POLITECNICO DI TORINO
Repository ISTITUZIONALE

Distributed ADMM for In-Network Reconstruction of Sparse Signals With Innovations

(Article begins on next page)

04 August 2020

Distributed ADMM for in-network reconstruction of sparse signals with innovations

Javier Matamoros[†], Sophie M. Fosson[⋆], Enrico Magli[⋆] and Carles Antón-Haro[†]

[†]Centre Tecnològic de Telecomunicacions de Catalunya, Barcelona (Spain)

[⋆] Department of Electronics and Telecommunications, Politecnico di Torino (Italy)

Abstract—In this paper, we tackle the in-network recovery of sparse signals with innovations. We assume that the nodes of the network measure a signal composed by a common component and an innovation, both sparse and unknown, according to the joint sparsity model 1 (JSM-1). Acquisition is performed as in compressed sensing, hence the number of measurements is reduced. Our goal is to show that distributed algorithms based on the alternating direction method of multipliers (ADMM) can be efficient in this framework to recover both the common and the individual components. Specifically, we define a suitable functional and we show that ADMM can be implemented to minimize it in a distributed way, leveraging local communication between nodes. Moreover, we develop a second version of the algorithm which requires only binary messaging, dramatically reducing the transmission load.

I. INTRODUCTION

In the last few years, an increasing attention has been paid to the implementation of the alternating direction method of multipliers (ADMM, [1, Chapter 2]) for distributed problems with sparsity constraints. On one hand, *sparsity* has many applications and its study has been boosted by the development of compressed sensing theory [2] in the last decade, which states that signals with few non-zero components can be recovered from a reduced number of measurements. On the other hand, the development of *distributed* algorithms has emerged due to the large diffusion of networked technologies, and specifically has addressed those sparse problems in which measurements are acquired by the nodes of a network [3]. Distributed compressed sensing [4] has then focused on different sparsity models and reconstruction methodologies, considering first centralized reconstruction at a fusion center that gathers all the data from the network, and lately tackling the more challenging in-network reconstruction problem [5]–[10].

In the framework of in-network reconstruction, the development of distributed versions of ADMM is very recent. Born in the 1970s, ADMM is not novel, but has been recently rediscovered thanks to its efficiency in large-scale computing systems. Its rationale is the merging of dual decomposition and augmented Lagrangian techniques to tackle convex optimization. In distributed optimization problems over networks, the use of ADMM combined with consensus

techniques has been largely studied in the last decade, see, e.g., [11], [12]. Concerning problems with sparsity constraints, an ADMM procedure has been presented in [5] to undertake a distributed basis pursuit problem, and then generalized to separable optimization problems [6]. Parsimonious in terms of communication steps, the distributed ADMM algorithm introduced in [5], named D-ADDM, has the drawback of requiring a coloring scheme for the underlying communication graph for synchronization purposes: the nodes with same color (i.e., nodes that are not neighbors) operate concurrently at same time, using information from the neighbors. Concerning general multi-agent problems, a distributed ADMM scheme has been also studied in [13], which is proved to converge and exhibit a faster convergence rate than state-of-the-art subgradient methods (specifically the error is $O\left(\frac{1}{t}\right)$ vs $O\left(\frac{1}{\sqrt{t}}\right)$, t being the iteration step). Again synchronization is an issue: this scheme requires a sequential update of the nodes. A further asynchronous version has been lately studied in [14]; in [15] the synchronization problem has been addressed for consensus optimization.

In [5], [6], [13], [14], [16], each node i of the network is associated with a local variable x_i and a function f_i, so that the functional f that one aims to minimize is separable and given by $f(x) = \sum_i f_i(x_i)$, where $x = (x_1, x_2, \dots)^T$. Moreover, the variables are coupled by a linear constraint $Ex = q$ where E and q are matrices of consistent dimensions. As in [5], this model can be used when the x_i's are local copies of a unique common signal, which implies that E imposes (local) consensus constraints.

The aim of this paper is to study a different and more general model, where different sparse signals with correlated support have to be in-network recovered. Specifically, we address the joint sparsity model 1 (JSM-1, [3], [4]), which assumes that each node has to recover a signal given by the sum of a component common to all the network nodes and an individual innovation component. Both components are sparse and unknown. From a mathematical viewpoint, each node is then associated with two variables, one for the common component (on which consensus constraints are expected), and one for the individual component. As we will see, this has some advantages in the ADMM iterative procedure in terms of synchronization: the two variables are updated sequentially, in such a way so that all the nodes can proceed in parallel, in contrast to [5], [13].

In practice, many applications can be represented using

JSM-1. For example, JSM-1 can describe a sensor network in which a common signal is measured, but different values are expected at each sensor, e.g., due to a shift in the spatial position or noise. The goal in this case may envisage both the reconstruction of the signal and the evaluation of the shift, which can be used, for example, for localization purposes. A more sophisticated example comes from distributed spectrum sensing in cooperative cognitive networks, whose aim is the evaluation of the spectrum occupancy through a network of collaborating cognitive radios [17], [18]. Sparsity can be exploited to build a suitable model for this problem, which actually consists in the estimation of a common component (due to the so-called primary users) and innovation components (due to secondary users). In [18, Section IV] a model very similar to JSM-1 was proposed, with an additional constraint of orthogonality between common component and innovations. A model with no innovations was instead considered in [17], where the problem was formulated as a distributed Lasso and tackled again via distributed ADMM. Sparsity was there assumed both in frequency and space.

Centralized reconstruction for JSM-1 has already been addressed in the literature. In [4] some asymptotic bounds were proved, and a single linear program algorithm was used for reconstruction, but its complexity was high. In [19], the Texas Hold' Em algorithm was used, which first estimates the common part by computing an average of the measurements (the innovation is considered as a noise with zero mean), and then exploits it to recover the innovations. This approach is guaranteed to work when the innovations are incoherent, and performance bounds are given in [19, Theorem 3.1]. In [20], [21] the JSM-1 problem is tackled assuming to have side information available, more precisely the decoder exactly knows one of the signals.

In this paper, in contrast, we study a *distributed* approach to JSM-1, which has been scarcely studied yet: to the best of our knowledge, the only relevant contributions are in the spectrum sensing framework [18]. As in [18], our approach is based on a Lasso reformulation of the problem that we tackle with ADMM techniques. We start by showing a centralized approach for reconstruction with ADMM, which turns out to be a simple case of classical ADMM implementation [1]; afterwards, we move to the distributed setting and propose a distributed ADMM, whose convergence is discussed using results from the recent work [22]. Further, we also propose a second version of distributed ADMM in which only binary messaging is required, which dramatically reduces the transmission load. We remark that the choice of considering ADMM for JSM-1 instead of trying to distribute algorithms like [4], [19]–[21], that are specific to JSM-1, has been motivated by ADMM efficiency, mathematical rigor, inclination to be distributed, and ease to extend to more general multi-agent contexts [13], [14]. These properties have spurred new interest towards ADMM, whose theory up to a few years ago was complete only in the standard case where the sum of two convex functions was considered. Results on the generalized case, where the sum of any number of convex functions is assumed (which is fundamental to work with networked systems), are very recent [13], [22].

The paper is organized as follows. In Section II, we formally present the model and the problem we undertake. Afterwards, we review the ADMM procedure (Section III) and explicitly derive the centralized ADMM for JSM-1 (Section IV). In Section V we introduce our distributed ADMM and discuss its convergence, and in Section VI we introduce a new version that only requires binary messaging. Section VII is then devoted to numerical results: convergence times for our distributed ADMM are analyzed and compared to the centralized algorithm, and performance results in terms of mean square error are shown and compared with Texas Hold'Em, which actually represents the state of the art for JSM-1 problems. Finally, some concluding remarks are collected in Section VIII.

A. Notation

Before proceeding let us introduce some notation. Given $x \in \mathbb{R}^n$, the ℓ_p-norm of x is denoted by $\|x\|_p$ for $p \in \mathbb{N}^+$, whereas $\|x\|_0$ gives the number of non-zero elements of x. The identity matrix of size $L \times L$ will be denoted by I_L. A $L \times N$ matrix with all entries zero will be denoted by $0_{L \times N}$. The L-length vectors with all zero entries and ones will be denoted by 0_L and 1_L, respectively. Notation $A \otimes B$ stands for the Kronecker product of matrices A and B. A graph $\mathcal{G}$ is defined as $\mathcal{G} := (\mathcal{N}, \mathcal{E})$ where $\mathcal{N}$ and $\mathcal{E}$ stand for the set of vertices and edges with cardinality $|\mathcal{N}|$ and $|\mathcal{E}|$ respectively.

II. SIGNAL MODEL

Consider a network composed of N nodes whose connectivity is described through the connected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$. Accordingly, node $i \in \mathcal{N}$ can communicate with node $j \in \mathcal{N}$ if the edge $\{i, j\}$ is included in $\mathcal{E}$, or, in other words, j belongs to the neighborhood set of i, denoted as $\mathcal{N}_i$ (see Figure 1).

In this scenario, each node observes a compressed version of a signal $\{x_i\}_{i \in \mathcal{N}} \in \mathbb{R}^L$ through a set of linear and local measurements, namely

$$y_i = A_i x_i + \eta_i \quad ; \quad i \in \mathcal{N}, \tag{1}$$

where $A_i \in \mathbb{R}^{M \times L}$ (with $M \ll L$) stands for the measurement matrix at the i-th node and $\eta_i \in \mathbb{R}^M$ for additive noise. We further assume that the observed signals follow the JSM-1 model [3], namely

$$x_i = \Psi z_c + \Phi z_i \quad ; \quad i \in \mathcal{N} \tag{2}$$

whith $\Psi, \Phi \in \mathbb{R}^{L \times L}$, and $z_c, z_i \in \mathbb{R}^L$. That is, the observed signal at each node is composed of a common component plus an innovation component, and we consider that these components are sparse in some domain, specified by the orthogonal bases Ψ and Φ. Hence, vectors z_c and $\{z_i\}$, which are both unknown and sparse, contain the signal coefficients in such domains, with the number of non-zero elements given by $k_c = \|z_c\|_0$ and $\{\|z_i\|_0\} = k_i$, respectively. As for the signal supports, defined as $\Omega_i := \{l | z_{i,l} \neq 0\}$ for $i \in \mathcal{N}$ and $\Omega_c := \{l | z_{c,l} \neq 0\}$, they do not necessarily
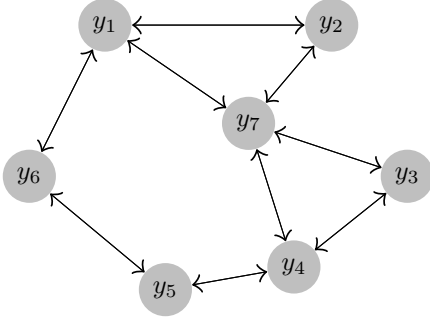
Figure 1. Example of 7 nodes with their respective observations connected according to a random graph.

Algorithm 1 Computation of $z_c(t+1), \{z_i(t+1)\}$

1: Initialize $z_c^\nu = 0_L$, $\{z_i^\nu\} = 0_L$ and $\nu = 0$
2: **while** $\|z_c^{\nu+1} - z_c^\nu\|_2^2 + \sum_{i=1}^N \|z_i^{\nu+1} - z_i^\nu\|_2^2 \leq \epsilon$ **do**
3:     $\nu \leftarrow \nu + 1$
4:     **for** $i = 1, \dots, N$ **do**
5:         $z_i^\nu \leftarrow \mathcal{S}_{\frac{\tau_1}{\rho}} \left[x_i(t+1) - z_c^\nu + \frac{\lambda_i(t)}{\rho} \right]$
6:     **end for**
7:     $z_c^\nu \leftarrow \mathcal{S}_{\frac{\tau_2}{\rho}} \left[\sum_{i=1}^N \frac{1}{N} \left(x_i(t+1) - z_i^\nu + \frac{\lambda_i(t)}{\rho} \right) \right]$
8: **end while**
9: $z_c(t+1) \leftarrow z_c^\nu$
10: $z_i(t+1) \leftarrow z_i^\nu$ for $i = 1, \dots, N$

---

coincide. Hereinafter, to ease the notation, we will consider[1] that $\Psi = \Phi = I_L$.

The ultimate goal is to reconstruct the triplets $\{x_i, z_c, z_i\}$ at each node in a distributed manner. To that end, we attempt to solve the following convex optimization problem:

$$\min_{\{x_i, z_i\}, z_c} \frac{1}{2} \sum_{i=1}^N \left(\|y_i - A_i x_i\|_2^2 + \tau_1 \|z_i\|_1 + \tau_2 \|z_c\|_1 \right) \quad (3)$$

$$\text{s.t.} \quad x_i = z_c + z_i; \quad i = 1, \dots, N, \quad (4)$$

with $\tau_1 > 0$ and $\tau_2 > 0$ denoting weights aimed to promote sparsity in the individual and common components, respectively.

III. ADMM REVIEW

In order to make the manuscript self-contained, this section briefly reviews the classical ADMM. For a detailed survey on this optimization method, the interested reader is referred to the seminal paper [1].

Let us consider the following structured convex optimization problem:

$$\min_{x,z} \quad f(x) + g(z) \quad (5)$$

$$\text{s.t.} \quad Gx + Bz - c = 0 \quad (6)$$

with variables $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$, $c \in \mathbb{R}^p$, matrices $G \in \mathbb{R}^{p \times n}$, $B \in \mathbb{R}^{p \times m}$ and convex functions $f(x) : \mathbb{R}^n \to \mathbb{R}$ and $g(z) :$

---

[1]However, the proposed algorithms also work for the case of different sparsity bases with minor modifications.

$\mathbb{R}^m \to \mathbb{R}$. In ADMM, the cost function is augmented by a quadratic term, thus yielding:

$$\min_{x,z} \quad f(x) + g(z) + \frac{\rho}{2} \|Ax + Bz - c\|_2^2 \quad (7)$$

$$\text{s.t.} \quad Gx + Bz - c = 0, \quad (8)$$

with $\rho > 0$. The augmented problem of (7)–(8) is still convex and its Lagrangian reads:

$$\mathcal{L}(x, z, \lambda) = f(x) + g(z) + \frac{\rho}{2} \|Ax + Bz - c\|_2^2$$
$$+ \lambda^T (Gx + Bz - c) \quad (9)$$

with $\lambda \in \mathbb{R}^p$ standing for the Lagrangian multipliers associated with the p constrains in (8). Essentially, ADMM consists in iterating the primal and dual variables of (9) as follows:

$$x(t+1) = \underset{x}{\arg\min} \mathcal{L}(x, z(t), \lambda(t)) \quad (10)$$

$$z(t+1) = \underset{z}{\arg\min} \mathcal{L}(x(t+1), z, \lambda(t)) \quad (11)$$

$$\lambda(t+1) = \lambda(t+1) + \rho (Gx(t+1) + Bz(t+1) - c) \quad (12)$$

As for the proof of convergence the interested reader is referred to [1, Appendix A].

IV. ADMM FOR JSM-1

Using the classical ADMM procedure described in the previous section, we now address the problem of centralized reconstruction in JSM-1. In the centralized scenario, it is assumed that sensors convey both their measurements $\{y_i\}$ and measurement matrices $\{A_i\}$ to a central coordinator where the triplets $\{x_i, z_c, z_i\}$ are reconstructed. As for the reconstruction method, we propose an ADMM algorithm, which, to the best of our knowledge, has not been yet addressed in the literature for JSM-1. We will use it as basis to develop our distributed schemes, which are our main purpose, and as benchmark to test them.

First, note that the problem of (3)–(4) can be rewritten as the standard problem of (5)–(6) by defining $x = [x_1^T, x_2^T, \dots x_N^T]^T$, $z = [z_c^T, z_1^T, z_2^T, \dots, z_N^T]^T$, $G = I_{N \cdot L}$, $c = 0_{N \cdot L}$ and matrix $B = [1_N \otimes I_L \quad I_N \otimes I_L]$. Therefore, following the rationale of (7), we can augment the cost function and obtain:

$$\min_{\{x_i, z_i\}, z_c} \frac{1}{2} \sum_{i=1}^N \left(\|y_i - A_i x_i\|_2^2 + \tau_1 \|z_i\|_1 + \tau_2 \|z_c\|_1 \right.$$
$$\left. + \frac{\rho}{2} \|x_i - z_i - z_c\|_2^2 \right) \quad (13)$$

$$\text{s.t.} \quad x_i = z_i + z_c; \quad i = 1, \dots, N \quad (14)$$

where ρ is a positive constant. Thus, the Lagrangian of the augmented problem reads:

$$\mathcal{L}(x, z; \lambda) = \frac{1}{2} \sum_{i=1}^N \|y_i - A_i x_i\|_2^2 + \sum_{i=1}^N \tau_1 \|z_i\|_1 + N\tau_2 \|z_c\|_1$$
$$+ \sum_{i=1}^N \frac{\rho}{2} \|x_i - z_i - z_c\|_2^2$$
$$+ \sum_{i=1}^N \lambda_i^T (x_i - z_i - z_c), \quad (15)$$

with $\{\lambda_i\} \in \mathbb{R}^n$ standing for the Lagrangian multipliers associated with the constraints in (14). Hence, according Section III, the ADMM iterates[2] are:

$$x_i(t+1) = (\rho I_L + A_i^T A_i)^{-1}(A_i^T y_i + \rho(z_i(t) + z_c(t)) - \lambda_i(t))$$

$$z_c(t+1), \{z_i(t+1)\} = \underset{z_c, \{z_i\}}{\arg\min} \mathcal{L}(x(t+1), z; \lambda(t))$$

$$\lambda_i(t+1) = \lambda_i(t) + \rho\left(x_i(t+1) - z_i(t+1) - z_c(t+1)\right). \tag{16}$$

As for the minimization step of (16), the solution must satisfy the following system of equations:

$$\partial_{z_i}\mathcal{L}(x(t+1), z; \lambda(t)) = 0_L; \quad i \in \mathcal{N}, \tag{17}$$

$$\partial_{z_c}\mathcal{L}(x(t+1), z; \lambda(t)) = 0_L, \tag{18}$$

with $\partial_x f$ denoting the subgradient of f with respect to x (see definition in [23]). From (17), the optimal variables $\{z_i(t+1), z_c(t+1)\}$ must satisfy:

$$\tau_1 s_i - \rho(x_i(t+1)$$
$$- z_i(t+1) - z_c(t+1)) - \lambda_i(t) = 0_L \tag{19}$$

for $i \in \mathcal{N}$. In the equation above, the L-length vector s_i stands for the subgradient of $\|z_i\|_1$ evaluated at $z_i(t+1)$ and its components are $s_{i,l} = 1$ if $z_{i,l}(t+1) > 0$, $s_{i,l} = -1$ for $z_{i,l}(t+1) < 0$ and

$$s_{i,l} = \frac{\rho}{\tau_1}(x_{i,l}(t+1) - z_{c,l}(t+1)) - \frac{\lambda_{i,l}(t)}{\tau_1} \in (-1, 1) \tag{20}$$

for $z_{i,l}(t+1) = 0$. From all the above and assuming $z_c(t+1)$ to be known, $z_i(t+1)$ reads

$$z_i(t+1) = \mathcal{S}_{\frac{\tau_1}{\rho}}\left[x_i(t+1) - z_c(t+1) + \frac{\lambda_i(t)}{\rho}\right], \tag{21}$$

with $\mathcal{S}_\alpha : \mathbb{R}^n \to \mathbb{R}^n$ standing for the component-wise soft-thresholding operator. That is, for any $x \in \mathbb{R}$ and $\alpha > 0$

$$S_\alpha(x) = \begin{cases} x - \alpha & \text{if } x > \alpha \\ x + \alpha & \text{if } x < -\alpha \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

As for the common component $z_c(t+1)$, let s_c be the subgradient of $\|z_c\|_1$ evaluated at $z_c(t+1)$ with entries given by $s_{c,l} = 1$ if $z_{c,l}(t+1) > 0$, $s_{c,l} = -1$ for $z_{c,l}(t+1) < 0$ and $s_{c,l} \in (-1, 1)$ for $z_{c,l}(t+1) = 0$ for $l = 1, \ldots, L$. Accordingly, we have that

$$\partial_{z_c}\mathcal{L}(x(t+1), z; \lambda(t)) = N\tau_2 s_c - \sum_{i=1}^{N} \lambda_i(t) \tag{23}$$

$$- \sum_{i=1}^{N} \rho(x_i(t+1) - z_i(t+1) - z_c(t+1)) \tag{24}$$

Hence, assuming $\{z_i(t+1)\}$ known, $z_c(t+1)$ is given by:

$$z_c(t+1) = \mathcal{S}_{\frac{\tau_2}{\rho}}\left[\sum_{i=1}^{N} \frac{1}{N}\left(x_i(t+1) - z_i(t+1) + \frac{\lambda_i(t)}{\rho}\right)\right]. \tag{25}$$

[2] Note that matrix $(\rho I + A_i^T A_i)^{-1}$ has to be computed only once. To lower the computational complexity, one may also resort to the matrix inversion lemma.

Bearing all the above in mind, we propose a coordinate descent method to find the set of $z_c(t+1), \{z_i(t+1)\}$. Essentially, we alternate minimizations with respect to $z_c(t+1)$ and $\{z_i\}$ until convergence. This procedure is summarized in Algorithm 1.

Proposition 1. *The coordinate descent method proposed in Algorithm 1 converges to the stationary solution of* (17)–(18).

Proof. The minimization of the augmented Lagrangian with respect to $z_c(t+1)$ and $\{z_i(t+1)\}$ turns out to be the minimization of a composite convex objective function, i.e. the sum of a smooth convex term and separable nonsmooth terms (ℓ_1-norms). In this case, the block coordinate descent method is known to converge to the optimal solution (see [24] for further details). $\square$

Besides, note that from (24) and (20), the stationary condition of (18) can be rewritten as follows:

$$s_c = \frac{\tau_1}{N\tau_2} \sum_{i=1}^{N} s_i \tag{26}$$

From (26), we conclude that the condition $\tau_2 \leq \tau_1$ must hold. To see this, consider the opposite case, i.e. when $\tau_2 > \tau_1$, and assume that $s_{i,l} = 1$ for all $l = 1, \ldots, L$. Then, from (26), we have that $s_{c,l} > 1$ which contradicts the definition of $s_{c,l}$ since $s_{c,l} \in [-1, 1]$. Finally, we remark that the procedure proposed in this section is based on the standard framework of ADMM, therefore the following convergence result is guaranteed.

Proposition 2. *The iterative procedure* (16) *(with Algorithm 1) converges to the optimal operating point.*

Proof. The proof can be deduced by the classical ADMM proof of convergence discussed in [1, Appendix A]. $\square$

V. DISTRIBUTED ADMM FOR JSM-1

This section goes one step beyond Section IV and attempts to find a *distributed* reconstruction method. To that end, we propose to solve the following optimization problem:

$$\min_{\{x_i, z_i, \zeta_i, c_i\}} \frac{1}{2} \sum_{i=1}^{N} \|y_i - A_i x_i\|_2^2 + \tau_1 \|z_i\|_1 + \tau_2 \|\zeta_i\|_1 \tag{27}$$

$$\text{s.t.} \quad x_i = z_i + \zeta_i; \quad i \in \mathcal{N} \tag{28}$$

$$\zeta_i = c_j; \quad j \in \tilde{\mathcal{N}}_i \tag{29}$$

with $\tilde{\mathcal{N}}_i = \mathcal{N}_i \cup i$. Here, we have introduced the local variables $\{\zeta_i\}, \{c_i\}$ that must be interpreted as the local and neighbors guesses on the common component. The consensus constraint of (29) and the fact that $\mathcal{G}$ is a connected graph make the problem above still equivalent to (3). In order to solve (27)–

(29), we resort to the generalized ADMM of [22] and build the following augmented cost function:

$$\min_{\{x_i, z_i, \zeta_i, c_i\}} \frac{1}{2} \sum_{i=1}^{N} \left\{ \|y_i - A_i x_i\|_2^2 + \tau_1 \|z_i\|_1 + \tau_2 \|\zeta_i\|_1 \right.$$

$$\left. + \frac{\rho}{2} \|x_i - z_i - \zeta_i\|_2^2 + \frac{\theta}{2} \sum_{j \in \tilde{\mathcal{N}}_i} \|\zeta_i - c_j\|_2^2 \right\} \tag{30}$$

$$\text{s.t.} \quad x_i = z_i + \zeta_i; \quad i \in \mathcal{N} \tag{31}$$

$$\zeta_i = c_j; \quad j \in \tilde{\mathcal{N}}_i \tag{32}$$

with ρ and θ standing for positive constants. As in the centralized approach (see Section IV), we write the Lagrangian:

$$\mathcal{L}(x, z, \zeta; \lambda, \mu) = \frac{1}{2} \sum_{i=1}^{N} \left\{ \|y_i - A_i x_i\|_2^2 + \tau_1 \|z_i\|_1 + \tau_2 \|\zeta_i\|_1 \right.$$

$$+ \frac{\rho}{2} \|x_i - z_i - \zeta_i\|_2^2 + \frac{\theta}{2} \sum_{j \in \tilde{\mathcal{N}}_i} \|\zeta_i - c_j\|_2^2$$

$$+ \lambda_i^T (x_i - z_i - \zeta_i)$$

$$\left. + \sum_{j \in \tilde{\mathcal{N}}_i} \mu_{i,j}^T (\zeta_i - c_j) \right\}. \tag{33}$$

Now, in an attempt to find a distributed solution to (30), we propose to sequentially update the primal variables $\{x_i, z_i, \zeta_i, c_i\}$ according to

$$x_i(t+1) = (\rho I_L + A_i^T A_i)^{-1} (A_i^T y_i + \rho(z_i(t) + \zeta_i(t)) - \lambda_i^T)$$

$$z_i(t+1) = \mathcal{S}_{\frac{\tau_1}{\rho}} \left[(x_i(t+1) - \zeta_i(t)) + \frac{\lambda_i(t)}{\rho} \right]$$

$$\zeta_i(t+1) = \mathcal{S}_{\frac{\tau_2}{\rho + \theta|\tilde{\mathcal{N}}_i|}} \left[\frac{1}{\rho + \theta|\tilde{\mathcal{N}}_i|} \left(\rho(x_i(t+1) - z_i(t+1)) \right. \right.$$

$$\left. \left. + \theta \sum_{j \in \tilde{\mathcal{N}}_i} \left(c_j(t) - \frac{\mu_{i,j}(t)}{\theta} \right) + \lambda_i(t) \right) \right] \tag{34}$$

$$c_i(t+1) = \frac{1}{|\tilde{\mathcal{N}}_i|} \sum_{j : i \in \tilde{\mathcal{N}}_j} \left(\zeta_j(t+1) + \frac{\mu_{j,i}(t)}{\theta} \right); \tag{35}$$

followed by the ascent updates of the dual variables, that is,

$$\lambda_i(t+1) = \lambda_i(t) + \alpha (x_i(t+1) - z_i(t+1) - \zeta_i(t+1))$$

$$\mu_{i,j}(t+1) = \mu_{i,j}(t) + \kappa (\zeta_i(t+1) - c_j(t+1)); \quad j \in \tilde{\mathcal{N}}_i,$$

where $\{\lambda_i\}$ and $\{\mu_{i,j}\}$ are the Lagrangian multipliers associated with constraints (31) and (32), respectively, and α and κ are positive constants[3]. Interestingly, this iterative method can be readily implemented in a distributed manner by exchanging information among neighbor nodes only. The proposed distributed ADMM for JSM-1 (referred to in the sequel as DADMM), is summarized in Algorithm 2. Regarding the convergence, applying the arguments in [22], we can prove the following result.

Proposition 3. *Algorithm 2 converges to an optimal solution.*

[3] In practice, we set $\alpha = \rho$ and $\kappa = \beta$.

Proof. The reader is referred to Appendix A for the proof. $\square$

Interestingly, Algorithm 2 requires $\tau_2 \leq \tau_1$ to achieve cooperation among the nodes. To see that, assume that at iteration $t = 1$ we have $z_{i,l}(t+1) > 0$ (the same reasoning applies for $z_{i,l}(t+1) < 0$ too). In this case, we have in step 8 of Algorithm 2 that $\zeta_i(t+1) = \mathcal{S}_{\frac{\tau_2}{\rho + \theta|\tilde{\mathcal{N}}_i|}} \left[\frac{\tau_1}{\rho + \theta|\tilde{\mathcal{N}}_i|} \right]$ which, according to the definition of the soft-thresholding operator, yields 0 if $\tau_2 > \tau_1$. Then, if $\tau_2 > \tau_1$, by iterating the algorithm one can observe that $\zeta_{i,l}(t) = c_{i,l}(t) = 0$ for any $t > 0$, which means that each sensor performs the reconstruction in an isolated manner.

Algorithm 2 Distributed ADMM (DADMM) for JSM-1

1: **for all** $i \in \mathcal{N}$ **do**
2:     Initialize variables:
        $x_i(1) = 0_L; z_i(1) = 0_L; \zeta_i(1) = 0_L; c_i(1) = 0_L$ and $\lambda_i = 0_L$
3: **end for**
4: **for** $t = 1, \ldots, T_{\max}$ **do**
5:     **for all** $i \in \mathcal{N}$ **do**
6:         $x_i(t+1) \leftarrow (\rho I + A_i^T A_i)^{-1} (A_i^T y_i + \rho(z_i(t) + \zeta_i(t)) - \lambda_i(t)^T)$
7:         $z_i(t+1) \leftarrow \mathcal{S}_{\frac{\tau_1}{\rho}} \left[(x_i(t+1) - \zeta_i(t)) + \frac{\lambda_i(t)}{\rho} \right]$
8:         $\zeta_i(t+1) \leftarrow \mathcal{S}_{\frac{\tau_2}{\rho + \theta|\tilde{\mathcal{N}}_i|}} \left[\frac{1}{\rho + \theta|\tilde{\mathcal{N}}_i|} \left(\rho(x_i(t+1) - z_i(t+1)) + \theta \sum_{j \in \mathcal{N}_i} \left(c_j(t) - \frac{\mu_{i,j}(t)}{\theta} \right) + \lambda_i(t) \right) \right]$
9:         Broadcast $\zeta_i(t+1)$ to each node j with $j \in \tilde{\mathcal{N}}_i$
10:         $c_i(t+1) \leftarrow \frac{1}{|\tilde{\mathcal{N}}_i|} \sum_{j : i \in \tilde{\mathcal{N}}_j} \left(\zeta_j(t+1) + \frac{\mu_{j,i}(t)}{\theta} \right)$
11:         Broadcast $c_i(t+1)$ to each node j with $j \in \tilde{\mathcal{N}}_i$
12:         $\lambda_i(t+1) \leftarrow \lambda_i(t) + \rho (x_i(t+1) - z_i(t+1) - \zeta_i(t+1))$
13:         **for all** $j \in \tilde{\mathcal{N}}_i$ **do**
14:             $\mu_{i,j}(t+1) \leftarrow \mu_{i,j}(t+1) + \theta (\zeta_i(t+1) - c_j(t+1))$
15:         **end for**
16:         **for all** $j : i \in \tilde{\mathcal{N}}_j$ **do**
17:             $\mu_{j,i}(t+1) \leftarrow \mu_{j,i}(t) + \theta (\zeta_j(t+1) - c_i(t+1))$
18:         **end for**
19:     **end for**
20: **end for**

VI. DISTRIBUTED ADMM WITH 1 BIT MESSAGES

The main drawback of the proposed DADMM, and the distributed algorithms in general (i.e. [13]), scheme is the large amount of information that needs to be exchanged among neighboring nodes [i.e. $\zeta_i(t+1)$ and $c_i(t+1)$] in each iteration). This in turn results in a large energy consumption and reduced network lifetime. To circumvent that, we propose to quantize the exchanged variables with 1 bit only. In order to retain most of the advantages of the scheme, we replace the exact minimization updates of primal variables (34) and (35) (steps 8 and 10 in Algorithm 2) by approximate minimization updates based on the subgradient method, that is

$$\zeta_i(t+1) = \zeta_i(t) - \epsilon \, \text{sign}\left(g_{\zeta_i^t} \right) \tag{36}$$

$$c_i(t+1) = c_i(t) - \epsilon \, \text{sign}\left(g_{c_i^t} \right), \tag{37}$$

where ϵ denotes the step length, $g_{\zeta_i^t}$ and $g_{c_i^t}$ stand for the subgradient of the augmented Lagrangian with respect to ζ_i and c_i at time t, and $\text{sign}(x)$ is defined component-wise as $\text{sign}(x) = 1$ if $x \geq 0$ and $\text{sign}(x) = -1$ otherwise. Consequently, in DADMM-1bit nodes only need to broadcast the sign of the innovations, namely $\text{sign}(g_{\zeta_i^t})$ and $\text{sign}(g_{c_i^t})$, in steps 9 and 11 of Algorithm 2.

As for the computation of $g_{\zeta_i^t}$, note that

$$g_{\zeta_i^t} := \tau_2 s - \rho(x_i(t+1) - z_i(t+1) - \zeta_i(t)) - \lambda_i(t) + \sum_{j \in \tilde{\mathcal{N}}_i} \theta\left(\zeta_i(t) - c_j(t)\right) + \mu_{i,j}(t), \qquad (38)$$

where the L-length vector s stands for the subgradient of $\|\zeta_i(t)\|_1$. Similarly, for $g_{c_i^t}$ we have that

$$g_{c_i^t} := -\theta \sum_{j : i \in \tilde{\mathcal{N}}_j} \left(\zeta_j(t) - c_i(t) + \frac{\mu_{j,i}(t)}{\theta}\right). \qquad (39)$$

VII. Numerical Results

In the simulations, we consider signals $\{x_i\}$ of length $L = 100$ with sparsity levels $k_i + k_c = 10$. The supports of the common and innovation signals $\{z_c, z_i\}$ are generated uniformly at random, with non-zero elements drawn from a standard Gaussian distribution. As performance metric, we adopt the normalized mean square error, which for a generic k-sparse signal x is defined as

$$\text{MSE}(x) = \frac{1}{k\sigma_x^2}\|\hat{x} - x\|_2^2$$

with $\hat{x}$ standing for the reconstruction of x and σ_x^2 for the average power of the non-zero values.

First, Figure 2 compares, in the case of noiseless measurements, the centralized ADMM with the Combined Community Texas Hold 'Em algorithm, Texas in short, proposed in [25]. In particular, performances are measured by the average MSE incurred in the reconstruction of $\{x_i, z_i, z_c\}$ when the sparsity level of the common component varies, but the total sparsity level is kept constant, i.e. $k_i + k_c = 10$. Numerical results reveal that ADMM outperforms Texas Algorithm in all cases. This is more evident for low values of k_c, where the iterative procedure of the proposed ADMM that jointly reconstructs the common and innovation signals pays off. On the contrary, for larger values of k_c, Texas performance approaches that of ADMM. This follows from the fact that, in this region, the common signal may be reconstructed quite accurately by simply treating the innovation components as additive noise. As for the noisy scenario, Figure 3 reports the attained MSE in the reconstruction of $\{x_i, z_i, z_c\}$ as a function of the SNR in the measurements. Interestingly, Texas is able to reconstruct the *common* signal more accurately than ADMM when the SNR is low. However, it is worth noting that ADMM exhibits a lower MSE in the reconstruction of $\{x_i\}$ for the whole range of SNR.

Next, Figure 4 depicts the attained MSE for the three proposed ADMM reconstruction methods (DADMM-1bit is tested for different values of ϵ defined in (36)). In this setting, we have considered $M = 25$ and a regular graph with

degree $d = 5$ for the distributed cases. Unsurprisingly, the centralized approach converges much faster than its distributed counterparts. Still, both the DADMM and the DADMM-1bit with $\epsilon = 0.01$ also achieve perfect reconstruction. For DADMM-1bit, we observe that ϵ impacts on the accuracy of the estimates and on the convergence speed: when ϵ increases, the algorithm converges faster at the price of less accurate estimation. Besides, this also explains the MSE oscillations for large values of ϵ. More interestingly, for small values, like $\epsilon = 0.01$, DADMM-1bit performs virtually identical to DADMM at the expense of 3 times more iterations to converge. From a signalling viewpoint this is still favorable: if, for instance, real values can be quantized over 16 bits, the signalling ratio is 3/16.

In an scenario with a reduced number of measurements (i.e. $M = 20$), Figure 5 shows the attained MSE in the reconstruction of the individual (z_i) and common signals (z_c). Again, all exhibit an identical performance after convergence. Interestingly, all algorithms achieve perfect reconstruction of the common component z_c thanks to the redundancy in the number of node measurements but are unable to reconstruct the innovations. To illustrate this point, we show in Figure 6 the MSE in reconstruction of the common and innovation components as a function of the number of measurements per sensor. As it can be observed, 18 measurements per sensor may suffice for an accurate reconstruction of the common signal. On the contrary, for an acceptable reconstruction quality on the innovation components one may need more than 24 measurements per sensor. Next, in Figure 7, we compare the DADMM for JSM-1 to a general distributed ADMM strategy, called in the sequel DADMM-c. DADMM-c first aims to achieve consensus on the common signal by considering the innovation components as noise[4]. After this consensus step, each sensors performs the reconstruction of its innovation component independently by an ADMM reconstruction method. Figure 7 shows the average MSE versus the sparsity level of the common signal. As we can observe, DADMM outperforms DADMM-c for the whole range of k_c. This result proves that the jointly reconstruction of the common and innovation components pays off.

Finally, Figure 8 shows the attained MSE versus SNR experienced in the sensors measurements for the distributed ADMM algorithms proposed in this paper. Numerical results reveal that DADMM is more robust to noisy measurements than its 1-bit counterpart.

VIII. Conclusions

In this paper we have addressed the problem of in-network reconstruction of jointly sparse signals with innovations. In this scenario, the signal model accounts for a certain structure on the sensor observations. That is, each sensor signal turns out to be the combination of a common sparse signal plus an innovation sparse component. As for the reconstruction method, we have proposed to solve a Lasso-type problem by means of ADMM due to its efficiency and fast convergence

---

[4]Note that this can be done by setting $z_i(t+1) = 0$ in Step 7 of Algorithm 2.
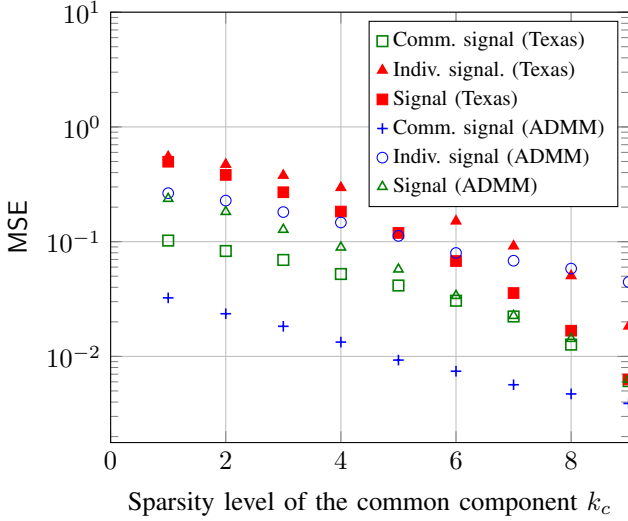
Figure 2. MSE in the reconstruction of $\{z_i\}$, z_c vs the sparsity level of the common component k_c ($N = 20$, $M = 25$, $k_i = 10 - k_c$, $L = 100$, $d = 5$, $\tau_1 = 3e - 2$, $\tau_2 = 1e - 2$).
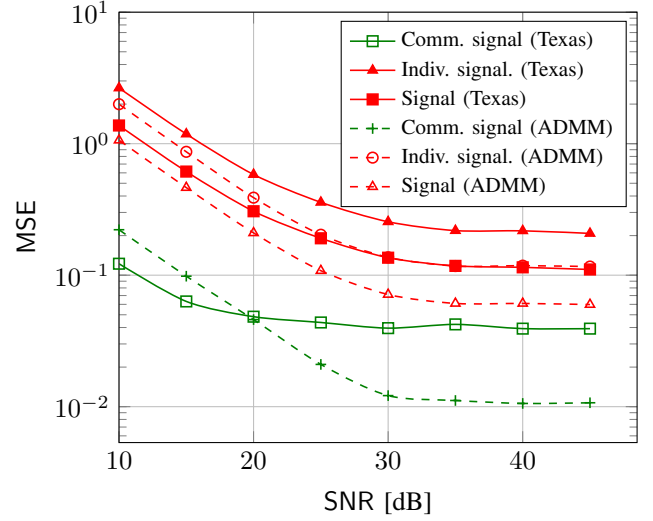

Figure 3. MSE in the (centralized) reconstruction of $\{z_i\}$, z_c vs SNR of the measurements ($N = 20$, $M = 25$, $k_c = k_i = 5$, $L = 100$, $d = 5$, $\tau_1 = 3e - 2$, $\tau_2 = 1e - 2$, $\rho = 0.01$).

properties. First, we have addressed the centralized scenario, where all sensor measurements are collected at a fusion center. In this case, we have derived an ADMM algorithm that jointly reconstructs the common and innovation signals. The proposed algorithm, which is based on the classical framework of ADMM, is shown to converge to the optimal operating point. Next, we have addressed the in-network reconstruction scenario, in which nodes must reconstruct their observed signals distributedly and without the aid of a fusion center. In this setting, the Lasso formulation has been modified accordingly and a distributed ADMM scheme has been proposed. To our best knowledge, this is the first distributed algorithm proposed for JSM-1. In particular, the distributed ADMM can be cast as the generalized ADMM, in which there exists more than two primal minimization blocks per algorithm iteration. Then, by standing on recent results on the generalized ADMM, we have proved that our solution satisfies the necessary conditions for convergence. Next, we have proposed modified version of the distributed ADMM, which merely requires the exchange of binary messages between neighboring nodes. Numerical results have revealed that the centralized ADMM outperforms other state-of-the-art methods and that the two distributed ADMM schemes converge to the centralized ADMM solution in a reasonable number of iterations. Interestingly, the 1-bit version is shown to reduce the total number of transmitted bits.

APPENDIX
CONVERGENCE OF DADMM

To prove the convergence of Algorithm 2, we stand on the results of [22]. To that end, we first pose the optimization problem of (27)–(29) as in [22] and then show that our problem satisfy the assumptions [22, Assumption A.(a)-(f), Corollary 3.1] for convergence.
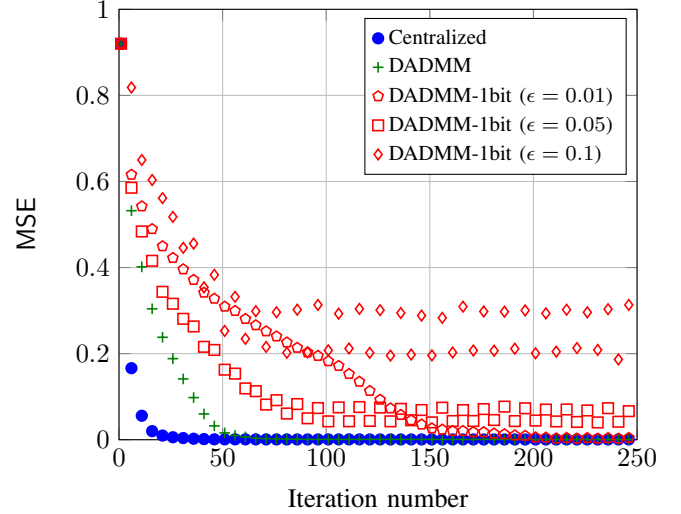

Figure 4. MSE in the reconstruction of $\{x_i\}$ vs iteration number ($N = 20$, $M = 25$, $k_i = k_c = 5$, $L = 100$, $d = 5$, $\tau_1 = 3 \cdot 10^{-3}$, $\tau_2 = 6 \cdot 10^{-4}$, $\rho = \theta = 0.01$).

Define vectors $x = \left[x_1^T, x_2^T, \ldots, x_N^T\right]^T$, $z = \left[z_1^T, z_2^T, \ldots, z_N^T\right]^T$, $\zeta = \left[\zeta_1^T, \zeta_2^T, \ldots, \zeta_N^T\right]^T$, $c = \left[c_1^T, c_2^T, \ldots, c_N^T\right]^T$ and functions

$$f_1(x) = \sum_{i=1}^{N} \|y_i - A_i x_i\|_2^2$$

$$f_2(z) = \tau_1 \sum_{i=1}^{N} \|z_i\|_1$$

$$f_3(\zeta) = \tau_2 \sum_{i=1}^{N} \|\zeta_i\|_1 \qquad (40)$$

$$f_4(c) = 0$$

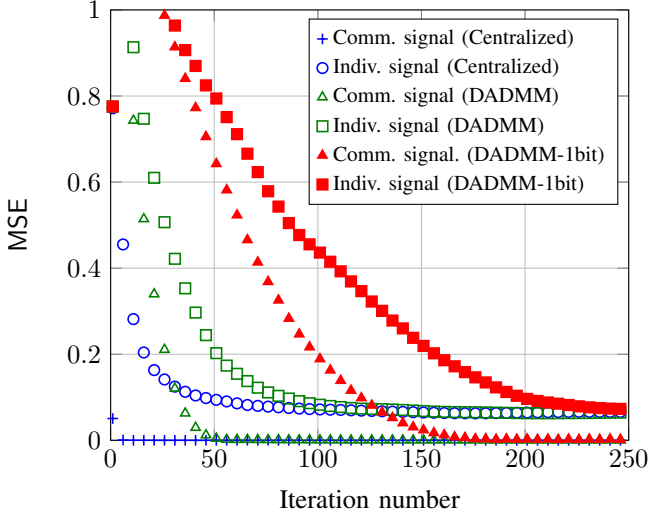Then, the problem in (27)–(29) can be conveniently rewrit-

Figure 5. MSE in the reconstruction of $\{z_i\}$ and z_c vs iteration number ($N = 20$, $M = 20$, $k_i = k_c = 5$, $L = 100$, $d = 5$, $\tau_1 = 3 \cdot 10^{-3}$, $\tau_2 = 6 \cdot 10^{-4}$, $\rho = \theta = 0.01$).
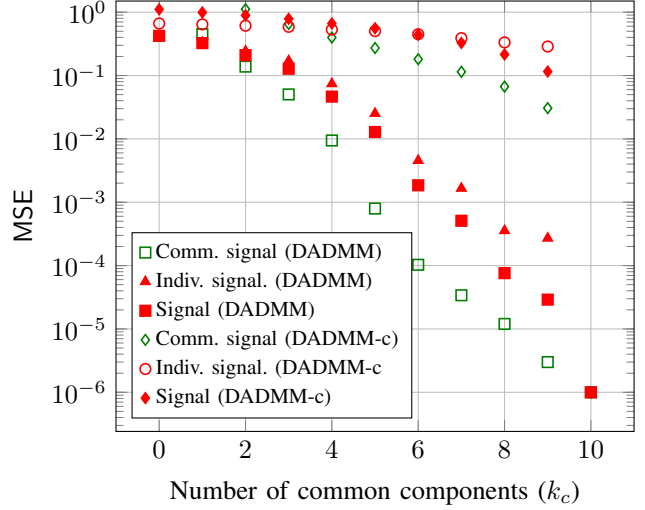

Figure 7. MSE in the (distributed) reconstruction of $\{z_i\}$, z_c vs k_c ($N = 20$, $M = 25$, $d = 5$ $L = 100$, $d = 5$, $k_i + k_c = 10$, $\tau_1 = 3e - 2$, $\tau_2 = 1e - 4$, $\rho = \theta = 0.01$).
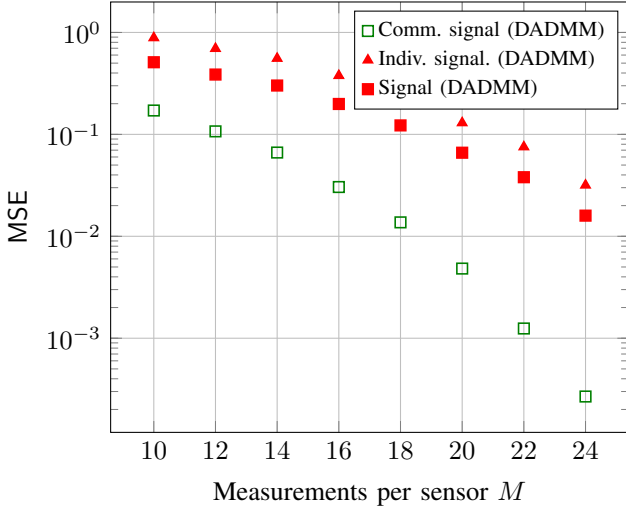

Figure 6. MSE attained by DADMM in the reconstruction of $\{z_i\}$, z_c and x_i vs the number of measurements per sensor M ($N = 20$, $M = 25$, $k_i = 10 - k_c$, $L = 100$, $d = 5$, $\tau_1 = 3e - 2$, $\tau_2 = 1e - 2$).
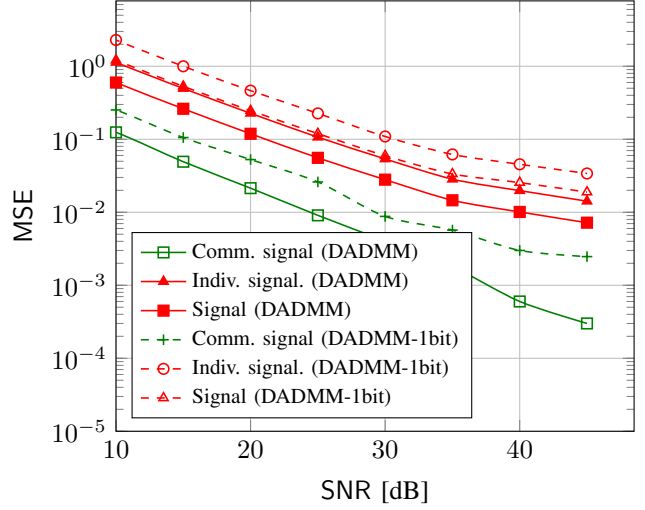

Figure 8. MSE in the (distributed) reconstruction of $\{z_i\}$, z_c vs SNR of the measurements ($N = 20$, $M = 25$, $d = 5$ $L = 100$, $d = 5$, $\tau_1 = 3e - 2$, $\tau_2 = 1e - 2$, $\rho = \theta = 0.01$, $\epsilon = 0.01$).

ten as follows:

$$\min_{\{x,z,\zeta,c\}} f_1(x) + f_2(z) + f_3(\zeta) + f_4(c) \tag{41}$$

$$\text{s.t.} \quad E_1 x + E_2 z + E_3 \zeta + E_4 c = 0_{NL+N(d+1)} \tag{42}$$

where, for the sake of notation, we have considered that $\mathcal{G}$ is a regular graph of degree d. Matrices E_1, E_2, E_3 and E_4 are defined as follows:

$$E_1 = \begin{bmatrix} \tilde{E}_1^T & 0_{NL(d+1) \times NL}^T \end{bmatrix}^T$$

$$E_2 = \begin{bmatrix} \tilde{E}_2^T & 0_{NL(d+1) \times NL}^T \end{bmatrix}^T$$

$$E_3 = \begin{bmatrix} \tilde{E}_3^T & \check{E}_3^T \end{bmatrix}^T$$

$$E_4 = \begin{bmatrix} 0_{NL \times NL}^T & B_1^T & \cdots & B_N^T \end{bmatrix}^T$$

where $\tilde{E}_1$ is $N \times N$ block diagonal matrix with diagonal blocks given by the identity matrix I_L. Similarly, $\tilde{E}_2 = \tilde{E}_3 = -\tilde{E}_1$ and

$$\check{E}_3^T = I_N \otimes \left(1_{(d+1)} \otimes I_L \right) \tag{43}$$

Finally, B_i is defined in blocks as follows:

$$B_i = \begin{bmatrix} \tilde{B}_{1,1}^{(i)} & \cdots & \tilde{B}_{1,N}^{(i)} \\ \vdots & \ddots & \vdots \\ \tilde{B}_{d+1,1}^{(i)} & \cdots & \tilde{B}_{d+1,N}^{(i)} \end{bmatrix} \tag{44}$$

with

$$\tilde{B}_{k,j}^{(i)} = \begin{cases} -I_L & \text{if } j \in \tilde{\mathcal{N}}_i \setminus \{0, \ldots, j-1\} \\ 0_{L \times L} & \text{otherwise} \end{cases} \tag{45}$$

With the equivalent formulation of (27)–(42) we only need to prove that our problem satisfy the following set of assumptions (see [22, Assumption A.(a)-(f), Corollary 3.1])

a) *The global minimum of (27)–(29) exists and can be attained.*

   This naturally comes from the construction of the problem.

b) *Let $\tilde{x}_1 = x$, $\tilde{x}_2 = z$, $\tilde{x}_3 = \zeta$, $\tilde{x}_4 = c$. For any $k \in \{1, 2, 3, 4\}$ the functions (40) are decomposable as follows:*

$$f_k(\tilde{x}_k) = g_k(\tilde{A}_k \tilde{x}_k) + h_k(\tilde{x}_k) \qquad (46)$$

   where g_k and h_k are both convex and continuous over their domains and $\tilde{A}_k$ are some matrices of proper dimensions.

   In our setting, on one hand, we have that $h_1(\tilde{x}_1) = 0$, $h_2(\tilde{x}_2) = f_2(\tilde{x}_2)$, $h_3(\tilde{x}_3) = f_3(\tilde{x}_3)$, and $h_4(\tilde{x}_4) = 0$. On the other, $g_k(\tilde{A}_k \tilde{x}_k) = 0$ for $k = 2, 3, 4$ and $g_1(\tilde{A}_1 \tilde{x}_1) = f_1(\tilde{x}_1)$ with $\tilde{A}_1$ being block-diagonal matrix, i.e. $\tilde{A}_1 = \text{diag}\,[A_1, A_2, \dots, A_N]$. Consequently, our functions satisfy this assumption.

c) *Each g_k is strictly convex and continuously differentiable over its domain with a uniform Lipschitz continuous gradient.*

   In our case, we only need to prove that the ℓ_2 norm of the Hessian of g_1 is bounded[5], that is,

$$\|\nabla^2 g_1(\tilde{A}x)\|_2 = \|\tilde{A}^T \tilde{A}\|_2 \le \kappa$$

   for some positive constant κ.

d) *Functions $h_k(\tilde{x}_k)$, $k \in \{1, 2, 3, 4\}$ are polyhedral functions.*

   To show this, note that functions h_2 and h_3 are both the sum of polyhedral functions (ℓ_1 norms) and, hence, from [26, Proposition 5.1.8], they are polyhedral too. Besides, functions h_1 and h_4 are particular cases of a linear function so they are also polyhedral.

e) *For any fixed and finite λ, μ and $\xi > 0$, $\sum_k h_k(\tilde{x}_k)$ if finite for all $\tilde{x} \in \{\tilde{x} : \mathcal{L}(\tilde{x}; \lambda, \mu) \le \xi\}$.*

   It is easy to check that our $\mathcal{L}(\tilde{x}; \lambda, \mu)$ is proper, that is, fixed $\xi > 0$, the levet set $\mathcal{L}(\tilde{x}; \lambda, \mu) \le \xi$ is bounded, say, $\tilde{x}$ can not tend to infinity. This implies that the h_k's are finite by their definition.

f) *Each submatrix E_k has full rank.*

   This fact comes from the definition of matrices $\{E_k\}$.

g) *The feasible set is polyhedral and the sequence of the primal-dual iterates lies in a compact set (see [22, Corollary 3.1]).*

   Since we are considering Euclidean spaces with usual Euclidean distances, we know that the iterates lie in a compact set whenever they are bounded. We then prove their boundedness. By Assumption a), the solution of problem (27)-(29) exists. Let us call such optimal point $\omega^\star = \{x_i^\star, z_i^\star, \zeta_i^\star, \lambda_i^\star, \mu_{i,j}^\star\}_{i \in \mathcal{N}}$. By the theory of the Lagrange multipliers, $\omega^\star$ is a stationary point for Algorithm 2.

Given this, if Algorithm 2 is non-expansive, then the iterates are bounded[6]. In fact, starting from any $X(0)$ with $\|X(0) - \omega^\star\| = \mathrm{d} < +\infty$, then the non-expansivity implies that $\|X(t) - \omega^\star\| \le \mathrm{d}$ for any $t \in \mathbb{N}$, assuming that the sequence $X(t)$ is updated according to Algorithm 2. In other words, $X(t)$ remains anchored to $\omega^\star$ and, thus, is bounded.

To prove the boundedness of the iterates, it is then sufficient to show their non-expansivity. To this purpose, we recall that the soft thresholding operator is non-expansive [27] and that $\left\|\rho I_L + A_i^T A_i)^{-1}\right\|_2 = \frac{1}{\rho}$ (which can be obtained via simple linear algebra, since the minimum eigenvalue of $A_i^T A_i$ is 0). These considerations, along with the assumption that the graph is regular, are sufficient to show the non-expansivity of each primal-dual update (computations are omitted for brevity.)

REFERENCES

[1] 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, 2010.

[2] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, pp. 1289–1306, 2006.

[3] M. F. Duarte, S. Sarvotham, D. Baron, M. B. Wakin, and R. G. Baraniuk, "Distributed compressed sensing of jointly sparse signals," in *Proc. 39th Asilomar Conf. Signals, Systems and Computers, Pacific Grove, CA*, 2005.

[4] D. Baron, M. F. Duarte, S. Sarvotham, M. B. Wakin, and R. G. Baraniuk, "An information theoretic approach to distributed compressed sensing," in *Allerton Conference on Communication, Control, and Computing*, Allerton, IL, 2005.

[5] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Püschel, "Distributed basis pursuit," *IEEE Trans. on Signal Processing*, vol. 60, no. 4, pp. 1942–1956, 2012.

[6] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Püschel, "D-ADMM: A communication-efficient distributed algorithm for separableoptimization," *IEEE Trans. on Signal Proc.*, vol. 61, no. 10, pp. 2718–2723, 2013.

[7] S. Patterson, Y. Eldar, and I. Keidar, "Distributed compressed sensing for static and time-varying networks," *IEEE Trans. Signal Process.*, vol. 62, no. 19, pp. 4931–4946, 2014.

[8] D. Sundman, S. Chatterjee, and M. Skoglund, "Distributed greedy pursuit algorithms," *submitted to EURASIP Signal Processing*, 2013.

[9] C. Ravazzi, S. M. Fosson, and E. Magli, "Distributed soft thresholding for sparse signal recovery," in *Globecom*, 2013.

[10] S. Fosson, J. Matamoros, C. Anton-Haro, and E. Magli, "Distributed support detection of jointly sparse signals," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[11] I. Schizas, A. Ribeiro, and G. Giannakis, "Consensus in ad hoc wsns with noisy links part i: Distributed estimation of deterministic signals," *IEEE Trans. Signal Process.*, vol. 56, no. 1, pp. 350–364, 2008.

[12] H. Zhu, G. Giannakis, and A. Cano, "Distributed in-network channel decoding," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 3970–3983, 2009.

[13] E. Wei and A. Ozdaglar, "Distributed alternating direction method of multipliers," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, 2012, pp. 5445–5450.

[14] ——, "On the $o(1/k)$ convergence of asynchronous distributed alternating direction method of multipliers," in *Global Conference on Signal and Information Processing (GlobalSIP), 2013 IEEE*, 2013, pp. 551–554.

[15] R. Zhang and J. Kwok, "Asynchronous distributed admm for consensus optimization," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, T. Jebara and E. P. Xing, Eds.   JMLR Workshop and Conference Proceedings, 2014, pp. 1701–1709.

---

[5]Note that according to [22] the strongly convex part may be absent, which is the case for f_1 and f_4

[6]by non-expansive, we mean that applying the update to two points with distance d, we obtain two points with distance $\le$ d

[16] Q. Ling, M. Tao, W. Yin, and X. Yuan, "A multi-block alternating direction method with parallel splitting for decentralized consensus optimization," *EURASIP J. Wireless Comm. and Netw.*, vol. 2012, no. 1, pp. 1–12, 2012.

[17] J. A. Bazerque and G. B. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1847–1862, 2010.

[18] F. Zeng, C. Li, and Z. Tian, "Distributed compressive spectrum sensing in cooperative multihop cognitive networks," *IEEE J. Sel. Top. Sign. Proces.*, vol. 5, no. 1, pp. 37–48, 2011.

[19] S. R. Schnelle, J. N. Laska, C. Hegde, M. F. Duarte, M. A. Davenport, and R. G. Baraniuk, "Texas Hold 'Em algorithms for distributed compressive sensing," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Dallas, TX, 2010.

[20] G. Coluccia, E. Magli, A. Roumy, and V. Toto-Zarasoa, "Lossy compression of distributed sparse sources: a practical scheme," in *2011 European Signal Processing Conference (EUSIPCO-2011)*, 2011, pp. 422–426.

[21] D. Valsesia, G. Coluccia, and E. Magli, "Joint recovery algorithms using difference of innovations for distributed compressed sensing," in *Signals, Systems and Computers, 2013 Asilomar Conference on*, 2013, pp. 414–417.

[22] M. Hong and Z.-Q. Luo, "On the linear convergence of the alternating direction method of multipliers," *preprint*, 2013, http://arxiv.org/abs/1208.3922.

[23] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 1993.

[24] P. Richtárik and M. Takáč, "Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function," *Mathematical Programming*, vol. 144, pp. 1–38, 2014.

[25] S. Schnelle, J. Laska, C. Hegde, M. Duarte, M. Davenport, and R. Baraniuk, "Texas hold 'em algorithms for distributed compressive sensing," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010, pp. 2886–2889.

[26] J. M. Borwein and A. S. Lewis, *Convex Analysis and Nonlinear Optimization, Theory and Examples*. CMS Books in Mathematics, 2006.

[27] M. Fornasier, "Numerical methods for sparse recovery," in *Theoretical Foundations and Numerical Methods for Sparse Recovery*, M. Fornasier, Ed. Radon Series Comp. Appl. Math., de Gruyter, 2010, pp. 93–200.