Intelligent Sensor Networks: Across Sensing, Signal Processing, and Machine Learning

Chapter: Compressive sensing and its application in wireless sensor networks

Authors: Jae-Gun Choi, Sang-Jun Park, and Heung-No Lee*

Gwangju Institute of Science and Technology (GIST)

261 Cheomdan-gwagiro, Buk-gu, Gwang-ju, South Korea

Email: {jgchoi, sjpark1, heungno*}@gist.ac.kr

Editor: Fei Hu

Department of Electrical and Computer Engineering

The University of Alabama

P.O. BOX 870286 TUSCALOOSA, AL 35487

Email: fei@eng.ua.edu

Publisher: Taylor & Francis LLC, CRC Press, 2012.

CRC Press, 711 Third Avenue, New York, NY 10017

Tel: 917-351-7146; Fax: 212-947-3027

Email: rich.ohanley@taylorandfrancis.com

Publication Data: April, 2012

Contents

1. Introduction

In this chapter, we discuss the application of a new compression technique called compressive sensing (CS) in wireless sensor networks (WSNs). The objective of a WSN we assume in this chapter is to collect information about events occurring in a region of interest. This WSN consists of a large number of wireless sensor nodes and a central fusion center (FC). The sensor nodes are spatially distributed over the said region to acquire physical signals such as sound, temperature, wind speed, pressure, and seismic vibrations. After sensing, they transmit the measured signals to the FC. In this chapter, we focus on the role of the FC which is to recover the transmitted signals in their original waveforms for further processing. By doing so, the FC can produce a global picture that illustrates the event occurring in the sensed region. Each sensor uses its onboard battery for sensing activities and makes reports to FC via wireless transmissions. Thus, limited power at the sensor nodes is the key problem to be resolved in the said WSN.

CS is a signal acquisition and compression framework recently developed in the field of signal processing and information theory [1],[2]. Donoho [1] says that "The Shannon–Nyquist sampling rate may lead to too many samples; probably not all of them are necessary to reconstruct the given signal. Therefore, compression may become necessary prior to storage or transmission." According to Baraniuk [3], CS provides a new method of acquiring compressible signals at a rate significantly below the Nyquist rate. This method employs non-adaptive linear projections that preserve the signal's structure; the compressed signal is then reconstructed from these projections using an optimization process. There are two tutorial articles good for further reading on CS [3],[4] published in the IEEE Signal Processing Magazine in 2007 and 2008, respectively.

Our aim in this chapter is to determine whether the CS can be used as a useful framework for the aforementioned WSN to compress and acquire signals and save transmittal

and computational power at the sensor node. This CS based signal acquisition and compression is done by a simple linear projection at each sensor node. Then, each sensor transmits the compressed samples to the FC; the FC which collects the compressed signals from the sensors jointly reconstructs the signal in polynomial time using a signal recovery algorithm. Illustrating this process in detail throughout this chapter, we check to see if CS can become an effective, efficient strategy to be employed in WSNs, especially for those with low-quality, inexpensive sensors.

In this chapter, as we assume a scenario in which a WSN is used for signal acquisition, we intend to pay some effort in modeling correlation between the signals acquired from the sensors. We discuss a few signal projection methods suggested in the literature which are known to give a good signal recovery performance from the compressed measurements. We also investigate a couple of well known signal recovery algorithms such as the orthogonal matching pursuit (OMP) (greedy approach) [13], and the primal-dual interior point method (PDIP) (gradient-type approach) [15]. Finally, we simulate the considered WSN system and examine how the presence of signal correlation can be exploited in the CS recovery routine and help reduce the amount of signal samples to be transmitted at the sensor node.

2. Compressed sensing: What is it?

In a conventional communication system, an analog-to-digital converter based on the Shannon–Nyquist sampling theorem is used to convert analog signals to digital signals. The theorem says that if a signal is sampled at a rate twice, or higher, the maximum frequency of the signal, the original signal can be exactly recovered from the samples. Once the sampled signals are obtained over a fixed duration of time, a conventional compression scheme can be used to compress them. Because the sampled signals often have substantial redundancy, compression is possible. Several compression schemes follow this approach, e.g., the MP3 and JPEG formats for audio or image data. However, conventional compression in a digital system is sometimes inefficient because it requires unnecessary signal processing stages, for example, retaining all of the sampled signals in one location before data compression. According to Donoho [1], the CS framework, as shown in **Figure 1**, can bypass these intermediate steps, and thus provides a light weight signal acquisition apparatus which is suitable for those sensor nodes in our WSN.

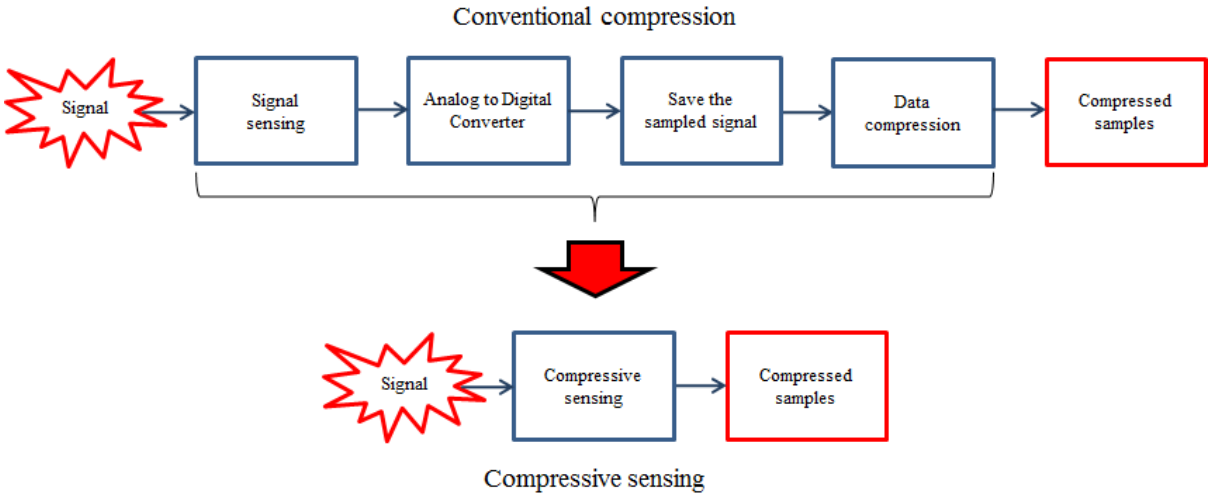

Figure 1. Conventional compression and Compressive sensing

The CS provides a direct method which acquires compressed samples without going

through the intermediate stages of conventional compression. Thus, CS provides a much simpler signal acquisition solution. In addition, the CS provides several recovery routines which the original signal can be regenerated perfectly from the compressed samples.

2.1. Background

Let a real-valued column vector $\mathbf{s}$ be a signal to be acquired. Let it be represented by

$$\mathbf{s} = \Psi \mathbf{x}, \tag{1}$$

where $\mathbf{x}$ and $\mathbf{s} \in \mathbf{R}^n$, and $\mathbf{x}$ is also a real-valued column vector. The matrix $\Psi \in \mathbf{R}^{n \times n}$ is an orthonormal basis, i.e., $\Psi^T \Psi = \Psi \Psi^T = I_N$, the identity matrix of size $\mathbf{R}^{n \times n}$. The signal $\mathbf{s}$ is called k-sparse if it can be represented as a linear combination of only k columns of Ψ, i.e., only the k components of the vector $\mathbf{x}$ are nonzero as represented Eq. (2).

$$\mathbf{s} = \sum_{i=1}^{n} x_i \psi_i \text{, where } \psi_i \text{ is a column vector of } \Psi. \tag{2}$$

A signal is called *compressible* if it has only a few significant (large in magnitude) components and a greater number of insignificant (close to zero) components. The compressive measurements $\mathbf{y}$ (compressed samples) are obtained via linear projections as follows:

$$\begin{aligned} \mathbf{y} &= \Phi \mathbf{s} \\ &= \Phi \Psi \mathbf{x} \\ &= \mathbf{A}\mathbf{x} \quad , \end{aligned} \tag{3}$$

where the measurement vector is $\mathbf{y} \in \mathbf{R}^m$, with $m < n$, and the measurement matrix $\mathbf{A} \in \mathbf{R}^{m \times n}$. Our goal is to recover $\mathbf{x}$ from the measurement vector $\mathbf{y}$. We note that Eq. (3) is an underdetermined system because it has fewer equations than unknowns; thus, it does not have a unique solution in general. However, the theory of CS asserts that, if the vector $\mathbf{x}$ is sufficiently sparse, an underdetermined system is guaranteed with high probability to have a unique solution.

In this section, we discuss the basics of CS in more detail.

i) k-sparse signal $\mathbf{x}$ in orthonormal basis

The k-sparse signal, $\mathbf{s}$ in Eq. (1), has k nonzero components in $\mathbf{x}$. The matrix Ψ is, again, an orthonormal basis, i.e., $\Psi^T \Psi = \Psi \Psi^T = I_N$, the identity matrix of size $\mathbf{R}^{n \times n}$.

ii) Measurement vector $\mathbf{y}$ and underdetermined system

$$\mathbf{y} = \Phi \mathbf{s} = \Phi \Psi \mathbf{x} = \mathbf{A}\mathbf{x} \tag{4}$$

The sensing matrices are Φ and $\mathbf{A}$, where $m < n$. When m is closer to k than n is, sufficient conditions for good signal recovery are satisfied. Then a compression effect exists. Note that Eq. (4) appears to be an ill-conditioned equation. That is, the number of unknowns n is larger than m the number of equations, $m < n$. However, if $\mathbf{x}$ is k-sparse and the locations of the k nonzero elements are known, the problem can be solved provided $m \geq k$. We can form a simplified equation by deleting all those columns and elements corresponding to the zero-elements, as follows:

$$\mathbf{y} = \mathbf{A}_{\kappa}\mathbf{x}_{\kappa}, \tag{5}$$

where $\kappa \in \{1,2,\cdots,n\}$ is the support set, which is the collection of indices corresponding to the nonzero elements of **x**. Note that the support set κ can be any size-k subset of the full index set, $\{1,2,3,...,n\}$. Eq. (5) has the unique solution $\mathbf{x}_{\kappa}$ if the columns of $\mathbf{A}_{\kappa}$ are linearly independent. The solution can be found using

$$\mathbf{x}_{\kappa} = \left(\mathbf{A}_{\kappa}^{T}\mathbf{A}_{\kappa}\right)^{-1}\mathbf{A}_{\kappa}^{T}\mathbf{y}. \tag{6}$$

Thus, if the support set κ can be found, the problem is easy to solve provided the columns are linearly independent.

iii) Incoherence condition

The incoherence condition is that the rows of Φ should be incoherent to the columns of Ψ. If the rows of Φ are coherent to the columns of Ψ, the matrix **A** cannot be a good sensing matrix. In the extreme case, we can show a matrix **A** having m rows of Φ that are the first m columns of Ψ.

$$\mathbf{A} = \Phi\Psi = \Psi_{(1:m,:)}^{T}\Psi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \tag{7}$$

If **A** of Eq. (7) is used as sensing matrix, the compressed measurement vector **y** captures only the first m elements of the vector **x**, and the rest of the information contained in

$\mathbf{x}$ is completely lost.

iv) Designing a sensing matrix Φ

One choice for designing a sensing matrix Φ is Gaussian. Under this choice, the sensing matrix Φ is designed as a Gaussian, i.e., matrix elements are independent and identically distributed Gaussian samples. This choice is deemed good since a Gaussian sensing matrix satisfies the incoherence condition with high probability for any choice of orthonormal basis Ψ. This randomly generated matrix acts as a random projection operator on the signal vector $\mathbf{x}$. Such a random projection matrix needs not depend on specific knowledge about the source signals. Moreover, random projections have the following advantages in the application to sensor networks [5].

1) *Universal incoherence*: Random matrices Φ can be combined with all conventional sparsity basis Ψ, and with high probability sparse signals can be recovered by an L_1 minimum algorithms from the measurements $\mathbf{y}$.

2) *Data independence*: The construction of a random matrix does not depend on any prior knowledge of the data. Therefore, given an explicit random number generator, only the sensors and the fusion center are required to agree on a single random seed for generating the same random matrices of any dimension.

3) *Robustness*: Transmission of randomly projected coefficients is robust to packet loss in the network. Even if part of the elements in measurement $\mathbf{y}$ is lost, the receiver can still recover the sparse signal, at the cost of lower accuracy.

2.2. The L_0, L_1, and L_2 norms

In CS, a core problem is to find a unique solution for an underdetermined equation. This problem is related to the signal reconstruction algorithm, which takes the measurement vector $\mathbf{y}$ as an input and outputs the k-sparse vector $\mathbf{x}$. To solve an underdetermined problem, we consider minimization criteria using different norms such as the L_2, L_1, and L_0 norms. The L_p norm of a vector $\mathbf{x}$ of length N is defined as

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{N} |x_i|^p \right)^{\frac{1}{p}}, \quad p > 0 . \tag{8}$$

Although we can define the L_2 and L_1 norms as $\|\mathbf{x}\|_2 = \left(\sum_{i=1}^{N} |x_i|^2 \right)^{\frac{1}{2}}$ and $\|\mathbf{x}\|_1 = \sum_{i=1}^{N} |x_i|$, respectively, using the definition of L_p norm, L_0 norm cannot be defined this way. The L_0 norm is a pseudo-norm that counts the number of nonzero components in a vector as defined by Donoho and Elad [6]. Using this definition of norms, we will discuss the minimization problem.

i) L_2 norm minimization

$$\begin{aligned} (L_2)\ \hat{\mathbf{x}} &= \arg\min \|\mathbf{x}\|_2 \ \text{ subject to } \mathbf{y} = \mathbf{A}\mathbf{x}, \ \text{where } \mathbf{A} \in \mathrm{R}^{m \times n}, \ rank(\mathbf{A}) = m \\ &= \mathbf{A}^T \left(\mathbf{A}\mathbf{A}^T \right)^{-1} \mathbf{y} \end{aligned} \tag{9}$$

However, this conventional solution yields a non-sparse solution, so it is not appropriate as a solution to the CS problem.

ii) L_0 norm minimization

$$\left(L_0\right) \text{ Minimize } \left\|\mathbf{x}\right\|_0 \text{ subject to } \mathbf{y} = \mathbf{A}\mathbf{x}, \text{ where } \mathbf{A} \in \mathrm{R}^{m\times n},\ rank\left(\mathbf{A}\right) = m \tag{10}$$

The L_0 norm of a vector is, by definition, the number of nonzero elements in the vector. In the CS literature, it is known that the L_0 norm problem can be solved by examining all the possible cases. Since this process involves a combinatorial search for all possible $\binom{n}{k}$ support sets, it is an NP-complete problem. Thus, we cannot solve it within polynomial time. Therefore, we consider L_1 norm minimization as an alternative.

iii) L_1 norm minimization

$$\left(L_1\right) \text{ Minimize } \left\|\mathbf{x}\right\|_1 \text{ subject to } \mathbf{y} = \mathbf{A}\mathbf{x}, \text{where } \mathbf{A} \in \mathrm{R}^{m\times n},\ rank\left(\mathbf{A}\right) = m \tag{11}$$

This L_1 norm minimization can be considered as a relaxed version of the L_0 problem. Fortunately, the L_1 problem is a convex optimization problem and in fact can be recast as a linear programming problem. For example, it can be solved by an interior point method. Many effective algorithms have been developed to solve the minimum L_1 problem, and it will be considered later in this chapter. Here, we aim to study the sufficient conditions under which Eq. (10) and (11) have unique solutions. We provide a theorem related to this issue.

L_0 / L_1 equivalence condition:

Let $\mathbf{A} \in \mathbf{R}^{m \times n}$ be a matrix with a maximum correlation definition μ, $\mu(\mathbf{A}) = \max_{i \neq j} \left| \langle \mathbf{a}_i, \mathbf{a}_j \rangle \right|$, where $\mathbf{a}_i$ is the i th column vector of $\mathbf{A}$ with $i = 1, 2, ..., n$, and $\mathbf{x}$ is a k-sparse signal. Then, if $k < \frac{1}{2}\left(1 + \frac{1}{\mu}\right)$ is satisfied, then the solution of L_1 coincides with that of L_0 [6].

Theorem 1. L_0 / L_1 Equivalence condition.

3. Wireless sensor networks

3.1. Network structure

We consider a WSN consisting of a large number of wireless sensor nodes and one FC (**Figure 2**). The wireless sensor nodes are spatially distributed over a region of interest and observe physical changes such as those in sound, temperature, pressure, or seismic vibrations. If a specific event occurs in a region of distributed sensors, each sensor makes local observations of the physical phenomenon as the result of this event taking place. An example of sensor network applications is area monitoring to detect forest fires. A network of sensor nodes can be installed in a forest to detect when a fire breaks out. The nodes can be equipped with sensors to measure temperature, humidity, and the gases produced by fires in trees or vegetation [7]. Other examples include military and security applications. Military applications vary from monitoring soldiers in the field, to tracking vehicles or enemy movement. Sensors attached to soldiers, vehicles and equipment can gather information about their condition and location to help planning activities on the battlefield. Seismic, acoustic and video sensors can be deployed to monitor critical terrain and approach routes; reconnaissance of enemy terrain and forces can be carried out [8].

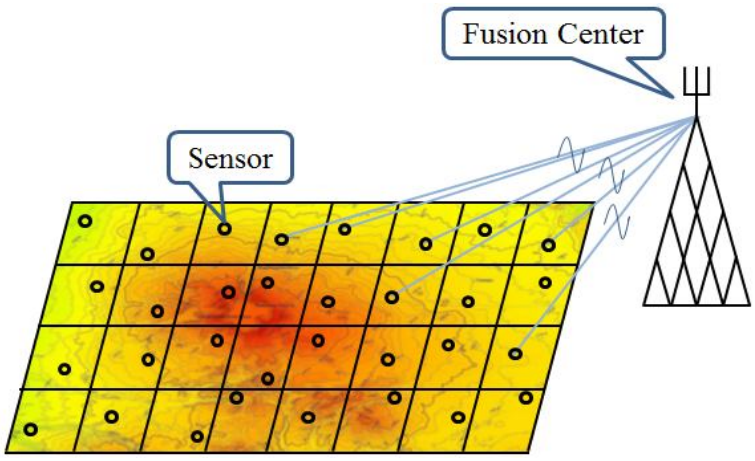

Figure 2. Wireless Sensor Network

After sensors observe an event taking place in a distributed region, they convert the sensed information into a digital signal and transmit the digitized signal to the FC. Finally, the FC assembles the data transmitted by all the sensors and decodes the original information. The decoded information at the FC provides a global picture of events occurring in the region of interest. Therefore, we assume that the objective of the sensor network is to determine accurately and rapidly reconstruct transmitted information and reconstruct the original signal.

We discuss the resource limitations of WSNs in the next section.

3.2. Resource limitations in WSNs

In this section, we describe the assumptions made in the sensor network we are interested in. We assume that the sensors are distributed and supposed to communicate with the FC through a wireless channel. Because each sensor is important components of WSN which observes event, they should typically be deployed in a large volume over the region of interest. Therefore, they are usually designed to be inexpensive and small. For that reason, each sensor operates on an onboard battery which is not rechargeable at all; thus, for simplicity, the hardware implementation of sensor nodes can provide only limited computational performance, bandwidth, and transmission power. As a result of limitations on the hardware implementation in sensor nodes, the FC has powerful computation performance and plentiful energy which naturally performs most of the complex computations.

Under the limited conditions stated above for a WSN, CS can substantially reduce the data volume to be transmitted at each sensor node. With the new method, it is possible to compress the original signal using only $O\left(k\log\left(n/k\right)\right)$ samples without going through many complex signal processing steps. These signals can be recovered successfully at the FC. All these are done under the CS framework. As the result, the consumption of power for transmission of signal contents at each sensor can be significantly reduced thanks to decreased data volume. Further, it should be noted that, this data reduction comes without utilizing onboard signal processing units since all the intermediate signal processing steps, shown **Figure 1**, are not needed. Namely, the sensor nodes can compress the signal while not spending any power for running complex compression algorithms onboard.

3.3. The usefulness of CS in WSNs

In this section, we provide a brief comparison of using CS and using the conventional compression in a WSN. This comparison illustrates why CS could be a useful solution for WSNs.

i) Sensor network scheme with conventional compression

For a conventional sensor system, suppose that the system designer has decided to gather all the uncompressed samples at a single location, say one of the sensors, in order to exploit inter-sensor correlation. See diagram shown in **Figure 3**. At the collection point, joint compression can be made and compressed information can be sent to the FC.

This option has a couple drawbacks. First, gathering the samples from all the sensors and jointly compressing them cause a transmission delay. Second, a lot of onboard power should be spent at the collaboration point. Third, each sensor should be collocated so that the transmitted information can be gathered at collaboration location.

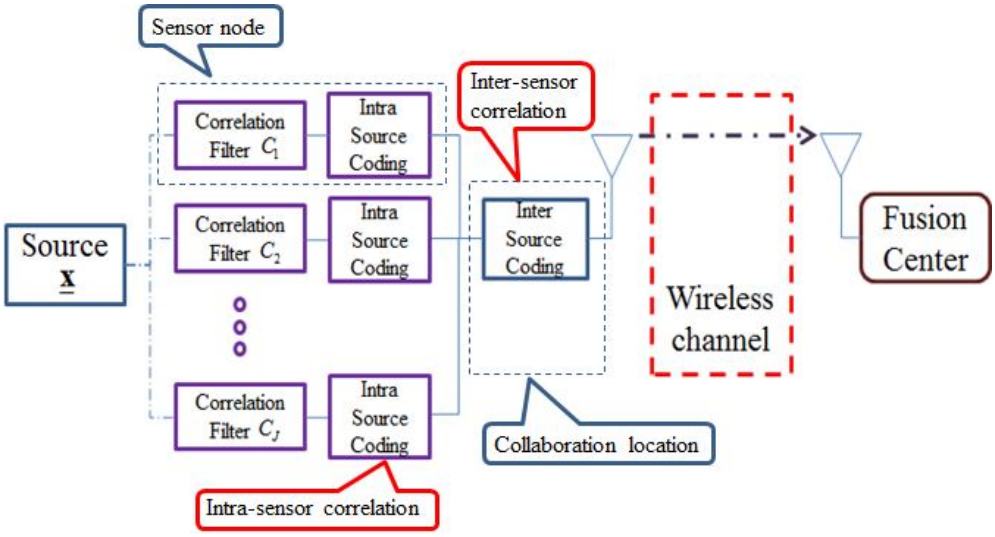

Figure 3. Conventional sensor network scheme

Now, we may suppose that the joint compression is not aimed at and each sensor compresses the signal on its own. First, the data reduction effect with this approach will be limited because inter-sensor correlation is not exploited at all. The total volume of the independently compressed data is much larger than that of jointly compressed data. This may produce a large traffic volume in the WSN and a large amount of transmission power will be wasted from the sensor nodes which transmit essentially the same information to the FC. Thus, this is an inefficient strategy as well.

ii) Sensor network scheme with compressive sensing

In contrast to the conventional schemes considered in the previous paragraph, the CS method aims to acquire compressed samples directly. If a high-dimensional observation vector $\mathbf{x}$ exhibits sparsity in a certain domain (by exploiting intra-sensor correlation), CS provides the *direct method* for signal compression as discussed in **Figure 1**. To compress the high-dimensional signal $\mathbf{x}$ into a low-dimensional signal $\mathbf{y}$, as Eq. (3), it uses a simple matrix multiplication with an $m \times n$ projection matrix $\mathbf{A}_j$, $j \in \{1,2,...J\}$, where j is the sensor index, as depicted in **Figure 4**.

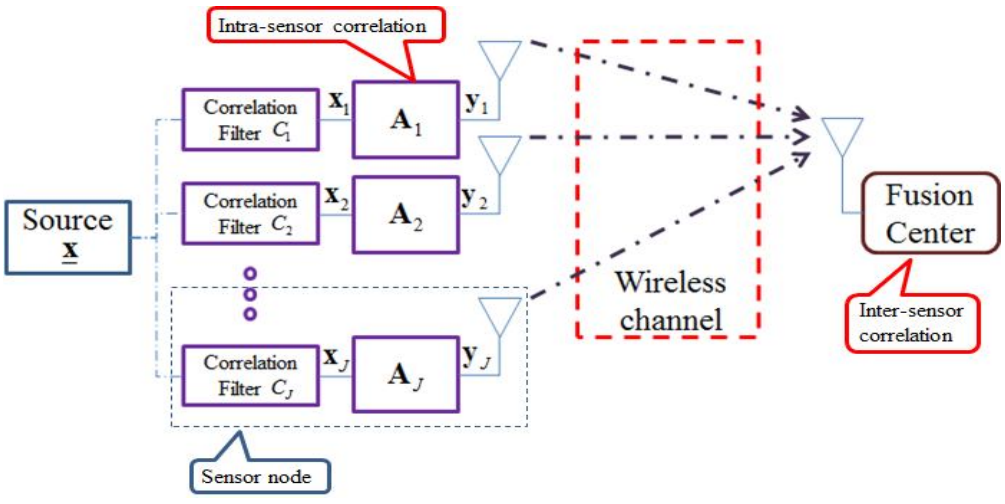

Figure 4. CS sensor network scheme

In the CS-based sensor network scheme, each sensor compresses the observed signals using a simple linear projection and transmits the compressed samples to the FC. Then, the FC can jointly reconstruct the received signals (by exploiting inter-sensor correlation) using one of the CS algorithms. Therefore, each sensor does not need to communicate with its neighboring sensors for joint compression. Our method is distributed compression without having the sensors to talk to each other; only the joint recovery at the FC is needed. Thus, no intermediate stages are required which are to gather all of the samples at a single location and carry out compression aiming to exploiting inter-sensor correlation. This free of intermediate stages allow us to reduce time delay significantly as well. Therefore, if the original data are compressed by CS, each sensor node produces much smaller traffic volume which can be transmitted to the FC at a much lower transmission power and with a smaller time delay.

4. Wireless sensor network system model

4.1. Multi-sensor systems and observed signal properties

Each sensor can observe only the local part of an entire physical phenomenon, and a certain event of interest is measured by one or more sensors. Therefore, the sensed signals are often partially correlated. These measured signals have two distinct correlations: intra-sensor correlation and inter-sensor correlation. Intra-sensor correlation exists in the signals observed by each sensor. Once a high-dimensional sensed signal has a sparse representation in a certain domain, we can reduce its size by using CS. This process exploits the intra-sensor correlation. In contrast, inter-sensor correlation exists between the signals sensed by different sensors. By exploiting inter-sensor correlation, further reduction in transmitted signals can be made.

These two correlations can be exploited to improve the system performance. As the number of sensors in a region becomes dense, each sensor has a strongly correlated signal that is similar to that of neighboring sensors. In contrast, if we decrease the density of sensors distributed in a given region, the sensed signals will obviously be more weakly correlated with each other. In this section, we discuss two strategies for transmitting signals in a multi-sensor CS-based system. One strategy uses only intra-sensor correlation, and the other uses both types of correlation. We illustrate that CS-based system in WSN exploits the inter-sensor correlation more effectively and simply than that of conventional sensor network.

i) Exploiting only intra-sensor correlation

In **Figure 5**, each sensor observes the source signal and independently compresses it to a low-dimensional signal. After compression, each sensor transmits the compressed signal to the FC. Without exploiting inter-sensor correlation between transmitted signals, the FC recovers these signals separately. In this case, even if there exists correlation among the sensed signals, because only intra-sensor correlation is exploited, we cannot gain any advantages from joint recovery. This method has the following characteristics:

1) Independent compression and transmission at each sensor

2) Signal recovery by exploiting only intra-sensor correlation at the FC

ii) Exploiting both intra- and inter-sensor correlation

Figure 6 shows the same process as in situation *i*) above, except that the FC exploits the inter-sensor correlation among sensed signals at signal reconstruction stage. In conventional sensor network system as shown in **Figure 3**, the sensor nodes communicate with their neighboring sensors to take advantage of joint compression by exploiting inter-sensor correlation. However, in the CS-based system, a stage for exploiting inter-sensor correlation is achieved at FC. It means that if inter-sensor correlation exists within the sensed signals, and the FC can exploit it. This is done with sensors communicating with the FC but not among the sensors themselves. We refer to this communication strategy as the *Distributed Compressive Sensing* (DCS). Exploitation of inter-sensor correlation should be manifested with the reduction of the measurement size m of matrix $\mathbf{A} \in \mathbf{R}^{m \times n}$, where $\mathbf{y} = \mathbf{A}\mathbf{x}$, required for good single recovery. The characteristics of our DCS sensor network are:

1) Independent compression and transmission at each sensor

2) Exploitation of inter-sensor signal correlation with the joint recovery scheme at the FC

3) Variation of the per sensor CS measurements to manipulate the level of signal correlation

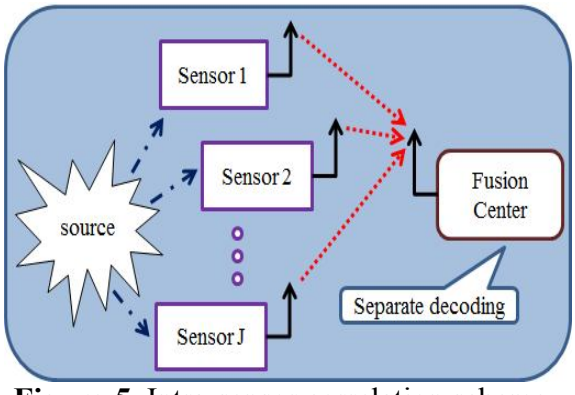

Figure 5. Intra-sensor correlation scheme

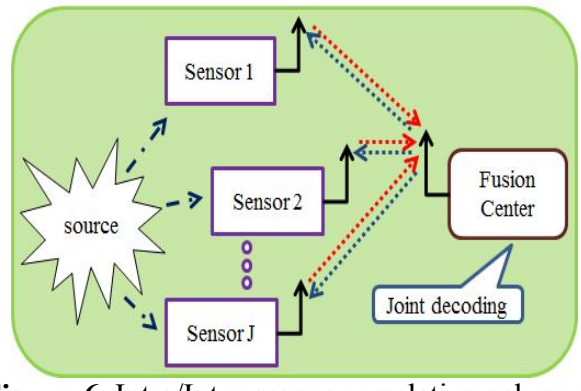

Figure 6. Intra/Inter-sensor correlation scheme

4.2. Correlated signal models and system equations

In this section, we introduce how signals with different degrees of correlation can be generated with sparse signal modes. Sparse signal is a correlated signal. The degree of sparseness, called the sparsity, is proportional to the amount of correlation. More correlated signal means sparser. In addition, inter-sensor signal correlation can be modeled *i*) by the degree of overlaps in the support sets of any two sparse signals, and *ii*) by the correlation of non-zero signal values. There are a number of papers which use interesting signal correlation models [5],[9],[10],[11]. The following table (**Table 1**) is useful for identifying the two models we take from these papers and use in the subsequent sections.

Correlation degree	Correlation characteristics					Model name
	Support set		Element value			
	Common	Innovation	Same	Different	Both	
Weak ↕ Strong	O	O	X	O	X	(Empty)
	O	O	X	X	O	**JSM-1**
	O	X	X	O	X	**JSM-2**
	O	X	O	X	X	(Empty)

Table 1. Synthetic signal models according to correlation degree.

Table 1 lists the signal models introduced in [9],[10]. In those references, the correlation signal is referred to as **JSM-1** (joint signal model) or **JSM-2** depending on the correlation type. In **JSM-1**, all of the signals share exactly the same common nonzero components that have the same values, whereas each signal also independently has different nonzero components, which is called innovation. Such a signal is expressed as

$$\mathbf{x}_j = \mathbf{z}_c + \mathbf{z}_j, \quad j \in \{1,2,\ldots,J\}, \ j \text{ is the index of the sensors,} \tag{12}$$

where $\|\mathbf{z}_c\|_0 = K$, and $\|\mathbf{z}_j\|_0 = K_j$. Obviously, $\mathbf{z}_c$ appears in all the signals. It can be recognized as the inter-sensor correlation. We note that the intra-sensor correlation is that all of the signals are sparse. The jth sensor transmits $\mathbf{y}_j = \mathbf{A}_j \mathbf{x}_j$ to the FC. After all the sensed signals are transmitted to the FC, the FC aims to recover all the signals. Because inter- sensor correlation exists in the sensed signals, we can obtain several benefits by using the correlated information in the transmitted signals. For ease of explanation, suppose that the WSN contains J sensors, and its sensed signal follows **JSM-1**. Then, the FC can exploit both intra- and inter-sensor correlation by solving Eq. (13) as described below.

i) Joint recovery scheme for **JSM-1**

The sensed signals from J sensors can be expressed as follows.

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{z}_c + \mathbf{z}_1 \in \mathbf{R}^n \\ \mathbf{x}_2 &= \mathbf{z}_c + \mathbf{z}_2 \in \mathbf{R}^n \\ &\vdots \\ \mathbf{x}_J &= \mathbf{z}_c + \mathbf{z}_J \in \mathbf{R}^n , \end{aligned}$$

where the sparsities of vectors $\mathbf{z}_c$ and $\mathbf{z}_j$ are K and K_j, respectively.

The transmitted signal $\mathbf{y}_j$ can be divided into two parts as follows.

$$\mathbf{y}_j = \mathbf{A}_j(\mathbf{z}_c + \mathbf{z}_j) = \mathbf{A}_j \mathbf{z}_c + \mathbf{A}_j \mathbf{z}_j$$

If the FC received all the signals transmitted from J sensors, it then concatenates the used sensing matrix and received signal using Eq. (13). Because the common sparsity $\mathbf{z}_c$

appears only once in the equation, the total sparsity is reduced from $J\times\left(K+K_j\right)$ to $K+\left(J\times K_j\right)$. In the underdetermined problem, low sparsity yields exact reconstruction. We will show the relationship between exact reconstruction and sparsity from simulation results in later section. By solving this equation, the FC can take advantage of exploiting inter-sensor correlation.

$$\begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\\ \mathbf{y}_3\\ \vdots\\ \mathbf{y}_J\end{bmatrix}=\begin{bmatrix}\mathbf{A}_1 & \mathbf{A}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbf{A}_2 & \mathbf{0} & \mathbf{A}_2 & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbf{A}_3 & \mathbf{0} & \mathbf{0} & \mathbf{A}_3 & \mathbf{0} & \vdots\\ \vdots & \vdots & \vdots & \vdots & \ddots & \mathbf{0}\\ \mathbf{A}_J & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_J\end{bmatrix}\begin{bmatrix}\mathbf{z}_c\\ \mathbf{z}_1\\ \mathbf{z}_2\\ \mathbf{z}_3\\ \vdots\\ \mathbf{z}_J\end{bmatrix} \tag{13}$$

However, if the FC recovers the received signals independently without using any correlation information, separate recovery is done. Even if the sensed signals are correlated, separate recovery offers no advantages for signal reconstruction because it does not exploit inter- sensor correlation.

ii) Separate recovery scheme for **JSM-1**

Even if a common correlated element exists in the sensed signals, separate recovery does not use that correlation information. Therefore, the received signals are recovered as follows.

$$\begin{bmatrix}\mathbf{y}_1\\ \mathbf{y}_2\\ \vdots\\ \mathbf{y}_J\end{bmatrix}=\begin{bmatrix}\mathbf{A}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{A}_2 & \mathbf{0} & \vdots\\ \vdots & \vdots & \ddots & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_J\end{bmatrix}\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\\ \vdots\\ \mathbf{x}_J\end{bmatrix} \tag{14}$$

To solve Eq. (13) and (14), we use the primal-dual interior point method (PDIP), which is an L_1 minimization algorithm, and compare the results of the two types of recovery. Using the comparison results in a later section, we can confirm that the measurement size required for perfect reconstruction is smaller for joint recovery than for separate recovery.

Now, we introduce **JSM-2**, which is simpler than **JSM-1**. All the signal coefficients are different, but their indices for nonzero components are the same. Suppose that there exist two signals, $\mathbf{x}_1$ and $\mathbf{x}_2$. The ith coefficient for $\mathbf{x}_1$ is nonzero if and only if the ith coefficient for $\mathbf{x}_2$ is nonzero. This property represents inter-sensor correlation, because if we know the support set for $\mathbf{x}_1$, then we automatically know the support set for $\mathbf{x}_2$.

iii) Recovery scheme for **JSM-2**

The inter-correlation prior becomes relevant when the number of sensors is more than two. To reconstruct the transmitted signals about **JSM-2**, we can solve the equation below jointly.

$$\mathbf{y}_j = \mathbf{A}_j \mathbf{x}_j,\ j \in \{1, 2, ..., J\} \tag{15}$$

Like the FC in **JSM-1**, the FC in **JSM-2** can exploit the fact that the support set is shared. By solving the Eq. (15) jointly in **JSM-2**, we obtain several benefits when the FC exploits inter-sensor correlation. If we solve this equation separately, but not jointly, it is separate recovery. As an algorithm for solving the equation of the **JSM-2** signal, we use a simultaneous OMP modified from an OMP algorithm in order to demonstrate the benefits when the FC exploits inter- sensor correlation. These algorithms are discussed in **Section 5**.

5. Recovery algorithms

In this section, we discuss the recovery algorithms used to solve the underdetermined equation. The recovery algorithms used in CS can be classified as the greedy type and the gradient type. We introduce representative algorithms from these two types, the orthogonal matching pursuit (OMP) and the primal-dual interior point method (PDIP) respectively.

5.1. Orthogonal matching pursuit (OMP) (greedy-type algorithm)

The orthogonal matching pursuit (OMP) is a famous greedy-type algorithm [12]. OMP produces a solution within k steps because it adds one index to the sparse set Λ at each iteration. The strategy of OMP is outlined in **Tables 2** and **3**.

Input	Output
A $m \times n$ measurement matrix $\mathbf{A}$ A $m-$dimensional data vector $\mathbf{y}$ The sparsity level k of the ideal signal	An estimate $\hat{\mathbf{x}}$ in R^n for the ideal signal. A set Λ_k containing k elements from $\{1,...,n\}$ An $m-$dimensional approximation $\hat{\mathbf{y}}_k$ of the data $\mathbf{y}$ An $m-$dimensional residual $\mathbf{r}_k = \mathbf{y} - \hat{\mathbf{y}}_k$

Table 2. Inputs and outputs of OMP algorithm.

The OMP algorithm:

1. Initialize:

Let the residual vector be $\mathbf{r}_0 = \mathbf{y}$, the sparse set $\Lambda_0 = \{\}$, and iteration number $t = 1$.

2. Find the index λ_t: $\lambda_t = \arg\max_{i=1,...,n} \left| \left\langle \mathbf{r}_{t-1}, \mathbf{a}_i \right\rangle \right|$. The $\mathbf{a}_i$ is the i th column vector of matrix $\mathbf{A}$.

3. Update set: $\Lambda_t = \Lambda_{t-1} \cup \{\lambda_t\}$.

4. Signal estimate: $\mathbf{x}_t(\Lambda_t) = \mathbf{A}_{\Lambda_t}^{\dagger}\mathbf{y}$ and $\mathbf{x}_t\left(\Lambda_t^C\right) = \mathbf{0}$, where $\mathbf{x}_t(\Lambda_t)$ is the set of elements whose indices are corresponding to the sparse set.

5. Get new residual: $\hat{\mathbf{y}}_t = \mathbf{A}_t \mathbf{x}_t$, $\mathbf{r}_t = \mathbf{y} - \hat{\mathbf{y}}_t$.

6. Increment t**:** Increase iteration number $t = t + 1$, and return to Step 2 if $t < k$.

Table 3. OMP algorithm.

Let us examine the above OMP algorithm. In step 2, OMP selects one index that has a dominant impact on the residual vector $\mathbf{r}$. Then, in step 3, the selected index is added to the sparse set, and the sub matrix $\mathbf{A}_{\Lambda_t}$ is constructed by collecting the column vectors of $\mathbf{A}$ corresponding to the indices of the sparse set Λ_t. OMP estimates the signal components corresponding to the indices of the sparse set and updates the residual vector by removing the estimated signal components in steps 4 and 5, respectively. Finally, OMP finishes its procedures when the cardinality of the sparse set is k.

OMP is a greedy-type algorithm because it selects the one index regarded as the optimal decision at each iteration. Thus, its performance is dominated by its ability to find the sparse set exactly. If the sparse set is not correctly reconstructed, OMP's solution could be wrong. Because OMP is very easy to understand, a couple of modified algorithms based on OMP have been designed and developed. For further information on the OMP algorithm and its modifications, interested readers are referred to two papers [13],[14].

We introduce another greedy-type algorithm based on OMP as an example: simultaneous orthogonal matching pursuit (SOMP) [13]. This greedy algorithm has been proposed for treating multiple measurement vectors for **JSM-2** when the sparse locations of all sensed signals are the same. Namely, SOMP algorithm handles with multiple measurements $\mathbf{y}_j$ as an input, when j is the index of distributed sensors, $j \in \{1, 2, ..., J\}$. In a later section, we use this algorithm to recover **JSM-2**. The pseudo code for SOMP is shown in **Table 4** and **5**.

Input	Output
A $m \times n$ measurement matrix $\mathbf{A}_j$ A $m-$dimensional data vector $\mathbf{y}_j$ The sparsity level k of the ideal signal	An estimate $\hat{\mathbf{x}}_j$ in R^n for the ideal signal. A set Λ_k containing k elements from $\{1,...,n\}$ An $m-$dimensional approximation $\hat{\mathbf{y}}_{j,k}$ of the data $\mathbf{y}_j$ An $m-$dimensional residual $\mathbf{r}_{j,k} = \mathbf{y}_j - \hat{\mathbf{y}}_{j,k}$

Table 4. Inputs and outputs of SOMP algorithm.

The SOMP algorithm:

1. Initialize:

Let the residual matrix be $\mathbf{r}_{j,0} = \mathbf{y}_{j,0}$. The sparse set $\Lambda_0 = \{\}$, and iteration number $t = 1$.

2. Find the index λ_t: $\lambda_t = \arg\max_{i=1,...,n} \sum_{j=1}^{J} \left| \left\langle \mathbf{r}_{j,t-1}, \mathbf{a}_{j,i} \right\rangle \right|$.

The $\mathbf{a}_{j,i}$ is the i th column vector of matrix $\mathbf{A}_j$.

3. Update set: $\Lambda_t = \Lambda_{t-1} \cup \{\lambda_t\}$.

4. Signal estimate: $\mathbf{x}_{j,t}(\Lambda_t) = \mathbf{A}_{j,\Lambda_t}^{\dagger} \mathbf{y}_j$ and $\mathbf{x}_{j,t}(\Lambda_t^C) = \mathbf{0}$, where $\mathbf{x}_{j,t}(\Lambda_t)$ is the set of elements whose indices are corresponding to the sparse set.

5. Get new residual: $\hat{\mathbf{y}}_{j,t} = \mathbf{A}_{j,t}\mathbf{x}_{j,t}$, $\mathbf{r}_{j,t} = \mathbf{y}_j - \hat{\mathbf{y}}_{j,t}$.

6. Increment t: Increase iteration number $t = t + 1$, and return to Step 2 if $t < k$.

Table 5. SOMP algorithm.

5.2. Primal-Dual interior point method (PDIP) (gradient-type algorithm)

The L_1 minimization in Eq. (11) can be recast as linear programming. Here we examine this relationship. Clearly, the L_1 minimization problem in Eq. (11) is not linear programming because its cost function is not linear. However, by using a new variable, we can transform it to linear programming. Thus, the problem that we want to solve is

$$\begin{aligned} &\min_{(x,u)} \sum_i u_i \\ &\text{subject to} \\ &\forall_i \left| x(i) \right| \le u_i \\ &\mathbf{Ax} = \mathbf{b} \end{aligned} \quad . \tag{16}$$

The solution of the above equation is equal to the solution of the L_1 minimization problem. Many approaches to solving Eq. (16) have been studied and developed. Here, we discuss the primal-dual interior point (PDIP) method, which is an example of gradient-type algorithms. First, we have the Lagrangian function of Eq. (16), as follows:

$$L(\mathbf{t}, \boldsymbol{\lambda}, \mathbf{v}) = \begin{bmatrix} \mathbf{0}_1^{\mathrm{T}} & \mathbf{1}^{\mathrm{T}} \end{bmatrix} \mathbf{t} + \mathbf{v}^{\mathrm{T}} \left(\begin{bmatrix} \mathbf{A} & \mathbf{0}_2 \end{bmatrix} \mathbf{t} - \mathbf{b} \right) + \boldsymbol{\lambda}^{\mathrm{T}} \left(\begin{bmatrix} \mathbf{e} & -\mathbf{e} \\ -\mathbf{e} & -\mathbf{e} \end{bmatrix} \mathbf{t} \right), \tag{17}$$

where $\mathbf{e}$ is the $n \times n$ identity matrix, $\mathbf{0}_1$ is the zero vector, $\mathbf{0}_2$ is the $m \times n$ zero vector, and $\mathbf{1}$ is the $n \times 1$ vector whose elements are all one, $\mathbf{t} := \begin{bmatrix} \mathbf{x} \\ \mathbf{u} \end{bmatrix} \in \mathbf{R}^{2n \times 1}$, $\mathbf{v} \in \mathbf{R}^{m \times 1}$, and $\boldsymbol{\lambda} \in \mathbf{R}^{2n \times 1} \ge 0$.

From the Lagrangian function, we have several KKT conditions,

$$\begin{bmatrix}\mathbf{0}\\ \mathbf{1}\end{bmatrix}+\begin{bmatrix}\mathbf{A}^{\mathrm{T}}\\ \mathbf{0}^{\mathrm{T}}\end{bmatrix}\mathbf{v}^{*}+\begin{bmatrix}\mathbf{e} & -\mathbf{e}\\ -\mathbf{e} & -\mathbf{e}\end{bmatrix}\boldsymbol{\lambda}^{*}=\mathbf{0}_3$$
$$\begin{bmatrix}\mathbf{A} & \mathbf{0}_2\end{bmatrix}\mathbf{t}^{*}-\mathbf{b}=\mathbf{0}_4$$
$$\begin{bmatrix}\mathbf{e} & -\mathbf{e}\\ -\mathbf{e} & -\mathbf{e}\end{bmatrix}\mathbf{t}^{*}\le\mathbf{0}_1 \quad , \tag{18}$$
$$\left(\boldsymbol{\lambda}^{*}\right)^{\mathrm{T}}\begin{bmatrix}\mathbf{e} & -\mathbf{e}\\ -\mathbf{e} & -\mathbf{e}\end{bmatrix}\mathbf{t}^{*}=0,\boldsymbol{\lambda}^{*}\ge\mathbf{0}_3$$

where $\mathbf{0}_3$ is the $2n\times 1$ zero vector, and $\mathbf{0}_4$ is the $m\times 1$ zero vector. The main point of the PDIP is to seek the point $\left(\mathbf{t}^{*},\boldsymbol{\lambda}^{*},\mathbf{v}^{*}\right)$ that satisfies the above KKT conditions. This is achieved by defining a mapping function $\mathrm{F}\left(\mathbf{t},\boldsymbol{\lambda},\mathbf{v}\right):\mathbf{R}^{(2n+m)\times 1}\to\mathbf{R}^{(2n+m)\times 1}$, which is

$$\mathrm{F}\left(\mathbf{t},\boldsymbol{\lambda},\mathbf{v}\right)=\begin{bmatrix}\begin{bmatrix}\mathbf{0}\\ \mathbf{1}\end{bmatrix}+\begin{bmatrix}\mathbf{A}^{\mathrm{T}}\\ \mathbf{0}^{\mathrm{T}}\end{bmatrix}\mathbf{v}+\begin{bmatrix}\mathbf{e} & -\mathbf{e}\\ -\mathbf{e} & -\mathbf{e}\end{bmatrix}\boldsymbol{\lambda}\\ \left(\boldsymbol{\lambda}^{*}\right)^{\mathrm{T}}\begin{bmatrix}\mathbf{e} & -\mathbf{e}\\ -\mathbf{e} & -\mathbf{e}\end{bmatrix}\mathbf{t}^{*}\\ \begin{bmatrix}\mathbf{A} & \mathbf{0}_2\end{bmatrix}\mathbf{t}-\mathbf{b}\end{bmatrix}=\mathbf{0}_4\in\mathbf{R}^{(2n+m)\times 1},\begin{bmatrix}\mathbf{e} & -\mathbf{e}\\ -\mathbf{e} & -\mathbf{e}\end{bmatrix}\mathbf{t}^{*}\le\mathbf{0}_1,\boldsymbol{\lambda}^{*}\ge\mathbf{0}_3, \tag{19}$$

where $\mathbf{0}_4$ is the $(2n+1)\times 1$ zero vector. Now, we would like to find the point $\left(\mathbf{t}^{*},\boldsymbol{\lambda}^{*},\mathbf{v}^{*}\right)$ satisfying $\mathrm{F}\left(\mathbf{t}^{*},\boldsymbol{\lambda}^{*},\mathbf{v}^{*}\right)=\mathbf{0}_4$. Here, we use a linear approximation method. From the Taylor expansions of the function $\mathrm{F}\left(\mathbf{t},\boldsymbol{\lambda},\mathbf{v}\right)$, we have

$$\mathrm{F}\left(\mathbf{t}+\Delta\mathbf{t},\boldsymbol{\lambda}+\Delta\boldsymbol{\lambda},\mathbf{v}+\Delta\mathbf{v}\right)\approx\mathrm{F}\left(\mathbf{t},\boldsymbol{\lambda},\mathbf{v}\right)+\nabla_{(\mathbf{t},\boldsymbol{\lambda},\mathbf{v})}\mathrm{F}\left(\mathbf{t},\boldsymbol{\lambda},\mathbf{v}\right)\begin{bmatrix}\Delta\mathbf{t}\\ \Delta\mathbf{v}\\ \Delta\boldsymbol{\lambda}\end{bmatrix}. \tag{20}$$

Thus, solving the above equations yields the direction $(\Delta\mathbf{t}, \Delta\mathbf{v}, \Delta\boldsymbol{\lambda})$. Next, we seek the proper step length along the direction that does not violate $\begin{bmatrix} \mathbf{e} & -\mathbf{e} \\ -\mathbf{e} & -\mathbf{e} \end{bmatrix} \mathbf{t}^* \le \mathbf{0}_1$ and $\boldsymbol{\lambda}^* \ge \mathbf{0}_3$. The pseudo code for the PDIP algorithm is shown in **Table 6**.

The primal-dual interior point method algorithm:

1. Initialize:

Choose $\mathbf{v}^0 \in \mathbf{R}^{m\times 1}$, $\boldsymbol{\lambda}^0 \ge \mathbf{0}_3$, and $\mathbf{t}^0 = \begin{bmatrix} \mathbf{x}^0 & \mathbf{u}^0 \end{bmatrix}^{\mathrm{T}}$, where $\mathbf{x} = \mathbf{A}^{\dagger}\mathbf{b}$, and $\mathbf{u}^0 = |\mathbf{x}^0| + \alpha |\mathbf{x}^0|$ and iteration number $k = 1$. (The $\mathbf{A}^{\dagger} = \left(\mathbf{A}^T\mathbf{A}\right)^{-1}\mathbf{A}^T$ is the Moore-Penrose pseudo-inverse of $\mathbf{A}$ and $\mathbf{A}^T$ denotes the transpose of $\mathbf{A}$.)

2. Find the direction vectors $(\Delta\mathbf{t}, \Delta\mathbf{v}, \Delta\boldsymbol{\lambda})$:

$$\begin{bmatrix} \Delta\mathbf{t} \\ \Delta\mathbf{v} \\ \Delta\boldsymbol{\lambda} \end{bmatrix} = -\left[\nabla_{(\mathbf{t}^k, \boldsymbol{\lambda}^k, \mathbf{v}^k)} \mathrm{F}\left(\mathbf{t}^k, \boldsymbol{\lambda}^k, \mathbf{v}^k\right)\right]^{-1} \mathrm{F}\left(\mathbf{t}^k, \boldsymbol{\lambda}^k, \mathbf{v}^k\right).$$

3. Find the proper step length:

Choose the largest α satisfying $\left\|\mathrm{F}\left(\mathbf{t}^k + \alpha, \boldsymbol{\lambda}^k + \alpha, \mathbf{v}^k + \alpha\right)\right\|_2^2 \le \left\|\mathrm{F}\left(\mathbf{t}^k, \boldsymbol{\lambda}^k, \mathbf{v}^k\right)\right\|_2^2$.

4. Update parameters:

$\mathbf{t}^{k+1} = \mathbf{t}^k + \alpha\Delta\mathbf{t}$, $\mathbf{v}^{k+1} = \mathbf{v}^k + \alpha\Delta\mathbf{v}$, $\boldsymbol{\lambda}^{k+1} = \boldsymbol{\lambda}^k + \alpha\Delta\boldsymbol{\lambda}$.

5. Update the signal:

$\mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{t}[1:n]$.

6. Increment the iteration number k:

Increase iteration number $k = k + 1$, and return to Step 2 if $\left\|\mathbf{y} - \mathbf{A}\mathbf{x}^k\right\|_2^2 > eps$.

Table 6. Primal-dual interior point method algorithm.

6. Performance evaluation

In this section, we investigate the performance of a WSN system that applies CS by using the PDIP or the SOMP. We divide this section into four sections that analyze the relationship between the number of measurements M, sparsity k depending on the number of sensors J, the degree of correlation, and the signal-to-noise ratio (SNR), respectively. To avoid confusion regarding the graphs, we define the notations and metrics used in the experiments in **Tables 7** and **8**, respectively.

Notation:
N : The length of the signal $\mathbf{x}$ at each sensor, M : the length of measurement $\mathbf{y}$.
j : Index of the sensors, $j \in \{1,2,...,J\}$.
$\mathbf{y}$: Signal transmitted from each sensor.
$\mathbf{A}$: Sensing matrix, $\mathbf{R}^{M \times N}$, which has a Gaussian distribution
$\mathbf{x}$: Sparse signal on the sensor; its elements also have a Gaussian distribution.
$\mathbf{n}$: Additive white Gaussian noise (AWGN).
K : Common sparsity number.
K_j : Innovation sparsity number.

Table 7. Notation used in experiments.

Metric:
i) SNR (dB) : Signal-to-noise ratio, $$10\log_{10}\frac{\|\mathbf{Ax}\|_2^2}{M\sigma_{\mathbf{n}}^2},$$ where $\sigma_{\mathbf{n}}^2$: The variance of noise.

ii) MSE : Mean square error,

$$\frac{\left\|\hat{\mathbf{x}}-\mathbf{x}\right\|_2^2}{\left\|\mathbf{x}\right\|_2^2},$$

where $\hat{\mathbf{x}}$: Recovered signal , and $\mathbf{x}$: Original signal .

Table 8. Metrics used in experiments.

The proposed correlation signals, **JSM-1** and **JSM-2**, as described in **Table 1**, will also be investigated in terms of various parameters, such as signal length, matrix size, and sparsity number. To recover the **JSM-1** signal (which includes both common and innovation components, and the common component has the same values for every sensor) from received signal $\mathbf{y}$, we use the PDIP algorithm. However, to recover the **JSM-2** signal (which includes only a common component that has different values for every sensor), we use SOMP. It is inappropriate to apply SOMP to the **JSM-1** signals because there exists the innovation component at every sensed signal $\mathbf{x}_j$. Although SOMP can identify the common part exactly, confusion may arise regarding the optimal selection for the innovation component. Because SOMP selects only one index that has the optimal value among the vector elements of length N in every iteration, if the selected index is included in the innovation component of only one sensor node, the solution cannot be correct.

For this reason, we use the SOMP algorithm to recover only the **JSM-2** signal. If we use SOMP to recover the **JSM-1** signal, we should improve the algorithm for finding the innovation component. From the results of simulations using those two recovery algorithms, we determined the relationship among the sensors, measurement, and amount of correlation in the unknown sensor signals.

6.1. Reconstruction performance as a function of sparsity

Figure 7 shows the results when the PDIP algorithm was used to reconstruct signals for **JSM-1**. We increased the common sparsity K of each sensor and number of sensors J while fixing the signal length, the number of measurements, and the innovation sparsity of each sensor at $N = 50$, $M = 20$, and $K_j = 3$, respectively. The FC concatenates the received signals $\mathbf{y}_j$, $j = \{1, 2, ..., J\}$ to $[\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_J]^T$,and puts the sensing matrices to the integrated one as $\mathbf{A}_{PDIP}$ of Eq. (21). Thus, the equation is $\mathbf{y}_{PDIP} \coloneqq \mathbf{A}_{PDIP}\mathbf{z}_{PDIP}$, and the number of measurements in this equation is $M_{PDIP} = M \times J$; it then uses PDIP algorithm to get $\mathbf{z}_{PDIP}$ from $\mathbf{y}_{PDIP}$. The recovered signal $[\hat{\mathbf{z}}_c, \hat{\mathbf{z}}_1, \hat{\mathbf{z}}_2, ..., \hat{\mathbf{z}}_J]^T$ from $[\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_J]^T$ was compared with the original $[\mathbf{z}_c, \mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_J]^T$ in order to calculate the probability of exact reconstruction.

$$\mathbf{y}_{PDIP} \coloneqq \begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_J \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_2 & \mathbf{0} & \mathbf{A}_2 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{A}_J & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_J \end{bmatrix}}_{\mathbf{A}_{PDIP}} \underbrace{\begin{bmatrix} \mathbf{z}_c \\ \mathbf{z}_1 \\ \mathbf{z}_2 \\ \vdots \\ \mathbf{z}_J \end{bmatrix}}_{\mathbf{z}_{PDIP} \coloneqq} \tag{21}$$

In this case, even if the original signal $\mathbf{x}_j = \mathbf{z}_c + \mathbf{z}_j \in \mathbf{R}^n$, $j \in \{1, 2, ..., J\}$ is not sparse, the signals $\mathbf{y}_j$ transmitted from sensors can be recovered perfectly at the FC if all the sensors have a small number of innovation component K_j that corresponds to $\mathbf{z}_j$. However, as the number of sensors increases, the integrated matrix also becomes large. Consequently, the computation is complex, and much time is required to obtain the solution.

Figure 8 illustrates the use of the SOMP algorithm to recover **JSM-2**. The fixed parameters are the signal length N and measurement size M of each sensor. To determine the

effect of the number of sensors and sparsity in the WSN, we increased the sparsity K and number of sensors J. Because the **JSM-2** signal has the same sparse location for every sensor, the sparse location can be found by using SOMP easily. As the number of sensors increases, the probability of making the optimal decision at each iteration is greater. As a result, exact reconstruction is achieved, as shown in **Figure 8**.

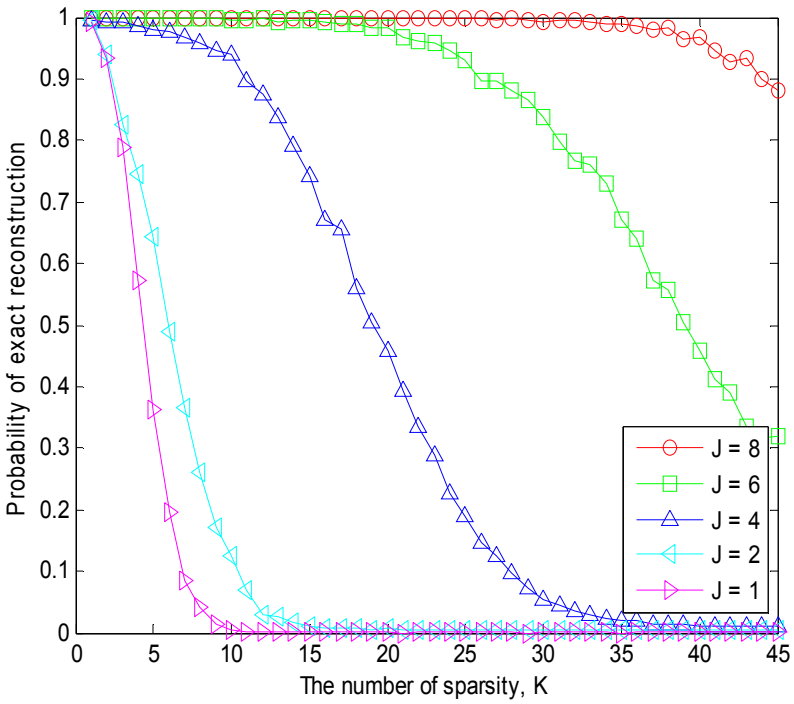

Figure 7. Signal reconstructed using PDIP algorithm for **JSM-1**. System parameters are $N = 50$, $M = 20$, and innovation sparsity $K_j = 3$.

In both cases, we notice that the probability of successful reconstruction increases as the number of sensors increases, because both algorithms use the prior information that the signals are correlated. For example, when we increase only the common sparsity K, we can reconstruct all of the signals by only increasing the number of sensors. Interestingly, the curve of **Figure 7** in **JSM-1** experiment does not show convergence as the number of sensors increases. On the other hand, that of **Figure 8** in **JSM-2** experiment converges to $M - 1$ as the number of sensors increases. These results are determined from the ratio of the number of measurement to sparsity (M / K) in compressive equation. In the case of **Figure 7**, as the

number of sensors increases, the number of measurement M_{PDIP} also increases. Thus, as the number of sensors increases, the ratio is also changed. (In our experiment, we choose $K_j = 3$, where $K_j \ll M$. Therefore, the ratio increases as the number of sensors increases.) In the case of **Figure 8**, there is no change for the ratio regardless of increasing the number of sensors. The varying ratio (M / K) of **JSM-1** experiment makes the result about no convergence in contrast with that of JSM-2 experiment.

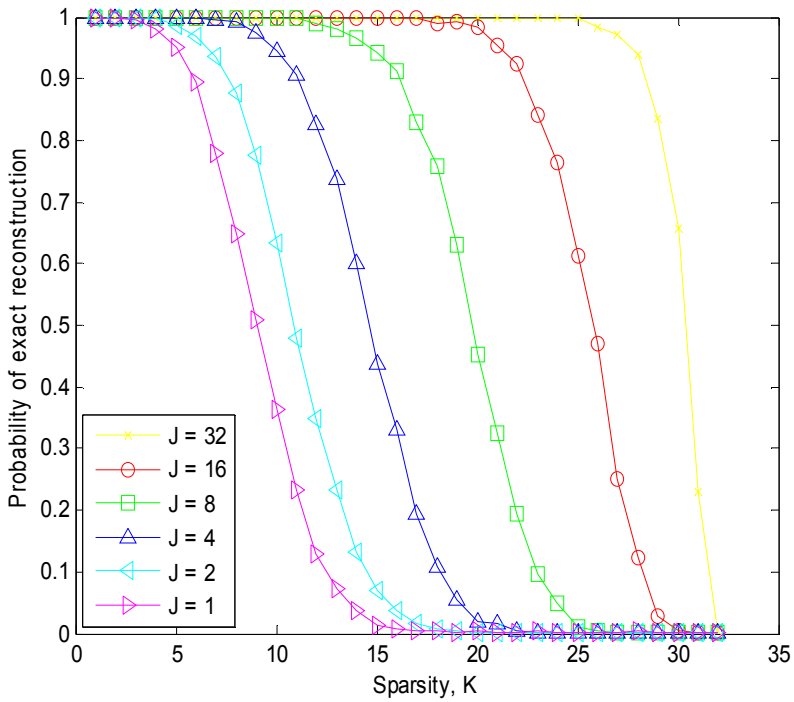

Figure 8. Signal reconstructed signal using SOMP for **JSM-2** for increasing common sparsity K and number of sensors J. System parameters are $N = 256$ and $M = 32$.

Summary 6.1 Reconstruction performance as a function of sparsity

We aim at investigating how the increase in sparsity K for signal at each sensor affects reconstruction performance of the joint recovery algorithms, while the signal length N and the number of measurements M are fixed at each sensor. As the common sparsity K of each sensor increases, the probability of exact reconstruction decreases. This is obvious. Eq. (21) is the result of **JSM-1** model which can be used to represent both common and innovative

elements in each sensor and allows exploitation of inter-sensor correlation. Thus, as the number of sensors increases, the total sparsity and the number of measurements M_{PDIP} also increases as shown in Eq. (21). In **JSM-2**, the sparsity K and the number of measurements M per sensor are fixed by the formulation in (15), regardless of the number of sensors. The varying ratio between the number of measurement and sparsity makes the results of **Figure 7** and **8**, respectively.

6.2. Relationship between the number of sensors and the number of measurements required for exact reconstruction

In **Figure 9**, we show the results when we increased the number of measurements and the number of sensors while fixing the signal length ($N = 50$), common component ($K = 9$), and innovation component ($K_j = 3$). As the number of sensors increased, the number of measurements required for the probability of exact reconstruction to converge to one decreased. Therefore, if we use many sensors to reconstruct the correlated signal, we can reduce the number of measurements, which in turn reduces the transmission power at each sensor. However, as **Figures 9** and **10** show, the decrease in measurement size is limited by the sparsity number $(K+1)$ in one sensor.

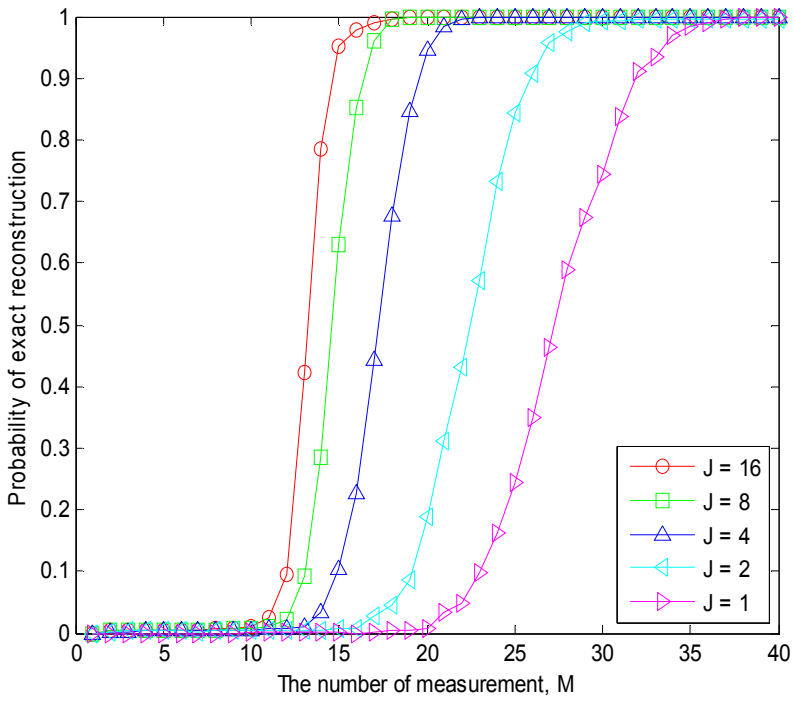

Figure 9. Signal reconstructed using PDIP algorithm for **JSM-1**. System parameters: $N = 50$, $K = 9$, $K_j = 3$.

For **JSM-2** signals, reconstruction is similar to that of **JSM-1** signals in terms of the effect of increasing the number of sensors when the correlated signal is jointly recovered (**Figure 10**, solid line). However, if signal reconstruction is performed separately, more

measurements per sensor are needed as the number of sensors J increases (**Figure 10**, dotted line). Because the transmitted signals from each sensor are reconstructed independently, if the probability p of successful reconstruction is less than or equal to 1, and then the total probability of successful reconstruction for all transmitted signals is p^J.

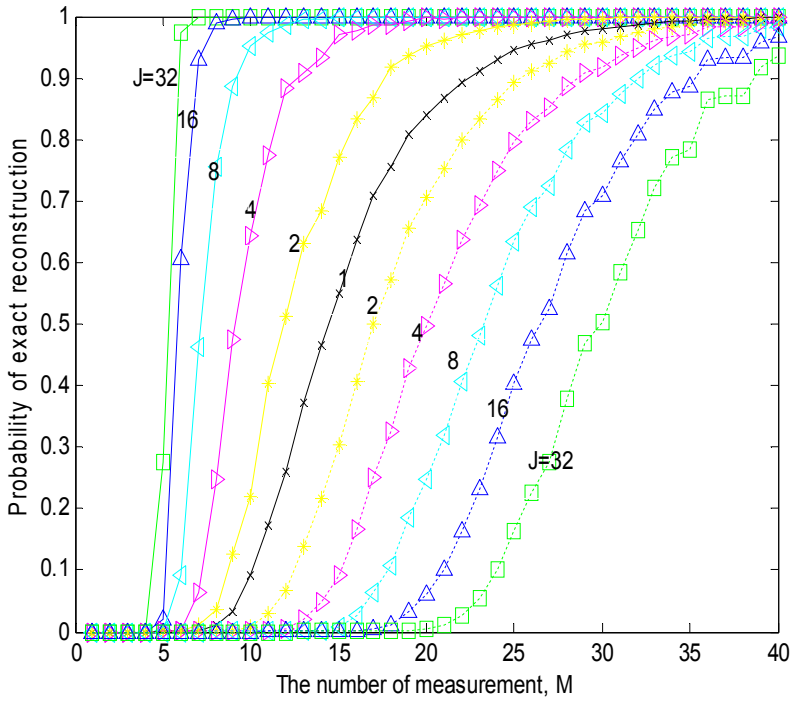

Figure 10. Reconstruction using SOMP for **JSM-2**. System parameters: $N = 50$, $K = 5$. Solid line: joint recovery, dotted line: separate recovery.

Summary 6.2 Relationship between the number of sensors and the number of measurements required for exact reconstruction

We aim at investigating how the probability of exact reconstruction changes with the number of sensors increased. As the number of sensors is increased, the signals FC collects are more inter-sensor correlated and the number of measurements per sensor required for exact reconstruction decrease. **Figure 9** and **Figure 10** show that the original signals can be recovered with high probability at the fixed measurement as $J \to \infty$ and the per-sensor measurements required for perfect signal recovery converges to $K+1$.

6.3. Performance as a function of SNR

In this section, we present the system performance of a WSN that uses CS in an additive white Gaussian noise (AWGN) channel. As in the other experiment, we used a Gaussian distribution to create the sensing matrix $\mathbf{A}_j$, $j \in \{1,2,...,J\}$ and sparse signal $\mathbf{x}_j$ and then added AWGN $\mathbf{n}$ to the measurement $\mathbf{y}_j = \mathbf{A}_j \mathbf{x}_j$. At the FC, the received signal $\tilde{\mathbf{y}}_j = \mathbf{y}_j + \mathbf{n}$ was recovered jointly. We increased the number of sensors while fixing the signal length, number of measurements, common sparsity, and innovation sparsity at $N = 50$, $M = 20$, $K = 3$, and $K_j = 2$, respectively. In this experiment, the SNR is set as shown below.

$$\text{SNR}\left(\text{Signal to Noise Ratio}\right) = 10\log_{10}\frac{\left\|\mathbf{A}_j\mathbf{x}_j\right\|^2}{M\sigma_{\mathbf{n}}^2}$$

The $\left\|\mathbf{A}_j\mathbf{x}_j\right\|^2$ is the transmitted signal power at sensor j, M is the number of measurements, and σ_n^2 is the noise variance. To estimate the reconstruction error between the original signal $\mathbf{x}_j$ and the reconstruction signal $\hat{\mathbf{x}}_j$, we used the mean square error (MSE) as follows.

$$\text{Mean square error} = \frac{\left\|\hat{\mathbf{x}}_j - \mathbf{x}_j\right\|^2}{\left\|\mathbf{x}_j\right\|^2}$$

We applied the PDIP algorithm to solve Eq. (21) for **JSM-1** and obtained the solution, $\left[\hat{\mathbf{z}}_c, \hat{\mathbf{z}}_1, \hat{\mathbf{z}}_2, ..., \hat{\mathbf{z}}_J\right]^T$. Because of the effect of noise, the solution $\left[\hat{\mathbf{z}}_c, \hat{\mathbf{z}}_1, \hat{\mathbf{z}}_2, ..., \hat{\mathbf{z}}_J\right]^T$ does not have a sparse solution. Therefore, we chose the largest $K + \left(J \times K_j\right)$ values from among the

elements of the solution. To compare the recovered signal $\hat{\mathbf{x}}_j$ to the original sensed signal $\mathbf{x}_j$, we divided the concatenated solution $\left[\hat{\mathbf{z}}_c, \hat{\mathbf{z}}_1, \hat{\mathbf{z}}_2, ..., \hat{\mathbf{z}}_J\right]^T$ by each recovered signal $\hat{\mathbf{x}}_j$. The results are shown in **Figure 11**.

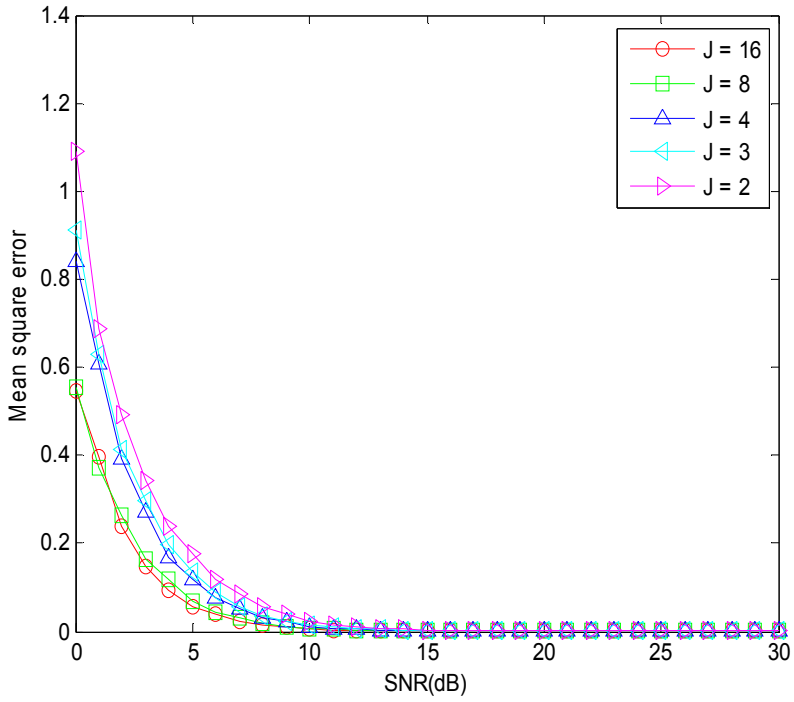

Figure 11. Signal reconstructed using primal-dual interior point method for **JSM-1**. System parameters: $N = 50$, $M = 20$, $K = 3$, and $K_j = 2$.

To obtain the results in **Figure 12**, we used the SOMP algorithm for **JSM-2** with the same processing. In contrast to the PDIP algorithm, the SOMP algorithm first searches the support set; therefore, it does not require a step in which the largest K values are chosen from among the elements of the solution. However, if the selected support set is wrong, the reconstruction is also wrong. Both results, **Figures 11** and **12**, show that if we increase the number of sensors, the MSE is improved and finally converges to zero as the SNR increases. Even if the transmitted signals contain much noise, having a large number of sensors to observe the correlated signal in the sensed region facilitates the search for the exact solution. In **Figure 12**, when the number of sensors is two or three, the MSE does not converge to zero even if the

SNR is high. Because the SOMP algorithm uses cross correlation to find the support set (**step 2** of **Table 5**), if the rank of sensing matrix **A** is smaller than the number of columns in **A** , then each column will exhibit significant correlation among themselves. Consequently, the SOMP algorithm selects the wrong support location. However, this problem can be solved by using a large number of sensors.

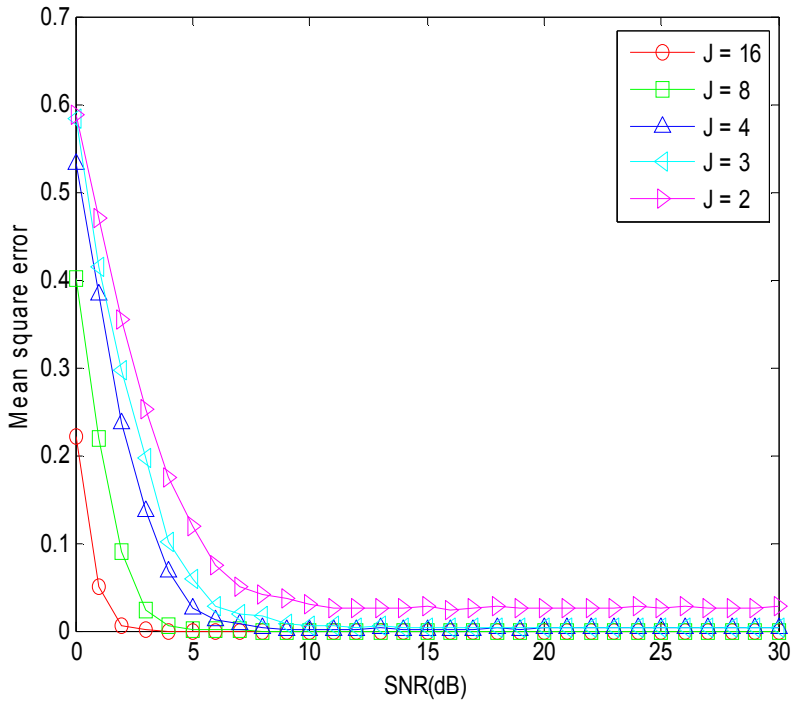

Figure 12. Signal reconstructed using SOMP for **JSM-2**. System parameters: $N = 50$, $M = 20$, $K = 5$.

Summary 6.3 Performance as a function of SNR

We aim to investigate the effect of noise in CS based WSN. In particular, we experiment how mean square error decreases as SNR increases. **Figure 11** and **12** show similar results. As the number of sensors increases, signals are more correlated. This helps signal recovery.

6.4. Joint vs. separate recovery performance as a function of correlation degree

Now, we compare the results of joint recovery and separate recovery (**Figure 13**). In joint recovery, if a correlation exists between the signals observed from the distributed sensors, the FC can use the correlated information to recover the transmitted signals. In separate recovery, correlated information is not used regardless of whether a correlation pattern exists between the observed signals. In **Figure 13**, solid lines were obtained from joint reconstructions, whereas dotted lines are the results of separate reconstructions.

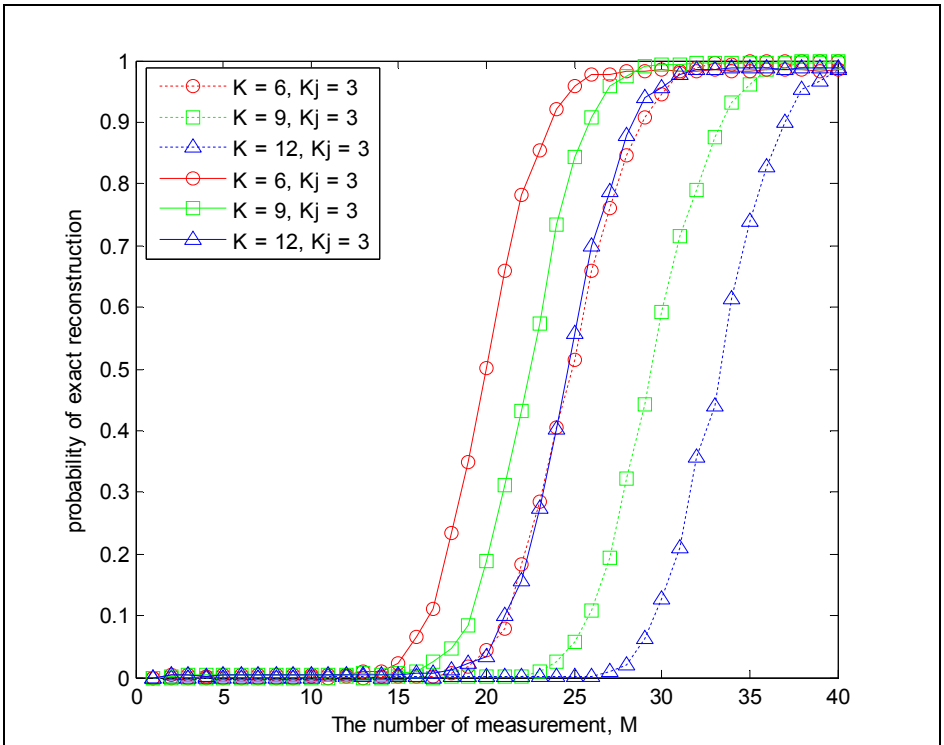

Figure 13. Joint (solid line) and separate (dotted line) reconstruction using PDIP algorithm for **JSM-1**. System parameters: $N = 50$, $J = 2$. The benefits of joint reconstruction depend on the sparsity number K.

When we use separate reconstruction, we cannot obtain any benefits from correlated information. However, when we use joint reconstruction, we can reduce the measurement size. For example, in **Figure 14**, the required number of measurements is almost 40 (dashed line and circles, $K = 6$) for perfect reconstruction when we use separate reconstruction. On the other hand, when we use joint reconstruction, it decreases to around 30 (solid line and circles, $K = 6$). Furthermore, as the common sparsity increases, the performance gap increases. For

example, when the common sparsity is 9, joint reconstruction has a 90% probability of recovering all the signals at $M = 30$. However, the probability that separate reconstruction can recover all the signals is only 70%. **Figure 13** also shows that joint reconstruction is superior to separate reconstruction. For example, we need at least 30 measurements for reliable recovery using separate reconstruction. However, we merely need at least 25 measurements for reliable recovery using joint reconstruction.

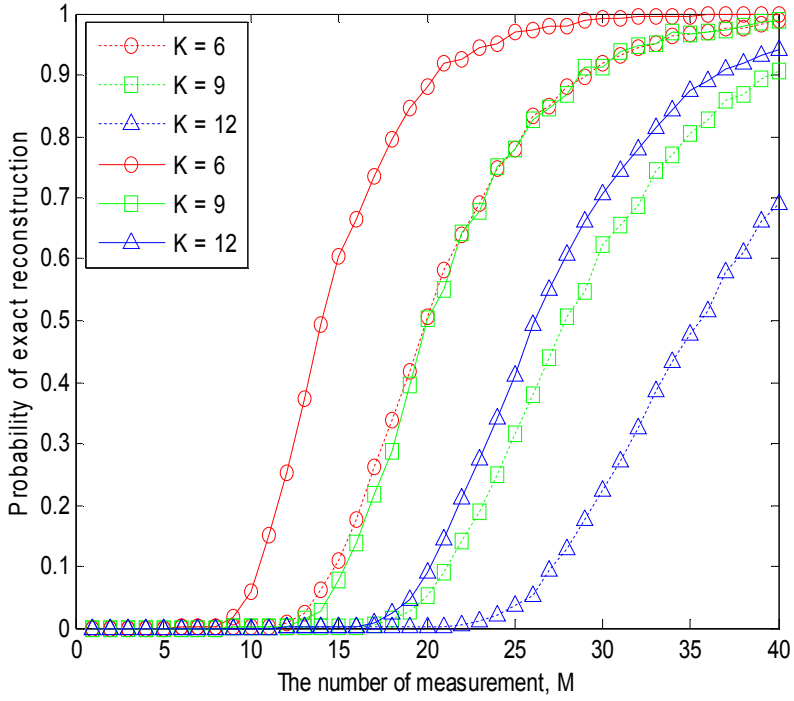

Figure 14. Joint (solid line) and separate (dotted line) reconstruction using SOMP for **JSM-2**. System parameters: $N = 50$, $J = 2$. Joint reconstruction has a higher probability of success than separate reconstruction.

Summary 6.4 Joint vs. separate recovery performance as a function of correlation degree

If a correlation exists between the signals observed from the distributed sensors, and if the FC uses the joint recovery, then it can reduce the measurement size required for exact reconstruction in comparison that with the separate recovery. As the degree of correlation increases, the gap in the results of two methods (Joint recovery and Separate recovery) widens as shown **Figure 13** and **14**.

7. Summary of Chapter

In this chapter, we discussed the application of compressive sensing (CS) for wireless sensor networks (WSNs). We assumed a WSN consisting of spatially distributed sensors and one fusion center (FC). The sensor nodes take signal samples and pass their acquired signal samples to the FC. When the FC receives the transmitted data from the sensor nodes, it aims to recover the original signal waveforms, for later identification of the events possibly occurring in the sensed region. (**Section 3.1**)

We discussed that CS is the possible solution which provides simpler signal acquisition and compression. CS is suitable for the wireless sensor networks since it allows removal of intermediate stages such as sampling the signal and gathering the sampled signals at one collaboration point which would usually be the case in a conventional compression scheme. Using CS, the amount of signal samples that need to be transferred to the FC from the sensors can be significantly reduced. This may lead to reduction of power consumption at the sensor nodes, which was discussed in **Section 3.3**. In summary, each sensor with CS can save power by not needing to run complex compression operations on board and by cutting down signal transmissions.

Distributed sensors usually observe a single globally occurring event and thus the observed signals are often correlated with each other. We considered two types of correlations: intra- and inter-sensor signal correlation. We provided the sparse signal models which encompass both types of correlation in **Sections 4.1** and **4.2**.

The FC receives the compressed signals from the sensors. The FC then recovers the original signal waveforms from the compressed signals using a CS recovery algorithm. We considered two types of algorithms. One is a greedy algorithm type, which includes the orthogonal matching pursuit (OMP) and the simultaneous orthogonal matching pursuit (SOMP) algorithms, discussed in **Section 5.1**. The other is a gradient type for which we used

the primal-dual interior point (PDIP) method, in **Section 5.2**.

Finally, we presented simulations results in which the CS based WSN system parameters such as the number of measurements, the sparsity, and the signal length were varied. We discussed the use of a joint recovery scheme at the FC. A CS recovery algorithm is referred to as the *joint recovery* scheme when it utilizes inter-sensor signal correlation as well. In contrast, when the inter-sensor signal correlation is not utilized, it is referred to as the *separate recovery* scheme. In the joint recovery scheme, inter-sensor signal correlation information is incorporated in the formation of recovery equation as shown Eq. (13) and (15). In the separate recovery scheme, a sensor signal recovery is done individually and independently from the recovery of other sensor signals. We compared the results of the joint recovery with those of the separate recovery scheme. We have shown that correlation information can be exploited and the number of measurements needed for exact reconstruction can be significantly reduced as shown in **Figure 14**. It means that the traffic volume transmitted from the sensors to the FC can decrease significantly without degrading the quality of the recovery performance. (**Section 6**)

We have shown that the CS is an efficient and effective signal acquisition and sampling framework for WSN which can be used to save transmittal and computational power significantly at the sensor node. This CS based signal acquisition and compression scheme is *very simple*, so it is suitable for inexpensive sensors. The number of compressed samples required for transmission from each sensor to the FC is *significantly small*, which makes it perfect for sensors whose operational power is drawn from onboard battery. Finally, the joint CS recovery at the FC exploits signal correlation and enables *Distributed Compressive Sensing*.

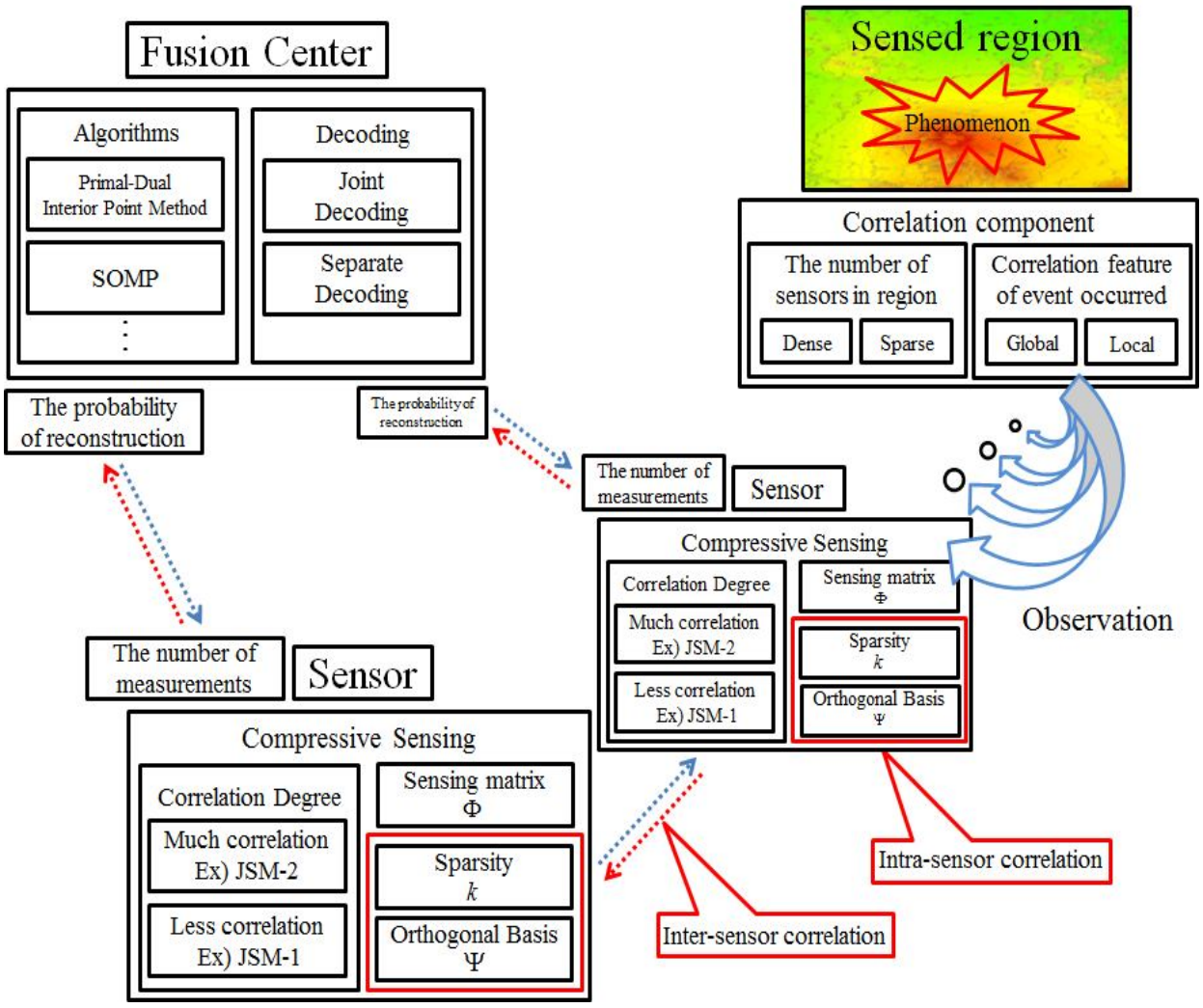

Figure 16. Summary of CS application in WSN

8. References

[1] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289-1306, Apr. 2006

[2] D. L. Donoho and J. Tanner, "Precise undersampling theorems," *Proc. IEEE*, vol. 98. pp. 913-924, May 2010.

[3] R. G. Baraniuk, "Lecture notes: Compressed sensing," *IEEE Signal Process. Mag.*, pp. 118-121, July 2007.

[4] J. Romberg, "Imaging via compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 14-20, March 2008.

[5] A. Y. Yang, M. Gastpar, R. Bajcsy, and S. S. Sastry, "Distributed sensor perception via sparse representation," to appear in *Proc. IEEE*.

[6] D. L. Donoho and M. Elad, "Maximal sparsity representation via ℓ_1 minimization," *Proc. Natl. Acad. Sci.*, vol. 100, pp. 2197-2202, March 4, 2003.

[7] J. Solobera, "Detecting forest fires using wireless sensor networks with Waspmote."

[8] A. Hac, "*Wireless Sensor Network Designs*," John Wiley & Sons, Ltd., 2003.

[9] D. Baron, M. F. Duarte, S. Sarvotham, M. B. Wakin, and R. G. Baraniuk, "An information theoretic approach to distributed compressed sensing," in *Proc. 43rd Allerton Conf. Comm., Control, Comput.*, Sept. 2005.

[10] M. F. Duarte, S. Sarvotham, D. Baron, M. B. Wakin, and R. G. Baraniuk, "Distributed compressed sensing of jointly sparse signals," *Asilomar Conf. on Signals, Systems and Computers*, pp. 1537-1541, 2005.

[11] M. Mishali and Y. C. Eldar, "Reduce and boost: Recovering arbitrary sets of jointly sparse vectors," *IEEE Trans. Signal Process.*, vol. 56, no. 10, pp. 4692-4702, 2008.

[12] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inform. Theor.*, vol. 53, no. 12, pp. 4655-4666, Dec. 2007.

[13] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Simultaneous sparse approximation via greedy pursuit," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 725, pp. v/72I-v/724, 2005.

[14] M. E. Davies and Y. C. Eldar, "Rank awareness in joint sparse recovery," *Arxiv preprint arXiv:1004.4529*, 2010.

[15] E. Candes and J. Romberg, Caltech, L_1-Magic: Recovery of sparse signals via convex programming, Oct 2005.