sensors

MDPI

Article

An Efficient Distributed Compressed Sensing Algorithm for Decentralized Sensor Network

Jing Liu *, Kaiyu Huang and Guoxian Zhang

School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China; huangkaiyu@stu.xjtu.edu.cn (K.H.); zgx3135@stu.xjtu.edu.cn (G.Z.)

* Correspondence: elelj20080730@gmail.com

Academic Editor: Leonhard M. Reindl

Received: 8 February 2017; Accepted: 13 April 2017; Published: 20 April 2017

Abstract: We consider the joint sparsity Model 1 (JSM-1) in a decentralized scenario, where a number of sensors are connected through a network and there is no fusion center. A novel algorithm, named distributed compact sensing matrix pursuit (DCSMP), is proposed to exploit the computational and communication capabilities of the sensor nodes. In contrast to the conventional distributed compressed sensing algorithms adopting a random sensing matrix, the proposed algorithm focuses on the deterministic sensing matrices built directly on the real acquisition systems. The proposed DCSMP algorithm can be divided into two independent parts, the common and innovation support set estimation processes. The goal of the common support set estimation process is to obtain an estimated common support set by fusing the candidate support set information from an individual node and its neighboring nodes. In the following innovation support set estimation process, the measurement vector is projected into a subspace that is perpendicular to the subspace spanned by the columns indexed by the estimated common support set, to remove the impact of the estimated common support set. We can then search the innovation support set using an orthogonal matching pursuit (OMP) algorithm based on the projected measurement vector and projected sensing matrix. In the proposed DCSMP algorithm, the process of estimating the common component/support set is decoupled with that of estimating the innovation component/support set. Thus, the inaccurately estimated common support set will have no impact on estimating the innovation support set. It is proven that under the condition the estimated common support set contains the true common support set, the proposed algorithm can find the true innovation set correctly. Moreover, since the innovation support set estimation process is independent of the common support set estimation process, there is no requirement for the cardinality of both sets; thus, the proposed DCSMP algorithm is capable of tackling the unknown sparsity problem successfully.

Keywords: distributed compressed sensing; JSM-1; distributed compact sensing matrix pursuit (DCSMP) algorithm

1. Introduction

Compressed sensing has received considerable attention recently and has been applied successfully in diverse fields, e.g., image processing [1], speech enhancement [2], sensor network [3,4] and radar systems [5]. As an important branch of compressed sensing, distributed compressed sensing (DCS) theory [6,7] rests on a new concept called the joint sparsity of a signal ensemble. A signal ensemble is composed of different signals from the various sensors of the same scene. Three joint sparsity models (JSM) are presented in [6]: JSM-1, JSM-2 and JSM-3. In JSM-1, each signal consists of a sum of two components: a common component that is present in all of the signals and an innovation component that is unique to each signal. In JSM-2, all signals are constructed from the same sparse set of basis vectors, but with different coefficient values. JSM-3 extends JSM-1 so that the common

component needs no longer to be sparse in any basis. The readers can refer to [6] for more details about JSM-1, JSM-2 and JSM-3.

In this paper, we focus on the JSM-1, and such signals may arise in a sensor network where large-scale phenomena affect all sensors and local phenomena affect individual sensors. The first example would be a network of temperature sensors in a forest, where the sun has a global effect, and shade, water and animals have more local effects [6]. The second example comes from distributed spectrum sensing in cooperative cognitive networks, which consists of the estimation of a common component (due to the so-called primary users) and innovation components (due to secondary users) [8]. The third example is about joint detection and tracking of multiple targets using a multistatic radar system, where different individual receivers observe the same surveillance region with different detection probabilities. Till recently, much research work has been carried out on data aggregation, data collection and data processing in wireless sensor networks [9–11].

Algorithms for the distributed compressed sensing problem can be developed either in a central manner or a distributed manner [12]. The central recovery methods assume the presence of a fusion center that gathers all of the information from the network, while the decentralized recovery methods perform the reconstruction in network, with no fusion center. The decentralized recovery methods have the advantages of convenience (for implementation) and robustness (to failures), over the central ones.

Centralized reconstruction for JSM-1 has already been addressed in the literature [13–16]. In this paper, in contrast, we study a decentralized approach to JSM-1. To the best of our knowledge, there are limited literature works about the JSM-1 in a decentralized network. In [17], a distributed alternating direction method of multipliers (ADMM) is applied in the JSM-1 framework to recover both the common and the individual components. In [12], the authors develop a distributed parallel pursuit (DIPP) algorithm based on the exchange of information about estimated support sets at sensors.

In the above work, the random matrix (e.g., the random Gaussian matrix) is chosen as the sensing matrix, which provides the worst case performance guarantees in the context of an adversarial signal/error model. In this work, we focus on the deterministic sensing matrices built directly on the real acquisition systems. These kinds of matrices eliminate the need for additional measurement matrices, reduce memory storage and accelerate the reconstruction algorithm. However, the deterministic sensing matrices often encounter the high coherence problem due to the high resolution of the sensors [18]. In our previous work [19], a general similar sensing matrix pursuit (GSSMP) algorithm is proposed to cope with the high coherence problem, which obtains much better performance compared with the subspace pursuit (SP) and basis pursuit (BP) algorithms. However, the GSSMP algorithm can only tackle a single sparse signal.

In this work, the GSSMP algorithm is extended to tackle the JSM-1 in a decentralized scenario. The individual sensing matrix for each sensor is constructed based on the real acquisition system. A compact sensing matrix is built based on the original sensing matrix using similarity analysis. The compact sensing matrix has a far smaller size compared with the original sensing matrix and is proven to have low coherence. For the signal model, we use the recent proposed mixed support set model of [12], which consists of a common support set and an innovation support set. Thus, the purpose of the proposed algorithm is to estimate both the common and innovation support sets, for the reconstruction of the sparse signal of each individual sensor. The proposed algorithm is referred to the distributed compact sensing matrix pursuit (DCSMP) algorithm, and it comprises two main parts: common support set estimation and innovation support set estimation processes.

In the common support set estimation process, a candidate support set is calculated based on the compact sensing matrix and the measurement vector, for each individual sensor. We prove that the candidate support set contains the true support set. Considering that the common support set is a subset of all individual support sets of all sparse signals, we can obtain an estimated common support set by fusing the candidate support set information from an individual node and its neighboring nodes.

Next, we are to pursue the innovation support set based on the estimated common support set, for each individual sensor. The measurement vector is projected into a subspace that is perpendicular

to the subspace spanned by the columns indexed by the estimated common support set, to remove the impact of it. We can then search for the innovation support set using an OMP algorithm based on the projected measurement vector and projected sensing matrix.

The main contribution of the paper has three components. First, in contrast to the conventional distributed compressed sensing algorithms adopting a random sensing matrix, the proposed DCSMP algorithm focuses on the deterministic sensing matrices built directly on the real acquisition systems, which eliminate the need for additional measurement matrices, reduce memory storage and accelerate the reconstruction algorithm. Secondly, in most algorithms addressing the JSM-1, the process of estimating the common component/support set is coupled with that of estimating the innovation component/support set; thus, the inaccurately estimated common support set will lead to failure in estimating the innovation support set. In this paper, the two processes are decoupled by projecting the measurement vector into a subspace that is perpendicular to the subspace spanned by the columns indexed by the estimated common support set, to remove the impact of it. It is proven that under the condition that the estimated common support set contains the true common support set, the proposed algorithm can find the true innovation set correctly. Thirdly, since the innovation support set estimation process is independent of the common support set estimation process, there is no requirement for the cardinality of both sets; thus, the proposed DCSMP algorithm is capable of tackling the unknown sparsity problem successfully.

The paper is organized as follows. Section 2 introduces the JSM-1 model in distributed compressed sensing. The proposed DCSMP algorithm is introduced in Section 3, which is the main contribution of this paper. The complexity and scalability analysis of the proposed DCSMP algorithm is in Section 4. In Section 5, we consider a simulation example of identifying multiple targets in a multistatic radar system, where different individual receivers observe the same surveillance region with different detection probabilities. A JSM-1 model is constructed based on the multistatic radar system, and sparse recovery is carried out in a decentralized manner across various receivers. Finally, the paper is summarized in Section 6.

1.1. Notation

For a set $T \subset \{1, 2, \cdots, n\}$, we use $|T|$ to denote its cardinality, i.e., the number of elements in T. We use T^c to denote its complement w.r.t. $\{1, 2, \cdots, n\}$, i.e., $T^c := i \in \{1, 2, \cdots, n\} : i \notin T$.

For a vector $\mathbf{v}$, $\mathbf{v}(i)$ denotes the i-th entry of $\mathbf{v}$, and $\mathbf{v}_T$ denotes a vector consisting of the entries of $\mathbf{v}$ indexed by T. We use $||\mathbf{v}||_p$ to denote the l_p norm of $\mathbf{v}$. The support set of $\mathbf{v}$, supp($\mathbf{v}$), is the set of indices at which $\mathbf{v}$ is nonzero, supp($\mathbf{v}$):$= \{i : \mathbf{v}(i) \neq 0\}$. We say that $\mathbf{v}$ is s-sparse if $|\text{supp}(\mathbf{v})| \leq s$.

For a matrix $\mathbf{B}$, $\mathbf{B}^*$ denotes its conjugate transpose and $\mathbf{B}^\dagger$ its pseudo-inverse. For a matrix with linearly-independent columns, $\mathbf{B}^\dagger = (\mathbf{B}^*\mathbf{B})^{-1}\mathbf{B}^*$.

We use $\mathbf{I}$ to denote an identity matrix of appropriate size. For an index set T and a matrix $\mathbf{B}$, $\mathbf{B}_T$ is the sub-matrix of $\mathbf{B}$ containing columns with indices in the set T. Notice that $\mathbf{B}_T = \mathbf{B}\mathbf{I}_T$. For a tall matrix $\mathbf{P}$, span($\mathbf{P}$) denotes the subspace spanned by the column vectors of $\mathbf{P}$.

2. Problem Formulation in the JSM-1 Framework

Consider a decentralized sensor network where a number of sensors acquire signals and communicate with neighbor nodes to reconstruct the original signals. Each individual sensor (e.g., the p-th sensor node) is not aware of the full network topology, and instead, it knows two sets of local neighbors; the incoming neighbor connections L_p^{in} and outgoing neighbor connections L_p^{out}. Here, incoming and outgoing connections correspond to communication links where a node can receive or send information, respectively. In particular, it is assumed that the signals sensed by these sensors exhibit both intra-sensor correlation and inter-sensor correlation. This correlated sensing model corresponds to the JSM-1 model, which is described as follows.

The p-th sensor monitors a discrete signal $\mathbf{x}_p \in \Re^N$ according to the following relation:

$$\mathbf{y}_p = \mathbf{\Phi}_p \mathbf{x}_p + \mathbf{e}_p, \quad \forall \quad p \in \Gamma, \tag{1}$$

where $\mathbf{y}_p \in \Re^M$ is the measurement vector, $\mathbf{\Phi}_p \in \Re^{M \times N}$ is the sensing matrix, $\mathbf{e}_p \in \Re^M$ is the measurement noise and Γ is a global set containing all nodes in the network. This setup describes an underdetermined system, where $M < N$. The signal vector $\mathbf{x}_p$ is K-sparse, and its support set is defined as S_p. The goal of the proposed algorithm is to reconstruct the original signal observed at each sensor node, i.e., to reconstruct $\mathbf{x}_p$ at the p-th node ($p \in \Gamma$).

In the JSM-1, each sensor has its own signal; that means signals across sensors are not the same, but have correlations. The sparse signal $\mathbf{x}_p$ can be represented as:

$$\mathbf{x}_p = \mathbf{z}_c + \mathbf{z}_p, \tag{2}$$

where $\mathbf{z}_c \in \Re^N$ denotes the common component of the sparse signal $\mathbf{x}_p$, which captures the inter-signal correlation and is common to all signals, and $\mathbf{z}_p \in \Re^N (p \in \Gamma)$ denotes the innovation component of the sparse signal $\mathbf{x}_p$, which captures the intra-signal correlation and is specific to the p-th sparse vector $\mathbf{x}_p$. We further define J and I_p as the support set of $\mathbf{z}_c$ and $\mathbf{z}_p$, respectively, and have:

$$S_p = J \cup I_p \quad and \quad J \bigcap I_p = \emptyset, \quad \forall \quad p \in \Gamma. \tag{3}$$

Here, the partial support set J is common to all sparse signals, which is defined as the common support set. The partial support set I_p is specific for the p-th node and defined as the innovation support set.

3. Distributed Compact Sensing Matrix Pursuit Algorithm

The goal of this section is to introduce the proposed DCSMP algorithm. A block diagram of the DCSMP algorithm is shown in Figure 1, which comprises four stages: (1) rough estimation, (2) data fusion, (3) innovation support set estimation and (4) final estimation. The estimated common support set is generated at the first and second stage; the estimated innovation support set is obtained at the third stage; and finally, we can obtain the final estimates of the support set and the original sparse signal at the fourth stage.

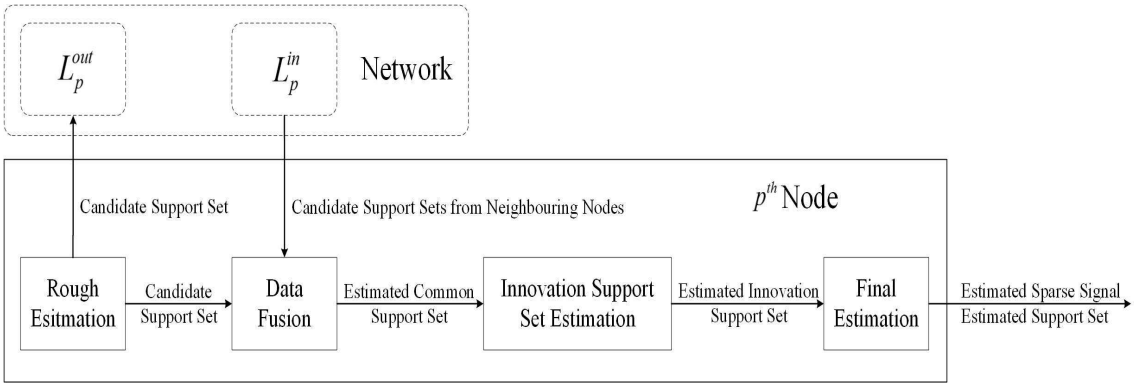


Figure 1. Block diagram of the DCSMP algorithm.

3.1. Rough Estimation

The goal of this section is to calculate the candidate support set for each individual sensor. In order to tackle the high coherence problem due to the high resolution of the sensors, we consider constructing a compact sensing matrix for each individual sensing matrix at each node, which is a compact version

of the original sensing matrix and has low coherence. The readers can refer to [19] for details of the construction process of the compact sensing matrix. An OMP algorithm is then utilized to calculate a rough estimate of the true support set, based on which we can obtain the candidate support set $\hat{F}_p$ in the original sensing matrix. Finally, we prove that the candidate support set contains the true support set for each individual sensor. The detailed procedures of the candidate support set estimation (CSSE) algorithm are presented in Algorithm 1.

Algorithm 1: Candidate support set estimation (CSSE).

Input: $\mathbf{\Phi}_p$, $\mathbf{y}_p$
Output: $\hat{F}_p$

1. $\mathbf{\Psi}_p \leftarrow$ ConstructCompact($\mathbf{\Phi}_p$).
2. $\hat{\mathbf{a}}_p \leftarrow$ OMP($\mathbf{\Psi}_p$, $\mathbf{y}_p$).
3. $\hat{\Xi}_p^{ini} \leftarrow$ MapToSubspace($\hat{\mathbf{a}}_p$).
4. $\hat{F}_p \leftarrow$ FindCandidateSupportSet ($\hat{\Xi}_p^{ini}$).

The candidate support set estimation algorithm consists of four steps. The first step uses the "ConstructCompact" function, which constructs the compact sensing matrix $\mathbf{\Psi}_p$ based on the original sensing matrix $\mathbf{\Phi}_p$. The process is the same as that described in Section 2.3 in [19]. At the second step, the OMP algorithm is used to find an estimate of the true support set, which is represented as $\hat{\mathbf{a}}_p = \{\hat{a}_p^1, \cdots, \hat{a}_p^k, \cdots, \hat{a}_p^{K'}\}$, $K' \leq K$, where $\hat{a}_p^k$ denotes the k-th element of $\hat{\mathbf{a}}_p$. At the third step, in the "MapToSubspace" function, each element in the estimated support set $\hat{a}_p^k$ corresponds to a condensed column of the compact sensing matrix. For example, $\hat{a}_p^k(\boldsymbol{\beta}_p^j)$ indicates that the k-th element of $\hat{\mathbf{a}}_p$ corresponds to the j-th condensed column $\boldsymbol{\beta}_p^j$. This column is defined as a contributing column. We can then obtain an initial estimate of the correct subspace, $\hat{\Xi}_p^{ini}$, spanned by K' contributing columns $\{\boldsymbol{\beta}_p^i, \cdots, \boldsymbol{\beta}_p^j, \cdots, \boldsymbol{\beta}_p^l\}$, as $\hat{\Xi}_p^{ini}$ =span$(\boldsymbol{\beta}_p^i, \cdots, \boldsymbol{\beta}_p^j, \cdots, \boldsymbol{\beta}_p^l)$. The final step uses the function "FindCandidateSupportSet", which is to find the candidate support set in the original sensing matrix $\mathbf{\Phi}_p$. Each contributing column corresponds to a similar column group in $\mathbf{\Phi}_p$, which is named the contributing similar column group. We can then obtain a set $\hat{\mathbf{\Lambda}}_p$ containing the indices of K' contributing similar column groups. All of the columns in each contributing similar column group from $\hat{\mathbf{\Lambda}}_p$ are listed out, and their indices form a candidate support set $\hat{F}_p$.

Proposition 1. *The true support set S_p is a subset of the candidate support set $\hat{F}_p$, i.e., $S_p \subset \hat{F}_p$.*

Proof. According to Proposition 2 in [19], the vectors spanning the true subspace are contained in K' contributing similar column groups. Thus, the true support set is contained in the candidate support set $\hat{F}_p$, which is the union of the indices of the columns contained in the K' contributing similar column groups. □

At the end of the rough estimation stage, the p-th node sends its own candidate support set to its neighboring nodes, as well as receives the candidate support sets from its neighboring nodes. The candidate support sets from both the p-th node and its neighboring nodes are sent to the data fusion stage, where the estimated common support set is generated by using some fusion strategy.

3.2. Data Fusion

For the p-th node, its true support set S_p (including the common support set J and innovation support set I_p) is contained in the candidate support set $\hat{F}_p$, according to Proposition 1. Considering that the common support set is joint to the p-th node and its connected neighboring nodes, it can be obtained by fusing the candidate support sets from both the p-th node and its neighboring nodes in the network. For fusion, we use a democratic voting strategy [12].

The data fusion algorithm is presented in Algorithm 2. The p-th node has access to the candidate support sets $\{\hat{F}_q\}_{q\in L_p^{in}}$ from neighbors and the local estimate $\hat{F}_p$. The estimated common support set $\hat{J}$ is formed (Step 5) such that each index in the estimated common support set is present in at least two candidate support sets from $\{\{\hat{F}_q\}_{q\in L_p^{in}}, \hat{F}_p\}$. Having more votes for a certain index increases the probability of this index being correct.

Algorithm 2: Data fusion.

Input: $\{\hat{F}_q\}_{q\in L_p^{in}}$, $\hat{F}_p$
Output: $\hat{J}$
Initialization: $\mathbf{z} \leftarrow \mathbf{0}_{N\times 1}$
1: $\mathbf{z} \leftarrow vote_1(\mathbf{z}, \hat{F}_p)$
2: for each $q \in L_p^{in}$ do
3: $\quad \mathbf{z} \leftarrow vote_1(\mathbf{z}, \hat{F}_q)$
4: end for
5: Choose $\hat{J}$ s.t. $(\mathbf{z}(i) \geq 2) \ \forall\, i \in \hat{J}$

In the above algorithm, "$vote_1$" denotes the voting procedure [12]. Since an index present in two nodes' candidate support sets will be treated as an element of the estimated common support set, the probability of the event that the estimated common support set contains the true common support set is very high. Thus, it is reasonable to assume that the true common support set is a subset of the estimated common support set.

3.3. Innovation Support Set Estimation

We consider a general condition that the true common support set is a subset of the estimated common support set. The related assumptions are listed as follows.

Assumption 1. *The true common support set J and the innovation support set I_p, satisfy the following conditions:*

1. *The true common support set J is a subset of the estimated common support set $\hat{J}$, i.e., $J \subseteq \hat{J}$.*
2. *The innovation support set I_p is a proper subset of the complement of $\hat{J}$, i.e., $I_p \subset \hat{J}^c$.*

The second assumption holds considering that the cardinality of I_p is far less than that of $\hat{J}^c$.

The goal of the innovation support set estimation stage is to calculate I_p. We prove Lemma 1 before we proceed to the detailed procedures of the proposed algorithm.

Lemma 1. *Assuming that $|\hat{J}| \leq 2K$, define the projection matrix $\mathbf{P}_p$ as:*

$$\mathbf{P}_p = \mathbf{I} - (\mathbf{\Phi}_p)_{\hat{J}}[(\mathbf{\Phi}_p)_{\hat{J}}^*(\mathbf{\Phi}_p)_{\hat{J}}]^{-1}(\mathbf{\Phi}_p)_{\hat{J}}^*, \tag{4}$$

and the projected measurement vector $\tilde{\mathbf{y}}_p$ as:

$$\tilde{\mathbf{y}}_p = \mathbf{P}_p\mathbf{y}_p. \tag{5}$$

The projected measurement vector $\tilde{\mathbf{y}}_p$ can be represented in a standard equation in compressed sensing as:

$$\tilde{\mathbf{y}}_p = \mathbf{A}_p(\mathbf{x}_p)_{\hat{J}^c} + \mathbf{e}'_p, \tag{6}$$

where $\mathbf{A}_p = \mathbf{P}_p(\mathbf{\Phi}_p)_{\hat{J}^c}$ is the projected sensing matrix and $\mathbf{e}'_p = \mathbf{P}_p\mathbf{e}_p$ is the equivalent noise. Moreover, the innovation support set I_p can be calculated using an OMP algorithm, based on the projected measurement vector $\tilde{\mathbf{y}}_p$ and projected sensing matrix $\mathbf{A}_p$.

Proof. The proof consists of three parts. First, we prove that $(\mathbf{\Phi}_p)_{\hat{J}}$ has full column rank under the condition that the cardinality of the estimated common support set $\hat{J}$ is less than or equal to $2K$, i.e., $|\hat{J}| \leq 2K$, where K is the sparsity level of $\mathbf{x}_p$.

According to Theorem 2.13 in [20], a unique s-sparse solution of the system $\mathbf{y} = \mathbf{A}\mathbf{x}$ implies that every set of $2s$ columns of $\mathbf{A}$ is linearly independent, where s indicates the sparsity level of $\mathbf{x}$. Theorem 2.13 applies to the deterministic sensing matrix in this work, since the system has a unique K-sparse solution. Thus, every set of $2K$ columns of $\mathbf{\Phi}_p$ is linearly independent. Under the condition that $|\hat{J}| \leq 2K$, the submatrix $(\mathbf{\Phi}_p)_{\hat{J}}$ contains at most $2K$ columns. Since every set of $2K$ columns of $\mathbf{\Phi}_p$ is linearly independent, $(\mathbf{\Phi}_p)_{\hat{J}}$ has full column rank.

Secondly, we prove that the projected measurement vector $\tilde{\mathbf{y}}_p$ can be represented in a standard equation in compressed sensing.

$$\begin{aligned} \tilde{\mathbf{y}}_p &= \mathbf{P}_p \mathbf{y}_p \\ &= \mathbf{P}_p \left(\mathbf{\Phi}_p \mathbf{x}_p + \mathbf{e}_p \right) \\ &= \mathbf{P}_p \left[\left(\mathbf{\Phi}_p \right)_{\hat{J}} \left(\mathbf{x}_p \right)_{\hat{J}} + \left(\mathbf{\Phi}_p \right)_{\hat{J}^c} \left(\mathbf{x}_p \right)_{\hat{J}^c} + \mathbf{e}_p \right] \\ &= \mathbf{P}_p \left(\mathbf{\Phi}_p \right)_{\hat{J}^c} \left(\mathbf{x}_p \right)_{\hat{J}^c} + \mathbf{P}_p \mathbf{e}_p, \\ &= \mathbf{A}_p (\mathbf{x}_p)_{\hat{J}^c} + \mathbf{e}'_p \end{aligned} \tag{7}$$

The fourth equality of (7) holds considering that the contribution of $\hat{J}$ to $\mathbf{y}_p$ can be nullified, by projecting $\mathbf{y}_p$ into a perpendicular space using the projection matrix $\mathbf{P}_p$.

$$\begin{aligned} & \mathbf{P}_p \left(\mathbf{\Phi}_p \right)_{\hat{J}} \left(\mathbf{x}_p \right)_{\hat{J}} \\ = & \left\{ \mathbf{I} - \left(\mathbf{\Phi}_p \right)_{\hat{J}} \left[\left(\mathbf{\Phi}_p \right)_{\hat{J}}^* \left(\mathbf{\Phi}_p \right)_{\hat{J}} \right]^{-1} \left(\mathbf{\Phi}_p \right)_{\hat{J}}^* \right\} \left(\mathbf{\Phi}_p \right)_{\hat{J}} \left(\mathbf{x}_p \right)_{\hat{J}} \\ = & \mathbf{0} \end{aligned} \tag{8}$$

Thus, we can obtain (6), a standard equation in compressed sensing.

Finally, we will prove that I_p is the support set of the sparse vector $(\mathbf{x}_p)_{\hat{J}^c}$. The support set of $\mathbf{x}_p$ can be represented as:

$$S_p = J \cup I_p \quad and \quad J \bigcap I_p = \emptyset, \quad \forall \quad p \in \Gamma. \tag{9}$$

According to Assumption 1, J is a subset of $\hat{J}$ and, thus, is the set of indices at which $(\mathbf{x}_p)_{\hat{J}}$ is nonzero. Similarly, I_p is the proper subset of $\hat{J}^c$ and is the set of indices at which $(\mathbf{x}_p)_{\hat{J}^c}$ is nonzero. Thus, I_p is the support set of the sparse vector $(\mathbf{x}_p)_{\hat{J}^c}$. □

In summary, (6) is a standard equation in compressed sensing, and we can calculate I_p using an OMP algorithm based on the projected measurement vector $\tilde{\mathbf{y}}_p$ and projected sensing matrix $\mathbf{A}_p$.

Algorithm 3: Innovation support set estimation (ISSE).

Input: $\mathbf{\Phi}_p$, $\mathbf{y}_p$, $\hat{J}$
Output: $\hat{I}_p$

1. $\mathbf{P}_p \leftarrow \mathbf{I} - (\mathbf{\Phi}_p)_{\hat{J}} \left[(\mathbf{\Phi}_p)_{\hat{J}}^* (\mathbf{\Phi}_p)_{\hat{J}} \right]^{-1} (\mathbf{\Phi}_p)_{\hat{J}}^*$.
2. $\tilde{\mathbf{y}}_p \leftarrow \mathbf{P}_p \mathbf{y}_p$.
3. $\mathbf{A}_p \leftarrow \mathbf{P}_p (\mathbf{\Phi}_p)_{\hat{J}^c}$.
4. $\hat{I}_p \leftarrow$ OMP($\tilde{\mathbf{y}}_p$, $\mathbf{A}_p$).

In the above algorithm (Algorithm 3), the first step is to construct the projection matrix $\mathbf{P}_p$, while the second and third steps are to construct the projected measurement vector $\tilde{\mathbf{y}}_p$ and projected sensing matrix $\mathbf{A}_p$. Finally, the innovation support set $\hat{I}_p$ can be calculated using an OMP algorithm, based on $\tilde{\mathbf{y}}_p$ and $\mathbf{A}_p$.

3.4. Final Estimation

The goal of this section is to estimate the support set S_p and the sparse vector $\mathbf{x}_p$, given the estimated common support set $\hat{J}$ and the estimated innovation support set $\hat{I}_p$. First, a combinational search is performed in $\hat{J}$ to find the true common support set J. Each combination together with $\hat{I}_p$ forms an estimate of the true support set, which is denoted as $\hat{S}_p^j,\ j = 1, 2, \cdots, N_{co}$, where N_{co} indicates the number of combinations. Based on each estimated support set, we can obtain an estimate of the sparse vector $\mathbf{x}_p$ (denoted as $\hat{\mathbf{x}}_p^j$), using the pseudo-inverse operation. Among the obtained estimated sparse vectors $\hat{\mathbf{x}}_p^j,\ j = 1, 2, \cdots, N_{co}$, we can find the one with the least residual, which is termed as the final estimate of the sparse vector $\mathbf{x}_p$. The detailed procedures are as follows.

The combinational search algorithm consists of six steps. The first step uses the "ListCombinations" function, which lists $C_E^1, C_E^2, \cdots, C_E^E$ combinations based on the indices in $\hat{J}$, where $E = |\hat{J}|$. Each combination is represented as $JI^j,\ j = 1, 2, \cdots, N_{co}$. At the second step, each combination together with $\hat{I}_p$ forms an estimate of the true support set, i.e., $\hat{S}_p^j = JI^j \bigcup \hat{I}_p,\ j = 1, 2, \cdots, N_{co}$. At the third step, the proposed algorithm solves a least squares problem to approximate the nonzero entries ($(\hat{\mathbf{x}}_p^j)_{\hat{S}_p^j} \leftarrow [(\mathbf{\Phi}_p)_{\hat{S}_p^j}]^\dagger\ \mathbf{y}_p$) and sets other entries as zero ($(\hat{\mathbf{x}}_p^j)_{(\hat{S}_p^j)^c} \leftarrow \mathbf{0}$), resulting in an estimate of the sparse vector, $\hat{\mathbf{x}}_p^j,\ j = 1, 2, \cdots, N_{co}$. The fourth step is to calculate the residual $\mathbf{r}_p^j\ (j = 1, \cdots, N_{co})$. The l_2 norm of $\mathbf{r}_p^j$ is indicated as $||\mathbf{r}_p^j||_2$. The fifth step uses the "MinimumResidual" function, which finds the residual with the least l_2 norm among the residuals, and denotes it as $\mathbf{r}_p^{min}$. Concurrently, we can find its associate sparse signal and support set, denoted as $\hat{\mathbf{x}}_p^{min}$ and $\hat{S}_p^{min}$, respectively. Finally, we can obtain the final estimates of the sparse signal and the true support set, by setting $\hat{\mathbf{x}}_p \leftarrow \hat{\mathbf{x}}_p^{min}$ and $\hat{S}_p \leftarrow \hat{S}_p^{min}$.

Proposition 2. *The proposed combinational search algorithm (Algorithm 4) can find the true support set S_p, provided that $\hat{I}_p$ is correctly estimated.*

Proof. In Algorithm 4, $C_E^1, C_E^2, \cdots, C_E^E$ combinations are listed based on the indices in $\hat{J}$, where $E = |\hat{J}|$. Each combination is represented as $JI^j,\ j = 1, 2, \cdots, N_{co}$, where N_{co} indicates the number of combinations. Since $J \subseteq \hat{J}$, J is contained in $JI^j,\ j = 1, 2, \cdots, N_{co}$. Thus, $S_p = J \bigcup I_p$ is contained in $JI^j \bigcup \hat{I}_p$ provided that $\hat{I}_p$ is correctly estimated. Therefore, we can find the true support set S_p via the combinational search algorithm. □

Algorithm 4: Combinational search.

Input: $\mathbf{\Phi}_p$, $\mathbf{y}_p$, $\hat{J}$, $\hat{I}_p$
Output: $\hat{S}_p$, $\hat{\mathbf{x}}_p$

1. $JI^j \leftarrow$ ListCombinations $(\hat{J})$, $j = 1, 2, \cdots, N_{co}$.
2. $\hat{S}_p^j \leftarrow JI^j \bigcup \hat{I}_p$, $j = 1, 2, \cdots, N_{co}$.
3. $(\hat{\mathbf{x}}_p^j)_{\hat{S}_p^j} \leftarrow [(\mathbf{\Phi}_p)_{\hat{S}_p^j}]^\dagger\ \mathbf{y}_p$, $(\hat{\mathbf{x}}_p^j)_{(\hat{S}_p^j)^c} \leftarrow \mathbf{0}$, $j = 1, 2, \cdots, N_{co}$.
4. $\mathbf{r}_p^j \leftarrow \mathbf{y}_p - \mathbf{\Phi}_p \hat{\mathbf{x}}_p^j$, $j = 1, 2, \cdots, N_{co}$.
5. $\mathbf{r}_p^{min} \leftarrow$ MinimumResidual $(\{\mathbf{r}_p^1, \mathbf{r}_p^2, \cdots, \mathbf{r}_p^{N_{co}}\})$. Concurrently, find $\hat{\mathbf{x}}_p^{min}$ and $\hat{S}_p^{min}$.
6. $\hat{\mathbf{x}}_p \leftarrow \hat{\mathbf{x}}_p^{min}$, $\hat{S}_p \leftarrow \hat{S}_p^{min}$.

3.5. DCSMP Algorithm

Using Algorithms 1∼4, we now develop the DCSMP algorithm presented in Algorithm 5. The inputs to Algorithm 5 for the p-th node are the measurement signal $\mathbf{y}_p$ and the sensing matrix $\mathbf{\Phi}_p$. Furthermore, Algorithm 5 knows L_p^{in} and L_p^{out}. We assume that some underlying communication scheme is provided for the transmit and receive functionality.

In Algorithm 5, local candidate support set estimates are exchanged over the network (Steps 2 and 3). The data fusion algorithm merges the local and neighboring candidate support set estimates to produce the estimated common support set $\hat{J}$ (Step 4). At Step 5, the contribution of $\hat{J}$ to $\mathbf{y}_p$ is nullified, by projecting $\mathbf{y}_p$ into a subspace perpendicular to the space spanned by the columns of $\mathbf{\Phi}_p$, indexed by $\hat{J}$. We can then calculate the estimated innovation support set $\hat{I}_p$ using an OMP algorithm based on the projected measurement vector and projected sensing matrix. Finally, a combinational search is performed in $\hat{J}$ to find J, and thus, we can obtain the final estimates of the support set and the original sparse vector.

Algorithm 5: Distributed CSMP.

Input: $\mathbf{y}_p$, $\mathbf{\Phi}_p$, L_p^{in}, L_p^{out}
Output: $\hat{S}_p, \hat{\mathbf{x}}_p$

1. $\hat{F}_p \leftarrow \text{CSSE}(\mathbf{\Phi}_p, \mathbf{y}_p)$;
2. Transmit: send $\hat{F}_p$ to its neighboring nodes;
3. Receive: Receive $\{\hat{F}_q\}_{q \in L_p^{in}}$ from the neighboring nodes;
4. $\hat{J} \leftarrow \text{DataFusion}(\{\hat{F}_q\}_{q \in L_p^{in}}, \hat{F}_p)$;
5. $\hat{I}_p \leftarrow \text{ISSE}(\mathbf{\Phi}_p, \mathbf{y}_p, \hat{J})$;
6. $(\hat{S}_p, \hat{\mathbf{x}}_p) \leftarrow \text{CombinationalSearch}(\mathbf{\Phi}_p, \mathbf{y}_p, \hat{J}, \hat{I}_p)$

Discussion: Two additional settings are considered when information from all sensors can be fused. The first setting is that each sensor has the capability of communicating with all other sensors in the network, and the second one is that the measurements of all sensors are sent to a fusion center, which generates an estimate of the original signal using information from all sensors.

It is straightforward to extend the proposed DCSMP algorithm to the above two settings. To deal with the first condition, we need only change the inputs of the data fusion algorithm (Algorithm 2). For instance, for the p-th node, change the original inputs of the data fusion algorithm, the candidate support sets from its neighbors, $\{\hat{F}_q\}_{q \in L_p^{in}}$, to the candidate support sets from all other sensors in the network, $\{\hat{F}_q\}_{q \neq p}$. For the second one, when the measurements from all of the sensors in the network are sent to the fusion center, we need only change the inputs of the data fusion algorithm (Algorithm 2) to the candidate support sets from all of the sensors in the network.

4. Complexity and Scalability Analysis

4.1. Complexity Analysis

The proposed DCSMP algorithm consists of two main parts: offline processing and online processing. The offline processing transforms the individual sensing matrix $\mathbf{\Phi}_p$ to a compact sensing matrix $\mathbf{\Psi}_p$ using similarity analysis. The computational complexity of the offline processing mainly focuses on the computation of the similarity between any two columns of the individual sensing matrix, which is of the order of $\text{O}(\frac{N(N-1)}{2}M)$ [19]. M and N are the numbers of the rows and columns of $\mathbf{\Phi}_p$, respectively.

The online processing procedure consists of four parts: (1) rough estimation; (2) data fusion; (3) innovation support set estimation; and (4) final estimation. First, for the rough estimation process, an OMP algorithm is used to find a rough estimate of the true support set for each individual signal.

The computational complexity is of the order of $O(MD_p)$ for the p-th node, where D_p is the number of columns of the compact sensing matrix $\mathbf{\Psi}_p$ [19].

Secondly, for the data fusion process, we use a democratic voting strategy [12], which is very simple and has negligible complexity compared with the other three processes.

In the innovation support set estimation process, first, the measurement vector $\mathbf{y}_p$ is projected into the subspace that is perpendicular to the space spanned by the columns of $\mathbf{\Phi}_p$, which are indexed by the estimated common support set $\hat{J}$, as:

$$\begin{aligned} \tilde{\mathbf{y}}_p &= \mathbf{P}_p \mathbf{y}_p \\ &= \left\{ \mathbf{I} - \left(\mathbf{\Phi}_p\right)_{\hat{J}} \left[\left(\mathbf{\Phi}_p\right)_{\hat{J}}^{*} \left(\mathbf{\Phi}_p\right)_{\hat{J}} \right]^{-1} \left(\mathbf{\Phi}_p\right)_{\hat{J}}^{*} \right\} \mathbf{y}_p \\ &= \mathbf{y}_p - \left(\mathbf{\Phi}_p\right)_{\hat{J}} \left(\mathbf{\Phi}_p\right)_{\hat{J}}^{\dagger} \mathbf{y}_p. \end{aligned} \tag{10}$$

The complexity of this step concentrates on the pseudo-inverse operation, i.e., computing $(\mathbf{\Phi}_p)_{\hat{J}}^{\dagger}\mathbf{y}_p$, using the least squares algorithm. Thus, the computational cost focuses on the least squares estimation and is of the order of $O(|\hat{J}| \cdot M)$, according to [21]. Secondly, the estimated innovation support set $\hat{I}_p$ is calculated using an OMP algorithm, and the complexity of this step is $O(M \cdot |\hat{J}^c|)$. Thus, the complexity of entire innovation support set estimation is $O(|\hat{J}| \cdot M)+O(M \cdot |\hat{J}^c|)$.

In the final estimation process, $C_E^1, C_E^2, \cdots, C_E^E$ combinations are listed out based on the indices in $\hat{J}$, where $E = |\hat{J}|$. Each combination is represented as JI^j, $j = 1, 2, \cdots, N_{co}$, where $N_{co} = C_E^1 + C_E^2 + \cdots + C_E^E$. Each combination together with $\hat{I}_p$ forms an estimate of the true support set, i.e., $\hat{S}_p^j = JI^j \bigcup \hat{I}_p$, $j = 1, 2, \cdots, N_{co}$. Based on each estimated support set, the nonzero entries of the estimated sparse vector are calculated using the least squares algorithm. The computational cost of the final estimation process is of the order of $C_E^1 \cdot O(|\hat{S}_p^1| \cdot M) + C_E^2 \cdot O(|\hat{S}_p^2| \cdot M) + \cdots + C_E^E \cdot O(|\hat{S}_p^E| \cdot M)$ [21]. Furthermore, since $max(|\hat{S}_p^1|, |\hat{S}_p^2|, \cdots, |\hat{S}_p^E|) \leq K$, we have:

$$C_E^1 \cdot O(|\hat{S}_p^1| \cdot M) + \cdots + C_E^E \cdot O(|\hat{S}_p^E| \cdot M) \quad < \quad C_E^1 \cdot O(KM) + \cdots + C_E^E \cdot O(KM) \tag{11}$$

Considering that $C_E^1 \cdot O(KM) + \cdots + C_E^E \cdot O(KM) = N_{co} \cdot O(KM)$, the computational cost of the final estimation process can be approximated as the order of $N_{co} \cdot O(KM)$.

In summary, the complexity analysis for online processing is listed in Table 1, and the computational complexity of the whole DCSMP algorithm is listed in Table 2.

Table 1. Complexity analysis for online processing.

Rough Estimation	Data Fusion	Innovation Support Set Estimation	Final Estimation
$O(MD_p)$	Negligible	$O(\|\hat{J}\| \cdot M)+O(M \cdot \|\hat{J}^c\|)$	$N_{co} \cdot O(KM)$

Table 2. Computational complexity of the DCSMP Algorithm.

Offline Processing	Online Processing
$O(\frac{N(N-1)}{2}M)$	$O(MD_p) + O(\|\hat{J}\| \cdot M)+O(M \cdot \|\hat{J}^c\|) + N_{co} \cdot O(KM)$

4.2. Scalability Analysis

In a typical wireless sensor network (WSN) scenario, signals are sampled at source nodes and aggregated at sink nodes. The correlation between source nodes causes redundancy. Many methods are proposed to reduce the redundancy, such as the structure fidelity data collection approach [11] and distributed compressed sensing [22].

This paper considers a decentralized sensor network where a number of densely-placed sensors acquire signals and communicates with neighbor nodes to reconstruct the original signals. All of the sensor nodes are equivalent. Each individual sensor (e.g., the p-th sensor node) is not aware of the full network topology, and instead, it knows two sets of local neighbors: the incoming neighbor connections L_p^{in} and outgoing neighbor connections L_p^{out}. Here, incoming and outgoing connections correspond to communication links where a node can receive or send information, respectively. The sparse signal is sampled and reconstructed at each sensor node, by exploiting the correlation between the sensor nodes. In particular, at the p-th node, the DCSMP algorithm fuses the candidate support set estimates from both the p-th node and its neighboring nodes, to enhance the reconstruction performance.

In the conventional sensor network, as the size of the network expands, the transmission burden and the computational cost of the data fusion process increase dramatically at each node. However, in the DCSMP algorithm-based sensor network, the candidate support sets are transmitted over the network, rather than the transmitted measurements as in the conventional network. This significantly reduces the transmission burden on the links. Moreover, the computational cost of the data fusion process grows linearly with the number of incoming neighboring connections. The computational complexity of the data fusion process is negligible since a very simple democratic voting strategy [12] is adopted in this work. Thus, the DCSMP algorithm provides a desirable structural scalability.

5. Simulation Results and Analysis

In this section, we consider a distributed multistatic radar system, which consists of a transmitter and a number of receivers (Figure 2). The transmitter emits the transmitted signal; the receivers receive the echoes from the targets. It is assumed that different individual receivers observe a same surveillance region with different detection probabilities. A JSM-1 is constructed based on the multistatic radar system, and sparse recovery is carried out in a decentralized manner across various receivers.

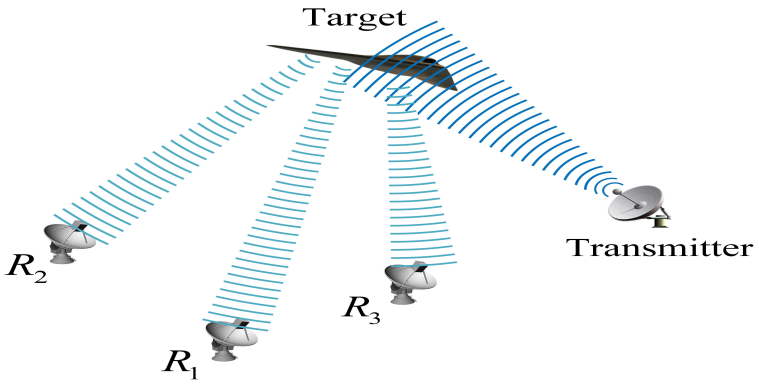


Figure 2. Multistatic radar system.

5.1. Sparse Representation in State Space

We consider N_T targets moving within the surveillance region. The state vector of the d-th target $(d = 1, \cdots, N_T)$ at the k-th scan is defined as $\mathbf{x}_k^d = \left[px_k^d, vx_k^d, py_k^d, vy_k^d, pz_k^d, vz_k^d\right]^T$, where px_k^d and vx_k^d denote respectively the position and velocity of the d-th target along the x axis of Cartesian frame at scan k; py_k^d and vy_k^d along the y axis and pz_k^d and vz_k^d along the z axis.

In the multistatic radar system, a transmitter Tr is at a known position $\mathbf{tr} = [x_0, y_0, z_0]^T$; a number of receivers R_i $(i = 1, \cdots, N_R)$ are placed at known locations $\mathbf{r}_i = [x_i, y_i, z_i]^T, i = 1, \cdots, N_R$, in a Cartesian coordinate system, where N_R denotes the number of receivers.

In practice, the number, locations and velocities of the targets are unknown during the tracking process. The state space at scan k is divided into N_g grids (possible values), listed as $\mathbf{g}_k^l$, $l = 1, \cdots, N_g$. An auxiliary parameter, namely grid reflection, is attached to each grid. If a grid is occupied by a target,

its grid reflection parameter is set as the reflection coefficient of the target; otherwise, it is set as zero. All of the grid reflection parameters are mapped into a grid reflection vector $\boldsymbol{\xi}_k$, which is an indicator vector that contains the true reflectivity of targets at each grid location. Considering that the number of grids occupied by targets is much smaller than that of the total grids in state space, $\boldsymbol{\xi}_k$ is a sparse vector.

Each grid in the state space represents a state vector of a potential target. The l-th grid $\mathbf{g}_k^l$ is transformed to a delay-Doppler set $(\tau_k^{i,l}, f_k^{i,l})$, according to Equations (12) and (13), under the condition of receiver R_i,

$$\tau_k^{i,l} = \frac{1}{c}\Big(||\mathbf{P}_k^l - \mathbf{tr}|| + ||\mathbf{P}_k^l - \mathbf{r}_i||\Big), \tag{12}$$

$$f_k^{i,l} = \frac{f_c}{c}\left(\left\langle \mathbf{V}_k^l, \mathbf{u}_k^{i,l}\right\rangle - \left\langle \mathbf{V}_k^l, \mathbf{u}_k^{tr,l}\right\rangle\right), \tag{13}$$

where $\mathbf{P}_k^l = [px_k^l, py_k^l, pz_k^l]^T$ and $\mathbf{V}_k^l = [vx_k^l, vy_k^l, vz_k^l]^T$ denote the position and velocity of the l-th grid at scan k, respectively; $\mathbf{u}_k^{tr,l}$ and $\mathbf{u}_k^{i,l}$ denote the unit vector from the transmitter to the l-th grid and the unit vector from the l-th grid to the i-th receiver, respectively.

Note: From (12) and (13), it can be seen that a grid in the state space, $\mathbf{g}_k^l$, corresponds to different delay-Doppler set $(\tau_k^{i,l}, f_k^{i,l})$, under the condition of different receivers R_i, $i = 1, \cdots, N_R$.

5.2. Compressed Sensing Model for an Individual Receiver

At the i-th receiver R_i, the received measurement signal can be represented via the grids in target spate space, as:

$$r_k^i(t) = \sum_{l=1}^{N_g}\left\{\alpha_k^{i,l}\cdot\sum_{n=1}^{W} p\Big(t-\Big(n-1\Big)\varepsilon-\Big(k-1\Big)\Delta T-\tau_k^{i,l}\Big)\cdot e^{j2\pi f_k^{i,l}\cdot t}\right\} + w_k^i(t), \tag{14}$$

where $\alpha_k^{i,l}$ denotes the reflection coefficient corresponding to the l-th grid between the transmitter and the i-th receiver; $\tau_k^{i,l}$ is the l-th grid originated delay, and $f_k^{i,l}$ is the Doppler shift frequency of the l-th grid, both measured by the i-th receiver; $w_k^i(t)$ is the complex envelope of the overall disturbance at the i-th receiver.

The state vector corresponding to the l-th grid, $\mathbf{g}_k^l$, contributes to the received signal if it is occupied by a target. We define $\varphi_k^{i,l}(t)$ as the l-th grid's contribution to the received signal, as:

$$\varphi_k^{i,l}(t) = \sum_{n=1}^{W} p\Big(t-\Big(n-1\Big)\varepsilon-\Big(k-1\Big)\Delta T-\tau_k^{i,l}\Big)e^{j2\pi f_k^{i,l}\cdot t}. \tag{15}$$

At the i-th receiver, a sequence of discrete outputs of the received signal is sampled, which forms a measurement vector $\mathbf{y}_k^i$, as:

$$\mathbf{y}_k^i = \Big[r_k^i\Big(1\Big), r_k^i\Big(2\Big), \cdots, r_k^i\Big(W\Big)\Big]^T, \tag{16}$$

where $r_k^i(n), n = 1, \cdots, W$ are discrete output samples collected by the i-th receiver, at scan k.

The measurement vector $\mathbf{y}_k^i$ can be represented in a compressed sensing framework, as in (17),

$$\mathbf{y}_k^i = \boldsymbol{\Phi}_k^i\boldsymbol{\xi}_k^i + \mathbf{e}_k^i, \tag{17}$$

where $\boldsymbol{\Phi}_k^i$ is a sensing matrix and $\boldsymbol{\xi}_k^i$ is a sparse grid reflection vector at the i-th receiver. We have:

$$\boldsymbol{\Phi}_k^i = \Big[\boldsymbol{\varphi}_k^{i,1}\cdots\boldsymbol{\varphi}_k^{i,l}\cdots\boldsymbol{\varphi}_k^{i,N_g}\Big], \tag{18}$$

where $\boldsymbol{\varphi}_k^{i,l}$ is the l-th column of the sensing matrix, i.e., $\boldsymbol{\varphi}_k^{i,l} = [\varphi_k^{i,l}(1), \varphi_k^{i,l}(2), \cdots, \varphi_k^{i,l}(W)]^T$. It can be seen from (15) that $\boldsymbol{\varphi}_k^{i,l}$ is the l-th grid's contribution to the received signal and is deterministic in the condition of fixed grids. Since the division of the state space with grids is assumed fixed prior in this work, the sensing matrix is deterministic in nature.

5.3. General JSM-1 in a Multistatic Radar System

In the multistatic radar system, assume that the i-th receiver R_i has bi-directional communication links with a number of neighboring nodes (receivers), which are denoted as $U_i^j, j = 1, \cdots, NE_i$, $i = 1, \cdots, N_R$, where NE_i denotes the number of neighboring nodes of the i-th receiver. For each neighboring node U_i^j, we can obtain a standard equation in compressed sensing, as:

$$\mathbf{y}_k^j = \boldsymbol{\Phi}_k^j \boldsymbol{\xi}_k^j + \mathbf{e}_k^j, \quad j = 1, \cdots, NE_i, \tag{19}$$

where $\mathbf{y}_k^j$, $\boldsymbol{\Phi}_k^j$, $\boldsymbol{\xi}_k^j$ and $\mathbf{e}_k^j$ denote the measurement vector, sensing matrix, sparse grid reflection vector and noise vector, respectively, for the j-th neighboring node of the i-th receiver.

It is assumed that the i-th receiver and its NE_i neighboring nodes observe a surveillance area. For each individual receiver, it cannot "see" all of the targets at a time. This is reasonable in practice considering that the receivers are located at different positions with different viewing angles, and each receiver has a different detection probability (less than one). Therefore, some targets are observed by all of the receivers, namely common targets; and the others are observed by individual receiver, which are named as innovation targets.

The common targets' states are the same as different observers (receivers) located at different positions, when the targets' states and receivers' positions are defined in the same coordinate system. Thus, for different receivers, their corresponding sparse grid reflection vectors share the same locations of part of the nonzero reflection coefficients (corresponding to the common targets), i.e., having the same common support set. However, the reflection coefficients of a common target observed by different receivers are not the same due to different parameters of each individual receiver; thus, the common components of different sparse grid reflection vectors are different, which does not fit the standard JSM-1 (2) presented in Section 2.

A general JSM-1 is adopted in this work, which relies only on Equation (3), focusing on common support set. In this condition, it is assumed that different sparse grid reflection vectors share a common support set, while having different reflection coefficients. Since the proposed DCSMP algorithm focuses on support set estimation instead of estimating the common component (the reflection coefficients), the proposed DCSMP algorithm can cope with the general JSM-1 efficiently.

5.4. Simulation Environment Setup

A multistatic radar system consisting of a transmitter and three receivers are considered in this simulation example. The transmitter is located at $\mathbf{tr} = [0, 0, 0]^T$ km; the receivers are located at $\mathbf{r}_1 = [0, 8, 0]^T$ km, $\mathbf{r}_2 = [8, 0, 0]^T$ km and $\mathbf{r}_3 = [8, 8, 0]^T$ km, respectively, in a 3D Cartesian coordinate system. The three receivers are connected with bi-directional communication links. The carrier frequency (f_c) for each receiver is 10 GHz. The number of discrete samples collected at each receiver (W) is 80. The volume of the surveillance position space (represented in px_k, py_k, pz_k) is $10^3 \times 10^3 \times 10^3$ m^3, which is divided into $10 \times 10 \times 10$ grid points; the volume of the surveillance velocity space (represented in vx_k, vy_k, vz_k) is $60 \times 60 \times 60$ (m/s)^3, which is divided into $2 \times 2 \times 2$ grid points. Therefore, the total number of grids in state space (N_g) is 8×10^3.

5.5. Identification of Multiple Targets' States

The proposed DCSMP algorithm is used to identify multiple targets (including common and innovation targets) in the state space. The achievable resolution of the sparse vector in state space

obtained by the DCSMP algorithm is evaluated and compared with the DIPP algorithm. We focus on the difference in position while assuming that all of the targets have the same velocity, for clarity in representation.

Three cases are considered in the simulation example: (a) separately-distributed targets; (b) closely-spaced common targets; (c) closely-spaced innovation targets observed by the receiver R1. The addressed scenarios are characterized by the signal-to-noise ratio (SNR), which is set to 20 dB. The simulation parameters for the three cases are listed in Tables 3–5.

Table 3. Case 1: Separately-distributed targets.

Target Number	Reflected Amplitude (dB)	Positions (m)	Observer
$C1$	2	$[1800, 2900, 18000]^T$	R1,R2,R3
$C2$	6	$[1900, 5100, 18100]^T$	R1,R2,R3
$I1$	13	$[4000, 1000, 18000]^T$	R1
$I2$	2	$[3500, 900, 17900]^T$	R2
$I3$	6	$[500, 500, 17500]^T$	R3

Table 4. Case 2: Closely-spaced common targets.

Target Number	Reflected Amplitude (dB)	Positions (m)	Observer
$C1$	2	$[1800, 4900, 18000]^T$	R1,R2,R3
$C2$	6	$[1700, 5000, 18000]^T$	R1,R2,R3
$I1$	13	$[3600, 600, 18100]^T$	R1
$I2$	2	$[3000, 1800, 17800]^T$	R2
$I3$	6	$[510, 500, 17500]^T$	R3

Table 5. Case 3: Closely-spaced innovation targets observed by R1.

Target Number	Reflected Amplitude (dB)	Positions (m)	Observer
$C1$	2	$[1800, 4900, 18000]^T$	R1,R2,R3
$C2$	6	$[1800, 2200, 18100]^T$	R1,R2,R3
$I11$	13	$[1800, 5000, 17900]^T$	R1
$I12$	13	$[1700, 4900, 17900]^T$	R1
$I2$	2	$[2500, 1500, 17400]^T$	R2
$I3$	6	$[500, 1500, 17500]^T$	R3

The simulation results are shown in Figures 3–8. The estimated positions shown in each figure are the combination of the results from three receivers, including the common targets observed by all of the receivers and the innovation targets observed by the individual receiver. This is reasonable since in practice, at the final stage of estimation, each receiver will send its estimated locations of the innovation targets to its neighboring nodes, and all of the connected receivers will have a common scene of the surveillance area.

5.5.1. Separately Distributed Targets

Figures 3 and 4 show the results of the identification of multiple separately-distributed targets in the state space, using the DCSMP and DIPP algorithm, respectively. In Figure 3, the true positions of targets are denoted as stars ("*"), with text indications "C1" and "C2" for common targets and "I1", "I2", "I3" for innovation targets. The estimated positions of targets are denoted as circles ("o"), with text indications "$C1_{estimate}$" and "$C2_{estimate}$" for estimated common targets and "$I1_{estimate}$", "$I2_{estimate}$", "$I3_{estimate}$" for estimated innovation targets. It can be seen from Figure 3 that the multiple separately-distributed targets (including both the common and innovation targets) are accurately

identified in the state space using the DCSMP algorithm, and similar results appear in Figure 4, for the DIPP algorithm.

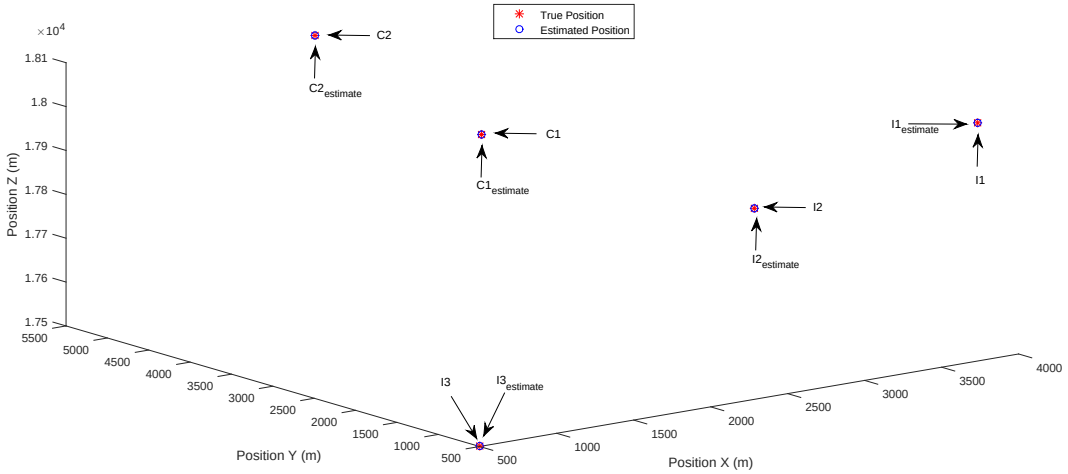


Figure 3. Case 1: Separately-distributed targets. Estimated positions in 3D space using the DCSMP algorithm.

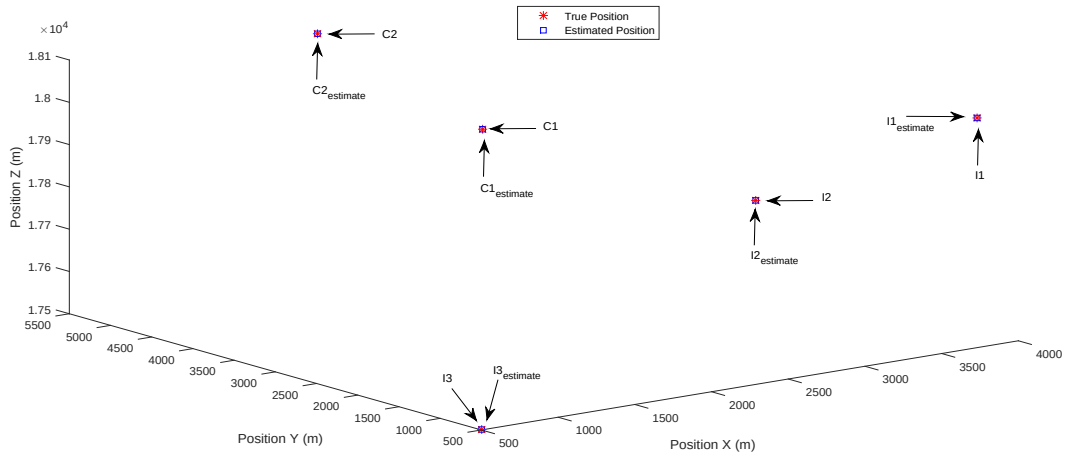


Figure 4. Case 1: Separately-distributed targets. Estimated positions in 3D space using the distributed parallel pursuit (DIPP) algorithm.

5.5.2. Closely-Spaced Common Targets

The main challenge of this scenario arises from the small separation between two closely-spaced common targets $C1$ and $C2$. Figure 6 shows that the DIPP algorithm fails in distinguishing $C1$ and $C2$. This is due to the reason that the DIPP algorithm cannot efficiently cope with the sensing matrix with high coherence due to the high resolution of the state space. Moreover, since the estimated common support set (i.e., estimated positions of the common targets) is adopted as the side information to calculate the innovation support set by the DIPP algorithm, the wrongly-estimated common support set leads to failure in estimating the innovation support set, for each receiver. Therefore, the positions of the innovation targets cannot be identified accurately (Figure 6).

In Figure 5, the common targets are accurately identified, which verifies that the proposed DCSMP algorithm is capable of dealing with the sensing matrix with high coherence. As a consequence, the innovation targets are accurately identified based on the correctly-estimated common support set.

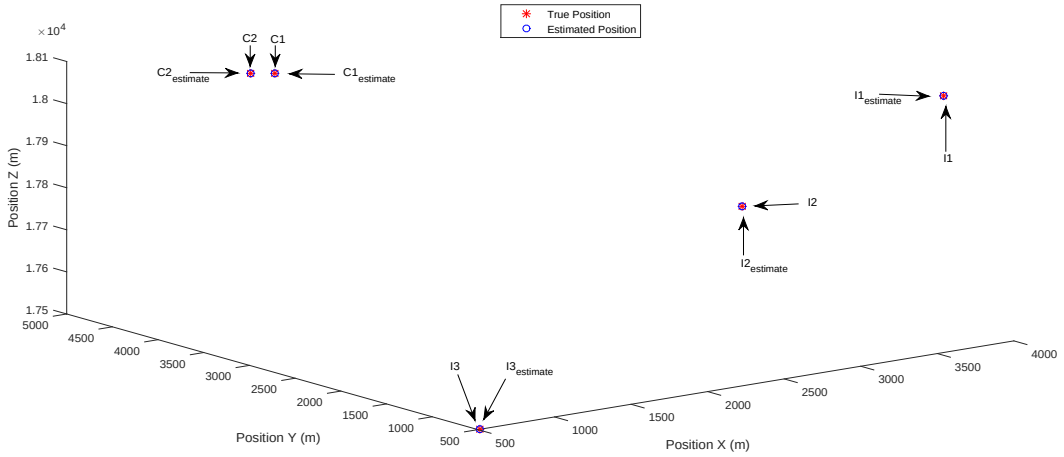


Figure 5. Case 2: Closely-spaced common targets. Estimated positions in 3D space using the DCSMP algorithm.

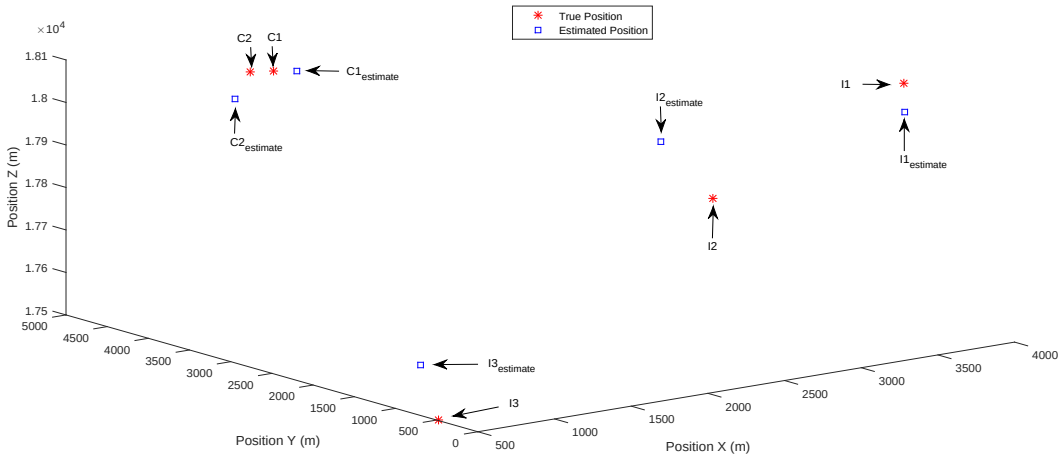


Figure 6. Case 2: Closely-spaced common targets. Estimated positions in 3D space using the DIPP algorithm.

5.5.3. Closely-Spaced Innovation Targets

The main challenge in this scenario arises from the small separation between two closely-spaced innovation targets $I11$ and $I12$, observed by the receiver R1. Figure 8 shows that the DIPP algorithm succeeds in estimating the positions of two separately-distributed common targets. The estimated common support set is then input as the side information to calculate the innovation support set, at three receivers, respectively. Figure 8 shows that the first receiver R1 fails in distinguishing the two closely-spaced innovation targets $I11$ and $I12$, while the other two receivers R2 and R3 succeed in identifying their corresponding innovation targets I2 and I3, respectively. In Figure 7, it can be seen that the proposed DCSMP algorithm succeeds in identifying both the common and innovation targets.

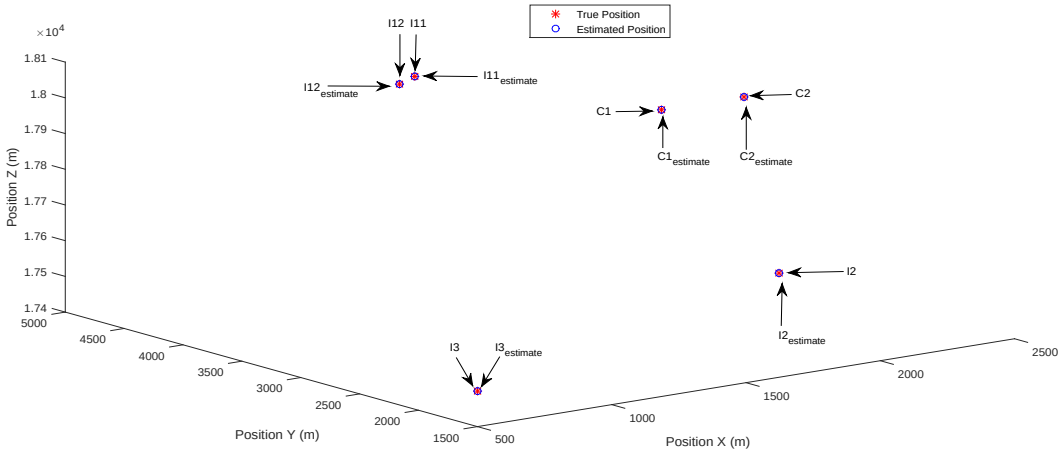


Figure 7. Case 3: Closely-spaced innovation targets observed by receiver R1. Estimated positions in 3D space using the DCSMP algorithm.

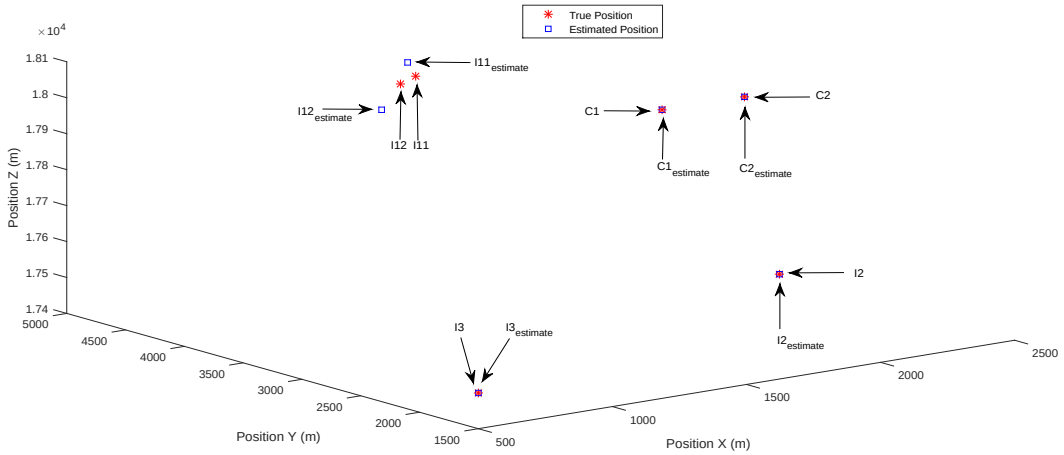


Figure 8. Case 3: Closely-spaced innovation targets observed by R1. Estimated positions in 3D space using the DIPP algorithm.

5.6. Random Testing Cases

A sequence of random cases is tested where 20 targets (including common and innovation targets) are randomly distributed in the three-dimensional position space. Five hundred Monte Carlo simulations are performed for each trail. Considering that each individual receiver cannot "see" all of the targets at a time, the numbers of common and innovation targets are set as 15 and 5, respectively. Thus, we have the following parameter setup for the simulation: the signal dimensionality $N = 8 \times 10^3$, the measurement dimensionality $M = 80$, the sparsity level $K = 20$, the common support set $J = 15$, the innovation support set $I_p = 5$, the number of all nodes (receivers) $N_R = 3$ and the number of Monte Carlo trails $N_{MC} = 500$.

Two metrics are utilized to evaluate the performance of the proposed algorithm. The first metric is called distributed reconstruction error (DRE), which is defined as:

$$DRE = \frac{1}{|N_R|}\Sigma_{p=1}^{|N_R|}\left[\frac{1}{N_{MC}}\Sigma_{i=1}^{N_{MC}}\frac{||\hat{\mathbf{x}}_p^i - \mathbf{x}_p^i||_2}{||\mathbf{x}_p^i||_2}\right], \tag{20}$$

where $\mathbf{x}_p^i$ represents the true signal for the p-th receiver in the i-th Monte Carlo trail and $\hat{\mathbf{x}}_p^i$ represents the estimated signal. Our objective is to achieve a lower DRE considering the whole decentralized network. We also adopt the average support-set cardinality error (ASCE) as a direct evaluation of the support-set recovery performance [12]. Note that the ASCE has the range $[0, 1]$, and our objective is to achieve a lower ASCE.

We now provide the average performance results using the performance measures DRE and ASCE, which are shown in Table 6. The performance of OMP and SP algorithms is included in the simulation as a benchmark characterizing a single-sensor (disconnected) scenario. From Table 6, it can be seen that the proposed DCSMP algorithm obtains the lowest DRE and ASCE, which verifies that the proposed DCSMP algorithm outperforms the three other algorithms (DIPP, SP and OMP), in reconstructing the sparse vector with high resolution, in a decentralized network.

Table 6. Performance results. DRE, distributed reconstruction error; ASCE, average support-set cardinality error.

Algorithm	DRE	ASCE
DCSMP	0.07	0.05
DIPP	0.23	0.17
SP	0.37	0.53
OMP	0.31	0.49

5.7. Simulation Results on a Large-Scale Sensor Network

This simulation is to validate the performance of the proposed DCSMP algorithm on a large-scale sensor network. The network topology is built along the random geometric graph model, presented by Penrose in [23], where a number of nodes are randomly distributed in an area. Each node connects with its neighbor nodes located within a certain distance from it. Figure 9 shows a typical setup of such a network consisting of 105 nodes. The proposed DCSMP algorithm is compared to the compressed sensing-based algorithms, e.g., the SP, OMP and DIPP algorithms, as well as non compressed sensing based algorithms, e.g., principal component analysis (PCA) [24] and the distributed wavelet compression (DWC) algorithm [25].

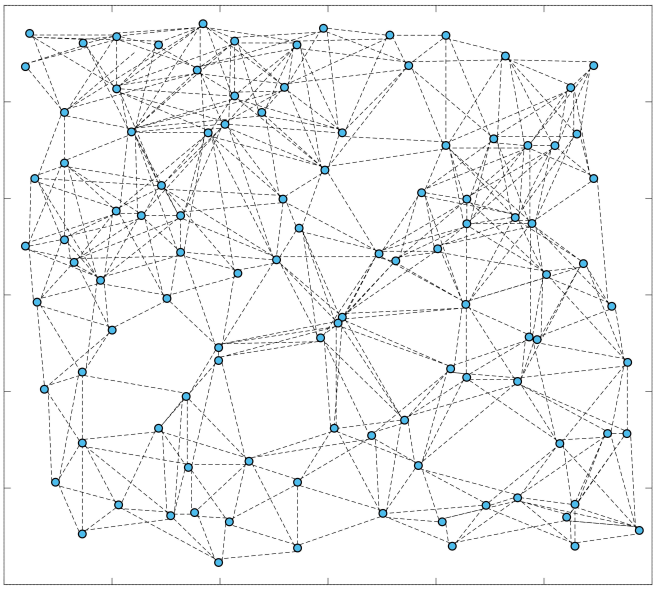
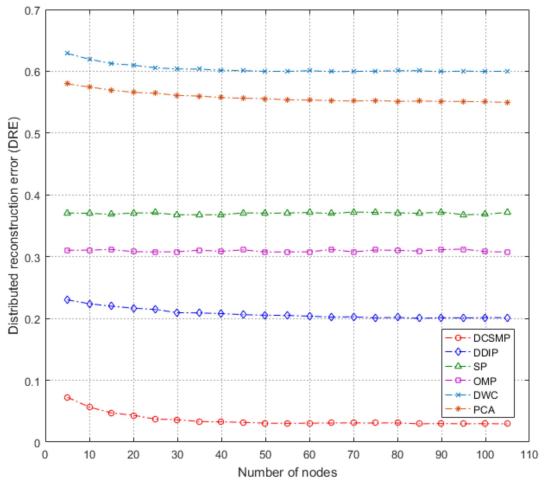


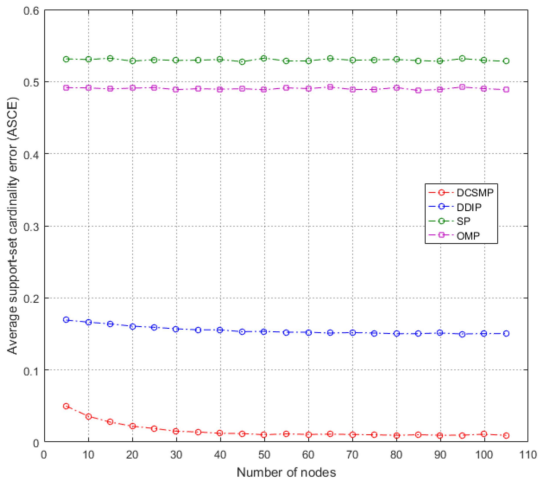
Figure 9. The connectivity topology of the tested network comprised of 105 nodes.

The signal model is constructed based on the multistatic radar system. The sparse grid reflection vector is chosen as the original signal. For the conventional compressed sensing-based algorithms, e.g., the SP and OMP algorithm, the measurements are compressed by a down-sampling measurement matrix before transmitting to the neighboring nodes. For the non-compressed sensing-based approaches, e.g., the PCA and DWC algorithm, the data compression is achieved after aggravating signals from neighboring nodes. For the above two kinds of algorithms, the measurements are transmitted in the network. In contrast, in the DCSMP and DDIP algorithm, only the estimated candidate support sets are transmitted between nodes in the network, which significantly reduces the transmission burden.

A sequence of random cases is tested where a number of targets (including common and innovation targets) are randomly distributed in the three-dimensional position space. The simulation parameters are the same as those in Section 5.6. Two metrics, DRE and ASCE, are utilized to evaluate the reconstruction performance of different algorithms. Figure 10a shows the variation of DRE with different number of nodes, for different algorithms. It can be seen from Figure 10a that the proposed DCSMP algorithm achieves the lowest DRE. The non-compressed sensing-based algorithms (e.g., PCA and DWC) cannot reconstruct the original signal perfectly at a down-sampling rate, thus they achieve large DREs. Though the conventional compressed sensing algorithms (e.g., SP and OMP) and DIPP algorithm can reconstruct the original signal accurately at a down-sampling rate, they cannot cope with the sensing matrix with high coherence, thus achieving moderate DREs, compared with the proposed DCSMP algorithm. Moreover, it can be seen from Figure 10a that the DREs of the two algorithms, DCSMP and DDIP, slightly decrease with the increase of network size. This is due to the reason that the information (the candidate support set or the measurements) received by each node increases with the size of the network expands, thus resulting in a more accurate estimated signal. Figure 10b shows the variation of ASCE with different numbers of nodes. The proposed DCSMP algorithm achieves the lowest ASCE. This verifies that the proposed DCSMP algorithm outperforms the three other algorithms (DIPP, SP and OMP) in dealing with the sensing matrix with high coherence.



(a) Variation of DRE with different number of nodes



(b) Variation of ASCE with different number of nodes

Figure 10. Performance comparison of different algorithms with a varying number of nodes. (**a**) Variation of DRE with different numbers of nodes; (**b**) variation of ASCE with different numbers of nodes.

6. Conclusions

A novel DCSMP algorithm is proposed to tackle the JSM-1 in a decentralized scenario. The proposed algorithm adopts deterministic sensing matrices built directly on the real acquisition systems. In the proposed algorithm, the process of estimating the common support set is to decouple

that of estimating the innovation support set. Thus, the inaccurately estimated common support set will have no impact on estimating the innovation support set. The simulation results show that the proposed algorithm can perform successful sparse recovery in a decentralized manner across various receivers, in a multistatic radar system.

Acknowledgments: This work was supported by the Natural Science Foundations of China (Nos. 61573276 and 61473217), the National 973 project of China through Grant 2013CB329405 and the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (61221063).

Author Contributions: Jing Liu developed the distributed compact sensing matrix pursuit (DCSMP) algorithm and wrote the paper; Kaiyu Huang answered part of the reviewers' questions and wrote part of the paper; Guoxian Zhang checked the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lin, L.; Liu, F.; Jiao, L. Compressed Sensing by Collaborative Reconstruction on Overcomplete Dictionary. *Signal Process.* **2014**, *103*, 92–102.
2. Wang, J.C.; Lee, Y.S.; Lin, C.H.; Wang, S.F.; Shih, C.H.; Wu, C.H. Compressive Sensing-Based Speech Enhancement. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2016**, *24*, 2122–2131.
3. Wang, D.; Wan, J.; Chen, J. An Online Dictionary Learning-Based Compressive Data Gathering Algorithm in Wireless Sensor Networks. *Sensors* **2016**, *16*, 1547.
4. Yin, J.; Yang, Y.; Wang, L. An Adaptive Data Gathering Scheme for Multi-Hop Wireless Sensor Networks Based on Compressed Sensing and Network Coding. *Sensors* **2016**, *16*, 462.
5. Bu, H.; Tao, R.; Bai, X. Regularized Smoothed l^0 Norm Algorithm and its Application to CS-based Radar Imaging. *Signal Process.* **2016**, *122*, 115–122.
6. Baron, D.; Wakin, M.; Duarte, M.; Sarvotham, S.; Baraniuk, R. Distributed compressed sensing. Available online: http://ai2-s2-pdfs.s3.amazonaws.com/6388/e944d5929cb6490792429085ed31c0ad776b.pdf (accessed on 20 April 2017).
7. Zhang, Q.; Fu, Y.; Li, H.; Rong, R. Optimised projections for generalised distributed compressed sensing. *Electron. Lett.* **2014**, *50*, 520–521.
8. Zeng, F.; Li, C.; Tian, Z. Distributed compressive spectrum sensing in cooperative multihop cognitive networks. *IEEE Trans. Signal Process.* **2011**, *5*, 37–48.
9. Tan, L.; Wu, M. Data reduction in wireless sensor networks: A hierarchical LMS prediction approach. *IEEE Sens. J.* **2016**, *16*, 1708–1715.
10. Wu, M.; Tan, L.; Xiong, N. Data prediction, compression, and recovery in clustered wireless sensor networks for environmental monitoring applications. *Inf. Sci.* **2016**, *329*, 800–818.
11. Wu, M.; Tan, L.; Xiong, N. A Structure Fidelity Approach for Big Data Collection in Wireless Sensor Networks. *Sensors* **2015**, *15*, 248–273.
12. Sundman, D.; Chatterjee, S.; Skoglund, M. Design and Analysis of a Greedy Pursuit for Distributed Compressed Sensing. *IEEE Trans. Signal Process.* **2016**, *64*, 2803–2818.
13. Baron, D.; Duarte, M.F.; Sarvotham, S.; Wakin, M.B.; Baraniuk, R.G. An information theoretic approach to distributed compressed sensing. In Proceedings of the 43rd Allerton Conference Communication Control, and Computing, Monticello, IL, USA, 28–30 September 2005.
14. Schnelle, S.R.; Laska, J.N.; Hegde, C.; Duarte, M.F.; Davenport, M.A.; Baraniuk, R.G. Texas hold 'Em algorithms for distributed compressive sensing. In Proceedings of the IEEE International Conference Acoustics, Speech and Signal Processing (ICASSP), Dallas, TX, USA, 14–19 March 2010; pp. 2886–2889.
15. Coluccia, G.; Magli, E.; Roumy, A.; Toto-Zarasoa, V. Lossy compression of distributed sparse sources: A practical scheme. In Proceedings of the 2011 19th European Signal Processing Conference (EUSIPCO 11), Barcelona, Spain, 29 August–2 September 2011; pp. 422–426.
16. Valsesia, D.; Coluccia, G.; Magli, E. Joint recovery algorithms using difference of innovations for distributed compressed sensing. In Proceedings of the 2013 Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, USA, 3–6 November 2013; pp. 414–417.
17. Matamoros, J.; Fosson, S.M.; Magli, E.; Antón-Haro, C. Distributed ADMM for In-Network Reconstruction of Sparse Signals With Innovations. *IEEE Trans. Signal Inf. Process. Netw.* **2015**, *1*, 225–234.

18. Liu, J.; Mallick, M.; Han, C.Z; Yao, X.H.; Lian, F. Similar sensing matrix pursuit: An efficient reconstruction algorithm to cope with deterministic sensing matrix. *Signal Process.* **2014**, *95*, 101–110.
19. Liu, J.; Mallick, M.; Lian, F.; Han, C.Z.; Sheng, M.X.; Yao, X.H. General similar sensing matrix pursuit: An efficient and rigorous reconstruction algorithm to cope with deterministic sensing matrix with high coherence. *Signal Process.* **2015**, *114*, 150–163.
20. Foucart, S.; Rauhut, H. *A Mathematical Introduction to Compressive Sensing*; Applied and Numerical Harmonic Analysis; Birkhäuser and Springer: New York, NY, USA, 2013.
21. Needell, D.; Tropp, J.A. COSAMP: Iterative signal recovery from incomplete and inaccurate samples. *Appl. Comput. Harmonic Anal.* **2009**, *26*, 301–321.
22. Chen, W.; Rodrigues, M.R.D.; Wassell, I.J. A Frechet Mean Approach for Compressive Sensing Date Acquisition and Reconstruction in Wireless Sensor Networks. *IEEE Trans. Wirel. Commun.* **2012**, *11*, 3598–3606.
23. Penrose, M. *Random Geometric Graphs*; Oxford University Press: Oxford, UK, 2004.
24. Chen, F.; Wen, F.; Jia, H. Algorithm of Data Compression Based on Multiple Principal Component Analysis over the WSN. In Proceedings of the 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM), Chengdu, China, 23–25 September 2010; pp. 1–4.
25. Ciancio, A.; Pattem, S.; Ortega, A.; Krishnamachari, B. Energy-efficient data representation and routing for wireless sensor networks based on a distributed wavelet compression algorithm. In Proceedings of the Fifth International Conference on Information Processing in Sensor Networks (IPSN 2006), Nashville, TN, USA, 19–21 April 2006; pp. 309–316.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).