Available online at www.sciencedirect.com

ScienceDirect

Journal of the Franklin Institute 350 (2013) 3128–3148

Journal of The Franklin Institute

www.elsevier.com/locate/jfranklin

Identification and control of delayed SISO systems through pattern search methods

J. Herrera[a,c,*], A. Ibeas[a], S. Alcántara[a], M. de la Sen[b], S.I. Serna-Garcés[c]

[a]*Departament de Telecomunicació i d'Enginyeria de Sistemes, Escola d'Enginyeria, Universitat Autònoma de Barcelona, 08193 Bellaterra, Barcelona, Spain*

[b]*Departamento de Electricidad y Electrónica, Facultad de Ciencia y Tecnología, Universidad del País Vasco, P.O. 644, 48080 Bilbao, Vizcaya, Spain*

[c]*Instituto Tecnológico Metropolitano, Medellín, Colombia*

Received 28 February 2012; received in revised form 12 February 2013; accepted 30 June 2013
Available online 12 July 2013

Abstract

In this study, an approach to identify and control stable, unstable and integrating systems with unknown delays, framed on the generalized Pattern Search Method is presented. The proposed method inherits the global convergence properties of the generalized Pattern Search Method, allowing us to make a stability analysis of the proposed approach and delay identification capabilities. The proposed approach identifies the delay and guarantees closed-loop stability, which could be a difficult task since in unstable and integrating cases, open-loop experiments are not feasible. Simulation examples show the usefulness of the proposed strategy proving that the scheme is capable of identifying the delay and stabilizing the system even with long delay.

1. Introduction

An open problem in the control community is the control of delayed systems, specially, when the delay is unknown or time varying. The external delay in a process causes the output signal being delayed with respect to the input. If a control strategy has to be designed for this system, the presence of the delay makes it a more difficult task, especially when the rational part of the system is unstable. Nevertheless, it is common to deal with systems with delay since it is usual

*Corresponding author at: Departament de Telecomunicació i d'Enginyeria de Sistemes, Escola d'Enginyeria, Universitat Autònoma de Barcelona, 08193 Bellaterra, Barcelona, Spain. Tel.: +34 935814799; fax: +34 935814031.

E-mail addresses: jorgeaurelio.herrera@uab.es, jorgeherrera@itm.edu.co (J. Herrera).

0016-0032/$32.00
http://dx.doi.org/10.1016/j.jfranklin.2013.06.022

to model higher-order systems as systems of first or second order plus dead time [20]. Various strategies have been used to counteract the effect of the delay, where the tuning of PID controllers is perhaps the most widely used. In [1,28] different tuning rules for stable/unstable first order plus dead time (FOPDT) and second order plus dead time (SOPDT) systems can be found. The disadvantage of such techniques is that they only work well when the delay is small compared with the time constant of the system [30]. For systems where the delay is dominant, different control strategies such as delay compensation schemes (DCS) should be used, being the most well-known the Smith Predictor and its modifications to include unstable systems [23].

Thus, an integrated design procedure for a modified Smith Predictor (MoSP) for unstable processes is developed in [25]. This work used a root locus technique to guarantee the asymptotic stability of the system. A partial optimization of the method proposed by [25] is presented in [26]. In this work, the obtained control shows better stability and a faster settling time than the previous one. Following the same theme than the previous works, [17] proposed a MoSP and a simple but very effective controller design procedure, that offers higher performance in both closed-loop tracking and robustness. Some drawbacks of the approaches mentioned above are that they use graphical methods to obtain an optimal tuning. This procedure has to be applied offline to the nominal model of the system, known beforehand. Hence, they cannot be applied online within an adaptive control setting, for instance if the delay of the process is unknown (i.e. there is uncertainty in the delay). An additional shortcoming of the mentioned approaches is the lack of robustness to small uncertainties in the delay.

Optimization methods are a useful tool used by the process control community, to deal with the problem of delayed systems with uncertainty in the delay. Different works have been focused in this direction. For instance, design of optimal controllers, based on the particle swarm optimization, genetic algorithms and H_2 optimization can be found in [11,19,24], respectively. Thus, in [5], an online controllers optimization for thermoacoustic instabilities of gas turbine combustors, using evolutionary computation is proposed. Finally, A LMI-based approach to treat the asymptotic stability, is developed in [27]. The drawback of these methods is that, in most cases, the optimization is done offline, making impossible a real time implementation. An additional shortcoming for the online optimization methods is that these methods do not have analytical convergence properties, which in some cases, difficult the stability analysis of the closed-loop system, being this one of the most important properties to be guaranteed.

The problem we are focusing on consists in the control of a possible unstable delay system with unknown delay and known rational component. This is an adequate starting point since the closed-loop is indeed quite robust to uncertainties in the rational component (originated by approximating nominal models) but very sensitive to delay uncertainty [15,16]. Consequently, solving the problem of delay identification implies being capable of guarantying the stability of the closed loop. The delay identification is formulated as an optimization problem, which is solved using Pattern Search Method (PSM).

Generalized PSM was proposed in [29] for derivative-free unconstrained optimization of continuously differentiable convex functions. The principal advantage is that our PSM inherits the convergence properties of the generalized PSM. Thus, analytical stability properties are formulated for the closed-loop adequately and easily since previous results can be used for this purpose. Another advantage is the fact that the PSM is implemented under simple mathematical operations, which makes its implementation relatively simple.

The PSM is implemented for practical purposes through a multi-model scheme running in parallel [7,9] and this framework is a novelty both in Control Theory as in Mathematics. The multi-model scheme contains a battery of models which is updated through time by using a modification rule called exploration move in the context of PSM. Each model possesses the same rational component but a different value for the delay. After an exploratory move the algorithm compares the mismatch between the actual system and each model and selects, at each time interval, the one that best describes the behavior of the real system, providing an online delay system estimation, while simultaneously ensuring closed-loop stability.

It is worth mentioning that the PSM is an optimization method that has been used in Mathematics and Optimization Theory [4,14], but its use in Control Theory is rather limited, existing very few works using it, [6,7]. Thus, a procedure for quantifying valve stiction in control loops based on global optimization is presented in [12] where a PSM subordinated by a least squares estimator was proposed for the parameter identification. A PSM has also been applied to power systems in [21], where, an object-oriented model of the system for making prediction is proposed; to achieve this goal, a online optimization problem is solved by means of PSM. In these two works, there is neither an analysis of convergence nor frame on the PSM. This contrasts with the present work, where both issues are treated.

The paper is organized as follows. Section 2 reviews the problem formulation, including an introduction to PSM. Section 3 presents the proposed control scheme and how it casts into the generalized PSM of Section 2. The stability analysis is performed in Section 4. Simulation examples are presented in Section 5. Finally, Section 6 summarizes the main conclusions.

Notation: $x \in A$ where $x \in \mathbb{R}^n$ and $A \in \mathbb{R}^{n \times n}$ stands for $x \in Im(A)$. $v \geq 0 (v > 0)$ stands for a vector in $\mathbb{R}^n$ with all its components being non-negative (positive).

2. Problem formulation

Let us consider the following model:

$$G(s) = G^{df}(s)e^{-hs} \tag{1}$$

where $G^{df}(s)$ denotes the part containing the rational component of the system. The following assumptions will be made about the system (1):

Assumption 1. The rational transfer function $G^{df}(s)$ is realizable, stabilizable and known. □

Assumption 2. The delay between input/output lies in a known compact interval. That is, there exist two known values $\overline{h}$, $\underline{h}$ $\in \mathbb{R}$ such that $\underline{h} \leq h \leq \overline{h}$. □

Assumption 1 is feasible in many control problems where a nominal model of the system is available beforehand and it does not posses unstable pole-zero cancelations. However, the delay may be unknown or even time-varying and hence has to be estimated, [18]. Note that there is no assumption on the stability of $G^{df}(s)$, being thus potentially unstable. Assumption 2 will be used in the proposed pattern-search-based algorithm to identify the delay in Section 3 and is feasible in many practical control problems, where bounds on the delay are known. The topology used in this paper is detailed below.

2.1. Modified smith predictor

The MoSP [17] is shown in Fig. 1. This structure can control stable, unstable and integrating processes, and has three controllers which are designed for different objectives.

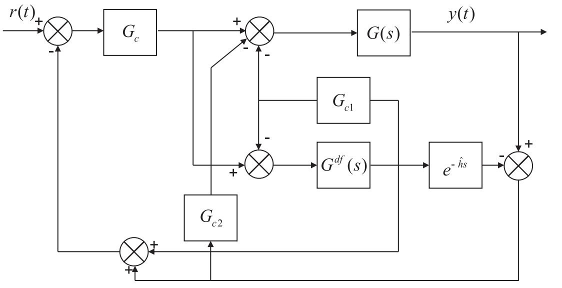

Fig. 1. MoSP structure.

Compensator G_{c1}, in the inner loop, is designed to prestabilize G^{df} in the unstable case. The other two controllers: G_c and G_{c2}, are used to take care of reference tracking and disturbance rejection, respectively and they must be designed maintaining the system stability. When $G_{c1}=0$ and $G_{c2}=0$, the standard Smith predictor for stable systems is obtained. Note that a nominal delay value $\hat{h}$ has been used in Fig. 1 (instead of the actual value for the delay) to design the compensator scheme. A full explanation of this structure can be found in the mentioned article [17]. The closed-loop response (with zero initial conditions) is given by

$$Y(s)=\frac{G^{df}G_c e^{-hs}(1+G_{c2}G^{df}e^{-\hat{h}s})}{(1+G^{df}[G_c+G_{c1}])(1+G_{c2}G^{df}e^{-hs})+G_c(Ge^{-hs}-G^{df}e^{-\hat{h}s})}R(s) \tag{2}$$

where $R(s)$ and $Y(s)$ are the Laplace transforms of the reference signal $r(t)$ and output $y(t)$ respectively. If Assumption 1 holds (the model perfectly matches the plant dynamics) and $\hat{h}=h$, then (2) reduces to

$$Y(s)=\frac{G^{df}G_c e^{-hs}}{1+G^{df}(G_c+G_{c1})}R(s) \tag{3}$$

It can be seen how the system with internal delay that appears in Eq. (2), becomes a system with external delay in Eq. (3). Therefore, this is precisely a topology that decouples the delay of the control strategy, making the system easier to control since compensators are designed regardless of the delay. On the other hand, if the delay is not known beforehand, then, the exact compensation cannot be performed despite the rational component is known and the closed-loop (2) can be potentially unstable. The problem we will face corresponds to the case when this delay is unknown and our objective is to obtain an estimation of the delay to guarantee the stability of the system. It is to highlight, that the approach is based on the MoSP [17], but it can be implemented on any other DCS. The only requirement is that the delay is bounded and decoupled from the control strategy. Thus, the proposed approach not only makes the DCS more robust with respect to uncertainty in the delay, but it allows its identification.

The way in which the delay is estimated consists of formulating the identification problem as an optimization problem solved online by Pattern Search Methods. Since this application is a novelty, a brief outline of the method is performed in the following subsection.

2.2. Generalized pattern search method

The generalized PSM consists of a sequence of iterations x_k, $k\in\mathbb{N}$. At each iteration, a number of *trial steps* s_k^i with $i=1,2,\ldots,p$ are added to the iteration x_k to obtain a number of *trial points*

$x_k^i = x_k + s_k^i$ at each iteration k. The objective function f is evaluated on these trial points through a series of *exploratory moves*, which defines a search mechanism in which the trial points are evaluated, and the obtained values are compared with $f(x_k)$. Then, the trial step s_k^* associated with minimum value of $f(x_k + s_k^i) - f(x_k) < 0$ is chosen to generate the next estimate of the patterns iterates $x_{k+1} = x_k + s_k^*$. The trial steps s_k^i are generated through a *step length* parameter $\Delta_k \in \mathbb{R}_+^n$ which is also updated through time depending on the value of x_{k+1}. The evolution of Δ_k establishes the convergence properties of the algorithm. In the following subsection, a more in-depth presentation is given:

2.2.1. The patterns

The objective of a pattern is to serve as a direction to generate the new trial points. Formally, to define a pattern two elements are necessary: a basis matrix B and a generating matrix C_k, which are used to determine the possible directions of the trial steps which will configure the next pattern. The basis matrix can be any nonsingular matrix $B \in \mathbb{R}^{n \times n}$. Frequently, for practical purposes $B = I_n$. The generating matrix is a matrix $C_k \in \mathbb{Z}^{n \times p}$, where $p > 2n$ and basically contains the directions in the space $\mathbb{R}^n$ on which the trial points are going to be generated. The structure of matrix C_k is given in Eq. (4).

$$C_k = [M_k - M_k L_k] \tag{4}$$

Usually, the matrix C_k is composed of a set of search directions defined as the columns of a matrix $M_k \in \mathbb{Z}^{n \times n}$ along with its opposites, $-M_k$ and a matrix $L_k \in \mathbb{Z}^{n \times (p-2n)}$ which contains possible columns that do not require opposite values and at least one column being identically zero to ensure that the original iterate x_k is also a possible trial point. It can be seen that the size of M_k and L_k are determined in function on the size of the domain of f which in this case is n.

A pattern is then defined by the columns of the matrix $P_k = BC_k$. The length of the direction vectors is modulated through the step-size parameter Δ_k. Therefore, the final trial step is formed by the columns of Eq. (5) below:

$$s_k^i = \Delta_k B c_k^i \tag{5}$$

where c_k^i denotes a column of $C_k = [c_k^1, c_k^2, \ldots, c_k^p]$. It can be seen from Eq. (5) that Δ_k serves as a step length parameter since acts modulating the modulus (length) of s_k^i. At iteration k, it defines a sequence of trial point $x_k + s_k^i$ with $i = 1, 2, \ldots, p$ as any point of the form $f(x_k^i) = f(x_k + s_k^i)$, where x_k is the current iterate.

For a better understanding of the role of the trial step in each pattern, in Fig. 2 it can be seen the possible pattern values when

$$C_k = \begin{bmatrix} 1 & 1 & 0 & -1 & -1 & -1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & -1 & -1 & -1 & 0 \end{bmatrix} \tag{6}$$

In this figure, open circles designate the possible trial points, solid circle indicates the trial point that is evaluated while open square specifies the trial points that have already been evaluated.

How to measure each pattern depends on the Exploratory moves selected, which will be explained below.

2.2.2. The exploratory moves

Pattern search methods proceed by evaluating the value of the objective function f on each trial point in a certain way before computing x_{k+1}. To allow the broadest possible choice of exploratory moves (and still maintain the properties required to prove convergence for the PSM)

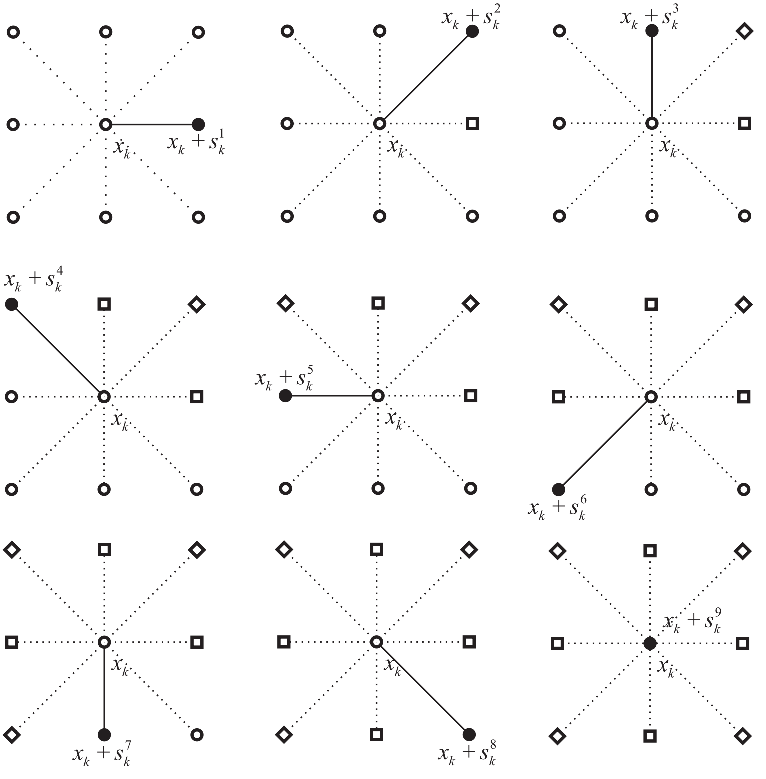

Fig. 2. All possible subsets of the trial steps for a PSM in $\mathbb{R}^2$ and matrix C_k (6).

there are two requirements on the exploratory moves. These requirements are given in the following Hypotheses [29]:

Hypothesis 1. $s_k \in \Delta_k P_k \equiv \Delta_k B C_k$

Hypothesis 2. If $\min\{f(x_k + s_k^i), s_k^i \in \Delta_k [M_k, -M_k]\} < f(x_k)$, then $x_{k+1} = x_k + s_k^*$.

The choice of exploratory moves must, therefore, ensure two things:

Concerning Hypotheses 1. The direction of any step s_k accepted at iteration k is defined by the pattern P_k, and its length is determined by Δ_k. This can be fulfilled easily since the trial points are precisely defined in this way by Eq. (5).

Concerning Hypotheses 2. A legitimate exploratory moves algorithm would be one that somehow guesses which of the steps defined by $\Delta_k P_k$ will produce a less value and then evaluates the function at only one such step. At worst case, a legitimate exploratory moves algorithm would be one that evaluates all s_k steps and returns the step that produced the least function value. This step trial is then used to generate the new estimate x_{k+1} being this process repeated until the minimum is reached.

2.2.3. The generalized PSM

Algorithm 1 describes formally the comments made in the previous sections and will be useful to introduce the concrete algorithm used for delay identification in Section 3. Algorithm 1 has a very simple operation. First, the objective function is evaluated in each trial points $x_k + s_k^i$, and choose the value s_k^* associated to the minimum of $\rho_k = f(x_k + s_k^i) - f(x_k)$ subject to $\rho_k \le 0$. Otherwise, $x_{k+1} = x_k$, this is the reason why L_k in Eq. (4) must contain a zero column. Finally the algorithm proceeds to update the values of C_k and Δ_k to generate a new sequence of exploratory moves.

Algorithm 1. The generalized pattern search method.

```
Let x_0 ∈ ℝ^n and Δ_0 > 0 be given
for k = 0, 1, … do
   Compute f(x_k)
   Determine a step s_k using an exploratory moves algorithm.
   Compute ρ_k = f(x_k + s_k^i) − f(x_k).
   if ρ_k ≤ 0 then
      x_{k+1} = x_k + s_k^i
   else
      x_{k+1} = x_k
   end if
   Update C_k and Δ_k.
end for
```

The Algorithm 2 specifies the requirements for updating Δ_k. The aim of the updating algorithm for Δ_k is to force $\rho_k^i = f(x_k + s_k^i) - f(x_k) < 0$. An iteration with $\rho_k < 0$ is successful; otherwise, the iteration is unsuccessful.

To define a particular pattern search method, it is necessary to specify the basis matrix B, the generating matrix C_k and the algorithms for updating C_k and Δ_k. In the next section the algorithm to update Δ_k is explained.

2.2.4. Δ_k updates

In the case that exploratory movement is successful, $\Delta_{k+1} = \lambda_k \Delta_k$ where $\lambda_k > 1$, thus, if an iteration is successful it may be possible to increase the step length parameter Δ_k. If the movement is unsuccessful $\Delta_{k+1} = \theta \Delta_k$ where $0 < \theta < 1$ and thus, the search space is reduced. The way to define λ and θ in Algorithm 2 is defined by [29].

Algorithm 2. Updating Δ_k.

```
Given 0 < θ < 1 and λ_k ≥ 1
if ρ_k ≤ 0 then
   Δ_{k+1} = θΔ_k
end if
if ρ_k > 0 then
   Δ_{k+1} = λ_k Δ_k.
end if
```

The algorithm for updating C_k depends on the pattern search method. For theoretical purposes, it is sufficient to choose the columns of C_k so that they satisfy (4) and the conditions for the matrices $M_k \in M \subset \mathbb{Z}^{n \times n}$ and $L_k \in \mathbb{Z}^{n \times (p-2n)}$.

2.2.5. Convergence properties of generalized PSM

In this section we will introduce the necessary conditions for the global convergence of generalized PSM formulated by [29] and introduced in the previous subsection. First, it is necessary that the set $L(x_0)=\{x\in\mathbb{R}^n : f(x)\leq f(x_0)\}$ is compact. Secondly that the objective function f is continuously differentiable on a neighborhood of $L(x_0)$, and, thirdly it is necessary a simple decrease on f which implies that the minimum has to be a unique global minimum of f, without the presence of local extrema. Then, it can be proved that the sequence of iterates $\{x_k\}$, obtained though Pattern Search Algorithm 1, converges to the minimum of the function f.

The details of the theorem can be consulted in [29], Theorem 3.1. The principal inconvenient of this algorithm is the necessity of simple decrease in f. The generalized PSM convergence properties were extended in [2] to the case when the objective function may posses local extrema but a unique global minimum. The key to obtain this generalization is to allow local exploration, in an asymptotically dense set of directions in the space of the optimization variables.

These two results will be the fundamentals ones that will be used to establish the convergence properties of the delay identification algorithm as well as the stability proof.

3. Proposed control scheme using PSM

The PSM is implemented using a so-called multi-model scheme [10]. Indeed, the multi-model approach represents an online implementation of the method, for the first time formulated in an adaptive control perspective. Henceforth, the term *trial point* is equivalent to *model*, this change in terminology is in order to facilitate understanding of the multi-model scheme and to frame it adequately in the generalized PSM introduced in Section 2.2. The equivalence between the PSM and the multi-model-based implementation is explained in detail in the following sections.

The basic structure of the proposed scheme is depicted in Fig. 3. As it can be seen, the MoSP and the Pattern Search Method are complemented with three elements: an objective function which evaluates the potential behavior of each model and a switching rule which monitors periodically this index and decides which model is the best, used to generate the control signal. The switching mechanism is intended to reduce the possible mismatch between the nominal and the actual system output. This methodology can be framed as a special case of generalized PSM, because the minimization is performed on real-valued functions with real arguments. In the following subsections the different elements of the presented architecture are considered in detail and framed into the generalized PSM.

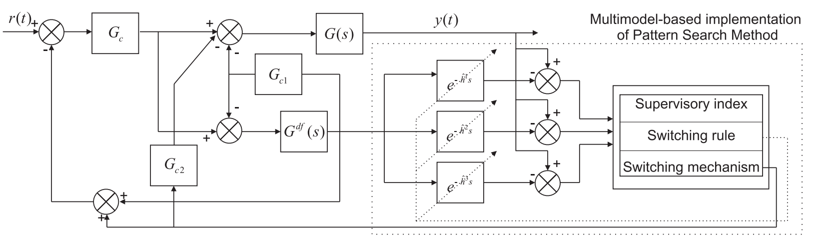

Fig. 3. Basic architecture (based on MoSP) of the control scheme.

3.1. Objective function

The first element of the proposed scheme is an objective function aimed at evaluating the behavior of the delay model $\hat{h}$. The suggested objective function is

$$J(\hat{h},t)=\int_{t-T_{res}}^{t}(y(\tau)-\hat{y}(\tau))^2\,d\tau \tag{7}$$

The integration takes place in a time interval in which the delay model does not being modified. In addition, $y(t)$ is the plant output while $\hat{y}(t)$ denotes the output associated to the delay model $\hat{h}$. Notice that both $y(t)$ as $\hat{y}(t)$ depends on the reference signal $r(t)$ and hence, so does (7). This dependence is expressed in Eq. (7) implicitly as dependence with t. T_{res} is the so-called residence time and defines the time interval window where the delay model $\hat{h}$ is evaluated. As we will show, the residence time is a positive number large enough be able to distinguish between the different delay values $\hat{h}$ but, as shown later, its concrete value does not modify the convergence properties of the algorithm. Also it can be seen that the objective function (7) differs from the proposed in [29] because this is time-dependent. However. this is not a problem in order to frame the present approach into the generalized PSM as explained in Subsection 3.4.

Defining the error between the plant output and the model output as

$$e(t)\doteq y(t)-\hat{y}(t), \tag{8}$$

the Laplace transform of the error (with zero initial conditions) is given by

$$E(s)=\frac{G_cG^{df}(e^{-hs}-e^{-\hat{h}s})(1+G_{c2}G^{df}e^{-\hat{h}s})}{(1+G_{c2}G^{df}e^{-hs})((1+G^{df}[G_c+G_{c1}])(1+G_{c2}G^{df}e^{-hs})+G_c(G^{df}e^{-hs}-G^{df}e^{-\hat{h}s}))}R(s), \tag{9}$$

It is readily seen from Eq. (9) that the error (8) is zero when $\tilde{h}=h-\hat{h}=0$, that is, when the delay model is equal to the real one. According to this fact, the search of the minimum of the function (7) leads to an estimation of the real delay value. Thus, the identification problem converts into an optimization one. It is important to notice that (7) is a continuously differentiable function, since it is defined as the square of the subtraction of two functions that are continuously differentiable as can be proved making the partial derivative of Eq. (2) with respect to $\hat{h}$, which leads to a continuous function. Also (7) satisfies the compactness condition stated in Section 2.2.5. On the other hand, the simple decrease condition (convexity) cannot be guaranteed to be satisfied.

3.2. The patterns

The proposed architecture is composed of patterns. These are the directions which the nominal delay model $\hat{h}_k$ is modified to obtain the delays which implement the delay models. These models are time-varying and automatically adjusted by the algorithm as corresponds to the PSM.

Taking Eq. (5), let the basis matrix $B=1$ and, $C_k=[1,-1,0]$, which is the simplest case. Now define the trial steps,

$$\Delta h_k=\Delta_k\bullet BC_k \tag{10}$$

where $\bullet$ is the Schur (or component-wise) product [3], $\Delta_k=[\Delta h_k^{up}\Delta h_k^{lo}0]$, in such a way that the models (trial points) take the next form.

$$\{\hat{h}_k^{upp},\hat{h}_k^{low},\hat{h}_k^{nom}\}=\{\hat{h}_k+\Delta h_k^{up},\hat{h}_k-\Delta h_k^{lo},\hat{h}_k\} \tag{11}$$

It can be observed that the patterns consists in adding and subtracting certain quantities to the current nominal delay $\hat{h}_k$.

Thus, the models consist of the same rational component and different delays that will take part of the multi-model strategy as depicted in Fig. 3. At each iteration the nominal model is used in the control law.

Remark 1. If we take

$$\Delta_k = [\Delta_k^1, \dots, \Delta_k^p, \Delta_k^{-1}, \dots, \Delta_k^{-p}, 0]$$

Δ_k would define $2p+1$ models, with p quantities added to the nominal model and p quantities subtracted, instead of 3 ($\Delta_k = [\Delta h_k^{up} \Delta h_k^{lo} 0]$) which is a generalization of the scheme. □

3.3. Proposed pattern search method

The PSM monitors at time instants, being integer multiple of the residence time, the objective function value and selects the nominal delay which is the best estimation of the real one. The nominal delay is used within each time interval to generate the control law. The initial model is selected by the designer with the initialization of $\hat{h}(0)$ and $\Delta\hat{h}^{up/lo}(0)$ using Assumption 2. From this moment onwards, the PSM can be expressed formally as the Algorithm 3.

Algorithm 3. Delay identification PSM.

$k=1$
$\hat{h}_k^{nom}$:Initial nominal delay
$\Delta_k = [\Delta\hat{h}_k^{up} \Delta\hat{h}_k^{lo} 0]$
$[\underline{h}, \overline{h}]$: Intervals of uncertainty of delays
$\gamma > 1$:Reduction factors for Δ_k
$T_{res} > 0$:Residence time
$S_m = nom$:Selected model
$P_m = nom$:Position the new nominal model
$\Delta h_k = \Delta_k \bullet BC_k$
$\hat{h}_k = \hat{h}_k^{nom} + \Delta h_k$
for $t > 0$ **do**
  if $t = mT_{res},\ m \in \mathbb{N}$ **then**
    $k \leftarrow k+1$
    $\rho_k^i = J_k(\hat{h}_k^i) - J_k(\hat{h}_k^{nom})$
    $S_m \leftarrow \arg\min_i \{\rho_k^i; i = upp, low, nom\}$
    $\hat{h}_k^{nom} = \hat{h}_{k-1}^{S_m}$
    Update Δ_k
    $\Delta h_k = \Delta_k \bullet Bc_k$
    $\hat{h}_k = \hat{h}_k^{nom} + \Delta h_k$
  end if
end for

In this algorithm we need to know the position (*upper, lower and nominal*) of the current nominal model in $\hat{h}_k^i$. For this purpose we make use of an instrumental variable: P_m, this variable controls change direction for $\Delta\hat{h}_k^{lo}$ and $\Delta\hat{h}_k^{up}$.

The comparisons between the different models are carried out in groups of three consisting of the nominal $\hat{h}_k^{nom}$, plus the additive $\hat{h}_k^{upp}$ and subtractive $\hat{h}_k^{low}$ disturbances. The models (11) are compared (Line 14) with nominal model using the performance index (7). As a result, the following outcomes are possible (Line 15). If $S_m = upp$ means that the model with the additive disturbance has been selected for the next time interval. $S_m = low$ stands for the subtractive disturbance and $S_m = nom$ for the previous nominal model. In this way the element associated with the lowest value of the function objective is obtained.

When comparing Algorithms 1 and 3, it can be seen that they have a similar behavior and whose difference is the way to evaluate models, because while the Algorithm 1 makes an exploratory move one at a time, in Algorithm 3 all exploratory moves are performed simultaneously, this allows that the convergence time is narrow down considerably. Since the best model is used rather only one that is good.

3.3.1. The updates

The limits $\Delta\hat{h}_k^{up}(t)$ and $\Delta\hat{h}_k^{lo}(t)$ are adjusted to ensure the convergence to the true delay. It can be seen that the reduction factor depends both of the last nominal model position P_m and the selected model S_m, as it is shown in Algorithm 4.

Algorithm 4. Update Δ_k.

1: **if** $P_m = low$ **then**

2: **if** $S_m = nom$ **then**

3: $$\{\Delta\hat{h}_k^{lo}, \Delta\hat{h}_k^{up}, P_m\} = \left\{-\frac{\Delta\hat{h}_{k-1}^{lo}}{\gamma}, \Delta\hat{h}_{k-1}^{up} + \frac{\Delta\hat{h}_{k-1}^{lo}}{\gamma}, nom\right\}$$

4: **else**

5: $$\{\Delta\hat{h}_k^{lo}, \Delta\hat{h}_k^{up}, P_m\} = \left\{\Delta\hat{h}_{k-1}^{lo} + \frac{\Delta\hat{h}_{k-1}^{up}}{\gamma}, -\frac{\Delta\hat{h}_{k-1}^{up}}{\gamma}, upp\right\}$$

6: **end if**

7: **else**

8: **if** $P_m = nom$ **then**

9: **if** $S_m = nom$ **then**

10 $$\{\Delta\hat{h}_k^{lo}, \Delta\hat{h}_k^{up}, P_m\} = \left\{-\frac{\Delta\hat{h}_{k-1}^{lo}}{\gamma}, \Delta\hat{h}_{k-1}^{up} + \frac{\Delta\hat{h}_{k-1}^{lo}}{\gamma}, low\right\}$$

11: **else**

12: $$\{\Delta\hat{h}_k^{lo}, \Delta\hat{h}_k^{up}, P_m\} = \left\{\frac{\Delta\hat{h}_{k-1}^{up}}{\gamma}, \frac{\Delta\hat{h}_{k-1}^{lo}}{\gamma}, upp\right\}$$

13: **end if**

14: **else**

15: **if** $S_m = nom$ **then**

16: $$\{\Delta\hat{h}_k^{lo}, \Delta\hat{h}_k^{up}, P_m\} = \left\{\frac{\Delta\hat{h}_{k-1}^{up}}{\gamma}, \frac{\Delta\hat{h}_{k-1}^{lo}}{\gamma}, nom\right\}$$

17: **else**

18: $$\{\Delta\hat{h}_k^{lo}, \Delta\hat{h}_k^{up}, P_m\} = \left\{\Delta\hat{h}_{k-1}^{lo} + \frac{\Delta\hat{h}_{k-1}^{up}}{\gamma}, -\frac{\Delta\hat{h}_{k-1}^{up}}{\gamma}, low\right\}$$

end if
end if
end if
$\Delta_k = [\Delta\hat{h}_k^{up}\,\Delta\hat{h}_k^{lo}\,0]$

It can be seen that the Algorithm 4 formulation is more complex than Algorithm 2, since in this case Δ_k do not relax and contract as is in Algorithm 2. In this case, at each iteration the search space is always reduced by changing the upper and lower limits, because the Algorithm 4 ensures that the global minimum can not be in a search space that was already evaluated. The limits value $\Delta\hat{h}_{k+1}^{up}$ and $\Delta\hat{h}_{k+1}^{lo}$ in the following residence time, depend of the reduction factor γ and the limits values $\Delta\hat{h}_k^{up}$ and $\Delta\hat{h}_k^{lo}$. This guarantees that all the models converge to the same value and hence the scheme tends to a time invariant system with the real delays.

Moreover, since Algorithms 3 and 4 use only simple mathematical and logical operations, they can be implemented easily in real world applications. Such implementations can be based on microcontrollers, low-cost chips or similar programmable devices, yielding a great source of possible applications.

3.4. Convergence results of the identification scheme

This section states the convergence results of Algorithms 3 and 4, guaranteeing the identification of the real delay. The proof is performed in two steps. First, the conditions under which (7) has a unique global minimum for $\hat{h} = h$ are stated. Having established that the real delay is the only global minimum of the function objective, it will be shown that Algorithm 3 is able to asymptotically find the global minimum of $J(\hat{h}, t)$.

From Eq. (7) it can be seen that $J(\hat{h}, t) = 0$ for $\hat{h} = h$ and that $J(\hat{h}, t) \geq 0$ since its integrand is non-negative. Thus, the global minimum is calculated when $J(\hat{h}, t) = 0$. The next Assumption 3 will be used subsequently.

Assumption 3. The input signal satisfies that there exists $T > 0$ such that $r(t) \neq r(t-\lambda)$ $\forall \lambda \in [\underline{h}, \overline{h}] \cap (0, +\infty)$ with $\underline{h}, \overline{h} \in \mathbb{R}$ and for all t belonging to an interval $I \subseteq [t-T, t]$ of positive measure. □

This assumption means that the values that takes the input signal $r(t)$ at an interval of time I, cannot be repeated in the next time interval. This assumption is necessary to ensure the non-periodicity of the input signal for different time-windows as those used in Eq. (7). It should be noted that this requirement is not difficult to perform in practice, since it is easily fulfil by a non-periodic signal obtained for instance as a sum of sinusoid with different amplitude, frequency and phases. Now we are in conditions to formulate the next Lemma.

Lemma 1. *The function (7) has a unique global minimum at $\hat{h} = h$ for all t provided that the reference signal $r(t)$ satisfies Assumption 3 for a given $T_{res} = T$.*

Proof. Eq. (9) can be equivalently represented in terms of the reference signal as

$$E(s)=\frac{G_cG^{df}(e^{-hs}-e^{-\hat{h}s})(1+G_{c2}G^{df}e^{-\hat{h}s})}{(1+G_{c2}G^{df}e^{-hs})((1+G^{df}[G_c+G_{c1}])(1+G_{c2}G^{df}e^{-hs})+G_cG^{df}(e^{-hs}-e^{-\hat{h}s}))}R(s) \tag{12}$$

Introducing,

$$Z(s)=\frac{G_cG^{df}(1+G_{c2}G^{df}e^{-\hat{h}s})}{(1+G_{c2}G^{df}e^{-hs})((1+G^{df}[G_c+G_{c1}])(1+G_{c2}G^{df}e^{-hs})+G_cG^{df}(e^{-hs}-e^{-\hat{h}s}))}R(s), \tag{13}$$

and $z(t)=\mathcal{L}^{-1}[Z(s)]$ then

$$y(\tau)-\hat{y}(\tau)=\mathcal{L}^{-1}[E(s)]=[z(\tau-h)-z(\tau-\hat{h})] \tag{14}$$

making use of the delay theorem of the Laplace transform. If $t=\tau-h$ and $\tilde{h}=\hat{h}-h$ then, (7) can be written as

$$J(\hat{h},t)=\int_{\tau-h-T_{res}}^{\tau-h}(z(t)-\hat{z}^{(i)}(t-\tilde{h}))^2\,dt \tag{15}$$

where $J(\hat{h},t)\neq 0$ if $\tilde{h}\neq 0$ because if (15) equals zero in the interval $[\tau-h-T_{res},\tau-h]$, for any other value of $\tilde{h}$, $z(t)$ would be periodic which is a contradiction with Assumption 3 on the input signal provided that $T_{res}=T$. Notice that $z(t)$ cannot be periodic if the reference signal satisfies Assumption 3, since a non-periodic signal still-remain as non-periodic when it is processed by a LTI system Eq. (13). Thus Lemma 1 is proved. □

Lemma 1 states that there exists an unique global minimum for each interval $[t-T_{res},t]$, despite for each time window $J(\hat{h},t)$ may take a different form. The approach presents the particularity that the global minimum is always the same. Thus, the proposed approach can be framed into the generalized PSM.

Next, we establish that the proposed algorithm is able to find the global minimum of the proposed function $J(\hat{h},t)$. Lemma 1 guarantees that under non-periodic input, Eq. (15) has a global minimum but there may be local minima, while the generalized PSM is designed for decreasing function (convex function) as commented in Section 2.2.5. Fortunately, as also commented in Section 2.2.5, the original generalized PSM given in [29] is extended in [2] to functions with multiple local minima. This has been done by converting the search into dense which can be achieved by Algorithm 4 making the parameter γ very close to unity. This implies that the updating of the new models does not lose the global minimum at each iteration, since new models are very close to old ones. This fact motivates that the proposed algorithm works well for the optimization of the time-varying function (7), since the problem at hand is equivalent to consider a family of time-invariant functions all of them with a unique global minimum exactly at the same point. Thus, if the search is dense, then, if the algorithm does not lose the minimum of a function during the updating of Δ_k, it is neither losing the minimum of any of the remaining functions. Therefore, the unique global minimum is not lost and algorithm works properly.

Under these circumstances, we are in conditions of applying the results from [2] and formulate the following result on the identification of the delay:

Theorem 1. *Consider the delay system given by (1) satisfying Assumptions 1 and 2. Thus , the generalized PSM based Algorithms 3 and 4 through models (11) can identify the real delay* *provided that the reference signal* $r(t)$ *satisfies Assumption 3 for a value of* $T_{res} = T$, γ *is sufficiently close to unity and* $(G_c + G_{c1})$ *stabilizes* $G^{df}(s)$. □

Proof. Using Lemma 1, and γ being close to unity we are in conditions of applying Theorem 4.3 of [2], guaranteeing the delay identification. □

Notice that the identification result has been easily established thanks to the fact that the identification problem has been formulated within a generalized PSM, taking advantage of this technique in its applications to Control Theory. Also, note that Theorem 1 requires γ being sufficiently close to unity. However, it has been observed in simulation examples, showing some of them in Section 5, that a finite value for it is suffices. It can be pointed out that the identification scheme does not depend on the specific value of T_{res}; it is only necessary to fulfil Assumption 3. for a T_{res} value which is a very slight condition.

4. Stability analysis

The closed-loop stability properties are stated in the following steps. Firstly, it is shown that the MoSP-based scheme given in Fig. 1 is robust under small (perhaps very-small) delay uncertainties. Secondly, we will use the identification properties of the proposed algorithm, stated in Theorem 1, to show that the nominal delay converges to a neighborhood of the actual delay in finite time and eventually becomes a time-invariant system. Thus, since the MoSP is robust under small perturbations, and the nominal delay is arbitrarily near the real delay, the closed-loop system is stable.

The following lemma will be used:

Lemma 2. *Consider that the MoSP control scheme depicted in Fig. 1 with* $G(s) = \hat{G}(s)$ and $(G_c + G_{c1})$ *stabilizes* $G^{df}(s)$. *Then, there exists a (sufficiently) small* $\Delta\hat{h}$ *such that the closed-loop system is still stable when* $\hat{G}(s) = G^{df}(s)e^{-(h+\Delta\hat{h})s}$.

Proof. The proof is along the lines of those in [13] and [22]. □

This result can be interpreted saying that the MoSP has a robust behavior for small uncertainties of delays not exceeding $\Delta\hat{h}$. Thus, the following stability theorem can be formulated.

Theorem 2. *The closed-loop system depicted in Fig. 3 obtained from Eqs. (1), and (11) through Algorithm 4 is stable provided that Assumptions 1 and 2 hold,* γ *is sufficiently close to unity and* $(G_c + G_{c1})$ *stabilizes* $G^{df}(s)$. □

Proof. The proof is made by contradiction. If the output is unbounded, the input signal $u(t)$ behaves as a non-periodic signal satisfying Assumption 3 for any value of T_{res}. Therefore, Theorem 1 guarantees that the nominal delay converges to the actual delay, and hence $\lim_{t\to\infty}\hat{h}(t) = h$. Hence, there exists a strictly ordered real positive sequence (except perhaps its first element) $\{t_n\}_{n=0}^{\infty}$ such that $|h-\hat{h}(t_n)| > |h-\hat{h}(t_{n+1})|$ for all $n \geq 1$. Thus, for a prescribed real constant ϵ, there exists a finite time $t_{n'}$ such that $|h-\hat{h}(t_n)| \leq \epsilon$ for all $t_n \geq t_{n'}$. Thus if $\epsilon < \Delta h$, where the existence of a positive Δh is guaranteed by Lemma 2, the algorithm reaches in finite time a neighborhood of the actual delay where the MoSP is robust. Consequently, the output cannot be unbounded, and this contradicts our assumptions. Accordingly, all the signals in the loop are bounded. □

It can be seen that the proof is straightforward, given that the scheme has been correctly framed within the PSM, inheriting its convergence properties and that the MoSP has a minimum robust behavior.

5. Simulation examples

In this section, we will show simulations with the proposed PSM. We use the scheme presented in [17], to show the improvement of the performance of the initial scheme when the delay is unknown. We choose this scheme because it offers acceptable closed-loop since both servo-tracking and disturbance rejection properties can be designed.

5.1. Modified Smith Predictor by S. Majhi and D. P. Atherton

The controller design procedure is performed in [17] for First-Order Delayed Unstable Processes (FODUP) and the Second-Order Delayed Unstable Processes (SODUP), which are described by equations (16) and (17) respectively serving as usual models for many industrial processes:

$$G(s) = \frac{Ke^{-hs}}{\tau s - 1} \tag{16}$$

$$G(s) = \frac{Ke^{-hs}}{(\tau s - 1)(as + 1)} \tag{17}$$

where $\tau, h, a \in \mathbb{R}^+$.

The scheme proposed in [17], has a good performance when the delay is known, but according to Eq. (2), when $\hat{h} \neq h$, an internal delay is obtained and the scheme exhibits a bad behavior. In Fig. 4 we make a variation for different delays in a range of $\pm 30\%$ of the real delay. It can be seen that the performance is degraded, even making the system unstable.

An example of how to use the approach to the delay identification is discussed below.

5.2. Delay identification

As mentioned in the introduction, the proposed scheme can be used to the delay identification for unstable or integrating system, with the advantage that the system will never become unstable. The rational component of the process is given by

$$G(s) = \frac{e^{-5s}}{10s - 1} \tag{18}$$

and $G_c = 10(0.2s+1)/0.2s$, $G_{c1} = 13.468$ and $G_{c2} = 1.414$. These selections are taken from. [17]. For this simulation the $T_{res} = 8$ s, the reference signal used is $r(t) = \sin(t/3)$ and the initial patterns are

$$\{\hat{h}_0^{upp}, \hat{h}_0^{low}, \hat{h}_0^{nom}\} = \{\hat{h}_0 + \Delta h_0^{up}, \hat{h}_0 - \Delta h_0^{lo}, \hat{h}_0\} = \{6, 1, 3.5\}$$

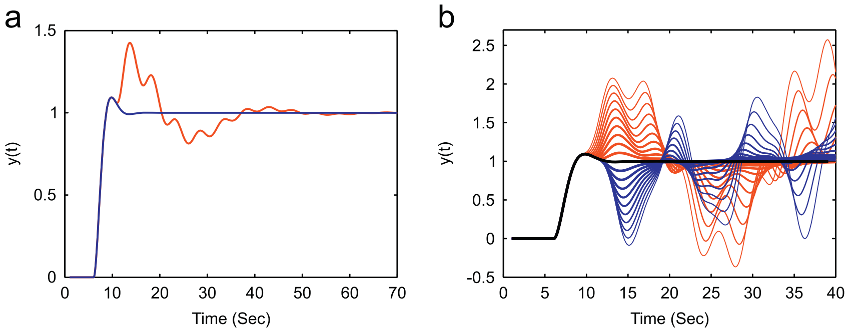

Fig. 4. (a) Error in the delay model by 15% and (b) delay variation from 0 to 30%.

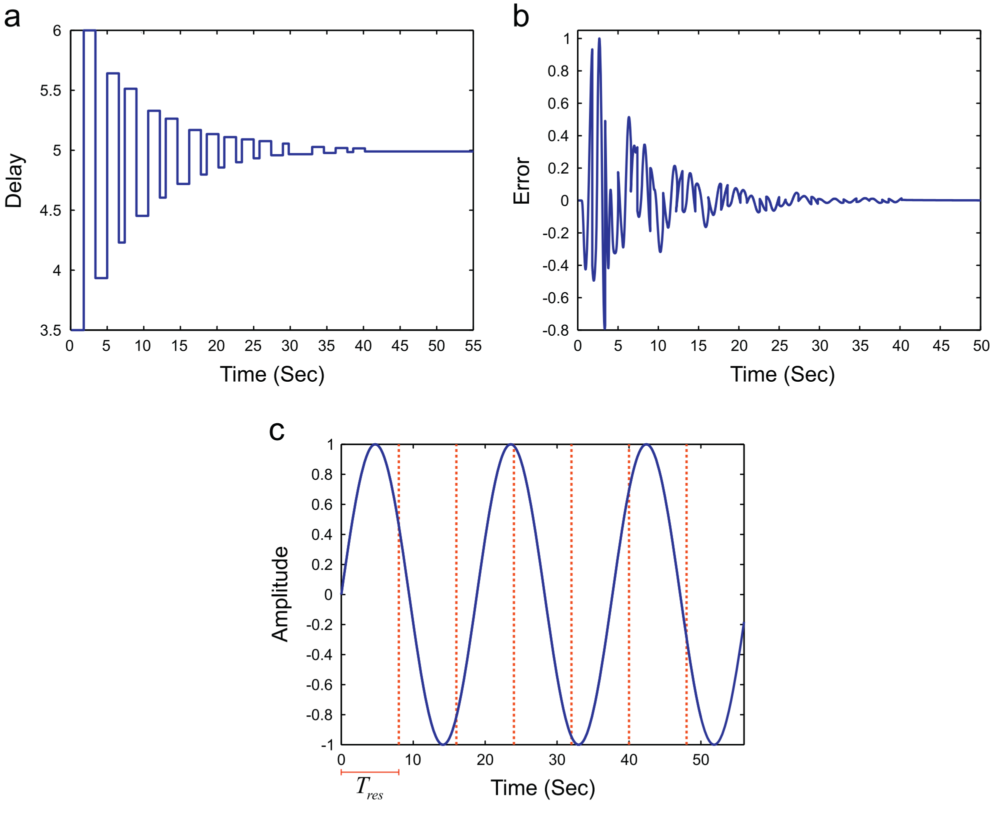

Fig. 5. (a) Delay obtained by identification, (b) error signal and (c) input signal $\sin(t/3)$ for different time windows.

In Fig. 5(a) it can be seen the nominal delay $\hat{h}_k$ behavior is stabilized in 4.99 s, which is a fairly accurate estimation to the actual delay $h = 5$ s. Fig. 5(b) shown the error signal between the actual and the perfectly compensated system. It can be seen that the error converges to zero.

There are two annotations to be considered in this simulation. Firstly, despite that the input signal is a periodic function $r(t) = \sin(t/3)$, this satisfies Lemma 1 for each time window, as shown in Fig. 5(c)) for the adequate choice of $T_{res}=8$. Secondly, note that a finite value for $\gamma=1.15$ s is sufficient to perform the delay identification.

5.3. First-order delayed unstable processes

In this case, the process and the controllers are the same as used in the previous simulation, the residence time $T_{res}=0.4$ s and the initial patterns are

$$\{\hat{h}_0^{upp}, \hat{h}_0^{low}, \hat{h}_0^{nom}\} = \{\hat{h}_0 + \Delta h_0^{up}, \hat{h}_0 - \Delta h_0^{lo}, \hat{h}_0\} = \{6, 2, 4\}$$

A comparison with the approach proposed by S. Majhi and D. P. Atherton is made, in this case we assumed that the nominal delay $\hat{h}_0^3 = 4$ s, compared with the real delay $h=5$ s which corresponds to an error of 25%. Fig. 6(a) clearly shows that the scheme is able to tackle the uncertainty in the delay while leading to remarkable performance. Fig. 6(b) shows the value that takes the delay for the control law at each time interval. In this case it has not been precisely identified, $\hat{h}=4.78$ s. It can be seen, that the delay identification is acceptable, accompanied of a good performance of the system response, despite that the reference signal is just a step. Notice that in this situation, the input signal does not meet the requirements of a strongly excited signal, which is a condition commonly used in the identification schemes.

5.4. Second-order delayed unstable processes

In this section we will inspect the behavior of the proposed scheme for SODUP systems. The rational component of the process used to perform the simulation is

$$G(s) = \frac{2e^{-5s}}{(10s-1)(2s-1)} \tag{19}$$

How to select the controllers is well explained in [17]. In this simulation, the $T_{res}=0.4$ and the initial patterns are:

$$\{\hat{h}_0^{upp}, \hat{h}_0^{low}, \hat{h}_0^{nom}\} = \{\hat{h}_0 + \Delta h_0^{up}, \hat{h}_0 - \Delta h_0^{lo}, \hat{h}_0\} = \{6.75, 2.75, 4.75\}$$

The output system is shown in Fig. 7(a) and it is compared with S. Majhi and D. P. Atherton's method, in which the value of delay model is $\hat{h}=4.75$ s. It can be seen that the response with the proposed scheme has a smaller settling time 130 s, compared with 280 s obtained by the other

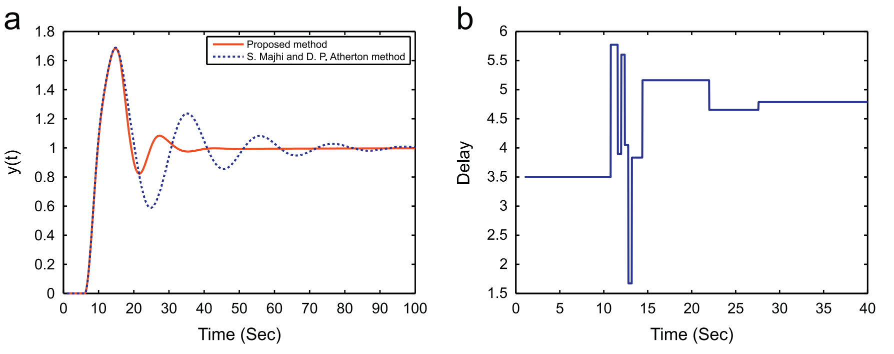

Fig. 6. (a) Output signal for first-order system and (b) delay evolution through time.

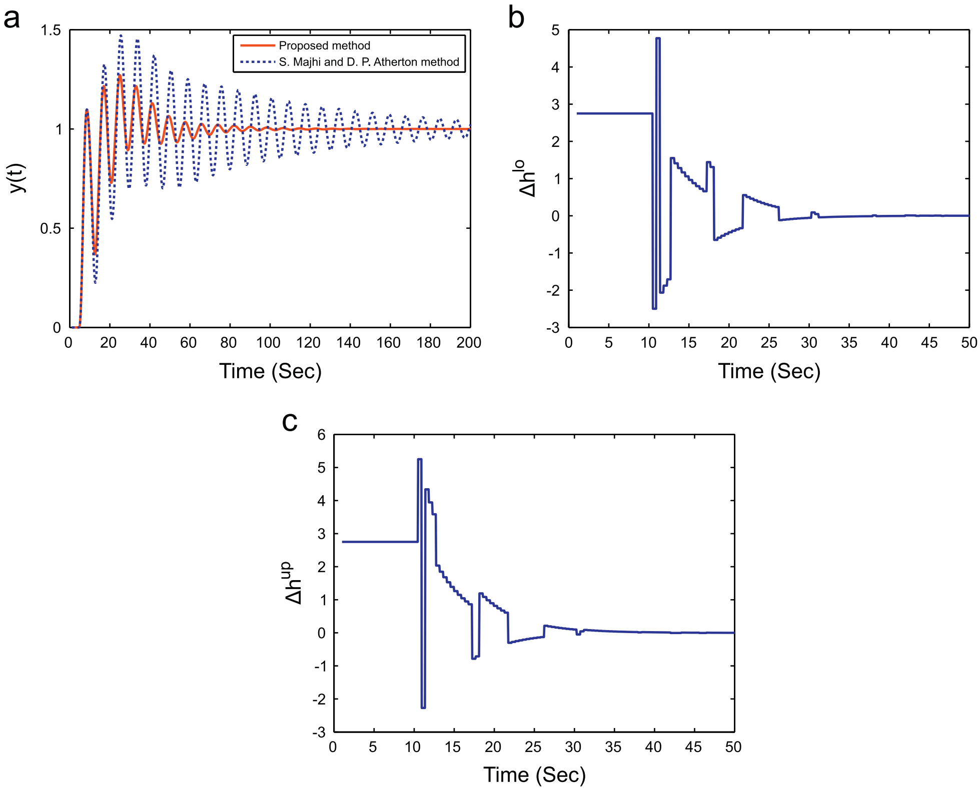

Fig. 7. (a) Trial steps performance, (b) Δh_k^{lo} and (c) Δh_k^{up}.

method. It also shows that the ripple value is much lower.

Fig. 7(b) and (c) shows how the trial steps Δh_k^{lo} and Δh_k^{up} tend to zero, respectively. The graph is drawn only up to $t=50$ s, given that from that instant of time the value remains constant.

Hence, the multi-model adaptive control strategy in combination with a delay compensation scheme allows to overcome the presented difficulty related to the uncertainty in the delay, despite that the input signal is a step.

5.5. Variation of residence time

In this section, the residence time influence in the identification capabilities of the scheme is discussed. For it, the process and the controllers are the same as used in the simulation 5.3. As it will be shown, its concrete value does not change this property, if it is chosen satisfying Assumption 3 for each reference input.

Fig. 8 shows some simulations varying the residence time. Hence, it can be seen that the residence time does not play an important role for a good identification as commented in Section 3.4 if selected conveniently. Note that if the input is non-periodic, Assumption 3 may not impose any restriction on the T_{res}, making this choice irrelevant. Since the delay accuracy has on error ranging between 0.02% and 3.5% of the actual value of the delay which is very acceptable estimation for a step signal. Moreover, the concrete value of the residence time does not threaten

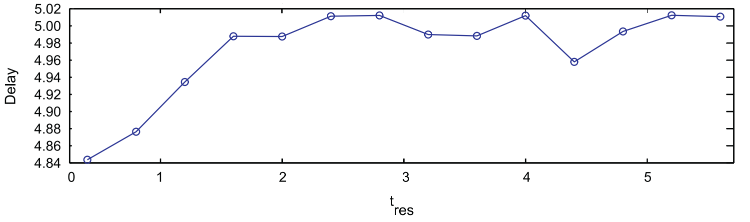

Fig. 8. Delay obtained with respect to the residence time.

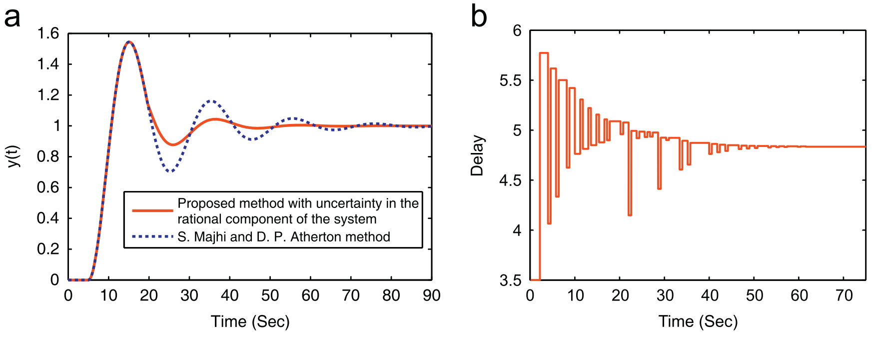

Fig. 9. (a) Output signal for system with a difference in the parameters and (b) delay obtained by identification.

the stability of closed-loop, since the proposed algorithm is designed to converge to a time-invariant system, and according to the results [8] it does not influence stability.

5.6. *Robustness of the controller*

Finally, we show some results to verify the robust performance of the proposed scheme under uncertainty in the rational component of the system. As in the previous case, the process and the controllers are the same as used in the simulation 5.3, but in this case we will assume an error of 15% in the parameters of the plant, as shown below.

$$G(s) = \frac{0.8696e^{-5s}}{11.5s - 0.8696} \tag{20}$$

Fig. 9(a) shows the output signal of the system modeling errors. In this case, results demonstrate that the proposed scheme has a robust behavior for uncertainties in modeling parameters of the plant, and indeed it can be seen that the system does not diverge. In Fig. 9(b) it can be observed the behavior of delay reaching a value of 4.83 s, which is quite acceptable provided that there is an uncertainty of 15% in the modeling parameters.

6. Conclusion

This paper has presented a multi-model adaptive control strategy to be applied to delay compensation schemes for stable/unstable systems. The main objectives are the delay identification and ensure the closed-loop stability, especially for unstable systems for which this problem is especially hard. The approach is formulated as an optimization problem and is framed into the generalized PSM, inheriting the convergence properties, which are a novelty both in the Control Theory as well as in Mathematics. The optimization has been implemented online by using a multiple-model scheme which is also a novel implementation of Pattern Search Methods.

It has been shown through simulation examples that the scheme can be used to the delay identification without turning the system unstable, with a great precision. It also possesses the advantage of being easy implementable in real systems, such on microcontrollers, low-cost chips or similar programmable devices, since its operation is quite simple, yielding a great source of possible applications.

Finally, it is shown that the proposed approach is robust even when the rational component of the system has a 15% error, providing versatility and could be implemented in real systems, since these usually have errors in modeling parameters. In authors' opinion, Pattern Search Methods, constitute a powerful optimization technique for control-oriented applications.

Acknowledgments

This work was partially supported by the Spanish Ministry of Economy and Competitiveness through grant DPI2012-30651, by the Basque Government (Gobierno Vasco) through grant IE378-10, by the University of the Basque Country through grant UFI 11/07 and by the Automatic and Electronic research group of the Instituto Tecnológico Metropolitano (ITM) under the projects "Maximización de extracción de energía en aerogeneradores para cogeneración urbana en el Valle de Aburrá", code P10-233.

References

[1] S. Alcántara, C. Pedret, R. Vilanova, W.D. Zhang, Simple analytical min-max model matching approach to robust proportional-integrative-derivative tuning with smooth set-point response, Industrial & Engineering Chemistry Research 49 (2) (2010) 690–700.

[2] C. Audet, J.E. Dennis, Mesh adaptive direct search algorithms for constrained optimization, SIAM Journal on Optimization 17 (2006) 188–217.

[3] Dennis S. Bernstein, Matrix Mathematics, Princeton University Press, 2005.

[4] C. Bogani, M.G. Gasparo, A. Papini, Generalized Pattern Search methods for a class of nonsmooth optimization problems with structure, Journal of Computational and Applied Mathematics 229 (2009) 283–293.

[5] N. Hansen, André Niederberger, L. Guzzella, P. Koumoutsakos, A method for handling uncertainty in evolutionary optimization with an application to feedback control of combustion, IEEE Transactions on Evolutionary Computation 13 (1) (2009) 180–197.

[6] J. Herrera, A. Ibeas, S. Alcantara, M. de la Sen, Multimodel-based techniques for the identification and adaptive control of delayed multi-input multi-output systems, IET Control Theory & Applications 5 (1) (2011) 188–202.

[7] J. Herrera, A. Ibeas, S. Alcantara, R. Vilanova, Identification and adaptive control of delayed unstable systems, in: Proceedings of the 2010 International Symposium on Intelligent Control (ISIC), September 8–10, Yokohama, Japan, 2010.

[8] J. Herrera, A. Ibeas, S. Alcantara, R. Vilanova, Multimodel-based techniques for the identification of the delay in mimo systems, in: Proceedings of the 2010 American Control Conference, June 30–July 2, Baltimore, Maryland, USA, 2010.

[9] A. Ibeas, M. de la Sen, Artificial intelligence and graph theory tools for describing switched linear control systems, Applied Artificial Intelligence 20 (9) (2006) 703–741.

[10] A. Ibeas, R. Vilanova, P. Balaguer, M. de la Sen, Multiple-model based intelligent control techniques for LTI systems with unknown external delays part i: known rational component, International Journal of Innovative Computing, Information and Control 4 (2) (2008) 839–848.

[11] K. Jabri, D. Dumur, E. Godoy, A. Mouchette, B. Bele, Particle swarm optimization based tuning of a modified smith predictor for mould level control in continuous casting, Journal of Process Control 21 (2011) 263–270.
[12] M. Jelali, Estimation of valve stiction in control loops using separable least-squares and global search algorithms, Journal of Process Control 18 (2008) 632–642.
[13] M. Krstic, Delay Compensation for Nonlinear, Adaptive, and PDE Systems, Birkhauser, 2009.
[14] L. Liu, X. Zhang, Generalized Pattern Search methods for linearly equality constrained optimization problems, Applied Mathematics and Computation 181 (2006) 527–535.
[15] T. Liu, Y. Cai, D. Gu, W. Zhang, New Modified Smith Predictor scheme for controlling integrating and unstable processes, IEE Control Theory & Applications 152 (2) (2003) 238–246.
[16] T. Liu, F. Gao, Closed-loop step response identification of integrating and unstable processes, Chemical Engineering Science 65 (10) (2010) 2884–2895.
[17] S. Majhi, D.P. Atherton, Modified Smith Predictor and controller for processes with time delay, Control Theory & Applications 146 (1999) 359–366.
[18] G. Marchetti, C. Scali, D.R. Lewin, Identification and control of open-loop unstable processes by relay methods, Automatica 37 (2001) 2049–2055.
[19] L. Mirkin, N. Raskin, Every stabilizing dead-time controller has an observer-predictor-based structure, Automatica 39 (2003) 1747–1754.
[20] B. Moore, Principal component analysis in linear systems: controllability, observability, and model reduction, IEEE Transactions on Automatic Control AC-26 (1981) 17–32.
[21] R.R. Negenborn, S. Leirens, B. De Schutter, J. Hellendoorn, Supervisory non linear MPC for emergency voltage control using Pattern Search, Control Engineering Practice 17 (2009) 841–848.
[22] S.I. Niculescu, Delay Effects on Stability. A Robust Control Approach, Springer-Verlag, 2001.
[23] J.E. Normey-Rico, E.F. Camacho, Control of Dead-time Processes, Springer-Verlag, 2007.
[24] I. Pan, S. Das, A. Gupta, Tuning of an optimal fuzzy PID controller with stochastic algorithms for networked control systems with random time delay, ISA Transactions 50 (2011) 28–36.
[25] A.M. De Paor, A Modified Smith Predictor and controller for unstable processes with time delay, International Journal of Control 41 (4) (1985) 1025–1036.
[26] A.M. De Paor, R.P.K. Egan, Extension and partial optimization of a Modified Smith Predictor and controller for unstable processes with time delay, International Journal of Control 50 (4) (1989) 1315–1326.
[27] W. Qian, J. Liu, Y. Sunb, S. Fei, A less conservative robust stability criteria for uncertain neutral systems with mixed delays, Mathematics and Computers in Simulation 80 (2010) 1007–1017.
[28] M. Chidambaram, R.P. Sree, M.N. Srinivas, A simple method of tuning PID controllers for stable and unstable FOPTD systems, Computers and Chemical Engineering 28 (2004) 2201–2218.
[29] V. Torczon, On the convergence of pattern search algorithms, SIAM Journal on Optimization 7 (1) (1997) 1–25.
[30] W.D. Zhang, Y.X. Sun, Modified Smith Predictor for controlling integrator/time delay processes, Industrial & Engineering Chemistry Research 35 (1996) 2769–2772.