An Online Parallel Algorithm for Recursive Estimation of Sparse Signals

Yang Yang, *Member, IEEE*, Marius Pesavento, *Member, IEEE*, Mengyi Zhang, *Member, IEEE*, and Daniel P. Palomar, *Fellow, IEEE*

***Abstract*—In this paper, we consider a recursive estimation problem for linear regression where the signal to be estimated admits a sparse representation and measurement samples are only sequentially available. We propose a convergent parallel estimation scheme that consists of solving a sequence of ℓ_1-regularized least-square problems approximately. The proposed scheme is novel in three aspects: 1) all elements of the unknown vector variable are updated in parallel at each time instant, and the convergence speed is much faster than state-of-the-art schemes which update the elements sequentially; 2) both the update direction and stepsize of each element have simple closed-form expressions, so the algorithm is suitable for online (real-time) implementation; and 3) the stepsize is designed to accelerate the convergence but it does not suffer from the common intricacy of parameter tuning. Both centralized and distributed implementation schemes are discussed. The attractive features of the proposed algorithm are also illustrated numerically.**

***Index Terms*—LASSO, linear regression, minimization stepsize rule, parallel algorithm, recursive estimation, sparse signal processing, stochastic optimization.**

I. Introduction

SIGNAL estimation has been a fundamental problem in a number of scenarios, such as wireless sensor networks (WSN) and cognitive radio (CR). WSN has received a lot of attention and is found application in diverse disciplines such as environmental monitoring, smart grids, and wireless communications [2]. CR appears as an enabling technique for flexible and efficient use of the radio spectrum [3], [4], since it allows unlicensed secondary users (SUs) to access the spectrum provided that the licensed primary users (PUs) are idle, and/or the interference generated by the SUs is below a certain level that is tolerable for the PUs [5], [6].

One prerequisite in CR systems is the ability to obtain a precise estimate of the PUs' power distribution map so that the SUs can avoid the areas in which the PUs are actively transmitting. This is usually realized through the estimation of the position, transmit status, and/or transmit power of PUs [7]–[10], and the estimation is typically obtained based on the minimum mean-square-error (MMSE) criterion [2], [9], [11]–[15].

The MMSE approach involves the calculation of the expectation of a squared ℓ_2-norm function that depends on the so-called regression vector and measurement output, both of which are random variables. This is essentially a stochastic optimization problem, but when the statistics of these random variables are unknown, it is impossible to calculate the expectation analytically. An alternative is to use the sample average function, constructed from sequentially available measurements, as an approximation of the expectation, and this leads to the well-known recursive least-square (RLS) algorithm [2], [12]–[14]. As the measurements are available sequentially, at each time instant of the RLS algorithm, an LS problem has to be solved, which furthermore admits a closed-form solution and thus can efficiently be computed. More details can be found in standard textbooks such as [11], [12].

In practice, the signal to be estimated may be sparse in nature [2], [8], [9], [15], [16]. In a recent attempt to apply the RLS approach to estimate a sparse signal, a regularization function in terms of ℓ_1-norm was incorporated into the LS function to encourage sparse estimates [2], [15]–[19], leading to an ℓ_1-regularized LS problem which has the form of the least-absolute shrinkage and selection operator (LASSO) [20]. Then in the recursive estimation of a sparse signal, the fundamental difference from the standard RLS approach is that at each time instant, instead of solving an LS problem as in the RLS algorithm, an ℓ_1-regularized LS problem in the form of LASSO is solved [2].

However, a closed-form solution to the ℓ_1-regularized LS problem does not exist because of the ℓ_1-norm regularization function and the problem can only be solved iteratively. As a matter of fact, iterative algorithms to solve the ℓ_1-regularized LS problems have been the center of extensive research in recent years and a number of solvers have been developed, e.g., `GP` [21], `l1_ls` [22], `FISTA` [23], `ADMM` [24], `FLEXA` [25], and `DQP-LASSO` [26]. Since the measurements are sequentially available, and with each new measurement, a new ℓ_1-regularized

Manuscript received March 17, 2015; revised September 15, 2015 and February 14, 2016; accepted April 19, 2016. Date of publication May 02, 2016; date of current version August 05, 2016. The work of Y. Yang was supported by the Seventh Framework Programme for Research of the European Commission under Grant ADEL-619647, the EXPRESS project within the DFG priority program CoSIP (DFG-SPP 1798), and the Hong Kong RGC 16207814 research grant. The work of M. Pesavento was supported by the Seventh Framework Programme for Research of the European Commission under Grant ADEL 619647 and the EXPRESS project within the DFG priority program CoSIP (DFG-SPP 1798). The work of M. Zhang and D. P. Palomar was supported by the Hong Kong RGC 16207814 research grant. The material in this paper was presented at the Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, November 2014 [1]. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Cédric Richard.

Y. Yang is with the Intel Deutschland GmbH, Neubiberg 85579, Germany (e-mail: yang1.yang@intel.com).

M. Pesavento is with the Communication Systems Group, Darmstadt University of Technology, Darmstadt 64283, Germany (e-mail: pesavento@nt.tu-darmstadt.de).

M. Zhang is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: zhangmy@cse.cuhk.edu.hk).

D. P. Palomar is with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Kowloon, Hong Kong (e-mail: palomar@ust.hk).

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/TSIPN.2016.2561703

LS problem is formed and solved, the overall complexity of using solvers for the whole sequence of ℓ_1-regularized LS problems is no longer affordable. If the environment is furthermore rapidly changing, this method is not suitable for real-time applications as new measurements may already arrive before the previous ℓ_1-regularized LS problem is solved.

To reduce the complexity of the estimation scheme so that it is suitable for online (real-time) implementation, the authors in [15], [17], [18] proposed algorithms in which the ℓ_1-regularized LS problem at each time instant is solved only *approximately*. For example, in the algorithm proposed in [15], at each time instant, the ℓ_1-regularized LS problem is solved with respect to (w.r.t.) only a single element of the unknown vector variable while the remaining elements are fixed, and the update of that element has a simple closed-form expression based on the so-called soft-thresholding operator [23]. With the next measurement that arrives, a new ℓ_1-regularized LS problem is formed and solved w.r.t. the next element only while the remaining elements are fixed. This sequential update rule is known in literature as the block coordinate descent method [27].

Intuitively, since only a single element is updated at each time instant, the online sequential algorithm proposed in [15] sometimes suffers from slow convergence, especially when the signal has a large dimension while large dimensions of sparse signals are universal in practice. It is tempting to use a parallel scheme in which the update directions of all elements are computed and updated simultaneously at each time instant, but the convergence properties of parallel algorithms are mostly investigated for deterministic optimization problems (see [25] and the references therein) and they may not converge for the stochastic optimization problem at hand. Besides this, the convergence speed of parallel algorithms heavily depends on the choice of the stepsizes. Typical rules for choosing the stepsizes are the Armijo-like successive line search rule, constant stepsize rule, and diminishing stepsize rule. The former two suffer from high complexity and slow convergence [25, Remark 4], while the decay rate of the diminishing stepsize is very difficult to choose: on the one hand, a slowly decaying stepsize is preferable to make notable progress and to achieve satisfactory convergence speed; on the other hand, theoretical convergence is guaranteed only when the stepsizes decays fast enough. It is a difficult task on its own to find the decay rate that yields a good trade-off.

Sparsity-aware learning over network algorithms have been proposed in [16], [28]–[30]. They are suitable for distributed implementation, but they do not converge to the exact MMSE estimate. Other schemes suitable for the online estimation of sparse signals include LMS-type algorithms [18], [31]–[33]. However, their convergence speed is typically slow and the free parameters (e.g., stepsizes) are difficult to choose: either the selection of the free parameters depends on information that is not easily obtainable in practice, such as the statistics of the regression vector, or the convergence is very sensitive to the choice of the free parameters.

A recent work on parallel algorithms for stochastic optimization is [34]. However, the algorithms proposed in [34] are not applicable for the recursive estimation of sparse signals. This is because the regularization function in [34] must be strongly convex and differentiable while the regularization gain must be lower bounded by some positive constant so that convergence can be achieved. However the regularization function in terms of ℓ_1-norm for sparse signal estimation is convex (but not strongly convex) and nondifferentiable while the regularization gain is decreasing to zero.

In this paper, we propose an online parallel algorithm with provable convergence for recursive estimation of sparse signals. In particular, our main contributions are summarized as follows.

Firstly, at each time instant, the ℓ_1-regularized LS problem is solved approximately and all elements are updated in parallel, so the convergence speed is greatly enhanced compared with [15]. As a nontrivial extension of [15] from sequential update to parallel update, and of [25], [35] from deterministic optimization problems to stochastic optimization problems, the convergence of the proposed algorithm is established.

Secondly, we propose a new procedure for the computation of the stepsize based on the so-called minimization rule (also known as exact line search) and its benefits are twofold: firstly, it is essential for the convergence of the proposed algorithm, which may however diverge under other stepsize rules; secondly, notable progress is achieved after each variable update and the common intricacy of complicated parameter tuning is saved. Besides this, both the update direction and stepsize of each element exhibit simple closed-form expressions, so the proposed algorithm is fast to converge and suitable for online implementation.

The rest of the paper is organized as follows. In Section II we introduce the system model and formulate the recursive estimation problem. The online parallel algorithm is proposed in Section III, and its implementations and extensions are discussed in Section IV. The performance of the proposed algorithm is evaluated numerically in Section V and finally concluding remarks are drawn in Section VI.

Notation: We use x, $\mathbf{x}$ and $\mathbf{X}$ to denote scalar, vector and matrix, respectively. X_{jk} is the (j,k)th element of $\mathbf{X}$; x_k and $x_{j,k}$ is the kth element of $\mathbf{x}$ and $\mathbf{x}_j$, respectively, and $\mathbf{x} = (x_k)_{k=1}^K$ and $\mathbf{x}_j = (x_{j,k})_{k=1}^K$. We use $\mathbf{x}_{-k}$ to denote the elements of $\mathbf{x}$ except x_k: $\mathbf{x}_{-k} \triangleq (x_j)_{j=1,j\neq k}^K$. We denote $\mathbf{d}(\mathbf{X})$ as a vector that consists of the diagonal elements of $\mathbf{X}$, $\text{diag}(\mathbf{X})$ as a diagonal matrix whose diagonal elements are the same as those of $\mathbf{X}$, and $\text{diag}(\mathbf{x})$ as a diagonal matrix whose diagonal vector is $\mathbf{x}$, i.e., $\text{diag}(\mathbf{X}) = \text{diag}(\mathbf{d}(\mathbf{X}))$. The operator $[\mathbf{x}]_{\mathbf{a}}^{\mathbf{b}}$ denotes the element-wise projection of $\mathbf{x}$ onto $[\mathbf{a}, \mathbf{b}]$: $[\mathbf{x}]_{\mathbf{a}}^{\mathbf{b}} \triangleq \max(\min(\mathbf{x}, \mathbf{b}), \mathbf{a})$, and $[\mathbf{x}]^+$ denotes the element-wise projection of $\mathbf{x}$ onto the nonnegative orthant: $[\mathbf{x}]^+ \triangleq \max(\mathbf{x}, \mathbf{0})$. The Moore–Penrose inverse of $\mathbf{X}$ is denoted as $\mathbf{X}^\dagger$, and $\lambda_{\max}(\mathbf{X})$ denotes the largest eigenvalue of $\mathbf{X}$.

II. System Model and Problem Formulation

Suppose $\mathbf{x}^\star = (x_k^\star)_{k=1}^K \in \mathbb{R}^K$ is a deterministic sparse signal to be estimated based on the the measurement $y_n \in \mathbb{R}$, and both quantities are connected through a linear regression model:

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

where N is the number of measurements at any time instant. The regression vector $\mathbf{g}_n = (g_{n,k})_{k=1}^K \in \mathbb{R}^K$ is assumed to be known, and $v_n \in \mathbb{R}$ is the additive estimation noise. Throughout the paper, we make the following assumptions on $\mathbf{g}_n$ and v_n for $n = 1, \ldots, N$:

(A1.1) $\mathbf{g}_n$ is a random variable with a bounded positive definite covariance matrix;

(A1.2) v_n is a random variable with zero mean and bounded variance;

(A1.3) $\mathbf{g}_n$ and v_n are uncorrelated.

Sometimes we may also need bounded assumptions on the higher order moments of $\mathbf{g}_n$ and v_n:

(A1.1') $\mathbf{g}_n$ is a random variable whose covariance matrix is positive definite and whose moments are bounded;

(A1.2') v_n is a random variable with zero mean and bounded moments;

(A1.3') $\mathbf{g}_n$ and v_n are uncorrelated.

Given the linear model in (1), the problem is to estimate $\mathbf{x}^\star$ from the set of regression vectors and measurements $\{\mathbf{g}_n, y_n\}_{n=1}^N$. Since both the regression vector $\mathbf{g}_n$ and estimation noise v_n are random variables, the measurement y_n is also random. A fundamental approach to estimate $\mathbf{x}^\star$ is based on the MMSE criterion, which has a solid root in adaptive filter theory [11], [12]. To improve the estimation precision, all available measurements $\{\mathbf{g}_n, y_n\}_{n=1}^N$ are exploited to form a cooperative estimation problem which consists of finding the variable that minimizes the mean-square-error [2], [9], [36]:

$$\begin{aligned}\mathbf{x}^\star &= \underset{\mathbf{x}=(x_k)_{k=1}^K}{\arg\min}\ \mathbb{E}\left[\sum_{n=1}^N \left(y_n - \mathbf{g}_n^T\mathbf{x}\right)^2\right] \qquad (2)\\ &= \underset{\mathbf{x}}{\arg\min}\ \frac{1}{2}\mathbf{x}^T\mathbf{G}\mathbf{x} - \mathbf{b}^T\mathbf{x},\end{aligned}$$

where $\mathbf{G} \triangleq \sum_{n=1}^N \mathbb{E}\left[\mathbf{g}_n\mathbf{g}_n^T\right]$ and $\mathbf{b} \triangleq \sum_{n=1}^N \mathbb{E}\left[y_n\mathbf{g}_n\right]$, and the expectation is taken over $\{\mathbf{g}_n, y_n\}_{n=1}^N$.

In practice, the statistics of $\{\mathbf{g}_n, y_n\}_{n=1}^N$ are often not available to compute $\mathbf{G}$ and $\mathbf{b}$ analytically. In fact, the absence of statistical information is a general rule rather than an exception. A common approach is to approximate the expectation in (2) by the sample average function constructed from the measurements (or realizations) $\{\mathbf{g}_n^{(\tau)}, y_n^{(\tau)}\}_{\tau=1}^t$ sequentially available up to time t [12]:

$$\begin{aligned}\mathbf{x}_{\text{rls}}^{(t)} &\triangleq \underset{\mathbf{x}}{\arg\min}\ \frac{1}{2}\mathbf{x}^T\mathbf{G}^{(t)}\mathbf{x} - (\mathbf{b}^{(t)})^T\mathbf{x} \qquad (3a)\\ &= \mathbf{G}^{(t)\dagger}\mathbf{b}^{(t)}, \qquad (3b)\end{aligned}$$

where $\mathbf{G}^{(t)}$ and $\mathbf{b}^{(t)}$ is the sample average of $\mathbf{G}$ and $\mathbf{b}$, respectively:

$$\mathbf{G}^{(t)} \triangleq \frac{1}{t}\sum_{\tau=1}^t\sum_{n=1}^N \mathbf{g}_n^{(\tau)}(\mathbf{g}_n^{(\tau)})^T,\ \mathbf{b}^{(t)} \triangleq \frac{1}{t}\sum_{\tau=1}^t\sum_{n=1}^N y_n^{(\tau)}\mathbf{g}_n^{(\tau)}. \qquad (4)$$

In literature, (3) is known as RLS, as indicated by the subscript "rls," and $\mathbf{x}_{\text{rls}}^{(t)}$ can be computed efficiently in closed-form, cf. (3b). Note that in (4) there are N measurements $(y_n^{(\tau)}, \mathbf{g}_n^{(\tau)})_{n=1}^N$ available at each time instant τ. For example, in a WSN, $(y_n^{(\tau)}, \mathbf{g}_n^{(\tau)})$ is the measurement available at the sensor n.

In many practical applications, the unknown signal $\mathbf{x}^\star$ is sparse by nature or by design, but $\mathbf{x}_{\text{rls}}^{(t)}$ given by (3) is not necessarily sparse when t is small [20], [22]. To overcome this shortcoming, a sparsity encouraging function in terms of ℓ_1-norm is incorporated into the sample average function in (3), leading to the following ℓ_1-regularized sample average function at any time instant $t = 1, 2, \ldots$ [2], [8], [15]:

$$L^{(t)}(\mathbf{x}) \triangleq \frac{1}{2}\mathbf{x}^T\mathbf{G}^{(t)}\mathbf{x} - (\mathbf{b}^{(t)})^T\mathbf{x} + \mu^{(t)}\left\|\mathbf{x}\right\|_1, \qquad (5)$$

where $\mu^{(t)} > 0$. Define $\mathbf{x}_{\text{lasso}}^{(t)}$ as the minimizing variable of $L^{(t)}(\mathbf{x})$:

$$\mathbf{x}_{\text{lasso}}^{(t)} = \underset{\mathbf{x}}{\arg\min}\ L^{(t)}(\mathbf{x}), \quad t = 1, 2, \ldots, \qquad (6)$$

In literature, problem (6) for any fixed t is known as the *least-absolute shrinkage and selection operator* (LASSO) [20], [22] (as indicated by the subscript "lasso" in (6)). Note that in batch processing [20], [22], problem (6) is solved only once when a certain number of measurements are collected (so t is equal to the number of measurements), while in the recursive estimation of $\mathbf{x}^\star$, the measurements are sequentially available (so t is increasing) and (6) is solved repeatedly at each time instant $t = 1, 2, \ldots$

The advantage of (6) over (2), whose objective function is stochastic and whose calculation depends on unknown parameters $\mathbf{G}$ and $\mathbf{b}$, is that (6) is a sequence of deterministic optimization problems whose theoretical and algorithmic properties have been extensively investigated and widely understood. A natural question arises in this context: is (6) equivalent to (2) in the sense that $\mathbf{x}_{\text{lasso}}^{(t)}$ is a strongly consistent estimator of $\mathbf{x}^\star$, i.e., $\lim_{t\to\infty}\mathbf{x}_{\text{lasso}}^{(t)} = \mathbf{x}^\star$ with probability one? The relation between $\mathbf{x}_{\text{lasso}}^{(t)}$ in (6) and the unknown variable $\mathbf{x}^\star$ is given in the following lemma [15].

Lemma 1: Suppose Assumption (A1) as well as the following assumptions are satisfied for problem (6):

(A2) $\mathbf{g}_n^{(t)}$ ($y_n^{(t)}$, respectively) is an independent identically distributed (i.i.d.) random process with the same probability density function of $\mathbf{g}_n$ (y_n, respectively).

(A3) $\left\{\mu^{(t)}\right\}$ is a positive sequence converging to 0, i.e., $\mu^{(t)} > 0$ and $\lim_{t\to\infty}\mu^{(t)} = 0$.

Then $\lim_{t\to\infty}\mathbf{x}_{\text{lasso}}^{(t)} = \mathbf{x}^\star$ with probability one.

An example of $\mu^{(t)}$ satisfying Assumption (A3) is $\mu^{(t)} = \alpha/t^\beta$ with $\alpha > 0$ and $\beta > 0$. Typical choices of β are $\beta = 1$ and $\beta = 0.5$ [15]. Note that the diminishing regularization gain $\mu^{(t)}$ differentiates our work from [37] in which the sparsity regularization gain is a positive constant $\mu_t = \mu$ for some $\mu > 0$: the algorithms proposed in [37] does not necessarily converge to $\mathbf{x}^\star$ while the algorithm to be proposed in the next section does.

Lemma 1 not only states the relation between $\mathbf{x}_{\text{lasso}}^{(t)}$ and $\mathbf{x}^\star$ from a theoretical perspective, but also suggests a simple algorithmic solution for problem (2): $\mathbf{x}^\star$ can be estimated by solving a sequence of deterministic optimization problems (6), one for each time instant $t = 1, 2, \ldots$. However, in contrast to the RLS algorithm in which each update has a closed-form expression, cf. (3b), problem (6) does not have a closed-form solution and it can only be solved numerically by an iterative algorithm such as `GP` [21], `l1_ls` [22], `FISTA` [23], `ADMM` [24], and `FLEXA` [25]. As a result, solving (6) repeatedly at each time instant $t = 1, 2, \ldots$ is neither computationally practical nor real-time applicable. The aim of the following sections is to develop an algorithm that enjoys easy implementation and fast convergence.

III. The Proposed Online Parallel Algorithm

The LASSO problem in (6) is convex, but the objective function is nondifferentiable and it cannot be minimized in closed-form, so solving (6) completely w.r.t. all elements of $\mathbf{x}$ by a solver at each time instant $t = 1, 2, \ldots$ is neither computationally practical nor suitable for online implementation. To reduce the complexity of the variable update, an algorithm based on inexact optimization is proposed in [15]: at time instant t, only a single element x_k with $k = \text{mod}(t-1, K) + 1$ is updated by its so-called best response, i.e., $L^{(t)}(\mathbf{x})$ is minimized w.r.t. x_k only: $x_k^{(t+1)} = \arg\min L^{(t)}(x_k, \mathbf{x}_{-k}^{(t)})$ with $\mathbf{x}_{-k} \triangleq (x_j)_{j \neq k}$, which can be solved in closed-form, while the remaining elements $\{x_j\}_{j \neq k}$ remain unchanged, i.e., $\mathbf{x}_{-k}^{(t+1)} = \mathbf{x}_{-k}^{(t)}$. At the next time instant $t+1$, a new sample average function $L^{(t+1)}(\mathbf{x})$ is formed with newly arriving measurements, and the $(k+1)$th element, x_{k+1}, is updated by minimizing $L^{(t+1)}(\mathbf{x})$ w.r.t. x_{k+1} only, while the remaining elements again are fixed. Although easy to implement, sequential updating schemes update only a single element at each time instant and they sometimes suffer from slow convergence when the number of elements K is large.

To overcome the slow convergence of the sequential update, we propose an online parallel update scheme, with provable convergence, in which (6) is solved *approximately* by simultaneously updating all elements only once based on their individual best response. Given the current estimate $\mathbf{x}^{(t)}$ which is available before the tth measurement arrives,[11] the estimate update $\mathbf{x}^{(t+1)}$ is determined based on all the measurements collected up to time instant t in a three-step procedure as described next.

Step 1 (Update Direction): In this step, all elements of $\mathbf{x}$ are updated *in parallel* and the update direction of $\mathbf{x}$ at $\mathbf{x} = \mathbf{x}^{(t)}$, denoted as $\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}$, is determined based on the best-response $\hat{\mathbf{x}}^{(t)}$. For each element of $\mathbf{x}$, say x_k, its best response at $\mathbf{x} = \mathbf{x}^{(t)}$ is given by:

$$\hat{x}_k^{(t)} \triangleq \underset{x_k}{\arg\min} \left\{ L^{(t)}(x_k, \mathbf{x}_{-k}^{(t)}) + \frac{1}{2} c_k^{(t)} (x_k - x_k^{(t)})^2 \right\} \quad \forall\, k, \tag{7}$$

where $\mathbf{x}_{-k} \triangleq \{x_j\}_{j \neq k}$ and it is fixed to the values of the preceding time instant $\mathbf{x}_{-k} = \mathbf{x}_{-k}^{(t)}$. An additional quadratic proximal term with $c_k^{(t)} > 0$ is included in (7) for numerical simplicity and stability [27], because it plays an important role in the convergence analysis of the proposed algorithm; conceptually it is a penalty (with variable weight $c_k^{(t)}$) for moving away from the current estimate $x_k^{(t)}$.

After substituting (5) into (7), the best-response in (7) can be expressed in closed-form:

$$\begin{aligned} \hat{x}_k^{(t)} &= \underset{x_k}{\arg\min} \left\{ \begin{array}{l} \frac{1}{2} G_{kk}^{(t)} x_k^2 - r_k^{(t)} \cdot x_k \\ \quad + \mu^{(t)} |x_k| + \frac{1}{2} c_k^{(t)} (x_k - x_k^{(t)})^2 \end{array} \right\} \\ &= \frac{\mathcal{S}_{\mu^{(t)}}(r_k^{(t)}(\mathbf{x}^{(t)}) + c_k^{(t)} x_k^{(t)})}{G_{kk}^{(t)} + c_k^{(t)}}, \quad k = 1, \ldots, K, \end{aligned} \tag{8}$$

or compactly: $\hat{\mathbf{x}}^{(t)} = (\hat{x}_k^{(t)})_{k=1}^K$ and

$$\begin{aligned} \hat{\mathbf{x}}^{(t)} = &\left(\text{diag}\left(\mathbf{G}^{(t)}\right) + \text{diag}\left(\mathbf{c}^{(t)}\right) \right)^{-1} \cdot \\ &\mathcal{S}_{\mu^{(t)}\mathbf{1}} \left(\mathbf{r}^{(t)}\left(\mathbf{x}^{(t)}\right) + \text{diag}\left(\mathbf{c}^{(t)}\right) \mathbf{x}^{(t)} \right), \end{aligned} \tag{9}$$

where

$$\begin{aligned} \mathbf{r}^{(t)}\left(\mathbf{x}^{(t)}\right) &= \left(r_k^{(t)}\left(\mathbf{x}^{(t)}\right) \right)_{k=1}^K \\ &\triangleq \text{diag}\left(\mathbf{G}^{(t)}\right) \mathbf{x}^{(t)} - \left(\mathbf{G}^{(t)} \mathbf{x}^{(t)} - \mathbf{b}^{(t)} \right), \end{aligned} \tag{10}$$

and

$$\mathcal{S}_{\mathbf{a}}(\mathbf{b}) \triangleq [\mathbf{b} - \mathbf{a}]^+ - [-\mathbf{b} - \mathbf{a}]^+$$

is the well-known soft-thresholding operator [23], [38]. From the definition of $\mathbf{G}^{(t)}$ in (4), $\mathbf{G}^{(t)} \succeq \mathbf{0}$ and $G_{kk}^{(t)} \geq 0$ for all k, so the matrix inverse in (9) is defined.[22]

Given the update direction $\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}$, an intermediate update vector $\tilde{\mathbf{x}}^{(t)}(\gamma)$ is defined

$$\tilde{\mathbf{x}}^{(t)}(\gamma) = \mathbf{x}^{(t)} + \gamma \left(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)} \right), \tag{11}$$

where $\gamma \in (0, 1]$ is the stepsize. The update direction $\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}$ is a descent direction of $L^{(t)}(\mathbf{x})$ in the sense specified by the following proposition.

Proposition 2 (Descent Direction): For $\hat{\mathbf{x}}^{(t)} = (\hat{x}_k^{(t)})_{k=1}^K$ given in (9) and the update direction $\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}$, the following holds for any $\gamma \in [0, 1]$:

$$\begin{aligned} &L^{(t)}\left(\tilde{\mathbf{x}}^{(t)}(\gamma)\right) - L^{(t)}\left(\mathbf{x}^{(t)}\right) \\ &\quad \leq -\gamma \left(c_{\min}^{(t)} - \tfrac{1}{2} \lambda_{\max}\left(\mathbf{G}^{(t)}\right) \gamma \right) \left\| \hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)} \right\|_2^2, \end{aligned} \tag{12}$$

where $c_{\min}^{(t)} \triangleq \min_k \{G_{kk}^{(t)} + c_k^{(t)}\} > 0$.

Proof: The proof follows the same line of analysis in [25, Prop. 8(c)] and is thus omitted here. ■

Step 2 (Stepsize): In this step, the stepsize γ in (11) is determined so that fast convergence is observed. It is easy to see from (12) that for sufficiently small γ, the right hand side of (12) becomes negative and $L^{(t)}(\tilde{\mathbf{x}}^{(t)}(\gamma))$ decreases as compared to $L^{(t)}(\mathbf{x}^{(t)})$. Thus, to minimize $L^{(t)}(\tilde{\mathbf{x}}^{(t)}(\gamma))$, a natural choice of the stepsize rule is the so-called "minimization rule" [39, Sec. 2.2.1] (also known as the "exact line search" [40,

[1] $\mathbf{x}^{(1)}$ could be arbitrarily chosen, e.g., $\mathbf{x}^{(1)} = \mathbf{0}$.

[2] Due to the diagonal structure of $\text{diag}(\mathbf{G}^{(t)}) + \text{diag}(\mathbf{c}^{(t)})$, the matrix inverse can be computed from the scalar inverse of the diagonal elements

Sec. 9.2]), which is the stepsize, denoted as $\gamma_{\text{opt}}^{(t)}$, that decreases $L^{(t)}(\tilde{\mathbf{x}}^{(t)}(\gamma))$ to the largest extent:

$$\begin{aligned}\gamma_{\text{opt}}^{(t)} &= \arg\min_{0\le\gamma\le1}\left\{L^{(t)}(\tilde{\mathbf{x}}^{(t)}(\gamma)) - L^{(t)}(\mathbf{x}^{(t)})\right\}\\ &= \arg\min_{0\le\gamma\le1}\left\{\begin{array}{l}\frac{1}{2}(\hat{\mathbf{x}}^{(t)}-\mathbf{x}^{(t)})^T\mathbf{G}^{(t)}(\hat{\mathbf{x}}^{(t)}-\mathbf{x}^{(t)})\cdot\gamma^2\\ +(\mathbf{G}^{(t)}\mathbf{x}^{(t)}-\mathbf{b}^{(t)})^T(\hat{\mathbf{x}}^{(t)}-\mathbf{x}^{(t)})\cdot\gamma\\ +\mu^{(t)}(\left\|\mathbf{x}^{(t)}+\gamma(\hat{\mathbf{x}}^{(t)}-\mathbf{x}^{(t)})\right\|_1-\left\|\mathbf{x}^{(t)}\right\|_1)\end{array}\right\}.\end{aligned} \tag{13}$$

Therefore by definition of $\gamma_{\text{opt}}^{(t)}$ we have for any $\gamma\in[0,1]$:

$$L^{(t)}\left(\mathbf{x}^{(t)}+\gamma_{\text{opt}}^{(t)}\left(\hat{\mathbf{x}}^{(t)}-\mathbf{x}^{(t)}\right)\right)\le L^{(t)}\left(\mathbf{x}^{(t)}+\gamma\left(\hat{\mathbf{x}}^{(t)}-\mathbf{x}^{(t)}\right)\right). \tag{14}$$

However, the applicability of the standard minimization rule (13) is usually limited in practice because of the high computational complexity of solving the optimization problem in (13). In particular, the nondifferentiable ℓ_1-norm function makes it impossible to find a closed-form expression of $\gamma_{\text{opt}}^{(t)}$ and the problem in (13) can only be solved numerically by a solver such as SeDuMi [41].

To obtain a stepsize that exhibits a good trade-off between convergence speed and computational complexity, we propose a *simplified* minimization rule which yields fast convergence but can be computed at a low complexity. Firstly, note that the high complexity of the standard minimization rule lies in the nondifferentiable ℓ_1-norm function in (13). It follows from the convexity of norm functions that for any $\gamma\in[0,1]$:

$$\begin{aligned}&\mu^{(t)}\left(\left\|\mathbf{x}^{(t)}+\gamma(\hat{\mathbf{x}}^{(t)}-\mathbf{x}^{(t)})\right\|_1-\left\|\mathbf{x}^{(t)}\right\|_1\right)\\ &=\mu^{(t)}\left\|(1-\gamma)\mathbf{x}^{(t)}+\gamma\hat{\mathbf{x}}^{(t)}\right\|_1-\mu^{(t)}\left\|\mathbf{x}^{(t)}\right\|_1\\ &\le(1-\gamma)\mu^{(t)}\left\|\mathbf{x}^{(t)}\right\|_1+\gamma\mu^{(t)}\left\|\hat{\mathbf{x}}^{(t)}\right\|_1-\mu^{(t)}\left\|\mathbf{x}^{(t)}\right\|_1 \qquad (15a)\\ &=\mu^{(t)}\left(\left\|\hat{\mathbf{x}}^{(t)}\right\|_1-\left\|\mathbf{x}^{(t)}\right\|_1\right)\cdot\gamma. \qquad (15b)\end{aligned}$$

The right hand side of (15b) is linear in γ, and equality is achieved in (15a) either when $\gamma=0$ or $\gamma=1$.

In the proposed simplified minimization rule, instead of directly minimizing $L^{(t)}(\tilde{\mathbf{x}}^{(t)}(\gamma))-L^{(t)}(\mathbf{x}^{(t)})$ over γ, its upper bound based on (15) is minimized:

$$\gamma^{(t)}\triangleq\arg\min_{0\le\gamma\le1}\left\{\begin{array}{l}\frac{1}{2}\left(\hat{\mathbf{x}}^{(t)}-\mathbf{x}^{(t)}\right)^T\mathbf{G}^{(t)}\left(\hat{\mathbf{x}}^{(t)}-\mathbf{x}^{(t)}\right)\cdot\gamma^2\\ +\left(\mathbf{G}^{(t)}\mathbf{x}^{(t)}-\mathbf{b}^{(t)}\right)^T\left(\hat{\mathbf{x}}^{(t)}-\mathbf{x}^{(t)}\right)\cdot\gamma\\ +\mu^{(t)}\left(\left\|\hat{\mathbf{x}}^{(t)}\right\|_1-\left\|\mathbf{x}^{(t)}\right\|_1\right)\cdot\gamma\end{array}\right\}. \tag{16}$$

The scalar optimization problem in (16) consists of a convex quadratic objective function along with a simple bound constraint and it has a closed-form solution, given by (17) at the bottom of this page. It is easy to verify that $\gamma^{(t)}$ is obtained by projecting the unconstrained optimal variable of the convex quadratic problem in (16) onto the interval $[0,1]$.

The advantage of minimizing the upper bound function of $L^{(t)}(\tilde{\mathbf{x}}^{(t)}(\gamma))$ in (16) is that the optimal γ, denoted as $\gamma^{(t)}$, always has a closed-form expression, cf. (17). At the same time, it also yields a decrease in $L^{(t)}(\mathbf{x})$ at $\mathbf{x}=\mathbf{x}^{(t)}$ as the standard minimization rule $\gamma_{\text{opt}}^{(t)}$ (13) does in (14), and this decreasing property is stated in the following proposition.

Proposition 3: Given $\tilde{\mathbf{x}}^{(t)}(\gamma)$ and $\gamma^{(t)}$ defined in (11) and (16), respectively, the following holds:

$$L^{(t)}\left(\tilde{\mathbf{x}}^{(t)}\left(\gamma^{(t)}\right)\right)\le L^{(t)}\left(\mathbf{x}^{(t)}\right),$$

and equality is achieved if and only if $\gamma^{(t)}=0$.

Proof: Denote the objective function in (16) as $\overline{L}^{(t)}(\tilde{\mathbf{x}}^{(t)}(\gamma))$. It follows from (15) that

$$L^{(t)}\left(\tilde{\mathbf{x}}^{(t)}\left(\gamma^{(t)}\right)\right)-L^{(t)}\left(\mathbf{x}^{(t)}\right)\le\overline{L}^{(t)}\left(\tilde{\mathbf{x}}^{(t)}\left(\gamma^{(t)}\right)\right), \tag{18}$$

and equality in (18) is achieved when $\gamma^{(t)}=0$ and $\gamma^{(t)}=1$.

Besides this, it follows from the definition of $\gamma^{(t)}$ that

$$\overline{L}^{(t)}\left(\tilde{\mathbf{x}}^{(t)}\left(\gamma^{(t)}\right)\right)\le\overline{L}^{(t)}\left(\tilde{\mathbf{x}}^{(t)}(\gamma)\right)\Big|_{\gamma=0}=L^{(t)}\left(\mathbf{x}^{(t)}\right). \tag{19}$$

Since the optimization problem in (16) has a unique optimal solution $\gamma^{(t)}$ given by (17), equality in (19) is achieved if and only if $\gamma^{(t)}=0$. Finally, combining (18) and (19) yields the conclusion stated in the proposition. ■

The signaling required to perform (17) (and also (9)) when implemented distributedly will be discussed in Section IV.

Step 3 (Dynamic Reset): In this step, the estimate update $\mathbf{x}^{(t+1)}$ is defined based on $\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)})$ given in (11) and (17). We first remark that although $\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)})$ yields a lower value of $L^{(t)}(\mathbf{x})$ than $\mathbf{x}^{(t)}$, it is not necessarily the solution of the optimization problem in (6), i.e.,

$$L^{(t)}\left(\mathbf{x}^{(t)}\right)\ge L^{(t)}\left(\tilde{\mathbf{x}}^{(t)}\left(\gamma^{(t)}\right)\right)\ge L^{(t)}\left(\mathbf{x}_{\text{lasso}}^{(t)}\right)=\min_{\mathbf{x}}L^{(t)}(\mathbf{x}). \tag{20}$$

This is because $\mathbf{x}$ is updated only once from $\mathbf{x}=\mathbf{x}^t$ to $\mathbf{x}=\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)})$, which in general can be further improved unless $\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)})=\mathbf{x}_{\text{lasso}}^{(t)}$, i.e., $\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)})$ already minimizes $L^{(t)}(\mathbf{x})$.

The definitions of $L^{(t)}(\mathbf{x})$ and $\mathbf{x}_{\text{lasso}}^{(t)}$ in (5)-(6) reveal that

$$0=L^{(t)}(\mathbf{x})\Big|_{\mathbf{x}=\mathbf{0}}\ge L^{(t)}\left(\mathbf{x}_{\text{lasso}}^{(t)}\right),\ t=1,2,\ldots. \tag{21}$$

$$\gamma^{(t)}=\left[-\frac{(\mathbf{G}^{(t)}\mathbf{x}^{(t)}-\mathbf{b}^{(t)})^T(\hat{\mathbf{x}}^{(t)}-\mathbf{x}^{(t)})+\mu^{(t)}(\left\|\hat{\mathbf{x}}^{(t)}\right\|_1-\left\|\mathbf{x}^{(t)}\right\|_1)}{(\hat{\mathbf{x}}^{(t)}-\mathbf{x}^{(t)})^T\mathbf{G}^{(t)}(\hat{\mathbf{x}}^{(t)}-\mathbf{x}^{(t)})}\right]_0^1. \tag{17}$$

Algorithm 1: The Online Parallel Algorithm for Recursive Estimation of Sparse Signals.

```
Initialization: $\mathbf{x}^{(1)} = \mathbf{0}$, $t = 1$.
At each time instant $t = 1, 2, \ldots$:
Step 1: Calculate $\hat{\mathbf{x}}^{(t)}$ according to (9).
Step 2: Calculate $\gamma^{(t)}$ according to (17).
Step 3-1: Calculate $\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)})$ according to (11).
Step 3-2: Update $\mathbf{x}^{(t+1)}$ according to (23).
```

Depending on whether $L^{(t)}(\mathbf{x}^{(t)})$ is smaller than 0 or not, it is possible to relate (20) and (21) in the following three ways:

$$\begin{aligned}
0 = L^{(t)}(\mathbf{0}) \geq L^{(t)}(\mathbf{x}^{(t)}) \geq L^{(t)}(\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)})) \geq L^{(t)}(\mathbf{x}_{\text{lasso}}^{(t)}),\\
L^{(t)}(\mathbf{x}^{(t)}) \geq 0 = L^{(t)}(\mathbf{0}) \geq L^{(t)}(\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)})) \geq L^{(t)}(\mathbf{x}_{\text{lasso}}^{(t)}),\\
L^{(t)}(\mathbf{x}^{(t)}) \geq L^{(t)}(\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)})) \geq 0 = L^{(t)}(\mathbf{0}) \geq L^{(t)}(\mathbf{x}_{\text{lasso}}^{(t)}).
\end{aligned} \tag{22}$$

The last case in (22) implies that $\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)})$ is not necessarily better than the point $\mathbf{0}$. Therefore we define the estimate update $\mathbf{x}^{(t+1)}$ to be the best point between the two points $\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)})$ and $\mathbf{0}$:

$$\begin{aligned}
\mathbf{x}^{(t+1)} &= \underset{\mathbf{x}\in\left\{\tilde{\mathbf{x}}^{(t+1)},\mathbf{0}\right\}}{\arg\min}\ L^{(t)}(\mathbf{x})\\
&= \begin{cases} \tilde{\mathbf{x}}^{(t)}(\gamma^{(t)}), & \text{if } L^{(t)}(\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)})) \leq L^{(t)}(\mathbf{0}) = 0,\\ \mathbf{0}, & \text{otherwise,} \end{cases}
\end{aligned} \tag{23}$$

and it is straightforward to infer the following relationship among $\mathbf{x}^{(t)}$, $\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)})$, $\mathbf{x}^{(t+1)}$ and $\mathbf{x}_{\text{lasso}}^{(t)}$:

$$L^{(t)}(\mathbf{x}^{(t)}) \geq L^{(t)}(\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)})) \geq L^{(t)}(\mathbf{x}^{(t+1)}) \geq L^{(t)}(\mathbf{x}_{\text{lasso}}^{(t)}).$$

Moreover, the dynamic reset (23) guarantees that

$$\mathbf{x}^{(t+1)} \in \left\{\mathbf{x} : L^{(t)}(\mathbf{x}) \leq 0\right\},\ t = 1, 2, \ldots. \tag{24}$$

Since $\lim_{t\to\infty} \mathbf{G}^{(t)} \succ \mathbf{0}$ and $\mathbf{b}^{(t)}$ converges from Assumptions (A1)-(A2), (24) guarantees that $\left\{\mathbf{x}^{(t)}\right\}$ is a bounded sequence.

Remark 4: Although $L^{(t)}(\mathbf{x}^{t+1}) \leq 0$ for any t according to (24), it may happen that $L^{(t+1)}(\mathbf{x}^{t+1}) > 0$ (unless $\mathbf{x}^{t+1} = \mathbf{0}$, which corresponds to the first two cases in (22)). The last case in (22) is thus still possible and it is necessary to check if $L^{(t+1)}(\tilde{\mathbf{x}}^{t+1}(\gamma^{t+1})) \leq 0$ as in (23).

To summarize the above development, the proposed online parallel algorithm is formally described in Algorithm 1. To analyze the convergence of Algorithm 1, we assume that the sequence $\{\mu^{(t)}\}$ monotonically decreases to 0:

(A3') $\left\{\mu^{(t)}\right\}$ is a positive decreasing sequence converging to 0, i.e., $\mu^{(t+1)} \geq \mu^{(t)} > 0$ for all t and $\lim_{t\to\infty} \mu^{(t)} = 0$.

We also assume that $c_k^{(t)}$ is selected such that:

(A4) $G_{kk}^{(t)} + c_k^{(t)} \geq c$ for some $c > 0$ and all $k = 1, \ldots, K$.

Theorem 5 (Strong Consistency): Suppose Assumptions (A1'), (A2), (A3') and (A4) are satisfied. Then $\mathbf{x}^{(t)}$ is a strongly consistent estimator of $\mathbf{x}^{\star}$, i.e., $\lim_{t\to\infty} \mathbf{x}^{(t)} = \mathbf{x}^{\star}$ with probability one.

Proof: See Appendix A. ■

Assumption (A1') is standard on random variables and is usually satisfied in practice. We can see from Assumption (A4) that if there already exists some value $c > 0$ such that $G_{kk}^{(t)} \geq c$ for all t, the quadratic proximal term in (7) is no longer needed, i.e., we can set $c_k^{(t)} = 0$ without affecting convergence. This is the case when t is sufficiently large because $\lim_{t\to\infty} \mathbf{G}^{(t)} \succ \mathbf{0}$. In practice it may be difficult to decide if t is large enough, so we can just assign a small value to $c_k^{(t)}$ for all t in order to guarantee the convergence. As for Assumption (A3'), it is satisfied by the previously mentioned choices of $\mu^{(t)}$, e.g., $\mu^{(t)} = \alpha/t^{\beta}$ with $\alpha > 0$ and $0.5 \leq \beta \leq 1$.

Theorem 5 establishes that there is no loss of strong consistency if at each time instant, (6) is solved only approximately by updating all elements simultaneously based on the best-response only once. In what follows, we comment on some of the desirable features of Algorithm 1 that make it appealing in practice:

1) Algorithm 1 belongs to the class of parallel algorithms where all elements are updated simultaneously at each time instant. Compared with sequential algorithms where only one element is updated at each time instant [15], the improvement in convergence speed is notable, especially when the signal dimension is large. This is illustrated numerically in Section V (cf. Figs. 1 and 2).
2) Algorithm 1 is easy to implement and suitable for online implementation, since both the computations of the best-response and the stepsize have closed-form expressions. With the simplified minimization stepsize rule, a notable decrease in objective function value is achieved after each variable update, and the difficulty of tuning the decay rate of the diminishing stepsize as required in [35] is saved. Most importantly, the algorithm may not converge under decreasing stepsizes.
3) Algorithm 1 converges under milder assumptions than state-of-the-art algorithms. The regression vector $\mathbf{g}_n$ and the noise v_n do not need to be uniformly bounded, which is required in [42], [43] and which is not satisfied in case of unbounded distributions, e.g., in the Gaussian distribution.

IV. Implementation and Extensions

A. A Special Case: $\mathbf{x}^{\star} \geq \mathbf{0}$

The proposed Algorithm 1 can be further simplified if $\mathbf{x}^{\star}$, the signal to be estimated, has additional properties. For example, in the context of CR studied in [8], $\mathbf{x}^{\star}$ represents the power vector and it is by definition always nonnegative. In this case, a nonnegative constraint on x_k in (7) is needed:

$$\hat{x}_k^{(t)} = \underset{x_k\geq 0}{\arg\min} \left\{L^{(t)}(x_k, \mathbf{x}_{-k}^{(t)}) + \frac{1}{2}c_k^{(t)}(x_k - x_k^{(t)})^2\right\} \quad \forall\, k,$$

and the best-response $\hat{x}_k^{(t)}$ in (9) simplifies to

$$\hat{x}_k^{(t)} = \frac{\left[r_k^{(t)} + c_k^{(t)} x_k^{(t)} - \mu^{(t)}\right]^+}{G_{kk}^{(t)} + c_k^{(t)}},\ k = 1, \ldots, K.$$

Furthermore, since both $\mathbf{x}^{(t)}$ and $\hat{\mathbf{x}}^{(t)}$ are nonnegative, we have

$$\mathbf{x}^{(t)} + \gamma(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}) \geq \mathbf{0},\ 0 \leq \gamma \leq 1,$$

and

$$\begin{aligned}\left\| \mathbf{x}^{(t)} + \gamma(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}) \right\|_1 &= \sum_{k=1}^{K} \left| x_k^{(t)} + \gamma(\hat{x}_k^{(t)} - x_k^{(t)}) \right| \\ &= \sum_{k=1}^{K} x_k^{(t)} + \gamma \left(\hat{x}_k^{(t)} - x_k^{(t)}] \right).\end{aligned}$$

Therefore the *standard*minimization rule (13) can be adopted directly and the stepsize is accordingly given as

$$\gamma^{(t)} = \left[-\frac{(\mathbf{G}^{(t)}\mathbf{x}^{(t)} - \mathbf{b}^{(t)} + \mu^{(t)}\mathbf{1})^T(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)})}{(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)})^T\mathbf{G}^{(t)}(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)})} \right]_0^1,$$

where $\mathbf{1}$ is a vector with all elements equal to 1.

B. Implementation Details and Complexity Analysis

Algorithm 1 can be implemented in a centralized and parallel or a distributed network architecture. To ease the exposition, we discuss the implementation details in the context of a WSN with a total number of N nodes.

Network With a Fusion Center: The fusion center first performs the computation of (9) and (17). Towards this end, signaling from the sensors to the fusion center is required: at each time instant t, each sensor n sends the values $(\mathbf{g}_n^{(t)}, y_n^{(t)}) \in \mathbb{R}^{K+1}$ to the fusion center. Note that $\mathbf{G}^{(t)}$ and $\mathbf{b}^{(t)}$ defined in (4) can be updated recursively

$$\mathbf{G}^{(t)} = \frac{t-1}{t}\mathbf{G}^{(t-1)} + \frac{1}{t}\sum_{n=1}^{N} \mathbf{g}_n^{(t)}(\mathbf{g}_n^{(t)})^T, \tag{25a}$$

$$\mathbf{b}^{(t)} = \frac{t-1}{t}\mathbf{b}^{(t-1)} + \frac{1}{t}\sum_{n=1}^{N} y_n^{(t)}\mathbf{g}_n^{(t)}. \tag{25b}$$

Then after updating $\mathbf{x}$ according to (11) and (23), the fusion center sends $\mathbf{x}^{(t+1)} \in \mathbb{R}^K$ back to all sensors.

We next discuss the computational complexity of Algorithm 1. Note that in (25), the normalization by t is immaterial as it appears in both the numerator and denominator. Among others, $(N+1)(K^2+K)/2$ multiplications and additions are required to compute (25a). Besides this, $3K^2$ multiplications and $3K(K-1)$ additions are required to perform the matrix-vector multiplications $\mathbf{G}^{(t)}\mathbf{x}^{(t)}$ of (10), $\mathbf{G}^{(t)}(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)})$ of (14) and $\mathbf{G}^{(t)}\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)})$ of (23). It is possible to verify that these operations dominate the others in terms of multiplications and additions, and the overall computational complexity is the same as the traditional RLS algorithm [12, Ch. 14].

We further remark that the computations specified in (9), (17) and (23), e.g., the matrix-vector and element-wise vector-vector multiplications, are easily parallelizable by using parallel hardware (e.g., FPGA) or multiple processors/cores. In this case, the computation time could be significantly reduced and this is of great interest in a centralized network as well.

Network Without a Fusion Center: In this case, the computational tasks are evenly distributed among the sensors and the computation in each step of Algorithm 1 is performed locally by each sensor at the price of some signaling exchange among different sensors.

We first define the sensor-specific variables $\mathbf{G}_n^{(t)}$ and $\mathbf{b}_n^{(t)}$ for sensor n as:

$$\mathbf{G}_n^{(t)} \triangleq \frac{1}{t}\sum_{\tau=1}^{t} \mathbf{g}_n^{(\tau)}(\mathbf{g}_n^{(\tau)})^T, \text{ and } \mathbf{b}_n^{(t)} = \frac{1}{t}\sum_{\tau=1}^{t} y_n^{(t)}\mathbf{g}_n^{(t)}, \tag{3}$$

so that $\mathbf{G}^{(t)} = \sum_{n=1}^{N} \mathbf{G}_n^{(t)}$ and $\mathbf{b}^{(t)} = \sum_{n=1}^{N} \mathbf{b}_n^{(t)}$. Note that $\mathbf{G}_n^{(t)}$ and $\mathbf{b}_n^{(t)}$ can be computed *locally* by sensor n without any signaling exchange required. It is also easy to verify that, similar to (25), $\mathbf{G}_n^{(t)}$ and $\mathbf{b}_n^{(t)}$ can be updated recursively by sensor n, so the sensors do not have to store all past data.

The information exchange among sensors in carried out in two phases. Firstly, for sensor n, to perform (9) [Step 1 of Algorithm 1], $\mathbf{d}(\mathbf{G}^{(t)})$ and $\mathbf{r}^{(t)}$ are required,[33] and they can be decomposed as follows:

$$\mathbf{d}(\mathbf{G}^{(t)}) = \sum_{n=1}^{N} \mathbf{d}(\mathbf{G}_n^{(t)}) \in \mathbb{R}^K, \tag{27a}$$

$$\mathbf{G}^{(t)}\mathbf{x}^{(t)} - \mathbf{b}^{(t)} = \sum_{n=1}^{N} \left(\mathbf{G}_n^{(t)}\mathbf{x}^{(t)} - \mathbf{b}_n^{(t)} \right) \in \mathbb{R}^K. \tag{27b}$$

Furthermore, to determine the stepsize (17) [Step 2 of Algorithm 1], the following computations must be available at sensor n:

$$\mathbf{G}^{(t)}\mathbf{x}^{(t)} = \sum_{n=1}^{N} \mathbf{G}_n^{(t)}\mathbf{x}^{(t)} \in \mathbb{R}^K \tag{27c}$$

$$\mathbf{G}^{(t)}\hat{\mathbf{x}}^{(t)} = \sum_{n=1}^{N} \mathbf{G}_n^{(t)}\hat{\mathbf{x}}^{(t)} \in \mathbb{R}^K, \tag{27d}$$

and

$$\begin{aligned}&(\mathbf{G}^{(t)}\mathbf{x}^{(t)} - \mathbf{b}^{(t)})^T(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}) \\ &\quad = \left(\sum_{n=1}^{N} (\mathbf{G}_n^{(t)}\mathbf{x}^{(t)} - \mathbf{b}_n^{(t)}) \right)^T (\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}),\end{aligned} \tag{27e}$$

however, computing (27e) does not require any additional signaling since $\sum_{n=1}^{N} (\mathbf{G}_n^{(t)}\mathbf{x}^{(t)} - \mathbf{b}_n^{(t)})$ is already available from (27b).

With $\hat{\mathbf{x}}^{(t)}$ and $\gamma^{(t)}$, each sensor n can locally calculate $\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)})$ according to (11) [Step 3-1 of Algorithm 1]. Note that $L^{(t)}(\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)}))$ [Step 3-2 of Algorithm 1] can be computed based on available information (27b)–(27d) because

$$\begin{aligned}&L^{(t)}(\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)})) \\ &\quad = \frac{1}{2}(\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)}))^T\mathbf{G}^{(t)}\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)}) \\ &\qquad - (\mathbf{b}^{(t)})^T\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)}) + \mu^{(t)}\|\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)})\|_1\end{aligned}$$

[3] Recall that $\text{diag}(\mathbf{G}^{(t)}) = \text{diag}(\mathbf{d}(\mathbf{G}^{(t)}))$.

$$\begin{aligned}&= \frac{1}{2}(\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)}))^T(\mathbf{G}^{(t)}\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)}) - 2\mathbf{b}^{(t)})\\ &\quad + \mu^{(t)}\|\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)})\|_1\\ &= \frac{1}{2}(\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)}))^T(2(\mathbf{G}^{(t)}\mathbf{x}^{(t)} - \mathbf{b}^{(t)}) - \mathbf{G}^{(t)}\mathbf{x}^{(t)}\\ &\quad + \gamma^{(t)}\mathbf{G}^{(t)}(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)})) + \mu^{(t)}\|\tilde{\mathbf{x}}^{(t)}(\gamma^{(t)})\|_1,\end{aligned}$$

where $\mathbf{G}^{(t)}\mathbf{x}^{(t)} - \mathbf{b}^{(t)}$ comes from (27b), $\mathbf{G}^{(t)}\mathbf{x}^{(t)}$ comes from (27c), and $\mathbf{G}^{(t)}(\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)})$ comes from (27c) and (27d). We can also infer from the above discussion that the most complex operations at each node are the computation of $\mathbf{G}_n^{(t)}$ in (26), which consists of $(K^2 + K)/2$ multiplications and additions, and the matrix-vector multiplications $\mathbf{G}_n^{(t)}\mathbf{x}^{(t)}$ in (27b), (27c) and $\mathbf{G}_n^{(t)}\hat{\mathbf{x}}^{(t)}$ in (27d), each of which consists of K^2 multiplications and $K(K-1)$ additions, leading to a total of $2.5K^2 + 0.5K$ multiplications and $2K(K-1)$ additions.

To summarize, in the first phase, each node needs to exchange $(\mathbf{d}(\mathbf{G}_n^{(t)}), \mathbf{G}_n^{(t)}\mathbf{x}^{(t)} - \mathbf{b}_n^{(t)}) \in \mathbb{R}^{2K\times 1}$, while in the second phase, the sensors need to exchange $(\mathbf{G}_n^{(t)}\mathbf{x}^{(t)}, \mathbf{G}_n^{(t)}\hat{\mathbf{x}}^{(t)}) \in \mathbb{R}^{2K\times 1}$; thus the dimension of the vector that needs to be exchanged at each time instant is $4K$. In what follows, we draw several comments on the information exchange and its implications.

The dimension of the vector to be exchanged is much smaller than in [2] and [8]. For example in [2, A.5], the optimization problem (6) is solved exactly at each time instant t (whereas it is solved only approximately in the proposed Algorithm 1, cf. (20)). In this sense it is essentially a double layer algorithm: in the inner layer, an iterative algorithm is used to solve (6) while in the outer layer t is increased to $t+1$ and (6) is solved again. Suppose the iterative algorithm in the inner layer converges in $T^{(t)}$ iterations; in general $T^{(t)} \gg 1$. In each iteration of the inner layer, the sensors should exchange a vector of the size $2K$, and this is repeated until the termination of the inner layer, leading to a total size of $2T^{(t)}K$, which is much larger than that of the proposed algorithm, namely, $4K$. Furthermore, since the information exchange must be repeated for $T^{(t)}$ times at each time instant, the incurred latency is much longer than that of the proposed algorithm, in which the information exchange is carried out only twice. The analysis for the distributed implementation of [15], proposed in [8], is similar and thus omitted.

In practice, the information exchange could be realized by broadcast, or consensus algorithms if only local communication with neighbor nodes is possible. Since consensus algorithms are of an iterative nature, the proposed distributed algorithm would have an additional inner layer if the consensus algorithm were explicitly counted: in the outer layer, the sensors perform the estimate update (11) and (23); in the inner layer, the sensors compute the average values (27) using an iterative consensus algorithm.[4]

Since the convergence of Algorithm 1 is based on perfect information exchange, we should use consensus algorithms under which the exact consensus is reached in a *finite* number of steps, for example, [44]. More specifically, the exact consensus in [44] is achieved in at most $T^{\max} \leq N + 1 - \min_n |\mathcal{N}_n|$ steps, where $|\mathcal{N}_n|$ is the number of neighbors of the sensor n, so the total signaling overhead at each time instant t of Algorithm 1 is $4T^{\max}K$. However, this specific choice of consensus algorithm imposes additional constraints on the network and the sensors (for example, each sensor should have the knowledge of topology of the global network and additional coordination is required among the sensors), which may impair the applicability of the proposed algorithm.

If consensus algorithms with asymptotic convergence are used for information exchange, they are typically terminated after finite iterations in practice. Then the information available at each sensor is a noisy estimate of the real information and the proposed algorithm may not converge. The convergence in this case requires further investigation.

C. Time- and Norm-Weighted Sparsity Regularization

For a given vector $\mathbf{x}$, its support $\mathcal{S}_{\mathbf{x}}$ is defined as the set of indices of nonzero elements:

$$\mathcal{S}_{\mathbf{x}} \triangleq \{1 \leq k \leq K : x_k \neq 0\}.$$

Suppose without loss of generality that $\mathcal{S}_{\mathbf{x}^\star} = \{1, 2, \ldots, \|\mathbf{x}^\star\|_0\}$, where $\|\mathbf{x}\|_0$ is the number of nonzero elements of $\mathbf{x}$. It is shown in [15] that with the time-weighted sparsity regularization (6), the estimate $\mathbf{x}_{\text{lasso}}^{(t)}$ does not necessarily satisfy the so-called "oracle properties": an estimator $\mathbf{x}^{(t)}$ is said to satisfy the oracle properties if

$$\lim_{t\to\infty} \text{Prob}\left[\mathcal{S}_{\mathbf{x}^{(t)}} = \mathcal{S}_{\mathbf{x}^\star}\right] = 1, \tag{28a}$$

and

$$\sqrt{t}\left(\mathbf{x}_{1:\|\mathbf{x}^\star\|_0}^{(t)} - \mathbf{x}_{1:\|\mathbf{x}^\star\|_0}^\star\right) \to_d \mathcal{N}\left(0, \sigma^2\mathbf{G}_{1:\|\mathbf{x}^\star\|_0,1:\|\mathbf{x}^\star\|_0}\right), \tag{28b}$$

where $\to_d$ means convergence in distribution and $\mathbf{G}_{1:k,1:k} \in \mathbb{R}^{k\times k}$ is the upper left block of $\mathbf{G}$. The first property (28a) and the second property (28b) is called support consistency and $\sqrt{t}$-estimation consistency, respectively [15].

To make the estimation satisfy the oracle properties, it was suggested in [15] that a time- and norm-weighted LASSO can be used, and the loss function $L^{(t)}(\mathbf{x})$ in (5) can be modified as follows:

$$\begin{aligned}L^{(t)}(\mathbf{x}) = \frac{1}{t}\sum_{\tau=1}^{t}\sum_{n=1}^{N}(y_n^{(\tau)} - (\mathbf{g}_n^{(\tau)})^T\mathbf{x})^2&\\ + \mu^{(t)}\sum_{k=1}^{K}\mathcal{W}_{\mu^{(t)}}(|x_{\text{rls},k}^{(t)}|)\cdot|x_k|,&\end{aligned} \tag{29}$$

where 1) $\mathbf{x}_{\text{rls}}^{(t)}$ is given in (3); 2) $\lim_{t\to\infty}\mu^{(t)} = 0$ and $\lim_{t\to\infty}\sqrt{t}\cdot\mu^{(t)} = \infty$, so $\mu^{(t)}$ must decrease slower than $1/\sqrt{t}$;

[4] The two-layer structure of the proposed algorithm is different from that of [2], [8]: since the average values in [2], [8] are also computed using an iterative consensus algorithm, the algorithms proposed in [2], [8] would have three layers if the consensus algorithm were explicitly counted.

3) the weight factor $\mathcal{W}_\mu(x)$ is defined as

$$\mathcal{W}_\mu(x) \triangleq \begin{cases} 1, & \text{if } x \le \mu, \\ \dfrac{a\mu - x}{(a-1)\mu}, & \text{if } \mu \le x \le a\mu, \\ 0, & \text{if } x \ge a\mu, \end{cases}$$

and $a > 1$ is a given constant. Therefore, the value of the weight function $\mu^{(t)}\mathcal{W}_{\mu^{(t)}}(|x_{\text{rls},k}^{(t)}|)$ in (29) depends on the relative magnitudes of $\mu^{(t)}$ and $x_{\text{rls},k}^{(t)}$.

After replacing the universal sparsity regularization gain $\mu^{(t)}$ by $\mu^{(t)}\mathcal{W}_{\mu^{(t)}}(\left|x_{\text{rls},k}^{(t)}\right|)$ for each element x_k in (9) and (17), Algorithm 1 can readily be applied to estimate $\mathbf{x}^\star$ based on the time- and norm-weighted loss function (29) and the strong consistency also holds. To see this, we only need to verify the nonincreasing property of the weight function $\mu^{(t)}\mathcal{W}_{\mu^{(t)}}(|x_{\text{rls},k}^{(t)}|)$. We remark that when t is sufficiently large, it is either $\mu^{(t)}\mathcal{W}_{\mu^{(t)}}(|x_{\text{rls},k}^{(t)}|) = 0$ or $\mu^{(t)}\mathcal{W}_{\mu^{(t)}}(|x_{\text{rls},k}^{(t)}|) = \mu^{(t)}$. This is because $\lim_{t\to\infty}\mathbf{x}_{\text{rls}}^{(t)} = \mathbf{x}^\star$ under the conditions of Lemma 1. If $x_k^\star > 0$, since $\lim_{t\to\infty}\mu^{(t)} = 0$, there exists for any arbitrarily small $\epsilon > 0$ some t_0 such that $a\mu^{(t)} < x_k^\star - \epsilon$ for all $t \ge t_0$; the weight factor in this case is 0 for all $t \ge t_0$, and the nonincreasing property is automatically satisfied. If, on the other hand, $x_k^\star = 0$, then $x_{\text{rls}}^{(t)}$ converges to $x_k^\star = 0$ at a rate of $1/\sqrt{t}$ [45]. Since $\mu^{(t)}$ decreases slower than $1/\sqrt{t}$, there exists some t_0 such that $x_{\text{rls},k}^{(t)} < \mu^{(t)}$ for all $t \ge t_0$. In this case, $\mathcal{W}_{\mu^{(t)}}(x_{\text{rls},k}^{(t)})$ is equal to 1 and the weight factor is simply $\mu^{(t)}$ for all $t \ge t_0$, which is nonincreasing.

D. Recursive Estimation of Time-Varying Signals

If the signal to be estimated is time-varying, the loss function (5) needs to be modified in a way such that the new measurement samples are given more weight than the old ones. Defining the so-called "forgetting factor" β, where $0 < \beta < 1$, the new loss function is given as [2], [12], [15]:

$$\min_{\mathbf{x}} \ \frac{1}{2t}\sum_{n=1}^{N}\sum_{\tau=1}^{t}\beta^{t-\tau}((\mathbf{g}_n^{(\tau)})^T\mathbf{x} - y_n^{(\tau)})^2 + \mu^{(t)}\left\|\mathbf{x}\right\|_1. \quad (30)$$

We observe that when $\beta = 1$, (30) is as same as (5). In this case, the only modification to Algorithm 1 is that $\mathbf{G}^{(t)}$ and $\mathbf{b}^{(t)}$ are updated according to the following recursive rule:

$$\mathbf{G}^{(t)} = \frac{1}{t}\left((t-1)\beta\mathbf{G}^{(t-1)} + \sum_{n=1}^{N}\mathbf{g}_n^{(t)}(\mathbf{g}_n^{(t)})^T\right),$$

$$\mathbf{b}^{(t)} = \frac{1}{t}\left((t-1)\beta\mathbf{b}^{(t-1)} + \sum_{n=1}^{N}y_n^{(t)}\mathbf{g}_n^{(t)}\right).$$

For problem (30), since the signal to be estimated is time-varying, the convergence analysis in Theorem 5 does not hold any more. However, simulation results show that there is little loss of optimality when optimizing (30) only approximately by Algorithm 1. This establishes the superiority of the proposed algorithm over the distributed algorithm in [2] which solves (30) exactly at the price of a large delay and a large signaling burden. Besides this, despite the lack of theoretical analysis, Algorithm 1 performs better than the online sequential algorithm [15] numerically, cf. Section V.

V. Numerical Results

In this section, the desirable features of the proposed algorithm are illustrated numerically. Unless otherwise stated, the simulation setup is as follows: 1) the number of sensors $N = 1$, so the subscript n in $\mathbf{g}_n$ is omitted; 2) the dimension of $\mathbf{x}^\star$: $K = 100$; 3) the proportion of the nonzero elements of $\mathbf{x}^\star$: 0.1; 4) both $\mathbf{g}$ and v are generated by i.i.d. standard normal distributions: $\mathbf{g} \in \mathcal{CN}(\mathbf{0}, \mathbf{I})$ and $v \in \mathcal{CN}(0, 0.2)$; 5) the sparsity regularization gain $\mu^{(t)} = 10/t$; 6) the simulations results are averaged over 100 realizations.

A. Convergence to the Optimal Value

We plot in Fig. 1 the relative error of the objective value $(L^{(t)}(\mathbf{x}^{(t)}) - L^{(t)}(\mathbf{x}_{\text{lasso}}^{(t)}))/L^{(t)}(\mathbf{x}_{\text{lasso}}^{(t)})$ versus the time instant t for two dimensions of $\mathbf{x}^\star$ (with $\mathbf{x}^{(0)} = \mathbf{0}$), namely, $K = 100$ in Fig. 1(a) and $K = 500$ in Fig. 1(b), where 1) $\mathbf{x}_{\text{lasso}}^{(t)}$ is defined in (6) and calculated by `MOSEK` [46]; 2) $\mathbf{x}^{(t)}$ is returned by Algorithm 1 in the proposed online parallel algorithm (coined as "parallel algorithm"); 3) $\mathbf{x}^{(t)}$ is returned by [15, Algorithm 1] in the online sequential algorithm (coined as "sequential algorithm"), where only one element of $\mathbf{x}$ is updated at each time instant; 4) in the "enhanced sequential algorithm," all elements of $\mathbf{x}$ are sequentially updated once at each time instant. Define

$$\mathbf{z}^{(t,k)} \triangleq [\hat{x}_1^{(t)}, \ldots, \hat{x}_k^{(t)}, x_{k+1}^{(t)}, \ldots, x_K^{(t)}]^T, \ 1 \le k \le K,$$

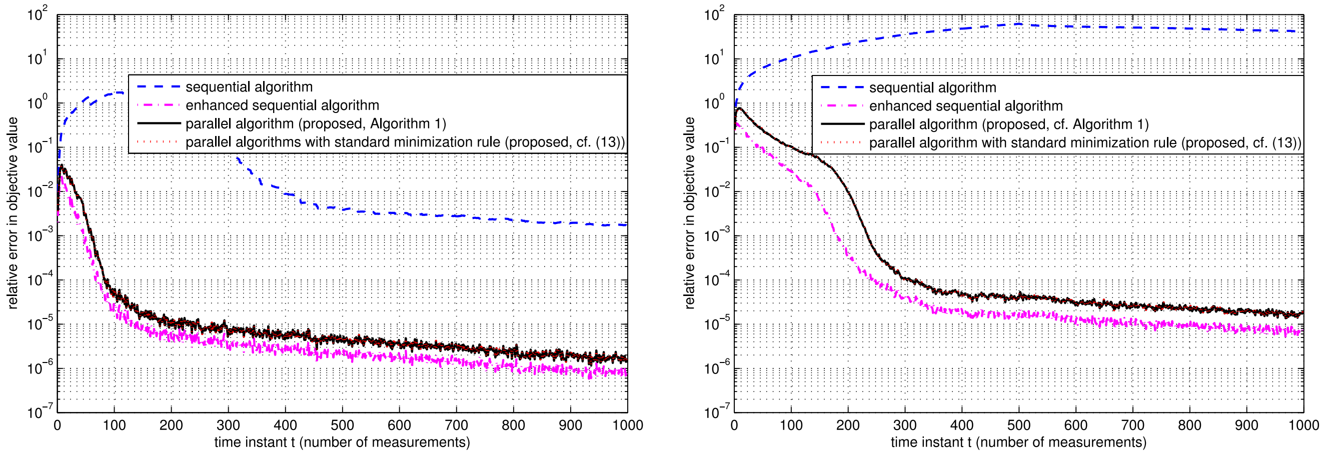
where $\hat{x}_k^{(t)} = (G_{kk}^{(t)} + c_k^{(t)})^{-1}\mathcal{S}_{\mu^{(t)}}(r_k^{(t)}(\mathbf{x}^{(t,k-1)}) + c_k^{(t)}x_k^{(t)})$; the variable update in the enhanced sequential algorithm can mathematically be expressed as[55]

$$\mathbf{x}^{(t+1)} = \mathbf{z}^{(t,K)}. \quad (31)$$

Note that $L^{(t)}(\mathbf{x}_{\text{lasso}}^{(t)})$ is by definition the lower bound of $L^{(t)}(\mathbf{x})$ and $L^{(t)}(\mathbf{x}^{(t)}) - L^{(t)}(\mathbf{x}_{\text{lasso}}^{(t)}) \ge 0$ for all t. From Fig. 1 it is clear that the proposed algorithm (black curve) converges to a precision of 10^{-2} with less than 200 measurements while the sequential algorithm (blue curve) either requires many more measurements (cf. Fig. 11(a)) or does not even converge with a reasonable number of measurements (cf. Fig. 1(b)). The improvement in convergence speed is thus notable, and the proposed online parallel algorithm outperforms the sequential algorithm both in convergence speed and solution quality. Besides this, a comparison of the proposed algorithm for different signal dimensions in Fig. 1(a) and (b) indicates that the proposed algorithm scales well and it is very practical.

We remark that the computational complexity per time instant of the sequential algorithm [15] is approximately $1/K$ that of the proposed algorithm, because the former updates a single element of $\mathbf{x}$ only according to (8), while the latter updates all

[5] The enhanced sequential algorithm is suggested by the reviewers.



(a)

(b)

Fig. 1. Convergence behavior in terms of objective function value. (a) Signal dimension: $K = 100$. (b) Signal dimension: $K = 500$.

elements of $\mathbf{x}$ simultaneously based on (9). The computational complexity per time instant of the enhanced sequential algorithm is roughly the same as that of the proposed algorithm, because the operation (8) is performed for K times after a complete cycle of element updates. However, the associated computational time per time instant of the enhanced sequential algorithm is (at least) K times as long as that of the proposed algorithm, because the proposed update (9) is parallelizable by using parallel hardware (e.g., FPGA) or multiple processors/cores. Thus the proposed algorithm is more suitable for online applications.

When implemented in a distributed manner, the enhanced sequential algorithm incurs a large signaling overhead. Following the line of analysis in Section IV, we remark that the nodes need to exchange $\mathbf{G}_n^{(t)}\mathbf{x}^{(t,k)} - \mathbf{b}^{(t)} \in \mathbb{R}^K$ after $\mathbf{x}^{(t,k)}$ is obtained so that $\mathbf{x}^{(t,k+1)}$ can be computed by each node locally. At each time instant, a complete cycle with K sequential element updates then leads to a total dimension of K^2, which is much larger than that of the proposed algorithm, namely, $4K$. The larger signaling overhead also increases the latency. Thus the proposed algorithm is more suitable for distributed implementation. Although we observe from Fig. 1 that the proposed algorithm converges slightly slower than the enhanced sequential algorithm, the significantly reduced computational time and signaling overhead justify the superiority of the proposed algorithm.

We also evaluate in Fig. 1 the performance loss incurred by the simplified minimization rule (16) (indicated by the black curve) compared with the standard minimization rule (13) (indicated by the red curve). It is easy to see from Fig. 1 that these two curves almost coincide with each other, so the extent to which the simplified minimization rule decreases the objective function is nearly the same as the standard minimization rule and the performance loss is negligible.

B. Convergence to the Optimal Variable

Then we consider in Fig. 2 the relative square error $\left\|\mathbf{x}^{(t)} - \mathbf{x}^\star\right\|_2^2 / \left\|\mathbf{x}^\star\right\|_2^2$ versus the time instant t. To compare the estimation approaches with and without sparsity regularization, the RLS algorithm in (3) is also implemented, where a ℓ_2 regularization term $10^{-4}\left\|\mathbf{x}\right\|_2^2$ is included into (3). Some observations are in order.

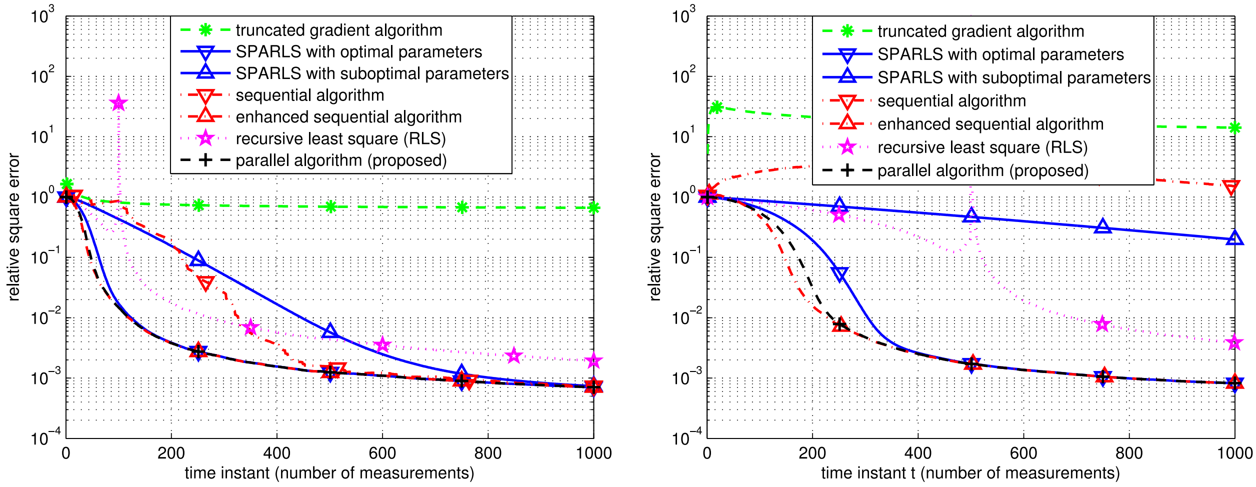
We see that the proposed online parallel algorithm (indicated by the black curve) and the enhanced sequential algorithm (indicated by the red curve with upper triangular) exhibit faster convergence than other algorithms. From Fig. 2 we see that when the signal dimension is increased from $K = 100$ to $K = 500$, the convergence speed of the proposed online parallel algorithm is not severely slowed down, which shows that the proposed algorithm scales well.

The enhanced sequential algorithm converges slightly faster than the proposed online parallel algorithm in the early iterations, but the difference is negligible. Note that the computational time and signaling overhead of the enhanced sequential algorithm does not scale well because they are proportional to K, the dimension of $\mathbf{x}^\star$. By comparison, as the update is parallelizable and signaling exchange is carried out only once at each time instant, the proposed algorithm achieves almost the same performance but at a reduced cost of computational time and signaling overhead than the enhanced sequential algorithm (cf. Section V-A).

We note that the estimation with sparsity regularization performs better than the classic RLS approach (indicated by the magenta curve), especially when t is small. This can be explained by the fact that a prior information of the sparsity of the signal $\mathbf{x}^\star$ is exploited.

The proposed algorithm performs better than the SPARLS algorithm with optimal parameters [17] (indicated by the blue curve with reversed triangular). However, to obtain the optimal parameters, the maximum eigenvalue of $\mathbf{G}^{(t)}$ must be computed, which is a computational prohibitive task in large-scale problems. If we use a suboptimal parameter tr$(\mathbf{G}^{(t)})$ instead of the optimal parameter $\lambda_{\max}(\mathbf{G}^{(t)})$, then the performance of SPARLS with suboptimal parameters (indicated by the blue curve with triangular) deteriorates significantly, especially when K is large, cf. Fig. 2(b).

We use the same choice of free parameters (e.g., stepsize and regularization gain) for the truncated gradient algorithm (indicated by the the green) in both settings $K = 100$ and $K = 500$.



(a)

(b)

Fig. 2. Relative square error for recursive estimation of time-varying signals. (a) Signal dimension: $K = 100$. (b) Signal dimension: $K = 500$.

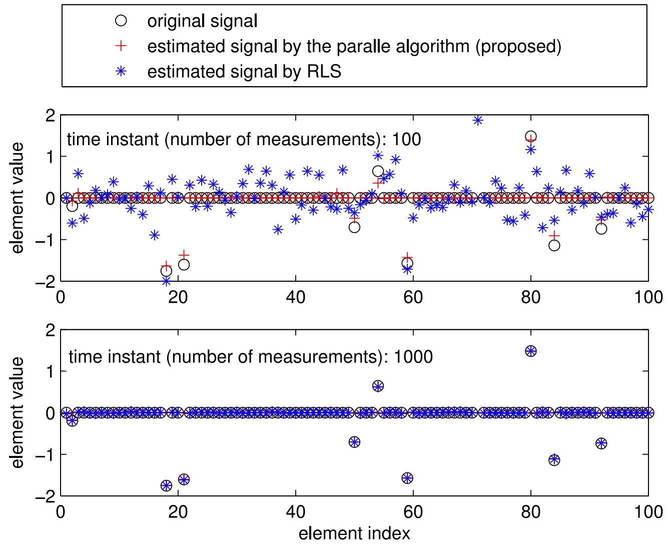


Fig. 3. Comparison of original signal and estimated signal at different time instant: $t = 100$ in the upper plot and $t = 1000$ in the lower plot.

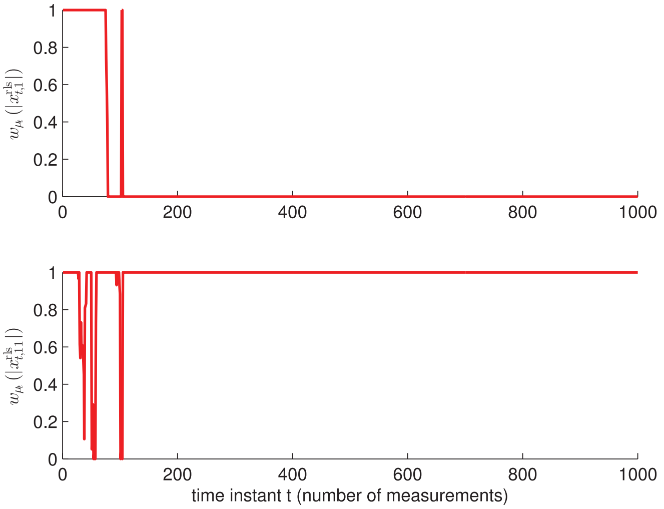


Fig. 4. Weight factor in time- and norm-weighted sparsity regularization.

It is observed from the comparison of Fig. 2(a) and (b) that the truncated gradient algorithm [18] is sensitive to the choice of free parameters and no general rule applies to all problem parameters. Furthermore, it converges slowly because it is essentially a gradient method. By comparison, no pretuning is required in the proposed algorithm and simple closed-form expressions exist for each update. The proposed algorithm is easy to use in practice and robust to changes in problem parameters.

The precision of the estimated signal by the proposed online parallel algorithm (after 100 and 1000 time instant, respectively) is shown element-wise in Fig. 3 when $K = 100$. Given 100 measurements, we observe from the upper plot of Fig. 3 that the proposed online parallel algorithm can accurately estimate the support of $\mathbf{x}^\star$, while as expected, the estimated signal based on the RLS algorithm is not sparse. When the number of measurements is increased to 1000, we can see from the lower plot of Fig. 3 that the value of $\mathbf{x}^\star$ is accurately estimated by the proposed online parallel algorithm. The same observation holds for the RLS algorithm as well because $\mathbf{x}^{(t)}_{\text{rls}} \to \mathbf{x}^\star$.

C. Weight Factor in Time- and Norm-Weighted Sparsity Regularization

In Fig. 4 we simulate the weight factor $\mathcal{W}_{\mu^{(t)}}(|x^{(t)}_{\text{rls},k}|)$ versus the time instant t in time- and norm-weighted sparsity regularization, where $K = 100$, $k = 1$ is used in the upper plot and $k = 11$ in the lower plot. The parameters are the same as in the previous simulation examples, except that $\mu^{(t)} = 1/t^{0.4}$ and $\mathbf{x}^\star$ are generated such that the first $0.1 \times K$ elements (where 0.1 is the proportion of nonzero elements of $\mathbf{x}^\star$) are nonzero while all other elements are zero. The weight factors of other elements are omitted because they exhibit similar behavior as the ones plotted in Fig. 4. As analyzed, $\mathcal{W}_{\mu^{(t)}}(|w^{(t)}_{\text{rls},1}|)$, the weight factor of the first element, where $x^\star_1 \neq 0$, quickly converges to zero, while $\mathcal{W}_{\mu^{(t)}}(|w^{(t)}_{\text{rls},11}|)$, the weight factor of the eleventh

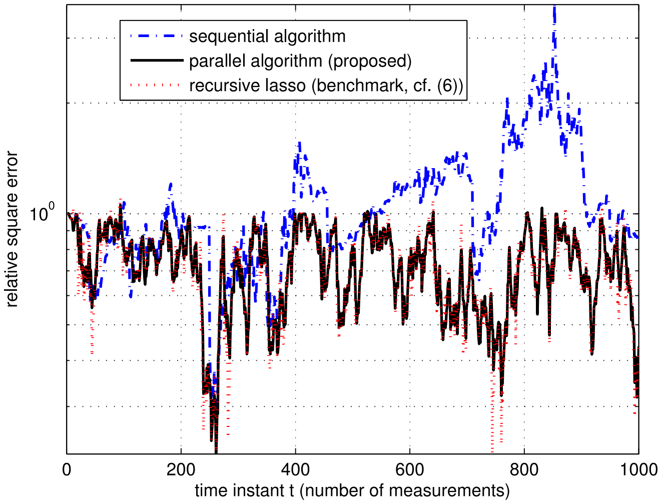


Fig. 5. Relative square error for recursive estimation of time-varying signals.

element, where $x_1^\star = 0$, quickly converges to one, making the overall weight factor monotonically decreasing, cf. (29). Therefore the proposed algorithm can readily be applied to the recursive estimation of sparse signals with time- and norm-weighted regularization.

D. Estimation of the Time-Varying Signal

When the signal to be estimated is varying, the theoretical analysis of the proposed algorithm is not valid anymore, but we can test numerically how the proposed algorithm performs compared with the online sequential algorithm. The time-varying unknown signal is denoted as $\mathbf{x}_t^\star \in \mathbb{R}^{100\times 1}$, and it is changing according to the following law:

$$x_{t+1,k}^\star = \alpha x_{t,k}^\star + w_{t,k},$$

where $w_{t,k} \sim \mathcal{CN}(0, 1-\alpha^2)$ for any k such that $x_{t,k}^\star \neq 0$, with $\alpha = 0.99$ and $\beta = 0.9$. In Fig. 5, the relative square error $\|\mathbf{x}_t - \mathbf{x}_t^\star\|_2^2 \,/\, \|\mathbf{x}_t^\star\|_2^2$ is plotted versus the time instant t. Despite the lack of theoretical analysis, we observe the estimation error of the proposed online parallel algorithm (indicated by the black curve) is almost as same as that of the benchmark in which the LASSO problem is solved exactly (indicated by the red curve), so the approximate optimization is not an impeding factor for the estimation accuracy. This is another advantage of the proposed algorithm over [2] where a distributed iterative algorithm is employed to solve (30) exactly, which inevitably incurs a large delay and extensive signaling.

VI. Concluding Remarks

In this paper, we have considered the recursive estimation of sparse signals and proposed an online parallel algorithm with provable convergence. The algorithm is based on approximate optimization but it converges to the exact solution. At each time instant, all elements are updated in parallel, and both the update direction and the stepsize can be calculated in analytical expressions. The proposed simplified minimization stepsize rule is well motivated and easily implementable, achieves a good trade-off between complexity and convergence speed, and avoids the common drawbacks of the standard stepsizes used in literature. Simulation results have demonstrated the notable improvement in convergence speed over state-of-the-art techniques. Our results show that the loss in convergence speed compared with the benchmark (where the LASSO problem is solved exactly at each time instant) is negligible. We have also considered numerically the recursive estimation of time-varying signals where the theoretical convergence does not necessarily hold, and the proposed algorithm performs better than state-of-the-art algorithms.

Appendix A
Proof of Theorem 5

Proof: It is easy to see that $L^{(t)}$ can be divided into the differentiable part $f^{(t)}(\mathbf{x})$ and the nondifferentiable part $h^{(t)}(\mathbf{x})$: $L^{(t)}(\mathbf{x}) = f^{(t)}(\mathbf{x}) + h^{(t)}(\mathbf{x})$,

$$f^{(t)}(\mathbf{x}) \triangleq \frac{1}{2}\mathbf{x}^T\mathbf{G}^{(t)}\mathbf{x} - (\mathbf{b}^{(t)})^T\mathbf{x}, \tag{32a}$$

$$h^{(t)}(\mathbf{x}) \triangleq \mu^{(t)}\left\|\mathbf{x}\right\|_1. \tag{32b}$$

We also use $f_k^{(t)}(x;\mathbf{x}^{(t)})$ to denote the smooth part of the objective function in (9):

$$f_k^{(t)}(x;\mathbf{x}^{(t)}) \triangleq \frac{1}{2}G_{kk}^{(t)}x^2 - r_k^{(t)}\cdot x + \frac{1}{2}c_k^{(t)}(x - x_k^{(t)})^2. \tag{33}$$

Functions $f_k^{(t)}(x;\mathbf{x}^{(t)})$ and $f^{(t)}(\mathbf{x})$ are related according to the following equation:

$$f_k^{(t)}(x_k;\mathbf{x}^{(t)}) = f^{(t)}(x_k, \mathbf{x}_{-k}^{(t)}) + \frac{1}{2}c^{(t)}(x_k - x_k^{(t)})^2, \tag{34}$$

from which it is easy to infer that $\nabla f_k^{(t)}(x_k^{(t)};\mathbf{x}^{(t)}) = \nabla_k f^{(t)}(\mathbf{x}^{(t)})$. Then from the first-order optimality condition, $h^{(t)}(x_k)$ has a subgradient $\xi_k^{(t)} \in \partial h^{(t)}(\hat{x}_k^{(t)})$ at $x_k = \hat{x}_k^{(t)}$ such that for any x_k:

$$(x_k - \hat{x}_k^{(t)})(\nabla f_k^{(t)}(\hat{x}_k^{(t)};\mathbf{x}^{(t)}) + \xi_k^{(t)}) \geq 0, \; \forall k. \tag{35}$$

Now consider the following equation:

$$\begin{aligned} &L^{(t)}(\mathbf{x}^{(t+1)}) - L^{(t-1)}(\mathbf{x}^{(t)}) = \\ &\quad L^{(t)}(\mathbf{x}^{(t+1)}) - L^{(t)}(\mathbf{x}^{(t)}) + L^{(t)}(\mathbf{x}^{(t)}) - L^{(t-1)}(\mathbf{x}^{(t)}). \end{aligned} \tag{36}$$

The rest of the proof consists of three parts. Firstly we prove in Part I that there exists a constant $\eta > 0$ such that $L^{(t)}(\mathbf{x}^{(t+1)}) - L^{(t)}(\mathbf{x}^{(t)}) \leq -\eta\left\|\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}\right\|_2^2$. Then we show in Part 2 that the sequence $\left\{L^{(t)}(\mathbf{x}^{(t+1)})\right\}_t$ converges. Finally we prove in Part 3 that any limit point of the sequence $\left\{\mathbf{x}^{(t)}\right\}_t$ is a solution of (2).

Part 1) Since $c_{\min}^{(t)} \geq c > 0$ for all t ($c_{\min}^{(t)}$ is defined in Proposition 2) from Assumption (A4), it is easy to see from (12) that the following is true:

$$\begin{aligned}&L^{(t)}(\mathbf{x}^{(t)}+\gamma(\hat{\mathbf{x}}^{(t)}-\mathbf{x}^{(t)}))-L^{(t)}(\mathbf{x}^{(t)})\\&\quad\leq-\gamma\Big(c-\frac{1}{2}\lambda_{\max}(\mathbf{G}^{(t)})\gamma\Big)\big\|\hat{\mathbf{x}}^{(t)}-\mathbf{x}^{(t)}\big\|_2^2,\ 0\leq\gamma\leq 1.\end{aligned}$$

Since $\lambda_{\max}(\mathbf{G}^{(t)})$ is a continuous function [47] and $\mathbf{G}^{(t)}$ converges to a positive definite matrix by Assumption (A1'), there exists a $\bar{\lambda}<+\infty$ such that $\bar{\lambda}\geq\lambda_{\max}(\mathbf{G}^{(t)})$ for all t. We thus conclude from the preceding inequality that for all $0\leq\lambda\leq 1$:

$$\begin{aligned}&L^{(t)}(\mathbf{x}^{(t)}+\gamma(\hat{\mathbf{x}}^{(t)}-\mathbf{x}^{(t)}))-L^{(t)}(\mathbf{x}^{(t)})\\&\quad\leq-\gamma\left(c-\frac{1}{2}\bar{\lambda}\gamma\right)\big\|\hat{\mathbf{x}}^{(t)}-\mathbf{x}^{(t)}\big\|_2^2.\end{aligned}\tag{37}$$

It follows from (15), (16) and (37) that

$$\begin{aligned}&L^{(t)}(\tilde{\mathbf{x}}^{(t+1)})\\&\leq f^{(t)}(\mathbf{x}^{(t)}+\gamma^{(t)}(\hat{\mathbf{x}}^{(t)}-\mathbf{x}^{(t)}))\\&\quad+(1-\gamma^{(t)})h^{(t)}(\mathbf{x}^{(t)})+\gamma^{(t)}h^{(t)}(\hat{\mathbf{x}}^{(t)})&(38)\\&\leq f^{(t)}(\mathbf{x}^{(t)}+\gamma(\hat{\mathbf{x}}^{(t)}-\mathbf{x}^{(t)}))\\&\quad+(1-\gamma)h^{(t)}(\mathbf{x}^{(t)})+\gamma h^{(t)}(\hat{\mathbf{x}}^{(t)})&(39)\\&\leq L^{(t)}(\mathbf{x}^{(t)})-\gamma(c-\frac{1}{2}\bar{\lambda}\gamma)\big\|\hat{\mathbf{x}}^{(t)}-\mathbf{x}^{(t)}\big\|_2^2.&(40)\end{aligned}$$

Since the inequalities in (40) are true for any $0\leq\gamma\leq 1$, we set $\gamma=\min(c/\bar{\lambda},1)$. Then it is possible to show that there is a constant $\eta>0$ such that

$$\begin{aligned}L^{(t)}(\mathbf{x}^{(t+1)})-L^{(t)}(\mathbf{x}^{(t)})&\leq L^{(t)}(\tilde{\mathbf{x}}^{(t+1)})-L^{(t)}(\mathbf{x}^{(t)})\\&\leq-\eta\big\|\hat{\mathbf{x}}^{(t)}-\mathbf{x}^{(t)}\big\|_2^2.\end{aligned}\tag{41}$$

Besides this, because of Step 3 in Algorithm 1, $\mathbf{x}^{(t+1)}$ is in the following lower level set of $L^{(t)}(\mathbf{x})$:

$$\mathcal{L}_{\leq 0}^{(t)}\triangleq\{\mathbf{x}:L^{(t)}(\mathbf{x})\leq 0\}.\tag{42}$$

Because $\|\mathbf{x}\|_1\geq 0$ for any $\mathbf{x}$, (42) is a subset of

$$\left\{\mathbf{x}:\frac{1}{2}\mathbf{x}^T\mathbf{G}^{(t)}\mathbf{x}-(\mathbf{b}^{(t)})^T\mathbf{x}\leq 0\right\},$$

which is a subset of

$$\bar{\mathcal{L}}_{\leq 0}^{(t)}\triangleq\left\{\mathbf{x}:\frac{1}{2}\lambda_{\max}(\mathbf{G}^{(t)})\,\|\mathbf{x}\|_2^2-(\mathbf{b}^{(t)})^T\mathbf{x}\leq 0\right\}.\tag{43}$$

Since $\mathbf{G}^{(t)}$ and $\mathbf{b}^{(t)}$ converges and $\lim_{t\to\infty}\mathbf{G}^{(t)}\succ\mathbf{0}$, there exists a bounded set, denoted as $\mathcal{L}_{\leq 0}$, such that $\mathcal{L}_{\leq 0}^{(t)}\subseteq\bar{\mathcal{L}}_{\leq 0}^{(t)}\subseteq\mathcal{L}_{\leq 0}$ for all t; thus the sequence $\{\mathbf{x}^{(t)}\}$ is bounded and we denote its upper bound as $\bar{\mathbf{x}}$.

Part 2) Combining (36) and (41), we have the following:

$$\begin{aligned}&L^{(t+1)}(\mathbf{x}^{(t+2)})-L^{(t)}(\mathbf{x}^{(t+1)})\\&\quad\leq L^{(t+1)}(\mathbf{x}^{(t+1)})-L^{(t)}(\mathbf{x}^{(t+1)})\\&\quad=f^{(t+1)}(\mathbf{x}^{(t+1)})-f^{(t)}(\mathbf{x}^{(t+1)})\\&\quad+h^{(t+1)}(\mathbf{x}^{(t+1)})-h^{(t)}(\mathbf{x}^{(t+1)})\\&\quad\leq f^{(t+1)}(\mathbf{x}^{(t+1)})-f^{(t)}(\mathbf{x}^{(t+1)}),\end{aligned}\tag{44}$$

where the last inequality comes from the decreasing property of $\mu^{(t)}$ by Assumption (A3'). Recalling the definition of $f^{(t)}(\mathbf{x})$ in (32), it is easy to see that

$$\begin{aligned}&(t+1)(f^{(t+1)}(\mathbf{x}^{(t+1)})-f^{(t)}(\mathbf{x}^{(t+1)}))\\&\quad=l^{(t+1)}(\mathbf{x}^{(t+1)})-\frac{1}{t}\sum_{\tau=1}^{t}l^{(\tau)}(\mathbf{x}^{(t+1)}),\end{aligned}$$

where

$$l^{(t)}(\mathbf{x})\triangleq\sum_{n=1}^{N}(y_n^{(t)}-(\mathbf{g}_n^{(t)})^T\mathbf{x})^2.$$

Taking the expectation of the preceding equation with respect to $\{y_n^{(t+1)},\mathbf{g}_n^{(t+1)}\}_{n=1}^N$, conditioned on the natural history up to time $t+1$, denoted as $\mathcal{F}^{(t+1)}$:

$$\begin{aligned}&\mathcal{F}^{(t+1)}=\\&\quad\left\{\mathbf{x}^{(0)},\ldots,\mathbf{x}^{(t+1)},\{\mathbf{g}_n^{(0)},\ldots,\mathbf{g}_n^{(t)}\}_n,\{y_n^{(0)},\ldots,y_n^{(t)}\}_n\right\},\end{aligned}$$

we have

$$\begin{aligned}&\mathbb{E}\left[(t+1)(f^{(t+1)}(\mathbf{x}^{(t+1)})-f^{(t)}(\mathbf{x}^{(t+1)}))|\mathcal{F}^{(t+1)}\right]\\&\quad=\mathbb{E}\left[l^{(t+1)}(\mathbf{x}^{(t+1)})|\mathcal{F}^{(t+1)}\right]-\frac{1}{t}\sum_{\tau=1}^{t}\mathbb{E}\left[l^{(\tau)}(\mathbf{x}^{(t+1)})|\mathcal{F}^{(t+1)}\right]\\&\quad=\mathbb{E}\left[l^{(t+1)}(\mathbf{x}^{(t+1)})|\mathcal{F}^{(t+1)}\right]-\frac{1}{t}\sum_{\tau=1}^{t}l^{(\tau)}(\mathbf{x}^{(t+1)}),\end{aligned}\tag{45}$$

where the second equality comes from the observation that $l^{(\tau)}(\mathbf{x}^{(t+1)})$ is deterministic as long as $\mathcal{F}^{(t+1)}$ is given. This together with (44) indicates that

$$\begin{aligned}&\mathbb{E}\left[L^{(t+1)}(\mathbf{x}^{(t+2)})-L^{(t)}(\mathbf{x}^{(t+1)})|\mathcal{F}^{(t+1)}\right]\\&\quad\leq\mathbb{E}\left[f^{(t+1)}(\mathbf{x}^{(t+1)})-f^{(t)}(\mathbf{x}^{(t+1)})|\mathcal{F}^{(t+1)}\right]\\&\quad\leq\frac{1}{t+1}\left(\mathbb{E}\left[l^{(t+1)}(\mathbf{x}^{(t+1)})|\mathcal{F}^{(t+1)}\right]-\frac{1}{t}\sum_{\tau=1}^{t}l^{(\tau)}(\mathbf{x}^{(t+1)})\right)\\&\quad\leq\frac{1}{t+1}\left|\mathbb{E}\left[l^{(t+1)}(\mathbf{x}^{(t+1)})|\mathcal{F}^{(t+1)}\right]-\frac{1}{t}\sum_{\tau=1}^{t}l^{(\tau)}(\mathbf{x}^{(t+1)})\right|,\end{aligned}$$

and

$$\begin{aligned}
&\left[\mathbb{E}\left[L^{(t+1)}(\mathbf{x}^{(t+2)}) - L^{(t)}(\mathbf{x}^{(t+1)})|\mathcal{F}^{(t+1)}\right]\right]_0\\
&\quad\le \frac{1}{t+1}\left|\mathbb{E}\left[l^{(t+1)}(\mathbf{x}^{(t+1)})|\mathcal{F}^{(t+1)}\right] - \frac{1}{t}\sum_{\tau=1}^{t} l^{(\tau)}(\mathbf{x}^{(t+1)})\right|\\
&\quad\le \frac{1}{t+1}\sup_{\mathbf{x}\in\mathcal{X}}\left|\mathbb{E}\left[l^{(t+1)}(\mathbf{x})|\mathcal{F}^{(t+1)}\right] - \frac{1}{t}\sum_{\tau=1}^{t} l^{(\tau)}(\mathbf{x})\right|,
\end{aligned} \tag{46}$$

where $[x]_0 = \max(x, 0)$, and $\mathcal{X}$ in (46) with $\mathcal{X} \triangleq \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots,\}$ is the complete path of $\mathbf{x}$.

Now we derive an upper bound on the expected value of the right hand side of (46):

$$\begin{aligned}
&\mathbb{E}\left[\sup_{\mathbf{x}\in\mathcal{X}}\left|\mathbb{E}\left[l^{(t+1)}(\mathbf{x})|\mathcal{F}^{(t+1)}\right] - \frac{1}{t}\sum_{\tau=1}^{t} l^{(\tau)}(\mathbf{x})\right|\right]\\
&\quad= \mathbb{E}\left[\sup_{\mathbf{x}\in\mathcal{X}}\left|\breve{y}^{(t)} - (\mathbf{r}_2^{(t)})^T\mathbf{x} + \mathbf{x}^T\mathbf{R}_3^{(t)}\mathbf{x}\right|\right]\\
&\quad\le \mathbb{E}\left[\sup_{\mathbf{x}\in\mathcal{X}}\left|\breve{y}^{(t)}\right| + \sup_{\mathbf{x}\in\mathcal{X}}\left|(\breve{\mathbf{b}}^{(t)})^T\mathbf{x}\right| + \sup_{\mathbf{x}\in\mathcal{X}}\left|\mathbf{x}^T\breve{\mathbf{G}}^{(t)}\mathbf{x}\right|\right]\\
&\quad= \mathbb{E}\left[\sup_{\mathbf{x}\in\mathcal{X}}\left|\breve{y}^{(t)}\right|\right] + \mathbb{E}\left[\sup_{\mathbf{x}\in\mathcal{X}}\left|(\breve{\mathbf{b}}^{(t)})^T\mathbf{x}\right|\right] + \mathbb{E}\left[\sup_{\mathbf{x}\in\mathcal{X}}\left|\mathbf{x}^T\breve{\mathbf{G}}^{(t)}\mathbf{x}\right|\right],
\end{aligned} \tag{47}$$

where

$$\begin{aligned}
\breve{y}^{(t)} &\triangleq \frac{1}{t}\sum_{\tau=1}^{t}\sum_{n=1}^{N}\left(\mathbb{E}_{y_n}\left[y_n^2\right] - (y_n^{(\tau)})^2\right),\\
\breve{\mathbf{b}}^{(t)} &\triangleq \frac{1}{t}\sum_{\tau=1}^{t}\sum_{n=1}^{N} 2\left(\mathbb{E}_{\{y_n,\mathbf{g}_n\}}\left[y_n\mathbf{g}_n\right] - y_n^{(\tau)}\mathbf{g}_n^{(\tau)}\right),\\
\breve{\mathbf{G}}^{(t)} &\triangleq \frac{1}{t}\sum_{\tau=1}^{t}\sum_{n=1}^{N}\left(\mathbb{E}_{\mathbf{g}_n}\left[\mathbf{g}_n\mathbf{g}_n\right] - \mathbf{g}_n^{(t)}\mathbf{g}_n^{(\tau)T}\right).
\end{aligned}$$

Then we bound each term in (47) individually. For the first term, since $\breve{y}^{(t)}$ is independent of $\mathbf{x}^{(t)}$,

$$\begin{aligned}
\mathbb{E}\left[\sup_{\mathbf{x}\in\mathcal{X}}\left|\breve{y}^{(t)}\right|\right] &= \mathbb{E}\left[\left|\breve{y}^{(t)}\right|\right] = \mathbb{E}\left[\sqrt{(\breve{y}^{(t)})^2}\right]\\
&\le \sqrt{\mathbb{E}\left[(\breve{y}^{(t)})^2\right]} \le \sqrt{\frac{\sigma_1^2}{t}}
\end{aligned} \tag{48}$$

for some $\sigma_1 < \infty$, where the second equality comes from Jensen's inequality. Because of Assumptions (A1') and (A2), $\breve{y}^{(t)}$ has bounded moments and the existence of σ_1 is then justified by the central limit theorem [48].

For the second term of (47), we have

$$\mathbb{E}\left[\sup_{\mathbf{x}}\left|(\breve{\mathbf{b}}^{(t)})^T\mathbf{x}\right|\right] \le \mathbb{E}\left[\sup_{\mathbf{x}}(\left|\breve{\mathbf{b}}^{(t)}\right|)^T\left|\mathbf{x}\right|\right] \le \left(\mathbb{E}\left[\left|\breve{\mathbf{b}}^{(t)}\right|\right]\right)^T\left|\bar{\mathbf{x}}\right|.$$

Similar to the line of analysis of (48), there exists a $\sigma_2 < \infty$ such that

$$\mathbb{E}\left[\sup_{\mathbf{x}}\left|(\breve{\mathbf{b}}^{(t)})^T\mathbf{x}\right|\right] \le \left(\mathbb{E}\left[\left|\breve{\mathbf{b}}^{(t)}\right|\right]\right)^T\left|\bar{\mathbf{x}}\right| \le \sqrt{\frac{\sigma_2^2}{t}}. \tag{49}$$

For the third term of (47), we have

$$\begin{aligned}
&\mathbb{E}\left[\sup_{\mathbf{x}\in\mathcal{X}}\left|\mathbf{x}^T\breve{\mathbf{G}}^{(t)}\mathbf{x}\right|\right]\\
&\quad= \mathbb{E}\left[\max_{1\le k\le K}\left|\lambda_k(\breve{\mathbf{G}}^{(t)})\right|\cdot\left\|\bar{\mathbf{x}}\right\|_2^2\right]\\
&\quad= \left\|\bar{\mathbf{x}}\right\|_2^2\cdot\mathbb{E}\left[\sqrt{\max\{\lambda_{\max}^2(\breve{\mathbf{G}}^{(t)}),\lambda_{\min}^2(\breve{\mathbf{G}}^{(t)})\}}\right]\\
&\quad\le \left\|\bar{\mathbf{x}}\right\|_2^2\cdot\sqrt{\mathbb{E}\left[\max\{\lambda_{\max}^2(\breve{\mathbf{G}}^{(t)}),\lambda_{\min}^2(\breve{\mathbf{G}}^{(t)})\}\right]}\\
&\quad\le \left\|\bar{\mathbf{x}}\right\|_2^2\cdot\sqrt{\mathbb{E}\left[\sum_{k=1}^{K}\lambda_k^2(\breve{\mathbf{G}}^{(t)})\right]}\\
&\quad= \left\|\bar{\mathbf{x}}\right\|_2^2\cdot\sqrt{\mathbb{E}\left[\operatorname{tr}\left(\breve{\mathbf{G}}^{(t)}(\breve{\mathbf{G}}^{(t)})^T\right)\right]} \le \sqrt{\frac{\sigma_3^2}{t}}
\end{aligned} \tag{50}$$

for some $\sigma_3 < \infty$, where the first equality comes from the observation that $\mathbf{x}$ should align with the eigenvector associated with the eigenvalue with largest absolute value. Then combing (48)–(50), we can claim that there exists $\sigma \triangleq \sqrt{\sigma_1^2} + \sqrt{\sigma_2^2} + \sqrt{\sigma_3^2} > 0$ such that

$$\mathbb{E}\left[\sup_{\mathbf{x}\in\mathcal{X}}\left|\mathbb{E}\left[l^{(t+1)}(\mathbf{x})|\mathcal{F}^{(t+1)}\right] - \frac{1}{t}\sum_{\tau=1}^{t} l^{(\tau)}(\mathbf{x})\right|\right] \le \frac{\sigma}{\sqrt{t}}.$$

In view of (46), we have

$$\mathbb{E}\left[\left[\mathbb{E}\left[L^{(t+1)}(\mathbf{x}^{(t+2)}) - L^{(t)}(\mathbf{x}^{(t+1)})|\mathcal{F}^{(t+1)}\right]\right]_0\right] \le \frac{\sigma}{t^{3/2}}. \tag{51}$$

Summing (51) over t, we obtain

$$\sum_{t=1}^{\infty}\mathbb{E}\left[\left[\mathbb{E}\left[L^{(t+1)}(\mathbf{x}^{(t+2)}) - L^{(t)}(\mathbf{x}^{(t+1)})|\mathcal{F}^{(t+1)}\right]\right]_0\right] < \infty.$$

Then it follows from the quasi-martingale convergence theorem (cf. [42, Th. 6]) that $\left\{L^{(t)}(\mathbf{x}^{(t+1)})\right\}$ converges almost surely.

Part 3) Combining (36) and (41), we have

$$\begin{aligned}
&L^{(t)}(\mathbf{x}^{(t+1)}) - L^{(t-1)}(\mathbf{x}^{(t)}) \le\\
&\qquad -\eta\left\|\hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)}\right\|_2^2 + L^{(t)}(\mathbf{x}^{(t)}) - L^{(t-1)}(\mathbf{x}^{(t)}).
\end{aligned} \tag{52}$$

Besides this, it follows from the convergence of $\left\{L^{(t)}(\mathbf{x}^{(t+1)})\right\}_t$

$$\lim_{t\to\infty} L^{(t)}(\mathbf{x}^{(t+1)}) - L^{(t-1)}(\mathbf{x}^{(t)}) = 0,$$

and the strong law of large numbers that

$$\lim_{t\to\infty} L^{(t)}(\mathbf{x}^{(t)}) - L^{(t-1)}(\mathbf{x}^{(t)}) = 0.$$

Taking the limit inferior of both sides of (52), we have

$$\begin{aligned}
0 =\ & \liminf_{t\to\infty} \left\{ L^{(t)}(\mathbf{x}^{(t+1)}) - L^{(t-1)}(\mathbf{x}^{(t)}) \right\} \\
\leq\ & \liminf_{t\to\infty} \left\{ -\eta \left\| \hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)} \right\|_2^2 + L^{(t)}(\mathbf{x}^{(t)}) - L^{(t-1)}(\mathbf{x}^{(t)}) \right\} \\
\leq\ & \liminf_{t\to\infty} \left\{ -\eta \left\| \hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)} \right\|_2^2 \right\} \\
& + \limsup_{t\to\infty} \left\{ L^{(t)}(\mathbf{x}^{(t)}) - L^{(t-1)}(\mathbf{x}^{(t)}) \right\} \\
=\ & -\eta \cdot \limsup_{t\to\infty} \left\| \hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)} \right\|_2^2 \leq 0,
\end{aligned}$$

so we can infer that $\limsup_{t\to\infty} \left\| \hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)} \right\|_2 = 0$. Since $0 \leq \liminf_{t\to\infty} \left\| \hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)} \right\|_2 \leq \limsup_{t\to\infty} \left\| \hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)} \right\|_2 = 0$, we can infer that $\liminf_{t\to\infty} \left\| \hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)} \right\| = 0$ and thus $\lim_{t\to\infty} \left\| \hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)} \right\| = 0$.

Consider any limit point of the sequence $\left\{ \mathbf{x}^{(t)} \right\}_t$, denoted as $\mathbf{x}^{(\infty)}$. Since $\hat{\mathbf{x}}$ is a continuous function of $\mathbf{x}$ in view of (9) and $\lim_{t\to\infty} \left\| \hat{\mathbf{x}}^{(t)} - \mathbf{x}^{(t)} \right\|_2 = 0$, it must be $\lim_{t\to\infty} \hat{\mathbf{x}}^{(t)} = \hat{\mathbf{x}}^{(\infty)} = \mathbf{x}^{(\infty)}$, and the minimum principle in (35) can be simplified as

$$(x_k - x_k^{(\infty)})(\nabla_k f^{(\infty)}(\mathbf{x}^{(\infty)}) + \xi_k^{(\infty)}) \geq 0,\ \forall x_k,$$

whose summation over $k = 1, \ldots, K$ leads to

$$(\mathbf{x} - \mathbf{x}^{(\infty)})^T (\nabla f^{(\infty)}(\mathbf{x}^{(\infty)}) + \boldsymbol{\xi}^{(\infty)}) \geq 0, \quad \forall \mathbf{x}.$$

Therefore $\mathbf{x}^{(\infty)}$ minimizes $L^{(\infty)}(\mathbf{x})$ and $\mathbf{x}^{(\infty)} = \mathbf{x}^\star$ almost surely by Lemma 1. Since $\mathbf{x}^\star$ is unique in view of Assumptions (A1'), the whole sequence $\{\mathbf{x}^{(t)}\}$ has a unique limit point and it thus converges to $\mathbf{x}^\star$. The proof is thus completed. ■

Acknowledgment

The authors would like to thank the reviewers whose comments have greatly improved the quality of the paper.

References

[1] Y. Yang, M. Zhang, M. Pesavento, and D. P. Palomar, "An online parallel algorithm for spectrum sensing in cognitive radio networks," in *Proc. 48th Asilomar Conf. Signals, Syst. Comput.*, 2014, pp. 1801–1805.

[2] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Distributed detection and estimation in wireless sensor networks," in *Academic Press Library in Signal Processing*, R. Chellappa and S. Theodoridis, Eds. New York, NY, USA: Academic, 2014, vol. 2, pp. 329–408.

[3] S. Haykin, "Cognitive radio: Brain-empowered wireless communications," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 2, pp. 201–220, Feb. 2005.

[4] J. Mitola and G. Maguire, "Cognitive radio: Making software radios more personal," *IEEE Pers. Commun.*, vol. 6, no. 4, pp. 13–18, Aug. 1999.

[5] R. Zhang, Y.-C. Liang, and S. Cui, "Dynamic resource allocation in cognitive radio networks," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 102–114, May 2010.

[6] Y. Yang, G. Scutari, P. Song, and D. P. Palomar, "Robust MIMO cognitive radio systems under interference temperature constraints," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 11, pp. 2465–2482, Nov. 2013.

[7] S. Haykin, D. Thomson, and J. Reed, "Spectrum sensing for cognitive radio," *Proc. IEEE*, vol. 97, no. 5, pp. 849–877, May 2009.

[8] S.-J. Kim, E. Dall'Anese, and G. B. Giannakis, "Cooperative spectrum sensing for cognitive radios using Kriged Kalman filtering," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 24–36, Feb. 2011.

[9] F. Zeng, C. Li, and Z. Tian, "Distributed compressive spectrum sensing in cooperative multihop cognitive networks," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 1, pp. 37–48, Feb. 2011.

[10] O. Mehanna and N. D. Sidiropoulos, "Frugal sensing: Wideband power spectrum sensing from few bits," *IEEE Trans. Signal Process.*, vol. 61, no. 10, pp. 2693–2703, May 2013.

[11] S. M. Kay, *Fundamentals of Statistical Signal Processing, Volume I: Estimation Theory*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993.

[12] A. Sayed, *Adaptive Filters*. Hoboken, NJ, USA: Wiley-Interscience, 2008.

[13] G. Mateos, I. Schizas, and G. Giannakis, "Distributed recursive least-squares for consensus-based in-network adaptive estimation," *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4583–4588, Nov. 2009.

[14] G. Mateos and G. B. Giannakis, "Distributed recursive least-squares: Stability and performance analysis," *IEEE Trans. Signal Process.*, vol. 60, no. 7, pp. 3740–3754, Jul. 2012.

[15] D. Angelosante, J. A. Bazerque, and G. B. Giannakis, "Online adaptive estimation of sparse signals: Where RLS meets the ℓ_1-norm," *IEEE Trans. Signal Process.*, vol. 58, no. 7, pp. 3436–3447, Jul. 2010.

[16] Y. Kopsinis, K. Slavakis, and S. Theodoridis, "Online sparse system identification and signal reconstruction using projections onto weighted ℓ_1 balls," *IEEE Trans. Signal Process.*, vol. 59, no. 3, pp. 936–952, Mar. 2011.

[17] B. Babadi, N. Kalouptsidis, and V. Tarokh, "SPARLS: The sparse RLS algorithm," *IEEE Trans. Signal Process.*, vol. 58, no. 8, pp. 4013–4025, Aug. 2010.

[18] J. Langford, L. Li, and T. Zhang, "Sparse online learning via truncated gradient," *J. Mach. Learn. Res.*, vol. 10, pp. 777–801, 2009.

[19] Y. Chen and A. O. Hero, "Recursive $\ell_{1,\infty}$ group Lasso," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 3978–3987, Aug. 2012.

[20] R. Tibshirani, "Regression shrinkage and selection via the lasso: A retrospective," *J. Roy. Statist. Soc. B, Statist. Methodol.*, vol. 58, no. 1, pp. 267–288, Jun. 1996.

[21] M. A. T. Figueiredo, R. D. Nowak, and S. J. 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, Dec. 2007.

[22] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale l_1-regularized least squares," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 606–617, Dec. 2007.

[23] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, Jan. 2009.

[24] T. Goldstein and S. Osher, "The split Bregman method for L1-regularized problems," *SIAM J. Imag. Sci.*, vol. 2, no. 2, pp. 323–343, 2009.

[25] F. Facchinei, G. Scutari, and S. Sagratella, "Parallel selective algorithms for nonconvex big data optimization," *IEEE Trans. Signal Process.*, vol. 63, no. 7, pp. 1874–1889, Nov. 2015.

[26] G. Mateos, J. A. Bazerque, and G. B. Giannakis, "Distributed sparse linear regression," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5262–5276, Oct. 2010.

[27] D. P. Bertsekas and J. N. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1989.

[28] S. Chouvardas, K. Slavakis, Y. Kopsinis, and S. Theodoridis, "A sparsity promoting adaptive algorithm for distributed learning," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5412–5425, Oct. 2012.

[29] P. Di Lorenzo and A. H. Sayed, "Sparse distributed learning based on diffusion adaptation," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1419–1433, Mar. 2013.

[30] P. Di Lorenzo, "Diffusion adaptation strategies for distributed estimation over Gaussian Markov random fields," *IEEE Trans. Signal Process.*, vol. 62, no. 21, pp. 5748–5760, Nov. 2014.

[31] Y. Chen, Y. Gu, and A. O. Hero, "Sparse LMS for system identification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, no. 3, pp. 3125–3128, 2009.

[32] Y. Chen, Y. Gu, and A. O. Hero, "Regularized least-mean-square algorithms," Tech. Rep., Jun. 2010. [Online]. Available: http://arxiv.org/abs/1012.5066

[33] Y. Liu, C. Li, and Z. Zhang, "Diffusion sparse least-mean squares over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4480–4485, Aug. 2012.

[34] Y. Yang, G. Scutari, D. P. Palomar, and M. Pesavento, "A parallel decomposition method for nonconvex stochastic multi-agent optimization problems," *IEEE Trans. Signal Process.,*vol. 64, no. 11, pp. 2949–2964, Jun. 2016.

[35] G. Scutari, F. Facchinei, P. Song, D. P. Palomar, and J.-S. Pang, "Decomposition by partial linearization: Parallel optimization of multi-agent systems," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 641–656, Feb. 2014.

[36] Z. Quan, S. Cui, H. Poor, and A. Sayed, "Collaborative wideband sensing for cognitive radios," *IEEE Signal Process. Mag.*, vol. 25, no. 6, pp. 60–73, Nov. 2008.

[37] S. Ghadimi and G. Lan, "Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization I: A generic

algorithmic framework," *SIAM J. Optimization*, vol. 22, no. 4, pp. 1469–1492, Nov. 2012.

[38] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[39] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA, USA: Athena Scientific, 1999.

[40] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[41] J. F. Sturm, "Using SeDuMi 1.02: A Matlab toolbox for optimization over symmetric cones," *Optimization Methods Softw.*, vol. 11, no. 1–4, pp. 625–653, Jan. 1999.

[42] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learning Res.*, vol. 11, pp. 19–60, 2010.

[43] M. Razaviyayn, M. Sanjabi, and Z.-Q. Luo, "A stochastic successive minimization method for nonsmooth nonconvex optimization," *Math. Program.*, Jun. 2013. [Online]. Available: http://arxiv.org/abs/1307.4457

[44] S. Sundaram and C. Hadjicostis, "Distributed function calculation and consensus using linear iterative strategies," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 4, pp. 650–660, May 2008.

[45] W. H. Greene, *Econometric Analysis*, 7th ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2011.

[46] MOSEK, "The MOSEK Optimization Toolbox for MATLAB Manual, Version 7.0." 2013. [Online]. Available: https://www.mosek.com/

[47] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.

[48] R. Durrett, *Probability: Theory and Examples*, 4th ed. Cambridge, U.K.: Cambridge Univ. Press, 2010.

Yang Yang (S'09–M'13) received the B.S. degree from the School of Information Science and Engineering, Southeast University, Nanjing, China, in 2009, and the Ph.D. degree from the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong. From November 2013 to November 2015, he had been a Postdoctoral Research Associate at the Communication Systems Group, Darmstadt University of Technology, Darmstadt, Germany. He joined Intel Deutschland GmbH as a Research Scientist in December 2015.

His research interests include distributed solution methods in convex optimization, nonlinear programming, and game theory, with applications in communication networks, signal processing, and financial engineering.

Marius Pesavento (M'00) received the Dipl.-Ing. and M.Eng. degrees from Ruhr-Universität Bochum, Bochum, Germany, and McMaster University, Hamilton, ON, Canada, in 1999 and 2000, respectively, and the Dr.-Ing. degree in electrical engineering from Ruhr-Universität Bochum in 2005. Between 2005 and 2007, he was a Research Engineer at FAG Industrial Services GmbH, Aachen, Germany. From 2007 to 2009, he was the Director of the Signal Processing Section at MIMOon GmbH, Duisburg, Germany. In 2010, he became an Assistant Professor for Robust Signal Processing and a Full Professor for Communication Systems in 2013, Department of Electrical Engineering and Information Technology, Darmstadt University of Technology, Darmstadt, Germany. His research interests include robust signal processing and adaptive beamforming, high-resolution sensor array processing, multiantenna and multiuser communication systems, distributed, sparse, and mixed-integer optimization techniques for signal processing and communications, statistical signal processing, spectral analysis, and parameter estimation. He has received the 2003 ITG/VDE Best Paper Award, the 2005 Young Author Best Paper Award of the IEEE Transactions on Signal Processing, and the 2010 Best Paper Award of the CROWNCOM conference. He is a Member of the Editorial board of the EURASIP Signal Processing Journal, an Associate Editor for the IEEE Transactions on Signal Processing. He is currently serving the second term as a Member of the Sensor Array and Multichannel Technical Committee of the IEEE Signal Processing Society.

Mengyi Zhang (S'09–M'13) received the B.Sc. degree from the Department of Electronic Information, Huazhong University of Science and Technology, Wuhan, China, in 2009, and the Ph.D. degree from the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Engineering, Hong Kong, in 2013. She is currently a Postdoctoral Research Associate in the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong. Her research interests include statistical signal processing, optimization, and machine learning with applications in wireless communications and financial systems.

Daniel P. Palomar (S'99–M'03–SM'08–F'12) received the Electrical Engineering and Ph.D. degrees from the Technical University of Catalonia (UPC), Barcelona, Spain, in 1998 and 2003, respectively.

He is a Professor in the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology (HKUST), Hong Kong which he joined in 2006. Since 2013, he is a Fellow of the Institute for Advance Study at HKUST. He had previously held several research appointments, namely, at King's College London, London, U.K.; Stanford University, Stanford, CA, USA; Telecommunications Technological Center of Catalonia, Barcelona, Spain; Royal Institute of Technology, Stockholm, Sweden; University of Rome "La Sapienza," Rome, Italy; and Princeton University, Princeton, NJ, USA. His current research interests include applications of convex optimization theory, game theory, and variational inequality theory to financial systems, big data systems, and communication systems.

He has received the 2004/06 Fulbright Research Fellowship, the 2004 and 2015 (coauthor) Young Author Best Paper Awards by the IEEE Signal Processing Society, the 2002/03 best Ph.D. prize in Information Technologies and Communications by the UPC, the 2002/03 Rosina Ribalta first prize for the Best Doctoral Thesis in information technologies and communications by the Epson Foundation, and the 2004 prize for the best Doctoral Thesis in advanced mobile communications by the Vodafone Foundation.

He is a Guest Editor of the IEEE Journal of Selected Topics in Signal Processing2016 Special Issue on "Financial Signal Processing and Machine Learning for Electronic Trading" and has been an Associate Editor of IEEE Transactions on Information Theory and of IEEE Transactions on Signal Processing, a Guest Editor of the IEEE Signal Processing Magazine 2010 Special Issue on "Convex Optimization for Signal Processing," the IEEE Journal on Selected Areas in Communications 2008 Special Issue on "Game Theory in Communication Systems," and the IEEE Journal on Selected Areas in Communications 2007 Special Issue on "Optimization of MIMO Transceivers for Realistic Communication Networks."