DIGITAL ACCESS TO
SCHOLARSHIP AT HARVARD

Adaptive Greedy Algorithm With Application to Nonlinear Communications

(Article begins on next page)

An adaptive greedy algorithm with application to nonlinear communications

Gerasimos Mileounis, *Student Member, IEEE,* Behtash Babadi,
Nicholas Kalouptsidis, and Vahid Tarokh, *Fellow, IEEE,*

Abstract—Greedy algorithms form an essential tool for compressed sensing. However, their inherent batch mode discourages their use in time-varying environments due to significant complexity and storage requirements. In this paper a powerful greedy scheme developed in [1], [2], is converted into an adaptive algorithm which is applied to estimation of a class of nonlinear communication systems. Performance is assessed via computer simulations on a variety of linear and nonlinear channels; all confirm significant improvements over conventional methods.

Index Terms—Adaptive filters, ARMA processes, Nonlinear systems, Equalizers, Compressed Sensing.

I. INTRODUCTION

Many real-life systems admit sparse representations, that is they are characterized by small number of non-zero coefficients. Sparse systems can be found in many signal processing [3] and communications applications [4]–[6]. For instance, in High Definition Television the significant echoes form a cluster, yet interarrival times between different clusters can be very long [4]. In wireless multipath channels there is a relatively small number of clusters of significant paths [5]. Finally, underwater acoustic channels exhibit long time delays between the multipath terms due to reflections off the sea surface or sea floor [6].

Two major algorithmic approaches to compressive sensing are ℓ_1-minimization (basis pursuit) and greedy algorithms (matching pursuit). Basis pursuit methods solve a convex minimization problem, which replaces the ℓ_0 quasi-norm by the ℓ_1 norm. The convex minimization problem can be solved using linear programming methods, and is thus executed in polynomial time [7]. Greedy algorithms, on the other hand, iteratively compute the support set of the signal and construct an approximation to it, until a halting condition is met [1], [2], [8]–[11]. Both of the above approaches pose their own advantages and disadvantages. ℓ_1-minimization methods provide theoretical performance guarantees, but they lack the speed of greedy techniques. Recently developed greedy algorithms, such as those developed in [1], [2], [10], deliver some of the same guarantee as ℓ_1-minimization approaches with less computational cost and storage.

G. Mileounis and N. Kalouptsidis are with the Department of Informatics and Telecommunications, Division of Communications and Signal Processing, University of Athens, Greece (email:{gmil,kalou}@di.uoa.gr).
B. Babadi and V. Tarokh are with the School of Engineering and Applied Sciences, Harvard University, Cambridge, MA02138 USA (e-mail: {behtash,vahid}@seas. harvard.edu).

Many signal processing applications [4]–[6] require adaptive estimation with minimal complexity and small memory requirements. Existing approaches to sparse adaptive estimation use the ℓ_1-minimization technique, in order to improve the performance of conventional algorithms. Chen et al. [12] incorporated two different sparsity constraints (the ℓ_1 and the log-sum penalty functions) into the quadratic cost of the standard Least Mean Squares (LMS) to improve the filtering performance on sparse systems. In [13], Angelosante et al. developed a recursive subgradient-based approach for solving the batch Lasso estimator. An ℓ_1-regularized RLS type algorithm based on a low complexity Expectation-Maximization is derived in [14] by Babadi et al. Sparse adaptive ℓ_1-regularized algorithms based on Kalman filtering and Expectation Maximization are reported in [15] by Kalouptsidis et al.

In contrast to the above work on adaptive sparse identification, this paper focuses on the greedy viewpoint. Greedy algorithms in their ordinary mode of operation, have an inherent batch mode, and hence are not suitable for time-varying environments. This paper establishes a conversion procedure that turns greedy algorithms into adaptive schemes for sparse system identification. In particular, a Sparse Adaptive Orthogonal Matching Pursuit (SpAdOMP) algorithm of linear complexity is developed, based on existing greedy algorithms [1], [2], that provide optimal performance guarantees. Also, the steady-state Mean Square Error (MSE) of the SpAdOMP algorithm is studied analytically. The developed algorithm is used to estimate ARMA and Nonlinear ARMA channels. It is shown that channel inversion for these channels, maintains sparsity and that it is equivalent to channel estimation. Computer simulations reveal that the proposed algorithm outperforms most existing adaptive algorithms for sparse channel estimation.

The paper is structured as follows. The problem formulation and literature review are addressed in section II. Section III describes the established algorithm, the steady-state error analysis and applications to nonlinear communication channels. Computer simulations are presented in section IV. Conclusions and future work are discussed in section V.

II. GREEDY METHODS AND THE COSAMP ALGORITHM

Consider the noisy representation of a vector $\boldsymbol{y}(n) = [y_1, \cdots, y_n]^T$ in terms of a basis arranged in the columns of a matrix $\boldsymbol{\Phi}(n)$ at time n

$$\boldsymbol{y}(n) = \boldsymbol{\Phi}(n)\boldsymbol{c} + \boldsymbol{\eta}(n) \tag{1}$$

where c is the parameter vector, $\boldsymbol{\Phi}(n) = [\boldsymbol{\phi}(1), \ldots, \boldsymbol{\phi}(n)]^T$ and $\boldsymbol{\eta}(n) = [\eta_1, \ldots, \eta_n]^T$ is the additive noise. The measurement matrix $\boldsymbol{\Phi}(n) \in \mathbb{C}^{n \times N}$ is often referred to as *dictionary* and the parameter vector c is assumed to be sparse, *i.e.*, $\|c\|_{\ell_0} \ll N$, where $\|\cdot\|_{\ell_0} = |\mathrm{supp}(\cdot)|$ is the ℓ_0 quasi-norm. We will call the parameter vector s-sparse when it contains at most s non-zero entries.

Recovery of the unknown parameter vector c can be pursued by finding the sparsest estimate of c which satisfies the ℓ_2 norm error tolerance δ

$$\min_{c}\|c\|_{\ell_0} \quad \text{subject to} \quad \|\boldsymbol{y}(n) - \boldsymbol{\Phi}(n)c\|_{\ell_2} \le \delta. \quad (P_{\ell_0})$$

Convex relaxation methods cope with the intractability of the above formulation by approximating the ℓ_0 quasi-norm by the convex ℓ_1 norm. The set of resulting techniques is referred to as ℓ_1-minimization. The ℓ_2 constraint can be interpreted as a noise removal mechanism when $\delta \ge \|\boldsymbol{\eta}(n)\|_{\ell_2}$. The ℓ_1-minimization approach is a convex optimization problem and can be solved by linear programming methods [7], [16], projected gradient methods [17] and iterative thresholding [18].

The exact conditions for retrieving the sparse vector rely either on the coherence of the measurement matrix [19] or on the Restricted Isometry Property (RIP) [16]. A measurement matrix $\boldsymbol{\Phi}(n)$ satisfies the Restricted Isometry Property for $\delta_s(n) \in (0,1)$ if we have

$$\left(1 - \delta_s(n)\right)\|c\|_{\ell_2}^2 \le \|\boldsymbol{\Phi}(n)c\|_{\ell_2}^2 \le \left(1 + \delta_s(n)\right)\|c\|_{\ell_2}^2 \quad (2)$$

for all s-sparse c. When $\delta_s(n)$ is small, the restricted isometry property implies that the set of columns of $\boldsymbol{\Phi}(n)$ approximately form an orthonormal system.

A. The CoSaMP greedy algorithm

Greedy algorithms provide an alternative approach to ℓ_1-minimization. For the recovery of a sparse signal in the presence of noise, greedy algorithms iteratively improve the current estimate for the parameter vector c by modifying one or more parameters until a halting condition is met. The basic principle behind greedy algorithms is to iteratively find the support set of the sparse vector and reconstruct the signal using the restricted support Least Squares (LS) estimate. The computational complexity of these algorithms depends on the number of iterations required to find the correct support set. One of the earliest algorithms proposed for sparse signal recovery is the Orthogonal Matching Pursuit (OMP) [8], [9], [19]. At each iteration, OMP finds the column of $\boldsymbol{\Phi}(n)$ most correlated with the residual, $\boldsymbol{v}(n) = \boldsymbol{y}(n) - \boldsymbol{\Phi}(n)\hat{c}$, using the proxy signal $\boldsymbol{p}(n) = \boldsymbol{\Phi}^{*T}(n)\boldsymbol{v}(n)$ (where $\boldsymbol{A}^{*T}(n)$ denotes the conjugate transpose of the matrix $\boldsymbol{A}(n) \in \mathbb{C}^{n \times N}$), and adds it to the support set. Then, it solves the following least squares problem:

$$\hat{c} = \arg\min_{z}\|\boldsymbol{y}(n) - \boldsymbol{\Phi}(n)z\|_{\ell_2}$$

and finally updates the residual by removing the contribution of the latter column. By repeating this procedure a total of s times, the support set of c is recovered. Although OMP is quite fast, it is unknown whether it succeeds on noisy measurements.

An alternative algorithm, called Stagewise OMP (StOMP), was proposed in [11]. Unlike OMP, it selects all components of the proxy signal whose values are above a certain threshold. Due to the multiple selection step, StOMP achieves better runtime than OMP. Parameter tuning in StOMP might be difficult and there are rigorous asymptotic results available.

A more sophisticated algorithm has been recently developed by Needell and Vershynin, and it is known as Regularized OMP (ROMP) [10]. ROMP chooses the s largest components of the proxy signal, followed by a regularization step, to ensure that not too many incorrect components are selected. For a measurement matrix $\boldsymbol{\Phi}(n)$ with RIP constant $\delta_{2s} = 0.03/\sqrt{\log s}$, ROMP provides uniform and stable recovery results. The recovery bounds obtained in [10] are optimal up to a logarithmic factor. Tighter recovery bounds which avoid the presence of the logarithmic factor are obtained by Needell and Tropp via the Compressed Sampling Matching Pursuit algorithm (CoSaMP) [1]. CoSaMP provides tighter recovery bounds than ROMP optimal up to a constant factor (which is a function of the RIP constants). An algorithm similar to the CoSaMP, was presented by Dai and Milenkovic and is known as Subspace Pursuit (SP) [2].

As with most greedy algorithms, CoSaMP takes advantage of the measurement matrix $\boldsymbol{\Phi}(n)$ which is approximately orthonormal ($\boldsymbol{\Phi}^{*T}(n)\boldsymbol{\Phi}(n)$ is close to the identity). Hence, the largest components of the signal proxy $\boldsymbol{p}(n) = \boldsymbol{\Phi}^{*T}(n)\boldsymbol{\Phi}(n)c$ is most likely to correspond to the non-zero entries of c. Next, the algorithm adds the largest components of the signal proxy to the running support set and performs least squares to get an estimate for the signal. Finally, it prunes the least square estimation and updates the error residual. The main ingredients of the CoSaMP algorithm are outlined below:

1) *Identification* of the largest $2s$ components of the proxy signal
2) *Support Merger*, which forms the union of the set of newly identified components with the set of indices corresponding to the s largest components of the least square estimate obtained in the previous iteration
3) *Estimation* via least squares on the merged set of components
4) *Pruning*, which restricts the LS estimate to its s largest components
5) *Sample update*, which updates the error residual.

The above steps are repeated until a halting criterion is met. The main difference between CoSaMP and SP is in the identification step where the SP algorithm chooses the s largest components.

In a time-varying environment, the estimates must be updated adaptively to take into consideration system variations. In such cases, the use of existing greedy algorithms on a measurement block requires that the system remain constant within that block. Moreover, the cost of repetitively applying a greedy algorithm after a new block arrives becomes enormous. Adaptive algorithms, on the other hand, allow online operation. Therefore, our primary goal is to convert existing greedy algorithms into an adaptive mode, while maintaining their superior performance gains. We demonstrate below that the conversion is feasible with linear complexity. We focus our

TABLE I
SpAdOMP ALGORITHM

Algorithm description		Complexity
$\boldsymbol{c}(0) = 0, \boldsymbol{w}(0) = 0, \boldsymbol{p}(0) = 0$	{Initiliazation}	
$v(0) = y(0)$	{Initial residual}	
$0 < \lambda \leq 1$	{Forgetting factor}	
$0 < \mu < 2\lambda_{\max}^{-1}$	{Step size}	
For $n := 1, 2, \ldots$ **do**		
1: $\quad \boldsymbol{p}(n) = \lambda\boldsymbol{p}(n-1) + \boldsymbol{\phi}^*(n-1)v(n-1)$	{Form signal proxy}	N
2: $\quad \Omega = \mathrm{supp}(\boldsymbol{p}_{2s}(n))$	{Identify large components}	N
3: $\quad \Lambda = \Omega \cup \mathrm{supp}(\boldsymbol{c}(n-1))$	{Merge supports}	s
4: $\quad \varepsilon(n) = y(n) - \boldsymbol{\phi}_{|\Lambda}^T(n)\boldsymbol{w}_{|\Lambda}(n-1)$	{Prediction error}	s
5: $\quad \boldsymbol{w}_{|\Lambda}(n) = \boldsymbol{w}_{|\Lambda}(n-1) + \mu\boldsymbol{\phi}_{|\Lambda}^*(n)\varepsilon(n)$	{LMS iteration}	s
6: $\quad \Lambda_s = \max(|\boldsymbol{w}_{|\Lambda}(n)|, s)$	{Obtain the pruned support}	s
7: $\quad \boldsymbol{c}_{|\Lambda_s}(n) = \boldsymbol{w}_{|\Lambda_s}(n), \quad \boldsymbol{c}_{|\Lambda_s^c}(n) = \boldsymbol{0}$	{Prune the LMS estimates}	
8: $\quad v(n) = y(n) - \boldsymbol{\phi}^T(n)\boldsymbol{c}(n)$	{Update error residual}	s
end For		$\mathcal{O}(N)$

analysis on CoSaMP/SP due to their superior performance, but similar ideas are applicable to other greedy algorithms as well.

III. SPARSE ADAPTIVE ORTHOGONAL MATCHING PURSUIT ALGORITHM

This section starts by converting CoSaMP and SP algorithms [1], [2] into an adaptive scheme. The derived algorithm is then used to estimate sparse Nonlinear ARMA channels.

The proposed algorithm relies on three modifications to the CoSaMP/SP structure: the proxy identification, estimation and error residual update. The error residual is now evaluated by

$$v(n) = y(n) - \boldsymbol{\phi}^T(n)\boldsymbol{c}(n). \tag{3}$$

The above formula involves the current sample only, in contrast to the CoSaMP/SP scheme which requires all the previous samples. Eq. (3) requires s complex multiplications, whereas the cost of the sample update in the CoSaMP/SP is sn multiplications. A new proxy signal that is more suitable for the adaptive mode, can be defined as:

$$\boldsymbol{p}(n) = \sum_{i=1}^{n-1} \lambda^{n-1-i}\boldsymbol{\phi}^*(i)v(i)$$

and is updated by

$$\boldsymbol{p}(n) = \lambda\boldsymbol{p}(n-1) + \boldsymbol{\phi}^*(n-1)v(n-1)$$

where the forgetting factor $\lambda \in (0, 1]$ is incorporated in order to give less weight in the past and more weight to recent data. This way the derived algorithm is capable of capturing variations on the support set of the parameter vector $\boldsymbol{c}$. In the case of a time-invariant environment, λ should be set to 1. The addition of the forgetting factor mechanism requires redefining the Restricted Isometry Property as follows:

Definition 1. *A measurement matrix* $\boldsymbol{\Phi}(n)$ *satisfies the Exponentially-weighted Restricted Isometry Property (ERIP) for* $\lambda \in (0, 1]$ *and* $\delta_s(\lambda, n) \in (0, 1)$*, if we have*

$$\left(1 - \delta_s(\lambda, n)\right)\|\boldsymbol{c}\|_{\ell_2}^2 \leq \|\boldsymbol{D}^{1/2}(n)\boldsymbol{\Phi}(n)\boldsymbol{c}\|_{\ell_2}^2 \leq \left(1 + \delta_s(\lambda, n)\right)\|\boldsymbol{c}\|_{\ell_2}^2 \tag{4}$$

where $\boldsymbol{D}(n) := diag(1, \lambda, \cdots, \lambda^{n-1})$.

The last modification attacks the estimation step. The estimate $\boldsymbol{w}(n)$ is updated by standard adaptive algorithms such as the LMS and RLS [20]. LMS is one of the most widely used algorithm in adaptive filtering due to its simplicity, robustness and low complexity. On the other hand, the RLS algorithm is an order of magnitude costlier but significantly improves the convergence speed of LMS. The LMS algorithm replaces the exact signal statistics by approximations, whereas RLS updates the inverse covariance matrix. The update rule for RLS cannot be directly restricted to the index support set Λ. Hence, a more sophisticated mechanism is required like the one proposed in [14]. For reasons of simplicity and complexity we focus on the LMS algorithm. At each iteration the current regressor $\boldsymbol{\phi}(n)$ and the previous estimate $\boldsymbol{w}(n-1)$ are restricted to the instantaneous support originated from the support merging step.

The resulting algorithm, the Sparse Adaptive Orthogonal Matching Pursuit (SpAdOMP), is presented in Table I. Note that $\boldsymbol{\phi}_{|\Lambda}$ and $\boldsymbol{w}_{|\Lambda}$ denote the sub-vectors corresponding to the index set Λ, $\max(|a|, s)$ returns s indices of the largest elements of a and Λ^c represents the complement of set Λ. An important point to note about step 5 of Table I is that, although it is simple to implement, it is difficult to choose the step-size parameter μ which assures convergence. The Normalized LMS (NLMS) update addresses this issue by scaling with the input power

$$\boldsymbol{w}_{|\Lambda}(n) = \boldsymbol{w}_{|\Lambda}(n-1) + \frac{\mu}{\epsilon + \|\boldsymbol{\phi}_{|\Lambda}(n)\|^2}\boldsymbol{\phi}_{|\Lambda}^*(n)\varepsilon(n)$$

where $0 < \mu < 2$ and ϵ is a small positive constant (to avoid division by small numbers for stability purposes). NLMS may be viewed as an LMS with time-varying step-size. This observation justifies the superior tracking ability of NLMS with respect to LMS in non-stationary environments.

A. Compressed Sensing Matrices satisfying the ERIP

We find it useful to provide an example of measurement matrices satisfying the ERIP, before proceeding with the steady-state analysis of SpAdOMP. Consider an $n \times N$ matrix $\boldsymbol{\Phi}(n)$ whose rows are i.i.d. samples from a random Gaussian vector process distributed according to $\mathcal{N}(\mathbf{0}, \boldsymbol{R})$. Let $\Lambda := \operatorname{supp}(\boldsymbol{c})$. Now, consider the matrix $\boldsymbol{\Psi}_\Lambda(n) := \boldsymbol{\Phi}_\Lambda^{*T}(n)\boldsymbol{D}(n)\boldsymbol{\Phi}_\Lambda(n)$, where $\boldsymbol{\Phi}_\Lambda(n)$ is the sub-matrix of $\boldsymbol{\Phi}(n)$ corresponding to the index set Λ. The matrix $\boldsymbol{\Psi}_\Lambda(n)$ appears in the definition of the ERIP and its eigen-distribution is of interest. The matrix $\boldsymbol{\Psi}_\Lambda(n)$ can be expressed as follows:

$$\boldsymbol{\Psi}_\Lambda(n) = \sum_{k=1}^{n} \lambda^{n-k} \boldsymbol{\phi}_{|\Lambda}(k)\boldsymbol{\phi}_{|\Lambda}^{*T}(k) \tag{5}$$

where $\boldsymbol{\phi}_{|\Lambda}(k)$ is the kth row of $\boldsymbol{\Phi}_\Lambda(n)$. Hence, the (i,j)th element of $\boldsymbol{\Psi}_\Lambda(n)$ can be expressed as $\Psi_{\Lambda,ij}(n) = \sum_{k=1}^{n} \lambda^{n-k}\phi_{|\Lambda,i}(k)\phi_{|\Lambda,j}^{*T}(k)$. For simplicity, we assume that $\boldsymbol{R} = \sigma_\phi^2 \boldsymbol{I}$, hence the elements of each row of $\boldsymbol{\Phi}(n)$ are distributed i.i.d. and according to $\mathcal{N}(0, \sigma_\phi^2)$. Hence, the set $\{\phi_i(k)\}$ for $i = 1, 2, \cdots, N$ and $k = 1, 2, \cdots, n$ consists of i.i.d. zero mean Gaussian random variables with variance σ_ϕ^2.

The exponentially weighted random matrix $\boldsymbol{\Psi}_\Lambda(n)$ formed by the set $\{\phi_{|i}(k)\}_{i\in\Lambda}$, can be identified as the empirical estimate of the covariance matrix through an exponentially weighted moving average. Such random matrices often arise in portfolio optimization applications (See, for example, [21]). In [21], using the resolvent technique (See, for example [22]) the eigen-distribution of such matrices is studied and compared to that of Wishart ensembles. The main result of [21] implies that in the limit of $N \to \infty$ and $\lambda \to 1$, with $\beta := s/N < 1$ and $Q := 1/(s(1-\lambda))$ fixed, and $n \to \infty$, the eigenvalues of the matrix $(1-\lambda)\boldsymbol{\Psi}_\Lambda(n)$ (denoted as x) are distributed according to the density

$$\rho(x) = \frac{Qv}{\pi} \tag{6}$$

where v is the solution to the non-algebraic equation

$$\frac{x}{\sigma_\phi^2} - \frac{vx}{\tan(vx)} + \log(v\sigma_\phi^2) - \log\sin(vx) - \frac{1}{Q} = 0. \tag{7}$$

For example, by solving the above equation numerically for $Q = 100$ and $\sigma_\phi = 1$, the range of the eigen-distribution of $(1-\lambda)\boldsymbol{\Psi}_\Lambda(n)$ is found to be $[0.8652, 1.1482]$. By appropriately scaling the elements of $\boldsymbol{\Phi}(n)$, e.g. $\sigma_\phi^2 = 1/N$, one can obtain an upper bound of $\delta_s(\lambda, n) \leq 0.1482$ on the ERIP constant, as $n \to \infty$ and $\lambda \to 1$ while β and Q are fixed and $\beta < 1/Q$. As it is shown in [21], for finite but large values of N and n and λ close enough to 1, the empirical eigen-distribution is very similar to the asymptotic limit. Therefore, by the standard continuity characteristics of the eigen-distribution of random matrices, one expects to have $\delta_s(\lambda, n) \leq 0.1482$ for finite but large values of n and N, and λ sufficiently close to 1, with overwhelming probability. Note that the above concentration result can be extended to the case of correlated input sequences, which is studied in [22].

In parallel to the above limit process for the matrix $\boldsymbol{\Psi}_\Lambda(n)$, one can consider the alternate limit process of $\lambda = 1$, s/n and N/n fixed, and $n \to \infty$. This limit process gives rise to

the well-known Wishart ensemble, whose eigen-distribution is known [23]. In fact, as it is argued in [21], in first limit process the parameter $1/\log(1/\lambda)$ can be intuitively interpreted as the effective row dimension of $\boldsymbol{\Phi}_\Lambda(n)$ as $n \to \infty$. Simulation results in [21] show that the eigen-distribution of the exponentially weighted random matrix $\boldsymbol{\Psi}_\Lambda(n)$ is indeed very similar to that of the corresponding Wishart ensemble, by considering $1/\log(1/\lambda)$ as the effective row dimension.

The above example reveals that there is a close connection between the RIP and ERIP conditions (by interpreting $1/\log(1/\lambda)$ as the effective row dimension). The RIP constant of Gaussian measurement matrices has been extensively studied by Blanchard et al. [24]. The above parallelism suggests that one might be able to extend such results regarding the RIP of random measurement matrices to those satisfying ERIP. However, study of the eigen-distribution of the exponentially weighted matrices seems to offer more difficulty than their non-weighted counterparts.

B. Steady-State MSE of SpAdOMP

The following Theorem establishes the steady-state MSE performance of the SpAdOMP algorithm:

Theorem 1. *(SpAdOMP). Suppose that the input sequence $\phi(n)$ is stationary, i.e., its covariance matrix $\boldsymbol{R}(n) := \mathbb{E}\{\boldsymbol{\phi}(n)\boldsymbol{\phi}^{*T}(n)\} = \boldsymbol{R}$ is independent of n. Moreover, assume that $\boldsymbol{R}$ is non-singular. Finally, suppose that for n large enough, the ERIP constants $\delta_s(\lambda, n)$, $\delta_{2s}(\lambda, n)$, $\cdots$, $\delta_{4s}(\lambda, n)$ exist. Then, the SpAdOMP algorithm, for large n, produces a s-sparse approximation $\boldsymbol{c}(n)$ to the parameter vector $\boldsymbol{c}$ that satisfies the following steady-state bound:*

$$\epsilon_1(n) := \|\boldsymbol{c} - \boldsymbol{c}(n)\|_{\ell_2} \tag{8}$$
$$\lesssim C_1(n)\big\|\boldsymbol{D}^{1/2}(n)\boldsymbol{\eta}(n)\big\|_{\ell_2} + C_2(n)\big\|\boldsymbol{\phi}_{|\Lambda(n)}(n)\big\|_{\ell_2}|e_o(n)|$$

where $e_o(n)$ is the estimation error of the optimum Wiener filter, and $C_1(n)$ and $C_2(n)$ are constants independent of $\boldsymbol{c}$ (which are explicitly given in the Appendix) and are functions of $\lambda_M > 0$ (the minimum eigenvalue of $\boldsymbol{R}$), the ERIP constants $\delta_s(\lambda, n)$, $\delta_{2s}(\lambda, n)$, $\cdots$, $\delta_{4s}(\lambda, n)$ and the step size μ. The approximation in the above inequality is in the sense of the Direct-averaging technique [20] employed in simplifying the LMS iteration.

The proof is supplied in the Appendix. The above bound can be further simplified if one considers the normalization $\|\boldsymbol{\phi}(n)\|_{\ell_2}^2 = 1$ for all n. Such a normalization is implicitly assumed for the above example on the i.i.d. Gaussian measurement matrix as $n, N \to \infty$ with $\sigma_\phi^2 = 1/N$. In this case, $\big\|\boldsymbol{\phi}_{|\Lambda(n)}(n)\big\|_{\ell_2} \leq 1$ and thus the second term of the error bound simplifies to $C_2(n)|e_o(n)|$. Note that for large values of n, the isometry constants can be controlled. As shown in the example above, for a suitably random input sequence (*e.g.*, i.i.d. Gaussian input) and for n large enough, the restricted isometry constants can be sufficiently small. For example, if for n large enough, $\delta_{4s}(\lambda, n) \leq 0.01$ and $\mu\lambda_M = 0.75$, then $C_1(n) \approx 38.6$ and $C_2(n) \approx 7.7$. The corresponding coefficient for the CoSaMP algorithm will be $C_1(n) \approx 6.1$. Hence, the

parameters $C_1(n)$ and $C_2(n)$ can be well controlled by feeding enough number of measurements to the SpAdOMP algorithm.

The first term on the right hand side of the Eq. (8) is analogous to the steady-state error of the CoSaMP/SP algorithm, corresponding to a batch of data of size n. The second term is the steady-state error induced by performing a single LMS iteration, instead of using the LS estimate. This error term does not exist in the error expression of the CoSaMP/SP algorithm. However, this excess MSE error can be compromised by the significant complexity reduction incurred by removing the LS estimate stage. Note that the promising support tracking behavior of the CoSaMP/SP algorithm is inherited by the LMS iteration, where only the sub-vector of $\phi(n)$ corresponding to $\Lambda(n)$ and $\boldsymbol{w}_{|\Lambda(n)}$ participate in the LMS iteration, and hence the error term. In other words, the SpAdOMP enjoys the low complexity virtue of LMS, as well as the support detection superiority of the CoSaMP/SP. Indeed, this observation is evident in the simulation results, where the MSE curve of SpAdOMP is shifted from that of LMS towards that of the genie-aided LS estimate (See Section IV).

C. Sparse NARMA identification

The nonlinear model that we will be concerned with, constitutes a generalization of the class of linear ARMA models [25] and is known as Nonlinear AutoRegressive Moving Average (NARMA) [26]. The output of NARMA models depends on past and present values of the input as well as the output

$$y_i = f(y_{i-1}, \ldots, y_{i-M_y}, x_i, \ldots, x_{i-M_x}) + \eta_i \qquad (9)$$

where y_i, x_i and η_i are the system output, input and noise, respectively; M_y, M_x denote the output and input memory orders; η_i is Gaussian and independent of x_i; and $f(\cdot)$ is a sparse polynomial function in several variables with degree of nonlinearity p. Known linearization criteria [25] provide sufficient conditions for the Bounded Input Bounded Output stability of (9).

Using Kronecker products, we write Eq. (9) as a linear regression model

$$y_i = \phi^T(i)\boldsymbol{c} + \eta_i \qquad (10)$$

where

$$\phi^T(i) = \begin{bmatrix} \phi_y^T(i) & \phi_x^T(i) & \phi_{yx}^T(i) \end{bmatrix}$$

$\boldsymbol{y}_i = \begin{bmatrix} y_{i-1}, \cdots, y_{i-M_y} \end{bmatrix}^T$ and $\boldsymbol{x}_i = \begin{bmatrix} x_i, \cdots, x_{i-M_x} \end{bmatrix}^T$. Consider the pth order Kronecker powers $\boldsymbol{y}_i^{(p)} = \boldsymbol{y}_i^{\otimes p}$ and $\boldsymbol{x}_i^{(p)} = \boldsymbol{x}_i^{\otimes p}$. Then, the output and input regressor vectors are respectively given by $\phi_y^T(i) = [\boldsymbol{y}_i^{(1)}, \cdots, \boldsymbol{y}_i^{(p)}]$ and $\phi_x^T(i) = [\boldsymbol{x}_i^{(1)}, \cdots, \boldsymbol{x}_i^{(p)}]$. $\phi_{yx}^T(i)$ denotes all possible Kronecker product combinations of $\boldsymbol{y}_i$ and $\boldsymbol{x}_i$ of degree up to p. The components of $\boldsymbol{c} = \begin{bmatrix} \boldsymbol{c}_y^T & \boldsymbol{c}_x^T & \boldsymbol{c}_{yx}^T \end{bmatrix}^T$ correspond to the coefficients of the polynomial $f(\cdot)$. Hence, if we collect n successive observations, recovery of the sparsest parameter vector can be accomplished by solving the mathematical program (P_{ℓ_0}).

It must be noted that in NARMA models, the input sequence is non-linearly related to the measurement matrix $\Phi(n)$ through the multi-fold Kronecker product procedure. Thus, the effective measurement matrix generated by an i.i.d. input sequence, will not necessarily maintain the i.i.d. structure. Nevertheless, in case of linear models, by invoking the frequently adopted *independence assumption* [20], the i.i.d. property of the input sequence is carried over to the corresponding measurement matrix, and thus one might be able to guarantee analytically-provable controlled ERIP constants for the measurement matrix (as in the example of Section III-A). Although we have not mathematically established any results regarding the isometry of such structured matrices, simulation results reveal that input sequences which give rise to measurement matrices satisfying the ERIP in linear models (e.g., i.i.d. Gaussian), also perform well in conjunction with non-linear models (See Section IV). Nevertheless, the problem of designing input sequences, with mathematical guarantees on the ERIP of the corresponding measurement matrices in the non-linear models, is of interest and remains open.

D. Equalization/Predistortion in nonlinear communication channels

Nonlinearities in communication channels are caused by Power Amplifiers (PA) operating near saturation [27] and are addressed by channel inversion. Right inverses are called *predistorters* and are placed at the transmitter side; left inverses are termed *equalizers* and are part of the receiver. Predistorters are the preferred solution in single transceiver multiple receiver systems, such as a base station and multiple GSM receivers.

Channel inversion is conveniently effected when Eq. (9) is restricted to

$$y_i = b_0 x_i + f(y_{i-1}, \ldots, y_{i-M_y}, x_{i-1}, \ldots, x_{i-M_x}) + \eta_i. \qquad (11)$$

In the above equation the present input sample enters linearly. If x_i entered polynomially, inversion would require finding the roots of a polynomial which does not always result in a unique solution and is computationally expensive. The inverse of Eq. (11) is given by

$$x_i = b_0^{-1} \left[y_i - f(y_{i-1}, \ldots, y_{i-M_y}, x_{i-1}, \ldots, x_{i-M_x}) - \eta_i \right],$$
$$\text{iff } b_0 \neq 0. \qquad (12)$$

Note that modulo the scaling by b_0 correction, the system and its inverse are generated by the same function. Hence, estimation of the direct process is equivalent to the estimation of the reverse process.

IV. EXPERIMENTAL RESULTS

In this section we compare through computer simulations the performance of existing algorithms and the algorithm proposed in this paper. Experiments were conducted on both linear and nonlinear channel setups. In all experiments the output sequence is disturbed by additive white Gaussian noise for various SNR levels ranging from 5 to $26dB$. SNR is the ratio of the noiseless channel output power to the noise power corrupting the output signal (σ_y^2/σ_η^2). The Normalized Mean Square Error, defined as

$$\mathbb{E}[\|\boldsymbol{c}(n) - \boldsymbol{c}\|_{\ell_2}^2] / \mathbb{E}[\|\boldsymbol{c}\|_{\ell_2}^2]$$

(a) ARMA channel   (b) Learning curve for SNR=23dB
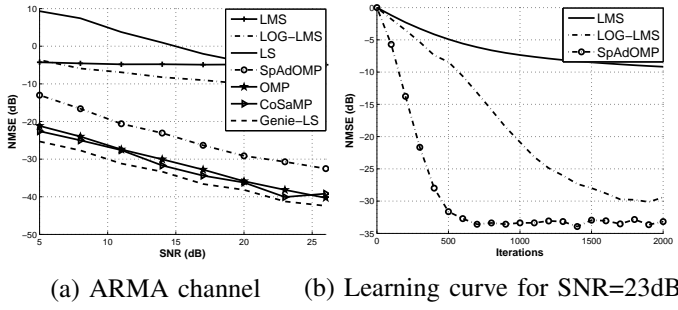
Fig. 1. NMSE of the channel estimates versus SNR on a linear channel

(a) Real part   (b) Imaginary part

Fig. 2. Time evolution of a_1 signal entry on a linear ARMA channel

TABLE II
CHOICE OF SPARSE PARAMETERS FOR LOG-LMS

SNR	5-8	11-17	20-26
γ^a	7×10^{-4}	8×10^{-4}	9×10^{-4}

$^a \mu = 2 \times 10^{-2}$, $\epsilon = 10$

is used to assess performance.

A. Sparse ARMA channel identification

In the first experiment sparse ARMA channel estimation is considered. The channel memory is $M_y = M_x = 50$ and the channel to be estimated is given by

$$y_n = a_1 y_{n-6} + a_2 y_{n-48} + x_n + b_1 x_{n-13} + b_2 x_{n-34}$$

where $[a_1, a_2] = [-0.5167 - j0.2828, 0.1801 + j0.1347]$ and $[b_1, b_2] = [-0.5368 - j0.9198, 1.0719 + j0.0318]$. The system is stable as the roots of the AR part are inside the unit circle.

The input sequence is drawn from a complex Gaussian distribution, $\mathcal{CN}(0, 1/5)$. To reduce the realization dependency, the parameter estimates were averaged over 30 Monte Carlo runs. Program (P_{ℓ_0}) is solved by the CoSaMP [1], OMP [8], [9], log-LMS [12] and SpAdOMP. Moreover, two conventional methods were used, namely, the Least Squares (LS) and the LMS algorithm. The number of samples processed was 500. The sparsity tuning parameter required by the log-LMS is summarized in Table II. The step size for the conventional LMS and the SpAdOMP was set to $\mu_{\text{LMS}} = 1 \times 10^{-2}$ and $\mu_{\text{SpAdOMP}} = 7 \times 10^{-2}$. Note that the choice of the step size μ is made near the instability point of each algorithm to provide the maximum possible convergence speed.

Fig. 1(a) shows the excellent performance match between the Genie LS, CoSaMP and OMP, all of which have quadratic complexity. The LMS, log-LMS and SpAdOMP have an order of magnitude less computational complexity, but only SpAdOMP achieves a performance gain close to Genie LS ($9dB$ less). If we repeat this experiment for a fixed SNR level of $23dB$ and process 2000 samples, then as shown in Fig. 1(b), log-LMS improves by $20dB$; however, it achieves $4dB$ less performance gain than SpAdOMP.

To demonstrate the support tracking ability of SpAdOMP, we run this experiment and after 300 iterations we set a_1 to zero. This time, since we have a support varying environment,
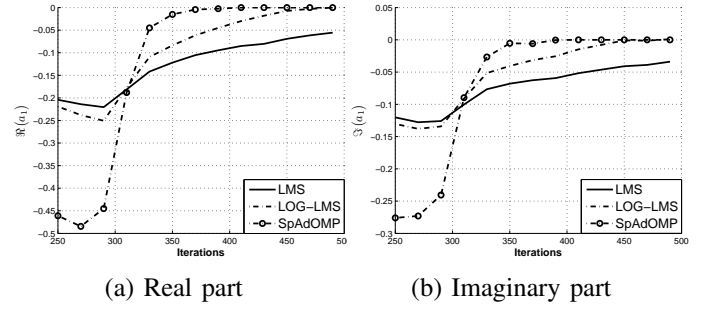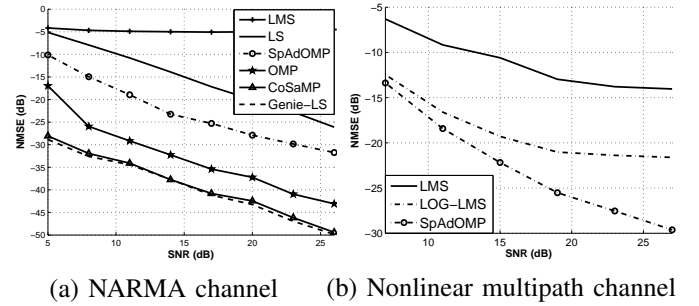

(a) NARMA channel   (b) Nonlinear multipath channel

Fig. 3. NMSE of the channel estimates versus SNR on nonlinear channels

λ is set to $\lambda = 0.8$ in SpAdOMP. Fig. 2 illustrates the time evolution of the estimates of a_1. We note from Fig. 2, that the conventional LMS does not take into account sparsity and hence the estimates are nonzero; while log-LMS and SpAdOMP succeed in estimating the zero entries. However, SpAdOMP has a much faster support tracking behavior for the estimation of the zero entries in comparison to log-LMS.

B. Sparse NARMA channel identification

In the second experiment the following NARMA channel is considered

$$y_n = a_1 y_{n-50} + a_2 y_{n-9}^2 + b_1 x_{n-8} + b_2 |x_{n-21}|^2 x_{n-21}$$

where $[a_1, a_2] = [-0.1586 - j0.7064, -0.1428 - j0.0478]$ and $[b_1, b_2] = [-0.8082 - j0.5221, -0.5177 + j0.7131]$ and the channel memory is $M_y = M_x = 50$.

The experiment is based on 30 Monte Carlo runs and the input sequence is generated from a complex Gaussian distribution, $\mathcal{CN}(0, 1/4)$, consisting of 1000 samples. This time, the methods used are CoSaMP, OMP and SpAdOMP, along with the standard LMS algorithm and least squares. The step size parameters $\mu_{LMS} = 6 \times 10^{-3}$ and $\mu_{SpAdOMP} = 0.3$ are used for the conventional LMS and SpAdOMP. OMP and SpAdOMP lag behind Genie LS by $5dB$ and $12dB$ respectively in performance. It is worth pointing out that SpAdOMP obtains an average gain of nearly $19dB$ over the conventional LMS. Note that this significant NMSE gain is the product of both the denoising mechanism and the compressed sampling virtue of the CoSaMP algorithm, which are lacking in conventional adaptive algorithms such as LMS.

C. Sparse nonlinear multipath channel identification

In this channel setup, a cubic baseband Hammerstein wireless channel with four Rayleigh fading rays (two on the liner and two on the cubic part) is employed; all rays fade at the same Doppler frequency of $f_D = 80Hz$ with sampling period $T_s = 0.8\mu s$. The channel memory length is equal to $M_1 = M_3 = 50$ (for both the linear and cubic parts) and the position of the fading rays is randomly chosen. In this experiment, 2000 samples from a complex Gaussian distribution $\mathcal{CN}(0, 1/4)$ were processed. Fig. 3(b) illustrates that SpAdOMP provides an average gain of $11dB$, over the conventional LMS and $5dB$ over the log-LMS developed in [12].

V. Conclusions

In this paper, an adaptive algorithm for sparse approximations with linear complexity was developed using the underlying principles of existing batch-greedy algorithms. Analytical bounds on the steady-state MSE are obtained, which highlight the superior performance of the proposed algorithm. The proposed algorithm was applied to sparse NARMA identification and in particular to NARMA channel equalization/predistortion. Simulation results validated the superior performance of the new algorithm. Future research is focused on blind algorithms for sparse system identification.

Appendix
Proof of Theorem 1

Note that, unlike CoSaMP, the iterations of SpAdOMP are not applied to a fixed batch of measurements. Hence, we need to revisit the error analysis of CoSaMP taking into account the time variations. Recall that the LMS update for $\boldsymbol{w}_{|\Lambda(n)}(n)$ is given by

$$
\begin{aligned}
\boldsymbol{w}_{|\Lambda(n)}(n) = \boldsymbol{w}_{|\Lambda(n)}(n-1) \qquad\qquad (13)\\
+ \mu\boldsymbol{\phi}_{|\Lambda(n)}^*(n)\big(y(n) - \boldsymbol{\phi}_{|\Lambda(n)}^T(n)\boldsymbol{w}_{|\Lambda(n)}(n-1)\big)
\end{aligned}
$$

Suppose that the estimate at time n is given by $\boldsymbol{c}(n)$. Let

$$
\epsilon_1(n) := \|\boldsymbol{c} - \boldsymbol{c}(n)\|_{\ell_2}, \quad \epsilon_2(n) := \|\boldsymbol{w}_{|\Lambda(n)}(n) - \boldsymbol{w}_{o|\Lambda(n)}\|_{\ell_2} \tag{14}
$$

where $\boldsymbol{w}_{o|\Lambda(n)}$ is the *optimum Wiener solution* restricted to the set $\Lambda(n)$, given by

$$
\boldsymbol{w}_{o|\Lambda(n)} := \boldsymbol{R}_{|\Lambda(n)}^{-1}\boldsymbol{r} \tag{15}
$$

with $\boldsymbol{R} := \mathbb{E}\{\boldsymbol{\phi}(n)\boldsymbol{\phi}^{*T}(n)\}$ and $\boldsymbol{r} := \mathbb{E}\{\boldsymbol{\phi}^*(n)y(n)\}$. One can write

$$
\begin{aligned}
\boldsymbol{w}_{|\Lambda(n)}(n) - \boldsymbol{w}_{o|\Lambda(n)} = \Big(\boldsymbol{I}_{|\Lambda(n)} - \mu\boldsymbol{\phi}_{|\Lambda(n)}(n)\boldsymbol{\phi}_{|\Lambda(n)}^{*T}(n)\Big)\\
\times \Big\{\big(\boldsymbol{w}_{|\Lambda(n-1)}(n-1) - \boldsymbol{w}_{o|\Lambda(n-1)}\big)\\
+ \big(\boldsymbol{w}_{|\Lambda(n)}(n-1) - \boldsymbol{w}_{|\Lambda(n-1)}(n-1)\big)\\
+ \big(\boldsymbol{w}_{o|\Lambda(n-1)} - \boldsymbol{w}_{o|\Lambda(n)}\big)\Big\} + \mu\boldsymbol{\phi}_{|\Lambda}^*(n)e_o(n) \tag{16}
\end{aligned}
$$

where $e_o(n)$ is the estimation error of the optimum Wiener filter, given by $e_o(n) := (y(n) - \boldsymbol{\phi}_{|\Lambda}^T(n)\boldsymbol{w}_{o|\Lambda})$. Invoking

the *Direct-Averaging* approximation [20], one can substitute $\boldsymbol{\phi}_{|\Lambda(n)}(n)\boldsymbol{\phi}_{|\Lambda(n)}^{*T}(n)$ with $\boldsymbol{R}_{|\Lambda(n)}$. Hence,

$$
\begin{aligned}
\epsilon_2(n) \leq (1 - \mu\lambda_M)\epsilon_2(n-1) + \mu\|\boldsymbol{\phi}_{|\Lambda}(n)\|_2|e_o(n)|\\
+ (1 - \mu\lambda_M)\Big\{\big\|\boldsymbol{w}_{|\Lambda(n)}(n-1) - \boldsymbol{w}_{|\Lambda(n-1)}(n-1)\big\|_{\ell_2}\\
+ \big\|\boldsymbol{w}_{o|\Lambda(n-1)} - \boldsymbol{w}_{o|\Lambda(n)}\big\|_{\ell_2}\Big\} \tag{17}
\end{aligned}
$$

where λ_M is the minimum eigenvalue of $\boldsymbol{R}$. Here we assume that the covariance matrix $\boldsymbol{R}$ is non-singular, *i.e.*, $\lambda_M > 0$. Note that the direct-averaging method yields a reasonable approximation particularly when $\mu \ll 1$ [28]. A more direct and rigorous convergence analysis of the LMS algorithm is possible, which is much more complicated [29]. Hence, for the sake of simplicity and clarity of the analysis, we proceed with the direct-averaging approach.

In order to obtain a closed set of difference equations for $\epsilon_1(n)$ and $\epsilon_2(n)$, we need to express the third and fourth terms of Eq. (17) in terms of $\epsilon_1(n)$ and $\epsilon_2(n)$ (and time-shifts thereof). First, we consider the third term. Let

$$
\boldsymbol{\delta}(n-1) := \boldsymbol{w}(n-1) - \boldsymbol{c}. \tag{18}
$$

Note that $\boldsymbol{w}(n-1)$ is supported on the index set $\Lambda(n-1)$. Hence,

$$
\begin{aligned}
\big\|\boldsymbol{w}_{|\Lambda(n)}(n-1) - \boldsymbol{w}_{|\Lambda(n-1)}(n-1)\big\|_{\ell_2}\\
= \big\|\boldsymbol{c}_{\Lambda(n)\Delta\Lambda(n-1)} + \boldsymbol{\delta}_{\Lambda(n)\Delta\Lambda(n-1)}(n-1)\big\|_{\ell_2}\\
\leq \big\|\boldsymbol{c}_{\Lambda(n)\Delta\Lambda(n-1)}\big\|_{\ell_2} + \big\|\boldsymbol{\delta}(n-1)\big\|_{\ell_2} \quad (19)
\end{aligned}
$$

where Δ represents the symmetric difference of $\Lambda(n)$ and $\Lambda(n-1)$. The key here is the fact that the support estimates $\Lambda(n-1)$ and $\Lambda(n)$ contain most of the energy of the true vector $\boldsymbol{c}$, due to the restricted isometry of the measurement matrix and the construction of the proxy signal. Consider the squared form of the first term in the above equation:

$$
\begin{aligned}
\big\|\boldsymbol{c}_{\Lambda(n)\Delta\Lambda(n-1)}\big\|_{\ell_2}^2 = \big\|\boldsymbol{c}_{\Lambda(n)\cap\Lambda^c(n-1)}\big\|_{\ell_2}^2 + \big\|\boldsymbol{c}_{\Lambda(n-1)\cap\Lambda^c(n)}\big\|_{\ell_2}^2\\
\leq \big\|\boldsymbol{c}_{\Lambda^c(n-1)}\big\|_{\ell_2}^2 + \big\|\boldsymbol{c}_{\Lambda^c(n)}\big\|_{\ell_2}^2 \tag{20}
\end{aligned}
$$

Hence,

$$
\big\|\boldsymbol{c}_{\Lambda(n)\Delta\Lambda(n-1)}\big\|_{\ell_2} \leq \sqrt{2}\max\Big\{\big\|\boldsymbol{c}_{\Lambda^c(n-1)}\big\|_{\ell_2}, \big\|\boldsymbol{c}_{\Lambda^c(n)}\big\|_{\ell_2}\Big\} \tag{21}
$$

Lemmas 4.2 and 4.3 of [1] provide the following bound on $\|\boldsymbol{c}_{\Lambda^c(n)}\|_{\ell_2}$:

$$
\big\|\boldsymbol{c}_{\Lambda^c(n)}\big\|_{\ell_2} \leq \gamma(n)\epsilon_1(n-1) + \xi(n)\big\|\boldsymbol{\eta}'(n)\big\|_{\ell_2} \tag{22}
$$

where

$$
\gamma(n) := \frac{\delta_{2s}(\lambda, n) + \delta_{4s}(\lambda, n)}{1 - \delta_{2s}(\lambda, n)}, \quad \xi(n) := \frac{2\sqrt{1 + \delta_{2s}(\lambda, n)}}{1 - \delta_{2s}(\lambda, n)}, \tag{23}
$$

and

$$
\boldsymbol{\eta}'(n) := \boldsymbol{D}^{1/2}(n)\boldsymbol{\eta}(n) + \text{Diag}\big(\boldsymbol{\Phi}(n)\boldsymbol{\Theta}(n)\big) \tag{24}
$$

with $\boldsymbol{\Theta}_{ij}(n) := \lambda^{n-j-1}\big(\boldsymbol{c}_i(n) - \boldsymbol{c}_i(j)\big)$, for $i = 1, 2, \cdots, N$ and $j = 1, 2, \cdots, n-1$. The *effective* noise vector $\boldsymbol{\eta}'(n)$ consists of two parts: the first term is the exponentially-weighted additive noise vector, and the second term is the excess error

due to the adaptive update of the proxy signal (in contrast to the batch construction used in the CoSaMP algorithm). Note that the isometry constants $\delta_s(\lambda, n), \cdots, \delta_{4s}(\lambda, n)$ are all functions of n, since the matrix $\mathbf{\Phi}(n)$ depends on n. If the input sequence is generated by a stationary source, for n large enough, one can approximate the covariance matrix $\mathbf{R}$ by the exponentially weighted sample covariance $\mathbf{\Phi}^{*T}(n)\mathbf{D}(n)\mathbf{\Phi}(n)$. Similarly, one can approximate $\mathbf{r}$ by $\mathbf{\Phi}^{*T}(n)\mathbf{D}(n)\mathbf{r}(n)$. In this case, we have $\mathbf{w}_{o|\Lambda(n)} \approx \mathbf{b}(n)$, where $\mathbf{b}(n)$ is the exponentially-weighted least squares solution restricted to the index set $\Lambda(n)$, given by

$$\mathbf{b}(n) := \begin{cases} \left(\mathbf{D}^{1/2}(n)\mathbf{\Phi}(n)\right)^{\dagger}_{|\Lambda(n)}\mathbf{D}^{1/2}(n)\mathbf{r}(n), & \text{on } \Lambda(n) \\ 0, & \text{elsewhere} \end{cases}$$
(25)

Using this approximation, the ℓ_2-norm of $\boldsymbol{\delta}(n-1)$ can be bounded as follows:

$$\left\|\boldsymbol{\delta}(n-1)\right\|_{\ell_2} \leq \left\|\mathbf{w}(n-1) - \mathbf{b}(n-1)\right\|_{\ell_2} + \left\|\mathbf{b}(n-1) - \mathbf{c}\right\|_{\ell_2}$$
$$\leq \epsilon_2(n-1) + \left\|\mathbf{b}(n-1) - \mathbf{c}\right\|_{\ell_2} \quad (26)$$

Moreover, using Lemmas 4.2, 4.3, and 4.4 of [1], one can express $\|\mathbf{c} - \mathbf{b}(n)\|_{\ell_2}$ in terms of $\epsilon_1(n)$ and $\boldsymbol{\eta}'(n)$ as follows:

$$\|\mathbf{c} - \mathbf{b}(n)\|_2 \leq \frac{1}{2}\alpha(n)\epsilon_1(n-1) + \frac{1}{2}\beta(n)\left\|\boldsymbol{\eta}'(n)\right\|_{\ell_2}, \quad (27)$$

where

$$\alpha(n) := 2\left(1 + \frac{\delta_{4s}(\lambda, n)}{1 - \delta_{3s}(\lambda, n)}\right)\gamma(n), \quad (28)$$

$$\beta(n) := \frac{2}{\sqrt{1 - \delta_{3s}(\lambda, n)}} + 2\left(1 + \frac{\delta_{4s}(\lambda, n)}{1 - \delta_{3s}(\lambda, n)}\right)\xi(n). \quad (29)$$

Denoting $\|\mathbf{c} - \mathbf{b}(n)\|_{\ell_2}$ by $\epsilon_3(n)$ and using Eqs. (21), (26), and (27), one can obtain the following recurrence relation for $\epsilon_2(n)$:

$$\epsilon_2(n) \leq (1 - \mu\lambda_M)\epsilon_2(n-1) + \mu\|\boldsymbol{\phi}_{|\Lambda}(n)\|_2|e_o(n)|$$
$$+ (1 - \mu\lambda_M)\left\{\left\|\mathbf{w}_{o|\Lambda(n-1)} - \mathbf{c}\right\|_{\ell_2} + \left\|\mathbf{w}_{o|\Lambda(n)} - \mathbf{c}\right\|_{\ell_2}\right.$$
$$\left. + \left\|\mathbf{w}_{|\Lambda(n)}(n-1) - \mathbf{w}_{|\Lambda(n-1)}(n-1)\right\|_{\ell_2}\right\}$$
$$\leq (1 - \mu\lambda_M)\epsilon_2(n-1) + \mu\|\boldsymbol{\phi}_{|\Lambda}(n)\|_{\ell_2}|e_o(n)|$$
$$+ (1 - \mu\lambda_M)\left\{\epsilon_3(n) + 2\epsilon_3(n-1) + \epsilon_2(n-1)\right\}$$
$$+ \sqrt{2}(1 - \mu\lambda_M)\max\left\{\gamma(n)\epsilon_1(n-1) + \xi(n)\|\boldsymbol{\eta}'(n)\|_{\ell_2},\right.$$
$$\left. \gamma(n-1)\epsilon_1(n-2) + \xi(n-1)\|\boldsymbol{\eta}'(n-1)\|_{\ell_2}\right\} \quad (30)$$

From Lemma 4.5 of Needell et al. [1], one can write

$$\epsilon_1(n) := \|\mathbf{c} - \mathbf{c}(n)\|_{\ell_2} \quad (31)$$
$$\leq \|\mathbf{c} - \mathbf{b}_s(n)\|_{\ell_2} + \|\mathbf{b}_s(n) - \mathbf{c}(n)\|_{\ell_2}$$
$$\leq 2\|\mathbf{c} - \mathbf{b}(n)\|_{\ell_2} + 4\|\mathbf{b}(n) - \mathbf{w}(n)\|_{\ell_2}$$
$$\leq 2\|\mathbf{c} - \mathbf{b}(n)\|_{\ell_2} + 4\|\mathbf{w}_{o|\Lambda(n)} - \mathbf{w}(n)\|_{\ell_2}$$
$$+ 4\|\mathbf{w}_{o|\Lambda(n)} - \mathbf{b}(n)\|_{\ell_2}$$

where the last line of Eq. (31) is obtained from the second line by adding and subtracting $\mathbf{w}_{o|\Lambda(n)}$ from $\mathbf{b}(n) - \mathbf{w}(n)$, and

using the triangle inequality. The last term on the right hand side of Eq. (31) denotes the difference between the optimum Wiener solution and the LS solution, both restricted to the index set $\Lambda(n)$. As mentioned earlier, one can approximate the covariance matrix $\mathbf{R}$ by the exponentially weighted sample covariance $\mathbf{\Phi}^{*T}(n)\mathbf{D}(n)\mathbf{\Phi}(n)$, and the correlation vector $\mathbf{r}$ by $\mathbf{\Phi}^{*T}(n)\mathbf{D}(n)\mathbf{r}(n)$. In this case, we have $\mathbf{w}_{o|\Lambda(n)} \approx \mathbf{b}(n)$, and hence the contribution of the last term on the right hand side of Eq. (31) to the steady-state error becomes negligible. Also, by construction, the estimate $\mathbf{w}(n)$ is supported on the index set $\Lambda(n)$. Hence, the second term of Eq. (31) can be identified as $4\|\mathbf{w}_{o|\Lambda(n)} - \mathbf{w}_{|\Lambda(n)}(n)\|_2 = 4\epsilon_2(n)$. With the above-mentioned simplifications, one can arrive at the following set of non-linearly coupled difference equations for $\epsilon_1(n)$, $\epsilon_2(n)$ and $\epsilon_3(n)$:

$$\begin{cases} \epsilon_1(n) \leq 2\epsilon_3(n) + 4\epsilon_2(n) \\ \epsilon_2(n) \leq (1 - \mu\lambda_M)\left\{2\epsilon_2(n-1) + 2\epsilon_3(n-1) + \epsilon_3(n)\right\} \\ \quad + \sqrt{2}(1 - \mu\lambda_M)\max\left\{\gamma(n)\epsilon_1(n-1) + \xi(n)\|\boldsymbol{\eta}'(n)\|_{\ell_2},\right. \\ \qquad \left. \gamma(n-1)\epsilon_1(n-2) + \xi(n-1)\|\boldsymbol{\eta}'(n-1)\|_{\ell_2}\right\} \\ \quad + \mu\|\boldsymbol{\phi}_{|\Lambda(n)}(n)\|_{\ell_2}|e_o(n)| \\ \epsilon_3(n) \leq \frac{1}{2}\alpha(n)\epsilon_1(n-1) + \frac{1}{2}\beta(n)\|\boldsymbol{\eta}'(n)\|_{\ell_2} \end{cases}$$
(32)

Although the above set of difference equations is sufficient to obtain the error measures $\epsilon_1(n)$, $\epsilon_2(n)$, and $\epsilon_3(n)$ for all n, the solution is non-trivial for general n due to its high non-linearity. However, for large n, it is possible to obtain the steady-state solution. It is easy to substitute $\epsilon_3(n)$ in terms of $\epsilon_1(n)$. Also, for large enough n, the arguments of the $\max\{\cdot, \cdot\}$ operator do not vary significantly with n. Hence, we can substitute the maximum with the second argument. Hence, the steady-state values of $\epsilon_1(n)$ and $\epsilon_2(n)$ can be obtained from the following equation:

$$\begin{pmatrix} 1 - \alpha(n) & -4 \\ -(1 - \mu\lambda_M)\left(\frac{3}{2}\alpha(n) + \sqrt{2}\gamma(n)\right) & 1 - 2(1 - \mu\lambda_M) \end{pmatrix}\begin{pmatrix} \epsilon_1(n) \\ \epsilon_2(n) \end{pmatrix}$$
$$\leq \|\mathbf{D}^{1/2}(n)\boldsymbol{\eta}(n)\|_{\ell_2}\begin{pmatrix} \beta(n) \\ (1 - \mu\lambda_M)\left(\frac{3}{2}\beta(n) + \sqrt{2}\xi(n)\right) \end{pmatrix}$$
$$+ \mu\|\boldsymbol{\phi}_{|\Lambda(n)}(n)\|_{\ell_2}|e_o(n)|\begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad (33)$$

Note that, the contribution of proxy error term in $\boldsymbol{\eta}'(n)$ becomes negligible for large n, due to the effect of forgetting factor, and the fact that the estimates $\mathbf{c}(n)$ do not vary much with n. Hence, we can approximate $\boldsymbol{\eta}'(n)$ by $\mathbf{D}^{1/2}(n)\boldsymbol{\eta}(n)$ for large n. In particular, the asymptotic solution to $\epsilon_1(n)$ is given by:

$$\epsilon_1(n) \lesssim C_1(n)\|\mathbf{D}^{1/2}(n)\boldsymbol{\eta}(n)\|_{\ell_2} + C_2(n)\|\boldsymbol{\phi}_{|\Lambda(n)}(n)\|_{\ell_2}|e_o(n)| \quad (34)$$

where,

$$C_1(n) := \frac{4(1 - \mu\lambda_M)\left(\frac{3}{2}\beta(n) + \sqrt{2}\xi(n)\right)}{\Delta(n)}$$
$$+ \frac{\left(1 - 2(1 - \mu\lambda_M)\right)\beta(n)}{\Delta(n)}, \quad (35)$$

$$C_2(n) := \frac{4\mu}{\Delta(n)}, \quad (36)$$

and

$$\Delta(n) := (2\mu\lambda_M - 1) - (5 - 4\mu\lambda_M)\alpha(n) - 4\sqrt{2}(1 - \mu\lambda_M)\gamma(n). \tag{37}$$

Note that a sufficient condition for the above bound to hold is $\Delta(n) > 0$.

REFERENCES

[1] D. Needell and J. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comput. Harmon. Anal*, vol. 26, pp. 301–321, 2009.

[2] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2230–2249, 2009.

[3] A. Bruckstein, D. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, no. 1, pp. 34–81, 2009.

[4] W. Schreiber, "Advanced television systems for terrestrial broadcasting: Some problems and some proposed solutions," vol. 83, pp. 958–981, 1995.

[5] W. Bajwa, J. Haupt, G. Raz, and R. Nowak, "Compressed channel sensing," in *Proc. IEEE CISS*, 2008, pp. 5–10.

[6] M. Kocic, D. Brady, and M. Stojanovic, "Sparse equalization for real-time digital underwater acoustic communications," in *Proc. IEEE OCEANS*, 1995, pp. 1417–1422.

[7] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, 1998.

[8] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *27th Asilomar Conf.on Signals, Systems and Comput.*, 1993, pp. 40–44.

[9] S. Davis, G.M. Mallat and Z. Zhang, "Adaptive time-frequency decompositions," *SPIE J. Opt. Engin.*, vol. 33, no. 7, pp. 2183–2191, 1994.

[10] D. Needell and R. Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Found. Comput. Math*, vol. 9, no. 3, pp. 317–334, 2009.

[11] D. Donoho, Y. Tsaig, I. Drori, and J. Starck, "Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit," *Submitted for publication*.

[12] Y. Chen, Y. Gu, and A. Hero, "Sparse LMS for system identification," in *Proc. IEEE ICASSP*, 2009, pp. 3125–3128.

[13] D. Angelosante and G. Giannakis, "RLS-weighted LASSO for adaptive estimation of sparse signals," in *Proc. IEEE ICASSP*, 2009.

[14] B. Babadi, N. Kalouptsidis, and V. Tarokh, "SPARLS: The sparse RLS algorithm," *submited in IEEE Trans. Signal Process.*, 2009.

[15] N. Kalouptsidis, G. Mileounis, B. Babadi, and V. Tarokh, "Adaptive algorithms for sparse nonlinear channel estimation," in *Proc. IEEE SSP*, 2009, pp. 221–224.

[16] E. Candés and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, 2004.

[17] M. Figueiredo, R. Nowak, and S. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 586–597, 2007.

[18] I. Daubechies, M. Defrise, and C. Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Comm. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, 2004.

[19] J. Tropp, "Greed is good: algorithmic results for sparse approximation," *IEEE Trans. Inf. Theory*, vol. 50, no. 10, pp. 2231–2242, 2004.

[20] S. Haykin, *Adaptive Filter Theory*. Prentice Hall, 1996.

[21] S. Pafka, M. Potters, and I. Kondor, "Exponential weighting and random-matrix-theory-based filtering of financial covariance matrices for portfolio optimization," *Arxiv preprint cond-mat/0402573 (available at http://arxiv.org/abs/cond-mat/0402573)*, 2004.

[22] A. M. Sengupta and P. P. Mitra, "Distributions of singular values for some random matrices," *Physical Review E*, vol. 60, no. 3, 1999.

[23] S. Geman, "A limit theorem for the norm of random matrices," *Ann. Probab.*, vol. 8, no. 2, pp. 252–261, 1980.

[24] J. Blanchard, C. Cartis, and J. Tanner, "Compressed sensing: How sharp is the restricted isometry property?" *submitted*.

[25] N. Kalouptsidis, *Signal Processing Systems: Theory & Design*. Wiley, 1997.

[26] S. Chen and S. Billings, "Representations of non-linear systems: the NARMAX model," *Int. J. Control*, vol. 49, no. 3, pp. 1013–1032, 1989.

[27] S. Benedetto and E. Biglieri, *Principles of Digital Transmission: with wireless applications*. Springer, 1999.

[28] H. Kushner, *Approximation and Weak Convergence Methods for Random Processes with Applications to Stochastic System Theory*. MIT Press, 1984.

[29] E. Eweda and O. Macchi, "Convergence of an adaptive linear estimation algorithm," *IEEE Trans. Autom. Control*, vol. 29, no. 2, pp. 119–127, 1984.

Gerasimos Mileounis (S'04) received the BEng. degree in Electronic Engineering and Computer Science from Aston University, UK, and the MSc. (Eng.) degree in Data Communications from University of Sheffield, UK, in 2004 and 2005, respectively.

He is currently working towards the Ph.D degree in the Department of Informatics and Telecommunications at the University of Athens, Greece. His research interests include nonlinear system identification/equalization for communications, higher-order statistics and compressed sensing.

Behtash Babadi (S'08) received the BSc. degree in Electrical Engineering from Sharif University of Technology, Tehran, Iran and the MSc. in Engineering Sciences from Harvard University, Cambridge, MA, in 2006 and 2008, respectively.

He is currently working towards the PhD degree in Engineering Sciences at the School of Engineering and Applied Sciences, Harvard University, Cambridge, MA. His research interests include dynamic spectrum access networks, adaptive signal processing and compressed sensing.

Nicholas Kalouptsidis (M'82-SM'85) was born in Athens, Greece, on September 13, 1951. He received the B.Sc. degree in mathematics (with highest honors) from the University of Athens, Athens, Greece, in 1973 and the M.S. and Ph.D. degrees in systems science and mathematics from Washington University, St. Louis, MO, in 1975 and 1976 respectively.

He has held visiting positions at Washington University, St. Louis, MO; Politecnico di Torino, Turin, Italy; Northeastern University, Boston, MA; and CNET Lannion, France. He has been an Associate Professor and Professor with the Department of Physics, University of Athens. In Fall 1998, he was a Clyde Chair Professor with the School of Engineering, University of Utah, Salt Lake City. In Spring 2008, he was a visiting scholar at Harvard university. He is currently a Professor with the Department of Informatics and Telecommunications, University of Athens. He is the author of the textbook *Signal Processing Systems: Theory and Design* (New York: Wiley, 1997) and coeditor, with S. Theodoridis, of the book *Adaptive System Identification and Signal Processing Algorithms* (Englewood Cliffs, NJ: Prentice-Hall, 1993). His research interests are in system theory and signal processing.

Vahid Tarokh (M'97-SM'02-F'09) received the Ph.D. degree in electrical engineering from the University of Waterloo, Waterloo, ON, Canada, in 1995.

He is a Perkins Professor of Applied Mathematics and Hammond Vinton Hayes Senior Fellow of Electrical Engineering at Harvard University, Cambridge, MA. At Harvard, he teaches courses and supervises research in communications, networking and signal processing.

Dr. Tarokh has received a number of major awards and holds two honorary degrees.