Online Adaptive Estimation of Sparse Signals: Where RLS Meets the ℓ_1-Norm

Daniele Angelosante, *Member, IEEE*, Juan Andrés Bazerque, *Student Member, IEEE*, and Georgios B. Giannakis, *Fellow, IEEE*

***Abstract*—Using the ℓ_1-norm to regularize the least-squares criterion, the batch least-absolute shrinkage and selection operator (Lasso) has well-documented merits for estimating sparse signals of interest emerging in various applications where observations adhere to parsimonious linear regression models. To cope with high complexity, increasing memory requirements, and lack of tracking capability that batch Lasso estimators face when processing observations sequentially, the present paper develops a novel time-weighted Lasso (TWL) approach. Performance analysis reveals that TWL cannot estimate consistently the desired signal support without compromising rate of convergence. This motivates the development of a time- and norm-weighted Lasso (TNWL) scheme with ℓ_1-norm weights obtained from the recursive least-squares (RLS) algorithm. The resultant algorithm consistently estimates the support of sparse signals without reducing the convergence rate. To cope with sparsity-aware recursive real-time processing, novel adaptive algorithms are also developed to enable online coordinate descent solvers of TWL and TNWL that provably converge to the true sparse signal in the time-invariant case. Simulated tests compare competing alternatives and corroborate the performance of the novel algorithms in estimating time-invariant signals, and tracking time-varying signals under sparsity constraints.**

***Index Terms*—Adaptive algorithms, compressive sampling, coordinate descent, RLS, sparse linear regression.**

I. Introduction

SPARSITY is an attribute present in a plethora of natural as well as man-made signals and systems. This is reasonable not only because nature itself is parsimonious but also because processing and simple models with minimal degrees of freedom are attractive from an implementation perspective. Exploitation of sparsity is critical in applications as diverse as variable selection in linear regression models for diabetes [24], image compression [5], distributed spectrum sensing for cognitive radios [3], estimation of wireless multipath channels [9], [23], and signal decomposition using overcomplete bases [8].

The notion of *variable selection* (VS) associated with sparse linear regression [24] is the cornerstone of the emerging area of compressive sampling (CS) [4], [5], [8]. VS is a combinatorially complex task closely related (but not identical) to the well-known *model order selection* problem tackled through Akaike's information [1], Bayesian information [20], and risk inflation [11] criteria. A typically convex function of the model fitting error is penalized with the ℓ_0-norm of the unknown vector which equals the number of nonzero entries, and thus accounts for model complexity (degrees of freedom). To bypass the nonconvexity of the ℓ_0-norm, VS and CS approaches replace it with convex penalty terms (e.g., the ℓ_1-norm) that capture sparsity but also lead to computationally efficient solvers.

Research on CS and VS has concentrated on batch processing, and various algorithms for sparse linear regression are available. Those include the basis pursuit and Lasso [8], [24], the Dantzig selector [6], and the ℓ_2-norm constrained ℓ_1-norm minimizer [4]. CS and VS estimators are nonlinear functions of the available observations which they process in a *batch* form using iterative algorithms. However, many sparse signals encountered in practice must be estimated based on noisy observations that become available *sequentially* in time. For such cases, batch signal estimators typically incur complexity and memory requirements that grow as time progresses. In addition, the sparse signal may vary with time both in its nonzero support, as well as in the values of its nonzero entries.

To cope with these challenges, the present paper develops adaptive algorithms for recursive estimation and tracking of (possibly time-varying) sparse signals based on noisy sequential observations adhering to a linear regression model. Adaptive estimation of sparse signals has received little consideration so far. Sequential noise-free signal recovery was considered in [18], and a sparsity-aware least mean-square (LMS) algorithm was pursued in [14]. Sparsity-aware "RLS-like" algorithms are reported in [2], while an effort to combine Kalman filtering and compressed sensing can be found in [26].

After preliminaries, the problem is stated in Section II. Section III deals with two pseudo-real time adaptive Lasso algorithms: the time-weighted Lasso (TWL), and the time- and norm-weighted Lasso (TNWL). The novel TWL replaces the ℓ_2-norm regularizing term of the RLS algorithm with the ℓ_1-norm that encourages sparsity. It turns out that if the sparse

Manuscript received September 02, 2009; accepted March 09, 2010. Date of publication March 29, 2010; date of current version June 16, 2010. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Hideaki Sakai. The work in this paper was supported by the NSF Grants CCF 0830480 and CON 0824007, and also through collaborative participation in the Communications and Networks Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon. Parts of this paper were presented at the IEEE International Conference on Acoustics, Speech and Signal Processing, Taipei, Taiwan, April 2009, and at the IEEE Workshop on Statistical Signal Processing, Cardiff, Wales, U.K., August 2009.

The authors are with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: angel129@umn.edu; bazer002@umn.edu; georgios@umn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSP.2010.2046897

signal vector is time-invariant, the TWL cannot estimate the signal support consistently while at the same time ensuring convergence comparable to RLS. To overcome this shortcoming, the TNWL introduces a weighted ℓ_1-norm regularization, where the weights are obtained from the RLS algorithm. TNWL estimates consistently the support of the sparse signal vector at converge rate identical to that of RLS.

Since TWL and TNWL estimates are not available in closed form, a sequence of convex programs has to be solved as new data are acquired, which is not desirable for real-time applications. Recent advances in sparse linear regression have showed that the coordinate descent approach provides an efficient means of solving Lasso-like problems since it is fast and numerically stable [13], [27]. This motivates the present paper's novel, low-complexity *online coordinate descent* algorithms developed in Section IV. Partial coordinate-wise updates have been considered for adaptive but sparsity-agnostic processing to lower the computational complexity of LMS and RLS [12], [28]. On top of lowering the computational burden, online coordinate descent is well motivated for sparsity-aware adaptive processing because the scalar coordinate estimates become available in closed form. Corroborating simulations are presented in Section V both for time-invariant and time-varying sparse signals. Conclusions are drawn in Section VI.

Notation. Column vectors (matrices) are denoted with lower-case (upper-case) boldface letters and sets with calligraphic letters; $(\cdot)^T$ stands for transposition, and $(\cdot)^\dagger$ for the Moore–Penrose pseudo-inverse. The pth entry of vector $\mathbf{x}$ is denoted as $x(p)$, and the (p,q)th entry of matrix $\mathbf{R}$ as $R(p,q)$. The function $\mathcal{N}(\mu,\sigma^2)$ stands for the Gaussian probability density function with mean μ and variance σ^2; the ℓ_1- and ℓ_2-norms of $\mathbf{x}\in\mathbb{R}^P$ are denoted as $\|\mathbf{x}\|_1 := \sum_{p=1}^P |x(p)|$ and $\|\mathbf{x}\|_2 := \sqrt{\sum_{p=1}^P |x(p)|^2}$, respectively.

II. Preliminaries and Problem Statement

Consider a vector $\mathbf{x}_o\in\mathbb{R}^P$ which is sparse, meaning that only a few of its entries $x_o(p)$, $p=1,\ldots,P$, are nonzero. Let $\mathcal{S}_{\mathbf{x}_o} := \{p : x_o(p)\neq 0\}$ denote its support, $P_1 := |\mathcal{S}_{\mathbf{x}_o}|$ the number of non-zero entries, and $P_0 := P - P_1$. Sparsity amounts to having $P_1 \ll P_0$. Suppose that such a sparse vector is to be estimated sequentially in time from scalar observations obeying the linear regression model

$$y_n := \mathbf{h}_n^T\mathbf{x}_o + v_n, \quad n=1,\ldots,N \tag{1}$$

where $\mathbf{h}_n\in\mathbb{R}^P$ is the regression vector at time n, and the additive noise v_n is assumed uncorrelated with $\mathbf{h}_n$, white, with mean zero, and variance σ^2. The goal of this paper is to develop sequential and adaptive estimators of $\mathbf{x}_o$ that is *a priori* known to be sparse, and perhaps slowly varying with n.

The least-squares (LS) criterion is the "workhorse" for linear regression analysis [19, p. 658]. If $\mathbf{y}_N := [y_1,\ldots,y_N]^T$ and $\mathbf{H}_N := [\mathbf{h}_1,\ldots,\mathbf{h}_N]^T$, the LS estimator of $\mathbf{x}_o$ at time N solves the minimization problem

$$\hat{\mathbf{x}}_N^{\mathrm{LS}} := \arg\min_{\mathbf{x}\in\mathbb{R}^P} \|\mathbf{y}_N - \mathbf{H}_N\mathbf{x}\|_2^2. \tag{2}$$

If $N<P$ or $\mathbf{H}_N$ is not full column rank, the problem in (2) does not admit a unique solution. For such cases, minimizing also the ℓ_2-norm of $\mathbf{x}$ renders the LS solver unique, and expressible as $\hat{\mathbf{x}}_N^{\mathrm{LS}} = \mathbf{H}_N^\dagger\mathbf{y}_N$, where † denotes matrix pseudo-inverse defined as in e.g., [15, p. 275].

In the sequential context considered herein, LS faces three challenges: i) increasing memory requirements for storing $\mathbf{y}_N$ and $\mathbf{H}_N$ as N grows large; ii) complexity of order $\mathcal{O}(P^3)$ per time instant n to perform the inversion in $\mathbf{H}_N^\dagger$; and iii) lack of capability to track possible variations of $\mathbf{x}_o$ with n.

These challenges are met by the recursive least-squares (RLS) estimator obtained as [19, Ch. 12]

$$\hat{\mathbf{x}}_N^{\mathrm{RLS}} := \arg\min_{\mathbf{x}\in\mathbb{R}^P}\sum_{n=1}^N \beta_{N,n}\left(y_n-\mathbf{h}_n^T\mathbf{x}\right)^2,\ N=1,2,\ldots \tag{3}$$

where the so-called "forgetting factor" $\beta_{N,n}$ describes one of the following data windowing choices:

(w1) *Infinite window* with $\beta_{N,n}=1$. This choice is adopted for time-invariant signals and, with proper initialization renders RLS equivalent to LS at complexity $\mathcal{O}(P^2)$ per datum.

(w2) *Exponentially decaying window* with $\beta_{N,n}=\beta^{N-n}$, and $0\ll\beta<1$. With this choice, RLS downweighs old samples, and can track time-varying signals.

(w3) *Finite window* with $\beta_{N,n}=1$ if $N-n\le M-1$ and $\beta_{N,n}=0$ otherwise. Here, only the most recent M samples are utilized to form $\hat{\mathbf{x}}_N^{\mathrm{RLS}}$ while the rest are discarded.

The RLS estimator in (3) can be expressed recursively in terms of $\hat{\mathbf{x}}_{N-1}^{\mathrm{RLS}}$. Supposing that $\{\mathbf{h}_n\}_{n=1}^P$ are linearly independent, setting $\beta_{N,n}=1$, and initializing this recursion with the LS solution for $N-1=P$, that is $\hat{\mathbf{x}}_{N-1}^{\mathrm{RLS}}=\hat{\mathbf{x}}_P^{\mathrm{LS}}$, the RLS coincides with the LS for successive instants $N>P$, provided that $\mathbf{x}_o$ remains invariant [19, p. 740].

For $N<P$ or when $\{\mathbf{h}_n\}_{n=1}^P$ are linearly dependent, the RLS estimator can be regularized by augmenting the LS cost with a scaled ℓ_2-norm of $\mathbf{x}$ [19, p. 739]. Specifically, the regularized RLS is

$$\hat{\mathbf{x}}_N^{\mathrm{RLS}} := \arg\min_{\mathbf{x}\in\mathbb{R}^P}\left[\sum_{n=1}^N \beta_{N,n}\left(y_n-\mathbf{h}_n^T\mathbf{x}\right)^2+\gamma_N\|\mathbf{x}\|_2^2\right],\ N=1,2,\ldots \tag{4}$$

where $\gamma_N>0$ is a pre-selected decreasing function of N that depends on the selected window and its effect vanishes for large N. Clearly, for $\gamma_N=0$ the regularized RLS in (4) reduces to the ordinary one in (3), but both do *not* exploit the sparsity present in $\mathbf{x}_o$.

Sparse linear regression is a topic of intense research in the last decade and Lasso is one of the most widely applied sparsity-aware estimators [5], [24]. The Lasso estimator is given by

$$\hat{\mathbf{x}}_N^{\mathrm{Lasso}} := \arg\min_{\mathbf{x}\in\mathbb{R}^P}\left[\frac{1}{2}\|\mathbf{y}_N-\mathbf{H}_N\mathbf{x}\|_2^2+\lambda\|\mathbf{x}\|_1\right]. \tag{5}$$

Thanks to the scaled ℓ_1-norm, the cost encourages sparse solutions [24]: the larger the chosen λ is, the more components are shrunk to zero. Interestingly, the Lasso performs well in sparse problems also when $N < P$, and the convex ℓ_1-norm regularization which can afford efficient solvers when optimizing (5) given batch data, performs similarly to its non-convex ℓ_0-norm counterpart [4]. The question that arises is how the ℓ_1-norm regularization can be effectively utilized in adaptive signal processing.

Specifically, given $\{y_n, \mathbf{h}_n\}_{n=1}^N$, we wish to develop recursive schemes to estimate the sparse signal of interest with: i) minimal memory requirements; ii) tracking capability; and iii) limited complexity.

III. Adaptive Pseudo-Real Time Lasso

Motivated by (3), a time-weighted Lasso (TWL) approach emerges naturally to endow the batch Lasso in (5) with ability to handle sequential processing. Specifically, the proposed TWL estimator is

$$\hat{\mathbf{x}}_N^{\text{TWL}} := \arg \min_{\mathbf{x} \in \mathbb{R}^P} J_N^{\text{TWL}}(\mathbf{x}), \quad N = 1, 2, \dots \tag{6}$$

where $J_N^{\text{TWL}}(\mathbf{x}) := 1/2 \sum_{n=1}^N \beta_{N,n} (y_n - \mathbf{h}_n^T \mathbf{x})^2 + \lambda_N \|\mathbf{x}\|_1$. In addition to windowing, note that λ_N is now allowed to vary with N.

Neglecting constant terms, the cost function in (6) can be re-written as

$$J_N^{\text{TWL}}(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{R}_N \mathbf{x} - \mathbf{x}^T \mathbf{r}_N + \lambda_N \|\mathbf{x}\|_1 \tag{7}$$

where

$$\mathbf{r}_N := \sum_{n=1}^N \beta_{N,n} y_n \mathbf{h}_n, \quad \mathbf{R}_N := \sum_{n=1}^N \beta_{N,n} \mathbf{h}_n \mathbf{h}_n^T. \tag{8}$$

Due to data windowing, $\mathbf{r}_N$ and $\mathbf{R}_N$ can be updated recursively as [cf. (w1)–(w3)]

$$\text{(w1)}: \ \mathbf{r}_N = \mathbf{r}_{N-1} + y_N \mathbf{h}_N, \quad \mathbf{R}_N = \mathbf{R}_{N-1} + \mathbf{h}_N \mathbf{h}_N^T \tag{9a}$$

$$\text{(w2)}: \ \mathbf{r}_N = \beta \mathbf{r}_{N-1} + y_N \mathbf{h}_N, \quad \mathbf{R}_N = \beta \mathbf{R}_{N-1} + \mathbf{h}_N \mathbf{h}_N^T \tag{9b}$$

$$\begin{aligned} \text{(w3)}: \ &\mathbf{r}_N = \mathbf{r}_{N-1} + y_N \mathbf{h}_N - y_{N-M} \mathbf{h}_{N-M}, \\ &\mathbf{R}_N = \mathbf{R}_{N-1} + \mathbf{h}_N \mathbf{h}_N^T - \mathbf{h}_{N-M} \mathbf{h}_{N-M}^T. \end{aligned} \tag{9c}$$

Relative to the batch Lasso in (5), any of the TWL updates in (9) offers memory savings. Clearly, choices (w2) and (w3) allow also the signal of interest to vary slowly with time. With respect to RLS in (4), the TWL estimator inherits the properties brought by the ℓ_1-norm, namely sparsity awareness and ability to deal with under-determined systems $(N < P)$. Summarizing, the attractive features of TWL are as follows:

i) reduced memory requirements with respect to batch Lasso;

ii) improved performance relative to RLS when $\mathbf{x}_o$ is sparse and time-invariant;

iii) enhanced tracking capability when $\mathbf{x}_o$ is sparse and time-varying, relative to batch Lasso and RLS for windows of size less than the dimension P of $\mathbf{x}_o$.

Despite these attractive features, the main limitation of TWL is that a convex program has to be solved per time N. While the RLS cost is differentiable, and thus amenable to closed-form minimization, $J_N^{\text{TWL}}(\mathbf{x})$ is not. However, initializing the convex program at time N with the solution $\hat{\mathbf{x}}_{N-1}^{\text{TWL}}$ at time $N-1$ provides a "warm start-up," which speeds up convergence to the optimum $\hat{\mathbf{x}}_N^{\text{TWL}}$. For these reasons, TWL is a "pseudo-real time" algorithm. Low-complexity real-time algorithms will be developed in Section IV. But for now, it is worth checking TWL for consistency.

A. (In)Consistency of the TWL Estimator

Since the non-zero support of $\mathbf{x}_o$ is unknown, and sparse vector estimators are nonlinear functions of the data not expressible in closed form, performance analysis is distinct from and far more challenging than that of LS estimators. Consider for simplicity that $\mathbf{x}_o$ is time-invariant for which (w1) is prudent to adopt, and suppose that the regressors and noise satisfy these regularity (ergodicity) conditions:

(r1) $\lim_{N \to \infty} 1/N \sum_{n=1}^N \mathbf{h}_n \mathbf{h}_n^T = \mathbf{R}_\infty$ with probability (w.p.) 1, with $\mathbf{R}_\infty$ positive definite;

(r2) $\lim_{N \to \infty} 1/N \sum_{n=1}^N v_n \mathbf{h}_n = \mathbf{r}_\infty^{vh}$ w.p. 1.

If the noise v_n and the regressors $\{\mathbf{h}_n\}$ are mixing, which is the case for most stationary processes with vanishing memory in practice, then (r1) and (r2) are readily met. Since v_n in (1) is zero mean and uncorrelated with $\mathbf{h}_n$, the cross-covariance in (r2) vanishes. With $\mathbf{r}_\infty^{vh} = \mathbf{0}$, it follows readily from (1) that $\mathbf{r}_\infty := \lim_{N \to \infty} N^{-1} \sum_{n=1}^N y_n \mathbf{h}_n = \mathbf{R}_\infty \mathbf{x}_o$ w.p. 1. Upon dividing both sides of (7) by N and taking limits, (r1) and (r2) then imply that $\lim_{N \to \infty} (1/N) J_N^{\text{TWL}}(\mathbf{x}) = (1/2) \mathbf{x}^T \mathbf{R}_\infty \mathbf{x} - \mathbf{x}^T \mathbf{r}_\infty := J_\infty(\mathbf{x})$ w.p. 1, if λ_N is chosen to grow slower than N. In this case, as $N \to \infty$ it holds that $\hat{\mathbf{x}}_N^{\text{TWL}} = \arg\min_{\mathbf{x}} J_N^{\text{TWL}}(\mathbf{x}) \to \arg\min_{\mathbf{x}} J_\infty(\mathbf{x}) := \mathbf{R}_\infty^{-1} \mathbf{r}_\infty = \mathbf{x}_o$ w.p. 1. This proves the following result.

Proposition 1. *For the model in (1) with (r1), (r2), and (w1) in effect, the TWL estimator is strongly consistent, provided that* λ_N *is chosen to satisfy* $\lim_{N \to \infty} (\lambda_N / N) = 0$.

At this point it is instructive to recall that under the conditions of Proposition 1, the LS estimator $\hat{\mathbf{x}}_N^{\text{LS}} = \mathbf{R}_N^{-1} \mathbf{r}_N$ also converges w.p. 1 to $\mathbf{x}_o = \mathbf{R}_\infty^{-1} \mathbf{r}_\infty$, and is thus strongly consistent [16]. For this reason, to asses performance of TWL and differentiate it from that of LS it is pertinent to consider sufficiently large (but preferably finite) N for which the standard sparsity-agnostic LS is unable to accurately estimate the zero entries of $\mathbf{x}_o$. It is thus of interest to check whether TWL can estimate jointly the nonzero support and the nonzero entries of $\mathbf{x}_o$ consistently for sufficiently large N. To this end, suppose that the first P_1 entries of $\mathbf{x}_o$ are non-zero; i.e., $\mathcal{S}_{\mathbf{x}_o} := \{1, \dots, P_1\}$; and partition accordingly the $\mathbf{R}_\infty$ matrix as

$$\mathbf{R}_\infty = \begin{pmatrix} \mathbf{R}_{11} & \mathbf{R}_{10} \\ \mathbf{R}_{01} & \mathbf{R}_{00} \end{pmatrix}.$$

Following the definitions in [10], *support consistency* amounts to having

$$\lim_{N \to \infty} \text{Prob} \left[\mathcal{S}_{\hat{\mathbf{x}}_N} = \mathcal{S}_{\mathbf{x}_o} \right] = 1 \tag{10}$$

and $\sqrt{N}$-*estimation consistency* requires convergence in distribution $(\to_d)$, that is

$$\sqrt{N} \left(\hat{\mathbf{x}}_N^{P_1} - \mathbf{x}_o^{P_1} \right) \to_d \mathcal{N} \left(\mathbf{0}_{P_1}, \sigma^2 \mathbf{R}_{11}^{-1} \right) \tag{11}$$

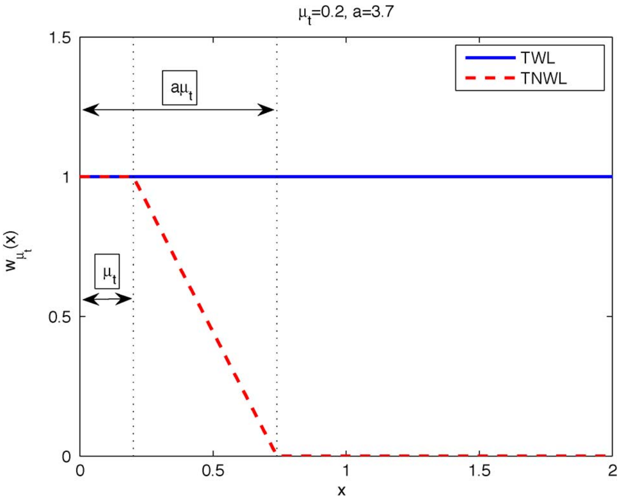

Fig. 1. Weight functions for TWL and TNWL estimators.

where $\mathbf{x}^{P_1}$ denotes the $P_1 \times 1$ vector obtained by extracting the first P_1 components of $\mathbf{x}$. Properties (10) and (11) are referred to as *oracle properties* because a sparsity-aware estimator possessing these properties is asymptotically as good as if the support $\mathcal{S}_{\mathbf{x}_o}$ was known in advance [10].

Under (w1), the TWL corresponds to a sequential version of the batch Lasso estimator. Hence, asymptotic properties of the latter derived in [16], [29] carry over to the TWL estimator introduced here.

Lemma 1. (See also [29, Prop. 1]). *For the model in (1) with (r1), (r2), and (w1) in effect, if* $\lim_{N\to\infty}(\lambda_N)/(\sqrt{N}) = \lambda_0 \geq 0$, *then* $\lim_{N\to\infty} \text{Prob}[\mathcal{S}_{\hat{\mathbf{x}}_N^{\text{TWL}}} = \mathcal{S}_{\mathbf{x}_o}] = c(\lambda_0) < 1$ *with* $c(\lambda_0)$ *denoting an increasing function of* λ_0.

In words, Lemma 1 asserts that if λ_N grows as $\sqrt{N}$, support consistency cannot be achieved. Since $c(\lambda_0)$ increases with λ_0, the hope for the TWL to satisfy the oracle properties is left for cases wherein λ_N grows faster than $\sqrt{N}$. Unfortunately, the next result discourages this.

Lemma 2. (See also [29, Lemma 3]). *For the model in (1) with (r1), (r2), and (w1) in effect, if* $\lim_{N\to\infty}(\lambda_N)/(\sqrt{N}) = \infty$ *and* $\lim_{N\to\infty}(\lambda_N/N) = 0$, *then* $\lim_{N\to\infty}(N/\lambda_N)(\hat{\mathbf{x}}_N^{\text{TWL}} - \mathbf{x}_o) = \mathbf{c}$, *where* $\mathbf{c}$ *is a non-random constant.*

Lemma 2 states that if λ_N grows faster than $\sqrt{N}$ but slower than N, the rate of convergence is (N/λ_N), that is slower than $\sqrt{N}$; hence, $\sqrt{N}(\hat{\mathbf{x}}_N^{\text{TWL}} - \mathbf{x}_o)$ diverges. Combining Lemmas 1 and 2, the following negative result holds for batch Lasso and thus for TWL.

Proposition 2. *For the model in (1) with (r1), (r2), and (w1) in effect, the TWL estimator can not achieve the oracle properties for any choice of* λ_N.

Before exploring alternatives to TWL that satisfy the oracle properties, one remark is in order.

Remark 1. If instead of the convex ℓ_1-norm the LS cost is regularized with suitably chosen non-convex functions of $\mathbf{x}$, it is possible to construct sparsity-aware estimators that asymptotically possess the oracle properties [10]. Of course, the price paid is inefficient optimization due to non-convexity. These considerations motivate searching for convex regularizing terms which result in pseudo real-time Lasso estimators satisfying the oracle properties. Such a class is developed next using the weighted ℓ_1-norm regularization, introduced in [29], [30] for the batch Lasso.

B. Time- and Norm-Weighted Lasso

Let $u(\cdot)$ denote the step function, $\{\mu_N\}$ a positive sequence dependent on the sample size, and $a > 1$ a constant tuning parameter. Based on these, define the weight function $w_{\mu_N}(\cdot) : \mathbb{R}^+ \to [0,1]$ as

$$w_{\mu_N}(x) := \frac{[a\mu_N - x]_+}{(a-1)\mu_N}\, u(x-\mu_N) + u(\mu_N - x) \qquad (12)$$

where $[\alpha]_+ := \max(\alpha, 0)$. Using w_{μ_N}, the novel time- *and* norm-weighted Lasso (TNWL) estimator weighs the ℓ_1-norm with coefficients depending on the entries of $\hat{\mathbf{x}}_N^{\text{RLS}}$; that is,

$$\hat{\mathbf{x}}_N^{\text{TNWL}} := \arg\min_{\mathbf{x}\in\mathbb{R}^P} \left[\frac{1}{2}\sum_{n=1}^{N} \beta_{N,n}\left(y_n - \mathbf{h}_n^T\mathbf{x}\right)^2 + \lambda_N \sum_{p=1}^{P} w_{\mu_N}\left(\left|\hat{x}_N^{\text{RLS}}(p)\right|\right)|x(p)| \right], \quad N = 1, 2, \ldots \qquad (13)$$

Fig. 1 shows the weight function $w_{\mu_N}(x)$ for $\mu_N = 0.2$ and $a = 3.7$. Notice that while λ_N in TWL weighs identically all summands $|x(p)|$ in the ℓ_1-norm, the TNWL estimator places higher weight to small entries, and lower weight to entries with large amplitudes. In fact, RLS estimates of size less than μ_N are penalized as in TWL, while estimates between μ_N and $a\mu_N$ are penalized in a linearly decreasing manner. Finally, RLS estimates larger than $a\mu_N$ are not penalized at all (cf. Fig. 1).

It is worth recalling at this point that albeit sparsity-agnostic, the RLS estimator is $\sqrt{N}$-estimation consistent [16], that is

$$\sqrt{N}\left(\hat{\mathbf{x}}_N^{\text{RLS}} - \mathbf{x}_o\right) \to_d \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{R}_\infty^{-1}\right). \qquad (14)$$

Based on (14), it is possible to establish the following result.

Proposition 3. (See also [30, Theorem 4]). *For the model in (1), with (r1), (r2), and (w1) in effect, if* $\lim_{N\to\infty}(\lambda_N)/(\sqrt{N}) = \infty$, $\lim_{N\to\infty}(\lambda_N/N) = 0$ *and* $\mu_N = (\lambda_N/N)$, *the TNWL estimator satisfies the oracle properties (10) and (11).*

Weighted ℓ_1-norm regularization was introduced in [7], [29], and [30] using different weight functions to effect sparsity and satisfy the oracle properties of the *batch* weighted Lasso estimator. The weight function in (12) corresponds to the local linear approximation of the smoothly clipped absolute deviation regularizer introduced by [30]. The difference here is its coupling with RLS to ensure consistency of the novel *adaptive* TNWL estimator.

Next, the implications of Propositions 2 and 3 are demonstrated through simulated tests.

C. Numerical Examples

Gaussian observations are generated according to (1) with a time-invariant $\mathbf{x}_o$, $P = 30$, $P_1 = 3$, $v_n \sim \mathcal{N}(0, \sigma^2)$, $\sigma^2 = 10^{-1}$, $\mathbf{h}_n \sim \mathcal{N}(\mathbf{0}_P, \mathbf{I}_P)$ and infinite windowing as in (w1).

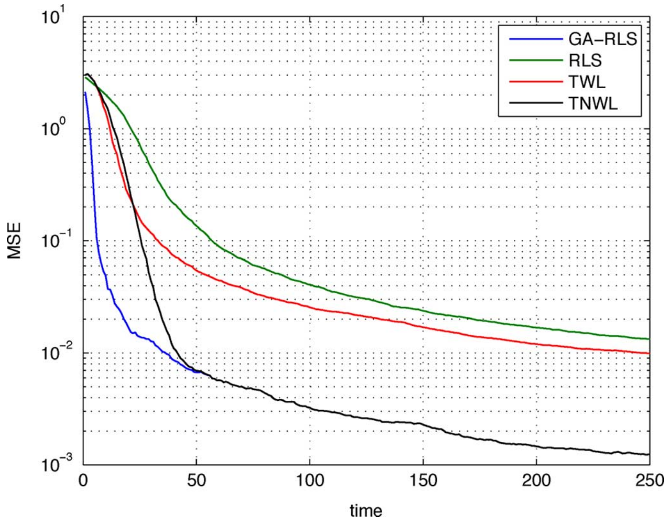

Fig. 2. MSE comparisons of pseudo real-time estimators (time-invariant $\mathbf{x}_o$).

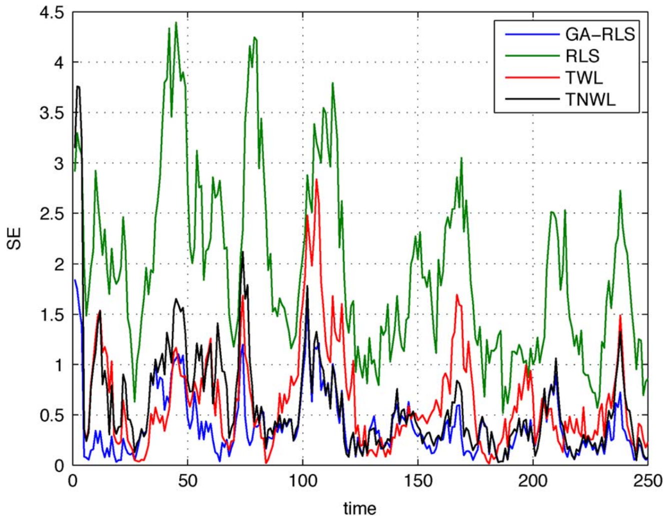

Fig. 3. Squared error comparisons of pseudo real-time estimators (time-varying $\mathbf{x}_n$ with exponentially decreasing window).

The penalty scale is set to $\lambda_N = \sqrt{2\sigma^2 N \log P}$ for the TWL (see also [29]), and $\lambda_N = \sqrt{2\sigma^2 N^{4/3} \log P}$ for TNWL with $\mu_N = (\lambda_N)/(N)$ and $a = 3.7$. The first three entries of $\mathbf{x}_o$ are chosen equal to unity, and all other entries are set to zero. Fig. 2 depicts the mean-square error (MSE), $\mathrm{E}[||\hat{\mathbf{x}}_N - \mathbf{x}_o||^2]$, across time for the TWL, TNWL, and RLS along with what is termed genie-aided (GA) RLS, which knows in advance the support and performs standard RLS to estimate the non-zero components. The convex optimization problem per time N is solved using the SeDuMi package [22] interfaced with Yalmip [17]. Observe that while TWL outperforms RLS, it is outperformed by TNWL, whose performance approaches that of the GA-RLS benchmark. Indeed, the TNWL does achieve the oracle properties in the considered simulation setting.

Next, Gaussian observations are generated according to (1) with a time-varying $\mathbf{x}_o$ (henceforth denoted as $\mathbf{x}_n$), and parameters $P = 30$, $P_1 = 3$, $v_n \sim \mathcal{N}(0, \sigma^2)$, $\sigma^2 = 10^{-1}$, and $\mathbf{h}_n \sim \mathcal{N}(\mathbf{0}_P, \mathbf{I}_P)$. A Gauss–Markov model is assumed for $\mathbf{x}_n$; that is, $x_n(p) = \alpha x_{n-1}(p) + w_n(p)$ with $x_0(p) \sim \mathcal{N}(0,1)$, $\alpha = 0.99$, and $w_n(p) \sim \mathcal{N}(0, 1-\alpha^2)$ for $p = 1,2,3$. Without loss of generality, (w2) is adopted with $\beta = 0.9$, and $\lambda_N = \sqrt{2\sigma^2 \log P}\sqrt{\sum_{n=1}^{N} \beta^{2(N-n)}}$ for both TWL and TNWL and $\mu_N = \lambda_N / (\sum_{n=1}^{N} \beta^{N-n})$. Clearly, in a time-varying setting these estimators are not expected to achieve the consistency properties established when $\mathbf{x}_o$ remains time-invariant. Fig. 3 depicts the squared error (SE) for a realization of the RLS, GA-RLS, TWL and TNWL. In the considered setting, TWL and TNWL perform similarly and both outperform RLS while approaching the performance of the GA-RLS benchmark.

Next, Gaussian observations are generated according to (1) with a time-varying $\mathbf{x}_n$, and parameters $P = 30$, $P_1 = 3$, $v_n \sim \mathcal{N}(0, \sigma^2)$, $\sigma^2 = 10^{-2}$, and $\mathbf{h}_n \sim \mathcal{N}(\mathbf{0}_P, \mathbf{I}_P)$. A Gauss–Markov model is assumed for $\mathbf{x}_n$ with $\alpha = 0.995$, and (w3) windowing is adopted. For brevity, only the regularized RLS in (4) with constant γ_N is shown along with the TWL estimators. In Fig. 4, two window sizes of length $M = 15$ and $M = 30$ are simulated. Interestingly, while RLS with $M = 30$ outperforms RLS with $M = 15$, TWL with $M = 15$ outperforms TWL with $M = 30$, and achieves the overall best performance. In fact, TWL performs well even for small window sizes, $M < P$, when the signal of interest is sparse. Thus, TWL exhibits better tracking capability than RLS which requires longer window size, and thus can track signals with slower variations.

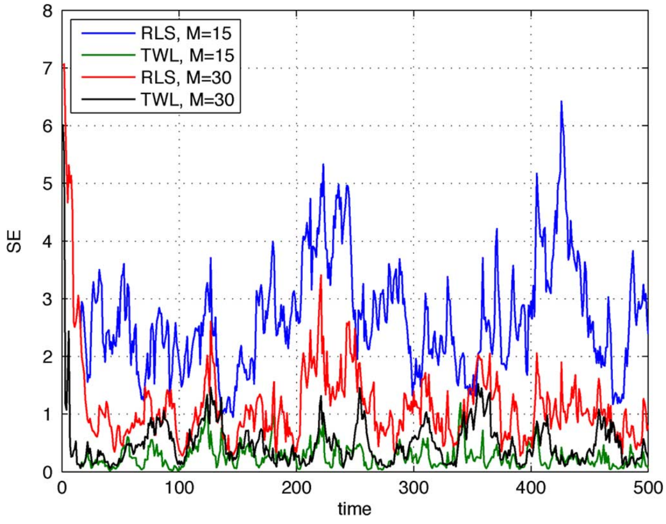

Fig. 4. Squared error comparisons of pseudo real-time estimators (time-varying $\mathbf{x}_n$ with finite window).

IV. Adaptive Real-Time Lasso

As mentioned earlier, TWL and TNWL estimators are not suitable for real-time implementation. In this section, online algorithms are developed and analyzed. The vector iterates developed in the next subsection provide online solvers of (6) and (13), admit a closed-form solution per iteration, and are proved convergent to $\mathbf{x}_o$ when the unknown vector is time-invariant. For notational brevity, the algorithms are developed for the TWL estimator but carry over to TNWL as well.

A. Online Coordinate Descent

One approach to finding the solution $\hat{\mathbf{x}}_N^{\mathrm{TWL}}$ in (6) is to run a cyclic coordinate descent (CCD) algorithm, which in its simplest form entails cyclic iterative minimization of $J_N^{\mathrm{TWL}}(\mathbf{x})$ in (7) with respect to one coordinate per iteration cycle. Let $\mathbf{x}_N^{(i-1)}$ denote the solution at time N and iteration $i-1$. The pth variable at the ith iteration is updated as

$$x_N^{(i)}(p) := \arg\min_x J_N^{\mathrm{TWL}}\left(x_N^{(i)}(1),\ldots,x_N^{(i)}(p-1),x,\right.$$
$$\left. x_N^{(i-1)}(p+1),\ldots,x_N^{(i-1)}(P)\right) \quad (15)$$

for $p=1,\ldots,P$. During the ith cycle each coordinate (here the pth) is optimized, while the pre-cursor coordinates (those with $p' < p$) are kept fixed to their values at the ith cycle, and the post-cursor coordinates (those with $p' > p$) are kept fixed to their values at the $(i-1)$st cycle.

Albeit convex, the cost $J_N^{\mathrm{TWL}}(\mathbf{x})$ is non-differentiable. Nonetheless, convergence of the CCD algorithm for Lasso-type problems follows readily using the results of [25]. In addition to affording effective initialization (with the all-zero vector), another attractive feature of CCD Lasso solvers is that each coordinate-wise minimizer is available in closed form. Recent comparative studies show that CCD exhibits computational complexity similar (if not lower) than state-of-the-art batch Lasso solvers and is numerically stable [13], [27].

The online coordinate descent (OCD) algorithm introduced next can be viewed as an adaptive counterpart of CCD Lasso, where a new datum is incorporated at each iteration; that is, the iteration index (i) in CCD is replaced in OCD by the time index N. The challenge arises because the cost function changes with N. The crux of OCD is to update only one variable per datum in the spirit of e.g., the partial least mean-squares (PLMS) algorithm [12]. Notwithstanding, PLMS is a sparsity-agnostic first-order algorithm, whereas OCD is sparsity-cognizant, it capitalizes on second-order statistics similar to RLS, and it is also provably convergent.

For notational convenience, express the time index as $N = kP+p$, where $p \in \{1,\ldots,P\}$ corresponds to the only entry of $\mathbf{x}$ to be updated at time N, and $k = \lceil N/P \rceil - 1$ indexes the number of cycles; that is, how many times the pth coordinate is updated. Let $\hat{\mathbf{x}}_{N-1}^{\mathrm{OCD}}$ denote the solution of the OCD algorithm at time $N-1$ and $\hat{x}_N^{\mathrm{OCD}}(q) = \hat{x}_{N-1}^{\mathrm{OCD}}(q)$ for $q \neq p$, which amounts to setting all but the pth coordinate at time N equal to those at time $N-1$, and selecting the pth one by minimizing $J_N^{\mathrm{TWL}}(\mathbf{x})$; that is,

$$\hat{x}_N^{\mathrm{OCD}}(p) := \arg\min_x J_N^{\mathrm{TWL}}\left(\hat{x}_{N-1}^{\mathrm{OCD}}(1),\ldots,\hat{x}_{N-1}^{\mathrm{OCD}}(p-1),x,\right.$$
$$\left.\hat{x}_{N-1}^{\mathrm{OCD}}(p+1),\ldots,\hat{x}_{N-1}^{\mathrm{OCD}}(P)\right). \quad (16)$$

In the cyclic update (16), the pre-cursor coordinates $\{\hat{x}_{N-1}^{\mathrm{OCD}}(q), q < p\}$ have been updated $k+1$ times, while the post-cursor entries $\{\hat{x}_{N-1}^{\mathrm{OCD}}(q), q > p\}$ have been updated k times. After isolating from $J_N^{\mathrm{TWL}}(\mathbf{x})$ only terms which depend on the pth coordinate that is currently optimized, recursion (16) can be rewritten as (cf. (6))

$$\hat{x}_N^{\mathrm{OCD}}(p) = \arg\min_x \left[\frac{1}{2}R_N(p,p)x^2 - r_{N,p}x + \lambda_N|x|\right] \quad (17)$$
$$r_{N,p} := r_N(p) - \sum_{q\neq p} R_N(p,q)\hat{x}_{N-1}^{\mathrm{OCD}}(q). \quad (18)$$

Algorithm 1: OCD-TWL

```
Initialize with $\hat{\mathbf{x}}_0^{\mathrm{OCD}} = \mathbf{0}, p = 1,\ldots,P$
for $k = 0,1,\ldots$ do
    for $p = 1,\ldots,P$ do
        S1. Acquire datum $y_N$, and regressor $\mathbf{h}_N$, $N = kP+p$.
        S2. Obtain $\mathbf{r}_N$ and $\mathbf{R}_N$ as in (8).
        S3. Set $\hat{x}_N^{\mathrm{OCD}}(q) = \hat{x}_{N-1}^{\mathrm{OCD}}(q)$ for all $q \neq p$.
        S4. Compute $r_{N,p}$ via (18).
        S5. Update $\hat{x}_N^{\mathrm{OCD}}(p)$ as in (19).
    end for
end for
```

Being a scalar optimization problem, it is well known that the minimization problem in (17) accepts a closed-form solution, namely [13]

$$\hat{x}_N^{\mathrm{OCD}}(p) = \frac{\mathrm{sgn}(r_{N,p})}{R_N(p,p)}[|r_{N,p}| - \lambda_N]_+. \quad (19)$$

Equation (19) amounts to a soft-thresholding operation that sets to zero inactive entries, thus facilitating convergence to sparse iterates. The OCD-TWL scheme is tabulated as Algorithm 1.

Convergence of OCD-TWL is established in Appendix A, and the main result can be summarized as follows.

Proposition 4. *For the model in (1) with (r1), (r2), and (w1) in effect, if* $\lim_{N\to\infty}(\lambda_N/N) = 0$, *it holds w.p. 1 that* $\lim_{N\to\infty}\hat{\mathbf{x}}_N^{\mathrm{OCD}} = \mathbf{x}_o$.

In words, Proposition 4 asserts that the OCD-TWL estimator is strongly consistent.

B. Online Selective Coordinate Descent

The OCD-TWL solver has low complexity but may exhibit slow convergence since each variable is updated every P observations. But since $P_1 \ll P_0$ due to sparsity, most of the time OCD-TWL resets to zero inactive entries of $\mathbf{x}_o$. On the other hand, updating zero variables cannot be skipped *a priori* since new nonzero entries may arise in time-varying scenarios. To address this dilemma, it is prudent to select which coordinate to update. A related selective approach has been pursued for batch Lasso in [27], and is extended here to the novel OCD solver.

Let $d_{\mathbf{e}_p}J_N^{\mathrm{TWL}}(\hat{\mathbf{x}}_{N-1}^{\mathrm{OSCD}})$ and $d_{-\mathbf{e}_p}J_N^{\mathrm{TWL}}(\hat{\mathbf{x}}_{N-1}^{\mathrm{OSCD}})$ denote the forward and backward directional derivatives w.r.t. $x(p)$ evaluated at $\hat{\mathbf{x}}_{N-1}^{\mathrm{OSCD}}$, which denotes the online selective coordinate descent (OSCD) estimate at time $N-1$. Define also the vectors $\mathbf{d}^+, \mathbf{d}^- \in \mathbb{R}^P$ whose pth entries are $d_{\mathbf{e}_p}J_N^{\mathrm{TWL}}(\hat{\mathbf{x}}_{N-1}^{\mathrm{OSCD}})$ and $d_{-\mathbf{e}_p}J_N^{\mathrm{TWL}}(\hat{\mathbf{x}}_{N-1}^{\mathrm{OSCD}})$, respectively. It is not difficult to verify that (see also [27])

$$\mathbf{d}^+ = \mathbf{R}_N\hat{\mathbf{x}}_{N-1}^{\mathrm{OSCD}} - \mathbf{r}_N + \lambda_N\mathbf{s}^+ \quad (20)$$
$$\mathbf{d}^- = \mathbf{r}_N - \mathbf{R}_N\hat{\mathbf{x}}_{N-1}^{\mathrm{OSCD}} + \lambda_N\mathbf{s}^- \quad (21)$$

with $\mathbf{s}^+, \mathbf{s}^- \in \mathbb{R}^P$, $s^+(p) = 1$ if $\hat{\mathbf{x}}_{N-1}^{\mathrm{OSCD}} \geq 0$ and $s^+(p) = -1$ otherwise; while $s^-(p) = 1$ if $\hat{\mathbf{x}}_{N-1}^{\mathrm{OSCD}} \leq 0$, and $s^+(p) = -1$ otherwise. After evaluating (20) and (21), the coordinate with the most negative directional derivative, either forward or backward, is updated. The OSCD-TWL scheme is summarized as Algorithm 2.

Remark 2. A subgradient-based LMS-like is developed in [2] for sparsity-aware online estimation. However, subgradient methods are first-order algorithms that posses slow convergence. For this reason, the OCD and OSCD alternatives developed here should be preferred.

Algorithm 2: OSCD-TWL

```
Initialize $\hat{\mathbf{x}}_0^{\mathrm{OSCD}} = \mathbf{0}, p = 1, \ldots, P$.
for $N = 1, 2, \ldots$ do
    S1. Acquire datum $y_N$, and regressor $\mathbf{h}_N$.
    S2. Evaluate $\mathbf{d}^+$ and $\mathbf{d}_-$ as in (20) and (21).
    S3. Select $p^* = \arg\min_p \{d^+(p), d^-(p)\}_{p=1}^P$.
    S4. Update $\hat{x}_N^{\mathrm{OSCD}}(q) = \hat{x}_{N-1}^{\mathrm{OSCD}}(q)$ for all $q \neq p^*$.
    S5. Compute $r_{N,p^*}$ via (18).
    S6. Update $\hat{x}_N^{\mathrm{OSCD}}(p^*)$ as in (19).
end for
```

C. Complexity Issues

Recall that the RLS algorithm requires $\mathcal{O}(P^2)$ algebraic operations per datum. On the other hand, the OCD-TWL Algorithm 1 requires $r_{N,p}$, whose computational burden is $\mathcal{O}(P)$ given $\mathbf{R}_N$ and $\mathbf{r}_N$. As far as OSCD is concerned, the selection step requires evaluation of $\mathbf{d}^+$ and $\mathbf{d}^-$ whose computation entails $\mathcal{O}(PP_{1,N})$ algebraic operations, where $P_{1,N}$ denotes the number of non-zero entries of $\hat{\mathbf{x}}_N^{\mathrm{OSCD}}$. However, the overall computational burden of the OCD-TWL algorithm is dominated by the update of $\mathbf{R}_N$, which requires $\mathcal{O}(P^2)$ algebraic operations. In this general case, the OCD (OSCD) can be implemented cyclically to update each coordinate per datum without affecting the overall complexity in order to speed up convergence. We summarize the online cyclic coordinate descent (OCCD) TWL as in Algorithm 3.

Algorithm 3: OCCD-TWL

```
Initialize $\hat{\mathbf{x}}_0^{\mathrm{OCCD}} = \mathbf{0}, p = 1, \ldots, P$
for $N = 1, 2, \ldots$ do
    S1. Acquire datum $y_N$, and regressor $\mathbf{h}_N$.
    S2. Obtain $\mathbf{r}_N$ and $\mathbf{R}_N$ as in (8).
    for $p = 1, \ldots, P$ do
        S3. Evaluate $r_{N,p} = r_N(p) - \sum_{q \neq p} R_N(p,q)\hat{x}_{N-1}^{\mathrm{OCCD}}(q)$
        S4. Update $\hat{x}_N^{\mathrm{OCCD}}(p)$ via (19).
    end for
end for
```

An important simplification which appears in problems such as system identification and beamforming is that regressors are sliding with time; that is $\mathbf{h}_n := [h(n), h(n-1), \ldots, h(n-P+1)]^T$ with $\mathbf{h}_0 = \mathbf{0}_P$. In this case, $\mathbf{R}_N$ updates and the RLS estimates incur complexity $\mathcal{O}(P)$ [19, p. 816], [28]. Likewise, OCD-TWL and OSCD-TWL in Algorithms 1 and 2 can be also implemented with complexity $\mathcal{O}(P)$.

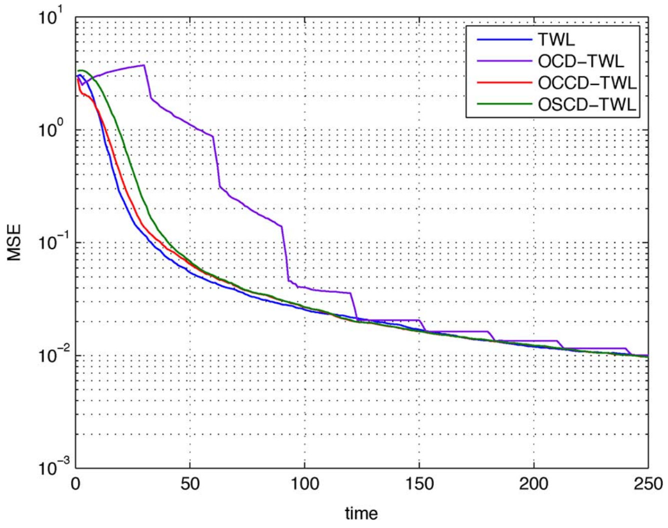

Fig. 5. MSE comparisons of online estimators (time-invariant $\mathbf{x}_o$).

Same conclusions can be drawn for online implementations of the TNWL through OCD or OSCD. In a nutshell, the novel online algorithms entail complexity analogous to RLS.

V. Simulated Tests

The online algorithms developed in Section IV are simulated here and compared with the TWL and TNWL algorithms of Section III and also with the RLS in both time-invariant as well as time-varying scenarios.

Gaussian observations are generated according to (1) with a time-invariant $\mathbf{x}_o$, $P = 30$, $P_1 = 3$, $v_n \sim \mathcal{N}(0, \sigma^2)$, $\sigma^2 = 10^{-1}$, $\mathbf{h}_n \sim \mathcal{N}(\mathbf{0}_P, \mathbf{I}_P)$, and windowing as in (w1). The first three entries of $\mathbf{x}_o$ are chosen equal to unity, and all other entries are set to zero. Fig. 5 depicts the MSE of the OCD-TWL, OCCD-TWL, OSCD-TWL, and TWL versus time. The scale is set to $\lambda_N = \sqrt{2\sigma^2 N \log P}$. As expected, the OCD-TWL converges to the TWL which requires the solution of a convex program per time N. Similar results holds for the OCCD-TWL and OSCD-TWL algorithms that also provide a means of enhancing the convergence speed.

Fig. 6 depicts the MSE of the OCD-TNWL, OCCD-TNWL, OSCD-TNWL, and TNWL versus time. The scale is set to $\lambda_N = \sqrt{2\sigma^2 N^{4/3} \log P}$ with $\mu_N = (\lambda_N / N)$ and $a = 3.7$ Also in this case the online algorithms converge to their pseudo real-time counterparts.

Next, Gaussian observations are generated according to (1) with a time-varying $\mathbf{x}_n$, and parameters $P = 30$, $P_1 = 3$, $v_n \sim \mathcal{N}(0, \sigma^2)$, $\sigma^2 = 10^{-1}$, and $\mathbf{h}_n \sim \mathcal{N}(\mathbf{0}_P, \mathbf{I}_P)$. A Gauss-Markov model is assumed for $\mathbf{x}_n$ with entries generated according to $x_n(p) = \alpha x_{n-1}(p) + w_n(p)$ with $x_0(p) \sim \mathcal{N}(0, 1)$, $\alpha = 0.99$, and $w_n(p) \sim \mathcal{N}(0, 1 - \alpha^2)$ for $p = 1, 2, 3$; (w2) is adopted with $\beta = 0.9$, and scale $\lambda_N = \sqrt{2\sigma^2 \log P}\sqrt{\sum_{n=1}^N \beta^{2(N-n)}}$. Fig. 7 shows a realization of the squared error (SE) for the OCD-TWL, OCCD-TWL, OSCD-TWL, TWL, and RLS. The OCD-TWL exhibits performance similar to that of the RLS. Indeed, updating one coordinate per observation in time-varying settings

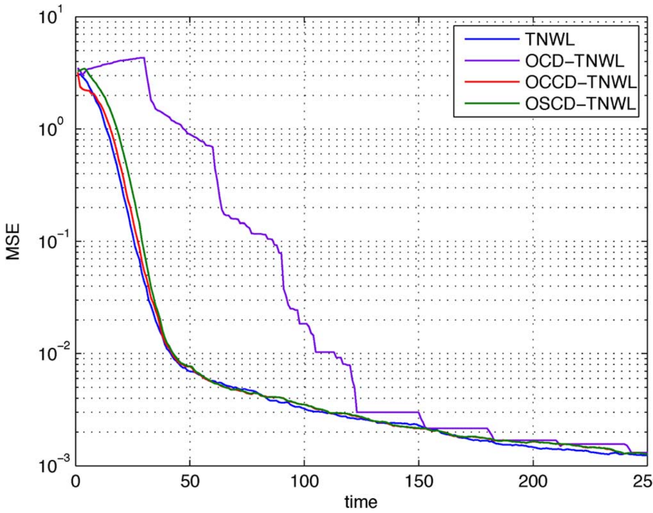

Fig. 6. MSE comparisons of online weighted-norm estimators (time-invariant $\mathbf{x}_o$).

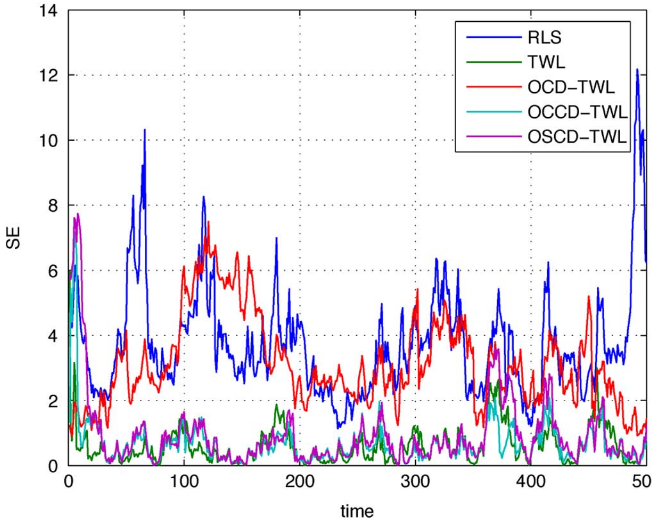

Fig. 7. Squared error comparisons of online estimators (time-varying $\mathbf{x}_n$ with exponentially decreasing window).

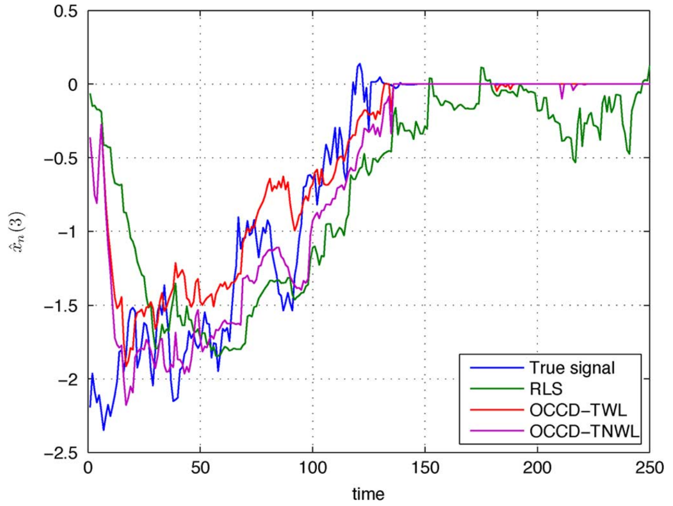

Fig. 8. Trajectory of a varying entry of the true signal vector and its estimates across time (tracking of a disappearing entry).

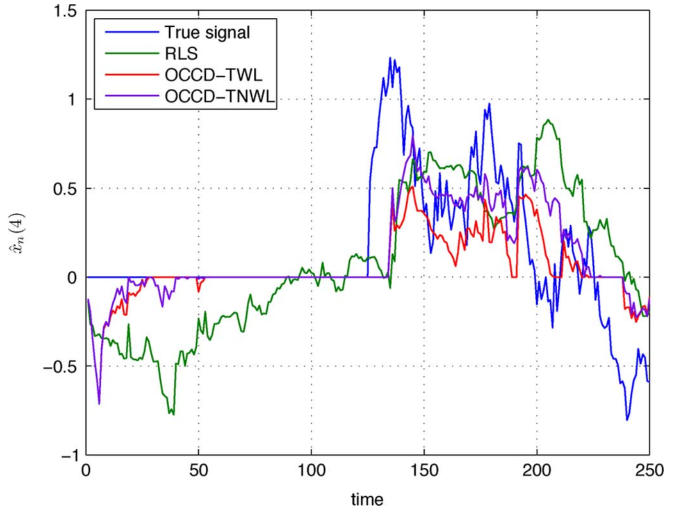

Fig. 9. Trajectory of a varying entry of the true signal vector and its estimates across time (tracking of an emerging entry).

weakens tracking capabilities [12]. However, the performance OCCD-TWL and OSCD-TWL approaches that of TWL, and both outperform the RLS algorithm.

Successively, simulated tests are performed to assess performance when the support of $\mathbf{x}_n$ changes with time. The setting in this example is identical to that of Fig. 7, except that here the support of the sparse $\mathbf{x}_n$ also undergoes step changes. Specifically, at $N = 125$ the third entry of $\mathbf{x}_n$ starts decreasing, and after $N = 150$ the same entry is set to zero. In addition, at $N = 125$ the fourth entry becomes nonzero. Figs. 8 and 9 depict, respectively, the true variations of $x_n(3)$ and $x_n(4)$ across time, along with their estimates obtained using the RLS, the OCCD-TWL, and the OCCD-TNWL with $\mu_N = (\lambda_N)/(\sum_{n=1}^{N} \beta^{N-n})$ and $a = 3.7$. Observe that the developed sparsity-aware algorithms can set to zero inactive entries while RLS estimates are not sparse and yield a nonzero value even if the true entry is zero. Moreover, after a few instants from the changing support points, the developed algorithms are able to track entries that become nonzero, and are further able to set to zero entries that disappear.

Finally, the novel algorithms are tested for identifying the sparse, finite impulse response of a discrete-time, linear system using input and noisy output data satisfying the input-output relationship

$$y_n = \sum_{p=0}^{P-1} h_p x_{n-p} + v_n = \mathbf{x}_n^T \mathbf{h}_o + v_n, \quad n = 1, \ldots, N \quad (22)$$

where $\mathbf{h}_o := [h_0, \ldots, h_{P-1}]^T$ collects the unknown impulse response coefficients, $\mathbf{x}_n := [x_n, \ldots, x_{n-P+1}]^T$ denotes the given input data (the regressor vector in (1)), and y_n the output at time n. As the system order maybe unknown, a large known upper bound P is selected. Since many entries of $\mathbf{h}_o$ maybe zero or negligible, the impulse response is sparse. In addition, nonzero entries may exhibit slow time variations, which gives rise to a time-varying impulse response $\mathbf{h}_n$. To assess performance of the introduced algorithms, a system with $P = 128$ and $P_1 = 6$ nonzero entries at unknown locations is simulated.

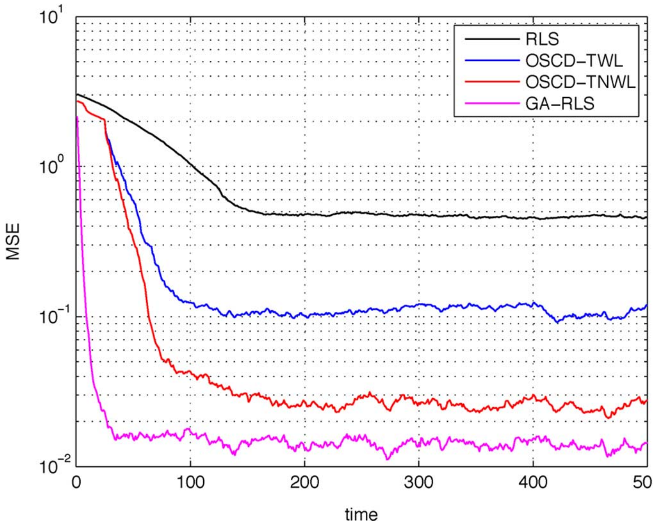

Fig. 10. MSE comparison of adaptive algorithms for estimating a sparse, linear, time-varying impulse response.

The input sequence is assumed zero-mean, white, Gaussian, with unit variance, and $v_n \sim \mathcal{N}(0, \sigma^2)$ with $\sigma^2 = 10^{-2}$. RLS, OSCD-TWL, and OSCD-TNWL are tested along with the GA-RLS for an exponential window with $\beta = 0.95$. Since the regressors here are shift-invariant, all algorithms incur computational burden that scales linearly with P. The impulse response is generated according to a first-order Gauss-Markov process with $\alpha = 0.999$. The tuning parameters of the OSCD-TWL and OSCD-TNWL have been chosen as in Fig. 8. Fig. 10 depicts the MSE (averaged over 100 realizations) across time. It is clear that both OSCD-TWL and OSCD-TNWL outperform the RLS. In particular, the gain of the OSCD-TNWL is more than one order of magnitude.

VI. Concluding Summary

Recursive algorithms were developed in this paper for estimation of (possibly time-varying) sparse signals based on observations that obey a linear regression model, and become available sequentially in time. The novel TWL and TNWL algorithms can be viewed as ℓ_1-norm regularized versions of the RLS. Simulations illustrated that TWL outperforms the sparsity-agnostic RLS scheme when estimating time-invariant and slowly-varying sparse signals. Moreover, the novel algorithms exhibit enhanced tracking capability with respect to RLS especially for short observation windows. Performance analysis revealed that TWL estimates cannot simultaneously recover the signal support and maintain convergence of RLS. This prompted the development of TNWL, which for proper selection of design parameters can achieve oracle consistency properties for time-invariant sparse signals. However, TWL and TNWL require solving a convex problem per time step, and may be less desirable for real-time applications. To overcome this limitation, low-complexity sparsity-aware online schemes were also developed. The crux of these schemes is a novel optimization algorithm that implements the basic coordinate descent iteration online. Albeit simple, the resulting OCD-TWL (OCD-TNWL) algorithm was proved convergent when the sparse signal is time invariant. At complexity comparable to OCD-TWL (OCD-TNWL) but with improved convergence speed, online selective variants choose the best coordinate to optimize and exhibit performance similar to the pseudo real-time TWL (TNWL).

Appendix A
Proof of Proposition 4

Define the vector $\boldsymbol{\chi}_k := \hat{\mathbf{x}}_N^{\mathrm{OCD}}$ for $N = kP$, that is, the one containing the iterates at the end of the kth cycle when each variable has been updated k times. The proof that $\boldsymbol{\chi}_k$ converges to $\mathbf{x}_o$ as $k \rightarrow \infty$ will proceed in five stages. In the first one, Algorithm 1 is put in the form of a noisy vector-matrix difference equation. The second and third stages prove that the corresponding discrete-time dynamical system is exponentially stable, and that the sequence $\{\boldsymbol{\chi}_k\}_{k=1}^{\infty}$ is bounded. In the fourth stage, convergence to a limit point $\boldsymbol{\chi}_\infty$ is proved. The proof concludes by showing that $\boldsymbol{\chi}_\infty = \mathbf{x}_o$.

A. Dynamical System

Let $\bar{\mathbf{R}}_k$ denote the matrix with entries $\bar{R}_k(p,q) := R_{kP+p}(p,q)/(kP + p)$, and $\bar{\mathbf{r}}_k$ the vector with entries $\bar{r}_k(p) := r_{kP+p}(p)/(kP + p)$. Conditions (r1) and (r2) guarantee that $\bar{\mathbf{R}}_k \; k \overset{\longrightarrow}{\rightarrow} \infty \; \mathbf{R}_\infty$ and $\bar{\mathbf{r}}_k \; k \overset{\longrightarrow}{\rightarrow} \infty \; \mathbf{r}_\infty$ w.p. 1. Consider the decomposition $\bar{\mathbf{R}}_k = \bar{\mathbf{D}}_k + \bar{\mathbf{L}}_k + \bar{\mathbf{U}}_k$ where $\bar{\mathbf{D}}_k$ is diagonal, and $\bar{\mathbf{L}}_k$ $(\bar{\mathbf{U}}_k)$ is strictly lower (upper) triangular. Observe that $\bar{\mathbf{L}}_k \neq \bar{\mathbf{U}}_k^T$.

Dividing the cost function of the problem in (17) by $kP + p$ yields

$$\chi_{k+1}(p) = \arg\min_{\chi} \left[\frac{1}{2} \left(\frac{R_{kP+p}(p,p)}{kP+p} \right) \chi^2 - \left(\frac{r_{kP+p,p}}{kP+p} \right) \chi + \left(\frac{\lambda}{kP+p} \right) |\chi| \right]. \tag{23}$$

The solution of this scalar minimization problem can be obtained in two steps. First, solve the differentiable linear-quadratic part of (23) using the auxiliary vector $\mathbf{z}_{k+1}$ to obtain

$$z_{k+1}(p) = \arg\min_{z} \left[\frac{1}{2} \left(\frac{R_{kP+p}(p,p)}{kP+p} \right) z^2 - \left(\frac{r_{kP+p}(p)}{kP+p} - \sum_{q<p} \frac{R_{kP+p}(p,q)}{kP+p} \chi_{k+1}(q) - \sum_{q>p} \frac{R_{kP+p}(p,q)}{kP+p} \chi_k(q) \right) z \right] \tag{24}$$

where $r_{kP+p,p}$ was expanded according to its definition in (18) and divided in two sums: the one already updated in the cycle $k+1$ $(q < p)$, and the second one updated in the kth cycle $(q > p)$. The second step to solve (23) is to pass $z_{k+1}(p)$ through the soft-threshold operator

$$\chi_{k+1}(p) = \operatorname{sgn}(z_{k+1}(p)) \left[|z_{k+1}(p)| - \frac{\lambda_{kP+p}}{kP+p} \right]_+ . \tag{25}$$

Using the decomposition of $\bar{\mathbf{R}}_k$, (24) can be rewritten as

$$z_{k+1}(p) = \arg\min_{z} \frac{1}{2} \bar{D}_k(p,p)\, z^2 - \left(\bar{r}_k(p) - \sum_{q<p} \bar{L}_k(p,q) \chi_{k+1}(q) - \sum_{q>p} \bar{U}_k(p,q) \chi_k(q) \right) z$$

whose solution is obtained (after equating the derivative to zero) as

$$\begin{aligned}\bar{D}_k(p,p)\, z_{k+1}(p) =& \bar{r}_k(p) - \sum_{q<p} \bar{L}_k(p,q)\chi_{k+1}(q) \\ & - \sum_{q>p} \bar{U}_k(p,q)\chi_k(q). \end{aligned} \tag{26}$$

Concatenating the latter with $p = 1, \ldots, P$ yields the matrix-vector difference equation

$$\bar{\mathbf{D}}_k \boldsymbol{z}_{k+1} = \bar{\mathbf{r}}_k - \bar{\mathbf{L}}_k \boldsymbol{\chi}_{k+1} - \bar{\mathbf{U}}_k \boldsymbol{\chi}_k. \tag{27}$$

The soft-thresholding operation in (25) can be accounted for by defining the error vector $\boldsymbol{\epsilon}_k := \chi_{k+1} - \mathbf{z}_{k+1}$ in which case (28) can be re-written as

$$\bar{\mathbf{D}}_k(\chi_{k+1} - \boldsymbol{\epsilon}_k) = \bar{\mathbf{r}}_k - \bar{\mathbf{L}}_k \boldsymbol{\chi}_{k+1} - \bar{\mathbf{U}}_k \boldsymbol{\chi}_k. \tag{28}$$

Assuming that there exists a $k^\star$ such that $\bar{\mathbf{D}}_k + \bar{\mathbf{L}}_k$ is invertible for each $k > k^\star$, (28) can be written as

$$\chi_{k+1} = \bar{\mathbf{G}}_k \boldsymbol{\chi}_k + (\bar{\mathbf{D}}_k + \bar{\mathbf{L}}_k)^{-1} \bar{\mathbf{r}}_k + (\bar{\mathbf{D}}_k + \bar{\mathbf{L}}_k)^{-1} \bar{\mathbf{D}}_k \boldsymbol{\epsilon}_k \tag{29}$$

with $\bar{\mathbf{G}}_k := (\bar{\mathbf{D}}_k + \bar{\mathbf{L}}_k)^{-1} \bar{\mathbf{U}}_k$. The key point to be used subsequently is that (25) guarantees that the entries of $\boldsymbol{\epsilon}_k$ are bounded by a vanishing sequence. Specifically, it holds that

$$|\epsilon_k(p)| \le \frac{\lambda_{kP+p}}{kP+p}, \quad \text{for } p = 1, \ldots, P \tag{30}$$

since the input-output variables of the soft-threshold operator $\chi = \mathrm{sgn}(z)[|z| - \lambda]_+$ obey $|\chi - z| \le \lambda$.

B. Exponential Stability

Let $\bar{\mathbf{G}}(l : k) := \prod_{i=l}^{k} \bar{\mathbf{G}}_i$ in (29) denote the product of the transition matrices $\bar{\mathbf{G}}_k$. The goal of this stage is to prove that $\|\bar{\mathbf{G}}(l : k)\| \le c\rho^{k-l+1}$, with $\rho < 1$.

The convergence of $\bar{\mathbf{R}}_k$ to $\mathbf{R}_\infty$ implies convergence of $\bar{\mathbf{G}}_k$ to $\mathbf{G}_\infty := -(\mathbf{D}_\infty + \mathbf{L}_\infty)^{-1}\mathbf{U}_\infty$, where $\mathbf{D}_\infty$, $\mathbf{L}_\infty$ and $\mathbf{U}_\infty$ are the diagonal, lower triangular, and upper triangular parts of $\mathbf{R}_\infty$, respectively. Since $\mathbf{R}_\infty$ is positive definite, the spectral radius of $\mathbf{G}_\infty$ is strictly less than one, i.e., $\varrho(G_\infty) < 1$ [15, p. 512]. Furthermore, for every $\delta > 0$ there exists a $c(\delta)$ constant w.r.t. k, such that $\|{\mathbf{G}_\infty}^k\| < c(\delta)[\varrho(\mathbf{G}_\infty) + \delta]^k$ ([15], p. 336). Then, by selecting $\delta = (1 - \varrho(\mathbf{G}_\infty))/2$, and defining $\rho_\infty := (1 + \varrho(\mathbf{G}_\infty))/2$, it holds that

$$\|{\mathbf{G}_\infty}^k\| < c\rho_\infty^k, \mathrm{with} \rho_\infty < 1. \tag{31}$$

Upon defining $\tilde{\mathbf{G}}_k := \bar{\mathbf{G}}_k - \mathbf{G}_\infty$, the following recursion is obtained

$$\begin{aligned}\bar{\mathbf{G}}(1:k) &= \bar{\mathbf{G}}_k \bar{\mathbf{G}}(1:k-1) \\ &= \mathbf{G}_\infty \bar{\mathbf{G}}(1:k-1) + \tilde{\mathbf{G}}_k \bar{\mathbf{G}}(1:k-1) \\ &= {\mathbf{G}_\infty}^k + \sum_{i=1}^{k} {\mathbf{G}_\infty}^{k-i} \tilde{\mathbf{G}}_i \bar{\mathbf{G}}(1:i-1), \bar{\mathbf{G}}(1:0) := \mathbf{I}.\end{aligned}$$

Using (31), the latter can be bounded as

$$\|\bar{\mathbf{G}}(1:k)\| \le c\rho_\infty^k + c\sum_{i=1}^{k} \rho_\infty^{k-i} \|\tilde{\mathbf{G}}_i\| \|\bar{\mathbf{G}}(1:i-1)\|$$

which after multiplying both sides by ρ_∞^{-k} yields

$$\rho_\infty^{-k} \|\bar{\mathbf{G}}(1:k)\| \le c + \sum_{i=1}^{k} c\rho_\infty^{-1} \|\tilde{\mathbf{G}}_i\| \|\bar{\mathbf{G}}(1:i-1)\| \rho_\infty^{-(i-1)} \tag{32}$$

and allows one to apply the discrete Bellman-Gronwall lemma (see, e.g., [21, p. 315]).

Lemma 5 (Bellman–Gronwall). *If* $c, \xi_k, h_k \ge 0 \; \forall k$ *satisfy the recursive inequality*

$$\xi_k \le c + \sum_{i=1}^{k} h_{i-1}\xi_{i-1} \tag{33}$$

then ξ_k *obeys the non-recursive inequality*

$$\xi_k \le c \prod_{i=1}^{k} (1 + h_{i-1}). \tag{34}$$

For $\xi_k = \rho_\infty^{-k} \|\bar{\mathbf{G}}(1:k)\|$ and $h_k = c\rho_\infty^{-1} \|\tilde{\mathbf{G}}_{k+1}\|$, (32) takes the form of (33), so that (34) holds and (after multiplying both sides by ρ_∞^k) results in

$$\|\bar{\mathbf{G}}(1:k)\| \le \rho_\infty^k c \prod_{i=1}^{k} \left(1 + c\rho_\infty^{-1} \|\tilde{\mathbf{G}}_i\|\right) = c \prod_{i=1}^{k} \left(\rho_\infty + c\|\tilde{\mathbf{G}}_i\|\right). \tag{35}$$

Raising both sides of (35) to the power of $1/k$ and applying the geometric-arithmetic mean inequality, it follows that

$$\|\bar{\mathbf{G}}(1:k)\|^{1/k} \le c^{1/k} \frac{1}{k} \sum_{i=1}^{k} (\rho_\infty + c\|\tilde{\mathbf{G}}_i\|)$$

which is readily rewritten as

$$\|\bar{\mathbf{G}}(1:k)\| \le c \left(\rho_\infty + c\frac{1}{k} \sum_{i=1}^{k} \|\tilde{\mathbf{G}}_i\|\right)^k.$$

Since $\bar{\mathbf{G}}_k \; k \overset{\longrightarrow}{\rightarrow} \infty \; \mathbf{G}_\infty$ and $\|\tilde{\mathbf{G}}_k\| \; k \overset{\longrightarrow}{\rightarrow} \infty \; \mathbf{0}$ w.p. 1, for every $\delta > 0$ there exists an integer k_0 such that if $k \ge k_0$, then $1/k \sum_{i=1}^{k} \|\tilde{\mathbf{G}}_i\| \le \delta$ w.p 1. Thus, if δ is selected as $(1 - \rho_\infty)/(2c)$, and ρ as $(1 + \rho_\infty)/2$, the following bound is obtained

$$\|\bar{\mathbf{G}}(1:k)\| \le c\rho^k, \quad \rho < 1, \; k \ge k_0, \mathrm{w.p.}\ 1.$$

It is clear, by inspection, that the proof so far carries over even if the product of transition matrices starts at $l > 1$; that is

$$\|\bar{\mathbf{G}}(l:k)\| \le c\rho^{k-l+1}, \; \rho < 1, \; k \ge k_0, \mathrm{w.p.}\ 1. \tag{36}$$

Certainly, c, ρ, and k_0 do not depend on l. However, k_0 does depend on the realization of the random sequence $\bar{\mathbf{G}}_k$, and its existence and finiteness are guaranteed w.p. 1.

C. Boundedness

Define $\mathbf{v}_k := (\bar{\mathbf{D}}_k + \bar{\mathbf{L}}_k)^{-1}(\bar{\mathbf{r}}_k + \bar{\mathbf{D}}_k \boldsymbol{\epsilon}_k)$, and rewrite (29) as

$$\chi_{k+1} = \bar{\mathbf{G}}_k \boldsymbol{\chi}_k + \mathbf{v}_k.$$

Using back substitution, χ_{k+1} can then be expressed as

$$\chi_{k+1} = \bar{\mathbf{G}}(k_0 : k)\boldsymbol{\chi}_{k_0} + \sum_{i=1}^{k-k_0} \bar{\mathbf{G}}(k-i+1 : k)\mathbf{v}_{k-i} + \mathbf{v}_k.$$

Since $\bar{\mathbf{D}}_k$, $\bar{\mathbf{L}}_k$, $\bar{\mathbf{r}}_k$, and $\boldsymbol{\epsilon}_k$ converge, the random sequence $\mathbf{v}_k$ converges too w.p. 1; hence, it can be stochastically bounded by a random variable v; that is, $\|\mathbf{v}_k\| < v$, $\forall k$, w.p. 1. This, combined with the exponential stability ensured by (36), guarantees that the realizations of the random sequence $\boldsymbol{\chi}_k$ are (stochastically) bounded; thus

$$\begin{aligned}\|\chi_{k+1}\| \leq c\rho^{-k+k_0-1}\|\boldsymbol{\chi}_{k_0}\| + \sum_{i=1}^{k-k_0} c\rho^{-i}v + \\ v \leq c\|\boldsymbol{\chi}_{k_0}\| + cv\left(\frac{1}{1-\rho}+1\right), \text{ w.p. 1.} \quad (37)\end{aligned}$$

D. Convergence

Define the error $\tilde{\mathbf{D}}_k := \bar{\mathbf{D}}_k - \mathbf{D}_\infty$, and similarly $\tilde{\mathbf{L}}_k := \bar{\mathbf{L}}_k - \mathbf{L}_\infty$, $\tilde{\mathbf{U}}_k := \bar{\mathbf{U}}_k - \mathbf{U}_\infty$, and $\tilde{\mathbf{r}}_k := \bar{\mathbf{r}}_k - \mathbf{r}_\infty$. Using these new variables, (28) can be rewritten in error form as

$$\begin{aligned}(\mathbf{D}_\infty + \tilde{\mathbf{D}}_k)(\chi_{k+1} - \boldsymbol{\epsilon}_k) = (\mathbf{r}_\infty + \tilde{\mathbf{r}}_k) - (\mathbf{L}_\infty + \tilde{\mathbf{L}}_k)\chi_{k+1} \\ -(\mathbf{U}_\infty + \tilde{\mathbf{U}}_k)\boldsymbol{\chi}_k \quad (38)\end{aligned}$$

and, after regrouping terms, as

$$\begin{aligned}\chi_{k+1} = -(\mathbf{D}_\infty + \mathbf{L}_\infty)^{-1}\mathbf{U}_\infty\boldsymbol{\chi}_k + (\mathbf{D}_\infty + \mathbf{L}_\infty)^{-1} \\ \times (\mathbf{r}_\infty + \tilde{\mathbf{r}}_k - (\tilde{\mathbf{D}}_k + \tilde{\mathbf{L}}_k)\chi_{k+1} + (\mathbf{D}_\infty + \tilde{\mathbf{D}}_k)\boldsymbol{\epsilon}_k - \tilde{\mathbf{U}}_k\boldsymbol{\chi}_k). \\ (39)\end{aligned}$$

Equation (39) describes an exponentially stable linear time-invariant system with transition matrix-$(\mathbf{D}_\infty + \mathbf{L}_\infty)^{-1}\mathbf{U}_\infty$, and input $\mathbf{u}_k := (\mathbf{D}_\infty + \mathbf{L}_\infty)^{-1}[\mathbf{r}_\infty + \tilde{\mathbf{r}}_k - (\tilde{\mathbf{D}}_k + \tilde{\mathbf{L}}_k)\chi_{k+1} + (\mathbf{D}_\infty + \tilde{\mathbf{D}}_k)\boldsymbol{\epsilon}_k - \tilde{\mathbf{U}}_k\boldsymbol{\chi}_k]$. The input can be divided into its limiting point $\mathbf{u}_\infty := (\mathbf{D}_\infty + \mathbf{L}_\infty)^{-1}\mathbf{r}_\infty$, and the error $\tilde{\mathbf{u}}_k := (\mathbf{D}_\infty + \mathbf{L}_\infty)^{-1}[\tilde{\mathbf{r}}_k - (\tilde{\mathbf{D}}_k + \tilde{\mathbf{L}}_k)\chi_{k+1} + (\mathbf{D}_\infty + \tilde{\mathbf{D}}_k)\boldsymbol{\epsilon}_k - \tilde{\mathbf{U}}_k\boldsymbol{\chi}_k]$. As $k \to \infty$, the vector $\tilde{\mathbf{u}}_k$ goes to zero almost surely because the sequence $\boldsymbol{\chi}_k$ is bounded, and the error $\tilde{\mathbf{r}}_k$, $\tilde{\mathbf{D}}_k$, $\tilde{\mathbf{L}}_k$ and $\tilde{\mathbf{U}}_k$ as well as $\boldsymbol{\epsilon}_k$, all go to zero w.p. 1.

With this notation and recalling the definition $\mathbf{G}_\infty := -(\mathbf{D}_\infty + \mathbf{L}_\infty)^{-1}\mathbf{U}_\infty$, (39) is rewritten as

$$\chi_{k+1} = \mathbf{G}_\infty\boldsymbol{\chi}_k + \mathbf{u}_\infty + \tilde{\mathbf{u}}_k$$

and back-substituting again the new expression for χ_{k+1} yields

$$\chi_{k+1} = \mathbf{G}_\infty{}^k\boldsymbol{\chi}_1 + \sum_{i=1}^{k} \mathbf{G}_\infty{}^{k-i}\mathbf{u}_\infty + \sum_{i=1}^{k} \mathbf{G}_\infty{}^{k-i}\tilde{\mathbf{u}}_i. \quad (40)$$

Convergence of this recursion will be established by showing that the first and third terms in the right-hand side vanish as $k \to \infty$, while the surviving one corresponds to a stable geometric series. Given that $\exists c > 0, \rho_\infty < 1$ such that $\forall n\ \|\mathbf{G}_\infty^n\| \leq c\rho_\infty^n$, convergence of the first term to zero follows readily from (31). The third term represents the limiting output value of a multiple input-multiple output stable linear time-invariant system with vanishing input; that is $\lim_{i\to\infty}\tilde{\mathbf{u}}_i = 0$. As $\forall k\ \|\mathbf{G}_\infty^k\| \leq c\rho_\infty^k$, (31) implies that it is possible to bound the sum under consideration as

$$\left\|\sum_{i=1}^{k} \mathbf{G}_\infty{}^{k-i}\tilde{\mathbf{u}}_i\right\| \leq c\sum_{i=1}^{k} \rho_\infty^{k-i}\|\tilde{\mathbf{u}}_i\|. \quad (41)$$

Since $\lim_{i\to\infty}\tilde{\mathbf{u}}_i = 0$, it holds by the definition of the limit that for any $\epsilon > 0\ \exists N \in \mathbb{N}$ so that $\|\tilde{\mathbf{u}}_i\| \leq \epsilon$, $\forall i \geq N$. Using the latter along with (41), it follows that for $k \geq N$

$$\begin{aligned}\left\|\sum_{i=1}^{k} \mathbf{G}_\infty{}^{k-i}\tilde{\mathbf{u}}_i\right\| &\leq c\sum_{i=1}^{N-1} \rho_\infty^{k-i}\|\tilde{\mathbf{u}}_i\| + c\epsilon\sum_{i=N}^{k} \rho_\infty^{k-i} \\ &= c\rho_\infty^{k-N}\sum_{i=1}^{N-1} \rho_\infty^{N-i}\|\tilde{\mathbf{u}}_i\| + c\epsilon\sum_{i=N}^{k} \rho_\infty^{k-i}. \\ & \quad (42)\end{aligned}$$

Because $\sum_{i=1}^{N-1}\rho_\infty^{N-i}\|\tilde{\mathbf{u}}_i\|$ does not depend on k, the limit of the first summand in (42) goes to zero; hence,

$$\lim_{k\to\infty}\left\|\sum_{i=1}^{k} \mathbf{G}_\infty{}^{k-i}\tilde{\mathbf{u}}_i\right\| \leq c\epsilon/(1-\rho_\infty).$$

The last inequality holds $\forall\epsilon > 0$; thus,

$$\lim_{k\to\infty}\left\|\sum_{i=1}^{k} \mathbf{G}_\infty{}^{k-i}\tilde{\mathbf{u}}_i\right\| = 0$$

which establishes convergence to zero of the third sum in the right-hand side of (40).

E. Limit Point

Once convergence is established, it is possible to take the limit as $k \to \infty$ in (38) to obtain

$$\mathbf{D}_\infty(\boldsymbol{\chi}_\infty - \boldsymbol{\epsilon}_\infty) = \mathbf{r}_\infty - \mathbf{L}_\infty\boldsymbol{\chi}_\infty - \mathbf{U}_\infty\boldsymbol{\chi}_\infty, \text{ w.p. 1.} \quad (43)$$

Recalling that $\|\boldsymbol{\epsilon}_\infty\| \leq \lim_{N\to\infty}(\lambda_N/N) = 0$, (43) reduces to

$$(\mathbf{D}_\infty + \mathbf{L}_\infty + \mathbf{U}_\infty)\boldsymbol{\chi}_\infty = \mathbf{r}_\infty, \text{ w.p. 1} \quad (44)$$

and since $\mathbf{D}_\infty + \mathbf{L}_\infty + \mathbf{U}_\infty = \mathbf{R}_\infty$, it holds that

$$\boldsymbol{\chi}_\infty = (\mathbf{R}_\infty)^{-1}\mathbf{r}_\infty = \mathbf{x}_o, \text{ w.p. 1} \quad (45)$$

which concludes the proof.

References

[1] H. Akaike, "A new look at the statistical model identification," *IEEE Trans. Autom. Control*, vol. 19, no. 6, pp. 716–723, 1974.

[2] D. Angelosante and G. B. Giannakis, "RLS-weighted Lasso for adaptive estimation of sparse signals," presented at the IEEE Int. Conf. Acoust., Speech, Signal Process., Taipei, Taiwan, Apr. 2009.

[3] J.-A. Bazerque and G. B. Giannakis, "Distributed spectrum sensing for cognitive radios by exploiting sparsity," presented at the 42nd Asilomar Conf., Pacific Grove, CA, Oct. 26–29, 2008.

[4] E. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Commun. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.

[5] E. Candès and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[6] E. Candès and T. Tao, "The Dantzig selector: Statistical estimation when p is much larger than n," *Ann. Statist.*, vol. 35, no. 6, pp. 2313–2351, 2007.

[7] E. Candès, M. Wakin, and S. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *J. Fourier Anal. Appl.*, vol. 14, pp. 877–905, 2007.

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

[9] S. F. Cotter and B. D. Rao, "Sparse channel estimation via matching pursuit with application to equalization," *IEEE Trans. Commun.*, vol. 50, no. 3, pp. 374–377, Mar. 2002.

[10] J. Fan and R. Li, "Variable selection via nonconcave penalized likelihood and its oracle properties," *J. Amer. Statist. Assoc.*, vol. 96, pp. 1348–1360, 2001.

[11] D. P. Foster and E. I. George, "The risk inflation criterion for multiple regression," *Ann. Statist.*, vol. 22, no. 4, pp. 1947–1975, 1994.

[12] M. Godavarti and A. O. Hero, III, "Partial update LMS algorithms," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2382–2399, Jul. 2005.

[13] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani, "Pathwise coordinate optimization," *Ann. Appl. Statist.*, vol. 1, pp. 302–332, Dec. 2007.

[14] Y. Gu, Y. Chen, and A. O. Hero, "Sparse LMS for system identification," presented at the IEEE Int. Conf. Acoustics, Speech, Signal Process., Taipei, Taiwan, Apr. 2009.

[15] G. Golub and C. Loan*, Matrix Computations*. Baltimore, MD: The Johns Hopkins Univ. Press, 1996.

[16] K. Knight and W. J. Fu, "Asymptotics for Lasso-type estimators," *Ann. Statist.*, vol. 28, pp. 1356–1378, 2000.

[17] J. Löfberg, "Yalmip: Software for solving convex (and nonconvex) optimization problems," presented at the Amer. Control Conf., Minneapolis, MN, Jun. 2006.

[18] D. M. Malioutov, S. Sanghavi, and A. S. Willsky, "Compressed sensing with sequential observations," presented at the Int. Conf. Acoustics, Speech, Signal Processing (ICASSP), Las Vegas, NV, Apr. 2008.

[19] A. H. Sayed*, Fundamentals of Adaptive Filtering*. New York: Wiley, 2003.

[20] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, no. 2, pp. 461–464, 1978.

[21] V. Solo and X. Kong*, Adaptive Signal Processing Algorithms: Stability and Performance*. Upper Saddle River, NJ: Prentice-Hall, 1995.

[22] J. Sturm, "Using Sedumi 1.02, a Matlab toolbox for optimization over symmetric cones," *Optimiz. Methods Softw.*, vol. 11, no. 12, pp. 625–653, 1999.

[23] G. Taubock and F. Hlawatsch, "A compressed sensing technique for OFDM channel estimation in mobile environments: Exploiting channel sparsity for reducing pilots," presented at the Int. Conf. Acoust., Speech, Signal Process., Las Vegas, NV, Apr. 2008.

[24] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *J. Roy. Statist. Soc.*, vol. 58, no. 1, pp. 267–288, 1996.

[25] P. Tseng, "Convergence of block coordinate descent method for nondifferentiable minimization," *J. Optimiz. Theory Appl.*, vol. 109, pp. 475–494, Jun. 2001.

[26] N. Vaswani, "Kalman filtered compressed sensing," presented at the Int. Conf. Image Process., San Diego, CA, Oct. 2008.

[27] T. T. Wu and K. Lange, "Coordinate descent algorithms for Lasso penalized regression," *Ann. Appl. Statist.*, vol. 2, pp. 224–244, Mar. 2008.

[28] G. P. White, Y. V. Zakharov, and J. Liu, "Low-complexity RLS algorithms using dichotomous coordinate descent iterations," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3150–3161, Jul. 2008.

[29] H. Zou, "The adaptive Lasso and its oracle properties," *J. Amer. Statist. Assoc.*, vol. 101, no. 476, pp. 1418–1429, 2006.

[30] H. Zou and R. Li, "One-step sparse estimates in nonconcave penalized likelihood models," *Ann. Statist.*, vol. 36, no. 4, pp. 1509–1533, 2008.

Daniele Angelosante (M'09) was born in Frosinone, Italy, on June 1, 1981. He received the B.Sc., Laurea Magistrale, and Ph.D. degrees in telecommunication engineering from the University of Cassino, Italy, in 2003, 2005, and 2009, respectively, and the M.Sc. degree in electrical engineering with specialization in signal and information processing for communication from the University of Aalborg, Denmark, in 2005.

From April to June 2007, he worked at the University Pompeu Fabra, Spain. In 2008, he was a visiting scholar at the University of Minnesota, Minneapolis. He is currently a Postdoctoral Associate with the Department of Electrical and Computer Engineering of the University of Minnesota. His research interests lie in the areas of statistical signal processing, with emphasis on wireless communications, tracking and compressed sensing.

Juan Andrés Bazerque (S'06) received the B.Sc. degree in electrical engineering from Universidad de la Republica (UdelaR), Montevideo, Uruguay, in 2003 and the M.Sc. degree in electrical engineering from the University of Minnesota (UofM), Minneapolis, in 2009. He is currently working towards the Ph.D. degree at the same university.

From 2000 to 2006, he was a Teaching Assistant with the Department of Mathematics and Statistics and with the Department of Electrical Engineering of UdelaR. From 2003 to 2006, he worked as a Telecommunications Engineer at the Uruguayan company Uniotel S.A., developing applications for voice-over-IP. Since August 2006, he has been a Research Assistant at UofM. His broad research interests lie in the general areas of networking, communications, and signal processing. His current research focuses on decentralized algorithms for in-network processing, cooperative wireless communications, cognitive radios, compressive sampling, and sparsity-aware statistical models.

Mr. Bazerque is the recipient of the UofM's Distinguished Master's Thesis Award 2009–2010 and corecipient of the Best Student Paper Award at the Second International Conference on Cognitive Radio Oriented Wireless Networks and Communication 2007.

Georgios B. Giannakis (F'97) received the Diploma degree in electrical engineering from the National Technical University of Athens, Greece, in 1981 and the MSc. degree in electrical engineering, the M.Sc. degree in mathematics, and the Ph.D. degree in electrical engineering from the University of Southern California (USC) in 1983, 1986 and 1986, respectively.

Since 1999, he has been a Professor with the University of Minnesota, where he now holds an ADC Chair in Wireless Telecommunications in the Electric and Computer Engineering Department and serves as Director of the Digital Technology Center. His general interests span the areas of communications, networking and statistical signal processingsubjects on which he has published more than 285 journal papers, 485 conference papers, two edited books, and two research monographs. Current research focuses on compressive sensing, cognitive radios, network coding, cross-layer designs, mobile ad hoc networks, wireless sensor, and social networks.

Dr. Giannakis is the (co)recipient of seven paper awards from the IEEE Signal Processing (SP) and Communications Societies, including the G. Marconi Prize Paper Award inWireless Communications. He also received Technical Achievement Awards from the SP Society (2000), from EURASIP (2005), a Young Faculty Teaching Award, and the G. W. Taylor Award for Distinguished Research from the University of Minnesota. He is a Fellow of EURASIP, has served the IEEE in a number of posts, and is also as a Distinguished Lecturer for the IEEE-SP Society.