Projection-Based and Look Ahead Strategies for Atom Selection

Saikat Chatterjee, Dennis Sundman, Mikko Vehkaperä, Mikael Skoglund

Abstract— **In this paper, we improve iterative greedy search algorithms in which atoms are selected serially over iterations, i.e., one-by-one over iterations. For serial atom selection, we devise two new schemes to select an atom from a set of potential atoms in each iteration. The two new schemes lead to two new algorithms. For both the algorithms, in each iteration, the set of potential atoms is found using a standard matched filter. In case of the first scheme, we propose an orthogonal projection strategy that selects an atom from the set of potential atoms. Then, for the second scheme, we propose a look ahead strategy such that the selection of an atom in the current iteration has an effect on the future iterations. The use of look ahead strategy requires a higher computational resource. To achieve a trade-off between performance and complexity, we use the two new schemes in cascade and develop a third new algorithm. Through experimental evaluations, we compare the proposed algorithms with existing greedy search and convex relaxation algorithms.**

Index Terms— **Sparse signal estimation, compressive sampling, orthogonal matching pursuit, orthogonal least squares, subspace pursuit.**

I. INTRODUCTION

A linear underdetermined system model based sparse signal estimation problem has attracted much attention in the current literature. The sparse signal estimation problem has many applications, for example, in linear regression [1], communication [2]-[4], multimedia [5]-[8], and recently in compressive sampling (CS) [9]-[10]. The algorithms proposed for solving such a problem may be categorized into three broad classes: convex relaxation, [9]-[18], Bayesian inference [20]-[24], and iterative greedy search [25]-[41]. The iterative greedy search (IGS) algorithms are of rising interest for solving large dimensional sparse signal estimation problems due to their algorithmic simplicity and lower complexity.

Generally an IGS algorithm constructs a support set of the underlying sparse signal vector through iterations. In the literature, the support set refers to the set of indices corresponding to the coordinates of non-zero elements of a sparse signal vector. Also the columns of the measurement matrix (or dictionary matrix) are referred to as atoms. Naturally, the support set's elements are the indices of atoms associated with the non-zero elements of the sparse vector. In the underdetermined setup, an atom-selection process for constructing the support set is a critical task. Depending on atom selection approaches, the IGS algorithms may be categorized into two classes:

serial (or sequential) and parallel (or simultaneous). Prominent examples of the serial atom selection based IGS algorithms are orthogonal matching pursuit (OMP) [26]-[27] and orthogonal least squares (OLS) [28]-[30]. For both the algorithms, a single atom is selected from a set of potential atoms in each iteration and then the atom's index is inducted into the support set under progressive construction. Therefore, the cardinality of the support set under construction is increased one-by-one through iterations serially[1]. At the end of all iterations, the algorithms provide a support set of full cardinality. On the other hand, for the parallel atom selection based IGS algorithms, such as subspace pursuit (SP) [35] and CoSaMP [37], a support set with a fixed (pre-determined) cardinality is estimated in each iteration[2]. A fixed number of atoms is selected simultaneously in each iteration to form the support set of full cardinality and the estimate of the underlying support set is refined through iterations. In practice, IGS algorithms provide different support set estimation performances (in turn different sparse signal estimation performances). For selecting (or detecting) an atom in a computationally efficient manner, the use of a matched filter remains prevalent in existing IGS algorithms.

In this paper, we develop new IGS algorithms to provide a better estimate of the underlying support set at the expense of computational complexity. Our interest is to deal with a highly underdetermined system that may arise in many applications, for example in a highly under-sampled CS. For such an underdetermined system, the reliability of the matched filter degrades. At the expense of complexity, we endeavor for improving atom selection performance beyond the scope of a matched filter. In the new algorithms, a trade-off between complexity and performance can be achieved by adjusting user defined tunable parameters.

We develop the new algorithms using the architectures of OMP and OLS algorithms where atom selection is performed serially over iterations. For the new algorithms, in each iteration, a set of potential atoms is chosen and then a single atom is finally selected from the set of potential atoms. We propose new atom selection strategies and incorporate them in the OMP and OLS architectures. Strategically, the new algorithms use a two stage mechanism: (a) first, the standard matched filter is used to choose a set of potential atoms from all the competing atoms, and (b) second, an atom is finally selected from the set

The authors are with Communication Theory Laboratory, School of Electrical Engineering, KTH - Royal Institute of Technology, Sweden. Emails: {sach, denniss, mikkov, skoglund}@kth.se

[1]Whilst both the OMP and OLS algorithms are structurally similar, there exists a subtle difference between them in the way an atom is selected in each iteration. A clear exposition on the difference between these two algorithms can be found in [31]. The reader is encouraged to see Appendix A.

[2]We briefly describe serial atom selection based OMP and OLS algorithms in Appendix A and parallel atom selection based SP in Appendix B.

of potential atoms by using the new selection strategies. In the first stage, the cardinality of the set of potential atoms is pre-determined. The use of matched filter in the first stage helps to reduce search space for the new selection strategies invoked in the second stage.

For the standard OMP, in each iteration, the matched filter is used to select an atom from all the competing atoms. In a highly underdetermined setup, atoms are more correlated and hence the atom selection performance using matched filter degrades. This performance degradation is due to fact that an underlying atom is subjected to a considerable interference from other closely correlated atoms. To improve atom selection, we propose to use the same matched filter for choosing a set of potential atoms and then select an atom from the set using an *orthogonal projection* strategy. The new algorithm is referred to as the projection based OMP (POMP) where the use of orthogonal projection is motivated by its existing use in parallel atom selection based IGS algorithms (such as in SP and CoSaMP). The orthogonal projection invokes signal estimation in least-squares (LS) sense and hence its use adds more strength on the top of matched filter.

Next, for the standard OLS, an atom is selected in each iteration such that the selection leads to the minimum norm of a fitting residual (LS residual) for that iteration. The selection of an atom in the current iteration does not depend on the final performance of the OLS algorithm when all the iterations would have been finished. Therefore, the selection of a new atom in the current iteration is blind to its effect on the future iterations in the sense of minimizing the final fitting residual norm. To overcome the shortcoming of blindness in OLS, we propose to use a *look ahead* selection strategy [39]. In the current iteration, an atom is selected from the set of potential atoms by evaluating its effect on the final performance measure in the sense of minimizing the norm of fitting residual at the end of all future iterations. The new algorithm is referred to as look ahead OLS (LAOLS). Use of a look ahead strategy is shown to provide a better performance, but at the expense of a higher computation.

Further, we combine the orthogonal projection-based and look ahead atom selection strategies in cascade for developing an IGS algorithm which provides a trade-off between computational complexity and reconstruction performance.

Considering compressive sampling (CS) as the application, we experimentally evaluate the new algorithms and compare their performances vis-a-vis several existing algorithms, such as OMP, OLS, SP and convex relaxation algorithms. For experiments, we endeavor for CS reconstruction of Gaussian and binary sparse signals at varying under-sampling rates and measurement noise levels.

Notations: Let $\mathbf{A} \in \mathbb{R}^{M \times N}$, $\mathbf{x} \in \mathbb{R}^N$, and $\mathcal{I} \subset \{1, 2, \ldots, N\}$. We denote $|\mathcal{I}|$ and $\overline{\mathcal{I}}$ as the cardinality and complement of the set $\mathcal{I}$, respectively. The matrix $\mathbf{A}_{\mathcal{I}} \in \mathbb{R}^{M \times |\mathcal{I}|}$ consists of the columns of $\mathbf{A}$ indexed by $i \in \mathcal{I}$, and $\mathbf{x}_{\mathcal{I}} \in \mathbb{R}^{|\mathcal{I}|}$ is composed of the components of $\mathbf{x}$ indexed by $i \in \mathcal{I}$. Also we denote $(.)^t$ and $(.)^\dagger$ as transpose and pseudo-inverse, respectively.

A. *Preliminaries of Compressive Sampling*

Let us state the standard CS problem where we acquire a K-sparse signal $\mathbf{x} \in \mathbb{R}^N$ via the linear measurements

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}, \tag{1}$$

where $\mathbf{A} \in \mathbb{R}^{M \times N}$ is a matrix representing the sampling system, $\mathbf{y} \in \mathbb{R}^M$ represents a vector of measurements and $\mathbf{w} \in \mathbb{R}^M$ is additive noise representing measurement errors. A K-sparse signal vector consists of at most K non-zero scalar components. For the setup $K < M < N$ (underdetermined system of linear equations), the task is to reconstruct $\mathbf{x}$ from $\mathbf{y}$ as $\hat{\mathbf{x}}$. With no a-priori statistical knowledge of $\mathbf{x}$ and $\mathbf{w}$, the objective in CS is to strive for a reduced number of measurements (M) as well as achieving a good reconstruction quality. Note that, in practice, we may wish to acquire a signal $\mathbf{x}$ that is sparse in a known orthonormal basis and the concerned problem can be recast as (1).

For a sparse signal vector $\mathbf{x} = [x_1, x_2, \ldots, x_N]^t$, the support set $\mathcal{I} \subset \{1, 2, \ldots, N\}$ is defined as $\mathcal{I} = \{i : x_i \neq 0\}$. For a K-sparse vector $\mathbf{x} \in \mathbb{R}^N$, $|\mathcal{I}| = \|\mathbf{x}\|_0 \leq K$. In this paper, we assume that $|\mathcal{I}| = K$. Denoting the i'th column (atom) of the measurement matrix $\mathbf{A}$ as $\mathbf{a}_i$, note from (1) that

$$\mathbf{y} = \sum_{i \in \mathcal{I}} x_i \, \mathbf{a}_i + \mathbf{w} = \mathbf{A}_{\mathcal{I}} \mathbf{x}_{\mathcal{I}} + \mathbf{w}, \tag{2}$$

where $\mathbf{x}_{\mathcal{I}} \in \mathbb{R}^K$ and $\mathbf{A}_{\mathcal{I}} \in \mathbb{R}^{M \times K}$. From $\mathbf{y}$, if the underlying support set $\mathcal{I}$ can be estimated, then we can estimate the non-zero values of $\mathbf{x}$ (i.e. $\mathbf{x}_{\mathcal{I}}$) using a standard least squares (LS) solution (as $K < M$, we can use the pseudo-inverse). Therefore, a better estimate of the support set leads to a better reconstruction performance. However, if the matrix $\mathbf{A}_{\mathcal{I}}$ is not full column rank (or ill-conditioned) then the LS solution will be erroneous. This may happen in case of the matrices where the correlation between the columns are considerably high, for example, in a highly under-sampled CS where $M \ll N$. Recently, theoretical properties of such matrices are investigated in [42], [43].

Next we consider the issue of constructing the sensing matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$. While it is possible to obtain deterministic constructions of $\mathbf{A} = \{a_{i,j}\}$ holding a specific structure, at present the most efficient designs (i.e., those requiring minimum number of rows) rely on random matrix constructions where the $a_{i,j}$'s are assumed realizations of independent and identically distributed (i.i.d.) random variables. A standard method is to draw $a_{i,j}$'s independently from a Gaussian source (i.e., $a_{i,j} \sim \mathcal{N}\left(0, \frac{1}{M}\right)$) and then to scale the columns of $\mathbf{A}$ to unit-norm. Note that once $\mathbf{A}$ is constructed, it remains fixed and made known to a CS reconstruction algorithm. In the following sections, we develop the new IGS algorithms.

II. PROJECTION BASED ORTHOGONAL MATCHING PURSUIT

Using the algorithmic structure of OMP, we develop projection based OMP (POMP) where the selection of an atom in the current iteration is carried out using an orthogonal projection method.

For developing the POMP algorithm, we first need to describe the orthogonal projection based selection method in Algorithm 1. In Algorithm 1, the inputs are $\mathbf{A}$, $\mathbf{y}$, previous

---

Algorithm 1 : Projection-Based Atom-index Selection

Input:
1: $\mathbf{A}$, $\mathbf{y}$;
2: Previous support set: $\mathcal{I}$;
3: Set of potential atoms' indices: $\mathcal{I}_{(p)}$;

Assumption:
1: $\mathcal{I}_{(p)} \cap \mathcal{I} = \emptyset$, $\hat{\mathbf{x}} \in \mathbb{R}^N$;

Execution:
1: $\mathcal{I}_{(u)} \leftarrow \mathcal{I} \cup \mathcal{I}_{(p)}$;
2: $\hat{\mathbf{x}}_{\mathcal{I}_{(u)}} \leftarrow \mathbf{A}_{\mathcal{I}_{(u)}}^{\dagger} \mathbf{y}$;   (Orthogonal projection)
3: $\hat{\mathbf{x}}_{\overline{\mathcal{I}}_{(p)}} \leftarrow \mathbf{0}$;   (Zeroing the coordinates indexed by $\overline{\mathcal{I}}_{(p)}$)
4: $i \leftarrow$ index of the highest amplitude of $\hat{\mathbf{x}}$

Output:
1: The index i.   (Note: $i \in \mathcal{I}_{(p)}$)

---

support set $\mathcal{I}$ and the set of potential atoms' indices $\mathcal{I}_{(p)}$. We refer to the previous support set $\mathcal{I}$ as the constructed support set in the previous iteration. The $\mathcal{I}_{(p)}$ is the set of potential atoms chosen using the standard matched filter in the current iteration. In Algorithm 1, a dummy set is formed as $\mathcal{I}_{(u)} = \mathcal{I} \cup \mathcal{I}_{(p)}$. Then the measurement vector $\mathbf{y}$ is projected on $\mathbf{A}_{\mathcal{I}_{(u)}}$ followed by constructing an estimate of $\hat{\mathbf{x}}$ where $\hat{\mathbf{x}}_{\overline{\mathcal{I}}_{(p)}} = \mathbf{0}$. The index of the most promising atom is then found using the standard criterion of finding the coordinate associated with the highest amplitude of $\hat{\mathbf{x}}$. Such an orthogonal projection based atom selection mechanism is used in parallel IGS algorithms, for example in SP [35] and CoSaMP [37], albeit differently. To compare, the SP algorithm is described briefly in Appendix B.

Using Algorithm 1, let us define the following function.

Function 1: (Projection-based atom-index selection) Let $\mathbf{y} \in \mathbb{R}^M$, $\mathbf{A} \in \mathbb{R}^{M \times N}$, previous support set $\mathcal{I}$ and the set of potential atoms' indices $\mathcal{I}_{(p)}$ are given. Suppose i denotes the index of the new atom selected in the current iteration. Then i is the output of the following algorithmic function

$$i = \text{proj\_atom\_index}\left(\mathbf{A}, \mathbf{y}, \mathcal{I}, \mathcal{I}_{(p)}\right) \qquad (3)$$

where the above function executes the Algorithm 1.

Using Function 1, the main steps of the POMP algorithm are summarized in Algorithm 2 where the support set is constructed serially. In the POMP algorithm, L is a user defined positive integer parameter that decides how many atoms need to be checked as potential atoms in each iteration. The algorithm starts with an initial empty support set $\mathcal{I}_0 = \emptyset$ and an initial residual $\mathbf{r}_0 = \mathbf{y}$. At the k'th iteration stage, it uses the standard matched filter to choose the set of L potential atoms' indices (in step 3), orthogonal projection method to select a new atom's index from the set (in step 4), inducts the selected index into the intermediate support set under construction (in step 5), solves a LS problem with the intermediate support set and produces a new residual by subtracting the LS fit (in step 6). Given the sparsity level

---

Algorithm 2 : POMP for CS Recovery

Input:
1: $\mathbf{A}$, $\mathbf{y}$, K;
2: L (L potential atoms and $1 \leq L \leq K$);

Initialization:
1: Iteration counter $k \leftarrow 0$;
2: $\mathbf{r}_0 \leftarrow \mathbf{y}$, $\mathcal{I}_0 \leftarrow \emptyset$;

Iterations:
1: **repeat**
2:    $k \leftarrow k + 1$;
3:    $\mathcal{I}_{(p)} \leftarrow \{$indices of L highest amplitudes of $\mathbf{A}^t \mathbf{r}_{k-1}\}$;
     (Assumption: chosen L indices $\notin \mathcal{I}_{k-1}$)
4:    $i_k \leftarrow \text{proj\_atom\_index}\left(\mathbf{A}, \mathbf{y}, \mathcal{I}_{k-1}, \mathcal{I}_{(p)}\right)$;
5:    $\mathcal{I}_k \leftarrow \mathcal{I}_{k-1} \cup i_k$;   (Note: $|\mathcal{I}_k| = k$)
6:    $\mathbf{r}_k \leftarrow \mathbf{y} - \mathbf{A}_{\mathcal{I}_k} \mathbf{A}_{\mathcal{I}_k}^{\dagger} \mathbf{y}$;   (Orthogonal projection)
7: **until** $\left((\|\mathbf{r}_k\|_2 > \|\mathbf{r}_{k-1}\|_2) \text{ or } (k > K)\right)$
8: $k \leftarrow k - 1$;   (Previous iteration)

Output:
1: $\hat{\mathbf{x}} \in \mathbb{R}^N$, satisfying $\hat{\mathbf{x}}_{\mathcal{I}_k} = \mathbf{A}_{\mathcal{I}_k}^{\dagger} \mathbf{y}$ and $\hat{\mathbf{x}}_{\overline{\mathcal{I}}_k} = \mathbf{0}$.

K, the algorithm usually executes K iterations and forms a support set of cardinality K at the end. Note that the support set cardinality is increased one-by-one by selecting a new atom's index in each iteration. We also have used a stopping criterion that ensures a decreasing trend of residual norm over iterations and the maximum allowable cardinality of the constructed support set. We assume that the sparsity level K is known a-priori like the cases of OMP (see Algorithm 3 of [33])), SP [35] and CoSaMP [37].

In contrast to the OMP algorithm (see Appendix A for a brief description), the new additions in the POMP algorithm are the use of the set of potential atoms and the orthogonal projection based atom selection strategy from the set. A natural question is why does the projection based atom selection method perform better than the standard matched filter. For understanding, let us consider an over-determined case where $M \geq N$ and assume that the columns of $\mathbf{A}$ (i.e. atoms) are orthonormal [3]. Now let us consider OMP for such an overdetermined $\mathbf{A}$ and assume clean measurement condition. In that case, the matched filter provides exact signal component values and hence the use of matched filter is optimal. The matched filter selects all the underlying K atoms perfectly one-by-one over iterations. The selection order of atoms will follow a strictly regular path as follows: the atom associated with largest amplitude scalar value will be selected first and then the atom associated with second largest amplitude scalar value will be selected, and so on. That means the atom selection order will follow the strict path associated with the decreasing amplitudes of underlying non-zero scalar signal values. Now let us consider the underdetermined CS setup (or sparse signal estimation problem) where $M < N$ and hence all the atoms can not be orthonormal. In that case, the matched filter can not provide exact signal component

---

[3]On this assumption, we mention that the construction of $\mathbf{A}$ by using i.i.d random variables guarantees a near-orthonormal condition by high probability.

values, but a mixed output and hence the use of matched filter is non-optimal. For any chosen atom, there exists a set of atoms which are highly correlated with the chosen atom; this set of atoms are coherent with the chosen atom. The mixed output of the matched filter is the outcome of the fact that an underlying atom is subjected to an interference from other atoms, mainly from the coherent atoms. The interference is due to non-orthogonality and hence the inner product operations in the matched filter do not provide reliable signal component values. For example, the highest amplitude coefficient of the output vector of the matched filter may not be associated with the corresponding underlying atom. Thus, the highest amplitude based detection strategy in the matched filter may not provide perfect selection of atoms over iterations. If $M \ll N$, the level of interference is considerably high and efficiency of the matched filter degrades severely, leading to poor atom selection. To circumvent the problem partially, we propose to use the orthogonal projection method in POMP. In each iteration of POMP, the orthogonal projection helps to estimate the signal component values associated with the chosen potential atoms in the LS sense. In general, the LS approach is powerful to estimate a signal. Thus, the LS signal estimation followed by the highest amplitude based atom selection strategy has a better potential than the sole use of matched filter. Note that the matched filter provides a LS solution for a single atom, without considering contributions from the non-orthogonal atoms.

Next we discuss the complexity of the POMP algorithm where the orthogonal projection operation is performed twice in each iteration. Considering that the matched filtering and orthogonal projection operations are the most computationally intensive among all the relevant computational operations, the POMP complexity can be seen loosely as less than twice of the OMP complexity. We assume that the other computational operations require negligible resources. To be quantitative, the POMP requires to perform K matched filter operations which is same as that of OMP. However, the POMP requires $2K$ orthogonal projection operations compared to the K orthogonal projection operations in OMP. In the POMP, we used the parameter L (associated with the number of potential atoms) such that $L \leq K$. Our engineering perspective for the choice of $L \leq K$ is that no more atoms, than the total number of atoms that will eventually be chosen (i.e., K atoms), need to be tested as the potential atoms in any iteration. Also, another consideration is that the atom selection performance saturates with increasing L. Through experimental evaluations in section VI-C, we show that an increase in L leads to a better atom selection performance initially and then saturates. Observe that for $L = 1$, the performance of POMP is the same as the OMP.

In the POMP, the atom selection strategy for the current iteration does not depend on the future iterations. In the next section, we propose an algorithm that sees the future.

III. LOOK AHEAD ORTHOGONAL LEAST SQUARES

In this section, we develop look ahead OLS (LAOLS) algorithm where the selection of an atom's index in the current iteration is carried out according to its future effect on minimizing the final residual norm. The final residual norm is evaluated at the end of all future iterations. For developing the LAOLS algorithm, we first need to realize a look ahead strategy in an algorithmic manner such that the future effect can be evaluated. Algorithm 3 shows such a look ahead strategy to evaluate the final residual norm.

Algorithm 3 : Look Ahead Residual Norm

Input:
  1: $\mathbf{A}$, $\mathbf{y}$, K;
  2: Previous support set $\mathcal{I}$, current atom-index i;
Assumption:
  1: $i \notin \mathcal{I}$;
Initialization:
  1: Iteration counter $k \leftarrow |\mathcal{I} \cup i|$;
  2: $\mathcal{I}_k \leftarrow \mathcal{I} \cup i$, $\mathbf{r}_k \leftarrow \mathbf{y} - \mathbf{A}_{\mathcal{I}_k} \mathbf{A}_{\mathcal{I}_k}^{\dagger} \mathbf{y}$;
Iterations:
  1: **repeat**
  2:   $k \leftarrow k + 1$;
  3:   $i_k \leftarrow$ index of the highest amplitude of $\mathbf{A}^t \mathbf{r}_{k-1}$
        (Assumption: $i_k \notin \mathcal{I}_{k-1}$);
  4:   $\mathcal{I}_k \leftarrow \mathcal{I}_{k-1} \cup i_k$;   (Note: $|\mathcal{I}_k| = k$)
  5:   $\mathbf{r}_k \leftarrow \mathbf{y} - \mathbf{A}_{\mathcal{I}_k} \mathbf{A}_{\mathcal{I}_k}^{\dagger} \mathbf{y}$;   (Orthogonal projection)
  6: **until** $((\|\mathbf{r}_k\|_2 > \|\mathbf{r}_{k-1}\|_2)$ or $(k > K))$
  7: $k \leftarrow k - 1$;   (Previous iteration)
Output:
  1: $\|\mathbf{r}_k\|_2$ (where $\mathbf{r}_k = \mathbf{y} - \mathbf{A}_{\mathcal{I}_k} \mathbf{A}_{\mathcal{I}_k}^{\dagger} \mathbf{y}$).

The Algorithm 3 can be seen as a look ahead part of the OMP algorithm in the current iteration where the future atoms are chosen using the matched filter and orthogonal projection operations. Given the sparsity level K, a previous support set $\mathcal{I}$ and a current atom-index i (index of the atom chosen in the current iteration), the algorithm usually finds a K-element support set at the end of all future iterations and returns the final LS residual norm. For the given previous support set $\mathcal{I}$, the final effect of the choice of a new atom in the current iteration can be evaluated through using this look ahead part.

Using Algorithm 3, let us define the following function.

Function 2: (Look ahead residual norm) Let $\mathbf{y} \in \mathbb{R}^M$, $\mathbf{A} \in \mathbb{R}^{M \times N}$ and the sparsity level K are given. Suppose that the previous support set is $\mathcal{I}$ and the current atom-index is i. Then the look ahead residual norm $n \in \mathbb{R}$ is the output of the following algorithmic function

$$n = \text{look\_ahead\_resid\_norm}\,(\mathbf{A}, \mathbf{y}, K, \mathcal{I}, i) \qquad (4)$$

where the above function executes the Algorithm 3.

Using Function 2, the main steps of the LAOLS algorithm are summarized in Algorithm 4. Like POMP, we fix an integer parameter $L \leq K$ that decides how many potential atoms are checked in each iteration using the look ahead strategy of Function 2 (or Algorithm 3). The atom with smallest look ahead residual norm is selected and added to the previous support set $\mathcal{I}_{k-1}$ to form the new support set $\mathcal{I}_k$ of cardinality k. Given the sparsity level K, the algorithm usually executes

---

Algorithm 4 : LAOLS for CS Recovery

Input:

1: $\mathbf{A}, \mathbf{y}, K$;

2: L (L potential atoms and $1 \leq L \leq K$);

Define:

1: $\mathbf{j} = [j_1 \ j_2, \ldots, j_L]^t$, $\mathbf{n} = [n_1 \ n_2, \ldots, n_L]^t$;

Initialization:

1: Iteration counters $k \leftarrow 0$ (Outer loop), $l \leftarrow 0$ (Inner loop);

2: $\mathbf{r}_0 \leftarrow \mathbf{y}$, $\mathcal{I}_0 \leftarrow \emptyset$;

Iterations:

1: **repeat**

2:    $k \leftarrow k + 1$;

3:    $\mathcal{I}_{(p)} \leftarrow \{$indices of L highest amplitudes of $\mathbf{A}^t\mathbf{r}_{k-1}\}$;
     (Assumption: chosen L indices $\notin \mathcal{I}_{k-1}$)

4:    $\mathbf{j} \leftarrow \mathcal{I}_{(p)}$;
     (Elements of $\mathcal{I}_{(p)}$ are assigned to components of $\mathbf{j}$)

5:    **for** $l = 1$ to L **do**

6:      $n_l \leftarrow$ look_ahead_resid_norm $(\mathbf{A}, \mathbf{y}, K, \mathcal{I}_{k-1}, j_l)$;
       (Final residual norm if j_l is selected)

7:    **end for**

8:    $l^\star \leftarrow$ coordinate of the lowest component of $\mathbf{n}$;

9:    $i_k \leftarrow j_{l^\star}$;
     (Index of atom providing lowest final residual norm)

10:    $\mathcal{I}_k \leftarrow \mathcal{I}_{k-1} \cup i_k$;   (Note: $|\mathcal{I}_k| = k$)

11:    $\mathbf{r}_k \leftarrow \mathbf{y} - \mathbf{A}_{\mathcal{I}_k}\mathbf{A}_{\mathcal{I}_k}^\dagger\mathbf{y}$;   (Orthogonal projection)

12: **until** $((\|\mathbf{r}_k\|_2 > \|\mathbf{r}_{k-1}\|_2)$ or $(k > K))$

13: $k \leftarrow k - 1$;   (Previous iteration)

Output:

1: $\hat{\mathbf{x}} \in \mathbb{R}^N$, satisfying $\hat{\mathbf{x}}_{\mathcal{I}_k} = \mathbf{A}_{\mathcal{I}_k}^\dagger\mathbf{y}$ and $\hat{\mathbf{x}}_{\overline{\mathcal{I}}_k} = \mathbf{0}$.

K iterations and forms a support set of cardinality K at the end. We mention that the LAOLS algorithm can be seen as a confluence of OMP and OLS algorithms. For the OMP connection, note the use of matched filter in the look ahead part to find the indices of future atoms (see step 3 of Algorithm 3). Also, for the OLS connection, note how the index of a new atom is selected in each iteration depending on the minimum norm of fitting residual (see steps 8 and 9 of Algorithm 4). Technically, for designing a look ahead OLS, it is possible to use the OLS strategy of atom selection in Algorithm 3 instead of using the OMP strategy. However, we refrain from such a strategy due to a heavy increase in computational complexity[4].

In contrast to the OLS algorithm (see Appendix A for a brief description), the new additions in the LAOLS algorithm are the use of a set of potential atoms and the look ahead atom selection from the set. A natural observation is that the use of look ahead strategy is computationally intensive. To achieve computational advantage, the orthogonal projection operations in the look ahead strategy can be performed recursively using block-wise matrix inversion. An exposition for performing recursive computation is shown in Appendix C. For each l at

---

[4]The atom selection strategy of OLS is more computationally intensive than the atom selection strategy of OMP. The reader is encouraged to see Appendix A.

---

the kth iteration, the look ahead strategy requires to perform $(K - (k - 1))$ matched filtering and orthogonal projection operations. Therefore, each of the total matched filtering and the total orthogonal projection operations in the LAOLS are $\approx \sum_{l=1}^{L}\sum_{k=1}^{K} L(K - (k-1)) = \sum_{k=1}^{K} L(K - (k-1)) = L(1 + 2 + \ldots + K) = \frac{1}{2}L(K^2 + K)$. Given the sparsity level K, the computational complexity increases linearly with the number of potential atoms L. A reduction in L leads to a lower complexity, but at an expense of CS recovery performance. In the next section, we develop an algorithm with a reduced complexity, but without much loss in performance.

IV. Structured Orthogonal Least Squares

In this section, we develop a structured IGS algorithm by combining POMP and LAOLS algorithms such that a trade-off between computational complexity and reconstruction performance can be established. In each iteration, we choose L potential atoms by the standard matched filter followed by reducing the cardinality of the set of potential atoms through an orthogonal projection based atom selection strategy. Then, we use the look ahead strategy to select an atom from the reduced set of potential atoms. The use of the reduced set allows for a sharp reduction in computational burden. However, as the cardinality reduction of the set of potential atoms is performed through an orthogonal projection method, a little performance loss can be expected. Note that an atom is finally selected from the set of L potential atoms through using the two selection schemes which are algorithmically in a cascade connection. The new algorithm is referred to as structured OLS (SOLS).

To develop the SOLS algorithm, we first need to describe the Algorithm 5 which is a slight modification of Algorithm 1. In the Algorithm 5, L' is a positive integer that decides how

---

Algorithm 5 : Projection-Based Multiple Atom-indices Selection

Input:

1: $\mathbf{A}, \mathbf{y}$;

2: Previous support set: $\mathcal{I}$;

3: Set of potential atoms' indices: $\mathcal{I}_{(p)}$;

4: L' ($L' \leq |\mathcal{I}_{(p)}|$);

Assumption:

1: $\mathcal{I}_{(p)} \cap \mathcal{I} = \emptyset$, $\hat{\mathbf{x}} \in \mathbb{R}^N$;

Execution:

1: $\mathcal{I}_{(u)} \leftarrow \mathcal{I} \cup \mathcal{I}_{(p)}$;

2: $\hat{\mathbf{x}}_{\mathcal{I}_{(u)}} \leftarrow \mathbf{A}_{\mathcal{I}_{(u)}}^\dagger\mathbf{y}$;   (Orthogonal projection)

3: $\hat{\mathbf{x}}_{\overline{\mathcal{I}}_{(p)}} \leftarrow \mathbf{0}$;   (Zeroing the coordinates indexed by $\overline{\mathcal{I}}_{(p)}$)

4: $\mathcal{I}_{(p')} \leftarrow$ indices of the L' highest amplitudes of $\hat{\mathbf{x}}$;

Output:

1: The indices' set $\mathcal{I}_{(p')}$.

---

many atoms are selected from the set of potential atoms $\mathcal{I}_{(p)}$ through orthogonal projection. Considering $L' \leq |\mathcal{I}_{(p)}|$, the set of L' atoms are used for further processing. For a given cardinality of the set $\mathcal{I}_{(p)}$, the choice of L' is again a

user's prerogative that depends on the trade-off between atom selection precision and search complexity.

Using Algorithm 5, let us define the following function.

Function 3: (Projection-based multiple atom-indices selection) Let $\mathbf{y} \in \mathbb{R}^M$, $\mathbf{A} \in \mathbb{R}^{M \times N}$, previous support set $\mathcal{I}$ and the set of potential atoms' indices $\mathcal{I}_{(p)}$ are given. Suppose $\mathcal{I}_{(p')} \subset \mathcal{I}_{(p)}$ denotes the set of indices of L' atoms selected in the current iteration. Then $\mathcal{I}_{(p')}$ is the output of the following algorithmic function

$$\mathcal{I}_{(p')} = \text{proj\_multi\_atom\_indices}\left(\mathbf{A}, \mathbf{y}, \mathcal{I}, \mathcal{I}_{(p)}, L'\right) \quad (5)$$

where the above function executes the Algorithm 5.

Using Functions 2 and 3, the main steps of the SOLS algorithm are summarized in Algorithm 6. In the algorithm, we

---

Algorithm 6 : SOLS for CS Recovery

Input:
1: $\mathbf{A}$, $\mathbf{y}$, K;
2: L (L potential atoms and $1 \le L \le K$);
3: γ (A subset selection parameter and $0 \le \gamma < 1$).

Define:
1: $L' = L - \lfloor \gamma L \rfloor$; ($L' > 0$; non-zero cardinality of subset)
2: $\mathbf{j} = [j_1\ j_2, \ldots, j_{L'}]^t$, $\mathbf{n} = [n_1\ n_2, \ldots, n_{L'}]^t$;

Initialization:
1: Iteration counters $k \leftarrow 0$ (Outer loop), $l \leftarrow 0$ (Inner loop);
2: $\mathbf{r}_0 \leftarrow \mathbf{y}$, $\mathcal{I}_0 \leftarrow \emptyset$;

Iteration:
1: **repeat**
2:    $k \leftarrow k + 1$;
3:    $\mathcal{I}_{(p)} \leftarrow$ indices of the L highest amplitudes of $\mathbf{A}^t \mathbf{r}_{k-1}$;
     (Assumption: chosen L indices $\notin \mathcal{I}_{k-1}$)
4:    $\mathcal{I}_{(p')} \leftarrow \text{proj\_multi\_atom\_indices}\left(\mathbf{A}, \mathbf{y}, \mathcal{I}_{k-1}, \mathcal{I}_{(p)}, L'\right)$;
     (Reducing the number of potential atoms: $|\mathcal{I}_{(p')}| = L'$)
5:    $\mathbf{j} \leftarrow \mathcal{I}_{(p')}$;
     (Elements of $\mathcal{I}_{(p')}$ are assigned to components of $\mathbf{j}$)
6:    **for** $l = 1$ to L' **do**
7:       $n_l \leftarrow \text{look\_ahead\_resid\_norm}\left(\mathbf{A}, \mathbf{y}, K, \mathcal{I}_{k-1}, j_l\right)$;
8:    **end for**
9:    $l^\star \leftarrow$ coordinate of the lowest component of $\mathbf{n}$;
10:   $i_k \leftarrow j_{l^\star}$;
     (Index of the atom providing lowest final residual norm)
11:   $\mathcal{I}_k \leftarrow \mathcal{I}_{k-1} \cup i_k$;
12:   $\mathbf{r}_k \leftarrow \mathbf{y} - \mathbf{A}_{\mathcal{I}_k} \mathbf{A}_{\mathcal{I}_k}^\dagger \mathbf{y}$;   (Orthogonal projection)
13: **until** $((\|\mathbf{r}_k\|_2 > \|\mathbf{r}_{k-1}\|_2)$ or $(k > K))$
14: $k \leftarrow k - 1$;   (Previous iteration count)

Output:
1: $\hat{\mathbf{x}} \in \mathbb{R}^N$, satisfying $\hat{\mathbf{x}}_{\mathcal{I}_k} = \mathbf{A}_{\mathcal{I}_k}^\dagger \mathbf{y}$ and $\hat{\mathbf{x}}_{\overline{\mathcal{I}_k}} = \mathbf{0}$.

---

introduce an input parameter γ ($0 \le \gamma < 1$) that decides the cardinality of reduced set of potential atoms as $L' = L - \lfloor \gamma L \rfloor$. In the iterations, the selection of atoms through orthogonal projection is performed in step 4 where L' atoms are chosen from the set of L potential atoms. Then the chosen L' atoms are subjected through look ahead strategy such that a single atom is finally selected from the L' atoms and added to the

---

TABLE I
COMPUTATIONAL COMPLEXITY COMPARISON

Algorithm	Order of complexity
Existing algorithms	
OMP	$\mathcal{O}(K(MN + K^2 + KM))$
OLS	$\mathcal{O}(K^2(KN + MN))$
SP	$\mathcal{O}(K(MN + K^2M))$
Convex relaxation	$\mathcal{O}(N^2M)$ or $\mathcal{O}(N^3)$
New algorithms	
POMP	$\mathcal{O}(K(MN + K^2M))$
LAOLS	$\mathcal{O}(LK^2(MN + K^2 + KM))$
SOLS	$\mathcal{O}((L - \lfloor \gamma L \rfloor)K^2(MN + K^2 + KM))$

intermediate support-set. For $\gamma = 0$, SOLS acts as LAOLS; on the other hand, for a γ near to one that results in $L' = 1$, SOLS acts as POMP.

Each of the total matched filtering and the total orthogonal projection operations in SOLS $\approx \sum_{k=1}^{K} [1 + L'(K - (k-1))] = K + \frac{1}{2}L'(K^2 + K) = K + \frac{1}{2}(L - \lfloor \gamma L \rfloor)(K^2 + K)$. Given the sparsity level K and the number of potential atoms L, the computational complexity decreases linearly with the increase in γ. For example, the choice of $\gamma = 0.5$ leads to a reduction of computational complexity by nearly half compared to the LAOLS algorithm.

V. COMPARISON OF COMPUTATIONAL COMPLEXITY

In this section, we perform an analysis of computational complexity of the new algorithms and compare them with existing algorithms. We mainly quantify the number of addition and multiplication operations.

Let us first consider the POMP algorithm. For k'th iteration, the algorithm requires one matched filter operation and two orthogonal projections. A matched filter operation requires $2MN$ computations and each orthogonal projection requires $\mathcal{O}(k^2M)$ computations. Overall the associated complexity is $\mathcal{O}(MN + K^2M)$ where we assume the worst case scenario that $k = K$. Now, considering K iterations, the overall complexity of the POMP algorithm is $\mathcal{O}(K(MN + K^2M))$.

Next we consider the LAOLS algorithm. In this case, the intensive computation is due to the look ahead part where it is necessary to perform matched filter and orthogonal projection operations. We compute orthogonal projections recursively using the block-wise matrix inversion mechanism shown in Appendix C. Using recursive computation and considering the worst case scenario that $k = K$, the necessary computation for each orthogonal projection is approximately $7K^2 + 4KM$. So, the total computation for each matched filter and each orthogonal projection is $\mathcal{O}(MN + K^2 + KM)$. Now, considering $\frac{1}{2}L(K^2 + K)$ matched filter operations and orthogonal projections, the total complexity of LAOLS is $\mathcal{O}(LK^2(MN + K^2 + KM))$. Following the similar arguments, the complexity of SOLS algorithm is $\mathcal{O}((L - \lfloor \gamma L \rfloor)K^2(MN + K^2 + KM))$.

We show the comparison of complexity between several algorithms in Table I. Following [44] (pages 6-8 of [44]), we assume that complexity of convex relaxation methods is either $\mathcal{O}(N^2M)$ (using linear program) or $\mathcal{O}(N^3)$ (using interior-point methods). The complexity of the existing IGS algorithms

are evaluated in Appendix A and B. Let us consider a quantitative analysis to show the complexity benefit of LAOLS over convex relaxation methods. From the literature, we note that most of the sparse reconstruction algorithms are successful when $M = \mathcal{O}(K \log N)$. For our comparison, we assume that K varies linearly with N as $K = \tau N$ (where $0 < \tau < 1$) and a typical setup where $K = \tau N \ll M = K \log N \ll N$. We also use $L = K$ for LAOLS and find its computational complexity as $\mathcal{O}(K^3 M N)$; the term $K^2 + KM$ is neglected as $K^2 + KM \ll MN$. Therefore, the computational complexity of LAOLS is $\mathcal{O}(K^3 M N) = \mathcal{O}(K^4 N \log N) = \mathcal{O}(\tau^4 N^5 \log N)$. We also assume that interior-point methods are used for convex relaxation algorithms and hence their complexity is $\mathcal{O}(N^3)$. To achieve a better computational advantage for LAOLS than the convex relaxation algorithms, we must need $\mathcal{O}(\tau^4 N^5 \log N) < \mathcal{O}(N^3)$. This may be satisfied when $\tau^4 N^5 \log N < N^3$, or $\tau < (N^2 \log N)^{-0.25}$. For example, when $N = 500$, we need $\tau < 0.0283$ and hence the allowed $K < 0.0283 \times 500 \approx 15$. The above analysis is also valid for SOLS.

To achieve complexity advantage for LAOLS/SOLS over convex relaxation algorithms, the above analysis shows the need of a lower K. The complexity increases heavily for a higher K and hence, it is natural to seek suboptimal solutions for a higher K. A possible suboptimal solution may be not to allow the full future evolution in look ahead strategy, but to allow some extent by restricting the number of future iterations. Considering a depth variable η that denotes the necessary allowable future iterations performed in look ahead strategy, a trade-off between performance and computational complexity may be achieved by proper choice of L and η. An engineering intuition is that the increase of any of them keeping the other fixed leads to a saturation in performance. In the next section, we allow the look ahead strategy to check the full future evolution and show saturation in performance for increasing L. We also show running time comparison results between competing algorithms.

VI. EXPERIMENTS AND RESULTS

We performed computer simulations in order to compare the performance of following CS reconstruction algorithms: OMP, OLS, SP, POMP, LAOLS, SOLS, and basis pursuit/basis pursuit denoising/LASSO (BP/BPDN/LASSO) [10]-[11], [14]-[15]. Among these algorithms, BP/BPDN/LASSO is a convex relaxation algorithm (l_1 norm minimization based) and the other algorithms are IGS methods. The BP/BPDN simulation code (matlab) was taken from the l_1-magic toolbox [18]. The LASSO simulation code (matlab) was taken from the SparseLab toolbox [19]. For SOLS, we used a fixed value of $\gamma = 0.5$. Experimentally we evaluated the effect of L on the trade-off between complexity and reconstruction performance for POMP/LAOLS/SOLS. We first discuss the reconstruction performance measures and experimental setups, and then discuss the BP/BPDN/LASSO algorithm briefly followed by reporting the performance results of all the algorithms for clean and noisy measurements.

A. Performance measures and experimental setups

We use two performance measures. For the first performance measure, we use signal-to-reconstruction-noise ratio (SRNR) defined as

$$\text{SRNR} = \frac{\mathcal{E}\{\|\mathbf{x}\|_2^2\}}{\mathcal{E}\{\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2\}}, \tag{6}$$

where $\hat{\mathbf{x}}$ is the reconstructed signal vector. Note that our objective is to achieve a higher SRNR.

Next we define another performance measure which provides a direct measure of estimating the underlying support set. For a K-sparse signal vector $\mathbf{x}$, the support set was denoted as $\mathcal{I}$ with cardinality K. Let us denote the support set of reconstructed vector $\hat{\mathbf{x}}$ as $\hat{\mathcal{I}}$. We assume that $\hat{\mathbf{x}}$ is also a K-sparse signal vector, i.e. $|\hat{\mathcal{I}}| = K$. To measure the support set estimation error, we consider to use the distortion $d(\mathcal{I}, \hat{\mathcal{I}}) = 1 - \left(|\mathcal{I} \cap \hat{\mathcal{I}}| / K \right)$ [45]. Considering a large number of realizations (signal vectors), we can compute the average of $d(\mathcal{I}, \hat{\mathcal{I}})$. We define the average support-cardinality error (ASCE) as follows

$$\text{ASCE} = \mathcal{E}\left\{ d(\mathcal{I}, \hat{\mathcal{I}}) \right\} = 1 - \frac{1}{K} \mathcal{E}\left\{ |\mathcal{I} \cap \hat{\mathcal{I}}| \right\}. \tag{7}$$

Note that the ASCE has the range $[0, 1]$ and our objective is to achieve a lower ASCE. Along-with SRNR, the ASCE is used as the second performance evaluation measure because the main objective of the IGS algorithms is to estimate the underlying support set.

Now we discuss experimental setups. In a CS setup, all sparse signal vectors are expected to be exactly reconstructed if the number of measurements is more than a certain threshold [35]. However, the computational complexity to test this uniform reconstruction ability is very high. Instead, for empirical testing, we can devise a strategy that can compute the performance measures for random measurement matrix ensemble. To measure the level of under-sampling, let us define the fraction of measurements (FoM)

$$\alpha = \frac{M}{N}. \tag{8}$$

Using α, steps of the testing strategy are listed as follows:

1) For given values of the parameters K and N, choose α such that the number of measurements M is an integer.
2) Randomly generate a sensing matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$ where the components are drawn independently from a Gaussian source (i.e., $a_{i,j} \sim \mathcal{N}\left(0, \frac{1}{M}\right)$) and then scale the columns of $\mathbf{A}$ to unit-norm.
3) Randomly generate a set of K-sparse data $\mathbf{x}$ where the support set $\mathcal{I}$ is chosen uniformly over the set $\{1, 2, \ldots, N\}$. Let we denote the size of data as S (i.e., the number of signal vectors $\mathbf{x}$ is S). The non-zero components of $\mathbf{x}$ are independently drawn by either of the following two methods.
   a) The non-zero components are drawn independently from a standard Gaussian source. This type of signal is referred to as Gaussian sparse signal.
   b) The non-zero components are set to ones. This type of signal is referred to as binary sparse signal.

Note that the Gaussian sparse signal is compressible in nature. That means, in the descending order, the sorted amplitudes of a Gaussian sparse signal vector's components decay fast with respect to the sorted indices. This decaying trend corroborates with several natural signals (for example, wavelet coefficients of an image). On the other hand, a binary sparse signal is not compressible in nature, but of special interest for comparative study, since it represents a particularly challenging case for OMP-type of reconstruction strategies [33], [35].

4) For each data, compute the measurement $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}$ and apply the CS reconstruction methods independently.
5) Repeat steps 2-4 for a given times (let T times). Then evaluate the CS performance evaluation measures (by averaging over ST data).
6) Repeat steps 1-5 for a new α.

This test can be performed for any chosen K and N.

Considering the measurement noise $\mathbf{w} \sim \mathcal{N}\left(\mathbf{0}, \sigma_w^2 \mathbf{I}_M\right)$, we define the signal-to-measurement-noise-ratio (SMNR) as

$$\text{SMNR} = \frac{\mathcal{E}\{\|\mathbf{x}\|_2^2\}}{\mathcal{E}\{\|\mathbf{w}\|_2^2\}}, \qquad (9)$$

where $\mathcal{E}\{\|\mathbf{w}\|_2^2\} = \sigma_w^2 M$. We report the experimental results at varying SMNRs.

In the presence of a measurement noise, it is impossible to achieve perfect CS recovery. On the other hand, for the clean measurement case, perfect CS recovery of a sparse signal is possible if α exceeds a certain threshold. In the spirit of using CS for practical applications with a less number of measurements at clean and noisy conditions, we are mainly interested in a lower range of α where performances of the contesting algorithms can be fairly compared.

B. BP/BPDN/LASSO

In this subsection, we discuss BP, BPDN and LASSO [10]. The BP/BPDN is a convex relaxation algorithm that looks for directly minimizing the l_1 norm of the solution vector with respect to convex constraint functions. For the case of clean measurement (i.e. $\mathbf{w} = 0$ in (1)), we use BP where the following optimization problem is solved

$$P_1: \quad \hat{\mathbf{x}} = \arg\min_{\mathbf{x} \in \mathbb{R}^N} \ \|\mathbf{x}\|_1 \text{ subject to } \mathbf{y} = \mathbf{A}\mathbf{x}. \qquad (10)$$

The above optimization problem can be recast as a linear program (LP) [18]. For the case of noisy measurements, we use BPDN [10]. Using a quadratic constraint, we solve for

$$P_1^\epsilon: \quad \hat{\mathbf{x}} = \arg\min_{\mathbf{x} \in \mathbb{R}^N} \ \|\mathbf{x}\|_1 \text{ subject to } \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \epsilon, \quad (11)$$

where $\epsilon \geq \|\mathbf{w}\|_2$ is a parameter that depends on the noise properties. The above problem can be recast as a second order cone program (SOCP) [18] and solved by an interior-point method. An interior-point method, such as the log-barrier method [44], is powerful and provides consistent performance. According to the suggestion of [12], we choose $\epsilon^2 = \sigma_w^2 \left(M + 2\sqrt{2M}\right)$. The l_1-magic toolbox [18] provides BP and BPDN codes. In the absence of a slack estimate ϵ, we can solve an l_1 norm penalized l_2 norm cost function in the following way:

$$Q_1^\lambda: \quad \hat{\mathbf{x}} = \arg\min_{\mathbf{x} \in \mathbb{R}^N} \ \{\lambda\|\mathbf{x}\|_1 + \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2\}. \qquad (12)$$

The above formulation is called LASSO where the regularization parameter λ needs to be optimally chosen. The optimal choice of λ can be performed by LARS (least angle regression) [15]. SparseLab toolbox [19] provides code for such an optimal LASSO.

We note that BP/BPDN/LASSO does not provide a solution vector that is guaranteed to be K-sparse in any experimental condition. Therefore, to compute ASCE performance measure, we assume that the support set is constructed from the solution vector of BP/BPDN/LASSO by considering the atom-indices associated with K highest amplitude coefficients. Then a new reconstructed signal $\hat{\mathbf{x}}$ is estimated using the orthogonal projection of $\mathbf{y}$ on the constructed support set of cardinality K (LS solution). We used the new reconstructed signal for our purpose of evaluating SRNR. In the literature [46], [47], such an oracle driven approach is shown to provide better SRNR performance for convex relaxation algorithms. Through experimental evaluations, we also verified that the oracle driven approach improves the performance of BP/BPDN/LASSO significantly [47].

C. Experimental Results

Using $N = 500$, $K = 20$, $S = 100$ and $T = 100$, we performed experiments. That means, we used 500-dimensional sparse signal vectors with sparsity level $K = 20$. Such a 4% sparsity level is chosen in accordance with real life scenarios, such as most of the energy of an image signal in the wavelet domain is concentrated within $2 - 4\%$ coefficients [10]. We used 100 realizations of $\mathbf{A}$ (i.e., $T = 100$). For each realization of $\mathbf{A}$, we used 100 signal vectors that are randomly generated (i.e., $S = 100$). Then, we incremented α from a lower limit to a higher limit in a small step-size (with the constraint that corresponding M is an integer for a value of α). Therefore, for each CS method at a chosen α, the performance is evaluated through averaging over $100 \times 100 = 10000$ realizations.

Let us first observe the effect of increasing L for the POMP, LAOLS and SOLS algorithms. For the Gaussian sparse signal at clean measurement condition, we show the SRNR performance of the three algorithms in Fig. 1. We show the results for $L = 2$, 5 and 20, in the range of α from 0.1 to 0.2. In Fig. 1 (c), we do not show the performance of SOLS for $L = 2$ (for $L = 2$, SOLS acts as POMP). From Fig. 1, we observe that the SRNR performance improves as L increases and the performance improvement by increasing to $L = 5$ from $L = 2$ is similar to the performance improvement by increasing to $L = 20$ from $L = 5$. This observation corroborates that the performance improvement saturates with increasing L. Considering the user defined choice that $L \leq K$, we choose to use $L = 20$ as it provides the best performance for all the three algorithms. For binary sparse signal, we also performed similar experiments and found the same trend in performance. In the later part of this subsection, we show
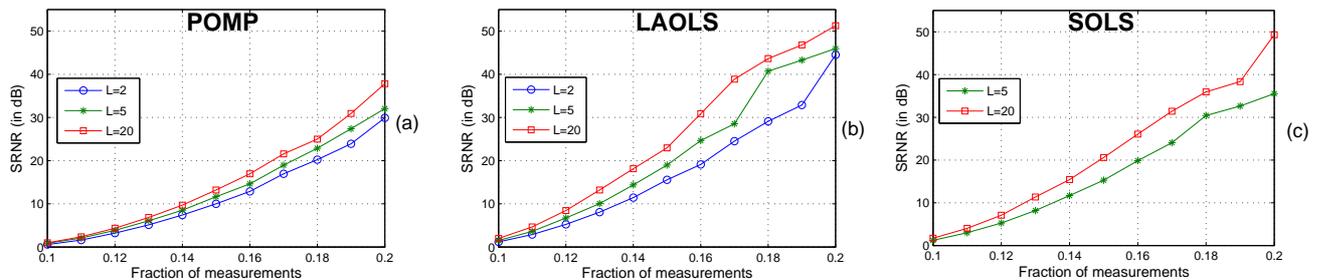
Fig. 1. Effect of increasing L on the POMP, LAOLS and SOLS algorithms for Gaussian sparse signal at the clean measurement condition. SRNR (in dB) versus fraction of measurements (α) is plotted: (a) POMP, (b) LAOLS, and (c) SOLS.

the results for POMP, LAOLS and SOLS algorithms where $L = 20$ is used.

Next, we compared between OMP, OLS, SP, BP/BPDN, LASSO, POMP, LAOLS and SOLS algorithms. We first show the SRNR and ASCE performance results for Gaussian sparse signal in Fig. 2. The results are shown for clean measurement condition and noisy measurement conditions with SMNR = 20 and 10 dB. Fig. 2 (a) and (b) compare the performance of all the algorithms for clean measurement condition. The SRNR results of Fig. 2 (a) is shown in Fig. 3 for a better view. We observe that the OLS and OMP provide similar performance. BP provides better SRNR than LASSO, but similar ASCE. The SP performs poorer than the OLS and OMP algorithms. For the proposed algorithms, we find that all the three new algorithms perform better than the existing algorithms. POMP provides a considerable improvement over OMP, and even performs better than the BP. At $\alpha = 0.2$, the POMP provides more than 6 dB SRNR performance improvement compared to OLS and OMP algorithms. The LAOLS outperforms all the other competing algorithms by a significant margin and can be considered the best. At $\alpha = 0.2$, the LAOLS provides nearly 20 dB SRNR performance improvement over OLS and OMP algorithms and 15 dB improvement over BP. The SOLS and POMP are found to be the second and third best, respectively. Now, let us observe the performance trends for noisy measurement conditions. Fig. 2 (c) and (d) show the comparative study at SMNR = 20 dB. In this case, we still find that the new algorithms are better than the existing algorithms. LAOLS and SOLS are found to show significant improvements than the others. A natural question is what happens if the noise power increases. At SMNR = 10 dB, Fig. 2 (e) and (f) show the results where we note that the BPDN and LASSO provide similar performance trends and they are better than the other IGS algorithms. For the Gaussian sparse signal, the results show that the IGS algorithms are non-robust at a higher measurement noise power.

Now we show the comparative results for binary sparse signal in Fig. (4). The results are shown for clean measurement condition and noisy measurement conditions with SMNR = 20 and 10 dB. Fig. 4 (a) and (b) compare the performance of all the algorithms for clean measurement condition. We again observe that the OLS and OMP provide similar performance. However, in this case, the SP is found to perform better than the OLS and OMP algorithms. The convex relaxation

algorithms (BP and LASSO) provide the best performance among all the algorithms. For the proposed algorithms, we find that POMP provides a considerable improvement over OMP. It can be said that LAOLS is the best performer among the IGS algorithms. Let us now compare the algorithms for noisy measurement conditions. At SMNR = 20 dB, Fig. 4 (c) and (d) display the results where the trends are similar to the clean measurement case results shown in Fig. 4 (a) and (b). However, the performance trends take an interesting turn at a higher noise power level. At SMNR = 10 dB, Fig. 4 (e) and (f) show the results where we observe that the BPDN/LASSO does not provide the best performance, but the LAOLS provides the best performance closely followed by the second best performance of SOLS. For the binary sparse signal, the LAOLS, SOLS and SP are found to perform better than the BPDN/LASSO at a higher measurement noise power.

Comparing all the algorithms for two different input signals at varying noise power and number of measurements, we note that the proposed POMP, LAOLS and SOLS have a promise to provide good performance. They consistently outperformed OMP and OLS algorithms. A specific example is that, for the Gaussian sparse signal at clean measurement condition, LAOLS provides 20 dB SRNR improvement than the OMP and OLS at $\alpha = 0.2$.

Next we performed a running time comparison between the competing algorithms for the CS reconstruction of Gaussian sparse signal at SMNR = 20 dB. The running time means the average execution time. To compute the running time, we used matlab codes executing on a desktop computer and the averaging is performed using CS execution times for 100 data (we used $T = 10$ ans $S = 10$). We performed the running time experiments for different (N, K) pairs. The chosen pairs are $(500, 10)$, $(500, 20)$, $(500, 30)$, $(1000, 20)$, $(1000, 30)$ and $(1000, 40)$. For each (N, K) pair, the choice of $M = \lceil K \log N \rceil$. The running time results are shown in Table II. Following the discussion in section V, we denote the allowable K by $K^\star$ for which the LAOLS and SOLS might have a lower complexity than the BPDN complexity (BPDN uses interior-point method). From Table II we find that the OMP has the minimum running time. The POMP and SP have similar order of running time. The running time of LAOLS and SOLS increases heavily with the increase of K. In this table, SOLS has the parameter $\gamma = 0.5$. Depending on the parameter γ, the SOLS running time can be adjusted. We performed the

Fig. 2. Comparison of OMP, OLS, SP, BP/BPDN, LASSO, POMP, LAOLS and SOLS algorithms for **Gaussian sparse signal in clean and noisy measurement conditions**. SRNR (in dB) and ASCE performances are plotted against the fraction of measurements (α). We show the results for clean measurement as well as noisy measurements with SMNR = 20 and 10 dB. BP/BPDN is denoted by BP(DN).
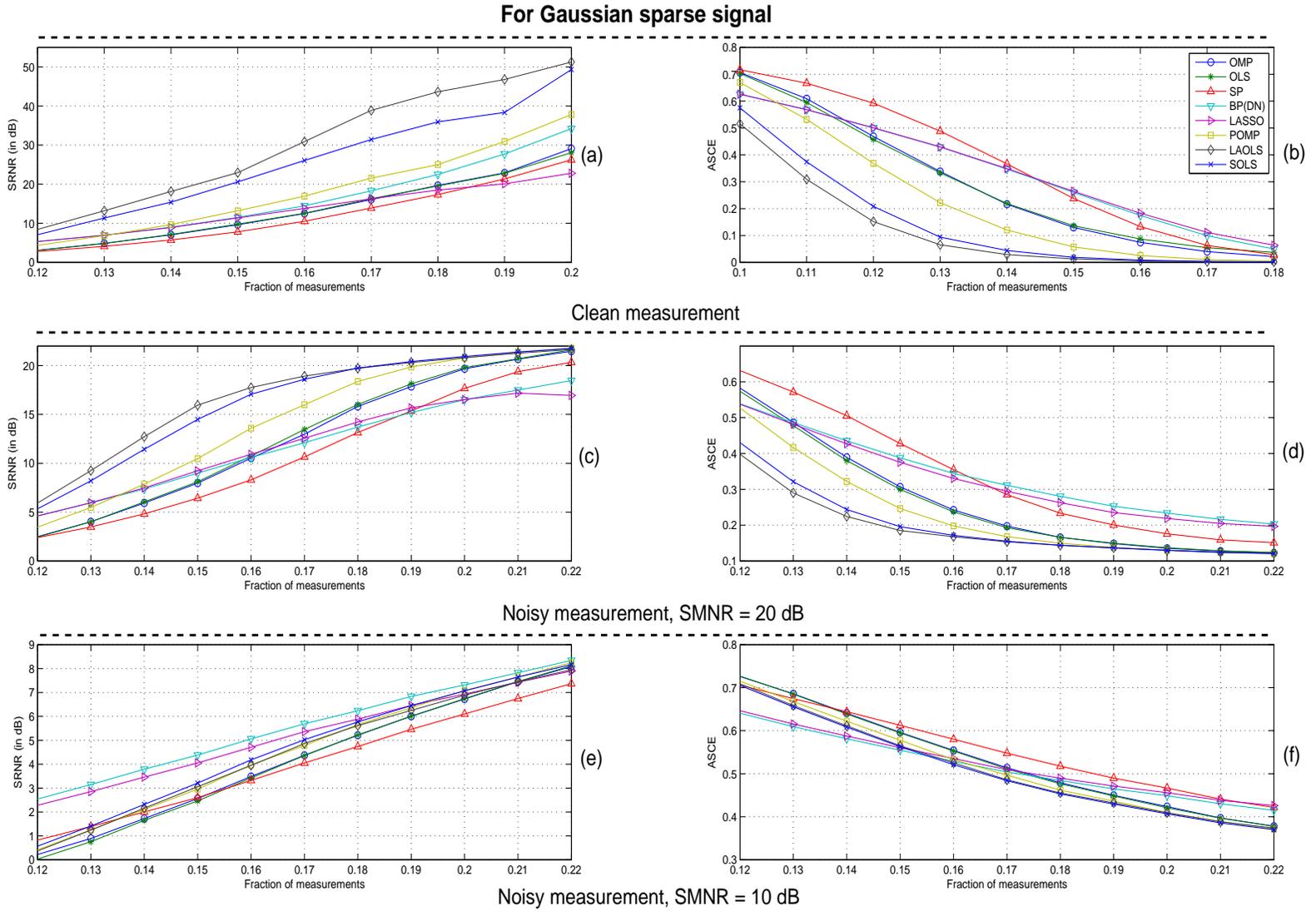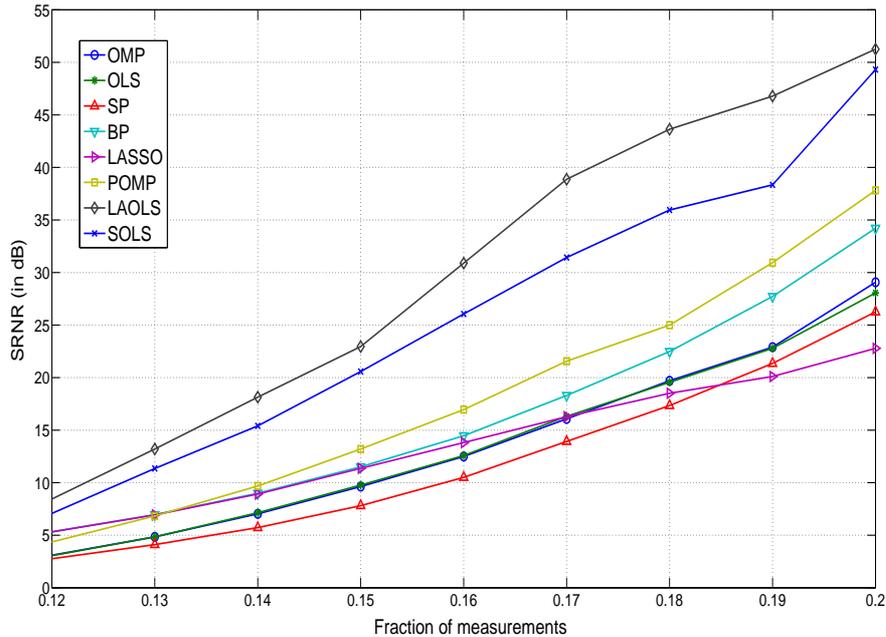
Fig. 3. Comparison of OMP, OLS, SP, BP, LASSO, POMP, LAOLS and SOLS algorithms for **Gaussian sparse signal in clean measurement condition**. SRNR (in dB) is plotted against the fraction of measurements (α). The same results are shown in Fig. 2 (a).

TABLE II

RUNNING TIME COMPARISON

Algorithm	$N = 500$			$N = 1000$		
	$K^\star = 15$			$K^\star = 20$		
	$K = 10$	$K = 20$	$K = 30$	$K = 20$	$K = 30$	$K = 40$
Existing algorithms						
OMP	0.0010	0.0023	0.0043	0.0042	0.0108	0.0220
OLS	0.3076	1.0100	2.6124	1.0522	2.6973	6.0095
SP	0.0057	0.0193	0.0482	0.0226	0.0541	0.1063
BPDN	0.4599	0.8140	1.1478	2.6550	2.9915	3.5564
LASSO	0.0442	0.1325	0.2700	0.2358	0.5379	1.1774
New algorithms						
POMP	0.0098	0.0340	0.0945	0.0407	0.1099	0.2465
LAOLS	0.0752	0.6696	2.7874	1.1794	6.4881	22.1076
SOLS	0.0337	0.2560	1.0172	0.4395	2.3020	7.6874

running time comparison for binary sparse signal also, but do not report the results as they are similar to the Gaussian sparse signal case.

We end this section with interesting observations. We note that, depending on the signal statistics (Gaussian and binary), the algorithms show different performance trends. For example, LAOLS provides better performance for a Gaussian sparse signal in the clean and moderately noisy measurement conditions. On the other hand, BP/BPDN/LASSO provides better performance for a binary sparse signal in the same testing conditions. Another example is that the SP provides poor performance for the Gaussian sparse signal, but found to be good for the binary sparse signal. Therefore, a challenging research problem is to develop adaptive IGS algorithms which are robust to underlying signal statistics.

Reproducible results: In the spirit of reproducible results, we provide necessary download-able matlab codes in the following website: https://sites.google.com/site/saikatchatt/softwares/. For CS reconstruction of Gaussian sparse signal at SMNR = 20 dB, the codes produce the SRNR and ASCE results shown in Fig. 2 (c) and (d), and the running time comparison results shown in Table II.

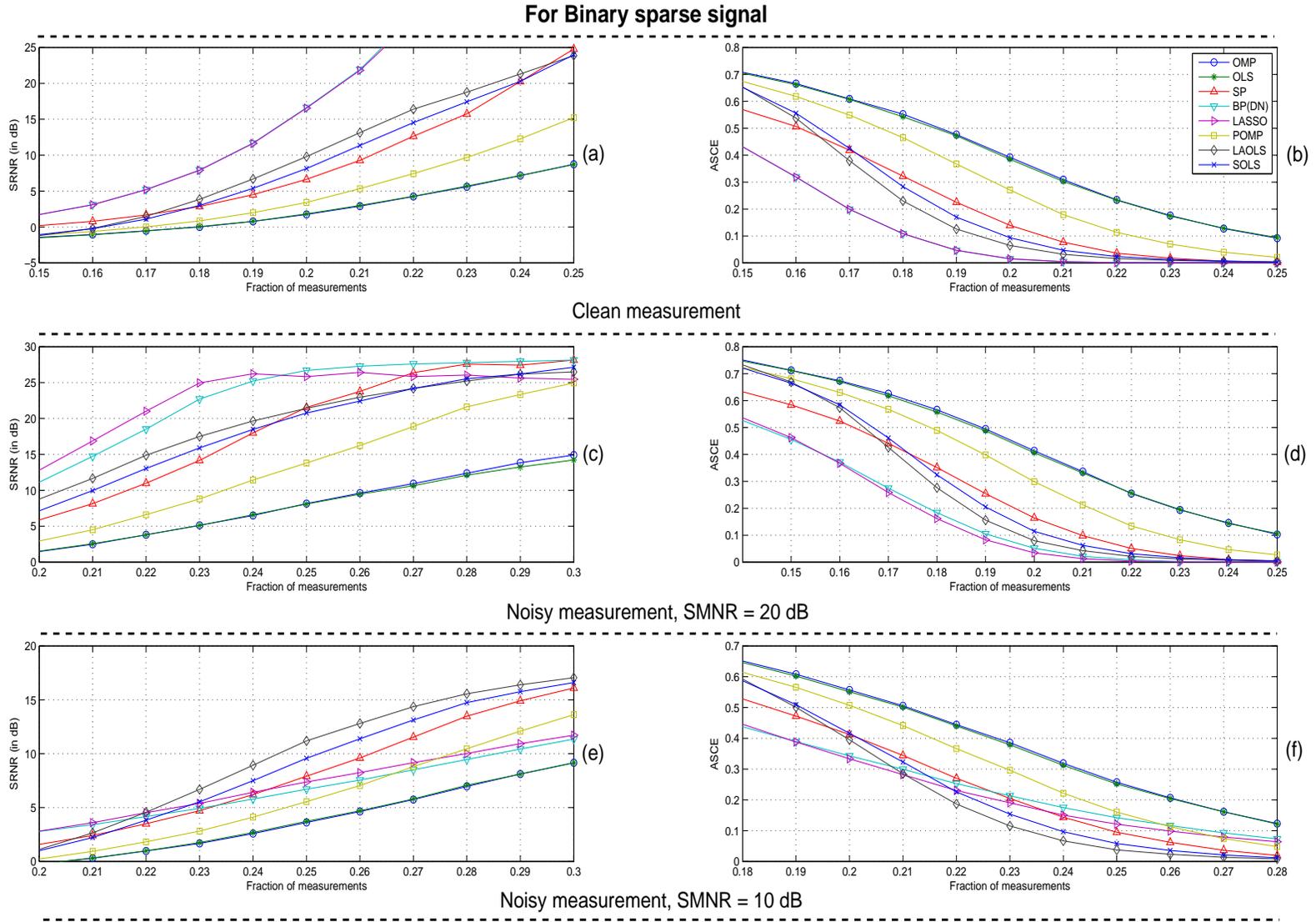## VII. DISCUSSIONS, CONCLUSIONS, AND FUTURE WORKS

For greedy algorithms where atoms are selected one-by-one serially over iterations, we show that new schemes can be developed for selecting an atom from a set of potential atoms. The potential atoms are found using a standard matched filter. The use of standard matched filter reduces the search space reasonably and allows us to use more sophisticated tools for atom selection. Among the developed tools, the look ahead strategy allows us to see the evaluation of an algorithm in future iterations. Overall, for selecting atoms, the new tools help to improve further on the state-of-art performance of matched filter.

Using our experimental framework, we also note that the performance of greedy search algorithms depend on the input signals' statistics. Future investigations can be performed to develop new greedy search algorithms that are robust to the statistics of input signals.

VIII. ACKNOWLEDGEMENT

For Binary sparse signal

Fig. 4. Comparison of OMP, OLS, SP, BP/BPDN, LASSO, POMP, LAOLS and SOLS algorithms for **Binary sparse signal in clean and noisy measurement conditions**. SRNR (in dB) and ASCE performances are plotted against the fraction of measurements (α). We show the results for clean measurement as well as noisy measurements with SMNR = 20 and 10 dB. BP/BPDN is denoted by BP(DN).

in part by VINNOVA and the Strategic Research Area project RICSNET.

APPENDIX

A. Serial Atom Selection Based OMP and OLS

In this section, we briefly describe two serial atom selection based IGS algorithms: OMP and OLS. First we summarize the main steps of the OMP in Algorithm 7 (see [31], [32] and Algorithm 3 of [33]). Note that, in each iteration, the OMP per-

---

Algorithm 7 : OMP for CS Recovery

Input:

  1: $\mathbf{A}$, $\mathbf{y}$, K;

Initialization:

  1: Iteration counter $k \leftarrow 0$;

  2: $\mathbf{r}_0 \leftarrow \mathbf{y}$, $\mathcal{I}_0 \leftarrow \emptyset$;

Iterations:

  1: **repeat**

  2:    $k \leftarrow k + 1$;

  3:    $i_k \leftarrow$ index of the highest amplitude of $\mathbf{A}^t \mathbf{r}_{k-1}$;

  4:    $\mathcal{I}_k \leftarrow \mathcal{I}_{k-1} \cup i_k$;   (Note: $|\mathcal{I}_k| = k$)

  5:    $\mathbf{r}_k \leftarrow \mathbf{y} - \mathbf{A}_{\mathcal{I}_k} \mathbf{A}_{\mathcal{I}_k}^{\dagger} \mathbf{y}$;   (Orthogonal projection)

  6: **until** $((\|\mathbf{r}_k\|_2 > \|\mathbf{r}_{k-1}\|_2)$ or $(k > K))$

  7: $k \leftarrow k - 1$;   (Previous iteration)

Output:

  1: $\hat{\mathbf{x}} \in \mathbb{R}^N$, satisfying $\hat{\mathbf{x}}_{\mathcal{I}_k} = \mathbf{A}_{\mathcal{I}_k}^{\dagger} \mathbf{y}$ and $\hat{\mathbf{x}}_{\overline{\mathcal{I}}_k} = \mathbf{0}$.

---

forms a matched filter operation and an orthogonal projection. Using block-wise matrix inversion, the orthogonal projection operations can be performed recursively. Considering the worst case scenario that $k = K$, the necessary computation for each orthogonal projection is approximately $7K^2 + 4KM$. So, the total computation for each matched filter and each orthogonal projection is $\mathcal{O}(MN + K^2 + KM)$. Now, considering K iterations, the total complexity is $\mathcal{O}(K(MN + K^2 + KM))$.

Next we summarize the OLS algorithm [31] in Algorithm 8. The overall structure of the OLS algorithm is similar to the OMP algorithm. The only and important difference is the way how the atoms are chosen through iterations. In each iteration, an atom is selected in such a way that its choice leads to the minimum residual norm for that iteration. This approach of atom selection is different from the OMP which uses a matched filter to select an atom, but does not guarantee to provide the minimum residual norm. However, OLS is computationally more intensive than the OMP. The OLS requires to perform approximately $[N + (N - 1) + \ldots + (N - K)] = \frac{1}{2}(KN - K^2 + K)$ orthogonal projection operations. As typically $N \gg K$, the computational complexity can be significantly high. However, in this case also the orthogonal projection operations can be performed recursively using block-wise matrix inversion. Considering the worst case scenario that $k = K$, the necessary computation for each orthogonal projection is approximately $7K^2 + 4KM$. So, the overall complexity of OLS is $\mathcal{O}(KN(K^2 + KM)) = \mathcal{O}(K^2(KN + MN))$.

Algorithm 8 : OLS for CS Recovery

Input:

  1: $\mathbf{A}$, $\mathbf{y}$, K;

Initialization:

  1: Iteration counter $k \leftarrow 0$;

  2: $\mathbf{r}_0 \leftarrow \mathbf{y}$, $\mathcal{I}_0 \leftarrow \emptyset$;

Iterations:

  1: **repeat**

  2:    $k \leftarrow k + 1$;

  3:    $i_k \leftarrow \arg\min_{i:\mathcal{I}_{(u)}=\mathcal{I}_{k-1}\cup i} \|\mathbf{y} - \mathbf{A}_{\mathcal{I}_{(u)}} \mathbf{A}_{\mathcal{I}_{(u)}}^{\dagger} \mathbf{y}\|_2$;

        $(i \notin \mathcal{I}_{k-1}$ and orthogonal projections)

  4:    $\mathcal{I}_k \leftarrow \mathcal{I}_{k-1} \cup i_k$;   (Note: $|\mathcal{I}_k| = k$)

  5:    $\mathbf{r}_k \leftarrow \mathbf{y} - \mathbf{A}_{\mathcal{I}_k} \mathbf{A}_{\mathcal{I}_k}^{\dagger} \mathbf{y}$;   (Orthogonal projection)

  6: **until** $((\|\mathbf{r}_k\|_2 > \|\mathbf{r}_{k-1}\|_2)$ or $(k > K))$

  7: $k \leftarrow k - 1$;   (Previous iteration)

Output:

  1: $\hat{\mathbf{x}} \in \mathbb{R}^N$, satisfying $\hat{\mathbf{x}}_{\mathcal{I}_k} = \mathbf{A}_{\mathcal{I}_k}^{\dagger} \mathbf{y}$ and $\hat{\mathbf{x}}_{\overline{\mathcal{I}}_k} = \mathbf{0}$.

B. Parallel Atom Selection Based SP

In this section, we describe the parallel atom selection based IGS algorithm: SP. We summarize the main steps of the SP algorithm in Algorithm 9 (see Algorithm 1 of [35]). The SP

Algorithm 9 : SP for CS Recovery

Input:

  1: $\mathbf{A}$, $\mathbf{y}$, K;

Initialization:

  1: Iteration counter $k \leftarrow 0$;

  2: $\mathcal{I}_0 \leftarrow$ indices of the K highest amplitudes of $\mathbf{A}^t \mathbf{y}$;

  3: $\mathbf{r}_0 \leftarrow \mathbf{y} - \mathbf{A}_{\mathcal{I}_0} \mathbf{A}_{\mathcal{I}_0}^{\dagger} \mathbf{y}$;

Iterations:

  1: **repeat**

  2:    $k \leftarrow k + 1$;

  3:    $\mathcal{I}_{(p)} \leftarrow \{$indices of K highest amplitudes of $\mathbf{A}^t \mathbf{r}_{k-1}\}$;

  4:    $\mathcal{I}_{(u)} \leftarrow \mathcal{I}_{k-1} \cup \mathcal{I}_{(p)}$; $(K \leq |\mathcal{I}_{(u)}| \leq 2K)$

  5:    $\hat{\mathbf{x}}_{\mathcal{I}_{(u)}} \leftarrow \mathbf{A}_{\mathcal{I}_{(u)}}^{\dagger} \mathbf{y}$; $\hat{\mathbf{x}}_{\overline{\mathcal{I}}_{(u)}} \leftarrow \mathbf{0}$; (Orthogonal projection)

  6:    $\mathcal{I}_k \leftarrow \{$indices of the K highest amplitudes of $\hat{\mathbf{x}}\}$;

  7:    $\mathbf{r}_k \leftarrow \mathbf{y} - \mathbf{A}_{\mathcal{I}_k} \mathbf{A}_{\mathcal{I}_k}^{\dagger} \mathbf{y}$;   (Orthogonal projection)

  8: **until** $(\|\mathbf{r}_k\|_2 > \|\mathbf{r}_{k-1}\|_2)$

  9: $k \leftarrow k - 1$;   (Previous iteration)

Output:

  1: $\hat{\mathbf{x}} \in \mathbb{R}^N$, satisfying $\hat{\mathbf{x}}_{\mathcal{I}_k} = \mathbf{A}_{\mathcal{I}_k}^{\dagger} \mathbf{y}$ and $\hat{\mathbf{x}}_{\overline{\mathcal{I}}_k} = \mathbf{0}$.

---

algorithm starts with an initial K-element support set $\mathcal{I}_0$ and an initial residual $\mathbf{r}_0 = \mathbf{y} - \mathbf{A}_{\mathcal{I}_0} \mathbf{A}_{\mathcal{I}_0}^{\dagger} \mathbf{y}$. At the k'th iteration stage, it forms the 'matched filter' $\mathbf{A}^t \mathbf{r}_{k-1}$, identifies the K highest amplitude coordinates, forms a dummy support set $\mathcal{I}_{(u)} = \mathcal{I}_{k-1} \cup \mathcal{I}_{(p)}$, refines out K-element support set $\mathcal{I}_k$ from $\mathcal{I}_{(u)}$, solves a LS problem with the selected indices in $\mathcal{I}_k$, subtracts the LS fit and produces a new residual. Given the sparsity level K, the algorithm estimates a support set of cardinality K in each iteration and runs until the residual norm minimization condition is violated. Note that, unlike in the

case of serial atom selection based OMP and OLS algorithms, here the support set cardinality is not increased one-by-one through iterations. Rather, a K-element support set is refined through iterations by addition of potential new atoms and deletion of unnecessary atoms. An important point is to note how the K-element support set $\mathcal{I}_k$ is chosen from the dummy support set $\mathcal{I}_{(u)}$ through using the orthogonal projection that invokes LS solution. The dummy support set $\mathcal{I}_{(u)}$ is formed through unionizing the previously estimated support set $\mathcal{I}_{k-1}$ with the set of K new atoms' indices. Then the observation $\mathbf{y}$ is orthogonally projected on the span of atoms that are indexed in $\mathcal{I}_{(u)}$ followed by picking up K indices corresponding to the highest amplitude coefficients of the solution vector. In the context of POMP/SOLS algorithm, we used similar strategy in Algorithm 1/Algorithm 5 for improving the atom selection strategy where a single atom/multiple atoms is/are selected from a set of potential atoms in each iteration. Normally, the SP algorithm converges less than K iterations. Assuming the worst case scenario that the SP algorithm runs at most K iterations, it performs K matched filtering and $2K$ orthogonal projection operations. Note that, like OMP and OLS, the orthogonal projections can not be performed recursively. Hence, the total complexity for each iteration is $\mathcal{O}(MN + K^2M)$. Therefore, considering K iterations, the overall complexity of SP algorithm is $\mathcal{O}(K(MN + K^2M))$.

C. Recursive Computation for Look Ahead Strategy

In Algorithm 3, a computationally intensive part is performing the orthogonal projection operation in each iteration. For orthogonal projection, we need to perform pseudo-inverse of $\mathbf{A}_{\mathcal{I}_k} \in \mathbb{R}^{M \times k}$ where $\mathbf{A}_{\mathcal{I}_k} = \begin{bmatrix} \mathbf{A}_{\mathcal{I}_{k-1}} & \mathbf{a}_{i_k} \end{bmatrix}$. The pseudo-inverse $\mathbf{A}_{\mathcal{I}_k}^{\dagger}\mathbf{y} = \begin{bmatrix} \mathbf{A}_{\mathcal{I}_k}^t \mathbf{A}_{\mathcal{I}_k} \end{bmatrix}^{-1} \mathbf{A}_{\mathcal{I}_k}^t \mathbf{y}$ and we note that the computation of inverse operation can be performed recursively. Denoting $\mathbf{P}_k^{-1} = \begin{bmatrix} \mathbf{A}_{\mathcal{I}_k}^t \mathbf{A}_{\mathcal{I}_k} \end{bmatrix}^{-1}$, we show (using block-wise matrix inversion) the recursive expression in (13) where $\mathbf{q}_k = \mathbf{A}_{\mathcal{I}_{k-1}}^t \mathbf{a}_{i_k}$ (note $\mathbf{q}_k \in \mathbb{R}^{k-1}$) and the scalar $\beta_k = 1 - \mathbf{q}_k^t \left[\mathbf{P}_{k-1} \right]^{-1} \mathbf{q}_k$. Here we use the fact that $\mathbf{P}_k$ is symmetric. Also note that the computation of $\mathbf{A}_{\mathcal{I}_k}^t \mathbf{y}$ can be performed recursively as

$$\mathbf{A}_{\mathcal{I}_k}^t \mathbf{y} = \begin{bmatrix} \mathbf{A}_{\mathcal{I}_{k-1}}^t \mathbf{y} \\ \mathbf{a}_{i_k}^t \mathbf{y} \end{bmatrix}. \tag{14}$$

Now, considering the fact that the multiplication of two matrices, $\mathbf{C} \in \mathbb{R}^{m \times n}$ and $\mathbf{D} \in \mathbb{R}^{n \times k}$, requires $2kmn$ operations, we can show that the total required computation for pseudo-inverse of $\mathbf{A}_{\mathcal{I}_k} \in \mathbb{R}^{M \times k}$ is $\approx 7k^2 + 4kM$.

REFERENCES

[1] A. Miller, "Subset Selection in Regression (second edition)," Monographs on Statistics and Applied Probability, Chapman & Hall/CRC, 2002.

[2] A.K. Fletcher, V.K. Goyal and S. Rangan, "A sparsity detection framework for on-off random access channels," *Proc. ISIT*, pp. 169-173, 2009.

[3] W.U. Bajwa, J. Haupt, A.M. Sayeed and R. Nowak, "Compressed Channel Sensing: A New Approach to Estimating Sparse Multipath Channels," *Proc. IEEE*, vol. 98, no. 6, pp. 1058-1076, 2010.

[4] A.Y. Yang, M. Gastpar, R. Bajcsy and S.S. Sastry, "Distributed sensor perception via sparse representation ," *Proc. IEEE*, vol. 98, no. 6, pp. 1077-1088, 2010.

[5] L. Daudet, "Sparse and structured decompositions of signals with the molecular matching pursuit," *IEEE Trans. Audio, Speech, Language Proc.*, vol. 14, no. 5, pp. 1808-1816, 2006.

[6] M.D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval and M.E. Davies, "Sparse representations in audio and music: from coding to source separation," *Proc. IEEE*, vol. 98, no. 6, pp. 995-1005, 2010.

[7] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Proc.*, vol. 15, no. 12, pp. 3736-3745, Dec. 2006.

[8] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, M. Yi, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal., Machine Intell.*, vol. 31, no. 2, pp. 210-227, 2009.

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

[10] E.J. Candes and M.B. Wakin, "An introduction to compressive sampling," *IEEE Signal Proc. Magazine*, vol. 25, pp. 21-30, March 2008.

[11] S.S. Chen, D.L. Donoho, and M.A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Scientif. Comput.*, vol. 20, no. 1, pp. 33-61, 1998.

[12] E.J. Candes, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. Pure Appl. Math*, vol. 59, no. 8, pp. 12071223, 2006.

[13] E.J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Info. Theory*, vol. 52, no. 2, pp. 489-509, 2006.

[14] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal. Statist. Soc., Ser. B*, vol. 58, no. 1, pp. 267-288, 1996.

[15] B. Efron, T. Haste, I. Johnstone and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407-499, 2004.

[16] E.J. Candes, M.B. Wakin and S. Boyd, "Enhancing sparsity by reweighted l_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877-905, 2008.

[17] E.J. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Info. Theory*, vol. 51, no. 12, pp. 4203-4215, 2005.

[18] E.J. Candes and J. Romberg, "l_1-MAGIC: Recovery of sparse signal via convex programming," A document in l_1-Magic toolbox shared online, http://www-stat.stanford.edu/~candes/l1magic/, Oct 2005.

[19] D. Donoho, V. Stodden and Y. Tsaig, "About SparseLab," SparseLab: Seeking Sparse Solutions to Linear Systems of Equations, SparseLab toolbox shared online, http://sparselab.stanford.edu/, 24th August 2007.

[20] S. Ji, Y. Xue, and L. Carin, "Bayesian compressive sampling," *IEEE Trans. Signal Proc.*, vol. 56, no. 6, pp. 2346-2356, 2008.

[21] L. He and L. Carin, "Exploiting structure in Wavelet-based Bayesian compressive sampling," *IEEE Trans. Signal Proc.*, vol. 57, no. 9, pp. 3488-3497, 2009.

[22] J.F. Murray, and K. Kreutz-Delgado, "An improved FOCUSS-based learning algorithm for solving sparse linear inverse problems ," *Conf. Rec. 35'th Asilomar conf. Signals, Systems, Computers*, pp. 347-351, 2001.

[23] M.E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Machine Learning Res.*, vol. 1, pp. 211-244, 2001.

[24] D.P. Wipf and B.D. Rao, "Sparse Bayesian learning for basis selection," *IEEE Trans. Signal Proc.*, vol. 52, no. 8, pp. 2153-2164, 2004.

[25] S.G. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. Signal Proc.*, vol. 41, no. 12, pp. 3397-3415, 1993.

[26] G. Davis, S. Mallat and Z. Zhang, "Adaptive time-frequency decompositions," *SPIE Jrnl. of Optical Engg.*, vol. 33, pp. 2183-2191, 1994.

[27] Y.C. Pati, R. Rezaiifar and P.S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," *Proc. Asilomar Conf. on Signals, Systems and Computers*, pp. 40-44, 1993.

[28] S. Chen, S.A. Billings and W. Luo, "Orthogonal least squares methods and their application to non-linear system identification," *International Journal of Control*, vol. 50, no. 5, pp. 1873-1896, 1989.

[29] L. Rebollo-Neira and D. Lowe, "Optimized orthogonal matching pursuit approach," *IEEE Signal Proc. Letters*, vol. 9, no. 4, pp. 137-140, 2002.

[30] B.K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM Journal on Computing*, vol. 24, no. 2, pp. 227-234, 1995.

[31] T. Blumensath and M.E. Davies, "On the difference between orthogonal mathcing pursuit and orthogonal least squares," Unpublished document.

[32] T. Blumensath and M.E. Davies, "Gradient pursuits," *IEEE Trans. Signal Proc.*, vol. 56, no. 6, pp. 2370-2382, 2008.

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

$$
\begin{aligned}
[\mathbf{P}_k]^{-1} &= \left[\left[\begin{array}{cc} \mathbf{A}_{\mathcal{I}_{k-1}}^t \mathbf{A}_{\mathcal{I}_{k-1}} & \mathbf{A}_{\mathcal{I}_{k-1}}^t \mathbf{a}_{i_k} \\ \mathbf{a}_{i_k}^t \mathbf{A}_{\mathcal{I}_{k-1}} & \mathbf{a}_{i_k}^t \mathbf{a}_{i_k} \end{array} \right] \right]^{-1} = \left[\begin{array}{cc} \mathbf{P}_{k-1} & \mathbf{q}_k \\ \mathbf{q}_k^t & 1 \end{array} \right]^{-1} \\
&= \left[\begin{array}{cc} [\mathbf{P}_{k-1}]^{-1} + \frac{1}{\beta_k} [\mathbf{P}_{k-1}]^{-1} \mathbf{q}_k \left[[\mathbf{P}_{k-1}]^{-1} \mathbf{q}_k \right]^t & -\frac{1}{\beta_k} [\mathbf{P}_{k-1}]^{-1} \mathbf{q}_k \\ -\frac{1}{\beta_k} \left[[\mathbf{P}_{k-1}]^{-1} \mathbf{q}_k \right]^t & \frac{1}{\beta_k} \end{array} \right]
\end{aligned}
\tag{13}
$$

[34] D.L. Donoho, et. al., "Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit," http://www-stat.stanford.edu/∼donoho/reports.html, March 2006.

[35] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sampling signal reconstruction," *IEEE Trans. Info. Theory*, vol. 55, no. 5, pp. 2230-2249, 2009.

[36] D. Needell and R. Vershynin, "Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit ," *IEEE Journal of Selected Topics in Signal Proc.*, vol. 4, no. 2, pp. 310-316, 2010.

[37] D. Needell and J.A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," Tech. Report, no. 2008-01, California Inst. Tech., July 2008.

[38] M.A. Davenport and M.B. Wakin, "Analysis of orthogonal matching pursuit using the restricted isometry property," *IEEE Trans. Info. Theory*, vol. 56, no. 9, pp. 4395-4401, 2010.

[39] S. Chatterjee, D. Sundman and M. Skoglund, "Look ahead orthogonal matching pursuit," *Proc. ICASSP*, pp. 4024-4027, 2011.

[40] T. Blumensath and M. Davies, "Iterative thresholding for sparse approximations," *J. Fourier Anal. and Appl.*, vol. 14, no. 5, pp. 629654, 2008.

[41] T. Blumensath and M. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, pp. 265274, 2009.

[42] D.G. Mixon, W.U. Bajwa and R. Calderbank, "Frame coherence and sparse signal processing," *accepted for ISIT 2011, http://www.duke.edu/ wb40/pubs/isit11_coherence.pdf*.

[43] W.U. Bajwa, R. Calderbank and D.G. Mixon, "Two are better than one: Fundamental parameters of frame coherence," http://arxiv.org/pdf/1103.0435v1.

[44] S. Boyd and L. Vandenberghe, "Convex optimization," Cambridge University Press, 2004.

[45] G. Reeves and M. Gastpar, "A note on optimal support recovery in compressed sensing," *Proc. Asilomar conf. Signals, Systems, Computers*, pp. 1576-1580, 2009.

[46] J.D Haupt, R.G Baraniuk, R.M. Castro and R.D. Nowak, "Compressive distilled sensing: Sparse recovery using adaptivity in compressive measurements," *Proc. Asilomar conf. Signals, Systems, Computers*, pp. 1551-1555, 2009.

[47] S. Chatterjee, D. Sundman and M. Skoglund, "Statistical post-processing improves basis pursuit denoising performance," *Proc. ISSPIT*, pp. 23-27, Dec 2010, Luxor, Egypt.

Mikko Vehkaperä Mikko Vehkaperä received the M.Sc. and Lic. Sc. degrees from University of Oulu, Finland, in 2004 and 2006, respectively, and the Ph.D. degree from Norwegian University of Science and Technology (NTNU) in 2010. Since 2010 he has been with the School of Electrical Engineering and ACCESS Linnaeus Center at Royal Institute of Technology (KTH), Sweden. From 2012 to 2013, he shares a research position between KTH and Aalto University School of Electrical Engineering, Finland. He held visiting appointments at Massachusetts Institute of Technology (MIT), US, Kyoto University, Japan, and University of Erlangen-Nuremberg, Germany. His research interests are in the field of wireless communications and signal processing.

Saikat Chatterjee Saikat Chatterjee is a researcher at Communication Theory Lab, KTH-Royal Institute of Technology, Sweden where he also pursued one year post-doctoral study. He was also with the Sound and Image Processing Lab at the same institution as a post-doctoral fellow for one year. Before moving to Sweden, he received Ph.D. degree in 2009 from Indian Institute of Science, India. He was a co-author of the paper that won the best student paper award at ICASSP 2010. His current research interests are source coding, speech and audio processing, estimation and detection, sparse signal processing, compressive sensing and wireless communications.

Dennis Sundman Dennis Sundman is a PhD student at the Communication Theory Lab at KTH - Royal Institute of Technology, Sweden, which he joined shortly after receiving his MSc degree in 2008. His research interest are towards compressive sensing and sensor networks.

Mikael Skoglund Mikael Skoglund (S'93-M'97-SM'04) received the Ph.D. degree in 1997 from Chalmers University of Technology, Sweden. In 1997, he joined the Royal Institute of Technology (KTH), Stockholm, Sweden, where he was appointed to the Chair in Communication Theory in 2003. At KTH, he heads the Communication Theory Lab and he is the Assistant Dean for Electrical Engineering.

Dr. Skoglund's research interests are in the theoretical aspects of wireless communications. He has worked on problems in source-channel coding, coding and transmission for wireless communications, Shannon theory and statistical signal processing. He has authored some 260 scientific papers, including papers that have received awards, invited conference presentations, and papers ranking as highly cited according to the ISI Essential Science Indicators. He has also consulted for industry, and he holds six patents.

Dr. Skoglund has served on numerous technical program committees for IEEE conferences. During 2003-08 he was an associate editor with the IEEE Transactions on Communications and he is presently on the editorial board for IEEE Transactions on Information Theory.