

# A gradient-free adaptation method for nonlinear active noise control

Emanuele Spiriti, Simone Morici, Luigi Piroddi\*

*Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria, Via Ponzo 34/5, 20133 Milano, Italy*

*Article history:*

Received 7 March 2013

Received in revised form

23 July 2013

Accepted 4 September 2013

Handling Editor: J. Lam

Available online 8 October 2013

## 1. Introduction

ANC is a methodology that employs (secondary) acoustic sources to attenuate a (primary) acoustic noise exploiting the principle of destructive interference (see, e.g., [1,2]). In the feedforward ANC setting the controller is an adaptive digital filter that processes an input reference signal (correlated with the primary noise), in order to drive the secondary acoustic source to provide the appropriate canceling signal at a given measurement point. The residual noise measured by the error microphone is used to tune the filter parameters (sometimes called weights), typically based on a LMS-type rule.

The ANC scheme configures an indirect identification problem where the control filter is adapted without the direct knowledge of its desired output signal, the error being measured only *after* the secondary path (i.e., the electroacoustic

---

\* Corresponding author. Tel.: +39 02 2399 3556; fax: +39 02 2399 3412.

E-mail addresses: [emanuele.spiriti@gmail.com](mailto:emanuele.spiriti@gmail.com) (E. Spiriti), [simone.morici@gmail.com](mailto:simone.morici@gmail.com) (S. Morici), [luigi.piroddi@polimi.it](mailto:luigi.piroddi@polimi.it) (L. Piroddi).

system accounting for the actuation and measurement chain of the control loop, as well as the involved acoustic channel). Nevertheless, in the linear case, the control filter and secondary path can be swapped in the slow adaptation hypothesis, transforming the indirect identification problem into a direct one. As a result, the gradient of the error cost function can be conveniently approximated in terms of products of the current error with an auxiliary signal, obtained by filtering the input or output of the control filter through the secondary path. This is the rationale behind popular ANC algorithms such as the Filtered-x Least Mean Squares (FXLMS) or the Filtered-u LMS (FULMS) algorithms [1].

Unfortunately, this reasoning does not generally apply in the nonlinear case. ANC is affected by several sources of nonlinearity, such as distortion or saturation on the involved microphones, amplifiers, and loudspeakers [3]. For these reasons several nonlinear ANC (NANC) methods have been recently introduced in the literature, based on different nonlinear model structures of the control filter, such as truncated Volterra expansions [4–7], radial basis functions [4], multilayer artificial neural networks (MLANN) [3,8], functional link artificial neural networks (FLANN) with trigonometric [6,9–12], or piecewise linear functional expansions [13], adaptive bilinear filters [14], general function expansion (GFE) nonlinear filters [15], recursive second order Volterra (RSOV) filters [16], polynomial nonlinear autoregressive models with exogenous variables (NARX)<sup>1</sup> [17,18].

Adaptive algorithms of the LMS family can be extended to several of these model classes, in the simplifying assumption that the secondary path dynamics be linear. For example, the Volterra FXLMS (VFXLMS) [5], the Filtered-S LMS (FSLMS) [9,10,19], the Bilinear FXLMS (BFXLMS) [14], and the Filtered-x Affine Projection (FAP) algorithm for both the adaptive Volterra filter and the FLANN structures [12,13], are documented in the literature. Indeed, the assumption of the linearity of the secondary path together with the special structure of the control filter allows the commutation of the secondary path with the linear portion of the control filter (containing the weights), transforming the ANC problem into a direct model identification problem, as with the FXLMS/FULMS schemes [17].

In the more general hypothesis that the secondary path is nonlinear, this commutation is no longer feasible and the adaptation process remains unavoidably indirect. As a result, the gradient of the cost function with respect to the control filter weights depends on the input–output gradient of the secondary path, by application of the chain rule [3,15–17]. This complicates the exact computation of the gradient, since it involves nonlinear recursive filtering tasks [17]. Besides the high computational effort required, the stability of such filters (which depend on the current estimates of the parameters) cannot be guaranteed. This problem is typically solved by simplifying the expressions for the computation of the gradient disregarding the recursive terms, along with the well-known Feintuch's assumption [20]. For example, in the case of polynomial NARX models this leads to the Nonlinear Filtered Gradient LMS (NFGLMS) algorithm [17]. However, the complexity of the ANC scheme is still significant, and the impact of Feintuch's assumption in terms of convergence speed is not always negligible.

As noted in [17], the indirect structure of the adaptation process is also a complication for model structure selection [21], which is crucial in nonlinear black-box identification applications to avoid overparametrization with all its related effects (see, e.g., [22,23]). Model selection is particularly important in the ANC context to cut down the on-line computational load (by reducing the size of the estimated models) in order to meet the real-time requirements of the on-line operation. Besides, it may also have an impact on the noise reduction performance itself, since the parameters of more compact and precisely matched model structures are estimated [17]. In addition, model selection may be necessary to track structural variations of system dynamics. Unfortunately, none of the adaptive model selection approaches available in the literature (see, e.g., [24–28]) are suitable in the NANC context, since – again – they rely on a direct identification setting.

To circumvent these problems a nongradient-based error minimization algorithm is developed in this work for the adaptation of general nonlinear control filters of the polynomial NARX family [29], inspired by Stochastic Approximation (SA) methods and evolutionary optimization algorithms. These are optimization algorithms that operate without directly exploiting the gradient of the loss function, but using only measurements of the function itself and heuristic policies to move in the parameter space [30–33]. SA and evolutionary algorithms can be advantageous in ANC since they can operate without a secondary path model, which is required for gradient calculation only (see, e.g., [34–39]). However, in order to obtain reliable measurements of the loss function, a transient time must be allowed after each parameter perturbation is effected, with a consequent increase in the convergence time. This is all the more relevant given that acoustic paths are typically nonminimum phase. In addition, a temporary performance loss may be experienced when worsening directions are explored in the parameter space.

In this paper, an SA-type method is employed to address the NANC problem exploiting its gradient-free optimization feature, which allows to circumvent the costly recursions required to account for the secondary path input–output gradient in the general NANC adaptation schemes of [15,17]. Contrary to the mentioned approaches, a model of the secondary path (which can be accurately estimated and tracked with well assessed algorithms in practical applications) is also employed, to perform a virtual backward reconstruction of the effects of a given parameter perturbation, accelerating the process convergence and avoiding temporary performance losses. This virtualization scheme allows the *parallel* simulation of multiple filter instances, subject to different perturbations. Based on it, a weight adaptation algorithm for NARX models is first proposed that, similar to SA approaches, uses local information obtained by the evaluation of the loss function resulting

---

<sup>1</sup> Notice that the polynomial NARX class strictly includes truncated Volterra expansions and bilinear models as well as RSOV filters.

from suitable perturbations of the control filter parameters to establish a convenient update direction in the parameter space. The update rule follows a classical steepest-descent paradigm.

A model selection algorithm for NARX models is also designed based on a similar rationale, virtually evaluating the effect of different structural variations. An heuristic policy is used to progressively construct the model by adding and pruning terms. To avoid affecting the control performance during the selection process, a model of the secondary path is used to assess each tentative structural modification, by estimating its effect on the output performance on a virtual channel. This allows to evaluate different structural modifications in parallel, compatibly with the computational resources available. Given that model selection is intrinsically computationally intensive, a downsampling of this task is here suggested to fit the computational capacity of the control hardware, while the main control loop provides noise attenuation based on a provisional control filter structure.

The rest of the paper is organized as follows. [Section 2](#) illustrates the basic ANC scheme, discussing the implications of the nonlinear setting. [Section 3](#) provides a brief overview of SA-type methods for ANC applications and presents the suggested gradient-free weight update method for the full NANC context. The method is then extended to address the model selection task in [Section 4.2](#). Several simulation examples that show the effectiveness of the proposed methods are illustrated in [Section 5](#) before the final conclusions.

## 2. The ANC problem

### 2.1. The basic ANC scheme

The basic feedforward ANC scheme (see [Fig. 1](#)) requires the availability of a reference input signal  $x(n)$ , that must be highly correlated with the offending noise  $d(n)$ . The primary path  $P$  describes the model relating  $x(n)$  to  $d(n)$ , essentially accounting for the acoustic path involved in the noise transmission. The reference signal is processed by an adaptive digital filter  $C$ , whose output  $y(n)$  drives a secondary acoustic source, with the aim to generate a secondary signal  $y'(n)$  interfering negatively with the primary noise  $d(n)$  at the error microphone. The secondary path  $S$  describes the model between  $y(n)$  and  $y'(n)$ . The control filter parameters (called weights) are adapted so as to minimize the (mean square) measured error  $e(n) = d(n) - y'(n)$ . In the basic ANC setting, the acoustic paths are assumed linear and time-invariant (or at least sufficiently slowly time varying), so that they can be characterized by their transfer functions  $P(z)$  and  $S(z)$ , or, in the time domain, by their impulse responses,  $p(n)$  and  $s(n)$ .

In the simplest case, the adaptive filter is assumed to have a Finite Impulse Response (FIR) structure:

$$y(n) = \sum_{l=0}^{L-1} w_l(n)x(n-l) = \mathbf{w}(n)^T \mathbf{x}(n),$$

where  $\mathbf{x}(n) = [x(n) \ x(n-1) \ \dots \ x(n-L+1)]^T$  and  $\mathbf{w}(n) = [w_1(n) \ w_2(n) \ \dots \ w_{L-1}(n)]^T$  is the weight vector at time  $n$ ,  $L$  being the FIR length. The secondary signal is obtained as the linear convolution of the secondary path impulse response sequence  $s(n)$  and the filter's output  $y(n)$ :

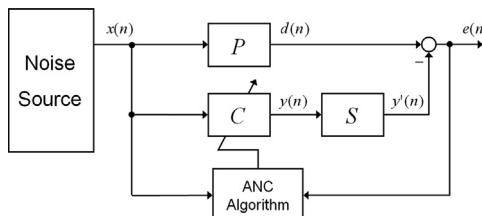
$$y'(n) = s(n) * y(n), \quad (1)$$

The weight update rule is the core of the feedforward ANC scheme, and is typically a simple gradient-based method of the LMS family. Indeed, under some simplifying assumptions, the gradient of the error cost function has a computationally convenient form. More precisely, expressing the error signal as

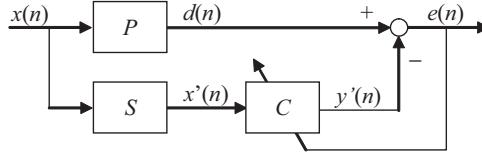
$$e(n) = d(n) - y'(n) = d(n) - s(n) * [\mathbf{w}(n)^T \mathbf{x}(n)]$$

and using the stochastic gradient approach [\[1\]](#) one obtains the following weight update equation:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{\mu}{2} \left( \frac{\partial e(n)^2}{\partial \mathbf{w}(n)} \right)^T = \mathbf{w}(n) + \mu \left( \frac{\partial y'(n)}{\partial \mathbf{w}(n)} \right)^T e(n), \quad (2)$$



**Fig. 1.** Block diagram of an ANC system.



**Fig. 2.** ANC scheme with commutation of  $C$  and  $S$ .

where  $\mu$  is the algorithm gain or step size. In the popular FXLMS algorithm [1], the derivative term in the previous expression is approximated as

$$\frac{\partial y'(n)}{\partial \mathbf{w}(n)} \cong s(n) * \mathbf{x}(n) = \mathbf{x}'(n),$$

where  $\mathbf{x}'(n) = [x'(n) \ x'(n-1) \ \dots \ x'(n-L+1)]^T$  and  $x'(n) = \hat{s}(n) * \mathbf{x}(n)$ ,  $\hat{s}(n)$  being an estimated version of  $s(n)$  [1]. This results in the following update equation:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mu \mathbf{x}'(n) e(n). \quad (3)$$

## 2.2. ANC as an indirect identification problem

The scheme of Fig. 1 configures a model identification problem, where the objective is to find a model of  $C$  such that the series between  $C$  and  $S$  emulates  $P$ , at least to the extent possible given the excitation characteristics of the reference signal  $x(n)$ . Such identification problem is *indirect*, in that the target output for model  $C$  is not accessible, and the only available information is not directly related to the output of  $C$ , but to a filtered version of it, through  $S$ .

Now, assume that blocks  $C$  and  $S$  could be commuted, as in Fig. 2 (an hypothesis which is commonly accepted in the linear context although  $C$  is time-varying, due precisely to the adaptation process). Then, the identification problem would be reformulated as a direct one, where the unknown system  $C$  has input  $x'(n)$  and  $e(n)$  represents the output error. Optimizing the parameters of  $C$  with the LMS algorithm, one would get exactly the FXLMS update law (3). Thus, block commutability of  $C$  and  $S$  stands at the basis of the FXLMS algorithm.

In the NANC context the commutability of blocks  $C$  and  $S$  does not generally apply. All the same, several methods have been developed for a specific sub-case, in which the secondary path is assumed linear, and the control filter can be decomposed as the series of a nonlinear block and a linear one, the latter containing the parameters. Then (in the slow adaptation assumption) commutability applies to  $S$  and the linear parameter-dependent part of  $C$ , resulting in a weight update law that extends the classical FXLMS approach. The mentioned VFXLMS, FSLMS, BFXLMS, FAP algorithms all fall into this category.

## 2.3. The full nonlinear case

Let us now consider the full nonlinear case. Without loss of generality we will assume that both systems  $C$  and  $S$  can be adequately described by NARX models, i.e. input–output recursive models where the current output is given by a nonlinear functional expansion of lagged inputs and outputs. Such model class is sufficiently general and subsumes all the cited nonlinear model structures.

Let the control filter be described as

$$y(n) = f(\mathbf{z}(n)), \quad (4)$$

where  $f(\cdot)$  is a generic nonlinear function and  $\mathbf{z}(n) = [y(n-1) \ \dots \ y(n-L) \ x(n) \ \dots \ x(n-L)]$ . Function  $f(\cdot)$  is often parameterized through a (truncated) polynomial expansion, so that (4) can be rewritten as a linear regression:

$$y(n) = \mathbf{w}(n)^T \mathbf{r}(\mathbf{z}(n)), \quad (5)$$

where  $\mathbf{w}(n) = [w_1(n) \ w_2(n) \ \dots \ w_R(n)]^T$  is the weight vector,  $\mathbf{r}(\mathbf{z}(n)) = [r_1(\mathbf{z}(n)) \ r_2(\mathbf{z}(n)) \ \dots \ r_R(\mathbf{z}(n))]^T$  is the regressor vector, with  $r_i(\mathbf{z}(n)) = \prod_{j=1}^{2L+1} z_j(n)^{l_{ij}}$ ,  $l_{ij}$  being nonnegative integers such that  $0 \leq \sum_{j=1}^{2L+1} l_{ij} \leq l$ , and  $l$  being the maximum degree of the expansion.

Similarly, let  $S$  be formulated as

$$y'(n) = g(\mathbf{z}'(n)) = \mathbf{v}^T \mathbf{q}(\mathbf{z}'(n)), \quad (6)$$

where  $\mathbf{z}'(n) = [y'(n-1) \ \dots \ y'(n-M) \ y(n) \ \dots \ y(n-M)]$ ,  $\mathbf{v} = [v_1 \ v_2 \ \dots \ v_Q]^T$  is the weight vector,  $\mathbf{q}(\mathbf{z}'(n)) = [q_1(\mathbf{z}'(n)) \ q_2(\mathbf{z}'(n)) \ \dots \ q_Q(\mathbf{z}'(n))]^T$  the regressor vector, and  $q_i(\mathbf{z}'(n)) = \prod_{j=1}^{2M+1} z'_j(n)^{m_{ij}}$ ,  $m_{ij}$  being nonnegative integers such that  $0 \leq \sum_{j=1}^{2M+1} m_{ij} \leq m$ .

The overall system with input  $x(n)$  and output  $y'(n)$  (i.e., the series of  $C$  and  $S$ ) cannot be generally characterized as a linear regression in the parameters  $\mathbf{w}(n)$ , even if both systems are described by linear-in-the-parameters models, unless very specific structural properties hold. One such condition occurs, e.g., if both models depend linearly on  $y(\cdot)$  [17]. Indeed, in that

case, the control filter can be reformulated (in operatorial notation) as

$$A(z)y(n) = \mathbf{w}_x(n)^T \mathbf{r}_x(\mathbf{x}(n)), \quad (7)$$

where  $A(z) = 1 - a_1 z^{-1} - \dots - a_L z^{-L}$  is a polynomial in the unit delay operator  $z^{-1}$  and  $\mathbf{r}_x$  is the vector of (nonlinear) regressors depending only on  $x(\cdot)$ . The vector of unknown parameters is  $\mathbf{w}(n) = [a_1 \dots a_L \mathbf{w}_x(n)^T]^T$ . Similarly, the secondary path becomes

$$y'(n) = \mathbf{v}_y^T \mathbf{q}_y(\mathbf{y}'(n)) + B(z)y(n), \quad (8)$$

where  $B(z) = 1 - b_1 z^{-1} - \dots - b_L z^{-M}$  is a polynomial in  $z^{-1}$ ,  $\mathbf{y}'(n) = [y'(n-1) \dots y'(n-M)]$  and  $\mathbf{q}_y$  is the vector of (nonlinear) regressors depending only on  $y'(\cdot)$ .

In these assumptions, the overall model becomes

$$A(z)(y'(n) - \mathbf{v}_y^T \mathbf{q}_y(\mathbf{y}'(n))) + \mathbf{w}_x(n)^T (B(z)\mathbf{r}_x(\mathbf{x}(n))), \quad (9)$$

which is a recursive equation with input  $x(n)$  and output  $y'(n)$  ( $y(n)$  does not appear anymore) that is linear in the parameters  $\mathbf{w}(n)$ . Then, LS-type algorithms can be used for parameter estimation.

#### 2.4. Gradient-based weight update algorithms for the nonlinear case

In the full nonlinear case, more complex algorithms are required to address the model updating task. Several gradient-based methods have been developed in the literature for this purpose (see, e.g., 15–17), whose distinctive features are the calculation of the error gradient as a (recursive) function of the input–output gradient of the secondary path and the use of Feintuch's assumption. We will here mainly refer to the NFGLMS algorithm [17], which is designed for recursive-type models of the NARX class.

The NFGLMS parameter update equation consists in the application of (2), where, using the notation  $y'_w(n) = \partial y'(n)/\partial \mathbf{w}(n)$  and  $y_w(n) = \partial y(n)/\partial \mathbf{w}(n)$ , the gradient term is computed using the recursive expression:

$$y'_w(n) = \frac{\partial y'(n)}{\partial \mathbf{z}'(n)} \mathbf{z}'_w(n), \quad (10)$$

where  $\mathbf{z}'_w(n) = [y'_w(n-1) \dots y'_w(n-M) y_w(n) \dots y_w(n-M)]$ , and the approximations  $\partial y'(n-k)/\partial \mathbf{w}(n) \approx \partial y'(n-k)/\partial \mathbf{w}(n-k) = y'_w(n-k)$  and  $\partial y(n-k)/\partial \mathbf{w}(n) \approx \partial y(n-k)/\partial \mathbf{w}(n-k) = y_w(n-k)$  have been used (in the slow convergence hypothesis).

The first factor on the RHS of (10) can be computed as

$$\frac{\partial y'(n)}{\partial \mathbf{z}'(n)} = \frac{\partial \mathbf{q}(\mathbf{z}'(n))^T}{\partial \mathbf{z}'(n)} \mathbf{v}, \quad (11)$$

where  $\partial q_i(\mathbf{z}'(n))/\partial z'_j(n)$  equals  $m_{ij} q_i(\mathbf{z}'(n))/z'_j(n)$  if  $m_{ij} \geq 1$ , and 0 otherwise,  $i = 1, \dots, Q, j = 1, \dots, 2M+1$ . The derivatives of  $y(n)$  appearing in the term  $\mathbf{z}'_w(n)$  of (10) also require the solution of suitable nonlinear recursions:

$$y_w(n) = \mathbf{r}(\mathbf{z}(n))^T + \frac{\partial \mathbf{r}(\mathbf{z}(n))^T}{\partial \mathbf{w}(n)} \mathbf{w}(n), \quad (12)$$

where  $\partial r_i(\mathbf{z}(n))/\partial z_j(n)$  equals  $l_{ij} r_i(\mathbf{z}(n))/z_j(n)$  if  $l_{ij} \geq 1$  and  $j \leq L$ , and 0 otherwise,  $i = 1, \dots, R, j = 1, \dots, 2L+1$  (the  $x(\cdot)$  terms are independent of  $\mathbf{w}(n)$ ).

Expressions (10) and (12) are complex nonlinear recursions, whose solution generally results in unacceptable computational load for ANC applications, let alone the problem of ensuring the stability of the corresponding (time varying) nonlinear filters. As typically done in ANC applications [1], Feintuch's assumption is used to eliminate all recursive terms from (10) and (12), resulting in the following simplified nonrecursive expressions that are used in the NFGLMS [17]:

$$y'_w(n) = \sum_{j=0}^M \frac{\partial y'(n)}{\partial y(n-j)} y_w(n-j) \quad (13)$$

$$y_w(n) = \mathbf{r}(\mathbf{z}(n))^T. \quad (14)$$

For additional details on the NFGLMS algorithm the interested reader is addressed to [17].

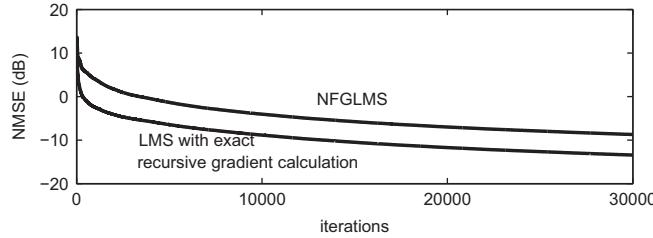
#### 2.5. Impact of the Feintuch assumption in the NANC case

While undoubtedly successful in reducing the computational load, these simplifications are not always without consequences on the convergence characteristics of the method. Consider, e.g., a NANC problem with the following secondary path model:

$$y'(n) = 0.1y'(n-1) + 0.3y(n) + 0.2y(n-1)y(n),$$

and a purely linear controller structure, for simplicity:

$$y(n) = w_1 y(n-1) + w_2 x(n).$$



**Fig. 3.** Impact of the Feintuch assumption.

The primary path coincides with the series of the controller, with parameters  $\mathbf{w} = [0.4 \ 0.2]$ , and the secondary path. Fig. 3 compares the NFGLMS to the exact gradient algorithm (i.e., employing expressions (10) and (12), starting from the initial parametrization  $\mathbf{w}_0 = [1 \ 0.2]$  and with a white noise reference input signal. The performance is measured in terms of the Normalized Mean Square Error (NMSE) index:

$$\text{NMSE}(n) = 10 \log_{10} \left( \frac{\sum_{i=1}^n e(i)^2}{n \cdot \text{var}(d(n))} \right). \quad (15)$$

Both methods eventually tend to the exact model parametrization, but the imprecise gradient calculation slows down the convergence of the NFGLMS algorithm, especially at the beginning of the adaptation process, when the model parametrization is far from the exact one.

### 3. A gradient-free weight update method

SA methods provide a convenient alternative that eliminates completely the mentioned nonlinear filtering operations (and, thereby, also the need to approximate them, as in the NFGLMS algorithm), by avoiding altogether the use of gradient information, and employing instead repeated measurements of the loss function and empirical weight perturbation policies. Methods of this type have been regarded as particularly appealing for NANC, in that they do not require the estimation of the secondary path model (see, e.g., [34,35]).

#### 3.1. Stochastic approximation methods

The simplest SA method is the Finite Difference Stochastic Approximation (FDSA) algorithm [30], which is essentially a gradient descent method, where the gradient is approximated by means of a finite difference. More in detail, denoting as  $J(\theta)$  the cost function, where  $\theta$  is the parameter vector, the  $i$ th component of the gradient is approximated as

$$J_{\theta_i}(\theta(n)) = \frac{J(\theta(n) + c(n)\mathbf{e}_i) - J(\theta(n) - c(n)\mathbf{e}_i)}{2c(n)}, \quad (16)$$

where  $\mathbf{e}_i$  is a unit vector with 1 at the  $i$ th component and 0 elsewhere, and  $c(n)$  is the magnitude of the perturbation (generally set to be decreasing with time). The parameter update is governed by

$$\theta(n+1) = \theta(n) - \mu(n)J_{\theta}(\theta(n))^T. \quad (17)$$

As many as  $2N$  cost function evaluations are required,  $N$  being the size of  $\theta$ , to obtain a gradient estimate. Furthermore, more than 1 sample may be necessary to robustly assess the value of  $J$  resulting from a parameter perturbation, raising to  $2NT$  the number of samples required to update the gradient,  $T$  being the time horizon used to evaluate the cost function. In all this period of time, the actual output performance may temporarily worsen since, on average, both good and bad directions are evenly explored.

The Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm [31] exploits the same principles but improves the performance by applying a parameter perturbation simultaneously to all elements of the parameter vector, thus reducing the number of cost function evaluations to 2. The gradient approximation becomes

$$J_{\theta_i}(\theta(n)) = \frac{J(\theta(n) + c(n)\delta(n)) - J(\theta(n) - c(n)\delta(n))}{2c(n)\delta(n)}, \quad (18)$$

where  $\delta(n)$  is the perturbation vector. The same update rule (17) applies to the SPSA algorithm. The convergence conditions discussed in [31] indicate that a convenient choice for the parameter perturbation is to extract it from a symmetric  $\pm 1$  Bernoulli distribution. The sequence of gains  $c(n)$  should also be decreasing to 0.

In the Particle Swarm Optimization (PSO) approach [33] a population of models (*particles*) is simultaneously adapted, sharing information related to the best performing particles to coordinate the population update. Each particle is characterized by a *position*  $\theta(n)$  (in the parameter space) and a *velocity*  $\mathbf{v}(n)$  (perturbation direction). At each update step all the particles'

positions and velocities are modified with respect to the particles that led to the best performances:

$$\theta(n+1)^{(i)} = \theta(n)^{(i)} + \mathbf{v}(n+1)^{(i)} \quad (19)$$

$$\mathbf{v}(n+1)^{(i)} = \rho \mathbf{v}(n)^{(i)} + \mu_1 r_1 (\theta(n)_{\text{best}}^{(i)} - \theta(n)^{(i)}) + \mu_2 r_2 (\theta(n)_{\text{best}} - \theta(n)^{(i)}), \quad (20)$$

where  $\rho$  is a leakage factor,  $\mu_1$  and  $\mu_2$  are gains,  $r_1$  and  $r_2$  are random numbers picked in  $[0, 1]$ ,  $\theta(n)_{\text{best}}^{(i)}$  and  $\theta(n)_{\text{best}}$  indicating the current best for the  $i$ th particle and the current best overall, respectively.

The SPSA algorithm has been applied for the reduction of periodic noise in a duct in [34], avoiding the use of secondary path information. Parameter perturbations are added and subsequently subtracted to the parameter vector and the output error of the ANC system is collected for a predefined number of steps to assess the effectiveness of each perturbation. An ANC application of the PSO approach is discussed in [35], where several control filters are updated in parallel. More precisely, each filter is fed with the same reference signal and outputs a different secondary signal. The different secondary signals are multiplexed in time, so that at each sample time the output error pertaining to a specific particle is recorded by the error microphone. Then, the parameters of each particle are updated with a logic similar to (19) and (20). Another PSO-based method designed for a decentralized multichannel ANC scheme is discussed in [36]. It is here worth mentioning that other types of evolutionary optimization algorithms have been exploited in the ANC context as well, to avoid secondary path modeling and local minima problems. For example, [37,38] propose ANC methods based on genetic algorithms, whereas a bacterial foraging optimization algorithm is employed in [39].

All the mentioned approaches suffer from the same two main drawbacks, related to the absence of a secondary path model. First of all, several consecutive steps are needed to reliably assess the effect of each perturbation, so that these methods will suffer from slow convergence problems. In addition, the noise attenuation performance is also affected, since worsening modifications may be tested at times. In view of these unwanted effects, their usage in on-line ANC schemes is questionable, unless it is really troublesome to obtain a reliable estimate of the secondary path model.

### 3.2. The virtualization scheme

On a different line, an SA-based approach is here adopted to address the NANC problem in the full nonlinear case because of its ability to circumvent the complexity of the gradient calculation, but a model of the secondary path is nevertheless employed to calculate *virtually* the effect of a model perturbation, providing an estimate of the control performance as a result of its application. This allows to decouple the parameter estimation phase from the actual control task, where the controller parameters are updated only in directions of expected performance improvement. In this way, both the noise reduction performance and the algorithm's convergence time can be improved.

With reference to the NANC setting where both the control filter and the secondary path are nonlinear systems, assume that the control filter is described by Eq. (4) and that a model for the secondary path is available in the form:

$$\hat{y}'(n) = \hat{g}(\hat{\mathbf{z}}'(n)) = \hat{\mathbf{v}}^T \hat{\mathbf{q}}(\hat{\mathbf{z}}'(n)), \quad (21)$$

where  $\hat{\mathbf{z}}'(n) = [\hat{y}'(n-1) \dots \hat{y}'(n-M) y(n) \dots y(n-M)]$ , and  $\hat{y}'(n)$  denotes the estimated secondary path output, with obvious notation (compare with (6)). Then, at each step an estimate of the noise can be obtained as

$$\hat{d}(n) = e(n) + \hat{y}'(n). \quad (22)$$

The obtained sequence  $\hat{d}(\cdot)$  can be used to evaluate at time step  $n$  the effect of a given perturbation on the current filter parameters, by simulating the series of (a copy of) the control filter (with perturbed parameters) and (a copy of) the estimated secondary path as shown in Fig. 4. The simulation is protracted for  $T$  steps starting from  $n-T+1$ ,  $T$  being a time horizon sufficiently long for the assessment of the perturbation effect, and at the same time not too long to significantly affect the overall computational load. In fact, in many simulations  $T=1$  provided more than satisfactory results.

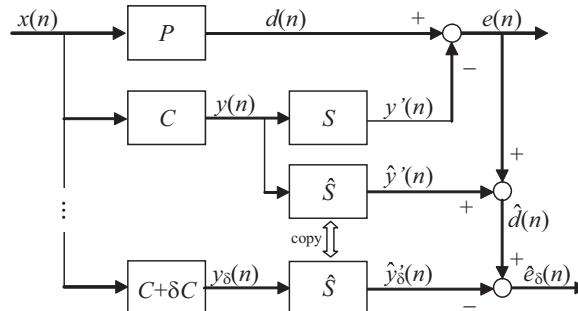


Fig. 4. The virtualization path.

### 3.3. The empirical weight update (EWU) algorithm

Based on this backward simulation scheme for the assessment of parameter perturbation effects, an SA-type policy can be suggested for the weight update, that tests different parameter perturbations in parallel, allowing to estimate the local gradient direction employing solely the corresponding cost function evaluations. More precisely, the EWU algorithm (see [Table 1](#)) operates similar to the FDSA algorithm, perturbing (in both directions) one parameter coordinate at a time.

At each time step  $n$ ,  $2N$  cost function evaluations are carried out, similar to the FDSA algorithm, but this time simultaneously, thanks to the use of the secondary path estimate. Parameter perturbations are performed on each parameter coordinate to estimate the gradient on each coordinate axis. Then, the improving perturbations are linearly combined to provide the actual parameter perturbation used to update the control filter.

The EWU algorithm employs several times the function eval() (see [Table 2](#)), which evaluates the virtual value of the cost function resulting from a parameter perturbation.

The EWU algorithm ends when perturbations do not lead to smaller errors for a predefined number of steps. The EWU converges generally much faster than the SA-based ANC schemes explained previously, thanks to the virtualization of the perturbation analysis. The algorithm is also insensitive to impulse-like error samples, which may blow up a gradient-based scheme.

### 3.4. Some remarks on the computational complexity of the EWU

[Table 3](#) compares the computational cost per iteration of the EWU scheme with that of the NFGLMS (a detailed comparison of the NFGLMS with several other nonlinear algorithms can be found in [\[17\]](#)). In the table  $R$  and  $Q$  denote the numbers of regressors of the control filter and secondary path model,  $M$  is the maximum lag in the latter model and  $T$  is the length of the evaluation period for the assessment of parametric perturbations.

The dominant cost of the EWU scheme is represented by the  $2R$  evaluations of the error function, each of which requires an order of  $2(R+Q)$  operations, for  $T$  samples. In the simplest case, in which  $T$  is set to 1, the cost is thus quadratic in the model size (notice however that the  $2R$  evaluations can be parallelized). This cost can be reduced by a factor  $R$  if an SPSA-type scheme is used, since in that algorithm a random perturbation is applied on the whole parameter vector at the same time. This extension is straightforward since it ultimately amounts to picking a direction at random in the parameter space

**Table 1**  
EWU iteration.

- 
- (1) Apply [\(4\)](#) to obtain the control output  $y(n)$ .
  - (2) Apply [\(21\)](#) to estimate the secondary path output  $\hat{y}'(n)$ . Then, measure  $e(n)$  and estimate the noise sample with [\(22\)](#).
  - (3) For  $i=1$  to  $R$  (number of weights) let  $\mathbf{w}_{i+}(n) = \mathbf{w}(n) + \mu_i \mathbf{e}_i$ , and calculate  $J_{i+}(n) = \text{eval}(\mathbf{w}_{i+}(n), T, n)$ .
  - (4) Repeat step (3) with  $\mathbf{w}_{i-}(n) = \mathbf{w}(n) - \mu_i \mathbf{e}_i$ , calculating  $J_{i-}(n) = \text{eval}(\mathbf{w}_{i-}(n), T, n)$ ,  $i = 1, \dots, R$ .
  - (5) For  $i=1$  to  $R$ , let  $J_i^{\min}(n) = \min(J(n), J_{i+}(n), J_{i-}(n))$ .
- Then, if  $J(n) = J_i^{\min}(n)$   $\delta w_i(n) = 0$  else if  $J_{i+}(n) = J_i^{\min}(n)$   $\delta w_i(n) = q_i \mu_i$  else  $\delta w_i(n) = -q_i \mu_i$ , where  $q_i = (J(n) - J_i^{\min}(n)) / \sqrt{\sum_{j=1}^R (J(n) - J_j^{\min}(n))^2}$  is a scaling factor that encourages perturbations in the most promising directions.
- (6) Finally, let  $\mathbf{w}(n+1) = \mathbf{w}(n) + \delta \mathbf{w}(n)$ .
- 

**Table 2**  
Function  $\bar{J} = \text{eval}(\mathbf{w}, T, n)$ .

- 
- (1) For  $k = n-T+1$  to  $n$  calculate  $\bar{y}(k) = \bar{\mathbf{w}}^T \mathbf{r}(\bar{\mathbf{z}}(k))$ , where  $\bar{\mathbf{z}}(k) = [\bar{y}(k-1) \dots \bar{y}(k-L) \mathbf{x}(k) \dots \mathbf{x}(k-L)]$ , and  $\bar{y}(k) = y(k)$  for  $k < n-T+1$ .
  - (2) For  $k = n-T+1$  to  $n$  calculate  $\bar{y}'(k) = \bar{\mathbf{v}}^T \mathbf{q}(\bar{\mathbf{z}}'(k))$ , where  $\bar{\mathbf{z}}'(k) = [\bar{y}'(k-1) \dots \bar{y}'(k-M) \bar{y}(k) \dots \bar{y}(k-M)]$ , and  $\bar{y}'(k) = \hat{y}'(k)$  for  $k < n-T+1$ .
  - (3) Calculate  $\bar{J} = (1/T) \sum_{k=n-T+1}^n (\hat{d}(k) - \bar{y}'(k))^2$ .
- 

**Table 3**  
Computational complexity of the EWU and NFGLMS algorithms.

Algorithm	No. of multiplications	No. of sums
NFGLMS	$(R+3Q)(M+1)+2R+Q+1$	$(M+2)(R+Q-1)-1$
EWU	$(2RT+1)(R+Q)+3R+1$	$(2RT+1)(R+Q-2)+6R$

and estimating the gradient on that direction by means of two error measurements. In this version the algorithm becomes linear in the model size  $R+Q$ , as opposed to the NFGLMS which has a quadratic dependence.

### 3.5. Step-size and time horizon adaptation

The step-size design is crucial for the convergence characteristics of the EWU. A fixed step-size may prevent the actual reaching of the desired minimum. For this reason, an adaptive scheme has been adopted that reduces by a prespecified factor the step-size on a specific coordinate axis if neither the positive nor the negative perturbation improve the performance. To this aim, the following instruction is added after step (5) of the EWU algorithm:

5') For  $i = 1$  to  $R$ , if  $J_i(n) = J_i^{\min}(n)$  then  $\mu_i(n+1) = \rho\mu_i(n)$ .

where  $0 < \rho < 1$  is the reduction factor.

If small  $T$  values are employed to save computational time, some imprecision is to be expected in the cost function evaluation. A consequence of this which is sometimes experienced in the algorithm usage, is that the step-size is taken prematurely to 0, before proper convergence. To prevent this, a gain increase was also introduced to reinforce good directions, adding the following instruction as well:

5'') For  $i = 1$  to  $R$ , if  $J_i^{\min}(k) = J_{i+}(k)$  for  $k = n-h+1$  to  $n$  or  
 $J_i^{\min}(k) = J_{i-}(k)$  for  $k = n-h+1$  to  $n$ , then  $\mu_i(n+1) = \eta\mu_i(n)$ .

Here  $\eta > 1$  is the amplification factor and  $h$  is the time horizon over which the action to be reinforced is consistently the same. For safety reasons a maximum threshold is set for  $\mu$ .

A particularly convenient version of the algorithm in terms of computational effort employs also an adaptation of parameter  $T$ . This parameter is normally set to 1 to speed up the algorithm, but when an apparent convergence has been reached, resulting in small  $\mu_i(n) \forall i$ , the step-size is tentatively reset to a pre-specified value  $\bar{\mu}$  and  $T$  is increased to robustify the estimation of the cost function associated to the different perturbed parameter vectors. If new improving directions of perturbation are found in this way, the weight vector is updated, the step-size in those coordinates is fixed to  $\bar{\mu}$ , and the normal update process is resumed with  $T=1$ .

## 4. The model selection task

The NARX model size (in terms of the number of regressors – and, consequently, parameters –  $R$ ) increases rapidly with  $L$  and  $l$  (curse of dimensionality), but in practice satisfactory models can be obtained with a small fraction of the terms. Therefore, a selection of the essential terms of the expansion to include in the model is crucial to avoid overparametrization issues.

Model structure selection is a formidably hard task, which takes much computational time to accomplish, and is especially hard to apply on-line, when strict real-time requirements are posed. However, it can help reducing significantly the model size, noticeably alleviating the ANC algorithm computational effort in the end. To get the benefits of model structure selection without affecting the efficiency of the ANC scheme, this task can be run in parallel to the main parameter update loop at a downsampled rate, distributing its load on larger time horizons.

### 4.1. Batch and adaptive model selection methods

Several model selection algorithms have been introduced in the literature both in the batch and the recursive case. Referring to the first class of methods, several algorithms rely on the use of Orthogonal Least Squares (OLS) to decouple the estimation of the parameters, such as the Forward Regression Orthogonal Estimator (FROE) [40] or the Fast Recursive Algorithm (FRA) [41]. An alternative technique based on an output error approach is developed in [23]. Statistical regularization methods have also been used for plain model reduction purposes, which is similar to model selection (see, e.g., [42–45]).

Adaptive model selection algorithms have also been developed in the literature. Among these the GFEX and GFSL (Givens rotation with Forward selection and EXponential/SLiding windowing) algorithms provide nice extensions of the OLS regression method to the adaptive case, [24,25]. Recursive regularization-based identification methods have also been recently proposed in the literature, [26,27]. Some recent developments in this area are provided in [28].

### 4.2. The heuristic iterative model selection policy

Unfortunately, all the mentioned (adaptive) selection methods are based on a direct identification setting and rely on a linear regression structure of the identification problem, which is not consistent with the general setting of the NANC problem. On the contrary, the EWU scheme is particularly suited for model selection, since testing a structural modification is not particularly different from testing a parametric modification. However, while the latter perturbs the current model only locally, a structural modification (i.e., the addition or pruning of a regressor) may provide a larger perturbation. Indeed,

**Table 4**

EWU-based model selection.

- 
- (0) Let  $\Phi = \{r_i(\mathbf{z}(n)), i = 1, \dots, R\}$  be the complete set of regressors (e.g., all the monomials of a polynomial expansion of degree  $l$  in the arguments  $\mathbf{z}(n) = [y(n-1) \dots y(n-L) \ x(n) \dots x(n-L)]$  for a given order  $L$ ).  
 Let also  $\Phi_{in} \subseteq \Phi$  and  $\Phi_{out} = \Phi \setminus \Phi_{in}$  be the sets of regressors included in and excluded from the model.  
 For example, one can initially set  $\Phi_{in} = \emptyset$ .  
 Let also  $\mathbf{r}_X$  denote the vector of regressors associated to a set of regressors  $X$ .
- (1) For each  $r_i(\mathbf{z}(n)) \in \Phi_{out}$ ,  $i = 1, \dots, |\Phi_{out}|$ , let  $X_i = \Phi_{in} \cup \{r_i(\mathbf{z}(n))\}$ . Apply the EWU algorithm with  $T = T_{sel}$  for  $\tau$  steps on a virtual path to estimate the parameters of the model  $y(n) = \mathbf{w}(n)^T \mathbf{r}_{X_i}$ , starting from  $\mathbf{w}(n) = \mathbf{0}$ .  
 At the end of the adaptation process calculate  $J_i(n) = eval(\mathbf{w}(n), T', n)$ , where  $T'$  is a sufficiently long horizon to evaluate the model performance, as well as the corresponding  $J(n)$  of the current model. If  $\min_i J_i(n) \leq J(n) - \delta J$ , then  $\Phi_{in} = X_i$ ,  $\Phi_{out} = \Phi \setminus \{r_i(\mathbf{z}(n))\}$ , where  $\delta J$  is a predefined threshold, else the algorithm ends.
- (2) If  $|\Phi_{in}| \geq 3$ , for each  $r_i(\mathbf{z}(n)) \in \Phi_{in}$ ,  $i = 1, \dots, |\Phi_{in}|$ , let  $X_i = \Phi_{in} \setminus \{r_i(\mathbf{z}(n))\}$  and proceed as in Step (1).  
 If  $\min_i J_i(n) \leq J(n)$ , then  $\Phi_{in} = X_i$ ,  $\Phi_{out} = \Phi \cup \{r_i(\mathbf{z}(n))\}$ .  
 Optional: Repeat Step (2) until no further regressor is pruned.
- 

the optimal parametrization of the model resulting from a structural modification is not necessarily proximal to that of the original model (at least in the common coordinates). As a consequence, to establish the effect of the structural variation the parameters of the modified model need to be adapted as well. For this reason, each structural modification is tested by applying the EWU algorithm (with  $T=1$ , for simplicity) for a given time period  $\tau$  on a virtual path, starting from null parametrization.

Following the heuristic scheme of [23], the model structure is progressively modified by applying elementary structural modifications (single regressor addition or pruning), as long as these modifications result in an improved model accuracy. At each iteration of the algorithm all possible terms are tested for inclusion in the model and the best one is added if it provides at least a pre-specified improvement. Each test consists in the application of the EWU algorithm for a given time period. All tests are run in parallel, exploiting the virtualization mechanism explained previously. After the specified time period, the corresponding cost functions are compared and the best model is retained. A pruning stage is then executed, with the same rationale, this time testing the elimination of a regressor from the model. A regressor can be pruned if its elimination from the model costs less than the accuracy improvement brought by the added regressor.

In detail, the model selection algorithm operates as described in Table 4.

The added computational complexity (multiple instances of the EWU algorithm must be executed in parallel) can be compensated by suitably downsampling the model selection task, distributing its load on larger time horizons. Notice, in this respect, that the first steps of the model selection procedure are quite fast, because the EWU is applied to (many) small-sized models. In any case, the main benefit of the model selection is to be evaluated in the long run, since a significant reduction in the model size may drastically cut the on-line computational effort (after the selection is completed). Moreover, the unpredictable effects of overparametrization are also avoided.

## 5. Simulation examples

### 5.1. Test 1: Weight adaptation

The first simulation example is designed to test the algorithm in an ideal condition where there exists an exact parametrization of the control filter, in order to see if it can retrieve the exact values of the parameters. The secondary path is given by

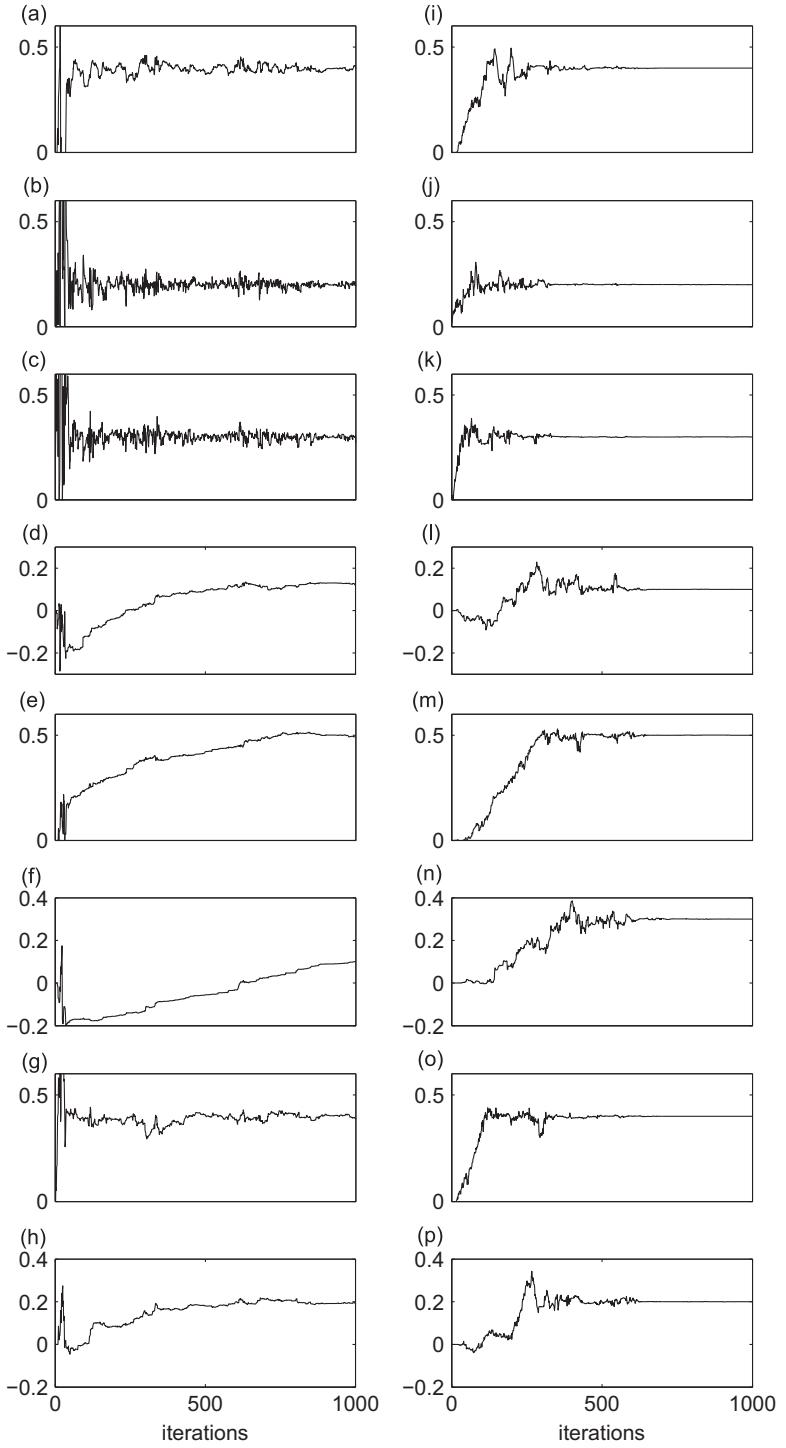
$$y'(n) = 0.1y'(n-1)y(n-2) + 0.3y(n) + 0.2y'(n-1)y(n) + 0.2y'(n-1)y(n)^2$$

and the controller structure is designed as follows:

$$\begin{aligned} y(n) = & w_1 y(n-5) + w_2 x(n-4) + w_3 x(n-1) + w_4 x(n-8)y(n-3) + w_5 x(n-3)y(n-8) \\ & + w_6 y(n-4)y(n-6) + w_7 x(n-6)x(n-8) + w_8 x(n-1)y(n-8) \end{aligned}$$

The primary path is designed so that there exists an ideal model for the control filter that solves the identification problem. Precisely, the model of the primary path is obtained as the series of the controller, with parameters  $\mathbf{w} = [0.4 \ 0.2 \ 0.3 \ 0.1 \ 0.5 \ 0.3 \ 0.4 \ 0.2]$ , and the secondary path.

Both the NFGLMS and the EWU algorithms have been tested on this example, employing the best step-size for each algorithm (obtained by trial and error). The corresponding parameter evolution is shown in Fig. 5. Both the algorithms achieve convergence, the EWU apparently being quicker and suffering less from parameter wandering (due to the stochastic gradient approximation) after the initial transient. Consider, e.g.,  $w_6$ : it takes the NFGLMS more than 3500 steps to reach convergence while the EWU converges to the optimal value after about 800 steps. The improved performance of the EWU compared to the NFGLMS is also apparent from Figs. 6 and 7, that report the evolution of the output signal and of the corresponding NMSE index.

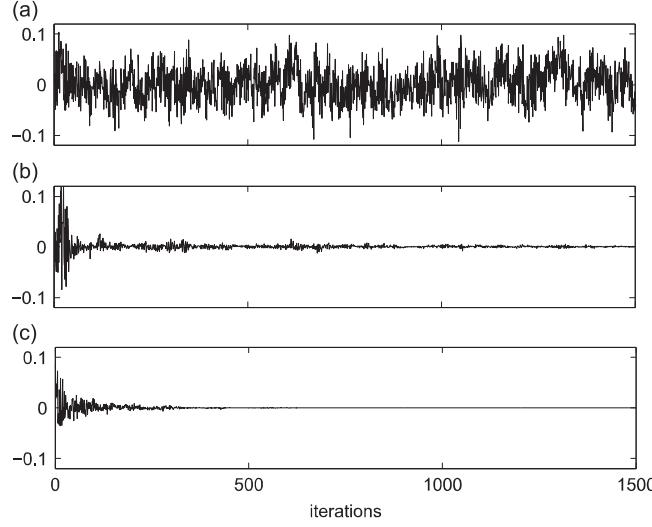


**Fig. 5.** Test 1: Adaptation of coefficients  $w_1$  to  $w_8$ , with NFGLMS (a–h) and EWU (i–p).

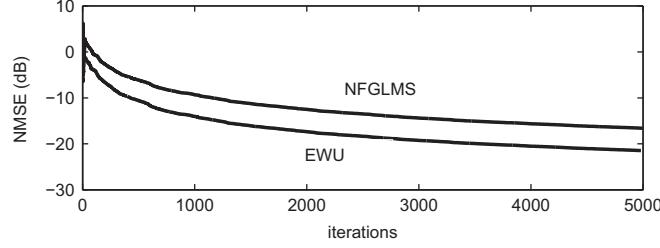
## 5.2. Test 2: Model selection in ideal conditions

The next simulation example is designed to test both the model selection and parameter update algorithms in an ideal condition where an exact structure and parametrization of the control filter exist. The secondary path is given by

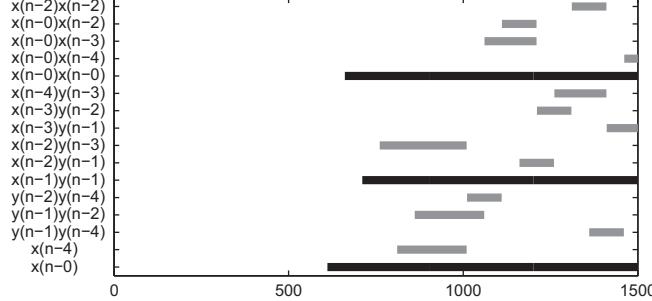
$$y'(n) = 0.1y'(n-1)y(n-2) + 0.3y(n) + 0.2y(n-1)y(n) + 0.2y'(n-1)y(n)^2$$



**Fig. 6.** Test 1: system output with ANC off (a), NFGLMS (b) and EWU (c).



**Fig. 7.** Test 1: NMSE of the NFGLMS and EWU algorithms.



**Fig. 8.** Test 2: Selected regressors during the model selection process (correct regressors in black).

and the controller structure is designed as follows:

$$y(n) = w_1 x(n)^2 + w_2 y(n-1)x(n-1) + w_3 x(n)$$

The primary path is designed so that there exists an ideal model for the control filter that solves the identification problem. Precisely, the model of the primary path is obtained as the series of the controller, with parameters  $\mathbf{w} = [0.4 \ 0.2 \ 0.3]$ , and the secondary path.

Model selection is conducted over a full quadratic NARX structure of order  $L=4$  (54 terms), with a downsampling factor of 50. In other words, the model selection process performs a step every 50 samples, using a window of 500 samples to evaluate the effect of a model structure modification ( $\tau = 500$  and  $T_{sel} = 1$ ). The selection process is started after 550 samples. New terms are included in the model if they provide a 15 percent improvement. By inspection of Fig. 8 (showing the regressors selected at least once), it is apparent that the correct regressors are picked out early in the selection process and never pruned. Comparatively, other regressors are selected only temporarily and for very small times. Furthermore,

when all three essential regressors have been included in the model their coefficients tend to the exact values, while spurious regressors are attributed negligible coefficient values (see Fig. 9).

### 5.3. Test 3: Model selection in nonideal conditions

The next test case is taken from [15] and is designed to test the model selection capabilities of the presented approach in a case where an exact model does not exist. The primary and secondary paths are given by the following nonlinear systems:

$$d(n) = x(n) + 0.8x(n-1) + 0.3x(n-2) + 0.4x(n-3) - 0.8x(n)x(n-1) + 0.9x(n)x(n-2) + 0.7x(n)x(n-3)$$

$$y'(n) = y(n) + 0.35y(n-1) + 0.09y(n-2) - 0.5y(n)y(n-1) + 0.4y(n)y(n-2)$$

The input signal is a zero mean white noise.

The EWU selection procedure operates on a full quadratic NARX model with maximum lag equal to 8. The initial model is assumed empty. A test-time of 500 samples was employed in the simulations and the selection procedure is started after the first 500 samples have been collected. The selection procedure is run at a rate of one step every 50 samples. The threshold for regressor inclusion in the model is a 15 percent improvement of the MSE. An initial step-size of 0.3 is assumed in the model selection phase. The final controller model structure and parameters are reported in Table 5.

The selection process (see Fig. 10) appears to catch consistently a subset of 5 terms which constitute the core of the model occasionally adding few terms with low coefficient values. A significant noise reduction is achieved early on during the selection process (see Fig. 11).

Fig. 12 compares the parameter estimation performance of the EWU (using the selected model), with that of the NFGLMS and the VFXLMS algorithms, according to the settings reported in Table 6. The step-size of each algorithm has been optimized independently to achieve the fastest convergence possible. The performance of the EWU almost equals that of the NFGLMS, implying that the model selection has made a good trade-off between model complexity and accuracy, thus succeeding in cutting the on-line computational requirements (notice that the number of parameters is reduced by more than one order of magnitude). Both methods outperform the employed Volterra filter.

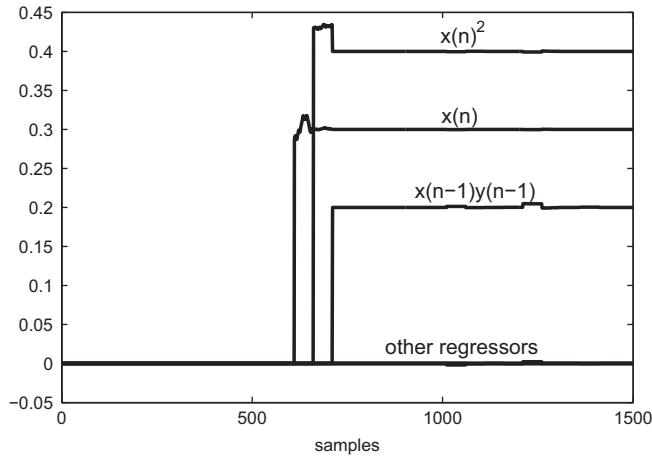
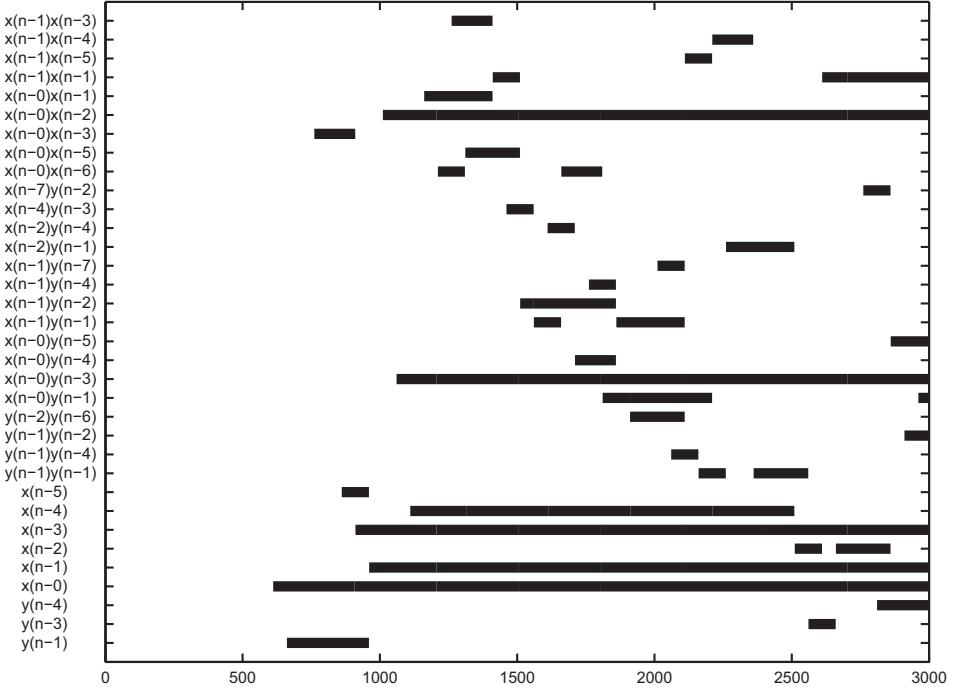


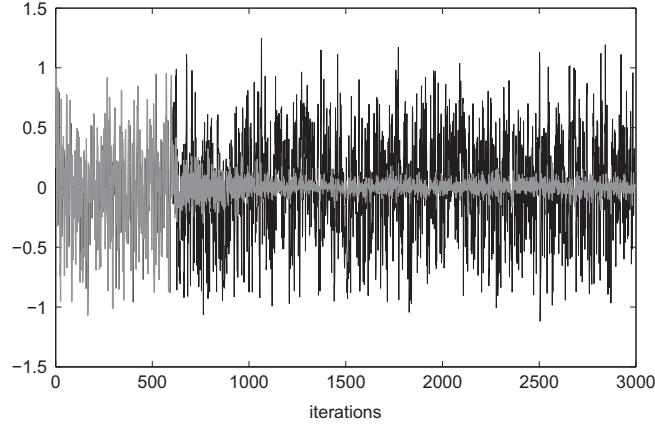
Fig. 9. Test 2: Values of the estimated coefficients during the model selection process.

Table 5  
Control filter for Test 3: regressors and parameters.

Regressors	Parameters
$x(n)$	0.9162
$x(n-1)$	0.4281
$x(n-3)$	0.3789
$x(n-4)$	-0.0864
$x(n)x(n-1)$	-0.2574
$x(n)x(n-2)$	0.6310
$x(n)x(n-5)$	-0.1718
$x(n)y(n-3)$	0.7190
$x(n-1)y(n-1)$	0.1083
$x(n-4)y(n-1)$	-0.0344



**Fig. 10.** Test 3: Regressor selection with the proposed approach.



**Fig. 11.** Test 3: Noise reduction during the selection process (original noise (black), reduced signal (gray); burn-in period of 500 samples).

#### 5.4. Test 4: The NANC scheme with saturating microphones

A particularly hard NANC problem concerns a system affected by microphone saturation both at the reference and the error microphone, as represented in Fig. 13, [14,17]. Following [17], the primary and secondary paths are modeled by two FIR filters, taken from [46]

$$\begin{aligned} d(n) &= 0.0179x(n) + 0.1005x(n-1) + 0.279x(n-2) + 0.489x(n-3) + 0.586x(n-4) + 0.489x(n-5) \\ &\quad + 0.279x(n-6) + 0.1005x(n-7) + 0.0179x(n-8), \\ y'(n) &= 0.7756y(n) + 0.5171y(n-1) + 0.362y(n-2). \end{aligned}$$

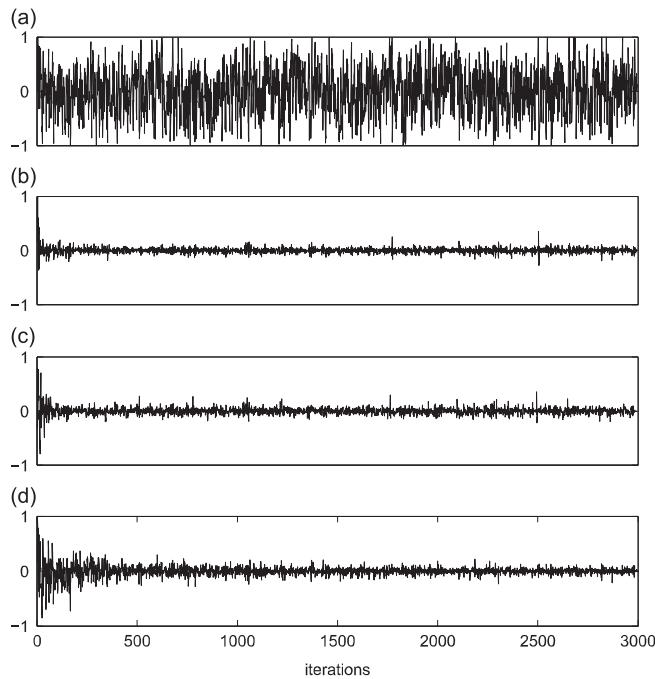
The reference and error microphone are subject to saturation, in the form of a 50 percent clipping of the signal with respect to its maximum value [14]. The reference signal is the sum of three sine waves at the normalized frequencies of 0.02, 0.04, and 0.08, with a sampling frequency of 8 kHz.

A quadratic NARX model with  $L=16$  (594 terms) is assumed as a starting point for the model selection phase. The model selection algorithm operates with a test-time equal to 500 and  $T_{\text{sel}} = 1$ . The algorithm performs a step every 50 samples. A core model with 3–4 terms is rapidly obtained that is capable of a significant noise reduction very early. The final selected

model includes only 10 terms and is given by

$$\begin{aligned}y(n) = & 0.5564y(n-1) - 0.0447y(n-6) + 0.392x(n-1) + 0.9098x(n-4) - 0.0314x(n-2)x(n-13) \\& - 0.0290x(n-1)y(n-7) + 0.0077x(n-2)y(n-9) \\& - 0.0217x(n-9)y(n-13) + 0.0669x(n-11)y(n-9) \\& - 0.0025y(n-1)y(n-14).\end{aligned}$$

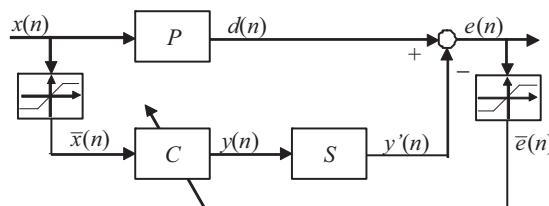
The following Figs. 14 and 15 report the attenuation performance achieved by the selected model using the EWU algorithm for on-line weight update. For comparison purposes, the performance of the NFGLMS, the VFXLMS [5], and the BFXLMS [14] algorithms, with the models and parameters listed in Table 7, is also reported. Notice that the larger the number of parameters, the smaller the step-size must be, to achieve convergence. As a consequence the EWU is significantly faster to converge than the other algorithms. Apparently, the compact size of the model, besides reducing of at least an order of magnitude the on-line computational effort, allows the update algorithm to converge faster and the steady-state noise reduction compares favorably with all the other methods. The result is all the more valuable, considering that in all simulations the EWU algorithm employs the FIR model of the secondary path as its estimate, not taking into account the presence of the saturation.



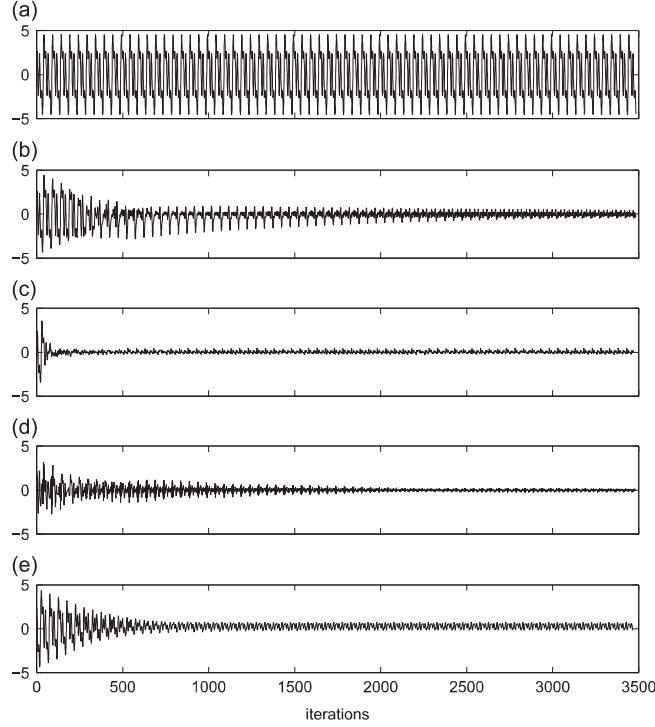
**Fig. 12.** Test 3: System outputs without ANC (a), with NFGLMS (b), EWU (c), and VFXLMS (d).

**Table 6**  
Test 3: Algorithm parameters.

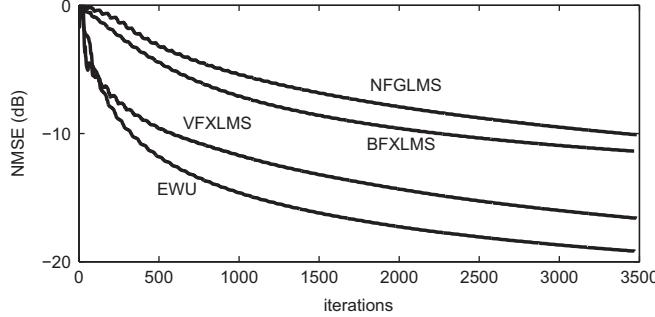
Algorithm	Step-size	Model type	Maximum lag	No. of terms
NFGLMS	0.5	Full quadratic model	8	170
EWU	0.03	Selected quadratic model	8	10
VFXLMS	0.05	Full quadratic Volterra series	8	54



**Fig. 13.** Block diagram of a NANC system in the presence of saturated reference and error signals.



**Fig. 14.** Test 4: Residual output with different ANC methods: (a) ANC off, (b) NFGLMS, (c) EWU, (d) VFXLMS, and (e) BFXLMS.



**Fig. 15.** Test 4: NMSE with different ANC methods.

**Table 7**  
Test 4: Algorithm parameters.

Algorithm	Step-size	Model type	Maximum lag	No. of terms
NFGLMS	$2 \times 10^{-5}$	Full quadratic model	16	594
EWU	0.02	Selected quadratic model	16	10
VFXLMS	$1 \times 10^{-4}$	Full quadratic Volterra series	16	170
BFXLMS	$3 \times 10^{-4}$	Full bilinear model	16	306

## 6. Conclusions

A gradient-free optimization method was employed in a general NANC setting to avoid the computational complexity and stability issues related to gradient-based approaches in the case of recursive nonlinear models. Contrary to other adaptations of SA algorithms to the ANC context previously presented in the literature, the availability of a secondary path

estimate has been assumed, making it possible to evaluate the effect of a given parameter perturbation virtually, without affecting the controller performance and improving the convergence properties. The EWU algorithm compares favorably with competitor NANC methods, such as the NFGLMS or the VFXLMS, both in terms of estimation accuracy and convergence speed.

An adaptive model selection algorithm for nonlinear model of the NARX family has also been developed based on the EWU algorithm, which is especially suited for indirect identification problems, as the NANC scheme. The model selection method has been tested on several cases, including a benchmark where the system nonlinearity is represented by microphone saturation.

## References

- [1] S.M. Kuo, D.R. Morgan, *Active Noise Control Systems: Algorithms and DSP Implementations*, Wiley-Interscience, New York (NY), USA, 1996.
- [2] S.J. Elliott, *Signal Processing for Active Control*, Academic Press, San Diego (CA), USA, 2001.
- [3] S.D. Snyder, N. Tanaka, Active control of vibration using a neural network, *IEEE Transactions on Neural Networks* 6 (1995) 819–828.
- [4] P. Strauch, B. Mulgrew, Active control of nonlinear noise processes in a linear duct, *IEEE Transactions on Signal Processing* 46 (1998) 2404–2412.
- [5] L. Tan, J. Jiang, Adaptive Volterra filters for active control of nonlinear noise processes, *IEEE Transactions on Signal Processing* 49 (2001) 1667–1676.
- [6] E.P. Reddy, D.P. Das, K.M.M. Prabhu, Fast adaptive algorithms for active control of nonlinear noise processes, *IEEE Transactions on Signal Processing* 56 (2008) 4530–4535.
- [7] H. Zhao, X. Zeng, J. Zhang, Z. He, T. Li, W. Jin, Adaptive extended pipelined second-order Volterra filter for nonlinear active noise controller, *IEEE Transactions on Audio, Speech, and Language Processing* 20 (2012) 1394–1399.
- [8] M. Bouchard, B. Paillard, C.T.L. Dinh, Improved training of neural networks for the nonlinear active control of sound and vibration, *IEEE Transactions on Neural Networks* 10 (1999) 391–401.
- [9] D.P. Das, G. Panda, Active mitigation of nonlinear noise processes using a novel filtered-s LMS algorithm, *IEEE Transactions on Speech and Audio Processing* 12 (2004) 313–322.
- [10] D.P. Das, S.R. Mohapatra, A. Routray, T.K. Basu, Filtered-s LMS algorithm for multichannel active control of nonlinear noise processes, *IEEE Transactions on Audio, Speech, and Language Processing* 14 (2006) 1875–1880.
- [11] H. Zhao, X. Zeng, J. Zhang, Adaptive reduced feedback FLNN nonlinear filter for active control of nonlinear noise processes, *Signal Processing* 90 (2010) 834–847.
- [12] G.L. Sicuranza, A. Carini, A generalized FLANN filter for nonlinear active noise control, *IEEE Transactions on Audio, Speech, and Language Processing* 19 (2011) 2412–2417.
- [13] G.L. Sicuranza, A. Carini, Piecewise-linear expansions for nonlinear active noise control, *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, pp. 209–212.
- [14] S. Kuo, H.-T. Wu, Nonlinear adaptive bilinear filters for active noise control systems, *IEEE Transactions on Circuits and Systems – I: Regular Papers* 52 (2005) 1–8.
- [15] D. Zhou, V. DeBrunner, Efficient adaptive nonlinear filters for nonlinear active noise control, *IEEE Transactions on Circuits and Systems – I: Regular Papers* 54 (2007) 669–681.
- [16] H. Zhao, X. Zeng, Z. He, T. Li, Adaptive RSOV filter using the FELMS algorithm for nonlinear active noise control systems, *Mechanical Systems and Signal Processing* 34 (2013) 378–392.
- [17] R. Napoli, L. Piroddi, Nonlinear active noise control with NARX models, *IEEE Transactions on Audio, Speech, and Language Processing* 18 (2010) 286–295.
- [18] D. Delvecchio, L. Piroddi, A nonlinear active noise control scheme with on-line model structure selection, *Proceedings of the 2011 IEEE Conference on Decision and Control*, Orlando (FL), USA, pp. 8014–8019.
- [19] D.P. Das, G. Panda, D.K. Nayak, Development of frequency domain block filtered-s LMS (FBFSLMS) algorithm for active noise control system, *Proceedings of the 2006 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Toulouse, France, pp. 289–292.
- [20] P.L. Feintuch, An adaptive recursive LMS filter, *Proceedings of the IEEE* 64 (1976) 1622–1624.
- [21] X. Hong, R.J. Mitchell, S. Chen, C.J. Harris, K. Li, G.W. Irwin, Model selection approaches for non-linear system identification: a review, *International Journal of Systems Science* 39 (2008) 925–946.
- [22] L.A. Aguirre, S.A. Billings, Dynamical effects of overparametrization in nonlinear models, *Physica D* 80 (1995) 26–40.
- [23] L. Piroddi, W. Spinelli, An identification algorithm for polynomial NARX models based on simulation error minimization, *International Journal of Control* 76 (2003) 1776–1871.
- [24] W. Luo, S.A. Billings, On-line structure detection and parameter estimation with exponential windowing for nonlinear systems, *European Journal of Control* 2 (1996) 291–304.
- [25] W. Luo, S.A. Billings, Adaptive model selection and estimation for nonlinear systems using a sliding data window, *Signal Processing* 46 (1995) 179–202.
- [26] P. Garrigues, L. El Ghaoui, An homotopy algorithm for the lasso with online observations, *Proceedings of the 2008 Conference on Neural Information Processing Systems*, pp. 489–496.
- [27] D. Angelosante, G.B. Giannakis, RLS-weighted lasso for adaptive estimation of sparse signals, *Proceedings of the 2009 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Taipei, Taiwan.
- [28] C. Cantelmo, L. Piroddi, Adaptive model selection for polynomial NARX models, *IET Control Theory and Applications* 4 (2010) 2693–2706.
- [29] I.J. Leontaritis, S.A. Billings, Input-output parametric models for non-linear systems – part II: stochastic non-linear systems, *International Journal of Control* 41 (1985) 329–344.
- [30] J. Kiefer, J. Wolfowitz, Stochastic estimation of the maximum of a regression function, *The Annals of Mathematical Statistics* 23 (1952) 462–466.
- [31] J.C. Spall, Multivariate stochastic approximation using a simultaneous perturbation gradient approximation, *IEEE Transactions on Automatic Control* 37 (1992) 332–341.
- [32] J.C. Spall, An overview of the simultaneous perturbation method for efficient optimization, *Johns Hopkins APL Technical Digest* 19 (1998) 482–492.
- [33] J. Kennedy, R. Eberhart, Particle swarm optimization, *Proceedings of the 1995 IEEE International Conference on Neural Networks*, Vol. 4, Perth, Australia, pp. 1942–1948.
- [34] Y.-L. Zhou, Q.-Z. Zhang, X.-D. Li, W.-S. Gan, On the use of an SPSA-based model-free feedback controller in active noise control for periodic disturbances in a duct, *Journal of Sound and Vibration* 317 (2008) 456–472.
- [35] N.K. Rout, D.P. Das, G. Panda, Particle swarm optimization based active noise control algorithm without secondary path identification, *IEEE Transactions on Instrumentation and Measurement* 61 (2012) 554–563.
- [36] N.V. George, G. Panda, A particle-swarm-optimization-based decentralized nonlinear active noise control system, *IEEE Transactions on Instrumentation and Measurement* 61 (2012) 3378–3386.
- [37] F. Russo, G.L. Sicuranza, Accuracy and performance evaluation in the genetic optimization of nonlinear systems for active noise control, *IEEE Transactions on Instrumentation and Measurement* 56 (2007) 1443–1450.
- [38] C.Y. Chang, D.R. Chen, Active noise cancellation without secondary path identification by using an adaptive genetic algorithm, *IEEE Transactions on Instrumentation and Measurement* 59 (2010) 2315–2327.

- [39] S. Gholami-Boroujeny, M. Eshghi, Non-linear active noise cancellation using a bacterial foraging optimisation algorithm, *IET Signal Processing* 6 (2012) 364–373.
- [40] M. Korenberg, S.A. Billings, Y.P. Liu, P.J. McIlroy, Orthogonal parameter estimation algorithm for non-linear stochastic systems, *International Journal of Control* 48 (1988) 193–210.
- [41] K. Li, J.-X. Peng, G.W. Irwin, A fast nonlinear model identification method, *IEEE Transactions on Automatic Control* 50 (2005) 1211–1216.
- [42] S. Chen, E.S. Chng, K. Alkadhimy, Regularised orthogonal least squares algorithm for constructing radial basis function networks, *International Journal of Control* 64 (1996) 829–837.
- [43] X. Hong, M. Brown, S. Chen, C.J. Harris, Sparse model identification using orthogonal forward regression with basis pursuit and d-optimality, *IEE Proceedings Control Theory and Applications* 151 (2004) 491–498.
- [44] S.L. Kukreja, J. Löfberg, M.J. Brenner, A least absolute shrinkage and selection operator (lasso) for nonlinear system identification, *2006 IFAC Symposium on System Identification*, pp. 814–819.
- [45] M. Bonin, V. Seghezza, L. Piroddi, NARX model selection based on simulation error minimization and LASSO, *IET Control Theory & Applications* 4 (2010) 1157–1168.
- [46] O. Tobias, R. Seara, Performance comparison of the FXLMS, nonlinear FXLMS and leaky FXLMS algorithms in nonlinear active noise applications, *Proceedings of the 2002 IEEE European Signal Processing Conference*, Vol. 1, Toulouse, France, pp. 155–158.