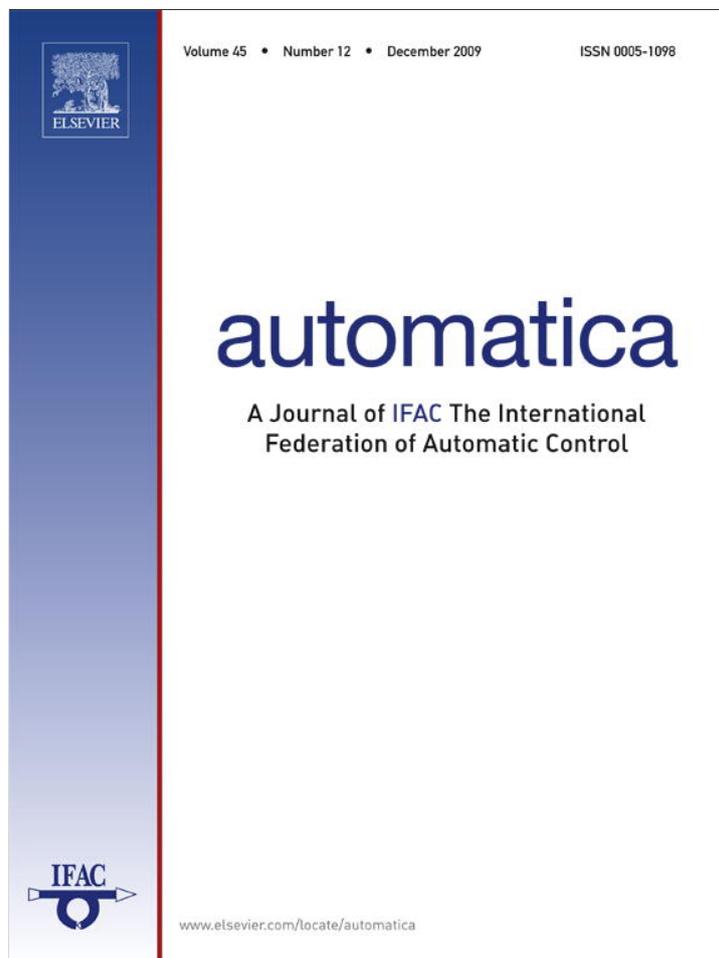


Provided for non-commercial research and education use.  
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



Contents lists available at ScienceDirect

Automatica

journal homepage: [www.elsevier.com/locate/automatica](http://www.elsevier.com/locate/automatica)

# Identification of linear dynamic systems operating in a networked environment<sup>☆</sup>

Jiandong Wang<sup>a</sup>, Wei Xing Zheng<sup>b,\*</sup>, Tongwen Chen<sup>c</sup>

<sup>a</sup> Department of Industrial Engineering & Management, College of Engineering, Peking University, Beijing 100871, China

<sup>b</sup> School of Computing & Mathematics, University of Western Sydney, Penrith South DC NSW 1797, Australia

<sup>c</sup> Department of Electrical & Computer Engineering, University of Alberta, Edmonton, Alberta, Canada T6G 2V4

## ARTICLE INFO

### Article history:

Received 2 January 2009  
Received in revised form  
8 June 2009  
Accepted 17 August 2009  
Available online 12 October 2009

### Keywords:

System identification  
Networked control systems  
Non-synchronized non-uniformly sampled data

## ABSTRACT

This paper studies a networked system identification problem, which aims at identifying mathematical models required in networked control/estimation/filtering systems. Specifically, we consider the off-line identification of open-loop stable linear time-invariant processes working in a networked environment. In the networked environment, how the actuators (D/A conversion) operate plays a key role in the complexity of the related identification problems. In particular, it is reasonable to consider the configuration of event-driven actuators subject to random network-induced delays and packet dropouts; as a result, the networked identification problem is formulated as the one to identify continuous-time linear time-invariant models, based on the general non-uniformly non-synchronized sampled data. A modified version of the simplified refined instrumental variable method is proposed to solve this problem, and is validated in a networked identification experiment based on the Matlab/Simulink simulator TrueTime.

© 2009 Elsevier Ltd. All rights reserved.

## 1. Introduction

The last decade has witnessed an accelerated integration of communication, computing networks and control systems at almost all levels of operation and information processing in various areas, including manufacturing industry, remote operation and tele-autonomy (Hespanha, Naghshtabrizi, & Xu, 2007; Hokayem & Abdallah, 2004; Tipsuwan & Chow, 2003; Yang, 2006). Ever since, networked control/estimation/filtering systems have been very challenging and promising research fields, and have received much attention from researchers and engineers; see, e.g., the special issues Antsaklis and Baillieul (2004, 2007), Bushnell (2001), Chow (2004) and Wang, Liu, Yang, and Li (2007) and numerous references therein.

One of the distinguishing features for networked control/estimation/filtering systems is data transmission over communication networks. As a matter of fact, this feature is not a new concept.

From tele-operation for space and hazardous environment to process regulation with distributed control systems, for instance, data transmission over communication networks has already been utilized for many years. These communication networks are usually specialized and dedicated to ensure the timeliness of the transmission. However, with the trend of Ethernet, Internet, and wireless networks becoming dominant, the networks are no longer specialized and are often shared for various concurrent general-purpose applications (Eidson & Cole, 1998; Kaplan, 2001; Liu, Chai, Mu, & Rees, 2008; Moyne & Tilbury, 2007; Tang & de Silva, 2006; Thompson, 2004; Yang, Chen, & Alty, 2003). As a result, the effects of communication networks, e.g., network-induced delays and packet dropouts, are too prominent to be ignorable in many applications, and need to be considered in the design and analysis of networked controllers/estimators/filters.

This paper studies the problem of identifying mathematical models of processes working in the networked environment, referred to as the networked identification problem. Needless to say, the identified models are indispensably required in the networked control/estimation/filtering systems. To the best of our knowledge, this problem has received relatively low attention so far. Brscic, Petrovic, and Peric (2000) considered constant round-trip delays and maximum delays, and performed an on-line identification of discrete-time (DT) linear time-invariant (LTI) models using a recursive least-squares algorithm. Fei, Du, and Li (2008) introduced an actuator buffer to deal with the network non-deterministic factors from the identifier to the actuator; by doing so, the problem is converted into identification of DT LTI models with some missing

<sup>☆</sup> This research was partially supported by the National Natural Science Foundation of China under grant Nos. 60704031 and 10832006, by a Research Grant from the Australian Research Council, by the Natural Sciences and Engineering Research Council of Canada and by the 111 Project. The material in this paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor Johan Schoukens under the direction of Editor Torsten Söderström.

\* Corresponding author. Tel.: +61 2 4736 0608; fax: +61 2 4736 0867.

E-mail addresses: [jiandong@pku.edu.cn](mailto:jiandong@pku.edu.cn) (J. Wang), [w.zheng@uws.edu.au](mailto:w.zheng@uws.edu.au) (W.X. Zheng), [tchen@ece.ualberta.ca](mailto:tchen@ece.ualberta.ca) (T. Chen).

output data points (this is equivalent to Case I in Section 2.3 in this paper).

The contribution of this paper is two-fold. (i) A fundamental question is addressed: what kind of identification problems are there for an LTI process working in the networked environment? This is not an easy question to answer. To have a feasible start, we limit ourselves to the off-line identification of open-loop stable LTI models. Under the effects of random network-induced delays and packet dropouts, it is found out in this paper that the distinct running modes of the actuator (D/A conversion) lead to several identification problems with very different complexities. Considering some reasonable assumptions (Assumptions 1 and 2 in Section 2), we formulate the networked identification problem under the configuration of event-driven actuators subject to random network-induced delays and packet dropouts. That is, a continuous-time (CT) LTI model is to be identified based on the general non-synchronized non-uniformly (NSNU) sampled data. (ii) To the best of our knowledge, the CT model identification problem based on general NSNU sampled data has seldom been considered in the literature (see the literature survey in Section 3.2). Taking the main idea of the simplified refined instrumental variable method for CT systems (SRIVC), proposed by Young and his coworkers (Huselstein & Garnier, 2002; Young, 2002, 2008; Young, Garnier, & Gilson, 2008; Young & Jakeman, 1980), a modified SRIVC method is proposed in this paper to solve the networked identification problem. The proposed method is validated in a networked identification experiment based on the Matlab/Simulink simulator TrueTime (Cervin, Henriksson, Lincoln, Eker, & Arzen, 2003).

The rest of the paper is organized as follows. Section 2 investigates the types of models to be identified, where two factors are considered, namely, the effects of communication networks and the running modes of nodes in the network. Section 3 formulates a particular networked identification problem to be solved in the rest of this paper. Section 4 proposes a modified version of the simplified refined instrumental variable method to identify CT LTI models based on the general NSNU sampled data. Section 5 presents a numerical example based on the Matlab/Simulink simulator TrueTime to illustrate the effectiveness of the proposed method in solving the networked identification problem. Finally, some concluding remarks are given in Section 6.

## 2. Different models arisen in networked environments

This section starts with a brief description of the identification problem in the networked environment. Two factors, namely, the effects of communication networks and the running modes of nodes, are analyzed in detail to reach two types of models including simple LTI models as well as complicated DT models which are time-varying in parameters and in structures.

### 2.1. Problem description

Consider a CT LTI process working in a networked environment depicted in Fig. 1, where the timeline of the signals goes as follows. The process output  $y(t)$  for  $t \in \mathbb{R}^+$  (the set of positive real numbers) is sampled at the time instant  $t_k$  by a sensor to yield a DT counterpart  $y(t_k)$ . The communication network transmits  $y(t_k)$  to a remote computer that receives  $y_c(t_l)$  at the time instant  $t_l$  to produce  $y_c(t_l)$ . The computer, acting as an identification device, sends out a DT excitation input  $u_c(t_j)$  via the communication network at the time instant  $t_j$ , which is received by an actuator at the time instant  $t_i$  to form  $u(t_i)$ . The actuator creates the CT input  $u(t)$  based on  $u(t_i)$  via zero-order hold (ZOH) to drive the process. Here the time instants  $t_k$ ,  $t_l$ ,  $t_j$  and  $t_i$  and their relationships

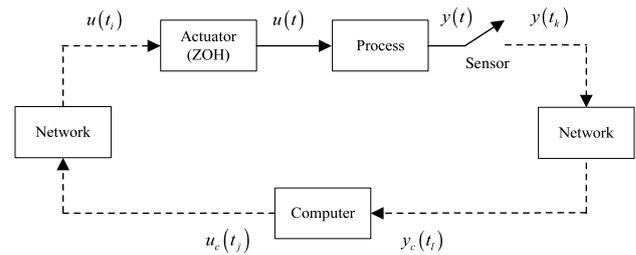


Fig. 1. A diagram of a CT LTI process working in the networked environment.

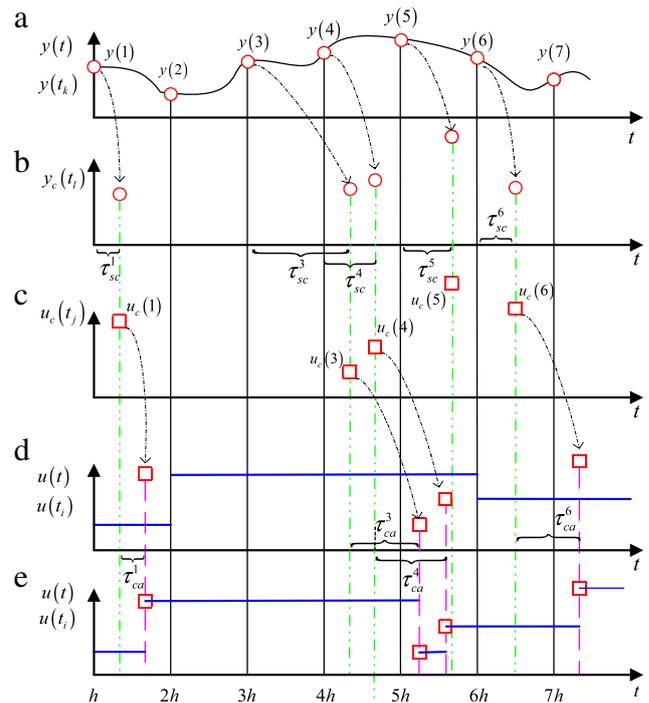


Fig. 2. The timeline of signals in the networked environment:  $y(t_k)$  and  $y_c(t_l)$  in circles,  $u_c(t_j)$  and  $u(t_i)$  in squares, and  $y(t)$  and  $u(t)$  in solid lines.

depend on the effects of communication networks, and the running modes of the sensor, the computer (identification device) and the actuator, and will be clarified later. Fig. 2 presents an illustrative example of these signals.

We study the problem of identifying a CT model or a DT counterpart of the process working in this networked environment, based on the available data  $y_c(t_l)$  and  $u_c(t_j)$  at the computer side. In other words, identification is performed in a remote manner via the communication network between the process and the computer. Henceforth, this problem is referred to as the networked identification problem. In particular, we limit ourselves to the off-line networked identification problem for open-loop stable LTI systems. In Fig. 1, even though  $t_j$  may depend on  $t_l$  (for an event-driven identification device), the value of  $u_c(t_j)$  is generated independent of  $y_c(t_l)$  so that an open-loop identification experiment is implemented. The problem studied in this paper could be a good starting point to tackle the on-line networked identification problem for closed-loop systems, which may be practically more important, but more complicated.

There are two factors which are distinctly different from the standard identification problems, namely, the effects of communication networks and the running modes of nodes in the network including the sensor, computer and actuator. The two factors are investigated in Sections 2.2 and 2.3.

## 2.2. Effects of communication networks

We consider two major effects of communication networks, namely, network-induced delays and packet dropouts. With the increasing usage of Ethernet, Internet and wireless communication networks, these effects of communication networks are too prominent to be ignorable in many applications.

First, there are two network-induced delays in Fig. 1:  $\tau_{sc}^k \in \mathbb{R}$  is the sensor-to-computer delay associated with  $y(t_k)$ , and  $\tau_{ca}^j \in \mathbb{R}$  is the computer-to-actuator delay associated with  $u_c(t_j)$ . The two delays  $\tau_{sc}^k$  and  $\tau_{ca}^j$  are inevitable, due to limited bandwidth and overhead in the network. In fact, they are sums of several small delays, including medium access delay, transportation delay, and congestion delay. The characteristics of  $\tau_{sc}^k$  and  $\tau_{ca}^j$  can be constant or random, depending on the network architectures, medium access control protocols, operating conditions, and the chosen hardware. In many cases, especially for Ethernet, Internet and wireless communication networks, the delays are time-varying in a random fashion (Lian, Moyne, & Tilbury, 2002; Nilsson, 1998; Srinivasagupta, Schattler, & Joseph, 2004; Yang et al., 2003; Yang, 2006; Zhang, Branicky, & Phillips, 2001). Hence, both  $\tau_{sc}^k$  and  $\tau_{ca}^j$  here are assumed to be random, e.g.,  $\tau_{sc}^k$  and  $\tau_{ca}^j$  in Fig. 2.

Second, data samples of  $y(t_k)$  and  $u_c(t_j)$  may be lost/dropped while in transmission through the communication network. The network packet dropout is an inherent problem with most communication networks due to several factors such as node failure, transmission errors and buffer overflow resulting from congestion. Although network protocols are usually equipped with transmission-retry mechanisms, they can only re-transmit for a limited time; after this time has expired, the packets are dropped (Hokayem & Abdallah, 2004; Zhang et al., 2001). The packet dropouts can be modelled either as stochastic or deterministic phenomena (Hespanha et al., 2007), against which the proposed off-line identification problem actually has no discrimination. Without loss of generality, we assume that some random packet dropouts may happen at the sensor-to-computer side in Fig. 1, i.e., some samples of  $y(t_k)$  may be lost/dropped with a certain probability in the transmission. Similarly, the packet dropouts of  $u_c(t_j)$  at the computer-to-actuator side may also happen; however, they have little effects on the open-loop identification owing to the fact that the ZOH keeps its previous value if the update does not arrive. For example,  $y(2)$  and  $u_c(5)$  in Fig. 2 are lost in the transmission.

Therefore, we make the following assumption on the effects of communication networks.

**Assumption 1.** The network-induced delays  $\tau_{sc}^k$  and  $\tau_{ca}^j$  are assumed to be random within the range  $[0, \tau_{\max}]$  for some real valued constant  $\tau_{\max}$ ; the communication network has a probability  $\alpha$  ( $0 < \alpha < 1$ ) of packet dropouts in the transmission.

Note that both  $\tau_{\max}$  and  $\alpha$  are not required in the identification. In addition,  $\tau_{\max}$  could be larger than the sampling period  $h$  that is defined later to be associated with  $y(t_k)$ . The probability  $\alpha$  can be time-varying, which often occurs if the communication traffic varies with time.

## 2.3. Running modes of the sensor, computer and actuator

The other factor different from the standard identification problems is the running mode of the nodes including the sensor, computer and actuator. The running mode has two types, namely, the event-driven running mode or the time-driven one.

First, from a sampled-data system perspective, it is natural to sample the process output  $y(t)$  equidistantly with a sampling period  $h \in \mathbb{R}^+$ . In other words, the sensor is in the time-driven running mode, and  $y(t_k)$  is uniformly sampled for  $t_k = kh$  where  $k \in \mathbb{Z}^+$  (the set of positive integer), see, e.g.,  $y(t_k)$  in Fig. 2(a).

Second, the running mode of the computer, acting as the identification device, does not really matter, because the identification is performed in the off-line manner. Without losing generality, the identification device is set to run in the event-driven mode: once  $y(t_k)$  is received as  $y_c(t_l)$  by the computer, the identification device transmits the excitation input  $u_c(t_j)$  out to the actuator via the communication network; see, e.g.,  $y_c(t_l)$  and  $u_c(t_j)$  in Fig. 2(b) and (c) respectively. However, the value of  $u_c(t_j)$  is independent of  $y_c(t_l)$ , i.e., it is an open-loop identification experiment. In addition, we ignore the computational delay in the computer and assume  $u_c(t_j)$  to be sent out from the computer at the moment that  $y_c(t_l)$  is received, i.e.,  $t_j = t_l$ .

Finally, under Assumption 1, the two running modes for the actuator lead to different types of models to be identified.

Case I: If the actuator is time-driven with updating period  $h$ , the CT input  $u(t)$  is piece-wise constant and is updated every  $h$  time unit by taking the value of the latest updated input  $u_c(t_j)$ , as shown in Fig. 2(d). The time-driven mode certainly requires the actuator having a buffer to store the received input  $u_c(t_j)$ . If there are several updated samples available within one sampling period, then all except the latest one are discarded, e.g., the time interval  $[5h, 6h)$  in Fig. 2(d). If no updating information is available within one sampling period, the actuator adopts the strategy of ZOH and preserves its previous value, e.g., the time interval  $[3h, 5h)$  in Fig. 2(d). For such a time-driven actuator, the CT input  $u(t)$  has no variation within one sampling period. If  $u(kh)$  and  $y(kh)$ , the equidistantly sampled points of  $u(t)$  and  $y(t)$  at the sampling instants  $kh$  respectively, can be recovered, then we are facing a standard identification problem for a DT/CT LTI model.

Case II: If the actuator is event-driven, the CT input  $u(t)$  may experience zero, one or multiple updates every  $h$  time unit due to the random network-induced delays (Branicky, Phillips, & Zhang, 2000; Chow & Tipsuwan, 2001; Zhang et al., 2001). For instance, the time intervals  $[2h, 3h)$ ,  $[h, 2h)$ , and  $[5h, 6h)$  in Fig. 2(e) have zero, one, and two variations, respectively. If the DT model is to be identified, then it is time-varying in parameters and in structures. This can be seen from the following example.

**Example 1.** Let the CT LTI process be in the state-space form,

$$\begin{aligned} \frac{dx(t)}{dt} &= Ax(t) + Bu(t), \\ y(t) &= Cx(t). \end{aligned}$$

Suppose that the CT input  $u(t)$  always experiences one variation within one sampling period  $h$ , i.e., one single updated excitation signal  $u_c(t_j)$  arrives at the actuator by a time instant  $t_i = (k-1)h + \Delta_k$  for some integer  $k \in \mathbb{Z}$  and  $0 < \Delta_k < h$ . By the step-invariant transformation (e.g., Chen and Francis (1995)), the DT model is

$$\begin{aligned} x((k+1)h) &= \Phi x(kh) + \Gamma_0(\Delta_k)u(kh) \\ &\quad + \Gamma_1(\Delta_k)u((k-1)h), \\ y(kh) &= Cx(kh), \end{aligned}$$

where

$$\Phi = e^{Ah}, \quad \Gamma_0(\Delta_k) = \int_0^{h-\Delta_k} e^{At} dt \cdot B,$$

$$\Gamma_1(\Delta_k) = \int_{h-\Delta_k}^h e^{At} dt \cdot B.$$

The DT model also can be written in the transfer function form as

$$\begin{aligned}
 y(kh) &= C(I - q\Phi)^{-1} \Gamma_0(\Delta_k)u(kh) \\
 &\quad + C(I - q\Phi)^{-1} \Gamma_1(\Delta_k)u((k-1)h) \\
 &=: \frac{B_0(q; k)}{A(q)}u(kh) + \frac{B_1(q; k)}{A(q)}u((k-1)h).
 \end{aligned}$$

In general, there are perhaps  $m$  variations within one sampling period  $h$ ; thus, the DT model contains  $m + 1$  delayed inputs with different numerator parameters. Since  $m$  and  $\Delta_k$  are time-varying due to the random network-induced delays, the DT model is time-varying in parameters and in structures.  $\square$

Identification of the DT model, being time-varying in parameters and in structures, would be very difficult. However, note that the time-varying nature is purely caused by the event-driven running mode of the actuator and the random network-induced delays, even though the underlying CT process is LTI. Therefore, it would be more reasonable to study the networked identification problem in the CT domain, i.e., the objective is to identify a CT LTI model based on the DT samples  $y(t_k)$  and  $u(t_i)$ .

The selection of the running mode for the actuator deserves a further discussion here. The introduction of a buffer in the time-driven actuator implies some extra waiting time required, that is, the waiting for the next actuator equidistantly-spaced updating instant to arrive before the process input can be updated. In other words, we sometimes are using earlier information than we need to (see Fig. 2(d)). This can lead to a degradation of performance, in particular for networked control systems, in comparison with an event-driven actuator. Thus, event-driven actuators are usually preferred to time-driven ones (Nilsson, 1998; Yang, 2006).

As a summary, we make the following assumption on the running modes of the sensor, computer and actuator.

**Assumption 2.** The sensor is time-driven with sampling period  $h \in \mathbb{R}^+$  time unit. The identification device in the computer is event-driven, sending out  $u_c(t_j)$  at the moment of receiving  $y_c(t_i)$ ; however, the value of  $u_c(t_j)$  is generated in advance before the identification experiment, being independent of  $y_c(t_i)$ . The actuator is in the event-driven running mode of updating  $u(t)$  at the moment  $t_i$  when  $u_c(t_j)$  is received as  $u(t_i)$ .

### 3. Formulation of a networked identification problem

As discussed in Section 2, it is reasonable to consider the networked identification problem under Assumptions 1 and 2 in the CT domain. This section first discusses the recovery of the process input  $u(t_i)$  and process output  $y(t_k)$ , under the help of the time-stamping technique. Next, a networked identification problem is formulated, i.e., a CT LTI model is to be identified based on the NSNU sampled data.

#### 3.1. Recovery of the process input and process output

The networked identification problem needs to recover the process input  $u(t_i)$  and process output  $y(t_k)$  in order to identify the process parameters, based on the available data  $y_c(t_i)$  and  $u_c(t_j)$  at the computer side.

To do so, we need to measure the information of network-induced delays and packet dropouts. This can be achieved by the so-called time-stamping technique. The transmitted output  $y(t_k)$  is marked with the time when it was generated. Once the packet including  $y(t_k)$  and its time stamp is received at the computer, the sensor-to-computer delay  $\tau_{sc}^k$  is calculated by comparing the time stamp of  $y(t_k)$  with the internal clock of the computer. Similarly, the actuator node can calculate the computer-to-actuator delay  $\tau_{ca}^j$  associated with  $u_c(t_j)$  and  $t_j$  sent by the computer. The information

of  $\tau_{ca}^j$  can be available to the computer in two ways. One way is to immediately send a message containing  $\tau_{ca}^j$  back from the actuator to the computer, while the other way is to pass  $\tau_{ca}^j$  on to the sensor and to include  $\tau_{ca}^j$  in the next sensor-to-computer data packet (Nilsson, 1998; Tang & de Silva, 2006). Certainly, the actuator is assumed here to be capable of additional computation (to calculate  $\tau_{ca}^j$ ) and communication (to send  $\tau_{ca}^j$  to the computer or to pass on it to the sensor). In most networks, the extra network load introduced by transmitting the time stamps or the network-induced delays is negligible in comparison with the load of transmitting signals and the network overhead. It is worthy to note that the time-stamping technique requires clock synchronization (being a research area in itself) in all nodes of the communication network, which can be achieved via the software synchronization or the hardware synchronization (Nilsson, 1998; Yang, 2006); see e.g. Faizulkhakov (2007), Ishikawa and Mita (2008) and Su (2008) for the state-of-the-art clock synchronization techniques. The time-stamping technique also needs the sequence numbering technique to assign a sequence number for each packet; otherwise, the time stamp alone cannot tell whether a packet is lost or not (see e.g. Forouzan (2007) (Pages 318 and 915 therein)). As a matter of fact, the time-stamping technique has been exploited in many studies of networked control systems, e.g., Lian et al. (2002), Nilsson (1998), Srinivasagupta et al. (2004) and Tang and de Silva (2006). Hence, we make an assumption on using the time-stamping technique.

**Assumption 3.** The time-stamping technique is used for all the data transmission.

Under Assumption 3, both the process input  $u(t_i)$  and process output  $y(t_k)$  can be completely recovered, based on the data points of  $y_c(t_j)$  and  $u_c(t_i)$  collected at the computer side, the information of  $\tau_{sc}^k$  and  $\tau_{ca}^j$ , the running modes of the sensor, computer and actuator, and the ZOH property of the actuator. The recovery certainly is not available at the moment of  $t_i$  or  $t_k$ , but is done after waiting for a certain period of time to receive all the required information; this is normally not a problem for the off-line identification here.

In particular,  $y(t_k)$  is recovered from the received packet containing the values and time stamp of  $y(t_k)$ . Here  $t_k = kh$  with  $k \in \mathbb{Z}^+$  because the sensor is in the time-driven running mode. At the ending stage of data collection, if the value and time stamp of  $y(t_k)$  at  $t_k = kh$  are not received due to packet dropouts,  $y(t_k)$  cannot be recovered and is regarded as the missing data.

The value of  $u_c(t_j)$ , its transmission status of being success or failure, and the associated computer-to-actuator delay  $\tau_{ca}^j$  tell us the full information of  $u(t_i)$ . Here the transmission status of  $u_c(t_j)$  is known to its sender computer, since most transmission protocols have the mechanism to provide this information. However, there is one exceptional scenario where  $u(t_i)$  cannot be recovered. As previously discussed,  $\tau_{ca}^j$  is transmitted back to the computer via either side of the communication channels. If the packet containing  $\tau_{ca}^j$  is dropped in the transmission, and the corresponding  $u_c(t_j)$  has a success transmission status, there will be no way of recovering the corresponding input  $u(t_i)$ . This exceptional scenario can be treated the same as the case that  $u(t)$  is taking the value of the last received input sample  $u(t_{i-1})$  without a new update; such a treatment inevitably introduces errors. To simplify the discussion, we ignore this exceptional scenario in what follows and leave it for future studies.

#### 3.2. Non-synchronized non-uniformly (NSNU) sampled data

Let us take a closer look at  $u(t_i)$  and  $y(t_k)$ . Besides the illustrative example in Fig. 2, a typical example of  $y(t_k)$  and  $u(t_i)$  from Section 5 is shown in Fig. 3.

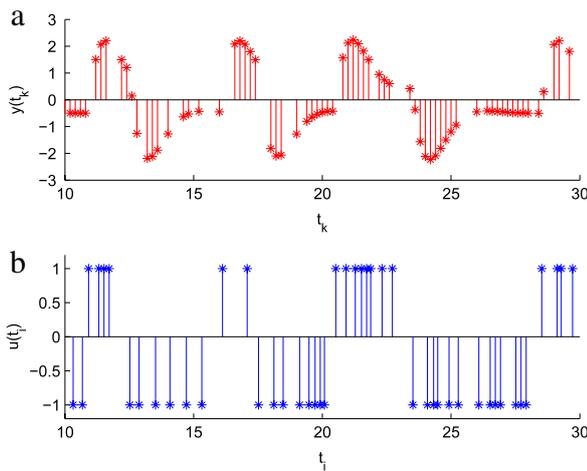


Fig. 3. An example of  $y(t_k)$  and  $u(t_i)$ .

First,  $y(t_k)$  is uniformly sampled for most of time, except at some instants where the data points are missed owing to the dropout of the packets containing  $y(t_k)$ . Second,  $u(t_i)$  is completely non-uniformly sampled, due to the random network-induced delays; in addition, the intersample behavior of  $u(t_i)$  is piece-wise constant because of the ZOH. As a result, the sampling patterns of  $u(t_i)$  and  $y(t_k)$  can be regarded as general non-synchronized non-uniformly (NSNU) sampled data, i.e., both the input and output are non-uniformly sampled, and they do not have to be synchronized. It is worthy to point out that the sensor does not have to work in the time-driven mode, i.e., the CT output  $y(t)$  can be sampled in a non-periodic fashion (Hespanha et al., 2007); in this case,  $y(t_k)$  in general is non-uniformly sampled.

To the best of our knowledge, the NSNU sampled data is seldom considered in the literature of CT model identification (Garnier & Wang, 2008). The most relevant work is based on non-uniformly sampled but synchronized data. Tsang and Billings (1995) studied the identification of CT model from non-uniformly sampled data via a state variable filter with the key step of adapting the Euler method or Runge–Kutta method for non-uniformly sampled data. Huselstein and Garnier (2002), Young (2002) and Young et al. (2008) discussed the capability of applying the simplified refined instrumental variable method for non-uniformly sampled data. Larsson, Mossberg, and Soderstrom (2007, 2008) devised a discrete approximation of the differentiation operator for CT model identification and extended the derivative approximation to handle the case of irregular sampling. Ahmed, Huang, and Shah (2008) took the linear filter method to provide the unmeasurable derivatives and estimated the time delay and system parameters simultaneously from the non-uniformly sampled data. In these references and examples presented therein, both the output and input are non-uniformly sampled, but are synchronized, i.e.,  $t_k$  is not evenly-spaced and  $t_i$  is the same as  $t_k$  in the current context. Even though some of the above methods have the potential of being applicable to the NSNU sampled data  $y(t_k)$  and  $u(t_i)$  here, none of them have indeed considered this particular sampling pattern arisen in the networked environment.

To be precise, the networked identification problem is stated as follows: *Given the recovered non-uniformly sampled output  $y(t_k)$  for  $t_k \in \mathbb{R}^+$ , and the non-uniformly sampled input  $u(t_i)$  for  $t_i \in \mathbb{R}^+$  with the piece-wise constant intersample behavior, identify in an off-line manner the asymptotically stable CT model working in an open-loop operation. Note that  $y(t_k)$  and  $u(t_i)$  are not synchronized, i.e.,  $t_k$  in general is different from  $t_i$ .*

#### 4. The modified SRIVC method

Among the related work mentioned in Section 3.2, the SRIVC method proposed by Young and his coworkers (Huselstein & Garnier, 2002; Young, 2002, 2008; Young et al., 2008; Young & Jakeman, 1980), referred to as the original SRIVC method in what follows, has been successfully validated via numerical and practical examples over years. This method is implemented by the function “srivc” in the latest version of CONTSID toolbox (Garnier, Gilson, Bastogne, & Mensler, 2008) (available at <http://www.cran.uhp-nancy.fr/contsid/>), and can be applied for non-uniformly sampled but synchronized input and output data. Taking the main idea of the original SRIVC method, we propose a modified SRIVC method that can deal with the general NSNU sampled data.

##### 4.1. Problem formulation

Let the CT LTI process in Fig. 1 be described by a constant coefficient differential-delay equation,

$$\begin{aligned} \frac{d^n x(t)}{dt^n} + a_1 \frac{d^{n-1} x(t)}{dt^{n-1}} + \dots + a_n x(t) \\ = b_0 \frac{d^m u(t - \tau)}{dt^m} + \dots + b_m u(t - \tau). \end{aligned} \quad (1)$$

The orders  $m$  and  $n$  ( $n \geq m$  for a proper model), as well as the time delay  $\tau$  between  $x(t)$  and  $u(t)$ , are assumed to be known *a priori*; otherwise, they can be determined together with the parameter estimation, e.g., in the same manner as that in Section 4.6 of Young et al. (2008). In what follows,  $\tau$  is ignored to simplify the notation. By ignoring  $\tau$  and taking  $p$  as the differential operator, i.e.,

$$p^i x(t) = \frac{d^i x(t)}{dt^i},$$

Eq. (1) is rewritten in a compact transfer function form

$$x(t) = G(p)u(t) = \frac{B(p)}{A(p)}u(t), \quad (2)$$

with

$$\begin{aligned} B(p) &= b_0 p^m + b_1 p^{m-1} + \dots + b_m, \\ A(p) &= p^n + a_1 p^{n-1} + \dots + a_n. \end{aligned}$$

Here  $A(p)$  and  $B(p)$  are assumed to be coprime, and  $G(p)$  is asymptotically stable. Consider an additive noise  $e(t)$  corrupting  $x(t)$  to yield the output measurement  $y(t)$ , i.e.,  $y(t) = x(t) + e(t)$ . If  $y(t)$  is non-uniformly sampled to yield  $y(t_k)$ , i.e.,  $t_k$  is not evenly-spaced, then  $e(t_k)$  is the noise associated with  $y(t_k)$ ,

$$y(t_k) = x(t_k) + e(t_k). \quad (3)$$

For the moment, assume that  $e(t_k)$  is a DT Gaussian white noise with zero mean and variance  $\sigma_e^2$ ; the colored noise case will be discussed later in Sections 4.2 and 4.3. The maximum likelihood estimate of  $e(t_k)$  is given by

$$\begin{aligned} \varepsilon(t_k) &= y(t_k) - x(t_k) \\ &= y(t_k) - \frac{B(p)}{A(p)}u(t_i). \end{aligned} \quad (4)$$

Here the expression  $x(t_k) = \frac{B(p)}{A(p)}u(t_i)$  stands for the operation of sampling at the time instant  $t_k$  the output of the CT filter  $B(p)/A(p)$  driven by the CT counterpart of  $u(t_i)$ . Note that  $u(t_i)$  is available here instead of  $u(t_k)$ . Since the polynomial operator commutes in the linear case, (4) becomes

$$\begin{aligned} \varepsilon(t_k) &= \frac{1}{A(p)}(A(p)y(t_k) - B(p)u(t_i)) \\ &=: A(p)y_f(t_k) - B(p)u_f(t_k), \end{aligned} \quad (5)$$

where

$$y_f(t_k) := \frac{1}{A(p)}y(t_k), \quad u_f(t_k) := \frac{1}{A(p)}u(t_k).$$

Eq. (5) yields a linear regression model,

$$y_f^{(m)}(t_k) = \varphi_f^T(t_k)\theta + \varepsilon(t_k), \quad (6)$$

where

$$\varphi_f(t_k) = \begin{bmatrix} -y_f^{(n-1)}(t_k), & -y_f^{(n-2)}(t_k), & \dots, & -y_f(t_k), \\ u_f^{(m)}(t_k), & u_f^{(m-1)}(t_k), & \dots, & u_f(t_k) \end{bmatrix}^T,$$

$$\theta = [a_1, a_2, \dots, a_n, b_0, b_1, \dots, b_m]^T.$$

Here  $y^{(i)}(t)$  or  $u^{(i)}(t)$  denotes the  $i$ th time derivative of the CT signal  $y(t)$  or  $u(t)$ . The procedure of obtaining the filtered signals  $y_f(t_k)$  and  $u_f(t_k)$  and their derivatives  $y_f^{(n)}(t_k), y_f^{(n-1)}(t_k), \dots, y_f^{(1)}(t_k)$  and  $u_f^{(m)}(t_k), u_f^{(m-1)}(t_k), \dots, u_f^{(1)}(t_k)$  will be addressed later in Section 4.2. The identification problem can now be stated as follows: Given the collected data  $u(t_i)$  and  $y(t_k)$ , identify the parameter vector  $\theta$ .

#### 4.2. Identification algorithm

We propose a modified SRIVC method for the general NSNU sampled data in the networked environment. The main idea is to introduce an iteration between estimating the parameter  $\theta$  based on (6) by the instrumental variable method (IVM), and updating the instrumental variable generated by an auxiliary model based on the previous estimated parameter.

A modified SRIVC method for the NSNU sampled Data:

Stage A. Initialization:

A1. Design a filter

$$f_c(p) = \frac{1}{(p + \lambda)^n}, \quad (7)$$

where  $\lambda \in \mathbb{R}$  is chosen so that it is equal to or larger than the bandwidth of  $G(p)$ . Note that  $n \geq m$  so that the order of  $f_c(p)$  is set to be  $n$ , and the knowledge of the bandwidth of  $G(p)$  is assumed to be known *a priori*, or can be obtained via some preliminary tests such as step tests that are normally performed before designing identification experiments in practice (see e.g., Section 3.2 of Zhu (2001)).

A2. Filter  $y(t_k)$  via the CT filter  $f_c(p)$  and sample the resulting signal and its derivatives at  $t_k$  as an estimation of  $y_f^{(n)}(t_k), y_f^{(n-1)}(t_k), \dots, y_f(t_k)$ . In the filtering operation (see Fig. 5), the intersample behavior of  $y(t_k)$  is not available and  $y(t_k)$  is interpolated in a first-order hold (FOH) fashion to obtain the CT counterpart  $y(t)$ .

A3. Filter  $u(t_i)$  via the CT filter  $f_c(p)$  and sample the resulting signal and its derivatives at  $t_k$  as an estimation of  $u_f^{(m)}(t_k), u_f^{(m-1)}(t_k), \dots, u_f(t_k)$ . Owing to the ZOH,  $u(t_i)$  is known to have the piece-wise constant intersample behavior to form the CT counterpart  $u(t)$ .

A4. Use the least-squares method to obtain an initial estimate  $\theta_0$  of  $\theta$  based on (6), i.e.,

$$\theta_0 = \left( \sum_{k=1}^N \varphi_f(t_k)\varphi_f^T(t_k) \right)^{-1} \sum_{k=1}^N \varphi_f(t_k)y_f^{(n)}(t_k).$$

Stage B. Iterative estimation: for  $j = 1$  : convergence

B1. Generate the estimates of the noise-free output  $x(t)$  that are sampled at the time instants  $t_k$  and  $t_{k,\Delta}$ , denoted as  $\hat{x}(t_{k,\Delta})$ , based on the previous estimate  $\theta_{j-1}$  and  $u(t_i)$ , i.e.,

$$\hat{x}(t_{k,\Delta}) = \frac{\hat{B}(p, \theta_{j-1})}{\hat{A}(p, \theta_{j-1})}u(t_i). \quad (8)$$

Once again,  $u(t_i)$  takes the piece-wise constant intersample behavior owing to the ZOH. Here the time index  $t_{k,\Delta}$  is evenly-spaced with a small incremental step comparing with the sampling period  $h$ .

B2. Filter  $y(t_k)$ ,  $u(t_i)$  and  $\hat{x}(t_{k,\Delta})$  via a CT filter  $f_c(p) = 1/\hat{A}(p, \theta_{j-1})$  and sample the resulting signals and their derivatives at  $t_k$  to reach

$$Y_f(t_k) := \begin{bmatrix} y_f^{(n)}(t_k), y_f^{(n-1)}(t_k), \dots, y_f(t_k) \end{bmatrix}^T,$$

$$U_f(t_k) := \begin{bmatrix} u_f^{(m)}(t_k), u_f^{(m-1)}(t_k), \dots, u_f(t_k) \end{bmatrix}^T,$$

$$X_f(t_k) := \begin{bmatrix} \hat{x}_f^{(n)}(t_k), \hat{x}_f^{(n-1)}(t_k), \dots, \hat{x}_f(t_k) \end{bmatrix}^T.$$

In the filtering operation, the intersample behaviors of  $y(t_k)$  and  $\hat{x}(t_{k,\Delta})$  are not available; both are interpolated in an FOH fashion. By contrast,  $u(t_i)$  has the piece-wise constant intersample behavior owing to the ZOH.

B3. Use the IVM to obtain the updated estimate  $\theta_j$ ,

$$\theta_j = \left( \sum_{k=1}^N \hat{\varphi}_f(t_k)\varphi_f^T(t_k) \right)^{-1} \sum_{k=1}^N \hat{\varphi}_f(t_k)y_f^{(n)}(t_k). \quad (9)$$

Here the instrumental variable  $\hat{\varphi}_f(t_k)$  is the noise-free counterpart of  $\varphi_f(t_k)$  where  $y_f^{(l)}(t_k)$  for  $l = 0, 1, \dots, n - 1$  are replaced by the estimate  $\hat{x}_f^{(l)}(t_k)$  of the noise-free output  $x_f^{(l)}(t_k)$ , i.e.,

$$\hat{\varphi}_f(t_k) = \begin{bmatrix} -\hat{x}_f^{(n-1)}(t_k), & -\hat{x}_f^{(n-2)}(t_k), & \dots, & -\hat{x}_f(t_k), \\ u_f^{(m)}(t_k), & u_f^{(m-1)}(t_k), & \dots, & u_f(t_k) \end{bmatrix}^T. \quad (10)$$

end

The steps in Stage B are diagrammatically illustrated in Fig. 4. In particular, the filtering operation is one of the kernel steps, and deserves a detailed illustration. Taking  $y_f(t_k) := \frac{1}{A(p)}y(t_k)$  for instance, the filtering by  $1/A(p)$  plays two roles of generating the filtered output  $y_f(t_k)$  and its derivatives  $y_f^{(n)}(t_k), y_f^{(n-1)}(t_k), \dots, y_f^{(1)}(t_k)$  required in (6), and of improving the estimation quality of  $\theta$ . The required filtering operation can be implemented by sampling the solution of the state-space model in (11) driven by  $y(t_k)$ :

$$\begin{bmatrix} y_f^{(n)}(t), & y_f^{(n-1)}(t), & \dots, & y_f^{(1)}(t) \end{bmatrix}^T \\ = \begin{bmatrix} -a_1 & -a_2 & \dots & -a_n \\ 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} y_f^{(n-1)}(t) \\ y_f^{(n-2)}(t) \\ \vdots \\ y_f(t) \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} y(t). \quad (11)$$

Here the intersample behavior of  $y(t_k)$  is not available and  $y(t_k)$  has to be interpolated in some manner (e.g., by taking the FOH intersample behavior) to obtain the CT counterpart  $y(t)$ . If the discretization interval used in the numerical integration is short comparing with the averaged sampling interval of  $y(t_k)$ , which is usually the case, then the error induced could be tolerable. This implementation is graphically illustrated in Fig. 5.

The filtering operation  $u_f(t_k) := \frac{1}{A(p)}u(t_i)$  is implemented in the same way to obtain  $u_f(t_k)$  and its derivatives  $u_f^{(m)}(t_k)$ ,

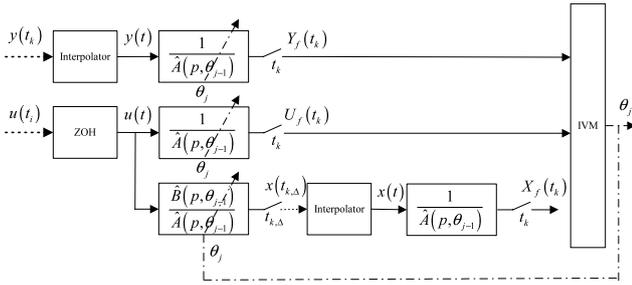


Fig. 4. The filtered signals  $y_f(t)$ ,  $u_f(t)$ ,  $x_f(t)$ , and their derivatives.

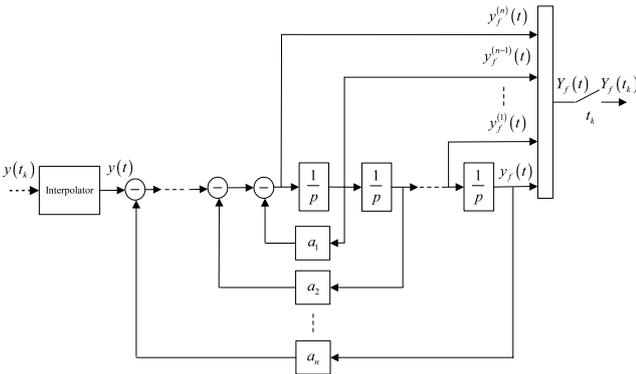


Fig. 5. The filtered output  $y_f(t) = y(t)/A(p)$  and its derivatives (This figure is based on Figure 4.1 in Young et al. (2008)).

$u_f^{(m-1)}(t_k), \dots, u_f(t_k)$ . Note that unlike  $y(t_k)$ , the intersample behavior of  $u(t_i)$  is piece-wise constant owing to the ZOH, so that the interpolation error does not exist.

Besides  $y_f(t_k)$  and  $u_f(t_k)$  and their derivatives, the modified SRIVC method also requires the estimates of  $x_f^{(n-1)}(t_k), x_f^{(n-2)}(t_k), \dots, x_f(t_k)$  in (9). The estimation process involves two filtering operations in steps B1 and B2, which can be implemented in a similar way as the filtering operation for  $y_f(t_k)$  and its derivatives. A special treatment is to introduce the time index  $t_{\Delta}$  along with  $t_i$  and  $t_k$ . By doing so, we can reduce the interpolation error of forming the CT noise-free output  $x(t)$  based on its discrete sample  $x(t_{k,\Delta})$  in step B2 to obtain  $X_f(t_k)$ , and can significantly improve the accuracy of the parameter estimation. Note that steps B1 and B2 cannot be lumped together, because  $x_f^{(n-1)}(t_k), x_f^{(n-2)}(t_k), \dots, x_f^{(1)}(t_k)$  need to be obtained in step B2.

Up until now,  $e(t_k)$  has been assumed to be a white noise, for which the proposed modified SRIVC method yields an efficient estimate of  $\theta$  (to be clarified in Section 4.3). If  $e(t_k)$  is colored, the resulting estimate of  $\theta$  is still unbiased and consistent, owing to the independence nature between  $e(t_k)$  and the instrumental variable  $\hat{\varphi}_f(t_k)$  that is solely correlated to the input  $u(t_i)$ . However, in this case, the parameter estimates are no longer efficient; finding a more effective method is left for future research.

### 4.3. Theoretical analysis

This section analyzes the convergence of the proposed modified SRIVC method, as well as the optimality and asymptotic distribution of the resulting parameter estimates.

Before starting the theoretical analysis, we need to clarify the relationship between the modified SRIVC method and the original one. The modified SRIVC method shares the same fundamental idea with the original SRIVC method, but has two differences: (i) The modified SRIVC method explicitly considers the general

NSNU sampled I/O data  $u(t_i)$  and  $y(t_k)$ , while the original SRIVC method cannot be applied to this special sampling pattern without proper changes. (ii) Besides  $t_i$  and  $t_k$ , the time index  $t_{\Delta}$  is introduced in order to reduce the interpolation error in the filtering operation in step B2 to obtain  $X_f(t_k)$ . By doing so, the accuracy of the parameter estimation can be improved significantly.

Because of the above relationship, the modified SRIVC method inherits from the original SRIVC method the same theoretical results. In terms of convergence, Section 4.1 of Young (2008) or Section 4.4.2 of Young et al. (2008) has already provided some preliminary analysis for the original SRIVC method by noticing the similarity with the iterative least-squares algorithm of Steiglitz and McBride (1965). In fact, the original SRIVC method shares the same spirit with the bootstrap IV-based estimator  $BE_1$  in Söderström and Stoica (1983) (Page 38 therein), the convergence of which has been established under mild conditions by Theorem 4.5 in Söderström and Stoica (1983). Thus, the local convergence of the modified SRIVC method can be established analogously.

In terms of the consistency, because the modified SRIVC method exploits the mechanism of using instrumental variables to eliminate noise effects, the converging estimate  $\theta_j$  in (9) is asymptotically unbiased and consistent, i.e.,  $\theta_j \rightarrow \theta$  as the data length  $N$  goes to infinity. For optimality, if the additive noise is white, the original SRIVC method is known to perform optimally (Young, 2008; Young et al., 2008; Young & Jakeman, 1980), i.e., the parameter estimates converge on the maximum likelihood estimates and are asymptotically efficient. This optimality result holds for the modified SRIVC method, too. In particular, we have the following asymptotic property for the parameter estimates.

**Theorem 1.** *If the CT LTI model  $G(p)$  in (2) is stable and identifiable, the noise  $e(t_k)$  in (3) is independent and identically distributed with zero mean and variance  $\sigma^2$ , and the input  $u(t_k)$  is persistently excited and is independent of  $e(t_k)$ , then the parameter estimate  $\theta_j$  in (9) obtained by the modified SRIVC method is asymptotically normal-distributed (abbreviated as AsN), i.e.,*

$$(\theta_j - \theta) \in \text{AsN}(0, P_{\theta_j}).$$

Here the covariance matrix of the parameter estimation error is

$$P_{\theta_j} = \frac{\sigma^2}{N} \left[ \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N \hat{\varphi}_f(t_k) \hat{\varphi}_f^T(t_k) \right]^{-1}, \quad (12)$$

where  $\hat{\varphi}_f(t_k)$  is the noise-free version of the vector  $\varphi_f(t_k)$  in (6),

$$\hat{\varphi}_f(t_k) = \left[ -x_f^{(n-1)}(t_k), \dots, -x_f(t_k), u_f^{(m)}(t_k), \dots, u_f(t_k) \right]^T.$$

**Proof of Theorem 1.** It is followed in the same manner as the counterpart of the original SRIVC method, see e.g., the proof of Theorem 4.1 in Young et al. (2008).  $\square$

After the iterative estimation of Stage B in Section 4.2 completes, the covariance matrix  $P_{\theta}$  associated with the converged estimate  $\theta_j$  can be estimated as

$$\hat{P}_{\theta_j} = \hat{\sigma}^2 \left[ \sum_{t=1}^N \hat{\varphi}_f(t_k) \hat{\varphi}_f^T(t_k) \right]^{-1}, \quad (13)$$

where  $\hat{\varphi}_f(t_k)$  is the instrumental variable in (10) associated with  $\theta_j$ , and  $\hat{\sigma}^2$  is the variance of the residual estimate,

$$\begin{aligned} \hat{\varepsilon}(t_k) &= y(t_k) - \hat{x}(t_k) \\ &= y(t_k) - \frac{\hat{B}(p, \theta_j)}{\hat{A}(p, \theta_j)} u(t_i). \end{aligned}$$

Thus,  $\hat{P}_{\theta_j}$  in (13) provides the estimate of the uncertainty in the

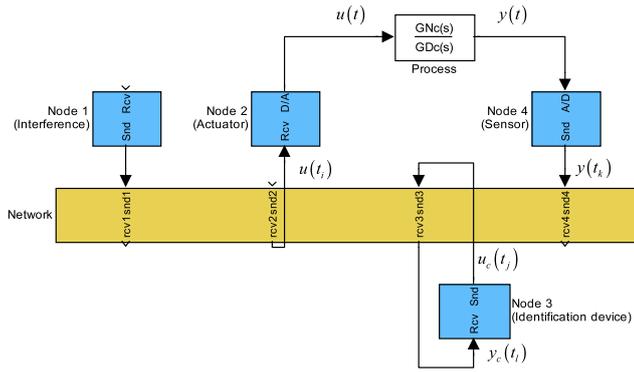


Fig. 6. The networked identification setup using TrueTime.

parameter estimates. The validity of Theorem 1, as well as  $\hat{P}_{\theta_j}$  in (13), is confirmed in the numerical example in Section 5.

### 5. Numerical example

This section presents a numerical example to illustrate the effectiveness of the modified SRIVC method in solving the networked identification problem, and to demonstrate the necessity of considering the effects of communication networks including the network-induced delays and packet dropouts.

**Example 2.** The true CT process is

$$G(p) = \frac{5p + 2}{p^2 + 2.8p + 4}.$$

The excitation input  $u_c(t_j)$  is a PRBS with magnitudes  $(-1, 1)$  and the frequency band  $[0, 0.2]$ . The output additive noise  $e(t_k)$  in (3) is independent of  $u_c(t_j)$ , and the properties of  $e(t_k)$  will be given later.

The Matlab/Simulink simulator TrueTime (Cervin et al., 2003) is used to simulate the effects of the communication network. Specifically, the networked identification configuration in Fig. 1 is implemented in Simulink using TrueTime as depicted in Fig. 6. There are one TrueTime network and four TrueTime kernel blocks (nodes 1–4), namely, the interference, the actuator, the identification device, and the sensor. A time-driven sensor node samples the process output periodically at the sampling instant  $t_k = kh$  for  $h = 0.2$  s, and sends the samples  $y(t_k)$  over the network to the identification device. Once the identification device receives  $y(t_k)$  as  $y_c(t_i)$  at the time instant  $t_i$ , it sends out the excitation input  $u_c(t_j)$  over the network to the actuator node that is subsequently actuated by the received excitation input  $u(t_i)$ . Here the values of the excitation input  $u_c(t_j)$  are generated before the identification experiment, i.e., this is indeed an open-loop identification experiment. All transmitted signals in the network are time-stamped so that the process output  $y(t_k)$  and input  $u(t_i)$  can be recovered from the received output  $y_c(t_i)$  and the sent input  $u_c(t_j)$ , as discussed in Section 3.1. The experiment duration is 40 s so that the length of collected data is about 200.

The TrueTime network takes the CSMA/CD mode (Carrier Sense Multiple Access with Collision Detection) that is commonly used by Ethernet, the data transmission rate is 2000 bits/s and the probability of packet dropouts in the network is  $\alpha = 0.1$ . The simulation also involves an interference node sending tasks over the network to cause disturbing traffic. The interference node is set to use 70% of the network bandwidth. The execution time for each node to read and send samples is very short, around  $5 \times 10^{-5}$  s to  $5 \times 10^{-4}$  s, relative to the sampling period  $h = 0.2$  s of  $y(t_k)$ . As a result, one noticeable effect of communication network lies at

the random time delays for the transmission from  $u_c(t_j)$  to  $u(t_i)$ . For the above network configurations, the averaged increment of  $t_i$  associated with  $u(t_i)$  in one typical experiment is 0.2 with a standard deviation about 0.0258. The recovered  $y(t_k)$  and  $u(t_i)$  in this typical example are shown in Fig. 3. In terms of packet dropouts, about 10% of output samples  $y(t_k)$  are missed. As an example,  $y(t_k)$  in Fig. 3(a) loses 19 data points out of 200.

First, let us investigate the performance of the modified SRIVC method proposed in Section 4.2 under various noise levels. This is achieved by varying the variance  $\sigma_e^2$  of the Gaussian white noise  $e(t_k)$ . The excitation input  $u_c(t_j)$  is fixed to be the same throughout the Monte Carlo simulations. Table 1 lists the means and standard deviations of the estimated parameters obtained by the modified SRIVC method for four noise levels, where 100 Monte Carlo simulations are implemented for each noise level. The parameter  $\lambda$  in (7) is chosen to be  $\lambda = 0.5$ ; in fact, the estimated parameters are almost the same for a quite wide range of  $\lambda$ . The small incremental step associated with  $t_{\Delta}$  is 0.05 s.

In the noise-free case, the modified SRIVC method reaches the true parameters  $\theta = [a_1, a_2, b_0, b_1]^T = [2.8, 4, 5, 2]^T$ . Note that for the noise-free case, i.e.,  $e(t_k) = 0, \forall t_k$ , the variations solely come from the communication network, caused by the random realizations of disturbing traffic generated in the interference node and the stochastic packet dropouts. In principle, some deviations from the true parameters may be encountered in the CT model identification, due to the interpolation errors introduced in the numerical integration operation. However, it seems that the instrumental variable mechanism in the modified SRIVC method is good at removing the interpolation errors. For the noisy cases, the estimates in the last three columns appear to be unbiased and consistent, which clearly demonstrates the effectiveness of the modified SRIVC method.

Table 2 lists the estimated parameters obtained by the modified SRIVC method in one typical simulation for different noise levels. The standard deviations associated with the estimated parameters are estimated based on the covariance matrix estimate in (13); they agree well with the counterparts in Table 1 obtained in 100 Monte Carlo simulations. Therefore, the asymptotic covariance matrix provides a good indication of the accuracy of the estimated parameters, even for a medium sample size  $N \approx 200$ .

Second, let  $e(t_k)$  be a colored noise generated by passing a zero-mean Gaussian white noise  $a(t_k)$  with the variance  $\sigma_a^2$  via a filter,

$$e(t_k) = \frac{1 + 0.2q^{-1}}{1 - 0.4q^{-1}} a(t_k).$$

Table 3 lists the means and standard deviations of the estimated parameters obtained in 100 Monte Carlo simulations for three noise levels  $\sigma_a^2 = 0.5 \times 10^{-4}$ ,  $\sigma_a^2 = 0.5 \times 10^{-3}$  and  $\sigma_a^2 = 0.5 \times 10^{-2}$ . These noise levels are selected to make the energies of  $e(t_k)$  close to those in Table 1 where  $e(t_k)$  is white noise. For colored noises, the estimates in Table 3 appear to be unbiased and consistent. This is consistent with the analysis in Section 4.3. However, these estimated parameters may not be efficient. We are currently working on methods that can deal with colored noises more effectively.

Finally, to demonstrate the necessity of considering the effects of communication networks, the original SRIVC method is exploited. It is implemented by the function 'srivc' in the CONTSID toolbox (Garnier et al., 2008) and can only deal with synchronized input and output data, being uniformly or non-uniformly sampled. Hence, the configuration of the networked identification experiment has to be simplified. Two simplifications are made: (i) The packet dropouts are not considered, i.e., the probability of packet dropouts in the communication network is set to  $\alpha = 0$ , while the rest of the configuration is preserved to be the same. As a result, the effects of the communication network reduce to

**Table 1**  
Estimated parameters for white noises with different noise levels in 100 simulations.

	$\sigma_e^2 = 0$	$\sigma_e^2 = 1 \times 10^{-4}$	$\sigma_e^2 = 1 \times 10^{-3}$	$\sigma_e^2 = 1 \times 10^{-2}$
$a_1 = 2.8$	$2.8 \pm 0.0$	$2.7994 \pm 0.0035$	$2.7976 \pm 0.0102$	$2.7933 \pm 0.0446$
$a_2 = 4$	$4.0 \pm 0.0$	$3.9966 \pm 0.0066$	$3.9894 \pm 0.0207$	$3.9857 \pm 0.0626$
$b_0 = 5$	$5.0 \pm 0.0$	$4.9991 \pm 0.0061$	$4.9967 \pm 0.0175$	$4.9894 \pm 0.0713$
$b_1 = 2$	$2.0 \pm 0.0$	$1.9963 \pm 0.0081$	$1.9887 \pm 0.0260$	$1.9869 \pm 0.0755$

**Table 2**  
Estimated parameters for white noises with different noise levels in one simulation.

	$\sigma_e^2 = 1 \times 10^{-4}$	$\sigma_e^2 = 1 \times 10^{-3}$	$\sigma_e^2 = 1 \times 10^{-2}$
$a_1 = 2.8$	$2.8017 \pm 0.0047$	$2.7903 \pm 0.0142$	$2.8130 \pm 0.0497$
$a_2 = 4$	$3.9989 \pm 0.0060$	$3.9803 \pm 0.0186$	$3.9575 \pm 0.0640$
$b_0 = 5$	$5.0025 \pm 0.0075$	$4.9992 \pm 0.0228$	$4.9325 \pm 0.0775$
$b_1 = 2$	$1.9958 \pm 0.0074$	$1.9749 \pm 0.0227$	$1.9670 \pm 0.0758$

**Table 3**  
Estimated parameters for color noises with different noise levels in 100 simulations.

	$\sigma_a^2 = 0.5 \times 10^{-4}$	$\sigma_a^2 = 0.5 \times 10^{-3}$	$\sigma_a^2 = 0.5 \times 10^{-2}$
$a_1 = 2.8$	$2.8000 \pm 0.0051$	$2.7998 \pm 0.0165$	$2.8007 \pm 0.0501$
$a_2 = 4$	$3.9955 \pm 0.0080$	$3.9862 \pm 0.0252$	$3.9564 \pm 0.0800$
$b_0 = 5$	$4.9999 \pm 0.0086$	$4.9997 \pm 0.0275$	$5.0013 \pm 0.0851$
$b_1 = 2$	$1.9951 \pm 0.0104$	$1.9849 \pm 0.0327$	$1.9520 \pm 0.1032$

**Table 4**  
Estimated parameters obtained by two methods for 100 noise-free simulations.

$e(t_k) = 0$	Modified SRIVC	SRIVC in CONTSID
$a_1 = 2.8$	$2.8 \pm 0.0$	$3.7848 \pm 0.0705$
$a_2 = 4$	$4.0 \pm 0.0$	$5.0034 \pm 0.0666$
$b_0 = 5$	$5.0 \pm 0.0$	$6.5207 \pm 0.0989$
$b_1 = 2$	$2.0 \pm 0.0$	$1.8659 \pm 0.0393$

the network-induced delays only. (ii) To synchronize with  $y(t_k)$ , an evenly-spaced input  $u(t_k)$  is sampled from the CT input  $u(t)$  reconstructed from  $u(t_i)$  and its ZOH intersample behavior. That is, the original SRIVC method ignores the possible variations among the sample point  $u(t_k)$ 's, and estimates the parameters based on  $u(t_k)$  and  $y(t_k)$ .

We compare the estimates of the modified SRIVC method and the original one for the noise-free case only, i.e.,  $e(t_k) = 0$ ,  $\forall t_k$ . The two methods take the same parameter  $\lambda = 0.5$  in (7). Table 4 lists the means and standard deviations of the estimated parameters obtained in 100 Monte Carlo simulations. Note that the variations in the 100 simulations solely come from the random realizations of disturbing traffic generated in the interference node. The modified SRIVC method reaches the true parameters, while the original SRIVC method in the CONTSID toolbox yields unsatisfactory estimates. This clearly illustrates the necessity of considering the effects of network-induced delays.

## 6. Conclusion

We studied the off-line identification problem for open-loop stable LTI processes working in the networked environment. Under the effects of random network-induced delays and packet dropouts, it was found out in Sections 2 and 3 that the running modes of the actuator lead to identification problems with very different complexities. In particular, the networked identification problem was formulated as identifying CT LTI models based on the general NSNU sampled data, under the configuration of having event-driven actuators subject to random network-induced delays and packet dropouts. A modified SRIVC method was proposed in Section 4 to solve this particular networked identification problem. The proposed method was validated in a networked identification experiment based on the Matlab/Simulink simulator TrueTime in Section 5.

There remain many interesting problems to be pursued in future studies. First of all, it was mentioned at the end of Section 4.2 that the modified SRIVC method does not provide efficient parameter estimates for colored noises  $e(t_k)$ . It would be necessary to develop another method to handle the colored noises more effectively. Second, Section 3.1 discusses an exceptional scenario of being unable to recover the input  $u(t_i)$  due to packet dropouts, whose effects on the modified SRIVC method certainly deserve a further investigation. Finally, an important task is to implement the networked identification experiment not only in the simulation of TrueTime, but also in the real-world applications in order to close the gap between theory and practice.

## Acknowledgments

The authors would like to thank the anonymous reviewers and the Associate Editor for their constructive comments and helpful suggestions. The first author is also grateful to Prof. P.C. Young for kindly providing references on the refined instrumental variable method.

## References

- Ahmed, S., Huang, B., & Shah, S. (2008). Process parameter and delay estimation from non-uniformly sampled data. In H. Garnier, & L. Wang (Eds.), *Identification of continuous-time models from sampled data*. London: Springer-Verlag.
- Antsaklis, P., & Baillieul, J. (2004). Special issue on networked control systems. *IEEE Transactions on Automatic Control*, 49(9), 1421–1597.
- Antsaklis, P., & Baillieul, J. (2007). Special issue on technology of networked control systems. *Proceedings of IEEE*, 95(1), 5–312.
- Branicky, M., Phillips, S., & Zhang, W. (2000). Stability of networked control systems: Explicit analysis of delay. In *Proc. of the 2000 American control conf., Chicago Illinois* (pp. 2352–2357).
- Brsic, D., Petrovic, I., & Peric, N. (2000). Atm available bit rate congestion control with on-line network model identification. In *Proc. of the 9th int. conf. on electronics, circuits and systems* (pp. 1243–1246).
- Bushnell, L. (2001). Special issue on networks and control. *IEEE Control Systems Magazine*, 21(1), 22–99.
- Cervin, A., Henriksson, D., Lincoln, B., Eker, J., & Arzen, K. (2003). How does control timing affect performance? Analysis and simulation of timing using Jitterbug and TrueTime. *IEEE Control Systems Magazine*, 23(3), 16–30.
- Chen, T., & Francis, B. (1995). *Optimal sampled-data control systems*. London: Springer.
- Chow, M. (2004). Special section on distributed networked-based control systems and applications. *IEEE Transactions on Industrial Electronics*, 51(6), 1126–1179.
- Chow, M., & Tipsuwan, Y. (2001). Network-based control systems: A tutorial. In *Proc. of the 27th annual conf. of the IEEE indust. electron. society* (pp. 1593–1602).
- Eidson, J., & Cole, W. (1998). Ethernet rules closed-loop system. *InTech*, 39–42.
- Faizulkhakov, Y. (2007). Time synchronization methods for wireless sensor networks: A survey. *Programming and Computer Software*, 33(4), 214–226.
- Fei, M., Du, D., & Li, K. (2008). A fast model identification method for networked control system. *Applied Mathematics and Computation*, 205(2), 658–667.
- Forouzan, B. (2007). *Data communications and networking* (4th ed.). McGraw-Hill.
- Garnier, H., Gilson, M., Bastogne, T., & Mensler, M. (2008). The contsid toolbox: A software support for data-based continuous-time modelling. In H. Garnier, & L. Wang (Eds.), *Identification of continuous-time models from sampled data*. London: Springer-Verlag.
- Garnier, H., & Wang, L. (2008). *Identification of continuous-time models from sampled data*. London: Springer-Verlag.
- Hespanha, J., Naghshtabrizi, P., & Xu, Y. (2007). A survey of recent results in networked control systems. *Processings of IEEE*, 95(1), 138–162.
- Hokayem, P., & Abdallah, C. (2004). Inherent issues in networked control systems: A survey. In *Proc. of the 2004 American control conf. Boston, Massachusetts* (pp. 4897–4902).
- Huselstein, E., & Garnier, H. (2002). An approach to continuous-time model identification from non-uniformly sampled data. In *Proc. of the 41st IEEE conf. on decision and control, Las Vegas, Nevada USA* (pp. 622–623).
- Ishikawa, K., & Mita, A. (2008). Time synchronization of a wired sensor network for structural health monitoring. *Smart Materials and Structures*, 17, 015016.

- Kaplan, G. (2001). Ethernet's winning ways. *IEEE Spectrum*, 38(1), 113–115.
- Larsson, E., Mossberg, M., & Soderstrom, T. (2007). Identification of continuous-time arx models from irregularly sampled data. *IEEE Transactions on Automatic Control*, 52(3), 417–427.
- Larsson, E., Mossberg, M., & Soderstrom, T. (2008). Estimation of continuous-time stochastic system parameters. In H. Garnier, & L. Wang (Eds.), *Identification of continuous-time models from sampled data*. London: Springer-Verlag.
- Lian, F., Moyne, J. R., & Tilbury, D. M. (2002). Network design consideration for distributed control systems. *IEEE Transactions on Control Systems Technology*, 10(2), 297–307.
- Liu, G., Chai, S., Mu, J., & Rees, D. (2008). Networked predictive control of systems with random delay in signal transmission channels. *International Journal of Systems Science*, 39(11), 1055–1064.
- Moyne, J., & Tilbury, D. (2007). The emergence of industrial control networks for manufacturing control, diagnostics, and safety data. *Proceedings of IEEE*, 95(1), 29–47.
- Nilsson, J. (1998). Real-time control systems with delays. Ph.D dissertation, Lund Inst. Technol., Lund, Sweden.
- Söderström, T., & Stoica, P. (1983). *Instrumental variable methods for system identification*. Berlin: Springer-Verlag.
- Srinivasagupta, D., Schattler, H., & Joseph, B. (2004). Time-stamped model predictive control: An algorithm for control of processes with random delays. *Computers and Chemical Engineering*, 28(8), 1337–1346.
- Steiglitz, K., & McBride, L. (1965). A technique for the identification of linear systems. *IEEE Transactions on Automatic Control*, 10(4), 461–464.
- Su, W. (2008). Time-synchronization challenges and techniques. In Y. Li, M. Thai, & W. Wu (Eds.), *Wireless sensor networks and applications*. London: Springer-Verlag.
- Tang, P., & de Silva, C. (2006). Compensation for transmission delays in an ethernet-based control network using variable-horizon predictive control. *IEEE Transactions on Control Systems Technology*, 14(4), 707–718.
- Thompson, H. (2004). Wireless and internet communications technologies for monitoring and control. *Control Engineering Practice*, 12(6), 781–791.
- Tipuwan, Y., & Chow, M. (2003). Control methodologies in networked control systems. *Control Engineering Practice*, 11(10), 1099–1111.
- Tsang, K., & Billings, S. (1995). Identification of systems from non-uniformly sampled data. *International Journal of Systems Science*, 26(10), 1823–1837.
- Wang, F., Liu, D., Yang, S., & Li, L. (2007). Special section on networking, sensing, and control for networked control systems: Architectures, algorithms, and applications. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, 37(2).
- Yang, S., Chen, X., & Alty, J. (2003). Design issues and implementation of internet-based process control systems. *Control Engineering Practice*, 11(6), 709–720.
- Yang, T. (2006). Networked control system: A brief survey. *IEE Proceedings—Control Theory and Applications*, 153(4), 403–412.
- Young, P. (2002). Optimal IV identification and estimation of continuous-time TF models. In *Proc. of the 15th triennial IFAC world congress on automatic control, Barcelona, Spain*, 25(5), 337–358.
- Young, P. (2008). The refined instrumental variable method: Unified estimation of discrete and continuous-time transfer function models. *Journal European des Systemes Automatisés*, 42(2–3), 149–179.
- Young, P., Garnier, H., & Gilson, M. (2008). Refined instrumental variable identification of continuous-time hybrid box jenkins models. In H. Garnier, & L. Wang (Eds.), *Identification of continuous-time models from sampled data*. London: Springer-Verlag.
- Young, P., & Jakeman, A. (1980). Refined instrumental variable methods of time-series analysis: Part III, extensions. *International Journal of Control*, 31(4), 741–764.
- Zhang, W., Branicky, M., & Phillips, S. (2001). Stability of networked control systems. *IEEE Control Systems Magazine*, 21(1), 84–99.
- Zhu, Y. (2001). *Multivariable system identification for processes control*. Oxford: Elsevier.



**Jiandong Wang** received the B.E. degree in automatic control from Beijing University of Chemical Technology, Beijing, China, in 1997, and the M.Sc and Ph.D. degrees from the University of Alberta, Edmonton, AB, Canada, in 2003 and 2007, respectively, all in Electrical and Computer Engineering.

From 1997 to 2001, he was a Control Engineer with the Beijing Tsinghua Energy Simulation Company, Beijing, China. From February 2006 to August 2006, he was a Visiting Scholar at the Department of System Design Engineering at the Keio University, Japan. Since December 2006, he has been an Assistant Professor with the Department of Industrial Engineering and Management in the College of Engineering at the Peking University, China. His current research interests include system identification, networked systems, signal processing, process monitoring and management, and their applications to industrial problems.

Dr. Wang received Peking University P&G Fellowship in 2008 and a Fellowship from the Japan Society for the Promotion of Science in 2006. He was a recipient of the Izzak Walton Killam Memorial Scholarship, the Alberta Ingenuity Ph.D. Studentship, and the iCore Graduate Student Scholarship at the University of Alberta.



**Wei Xing Zheng** received the Ph.D. degree in Electrical Engineering from Southeast University, China in 1989. He has held various faculty/research/visiting positions at Southeast University, China, Imperial College of Science, Technology and Medicine, UK, University of Western Australia, Curtin University of Technology, Australia, Munich University of Technology, Germany, University of Virginia, USA, and University of California-Davis, USA. Currently he holds the rank of Full Professor at University of Western Sydney, Australia. Dr. Zheng has served as an Associate Editor for four flagship journals: *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications* (2002–2004), *IEEE Transactions on Automatic Control* (2004–2007), *IEEE Signal Processing Letters* (from 2007 onwards), and *IEEE Transactions on Circuits and Systems-II: Express Briefs* (from 2008 onwards). He is also an Associate Editor of *IEEE Control Systems Society's Conference Editorial Board* (from 2000 onwards). Currently, he serves as a Guest Associate Editor of Special Issue on Blind Signal Processing and Its Applications for *IEEE Transactions on Circuits and Systems-I: Regular Papers*. He has also served as the Chair of *IEEE Circuits and Systems Society's Technical Committee on Neural Systems and Applications* and as the Chair-Elect of *IEEE Circuits and Systems Society's Technical Committee on Blind Signal Processing*.



**Tongwen Chen** is presently a Professor of Electrical and Computer Engineering at the University of Alberta, Edmonton, Canada. He received the B.Eng. degree in Automation and Instrumentation from Tsinghua University (Beijing) in 1984, and the M.A.Sc. and Ph.D. degrees in Electrical Engineering from the University of Toronto in 1988 and 1991, respectively.

His research interests include computer and network based control systems, and their applications to the process and power industries. Dr. Chen is an IEEE Fellow and a Fellow of the Engineering Institute of Canada. He has served as an Associate Editor for several international journals, including *IEEE Transactions on Automatic Control*, *Automatica*, and *Systems and Control Letters*. He is a registered Professional Engineer in Alberta, Canada.