Efficient Target Estimation in Distributed MIMO Radar via the ADMM

Bo Li and Athina Petropulu
ECE Department
Rutgers, The State University of New Jersey, NJ, 08854
Email: {paul.bo.li, athinap}@rutgers.edu

***Abstract*— We consider the problem of target estimation in distributed MIMO radars that employ compressive sensing. The problem is formulated as a sparse signal recovery problem with magnitude constraints on the target reflection coefficients, where the signal to be recovered consist of equal size blocks that have the same sparsity profile. A solution is proposed based on the alternating direction method of multipliers (ADMM), which significantly lowers the computational complexity of sparse recovery and improves the estimation accuracy. Due to the block diagonal structure of the sensing matrix, the iterations of all ADMM subproblems are amenable to parallel implementation, which can reduce the running time. A semi-distributed implementation, which relaxes the need of a powerful fusion center is also discussed.**

***Index Terms*— Distributed MIMO radar, sparse sensing, ADMM.**

I. Introduction

Multiple-input multiple-output (MIMO) radars [1]–[4] have received considerable attention in recent years due to their improved performance over traditional phased arrays radar systems. Distributed MIMO radars are a special class of MIMO radars in which the antennas are widely separated. The multiple independent paths between the targets and the transmit/receive antennas introduce spatial diversity, thus contributing to the improved target estimation performance of distributed MIMO radars. The introduction of compressive sensing (CS) [5] in distributed MIMO radars has been shown to enable good performance with fewer data [6]–[8]. In CS based distributed MIMO radars, the target estimation problem is formulated as a sparse vector recovery problem, where the nonzero elements of the sparse vector indicate the presence of targets in the discretized target space. By grouping together the columns of the sensing matrix corresponding to the same grid point, the sparse vector becomes group sparse (also known as block sparse) [6]–[8], which can be exploited to achieve improved target estimation and further reduction of the number of measurements needed; this was validated in [6], [7] via simulations, while the theoretical justification was provided in [8].

The recovery of the block-sparse vector can be obtained via Block Orthogonal Matching Pursuit (BOMP) [6], or Group Lasso with proximal gradient algorithm (GLasso) [7], or mixed ℓ_1/ℓ_2 norm optimization (L-OPT) with interior point method [8]. GLasso and L-OPT achieve better estimation performance than BOMP but involve higher computational complexity and require careful tuning of manually chosen parameters. By exploiting the block diagonal structure in the sensing matrix, [7] proposed to decouple the original problem into smaller size problems, which reduces the complexity of the recovery process. However, the scheme of [7] does not take advantage of the identical sparsity profiles of the sub-blocks in the target vector to improve estimation performance.

This paper considers the same scenario as in [6]–[8]. We formulate the target estimation problem as a sparse signal recovery problem, with the signal to be recovered consisting of equal-size blocks of the same sparsity profile, and with magnitude constraints on the target reflection coefficients. By introducing auxiliary variables, we express the objective of the optimization problem as a function of three variables. The alternating direction method of multipliers (ADMM) is applied to obtain the optimal solution by alternately optimizing over these three variables. Each of the ADMM subproblems in one iteration either has a closed form solution, or can be solved fast. This significantly lowers the computational complexity as compared to algorithms based on interior point method. In addition, the assumed prior information on the magnitude of target reflection coefficients allows for improved estimation accuracy. The iterations of all ADMM subproblems are amenable to parallel implementation, which allows for reduction of running time. The parallel implementation here is different from the decoupled scheme of [7], because here, the identical sparsity profile in the sub-blocks of the target vector is utilized via an auxiliary variable. We also discuss a semi-distributed implementation of the solution, in which the computations are distributed among all the receive nodes thus obviating the need of a powerful fusion center. Simulations validate the efficiency of the proposed algorithm and show that the proposed algorithm is robust over a wide range of a manually chosen parameter.

The paper is organized as follows. The sparse model for distributed MIMO radar system is presented in Section II. In Section III, a constrained optimization problem is proposed for the target estimation, which is efficiently solved via ADMM based iterations. Parallel and semi-decentralized implementation schemes are discussed in Section IV. Simulation results are given in Section V, and conclusions are presented in

This work was supported by ONR under Grant N00014-12-1-0036 and the ECE Department of Rutgers.

Section VI.

II. Signal Model

We consider a MIMO radar system with M_t transmit nodes (TX) and M_r receive nodes (RX) that are widely separated. Let (x_i^t, y_i^t) and (x_i^r, y_i^r) denote the locations of the ith transmit and receive antennas in cartesian coordinates, respectively. The ith TX antenna transmits the modulated waveform $x_i(t)e^{j2\pi f_i t}$, where f_i is the carrier frequency.

Let us assume that there are K stationary point targets present in the space. For simplicity, we consider a clutter-free environment. To exploit the spatial sparsity of the targets, the location space is discretized on the grid $\mathbf{\Theta} = \{(x_n, y_n), n = 1, \ldots, N\}$. The received baseband signal $z_{ij}(t)$ can be rewritten as a linear combination of the signal reflected from all grid points, i.e.,

$$z_{ij}(t) = \sum_{n=1}^{N} s_{ij}^n x_i(t - \tau_{ij}^n) + n_{ij}(t) \tag{1}$$

where $n_{ij}(t)$ is the additive complex Gaussian noise; τ_{ij}^n and s_{ij}^n represent the propagation time and reflection coefficient associated with the n-th grid point and the transmit/receive antennas (TX/RX) pair (i, j), respectively. It holds that

$$\begin{aligned} \tau_{ij}^n &= (d_{in}^t + d_{jn}^r)/c, \\ d_{in}^{t/r} &= \sqrt{(x_i^{t/r} - x_n)^2 + (y_i^{t/r} - y_n)^2}. \end{aligned} \tag{2}$$

where c is the propagation speed, and (x_n, y_n) is the coordinate of the n-th grid point. If there is a target located on the n-th grid point, then s_{ij}^n is nonzero and models the combined effects of target attenuation and fading during propagation; otherwise s_{ij}^n is zero. In this paper, we consider complex attenuation factors with magnitude less than ω_0, which means $|s_{ij}^n| \in [0, \omega_0] \triangleq \Omega$. Such prior can be obtained, for example, based on the distance between the region of interest and the TX/RX pairs. This magnitude prior will be used later to improve estimation accuracy.

On letting L denote the number of T_s-spaced samples from the continuous waveforms, model (1) can be written into vector form as

$$\mathbf{z}_{ij} = \mathbf{\Psi}_{ij}\mathbf{s}_{ij} + \mathbf{n}_{ij} \tag{3}$$

where $\mathbf{z}_{ij} = [z_{ij}(t_0 + 0T_s), \ldots, z_{ij}(t_0 + (L-1)T_s)]^T$, $\mathbf{n}_{ij} = [n_{ij}(t_0 + 0T_s), \ldots, n_{ij}(t_0 + (L-1)T_s)]^T$, $\mathbf{s}_{ij} = \left[s_{ij}^1, \ldots, s_{ij}^N\right]^T$ and

$$\mathbf{\Psi}_{ij} = \begin{bmatrix} x_i(t_0+0T_s-\tau_{ij}^1) & \cdots & x_i(t_0+0T_s-\tau_{ij}^N) \\ \vdots & \ddots & \vdots \\ x_i(t_0+(L-1)T_s-\tau_{ij}^1) & \cdots & x_i(t_0+(L-1)T_s-\tau_{ij}^N) \end{bmatrix}_{L\times N} \tag{4}$$

with t_0 being the sampling start time.

Let us define $\mathbf{\Psi} = \text{diag}(\mathbf{\Psi}_{11}, \ldots, \mathbf{\Psi}_{M_tM_r})$. We can stack the received samples from all the pairs of transmit and receive antennas into a column vector $\mathbf{z}$ of length LM_tM_r,

$$\mathbf{z} = \left[(\mathbf{z}_{11})^T, \ldots, (\mathbf{z}_{M_tM_r})^T\right]^T = \mathbf{\Psi}\mathbf{s} + \mathbf{n}, \tag{5}$$

where $\mathbf{s} = \left[(\mathbf{s}_{11})^T, \ldots, (\mathbf{s}_{M_tM_r})^T\right]^T$ and $\mathbf{n} = [(\mathbf{n}_{11})^T, \ldots, (\mathbf{n}_{M_tM_r})^T]^T$. Note that for every (i, j) pair, the entries of vector $\mathbf{s}_{ij}$ are zero except those corresponding to grid points occupied by targets. Thus, vector $\mathbf{s}$ exhibits group sparsity: it is composed by M_tM_r sub-blocks, which share the same sparsity profile, and each block has exactly K nonzero entries. In addition, all the entries have magnitude in the range $\Omega \triangleq [0, \omega_0]$.

III. Fast Signal Recovery based on ADMM

A. Reformulation with real variables

Note that $\mathbf{z}, \mathbf{s}$ and $\mathbf{n}$ are all complex vectors. We can easily obtain the equivalent reformulation with real variables as follows

$$\underbrace{\begin{bmatrix} \Re\{\mathbf{z}\} \\ \Im\{\mathbf{z}\} \end{bmatrix}}_{\tilde{\mathbf{z}}} = \underbrace{\begin{bmatrix} \mathbf{\Psi} & \\ & \mathbf{\Psi} \end{bmatrix}}_{\tilde{\mathbf{\Psi}}} \underbrace{\begin{bmatrix} \Re\{\mathbf{s}\} \\ \Im\{\mathbf{s}\} \end{bmatrix}}_{\tilde{\mathbf{s}}} + \underbrace{\begin{bmatrix} \Re\{\mathbf{n}\} \\ \Im\{\mathbf{n}\} \end{bmatrix}}_{\tilde{\mathbf{n}}} \tag{6}$$

where $\Re\{\cdot\}$ and $\Im\{\cdot\}$ denote respectively the real and imaginary parts of a complex variable. It is clear that $\tilde{\mathbf{\Psi}}$ is still block diagonal and $\tilde{\mathbf{s}}$ is composed by $2M_tM_r$ sub-blocks that share the same sparsity profile and have exactly K nonzero real entries.

We can recover the sparse target vector $\tilde{\mathbf{s}}$, and thus $\mathbf{s}$, by solving the optimization problem

$$\begin{aligned} &\min \frac{1}{2}\|\tilde{\mathbf{z}} - \tilde{\mathbf{\Psi}}\tilde{\mathbf{s}}\|_2^2 + \lambda \sum_{n=1}^{N} \|\tilde{\mathbf{s}}[\mathcal{I}_n]\|_2 \\ &s.t.\ \tilde{\mathbf{s}} \in \Omega^{2NM_tM_r} \end{aligned} \tag{7}$$

where λ is the regularization parameter, and the set $\mathcal{I}_n, \forall n \in \mathbb{N}_N^+$ has cardinality $2M_tM_r$ and contains the indices of the n-th entries of all equal-length sub-blocks $\tilde{\mathbf{s}}_{11}, \ldots, \tilde{\mathbf{s}}_{2M_tM_r}$. The constraint $\tilde{\mathbf{s}} \in \Omega^{2NM_tM_r}$ is satisfied if $\|\left[\tilde{\mathbf{s}}[i], \tilde{\mathbf{s}}[i + NM_tM_r]\right]\|_2 \in \Omega,\ \forall i \in \mathbb{N}_{NM_tM_r}^+$, where $\tilde{\mathbf{s}}[i]$ denotes the i-th entry of $\tilde{\mathbf{s}}$. The above optimization problem is convex and can be solved by interior point methods. However, such methods would involve high complexity. In the following, we use the alternating direction method of multipliers (ADMM) [9] to solve the problem at reduced complexity.

B. Fast algorithm based on ADMM

We introduce auxiliary variables $\mathbf{y}$ and $\mathbf{x}$ and rewrite (7) equivalently as

$$\begin{aligned} &\min \frac{1}{2}\|\tilde{\mathbf{z}} - \tilde{\mathbf{\Psi}}\tilde{\mathbf{s}}\|_2^2 + \sum_{n=1}^{N} \lambda\|\mathbf{y}_n\|_2 \\ &s.t.\ \mathbf{y}_n = \mathbf{D}_n\tilde{\mathbf{s}},\ \forall n \in \mathbb{N}_N^+; \\ &\quad \mathbf{x} = \tilde{\mathbf{s}}, \quad \mathbf{x} \in \Omega^{2NM_tM_r} \end{aligned} \tag{8}$$

where $\mathbf{D}_n$ is an $(2M_tM_r) \times (2NM_tM_r)$ matrix that selects the entries of $\tilde{\mathbf{s}}$ indexed by $\mathcal{I}_n$; the vector $\mathbf{y}$ is defined as $[\mathbf{y}_1^T, \ldots, \mathbf{y}_N^T]^T$. We have $\mathbf{y} = \mathbf{D}\tilde{\mathbf{s}}$ where $\mathbf{D} = [\mathbf{D}_1^T, \ldots, \mathbf{D}_N^T]$ permutates $\tilde{\mathbf{s}}$ into $\mathbf{y}$. The auxiliary variable $\mathbf{y}$ is used to isolate $\tilde{\mathbf{s}}$ from the group sparsity-inducing term $\sum\|\cdot\|_2$; the magnitude constraint is now imposed on $\mathbf{x}$ instead of $\tilde{\mathbf{s}}$. Now ADMM is

applicable if we group the variables into two blocks, i.e., $(\mathbf{y}, \mathbf{x})$ and $\tilde{\mathbf{s}}$. In the following, we show that the ADMM subproblems can be solved fast.

The augmented Lagrangian of the above optimization problem can be written as follows

$$\mathcal{L}(\tilde{\mathbf{s}}, \mathbf{y}, \mathbf{x}; \mu, \nu) = \frac{1}{2}\|\tilde{\mathbf{z}} - \tilde{\mathbf{\Psi}}\tilde{\mathbf{s}}\|_2^2 + \nu^T(\mathbf{x} - \tilde{\mathbf{s}}) + \frac{\rho_2}{2}\|\mathbf{x} - \tilde{\mathbf{s}}\|_2^2 + \sum_{n=1}^{N}\left(\lambda\|\mathbf{y}_n\|_2 + \mu_n^T(\mathbf{y}_n - \mathbf{D}_n\tilde{\mathbf{s}}) + \frac{\rho_1}{2}\|\mathbf{y}_n - \mathbf{D}_n\tilde{\mathbf{s}}\|_2^2\right) \tag{9}$$

where $\rho_1, \rho_2 > 0$ and $\mu \triangleq [\mu_1^T, \ldots, \mu_N^T]^T \in \mathbb{R}^{2NM_tM_r}$ and $\nu \in \mathbb{R}^{2NM_tM_r}$ are the Lagrangian multipliers.

Based on the framework of ADMM, we can solve (8) by the following iterations

$$\mathbf{y}^{k+1} = \arg\min_{\mathbf{y}} \mathcal{L}(\tilde{\mathbf{s}}^k, \mathbf{y}, \mathbf{x}^k; \mu^k, \nu^k), \tag{10a}$$

$$\mathbf{x}^{k+1} = \arg\min_{\mathbf{x}\in\Omega^{NM_tM_r}} \mathcal{L}(\tilde{\mathbf{s}}^k, \mathbf{y}^{k+1}, \mathbf{x}; \mu^k, \nu^k), \tag{10b}$$

$$\tilde{\mathbf{s}}^{k+1} = \arg\min_{\tilde{\mathbf{s}}} \mathcal{L}(\tilde{\mathbf{s}}, \mathbf{y}^{k+1}, \mathbf{x}^{k+1}; \mu^k, \nu^k), \tag{10c}$$

$$\nu^{k+1} = \nu^k + \rho_2(\mathbf{x}^{k+1} - \tilde{\mathbf{s}}^{k+1}), \tag{10d}$$

$$\mu^{k+1} = \mu^k + \rho_1(\mathbf{y}^{k+1} - \mathbf{D}\tilde{\mathbf{s}}^{k+1}). \tag{10e}$$

The $\mathbf{y}$-subproblem is well studied in the literature [10] and its solution is given explicitly by the shrinkage operator

$$\mathbf{y}_n^{k+1} = \max\left\{\|\bar{\mathbf{s}}_n^k\|_2 - \frac{\lambda}{\rho_1}, 0\right\}\frac{\bar{\mathbf{s}}_n^k}{\|\bar{\mathbf{s}}_n^k\|_2}, \ \forall n \in \mathbb{N}_N^+. \tag{11}$$

where $\bar{\mathbf{s}}_n^k = \mathbf{D}_n\tilde{\mathbf{s}}^k - \mu_n^k/\rho_1$. In total, the computation cost of (11) scales as $\mathcal{O}(NM_tM_r)$.

For the $\mathbf{x}$-subproblem, we have

$$\mathbf{x}^{k+1} = \mathcal{P}_\Omega\left(\tilde{\mathbf{s}}^{k+1} - \frac{\nu^k}{\rho_2}\right) \tag{12}$$

where $\mathcal{P}_\Omega(\mathbf{x})$ denotes the projection of $\big(\mathbf{x}[i], \mathbf{x}[i+NM_tM_r]\big)$ onto a circle with radius ω_0 for all $i \in \mathbb{N}_{NM_tM_r}^+$. The computation of this projection involves $\mathcal{O}(NM_tM_r)$ operations.

The minimizer of the objective function is found as

$$0 \in \frac{\partial}{\partial\tilde{\mathbf{s}}}\mathcal{L}(\tilde{\mathbf{s}}, \mathbf{y}^{k+1}, \mathbf{x}^{k+1}; \mu^k, \nu^k) = \mathbf{A}\tilde{\mathbf{s}} - \mathbf{b}^k \tag{13}$$

where $\mathbf{A} = \tilde{\mathbf{\Psi}}^T\tilde{\mathbf{\Psi}} + (\rho_1 + \rho_2)\mathbf{I}_{2NM_tM_r}$ and $\mathbf{b}^k = \tilde{\mathbf{\Psi}}^T\tilde{\mathbf{z}} + \mathbf{D}^T\mu^k + \rho_1\mathbf{D}^T\mathbf{y}^{k+1} + \nu^k + \rho_2\mathbf{x}^{k+1}$, and $\mathbf{I}_M$ denotes an identity matrix of dimension $M \times M$. The solution can be obtained by solving the following system of linear equations

$$\mathbf{A}\tilde{\mathbf{s}}^{k+1} = \mathbf{b}^k. \tag{14}$$

Given the signal model, $\mathbf{A}$ is fixed for all iterations. The computational effort for $\mathbf{b}^k$ in each iteration only involves permutation and addition of vectors; this is because $\tilde{\mathbf{\Psi}}^T\tilde{\mathbf{z}}$ is also fixed. In addition, $\mathbf{A}$ is block diagonal because $\tilde{\mathbf{\Psi}}^T\tilde{\mathbf{\Psi}}$ is block diagonal. System (14) can be written into a set of subsystems of linear equations as follows

$$\mathbf{A}_m\tilde{\mathbf{s}}_m^{k+1} = \mathbf{b}_m^k, \ \forall m \in \mathbb{N}_{2M_tM_r}^+, \tag{15}$$

where $\mathbf{A}_m$ denotes the m-th diagonal block of matrix $\mathbf{A}$; $\mathbf{v}_m$ denotes the m-th uniformly partitioned block of vector $\mathbf{v}$. From the definition of $\mathbf{A}$, we know that

$$\mathbf{A}_m = \begin{cases} \mathbf{\Psi}_{ij}^T\mathbf{\Psi}_{ij} + (\rho_1 + \rho_2)\mathbf{I}_N & \text{if } m \in [1, M_tM_r] \\ \mathbf{A}_{m-M_tM_r} & \text{otherwise} \end{cases} \tag{16}$$

where $j = \lfloor\frac{m-1}{M_t}\rfloor + 1$ and $i = m - (j-1)M_t$. $\lfloor a\rfloor$ denotes the largest integer that is smaller than a. As observed in [8], $\mathbf{\Psi}_{ij}^T\mathbf{\Psi}_{ij}$ is symmetric with very small off-diagonal entries. Therefore, $\mathbf{A}_m$ is guaranteed to be strictly diagonal dominant and symmetric for any $\rho_1, \rho_2 > 0$. Each system in (15) can be solved efficiently with cost at most $\mathcal{O}(N^2)$. The total number of operations to solve (14) is of the order of $\mathcal{O}(N^2M_tM_r)$.

Finally, the update for multipliers μ and ν can be carried out as in (10) with linear complexity.

The convergence of the above iterations is guaranteed by results in the ADMM literature [9]. The iterations stop when the decrease of the objective value in (7) drops below a certain threshold, or when the number of iterations reaches a certain value.

The advantages of the proposed algorithm can be summarized as follows. First, the computational cost is low. As we know, solving (8) using an interior point method would involve $\mathcal{O}((NM_tM_r)^3)$ operations [11]. For the proposed algorithm, the computational cost in each iteration is dominated by solving the system of linear equations (14), which is $\mathcal{O}(N^2M_tM_r)$. The reduction of computations is more significant as the number of antennas increases. Second, the estimation accuracy of $\mathbf{s}$ is improved by introducing the amplitude constraints on the sparse target vector. Also, the performance is robust over wide range of regularization parameter λ. This is validated via simulations in Section V. Lastly, due to the block diagonal structure in $\mathbf{\Psi}$, the update of $\tilde{\mathbf{s}}^{k+1}$ in (14) can be achieved by updating independent sub-blocks in $\tilde{\mathbf{s}}^{k+1}$. The good separability in the update of all variables facilitates the parallel implementation and the decentralized scheme as discussed in the following section.

IV. Implementation Schemes and Discussions

A. Parallel Implementation

In the $(k+1)$th iteration, it is clear that all pairs $(\mathbf{x}^k[i], \mathbf{x}^k[i+NM_tM_r])$ in $\mathbf{x}^k$ are updated independent of the others, thus, the computations can be done in parallel. Similar parallelism scheme applies to μ^k and ν^k, and the update of $\mathbf{y}_n^k$. The subsystems in (15) can also be solved in parallel. Assuming that there are multiple computing units available at the fusion center, the target estimation can be significantly reduced.

B. Fusion Center Aided Semi-Distributed Implementation

The ADMM based approach described in Section III requires a fusion center to perform the computations. However, a semi-distributed implementation is also possible, as described in the following.

$\mathbf{x}$ (respectively for $\mathbf{s}$ and ν) can be divided into blocks, each of which can be updated locally at one receive antenna.

However, the update of $\mathbf{y}$ and μ cannot be done locally. One receive node would have to be assigned to act as a fusion center, and perform the update of $\mathbf{y}$ and μ. The computational requirements of that fusion center would be much lower than those of the fusion center of the Section III.

The fusion center aided semi-distributed scheme is summarized in Algorithm 1. In the implementation of Algorithm 1, $\tilde{\mathbf{s}}_m^{k+1} \in \mathbb{R}^N$ (respectively for $\nu_m^{k+1}, \mathbf{x}_m^{k+1}$), $m \in \mathbb{N}_{2M_tM_r}^+$, denotes the m-th block of the uniformly partitioned $\tilde{\mathbf{s}}^{k+1}$. $\mathbf{y}_m^{k+1} \in \mathbb{R}^N$ denotes the m-th block of the uniformly partitioned $\mathbf{D}^T\mathbf{y}^{k+1}$. The receive node j updates $\mathbf{x}_m^{k+1}, \nu_m^{k+1}$ and $\mathbf{s}_m^{k+1}$ for all $m \in \mathcal{T}_j \triangleq \{(j-1)M_t + i, M_tM_r + (j-1)M_t + i | i \in \mathbb{N}_{M_t}^+\}$. The fusion center updates $\mathbf{y}$ and μ. Thus, the computation cost is $\mathcal{O}(N^2M_t)$ at each node and $\mathcal{O}(NM_tM_r)$ at the fusion center. We can see that the computations are distributed among all receive nodes. The computation and memory required at the fusion center are only linear with the dimension of sparse vector $\mathbf{y}$. In each iteration, each node communicates $\tilde{\mathbf{s}}_m^{k+1}$ and $\mathbf{y}_m^{k+1}$ to the fusion center. After a few iterations, the nonzero entries in $\tilde{\mathbf{s}}_m^{k+1}$ and $\mathbf{y}_m^{k+1}$ would be of the order of $\mathcal{O}(K)$.

A fully distributed scheme would also be possible, but would require consensus; consensus-based implementations converge slowly, which would be a problem in target estimation and tracking applications.

Algorithm 1 Semi-Distributed Implementation

One peer receive node is chosen as the fusion center.
Input $\tilde{\mathbf{\Psi}}, \tilde{\mathbf{z}}, \lambda, \rho_1, \rho_2$
Initialization $\tilde{\mathbf{s}}^{(0)} = \mathbf{x}^{(0)} = \mathbf{y}^{(0)} = 0$, $\mu^{(0)} = \nu^{(0)} = 0$
Iteration
Fusion Center:
  compute $\mathbf{y}_n^{k+1}$, $n \in \mathbb{N}_N^+$ by (11);
Node $j \in \mathbb{N}_{M_r}^+$: for all $m \in \mathcal{T}_j$
  download $\mathbf{y}_m^{k+1}$ from the fusion center;
  compute $[\mathbf{x}_m^{k+1}; \mathbf{x}_{m+M_tM_r}^{k+1}] = \mathcal{P}_\Omega([\hat{\mathbf{x}}_m^{k+1}; \hat{\mathbf{x}}_{m+M_tM_r}^{k+1}])$
    where $\hat{\mathbf{x}}_m^{k+1} = \tilde{\mathbf{s}}_m^{k+1} - \nu_m^k/\rho_2$;
  compute $\tilde{\mathbf{s}}_m^{k+1}$ by solving (15);
  compute $\nu_m^{k+1} = \nu_m^k + \rho_2(\mathbf{x}_m^{k+1} - \tilde{\mathbf{s}}_m^{k+1})$;
  upload $\tilde{\mathbf{s}}_m^{k+1}$ to the fusion center;
Fusion Center:
  compute μ^{k+1} by (10e);

V. Numerical Results

In this section, we evaluate the performance of the proposed algorithm, described in Section III. We consider a MIMO radar system with 4 TX and 4 RX antennas, distributed uniformly on a circle of radius of $6,000m$ and $3,000m$, respectively. Each TX radar transmits pulses with repetition interval $0.125ms$ and carrier frequency of 5GHz. Each RX radar operates with sampling frequency 5MHz. The signal-to-noise ratio (SNR) is defined as $10\log_{10}(\sigma_0^2/\sigma_n^2)$, and is set to be 5 dB. The space of interest is discretized into 25×10 grid points with 10 meters grid size. We randomly generate K targets on the grid. The dimension of the target vector in (6) is 8000 with $32 \times K$ nonzero entries. The magnitude of the complex reflection coefficients for each target in each trial is randomly generated from uniform distribution in the range of $[0.1, 0.8]$. ω_0 is chosen as 1.

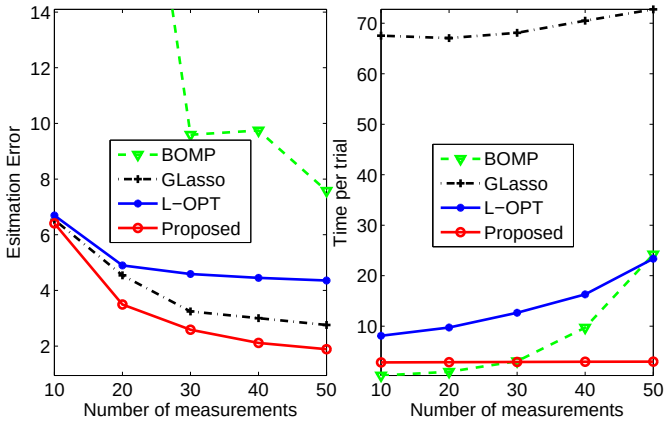

Fig. 1. Performance under different number of measurements. 10 targets; for GLasso $\lambda = 0.02$; and for the proposed method $\lambda = 2$, $\rho_1 = \rho_2 = 1$.

The BOMP [6], GLasso [7] and L-OPT [8] are implemented for comparison. For GLasso, we choose $\lambda = 0.02$ for the best performance. For L-OPT, we set $\epsilon = 2\sqrt{LM_tM_r}\sigma_n$ [8] with the knowledge of σ_n. As for the proposed algorithm, preconditioned conjugate gradient is used to solve the system (14). The estimation error $\|\hat{\mathbf{s}} - \mathbf{s}\|_2$ and the CPU running time are used as the performance metrics. All results are averaged over 100 independent trials.

We first fix $K = 10$ and evaluate the performance under different number of measurements L. The results are plotted in Figure 1. The proposed algorithm achieves lower estimation errors with less CPU run time as compared to GLasso and L-OPT under all L's. The CPU run time of the proposed algorithm remains less than $5s$, while the run time of L-OPT grows superlinearly with L.

Next, we consider the performance of the proposed scheme for different number of targets and fixed number of measurements $L = 50$. The results are plotted in Figure 2. For all values of K, the proposed algorithm achieves lower estimation errors than L-OPT using around one quarter CPU run time.

In the above simulation, L-OPT requires knowledge of the noise variance σ_n^2. The regularization parameter λ in GLasso and the proposed algorithm also need to be manually tuned. In fact, the choice of such parameters are critical for the estimation performance. In Fig. 3, we plot the estimation errors for a wide range of λ. We observe that the estimation error of the proposed algorithm remains very small for a wide range of λ's, while for GLasso, the range of good λ's is very narrow. The robustness to λ makes the proposed algorithm good candidate for real world applications.

VI. Conclusions

In this paper, we have proposed an ADMM-based efficient sparse signal recovery algorithm for target estimation in dis-

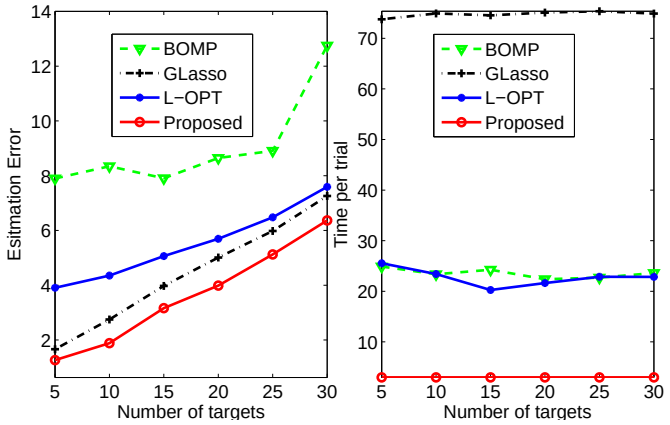

Fig. 2. Performance under different number of targets. 50 measurements; for GLasso $\lambda = 0.02$; and for the proposed method $\lambda = 2$, $\rho_1 = \rho_2 = 1$.

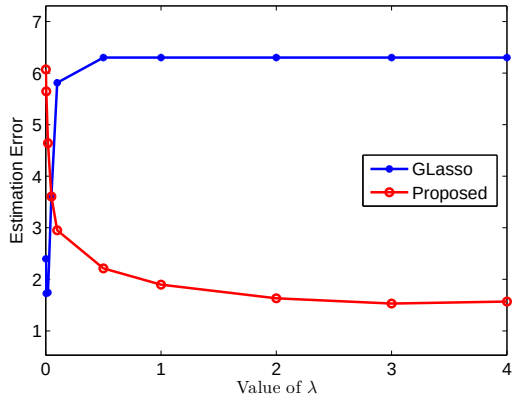

Fig. 3. Performance under different values of λ.

tributed MIMO radar. Simulation results have indicated that the proposed algorithm significantly lowers the computational complexity for target estimation with improved accuracy. Parallel implementation has also been considered for further reduction of the execution time. A semi-distributed implementation, requiring a fusion enter with minimal computational power, has also been discussed. The proposed algorithm can be readily extended for distributed compressive sensing systems in [12], [13].

REFERENCES

[1] J. Li and P. Stoica, "MIMO radar with colocated antennas," *IEEE Signal Processing Magazine*, vol. 24, no. 5, pp. 106–114, 2007.

[2] J. Li, Petre Stoica, L. Xu, and W. Roberts, "On parameter identifiability of MIMO radar," *IEEE Signal Processing Letters*, vol. 14, no. 12, pp. 968–971, 2007.

[3] A. M. Haimovich, R. S. Blum, and L. J. Cimini, "MIMO radar with widely separated antennas," *IEEE Signal Processing Magazine*, vol. 25, no. 1, pp. 116–129, 2008.

[4] H. Godrich, A. M. Haimovich, and R. S. Blum, "Target localization accuracy gain in MIMO radar-based systems," *IEEE Transactions on Information Theory*, vol. 56, no. 6, pp. 2783–2803, 2010.

[5] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[6] S. Gogineni and A. Nehorai, "Target estimation using sparse modeling for distributed mimo radar," *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5315–5325, 2011.

[7] A. P. Petropulu, Y. Yu, and J. Huang, "On exploring sparsity in widely separated MIMO radar," in *Proceedings of the Forty Fifth Asilomar Conference on Signals, Systems and Computers*, 2011, pp. 1496–1500.

[8] B. Li and A. P. Petropulu, "Performance guarantees for distributed MIMO radar based on sparse modeling," in *Proc. IEEE Radar Conference*, 2014, pp. 1369–1372.

[9] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, pp. 1-122, 2011.

[10] Y. Wang, J. Yang, W. Yin, and Y. Zhang " A new alternating minimization algorithm for total variation image reconstruction," *SIAM Journal of Imaging Science*, vol. 1, pp. 248-272, 2008.

[11] D. Malioutov, M. Cetin, and A. S. Willsky. "A sparse signal reconstruction perspective for source localization with sensor arrays," *IEEE Transactions on Signal Processing*, vol. 53 no. 8, pp. 3010-3022, 2005.

[12] D. Baron, M. F. Duarte, M. B. Wakin, S. Sarvotham, and R. G. Baraniuk, "Distributed compressive sensing," *arXiv preprint*, arXiv:0901.3403v1, Jan. 2009.

[13] B. Li and A. Petropulu, "Structured sampling of structured signals," in *Proc. IEEE Global Conference on Signal and Information Processing*, 2013, pp. 1009–1012.