Sparsity-Aware Estimation of Nonlinear Volterra Kernels

Vassilis Kekatos, Daniele Angelosante, Georgios B. Giannakis
University of Minnesota, ECE Dept.,
Minneapolis, MN 55455, USA
Emails:{kekatos,angel129,georgios}@umn.edu

***Abstract*—The Volterra series expansion has well-documented merits for modeling smooth nonlinear systems. Given that nature itself is parsimonious and models with minimal degrees of freedom are attractive from a system identification viewpoint, estimating *sparse* Volterra models is of paramount importance. Based on input-output data, existing estimators of Volterra kernels are sparsity agnostic because they rely on standard (possibly recursive) least-squares approaches. Instead, the present contribution develops batch and recursive algorithms for estimating sparse Volterra kernels using the least-absolute shrinkage and selection operator (Lasso) along with its recent weighted and online variants. Analysis and simulations demonstrate that weighted (recursive) Lasso has the potential to obviate the "curse of dimensionality," especially in the under-determined case where input-output data are less than the number of unknowns dictated by the order of the expansion and the memory of the kernels.**

I. Introduction

Nonlinear time-invariant systems with memory appear frequently in science and engineering. Widespread applications span the gamut of physiological and biological processes, power amplifiers, loudspeakers, speech, and image models to name a few; see e.g., [1]. If the nonlinearity is sufficiently smooth, the Volterra series offers a well-appreciated model of the output expressed as a polynomial expansion of the input using Taylor's theorem. The expansion coefficients of order $p > 1$ are p-dimensional kernel sequences generalizing the one-dimensional impulse response sequence encountered with linear systems. Their support is dictated by the memory.

The popularity of Volterra models stems from the fact that the input-output relationship is linear in the unknown kernel parameters. This allows their estimation based on input-output data via least-squares (LS). The major bottleneck is the "curse of dimensionality." Indeed, even when the expansion is truncated at a finite order P and the memory L is also finite, the number of unknowns is exponentially large, namely of order L^P. For reliable estimation, this necessitates long data records which may violate the time-invariance assumption.

On the other hand, in various applications multiple polynomial orders are missing, and/or, only a few kernel coefficients are nonzero because the memory is selectively nonzero. The nonlinearity order, the memory size, and which kernel coefficients are nonzero may all be unknown. Nonetheless, the Volterra expansion in such applications is sparse - a fact that can be attributed either to a parsimonious underlying physical system, or, to an overparameterized model assumed. Sparsity has not been so far exploited in the identification of Volterra systems and constitutes the motivation behind this work. It is well known that LS estimators do not account for sparsity; whereas the least-absolute shrinkage and selection operator (Lasso) [2] and the recent advances on compressive sampling offer a precious toolbox for estimating sparse signals. Existing sparsity-aware algorithms however, deal with linear systems.

The major contribution of the present work is the development of sparsity-aware kernel estimators of nonlinear Volterra systems. After expressing the estimation problem in matrix form, two batch estimators are developed based on the Lasso [2], and the weighted Lasso [3]. Their sequential counterparts are further introduced by solving a sequence of sparsity-promoting convex optimization problems. To reduce the computational burden and allow for recursive estimation, the coordinate descent approach is employed. By applying a single cycle of the coordinate descent scheme per time instant, two novel sparsity-aware recursive algorithms are also developed having computational complexity comparable to the recursive least-squares (RLS) algorithm. Simulated tests demonstrate that the novel batch and recursive estimators can cope with the curse of dimensionality present when identifying Volterra kernels, and yield parsimonious and accurate models with relatively short data records.

II. Preliminaries and Problem Formulation

Consider a nonlinear, discrete-time, time-invariant and causal system described by the following input-output (I/O) relationship

$$y(n) = f\left(x(n), \ldots, x(1)\right) \tag{1}$$

where $x(n)$ and $y(n)$ denote respectively the input and output samples at time n. While (1) can capture nonlinear dependencies of infinite memory, the finite-memory assumption adopted frequently in practice amounts to having $y(n) = f\left(x(n), \ldots, x(n-L+1)\right)$ with L finite. Under smoothness conditions, this I/O relationship can be approximated by a Volterra expansion often truncated to order P as [4], [1]

$$y(n) = \bar{h}_0 + \sum_{p=1}^{P} \bar{h}_p\left[x(n), \ldots, x(n-L+1)\right] + v(n) \tag{2}$$

This research was supported by the NSF grants CCF 0830480, CON 014658; and also by the Marie Curie Intl. Outgoing Fellowship "DIESIS"-No. 234914 within the 7th European Community Framework Programme.

where $v(n)$ captures unmodeled dynamics as well as observation noise, and $h_p(k_1,\ldots,k_p)$ denotes the p-th order Volterra kernel given by

$$\bar{h}_p\left[x(n),\ldots,x(n-L+1)\right] := \sum_{k_1=0}^{L-1}\cdots\sum_{k_p=0}^{L-1} h_p(k_1,\ldots,k_p) \times \prod_{i=1}^{p} x(n-k_i). \quad (3)$$

Given I/O samples $\{x(n),y(n)\}_{n=1}^{N}$, the goal is to estimate the Volterra kernels $h_p(k_1,\ldots,k_p)$ for $p=0,1,\ldots,P$, and $k_i=0,1,\ldots,L-1$, when the upper bounds P and L on the expansion order and the memory size are known. Although the task of truncated Volterra kernel estimation has been extensively studied [1], the sparsity present in the Volterra kernel representation of many nonlinear systems will be exploited hereafter to develop efficient kernel estimators.

To this end, the I/O relationship will be first expressed in a linear matrix-vector form. With the $L\times 1$ vector $\bar{\mathbf{x}}_1(n) := [x(n)\ \cdots\ x(n-L+1)]^T$ collecting the input samples affecting the output at time n, the input corresponding to the p-th order Volterra kernel is

$$\bar{\mathbf{x}}_p(n) := \underbrace{\bar{\mathbf{x}}_1(n)\otimes\ldots\otimes\bar{\mathbf{x}}_1(n)}_{p\ \text{times}},\ p=1,\ldots,P \quad (4)$$

where $\otimes$ denotes Kronecker product. The output of the p-th order Volterra kernel can be written as the inner product

$$\bar{h}_p\left[x(n),\ldots,x(n-L+1)\right] = \bar{\mathbf{x}}_p^T(n)\bar{\mathbf{h}}_p,\ p=1,\ldots,P \quad (5)$$

where vector $\bar{\mathbf{h}}_p$ contains the coefficients of the kernel $h_p(k_1,\ldots,k_p)$ arranged accordingly. Using (5), the model (2) can be rewritten as

$$\begin{aligned} y(n) &= \mathbf{x}^T(n)\mathbf{h}+v(n),\ n=1,\ldots,N, \text{ where} && (6)\\ \mathbf{x}(n) &:= \begin{bmatrix} 1 & \bar{\mathbf{x}}_1^T(n) & \ldots & \bar{\mathbf{x}}_P^T(n)\end{bmatrix}^T, \text{ and} && (7)\\ \mathbf{h} &:= \begin{bmatrix} h_0 & \bar{\mathbf{h}}_1^T & \ldots & \bar{\mathbf{h}}_P^T\end{bmatrix}^T. && (8)\end{aligned}$$

Upon considering the time instances $n=1,\ldots,N$, and defining $\mathbf{y} := [y(1)\ \cdots\ y(N)]^T$, the Volterra series expansion in (2) reduces to

$$\mathbf{y} = \mathbf{X}\mathbf{h}+\mathbf{v} \quad (9)$$

where $\mathbf{X} := [\mathbf{x}(1)\ \cdots\ \mathbf{x}(N)]^T$, and $\mathbf{v} := [v(1)\ \cdots\ v(N)]^T$.

Note that the number of coefficients in the p-th order kernel $h_p(k_1,\ldots,k_p)$ is L^p, that is exponential in the order of the system nonlinearity. However, all possible permutations of a fixed set of indices $\{k_1,\ldots,k_p\}$ multiply the same input term $x(n-k_1)\cdots x(n-k_p)$. To obtain a unique representation of (3), only one of these permutations must be retained. Thus, by properly discarding the redundant coefficients, the dimension of the vector $\bar{\mathbf{h}}_p$, and similarly for the input vectors $\bar{\mathbf{x}}_p(n)$, can be reduced to $\binom{L+p-1}{p}$. By exploiting the redundancy of all kernels, vectors $\mathbf{h}$ and $\mathbf{x}(n)$ can be shortened from $\frac{L^{P+1}-1}{L-1}$ to $M := \binom{L+P}{P}$ [1]. Taking this redundancy into account, matrix $\mathbf{X}$ in (9) will be hereafter considered to have dimension $N\times M$.

III. Estimation of Sparse Volterra Kernels

One of the nice properties of the Volterra representation is that the output $y(n)$ is a linear function of the kernel coefficients $h_p(k_1,\ldots,k_p)$. Thus, one can utilize the linear regression model (9) to develop standard estimators for $\mathbf{h}$ [1]. However, the number of kernel coefficients M is large for reasonable values of P and L. Thus, long observation intervals are needed for accurate estimation of $\mathbf{h}$.

In many applications on the other hand, it can be argued that the associated Volterra kernels are sparse, meaning that many of the entries of $\mathbf{h}$ are zero. Typical examples of sparse Volterra series expansions are outlined next.

Consider first the Linear - Nonlinear - Linear (LNL) model employed in various applications to model e.g., the effect of nonlinear amplifiers in OFDM, the satellite communication channel, or the transfer function of loudspeakers and headphones. The LNL model consists of a linear filter $\{h_a(k)\}_{k=0}^{L_a-1}$, in cascade with a memoryless nonlinearity $f(x)$, and a second linear filter $\{h_b(k)\}_{k=0}^{L_b-1}$. The overall memory is thus $L=L_a+L_b-1$. If the nonlinear function is analytic on an open set (a,b), it accepts a Taylor series expansion: $f(x)=\sum_{p=0}^{\infty}c_p x^p, \forall x\in(a,b)$. It can be shown that the p-th order Volterra kernel is then given by [1]

$$h_p(k_1,\ldots,k_p) = c_p\sum_{k=0}^{L_b-1} h_b(k)h_a(k_1-k)\ldots h_a(k_p-k). \quad (10)$$

In (10), there exist p-tuples $(k_1,\ldots,k_p)$ for which there is no $k\in\{0,\ldots,L_b-1\}$ such that $(k_i-k)\in\{0,L_a-1\}$ for all $i=1,\ldots,p$. For these p-tuples, the Volterra kernel equals zero. As an example, for filters of length $L_a=L_b=6$ and for $P=3$, among the 364 non-redundant kernel coefficients, the nonzero ones are no more than 224. Furthermore, when the LNL cascade is used to model, e.g., the satellite channel, the number of nonzero Volterra coefficients may be further reduced depending on the input; e.g., if $x(n)$ is drawn from a constant modulus constellation [5]. Hence, when the constellation is unknown (as in blind demodulation problems), extra sources of sparsity may be possible to exploit.

If the second filter in the LNL model is dropped, then the Wiener model is obtained, for which the p-th order Volterra kernel is expressed as

$$h_p(k_1,\ldots,k_p) = c_p h_a(k_1)\ldots h_a(k_p). \quad (11)$$

Due to the separability of the kernel in (11), if the impulse response $h_a(k)$ is also sparse, then the Volterra kernel becomes even sparser.

Apart from these nonlinear systems with special structure, it has been observed that in many applications only a few kernel coefficients contribute to the output [5]. Furthermore, the sparsity of the Volterra representation can arise when the degree of the nonlinearity and the system memory are not known a priori. In this case, kernel estimation must be performed jointly with model order selection. Based on these considerations, exploiting the sparsity present in many

Volterra representations is well motivated. Batch sparsity-aware estimators are described next.

A. Batch Estimators of Volterra Kernels

The system identification problem stated in Section II can be solved by using the LS approach as

$$\hat{\mathbf{h}}^{LS} = \arg\min_{\mathbf{h}} \|\mathbf{y} - \mathbf{X}\mathbf{h}\|_2^2. \tag{12}$$

If $N \geq M$ and matrix $\mathbf{X}$ has rank M, the solution of the problem is uniquely found as $\hat{\mathbf{h}}^{LS} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$. If the input samples $\{x(n)\}_{n=1}^N$ are drawn from a continuous distribution, then $\mathbf{X}^T\mathbf{X}$ is invertible with probability (w.p.) 1 [6]. However, the condition number of the matrix $\mathbf{X}^T\mathbf{X}$ grows with L and P [7]. A large condition number translates to numerically ill-posed inversion of the matrix and amplification of the noise.

If $N < M$, the solution of the convex optimization problem (12) is not unique but can be rendered unique if one chooses the minimum ℓ_2-norm solution. Alternatively, one may resort to the ridge regression (ℓ_2-norm regularized) solution given by

$$\hat{\mathbf{h}}^{Ridge} = \left(\mathbf{X}^T\mathbf{X} + \delta\mathbf{I}_M\right)^{-1}\mathbf{X}^T\mathbf{y} \tag{13}$$

for some $\delta > 0$. In any case, both $\hat{\mathbf{h}}^{LS}$ and $\hat{\mathbf{h}}^{Ridge}$ are not sparse.

To capitalize on the prior information about the Volterra kernel coefficients, sparsity can be effected by the ℓ_1-norm penalized regression [3]

$$\hat{\mathbf{h}} = \arg\min_{\mathbf{h}} \frac{1}{2}\|\mathbf{y} - \mathbf{X}\mathbf{h}\|_2^2 + \lambda_N \sum_{i=1}^{M} w_i |h_i| \tag{14}$$

where h_i is the i-th entry of $\mathbf{h}$, and $w_i > 0,\ i = 1, \ldots, M$. Two choices of w_i are commonly adopted:

(w1) $w_i = 1$ for $i = 1, \ldots, M$, which corresponds to the conventional Lasso estimator [2]; or,

(w2) $w_i = |\hat{h}_i^{Ridge}|^{-1}$ for $i = 1, \ldots, M$, which leads to the weighted Lasso estimator [3].

The weighted Lasso estimator exhibits improved asymptotic properties over the conventional one at the price of requiring the ridge regression estimates to evaluate the w_i's [3].

B. Recursive Estimators of Volterra Kernels

Contrary to the batch estimators, their recursive counterparts offer computational and memory savings, and enable tracking of time-varying systems. The RLS algorithm represents an efficient sequential implementation of the LS, as well as the ridge regression estimator. Indeed, it solves sequentially in time the following optimization problem:

$$\hat{\mathbf{h}}_N^{RLS} = \arg\min_{\mathbf{h}} J_N^{RLS}(\mathbf{h}) \tag{15}$$

$$J_N^{RLS}(\mathbf{h}) := \sum_{n=1}^{N} \beta^{N-n}\left(y(n) - \mathbf{x}^T(n)\mathbf{h}\right)^2 + \beta^N \delta \|\mathbf{h}\|_2^2$$

where β denotes the forgetting factor and δ a small positive constant. For time-invariant systems, β is set to 1, while $0 \ll \beta < 1$ enables tracking of slowly time-varying systems. Similar to the batch LS estimator, the RLS estimator does not exploit the prior knowledge on the sparsity of $\mathbf{h}$, and suffers from numerical instability especially when the effective memory of the algorithm, $1/(1-\beta)$, becomes comparable to the dimension M of the desired vector.

TABLE I
RECURSIVE CYCLIC COORDINATE DESCENT (RCCD) ALGORITHM

```
Initialize $\hat{\mathbf{h}}_0 = \mathbf{0}_M$ and $\mathbf{R}_0 = \epsilon\mathbf{I}_M$.
for $N = 1, 2, \ldots,$ do
    Update $\mathbf{r}_N$ and $\mathbf{R}_N$ using (17) and (18), respectively.
    for $i = 1, \ldots, M$ do
        Calculate $z_{N,i}$ using (20).
        Update $\hat{h}_{N,i}$ according to the thresholding rule (19).
    end for
end for
```

To overcome these limitations, the following estimation criterion is advocated; see also [8]

$$\hat{\mathbf{h}}_N = \arg\min_{\mathbf{h}} J_N^L(\mathbf{h}) \tag{16}$$

$$J_N^L(\mathbf{h}) := \sum_{n=1}^{N} \beta^{N-n}\left(y(n) - \mathbf{x}^T(n)\mathbf{h}\right)^2 + \lambda_N \sum_{i=1}^{M} w_{N,i}|h_i|$$

where, $w_{N,i}$ can be chosen as

(a1) $w_{N,i} = 1\ \forall N,\ i = 1, \ldots, M$, leading to the time-weighted Lasso (TWL); or,

(a2) $w_{N,i} = |\hat{h}_{N,i}^{RLS}|^{-1}\ \forall N,\ i = 1, \ldots, M$, which corresponds to the time penalty-weighted Lasso (TPWL).

The sequence $\{\hat{\mathbf{h}}_N\}$ cannot be updated recursively, and each one of the problems in (16) calls for a convex optimization solver. To avoid the computational burden involved, several methods have been developed [8], [9]. Coordinate descent-based recursive algorithms that approximately solve (16) are described next.

Solving (16) separately for each entry of $\mathbf{h}$ admits a simple closed-form solution. Using the recursive updates

$$\mathbf{r}_N = \beta\mathbf{r}_{N-1} + \mathbf{x}(N)y(N) \tag{17}$$

$$\mathbf{R}_N = \beta\mathbf{R}_{N-1} + \mathbf{x}(N)\mathbf{x}^T(N) \tag{18}$$

and a provisional solution at time $N-1$, namely $\hat{\mathbf{h}}_{N-1}$, the recursive update of the i-th component of $\hat{\mathbf{h}}_N$ is

$$\hat{h}_{N,i} = \frac{\operatorname{sgn}(z_{N,i})}{R_N(i,i)}\left[|z_{N,i}| - \lambda_N w_{N,i}\right]_+ \tag{19}$$

where $[x]_+ := \max(x, 0)$, $R_N(i,i)$ is the (i,i)-th entry of matrix $\mathbf{R}_N$, and $z_{N,i}$ is given by

$$z_{N,i} = r_{N,i} - \sum_{j=1}^{i-1} R_N(i,j)\hat{h}_{N,j} - \sum_{j=i+1}^{M} R_N(i,j)\hat{h}_{N-1,j}. \tag{20}$$

The developed algorithm, called hereafter recursive cyclic coordinate descent (RCCD), is summarized in Table I. Its complexity is dominated by the $\mathcal{O}(M^2)$ computations needed for updating the matrix $\mathbf{R}_N$, which is of the same order as the RLS. If $w_{N,i}$ in (19) is set to $1\ \forall N$ and $i = 1, \ldots, M$,

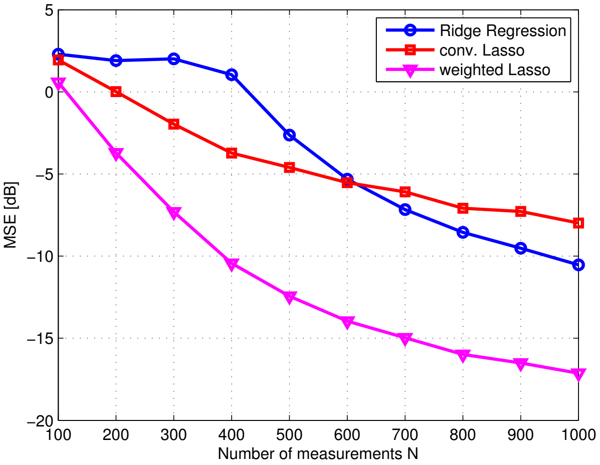

Fig. 1. MSE of batch estimators versus the number observations.

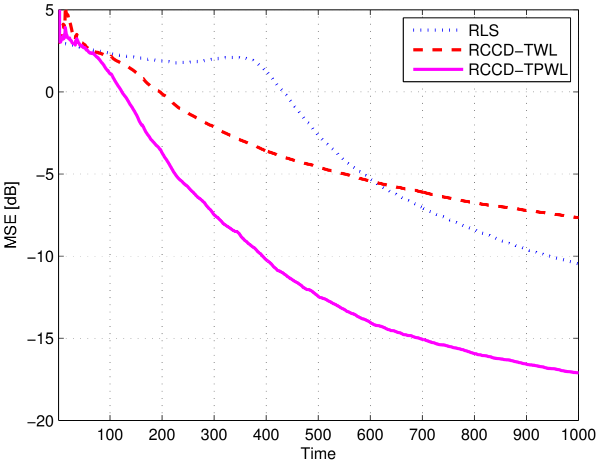

Fig. 2. MSE of recursive estimators.

the RCCD algorithm approximately solves the TWL problem, and is thus abbreviated as RCCD-TWL. Similarly, if $w_{N,i} = |\hat{h}_{N,i}^{RLS}|^{-1}$ $\forall N$ and $i = 1, \ldots, M$, the RCCD approximates the solution of TPWL and is referred to as RCCD-TPWL. It is worth stressing that the RLS, RCCD-TWL, and RCCD-TPWL algorithms are implemented in $\mathcal{O}(M^2)$.

IV. Simulated Tests

The developed estimators were tested through computer simulations. The system under study was an LNL one, consisting of a linear filter with impulse response $\mathbf{h}_f = [0.36\ 0\ 0.91\ 0\ 0\ 0.19]^T$, in cascade with the memoryless nonlinearity $f(x) = -0.5x^3+0.4x^2+x$, and the same linear filter. This system is exactly described by a Volterra expansion with $L = 11$ and $P = 3$, leading to a total of $M = \binom{L+P}{P} = 364$ kernel coefficients stored in the vector $\mathbf{h}_0$. Out of the 364 coefficients only 48 are nonzero. The system input was modeled as $x(n) \sim \mathcal{N}(0, 1)$, while the output was corrupted by additive noise $v(n) \sim \mathcal{N}(0, 0.1)$. First, the batch estimators of Section III-A were tested, followed by their sequential counterparts.

In Fig. 1, the obtained mean-square error (MSE), $\mathbb{E}\left[\|\mathbf{h}_0 - \hat{\mathbf{h}}\|_2^2\right]$, averaged over 100 Monte Carlo experiments, is plotted against the number of observations, N, for the following estimators: (i) the ridge regression estimator of (13) with δ=1; (ii) the Lasso estimator with λ_N=$0.7\sqrt{N}$; and, (iii) the weighted Lasso estimator with λ_N=$0.08 \log N$. It can be seen that the sparsity-agnostic ridge regression estimator is outperformed by the Lasso estimator for short observation intervals (N<600). For larger N, where $\mathbf{X}^T\mathbf{X}$ becomes well-conditioned, the former provides improved estimation accuracy. However, the weighted Lasso estimator offers the lowest MSE for every N, and provides a reasonable accuracy even for the underdetermined case (N<364).

Performance of the sequential estimator of Section III-B was assessed in the same setup. Fig. 2 illustrates the convergence of the MSE, averaged over 100 Monte Carlo runs, for the following three recursive algorithms: (i) the conventional RLS of (15); (ii) the RCCD-TWL; and, (iii) the RCCD-TPWL. Since the system was time-invariant, the forgetting factor was set to $\beta = 1$. It can be observed that the conclusions drawn for the batch case carry over to the recursive algorithms too. Moreover, the sparsity-aware iterates of Table I are a close approximation to the problem in (16).

V. Conclusions

The idea of exploiting sparsity in the representation of a system, already widely adopted for linear regression and linear system identification, has been permeated here to estimate sparse kernels of nonlinear Volterra models. The resultant weighted Lasso batch estimator outperforms the LS-based kernel estimator, and provides reasonable estimation accuracy even for a limited number of input-output observations. Furthermore, the novel RCCD-TPWL algorithm, derived as an approximate solution of the time penalty weighted Lasso cost function, is shown to converge fast to the exact solution. Thus, it offers an accurate and sparse solution, recursively updated at complexity comparable to the conventional RLS.

References

[1] V. Mathews and G. Sicuranza, *Polynomial Signal Processing*. John Wiley & Sons Inc., 2000.

[2] R. Tibshirani, “Regression shrinkage and selection via the Lasso,” *J. R. Stat. Soc. Ser.*, vol. 58, no. 1, pp. 267–288, 1996.

[3] H. Zou, “The adaptive Lasso and its oracle properties,” *J. of the American Stat. Assoc.*, vol. 101, no. 476, pp. 1418–1429, Dec. 2006.

[4] G. Palm and T. Poggio, “The Volterra representation and the Wiener expansion: Validity and pitfalls,” *SIAM Journal on Applied Math.*, vol. 33, no. 2, pp. 195–216, Sept. 1977.

[5] S. Benedetto and E. Biglieri, “Nonlinear equalization of digital satellite channels,” *IEEE J. Select. Areas Commun.*, no. 1, pp. 57–62, Jan. 1983.

[6] R. Nowak and B. V. Veen, “Invertibility of higher order moment matrices,” *IEEE Trans. Signal Processing*, vol. 43, no. 3, pp. 705–708, Mar. 1995.

[7] ——, “Random and pseudorandom inputs for Volterra filter identification,” *IEEE Trans. Signal Processing*, vol. 42, no. 8, pp. 2124–2135, Aug. 1994.

[8] D. Angelosante and G. Giannakis, “RLS-weighted Lasso for adaptive estimation of sparse signals,” in *Proc. of the IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Taipei, Taiwan, Apr. 2009.

[9] D. Angelosante, J.-A. Bazerque, and G. B. Giannakis, “Online coordinate descent for adaptive estimation of sparse signals,” in *Workshop on Statistical Signal Processing*, Cardiff, Wales, UK, Aug. 2009.