2013 American Control Conference (ACC)
Washington, DC, USA, June 17-19, 2013

Optimal Time Delay Estimation for System Identification

Vitor Alex Oliveira Alves, Rodrigo Juliani Correa de Godoy and Claudio Garcia

Abstract— **This paper presents a novel strategy for optimal time delay estimation for system identification purposes. A brief review of classical time delay estimation techniques is presented and the problem of time delay estimation is further inspected from a system identification perspective. A solution based on optimization of the fit index is then proposed to find the best time delay, so that a given model structure with a certain order and adjusted for an identification dataset best represents the true system. A set of examples illustrates and validates the application of the proposed technique.**

I. INTRODUCTION

The goal of system identification is to derive a mathematical model that fairly describes the dynamics of the process under study. Under the Prediction Error Method (PEM) framework, the general polynomial model structure employed in linear system identification is [1]

$$A(q)y(t) = \frac{B(q)}{F(q)}u(t) + \frac{C(q)}{D(q)}e(t), \tag{1}$$

$$B(q) = \sum_{i=1}^{n_b+n_k} b_i q^{-i} \ , \ \ b_i = 0 \ \ \forall\, i \leq n_k. \tag{2}$$

In (1) and (2) q is the forward shift operator, $u(t)$ and $y(t)$ are the input and output signals, respectively, and $e(t)$ is a white noise signal. One or more of the polynomials $A(q)$, $B(q)$, $C(q)$, $D(q)$ and $F(q)$ in (1) can be fixed to unity, deriving some well known model structures such as ARX, ARMAX, OE and BJ [1].

Due to the system time delay, the dynamics from $u(t)$ to $y(t)$ contains a delay of n_k samples, so some leading coefficients of $B(q)$ are zero [1] and this polynomial can be written as in (2). It is important to stress that the n_k parameter *does not* correspond exactly to the true system time delay. It is merely a vehicle to obtain a reliable model, emulating the effect of the time delay in the dynamics from $u(t)$ to $y(t)$. In fact, there are three different types of time delay. The first one is the true process time delay, τ_d. The second one is the most suitable time delay, n_{k_0}, for a given model structure so that such model best represents the process dynamics. Finally, the third one is the value of n_k, denoted by n_k^*, that gives the best results for an identification procedure. Hence, one can not assume to find the best value for n_k by determining the true time delay, but the best n_k depends on the selected model structure and parameter estimation algorithm applied in the identification procedure for a given dataset – see Section III.

In the described scenario, this paper analyses the influence n_k in the identification process and its relationship with the true system time delay τ_d and also with the best theoretical time delay n_{k_0} for a given structure. A novel technique for finding the optimal n_k^* is then proposed. Examples addressing the discussed issues and the application of the proposal are presented. Comparisons between often used time delay estimation techniques and the proposed strategy are drawn.

The remainder of this paper is organized as follows. Section II briefly presents some techniques used to estimate the time delay. Two of them aim to find τ_d, another tries to determine n_k^* by exhaustive search on a relative small set. In Section III the influence of n_k in the identification procedure is inspected in detail. A novel approach for choosing n_k^*, in the fit index sense [2], is presented in Section IV. Section V shows application examples and performance comparison analysis. The conclusions are drawn in Section VI.

II. TIME DELAY ESTIMATION TECHNIQUES

There are several methods in the specialized literature for the estimation of time delays, see survey papers [3], [7], [4], [5] and [6]. Such studies focus on the identification of the *real* process time delay τ_d. In this paper a different approach is proposed, where a model delay n_k^* that optimizes the model behavior is sought. Nevertheless, it is useful to compare the results obtained by some of these time delay estimation methods with those derived by the strategy here proposed (see Section IV). Of direct interest are the methods presented by Björklund [7] in a thorough investigation of the time delay estimation problem. In his work, such problem is defined by the estimation of τ_d – the true time delay – in

$$y(t) = G(p)u(t) + v(t) = G_r(p)u(t-\tau_d) + v(t), \tag{3}$$

in which $G_r(p)$ is a continuous time-invariant linear system without time delay and p is the derivative operator.

Several techniques were analysed and a few more proposed in [7]. They can be divided into three categories: time delay approximation methods, explicit time delay parameter methods and area, moment and higher-order statistics methods. The brief review of subsections II-A and II-B refers to a subclass of the time delay approximation methods, known as *time domain approximation methods*. Subsection II-C discuss the use of the MATLAB® routine *delayest* in the estimation of an adequate value for n_k – an approach that matches the basic concepts of the novel technique in Section IV.

Research project sponsored and financially supported by Petrobras SA under process 0050.0058968.10.9/2010.

V. A. O. Alves is with Mauá School of Engineering of the Mauá Institute of Technology – IMT, Praça Mauá 01, São Caetano do Sul, SP, Brazil - CEP 09580-900. vitoralex.alves@maua.br

R. Juliani C. G. and C. Garcia are with Dept. of Telecommunications and Control Engineering, Polytechnic School of the University of São Paulo, Av. Prof. Luciano Gualberto, 158, São Paulo, SP, Brazil - CEP 05508-970. rodrigo.juliani@usp.br, clgarcia@lac.usp.br

A. Impulse Response and Cumulative Sum (CUSUM)

The time delay can be interpreted as the time needed for the impulse response of a system to start [7]. The Cumulative Sum (CUSUM) method [8] aims at detecting the start of such response. Therefore, it requires an estimate $\hat{g}(t)$ of the impulse response $g(t)$. In this study, such task is performed using three different approaches: cross-correlation analysis, FIR model estimation using MATLAB® routine *impulseest* and optimal high order ARX model estimation [9], [10].

The cross-correlation analysis is based on the fact that the impulse response can be expressed as $g(\tau) = R_{yu}(\tau)/\lambda$ when the input signal $u(t)$ is white noise with autocorrelation function $R_u(\tau) = E\{u(t+\tau)u(t)\} = \lambda\delta(\tau)$ [1], in which λ is the power of the input signal. The estimates of the cross-correlation function $R_{yu}(\tau) = E\{y(t+\tau)u(t)\}$ and the input signal power λ can be obtained from input-output data:

$$\hat{R}_{yu}(\tau) = \frac{1}{N}\sum_{t=\tau+1}^{N} y(t)u(t-\tau) \ \ , \ \ \hat{\lambda} = \frac{1}{N}\sum_{t=1}^{N} u^2(t). \quad (4)$$

If the input is white noise, an estimate of the impulse response is then

$$\hat{g}(\tau) = \hat{R}_{yu}(\tau)/\hat{\lambda}. \quad (5)$$

For system identification purposes, $u(t)$ is not a white signal. In fact, $u(t)$ must be carefully designed for the system to be excited in the frequency band of interest for the considered application, specially when the process is ill-conditioned [11]. Typical excitation (input) signals applied in identification experiments, such as pseudo-random binary sequences (PRBS) and generalized binary noise (GBN) [12], can be designed to present *quasi*-white characteristics in a particular frequency band. Hence, the cross-correlation approach can be employed under such conditions [13].

Another way of estimating the impulse response of a system is to estimate a parametric FIR model, i.e. $y(t) = B(q)u(t)$, through PEM. The parameter vector $\hat{\theta}$ contains the coefficients of $B(q)$. The model will be a linear regression and the estimate is given by the least-squares expression [1]

$$\hat{\theta} = \left[\frac{1}{N}\sum_{t}\phi(t)\phi^T(t)\right]^{-1}\left[\frac{1}{N}\sum_{t}\phi(t)y(t)\right] = \\ = R^{-1}(N)f(N), \quad (6)$$

with $\phi(t) = [u(t), u(t-1), ...]$. This is essentially the same as (5). If the input signal is white, then $R(N)$ and $f(N)$ in (6) are $R(N) = \hat{\lambda}I$ and $f(N) = [\hat{R}_{yu}(0) \ \hat{R}_{yu}(1) \ \cdots]^T$. Therefore, $\hat{\theta} = [\hat{g}_0 \ \hat{g}_1 \ \cdots]^T$ is exactly the same as $\hat{g}(\tau) = [\hat{g}_0 \ \hat{g}_1 \ \cdots]^T = [\hat{g}(0) \ \hat{g}(1) \ \cdots]^T$ in (5). Once again, when PRBS and GBN input signals are used in the identification experiment, the same results hold [13]. In this study, (6) is solved with MATLAB® *impulseest*. This routine internally determines a suitable number of free parameters for θ.

Finally, the third approach in the estimation of the impulse response applies an optimal high order ARX model $A(q)y(t) = B(q)u(t) + e(t)$, with order (number of free coefficients in $A(q)$ and $B(q)$) obtained by the optimization of the fit index (10). See details in [9], [10].

Fig. 1 shows the structure of time delay estimation in the time domain [7], the category of methods in which the CUSUM detector is included.

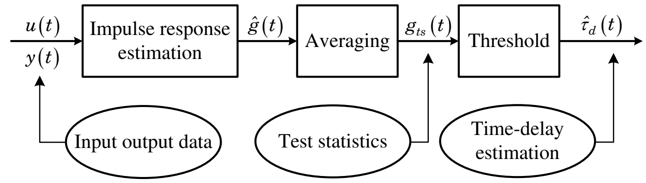

Fig. 1. Steps in time delay estimation in the time domain.

As mentioned before, the first step is the estimation of the impulse response and its uncertainty. Afterwards, an averaging operation is performed to reduce the noise, resulting in the test statistics $g_{ts}(t)$, which is then thresholded to detect the start of the impulse response. Finally, the time delay estimate $\hat{\tau}_d$ derived from the CUSUM method is equal to the estimate of the change time $\hat{k}$ of the impulse response $g(t)$ – see Table I. There are two user-selected parameters: the *drift* $\nu(t)$ and the *threshold* $h(t)$. The latter can be fixed (data independent) or relative, i.e., it depends on the uncertainty of the impulse response estimate. The choice of the threshold is a key feature of CUSUM. If it is proportional to the uncertainty of the impulse response, a specified risk for false alarm is determined. On the other hand, the reduction of such risk (by increasing the threshold) causes a reduction in the probability of detection. In this study, in accordance with [7], the drift and threshold parameters employed are $\nu(t) = \nu = \hat{g}_{std}(0)$ and $h(t) = h = 3\,\hat{g}_{std}(0)$. The subscript *std* denotes the standard deviation of the coefficient $\hat{g}(0)$. The complete algorithm is shown in Table I [7].

TABLE I

CUSUM ALGORITHM: $\hat{\tau}_d = \hat{k}$.

Design parameters: Drift $\nu(t)$ and threshold $h(t)$.
Output: Detection time t_a and the estimate of the change time $\hat{k}$.
Input: Impulse response estimate $\hat{g}(t)$.
Internal variable: Test statistics $g_{ts}(t)$.

1. $t = 0$, $g_{ts}(-1) = 0$
2. $g_{ts}(t) = g_{ts}(t-1) + \hat{g}(t) - \nu(t)$
3. If $g_{ts}(t) < 0$ then $g_{ts}(t) = 0$, $\hat{k} = t$
4. If $g_{ts}(t) > h(t) > 0$ then $t_a = t$, goto 6
5. $t = t + 1$, goto 2
6. The detection time is t_a and the estimate of the change time is $\hat{k}$

B. Separating Time Delay from Dynamics

This subsection briefly describes a technique that separates the time delay from the dynamics of the system, as seen in (3). A detailed discussion can be found in [7]. The method requires the impulse response $g(t) = g_r(t) * \delta(t - \tau_d)$. The Fourier transform of $g(t)$ is $G(j\omega) = G_r(j\omega) \cdot e^{-j\omega\tau_d}$ and its real part is given by:

$$G_{fr}(j\omega) = A_{fr}(j\omega)\cos(\tau_d\omega + \varphi(j\omega)), \quad (7)$$

$$A_{fr}(j\omega) = \sqrt{[\text{Re}\{G_r(j\omega)\}]^2 + [\text{Im}\{G_r(j\omega)\}]^2}, \quad (8)$$

$$\varphi(j\omega) = \text{arctg}\left(\frac{\text{Im}\{G_r(j\omega)\}}{\text{Re}\{G_r(j\omega)\}}\right). \quad (9)$$

As can be seen in (7), $G_{fr}(j\omega)$ depends on $A_{fr}(j\omega)$, which is modulated with a sinusoid with "frequency" τ_d and phase shift $\varphi(j\omega)$ [7]. One way to estimate such "frequency" is to analyse the "spectrum" of $G_{fr}(j\omega)$ and look for peaks at "frequency" τ_d, which is the time delay. Typically, $G_{fr}(j\omega)$ presents oscillations with relatively small amplitudes. To increase the effect of the oscillation, Björklund [7] suggests to take the logarithm of the absolute value of $G_{fr}(j\omega)$ before the determination of its "spectrum". In this case, one needs to look for peaks at "frequency" $2\tau_d$. If the input signal $u(t)$ is not white, and especially when it is oscillatory (i.e. there are dominating frequencies), then the impulse response will have several local maxima [14], [15].

C. MATLAB® delayest

The *delayest* algorithm estimates the n_k that minimizes the loss function [1] associated with an ARX model $A(q)y(t) = B(q)u(t)+e(t)$ with fixed number of parameters in $A(q)$ $[n_a]$ and $B(q)$ $[n_b]$, chosen by the user. It employs an exhaustive search strategy by creating several ARX models – derived for the chosen order and with n_k varying in a specified domain – and returning the n_k value for which the loss function is minimized. The major drawbacks of *delayest* are: *i)* only ARX model structures are contemplated; and *ii)* exhaustive search solution results in large computational time.

III. INFLUENCE OF n_k IN SYSTEM IDENTIFICATION

This section addresses the problem of how the n_k parameter affects the identification results. This is done through examples in which it is observed how the model quality, measured in terms of the fit index [2], varies with the employed n_k. Such index is defined by

$$\text{fit} = 100 \cdot \left(1 - \frac{\|\hat{y} - y\|}{\|y - \bar{y}\|}\right) (\%), \tag{10}$$

in which y denotes the vector of measured outputs, $\hat{y}$ represents the vector of the model predicted outputs and $\bar{y}$ is the mean value of y. It is important to stress that in this paper y in (10) is obtained in the absence of noise, i.e., y is a "clean" version of the process output signal acquired in the identification test. It generates the so called *noise-free fit*, in opposition to the *noisy fit*, calculated with the original process data [9], [10]. This is, of course, a theoretical evaluation that indicates how well the process model prediction fits to the real system behavior not corrupted by noise.

Hereafter, *model order* represents a vector composed by the orders of the polynomials of the selected model structure and n_k is the number of fixed null parameters in $B(q)$.

Example 1 considers an identification test in which the model is derived with the correct order and different n_k values under distinct SNR (Signal-to-Noise Ratio) scenarios. Example 2 is similar to the previous one, but with fixed SNR and a model order different from the correct one. Finally, Example 3 presents how the change of the identification method (or the model structure) influences the effect of the n_k parameter in the model quality. The following process model is employed in all the examples:

$$G(s) = \frac{-5 \cdot (s - 0.2)}{(10s + 1) \cdot (5s + 1)} \cdot \mathrm{e}^{-10s}. \tag{11}$$

The discrete form of (11), employing a zero order hold with unitary sampling time is given by:

$$G(z) = \frac{-0.07705\, z^{-11} + 0.0943\, z^{-12}}{1 - 1.724\, z^{-1} + 0.7408\, z^{-2}}. \tag{12}$$

From (12) it can be seen that the correct model order is $[n_a\, n_b]^T = [2\ 2]^T$ and $\tau_d = 11$.

All identification tests are performed exciting the process model (11) with a PRBS signal and acquiring 1000 points (600 points for identification and the complete set for model validation). The SNR is defined in terms of variance.

Example 1: An ARX model structure is employed in the process identification using the correct order. The n_k parameter varies from 0 to 50. The fit index is calculated for *k*-step ahead predictions, $k = 1, 2, \cdots, 50, \infty$. The results for SNR = ∞, 30 and 3 are shown in Figures 2, 3 and 4.

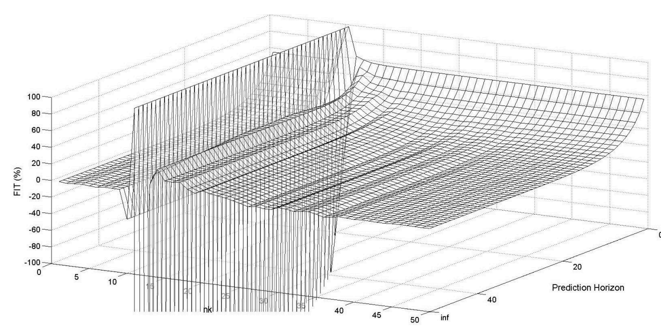

Fig. 2. ARX $[2\ 2]^T$ - Noise-free fit values for ID test under SNR = ∞.

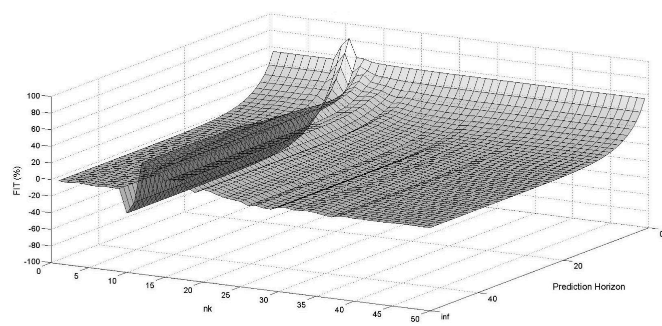

Fig. 3. ARX $[2\ 2]^T$ - Noise-free fit values for ID test under SNR = 30.

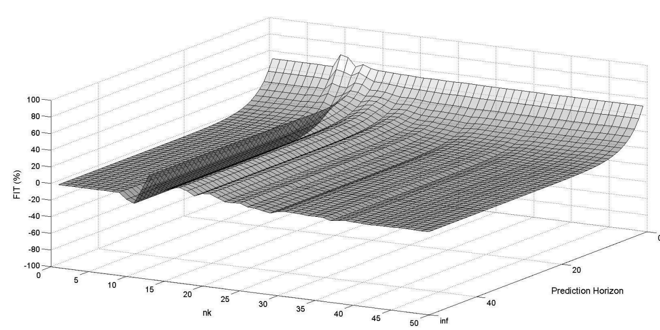

Fig. 4. ARX $[2\ 2]^T$ - Noise-free fit values for ID test under SNR = 3.

The inspection of Figures 2, 3 and 4 reveals that the model quality, measured in terms of the fit index, greatly varies with n_k. This alone justifies the search for its best possible value.

A close inspection of the figures reveals that n_k^*, the optimal n_k, can result not equal to the real value n_{k_0}, that, in this case, is equal to τ_d since the correct order is employed. It is also observed that close to the optimal there is a strong decay in the fit index, what corroborates the importance of finding n_k^* for each dataset. Table II illustrates these statements for an infinite step-ahead prediction horizon. Such case is selected because in it there is the strongest influence of n_k, since only the process model (which is directly affected by n_k) is employed in the prediction procedure.

TABLE II
NOISE-FREE FIT VALUES FOR DIFFERENT SNR.

	SNR ∞		**SNR 30**		**SNR 3**	
n_k	n_{k_0} 11	n_k^* 11	n_{k_0} 11	n_k^* 12	n_{k_0} 11	n_k^* 13
FIT (%)	100	100	-14.5	34.3	-11.1	25.0

As seen in Table II, the optimal value n_k^* results different from the true one in the SNR 30 and 3 scenarios. In fact, Figures 2, 3 and 4 show that a small deviation in the value of n_k can result in a great model quality loss. For SNR ∞, the worst fit value ($-8.3204 \cdot 10^8\%$) is obtained with $n_k = n_{k_0} + 1 = 12$; for SNR 30, the worst fit (-28.7%) occurs when $n_k = n_{k_0} - 1 = 10$; and for SNR 3, $n_k = n_{k_0}$ provides the worst model (fit = -11.1%). ■

Example 2: In this example, it is adopted $[n_a\, n_b]^T = [1\ \ 1]^T$, an incorrect order for (12). For this chosen order $n_{k_0} = 18$ is the value that best approximates the true step response of the process.

Using the same datasets from Example 1, ARX models with the adopted order and n_k varying from 0 to 50 were identified. Fig. 5 shows the fit values for SNR = 30.

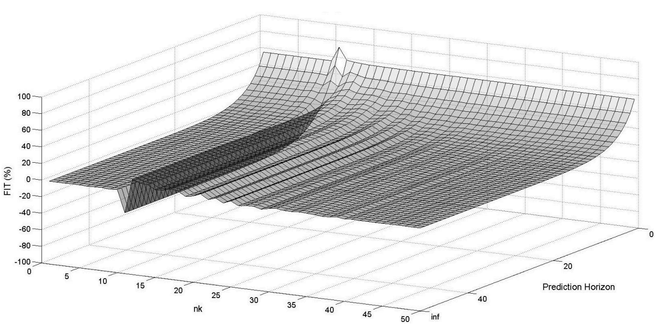

Fig. 5. ARX $[1\ 1]^T$ - Noise-free fit values for ID test under SNR = 30.

Comparing figures 3 and 5 it can be seen that a mismatch in the order causes a decay in the model quality, particularly in the range of n_k between 10 and 20, as expected. Nevertheless, the previously observed n_k influence remains. So, the value of n_k^* depends on the dataset, the SNR and also on the model order. These conclusions are summarized in Table III, in which it is considered infinite step-ahead predictions.

In Table III it can be seen that, with the same dataset employed in Example 1, n_k^* results the same for different model orders (which can not be considered the general case). However, an incorrect order provides a worse fit compared to the one achieved with correct order. Also, the fact that $n_k^* \neq n_{k_0}$ is once again observed. ■

TABLE III
NOISE-FREE FIT VALUES FOR DIFFERENT MODEL ORDERS.

$[n_a\, n_b]^T$	$[1\ 1]^T$		$[2\ 2]^T$	
n_k	n_{k_0} 18	n_k^* 12	n_{k_0} 11	n_k^* 12
FIT (%)	-26.6	14.9	-14.5	34.3

Example 3: In this example an OE structure (identical to the discrete process (12)) is applied with the correct order $[n_b\, n_f]^T = [2\ 2]^T$. The identification procedure is analogous to the previous ones, with SNR 30 and 3. Fig. 6 shows the fit values for each identified model with SNR 30. Since the OE model structure does not have a disturbance model, its *k*-step ahead predictions are the same for any horizon.

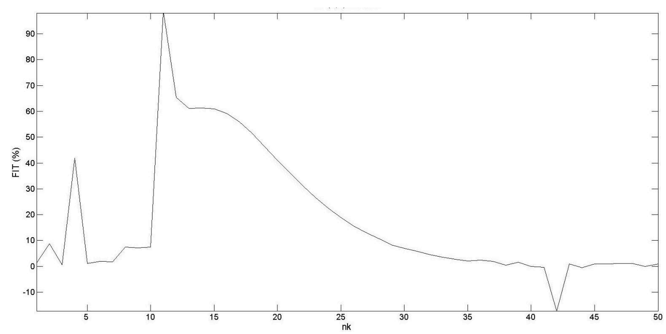

Fig. 6. OE $[2\ 2]^T$ - Noise-free fit values for ID test under SNR = 30.

Once again, a difference in the influence of n_k is observed when compared to the one in Figure 3. Therefore, n_k^* is dependent on the dataset, the SNR, the model order and the model structure. This effect is presented in Table IV.

TABLE IV
NOISE-FREE FIT VALUES FOR DIFFERENT MODEL STRUCTURES.

	ARX				**OE**			
	SNR 30		**SNR 3**		**SNR 30**		**SNR 3**	
n_k	n_{k_0} 11	n_k^* 12	n_{k_0} 11	n_k^* 13	n_{k_0} 11	n_k^* 11	n_{k_0} 11	n_k^* 11
FIT(%)	-14.1	34.3	-11.1	25.0	98.1	98.1	57.4	57.4

In Table IV it can be seen that when structure and order are correct, it is most likely that $n_k^* = n_{k_0}$, as in this case. But even so, the SNR affects the fit index. Another important fact is the improvement in the optimal fit value, that increases as the correct structure is employed. ■

IV. PROPOSED METHOD

The main goal of the present study is to determine what is the best value for n_k (in the fit index sense) to be assigned to $B(q)$ in the general model structure (1). Examples 1, 2 and 3 revealed the great importance of n_k, its dependence on the identification scenario and its choice effects in model quality. In this section, an optimization problem that deals with such goal is formally postulated in subsection IV-A.

Then, a solution to the problem is proposed in a two-step strategy shown in subsection IV-B.

A. Optimal Time Delay Estimation Problem

The search for an adequate n_k value associated with the model structure (1) can be formally stated as an optimization problem expressed by

$$\max_{n_k} \ \text{fit}(\hat{y}, y_{val}), \tag{13}$$

subject to:

$$\hat{y} = \text{predict}(\text{Model}, \text{Data}_{\text{val}}, k) \tag{14}$$

$$\text{Model} = \text{estimation}(\text{Data}_{\text{ident}}, \text{Order}, n_k) \tag{15}$$

$$\text{Order} = [n_a \ n_b \ n_c \ n_d \ n_f]^T \tag{16}$$

In (13) the objective function $\text{fit}(\hat{y}, y_{val})$ is defined as in (10), replacing y by y_{val}, in which the subscript *val* denotes the use of a cross-validation dataset. Equations (14)-(16) constitute the set of constraints to which the optimization problem is subject. Constraint (14) represents the k-step ahead prediction [1] of the process output employing the validation dataset. Equation (15) represents the parameter estimation procedure that derives a model with a chosen structure and order for a given identification dataset. The optimization variable n_k is determined so that (13) is maximized. Constraint (16) defines which of the polynomials $A(q)$, $B(q)$, $C(q)$, $D(q)$ and $F(q)$ of (1), and their corresponding order, form the chosen model structure.

It is important to stress that the obtained n_k^* must provide a fit index close to its global optimal, but it is irrelevant if the identified model time delay itself is close to its global optimal. It is also relevant to note that the cross-validation was applied in (13) and (14) because *"it is not so surprising that a model will be able to reproduce the estimation data, [...] but the real test is whether it will be capable of also describing fresh data-sets from the process"* [1].

B. Optimization Problem Solution

To optimize (13), subject to (14)-(16), a two-step optimization strategy is proposed. In the first step, (13) is solved via a global optimization algorithm, then, in the second step, the solution is refined employing a direct search algorithm. Both steps are described next.

1) Global Optimization Method – Simulated Annealing: The optimization problem (13) requires a large computational effort when solved by exhaustive search. Furthermore, the search domain grows exponentially with the increase in the range of variation of the optimization variable. Thus, it is desirable to find a solution that presents a fit close to the optimum in a feasible computational time. The Simulated Annealing algorithm [16], [17] is employed in this task. The name and inspiration for this algorithm come from annealing in metallurgy, a technique that involves heating and controlled cooling of a material in order to increase the size of its crystals and reduce their defects.

At each step or iteraction, the Simulated Annealing heuristic considers some neighbouring state s' of the current state s, and probabilistically decides between moving the solution to state s' or staying in state s. Such probabilities ultimately lead the solution to move to states of "lower energy". Typically, this basic step is repeated until the solution reaches a state that satisfies the objective function under some well defined criteria, or until a given computation budget has been exhausted. See a detailed discussion in [16], [17].

2) Direct Search Optimization Method – Pattern Search: The Simulated Annealing algorithm finds a solution that presents a performance near the optimum but, in some cases, is not exactly the optimum. Hence, to refine such solution, it is employed a direct search algorithm: the Pattern Search [18], [19]. In the proposed optimization strategy, this second step moves the previous solution closer to the optimum.

The previous algorithms were selected due to their widely application to optimization problems and for being available in computational tools employed in simulation, identification and optimization, such as MATLAB®. Any other optimization algorithm can be employed in the solution of (13)-(16) – in fact, other algorithms were tested, and Simulated Annealing and Direct Search were chosen for presenting a good performance for this problem.

V. APPLICATION EXAMPLES

In this section the proposed approach is applied and compared with the classical time delay estimation techniques described in Section II. Hereafter, CUSUM-CrossCorr refers to the CUSUM method with impulse response estimated by the cross-correlation analysis; CUSUM-FIR employs a FIR model (derived from *impulseest*) whose coefficients represent the impulse response and CUSUM-ARX-High uses an optimized high order ARX model for the generation of the impulse response. The "frequency domain" technique that separates the time delay from the dynamics of the process is denoted by Freq-Domain. In it, the impulse response is estimated by *impulseest*. Finally, OPT represents the proposed optimization strategy.

All techniques are applied to the five scenarios (ARX model structure with correct order and SNR ∞, 30 and 3; ARX structure with incorrect order and SNR 30; correct structure OE and order, SNR 30) presented in Examples 1, 2 and 3 of Section III. Table V shows the n_k values obtained by each of the time delay estimation techniques and the infinite step-ahead fit index achieved by the model identified with such n_k.

It can be seen in Table V that, as stated in Section III, n_k^* can be different from n_{k_0}. Also, the use of n_{k_0} can result in poor fit values, as in SNR 30 and 3 scenarios for ARX $[2\,2]^T$. From Section II one can see that all CUSUM, Freq-Domain and *delayest* methods ignore the model structure and order to be employed in the identification procedure, deriving a unique n_k value for each SNR scenario (since the dataset is always the same). The results corroborate it. The n_k values estimated by these methods can generate a poor model. Meanwhile, the optimization approach considers all features that affect how n_k influences the model quality: dataset, model structure and model order, which characterizes a more

TABLE V

COMPARISON OF TIME DELAY ESTIMATION TECHNIQUES – NOISE-FREE FIT(%) AND n_k VALUES.

	Example 1 – ARX $[n_a\, n_b]^T = [2\ 2]^T$						**Example 2 – ARX** $[n_a\, n_b]^T = [1\ 1]^T$		**Example 3 – OE** $[n_b\, n_f]^T = [2\ 2]^T$			
	SNR ∞		**SNR 30**		**SNR 3**		**SNR 30**		**SNR 30**		**SNR 3**	
	n_k	FIT	n_k	FIT	n_k	FIT	n_k	FIT	n_k	FIT	n_k	FIT
n_{k_0}	11	100	11	-14.5	11	-11.1	18	7.76	11	98.1	11	94.0
n_k^*	11	100	12	34.3	13	25.0	12	14.9	11	98.1	11	94.0
CUSUM-CrossCorr	14	23.5	14	20.3	15	2.33	14	5.27	14	61.3	15	60.6
CUSUM-FIR	14	23.5	14	20.3	14	4.96	14	5.27	14	61.3	14	61.0
CUSUM-ARX-High	14	23.5	14	20.3	15	2.33	14	5.27	14	61.3	15	60.6
Freq-Domain	12	$-\infty$	12	34.3	12	6.41	12	14.9	12	65.3	12	65.2
delayest	11	100	11	-14.5	11	-11.1	11	-26.6	11	98.1	11	94.0
OPT	11	100	13	19.1	20	11.1	21	7.60	11	98.1	11	94.0

robust approach. As can be observed, in most cases, the optimization procedure finds a n_k that generates an optimal or quasi-optimal fit index. However, since the optimized variable is the noisy fit and the assessed index is the noise-free fit, in some cases the optimized n_k for the noisy fit can result different from the optimal n_k^* for the noise-free fit. This difference occurs due to the effect of the noise in the identification process, which is aggravated by the fact that the noise whiteness hypothesis is not valid for a finite length dataset, specially for a short one, as usually employed in system identification.

VI. CONCLUSIONS

This paper dealt with the problem of time delay estimation in system identification. The concept of time delay is threefold: τ_d is the true process time delay; n_{k_0} is the most suitable time delay for a given model structure that best represents the process dynamics; n_k^* is the time delay that provides the best results (in some criterion sense) for an identification procedure.

The presented examples show that n_{k_0} is not always equal to n_k^*. In fact, the latter depends on the dataset, the SNR under which such data were acquired and the adopted model structure. Hence, it is imperative to determine n_k such that the model quality (in this paper measured by the fit index) is maximized. However, an exhaustive search can be time consuming (specially when MIMO systems are concerned). To deal with that, an optimization strategy was proposed.

In order to test the optimization validity, a simulated experiment was designed with some SNR, model structure and order scenarios. Classical time delay estimation techniques were reviewed and applied to these scenarios. The tests revealed that classical approaches do not always give good results, and the proposed strategy is promising.

Future work involves a deeper study of the method performance, the multivariable estimation problem, the study of time delay estimation in closed-loop operation and also a combined strategy for time delay estimation along with the other order parameters, once it was detected a high dependence of the best time delay with the order parameters.

ACKNOWLEDGMENTS

Authors thank the support provided by Center of Excellence for Industrial Automation Technology (CETAI) and Research Center (CENPES) of Petrobras S.A. and by Mauá Institute of Technology - IMT.

REFERENCES

[1] L. Ljung, *System Identification: Theory for the user*, Upper Saddle River, NJ: Prentice Hall PTR, 1999, 609 p.

[2] L. Jung, *The system identification toolbox: The manual*, 7th edition, The MathWorks, Inc., MA, USA: Natick, 2007.

[3] B. Ni, D. Xiao, S. L. Shah, Time delay estimation for MIMO dynamical susyems - With time-frequency domain analysis, *Journal of Process Control*, vol. 20, 2010, pp. 83-94.

[4] S. Ahmed, *Parameter and delay estimation of continuous-time models from uniformly and non-uniformly sampled data*, Ph.D. Thesis, Edmonton, T6G 2G6, University of Alberta, 2006.

[5] J. P. Richard, Time-delay systems: an overview of some recent advances and open problems, *Automatica*, vol. 39, 2003, pp. 1667-1694.

[6] J. D. Chen, J. Benesty, Y. T. Huang, Time delay estimation in room acoustic environments: an overview, *EURASIP Journal on Applied Signal Processing*, 2006, pp. 1-19.

[7] S. Björklund, *A Survey and Comparison of Time-Delay Estimation Methods in Linear Systems*, Licentiate Thesis no. 1061, Department of Electrical Enginnering, Linköping University, Sweden, 2003.

[8] F. Gustafsson, *Adaptive Filtering and Change Detection*, West Sussex, England: Wiley, 2000, 492 p.

[9] V. A. O. Alves, R. Juliani C. G. and C. Garcia, "Searching the Optimal Order for High Order Models - SISO Case", *in IEEE Multi-Conference on Systems and Control*, Dubrovnik, Croatia, 2012, pp. 843-848.

[10] V. A. O. Alves, R. Juliani C. G. and C. Garcia, "Searching the Optimal Order for High Order Models - MIMO Case", *in IEEE Multi-Conference on Systems and Control*, Dubrovnik, Croatia, 2012, pp. 98-103.

[11] A. S. R. Kuramoto, O. R. Vaillant and C. Garcia, "Effectiveness of signal excitation design methods for ill-conditioned process identification", *in 8th IFAC Symposium on Advanced Control of Chemical Processes - AdChem 2012*, Singapore, 2012, pp. 337-342.

[12] H. J. A. F. Tulleken, Generalized binary noise test-signal concept for improved identication-experiment design. *Automatica*, vol. 27, no. 1, pp. 37-49, 1990.

[13] L. A. Aguirre, *Introduction to System Identification - Linear and Non-Linear Techniques Applied to Real Systems*, Belo Horizonte, Brazil: Editora UFMG, 2007, 729p, *in Portuguese*.

[14] R. Pupeikis, "Recursive estimation of the parameters of linear systems with time delay", *in 7th IFAC/IFORS Symposium on Identification and System Parameter Estimation*, York, UK, 1985, pp. 787-792.

[15] D. H. Johnson and D. E. Dudgeon, *Array Signal Processing. Concepts and Techniques*, Prentice Hall, 1993.

[16] S. Kirkpatrick, C. D. Gelatt and M. P. Vecchi, Optimization by Simulated Annealing, *Science*, vol. 220 n. 4598, 1983, pp. 671-680.

[17] V. Cerny, Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm, *Journal of Optimization Theory and Applications*, vol. 45, 1985, pp. 41-51.

[18] R. Hooke and T. A. Jeeves, 'Direct search' solution of numerical and statistical problems, *Journal of the Association for Computing Machinery (ACM)*, vol. 8, n. 2, 1961, pp. 212-229.

[19] W. C. Davidon, Variable metric method for minimization, *SIAM Journal on Optimization*, vol. 1, n. 1, 1991, pp. 1-17.