To cite this version:
Baysse, Arnaud and Carrillo, Francisco and Habbadi , Abdallah *Least squares and output error identification algorithms for continuous time systems with unknown time delay operating in open or closed loop.* (2012) In: Sysid 2012, 16th IFAC Symposium on System Identification, 11-13 July 2012, Brussels, Belgium.

Least squares and output error identifcation algorithms for continuous time systems with unknown time delay operating in open or closed loop

A. Baysse, F. J. Carrillo and A. Habbadi

Université de Toulouse, INP/ENIT, LGP,
47 av. d'Azereix - BP 1629 - 65016 Tarbes Cedex
e-mail: [Francisco.Carrillo, arnaud.baysse, Abdallah.Habbadi]@enit.fr

Abstract: This paper presents two off-line output error identification algorithms for linear continuous-time systems with unknown time delay from sampled data. The proposed methods (for open and closed loop systems) use a Nonlinear Programming algorithm and needs an initialization step that is also proposed from a modification of the Yang algorithm. Simulations, as illustrated by Monte-Carlo runs, show that the obtained parameters are unbiased and very accurate.

Keywords: identification of continuous-time systems with time delay, least squares and output error identification algorithms. Open-loop and Closed-loop identification

1. INTRODUCTION

Many industrial processes possess a time delay, such as chemical, thermal or biological systems. It is important to identify the value of this delay, in particular for control. For example, the design of a Smith predictor implies a good knowledge of the time delay because this information is directly included in the controller.

There are several papers coping with the identification of open loop systems with time delay. Almost all are proposed to identify first and second order delayed systems from step signals. Amongst the existing methods, most of them use integrals to obtain a parameter vector that contains the time delay. There are off-line (Liu et al. (2007); Hwang and Lai (2004); Wang et al. (2001)) and on-line methods (Ahmed et al. (2006); Garnier and Wang (2008, chap. 11)). The parameters are then identified using least-squares, or instrumental variable methods. Other methods separate the model in two different parts: the first one contains the linear parameters and the second one the nonlinear parameter, i.e. the time delay. There are also methods that use nonlinear programming (NLP) to estimate the nonlinear parameter. Amongst these, Yang *et al.* (Yang et al. (2007); Garnier and Wang (2008, chap. 12)), proposed an interesting method which permits to identify multiple input single output (MISO) systems with little prior knowledge, using the least-squares, this algorithm will be adapted and used for initialization of the new output error identification algorithms (for open and closed-loop systems).

This paper, propose two algorithms which permit the identification of all the parameters of a linear continuous-time system with unknown time delay. They can be considered as extensions of the sensitivity based identification algorithm previously proposed in (Carrillo et al. (2009)). These algorithms are both based on the output error method and can identify a linear stable system of any order with time delay in a very accurate manner. The first algorithm was previously presented in an open loop form in (Baysse et al. (2011)) and it will be called Continuous-Time Output Error CTOE algorithm. The second proposed algorithm will be useful for the case of closed loop systems and it will be called Continuous-Time Closed-Loop Output Error CTCLOE algorithm. The paper is organized as follows. In the second section the process and model description are presented. Then, the new output error methods are detailed in the third section for open and closed loop systems. Section four describes a least squares algorithm that will be used for initialization. Simulation results for open and closed-loop systems are described in section 5. Finally, some conclusions are presented in section 6.

2. PROCESS AND MODEL DESCRIPTION

When using sampled data, the system with input $u(t_k)$ and output $y(t_k)$ can be represented by (see Garnier and Wang (2008)):

$$y(t_k) = \frac{B(p)}{A(p)} e^{-T_d p} u(t_k) + w(t_k) \tag{1}$$

where $w(t_k)$ is assumed to be a Gaussian white noise, p is the derivative operator $p = \dfrac{d}{dt}$, T_d is the time delay and $e^{-T_d p}$ has to be understood as:

$$e^{-T_d p} u(t) = u(t - T_d) \tag{2}$$

$A(p)$ and $B(p)$ are polynomials with the following structure:

$$A(p) = 1 + a_1 p + \cdots + a_n p^n \quad (3)$$

$$B(p) = b_0 + b_1 p + \cdots + b_m p^m; \qquad m < n \quad (4)$$

$y(t_k)$ is the amplitude of the continuous-time signal $y(t)$ at time $t_k = kT_s$, with T_s the sampling period. For the rest of the presentation, the signals are written as a function of the time (t). For implementation purpose, they have to be considered at sampling intervals.

2.1 Open loop system

Let us denote $\widehat{y}(t)$ the estimated output of the model, defined as:

$$\widehat{y}(t) = \frac{\widehat{B}(p)}{\widehat{A}(p)} e^{-\widehat{T}_d p} u(t) \quad (5)$$

Where $\widehat{T}_d$ is the estimated time delay ; $\widehat{A}(p)$ and $\widehat{B}(p)$ are polynomials of estimated parameters. The output can be expressed in a regressor form as it was shown in Baysse et al. (2011).

The proposed approach identifies at the same time the parameters $\widehat{\theta}$ and the parameter $\widehat{T}_d$. Thus, a vector $\widehat{\Theta}$ which contains all the parameters is defined as $\widehat{\Theta}^T = \left[\, \widehat{\theta}^T \ \widehat{T}_d \,\right]$.

2.2 Closed-loop system

If the process is operating in closed loop, with a controller $C(p)$ defined as:

$$u(t) = C(p)\left[r(t) - y(t)\right]; \quad (6)$$

$$C(p) = \frac{R(p)}{S(p)} \quad (7)$$

where $r(t)$ is the reference signal.

Using the output error approach, the estimated output of the model $\widehat{y}(t)$, is defined as:

$$\widehat{y}(t) = \frac{\widehat{B}(p)}{\widehat{A}(p)} e^{-\widehat{T}_d p} \widehat{u}(t) \quad (8)$$

$$\widehat{u}(t) = C(p)\left[r(t) - \widehat{y}(t)\right] \quad (9)$$

The output can be expressed in a regressor form:

$$\begin{aligned} \widehat{y}(t) &= -\widehat{a}_1 \widehat{y}^{(1)}(t) - \cdots - \widehat{a}_n \widehat{y}^{(n)}(t) \\ &\quad + \widehat{b}_0 \widehat{u}(t - \widehat{T}_d) + \cdots + \widehat{b}_m \widehat{u}^{(m)}(t - \widehat{T}_d) \\ \widehat{y}(t) &= \widehat{\theta}^T \widehat{\phi}(t) \end{aligned} \quad (10)$$

with:

$$\widehat{\theta}^T = \left[\, \widehat{a}_1 \ \cdots \ \widehat{a}_n \ \widehat{b}_0 \ \cdots \ \widehat{b}_m \,\right], \quad (11)$$

$$\begin{aligned} \widehat{\phi}^T(t) = \big[& -\widehat{y}^{(1)}(t) \ \cdots \ -\widehat{y}^{(n)}(t) \\ & \widehat{u}(t - \widehat{T}_d) \ \cdots \ \hat{u}^{(m)}(t - \widehat{T}_d) \,\big] \end{aligned} \quad (12)$$

where $\widehat{u}^{(m)}(t) = p^m \widehat{u}(t)$ and $\widehat{y}^{(n)}(t) = p^n \widehat{y}(t)$.

3. THE OFF-LINE OUTPUT ERROR METHOD

The quadratic criterion to be minimized from N sampled data is:

$$J = \sum_{k=1}^{N} \left(y(t_k) - \widehat{y}(t_k)\right)^2 = \sum_{k=1}^{N} \varepsilon^2(t_k) \quad (13)$$

Since this criterion is nonlinear in the parameters, one must use a NonLinear Programming (NLP) method. These methods are based on an iterative approach to converge. At the iteration $j+1$, the parameters are updated using:

$$\widehat{\Theta}_{j+1} = \widehat{\Theta}_j + \Delta\widehat{\Theta}_j \quad (14)$$

where $\Delta\widehat{\Theta}_j$ is the increment of the parameters $\widehat{\Theta}_j$. The three NLP methods most used are the gradient method, the Gauss-Newton method, and the Levenberg-Marquardt method, where the increment is respectively described as:

$$\Delta\widehat{\Theta}_j = -\mu J'(\widehat{\mathbf{\Theta}}_j) \quad (15)$$

$$\Delta\widehat{\Theta}_j = -\mu \left[J''(\widehat{\mathbf{\Theta}}_j)\right]^{-1} J'(\widehat{\mathbf{\Theta}}_j) \quad (16)$$

$$\Delta\widehat{\Theta}_j = -\left[J''(\widehat{\mathbf{\Theta}}_j) + \mu \mathrm{diag}\left(J''(\widehat{\mathbf{\Theta}}_j)\right)\right]^{-1} J'(\widehat{\mathbf{\Theta}}_j) \quad (17)$$

where $J'(\widehat{\mathbf{\Theta}}_j)$ and $J''(\widehat{\mathbf{\Theta}}_j)$ are respectively the gradient and the Hessian, μ is a parameter which permits the tuning of the algorithms. The latter algorithm (17) is the most performant due to its tuning parameter μ. For high μ values the algorithm is close to the gradient method, for low μ values the algorithm is close to the Gauss-Newton method. The gradient and the Hessian are computed using the sensitivity functions. Denoting $\sigma_{\widehat{\theta}}(t) = \frac{\partial \widehat{y}(t)}{\partial \widehat{\theta}}$ the output sensitivity function w.r.t. $\widehat{\theta}$, and $\sigma_{\widehat{T}_d}(t) = \frac{\partial \widehat{y}(t)}{\partial \widehat{T}_d}$ the output sensitivity function w.r.t. $\widehat{T}_d$, the sensitivity vector $\sigma(t)$ is written in the following form:

$$\begin{aligned} \sigma^T(t) &= \left[\, \sigma^T_{\widehat{\theta}}(t) \ \sigma_{\widehat{T}_d}(t) \,\right] \\ &= \left[\, \sigma_{\widehat{a}_1}(t) \ \cdots \ \sigma_{\widehat{a}_n}(t) \ \sigma_{\widehat{b}_0}(t) \ \cdots \ \sigma_{\widehat{b}_m}(t) \ \sigma_{\widehat{T}_d}(t) \,\right] \end{aligned} \quad (18)$$

The gradient and the Hessian, with the Gauss-Newton approximation, are then respectively given by:

$$J'(\widehat{\mathbf{\Theta}}) = -2 \sum_{k=1}^{N} \varepsilon(t_k)\, \sigma(t_k) \quad (19)$$

$$J''(\widehat{\mathbf{\Theta}}) \simeq 2 \sum_{k=1}^{N} \sigma(t_k)\, \sigma^T(t_k) \quad (20)$$

3.1 Sensitivity functions computation

Open loop system

The following computations for an open loop system were developed in a previous paper (Baysse et al. (2011)). Only the main results are presented here.

The sensitivities of a numerator parameter $\widehat{b}_i$ and of a denominator parameter $\widehat{a}_i$ are respectively computed in the following way:

$$\sigma_{\widehat{b}_i}(t) = \frac{p^i}{\widehat{A}(p)} e^{-\widehat{T}_d p} u(t) \; ; \; \sigma_{\widehat{a}_i}(t) = \frac{-p^i}{\widehat{A}(p)} \widehat{y}(t) \qquad (21)$$

Notice that the sensitivity functions $\sigma_{\widehat{\theta}}^T(t)$ can be written in compact notation: $\sigma_{\widehat{\theta}}^T(t) = \frac{1}{\widehat{A}(p)} \widehat{\phi}(t, \widehat{T_d})$. So the sensitivity functions $\sigma_{\widehat{\theta}}^T(t)$ is the regression vector $\widehat{\phi}(t, \widehat{T_d})$ filtered by $\frac{1}{\widehat{A}(p)}$. This implicit filtering gives an optimal estimate as shown in Garnier and Wang (2008), when $\widehat{A}(p) \backsimeq A(p)$.

The sensitivity of the simulated output with respect to the time delay $\widehat{T}_d$ is computed in the following manner:

$$\sigma_{\widehat{T}_d}(t) = \frac{\partial \widehat{y}(t)}{\partial \widehat{T}_d} = \frac{-p\widehat{B}(p)}{\widehat{A}(p)} e^{-\widehat{T}_d p} u(t) = -p\widehat{y}(t) \qquad (22)$$

It must be noted that the sensitivity function $\sigma_{\widehat{T}_d}$ is realizable only if $m < n$.

Closed-loop system

In this case the sensitivity function of the estimated output with respect of parameter of the numerator is given by

$$\sigma_{\widehat{b}_i}(t) = \frac{\partial \widehat{y}(t)}{\partial \widehat{b}_i} = \frac{\partial}{\partial \widehat{b}_i} \left[\frac{\widehat{B}(p) e^{-\widehat{T}_d p}}{\widehat{A}(p)} \widehat{u}(t) \right] \qquad (23)$$

Using (9), it can be verified as in Carrillo et al. (2009) that:

$$\sigma_{\widehat{b}_i}(t) = \frac{S(p) e^{-\widehat{T}_d p} p^i}{\widehat{A}(p) S(p) + \widehat{B}(p) R(p) e^{-\widehat{T}_d p}} \widehat{u}(t)$$

$$\sigma_{\widehat{b}_i}(t) = \frac{S(p) e^{-\widehat{T}_d p} p^i}{\widehat{P}_d(p)} \widehat{u}(t); \qquad (24)$$

$$\widehat{P}_d(p) = \widehat{A}(p) S(p) + \widehat{B}(p) R(p) e^{-\widehat{T}_d p} \qquad (25)$$

The sensitivity function with respect to a parameter of the denominator is calculated in the following manner:

$$\sigma_{\widehat{a}_i}(t) = \frac{\partial \widehat{y}(t)}{\partial \widehat{a}_i} = \frac{\partial}{\partial \widehat{a}_i} \left[\frac{\widehat{B}(p) e^{-\widehat{T}_d p}}{\widehat{A}(p)} \widehat{u}(t) \right] \qquad (26)$$

Then, with (9) the following result is obtained:

$$\sigma_{\widehat{a}_i}(t) = \frac{-S(p) p^i}{\widehat{A}(p) S(p) + \widehat{B}(p) R(p) e^{-\widehat{T}_d p}} \widehat{y}(t)$$

$$\sigma_{\widehat{a}_i}(t) = \frac{-S(p) p^i}{\widehat{P}_d(p)} \widehat{y}(t) \qquad (27)$$

With $\widehat{P}_d(p)$ given by (25).

The sensitivity functions $\sigma_{\widehat{\theta}}$ can be written in the compact notation:

$$\sigma_{\widehat{\theta}}(t) = \frac{S(p)}{\widehat{P}_d(p)} \widehat{\phi}(t) \qquad (28)$$

So the sensitivity function can be seen as a filtering (by $\frac{S}{P}$) of the regression vector. This implicit filtering is the same that Landau Landau and Karimi (1997) obtained in the discrete-time case for the F-CLOE algorithm which is well known to have excellent convergence properties.

Finally, the output sensitivity function with respect to the time delay is computed by:

$$\sigma_{\widehat{T}_d}(t) = \frac{\partial \widehat{y}(t)}{\partial \widehat{T}_d} = \frac{\partial}{\partial \widehat{T}_d} \left[\frac{\widehat{B}(p)}{\widehat{A}(p)} e^{-\widehat{T}_d p} \widehat{u}(t) \right] \qquad (29)$$

Using (9), this gives:

$$\sigma_{\widehat{T}_d}(t) = \frac{-p\widehat{B}(p) S(p)}{\widehat{P}_d(p)} e^{-\widehat{T}_d p} \widehat{u}(t) \qquad (30)$$

With $\widehat{P}_d(p)$ given by (25).

Because the time delay is contained in the denominator of all the sensitivities functions (24), (27), (30) it is not obvious how they can be implemented. To cope with, the three implementable structures shown in the figure 1 are proposed. For example, it can be easily verified (from standard properties of the first proposed closed-loop structure) that:

$$\sigma_{\widehat{b}_i}(t) = \frac{p^i}{\widehat{A}(p)} \frac{1}{1 + \frac{\widehat{B}(p)}{\widehat{A}(p)} \frac{R(p)}{S(p)} e^{-\widehat{T}_d p}} e^{-\widehat{T}_d p}$$

$$= \frac{p^i S(p)}{\widehat{A}(p) S(p) + \widehat{B}(p) R(p) e^{-\widehat{T}_d p}} e^{-\widehat{T}_d p} \qquad (31)$$

$$\sigma_{\widehat{b}_i}(t) = \frac{p^i S(p)}{\widehat{P}_d(p)} e^{-\widehat{T}_d p} \qquad (32)$$

This will be also the case for all the others sensitivities functions.

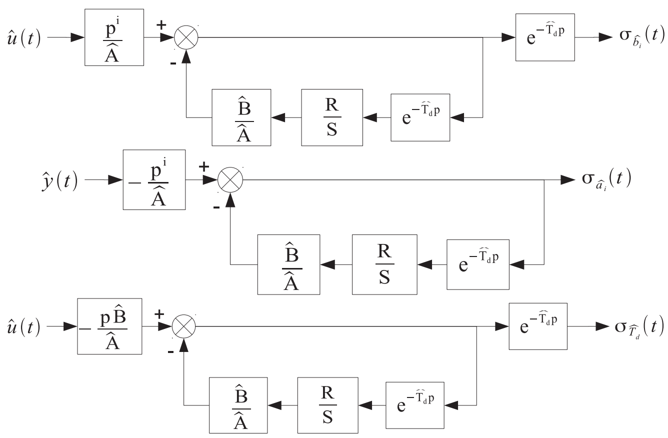

Figure 1. Sensitivity functions realization

3.2 Algorithm description of the proposed CTOE and CTCLOE methods

The algorithm is the same for the open loop or closed loop system. For the open loop system, the algorithm was previously presented in Baysse et al. (2011). Here the algorithm is presented for the closed loop system.

(1) Let j=0. Use an initial value of $\widehat{\Theta}_0$, μ_0 and J_0.
(2) Simulate the model using (8) to obtain $\widehat{y}(t)$.
(3) Compute the sensitivity function $\sigma^T(t_k)$ using (24), (27) and (30).
(4) Compute the gradient and the Hessian with (19) and (20).
(5) Perform the following:
   Compute
   $\widehat{\Theta}_{j+1} = \widehat{\Theta}_j - \left[J''(\widehat{\boldsymbol{\Theta}}_j) + \mu \text{diag}\left(J''(\widehat{\boldsymbol{\Theta}}_j)\right)\right]^{-1} J'(\widehat{\boldsymbol{\Theta}}_j)$.
   (a) Simulate the model (8) to obtain the new estimated output $\widehat{y}(t)$.
   (b) Compute $J(\widehat{\boldsymbol{\Theta}}_{j+1}) = \sum_{k=1}^{N} \left(y\left(t_k\right) - \widehat{y}\left(t_k\right)\right)^2$.
   (c) Check if $J(\widehat{\boldsymbol{\Theta}}_{j+1}) \leqslant J(\widehat{\boldsymbol{\Theta}}_j)$. If not, do $\mu_{j+1} = 10\mu_j$ and go back to (a).
   (d) Do $\mu_{j+1} = \frac{1}{4}\mu_j$.
(6) Terminate the algorithm if the convergence condition is satisfied. Otherwise, let $j = j + 1$ and go back to step 2.

4. LEAST SQUARES ALGORITHM USED FOR INITIALIZATION

The proposed output error method needs an initialization step for the estimated model (5) or (8). This task can be done by an adaptation of the method proposed by Yang et al. (2007). This method, based on the separable least-squares method, identify firstly the nonlinear parameter with a NLP algorithm, and then the linear parameters with the classical linear least-squares algorithm. This algorithm is called GSEPNLS for Global Separable Nonlinear Least-Squares.

The nonlinear parameter, i.e. the time delay, is identified using a Gauss Newton algorithm. It is well-known that this NLP algorithm can only insures local convergence. Thus, to avoid local minima, Yang et al. (2007) propose to use a random variable added to the gradient. The algorithm permits then to obtain a better convergence.

It is proposed in the following an adaptation of the GSEPNLS algorithm. The main differences come from its use in a SISO system, the use of a continuous-time filter to compute the derivatives, and the use of the filtered derivative as output.

When using the least-squares algorithm, the model (8) can be written in regressor form as:

$$\begin{aligned}\widehat{y}^{(n)}(t) &= -\frac{1}{\widehat{a}_n}y(t) - \frac{\widehat{a}_1}{\widehat{a}_n}y^{(1)}(t) - - \cdots - \frac{\widehat{a}_{n-1}}{\widehat{a}_n}y^{(n-1)}(t) \\ &\quad + \frac{\widehat{b}_0}{\widehat{a}_n}e^{-\widehat{T}_d p}u(t) + \cdots + \frac{\widehat{b}_m}{\widehat{a}_n}e^{-\widehat{T}_d p}u^{(m)}(t) \\ \widehat{y}^{(n)}(t) &= \widehat{\vartheta}^T\phi(t,\widehat{T}_d) \end{aligned} \tag{33}$$

where

$$\widehat{\vartheta}^T = \left[\, \frac{1}{\widehat{a}_n}\ \frac{\widehat{a}_1}{\widehat{a}_n}\ \cdots\ \frac{\widehat{a}_{n-1}}{\widehat{a}_n}\ \frac{\widehat{b}_0}{\widehat{a}_n}\ \cdots\ \frac{\widehat{b}_m}{\widehat{a}_n} \,\right] \tag{34}$$

$$\begin{aligned}\phi^T(t,\widehat{T}_d) = \big[&-y(t)\ -y^{(1)}(t)\ \cdots\ -y^{(n-1)}(t) \\ &e^{-\widehat{T}_d p}u(t)\ \cdots\ e^{-\widehat{T}_d p}u^{(m)}(t) \big]\end{aligned} \tag{35}$$

But, because of the time delay, (33) is a nonlinear model. To make the vector $\phi^T(t,\widehat{T}_d)$ realizable, it is necessary to compute the derivatives of the signals $u(t)$ and $y(t)$. This is achieved using a low-pass filter, defined as:

$$F(p) = \frac{1}{\Lambda(p)} = \frac{1}{(1+\lambda p)^n} = \frac{1}{1+\lambda_1 p + \ldots + \lambda_n p^n} \tag{36}$$

where λ is a constant which determines the pass band of the filter, and $\lambda_1, \ldots, \lambda_n$ are polynomial coefficient of the filter. The choice of this parameter implies prior knowledge of the system band pass.

Denoting with the subscript f the filtered signals, the regressor model (33) can be written:

$$\begin{aligned}\widehat{y}_f^{(n)}(t) &= -\frac{\widehat{a}_1}{\widehat{a}_n}y_f^{(1)}(t) + \cdots - \frac{\widehat{a}_{n-1}}{\widehat{a}_n}y_f^{(n-1)}(t) \\ &\quad + \frac{\widehat{b}_0}{\widehat{a}_n}e^{-\widehat{T}_d p}u_f(t) + \cdots + \frac{\widehat{b}_m}{\widehat{a}_n}e^{-\widehat{T}_d p}u_f^{(m)}(t) \\ \widehat{y}_f^{(n)}(t) &= \widehat{\vartheta}^T\phi_f(t,\widehat{T}_d)\end{aligned} \tag{37}$$

where

$$\begin{aligned}\phi_f^T(t,\widehat{T}_d) = \big[&-y_f(t)\ \cdots\ -y_f^{(n-1)}(t) \\ &e^{-\widehat{T}_d p}u_f(t)\ \cdots\ e^{-\widehat{T}_d p}u_f^{(m)}(t) \big]\end{aligned} \tag{38}$$

Using (37), the quadratic criterion that have to be minimized is written:

$$J(\widehat{T}_d) = \frac{1}{2}\sum_{k=1}^{N}\left(\varepsilon_f(t_k)\right)^2 \tag{39}$$

$$\varepsilon_f\left(t_k\right) = y_f^{(n)}(t_k) - \widehat{\vartheta}^T\phi_f(t,\widehat{T}_d) \tag{40}$$

In the following, for the presentation of this algorithm, it is easier to use matrix notation. Thus, the following variables are defined:

$$\Phi_f(\widehat{T}_d) = \left[\, \phi_f(1,\widehat{T}_d)\ \cdots\ \phi_f(N,\widehat{T}_d) \,\right]^T \tag{41}$$

$$Z = \left[\, y_f^{(n)}(1)\ \cdots\ y_f^{(n)}(N) \,\right]^T \tag{42}$$

$$E = \left[\, \varepsilon_f(1)\ \cdots\ \varepsilon_f(N) \,\right]^T \tag{43}$$

With this notation, the quadratic criterion (39), (40) can be written as:

$$J(\widehat{T}_d) = \frac{1}{2}E^T E \tag{44}$$

$$E = Z - \Phi_f(\widehat{T}_d)\widehat{\vartheta} \tag{45}$$

The principle of the separable algorithm is to firstly identify the nonlinear parameter $\widehat{T}_d$ with a NLP method then the linear parameters $\widehat{\vartheta}$ with the linear least squares algorithm. To show how the gradient and the Hessian are computed, the quadratic criterion (44) must be written as a function of the time delay $\widehat{T}_d$. The estimated vector $\widehat{\vartheta}$, can be replaced by its identified value using the linear Least Squares (LS) algorithm.

$$\widehat{\vartheta} = \left[\Phi_f^T(\widehat{T}_d)\Phi_f(\widehat{T}_d)\right]^{-1}\Phi_f^T(\widehat{T}_d)Z = \Phi_f^{\dagger}(\widehat{T}_d)Z \quad (46)$$

where $\Phi_f^{\dagger}$ is the pseudo-inverse of Φ_f. So the quadratic criterion (44) can be expressed only as a function of the time delay:

$$J(\widehat{T}_d) = \frac{1}{2}\left\|Z - \Phi_f(\widehat{T}_d)\Phi_f^{\dagger}(\widehat{T}_d)Z\right\|_2^2 \quad (47)$$

To compute the gradient and the Hessian, a notation similar to Ngia (2001) is used:

$$E_{\widehat{T}_d} = \frac{dE}{d\widehat{T}_d} = -\frac{d\Phi_f(\widehat{T}_d)}{d\widehat{T}_d}\widehat{\vartheta} = -\frac{d\Phi_f(\widehat{T}_d)}{d\widehat{T}_d}\Phi_f^{\dagger}(\widehat{T}_d)Z \quad (48)$$

$$E_{\widehat{\vartheta}} = \frac{dE}{d\widehat{\vartheta}} = -\Phi_f(\widehat{T}_d) \quad (49)$$

The gradient and the Hessian, with the Gauss-Newton approximation, will have the form (as shown in Ngia (2001)):

$$J'(\widehat{T}_d) = E_{\widehat{T}_d}^T E \quad (50)$$

$$J''(\widehat{T}_d) = E_{\widehat{T}_d}^T E_{\widehat{T}_d} - E_{\widehat{T}_d}^T E_{\widehat{\vartheta}}\left[E_{\widehat{\vartheta}}^T E_{\widehat{\vartheta}}\right]^{-1} E_{\widehat{\vartheta}}^T E_{\widehat{T}_d} \quad (51)$$

Yang *et al.* propose to use the Gauss-Newton NLP method, with a stochastic term added to the gradient. Denoting η a Gaussian random variable, the evolution of the estimated time delay with the Gauss-Newton algorithm is then written:

$$\widehat{T}_{d,j+1} = \widehat{T}_{d,j} - \left[J''(\widehat{T}_d)\right]^{-1}\left(J'(\widehat{T}_d) - \beta_j\eta\right) \quad (52)$$

where β_j is the variance of the random variable η. This variance is computed using $\beta_j = \beta_0 J(\widehat{T}_d)$, with β_0 a positive constant chosen sufficiently large. The idea is if the quadratic criterion becomes smaller, the stochastic perturbation $\beta_j\eta$ becomes weaker and the algorithm tend to be closer to the classical Gauss-Newton algorithm. The algorithm used has the form:

(1) Let j=0. Set β_0, the initial estimate $\widehat{T}_{d,0}$ and an upper bound of time delays $\overline{T}_d$.
(2) Compute prefiltered signals that are part of $\Phi_f(\widehat{T}_d)$, with the low-pass filter (36).
(3) Set $\beta_j = \beta_0 J(\widehat{T}_d)$ using (47).
(4) Perform the following
  (a) Compute
  $$\Delta\widehat{T}_{d,j+1} = -\left[J''(\widehat{T}_{d,j})\right]^{-1}\left(J'(\widehat{T}_{d,j}) - \beta_j\eta\right).$$
  (b) Compute $\widehat{T}_{d,j+1} = \widehat{T}_{d,j} + \Delta\widehat{T}_{d,j+1}$.
  (c) Check if $0 \leqslant \widehat{T}_{d,j+1} \leqslant \overline{T}_d$. If not, let $\Delta\widehat{T}_{d,j+1} = 0.5\Delta\widehat{T}_{d,j+1}$ and go back to (b).
  (d) Check if $J(\widehat{T}_{d,j+1}) \leqslant J(\widehat{T}_{d,j})$. If not, let $\Delta\widehat{T}_{d,j+1} = 0.5\Delta\widehat{T}_{d,j+1}$ and go back to (b).
(5) Terminate the algorithm if the stopping condition is satisfied. Otherwise, let $j = j+1$ and go back to step 3.
(6) Finally, by substituting the final value of $\widehat{T}_d$, the linear parameter vector $\widehat{\vartheta}$ can be estimated by the LS method given in (46).

5. SIMULATION RESULTS

Consider a third order system with a zero, defined by:

$$G(p) = \frac{-1+2p}{(p+2)\left(p^2+2\xi\omega_n p+\omega_n^2\right)}e^{-T_d p} \quad (53)$$

$$G(p) = \frac{-b_0+b_1 p}{1+a_1 p+a_2 p^2+a_3 p^3}e^{-T_d p} \quad (54)$$

with

$$\begin{array}{lll} \xi = 0.2; & \omega_n = \sqrt{3}\ \text{rad/s}; & \\ b_0 = 0.167; & b_1 = 0.333; & a_1 = 0.731; \\ a_2 = 0.449; & a_3 = 0.167; & T_d = 1.3\ \text{s}. \end{array} \quad (55)$$

The system is excited with a Pseudo Random Binary Signal (PRBS). An additive white noise is added to the output with a Signal to Noise Ratio (SNR) of 15 dB. This SNR is defined as:

$$SNR = 10\log_{10}\left(\frac{var\,(y_0)}{var\,(y-y_0)}\right) \quad (56)$$

where y_0 is the noiseless output and y is the noisy output. The signals used for identification are presented in figure 2.

For the closed-loop simulations, the controller is tuned using the direct method, with the parameter values obtained with the CTOE algorithm. With this approach the desired closed loop transfer function model G_{CL} is chosen to have the following form:

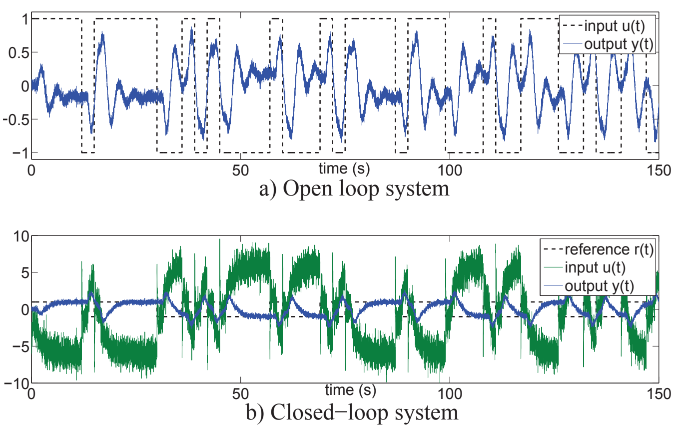

Figure 2. Signal used for identification in a) open loop and b) closed-loop

$$G_{CL} = \frac{-6\left(-1+2p\right)}{\left(p+2\right)\left(p^2+2\times 1.5\times \sqrt{3}p+3\right)} e^{-T_d p} \quad (57)$$

Where it can be seen that G_{CL} have the same unstable zero and the same time delay value (as in $G(p)$) but the static gain is now equal to one and the desired closed loop damping factor is now equal to 1.5 (well damped poles). Then, the controller is computed using the formula:

$$C(p) = \frac{1}{G(p)}\left[\frac{G_{CL}(p)}{1-G_{CL}(p)}\right] \quad (58)$$

To obtain a finite dimension controller $C(p)$, the following second order Padé approximation of the time delay T_d was used:

$$e^{-T_d p} = \frac{1-\frac{1}{2}T_d p+\frac{1}{12}\left(T_d p\right)^2}{1+\frac{1}{2}T_d p+\frac{1}{12}\left(T_d p\right)^2} \quad (59)$$

With this approximation the controller becomes :

$$C(p) = \frac{-256-353p-272p^2-144p^3-43.9p^4-6p^5}{236p+57.5p^2+66p^3+11.8p^4+p^5} \quad (60)$$

100 Monte-Carlo simulations are conducted to analyze the statistical properties of the CTOE and CTCLOE proposed algorithms. The results are presented in terms of the mean and the standard deviation. The parametric distance, computed using $D = \sqrt{\sum_{l=1}^{n+m+1}\sum_{i=1}^{100}\left(\frac{\Theta_l-\widehat{\Theta}_{l,i}}{\Theta_l}\right)^2}$ is also presented.

where i is the index of the Monte-Carlo simulation, and l the index of the parameter vector Θ.

The sampling period chosen is $T_s = 10$ ms. The initial parameter for the proposed algorithm is $\mu = 0.1$; for the GSEPNLS method, the initial variance of the stochastic variable is $\beta_0 = 10^5$, as proposed in Yang et al. (2007). The initial value of T_d is 5 s and the parameter for the filter (36) is chosen as $\lambda = 0.2$ which corresponds approximately to the bandpass of the system.

The system is firstly identified in open loop. The simulation procedure is to obtain an initial estimate of the parameters using the GSEPNLS method. Then, the proposed output error methods start with this initial estimate. The simulation results are summarized in table 1.

With CTOE and CTCLOE the parameter estimation quality is very similar and the estimated system are unbiased.

6. CONCLUSION

This paper has presented two time domain output error algorithms called CTOE and CTCLOE to identify a continuous-time linear system with time delay, in open or closed-loop with very good statistical properties. It has been also proposed a modified version of the GSEPNLS to initialize them.

Table 1. Simulation results of the 100 Monte-Carlo runs

Parameter	GSEPNLS	CTOE	CTCLOE
$\widehat{b}_0$(-0.167)	-0.1523	-0.1666	-0.1668
	±0.0010	±0.0005	±0.0005
$\widehat{b}_1$(0.333)	0.2835	0.3333	0.3333
	±0.0016	±0.0007	±0.0010
$\widehat{a}_1$(0.731)	0.4375	0.7309	0.7307
	±0.0039	±0.0032	±0.0046
$\widehat{a}_2$(0.449)	0.4360	0.4488	0.4485
	±0.0012	±0.007	±0.0017
$\widehat{a}_3$(0.167)	0.0718	0.1667	0.1665
	±0.0020	±0.0012	±0.0021
$\widehat{T}_d$(1.3)	1.3985	1.2999	1.3005
	±0.0081	±0.0024	±0.0037
D	7.2	0.1	0.15
Average of iterations	23	10	11

REFERENCES

Ahmed, S., Huang, B., and Shah, S. (2006). Parameter and delay estimation of continuous-time models using a linear filter. *Journal of Process Control*, 16(4), 323 – 331.

Baysse, A., Carrillo, F.J., and Habbadi, A. (2011). Time domain identification of continuous-time systems with time delay using output error method from sampled data. In *18th IFAC World Congress*, volume 18. Milano, Italy.

Carrillo, F.J., Baysse, A., and Habbadi, A. (2009). Output error identification algorithms for continuous-time systems operating in closed-loop. In *15th IFAC Symposium on System Identification, SYSID 2009*, volume 15 of *System Identification*. St Malo, France.

Garnier, H. and Wang, L. (eds.) (2008). *Identification of Continuous-time Models from Sampled Data*. Advances in Industrial Control. Springer.

Hwang, S.H. and Lai, S.T. (2004). Use of two-stage least-squares algorithms for identification of continuous systems with time delay based on pulse responses. *Automatica*, 40(9), 1561 – 1568.

Landau, I.D. and Karimi, A. (1997). Recursive algorithms for identification in closed loop: A unified approach and evaluation. *Automatica*, 33(8), 1499 – 1523.

Liu, M., Wang, Q.G., Huang, B., and Hang, C.C. (2007). Improved identification of continuous-time delay processes from piecewise step tests. *Journal of Process Control*, 17(1), 51 – 57.

Ngia, L. (2001). Separable nonlinear least-squares methods for efficient off-line and on-line modeling of systems using kautz and laguerre filters. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 48(6), 562 –579.

Wang, Q.G., Guo, X., and Zhang, Y. (2001). Direct identification of continuous time delay systems from step responses. *Journal of Process Control*, 11(5), 531 – 542.

Yang, Z.J., Iemura, H., Kanae, S., and Wada, K. (2007). Identification of continuous-time systems with multiple unknown time delays by global nonlinear least-squares and instrumental variable methods. *Automatica*, 43(7), 1257 – 1264.