A GLOBAL NONLINEAR INSTRUMENTAL VARIABLE METHOD FOR IDENTIFICATION OF CONTINUOUS-TIME SYSTEMS WITH UNKNOWN TIME DELAYS

Zi-Jiang Yang * **Hideto Iemura** * **Shunshoku Kanae** *
Kiyoshi Wada *

* *Department of Electrical and Electronic Systems Engineering,*
Kyushu University, Hakozaki, Fukuoka 812-8581, Japan
TEL: (+81)92-642-3904, FAX: (+81)92-642-3939
E-mail: yoh@ees.kyushu-u.ac.jp

Abstract: This paper considers the identification problem of continuous-time systems with unknown time delays from sampled input-output data. An iterative global separable nonlinear least-squares (GSEPNLS) method which estimates the time delays and transfer function parameters separably is derived, by using stochastic global-optimization technique to avoid convergence to a local minimum. Futhermore, the GSEPNLS method is modified to a novel global separable nonlinear instrumental variable (GSEPNIV) method to yield consistent estimates if the algorithm converges to the global minimum. Simulation results show that the proposed method works quite well.

Keywords: Identification, continuous-time system, time-delay, instrumental variable method, stochastic approximation with convolution smoothing.

1. INTRODUCTION

In the discrete-time models with time delay, the sampling period is usually required to be very small such that the time delay is an integral multiple of the sampling period, whereas if the sampling period is too small, the identification problem may become numerically difficult. Moreover, the parameters in the discrete-time models usually do not correspond to the physical parameters. Therefore, the importance of identification based on the continuous-time model has been recognized in recent years.

There have been some typical approaches to identification of continuous-time models with unknown delay. One approach is based on the nonlinear estimation method like nonlinear least-squares (LS) method that searches for the minimum by using a gradient-following technique (Gawthrop *et al.*, 1989; Tuch *et al*, 1994). A major problem of nonlinear estimation methods is that the estimates by such methods may be stuck at local minima. Therefore, the results may be sensitive to the initial values. The problem is much more difficult for multi-input single-output (MISO) systems with multiple time delays, In our previous study, we proposed an iterative GSEPNLS method by using of stochastic global-optimization techniques (Iemura *et al.*, 2004). In this further work, we propose a novel GSEPNIV method that yields consistent estimates in the presence of high measurement noise if the algorithm converges to the global minimum.

2. STATEMENT OF THE PROBLEM

Consider the strictly stable MISO continuous-time system with unknown time delays:

$$\sum_{i=0}^{n} a_i p^{n-i} x(t) = \sum_{j=1}^{r}\sum_{i=1}^{m_j} b_{ji} p^{m_j-i} u_j(t-\tau_j) \quad (a_0=1, b_{j1}\neq 0) \tag{1}$$

where p is a differential operator, $u_j(t)$ is the jth input with time delay τ_j, $x(t)$ is the real input and output. n and m_j are assumed to be known ($n \geq m_j$).

It is assumed that a zero-order hold is utilized such that

$$u_j(t) = \bar{u}_j(k) \qquad (k-1)T \leq t < kT \tag{2}$$

where T is the sampling period.

Practically the discrete-time measurement of the output variable is corrupted by a stochastic measurement noise:.

$$y(k) = x(k) + v(k) \tag{3}$$

where $y(k), x(k), v(k)$ denote $y(kT), x(kT), v(kT)$ respectively. It is assumed that each zero-order hold input $\bar{u}_j$ is a quasi-stationary deterministic or random signal and the noise $v(k)$ is a quasi-stationary zero-mean random signal uncorrelated with each input such that

$$\lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} \bar{u}_j(k) v(k) = 0$$

3. APPROXIMATED DISCRETE-TIME ESTIMATION MODEL

To avoid direct signal derivatives, we introduce a low-pass pre-filter $Q(p)$ as

$$Q(p) = \frac{1}{(\alpha p + 1)^n} \tag{4}$$

where α is the time constant which determines the pass-band of $Q(p)$.

Multipling both sides of (1) by $Q(p)$ and using the bilinear transformation based on the block-pulse functions, we can obtain the following approximated discrete-time estimation model of the original system (Yang *et al.*, 1997) :

$$\xi_{0\bar{y}}(k) + \sum_{i=1}^{n} a_i \xi_{i\bar{y}}(k) = \sum_{j=1}^{r}\sum_{i=1}^{m_j} b_{ji}\xi_{(n-m_j+i)\bar{u}_j}(k-\tilde{\tau}_j) + r(k) \tag{5}$$

where

$$r(k) = \sum_{i=0}^{n} a_i \xi_{i\bar{v}}(k) \tag{6}$$

and

$$\begin{aligned} \xi_{i\bar{u}_j}(k) &= F_i(z^{-1})\bar{u}_j(k) \\ \xi_{i\bar{y}}(k) &= F_i(z^{-1})\bar{y}(k) \\ \xi_{i\bar{v}}(k) &= F_i(z^{-1})\bar{v}(k) \end{aligned} \tag{7}$$

$$F_i(z^{-1}) = \frac{\left(\frac{T}{2}\right)^i (1+z^{-1})^i (1-z^{-1})^{n-i}}{\left[\alpha(1-z^{-1}) + \frac{T}{2}(1+z^{-1})\right]^n} \tag{8}$$

where z^{-1} is the backward shift operator, $\bar{v}(k) = (1+z^{-1})v(k)/2$ and $\bar{y}(k) = (1+z^{-1})y(k)/2$.

$\tilde{\tau}_j$ in (5) is given by

$$\tilde{\tau}_j = \tau_j/T = l + \Delta/T \tag{9}$$

where $0 \leq \Delta < T$ and l is a non-negative integer.

Remark 1: Our approximated discrete-time estimation model does not require that the time-delay τ_j is an integral multiple of the sampling period. In the case of $\Delta \neq 0$, we can get $\xi_{(n-m_j+i)\bar{u}_j}(k-\tilde{\tau}_j)$ by linear interpolation between $\xi_{(n-m_j+i)\bar{u}_j}(k-l)$ and $\xi_{(n-m_j+i)\bar{u}_j}(k-l-1)$.

(5) can be written in vector form:

$$\begin{aligned} \xi_{0\bar{y}}(k) &= \boldsymbol{\varphi}^T(k,\boldsymbol{\tau})\boldsymbol{\theta} + r(k) \\ \boldsymbol{\varphi}^T(k,\boldsymbol{\tau}) &= [-\boldsymbol{\varphi}_{\bar{y}}^T(k), \boldsymbol{\varphi}_{\bar{u}_1}^T(k-\tilde{\tau}_1), \cdots, \boldsymbol{\varphi}_{\bar{u}_r}^T(k-\tilde{\tau}_r)] \\ \boldsymbol{\varphi}_{\bar{y}}^T(k) &= [\xi_{1\bar{y}}(k), \cdots, \xi_{n\bar{y}}(k)] \\ \boldsymbol{\varphi}_{\bar{u}_j}^T(k-\tilde{\tau}_j) &= [\xi_{(n-m_j+1)\bar{u}_j}(k-\tilde{\tau}_j), \cdots, \xi_{n\bar{u}_j}(k-\tilde{\tau}_j)] \\ \boldsymbol{\theta}^T &= [\boldsymbol{a}^T, \boldsymbol{b}_1^T, \cdots, \boldsymbol{b}_r^T] \\ \boldsymbol{\tau}^T &= [\tau_1, \cdots, \tau_r] \\ \boldsymbol{a}^T &= [a_1, \cdots, a_n] \\ \boldsymbol{b}_j^T &= [b_{j1}, \cdots, b_{jm_j}] \end{aligned} \tag{10}$$

4. SEPNLS METHOD

Given a fixed set of filtered input-output data $\{\xi_{0\bar{y}}(k), \boldsymbol{\varphi}_{\bar{y}}^T(k), \boldsymbol{\varphi}_{\bar{u}_1}^T(k), \cdots, \boldsymbol{\varphi}_{\bar{u}_r}^T(k)\}_{k_s+1}^{N}$, the off-line parameter estimates are defined as the minimizing arguments of the following LS criterion

$$\begin{aligned} V_N(\boldsymbol{\theta},\boldsymbol{\tau}) &= \frac{1}{N-k_s}\sum_{k=k_s+1}^{N} \frac{1}{2}\epsilon^2(k,\boldsymbol{\theta},\boldsymbol{\tau}) \\ \epsilon(k,\boldsymbol{\theta},\boldsymbol{\tau}) &= \xi_{0\bar{y}}(k) - \boldsymbol{\varphi}^T(k,\boldsymbol{\tau})\boldsymbol{\theta} \end{aligned} \tag{11}$$

such that

$$\left[\widehat{\boldsymbol{\theta}}_N^T, \widehat{\boldsymbol{\tau}}_N^T\right]^T = \arg\min_{\boldsymbol{\theta},\boldsymbol{\tau}} V_N(\boldsymbol{\theta},\boldsymbol{\tau}) \tag{12}$$

The SEPNLS method estimates the time delay vector $\boldsymbol{\tau}$ and the linear parameter vector $\boldsymbol{\theta}$ in a separable manner. When the time delays are known, the linear parameters can be estimated by the linear LS method as

$$\begin{aligned} \widehat{\boldsymbol{\theta}}_N(\boldsymbol{\tau}) &= \boldsymbol{R}^{-1}(N,\boldsymbol{\tau})\boldsymbol{f}(N,\boldsymbol{\tau}) \\ \boldsymbol{R}(N,\boldsymbol{\tau}) &= \frac{1}{N-k_s}\sum_{k=k_s+1}^{N} \boldsymbol{\varphi}(k,\boldsymbol{\tau})\boldsymbol{\varphi}^T(k,\boldsymbol{\tau}) \\ \boldsymbol{f}(N,\boldsymbol{\tau}) &= \frac{1}{N-k_s}\sum_{k=k_s+1}^{N} \boldsymbol{\varphi}(k,\boldsymbol{\tau})\xi_{0\bar{y}}(k) \end{aligned} \tag{13}$$

Then the LS criterion $V_N(\boldsymbol{\theta},\boldsymbol{\tau})$ becomes

$$\check{V}_N(\boldsymbol{\tau}) = \frac{1}{N-k_s}\sum_{k=k_s+1}^{N} \frac{1}{2}\check{\varepsilon}^2(k,\boldsymbol{\tau}) \tag{14}$$

where

$$\breve{\varepsilon}(k,\boldsymbol{\tau}) = \xi_{0\bar{y}}(k) - \boldsymbol{\varphi}^T(k,\boldsymbol{\tau})\boldsymbol{R}^{-1}(N,\boldsymbol{\tau})\boldsymbol{f}(N,\boldsymbol{\tau}) \quad (15)$$

The time delay vector $\boldsymbol{\tau}$ and the linear parameter vector $\boldsymbol{\theta}$ can be estimated separably according to the following theorem. See Ruhe & Wedin (1980), Ngia (2001) for proof and more detailed explanations.

Theorem 1. Let $\boldsymbol{\theta} = \widehat{\boldsymbol{\theta}}_N(\boldsymbol{\tau}) = \boldsymbol{R}^{-1}(N,\boldsymbol{\tau})\boldsymbol{f}(N,\boldsymbol{\tau})$ *denotes one solution of the LS criterion (11). Then*

$$\left[\widehat{\boldsymbol{\theta}}_N^T, \widehat{\boldsymbol{\tau}}_N^T\right]^T = \underset{\boldsymbol{\theta},\boldsymbol{\tau}}{\operatorname{argmin}}\, V_N(\boldsymbol{\theta},\boldsymbol{\tau}) = \underset{\boldsymbol{\tau}}{\operatorname{argmin}}\, \breve{V}_N(\boldsymbol{\tau}) \quad (16)$$

Therefore, the estimate of time delays can be obtained as

$$\widehat{\boldsymbol{\tau}}_N = \underset{\boldsymbol{\tau}}{\operatorname{argmin}}\, \breve{V}_N(\boldsymbol{\tau}) \quad (17)$$

through the following iterative search algorithm.

$$\widehat{\boldsymbol{\tau}}_N^{(j+1)} = \widehat{\boldsymbol{\tau}}_N^{(j)} - \mu^{(j)} \left[\breve{\boldsymbol{R}}_N\left(\widehat{\boldsymbol{\tau}}_N^{(j)}\right)\right]^{-1} \breve{\boldsymbol{V}}'_N\left(\widehat{\boldsymbol{\tau}}_N^{(j)}\right) \quad (18)$$

where $\mu^{(j)}$ is the step-size which assures that $\breve{V}_N(\boldsymbol{\tau})$ decreases and that $\widehat{\boldsymbol{\tau}}_N$ stays in a preassigned interval, i.e., for $i = 1, \cdots, r$,

$$\widehat{\tau}_{Ni}^{(j+1)} \in \Omega_{\tau_i} = \left\{\widehat{\tau}_{Ni}^{(j+1)} \middle| 0 \le \widehat{\tau}_{Ni}^{(j+1)} \le \bar{\tau}_i\right\}$$

$\breve{\boldsymbol{V}}'_N(\boldsymbol{\tau})$ and $\breve{\boldsymbol{R}}_N(\boldsymbol{\tau})$ are respectively the gradient and the estimate of the Hessian of the LS criterion:

$$\begin{aligned} \breve{\boldsymbol{V}}'_N(\boldsymbol{\tau}) &= -\frac{1}{N-k_s}\sum_{k=k_s+1}^{N} \boldsymbol{\psi}(k,\boldsymbol{\tau})\breve{\varepsilon}(k,\boldsymbol{\tau}) \\ \breve{\boldsymbol{R}}_N(\boldsymbol{\tau}) &= \frac{1}{N-k_s}\sum_{k=k_s+1}^{N} \boldsymbol{\psi}(k,\boldsymbol{\tau})\boldsymbol{\psi}^T(k,\boldsymbol{\tau}) \end{aligned} \quad (19)$$

$\boldsymbol{\psi}(k,\boldsymbol{\tau}) = [\psi_1(k,\boldsymbol{\tau}), \cdots, \psi_r(k,\boldsymbol{\tau})]^T$ can be obtained through tedious but straightforward calculations as follows, for $j = 1, \cdots, r$.

$$\begin{aligned} &\boldsymbol{\psi}_j(k,\boldsymbol{\tau}) = -\frac{\partial\breve{\varepsilon}(k,\boldsymbol{\tau})}{\partial\tau_j} \\ &= \boldsymbol{\varphi}_{\tau_j}^T(k,\boldsymbol{\tau})\boldsymbol{R}^{-1}(N,\boldsymbol{\tau})\boldsymbol{f}(N,\boldsymbol{\tau}) + \boldsymbol{\varphi}^T(k,\boldsymbol{\tau})\boldsymbol{R}^{-1}(N,\boldsymbol{\tau})\boldsymbol{f}_{\tau_j}(N,\boldsymbol{\tau}) \\ &-\boldsymbol{\varphi}^T(k,\boldsymbol{\tau})\boldsymbol{R}^{-1}(N,\boldsymbol{\tau})\left[\boldsymbol{R}_{\tau_j}(N,\boldsymbol{\tau}) + \boldsymbol{R}_{\tau_j}^T(N,\boldsymbol{\tau})\right]\boldsymbol{R}^{-1}(N,\boldsymbol{\tau})\boldsymbol{f}(N,\boldsymbol{\tau}) \end{aligned} \quad (20)$$

where

$$\begin{aligned} \boldsymbol{R}_{\tau_j}(N,\boldsymbol{\tau}) &= \frac{1}{N-k_s}\sum_{k=k_s+1}^{N} \boldsymbol{\varphi}_{\tau_j}(k,\boldsymbol{\tau})\boldsymbol{\varphi}^T(k,\boldsymbol{\tau}) \\ \boldsymbol{f}_{\tau_j}(N,\boldsymbol{\tau}) &= \frac{1}{N-k_s}\sum_{k=k_s+1}^{N} \boldsymbol{\varphi}_{\tau_j}(k,\boldsymbol{\tau})\xi_{0\bar{y}}(k) \\ \boldsymbol{\varphi}_{\tau_j}(k,\boldsymbol{\tau}) &= \frac{\partial\boldsymbol{\varphi}(k,\boldsymbol{\tau})}{\partial\tau_j} \\ &= \Big[\boldsymbol{0}_{1\times n}, \boldsymbol{0}_{1\times m_1}, \cdots, \boldsymbol{0}_{1\times m_{j-1}}, \\ &\quad \boldsymbol{\varphi}_{\tau_j}^T(k-\tilde{\tau}_j), \boldsymbol{0}_{1\times m_{j+1}}, \cdots, \boldsymbol{0}_{1\times m_r}\Big]^T \\ \boldsymbol{\varphi}_{\tau_j}^T(k-\tilde{\tau}_j) &= [-\xi_{(n-m_j)\bar{u}_j}(k-\tilde{\tau}_j), -\xi_{(n-m_j+1)\bar{u}_j}(k-\tilde{\tau}_j), \\ &\quad \cdots, -\xi_{(n-1)\bar{u}_j}(k-\tilde{\tau}_j)] \end{aligned} \quad (21)$$

Finally, by substituting $\widehat{\boldsymbol{\tau}}_N$ into (13), the linear parameter vector $\boldsymbol{\theta}$ can be estimated by the linear LS method (13).

5. GSEPNLS METHOD

Stochastic approximation with convolution smoothing (SAS) is a global-optimization algorithm for minimizing a nonconvex function

$$\underset{\boldsymbol{\tau}\in\Omega_r}{\operatorname{argmin}}\, \breve{V}_N(\boldsymbol{\tau}) \quad (22)$$

The smoothing process represents the convolution of $\breve{V}_N(\boldsymbol{\tau})$ with a smoothing function $\check{h}(\boldsymbol{\eta},\beta)$, where $\boldsymbol{\eta} \in R^r$ is a random vector used to perturb $\boldsymbol{\tau}$, and β controls the degree of smoothing. This smoothed functional described in Rubinstein (1981) is given by

$$\begin{aligned} \check{\breve{V}}_N(\boldsymbol{\tau},\beta) &= \int_{-\infty}^{\infty} \check{h}(\boldsymbol{\eta},\beta)\breve{V}_N(\boldsymbol{\tau}-\boldsymbol{\eta})d\boldsymbol{\eta} \\ &= \int_{-\infty}^{\infty} \check{h}(\boldsymbol{\tau}-\boldsymbol{\eta},\beta)\breve{V}_N(\boldsymbol{\eta})d\boldsymbol{\eta} \end{aligned} \quad (23)$$

which represents an averaged version of $\breve{V}_N(\boldsymbol{\tau})$ weighted by $\check{h}(\cdot,\beta)$. To yield a properly-smoothed functional $\check{\breve{V}}_N(\boldsymbol{\tau},\beta)$, the kernel functional $\check{h}(\boldsymbol{\eta},\beta)$ must have the following properties (Rubinstein, 1981):

(1) $\check{h}(\boldsymbol{\eta},\beta) = (1/\beta^r)h(\boldsymbol{\eta}/\beta)$ is piecewise differenciable with respect to β;
(2) $\lim_{\beta\to 0}\check{h}(\boldsymbol{\eta},\beta) = \delta(\boldsymbol{\eta})$; ($\delta(\boldsymbol{\eta})$ is the Dirac delta function);
(3) $\lim_{\beta\to 0}\check{\breve{V}}_N(\boldsymbol{\tau},\beta) = \breve{V}_N(\boldsymbol{\tau})$;
(4) $\check{h}(\boldsymbol{\eta},\beta)$ is a probability density function (pdf).

One of the possible choices for $h(\boldsymbol{\eta})$ is a Gaussian pdf (Rubinstein, 1981), which leads to the following expression for $\check{h}(\boldsymbol{\eta},\beta)$:

$$\check{h}(\boldsymbol{\eta},\beta) = \frac{1}{(2\pi)^{(r/2)}\beta^r}\exp\left[-\frac{1}{2}\sum_{i=1}^{r}\left(\frac{\eta_i}{\beta}\right)^2\right] \quad (24)$$

Under these conditions, we can rewrite (23) as the expectation with respect to $\boldsymbol{\eta}$

$$\check{\breve{V}}_N(\boldsymbol{\tau},\beta) = \mathrm{E}[\breve{V}_N(\boldsymbol{\tau}-\boldsymbol{\eta})] \quad (25)$$

In our case, $\check{h}(\boldsymbol{\eta},\beta)$ will be the sampled values of its pdf, which is convolved with the original objective function for smoothing.

The value of β plays a dominant role in the smoothing process by controlling the variance of $\check{h}(\boldsymbol{\eta},\beta)$; see properties 2 and 3. Furthemore, property 3 states that to avoid convergence to a local minimum, β has to be large at the start of the optimization process and is then reduced to approximately zero as the global minimum is reached.

The application of this technique to the SEPNLS method requires a gradient operation on the functional $\breve{\breve{V}}_N(\boldsymbol{\tau},\beta)$, i.e., $\breve{\breve{\boldsymbol{V}}}'_N(\boldsymbol{\tau},\beta)$. As described in (Styblinski & Tang, 1990; Rubinstein 1981), if $\breve{h}(\boldsymbol{\eta},\beta)$ is a Gaussian distribution, then the unbiased gradient estimate of the smoothed functional can be expressed as

$$\breve{\breve{\boldsymbol{V}}}'_N(\boldsymbol{\tau},\beta) = \frac{1}{M}\sum_{i=1}^{M}\breve{\boldsymbol{V}}'_N(\boldsymbol{\tau}-\beta\boldsymbol{\eta}_i) \tag{26}$$

In (26) M points $\boldsymbol{\eta}_i$ are sampled with the pdf $h(\boldsymbol{\eta})$. Substituting $M = 1$ in (26) one obtains the one-sample gradient estimator usually used in the stochastic approximation algorithms (Styblinski & Tang, 1990).

$$\breve{\breve{\boldsymbol{V}}}'_N(\boldsymbol{\tau},\beta) = \breve{\boldsymbol{V}}'_N(\boldsymbol{\tau}-\beta\boldsymbol{\eta}) \tag{27}$$

In Styblinski & Tang. (1990), using $\breve{\boldsymbol{V}}'_N(\boldsymbol{\tau} - \beta\boldsymbol{\eta})$ in (26), SAS is applied to the normalized steepest descent method. Edmonson *et al.* (1998) proposed a simplification that involves expressing the gradient $\boldsymbol{\tau} - \beta\boldsymbol{\eta}$ as a Taylor series around the operating point:

$$\breve{\boldsymbol{V}}'_N(\boldsymbol{\tau}-\beta\boldsymbol{\eta}) = \breve{\boldsymbol{V}}'_N(\boldsymbol{\tau}) - \beta\breve{\boldsymbol{V}}''_N(\boldsymbol{\tau})\boldsymbol{\eta} + \cdots \tag{28}$$

Additonally, $\breve{\boldsymbol{V}}''_N(\boldsymbol{\tau})$ in the above equation is approximated as an identity matrix and only the first two terms of the Taylor series are kept such that

$$\breve{\boldsymbol{V}}'_N(\boldsymbol{\tau}-\beta\boldsymbol{\eta}) \approx \breve{\boldsymbol{V}}'_N(\boldsymbol{\tau}) - \beta\boldsymbol{\eta} \tag{29}$$

Then $\breve{\boldsymbol{V}}'_N(\boldsymbol{\tau}-\beta\boldsymbol{\eta})$ is used to modify the least mean square (LMS) algorithm.

In this study, we extend the idea in Edmonson *et al.* (1998) to our SEPNLS method. Replacing $\breve{\boldsymbol{V}}'_N(\boldsymbol{\tau})$ in (18) by $\breve{\boldsymbol{V}}'_N(\boldsymbol{\tau} - \beta\boldsymbol{\eta})$, we obtain the following result.

$$\hat{\boldsymbol{\tau}}_N^{(j+1)} = \hat{\boldsymbol{\tau}}_N^{(j)} - \mu^{(j)}\left[\breve{\boldsymbol{R}}_N\left(\hat{\boldsymbol{\tau}}_N^{(j)}\right)\right]^{-1}\left(\breve{\boldsymbol{V}}'_N\left(\hat{\boldsymbol{\tau}}_N^{(j)}\right) - \beta^{(j)}\boldsymbol{\eta}\right) \tag{30}$$

This is our GSEPNLS method which modifies the SEPNLS method with an addition of a stochastic perturbation term.

Remark 2: As suggested in Styblinski & Tang (1990), β has to be chosen large at the start of the iterations and is then decreased to approximately zero as the global minimum is reached. And in Edmonson *et al.* (1998), the sequence of $\beta^{(j)}$ is chosen as a discrete exponentially decaying function of iteration number j. However, in both works β are chosen by trial and errors. And we have not found in the literature any reliable policy telling us how to determine reliable and efficient values of β. In this paper, however, based on empirical studies, we recommend the following choice: $\beta^{(j)} = \beta_0\breve{V}_N\left(\hat{\boldsymbol{\tau}}_N^{(j)}\right)$, where β_0 is a sufficiently large positive constant. It can be understood that if $\breve{V}_N\left(\hat{\boldsymbol{\tau}}_N^{(j)}\right)$ is far from the global minimum, $\beta^{(j)}$ is large, and if it becomes near the global minimum, $\beta^{(j)}$ becomes small. Finally,it should be mentioned here that the results are not sensitive to the constant β_0.

The algorithm of the GSEPNLS method can be summarized as follows.

(1) Let $j = 0$. Set β_0, the initial estimate $\hat{\boldsymbol{\tau}}_N^{(0)}$ and the considerable upper bound of time delays $\bar{\boldsymbol{\tau}}$.

(2) Set $\beta^{(j)} = \beta_0\breve{V}_N\left(\hat{\boldsymbol{\tau}}_N^{(j)}\right)$.

(3) Perform the following.

  (a) Compute $\Delta\hat{\boldsymbol{\tau}}_N^{(j+1)} = -\breve{\boldsymbol{R}}_N^{-1}\left(\hat{\boldsymbol{\tau}}_N^{(j)}\right)\left(\breve{\boldsymbol{V}}'_N\left(\hat{\boldsymbol{\tau}}_N^{(j)}\right) - \beta^{(j)}\boldsymbol{\eta}\right)$

  (b) Compute $\hat{\boldsymbol{\tau}}_N^{(j+1)} = \hat{\boldsymbol{\tau}}_N^{(j)} + \Delta\hat{\boldsymbol{\tau}}_N^{(j+1)}$

  (c) Check if $0 \le \hat{\tau}_{Ni}^{(j+1)} \le \bar{\tau}_i (i = 1,\cdots,r)$. If not, let $\Delta\hat{\boldsymbol{\tau}}_N^{(j+1)} = 0.5\Delta\hat{\boldsymbol{\tau}}_N^{(j+1)}$ and go back to (b).

  (d) Check if $\breve{V}_N\left(\hat{\boldsymbol{\tau}}_N^{(j+1)}\right) \le \breve{V}_N\left(\hat{\boldsymbol{\tau}}_N^{(j)}\right)$. If not, let $\Delta\hat{\boldsymbol{\tau}}_N^{(j+1)} = 0.5\Delta\hat{\boldsymbol{\tau}}_N^{(j+1)}$ and go back to (b).

(4) Terminate the algorithm if the stopping condition is satisfied. Otherwise, let $j = j + 1$ and go back to step 2.

Finally, by substituting $\hat{\boldsymbol{\tau}}_N$ into (13), the linear parameter vector $\boldsymbol{\theta}$ can be estimated by the linear LS method (13).

6. GSEPNIV METHOD

Although the GSEPNLS is able to converge to the global minimum, the estimetes are acceptable only in the case of low measurement noise. To achieve consistent estimates in the case of high measurement noise, we modify the GSEPNLS method to the GSEPNIV method.

We first introduce the following IV vector by using the input signals $\bar{u}_j(k)$ and sampled noise-free output $x(k)$:

$$\begin{aligned}\boldsymbol{m}^T(k,\boldsymbol{\tau}) &= [-\boldsymbol{\varphi}_{\bar{x}}^T(k), \boldsymbol{\varphi}_{\bar{u}_1}^T(k-\tilde{\tau}_1),\cdots,\boldsymbol{\varphi}_{\bar{u}_r}^T(k-\tilde{\tau}_r)]\\ \boldsymbol{\varphi}_{\bar{x}}^T(k) &= [\xi_{1\bar{x}}(k),\cdots,\xi_{n\bar{x}}(k)]\end{aligned} \tag{31}$$

where

$$\xi_{i\bar{x}}(k) = \frac{\left(\dfrac{T}{2}\right)^i(1+z^{-1})^i(1-z^{-1})^{n-i}}{\left[\alpha(1-z^{-1}) + \dfrac{T}{2}(1+z^{-1})\right]^n}\bar{x}(k) \tag{32}$$

and $\bar{x}(k) = (1 + z^{-1})x(k)/2$.

Remark 3: In practice, however, the noise-free output is never known. Therefore, a boot-strap scheme is usually used to generate the instrumental variables (Young & Jakeman, 1979). The estimated noise-free output $\hat{\bar{x}}(k)$ is obtained by discretizing the estimated system model by the bilinear transformation.

By using the IV vector, we can estimate the linear transfer function parameters by the linear IV method as

$$\begin{aligned}\widehat{\boldsymbol{\theta}}_{IVN}(\boldsymbol{\tau}) &= \boldsymbol{R}_{IV}^{-1}(N,\boldsymbol{\tau})\boldsymbol{f}_{IV}(N,\boldsymbol{\tau})\\ \boldsymbol{R}_{IV}(N,\boldsymbol{\tau}) &= \frac{1}{N-k_s}\sum_{k=k_s+1}^{N}\boldsymbol{m}(k,\boldsymbol{\tau})\boldsymbol{\varphi}^T(k,\boldsymbol{\tau})\\ \boldsymbol{f}_{IV}(N,\boldsymbol{\tau}) &= \frac{1}{N-k_s}\sum_{k=k_s+1}^{N}\boldsymbol{m}(k,\boldsymbol{\tau})\xi_{0\bar{y}}(k)\end{aligned} \tag{33}$$

In this case, the residual is given as

$$\breve{\varepsilon}_{IV}(k,\boldsymbol{\tau}) = \xi_{0\bar{y}}(k) - \boldsymbol{\varphi}^T(k,\boldsymbol{\tau})\boldsymbol{R}_{IV}^{-1}(N,\boldsymbol{\tau})\boldsymbol{f}_{IV}(N,\boldsymbol{\tau}) \tag{34}$$

Then SEPNLS method (17) is modified to the following SEPNIV method:

$$\widehat{\boldsymbol{\tau}}_{IVN}^{(j+1)} = \widehat{\boldsymbol{\tau}}_{IVN}^{(j)} - \mu^{(j)}\left[\breve{\boldsymbol{R}}_{IVN}\left(\widehat{\boldsymbol{\tau}}_{IVN}^{(j)}\right)\right]^{-1}\breve{\boldsymbol{V}}'_{IVN}\left(\widehat{\boldsymbol{\tau}}_{IVN}^{(j)}\right) \tag{35}$$

where

$$\begin{aligned}\breve{\boldsymbol{V}}'_{IVN}(\boldsymbol{\tau}) &= -\frac{1}{N-k_s}\sum_{k=k_s+1}^{N}\boldsymbol{\psi}_{IV}(k,\boldsymbol{\tau})\breve{\varepsilon}_{IV}(k,\boldsymbol{\tau})\\ \breve{\boldsymbol{R}}_{IVN}(\boldsymbol{\tau}) &= \frac{1}{N-k_s}\sum_{k=k_s+1}^{N}\boldsymbol{\psi}_{IV}(k,\tau)\boldsymbol{\psi}_m^T(k,\boldsymbol{\tau})\end{aligned} \tag{36}$$

$\boldsymbol{\psi}_m(k,\boldsymbol{\tau}) = [\psi_{m1}(k,\boldsymbol{\tau}),\cdots,\psi_{mr}(k,\boldsymbol{\tau})]^T$ is a slight modification of $\boldsymbol{\psi}(k,\boldsymbol{\tau})$ given in (20):

$$\begin{aligned}\boldsymbol{\psi}_{mj}(k,\boldsymbol{\tau}) &= \boldsymbol{\varphi}_{\tau_j}^T(k,\tau)\boldsymbol{R}_{IV}^{-1}(N,\boldsymbol{\tau})\boldsymbol{f}_{IV}(N,\boldsymbol{\tau})\\ &+\boldsymbol{\varphi}^T(k,\boldsymbol{\tau})\boldsymbol{R}_{IV}^{-1}(N,\boldsymbol{\tau})\boldsymbol{f}_{\tau_j}(N,\boldsymbol{\tau})\\ &-\boldsymbol{\varphi}^T(k,\boldsymbol{\tau})\boldsymbol{R}_{IV}^{-1}(N,\boldsymbol{\tau})\boldsymbol{R}_{\tau_j}(N,\boldsymbol{\tau})\boldsymbol{R}_{IV}^{-1}(N,\boldsymbol{\tau})\boldsymbol{f}_{IV}(N,\boldsymbol{\tau})\\ &-\boldsymbol{\varphi}^T(k,\boldsymbol{\tau})\boldsymbol{R}_{IV}^{-1}(N,\boldsymbol{\tau})\boldsymbol{R}_{\tau_j}^T(N,\tau)\boldsymbol{R}_{IV}^{-1}(N,\boldsymbol{\tau})\boldsymbol{f}_{IV}(N,\boldsymbol{\tau})\end{aligned} \tag{37}$$

and $\boldsymbol{\psi}_{IV}(k,\boldsymbol{\tau}) = [\psi_{IV1}(k,\boldsymbol{\tau}),\cdots,\psi_{IVr}(k,\boldsymbol{\tau})]^T$ is the IV vector to make $\breve{V}'_{IVN}(\boldsymbol{\tau})$ and $\breve{R}_{IVN}(\boldsymbol{\tau})$ unbiased:

$$\begin{aligned}\boldsymbol{\psi}_{IVj}(k,\boldsymbol{\tau}) &= \boldsymbol{\varphi}_{\tau_j}^T(k,\boldsymbol{\tau})\boldsymbol{R}_{IV}^{-1}(N,\boldsymbol{\tau})\boldsymbol{f}_{IV}(N,\boldsymbol{\tau})\\ &+\boldsymbol{m}^T(k,\boldsymbol{\tau})\boldsymbol{R}_{IV}^{-1}(N,\boldsymbol{\tau})\boldsymbol{f}_{\tau_j}(N,\boldsymbol{\tau})\\ &-\boldsymbol{m}^T(k,\boldsymbol{\tau})\boldsymbol{R}_{IV}^{-1}(N,\boldsymbol{\tau})\boldsymbol{R}_{\tau_j}(N,\boldsymbol{\tau})\boldsymbol{R}_{IV}^{-1}(N,\boldsymbol{\tau})\boldsymbol{f}_{IV}(N,\boldsymbol{\tau})\\ &-\boldsymbol{m}^T(k,\boldsymbol{\tau})\boldsymbol{R}_{IV}^{-1}(N,\boldsymbol{\tau})\boldsymbol{R}_{\tau_j}^T(N,\boldsymbol{\tau})\boldsymbol{R}_{IV}^{-1}(N,\boldsymbol{\tau})\boldsymbol{f}_{IV}(N,\boldsymbol{\tau})\end{aligned} \tag{38}$$

It can be shown through correlation analysis that the solution by the SEPNIV method is is equivalent to the noise-free solution by the SEPNLS method, i.e., the estimate is consistent if the algorithm converges to the global minimum. Detailed analysis is omitted here due to the limit of the paper length.

We should notice that the SEPNIV estimate does not minimize the LS criterion. However if the estimate is consistent, it should minimize the mean squares of the output error:

$$V_{IV}(\boldsymbol{\theta}_{IV},\boldsymbol{\tau}_{IV}) = \frac{1}{N-ks}\sum_{k=ks+1}^{N}\left(\bar{y}(k) - \hat{\bar{x}}(k)\right)^2 \tag{39}$$

The GSEPNLS method (30) is therefore modified to the following GSEPNIV method:

$$\widehat{\boldsymbol{\tau}}_{IVN}^{(j+1)} = \widehat{\boldsymbol{\tau}}_{IVN}^{(j)} - \mu^{(j)}\left[\breve{\boldsymbol{R}}_{IVN}\left(\widehat{\boldsymbol{\tau}}_{IVN}^{(j)}\right)\right]^{-1}\left(\breve{\boldsymbol{V}}'_{IVN}\left(\widehat{\boldsymbol{\tau}}_{IVN}^{(j)}\right) - \beta^{(j)}\boldsymbol{\eta}\right) \tag{40}$$

The algorithm of the GSEPNIV method is summarized as follows.

(1) Let $j = 0$. Set β_0, the initial estimate $\widehat{\boldsymbol{\theta}}_{IVN}^{(0)}$ and $\widehat{\boldsymbol{\tau}}_{IVN}^{(0)}$, and the considerable upper bound of time delays $\bar{\boldsymbol{\tau}}$. Generate the estimated noise-free output by using $\widehat{\boldsymbol{\theta}}_{IVN}^{(0)}$ and $\widehat{\boldsymbol{\tau}}_{IVN}^{(0)}$.

(2) Set $\beta^{(j)} = \beta_0 V_{IV}\left(\widehat{\boldsymbol{\theta}}_{IVN}^{(j)}, \widehat{\boldsymbol{\tau}}_{IVN}^{(j)}\right)$

(3) Perform the following.
   (a) Compute
   $$\Delta\widehat{\boldsymbol{\tau}}_{IVN}^{(j+1)} = -\breve{\boldsymbol{R}}_{IVN}^{-1}\left(\widehat{\boldsymbol{\tau}}_{IVN}^{(j)}\right)\left(\breve{\boldsymbol{V}}'_{IVN}\left(\widehat{\boldsymbol{\tau}}_{IVN}^{(j)}\right) - \beta^{(j)}\boldsymbol{\eta}\right)$$
   (b) Compute
   $$\widehat{\boldsymbol{\tau}}_{IVN}^{(j+1)} = \widehat{\boldsymbol{\tau}}_{IVN}^{(j)} + \Delta\widehat{\boldsymbol{\tau}}_{IVN}^{(j+1)}$$
   (c) Check if $0 \le \widehat{\tau}_{IVNi}^{(j+1)} \le \bar{\tau}_i (i = 1,\cdots,r)$. If not, let $\Delta\widehat{\boldsymbol{\tau}}_{IVN}^{(j+1)} = 0.5\Delta\widehat{\boldsymbol{\tau}}_{IVN}^{(j+1)}$ and go back to (b).
   (d) Compute
   $$\widehat{\boldsymbol{\theta}}_{IVN}^{(j+1)} = \boldsymbol{R}_{IV}^{-1}\left(N,\widehat{\boldsymbol{\tau}}_{IVN}^{(j+1)}\right)\boldsymbol{f}_{IV}\left(N,\widehat{\boldsymbol{\tau}}_{IVN}^{(j+1)}\right).$$
   (e) Check if the estimated system model that generates the estimated noise-free output is stable. If not, let $\widehat{\boldsymbol{\theta}}_{IVN}^{(j+1)} = \widehat{\boldsymbol{\theta}}_{IVN}^{(j)}$.
   (f) Generate the estimated noise-free output by using $\widehat{\boldsymbol{\theta}}_{IVN}^{(j+1)}$ and $\widehat{\boldsymbol{\tau}}_{IVN}^{(j+1)}$.
   (g) Check if
   $$V_{IV}\left(\widehat{\boldsymbol{\theta}}_{IVN}^{(j+1)}, \widehat{\boldsymbol{\tau}}_{IVN}^{(j+1)}\right) \le V_{IV}\left(\widehat{\boldsymbol{\theta}}_{IVN}^{(j)}, \widehat{\boldsymbol{\tau}}_{IVN}^{(j)}\right).$$
   If not, let $\Delta\widehat{\boldsymbol{\tau}}_{IVN}^{(j+1)} = 0.5\Delta\widehat{\boldsymbol{\tau}}_{IVN}^{(j+1)}$ and go back to (b).

(4) Terminate the algorithm if the stopping condition is satisfied. Otherwise, let $j = j + 1$ and go back to step 2.

7. NUMERICAL RESULTS AND CONCLUSIONS

Consider the following MISO continuous-time system:

$$\begin{aligned}&\ddot{x}(t) + a_1\dot{x}(t) + a_2 x(t) = b_{11}u_1(t-\tau_1) + b_{21}u_2(t-\tau_2)\\ &a_1 = 3.0,\quad a_2 = 4.0,\quad b_{11} = 2.0,\quad b_{21} = 1.0,\\ &\tau_1 = 9.130,\quad \tau_2 = 2.570\end{aligned} \tag{41}$$

Each input signal is a filtered white signal. The sampling period taken as $T = 0.05$, and α in the low-pass pre-filter $Q(p)$ is 0.4. β_0 is chosen as 10^5, as suggested in remark 2. The algorithms are terminated after 200 iterations.

We have found the proposed global search algorithms converge to the global minimum almostly in all the cases of various combinations of the initial estimates, input signals and measurement noise. On the other hand, the conventional local search algorithms failed in most cases.

For one fixed realization of $\boldsymbol{\eta}$, the algorithms were implemented for 20 realizations of the measurement noise whose NSR (noise to signal ratio) is 30% with a data length of 4000. The initial estimates were set as $\widehat{\boldsymbol{\tau}}_N^{(0)} = \widehat{\boldsymbol{\tau}}_{IVN}^{(0)} = [1, 1]^T$.

The results are shown in a Table 1, where A and B denote respectively GSEPNLS and GSEPNIV methods. The table includes the mean and standard deviation of the estimates. It can be seen that the biases by the GSEPNLS method are significant, whereas the GSEPNIV method yields consistent estimates. An example of the convergency behaviour of the estimates of the time delays is shown in Fig. 1. And the locus of the time delay estimates on the contour of the output error criterion (39) is shown in Fig. 2.

Table 1. Estimates of GSEPNLS and GSEPNIV methods.

	$\widehat{a_1}(3.0)$	$\widehat{a_2}(4.0)$	$\widehat{b_{11}}(2.0)$	$\widehat{b_{21}}(1.0)$
A	2.6078 ±0.0414	3.5944 ±0.0458	1.7678 ±0.0269	0.8863 ±0.0186
B	3.0042 ±0.0625	4.0083 ±0.0772	2.0065 ±0.0417	1.0014 ±0.0252

	$\widehat{\tau_1}(9.13)$	$\widehat{\tau_2}(2.57)$
A	9.1032 ±0.0041	2.5463 ±0.0064
B	9.1297 ±0.0057	2.5689 ±0.0072

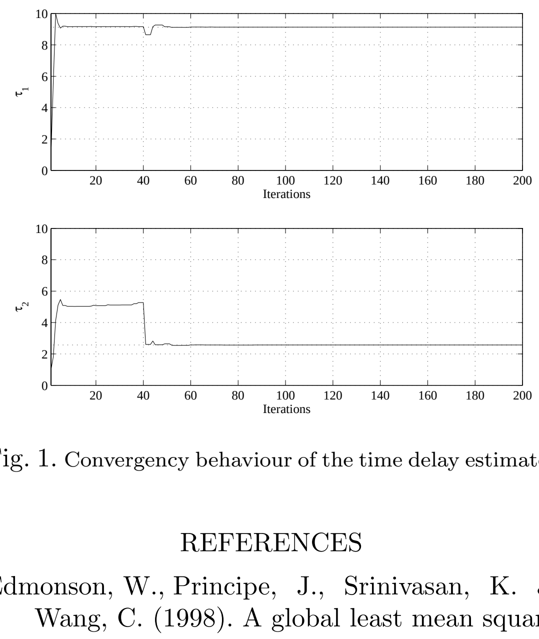

Fig. 1. Convergency behaviour of the time delay estimates

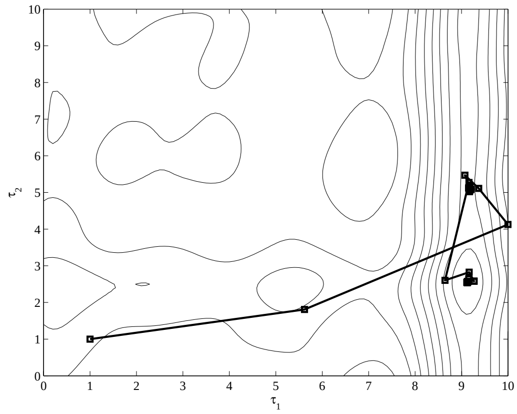

Fig. 2. Locus of the time delay estimates on the contour of the output error criterion

REFERENCES

Edmonson, W., Principe, J., Srinivasan, K. & Wang, C. (1998). A global least mean square algorithm for adaptive IIR filtering. *I.E.E.E. Trans.on Circuits and Systems II, 45*(3), 379-384.

Iemura, H., Yang, Z. J., Kanae, S., & Wada, K. (2004). Identification of continuous-time systems with unkown time delays by global nonlinear least-squares method. *IFAC Workshop on ALCOSP 04 and PSYCO 04*, Yokohama, Japan.

Gawthrop, P. J., Nihtilä, M. T. & Besharati-Rad A. (1989) Recursive parameter estimation of continuous-time systems with unknown time delay. *Control–Theory and Advanced Technology, 5*(3), 227-248.

Ngia, L. S. H. (2001). Separable nonlinear least-squares methods for efficient off-line and on-line modeling of systems using Kautz and Laguerre filters. *I.E.E.E. Trans.on Circuits and Systems II, 48*(6), 562-579.

Rubinstein, R.Y. (1981). *Simulation and the Monte Carlo method*. New York: John Wiley.

Ruhe, A., & Wedin, P.A. (1980). Algorithms for separable nonlinear least squares problems. *SIAM Review, 22*, 318-337.

Styblinski, M.A., & Tang, T.S. (1990). Experiments in nonconvex optimization:stochastic approximation with function smoothing and simulated annealing. *Neural Networks, 3*, 467-483.

Tuch, J., Feuer, A., & Palmor, Z. J. (1994). Time delay estimation in continuous linear time-invariant systems. *I.E.E.E. Trans.on Automatic Control, 39*, 823-827.

Yang, Z.J., , Hachino, T & Tsuji T. (1997). On-line identification of continuous time-delay systems combining the least-squares techniques with the genetic algorithm, *International Journal of Control, 66* (1), 23-42.

Young, P., & Jakeman, A. (1979). Refined instrumental variable methods of recursive time series analysis. Part 1. Single input, single output systems, *International Journal of Control, 19*(1), 1-30.