel size.

To ensure that the proposed algorithm can track the desired output of the controlled system, the convergence condition for the proposed algorithm should be satisfied and is derived as follows.

Theorem 1. *If the learning factor satisfies $0 < \eta < 1/\max_i |\lambda_i|$, the weights of the neural networks are convergent in mean square sense.*

The weights of the neural networks can be rearranged in a line and denoted as a vector W . Equations (12) and (17) can be summarized as follows:

$$W(k+1) = W(k) + \eta \nabla V(W(k)), \quad (23)$$

where $\nabla V(W(k))$ is the gradient of V_k given in (13) or (18) evaluated at $W(k)$. Perform the Taylor series expansion of the gradient $\nabla V(W(k))$ around the optimal weight vector W^* as

$$\nabla V(W(k)) = \nabla V(W^*) + \frac{\partial \nabla V(W^*)}{\partial W} (W - W^*). \quad (24)$$

Define a new weight vector space $\bar{W} = W - W^*$ and yield

$$\bar{W}(k+1) = [I + \eta R] \bar{W}(k), \quad (25)$$

where the Hessian matrix $R = \partial \nabla V(W^*) / \partial W = \partial^2 V(W^*) / \partial W^2$. Let $\Omega = \bar{Q}^T \bar{W}$, and \bar{Q} is the orthonormal matrix consisting of the eigenvalues of R . Thus,

$$\Omega(k+1) = [I + \eta \Lambda] \Omega(k), \quad (26)$$

where Λ is the diagonal eigenvalue matrix with entries ordered in correspondence with the ordering in \bar{Q} . It yields

$$\begin{aligned} \Omega_i(k+1) &= [1 + \eta \lambda_i] \Omega_i(k), \\ i &= 1, 2, \dots, P(Q+1) + 3(P+1). \end{aligned} \quad (27)$$

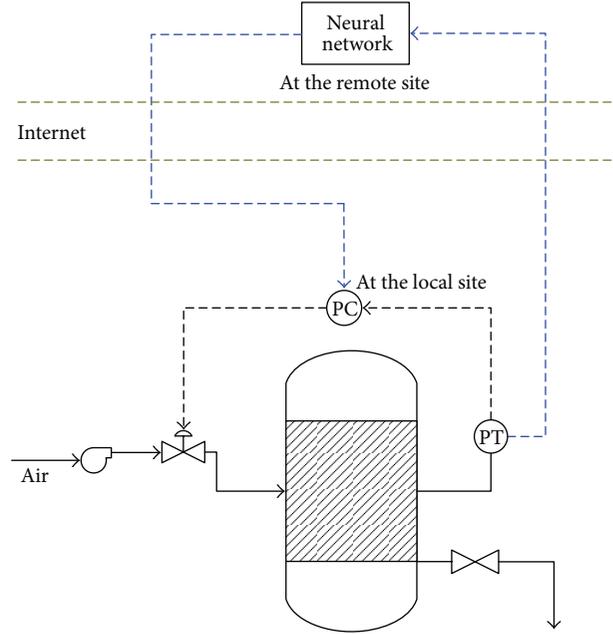


FIGURE 3: Schematic diagram of the air-pressure tank control loop over the network.

Define $\Delta \Omega_i(k) = \Omega_i(k+1) - \Omega_i(k)$. Since the eigenvalues of the Hessian matrix R are negative, $|1 + \eta \lambda_i| < 1$ can guarantee that the i th weight of the neural networks obtains a stable dynamics in mean square sense. Therefore, the following inequality can ensure that the proposed algorithm is convergent in mean square sense:

$$0 < \eta < \frac{1}{\max_i |\lambda_i|}. \quad (28)$$

This means

$$\lim_{k \rightarrow \infty} E \{ \Delta \Omega_i^2(k) \} = 0, \quad (29)$$

where $E(\cdot)$ is mathematical expectation.

4. Case Studies: Air-Pressure Tank

As for dealing with highly nonlinear and time varying chemical processes, although several schemes of self-tuning PID controllers were proposed in the past, their approaches were limited to the process disturbance with the Gaussian distribution assumption. If the data collected have non-Gaussian distribution, those methods would get weird responses and they would not be able to achieve the expected performance using minimizing variance. In this case study, the comparisons between the proposed network control method and the conventional PID tuning method should be made.

A laboratory scale air-pressure tank system shown in Figure 3 is made accessible for control over the computer network. At the local site, the tank is equipped with differential-pressure-to-current (DP/I) transducers to provide a continuous measurement of the pressure. The computer is connected to a PCI-1710 analog/digital I/O expansion card from

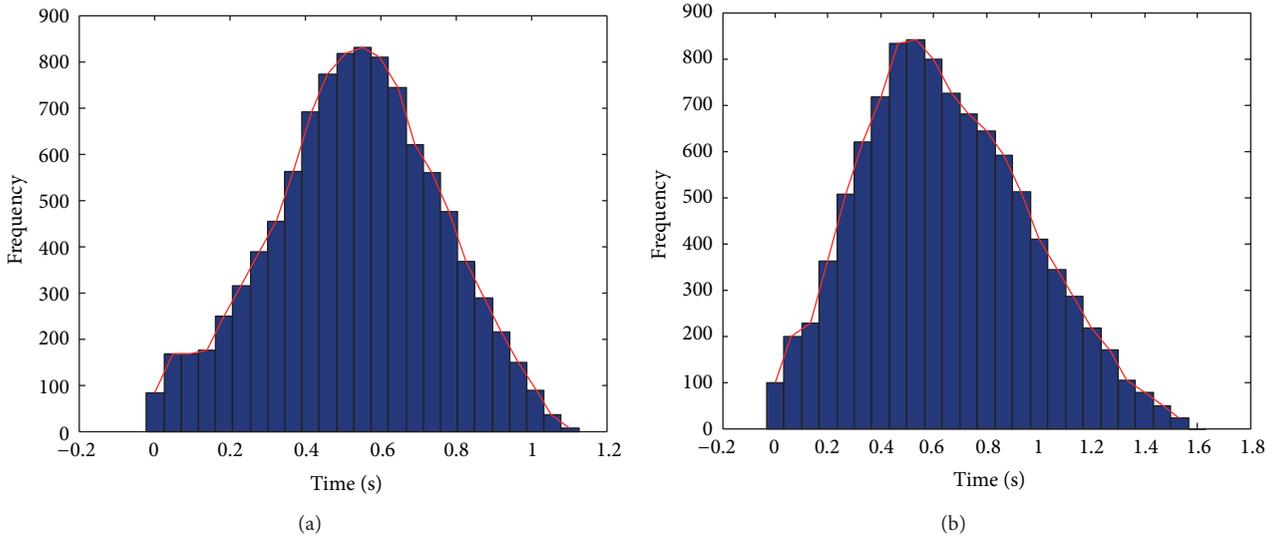


FIGURE 4: The distributions of (a) delay between the controller and the actuator and (b) delay between the sensor and the controller in network control of the air-pressure tank process.

Advantech. The expansion board uses a 12 bit converter; therefore, the digital signals are 12 bit. The analog signals from the measured pressure are amplified and conditioned EDM35 (4–20 mA/0–5 volts) modules. The embedded applications, including data acquisition software and the PID controller algorithm, are developed using MATLAB. At the remote site, the Windows operating system and TCP/IP over wireless networks that are implemented on laboratory floors to support mobility laptop are chosen. The Linux operation system and TCP/IP over Ethernet for the data storage server are implemented. The time constant of the air-pressure tank is significantly larger than the time needed during controller computation. Thus, the controller under the Microsoft Windows operating system is pretty close to the real-time environment. Furthermore, with the configuration mentioned above, it is easy to apply this test structure to the laboratory process and demonstrate queuing and buffering delays in the network system although an industrial system should be incorporated with a real-time OS using an appropriate network scheme. Moreover, in the discrete control system, if the controller can be actually acted at each fixed time point and the proper information is collected at the interval of two time points, the operating system can be treated as real-time control. The distributions of the delay between the controller and the actuator and those of the delay between the sensor and the controller are shown in Figures 4(a) and 4(b), respectively. The sensor-to-controller time delay can be calculated by the time difference between the time point when the controlled variable is measured and the time point when the controlled variable is stored on the data server. Likewise, the controller-to-actuator time delay is obtained by computing the time difference between the time point when the output neural network is sent out and the time point when the local controller receives the controller parameters. The objective is to assess the control

performance in the presence of communication and measurement delays when the proposed NCS method is applied. The variation of the process output with control inputs over time will be used to evaluate the performance of the proposed method.

In this application, the sampling period $T = 1$ sec. The setpoints of the pressure values have been changed from 3.5 (Kg/cm²) to 2.5 (Kg/cm²) and then from 2.5 (Kg/cm²) to 3.0 (Kg/cm²). The number of nodes in the hidden layer of the neural controller is 3, the learning factor $\eta = 0.01$, the kernel size used to estimate the entropy is experimentally set at $\sigma = 0.5$, and the samples for neural controller are trained with a segment of $L = 50$ at each instant. The procedures of designing the controller are implemented according to the summarized algorithm in Section 3. The proposed controller uses an instant performance index shown as (7), whereas the conventional PID controller turning method, ITAE, uses a cumulative performance index (integral means accumulating a series of data). To avoid improper initial weights of NN in our controller and produce large overshoot, NN is initially trained using the historical operation data. In this case, NN is trained before the time point 580 when the controller parameters are fixed without change. The performance using the proposed neural network PID controller is shown in Figure 5, where the 95% control limit is also included. Due to the fluctuation of the upstream pressure, the controlled pressure is still oscillatory even though the required time completes responses for the new set point change. With the new coming measurement, the PDF is regularly updated. The control limit is recalculated based on (22) when PDF is significantly changed. Thus, the control limits shown in Figure 5 are different at different time periods. In order to detect nonrandom changes in operating conditions, the superimposed rules on the chart are used to decide whether the pressure is “in statistical control” or not. The following

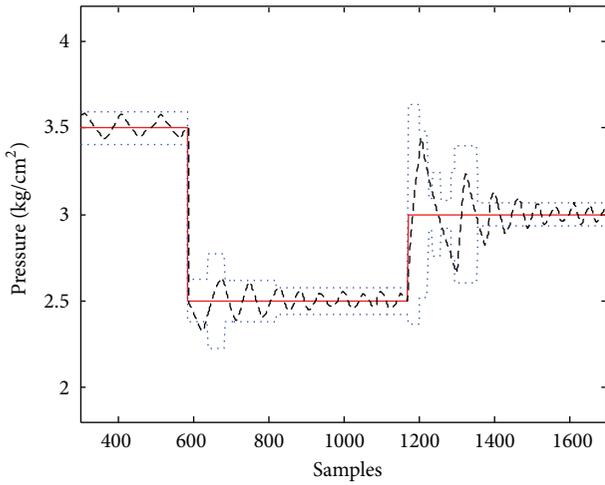


FIGURE 5: Network PID controller in setpoint change results in the air-pressure tank process: pressure (dashed line) and setpoint (solid line). The dotted line represents the 95% control limits.

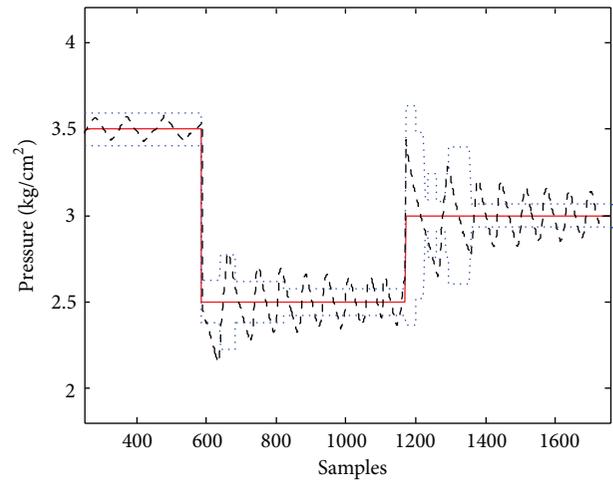


FIGURE 7: The PID controller with the fixed controller parameters in setpoint change results in the air-pressure tank process: pressure (dashed line) and setpoint (solid line). The dotted line represents the 95% control limits.

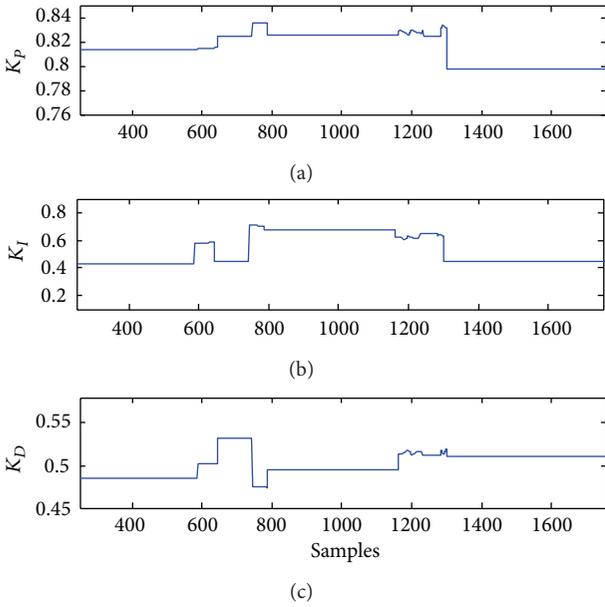


FIGURE 6: Network PID controller parameters in the air-pressure tank process: (a) K_p ; (b) K_i ; (c) K_d .

decision-making rules are adopted [26] when the control parameters are updated:

- eight consecutive points increasing in value;
- eight consecutive points decreasing in value;
- three consecutive points above the control limit;
- three consecutive points below the control limit;

The control parameters shown in Figure 6 are calculated with the new coming measurement, but the PID controller is not regularly updated until the pressure is significantly changed. Using the proposed rule to automatically adjust the control

parameters, the controlled results will not aggressively react to the random changes beforehand if the pressure is not “in statistical control.”

If the response to the controlled system with the same controller parameters is tested, the performance is shown in Figure 7. Note that the controller parameters are computed based on the process reaction curve [27]. First, based on the minimum ITAE tuning formula, the controller parameters are computed when the system is operated at the pressure around $3.5 \text{ (Kg/cm}^2\text{)}$. Figure 7 indicates that after the setpoint changes, the fixed tuning parameters result in a larger overshoot than the proposed neural network PID controller shown in Figure 5. The large oscillations in the process output are caused by the oscillatory control inputs. On the contrary, the neural network PID controller of the process output is less oscillatory and reaches the desired setpoint much more quickly. To clearly illustrate the improvement of the performance, Figure 8 shows the comparison of the error distributions between PID and neural network PID controllers at time point 1,500 where the set point has been changed for a while. The error distribution of the neural network PID controller is sharper and narrower after the controlled process reaches the desired set point. To clearly illustrate the improvement of the PDF distribution based on the proposed method, PDF at time points 1,250, 1,350, and 1,500 is replotted in Figure 9. It is shown that the PDF distribution obtained from the last time points is narrower and sharper than that from the early time points. As would be expected, the adaptive tuning parameters are needed for the nonlinear air-pressure tank process.

5. Conclusions

In recent years, there has been a growing interest in the study of the effects of NCSs mainly because of the broad use

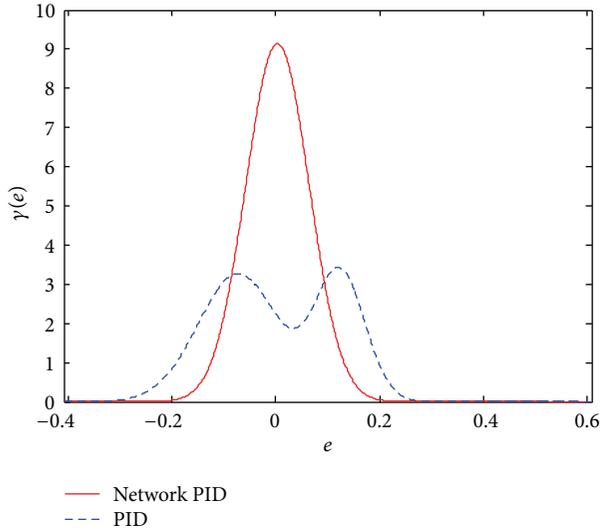


FIGURE 8: The estimated error distributions of the PID controller (dashed line) and the network PID controller (solid line) in the air-pressure tank process.

in NCSs. NCSs can provide Web clients a platform not only for remote monitoring of the current behavior of the operation plants but also for remote control of the plant. These advantages make NCSs applicable to many fields, including spacecraft automotive, remote robots, and manufacturing processes, but not much work has been done in the process control field so far. In a networked environment, the communication delays (also called network-induced time delays) and packet dropouts (probabilistic information missing, missing measurement) are two main problems that would degrade the system performance. In most relevant literature, the network-induced time delays have been commonly assumed to be deterministic, which is fairly unrealistic as, by nature, delays resulting from network transmissions are typically time varying and random.

In this paper, the proposed neural network PID controller for the controlled processes with the communication delays in various probabilistic ways is developed. In comparison with the past work, the merits of the proposed method are as following.

- (i) Instead of Monte Carlo that generates a set of tracking errors at the instant time point, the proposed method uses the sliding windowing technique to obtain the estimation of PDF, entropy, and information potential of the tracking error. Then the design of the controller is straightforward based on minimizing the entropy of the tracking error or maximizing the information potential of the tracking error. This is a more practical solution.
- (ii) Due to the stochastic behavior, the SPC monitoring chart is introduced to solve practical problems. It improves the control performance of the nonlinear process in the network. The implementation is much simpler and less computationally demanding.

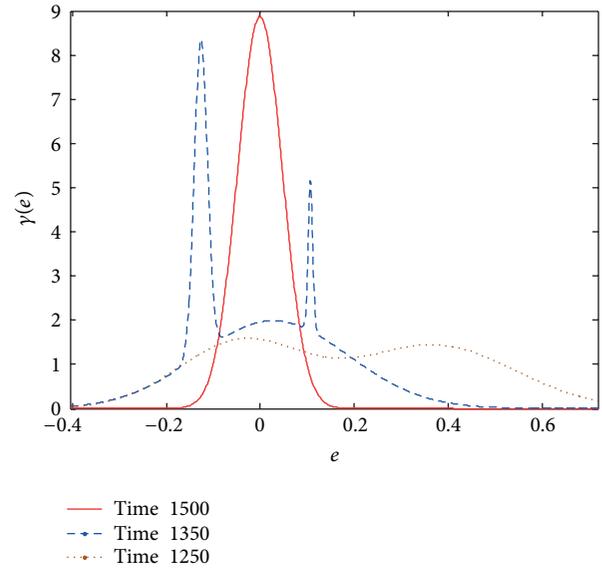


FIGURE 9: Plot of PDF for the time point at 1,250, 1,350, and 1,500 in the air-pressure tank process.

The control strategy is the balance between the network delays and the conventional nonlinear control designs.

- (iii) To verify the proposed practice-oriented algorithm, one practical case is conducted in an air-pressure tank. The results show that the proposed method is effective in following the setpoint and reducing the variance of the output caused by unknown stochastic network delays.

This strategy is particularly suitable for the industrial processes that is characterized by high nonlinearity and analytical difficulty in network process control. Because of communication delay, packet dropouts, noises, and disturbance, the tracking error may not obey the Gaussian distribution, the Rényi quadratic entropy of the tracking error is used to design the neural network PID controller for networked operation processes. Hence, the method proposed in this paper can be directly extended to deal with the controller design of the NCS with nonlinear plants, noises, and random delays. All the results presented in this paper yield encouraging outcomes in the single variable process, but sensitivity analysis on different initial PID parameters is not considered. The extension of the neural network PID to the sensitivity problem using theoretic analysis and MIMO systems is needed in the future.

Acknowledgments

This work was supported by the National Basic Research Program of China under Grant (973 Program 2011 CB710706) and the Doctoral Fund of the Ministry of Education of China (20110036110005). These are gratefully acknowledged.

References

- [1] W. Zhang, Y. Tang, J. Fang, and X. Wu, "Stability of delayed neural networks with time-varying impulses," *Neural Networks*, vol. 36, pp. 59–63, 2012.
- [2] Y. Tang, H. Gao, W. Zou, and J. Kurths, "Identifying controlling nodes in neuronal networks in different scales," *PLoS ONE*, vol. 7, no. 7, Article ID e41375, 2012.
- [3] E. Tian, D. Yue, and C. Peng, "Quantized output feedback control for networked control systems," *Information Sciences*, vol. 178, no. 12, pp. 2734–2749, 2008.
- [4] E. Tian and D. Yue, "Reliable H_∞ filter design for T-S fuzzy model-based networked control systems with random sensor failure," *International Journal of Robust and Nonlinear Control*, vol. 23, pp. 15–32, 2013.
- [5] D. Srinivasagupta, H. Schättler, and B. Joseph, "Time-stamped model predictive control: an algorithm for control of processes with random delays," *Computers and Chemical Engineering*, vol. 28, no. 8, pp. 1337–1346, 2004.
- [6] Y. Zhang and S. Li, "Networked model predictive control based on neighbourhood optimization for serially connected large-scale processes," *Journal of Process Control*, vol. 17, no. 1, pp. 37–50, 2007.
- [7] J. Liu, D. M. de la Peña, B. J. Ohran, P. D. Christofides, and J. F. Davis, "A two-tier architecture for networked process control," *Chemical Engineering Science*, vol. 63, no. 22, pp. 5394–5409, 2008.
- [8] N. H. Ei-Farra, A. Gani, and P. D. Christofides, "Fault-tolerant control of process systems using communication networks," *AIChE Journal*, vol. 51, no. 6, pp. 1665–1682, 2005.
- [9] D. M. de la Peña and P. D. Christofides, "Output feedback control of nonlinear systems subject to sensor data losses," *Systems and Control Letters*, vol. 57, no. 8, pp. 631–642, 2008.
- [10] R. Luck and A. Ray, "Experimental verification of a delay compensation algorithm for integrated communication and control systems," *International Journal of Control*, vol. 59, no. 6, pp. 1357–1372, 1994.
- [11] L. Guo and H. Wang, "Generalized discrete-time PI control of output PDFs using square root B-spline expansion," *Automatica*, vol. 41, no. 1, pp. 159–162, 2005.
- [12] H. T. Su and T. J. McAvoy, "Integration of multilayer perceptron networks and linear dynamic models: a hammerstein modeling approach," *Industrial and Engineering Chemistry Research*, vol. 32, no. 9, pp. 1927–1936, 1993.
- [13] D. C. Psychogios and L. H. Ungar, "Direct and indirect model based control using artificial neural networks," *Industrial & Engineering Chemistry Research*, vol. 30, no. 12, pp. 2564–2573, 1991.
- [14] E. P. Nahas, M. A. Henson, and D. E. Seborg, "Nonlinear internal model control strategy for neural network models," *Computers and Chemical Engineering*, vol. 16, no. 12, pp. 1039–1057, 1992.
- [15] M. Nikolaou and V. Hanagandi, "Control of nonlinear dynamical systems modeled by recurrent neural networks," *American Institute of Chemical Engineers Journal*, vol. 39, no. 1, pp. 1890–1894, 1993.
- [16] I. Pan, S. Das, and A. Gupta, "Tuning of an optimal fuzzy PID controller with stochastic algorithms for networked control systems with random time delay," *ISA Transactions*, vol. 50, no. 1, pp. 28–36, 2011.
- [17] J. H. Zhang, "Neural controllers for networked control systems based on minimum tracking error entropy," *Journal of Systems and Control Engineering*, vol. 222, no. 7, pp. 671–679, 2008.
- [18] R. Ghostine, J. M. Thiriet, and J. F. Aubry, "Variable delays and message losses: influence on the reliability of a control loop," *Reliability Engineering and System Safety*, vol. 96, no. 1, pp. 160–171, 2011.
- [19] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.
- [20] P. Wen, J. Cao, and Y. Li, "Design of high-performance networked real-time control systems," *IET Control Theory and Applications*, vol. 1, no. 5, pp. 1329–1335, 2007.
- [21] Y. Tipsuwan and M. Y. Chow, "Control methodologies in networked control systems," *Control Engineering Practice*, vol. 11, no. 10, pp. 1099–1111, 2003.
- [22] D. Erdogmus and J. C. Principe, "An error-entropy minimization algorithm for supervised training of nonlinear adaptive systems," *IEEE Transactions on Signal Processing*, vol. 50, no. 7, pp. 1780–1786, 2002.
- [23] W. E. Deming, *Quality, Productivity and Competitive Position*, MIT Center for Advanced Engineering Study, Cambridge, Mass, USA, 1982.
- [24] D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*, John Wiley & Sons, New York, NY, USA, 4th edition, 2007.
- [25] ASTM, *Manual on Presentation of Data and Control Chart Analysis*, Publication MNL-7, American Society for Testing & Materials, Philadelphia, Pa, USA, 6th edition, 1990.
- [26] P. C. Badavas, *Real-Time Statistical Process Control*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1993.
- [27] C. A. Smith and A. B. Corripio, *Principles and Practices of Automatic Process Control*, John Wiley & Sons, New York, NY, USA, 3rd edition, 2006.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

