A Node-Based Time Slot Assignment Algorithm for STDMA Wireless Mesh Networks

Authors: W. Chen, and Chin-Tau Lea

Publication: IEEE Trans. Veh. Tech., Jan. 2013
Speaker: Asif Raza

Short summary: In this paper authors present a link capacity model for spatial time-division multiple access (STDMA) mesh networks. It makes use of a simplified transmission model that also considers channel fading. The model then forms the basis of a node-based slot-assignment and scheduling algorithm. This algorithm enables the user to exploit multiuser diversity that results in optimizes network throughput. The presented algorithm shows significant improvement in the throughput when compared with existing slot-assignment methods.

I. Introduction

In STDMA network the transmission time of a channel is divided into slots where multiple slots constitute a frame. These slots are assigned to potential users of the network. The goal of slot assignment scheme is to maximize network throughput. Existing assignment algorithms in STDMA make use of simplified transmission model which do not consider the time-varying fading behavior of a wireless channel. This results in slot wastage when link is in deep fade. The slot is also wasted if scheduled link has no traffic to transmit. This degrades the STDMA network throughput. Therefore a dynamic slot-assignment with that should exploit multiuser diversity is required. However sheer complexity involved in coordinating with all nodes and generating scheduling map in a reasonable time makes this approach impractical. In order to fix these issues the authors present a node-based slot-assignment scheme in which scheduling in each slot is done for nodes not for links. Their contributions include:

- Defining link capacity: a model that includes channel fading. It ensures that whichever link is used by a node will not change the interference profiles on the links selected by other users.
- Node-based time-slot assignment and scheduling algorithms.

II. System Model

Wireless STDMA mesh network with fixed routers.

Transmissions are organized in frames.

Synchronization among nodes provided through GPS.

Set of nodes are identified and assigned to a slot for their transmission.

Each node maintains a separate queue for each outgoing link and performs scheduling without coordination with other nodes.

Multiprotocol Label Switching (MPLS) multipath routing is used for routing however packets are transmitted in sequence.

Adaptive modulation and time varying fading channels are considered. It is also assumed that wireless channels undergo slow fading. Due to fading channel an instant channel gain will be fed back to transmitter. The duration for feedback is no longer than coherence time (the time for which channel conditions remain same)

Adaptive modulation is implemented that each data packet can be fragmented into multiple segments and each segment can be transmitted in with lowest data rate. If high data rate is available then multiple segments can be transmitted per slot duration.

III. Link Capacity Modeling

Each node has multiple links and it can exploit multiuser diversity i.e. different links have different traffic and fading conditions. A channel model is presented that includes shadowing and slow fading.

A. Signal to interference and noise ratio (SINR) Formulation

$h_{r,t}$: Channel response function from transmitter 't' and receiver 'r'

x_t: Signal from 't'

$I_r, n_I, t_i^{'}$: Set of transmitters causing interference to 'r', number of transmitters and i[th] transmitter in I_r respectively. Power control is not considered therefore transmission power of 't' is $pt = E\left(\left|x_t\right|^2\right)$. Let n_0 be thermal noise with power equal to k then received power at 'r' is

$$y_{r,t} = h_{r,t}x_t + \sum_{i=1}^{n_I} h_{r,t_i^{'}} x_{t_i^{'}} + n_0 \qquad (1)$$

SINR at receiver 'r' is expressed as:

$$\gamma_{r,t} = \frac{\left|h_{r,t}x_t\right|^2}{\sum_{i=1}^{n_I}\left|h_{r,t_i^{'}}x_{t_i^{'}}\right|^2 + \kappa} = \frac{\left|h_{r,t}\right|^2 p_t}{\sum_{i=1}^{n_I}\left|h_{r,t_i^{'}}\right|^2 p_{t_i^{'}} + \kappa} = \frac{s_0}{\sum_{i=1}^{n_I} s_i + \kappa} \qquad (2)$$

Here $s_0 = \left|h_{r,t}\right|^2 p_t$ *and* $s_i = \left|h_{r,t_i^{'}}\right|^2 p_{t_i^{'}}$. The Channel response function consists of three parts:

- Path loss
- Shadowing
- Fading

$$h_{r,t} = \sqrt{l_{r,t}^{-\alpha} 10^{\frac{f_{r,t}}{10}} \pi_{r,t}} \qquad (3)$$

Where $l_{r,t}$ is distance between 't' and 'r', $\alpha \mapsto [2-4]$ (constant), $10^{\frac{f_{r,t}}{10}}$ is shadowing effect and it is modeled as a log-normal distributed random variable. $\pi_{r,t}$ is fading effect and it is defined as complex Gaussian RV with mean and variance equal to 0 and 1 respectively. PDF of s_0 and s_i are defined as:

$$p_{s_0}(\alpha_0) = \tfrac{1}{\rho_0} e^{-\frac{\alpha_0}{\rho_0}} \qquad (4a)$$

$$p_{s_i}(\alpha_i) = \tfrac{1}{\rho_i} e^{-\frac{\alpha_i}{\rho_i}} \qquad (4b)$$

Here $\rho_0 = E(s_0) = l_{r,t}^{-\alpha} 10^{\frac{f_{r,t}}{10}} p_t$ and $\rho_i = E(s_i) = l_{r,t_i'}^{-\alpha} 10^{\frac{f_{r,t_i'}}{10}} p_{t_i'}$

B. PDF of SINR

Case 1: no interference is observed by receiver 'r' i.e. (I_r=0, n_I=0) then PDF of $\gamma_{r,t}$, is defined as: let $\delta = \gamma_0 / \kappa$

$$p_{\gamma_{r,t}}(z) = p_{s_0}(z\kappa) = \tfrac{1}{\delta} e^{-\frac{z}{\delta}} \qquad (5)$$

Probability that $\gamma_{r,t}$ is smaller than w is defined as:

$$Pr(\gamma_{r,t} \geq w) = \int_w^\infty p_{\gamma_{r,t}}(z)dz = e^{-\frac{w}{\delta}} \qquad (6)$$

Case 2: unit interference is observed by 'r' i.e. (I_r>0, n_I=1) then PDF of term (s_i+k i.e. denominator of equ.2) is defined as:

$$p_{s_1+\kappa}(v) = \tfrac{1}{\sigma_1} e^{-\frac{v-\kappa}{\sigma_1}} \qquad (7)$$

Finally PDF of $\gamma_{r,t}$ is defined as:

$$p_{\gamma_{r,t}}(z) = \int_\kappa^\infty v p_{s_0}(vz) p_{s_1+\kappa}(v)dv = \int_\kappa^\infty \tfrac{v}{\sigma_0} e^{-\frac{vz}{\sigma_0}} \tfrac{1}{\sigma_1} e^{-\frac{v-\kappa}{\sigma_1}} dv = \tfrac{\kappa + \frac{1}{u}}{\sigma_0 \sigma_1 u} e^{\left(\frac{\kappa}{\sigma_1} - u\kappa\right)} \qquad (8)$$

Probability that $\gamma_{r,t}$ is smaller than w is defined as:

$$Pr(\gamma_{r,t} \geq w) = \int_w^\infty p_{\gamma_{r,t}}(z)dz = \frac{\sigma_0}{\sigma_0 + w\sigma_1} e^{-\frac{w}{\delta}} \qquad (9)$$

Case 3: more than one interferers are present in I_r i.e. ($n_I > 1$) then PDF of $\sum_{i=1}^{n_I} s_i$ can be defined as:

$$p_I(v) = P_{\sum_{i=1}^{n_I} s_i}(v) \otimes p_{s_{n_I}}(v) = \sum_{i=1}^{n_I} \frac{b_i}{\sigma_i} e^{-\frac{v}{\sigma_i}} \qquad (10)$$

Here $b_i = \prod_{j=1, j\neq i}^{n_I} \left(\sigma_i / \sigma_i - \sigma_j \right) and \sum_{i=1}^{n_I} b_i = 1$. The PDF of term ($s_i$+k i.e. denominator of equ.2) is defined as: $p_{I+\kappa}(v) = p_I(v-\kappa)$. Finally the PDF of $\gamma_{r,t}$:

$$p_{\gamma_{r,t}}(z) = \int_{\kappa}^{\infty} v p_{s_0}(vz) p_{I+\kappa}(v) dv = \sum_{i=1}^{n_I} d_i \left(\frac{\kappa}{q_i} + \frac{\kappa}{q_i^2} \right) e^{-\kappa q_i} \qquad (11)$$

Here $q_i = \left(z / \sigma_0 \right) + \left(1 / \sigma_i \right)$ and $d_i = \left(b_i / \sigma_0 \sigma_i \right) e^{-\kappa q_i}$. Probability that $\gamma_{r,t}$ is smaller than w is defined as:

$$Pr\left(\gamma_{r,t} \geq w \right) = \int_{w}^{\infty} p_{\gamma_{r,t}}(z) dz = \sigma_0 e^{-\frac{w}{\delta}} \sum_{i=1}^{n_I} \frac{b_i}{\sigma_0 + \sigma_i w} \qquad (12)$$

Finally Link Capacity can then be determined as:

$$c_{r,t}(I_r) = \sum_{i=1}^{\xi-1} c_i \cdot Pr\left(\gamma_{thr}^{i} \leq \gamma_{r,t} < \gamma_{thr}^{i+1} \right) + c_{\xi} \cdot Pr\left(\gamma_{r,t} \geq \gamma_{thr}^{\xi} \right)$$

Where $c_{r,t}(I_r)$ is average data rate between 't' and 'r', given interference set I_r $c_{r,t}(I_r))$ $c_{r,t}(I_r)$

IV. Proposed Time-Slot Assignment Algorithm

TDMA frame consists of a fixed number of slots is considered. The set of transmitting links that are activated in a given slot is called a **link pattern**, and the set of nodes activated in a given slot is called a **node pattern**.

A. Formulation of Node-Based Time-Slot Algorithm

Notations:

V: set of nodes

E: set of links

NP: Node Pattern

$tx_e, rx_e; e \in E$ transmitter and the receiver of link e, respectively,

$E_{s,p} = \{ e \mid e \in E, p \in s, s \in NP, tx_e = p \}$ set of links that can be used at node p, where p∈s i.e p is activated in node pattern *s*);

μ_s = portion of time that is assigned to node pattern (s) in a frame, where,

$\sum_{s \in NP} \mu_s = 1$

$\left\{ò_{s,p,e} \mid e \in E_{s,p}\right\}$:portion of time that is assigned to each link of node p in node pattern s

F: set of flows in the system; where flow defines all traffic that belongs to (S, D) pair

h_f : traffic demand for flow f, where $f \in F$

S_f : source of flow f

D_f : destination of flow f

$x_{f,e}$: percentage of traffic that flow f passes through link e,

Calculations

Link congestion: it is total amount of traffic routed through the link 'e' over its average capacity (c_e) i.e. $r_e = \left(\sum_{f \in F} \frac{x_{f,e}}{c_e} \right)$ where link capacity (data rate between transmitter 't' and receiver 'r' is defined as: $c_e = \sum_{\{s \mid s \in NP, p \in s, e \in E_{s,p}\}} c_{s,e} ò_{s,p,e}$.

Thus network congestion ratio 'r' is the maximum of all link congestion ratios, i.e. $r = \max_{e \in E} r_e$

Optimal node-based slot assignment scheme is one which minimizes congestion 'r':

$$\min \quad r \qquad (13a)$$

$$s.t \quad \frac{\sum_{f \in F} x_{f,e} h_f}{\sum_{\{e \mid s \in NP, p \in s, e \in E_{s,p}\}} c_{s,e} ò_{s,p,e}} \le r \quad (13b)$$

$$\sum_{\{e \mid e \in E_{s,p}\}} ò_{s,p,e} \le \mu_s \qquad (13c)$$

$$\sum_{\{e \mid p \in s, e \in E_{s,p}, rx_e = q\}} ò_{s,p,e} \le \mu_s \qquad (13d)$$

$$\sum_{s \in NP} \mu_s = 1 \qquad (13e)$$

$$\mu_s \ge 0, ò_{s,p,e} \ge 0, r \ge 0 \qquad (13f)$$

$$\sum_{\{e \mid tx_e = v\}} x_{f,e} - \sum_{\{e \mid rx_e = v\}} x_{f,e} = 0 \quad (13g)$$

$$\sum_{\{e \mid tx_e = S_f\}} x_{f,e} - \sum_{\{e \mid rx_e = S_f\}} x_{f,e} = 1 (13h)$$

$$x_{f,e} \ge 0 \qquad (13i)$$

Problem 13 is the optimization problem, whose purpose is to find the set of $\grave{o}_{s,p,e}$ that will lead to the optimal objective function. Constraint 13c represents that in node pattern s, for any node p∈s, p can transmit to only one node at one time. 13d ensures that a node q can receive from only one node at one time while q ∉ s and p∈s. 13f and 13i ensures non-negativity constraints. Constraint 13b is non-linear therefore $\mu_s r$ and $\grave{o}_{s,p,e} r$ are replaced by $\theta_s \; and \; \rho_{s,p,e}$ respectively. Therefore final formulation is defined as:

$$\min \quad \sum_{s \in NP} \theta_s \qquad (14a)$$

$$s.t \quad \sum_{f \in F} x_{f,e} h_f \le \sum_{\{e | s \in NP, p \in s, e \in E_{s,p}\}} c_{s,e} \grave{o}_{s,p,e} \qquad (14b)$$

$$\sum_{\{e | e \in E_{s,p}\}} \rho_{s,p,e} \le \theta_s \qquad (14c)$$

$$\sum_{\{e | p \in s, e \in E_{s,p}, rx_e = q\}} \rho_{s,p,e} \le \theta_s \qquad (14d)$$

$$\theta_s \ge 0, \rho_{s,p,e} \ge 0 \qquad (14e)$$

$$\sum_{\{e | tx_e = v\}} x_{f,e} - \sum_{\{e | rx_e = v\}} x_{f,e} = 0 \qquad (14f)$$

$$\sum_{\{e | tx_e = S_f\}} x_{f,e} - \sum_{\{e | rx_e = S_f\}} x_{f,e} = 1 \qquad (14g)$$

$$x_{f,e} \ge 0 \qquad (14h)$$

Authors describe that the presented formulation can handle scheduling of node patterns by using Linear Programming approach. However for link based approach, listing all link patterns does not work by using LP formulation. Therefore column generation method is used to tackle the problem.

B. Frame Construction and Throughput Loss due to Frame Quantization

Frame is constructed as: $n_f = \sum_{s \in NP} \left[z \mu_s \right]$ here z is frame length and function [x] rounds 'x' to nearest integer.

The frame quantization will change the portion of time assigned to all patterns (μ_s). Therefore parameters like minimum congestion ratio r_z, the optimal link capacities (c_e) and the routing scheme $x_{f,e}$

will change. These parameters need to be recomputed as follows. let z_s be number of slots assigned to node pattern 's' in a frame.

$$\min \quad \varsigma_z \qquad (15a)$$

$$s.t \quad \sum_{f\in F} y_{f,e} h_f \le \sum_{\{s|s\in NP, p\in s, e\in E_{s,p}\}} c_{s,e} \grave{o}_{s,p,e} \qquad (15b)$$

$$\sum_{\{e|e\in E_{s,p}\}} \grave{o}_{s,p,e} \le \frac{z_s}{\sum_{s\in NP} z_s} \qquad (15c)$$

$$\sum_{\{e|p\in s, e\in E_{s,p}, rx_e=q\}} \grave{o}_{s,p,e} \le \frac{z_s}{\sum_{s\in NP} z_s} \qquad (15d)$$

$$\sum_{\{e|tx_e=v\}} y_{f,e} - \sum_{\{e|rx_e=v\}} y_{f,e} = 0 \qquad (15e)$$

$$\sum_{\{e|tx_e=S_f\}} y_{f,e} - \sum_{\{e|rx_e=S_f\}} y_{f,e} = h_f \varsigma_z \qquad (15f)$$

$$y_{f,e} \ge 0,\ \grave{o}_{s,p,e} \ge 0 \qquad (15g)$$

Here $\varsigma_z = 1/r_z, \mu_s = \left(z_s / \sum_{s\in NP} z_s \right), y_{f,e} = \left(x_{f,e} / r_z \right) = x_{f,e}\varsigma_z$

C. Column Generation Method

Column generation is an algorithm for solving large LP problems. Most of the variables are usually non-basic and assume zero values in the optimal solution, only a subset of variables are needed for solving the problem. Column generation method considers only the variables which have potential to improve the objective function. It splits the problem into master problem and subproblem. Master problem is the original problem with subset of variables being considered. In subproblem it uses duality approach to select new variables to be added to master problem to improve its result.

Master Problem: it is same as defined in problem 14 except that NP is replaced with NP' (subset of NP which is feasible for 14). Solution of master problem shall provide a routing and slot-assignment scheme.

Subproblem: is a new problem created to identify a new node pattern to add to master problem and it is defined as:

$$\min_{s\in\{NP\setminus NP'\}} rp_s \qquad (15)$$

Here rp_s is reduced cost of node pattern 's' in the column generation algorithm and it is optimal value of following problem:

$$\max\left(1-\sum_{\{p|p\in s\}}\omega_{s,p}-\sum_{\{q|p\in s,e\in E_{s,p},rx_e=q\}}\tau_{s,q}\right) \quad (16)$$
$$s.t \qquad \omega_{s,p}+\tau_{s,p}-\phi_e c_{s,e}\geq 0$$

Here $\omega_{s,p}\ and\ \tau_{s,q}$ are variables that are associated with the transmitter p and the receiver q in node patterns s. Well the question is which node pattern should be included into NP' ?

According to duality theory if master problem is optimal then rp_s is always non-negative for any pattern in NP. The node patterns with negative rp_s can improve the result if they are added into NP'. So algorithm will iterate between two phases until no more patterns can be added to NP'.

Algorithm steps are defined as follows:

Step 1: Set node pattern A= ϕ and rp_A =0,

Step 2: Identify $A^c = v$ and compute $rp_{A'_v}$ for node pattern $A'_v\, s.t\, A'_v = \{A, v\}$.

Step 3: select v from A^c with minimum $rp_{A'_v}$ and compute rp_A of A.

Step 4: If $rp_{A'_v} \geq rp_A$, node v will be deleted from A^c and add it A.

Step 5: If $A^c \neq \phi$ stop else go to step 3.

V. SCHEDULING ALGORITHMS

Two scheduling algorithms are proposed in which each node will locally schedule its link transmissions without inter-node coordination and without disturbing interference profiles of other nodes.

A. Scheme 1

Every node 't' in node pattern 's' assigns a transmission probability to every link associated with 't'. The set of transmission probabilities is then defined as:

$$P_{s,t}=\left\{p_{s,t,e}\,|\,e\in E_{s,t}, p_{s,t,e}=\frac{\partial_{s,t,e}}{\mu_s}\right\} \qquad (17)$$

The region [0,1] is then divided into subregions, one for each link in $|E_{s,t}|$, and length of regions is set according to $P_{s,t}$. The algorithm works as follows; Suppose a node pattern 's' is activated in slot x. Each node $t \in s$ will generate a RV w, uniformly distributed within [0, 1]. The node will then schedule

link into which subregion w falls. If selected link (e) is not usable (either due to fading or no traffic) the scheduler will check link next to 'e' one by one until a usable link is found.

B. Scheme 2

Scheme 1 does not consider link quality while scheduling the links. Therefore authors presented another scheduling mechanism.

Selection criteria = (queue length * link capacity)

Each node maintains two queues for each of its link:

1) a real data queue to store packets and

2) A shadow queue for scheduling.

These queues of link 'e' whose transmitter can be activated in slot 'x', are defined as:

$$q_e(x) = q_e(x-1) + a_e(x) - d_e(x)$$
$$q'_e(x) = q'_e(x-1) + a'_e(x) - d'_e(x)$$

Here $q_e(x)$ and $q'_e(x)$ are lengths of the real queue and shadow queue respectively. $a_e(x)$, $a'_e(x)$, $d_e(x)$ and $d'_e(x)$ are the number of arrivals and departures for the two queues in 'x', respectively. In shadow queue the term $a'_e(x)$ is defined as: $a'_e(x) = \left(1 + \gamma_x\right)\sum_{t=0}^{x} a_e(t)$ i.e. it is used to smooth the incoming traffic from source or previous hop.

Packets departing from link 'e' are defined as: $d_e(x) = \min\left\{\tilde{c}_e(x), q_e(x)\right\}$. Here $\tilde{c}_e(x)$ is instant capacity of link 'e' in slot 'x'. Thus scheduling, in slot 'x', the scheduler in node $t \in s$ will select the link from all its associated links with a maximum value of $q'_e(x)\tilde{c}_e(x)$. In doing so, it tries to strike the optimal balance between link quality and traffic backlog.

VI. Simulation and Results

A. Simulation Environment and Settings

Linear optimization toolbox of MATLAB is used for proposed routing and slot-assignment algorithm. C++ program is then used to inspect maximum achievable throughput for different scheduling schemes.

The physical-layer parameters are summarized as follows:

- Transmission power: 20 dBm.
- Thermal noise:−90 dBm.
- Path loss(α):3.5.
- Variance of shadow fading: 4 dBm.

• Minimal distance of two nodes: 15 m.

• Slot duration: 0.22 ms.

• Frame size: 100 slots. So frame length = 22 ms.

The mapping between the following data rates and SINR threshold is summarized as follows.

• 54 Mb/s: 24.56 dBm.

• 48 Mb/s: 24.05 dBm.

• 36 Mb/s: 18.80 dBm.

• 24 Mb/s: 17.04 dBm.

• 18 Mb/s: 10.79 dBm.

• 12 Mb/s: 9.03 dBm.

• 9 Mb/s: 7.78 dBm.

• 6 Mb/s: 6.02 dBm.

Network Topology: two networks 15-node and 30-node with two gateway nodes and three gateway nodes are considered, respectively.

The traffic load of each flow is assumed to be the same i.e.,h_f =1Mb/s,

Throughput loss due to Frame Quantization:

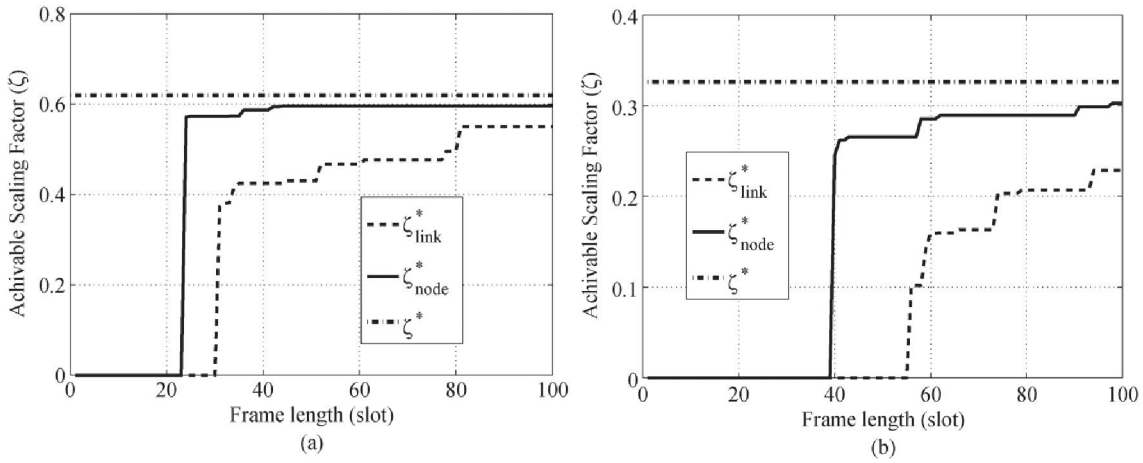


Fig. 1. Achievable throughput after frame generation for (a) 15- and (b) 30-node networks.

The solid (ζ^*_{node}), dashed (ζ^*_{link}) and dashed–dotted (ζ^*) lines indicate the achievable throughput in node-based, link based and before frame construction (I.e. upper bound on throughput) respectively. The flat area represents the range where the performance does not improve. Note that (ζ^*_{node}) and (ζ^*_{link}) are function of 'z' and are not always monotonically increasing due to the quantization involved in the process, and small oscillation occurs within a short range of z. This is why, in Fig. 1(a) and (b), the curves move up in steps.

TABLE I
PERFORMANCE COMPARISONS FOR THE LINK- AND NODE-BASED SCHEMES

N	ζ^*	ζ^*_{node} (ζ^*_{link})	n_{use}	$\zeta^*_{exp,1}$	$\zeta^*_{exp,2}$	$\zeta^*_{det,1}$	$\zeta^*_{det,2}$
15 (Node)	0.62	0.60	17	0.62	0.77	0.62	0.76
15 (Link)	0.62	0.55	26	NA	NA	NA	NA
30 (Node)	0.33	0.30	30	0.31	0.39	0.31	0.39
30 (Link)	0.33	0.23	54	NA	NA	NA	NA

From the table it is clear that ζ^*_{node} approaches ζ^* much faster than ζ^*_{link}. Moreover difference in throughput between ζ^*_{node} and ζ^*_{link} is also significant as shown in table 1.

The optimal scaling factors of the ζ^*_{node} for schemes 1 and 2 under the Poisson and a deterministic arrival process are denoted as shown by $\zeta^*_{\text{exp},1}$, $\zeta^*_{\text{exp},2}$, $\zeta^*_{\text{det},1}$, $\zeta^*_{\text{det},2}$ respectively.

{the ζ^*_{node} is derived from problem 14 and it does not include multi-user diversity gain. Therefore, it can be viewed as a lower bound of the two proposed scheduling schemes 1 and 2 As shown in the Table I.

It is also clear from the table that , both (posisson and deterministic arrival rates) $\zeta^*_{\text{exp},1}$, $\zeta^*_{\text{det},1}$ are only slightly larger than ζ^*_{node} for the 15- and 30-node networks. The difference is only about 3%. This is because scheme 1 tries to follow $\grave{o}_{s,p,e}$ i.e. portion of time that is assigned to each link of node p in node pattern s and does not select a link with the best quality. However, the situation is different in scheme 2, because link quality is part of the selection criteria. With scheme 2, $\zeta^*_{\text{exp},1}$, $\zeta^*_{\text{det},1}$ are about 26% larger than ζ^*_{node} for the 15-node network and 30% larger for the 30-node network.

On the Recovery Limit of Sparse Signal Using Orthogonal Matching Pursuit

Authors: Jian Wang and Byonghyo Shim

Publication: IEEE T. Signal Processing, SEPT.2012
Speaker: Sangjun Park

Short summary: In the paper, the authors give a sufficient condition of the Orthogonal Matching Pursuit (OMP) algorithm. In [2], Wakin and Davenport insisted that OMP can reconstruct any K sparse signal if $\delta_{K+1} < 1/\left(3\sqrt{K}\right)$, where δ_K is the restricted isometry constant. However, in this talk, an improved sufficient condition that guarantees the perfect recovery of OMP is presented

I. FINAL SUMMARY OF THE PAPER

a. A strategy of the proof of Theorem 1.

1) We aim to find a condition such that the OMP algorithm selects a correct index in the first iteration.

=> We need to show that $\min_{i\in\mathcal{I}} \left|\langle \mathbf{a}_i, \mathbf{y}\rangle\right| > \max_{j\notin\mathcal{I}} \left|\langle \mathbf{a}_j, \mathbf{y}\rangle\right|$. (e.g., see from (7) to (10).)

2) Let us suppose that the initial k iterations of the OMP algorithm are successful, and that $\mathcal{T}^k$ is the estimated support set after the initial k iterations. Now, the OMP algorithm selects a correct index, which belongs to $\mathcal{I}\setminus\mathcal{T}^k$, in the $k+1$ iteration.

=> Clearly, $\mathcal{T}^k \subseteq \mathcal{I}$, therefore $\mathbf{r}^k = \mathbf{y} - \mathbf{A}_{\mathcal{T}^k}\hat{\mathbf{x}}_{\mathcal{T}^k} = \mathbf{P}^{\perp}_{\mathcal{T}^k}\mathbf{y} \in span\left(\mathbf{A}_{\mathcal{I}}\right)$ can be considered as a linear combination of the K columns of $\mathbf{A}_{\mathcal{I}}$. Thus, $\mathbf{r}^k = \mathbf{A}\mathbf{b}$, where $\|\mathbf{b}\|_0 \le K$, and $\mathrm{supp}(\mathbf{b}) \subseteq \mathrm{supp}(\mathbf{x}) = \mathcal{I}$.

=> Again, we find a condition such that the OMP algorithm selects a correct index t^{k+1} which belongs to $\mathrm{supp}(\mathbf{b})$.

=> Furthermore, for any $i \in \mathcal{T}^k$, we have $\left|\langle \mathbf{r}^k, \mathbf{a}_i\rangle\right| = 0$. Thus, $t^{k+1} \in \mathcal{I}\setminus\mathcal{T}^k$.

3) Thus, we conclude that the OMP algorithm can reconstruct K sparse signal provided that the condition of 1) and the condition of 2) are satisfied

b. Comparison between the result by the authors in this paper and the result by Davenport and Wakin.

According to the authors, the improvement is possible due to 1) contradiction based construction of the success condition in the first iteration ($\min_{i\in\mathcal{I}}\left|\langle \mathbf{a}_i,\mathbf{y}\rangle\right| > \max_{j\notin\mathcal{I}}\left|\langle \mathbf{a}_j,\mathbf{y}\rangle\right|$) , and 2) observation that the residual in the general iteration preserves the sparsity level of the input signal. ($\mathbf{r}^k = \mathbf{Ab}$, where $\|\mathbf{b}\|_0 \le K$, and $\mathrm{supp}(\mathbf{b}) \subseteq \mathrm{supp}(\mathbf{x}) = \mathcal{I}$).

In fact, the authors again improved the result by Davenport and Wakin.

The more detailed explanations are referred to the paper.

c. Future Works

1) Can we apply the techniques, which are used in the proof, to find a sufficient condition of an algorithm based from the OMP algorithm? For example, the SOMP algorithm selects a index i such as $\arg\max_i \left\|\mathbf{a}_i^T\mathbf{R}^{(k)}\right\|_q$, where $\mathbf{R}^{(k)} = \left[\mathbf{r}_1^{(k)} \quad \cdots \quad \mathbf{r}_S^{(k)}\right]$, $\mathbf{r}_i^{(k)} = \mathbf{y}_i - \mathbf{A}_{T^k}\hat{\mathbf{x}}_{i,T^k}$, and $q = 1 \text{ or } 2$. Can we find a condition such that the SOMP algorithm selects a correct index?

II. History of sufficient conditions of the OMP algorithm

In the below table 1, sufficient conditions that the OMP algorithm reconstructs a K spars signal from a set of linear measurements $\mathbf{y} = \mathbf{Ax}$, where $\mathbf{A} \in \Re^{M\times N}$ ($N > M$), are given.

Year	A sufficient condition
2007[1]	$\mu < 1/(2K-1)$
2010[2]	$\delta_{K+1} < 1/\left(3\sqrt{K}\right)$

Besides, there are many theoretical papers which analyze algorithms based on the OMP algorithm. In here, it is not scope of this seminar. Therefore, we do not care about them.

III. System Model

Let us consider the below equation:

$$\mathbf{y} = \mathbf{Ax}, \tag{1}$$

where $\mathbf{A} \in \Re^{M\times N}$ ($N > M$), and $\mathbf{x} \in \Re^N$ is a K sparse signal, and $\mathbf{y} \in \Re^M$ is a set of linear measurements. The smallest constant δ_K called “the restricted isometry constant” satisfies

$$(1-\delta_K)\|\mathbf{x}\|_2^2 \le \|\mathbf{A}\mathbf{x}\|_2^2 \le (1+\delta_K)\|\mathbf{x}\|_2^2 \tag{2}$$

for any K sparse signal $\mathbf{x}$.

IV. Main Results

A. Improved Recovery Bound of the OMP algorithm

Theorem 1: For any K sparse signal $\mathbf{x}$, the OMP algorithm perfectly reconstructs $\mathbf{x}$ from $\mathbf{y}$ if the isometry constant δ_{K+1} satisfies

$$\delta_{K+1} < \frac{1}{\sqrt{K}+1}. \tag{3}$$

In this talk, we try to understand a proof of Theorem 1.

Before we study the proof, let us consider whether the OMP algorithm perfectly reconstructs $\mathbf{x}$ or not if $\delta_{K+1} = 1/\sqrt{K}$.

B. The OMP algorithm can fail under $\delta_{K+1} = 1/\sqrt{K}$.

Example 1: Let us consider the problem of reconstructing a K sparse signal $\mathbf{x} \in \Re^{K+1}$ such as $x_{K+1} = 0$, and $x_i = 1$ for $i = 1, \cdots, K$ from $\mathbf{y} = \mathbf{A}\mathbf{x}$, where

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} 1 & b & \cdots & b \\ b & 1 & & \vdots \\ \vdots & & \ddots & b \\ b & \cdots & b & 1 \end{bmatrix} \in \Re^{(K+1)\times(K+1)}.$$

Obviously, all the Eigen values of $\mathbf{A}^T\mathbf{A}$ are $\lambda_1 = \lambda_2 = \cdots = \lambda_K = 1-b$, and $\lambda_{K+1} = 1+Kb$. (See Example 1 on Appendix). When we assume $b = -1/\left(K\sqrt{K}\right)$, $\mathbf{A}^T\mathbf{A}$ becomes

$$\mathbf{A}^T\mathbf{A} = \begin{bmatrix} 1 & -1/\left(K\sqrt{K}\right) & \cdots & -1/\left(K\sqrt{K}\right) \\ -1/\left(K\sqrt{K}\right) & 1 & & \vdots \\ \vdots & & \ddots & -1/\left(K\sqrt{K}\right) \\ -1/\left(K\sqrt{K}\right) & \cdots & -1/\left(K\sqrt{K}\right) & 1 \end{bmatrix} \in \Re^{(K+1)\times(K+1)}, \tag{4}$$

and the smallest and biggest Eigen values are

$$\lambda_{\min} = 1-1/\sqrt{K}, \text{ and } \lambda_{\max} = 1+1/\left(K\sqrt{K}\right).$$

Therefore, we have $\delta_{K+1} = 1/\sqrt{K}$ (In fact, all the Eigen values of $\mathbf{A}_{\mathcal{S}}^T \mathbf{A}_{\mathcal{S}}$ must be contained in the interval $\left[1-\delta_{|\mathcal{S}|}, 1+\delta_{|\mathcal{S}|}\right]$, Thus, $\delta_{K+1} = \max\left\{\lambda_{\max}\left(\mathbf{A}_{\mathcal{S}}^T \mathbf{A}_{\mathcal{S}}\right)-1, 1-\lambda_{\min}\left(\mathbf{A}_{\mathcal{S}}^T \mathbf{A}_{\mathcal{S}}\right)\right\}$). Now, we investigate a quantity $\left|\langle \mathbf{a}_i, \mathbf{y}\rangle\right|$ for $i = 1, \cdots, K+1$. For the OMP algorithm to reconstructs $\mathbf{x}$, $\left|\langle \mathbf{a}_{K+1}, \mathbf{y}\rangle\right|$ must be less than any $\left|\langle \mathbf{a}_i, \mathbf{y}\rangle\right|$ for $i = 1, \cdots, K$. This is reason that we investigate the quantities. First, for $i \in \{1, \cdots, K\}$, we have

$$\begin{aligned} \left|\langle \mathbf{a}_i, \mathbf{y}\rangle\right| &\overset{(a)}{=} \left|\langle \mathbf{a}_i, \mathbf{A}\mathbf{x}\rangle\right| \\ &\overset{(b)}{=} \left|\left\langle \mathbf{A}^T \mathbf{a}_i, \mathbf{x}\right\rangle\right| \\ &\overset{(c)}{=} 1 - \frac{K-1}{K\sqrt{K}}, \end{aligned} \tag{5}$$

where (a) from the fact $\mathbf{y} = \mathbf{A}\mathbf{x}$, (b) from the fact $\langle \mathbf{a}_i, \mathbf{A}\mathbf{x}\rangle = \mathbf{a}_i^T \mathbf{A}\mathbf{x} = \mathbf{x}^T \mathbf{A}^T \mathbf{a}_i = \left\langle \mathbf{x}, \mathbf{A}^T \mathbf{a}_i\right\rangle$, and (c) from the fact that $\mathbf{A}^T \mathbf{a}_i$ is the i^{th} column of $\mathbf{A}^T \mathbf{A}$ presented in (4), and $\mathbf{x}$ such as $x_{K+1} = 0$, and $x_i = 1$ for $i = 1, \cdots, K$. Second, for $i = K+1$, we have

$$\begin{aligned} \left|\langle \mathbf{a}_{K+1}, \mathbf{y}\rangle\right| &= \left|\langle \mathbf{a}_{K+1}, \mathbf{A}\mathbf{x}\rangle\right| \\ &= \left|\left\langle \mathbf{A}^T \mathbf{a}_{K+1}, \mathbf{x}\right\rangle\right| \\ &= \frac{1}{\sqrt{K}}. \end{aligned} \tag{6}$$

Obviously, the OMP algorithm must fail in the first iteration if an inequality $\left|\langle \mathbf{a}_{K+1}, \mathbf{y}\rangle\right| \geq \left|\langle \mathbf{a}_i, \mathbf{y}\rangle\right|$ for all $i \in \{1, \cdots, K\}$. The inequity becomes

$$\frac{1}{\sqrt{K}} \geq 1 - \frac{K-1}{K\sqrt{K}}$$

which is always true if $K = 2$. Thus, the OMP algorithm in the first iteration selects an incorrect index.

V. Proof of Theorem 1

A. Notations

The below notations will be used throughout the rest of this presentation. $\mathcal{T} = \text{supp}(\mathbf{x}) := \{i | x_i \neq 0\}$ is the set of indices corresponding to non-zero coefficients of $\mathbf{x}$. $|\mathcal{T}|$ is the cardinality of $\mathcal{T}$, and $\mathcal{T} \setminus \mathcal{I}$ is the set of elements belonging to $\mathcal{T}$ but not to $\mathcal{I}$. $\mathbf{A}_{\mathcal{T}} \in \Re^{M \times |\mathcal{T}|}$ is a sub-matrix of $\mathbf{A}$ which contains columns

corresponding to indices of $\mathcal{T}$. $\mathbf{x}_{\mathcal{T}} \in \Re^{|\mathcal{T}|}$ is a restriction of $\mathbf{x}$ to the elements indexed by $\mathcal{T}$. $span(\mathbf{A}_{\mathcal{T}})$ is the span of columns in $\mathbf{A}_{\mathcal{T}}$, $\mathbf{A}_{\mathcal{T}}^T$ is the transpose of $\mathbf{A}_{\mathcal{T}}$, and $\mathbf{A}_{\mathcal{T}}^\dagger = \left(\mathbf{A}_{\mathcal{T}}^T \mathbf{A}_{\mathcal{T}}\right)^{-1} \mathbf{A}_{\mathcal{T}}^T$ is the pseudo inverse of $\mathbf{A}_{\mathcal{T}}$. $\mathbf{P}_{\mathcal{T}} = \mathbf{A}_{\mathcal{T}} \mathbf{A}_{\mathcal{T}}^\dagger$ is the orthogonal projection onto $span(\mathbf{A}_{\mathcal{T}})$, and $\mathbf{P}_{\mathcal{T}}^\perp = \mathbf{I} - \mathbf{P}_{\mathcal{T}}$ is the orthogonal projection onto the orthogonal complement of $span(\mathbf{A}_{\mathcal{T}})$.

B. Lemmas

We need the below lemmas to prove Theorem 1.

Lemma 1: For a set $\mathcal{I}$, if $\delta_{|\mathcal{I}|} < 1$, then

$$\left(1 - \delta_{|\mathcal{I}|}\right) \|\mathbf{v}\|_2 \le \left\| \mathbf{A}_{\mathcal{I}}^T \mathbf{A}_{\mathcal{I}} \mathbf{v}_{\mathcal{I}} \right\|_2 \le \left(1 + \delta_{|\mathcal{I}|}\right) \|\mathbf{v}\|_2$$

holds for any $\mathbf{v}$ supported on $\mathcal{I}$.

Lemma 2: For disjoint sets $\mathcal{I}, \mathcal{J}$, if $\delta_{|\mathcal{I}|+|\mathcal{J}|} < 1$, then

$$\left\| \mathbf{A}_{\mathcal{I}}^T \mathbf{A} \mathbf{v} \right\|_2 = \left\| \mathbf{A}_{\mathcal{I}}^T \mathbf{A}_{\mathcal{J}} \mathbf{v}_{\mathcal{J}} \right\|_2 \le \delta_{|\mathcal{I}|+|\mathcal{J}|} \|\mathbf{v}\|_2$$

holds for any $\mathbf{v}$ supported on $\mathcal{J}$.

Lemma 3: If the sensing matrix satisfies the RIP of both orders K_1 and K_2, then $\delta_{K_1} \le \delta_{K_2}$ for any $K_1 \le K_2$

All proofs of the above lemmas are given in [3].

C. Proof of Theorem 1

1) We provide a condition under which the OMP algorithm selects a correct index in the first iteration. 2) We show that the residual in the general iteration preservers the sparsity of a K sparse signal. 3) The condition for the first iteration can be extended to the general iteration. 4) Theorem 1 is established from the conditions. The statements are an overall strategy of Proof of Theorem1.

First, we need investigate the condition when the OMP algorithm selects a correct index in the first iteration. Let us denote t^k be the index of the column maximally correlated with the residual $\mathbf{r}^{k-1}$. In the first iteration, we have

$$t^1 = \arg\max_i \left| \left\langle \mathbf{a}_i, \mathbf{r}^0 \right\rangle \right| = \arg\max_i \left| \left\langle \mathbf{a}_i, \mathbf{y} \right\rangle \right|. \quad (7)$$

Now, let us suppose that t^1 always belong to the support set $\mathcal{I}$ of $\mathbf{x}$. From (7), we have

$$\begin{aligned}\left\|\left\langle \mathbf{a}_{t^1}, \mathbf{y}\right\rangle\right\| &= \left\|\mathbf{A}_{\mathcal{I}}^T \mathbf{y}\right\|_\infty \\ &\overset{(a)}{\geq} \frac{1}{\sqrt{K}}\left\|\mathbf{A}_{\mathcal{I}}^T \mathbf{y}\right\|_2 \\ &\overset{(b)}{\geq} \frac{1}{\sqrt{K}}\left(1-\delta_K\right)\left\|\mathbf{x}_{\mathcal{I}}\right\|_2,\end{aligned} \tag{8}$$

where (a) from the norm inequalities, and (b) from the fact that $y = \mathbf{A}_{\mathcal{I}}\mathbf{x}_{\mathcal{I}}$ and Lemma 1.Suppose that t^1 does not belong to the support set $\mathcal{I}$, then

$$\begin{aligned}\left\|\left\langle \mathbf{a}_{t^1}, \mathbf{y}\right\rangle\right\| &= \left\|\mathbf{a}_{t^1}^T \mathbf{A}_{\mathcal{I}}\mathbf{x}_{\mathcal{I}}\right\| \\ &\overset{(a)}{\leq} \left(1-\delta_{K+1}\right)\left\|\mathbf{x}_{\mathcal{I}}\right\|_2,\end{aligned} \tag{9}$$

where (a) from Lemma 2. Clearly, t^1 must belong to the support set $\mathcal{I}$. Thus, if

$$\frac{1}{\sqrt{K}}\left(1-\delta_K\right)\left\|\mathbf{x}_{\mathcal{I}}\right\|_2 > \left(1-\delta_{K+1}\right)\left\|\mathbf{x}_{\mathcal{I}}\right\|_2 \tag{10}$$

then, the OMP algorithm selects a correct index in the first iteration. The equation (10) becomes $\sqrt{K}\delta_{K+1} + \delta_K < 1$. From Lemma 3, the inequality becomes $\sqrt{K}\delta_{K+1} + \delta_{K+1} < 1$ which leads to

$$\delta_{K+1} < \frac{1}{\sqrt{K}+1} \tag{11}$$

In short, if (11) is true, then the OMP algorithm always selects a correct index in the first iteration.

Now, we investigate a condition such that the OMP algorithm selects a correct index in the $(k+1)^{\text{th}}$ iteration. Let us suppose that initial k iterations of the OMP algorithm are successful. Namely, $\mathcal{T}^k = \left\{t^1, \cdots, t^k\right\} \in \mathcal{I}$. Then, $\mathbf{r}^k = \mathbf{y} - \mathbf{A}_{\mathcal{T}^k}\hat{\mathbf{x}}_{\mathcal{T}^k} \in span\left(\mathbf{A}_{\mathcal{I}}\right)$ because $\mathbf{y} = \mathbf{A}_{\mathcal{I}}\mathbf{x}_{\mathcal{I}}$ and $\mathbf{A}_{\mathcal{T}^k}$ is a sub-matrix of $\mathbf{A}_{\mathcal{I}}$. Thus, $\mathbf{r}^k$ can be expressed as $\mathbf{r}^k = \mathbf{A}\mathbf{x}^k$ (i.e., $\mathbf{r}^k$ is a linear combination of the K columns of $\mathbf{A}_{\mathcal{I}}$), where the support set of $\mathbf{x}^k$ belongs to the support set of $\mathbf{x}$. If the OMP algorithm selects a correct index belonging to the support set of $\mathbf{x}^k$, then the OMP algorithm also selects a correct index belonging to the support set of $\mathbf{x}$. Clearly, if $\sqrt{K}\delta_{K+1} + \delta_{K+1} < 1$ is satisfied, then the OMP algorithm success in the $(k+1)^{\text{th}}$ iteration.

Last, we need to show that the index t^{k+1} selected at the $(k+1)^{\text{th}}$ iteration of the OMP algorithm does not belong to T^k. First, we have $\hat{\mathbf{x}}_{\mathcal{T}^k} = \mathbf{A}_{\mathcal{T}^k}^{\dagger}\mathbf{y}$, and $\mathbf{r}^k = \mathbf{y} - \mathbf{A}_{\mathcal{T}^k}\hat{\mathbf{x}}_{\mathcal{T}^k} = \mathbf{P}_{\mathcal{T}^k}^{\perp}\mathbf{y}$. Second, for all $i \in \mathcal{T}^k$, we have

$$\begin{aligned}\left\langle \mathbf{a}_i, \mathbf{r}^k \right\rangle &= \left\langle \mathbf{a}_i, \mathbf{y} - \mathbf{A}_{\mathcal{T}^k} \hat{\mathbf{x}}_{\mathcal{T}^k} \right\rangle \\ &= \left\langle \mathbf{a}_i, \mathbf{y} \right\rangle - \left\{ \mathbf{a}_i, \mathbf{A}_{\mathcal{T}^k} \hat{\mathbf{x}}_{\mathcal{T}^k} \right\} \\ &= \mathbf{a}_i^T A_{\mathcal{I}} x_{\mathcal{I}} - \mathbf{a}_i^T \mathbf{A}_{\mathcal{T}^k} \mathbf{A}_{\mathcal{T}^k}^{\dagger} \mathbf{y} \\ &= 0.\end{aligned}$$

Therefore, we conclude that $\mathbf{r}^k$ is orthogonal to the columns $\mathbf{a}_i$ for all $i \in T^k$. It leads to $t^{k+1} \notin \mathcal{T}^k$. Furthermore, if $\mathbf{r}^k \neq \mathbf{0}$ and $\mathbf{r}^k \in span\left(\mathbf{A}_{\mathcal{I}}\right)$, then there exists $i \in \mathcal{I}$ such as $\left\langle \mathbf{a}_i, \mathbf{r}^k \right\rangle \neq 0$. Therefore, the OMP algorithm selects $i \in \mathcal{I} \setminus \mathcal{T}^k$.

Now, we apply the mathematical induction. First, we proved that the OMP algorithm selects a correct index if $\delta_{K+1} < \frac{1}{\sqrt{K}+1}$. Second, when we assume that the initial k iterations of the OMP algorithm are successful, the OMP algorithm selects a correct index in the $(k+1)^{\text{th}}$ iteration if $\delta_{K+1} < \frac{1}{\sqrt{K}+1}$. Thus, the OMP algorithm will terminate after the K^{th} iteration if $\delta_{K+1} < \frac{1}{\sqrt{K}+1}$.

VI. Discussion on Theorem 1

It is hard for us to determine δ_{K+1} from a sensing matrix because we need to examine all possible *K* sparse signal. However, the below result is known

Result [4]: If an $M \times N$ sensing matrix $\mathbf{A}$ whose entries are i.i.d. $\mathcal{N}(0,1/M)$, then $\mathbf{A}$ obeys the RIP condition $\delta_K \leq \varepsilon$ with high probability under

$$M \geq \frac{\rho K \log\left(\frac{N}{K}\right)}{\varepsilon^2} \tag{12}$$

where ρ is a positive constant. When we utilize the above inequalities, we indirectly compare the result obtained by [2].

	A sufficient condition	A sufficient condition on M
[1]	$\delta_{K+1} < 1/\left(3\sqrt{K}\right)$	$M \geq \rho 9K(K+1)\log\left(\frac{N}{K+1}\right)$

The paper	$\delta_{K+1} < 1/\left(\sqrt{K}+1\right)$	$M \geq \rho(K+1)\left(\sqrt{K}+1\right)^2 \log\frac{N}{K+1}$

Appendix

Example 1) computing all the Eigen values of $\begin{bmatrix} 1 & b & \cdots & b \\ b & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & b \\ b & \cdots & b & 1 \end{bmatrix}$.

$$\begin{vmatrix} 1-\lambda & b & b \\ b & 1-\lambda & b \\ b & b & 1-\lambda \end{vmatrix} = \begin{vmatrix} 1-\lambda & b & b \\ 0 & 1-\lambda-b & b-(1-\lambda) \\ b & b & 1-\lambda \end{vmatrix} = \begin{vmatrix} 1-\lambda-b & 0 & b-(1-\lambda) \\ 0 & 1-\lambda-b & b-(1-\lambda) \\ b & b & 1-\lambda \end{vmatrix}$$

$$= \begin{vmatrix} 1-\lambda-b & 0 & b-(1-\lambda) \\ 0 & 1-\lambda-b & b-(1-\lambda) \\ 0 & b & 1-\lambda+b \end{vmatrix} = \begin{vmatrix} 1-\lambda-b & 0 & b-(1-\lambda) \\ 0 & 1-\lambda-b & b-(1-\lambda) \\ 0 & 0 & 1-\lambda+2b \end{vmatrix}$$

$$= (1-\lambda-b)^2(1-\lambda+2b)$$

Therefore, $\lambda_1 = \lambda_2 = 1-b$, and $\lambda_3 = 1+2b$.

$$\begin{vmatrix} 1-\lambda & b & b & b \\ b & 1-\lambda & b & b \\ b & b & 1-\lambda & b \\ b & b & b & 1-\lambda \end{vmatrix} = \begin{vmatrix} 1-\lambda & b & b & b \\ 0 & 1-\lambda-b & b-(1-\lambda) & 0 \\ b & b & 1-\lambda & b \\ b & b & b & 1-\lambda \end{vmatrix} = \begin{vmatrix} 1-\lambda & b & b & b \\ 0 & 1-\lambda-b & b-(1-\lambda) & 0 \\ 0 & 0 & 1-\lambda-b & b-(1-\lambda) \\ b & b & b & 1-\lambda \end{vmatrix}$$

$$= \begin{vmatrix} 1-\lambda-b & 0 & 0 & b-(1-\lambda) \\ 0 & 1-\lambda-b & b-(1-\lambda) & 0 \\ 0 & 0 & 1-\lambda-b & b-(1-\lambda) \\ b & b & b & 1-\lambda \end{vmatrix} = \begin{vmatrix} 1-\lambda-b & 0 & 0 & b-(1-\lambda) \\ 0 & 1-\lambda-b & b-(1-\lambda) & 0 \\ 0 & 0 & 1-\lambda-b & b-(1-\lambda) \\ 0 & b & b & 1-\lambda+b \end{vmatrix}$$

$$= \begin{vmatrix} 1-\lambda-b & 0 & 0 & b-(1-\lambda) \\ 0 & 1-\lambda-b & b-(1-\lambda) & 0 \\ 0 & 0 & 1-\lambda-b & b-(1-\lambda) \\ 0 & 0 & 2b & 1-\lambda+b \end{vmatrix} = \begin{vmatrix} 1-\lambda-b & 0 & 0 & b-(1-\lambda) \\ 0 & 1-\lambda-b & b-(1-\lambda) & 0 \\ 0 & 0 & 1-\lambda-b & b-(1-\lambda) \\ 0 & 0 & 0 & 1-\lambda+3b \end{vmatrix}$$

$$= (1-\lambda-b)^3(1-\lambda+3b)$$

Therefore, $\lambda_1 = \lambda_2 = \lambda_3 = 1-b$, and $\lambda_4 = 1+3b$

Thus, we concluded all the Eigen values of a $(K+1)\times(K+1)$ $\begin{bmatrix} 1 & b & \cdots & b \\ b & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & b \\ b & \cdots & b & 1 \end{bmatrix}$ are

$\lambda_1 = \cdots = \lambda_K = 1-b,$ and $\lambda_{K+1} = 1+Kb$.

Reference

[1] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” IEEE Trans. Inf. Theory, vol. 53, pp. 4655 – 4666, Dec. 2007.
[2] M. A. Davenport and M. B. Wakin, “Analysis of Orthogonal Matching Pursuit using the restricted i sometry property,” IEEE Trans. Inf. Theory, vol. 56, pp. 589 – 592, 2008.
[3] W. Dai and O. Milenkovic, “Subspace pursuit for compressive sensing signal reconstruction,” IEEE Trans. Inf. Theory, vol. 55, no. 5, pp. 2230 – 2249, May 2009.
[4] R. Baraniuk, M. Davenport, R. De Vore, and M. Wakin, “A simple proof of the restricted isometry property for random matrices,” Construct. Approx., vol. 28, no. 3, pp. 253– 263, 2008.

Capacity of OFDM Systems over Fading Underwater Acoustic Channels

Authors: Chantri Polprasert, *Member, IEEE*, James A. Ritcey, *Fellow, IEEE*, and Milica Stojanovic, *Fellow, IEEE*
Publication: IEEE Journal of Oceanic Engineering, vol. 36, no. 4, Oct. 2011
Speaker: Zafar Iqbal

Short Summary:

This paper derives the upper and lower bounds for channel capacity of the OFDM systems over underwater acoustic channels as a function of distance between the transmitter and the receiver. The upper bound is obtained using perfect CSI at the receiver while the lower bound is obtained by assuming that the input is drawn from a PSK constellation which results in non-Gaussian distribution of the output signal and no CSI. It incorporates frequency dependent path loss at each arrival path at the receiver due to acoustic propagation. This leads the UW channel to be modeled as wide sense stationary and correlated scattering (WSS-non-US) fading channel. Results from both Rayleigh and Rician fading show a gap between the upper and lower bounds which depends, not only on the ranges and shape of the scattering function of the UW channel but also on the distance between the transmitter and the receiver.

I. INTRODUCTION

Recently, OFDM has been applied to the UWA communications and yields high data rate with strong bit error rate performance [2-5].

Time and frequency spreading are the main challenges for data transmission through UW channels. Several attempts have been made to characterize the UW channel, most of which view the UW channel as a linear time-varying channel with wide sense stationary and uncorrelated scattering (WSSUS) [8-10]. However, this approach treats the entire frequency band as a whole and neglects the frequency dependent path loss. This model is acceptable for transmissions at low bandwidth (<10 kHz) [9].

Channel capacity over WSSUS fading channel has been studied [12-15] under these assumptions: 1) no CSI is available at the transmitter or receiver, and 2) peak power constraints. It is shown that channel capacity is achieved at capacity maximizing bandwidth, which depends

on the ranges and shape of the scattering function of the fading channel. These studies are conducted over wireless fading channels which assume constant power spectral density (PSD) and AWGN noise. There has been some research on the capacity of UW channels [6, 16, 17] but all assume no fading in their UW channels.

This paper investigates the capacity of OFDM systems over the UW fading channels with no CSI at the transmitter or the receiver. The UW channel is modeled by taking into account frequency-dependent path loss. This invalidates the assumption of stationarity in frequency of the WSSUS model and leads to a frequency-dependent doubly spread (DS) fading channel characterized by the WSS-non-US [18] assumptions. The conventional WSSUS model is uncorrelated in both delay and Doppler domains but the proposed model is uncorrelated in the Doppler domain and correlated in the delay domain.

Using this channel model and assuming that the acoustic propagation and ambient noise PSD are available at both the transmitter and receiver, capacity upper and lower bounds are derived. Capacity upper bound is derived by assuming perfect CSI at the receiver, while lower bound is obtained by the mutual information rate whose input is an i.i.d. random variable and is drawn from a PSK modulation [12,19], which results in a non-Gaussian distribution of the output signal. Results are obtained for both Rayleigh and Rician fading of the UW channel. Simulation results show a gap between the upper and lower bounds which depends not only on the ranges and shape of the scattering function of the UW channel, but also on the distance between the transmitter and receiver. Results are confirmed with the scattering function obtained from the 2008 rescheduled Acoustic Communications Experiment (RACE08) experimental data.

II. OFDM System and UW Channel Model

In this section, an OFDM system model for UW acoustic communications is developed. Physical and statistical properties of the channel as well as PSD of the ambient noise are investigated and a frequency-dependent UW DS fading channel has been proposed.

A. OFDM System

A conventional CP-OFDM system is considered as shown in Fig. 1. Let $\mathbf{X}_n = \left[X_{n,0} ... X_{n,K-1} \right]^T$ and $\mathbf{Y}_n = \left[Y_{n,0} ... Y_{n,K-1} \right]^T$ be the sent and received block of data at the *n*th OFDM symbol duration, respectively.

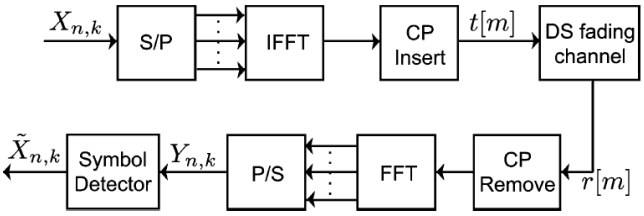


Fig. 1. System model

Assuming the guard interval L_{cp} is longer than the channel length L to avoid the interblock interference (IBI), the input/output relationship can be written as,

$$Y_{n,k} = G_{n,k}(d) X_{n,k} + N_{n,k} \tag{1}$$

where $k \in [0,...,K-1]$ is the subcarrier index and $n \in [0,...,N-1]$, while d is the distance between transmitter and receiver. $G_{n,k}(d)$ denotes the channel transfer function at the kth subcarrier. $N_{n,k}$ is the ambient noise in the ocean. This simplifies the fading effect into multiplicative coefficient, which is the basis for analysis of the UW channel in this paper. The impact of ICI is assumed to be negligible through appropriate parameter settings (Justified in App. I). For simplicity, the overall system input/output of the entire N OFDM transmissions is characterized by a vector of size $NK \times 1$, as follows.

$$\mathbf{Y} = \text{diag}(\mathbf{X})\mathbf{G}(d) + \mathbf{N} = \text{diag}(\mathbf{G}(d))\mathbf{X} + \mathbf{N} \tag{2}$$

where

$$\mathbf{Y} = \left[\mathbf{Y}_0^T ... \mathbf{Y}_{N-1}^T \right]^T \text{ and } \mathbf{Y}_n = \left[Y_{n,0} ... Y_{n,K-1} \right]^T \tag{3}$$

$$\mathbf{X} = \left[\mathbf{X}_0^T ... \mathbf{X}_{N-1}^T \right]^T \text{ and } \mathbf{X}_n = \left[X_{n,0} ... X_{n,K-1} \right]^T \tag{4}$$

$$\mathbf{N} = \left[\mathbf{N}_0^T ... \mathbf{N}_{N-1}^T \right]^T \text{ and } \mathbf{N}_n = \left[N_{n,0} ... N_{n,K-1} \right]^T \tag{5}$$

$$\mathbf{G}(d) = \left[\mathbf{G}_0^T(d) ... \mathbf{G}_{N-1}^T(d) \right]^T \text{ and } \mathbf{G}_n(d) = \left[G_{n,0}(d) ... G_{n,K-1}(d) \right]^T \tag{6}$$

B. Characterization of Approximate DS Fading Channels

UW channel is modeled using both the physical property, which is the attenuation depending on the propagation distance and bandwidth of the transmitted signal, and the statistical property for which the channel is usually assumed WSSUS.

1) Frequency-dependent Path Loss

For the signal propagated through UW medium, the attenuation or path loss, which is a function of distance and signal frequency, is a combination of geometric spreading and absorption, written as,

$$Q^2(d, f) = d^{-sp}(q^2(f))^{-d} \tag{7}$$

where d is the propagated distance in meter and f is the frequency in kilohertz. d^{-sp} represents the spreading loss and sp is the spreading factor which is set to 1.5. $q^2(f)$ is the absorption coefficient in seawater which is given by,

$$10\log(q^2(f)) = 2.49\times10^{-7} f^2 + 0.99\frac{f^2}{f^2+1.23\times10^4} + 1.48\times10^{-4}\frac{f^2}{f^2+1.522} \text{ dB/m} \tag{8}$$

Eq. (8) is calculated when the salinity S is 35 parts per thousand (ppt), gauge pressure P_a is 1 atm, temperature T=14 ^{o}C, and the relaxation frequency is 111 kHz.

2) Conventional Statistical Model

The CIR is modeled by a sum of several multipath components [9], [10]. Let $h(t, \tau)$ denote a continuous-time CIR of linear time-variant (LTV) UW channels and its corresponding transfer function $H(t, f)$ is,

$$h(t,\tau) = \sum_{i=0}^{I-1} h_i(t)\delta(\tau-\tau_i), \quad H(t, f) = \sum_{i=0}^{I-1} h_i(t)e^{-j2\pi f\tau_i} \tag{9}$$

where I is the number of arrival paths. WSSUS is commonly assumed to characterize the channel, i.e., $E\{h[t,\tau]h^*[t',\tau']\} = R_{h_c}(t-t',\tau)\delta(\tau-\tau')$ where $R_{h_c}(t-t',\tau)$ is the autocorrelation function of the delay τ between time t and t'. Its corresponding scattering function is $S_c(\tau, v) = \int R_{h_c}(\Delta t,\tau)\exp(-j2\pi\Delta tv)d\Delta t$ where $\tau \in [0, \tau_m]$. For a bandwidth of less than 10 kHz, let τ_m and f_d denote the maximum channel delay spread and 3-dB Doppler spread of $S_c(\tau,v)$, respectively.

3) Frequency-dependent DS Fading Channels

Conventionally, UW models use WSSUS properties to characterize LTV UW channels, assuming equal attenuation across all the signal bandwidth, treating the entire frequency band as

flat and neglecting frequency-dependent parameters of the individual arrival path. In reality, various factors from channel physics such as the attenuation, reflection loss, or tx/rx operating ranges influence frequency dependency on the path loss. In this paper, the impact of channel physics is limited to only the attenuation $Q^2(d_i, f)$ (7) where d_i is the propagation distance of ith delay path.

Let $\chi_{d_i}(\tau)$ denote a CIR of the ith delay path corresponding to $Q(d_i, f)$ i.e., $Q(d_i, f) = \int \chi_{d_i}(\tau)\exp(-j2\pi f\tau)d\tau$ where $Q^2(d_i, f) = Q(d_i, f)Q^*(d_i, f)$. Taking into account $\chi_{d_i}(\tau)$ yields a modified CIR, $g_d(t, \tau)$

$$g_d(t,\tau) = \sum_{i=0}^{I-1} h_i(t)\chi_{d_i}(\tau) \otimes \delta(\tau - \tau_i) \tag{10}$$

$$\begin{aligned} G_d(t, f) &= \int g_d(t,\tau)e^{-j2\pi\tau f}d\tau \\ &= \sum_{i=0}^{I-1} h_i(t)Q(d_i, f)e^{-j2\pi\tau_i f} \\ &\approx Q(d0, f)\sum_{i=0}^{I-1} h_i(t)e^{-j2\pi\tau_i f} \quad \because Q(d, f) \approx Q(d_{I-1}, f) \\ &= Q(d, f)H(t, f) \end{aligned} \tag{11}$$

d_0 is the distance between transmitter and receiver and the subscript of d is neglected for simplicity. Hence the modified CIR is

$$g_d(t,\tau) = \chi_d(\tau) \otimes \sum_{i=0}^{I-1} h_i(t)\delta(\tau - \tau_i) \tag{12}$$

From the sampling theorem, the T_s-spaced discrete time CIR is,

$$\begin{aligned} g_{0,d}[m, p_l] &= \int_{\tau} g_d(mT_s, \tau)\,\mathrm{sinc}(B\tau - p_l)d\tau \\ &\approx g_d(mT_s, p_lT_s) \quad \because B \text{ is large} \\ &= \chi_d[l] \otimes h_0[m, p_l] \end{aligned} \tag{13}$$

where B=1/T_s. From (13) the channel transfer function can be written as

$$
\begin{aligned}
G_{n,k}(d) &= \sum_{m'=0}^{K-1}\left(\frac{1}{K}\sum_{l=0}^{L-1} g_{0,d}^{n}\left[m'+L_{cp},l\right]\sum_{p=0}^{K-1} e^{j2\pi p(m'-l)/K}\right)\times e^{-j2\pi m'k/K} \\
&= \sum_{p=0}^{K-1}\sum_{l=0}^{L-1}\left(\frac{1}{K}\sum_{m'=0}^{K-1}\left(\chi_d[l]\otimes h_0^n\left[m'+L_{cp},l\right]\right)\times e^{j2\pi m'(p-k)/K}\right)e^{-j2\pi pl/K} \\
&\approx \sum_{p=0}^{K-1}\sum_{l=0}^{L-1} G_{n,l}^{d}[0]e^{-j2\pi kl/K} \\
&= Q(d,f_k)H_{n,k}
\end{aligned}
\tag{14}
$$

where $f_k = f_c + k/(KT_s)$ and f_c is the center frequency corresponding to the zeroth subcarrier. $g_{0,d}^{n}\left[m',l\right] = g_{0,d}\left[nN_s + p_0 + m', p_l\right]$ and $h^n[m',l] = h_0\left[nN_s + p_0 + m', p_l\right]$ where $N_s = L_{cp} + K$ is the OFDM symbol length and p_0 is the arrival time of the first arrival path. Eq. (14) is derived under negligible ICI. Moreover,

$$
\begin{aligned}
H_{n,k} &= \sum_{l=0}^{L-1}\left(\frac{1}{K}\sum_{m'=0}^{K-1} h_0^n\left[m'+L_{cp},l\right]\right)e^{-j2\pi lk/K} \\
&= \sum_{l=0}^{L-1} h_{n,l}[0]e^{-j2\pi lk/K}
\end{aligned}
\tag{15}
$$

$$
Q(d,f_k) = \sum_{l=0}^{L-1}\chi_d[l]e^{-j2\pi lk/K} \tag{16}
$$

and

$$
G_{n,l}^{d}[p-k] = \frac{1}{K}\sum_{m'=0}^{K-1}\left(\chi_d[l]\otimes h_0^n\left[m'+L_{cp},l\right]\right)e^{j2\pi m'(p-k)/K} \tag{17}
$$

$G_{n,k}(d)$ is the fading gain encountered by the signal transmitted on the kth subcarrier. $Q(d, f_k)$ is assumed constant within a subcarrier with center frequency f_k. $H_{n,k}[0]$ is the approximate CIR. Eq. (14) simplifies the transfer function of frequency-dependent UW DS channel into a multiplication of the attenuation $Q(d,f_k)$ and statistical part $H_{n,k}$ governed by the scattering function $S[l, \lambda]$. Assuming $S[l,\lambda] \approx S_c\left(lT_s, \lambda/T_b\right)$ when the variation of $h_0[m, p_l]$ within $T_b(T_b=N_sT_s)$ is negligible [32]. T_b is the OFDM symbol interval $\lambda\in$[-0.5,0.5]. Its range (L, λ_d) is related to (τ,f_d) of $S_c(\tau,v)$ through $L = \lceil \tau_m B \rceil$ and $\lambda_d=f_dT_b$. This leads $G_{n,k}(d)$ to be a WSS but non-US fading channel [18].

$$
E\left\{G_{n',k'}(d)G_{n,k}^{*}(d)\right\} = Q(d,f_k)Q^{*}(d,f_{k'})R_H\left[n-n',k-k'\right] \tag{18}
$$

where $R_H\left[n-n',k-k'\right] = E\left[H_{n,k}H_{n,k}^{*}\right]$. Compared to the conventional WSSUS model (uncorrelated in both delay and Doppler domains), the proposed model is still uncorrelated in

Doppler but correlated in the delay domain because of attenuation. To be consistent, a vector form for $G_{n,k}(d)$ from (14) is

$$\mathbf{G}_n(d) = \mathbf{Q}(d)\mathbf{H}_n \tag{19}$$

where $\mathbf{Q}(d) = \mathrm{diag}\left(\left[Q(d, f_0)...Q(d, f_{K-1})\right]\right)$ and $\mathbf{H}_n = \left[H_{n,0}...H_{n,K-1}\right]^T$ from (15). Fig 2 shows a realization of $\left|G_{n,k}(d)\right|^2$ when d=5 and 20 km. $H_{n,k}$ is assumed zero-mean complex Gaussian random variable with exponentially decaying PDP with 20-dB power difference between the first and last paths. Transmit bandwidth is 51.2 kHz. Channel delay length is 5 ms which corresponds to L=256. The number of subcarriers K is 512. We can see that the propagation distance and signal frequency have a significant impact on the realization of $\left|G_{n,k}(d)\right|^2$.

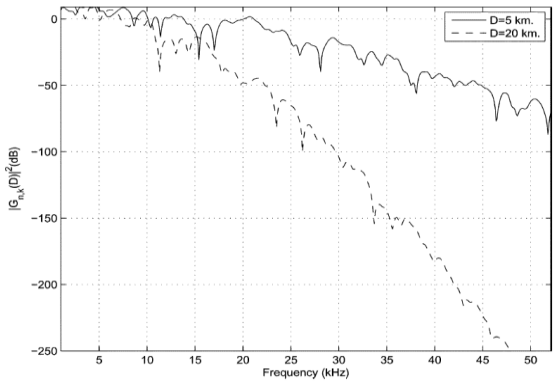


Fig. 2. Impact of attenuation on CIR

C. Ambient Noise

$N_{n,k}$ in (1) is assumed the ambient noise in the ocean which consists of four sources [6]: turbulence $A_t(f)$, shipping $A_s(f)$, waves $A_w(f)$, and thermal noise $A_{th}(f)$, described by Gaussian statistics with a continuous PSD in dBre/μPa per hertz,

$$\begin{aligned} A_t(f) &= 17 - 30\log f \\ A_s(f) &= 40 + 20(s - 0.5) + 26\log f - 60\log(f + 0.03) \\ A_w(f) &= 50 + 7.5\sqrt{w} + 20\log f - 40\log(f + 0.4) \\ A_{th}(f) &= -15 + 20\log f \end{aligned} \tag{20}$$

where f is the frequency in kilohertz, $s \in [0,1]$ is the shipping activity, w is the wind speed in meters per second, and overall noise PSD is

$$A(f) = 10\log\left(10^{A_t(f)/10} + 10^{A_s(f)/10} + 10^{A_w(f)/10} + 10^{A_{th}(f)/10}\right) \tag{21}$$

III. CAPACITY OF THE UW CHANNELS

The upper bound $U_c(d)$ and lower bound $L_c(d)$ are derived as a function of distance d between the transmitter and receiver. The capacity $C(d)$ is given in bits per second by

$$C(d) = \lim_{N\to\infty} \frac{1}{NT_b} \sup_{p(\mathbf{X})} I(\mathbf{Y};\mathbf{X}) \tag{22}$$

where the maximization is over the set $p(\mathbf{X})$ of all input distributions that satisfy a given average-power constraint. $U_c(d)$ is obtained when the input vector follows a join complex Gaussian distribution. $L_c(d)$ is obtained under imperfect CSI whose reduction from $U_c(d)$ comes from limited mutual information from PSK constellation and the MMSE prediction error related to channel uncertainty [12], [14]. This bounding technique is used in [12] for wireless fading channels while this paper uses it for UW channels. The bounds are derived under the following assumptions:

- Information of attenuation (7) and ambient noise PSD (21) of UW channels are available at both the transmitter and receiver.
- For statistical part [$H_{n,k}$ (15)] of UW channels, its approximate CIR $h_{n,l}[0]$ is assumed a WSSUS random process with variance σ_l^2 where $\sum_l \sigma_l^2 = E\left[\left|H_{n,k}\right|^2\right] = 1$. Rayleigh and Rician fading are also considered. A scattering function which characterizes $H_{n,k}$ is available at the receiver.
- The noise vector $\mathbf{N} \sim \mathcal{CN}\left(\mathbf{0}, diag(\mathbf{A})\right)$. Where $\mathbf{A} = \left[\mathbf{A}_0 ... \mathbf{A}_{N-1}\right]^T$ and $\mathbf{A}_n = \left[A(f_0), ..., A(f_{K-1})\right]^T$.
- The impact of ICI is negligible compared to $A(f_k)$.

Let F denote the subcarrier spacing and $B=KF$, the signal bandwidth. P is the signal transmit power in dBre/μPa.

A. Upper Bound $U_c(d)$

To bound $\sup_{p(\mathbf{X})} I(\mathbf{Y};\mathbf{X})$, we use the chain rule $I(\mathbf{Y};\mathbf{X}) = I(\mathbf{Y};\mathbf{X},\mathbf{G}(d)) - I(\mathbf{Y};\mathbf{G}(d)\,|\,\mathbf{X})$. The output vector $\mathbf{Y}$ depends on the input vector $\mathbf{X}$ through $\mathbf{b}$=daig($\mathbf{X}$)$\mathbf{G}(d)$, so $I(\mathbf{Y};\mathbf{X},\mathbf{G}(d))=I(\mathbf{Y};\mathbf{b})$. The upper bound of I(Y;b) is achieved when the input $\mathbf{b} \sim \mathcal{CN}(\mathbf{0}, \mathbf{I} + \mathbf{R}_\mathbf{X}(d) \odot \mathbf{R}_\mathbf{G}(d))$. Where

$\mathbf{R}_{\mathbf{G}}(d) = E[\mathbf{G}(d)\mathbf{G}^H(d)]$. $\mathbf{R}_{\mathbf{X}}(d) = \text{diag}\left(\left[\mathbf{R}_{\mathbf{X}_0}(d),...,\mathbf{R}_{\mathbf{X}_{N-1}}(d)\right]\right)$ where

$\mathbf{R}_{\mathbf{X}_n}(d) = \text{diag}\left(\left[\sigma_x^2(d, f_0)...\sigma_x^2(d, f_{K-1})\right]\right)$ and $\sigma_x^2(d, f_k) = E\left[\left|X_{n,k}\right|^2\right]$.

The upper bound $U_c(d)$ is [27],

$$\begin{aligned} C(d) &\le \lim_{N\to\infty} \frac{1}{NT_b} \sup_{\mathbf{R}_{\mathbf{X}}(d)} \log\det\left(\mathbf{I} + \left(\mathbf{R}_{\mathbf{X}}(d) \odot \mathbf{R}_{\mathbf{G}}(d)\right)\text{diag}(\mathbf{A})^{-1}\right) \\ &\le \lim_{N\to\infty} \frac{1}{NT_b} \sup_{\mathbf{R}_{\mathbf{X}}(d)} \sum_{n=0}^{N-1}\sum_{k=0}^{K-1} \log\left(1 + E\left[\left|X_{n,k}\right|^2\right]\frac{Q^2(d, f_k)}{A(f_k)}\right) \\ &= \frac{1}{T_b}\sum_{k=0}^{K-1} \sup_{\sigma_x^2(d,f_k)} \log\left(1 + \sigma_x^2(d, f_k)\frac{Q^2(d, f_k)}{A(f_k)}\right) \\ &= U_c(d) \end{aligned} \tag{23}$$

where the inequality follows from Hadamard's inequality [11]. This result is similar to [6] which is the capacity of time-invariant UW channels but is scaled by a factor of FT_b which is greater than 1 to avoid IBI. $\sigma_x^2(d, f_k)$ is subject to the source power constraint

$$F\sum_{k=0}^{K-1}\sigma_x^2(d, f_k) = P \tag{24}$$

U_c is obtained when energy allocation across all subcarriers satisfies

$$\sigma_x^2(d, f_k) = \begin{cases} \max\left(Th - \left(\frac{A(f_k)}{Q(d,f_k)^2}\right), 0\right) & f_k \in B \\ 0 & otherwise \end{cases} \tag{25}$$

where Th is chosen so that (24) is satisfied according to the water-filling algorithm [11].

B. Lower Bound $L_c(d)$ over Rayleigh Fading Channels

For lower bound, channel fading statistics are assumed available at the receiver, not the transmitter. Our results show, for the first time, that decrease in $L_c(d)$ depends not only on the channel variations but also on the propagation distance d between the transmitter and receiver.

Consider $I(\mathbf{Y};\mathbf{X})$ where each entry of $\mathbf{X}$, $X_{n,k}$ is an i.i.d r.v. drawn from PSK modulation whose amplitude $\left|X_{n,k}\right| = \sigma_x$ and phase $\angle X_{n,k}$ has a uniform discrete distribution across a circle. $I(\mathbf{Y};\mathbf{X})$ can be written as,

$$\begin{aligned} I(\mathbf{Y};\mathbf{X}) &= I\left(\mathbf{Y};\mathbf{X},\mathbf{G}(d)\right) - I\left(\mathbf{Y};\mathbf{G}(d)\,|\,\mathbf{X}\right) \\ &\ge I\left(\mathbf{Y};\mathbf{X}\,|\,\mathbf{G}(d)\right) - I\left(\mathbf{Y};\mathbf{G}(d)\,|\,\mathbf{X}\right) \end{aligned} \tag{26}$$

The inequality is due to the non-negativity property of mutual information. Exact calculation of the mutual information is infeasible due the non-Gaussian distribution of **Y** [28]. Note that [29]

$$I(\mathbf{Y};\mathbf{X}\,|\,\mathbf{G}(d)) = N\ I(\mathbf{Y}_N;\mathbf{X}_N\,|\,\mathbf{G}_N(d)) \tag{27}$$

where $\mathbf{G}_N(d)=\mathbf{G}_n(d)$, $\mathbf{X}_N=\mathbf{X}_n$, and $\mathbf{Y}_N=\mathbf{Y}_n$ since the input $X_{n,k}$ has an i.i.d. distribution and every block of the channel coefficients $\mathbf{G}_n(d)$ has the same distribution. $\sigma_x^2(d, f_k)$ is set according to (25) under constraint (24) and apply it to $I(\mathbf{Y}_N;\mathbf{X}_N\,|\,\mathbf{G}_N(d))$. This water-filling policy is suboptimal for PSK constellation [30]. $I(\mathbf{Y};\mathbf{G}(d)|\mathbf{X})$ is calculated in App. II which yields

$$I(\mathbf{Y};\mathbf{G}(d)\,|\,\mathbf{X}) = \sum_{n=0}^{N-1}\log\det\left(\mathbf{I}+\mathbf{B}_n(d)\,\mathrm{diag}\left(\mathbf{S}(d)\right)\right) \tag{28}$$

where $\mathbf{S}(d)$ is the $K\times 1$ vector whose kth entry is $\sigma_x^2(d, f_k)Q^2(d, f_k)/A(f_k)$. $\mathbf{B}_n(d)$ is the linear MMSE prediction error matrix which depends on both the transmission distance d and channel variation $R_H[m,k]$. Substituting (27) and (28) into (26), the mutual information is

$$I(\mathbf{X};\mathbf{Y}) \ge N\ I\left(\mathbf{Y}_N;\mathbf{X}_N\,|\,\mathbf{G}_N(d)\right) - \sum_{n=0}^{N-1}\log\det\left(\mathbf{I}+\mathbf{B}_n(d)\,\mathrm{diag}\left(\mathbf{S}(d)\right)\right) \tag{29}$$

Finally, the lower bound $L_c(d)$ of the capacity $C(d)$ can be written as,

$$\begin{aligned} C(d) &\ge \lim_{N\to\infty}\frac{1}{NT_b}I(\mathbf{Y};\mathbf{X}) \\ &\ge \frac{1}{T_b}I\left(\mathbf{Y}_\infty;\mathbf{X}_\infty\,|\,\mathbf{G}_\infty(d)\right) - \frac{1}{T_b}\log\det\left(\mathbf{I}+\mathbf{B}_\infty(d)\,\mathrm{diag}\left(\mathbf{S}(d)\right)\right) \\ &= L_c(d) \end{aligned} \tag{30}$$

where $\mathbf{B}_\infty(d)$ is calculated given infinite past channel symbols. From (30), unlike [12] and [19], channel scattering function is not explicit but lies within $\mathbf{B}_\infty(d)$.

C. Lower Bound $L_c(d)$ over Rician Fading Channels

Let ρ denote a Rician fading parameter which is the ratio of the fixed to a scatter part. ρ is assumed independent of the transmission distance d and identical for every delay path. The approximate CIR $h_{n,l}[0]$ of the lth path is modeled as

$$h_{n,l}[0] = \sigma_l\left(A_l e^{j\phi_l} + s_{n,l}\right) \tag{31}$$

$$\text{where } \rho = \frac{A_l^2}{E\left[\left|s_{n,l}\right|^2\right]} \qquad \text{and } A_l^2 = \frac{\rho}{\rho+1}, E\left[\left|s_{n,l}\right|^2\right] = \frac{1}{\rho+1} \tag{32}$$

$E\left[\left|h_{n,l}[0]\right|^2\right] = \sigma_l^2$. ϕ_l is assumed uniformly distributed from –π to π and uncorrelated across different delay paths. For $h(\mathbf{Y})$, using (31), $H_{n,k}$ is

$$H_{n,k} = \sum_{l=0}^{L-1} \sigma_l A_l e^{j\phi_l} e^{-j2\pi lk/K} + \sum_{l=0}^{L-1} \sigma_l s_{n,l} e^{-j2\pi lk/K} \tag{33}$$

From (33), sum of scatter part follows $\mathcal{CN}\left(0,1/(\rho+1)\right)$. This causes $H_{n,k} \sim \mathcal{CN}\left(D_k,1/(\rho+1)\right)$ where $Dk = \sum_{l=0}^{L-1} \sigma_l A_l e^{j\phi_l} e^{-j2\pi lk/K}$.

For $h(\mathbf{Y}|\mathbf{X})$, we assume that the receiver can successfully track the fixed part $A_l e^{j\phi_l}$ and the autocorrelation function of the approximate CIR is

$$E\left[h_{n,l}[0]h_{n',l'}^*[0]\right] = \left(A_l^2 + R_s\left[n-n',l\right]\right)\sigma_l^2\delta\left[l-l'\right] \tag{34}$$

Where $R_s\left[n-n',l\right] = E\left[s_{n,l}s_{n',l}^*\right]$. Apply (34) to calculate $\mathbf{B}_n(d)$ and obtain $h(\mathbf{Y}|\mathbf{X})$.

IV. Simulation results

The UW fading channel is modeled by two parts, attenuation and statistical as explained earlier. The delay profile is assumed exponentially decaying whose maximum delay spread τ_m is set where the first and last arrival paths have 10-dB power difference. The range of the Doppler profile scattering function is determined by f_d, the 3-dB bandwidth of the frequency response. For $A(f_k)$, the shipping activity s = 0.5 and wind speed w = 10 m/s. OFDM symbols are transmitter at frequency beyond 1 KHz. Energy allocation across transmit bandwidth $B_c(d)$ is implemented using (25) subject to power constraint (24). P=145 dBre/μPa and Rayleigh fading is assumed unless stated otherwise.

1) Limitations due to the ICI

Because of the attenuation, the variance of ICI is frequency dependent. This model assumes the ICI variance is negligible compared to that of the ambient noise. In simulation, the ICI variance is limited to at least 3 dB lower than ambient noise variance. The ICI variance depends

on attenuation, $\sigma_x^2(f_k)$, and shape of the scattering function. Two scattering functions, AR-1 and uniform scattering are considered whose 3 dB bandwidth is equal to λ_d. Let $S_1[l, \lambda]$ and $S_2[l, \lambda]$ denote these scattering functions of $h_{n,l}[0]$, respectively, given by

$$S_1[l,\lambda] = \frac{\sigma_l^2}{\left|1-\alpha_l e^{-j2\pi\lambda}\right|^2},\ \lambda \in [-0.5, 0.5]$$
$$S_2[l,\lambda] = \begin{cases} \frac{\sigma_l^2}{2\lambda_d}, & |\lambda| \le \lambda_d \\ 0, & \lambda_d < |\lambda| \le 0.5 \end{cases} \tag{35}$$

These scattering functions are assumed unchanged over the transmission ranges of interest. Fig. 3 displays variance of the ICI at their widest spread of both scattering functions when d=5 km such that its variance is at least 3 dB lower than that of the ambient noise for most of the transmission bandwidth. For the AR-1 model, τ_m =1 ms and f_d =1 Hz. For the uniform model, τ_m=5 ms and f_d =7 Hz. We notice that the 3-dB gap is violated when signal bandwidth is greater than 31 kHz. These account for only 0.39% of the total signal energy and have negligible impact on the capacity as justified in Appendix I.

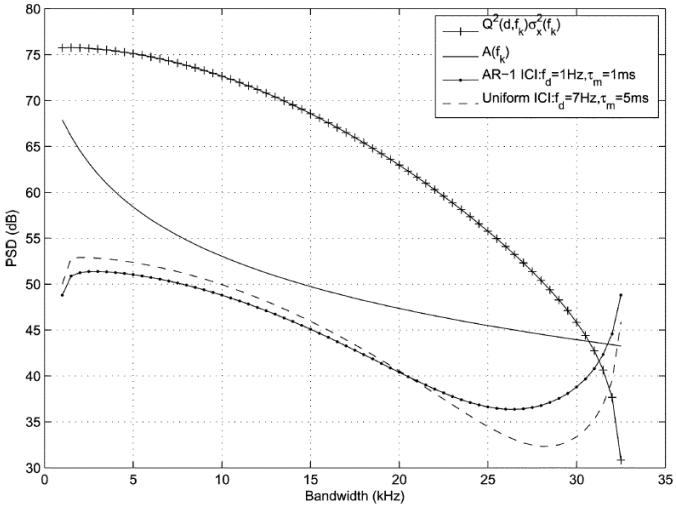


Fig. 3. PSD of the received signal, ambient noise, and the ICI variance at d=5 km

2) Impact of Signal Bandwidth

From Fig. 4, we can see that both $U_c(d)$ and $L_c(d)$ increase as a function of signal bandwidth B and remain fixed when B is greater than a certain value. We define this value as the capacity-maximizing bandwidth $B_c(d)$ which is a signal bandwidth that maximizes both $L_c(d)$ and $U_c(d)$. The gap beyond $B_c(d)$ is rather wide due to the limited mutual information that can be conveyed by the PSK constellation.

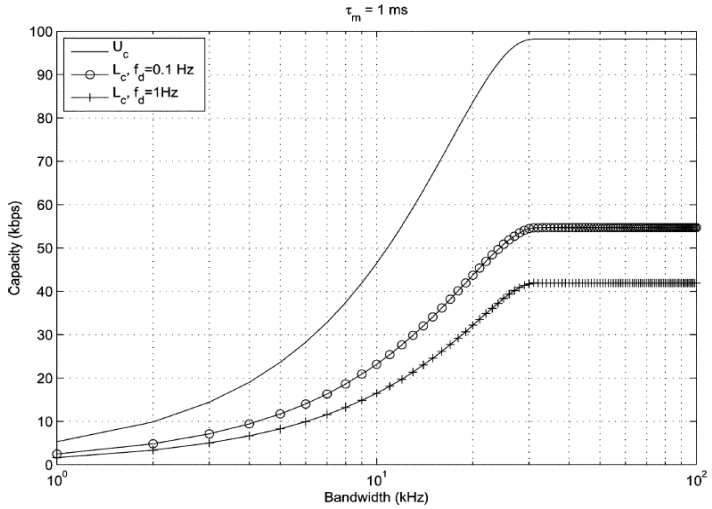


Fig. 4. $U_c(d)$ and $L_c(d)$ versus bandwidth for AR-1 model at d=5km

3) Impact of Ranges and Shape of the Scattering Function

Figs. 5 and 6 show the impact of the ranges of $(f_{d,}\tau_m)$ on $L_c(d)$ over the distance for $S_1[l, \lambda]$ and $S_2[l, \lambda]$, respectively.

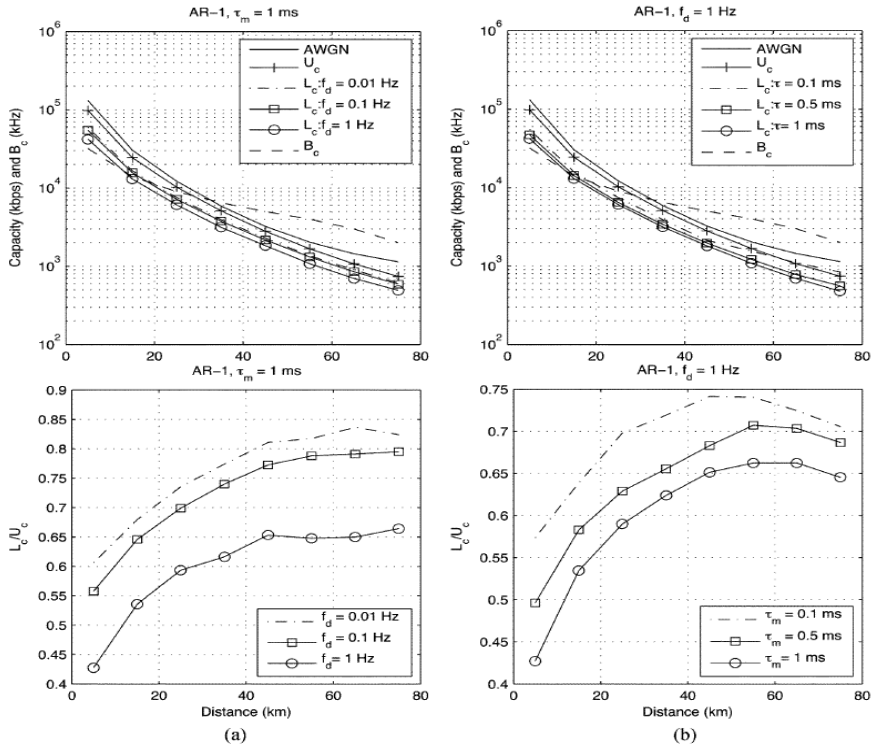


Fig. 5. Impact of (a) Doppler spread and (b) delay spread on $L_c(d)$ for AR-1 scattering function

As expected, the ratio between $L_c(d)$ and $U_c(d)$ increases as either f_d or τ_m increases. This is due to the higher prediction error influenced by stronger channel variations.

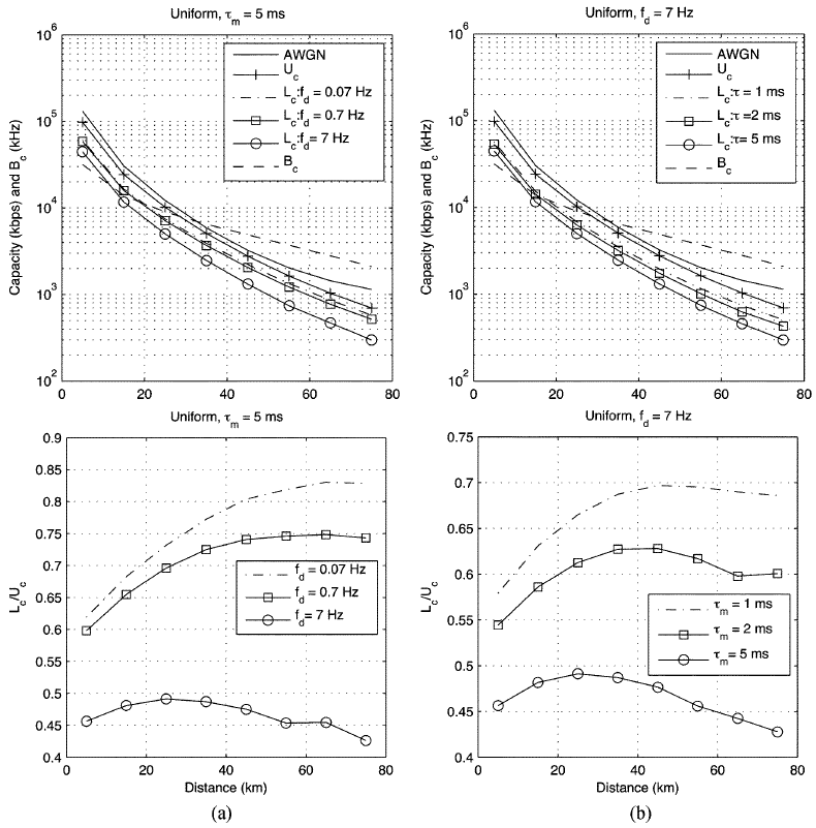


Fig. 6. Impact of (a) Doppler spread and (b) delay spread on $L_c(d)$ for uniform scattering

The impact of the shape of the scattering function is compared in Fig. 7 when f_d=1 Hz and τ_m=1 ms. We set F=500 kHz and T_b=15 ms. From the figure, $L_c(d)$ from $S_1[l, \lambda]$ is lower than that of $S_2[l, \lambda]$ as shown in Fig. 7(a). Fig. 7(b) shows the ratio of $L_c(d)$/ $U_c(d)$.

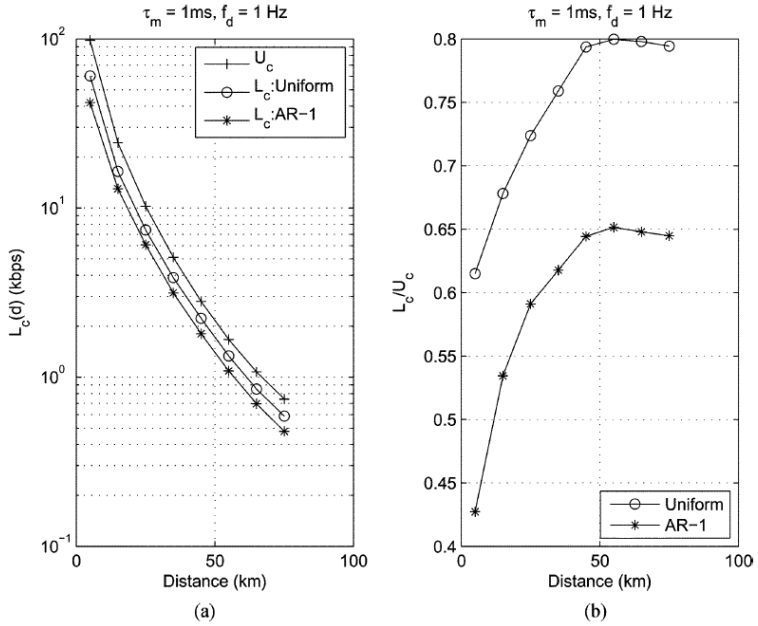


Fig. 7. Impact of the shape of scattering function on (a) $L_c(d)$ and (b) $L_c(d)$/ $U_c(d)$

4) Impact of Over Rician Fading Channels

The Ricean fading parameter ρ is set to -5, 0, 5, and 10 dB, identical for every path and independent of the distance. The Doppler spread profile of the scatter part (34) is assumed uniformly distributed. The fixed part is perfectly known at the receiver. From Fig. 8, the gap between the upper and lower bounds decreases as ρ increases which is due to the reduced power in the scatter part of the channel.

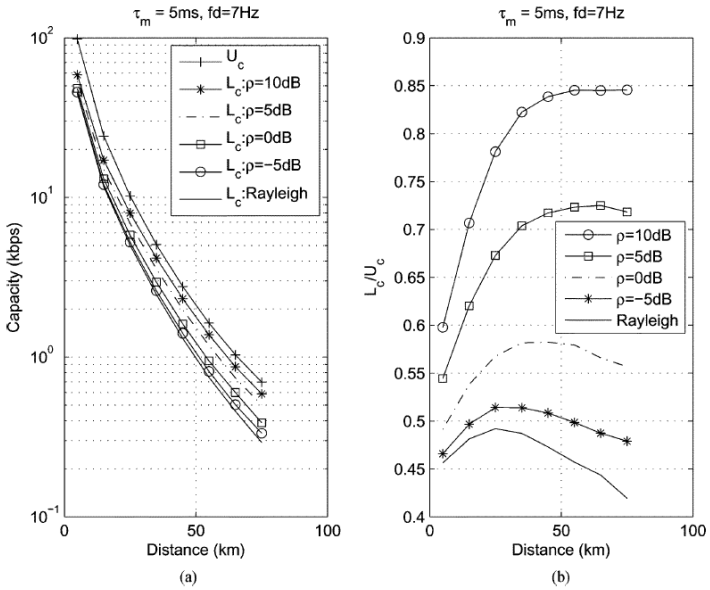


Fig. 8. $U_c(d)$ and $L_c(d)$ to the channel capacity over (a) Rician fading channel, (b) $L_c(d)/ U_c(d)$. Uniform Doppler spread profile

5) Impact of the Transmission Distance

From Figs. 5 and 6, both $L_c(d)$ and $U_c(d)$ decrease at longer distance owing to strong channel attenuation which determines $B_c(d)$. The gap at a short transmission distance is due to the energy wasted because of the PSK constellation while the gap at a very long distance is due to the higher prediction error because of the stronger attenuation.

6) Impact of Transmit Power

Fig. 9 shows the impact of transmit power on $L_c(d)$ and $B_c(d)$ for AR-1 scattering. A significant decrease in $L_c(d)$ and $B_c(d)$ occurs especially at long distance. This shows that for data transmission at low power, a short distance or multiple short hops across the transducers are preferred to one long transmission.

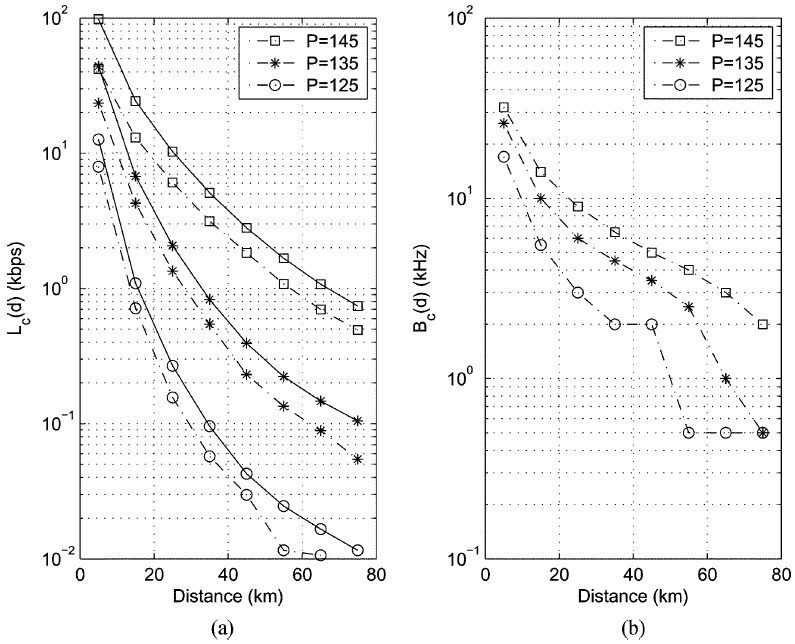


Fig. 9. Impact of transmit power on (a) $L_c(d)$ and (b) $B_c(d)$ for AR-1 scattering

V. EXPERIMENTAL DATA

The capacity of OFDM systems is investigated using the scattering function from real UW environments measured from the RACE08 experiment. Data is selected from the receiving arrays which are 1000 m from the transducer. The array is a 12-element vertical array with 12-cm spacing between elements. 8-PSK signals are upsampled by a factor of ten and filtered by a square root raised-cosine filter with a rolloff factor 0.25. A block of data which contain 64 data symbols are transmitted every 28.7 ms. A guard period is inserted between blocks to avoid the IBI. The bandwidth is 4.8 kHz at 12-kHz carrier frequency. Fig. 10(a) shows a contour plot of the estimates of the scattering function and Fig. 10(b) shows their corresponding PDP of process I–IV obtained from four different measurement periods.

Fig. 11(a) shows $L_c(d)$ and $U_c(d)$ from process I–IV over a range of the distance. Their corresponding $L_c(d)$ / $U_c(d)$ are displayed in Fig. 11(b). From the results, process II yields the best performance while process IV yields the worst. This is due to high Doppler spread at the dominant arrival paths in process IV while process II experiences smallest Doppler spread for almost every arrival path as shown in Fig. 10. Processes I and III exhibit similar results although process III is slightly worse since more dominant paths experience stronger Doppler spread compared to process I.

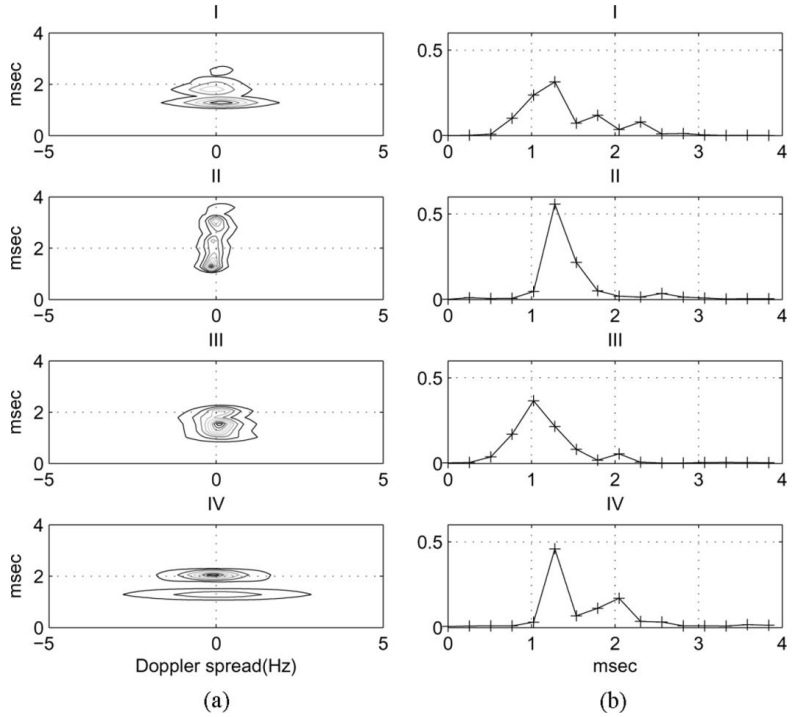


Fig. 10. (a) Scattering function estimates and (b) corresponding normalized PDP

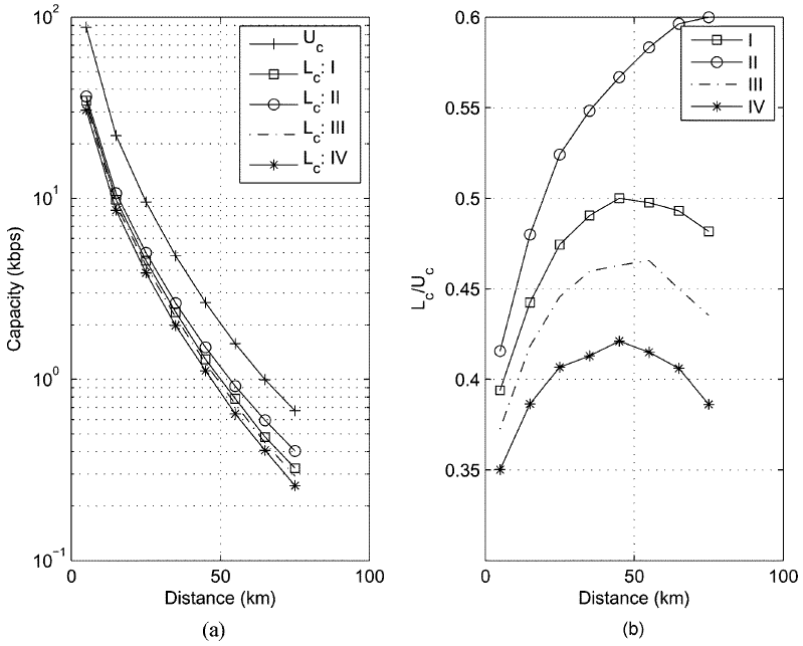


Fig. 11. (a) $L_c(d)$ $U_c(d)$ and (b) corresponding $L_c(d)/ U_c(d)$ over experimental UW fading channels

VI. DISCUSSION

After meeting, please write discussion in the meeting and update your presentation file.

Appendix

A. ICI Justification

To investigate the ICI impact, a simulation is run assuming that the ICI behaves as an independent complex Gaussian r.v. Therefore, the total noise accumulated in the simulation is the ICI plus the ambient noise. From (1), by including the ICI, the received signal can be written as

$$\begin{aligned} Y_{n,k} &= G_{n,k}(d)X_{n,k} + C_{n,k} + N_{n,k} \\ &= G_{n,k}(d)X_{n,k} + Z_{n,k} \end{aligned} \tag{36}$$

where $Z_{n,k}$ is the complex Gaussian noise consisting of the ambient and ICI noise whose variance is $E\left[\left|C_{n,k}\right|^2\right] + A(f_k)$. Using this assumption, Fig. 12 shows the $U_c(d)$ and $L_c(d)$ bounds at 5 km distance between transmitter and receiver. This distance gives highest ICI variance since longer distance means higher attenuation resulting in lower ICI.

In conclusion, it is shown that by taking into account the ICI as an additive complex Gaussian noise, $U_c(d)$ is reduced by at most 5.89% while $L_c(d)$ is reduced by at most 3.03%. This reduction is quite small and has little impact on the overall performance, and justifies our ICI setting.

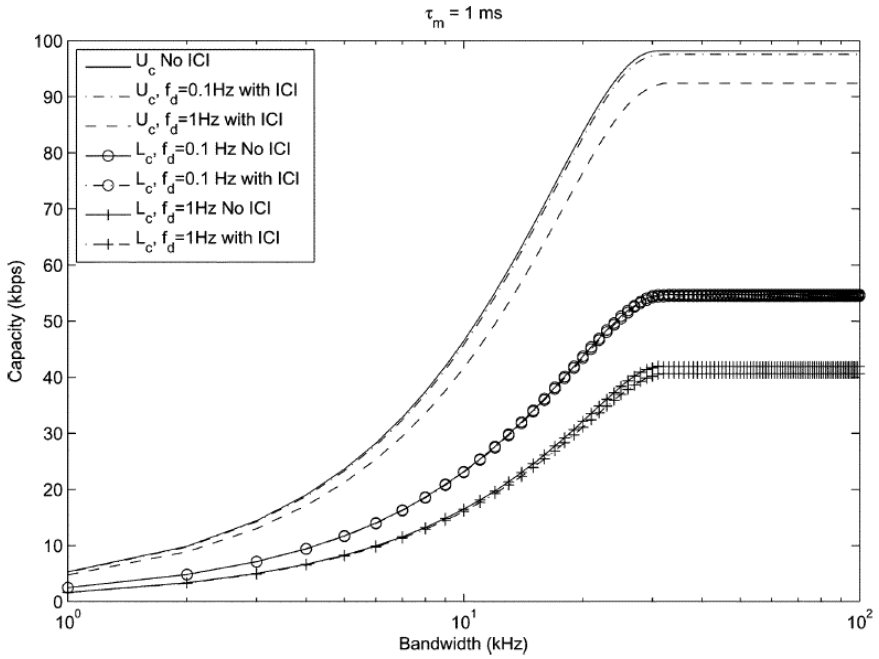


Fig. 12. Impact of ICI on $L_c(d)$ and $U_c(d)$ for AR-1 scattering function

B. I(Y;G(d)|X) Derivation

To calculate $I(\mathbf{Y};\mathbf{G}(d)|\mathbf{X})$, use the chain rule of differential entropy [11],

$$\begin{aligned} I\left(\mathbf{Y};\mathbf{G}(d)\,|\,\mathbf{X}\right) &= h\left(\mathbf{Y}\,|\,\mathbf{X}\right) - h\left(\mathbf{Y}\,|\,\mathbf{G}(d),\mathbf{X}\right) \\ &= h\left(\mathbf{Y}_0,\mathbf{Y}_1,...,\mathbf{Y}_{N-1}\,|\,\mathbf{X}\right) - N\sum_{k=0}^{K-1}\log\left(\pi eA(f_k)\right) \\ &= \sum_{n=0}^{N-1} h\left(\mathbf{Y}_n\,|\,\mathbf{Y}_0...\mathbf{Y}_{n-1},\mathbf{X}\right) - N\sum_{k=0}^{K-1}\log\left(\pi eA(f_k)\right) \end{aligned} \tag{37}$$

where

$$h\left(\mathbf{Y}_n\,|\,\mathbf{Y}_0...\mathbf{Y}_{n-1},\mathbf{X}\right) \triangleq \log\left(\left(\pi e\right)^K \det\left(\mathrm{Cov}\left[\mathbf{Y}_n\,|\,\mathbf{Y}_0...\mathbf{Y}_{n-1},\mathbf{X}\right]\right)\right) \tag{38}$$

To calculate $\mathrm{Cov}\left[\mathbf{Y}_n\,|\,\mathbf{Y}_0...\mathbf{Y}_{n-1},\mathbf{X}\right]$, we begin with mean

$$\begin{aligned} E\left[\mathbf{Y}_n\,|\,\mathbf{Y}_0...\mathbf{Y}_{n-1},\mathbf{X}\right] &= \mathrm{diag}(\mathbf{X}_n)E\left[\mathbf{G}_n(d)\,|\,\mathbf{Y}_n\,|\,\mathbf{Y}_0...\mathbf{Y}_{n-1},\mathbf{X}\right] \\ &= \left(\mathrm{diag}\left(\mathbf{X}_n\right)\mathbf{Q}(d)\right)\hat{\mathbf{H}}_n(d) \end{aligned} \tag{39}$$

Where (39) is obtained using (2) and (19). $\hat{\mathbf{H}}_n(d) = E\left[\mathbf{H}_n(d)\,|\,\mathbf{Y}_0,...,\mathbf{Y}_{n-1},\mathbf{X}\right]$ is the MMSE channel estimate given the current and past detected symbols and can be written as the 1-step output of the linear $K\times K$ MIMO predictor filter of length J

$$\hat{\mathbf{H}}_n(d) = \sum_{j=1}^{J} E_j(d)\tilde{\mathbf{H}}_{n-j}(d) \tag{40}$$

$\tilde{\mathbf{H}}_n(d) = \left[\tilde{H}_{n,0}(d)...\tilde{H}_{n,K-1}(d)\right]^T$ and $\mathbf{E}_j(d)$ is the predictor coefficient of size $K\times K$. With (1), the observation $\tilde{H}_{n,K-1}(d)$ is obtained by

$$\tilde{H}_{n,k}(d) = \frac{Y_{n,k}}{X_{n,k}Q(d,f_k)} = H_{n,k} + \frac{N_{n,k}}{X_{n,k}Q(d,f_k)} \tag{41}$$

Then from (39), $\mathrm{Cov}\left[\mathbf{Y}_n\,|\,\mathbf{Y}_0...\mathbf{Y}_{n-1},\mathbf{X}\right]$ is

$$\begin{aligned} &\mathrm{Cov}\left[\mathbf{Y}_n\,|\,\mathbf{Y}_0.s\mathbf{Y}_{n-1},\mathbf{X}\right] \\ &\triangleq E\left[\left(\mathbf{Y}_n - \left(\mathrm{daig}(\mathbf{X}_n)\mathbf{Q}(d)\right)\hat{\mathbf{H}}_n(d)\right)\times\left(\mathbf{Y}_n - \left(\mathrm{daig}(\mathbf{X}_n)\mathbf{Q}(d)\right)\hat{\mathbf{H}}_n(d)\right)^H\right] \\ &= \left(\mathrm{daig}(\mathbf{X}_n)\mathbf{Q}(d)\right)\mathbf{B}_n(d)\left(\mathrm{daig}(\mathbf{X}_n^*)\mathbf{Q}(d)\right) + \mathrm{diag}(\mathbf{A}_n) \end{aligned} \tag{42}$$

Where $\mathbf{B}_n(d)$ is the linear MMSE prediction error matrix obtained using the orthogonality principles.

Substituting (42) into (38) and into (37), $I(\mathbf{Y};\mathbf{G}(d)|\mathbf{X})$ is given as

$$\mathbf{I}(\mathbf{Y};\mathbf{G}(d)\,|\,\mathbf{X}) = \sum_{n=0}^{N-1} \log\det\left(\mathbf{I} + \mathrm{diag}\left(\mathbf{X}_n\right)\mathbf{B}_n(d)\,\mathrm{diag}\left(\mathbf{X}_n^*\right)\mathrm{diag}(\mathbf{A}_n)^{-1}\right)$$

$$= \sum_{n=0}^{N-1} \log\det\left(\mathbf{I} + \mathbf{B}_n(d)\,\mathrm{diag}\left(\mathbf{X}_n^*\right)\mathrm{diag}(\mathbf{A}_n)^{-1}\,\mathrm{diag}\left(\mathbf{X}_n\right)\right) \quad \because \det(\mathbf{I}+\mathbf{XY}) = \det(\mathbf{I}+\mathbf{YX}) \tag{43}$$

$$= \sum_{n=0}^{N-1} \log\det\left(\mathbf{I} + \mathbf{B}_n(d)\,\mathrm{diag}\left(\mathbf{S}(d)\right)\right)$$

The kth entry of the $K\times 1$ vector $\mathbf{S}(d)$ is $\sigma_x^2(d, f_k)Q^2(d, f_k) / A(f_k)$.

References

[1] Same as paper.

[2]

Master Thesis:
Compressive sensing and its application in Wireless Sensor Network & Correlated Signal Recovery Method

Abstract

In this paper, we consider the application of compressive sensing (CS) in wireless sensor networks (WSNs). CS is a signal acquisition and compression framework recently developed in the field of signal processing and information theory. We applied this CS technique to WSN which consists of a large number of wireless sensor nodes and a central fusion center (FC). 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 signals in polynomial time using a signal recovery algorithm. The distributed sensors observe similar event in designated region. Therefore, the observed signals have considerable correlation each other. We make some effort in modeling correlation between the signals acquired from the sensors and analyze the component in observed signals. After modeling the correlated signals, we propose POMP (Phased-OMP) which can recover any type of correlated signals stably and effectively. We introduce the idea of our proposed algorithm in detail and then compare the reconstruction performance of POMP with previous algorithms ReMBo, MEM, SOMP, etc.

©2012

Jae-Gun Choi

ALL RIGHTS RESERVED

Contents

List of Figures

List of Tables

1. Introduction

In this paper, we discuss the application of a new compression technique called compressive sensing (CS) in wireless sensor networks (WSNs). The objective of a WSN which we assume in this paper 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 paper, 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.

We applied this CS technique to WSN. One of our aims in this paper 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.

The distributed sensors observe similar event in designated region. Therefore, the observed signals have considerable correlation each other. In this paper, 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. Then, we divide the correlated signals to three parts for example, common sparsity, innovation sparsity, and total sparsity. Those terminologies give more easy understanding to solve multiple measurement vector (MMV) modeled from WSN structure.

If we will use the correlated information to recover signals transmitted from each sensor, its reconstruction performance will increase over that not using correlated information. We demonstrated this assumption by showing a simulation result. Finally, we proposed advanced algorithm to recovery the correlated signals effectively. The proposed algorithm is called phased advanced orthogonal matching pursuit (POMP). POMP has better performance about reconstruction probability than previous algorithms, for examples, SOMP, ReMBo etc. We will introduce the idea of our proposed algorithm in detail and then compare the reconstruction performance of our algorithms with previous algorithms

2. Wireless sensor network

2.1. Network structure

We consider a WSN consisting of a large number of wireless sensor nodes and one FC (**Figure 1**). 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].

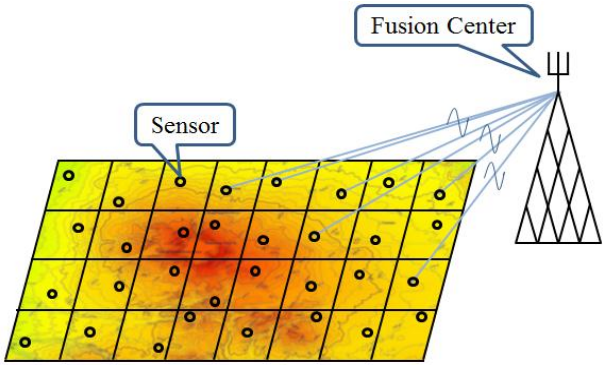


Figure 1. Wireless Sensor Network (WSN)

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.

2.2. 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. Moreover, this data reduction comes without utilizing complex signal processing. Namely, the sensor nodes can compress the signal while not spending any power for running complex compression algorithms onboard.

We discuss the new technique CS in the next section and check how CS can get the advantages like data reduction and simple data compression.

3. Compressive sensing (Literature survey)

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 2**, can bypass these intermediate steps, and thus provides a light weight signal acquisition apparatus which is suitable for those sensor nodes in our WSN.

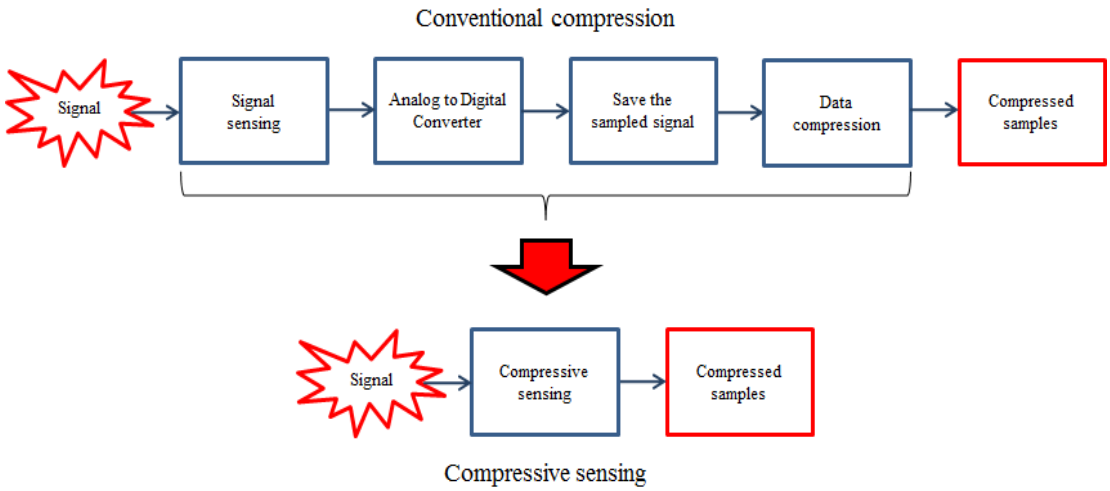


Figure 2. Conventional compression and compressive sensing

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.

3.1 Theoretical 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.**Error! Reference source not found.** .

.

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

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 (**Figure 3**):

$$\mathbf{y} = \Phi\mathbf{s} = \Phi\Psi\mathbf{x} = \mathbf{A}\mathbf{x} \tag{3}$$

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. **Error! Reference source not found.** 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.

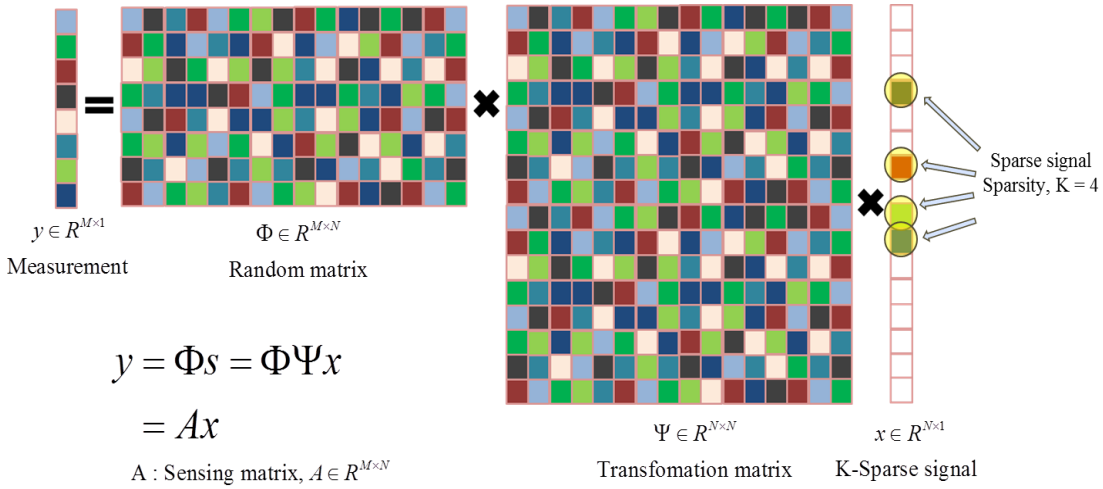


Figure 3. The summary of compressive sensing

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

The k -sparse signal, $\mathbf{s}$ in Eq. **Error! Reference source not found.**, 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

The sensing matrices are Φ and $\mathbf{A}$ in Eq. **Error! Reference source not found.**, where its dimension $\mathbf{R}^{m\times n}$, $m < n$. 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.

Error! Reference source not found. 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:

$$\mathbf{y} = \mathbf{A}_{\kappa}\mathbf{x}_{\kappa} \tag{4}$$

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. **Error! Reference source not found.** has the unique solution $\mathbf{x}_{\kappa}$ if the columns of $\mathbf{A}_{\kappa}$ are linearly independent. The solution can be found using pseudo inverse easily as Eq. **Error! Reference source not found.**

$$\mathbf{x}_{\kappa} = \left(\mathbf{A}_{\kappa}^{T}\mathbf{A}_{\kappa}\right)^{-1}\mathbf{A}_{\kappa}^{T}\mathbf{y} \tag{5}$$

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 $\mathbf{A}$ cannot be a good sensing matrix. In the extreme case, we can show a matrix $\mathbf{A}$ having m rows of Φ that are the first m columns of Ψ.

$$\mathbf{A} = \Phi\Psi = \Psi^{T}_{(1:m,:)}\Psi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \tag{6}$$

If $\mathbf{A}$ of Eq. **Error! Reference source not found.** is used as sensing matrix, the compressed measurement vector $\mathbf{y}$ captures only the first m elements of the vector $\mathbf{x}$, 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.

3.2 System equations

We knew the method how to find a unique solution of CS problem in previous section. In this section, we discuss various equations which are handled in CS theory as single measurement vector (SMV) and multiple measurement vector (MMV). The SMV is a basic equation for CS. It is expressed as Eq. **Error! Reference source not found.**. Many CS paper about this SMV problem is researched in [Ref],[Ref].

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{n-1} \\ y_n \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & & a_{2,n} \\ \vdots & \vdots & & & \vdots \\ \vdots & \vdots & & & \vdots \\ a_{m,1} & a_{m,2} & \cdots & \cdots & a_{m,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{bmatrix} \quad (7)$$

Otherwise, the MMV has multiple measurement vectors and sparse matrix as Eq. **Error! Reference source not found.**. The sparse vector in each SMV results in MMV. It has much unknowns compared with SMV. The many number of unknowns may make the MMV to be solved hard. To solve this equation effectively, some algorithms are proposed as SOMP, ReMBo, M-FOCUSS. If each column of sparse matrix $\mathbf{X}$ has similar support set, the priori information about support location can be used to get exact solution easily.

$$\mathbf{Y} = \mathbf{AX}$$

$$\begin{bmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,J} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,J} \\ & & & \\ y_{m,1} & y_{m,2} & \cdots & y_{m,J} \end{bmatrix}_{1,} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & & a_{2,n} \\ \vdots & \vdots & & & \vdots \\ \vdots & \vdots & & & \vdots \\ a_{m,1} & a_{m,2} & \cdots & \cdots & a_{m,n} \end{bmatrix} \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,J} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,J} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ & & & \\ \vdots & \vdots & & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,J} \end{bmatrix}_{1} \quad (8)$$

The MMV equation can have the more number of equations by transforming Eq. **Error! Reference source not found.**. It means that the MMV equation has more information to solve underdetermined equation. The modified MMV equation is expressed as below Eq. **Error! Reference source not found.**. Furthermore, infinite measurement vector (IMV) consists of an infinite set of jointly sparse vectors.

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

$$\begin{bmatrix} y_{1,1} \\ y_{2,1} \\ \vdots \\ \\ y_{m-1,1} \\ y_{m,1} \end{bmatrix}_1 = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & & a_{2,n} \\ \vdots & \vdots & & & \vdots \\ \vdots & \vdots & & & \vdots \\ a_{m,1} & a_{m,2} & \cdots & \cdots & a_{m,n} \end{bmatrix}_1 \begin{bmatrix} x_{1,1} \\ x_{2,1} \\ \vdots \\ \vdots \\ \\ \vdots \\ x_{n,1} \end{bmatrix}_1 ,\cdots, \begin{bmatrix} y_{1,J} \\ y_{2,J} \\ \vdots \\ \\ y_{m-1,J} \\ y_{m,J} \end{bmatrix}_J = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & & a_{2,n} \\ \vdots & \vdots & & & \vdots \\ \vdots & \vdots & & & \vdots \\ a_{m,1} & a_{m,2} & \cdots & \cdots & a_{m,n} \end{bmatrix}_J \begin{bmatrix} x_{1,J} \\ x_{2,J} \\ \vdots \\ \vdots \\ \\ \vdots \\ x_{n,J} \end{bmatrix}_J$$

$$\mathbf{Y} = \mathbf{A}_j \mathbf{X}, \text{ where } j \in \{1, 2, ..., J\}$$

$$\Rightarrow \begin{bmatrix} y_{1,1} & y_{1,2} & \cdots & y_{1,J} \\ y_{2,1} & y_{2,2} & \cdots & y_{2,J} \\ & & & \\ y_{m,1} & y_{m,2} & \cdots & y_{m,J} \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & & a_{2,n} \\ \vdots & \vdots & & & \vdots \\ \vdots & \vdots & & & \vdots \\ a_{m,1} & a_{m,2} & \cdots & \cdots & a_{m,n} \end{bmatrix}_j \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,J} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,J} \\ \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots \\ & & & \\ \vdots & \vdots & & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,J} \end{bmatrix} \quad (9)$$

Therefore, we can draw the relationship among the SMV, MMV and IMV as **Figure 4.** As the **Figure 4** shows, the MMV includes all of the SMV. It means that the MMV has all the information of the SMV. Therefore, if we solve the MMV equation exactly, it results the solution of each SMV also.

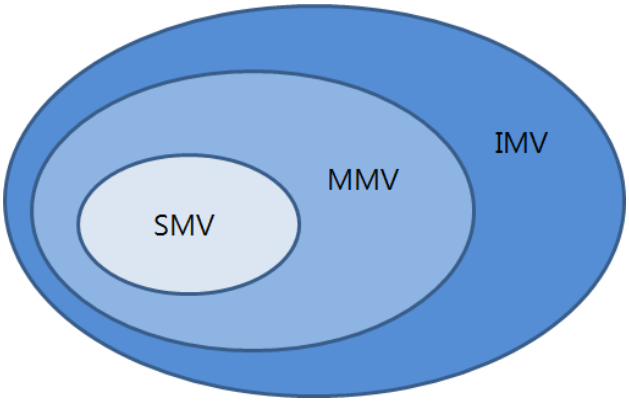


Figure 4. The relationship among SMV, MMV and IMV

3.3. Unique solution condition of SMV and MMV

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 the k-sparse vector $\mathbf{x}$ as an output. 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

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{\frac{1}{p}}, \quad p > 0 \tag{10}$$

.

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 to get solution $\mathbf{x}$.

i) The minimization problem in SMV

1) L_2 norm minimization in SMV

$$\begin{aligned} (L_2)\ \hat{\mathbf{x}} &= \arg\min \|\mathbf{x}\|_2 \quad \text{subject to } \mathbf{y} = \mathbf{A}\mathbf{x}, \text{ where } \mathbf{A} \in \mathrm{R}^{m \times n},\ rank(\mathbf{A}) = m \\ &= \mathbf{A}^T \left(\mathbf{A}\mathbf{A}^T \right)^{-1} \mathbf{y} \end{aligned} \tag{11}$$

However, this conventional solution yields a non-sparse solution, so it is not appropriate as a solution to the CS problem. Thus, we do not consider this method for finding solution.

2) L_0 norm minimization in SMV

$$(L_0) \text{ Minimize } \|\mathbf{x}\|_0 \text{ subject to } \mathbf{y} = \mathbf{A}\mathbf{x}, \text{ where } \mathbf{A} \in \mathrm{R}^{m \times n},\ rank(\mathbf{A}) = m \tag{12}$$

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. In literature [Ref], the unique solution of the L_0 minimization is known as following,

$$k < \frac{\mathrm{spark}(\mathbf{A})}{2} \tag{13}$$

The $\mathrm{spark}(\mathbf{A})$ is the smallest number n such that there exists a set of n columns in $\mathbf{A}$ which are linearly dependent. In summary, if the above equation is satisfied, then the unique solution of the Eq. **Error! Reference source not found.** is guaranteed.

3) L_1 norm minimization in SMV

$$(L_1) \text{ Minimize } \|\mathbf{x}\|_1 \text{ subject to } \mathbf{y} = \mathbf{A}\mathbf{x}, \text{where } \mathbf{A} \in \mathrm{R}^{m \times n},\ rank(\mathbf{A}) = m \tag{14}$$

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. **Error! Reference source not found.** and **Error! Reference source not found.** have unique solutions. We provide a theorem related to this issue.

L_0 / L_1 **equivalence condition in SMV:**
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] .

Table 1. L_0 / L_1 Equivalence condition.

ii) The minimization problem in MMV

To get the unique solution of MMV, it can be considered similar method with that of SMV. We introduce theorems from references [Ref]. To explain the uniqueness condition for MMV, we introduce the following definitions $R(\mathbf{X})$ and $relax(\mathbf{X})$.

$$R(\mathbf{X}) = \left\| m(\mathbf{x}_i)_{n \times 1} \right\|_0 \tag{15}$$

where $x_i \in R^L$ is the transpose of the ith row of matrix $\mathbf{X}$, ie $\mathbf{X} = [x_1, x_2, ..., x_n]^T$, $m(\cdot)$ is any vector norm in R^L. Therefore, $R(\mathbf{X})$ is the number of rows which have nonzero element in matrix $\mathbf{X}$. When norm of $m(\mathbf{x}_i)_{n\times 1}$ is one, then it is defined as $relax(\mathbf{X})$.

$$relax(\mathbf{X}) = \left\| \left(m(\mathbf{x}_i) \right)_{n\times 1} \right\|_1 \tag{16}$$

1) L_0 norm minimization in MMV

$$(L_0) \quad \text{Minimize } R(\mathbf{X}) = \left\| m(\mathbf{x}_i)_{n\times 1} \right\|_0 \quad \text{subject to} \quad \mathbf{A} \in \mathbb{R}^{m\times n}, \text{with } rank(\mathbf{A}) = m \tag{17}$$

In literature [Ref], the MMV unique solution of the L_0 minimization is known as following,

$$R(\mathbf{X}) < \frac{spark(\mathbf{A}) - 1 + rank\left(Cols(\mathbf{Y})\right)}{2} \tag{18}$$

The $rank\left(Cols(\mathbf{Y})\right)$ is the column rank of matrix $\mathbf{Y}$. If the above equation is satisfied, then the unique solution of the Eq. **Error! Reference source not found.** is guaranteed.

2) L_1 norm minimization in MMV

$$(L_1) \quad \text{Minimize } relax(\mathbf{X}) = \left\| m(\mathbf{x}_i)_{n\times 1} \right\|_1 \quad \text{subject to} \quad \mathbf{A} \in \mathbb{R}^{m\times n}, \text{with } rank(\mathbf{A}) = m \tag{19}$$

A sufficient condition to be the unique solution to 2) of MMV is that

$$\left\| \mathbf{A}_S^{\dagger} \mathbf{A}_j \right\|_1 < 1, \ \forall j \notin S \tag{20}$$

.

$\mathbf{A}_S$ is reduced matrix of $\mathbf{A}$ corresponding to indices from support location of $R(\mathbf{X})$. So, we can write $\mathbf{Y} = \mathbf{A}_S \mathbf{X}_S$, where matrix $\mathbf{X}_S$ is made by nonzero rows of $\mathbf{X}$. $\mathbf{A}_S$ is of full column rank. $\mathbf{A}_S^{\dagger}$ is pseudo-inverse which is defined by $\mathbf{A}_S^{\dagger} = \left(\mathbf{A}_S^T \mathbf{A}_S \right)^{-1} \mathbf{A}_S^T$. Because $\mathbf{A}_S$ is of full column rank, the generalized inverse is well defined. The above is the Exact Recovery Condition (ERC) in Tropp's "Greed is good: Algorithmic results for sparse approximation"

L_0 / L_1 **equivalence condition in MMV:**
If $R(\mathbf{X}) < \frac{spark(\mathbf{A})}{2}$ is satisfied, then the solution of L_1 in MMV coincides with that of L_0 [Ref] .

Table 1. L_0 / L_1 Equivalence condition.

4. Compressive sensing and its application in WSN

4.1 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, the distributed sensors observe physical changes in designed area. Since each sensor observes similar physical changes, the signals observed from each sensor have much correlation. The correlated signal can be compressed for reducing data. The conventional compression for WSN requires exchanging information between distributed sensors in order to exploit inter-sensor correlation. Such a transmission strategy makes the network system complex below **Figure 9**.

The conventional compression needs to get together redundant data for compression as **Figure 10**. 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.

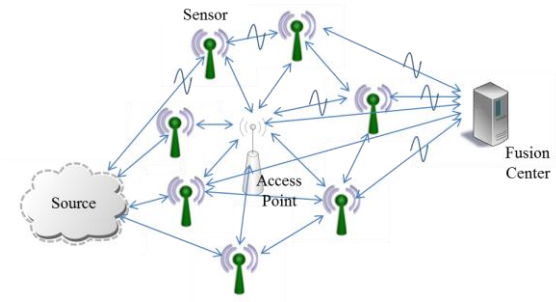


Figure 9. Conventional sensor network structure

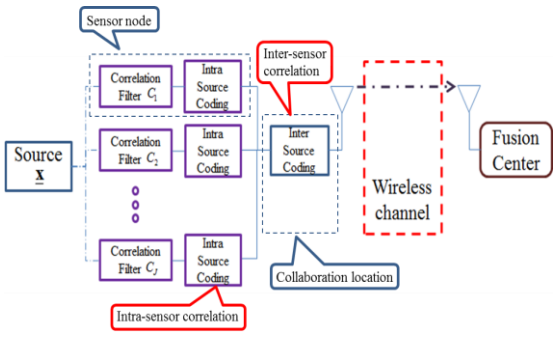


Figure 10. Conventional sensor network structure

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 2**. To compress the high-dimensional signal $\mathbf{x}$ into a low-dimensional signal $\mathbf{y}$, as Eq. **Error! Reference source not found.**, 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 12**.

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. The CS sensor network structure applied for WSN is as below (**Figure 11**). You can check the simplicity of transmission strategy of CS based WSN compared with conventional network.

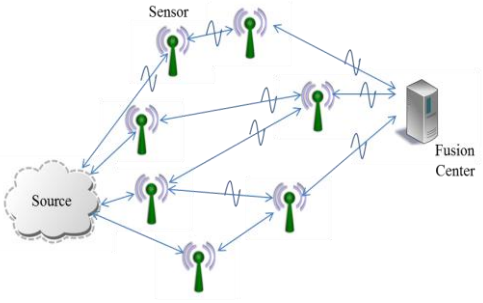


Figure 11. CS sensor network scheme

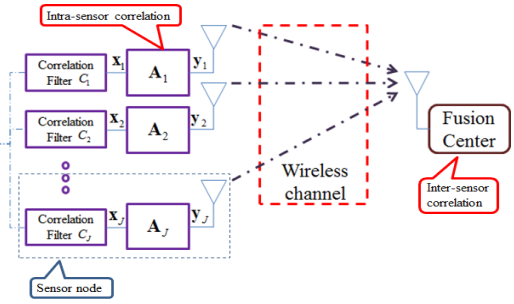


Figure 12. CS sensor network structure

4.2 Distributed compressive sensing

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 13**, 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

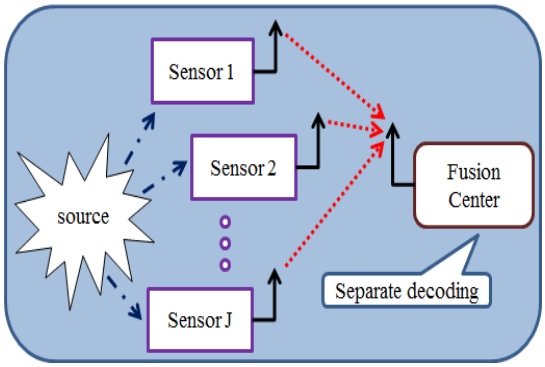
ii) Exploiting both intra- and inter-sensor correlation

Figure 14 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 10**, 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



F i g u r e .1 B n t r a - s e n s o r c o r r e l

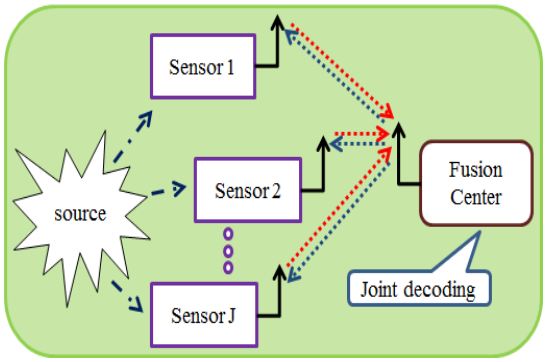
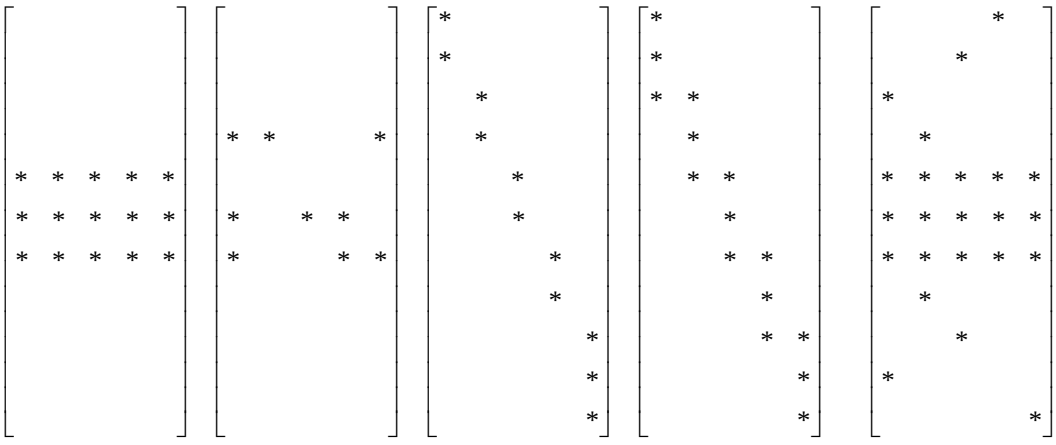


Figure 14. Intra/Inter-sensor correlation scheme

4.3. Correlated signal models

We assumed the WSN which consists of large number of sensor with a built-in CS and one fusion center. Because of the feature of considered WSN, the observed signals have inter-sensor correlation. We can model this WSN as Eq. **Error! Reference source not found.** or Eq. **Error! Reference source not found.**. Those two equations have sparse signal matrix **X** which consists of signals transmitted from each sensor.

In this section, we introduce how the signal matrix with different degrees of correlation can be generated as sparse signal models. The sparse signal matrix in WSN has correlated properties. The degree of sparseness which is called the sparsity, is proportional to the amount of correlation. More correlated signal means sparser in terms of intra-sensor correlation. 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. By using those two properties, we can model correlated sparse signal matrix **X** as below examples **Figure 15**.



*: Unknown nonzero value

Figure 15. The examples of correlated signals

We can divide those correlated signals in **Figure 15** as three components; common sparsity part, innovation sparsity part, and total sparsity part. The common sparsity part has one more nonzero value in the row of sparse signal matrix **X**. The Innovation sparsity part has only one nonzero value in the row of signal matrix. Lastly, the total sparsity part is the total number of rows which have nonzero elements. The common sparsity is a correlated part. Therefore, if we find the location of common part, we can also use it to solve another SMV. The innovation sparsity is a uncorrelated part. Even if we find the location of innovation part, we cannot use it to solve other SMV equations. Finally, the total sparsity is related with the degree of correlation among observed signals. We re-expressed the correlated signals as following **Figure 16** by using three terminologies mentioned.

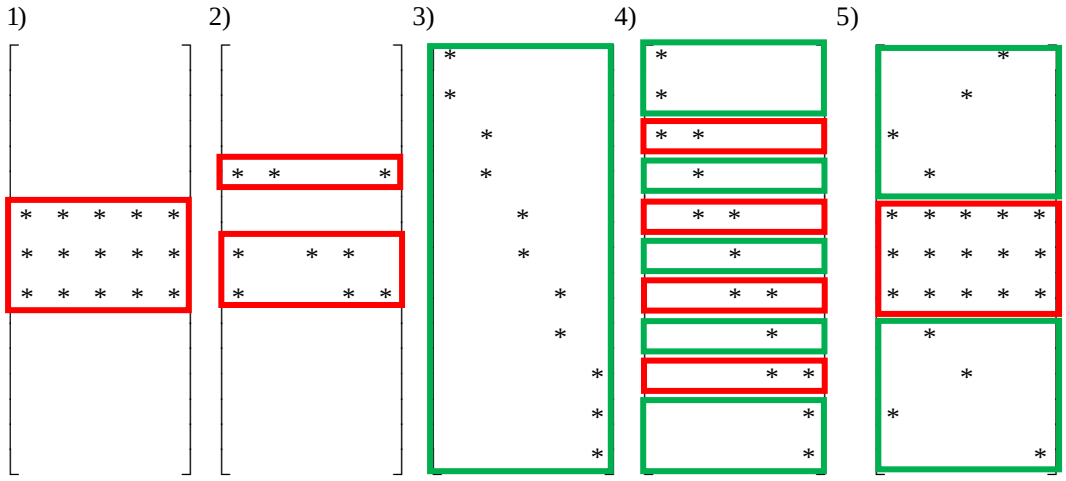


Figure 16. The components of correalted signals

The first and second correlated signals of **Figure 16** have only common sparsity part. The third signal consists of only innovation sparsity. Therefore, there is no common part which has one

more nonzero element in same row. The 4), 5) signals have both common part and innovation part. If we know the prior information of the heuristic signal **X** about support location, we can use it to find solution effectively. We can use those correlated properties to recover signals transmitted from each sensor, and its reconstruction performance will increase over that not using correlated information. We discuss the ideas for recovering those correlated signals in next section.

5. The recovery ideas for correlated signals

5.1 Joint decoding and separate decoding

We discussed the correlated signals which consist of common, innovation, and total parts in the previous section. The understanding of various correlated signal models gives more clues to get solution. In this section, we argue ideas using correlated information to get solution effectively.

Some specific correlated signals also are handled in [Ref],[Ref]. In those references, the correlation signals are 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. In JSM-2, it shares same support location that has different value. Those two signals is expressed below, **Figure 17**. In [Ref],[Ref], they proposed methods which find the solution of the correlated signals consisting of those specific pattern in Eq. **Error! Reference source not found.**.

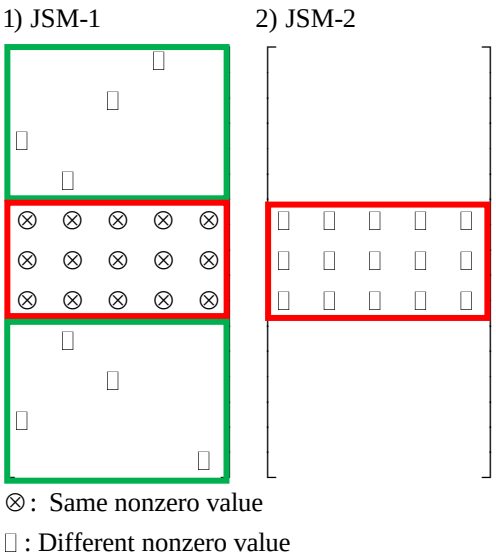


Figure 17. Joint signal models, JSM-1, JSM-2

The JSM-1 is expressed as

$$\mathbf{x}_j = \mathbf{z}_c + \mathbf{z}_j,\ j \in \{1,2,...,J\},\ j \text{ is the index of the sensors} \tag{21}$$

where $\|\mathbf{z}_c\|_0 = k_c$ (Common part), and $\|\mathbf{z}_j\|_0 = k_j$ (Innovation part) in each sensed signal. Obviously, $\mathbf{z}_c$ appears in all the columns of the correlated 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 j th 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 pattern. Then, the FC can exploit both intra- and inter- sensor correlation by solving Eq. **Error! Reference source not found.** as described below.

i) Joint recovery scheme for JSM-1 (Modified equation method)

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 sparsity of vectors $\mathbf{z}_c$ and $\mathbf{z}_j$ are k_c and k_j, respectively and each sensor has same spasity $k = k_c + k_j$. Then, 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. **Error! Reference source not found.**. Therefore, the sensed signal in JSM-1 is transformed into **Figure 18**. This idea is handled in [Ref],[Ref].

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \\ \vdots \\ \mathbf{y}_J \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{A}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_2 & \mathbf{0} & \mathbf{A}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{A}_3 & \mathbf{0} & \mathbf{0} & \mathbf{A}_3 & \mathbf{0} & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{A}_J & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_J \end{bmatrix} \begin{bmatrix} \mathbf{z}_c \\ \mathbf{z}_1 \\ \mathbf{z}_2 \\ \mathbf{z}_3 \\ \vdots \\ \mathbf{z}_J \end{bmatrix} \tag{22}$$

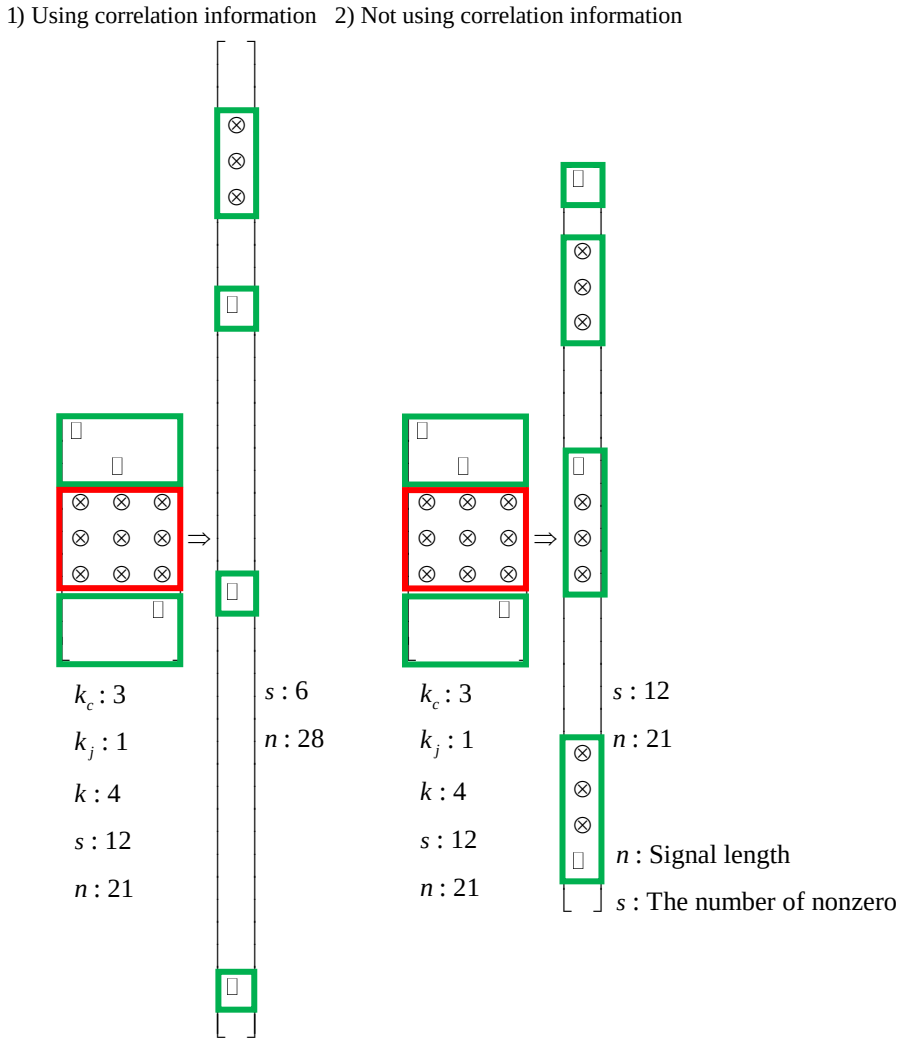


Figure 18. Concatenating JSM-1 to a column signal

Because JSM-1 shares common part $\mathbf{z}_c$ in the equation, we can reduce the number of nonzero value as 1) of **Figure 18**. In conclusion, the total number of nonzero in matrix $\mathbf{X}$ is 12, but in transforming equation, it is 6 only. Thus, the total number of nonzero, s is reduced from $J \times \left(k_c + k_j \right)$ to $k_c + \left(J \times k_j \right)$. The total number of sparsity affects the probability of exact reconstruction. By solving this equation, the FC can take advantage of exploiting inter-sensor correlation. 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 as before example. Therefore, the received signals are recovered as follows and its concatenated signal is express as 2) of **Figure 18**.

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \vdots \\ \mathbf{y}_J \end{bmatrix} = \begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{0} & \vdots \\ \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{A}_J \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_J \end{bmatrix} \tag{23}$$

To solve Eq. **Error! Reference source not found.** and **Error! Reference source not found.**, we use the primal-dual interior point method (PDIP) in **Appendix 7.1**, which is an L_1 minimization algorithm, and compare the results of the two types of recovery, joint decoding and separate decoding respectively. Using the comparison results, 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 i th coefficient for $\mathbf{x}_1$ is nonzero if and only if the i th 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) Joint and separate decoding scheme for JSM-2

The prior inter-correlation becomes relevant when the number of sensors is more than two. To get the advantages of exploiting inter-sensor correlation about JSM-2, we should solve the Eq. **Error! Reference source not found.** and Eq. **Error! Reference source not found.** jointly. Like the FC in JSM-1, the FC in JSM-2 can exploit the fact that the support set is shared. By solving the MMV jointly, we obtain several benefits as high reconstruction probability on same number of measurement. If we solve those two equations 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 (SOMP) modified from an OMP algorithm for joint decoding and apply OMP for separate decoding. These algorithms are introduced in **Appendix 7.2** and **7.3** correspondingly.

iv) Joint vs. separate recovery performance for JSM-1 and JSM-2

Now, we compare the results of joint recovery and separate recovery. 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 19**, solid lines were obtained from joint reconstructions, whereas dotted lines are the results of separate reconstructions.

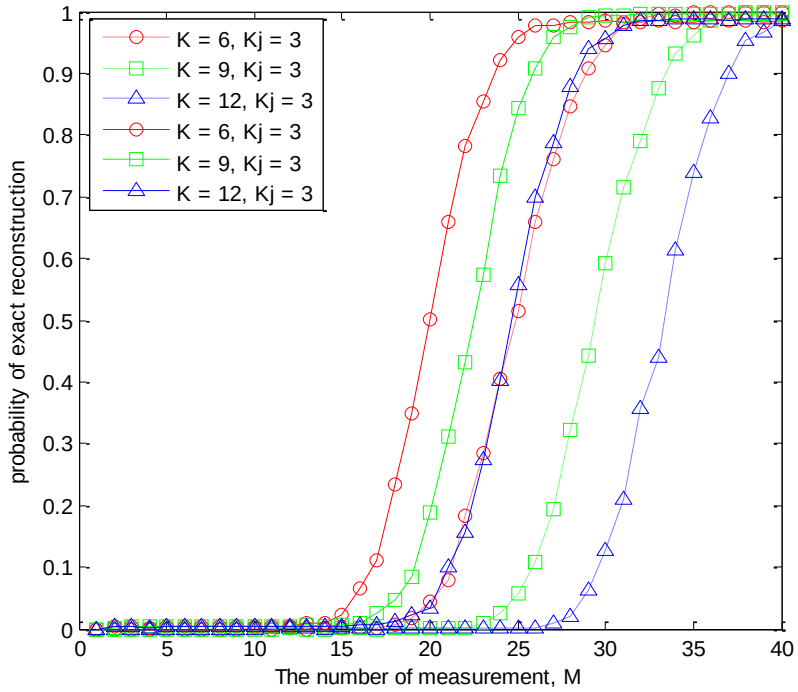


Figure 19. Joint (solid line) and separate (dotted line) reconstruction using PDIP algorithm for JSM-1. System parameters: $N = 50$, $J = 2$. The benefits of joint reconstruction depend on the common sparsity k_c.

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 20**, 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 19** 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.

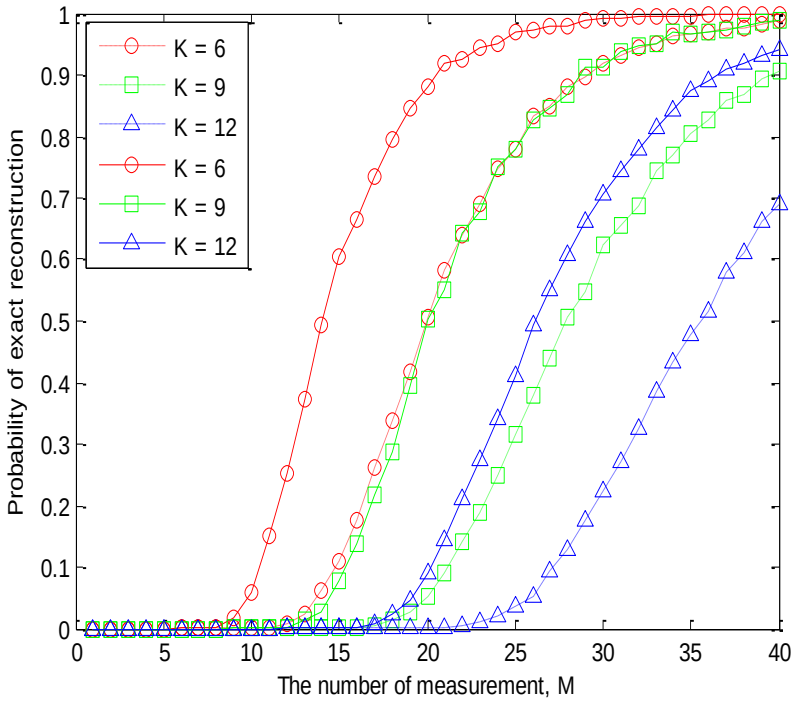


Figure 20. Joint (solid line) and separate (dotted line) reconstruction using SOMP for **JSM-2**. System parameters: $N = 50$, $J = 2$. Joint reconstruction has a higher probability of success than separate reconstruction.

5.2 Phased-Orthogonal matching pursuit (POMP)

In previous section, we discussed joint decoding methods for specific correlated signal, JSM-1 and JSM-2. The joint decoding method for JSM-1 cannot apply for JSM-2 which shares same support location only, since JSM-2 does not have same nonzero value. Therefore, if we use same idea for JSM-2, we cannot get the advantages of exploiting inter-sensor correlation. In reverse, SOMP, joint decoding algorithm for JSM-2, cannot apply for JSM-1 which has a large number of innovation sparsity. If we use SOMP algorithm for JSM-1, it may not find solution exactly. In summary, those two methods cannot apply all of the correlated signals which have various correlated pattern. To get exact solution of various correlated signals, we proposed joint decoding algorithm. The proposed algorithm is called phased orthogonal matching pursuit (POMP). POMP has better performance about the exact reconstruction probability of correlated signals than previous algorithms, for examples, PDIP, SOMP, ReMBo, etc. We will introduce the idea of our proposed algorithm in detail and then compare the reconstruction performance of our algorithms with previous algorithms.

i) Previous algorithm for MMV

1) One-step greedy algorithm

Figure 21 plots the probability of success in recovering the support set by using the one-step greedy algorithm (OSGA). OSGA finds common support location by using method described in **Table 9**, for the JSM-2 signal. In comparison with other greedy algorithms, it finds all the nonzero location at once so its performance is lower than that of SOMP.

The result of **Figure 21** suggests that the number of required measurements decreases for the same probability of exact reconstruction as the number of sensors increases. The OSGA works for a small

number of measurements m if the number of sensors is sufficiently large. Therefore, if many distributed sensors observe a correlated signal, each sensor is enough to send only a small number of compressed signals to achieve perfect reconstruction probability. Consequently, the transmission power of each sensor can be reduced because only the traffic volume required for exact reconstruction, which decreases significantly, must be transmitted. However, OSGA works poorly when there are fewer sensors, so it is not good method finding correlated signals. The OSGA is described in more details in **Error! Reference source not found.**.

The one-step greedy algorithm (OSGA):

1. Make greedy choice: Given all of the measurements, compute the test statistics

$$\varepsilon_n = \frac{1}{J}\sum_{j=1}^{J}\left\langle y_j, \phi_{j,n}\right\rangle^2$$

for $n \in \{1,2,...,N\}$ and estimate the common coefficient support set by

$\hat{\Omega} = \{n$ having one of the k largest $\varepsilon_n\}$.

Table 9. The one-step greedy algorithm (OSGA).

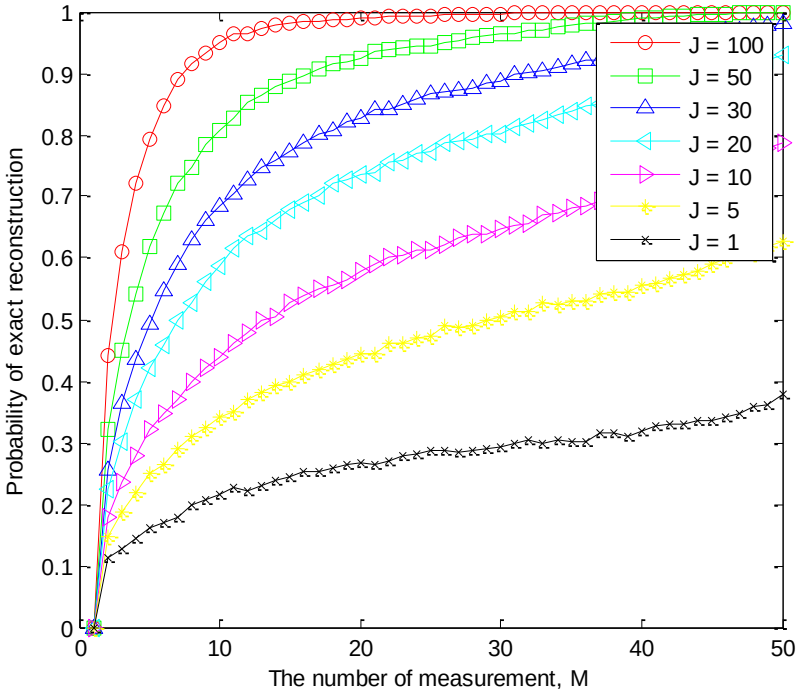
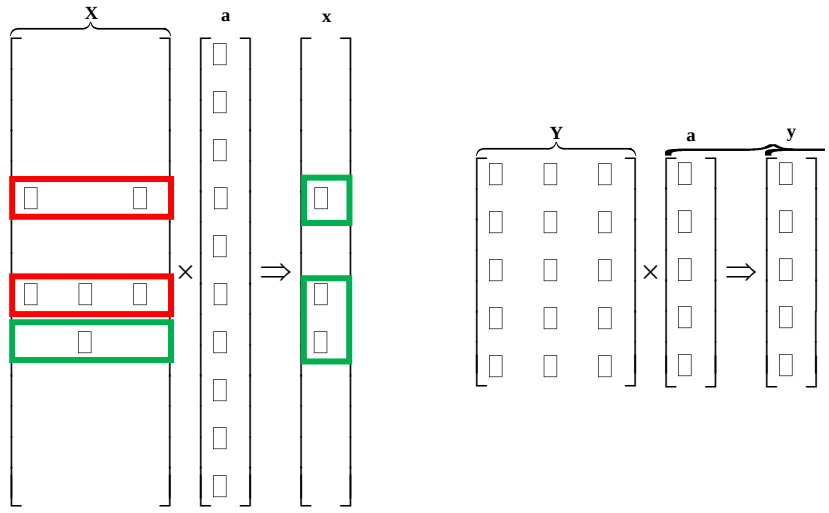


Figure 21. Reconstruction using OSGA for **JSM-2**. Experimental probability of success in recovering the support set in **JSM-2**. Signal length and sparsity are $n = 50$ and $k = 5$, respectively.

2) Reduced and boost algorithm

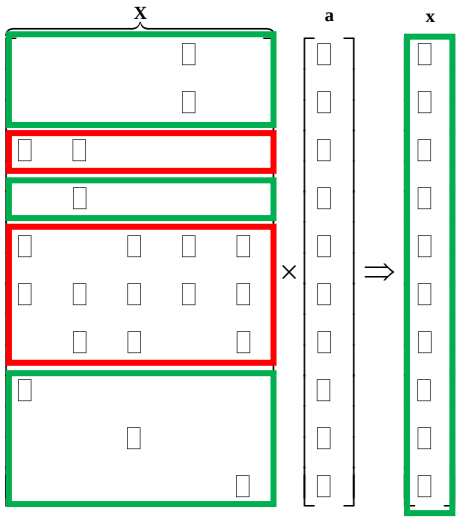
The Reduced and boost algorithm (ReMBo) is introduced in [Ref]. They reduce the correlated signal matrix $\mathbf{X}$ to one column signal and then solve reduced SMV problem by using greedy or gradient algorithm. The algorithm is summarized in Appendix 7.4. To get solution of MMV, ReMBo makes $\mathbf{Y} = \mathbf{AX}$ to $\mathbf{y} = \mathbf{Ax}$ by multiplying randomly generated vector $\mathbf{a}$ as **Figure 22**.



□ : Different value

Figure 22. Reduce MMV to SMV in ReMBo

The reduced SMV can be solved by using any one of SMV algorithm and then ReMBo algorithm saves the support location of SMV solution. From the information of support set, they can get the exact solution of matrix $\mathbf{X}$. This algorithm has easy stage for understanding and its algorithm speed is fast and effective. However, it is possible to apply only Eq. **Error! Reference source not found.** not **Error! Reference source not found.**, and if the original matrix $\mathbf{X}$ has much number of distributed innovation part as **Figure 23**, it cannot find solution. In the case of **Figure 23**, it makes the reduced signal $\mathbf{x}$ which is not sparse. The transformed equation $\mathbf{y} = \mathbf{A}\mathbf{x}$ from the signal of **Figure 23** cannot be solved. Therefore, it has limitations to apply various correlated signals.



□ : Different value

Figure 23. The limition of ReMBo algorithm

ii) Phased-OMP algorithm

Until now, we discussed various methods for recovering correlated signals. They are modified equation method, SOMP, OSGA, ReMBo. Those methods can apply only specific correlated signals as JSM-1, JSM-2 which have fixed pattern. If those methods are applied to other various signal models, it would not work properly. Thus, we proposed one method for any kind of correlated signals. It is called to phased-OMP (POMP). In this section, we explain how the algorithm works to recover various correlated signals. To help understanding algorithm, we will use the terminologies which are mentioned in previous section.

1) Basic idea of correlated signal recovery

We already talked about the unique solution of SMV problem before. The condition satisfied for solving SMV equation is $k < \frac{spark(A)}{2}$. It is proved in [Ref]. We used this proof as an idea for making

our proposed algorithm. If the total sparsity made from matrix X satisfies $T < \frac{spark(\mathbf{A})}{2}$, it guarantees each column has unique solution. Otherwise, even though each column satisfies $k < \frac{spark(A)}{2}$, it doesn't mean that $T < \frac{spark(\mathbf{A})}{2}$ is satisfied due to distributed innovation part

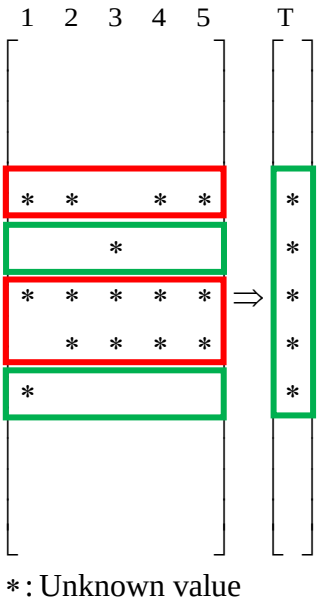


Figure 24. Total sparsity of MMV equation

Consider this specific signal which has only common sparsity and its total sparsity satisfy $T < \frac{spark(\mathbf{A})}{2}$. Because each column in MMV also satisfies $k < \frac{spark(A)}{2}$, and then we can get unique solution and its support location by using separate decoding. From the support set information of first SMV problem, we can also solve next SMV problem easily by using pseudo-inverse, if the next support set also has same support set. Therefore, it is important to know common part information since exact common part can reduce calculation time and complexity.

The idea for proposed algorithm follows. Common sparsity is a correlated part. Therefore, if we find the location of common part, we can also use it to solve another SMV. Innovation sparsity is an uncorrelated part. Therefore, even if we know the location of innovation part, we cannot use it to recover the innovation of other signals. In SMV problem, if we want to find the unique solution, each SMV should satisfy the unique condition. However, if we use correlated sparsity information in MMV equation, each SMV problem can get more guarantee for exact solution by using correlated information.

To use correlated information in POMP, we will find the common support location at first by using joint decoding. Then, by using separate decoding, we will find the remaining support location for each SMV problem. Therefore, we use the specific characters of both the joint decoding and separate decoding for effective reconstruction. Although the proposed algorithm works with easily understanding, its performance is better than previous methods (Modified equation method, POMP, SOMP) for correlated signal reconstruction. In addition, it doesn't be related with the number of total sparsity.

2) Pseudo-code of POMP

POMP uses the method which finds support set at every iteration. It is similar with the OMP method about finding support set, but we applied many ideas different with original OMP. We illustrate the pseudo code of POMP algorithm as following table.

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 The estimate number of common sparisty C Stop conditon ε	An estimate $\hat{\mathbf{x}}_j$ in $\mathbf{R}^n$ for the ideal signal. A set $\Lambda_{j,k}$ 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 POMP algorithm:

Phase 1: For find common sparsity

1. Initialize:

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

2. Find the common sparsity index $\lambda_{j,t}$ for each j:

$\lambda_{j,t} = \arg\max_{i=1,...,n} \sum_{j=1}^{J} \left| \left\langle \mathbf{r}_{j,t-1}, \mathbf{a}_{j,i} \right\rangle \right|$. The $\mathbf{a}_{j,i}$ is the i th column vector of matrix $\mathbf{A}_j$.

3. Update set:

$\Lambda_{j,t} = \Lambda_{j,t-1} \cup \{\lambda_{j,t}\}$.

4. Signal estimate:

$\mathbf{x}_{j,t}\left(\Lambda_{j,t}\right) = \mathbf{A}_{j,\Lambda_t}^{\dagger} \mathbf{y}_j$ and $\mathbf{x}_{j,t}\left(\Lambda_{j,t}^{C}\right) = \mathbf{0}$, where $\mathbf{x}_{j,t}\left(\Lambda_{j,t}\right)$ 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

if $t < C$ return to **Step 2 of Phase 1**

otherwise, $t > C$ go to **Phase 2**

Phase 2: For find innovation sparsity for each j

7. Find the index $\lambda_{j,t}$ **for each** j **:**

$\lambda_{j,t} = \arg\max_{i=1,\ldots,n} \left| \left\langle \mathbf{r}_{j,t-1}, \mathbf{a}_{j,i} \right\rangle \right|$ for every j. The $\mathbf{a}_{j,i}$ is the i th column vector of matrix $\mathbf{A}_j$.

8. Update set:

$\Lambda_{j,t} = \Lambda_{j,t-1} \cup \{\lambda_{j,t}\}$ for each j.

9. Signal estimate:

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

10. Get new residual:

$\hat{\mathbf{y}}_{j,t} = \mathbf{A}_{j,t}\mathbf{x}_{j,t}$, $\mathbf{r}_{j,t} = \mathbf{y}_j - \hat{\mathbf{y}}_{j,t}$.

11. Increment t **:**

Increase iteration number $t = t+1$, and

return to **Step 7 of phase 2** if $\sum_{j=1}^{J} \left\| \mathbf{y}_j - \mathbf{A}_j \mathbf{x}_j \right\|_2 > \varepsilon$

otherwise stop the algorithm.

Table 5. POMP algorithm.

For **Phase 1**, it is a stage for finding common sparsity Because the common sparsity is the correlated part of the signal matrix $\mathbf{X}$, we use joint decoding method for finding the location of common part. The joint decoding method is able to find support location successfully. We already knew the advantages of joint decoding from the comparison of joint decoding and separate decoding in **Section 5.1**. Therefore, if it is possible to use joint decoding for MMV equation, we should use it for advantages about signal reconstruction. It results in better performance for solving MMV equation. According to pseudo-code of POMP, it finds the location of common part in **Phase 1** and memorizes the index as **Figure 26**. After the stage of **Phase 1** is finished, POMP algorithm tries to find the

remaining support set by separate decoding. Due to separate decoding of **Phase 2** for remaining support set, POMP can find the missed common sparsity in previous stage.

We draw **Figure 26** which expresses the movement of POMP algorithm. The nonzero values in red box are common sparsity which is exploited in **Phase 1** and then the remaining nonzero values in green box can be exploited in **Phase 2**. The index of row having nonzero values is added to Λ at every iteration until the criteria $\sum_{j=1}^{J}\left\|\mathbf{y}_j - \mathbf{A}_j\mathbf{x}_j\right\|_2 \le \varepsilon$ is satisfied. After finishing the movement of POMP, we can get the original solution by using pseudo-inverse based on the estimated support set.

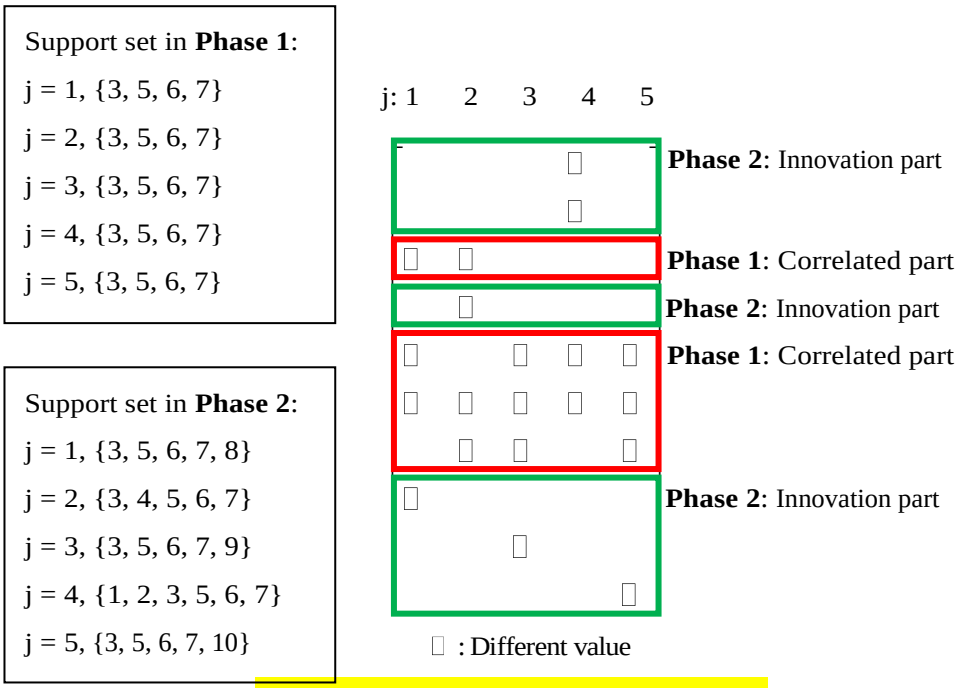


Figure 25. The movement of POMP algorithm

Moore-Penrose pseudo inverse:

If we define that Λ_j is the support set of the j th column in matrix $\mathbf{X}$, we can reduce the sensing matrix $\mathbf{A}_j$ to $\mathbf{A}_{\Lambda_j}$ corresponding to the nonzero elements of $\mathbf{x}_j$. If the columns of the reduced matrix $\mathbf{A}_{\Lambda_j}$ are linearly independent, Moore-Penrose pseudo inverse equation is accepted.

$$\left(\mathbf{A}_{\Lambda_j}\right)^{\dagger}\mathbf{A}_{\Lambda_j}=\mathbf{I}\text{, where }\ \mathbf{A}_{\Lambda_j}^{\dagger}=\left(\mathbf{A}_{\Lambda_j}^{T}\mathbf{A}_{\Lambda_j}\right)^{-1}\mathbf{A}_{\Lambda_j}^{T}$$

Therefore, if we know the support set and the reduced matrix $\mathbf{A}_{\Lambda_j}$ are linearly independent, then the original signal $\mathbf{x}_j$ can be found by using pseudo-inverse.

$$\mathbf{x}_{\Lambda_j}=\left(\mathbf{A}_{\Lambda_j}^{T}\mathbf{A}_{\Lambda_j}\right)^{-1}\mathbf{A}_{\Lambda_j}^{T}\mathbf{y}$$

Table 9. **Moore-Penrose** pseudo inverse.

5.3 The properties of POMP algorithm

i) The advantages of using prior correlation information

If we know the prior information of correlated signal like the number of common sparstiy, innovation sparsity, or the distribution of support location, we can use that information for signal recovery. If we know the number of common sparsity as prior information, we can choice parameter C as the number of iteration used for finding common part exactly. To select the number of iteration exactly in POMP affects the reconstruction performance as **Figure 25**. The parameters of simulation are $N=150$, $C=10$, $I=10$. Even though the value of estimated C is not exact correct as red and or blue, its performance is stable. However, it requires much number of measurements for perfect signal recovery.

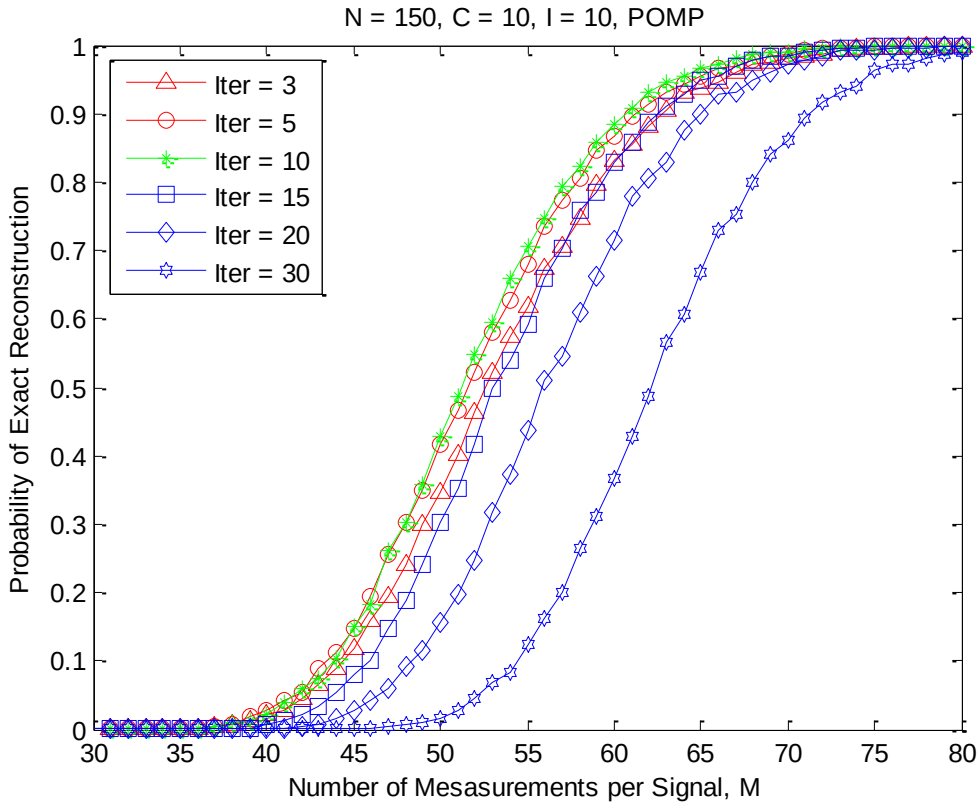


Figure 25. The advantages of using prior information

ii) The complexity of POMP

1) The complexity of Moore-Penrose pseudo inverse

We express how POMP works for MMV equation throughout pseudo-code. According to POMP algorithm, it requires pseudo-inverse calculation to get unique solution when we know the support set of original signal. In general, the calculation of matrix multiplication and inverse matrix require high complexity respectively $O\left(n^2\right)$ and $O\left(n^3\right)$ respectively. However, we already know that the number of sparsity k is very short in comparison with the length of signal n as $k << m < n$ so, the complexity of those calculations is simple as below **Table**.

The complexity of pseudo-inverse:
By using pseudo-inverse, we can get the nonzero values corresponding to support set. The pseudo inverse complexity is not high. From the relation $k << m < n$, we can get the complexity.
$\underset{k\times m\ m\times k}{\left(A_S^T A_S\right)^{-1}} \underset{k\times m\ m\times 1}{A_S^T y} = x_S$
Multiplication: $k\times m\times k + k\times k\times m + k\times m\times 1 \approx 2k^2 m + km \approx O\left(mk^2\right)$
Inverse: $O\left(k^3\right)$. Therefore, total:$O\left(k^3\right)+O\left(mk^2\right)\approx O\left(mk^2\right)$ $\because k << m$

Table 9. Pseudo-inverse in MMV.

2) The complexity of POMP algorithm in terms of sparsity.

POMP algorithm has two stages for find support location which consists of common sparsity and innovation sparsity. In this section, we analyze the complexity of POMP algorithm related with sparsity and the number of sensors J . We assumed that the observed signal has both the C number of common sparstiy and the I number of innovation sparsity. In **Phase 1**, it will find the C number of sparsity and it requires such as calculations, for examples $\lambda_{j,t} = \underset{i=1,\ldots,n}{\arg\max} \sum_{j=1}^{J} \left|\left\langle \mathbf{r}_{j,t-1}, \mathbf{a}_{j,i} \right\rangle\right|$ and

$\mathbf{x}_{j,t}\left(\Lambda_{j,t}\right)=\mathbf{A}_{j,\Lambda_t}^{\dagger}\mathbf{y}_j$. After finishing the calculations of **Phase 1**, it starts the calculation of **Phase 2** for finding innovation sparsity. We consider $\lambda_{j,t}=\arg\max_{i=1,\ldots,n}\left|\left\langle \mathbf{r}_{j,t-1},\mathbf{a}_{j,i}\right\rangle\right|$ and $\mathbf{x}_{j,t}\left(\Lambda_{j,t}\right)=\mathbf{A}_{j,\Lambda_t}^{\dagger}\mathbf{y}_j$ in **Phase 2**. The complexity of POMP algorithm is below.

The complexity of POMP

Phase 1

Considered parameter: signal length n, measurement m, sparsity k, the number of sensors J, common sparstiy C, innovation sparsity I. We already know the relationship $(C\approx I)<k<<m<n$

1) $\lambda_{j,t}=\arg\max_{i=1,\ldots,n}\underbrace{\sum_{j=1}^{J}\left|\left\langle \underset{1\times m}{\mathbf{r}_{j,t-1}},\underset{m\times 1}{\mathbf{a}_{j,i}}\right\rangle\right|}_{J\text{ times summation}}$

Inner product and sigma summation: $O(m)+O(J)\approx O(m)$ ∵ In general $J<m$

By $J\times n$ iteration: $JnO(m)=O(Jnm)$

2) $\mathbf{x}_{j,t}\left(\Lambda_{j,t}\right)=\mathbf{A}_{j,\Lambda_t}^{\dagger}\mathbf{y}_j$

Pseudo-inverse: $O\left(mk^2\right)$

3) The number of iteration: C

Therefore, $C\left(O(Jmn)+O\left(mk^2\right)\right)\approx O(CJmn)$ ∵ In general $k^2<n$

In conclusion, the complexity of **Phase 1** is $O(CJmn)$

Phase 2

1) $\lambda_{j,t}=\arg\max_{i=1,\ldots,n}\left|\left\langle \mathbf{r}_{j,t-1},\mathbf{a}_{j,i}\right\rangle\right|$

Inner product : $O(m)$

By $J\times n$ iteration: $JnO(m)=O(Jnm)$

2) $\mathbf{x}_{j,t}\left(\Lambda_{j,t}\right)=\mathbf{A}_{j,\Lambda_t}^{\dagger}\mathbf{y}_j$

Pseudo-inverse: $O(mk^2)$

3) The number of iteration: I

Therefore, $I\left(O(Jmn)+O(mk^2)\right)\approx O(IJmn)$ $\because$ In general $k^2 < n$

In conclusion, the complexity of **Phase 2** is $O(IJmn)$

By **Phase 1** + **Phase 2**,

In conclusion, the complexity of POMP is $O(CJmn)+O(IJmn)=O\left(Jmn(C+I)\right)\approx O(Jmnk)$.

The complexity of POMP algorithm is affected by the parameters J, m, n, k.

Table 9. The complexity of POMP algorithm

iii) The recovery condition of POMP

6. Performance evaluation

In previous section, we already discuss the various correlated signals which are handled with references. For distinguishing those correlated signals, we named it correlated signal model (CSM) as following **Figure 27** and then we solved MMV equation **Error! Reference source not found.** and **Error! Reference source not found.** by using algorithms like modified equation method (MEM), SOMP, ReMBo, POMP. All of the algorithms are handled in previous section.

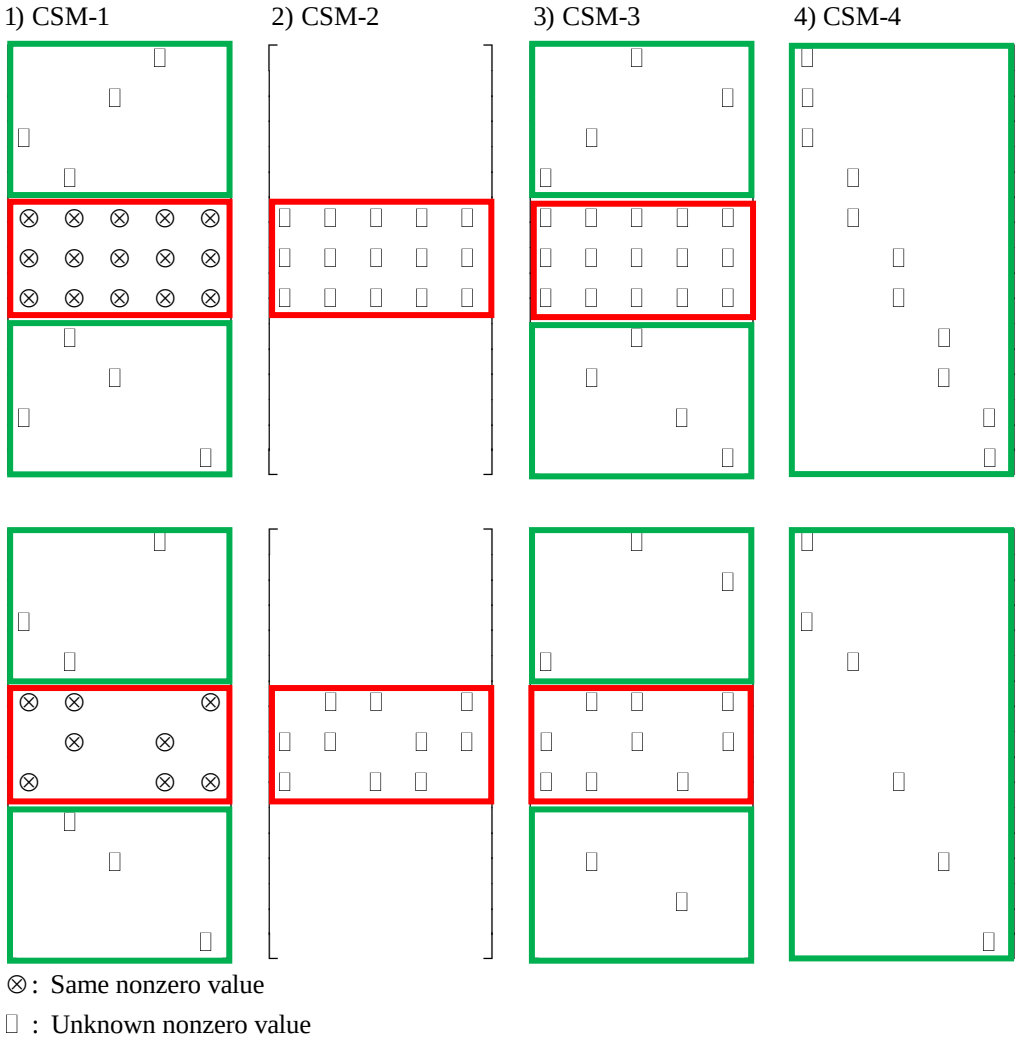
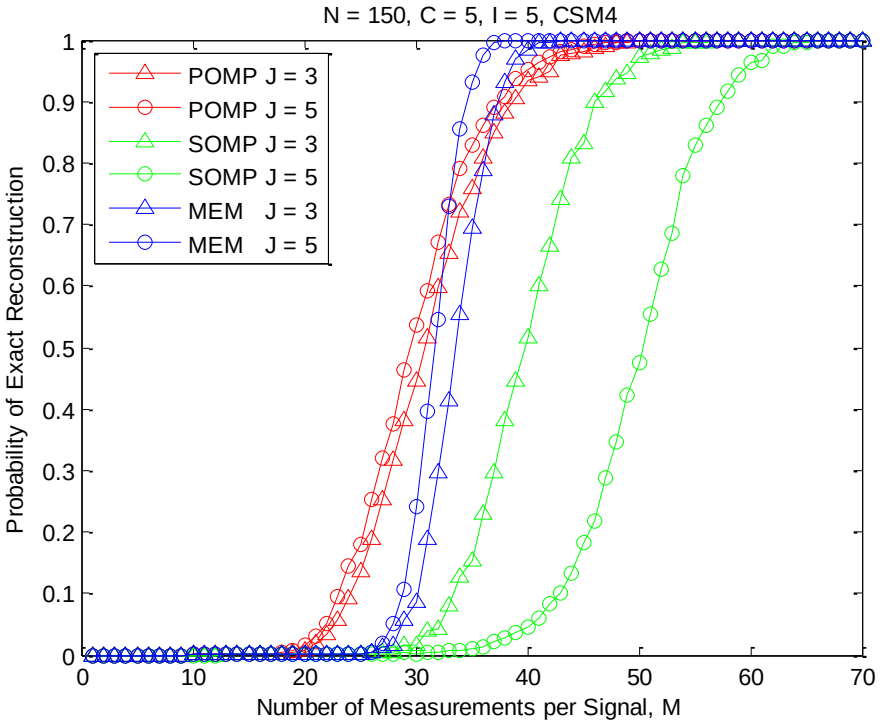


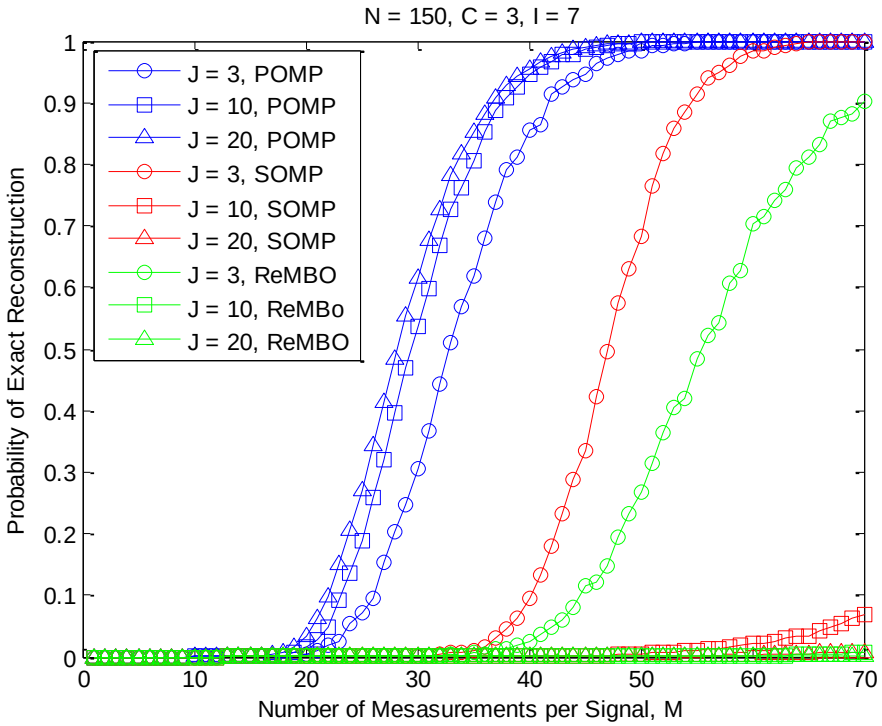
Figure 27. Correlated signal model (CSM)

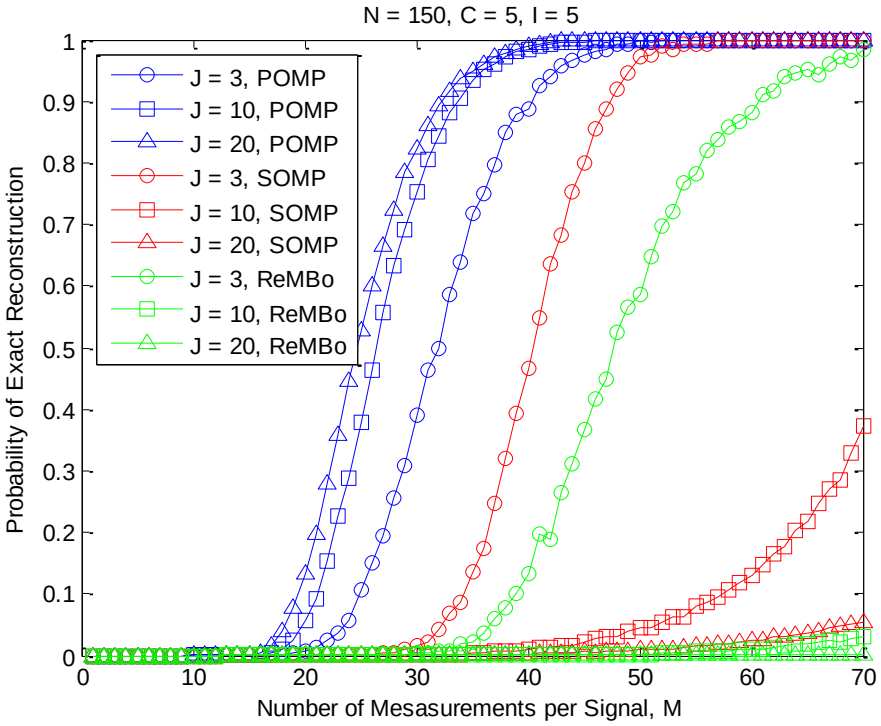
In **Figure 27**, we define correlated signal model which has various kinds of pattern. Most of the signal pattern can be defined by our CSM. CSM-1 and CSM-2 is similar with JSM-1 and JSM-2 which are handled in [Ref] but we consider the case that has some vacancies in common part and innovation part is also same signal model and we changed the definition of innovation part. In our case, innovation sparsity exist only one nonzero value in same row in different with [Ref]. CSM-1 and CSM-3 have common part and innovation part together but CSM-1 has same value for common part. CSM-2 has only common part and CSM-4 has only innovation part. Other case which does not exist in **Figure 27** will not be considered in this paper. Now, we simulate the performance POMP algorithm compared with other methods as ReMbo, SOMP, MEM.

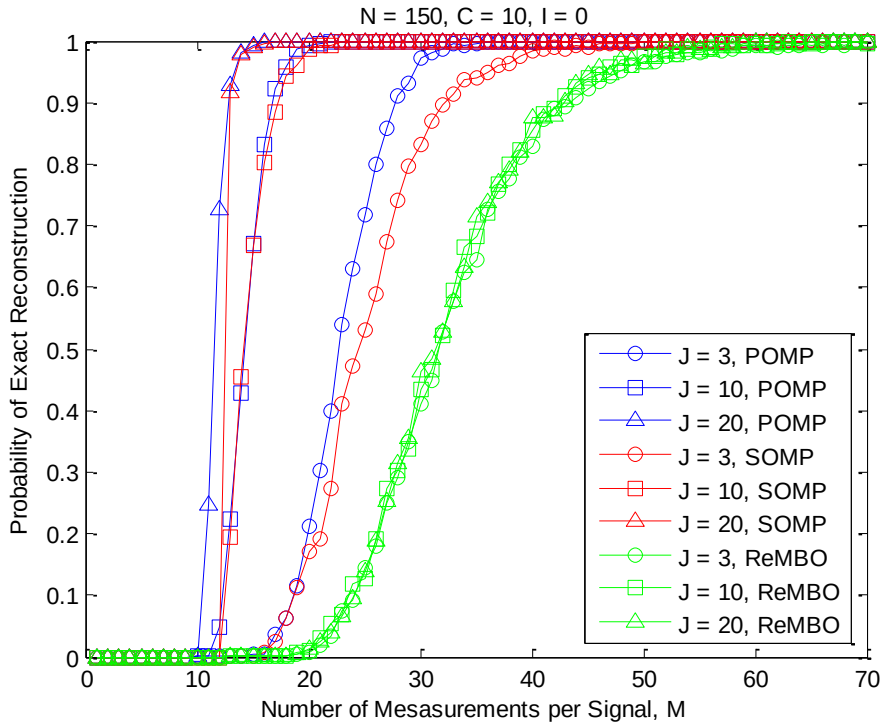
1) CSM-1 and Different matrix **A**

We generated the CSM-1 which has common and innovation part. Its common part has same values. We observed the reconstruction performance for MEM, SOMP, and POMP algorithm when the signal has CSM-1 pattern and each sensing matrix is different.


N = 150, C = 5, I = 5, CSM4
POMP J = 3
POMP J = 5
SOMP J = 3
SOMP J = 5
MEM J = 3
MEM J = 5
Probability of Exact Reconstruction
Number of Mesasurements per Signal, M
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
10
20
30
40
50
60
70


N = 150, C = 3, I = 7
J = 3, POMP
J = 10, POMP
J = 20, POMP
J = 3, SOMP
J = 10, SOMP
J = 20, SOMP
J = 3, ReMBO
J = 10, ReMBo
J = 20, ReMBO
Probability of Exact Reconstruction
Number of Mesasurements per Signal, M
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
0
10
20
30
40
50
60
70


N = 150, C = 5, I = 5
Probability of Exact Reconstruction
Number of Mesasurements per Signal, M
J = 3, POMP
J = 10, POMP
J = 20, POMP
J = 3, SOMP
J = 10, SOMP
J = 20, SOMP
J = 3, ReMBo
J = 10, ReMBo
J = 20, ReMBo


N = 150, C = 10, I = 0
Probability of Exact Reconstruction
Number of Mesasurements per Signal, M
J = 3, POMP
J = 10, POMP
J = 20, POMP
J = 3, SOMP
J = 10, SOMP
J = 20, SOMP
J = 3, ReMBO
J = 10, ReMBO
J = 20, ReMBO

7. Conclusion

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 2.1)

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 4.1. 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 4.2 and 4.3.

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

Error! Reference source not found.. The other is a gradient type for which we used the primal-dual interior point (PDIP) method, in Section **Error! Reference source not found.**.

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. **Error! Reference source not found.** and **Error! Reference source not found.**. 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 **Error! Reference source not found.**)

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.

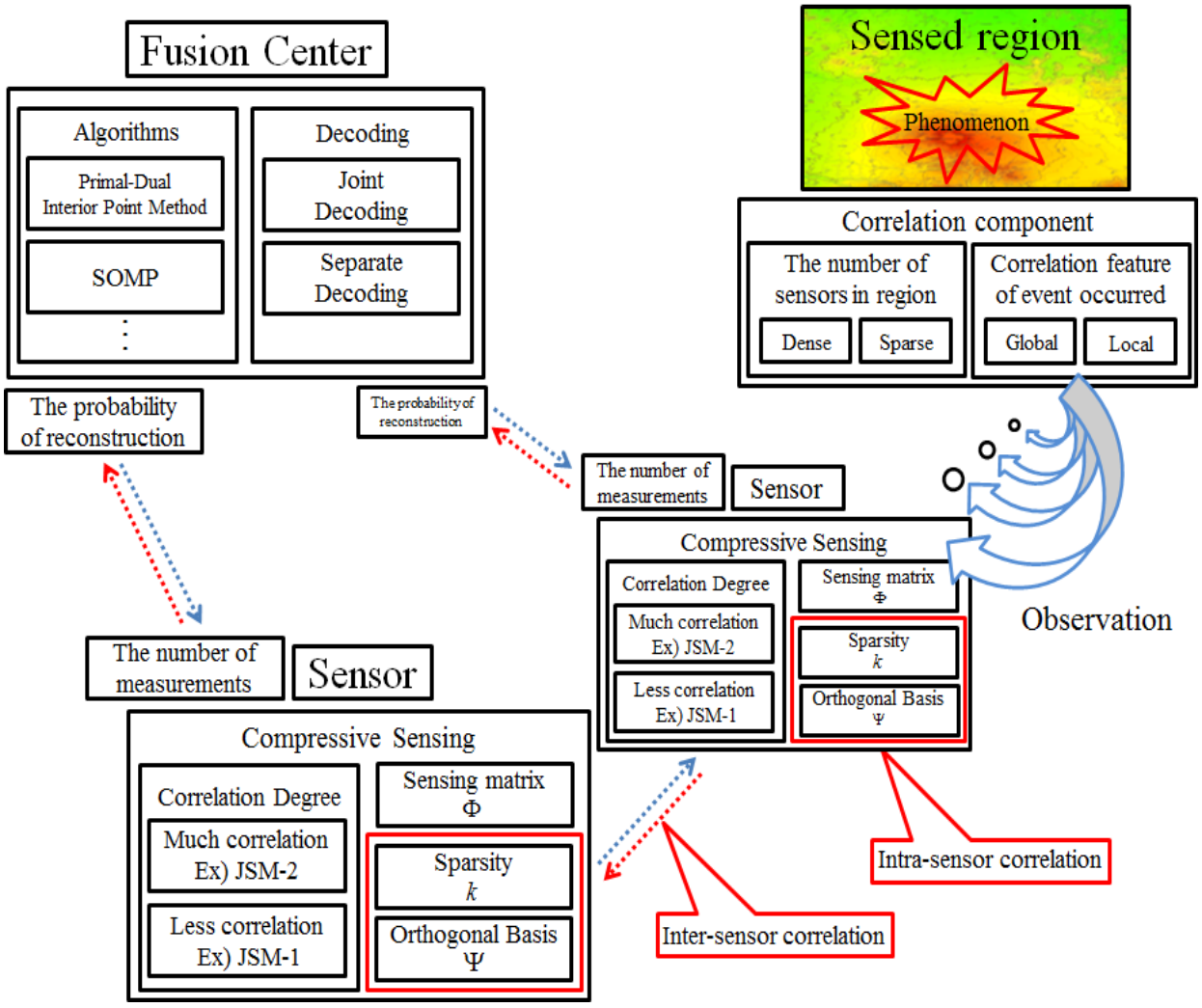


Figure 16. Summary of CS application in WSN

8. Appendix

8.1. Primal-dual interior point method (PDIP)

The L_1 minimization in Eq. **Error! Reference source not found.** can be recast as linear programming. Here we examine this relationship. Clearly, the L_1 minimization problem in Eq. **Error! Reference source not found.** 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

$$\begin{aligned} &\min_{(x,u)} \sum_i u_i \\ &\text{subject to} \\ &\forall_i \left| x(i) \right| \le u_i \\ &\mathbf{Ax} = \mathbf{b} \end{aligned} \tag{24}$$

The solution of the above equation is equal to the solution of the L_1 minimization problem. Many approaches to solving Eq. **Error! Reference source not found.** 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. **Error! Reference source not found.**, as follows:

$$L(\mathbf{t}, \boldsymbol{\lambda}, \mathbf{v}) = \begin{bmatrix} \mathbf{0}_1^{\mathrm{T}} & \mathbf{1}^{\mathrm{T}} \end{bmatrix} \mathbf{t} + \mathbf{v}^{\mathrm{T}} \left(\begin{bmatrix} \mathbf{A} & \mathbf{0}_2 \end{bmatrix} \mathbf{t} - \mathbf{b} \right) + \boldsymbol{\lambda}^{\mathrm{T}} \left(\begin{bmatrix} \mathbf{e} & -\mathbf{e} \\ -\mathbf{e} & -\mathbf{e} \end{bmatrix} \mathbf{t} \right) \tag{25}$$

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 $\boldsymbol{\lambda}\in\mathbf{R}^{2n\times 1}\geq 0$. From the Lagrangian function, we have several KKT conditions,

$$\begin{aligned}
&\begin{bmatrix}\mathbf{0}\\ \mathbf{1}\end{bmatrix}+\begin{bmatrix}\mathbf{A}^{\mathrm{T}}\\ \mathbf{0}^{\mathrm{T}}\end{bmatrix}\mathbf{v}^{*}+\begin{bmatrix}\mathbf{e} & -\mathbf{e}\\ -\mathbf{e} & -\mathbf{e}\end{bmatrix}\boldsymbol{\lambda}^{*}=\mathbf{0}_3\\
&\begin{bmatrix}\mathbf{A} & \mathbf{0}_2\end{bmatrix}\mathbf{t}^{*}-\mathbf{b}=\mathbf{0}_4\\
&\begin{bmatrix}\mathbf{e} & -\mathbf{e}\\ -\mathbf{e} & -\mathbf{e}\end{bmatrix}\mathbf{t}^{*}\leq\mathbf{0}_1\\
&\left(\boldsymbol{\lambda}^{*}\right)^{\mathrm{T}}\begin{bmatrix}\mathbf{e} & -\mathbf{e}\\ -\mathbf{e} & -\mathbf{e}\end{bmatrix}\mathbf{t}^{*}=0,\boldsymbol{\lambda}^{*}\geq\mathbf{0}_3
\end{aligned}\qquad(26)$$

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 $\mathrm{F}\left(\mathbf{t},\boldsymbol{\lambda},\mathbf{v}\right):\mathbf{R}^{(2n+m)\times 1}\rightarrow\mathbf{R}^{(2n+m)\times 1}$, which is

$$\mathrm{F}\left(\mathbf{t},\boldsymbol{\lambda},\mathbf{v}\right)=\begin{bmatrix}\begin{bmatrix}\mathbf{0}\\ \mathbf{1}\end{bmatrix}+\begin{bmatrix}\mathbf{A}^{\mathrm{T}}\\ \mathbf{0}^{\mathrm{T}}\end{bmatrix}\mathbf{v}+\begin{bmatrix}\mathbf{e} & -\mathbf{e}\\ -\mathbf{e} & -\mathbf{e}\end{bmatrix}\boldsymbol{\lambda}\\ \left(\boldsymbol{\lambda}^{*}\right)^{\mathrm{T}}\begin{bmatrix}\mathbf{e} & -\mathbf{e}\\ -\mathbf{e} & -\mathbf{e}\end{bmatrix}\mathbf{t}^{*}\\ \begin{bmatrix}\mathbf{A} & \mathbf{0}_2\end{bmatrix}\mathbf{t}-\mathbf{b}\end{bmatrix}=\mathbf{0}_4\in\mathbf{R}^{(2n+m)\times 1},\begin{bmatrix}\mathbf{e} & -\mathbf{e}\\ -\mathbf{e} & -\mathbf{e}\end{bmatrix}\mathbf{t}^{*}\leq\mathbf{0}_1,\boldsymbol{\lambda}^{*}\geq\mathbf{0}_3\qquad(27)$$

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

$$\mathrm{F}(\mathbf{t}+\Delta\mathbf{t}, \boldsymbol{\lambda}+\Delta\boldsymbol{\lambda}, \mathbf{v}+\Delta\mathbf{v}) \approx \mathrm{F}(\mathbf{t}, \boldsymbol{\lambda}, \mathbf{v}) + \nabla_{(\mathbf{t},\boldsymbol{\lambda},\mathbf{v})}\mathrm{F}(\mathbf{t}, \boldsymbol{\lambda}, \mathbf{v}) \begin{bmatrix} \Delta\mathbf{t} \\ \Delta\mathbf{v} \\ \Delta\boldsymbol{\lambda} \end{bmatrix} \tag{28}$$

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 $\mathbf{A}^{\dagger} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ is the Moore-Penrose pseudo-inverse of $\mathbf{A}$ and $\mathbf{A}^T$ denotes the transpose of $\mathbf{A}$.)

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

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

3. Find the proper step length:

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

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.

8.2 Orthogonal matching pursuit (OMP)

The orthogonal matching pursuit (OMP) is a famous greedy-type algorithm **Error! Reference source not found.**. 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 $\mathbf{R}^n$ 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 : $\lambda_t = \arg\max_{i=1,...,n} \left\| \langle \mathbf{r}_{t-1}, \mathbf{a}_i \rangle \right\|$. 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 $\mathbf{x}_t(\Lambda_t^C) = \mathbf{0}$, 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].

8.3 Simultaneous orthogonal matching pursuit (SOMP)

We introduce another greedy-type algorithm based on OMP as an example: simultaneous orthogonal matching pursuit (SOMP) **Error! Reference source not found.**. 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 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 $\mathbf{R}^n$ 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.

8.4 Reduce and boost (ReMBo)

ReMBo algorithm is for recovering correlated signals. The authors in [Ref] insisted that the algorithm improves the recovery probability of any suboptimal methods for signal matrix $\mathbf{X}$. Its idea is simple and effective. They transformed the matrix $\mathbf{X}$ to a single vector $\mathbf{x}$ and do $\mathbf{Y}$ to a single measurement vector $\mathbf{y}$. After modifying MMV equation to SMV, they apply any algorithm for SMV. We attached ReMBo algorithm from [Ref].

Input	Output
A $m \times n$ measurement matrix $\mathbf{A}_j$	An estimate $\hat{\mathbf{x}}_j$ in $\mathbf{R}^n$ for the ideal signal.
A $m-$dimensional data vector $\mathbf{y}_j$	Support set $\hat{S}$
The sparsity level k of the ideal signal	flag

Table 4. Inputs and outputs of ReMBo algorithm.

The ReMBo algorithm:

Control parameters : k **,** ε **,** Maxiters

1. Initialize:

Set iter = 1, flag = false.

2. while (iter $\leq$ Maxiter) and (flag is false) **do**

Draw a random vector $\mathbf{a}$ of length j according to randomly generated distribution.

$\mathbf{y} = \mathbf{Aa}$

Solve $\mathbf{y} = \mathbf{Ax}$ using SMV algorithm and save the solution $\mathbf{x}$.

$\hat{S} = I(\mathbf{x})$

If $\left(\left|\hat{S}\right| \leq K\right)$ and $\left(\|\mathbf{y} - \mathbf{Ax}\|_2 \leq \varepsilon\right)$ **then**

flag = true

else

flag = false

end if

Construct $\mathbf{X}$ using $\hat{S}$ and pseudo inverse

iter = iter + 1

end while

return $\mathbf{X}$, $\hat{S}$, flag

Table 5. ReMBo algorithm.

9. Reference

[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] Jie Chen and Xiaoming Huo, “Theoretical results on sparse representations of multiple measurement vectors”, *IEEE Signal Process.*, pp. 4634-4643, Dec. 2006.

[8] J. Solobera, “Detecting forest fires using wireless sensor networks with Waspmote.”

[9] A. Hac, “*Wireless Sensor Network Designs*,” John Wiley & Sons, Ltd., 2003.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

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