Nonlinear System Identification with LAR

Catia Valdman, Marcello L. R. de Campos and José A. Apolinário Jr.

***Abstract*— In this paper, the use of the Least Angle Regression (LAR) algorithm in combination with a Volterra filter is proposed for nonlinear system identification. The LAR algorithm has been used successfully in applications with sparse systems. Volterra filters are known as a good model of nonlinear systems and have been useful in a number of applications. However, only low order Volterra models are usually employed due to the large number of coefficients. Since the LAR algorithm indicates the most correlated coefficients in an increasing way, one by one, we propose to use this information to stop the LAR algorithm when a number of desired coefficients are already calculated. Hence, for a large order Volterra filter, the most important coefficients will be evaluated independently of its kernel position. To validate the proposition, we use third-order and fifth-order Volterra filters with the LAR algorithm to identify two nonlinear systems. Results of the LAR algorithm are compared to results of the Least Squares and the Subset Selection algorithms.**

***Index Terms*— LAR algorithm, nonlinear system, Volterra filter.**

***Abstract*— Neste artigo o uso do algoritmo Least Angle Regression (LAR) em conjunto com um filtro Volterra é proposto para a identificação de sistemas não lineares. O algoritmo LAR vem sendo usado em sistemas esparsos com resultados satisfatórios. Filtros Volterra são conhecidos por serem eficientes para modelagem de sistemas não lineares e já foram utilizados em diversas aplicações. Entretanto, na maioria das vezes, apenas modelos Volterra de baixa ordem são utilizados devido ao seu elevado número de coeficientes. Uma vez que o algoritmo LAR indica os coeficientes mais correlatos acrescentando-os ao modelo sempre de um em um, nossa proposta é utilizar esta informação para interromper o algoritmo LAR quando um número desejado de coeficientes já tiver sido calculado. Desta forma, para um filtro Volterra de alta ordem, os coeficientes mais importantes serão estimados, independente de sua posição no kernel. Para validar esta proposta, utilizamos um filtro Volterra de terceira ordem e um filtro Volterra de quinta ordem para identificar dois sistemas não lineares. Os resultados obtidos pelo algoritmo LAR são comparados com os resultados dos algoritmos Least Squares e Subset Selection.**

***Index Terms*— Algoritmo LAR, sistema não linear, filtro Volterra.**

I. Introduction

Nonlinear system models are used in many areas, such as communication systems, power amplifiers, loudspeakers with harmonic distortion and others [1]. The Volterra filter is commonly used to identify nonlinear systems, however, standard approaches tend to limit the order of the filter to avoid a large number of coefficients. For example, in [2] and [3], adaptive second-order Volterra filters were used to model nonlinear acoustic echo paths using the NLMS algorithm. In [4], stationary and non-stationary signals which arise from Volterra models were estimated using neural networks; again, the second-order Volterra model was considered sufficient for the purpose. In [5], a study was carried out for several algorithms combined with Volterra filter to identify nonlinear systems. The algorithms studied were: LMS, NLMS, RLS, affine projection and summation affine projection; once again, only second-order nonlinear components were treated [5].

The LAR algorithm was first developed and based on diabetes studies [6]. Since then, it has been used in several applications. In [7] and [8], models to text classification were developed and up to 2,000 coefficients were chosen. In [9], the LAR algorithm was used to estimate performance variability of integrated circuits with a larger number of coefficients, in the order of 10^4 to 10^6. By comparing the response of the LS and the LAR algorithms, the authors concluded that the LAR algorithm achieves up to 25x runtime speedup without compromising any accuracy [9]. Recently, it has been employed in image processing, for face representation and recognition [10], and for face age estimation [11].

Based on the fact that the LAR algorithm is very useful when dealing with sparse systems, indicating the most important coefficients to be used, and on its proven success in several applications, we propose its use in combination with the Volterra filter to identify nonlinear systems. In [12], one nonlinear system using a third-order Volterra filter and one using a fifth-order Volterra filter were identified with the LAR algorithm. In this paper we provide a more in depth description of this algorithm and add a new simulation where the coefficients have higher magnitudes.

This paper is organized as follows. Section II describes how a nonlinear system can be modeled and the importance of the Volterra filter for this task. The LAR algorithm is addressed in Section III. The proposed configuration is tested in simulated scenarios and the results are shown in Section IV for Volterra filters of third and fifth orders. Conclusions are drawn in Section V.

II. Nonlinear systems and the Volterra Series

Certain classes of nonlinear systems can be represented by one of the three following cascade models [13]:

- LN – a linear filter followed by a memoryless nonlinearity, known as the Wiener model;
- NL – a memoryless nonlinearity followed by a linear filter, known as the Hammerstein model; and
- LNL – a linear filter, a memoryless nonlinearity and a second linear filter.

It may be desirable to model them using a single Volterra based filter [13], i.e., to use a Volterra series for describing the

Manuscrito recebido em 30 de novembro de 2010; revisado em 1 de abril de 2011.

C. Valdman (catia@valdman.com) and M. L. R. de Campos (mcampos@ieee.org) are with the Federal University of Rio de Janeiro - UFRJ - P.O. Box 68504 - 21941–972 - Rio de Janeiro - RJ - Brazil.

J. A. Apolinário Jr. (apolin@ime.eb.br) is with the Military Institure of Engineering - IME - Praça General Tibúrcio, 80 - 22290–270 - Rio de Janeiro - RJ - Brazil.

input-output relationship of a nonlinear device with memory [14].

One advantage of using LNL system modeling is the small number of coefficients when compared to the Volterra filter. However, computing LNL coefficients may be more complicated than calculating coefficients based on the Volterra filter [14].

The Volterra series is a generalized extension of the linear series and can be regarded as a general Taylor series of a function with memory [14]. A discrete time-invariant and causal nonlinear system with memory can be represented by the following Volterra series expansion [5]

$$\begin{aligned} y(k) &= h_0 + \sum_{m_1=0}^{\infty} h_1(m_1)u(k-m_1) \\ &\quad + \sum_{m_1=0}^{\infty}\sum_{m_2=0}^{\infty} h_2(m_1,m_2)u(k-m_1)u(k-m_2) \\ &\quad + \cdots + \sum_{m_1=0}^{\infty}\sum_{m_2=0}^{\infty}\cdots\sum_{m_l=0}^{\infty} h_l(m_1,\cdots,m_l) \\ &\quad u(k-m_1)u(k-m_2)\cdots u(k-m_l) + \cdots \end{aligned} \quad (1)$$

where $u(k)$ is the input signal, $y(k)$ is the output signal and $h_l(m_1,\cdots,m_l)$ is l-th order discrete Volterra kernel, i.e., h_0 is the bias coefficient (DC component), $h_1(m_1)$ are the linear coefficients, $h_2(m_1,m_2)$ are the quadratic coefficients, $h_3(m_1,m_2,m_3)$ are the cubic coefficients, and so on [4]. The number of coefficients in the kernel in a truncated Volterra representation (finite number of delays) is calculated by [15]

$$\#\text{Coefficients} = \left(\begin{array}{c} l+m+1 \\ l \end{array}\right) = \frac{(l+m+1)!}{l!(m+1)!} \quad (2)$$

where l is the Volterra order (nonlinear degree), and m is its memory length (also known as dynamic order).

In signal processing, it is usual to represent the truncated Volterra filter in a vector way [14]. For example, for a second-order Volterra filter with memory length of one, i.e., $l = 2$ and $m = 1$, the output signal is given by

$$y = \mathbf{h}^T\mathbf{u},$$

being

$$\mathbf{u} = [u(k) \;\; u(k-1) \;\; u^2(k) \;\; u(k)u(k-1) \;\; u^2(k-1)]^T$$

the input signal and

$$\mathbf{h} = [h_1(0) \;\; h_1(1) \;\; h_2(0,0) \;\; h_2(0,1) \;\; h_2(1,1)]^T$$

the coefficient vector, where h_0 is not represented. For example, with

$$\mathbf{h} = [1 \;\; -0.5 \;\; 0 \;\; -0.1 \;\; 0]^T,$$

the output signal is

$$y = u(k) - 0.5u(k-1) - 0.1u(k)u(k-1).$$

This paper works with the third-order ($l = 3$) and fifth-order ($l = 5$) filters and assumes that all the Volterra kernels have finite memory length of four ($m = 4$) and six ($m = 6$), respectively. Therefore, Eq. (2) yields either 55 or 791 possible coefficients in the kernel (plus the DC component).

The main problem using Volterra-based adaptive filters is that it is usually not practical, either because it takes a long time to converge when using LMS-like algorithms, or because it is too complex (too many coefficients) for LS-based algorithms. However, modeling an LNL nonlinear system using a Volterra filter yields a large number of zero coefficients. Because of this sparsity characteristic, it is advantageous to combine the Volterra filter with the LAR algorithm, described in the next section.

III. THE LAR ALGORITHM

The Least Angle Regression (LAR) algorithm is a versatile linear model algorithm first developed in 2004 [6]. LAR is a forward stepwise algorithm, i.e., at each iteration a new coefficient is added to the model [16]. Due to its increasing order characteristic, it can be very useful to identify systems with many null coefficients, such as nonlinear systems.

In order to build the model, as described in [6] and [16], the LAR algorithm uses two main variables: the prediction vector ($\tilde{\mathbf{y}}$) and the correlation vector ($\mathbf{c}$). Both are updated at each iteration and should be defined as $\tilde{\mathbf{y}}_{step}$ and $\mathbf{c}_{step}$ where $step$ is the algorithm iteration, but, for conciseness, the argument $step$ will be omitted. In that way, the first variable is defined as

$$\tilde{\mathbf{y}} = \mathbf{X}\mathbf{w}_{LAR}, \quad (3)$$

where $\mathbf{w}_{LAR}$ is the estimated coefficient vector and $\mathbf{X}$ is the $K \times J$ input signal matrix defined as

$$\mathbf{X} = [\mathbf{x}(1)\cdots\mathbf{x}(k)\cdots\mathbf{x}(K)]^T \quad (4)$$

being $\mathbf{x}(k)$ the input signal vector defined as

$$\mathbf{x}(k) = [x_1(k) \;\cdots\; x_j(k) \;\cdots\; x_J(k)]^T \quad (5)$$

where k is the time index, $k = 1, 2, \cdots, K$, and j is the coefficient index, $j = 1, 2, \cdots, J$. In Eq. (5), $x_j(k)$ is the input signal of the jth coefficient at time instant k. The second main variable, the correlation vector, is defined as

$$\mathbf{c} = \mathbf{X}^T(\mathbf{y} - \tilde{\mathbf{y}}) \quad (6)$$

being $\mathbf{y}$ the reference signal vector defined as

$$\mathbf{y} = [y(1)\cdots y(k)\cdots y(K)]^T. \quad (7)$$

In Eq. (6), the elements of vector $\mathbf{c}$ are designed as $c(j)$, the correlation value of the jth coefficient.

The key of the LAR algorithm is how to compute the prediction vector ($\tilde{\mathbf{y}}$) and, hence, estimate the coefficient vector, $\mathbf{w}_{LAR}$ from Eq. (3). The prediction vector is influenced by the coefficients in the active set ($\mathcal{A}$), the subset of non-zero coefficients. The coefficients that are not in the active set compose the inactive set ($\mathcal{I}$), meaning that they are still equal to zero. At each iteration, a new coefficient is transfered from the inactive set to the active set, meaning that this new coefficient is, together with the others already in the active set, more relevant to form the output signal. All coefficients in the active set are equally correlated, i.e., they have the same value of $|c(j)|, j \in \mathcal{A}$.

The prediction vector is initialized to be equal to zero. In the first step, the coefficient with the highest value of

$|c(j)|$, i.e. the most correlated, is identified and the prediction vector is implemented in its direction. The prediction vector is implemented as much as another coefficient is as correlated as the first one, i.e., having the same value of $|c(j)|, j \in \mathcal{A}$. Then, in the second step, instead of implementing the prediction vector in the direction of the second most correlated coefficient, the algorithm proceeds to an equiangular direction between the two most correlated coefficients. This procedure continues throughout the execution of the algorithm: the direction followed by the prediction vector is equiangular among the coefficients in the active set, thus giving the name of the algorithm Least Angle Regression. Therefore, the maximum number of steps, or algorithm iterations, corresponds to the total number of coefficients. The LAR pseudocode is shown in Algorithm 1, whereas a detailed explanation can be found in [6].

Algorithm 1 LAR

input $\mathbf{X}$, $\mathbf{y}$
$\tilde{\mathbf{y}} \leftarrow 0$
for $step = 1$ to J **do**
  $\mathbf{c}_{step} \leftarrow \mathbf{X}^T(\mathbf{y} - \tilde{\mathbf{y}})$
  $C \leftarrow max(\mathbf{c}_{step})$
  if $step = 1$ **then**
    $\mathcal{A} \leftarrow j$ which resulted C
  else
    $\mathcal{A} = \mathcal{A} \cup j$
  end if
  $\mathcal{I} = j \notin \mathcal{A}, j = 1, \cdots, J$
  $\mathbf{X}_A \leftarrow \{\}$
  for $j = 1$ to length($\mathcal{A}$) **do**
    $s_j = \text{sign}(c_{step}(\mathcal{A}(j)))$
    $\mathbf{X}_A = \mathbf{X}_A \cup s_j.\mathbf{x}_j$
  end for
  $\mathbf{R} = (\mathbf{X}_A^T\mathbf{X}_A)^{-1}$
  $\mathbf{1} = \text{ones(length}(\mathcal{A}))$
  $B = (\mathbf{1}^T\mathbf{R}\mathbf{1})^{-1/2}$
  $\mathbf{v} = B\mathbf{R}\mathbf{1}$
  $\mathbf{u} = \mathbf{X}_A\mathbf{v}$
  $\mathbf{a} = \mathbf{X}\mathbf{u}$
  if $step = J$ **then**
    $\gamma \leftarrow C/B$
  else
    for $cont = 1$ to length($\mathcal{I}$) **do**
      $val1 \leftarrow \dfrac{C - c_{step-1}(j)}{B - a(\mathcal{I}(j))}$
      $val2 \leftarrow \dfrac{C + c_{step-1}(j)}{B + a(\mathcal{I}(j))}$
    end for
    $\gamma \leftarrow \min(val1, val2), val1 > 0, val2 > 0$
    $j \leftarrow \mathcal{I}(cont)$ which resulted γ
  end if
  $\tilde{\mathbf{y}} \leftarrow \tilde{\mathbf{y}} + \gamma\mathbf{u}$
end for
$\mathbf{w}_{LAR} \leftarrow (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\tilde{\mathbf{y}}$

As seen in Algorithm 1, the prediction vector is updated according to the following equation ($\tilde{\mathbf{y}}_0 = \mathbf{0}$)

$$\tilde{\mathbf{y}}_{step} = \tilde{\mathbf{y}}_{step-1} + \gamma_{step}\mathbf{u}_{step}, \tag{8}$$

where γ_{step} is the step size, $\mathbf{u}_{step}$ is the direction given by the coefficients in the active set, and $step = 0, 1, \cdots, J$ denotes the algorithm iteration. The cleverness of the LAR algorithm is how to calculate $\mathbf{u}$ and γ at each step, supported by the correlation vector.

As a consequence, the absolute value of the current correlation vector decreases as the number of coefficients in the active set increases. The more coefficients are used to predict the final value, the smaller the error. In the last step, we wish to have

$$\mathbf{c} = \mathbf{X}^T(\mathbf{y} - \tilde{\mathbf{y}}) = \mathbf{0} \tag{9}$$

leading to

$$\mathbf{X}^T\mathbf{y} = \mathbf{X}^T\tilde{\mathbf{y}} \tag{10}$$

which, using Eq. (3), yields to

$$\mathbf{X}^T\mathbf{y} = \mathbf{X}^T\mathbf{X}\mathbf{w}_{LAR}. \tag{11}$$

LAR's final solution, when all coefficients are calculated, corresponds to the Least Squares (LS) solution $\mathbf{w}_{LS} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$. (9) is known as the orthogonality principle.

The data normalization imposed by the LAR algorithm assumes that vectors containing each coefficient input signal at all time instants, i.e., $\mathbf{x}_j = [x_j(1) \cdots x_j(k) \cdots x_j(K)]^T$ with $j = 1, 2, \cdots, J$, have zero-mean and unitary length and that the reference signal vector, $\mathbf{y}$, as defined in Eq. (7), has zero-mean. Besides, as proposed in [17], we also impose that the reference signal vector, $\mathbf{y}$, has unitary length. This means, respectively,

$$\sum_{k=1}^{K} x_j(k) = 0 \implies m_{\mathbf{x}_j} = 0; \tag{12}$$

$$\sum_{k=1}^{K} x_j^2(k) = 1 \implies \|\mathbf{x}_j\|^2 = 1; \tag{13}$$

$$\sum_{k=1}^{K} y(k) = 0 \implies m_{\mathbf{y}} = 0; \text{ and} \tag{14}$$

$$\sum_{k=1}^{K} y^2(k) = 1 \implies \|\mathbf{y}\|^2 = 1. \tag{15}$$

Therefore, to conform to these conditions, the coefficient input signal vector should be transformed as

$$\bar{\mathbf{x}}_j = \frac{\mathbf{x}_j - m_{\mathbf{x}_j}}{||\mathbf{x}_j - m_{\mathbf{x}_j}||}, j = 1, \cdots, J \tag{16}$$

where $m_{\mathbf{x}_j} = \dfrac{\sum_{k=1}^{K} x_j(k)}{K}$ is the mean value of the elements of the coefficient input signal vector such that $\bar{\mathbf{X}} = [\bar{\mathbf{x}}_1 \;\; \bar{\mathbf{x}}_2 \cdots \;\; \bar{\mathbf{x}}_J]$. The reference signal vector should also be transformed as

$$\bar{\mathbf{y}} = \frac{\mathbf{y} - m_{\mathbf{y}}}{||\mathbf{y} - m_{\mathbf{y}}||}. \tag{17}$$

where $m_{\mathbf{y}} = \dfrac{\sum_{k=1}^{K} y(k)}{K}$ is the mean value of the elements of the reference signal vector.

Finally, the coefficient element is estimated by the LAR algorithm, from Eq. (9) and data normalization, as

$$w_{j,LAR} = ||\mathbf{y} - m_{\mathbf{y}}|| \frac{(\bar{\mathbf{X}}^T \bar{\mathbf{X}})^{-1} \bar{\mathbf{X}}^T \tilde{\mathbf{y}}}{||\mathbf{x}_j - m_{\mathbf{x}_j}||}, j = 1, \cdots, J \quad (18)$$

and, thus, defining the coefficient vector as

$$\mathbf{w}_{LAR} = [w_{1,LAR} \ \cdots \ w_{J,LAR}]^T. \quad (19)$$

IV. The LAR algorithm in a nonlinear system identification

In order to evaluate the performance of the Volterra filter in combination with the LAR algorithm in modeling a nonlinear system, the setup of a system identification, as depicted in Fig. 1, was employed.

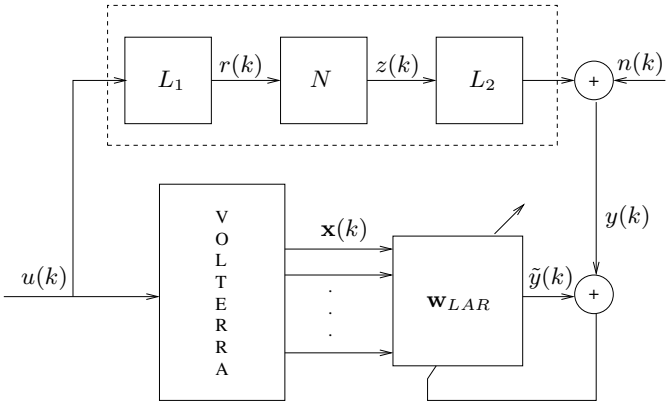


Fig. 1. Nonlinear system identification with Volterra filter and the LAR algorithm.

The LNL model used to represent the unknown nonlinear system has two linear filters with memory (L_1 and L_2) and one nonlinearity (N). Two different scenarios were simulated. In the first scenario, each linear filter had memory length equal to two and a nonlinearity of third-order. In the second scenario, each linear filter had memory length equal to three and a nonlinearity of fifth-order. As a consequence, the Volterra filter was composed by a nonlinearity of the third-order and memory length four in the first case, and by a nonlinearity of the fifth-order and memory length six in the second case. Therefore, the number of Volterra coefficients were 55 and 791, plus the DC component, for the first and second experiments, respectively. In Fig. 1, $n(k)$ is a random observation noise.

In order to compare the performance of the LAR algorithm in these experiments, we have computed the ordinary Least Squares (LS) and the Subset Selection (SSS) solutions. The SSS solution was obtained from the LS solution after forcing the smallest coefficients (in magnitude) to be equal to zero.

A. First Scenario: Third-Order Volterra + LAR

For the first scenario, a third-order Volterra filter was simulated. In that case, the first filter (L_1) has a coefficient vector set to

$$\mathbf{w}_1 = [0.5 \quad 1 \quad 0.5]^T \quad (20)$$

with an input signal vector defined as

$$\mathbf{u} = [u(k) \quad u(k-1) \quad u(k-2)]^T, \quad (21)$$

whereas the second filter (L_2) has a coefficient vector set to

$$\mathbf{w}_2 = [0.1 \quad -0.5 \quad 0.1]^T \quad (22)$$

with an input signal vector defined as

$$\mathbf{z} = [z(k) \quad z(k-1) \quad z(k-2)]^T. \quad (23)$$

The optimal coefficient vector of the Volterra filter has 55 coefficients, but 28 result equal to zero. As a consequence, the number of coefficients forced to be equal to zero in the SSS algorithm was 28 and the number of coefficients forced to be estimated in the LAR algorithm was 27.

1) Nonlinear System 1 (NL1): The LNL model to represent the first unknown linear system was constructed as

$$\begin{aligned} L_1: \quad r(k) &= \mathbf{w}_1^T \mathbf{u} \\ N: \quad z(k) &= 0.1r(k) - 0.01r^3(k) \\ L_2: \quad y(k) &= \mathbf{w}_2^T \mathbf{z} + n(k) \end{aligned} \quad (24)$$

where $\mathbf{w}_1$, $\mathbf{u}$, $\mathbf{w}_2$ and $\mathbf{z}$ are defined as in Eq. (20), (21), (22) and (23), respectively, and $r(k)$ is the nonlinearity input signal.

Fig. 2 shows the mean squared error (MSE) of the LAR algorithm as a function of the algorithm steps, averaged in time domain for $3,000$ samples. It is clear from this figure that the number of relevant coefficients is not larger than 27, as expected. Note that the observation error, set with variance $\sigma_r^2 = 10^{-6}$, is responsible for the -60dB of minimum MSE.

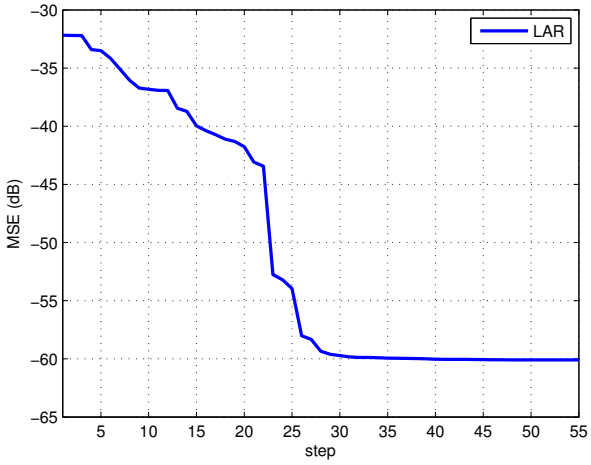


Fig. 2. 3rd order, NL1: MSE result (time domain with $K = 3,000$) for a Volterra filter using the LAR algorithm.

To evaluate the influence of the amount of data, the MSE of the LAR algorithm is shown in Fig. 3 as a function of the algorithm steps and the number of samples, averaged for a hundred runs for twenty sets of K, from $K = 100$ to $K = 5,000$. The dark line highlights when the MSE reaches the minimum. It can be seen that for over approximately $K = 1,800$, from $step = 27$, the MSE is already -60dB, i.e. the algorithm identifies as 27 the number of relevant coefficients,

the correct amount. For $K < 1,800$, the algorithm needs to estimate more coefficients to get $MSE = -60dB$, but no more than 40 coefficients (40 steps).

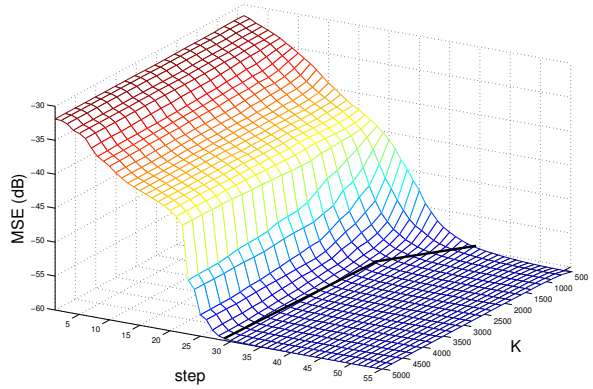


Fig. 3. 3rd order, NL1: MSE result (time domain) over twenty sets of K for a Volterra filter using the LAR algorithm.

The LAR algorithm was run again aiming the calculation of 27 coefficients with a hundred runs averaged to calculate the mean squared error for thirty different values of K, from $K = 100$ to $K = 3,000$, in steps of 100. For each set of data, the last signal vector was used to calculate the *a priori* error, not used to estimate the coefficients. The MSE result and the norm of the difference between the estimated and known coefficients, obtained from the LNL model, are shown in Fig. 4. From the first figure, it can be seen that the MSE converged for the LAR algorithm for $K \geq 1,800$, as in Fig. 3, while the LS and SSS algorithms need less input data to converge. From the second figure, it can be inferred that the estimated coefficient vector is accurate, since the distance between the coefficient vector estimated and known is very small, less than -100dB for all algorithms, as can be evidenced in Fig. 5 for the LAR algorithm.

Once, in this scenario, the LS algorithm reaches fast the minimum MSE, also does the SSS algorithm. Despite the fact that the SSS algorithm provides zero coefficients as does the LAR algorithm, the SSS solution needs the full 55 LS coefficients to make 28 of them equal to zero, while the LAR algorithm uses only the 27 most correlated coefficients.

2) Nonlinear System 2 (NL2): The LNL model to represent the second unknown linear system was constructed as

$$\begin{aligned} L_1: \quad r(k) &= \mathbf{w}_1^T \mathbf{u} \\ N: \quad z(k) &= r(k) - r^3(k) \\ L_2: \quad y(k) &= \mathbf{w}_2^T \mathbf{z} + n(k) \end{aligned} \tag{25}$$

The difference, when comparing NL1 to NL2, Eq. (24) to Eq. (25), is only in the final coefficient magnitude, not in their kernel position. The main objective here was to evaluate how much the coefficient magnitude influences the results.

Fig. 6 shows the mean squared error (MSE), averaged in time domain for $3,000$ samples, of the LAR algorithm as a function of the algorithm steps. This time, the number of

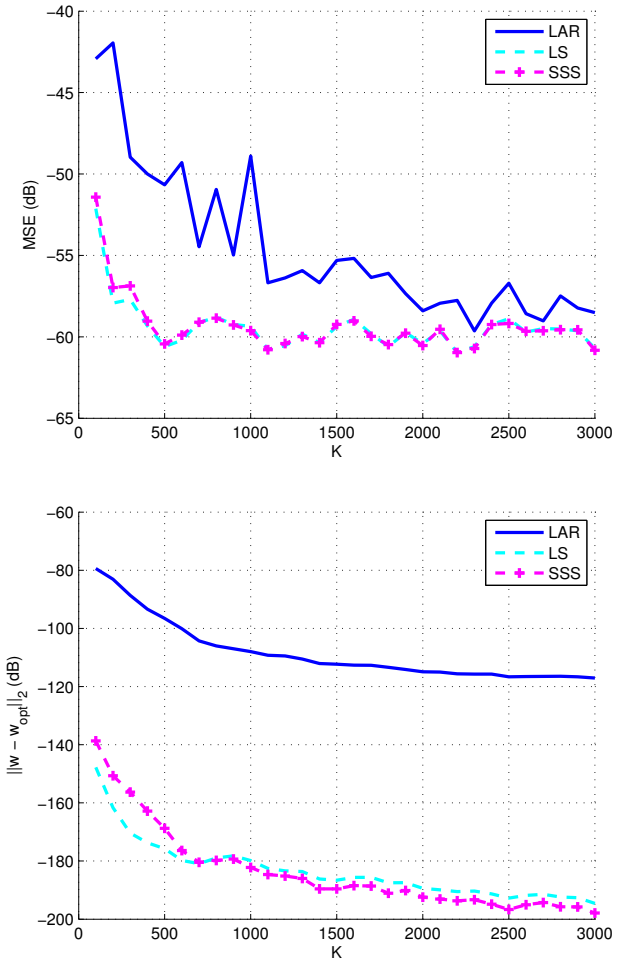


Fig. 4. 3rd order, NL1: MSE and the norm-2 of the difference between the estimated and optimal coefficient vector.

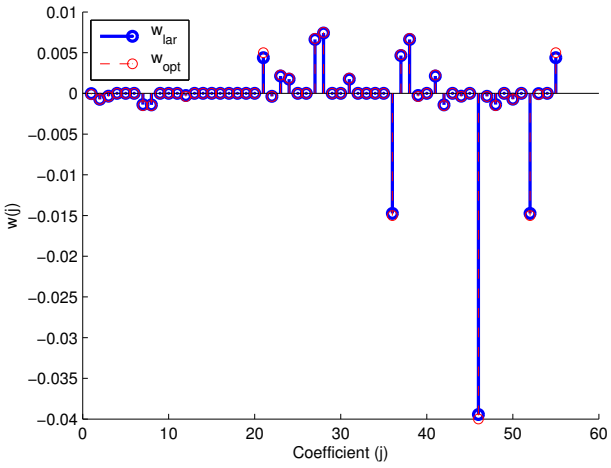


Fig. 5. 3rd order, NL1: Coefficients of the third-order Volterra filter: optimal, known from the LNL model, and LAR estimate.

relevant coefficients was not larger than 29 instead of 27, result obtained in NL1. This already suggests that the coefficient

magnitude does influence the number of estimated coefficients for a given amount of data (value of K). Still, the observation error was responsible for the -60dB of minimum MSE.

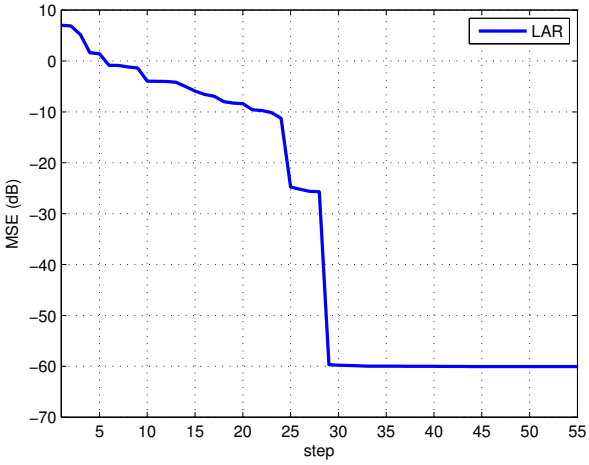


Fig. 6. 3rd order, NL2: MSE result (time domain with $K = 3,000$) for a third-order Volterra filter using the LAR algorithm.

Therefore, to evaluate the influence of the amount of data, the MSE of the LAR algorithm is shown in Fig. 7 as a function of the algorithm steps and the number of samples, averaged for a hundred runs for twenty sets of K, from $K = 100$ to $K = 5,000$. It can be seen that even with a high number of samples, it is necessary to have 30 steps, i.e., 30 coefficients estimated for the MSE converge to -60dB; for over approximately $K = 2,000$ this number is already reached, as it can be evidenced by the dark line. This time the LAR algorithm could not be precise to define that there are 27 nonzero coefficients, but still has close to 90% accuracy. This suggests that the minimum number of coefficients to be estimated is somehow influenced by the magnitudes of the nonlinear plant.

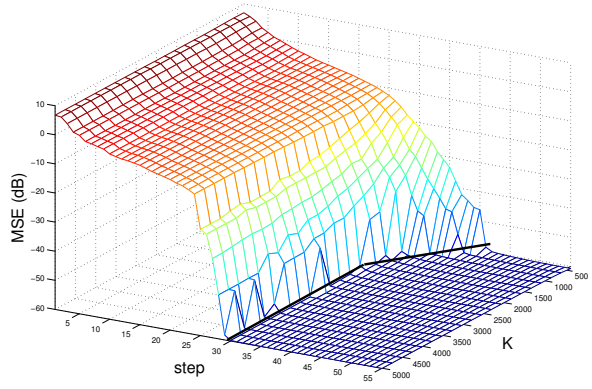


Fig. 7. 3rd order, NL2: MSE result over all sets of K for a Volterra filter using the LAR algorithm.

To observe the efficiency of the LAR algorithm estimating the correct number of coefficients known from the LNL model, the algorithm was then run aiming the calculation of 27 coefficients with a hundred runs averaged to calculate the mean squared error for thirty different values of K, from $K = 100$ to $K = 3,000$, in steps of 100. For each set of data, the last signal vector was used to calculate the *a priori* error, not used to estimate the coefficients.

The MSE result and the difference between the estimated and known coefficients, obtained from the LNL model, are shown in Fig. 8. The MSE result for the LAR algorithm was not as satisfactory as it was in the first nonlinear system simulated. This was, however, already expected, since the number of coefficients to be estimated, from Fig. 7, should be at least 30, instead of 27 previously known by the LNL model and the number used of coefficients to be estimated by the LAR algorithm. It can thus be concluded that the number of coefficients to be estimated has a high influence in the final results. Neither was the distance between the coefficient vector estimated as good as it was in NL1, still reaching, however, -60dB. Fig. 9 shows the coefficients (optimal and pbtained by the LAR algorithm) for this experiment.

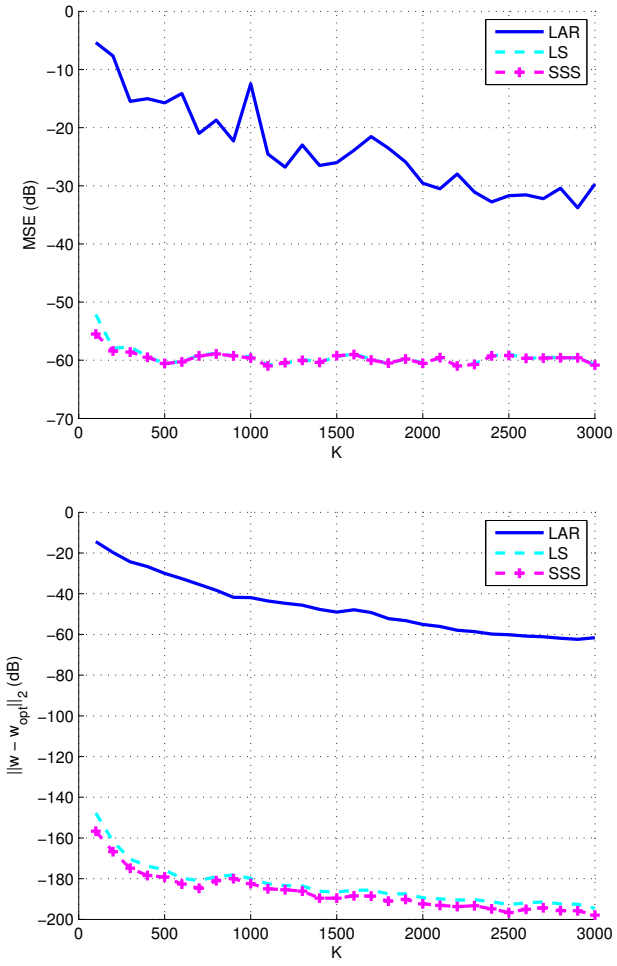


Fig. 8. 3rd order, NL2: MSE and the norm-2 of the difference between the estimated and optimal coefficient vector

3) First Scenario Conclusion: Based on the results so far, we can confirm that the LAR solution was close to the

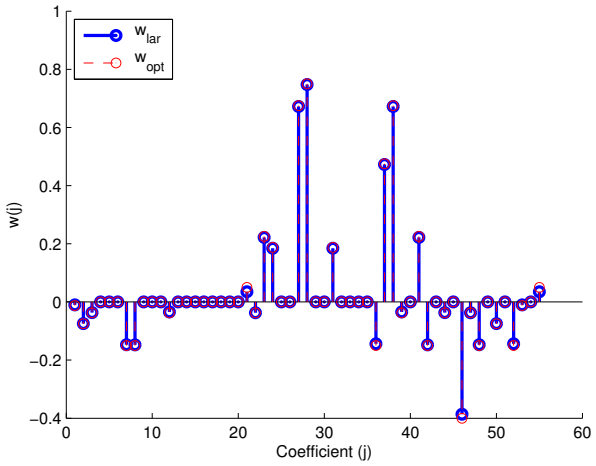


Fig. 9. 3rd order, NL2: Coefficients of the third-order Volterra filter: optimal, known from the LNL model, and LAR estimated.

optimal solution, although we could notice a depreciation of the results in the second nonlinear system simulated, when the coefficients magnitude increase. How the coefficients magnitudes offers the convergence of the LAR algorithm is subject of resarch. The number of coefficients, in our example previously known by the LNL model, is unknown in many real scenarios. Still, 30 coefficients was the maximum number of coefficients to be estimated, being 27 coefficients the correct number knwon by the LNL model. It should be noted that the coefficient Volterra kernel position is not important here; the LAR algorithm selects the most correlated coefficients in an independent way, occurring that in both nonlinear systems simulated.

Considering just the algorithms which have zero coefficients indeed, it could be concluded that the SSS solution presented a better result than the LAR algorithm. However, it was just possible because the number of zero coefficients was known from the LNL model. Using the LAR algorithm this value could be inferred. Even if the number of zero coefficients is not know, the LAR algorithm can be forced to stop when MSE has converged. Besides, the SSS algorithm needs to calculate all LS coefficients and then force to zero a given number. The LAR algorithm estimates the coefficients one by one and, since less coefficients are estimated, computational complexity can be reduced. Therefore, for sparse systems, the LAR algorithm could be a good choice.

B. Second Scenario: Fifth-Order Volterra + LAR

A second experiment was designed, where a fifth-order Volterra filter was employed. In that case the first filter (L_1) has its coefficient vector set to

$$\mathbf{w}_1 = [0.5 \quad 0.5 \quad 1 \quad 0.5]^T \tag{26}$$

with an input signal vector defined as

$$\mathbf{u} = [u(k) \quad u(k-1) \quad u(k-2) \quad u(k-3)]^T, \tag{27}$$

whereas the second filter (L_2) has a coefficient vector set to

$$\mathbf{w}_2 = [0.1 \quad -0.5 \quad 1 \quad -0.5]^T \tag{28}$$

with an input signal vector defined as

$$\mathbf{z} = [z(k) \quad z(k-1) \quad z(k-2) \quad z(k-3)]^T. \tag{29}$$

The optimal coefficient vector of the Volterra filter has 791 coefficients, but 580 result equal to zero. Therefore, the number of coefficients forced to be equal to zero in the SSS algorithm was 580 and the number of coefficients forced to be estimated in the LAR algorithm was 211.

1) Nonlinear System 1 (NL1): In this scenario, the only LNL model to represent the unknown linear system was constructed as

$$\begin{aligned} L_1: \quad r(k) &= \mathbf{w}_1^T \mathbf{u} \\ N: \quad z(k) &= 0.1r(k) - 0.01r^3(k) + 0.01r^5(k) \\ L_2: \quad y(k) &= \mathbf{w}_2^T \mathbf{z} + n(k) \end{aligned} \tag{30}$$

where $\mathbf{w}_1$, $\mathbf{u}$, $\mathbf{w}_2$ and $\mathbf{z}$ are defined as in Eq. (26), (27), (28) and (29), respectively, and $r(k)$ is the nonlinearity input signal.

Fig. 10 shows the mean squared error (MSE), averaged in time domain for $10,000$ samples, of the LAR algorithm as a function of the algorithm steps, where we can observe that the number of relevant coefficients identified was close to 350 – 400. Hence, the number of null coefficients identified by the LAR algorithm was not precise (only 211 should be different from zero, known from the LNL model), but still identifies close to 75% of the null coefficients. Once again, the observation error, set with variance $\sigma_r^2 = 10^{-6}$, was responsible for the -60dB of minimum MSE.

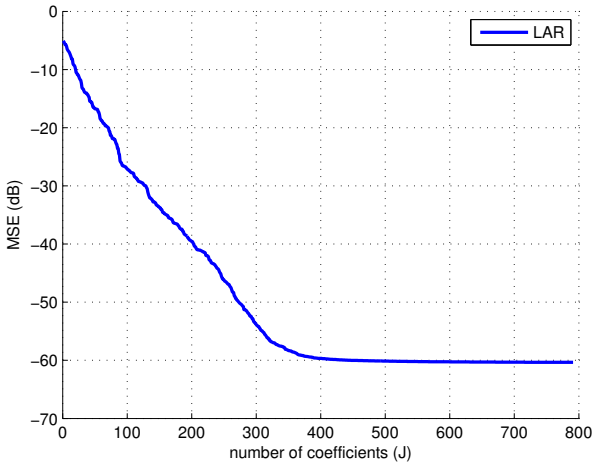


Fig. 10. 5th order, NL1: MSE result (time domain with $K = 10,000$) for a Volterra filter using the LAR algorithm.

To evaluate the influence of the amount of data, the MSE of the LAR algorithm is shown in Fig. 11 as a function of the algorithm steps and the number of samples, averaged for sixteen runs for eleven sets of K, from $K = 1,000$ to $K = 10,000$. The algorithm was not run for all steps, i.e., it was not implemented till the estimation of the 791 coefficient,

but just till the estimation of its 500 coefficient, since the number of non-zero coefficients known by the LNL model (211) was already reached by far. The dark line highlights when the MSE reaches the minimum. It can be seen that for over approximately $K = 6,000$, from $step = 350$, the MSE is -60dB. It can be seen that from this ammount, the rate of convergence becomes slower.

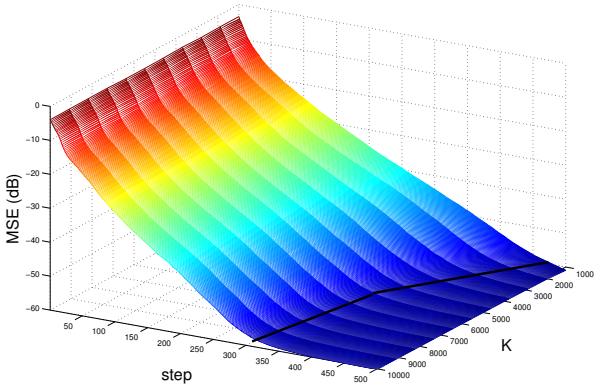


Fig. 11. 5th order, NL1: MSE result (time domain) over eleven sets of K for a Volterra filter using the LAR algorithm.

The LAR algorithm was run again aiming the calculation of 211 coefficients with fifty runs averaged to calculate the mean squared error for ten sets of K samples, from $K = 1,000$ to $K = 10,000$, in steps of $1,000$. For each set of data, the last signal vector was used to calculate the *a priori* error, not used to estimate the coefficients. The MSE result and the difference between the estimated and known coefficients, obtained from the LNL model, are shown in Fig. 12.

It can be seen that the MSE resulted from the LAR algorithm did not reach the minimum imposed by the noise variance (-60dB) for any value of K, but reached the minimum imposed by the number of coefficients calculated (close to -40dB when $J = 211$ and $K = 10,000$, from Fig. 10). The number of coefficients estimated must be increased to improve the results. It can be observed that the LS algorithm reaches the minimum MSE imposed by the noise variance with, approximately, $K = 3,000$. From the second figure, it can be inferred that the difference between the coefficients estimated and the optimum value, obtained by the LNL model, is close to -80dB for the LAR algorithm, still close to the optimal solution. As in the first experiment, the SSS algorithm had a good result, but it was just possible because the number of zero coefficients was known from the LNL model.

Finally, the coefficient are shown in Fig. 13, we see that the LAR solution is close to the optimal solution. Once again, the coefficient Volterra kernel position is not important.

2) Second Scenario Conclusion: Based on the results, it was evident that for higher order systems (over the fifth-order), the convergence rate can be very slow. However, if reaching the exact number of non-zero coefficients is not of prime importance, the LAR algorithm could be an option; for the simulated scenario, the minimum MSE can be obtained with

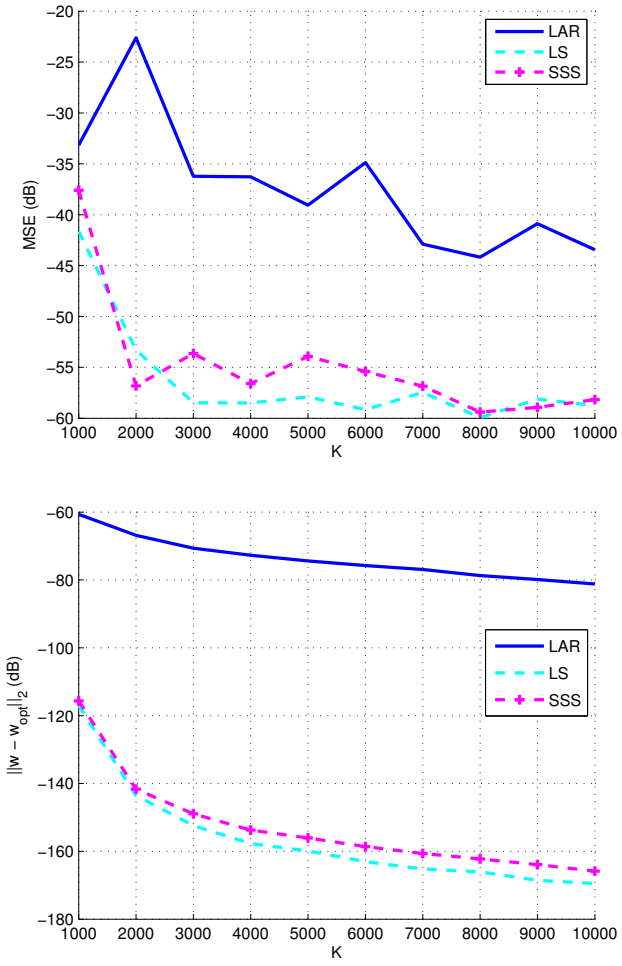


Fig. 12. 5th order, NL1: MSE and the norm-2 of the difference between the estimated and optimal coefficient vector

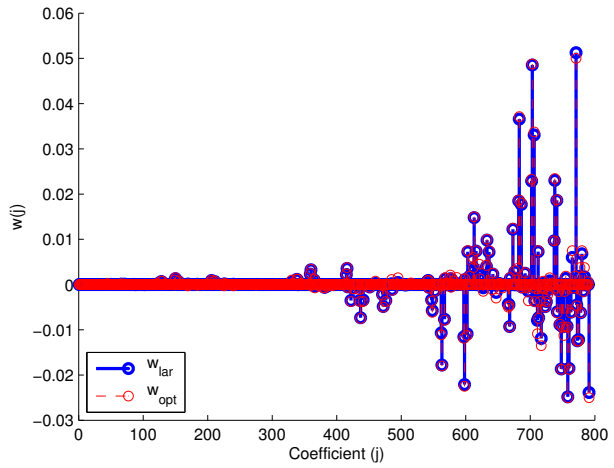


Fig. 13. 5th order, NL1: Coefficient of the fifth-order Volterra filter acting: optimal, known from the LNL model, and LAR estimated.

350 to 400 coefficients, reducing computational complexity when compared to LS-like algorithms.

V. Conclusion

From this work, it can be observed that, using the LAR algorithm with a Volterra filter, it is possible to identify the most relevant coefficients in nonlinear system modeling, independently of its Volterra kernel, allowing the use of filter with higher orders. First, the Volterra filter and the LAR algorithm were briefly reviewed and a detailed pseudo code was presented. The behavior of the LAR, the LS, and the SSS algorithms were then compared in two simulated scenarios, one using a third-order Volterra filter and another one using a fifth-order Volterra filter. From the simulation results, we were able to conclude that the LAR algorithm identifies the number of coefficients to be estimated without requiring any knowledge from the scenario; just with the input signal matrix and the reference signal vector it is possible to identify a non-linear system. Although the SSS algorithm has presented better results (lower MSE and difference between the estimated and the optimal coefficient vector) than the LAR algorithm, this was only possible due the previous knowledge of the correct order (the LNL model was known in advance); besides, the SSS needs to fulfill the LS coefficients to force some of them to zero. It was observed that the LAR algorithm has a convergence rate which depends on the coefficients magnitude and the system order; the convergence is slower with a large amount of input data but this draw back could be overcome by allowing a large number of coefficients when not enough data is available. Therefore, using the LAR algorithm, one can infer the number of coefficients to be estimated with the amount of data available, or the amount of data required to estimate an exact number of coefficients. A "stop criteria" as well a convergence analysis of the LAR algorithm are currently topics of our investigation.

Acknowledgements

The authors want to express their gratitude to CAPES, CNPq and INPI (Instituto Nacional da Propriedade Industrial) for partial financial support of this work.

References

[1] V. J. Mathew and G. L. Sicuranza, *Polynomial Signal Processing*. John Wiley and Sons, 2001.

[2] F. Kuech and W. Kellermann, "Proportionate NLMS algorithm for second-order Volterra filters and its application to nonlinear echo cancellation," in *Conf. Rec. Intl. Workshop on Acoustic Echo and Noise Control (IWAENC)*, Kyoto, Japan, Sep. 2003.

[3] A. Stenger, L. Trautmann, and R. Rabenstein, "Nonlinear acoustic echo cancellation with 2nd order adaptive Volterra filters," in *Acoustics, Speech, and Signal Processing, 1999. ICASSP '99. Proceedings., 1999 IEEE International Conference on*, vol. 2, Mar. 1999, pp. 877 – 880.

[4] S. Chan, T. Stathaki, and A. G. Constantinides, "Adaptive weighted Least Squares algorithm for Volterra signal modeling," *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, vol. 47, no. 4, pp. 545 – 554, Apr. 2000.

[5] Y. Kajikawa, "The adaptive Volterra filter: Its present and future," in *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, vol. 83, no. 12, 2000, pp. 51 – 61.

[6] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least Angle Regression," in *Annals of Statistics, Stanford University*, vol. 32, no. 2, 2004, pp. 407 – 409.

[7] L. Xi, D. Hang, and W. Mingwen, "Two-stage feature selection method for text classification," in *Multimedia Information Networking and Security, 2009. MINES '09. International Conference on*, vol. 1, Nov. 2009, pp. 234 – 238.

[8] S. S. Keerthi, "Generalized LARS as an effective feature selection tool for text classification with SVMs," in *ICML*, 2005, pp. 417 – 424.

[9] X. Li, "Finding deterministic solution from underdetermined equation: Large-scale performance modeling by least angle regression," in *Design Automation Conference, 2009. DAC '09. 46th ACM/IEEE*, Jul. 2009, pp. 364 – 369.

[10] C. Xiao, "Two-dimensional sparse principal component analysis for face recognition," in *Future Computer and Communication (ICFCC), 2010 2nd International Conference on*, vol. 2, May 2010, pp. 561 – 565.

[11] C. Chen, Y. Chang, K. Ricanek, and Y. Wang, "Face age estimation using model selection," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, Jun. 2010, pp. 93 – 99.

[12] C. Valdman, M. L. R. de Campos, and J. A. Apolinário Jr, "Nonlinear system identification with LAR," *Proceedings of the International Workshop on Telecomunications - IWT/2011*, pp. 195–200, May 2011. [Online]. Available: http://www.inatel.br/iwt/slide-show/accepted-papers

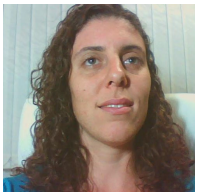
[13] V. Hegde, C. Radhakrishnan, D. Krusienski, and W. Jenkins, "Series-cascade nonlinear adaptive filters," in *Circuits and Systems, 2002. MWSCAS-2002. The 2002 45th Midwest Symposium on*, vol. 3, Aug. 2002, pp. III–219 – III–222.

[14] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementations*, 3rd ed. Springer, 2008.

[15] P. F. J. Doyle, R. K. Pearson, and B. A. Ogunnaike, *Identification and control using Volterra models*, 1st ed. Springer, 2002.

[16] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer, Feb. 2009.

[17] F. V. Berghen, "LARS Library: Least Angle Regression laSso/Stagewise Library," Université Libre de Bruxelles, Tech. Rep., 2005, accessed: 28 july 2010. [Online]. Available: http://www.applied-mathematics.net/identification/lars.html



Catia Valdman was born in Miguel Pereira, Rio de Janeiro, Brazil, in 1980. She received her B.S. degree in telecommunication engineering from Universidade Federal Fluminense (UFF), Rio de Janeiro, Brazil, in 2003 and the M.S. degree in electronic engineering from the Instituto Militar de Engenharia (IME), Rio de Janeiro, Brazil, in 2006.

Since 2006 she held a patent examiner position in the Brazilian patent office, Insituto Nacional de Propriedade Industrial – INPI, examining patents in the telecommunication field. Frequently, she composes the Brazilian delegation in meetings about the International Patent Classification – IPC – held in the World Intellectual Property Organization – WIPO, Geneva. In addition, she is currently a doctorate student in Universidade Federal do Rio de Janeiro – COPPE/UFRJ, where her research area is digital signal processing, interested specially in algorithms for non-linear systems.

Marcello L. R. de Campos was born in Niteroi, Brazil, in 1968. He received the Engineering degree (cum laude) from the Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil, in 1990, the M.Sc. degree from COPPE/UFRJ in 1991, and the Ph.D. degree from the University of Victoria, Victoria, BC, Canada, in 1995, all in electrical engineering. In 1996, he was post-doctoral fellow with the Department of Electronics, School of Engineering, UFRJ, and with the Program of Electrical Engineering, COPPE/UFRJ. From January 1997 until May 1998, he was Associate Professor with the Department of Electrical Engineering, Military Institute of Engineering, Rio de Janeiro.

He is currently Associate Professor of the Program of Electrical Engineering, COPPE/UFRJ, where he served as Department Vice-Chair and Chair in the years 2004 and 2005, respectively. From September to December 1998, he was visiting the Laboratory for Telecommunications Technology, Helsinki University of Technology, Espoo, Finland. He served as IEEE Communications Society Regional Director for Latin America in 2000 and 2001. In 2001, he received a Nokia Visiting Fellowship to visit the Centre for Wireless Communications, University of Oulu, Oulu, Finland. In 2008, he visited Unik, the University Graduate Center of the University of Oslo, Oslo, Norway. His research interests include adaptive signal processing, statistical signal processing, signal processing for communications, mobile and wireless communications, and MIMO systems.

José Antonio Apolinário Junior (SM‘04) was born in Taubaté, Brazil, in 1960. He graduated from the Military Academy of Agulhas Negras (AMAN), Resende, Brazil, in 1981 and received the B.Sc. degree from the Military Institute of Engineering (IME), Rio de Janeiro, Brazil, in 1988, the M.Sc. degree from the University of Brasília (UnB), Brasília, Brazil, in 1993, and the D.Sc. degree from the Federal University of Rio de Janeiro (COPPE/UFRJ), Rio de Janeiro, Brazil, in 1998, all in electrical engineering.

He is currently an Adjoint Professor with the Department of Electrical Engineering, IME, where he has already served as the Head of Department and as the Vice- Rector for Study and Research. He was a Visiting Professor at the Escuela Politécnica del Ejército (ESPE), Quito, Ecuador, from 1999 to 2000 and a Visiting Researcher and twice a Visiting Professor at Helsinki University of Technology (HUT), Finland, in 1997, 2004 and 2006, respectively. His research interests comprise many aspects of linear and nonlinear digital signal processing, including adaptive filtering, speech, and array processing. Dr. Apolinário has organized and been the first Chair of the Rio de Janeiro Chapter of the IEEE Communications Society. He has recently edited the book “QRDRLS Adaptive Filtering” (Springer, 2009) and served as the Finance Chair of IEEE ISCAS 2011 (Rio de Janeiro, August 2011).