Batch Look Ahead Orthogonal Matching Pursuit

Muralikrishnna G. S. *, Sooraj K. Ambat†, K.V.S. Hari‡
*National Institute of Technology, Tiruchirappalli, 620 015, India
†Naval Physical and Oceanographic Laboratory (NPOL), DRDO, Kochi, 682 021, India
‡ SSP Laboratory, Dept. of ECE, Indian Institute of Science, Bangalore, 560 012, India
Email: *muralikgs11@gmail.com, †sooraj.k.ambat@ieee.org, ‡hari@ece.iisc.ernet.in

***Abstract*—Compressed sensing (CS) is a sampling paradigm that enables sampling signals at sub Nyquist rates by exploiting the sparse nature of signals. One of the main concerns in CS is the reconstruction of the signal after sampling. Many reconstruction algorithms have been proposed in the literature for the recovery of the sparse signals - Basis Pursuit, Orthogonal Matching Pursuit (OMP), Look Ahead Orthogonal Matching Pursuit (LAOMP) are some of the popular reconstruction algorithms. LAOMP, a modification of OMP, improves the reconstruction accuracy of OMP by employing a *look ahead* procedure. But LAOMP suffers from the drawback of being very expensive in terms of the computational time. In this paper we propose a modified version of the LAOMP algorithm called Batch-LAOMP which has a lesser computational complexity and also gives better performance in terms of reconstruction accuracy as seen from the results of the numerical experiments.**

keywords - Compressed Sensing, OMP, LAOMP, Batch-LAOMP

I. INTRODUCTION

In Compressed sensing, a sparse vector is to be reconstructed from incomplete and inaccurate measurements . This demands to solve an undetermined system of linear equations, in the presence of noise. Some of the most popular algorithms for the reconstruction problem are, Basis Pursuit (a convex relaxation to l_0 norm minimization) [1], Matching Pursuit (MP) [2], Orthogonal Matching Pursuit (OMP) [3], Look Ahead OMP (LAOMP) [4], Subspace Pursuit (SP) [5], CoSaMP [6], FACS [7] etc. These algorithms can be broadly classified into Convex relaxation methods [1] and greedy pursuit algorithms [2]–[7]. In convex relaxation methods, the ℓ_1 norm of the vector to be reconstructed is minimized, subject to the constraint that the reconstructed vector gives the same measurement vector under the sampling. The convex relaxation methods are generally more computationally complex than greedy pursuit algorithms. Therefore, in most practical cases, greedy pursuit algorithms are preferred. Greedy pursuit algorithms iteratively try to estimate the best index set based on the measurements. The size of the index set increases after every iteration. In the case of OMP, this increment is one. Whereas, in SP and CoSaMP the index set size grows by more than one after every iteration. Least square estimate is used for finding the reconstructed vector once the index set is identified. There have been many modifications proposed to OMP which either improve its accuracy or the speed of the algorithm. LAOMP is one such algorithm which improves the reconstruction accuracy of OMP using a *LookAheadResidue*, but this comes at the cost of increased computational complexity (further explanation in Section III). In the proposed algorithm (Batch-LAOMP), instead of selecting the atom which gives the least final residue, a variable number of atoms are selected (described in Section IV). The following Section describes the numerical conventions used and the framework of compressed sensing. OMP and LAOMP are discussed in Sections III and in IV the proposed algorithm is described in detail. The setup for the numerical experiment followed by the results of the experiment are discussed in Sections V and VI.

*The first author was an intern from May to July, 2017 at the Statistical Signal Processing Lab at the Indian Institute of Science, Bangalore and was supported by a joint fellowship from the Indian Academy of Sciences (IAS), the National Academy of Sciences India (NASI) and the Indian National Science Academy (INSA).

II. COMPRESSED SENSING

In the framework of compressed sensing, an unknown sparse vector $\mathbf{x} \in \mathbb{R}^N$, of sparsity K ($\|\mathbf{x}\|_0 \leq K$), is to be estimated from a set of incomplete measurements known as measurement vector $\mathbf{y} \in \mathbb{R}^M$, where $M \ll N$, obtained by sampling the vector $\mathbf{x}$ using a measurement matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$, the measurement matrix is sometimes also called as an overcomplete dictionary. The above relationship can be formulated by the following matrix equation,

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

where $\mathbf{w} \in \mathbb{R}^M$ is the additive Gaussian noise. With this setup, the task at hand is to obtain an estimate of $\mathbf{x}$, say $\hat{\mathbf{x}}$, from $\mathbf{y}$ and $\mathbf{A}$, assuming the sparsity (K) is known beforehand. With $M \ll N$ there are infinitely many possible solutions to the inverse problem of (1). This obstacle can be overcome by only considering vectors that are K sparse

978-1-5386-1224-8/18/$31.00 ©2018 IEEE

and constructing an overcomplete dictionary that satisfies the *Restricted Isometry Property* (RIP) [8] of order $2K$. A matrix $\mathbf{A}$ satisfies RIP of order K if,

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

for all $\mathbf{x} \in \Sigma_K$, where Σ_K denotes the space of all K-sparse vectors and $\delta_K \in (0,1)$. If $\mathbf{A}$ satisfies RIP of order $2K$ then the distance between any two K-sparse vectors is approximately preserved under the transformation. Hence it is possible to recover $\mathbf{x}$ with high probability, provided the size of the measurements (M) is large enough. Checking if a matrix satisfies RIP is a computationally expensive task (NP-Hard). Fortunately it is known that matrices filled with independent and identically distributed (i.i.d) random variables sampled from Gaussian or Bernoulli's distribution satisfy the *Restricted Isometry Property*.

A. Notational Convention

The notations used in the paper are as follows,

- **Vectors**, denoted using bold faced lowercase letters, ($\mathbf{x}$). $\|\mathbf{x}\|_p$ denotes the l_p norm of the vector $\mathbf{x}$.
- **Matrices**, denoted using bold faced uppercase letters, ($\mathbf{A}$). $\mathbf{A}^\dagger$ denotes the pseudo-inverse of $\mathbf{A}$, and $\mathbf{a}_p$ denotes the p^{th} atom of the matrix $\mathbf{A}$.
- **Sets**, denoted by uppercase Greek alphabets, (Γ). $|\Gamma|$ denotes the cardinality of the set, Γ^c denotes the complement of the set. The notation $\mathbf{A}_\Gamma$ denotes the matrix consisting of columns of $\mathbf{A}$ indexed by the elements of Γ. Similarly, $\mathbf{x}_\Gamma$ denotes a vector whose components are that of $\mathbf{x}$ indexed by Γ.

III. Previous Work

A. Orthogonal Matching Pursuit

In OMP [3], the atom giving the maximum inner product with the current residue is selected from $\mathbf{A}$ and is taken to be a part of the predicted support set at the end of each iteration. The coefficient corresponding to the selected atom is computed by taking the inner product between the residue and the chosen atom. The residue is then updated by subtracting its projection onto the space spanned by the chosen columns, from itself. There are many ways to define the stopping criteria for OMP, the most popular choice is to limit the size of the index set to K (sparsity of the vector $\mathbf{x}$). The main drawback of OMP is that the atom having the highest inner product may not always be a part of the original index set of $\mathbf{x}$, as the matrix $\mathbf{A}$ is not a full rank matrix. The performance of OMP for the cases when non-zero elements of $\mathbf{x}$ are sampled from (i) Gaussian distribution and (ii) the set $\{-1,+1\}$ is shown in figures 1 and 2.

Algorithm 1 Look Ahead Residue [4]

Input: The measurement matrix $\mathbf{A} \in \mathbb{R}^{M\times N}$, the measurement vector $\mathbf{y} \in \mathbb{R}^M$, the intermediate support set Γ, the chosen atom Ω and the sparsity of the vector K.
Output: The final reconstruction residue $\mathbf{r}$ and the predicted support set Γ.
Initialization :
$\Gamma \leftarrow \Gamma \cup \Omega$
$n \leftarrow |\Gamma|$
$\mathbf{r}_n \leftarrow \mathbf{y} - \mathbf{A}_\Gamma \hat{\mathbf{x}}_\Gamma$
LOOP Process
while $n < K$ **do**
  $n \leftarrow n+1$
  $l_n \leftarrow$ index of column corresponding to the largest value of $\mathbf{A}_{\Gamma^c}^T \mathbf{r}_n$.
  $\Gamma \leftarrow \Gamma \cup l_n$
  $\hat{\mathbf{x}}_\Gamma \leftarrow \mathbf{A}_\Gamma^\dagger \mathbf{y}$, $\hat{\mathbf{x}}_{\Gamma^c} \leftarrow 0$
  $\mathbf{r}_n \leftarrow \mathbf{y} - \mathbf{A}_\Gamma \hat{\mathbf{x}}_\Gamma$
end while
return $\mathbf{r}, \Gamma$

B. Look Ahead OMP

To overcome the drawback of OMP, Look Ahead Orthogonal Matching Pursuit (LAOMP) was proposed in [4]. Like OMP, a single atom is selected at every iteration, with the difference being in the selection process of the atom. Indices corresponding to L atoms having the largest inner products with the current residue are chosen and are then successively passed onto a look ahead residue procedure (explained in the next section) where an estimate of the final reconstruction residue is made. This is done by assuming the selected atom to be a part of the support set of $\mathbf{x}$. Among the chosen atoms, the atom for which the look ahead procedure returns the least norm residue is taken to be a part of the support set of $\mathbf{x}$. The look ahead residue procedure is illustrated above as algorithm 1.

LAOMP performs better than OMP and has much lesser reconstruction errors but is significantly more computationally complex than OMP. Hence, it takes a much longer time than OMP to arrive at the solution. The parameter L provides a trade-off between complexity and reconstruction accuracy, higher the value of L higher reconstruction accuracy but also high computational complexity. When L is low the computational complexity is low, but the reconstruction accuracy is compromised. Generally, $L \leq K$.

Algorithm 2 Batch Look Ahead Orthogonal Matching Pursuit

Input: The measurement matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$, the measurement vector $\mathbf{y} \in \mathbb{R}^{M}$, the sparsity of the vector K and the look ahead parameter L.

Output: The reconstructed vector $\hat{\mathbf{x}}$ and the predicted support set Γ

Initialization :

$n \leftarrow 0$

$\mathbf{r}_0 \leftarrow \mathbf{y}$

LOOP Process

while $n < K$ **do**

  $\Omega \leftarrow$index of L columns corresponding to L largest values of $\mathbf{A}_{\Gamma^c}^T \mathbf{r}_n$.

  for $j = 1$ to L **do**

    $[\mathbf{r}_j, \Lambda_j] \leftarrow \text{LookAheadResidue}(\mathbf{A}, \mathbf{y}, \Gamma, \Omega_j, k)$

    $n_j \leftarrow \|\mathbf{r}_j\|_2$

  end for

  $p = \underset{j}{\operatorname{argmin}}\, n_j$

  $\tilde{\Gamma} \leftarrow \Lambda_p \cap \Omega$

  $\Gamma \leftarrow \Gamma \cup \tilde{\Gamma}$

  $\hat{\mathbf{x}}_\Gamma \leftarrow \mathbf{A}_\Gamma^\dagger \mathbf{y}$, $\hat{\mathbf{x}}_{\Gamma^c} \leftarrow 0$

  $\mathbf{r}_n \leftarrow \mathbf{y} - \mathbf{A}_\Gamma \hat{\mathbf{x}}_\Gamma$

  $n \leftarrow |\Gamma|$

end while

return $\hat{\mathbf{x}}, \Gamma$

IV. Batch LAOMP

Based on the look ahead procedure developed in LAOMP, we propose Batch Look Ahead Orthogonal Matching Pursuit (BLAOMP), which iteratively finds the index set of the vector $\mathbf{x}$. In each iteration, a set of L atoms ($\Omega := [\mathbf{a}_1, \ldots, \mathbf{a}_L]$) that give the largest inner products with the current residue are selected and are passed onto to the *LookAhead-Residue*() as defined in [1], with the difference here being that the index set of the reconstructed vector is also returned by the procedure. The *LookAhead-Residue*() assumes the chosen atom to be a part of the index set of $\mathbf{x}$ and uses OMP [6] for computing the remaining elements in the support of $\mathbf{x}$ and the final predicted residue $\mathbf{r}$. The reconstruction can be made to converge faster if more than one atoms are taken in each iteration. This is followed in few greedy pursuit algorithms like SP [5] and CoSaMP [6], but the strategy used in the proposed algorithm is different from that employed in (SP and CoSaMP). Generally, the atoms which results in high magnitudes of inner product with the residue have a high chance of being a part of the original support set of the vector x. In LAOMP, when only one of the L atoms are selected, there is a very high chance that some of the $L-1$ atoms which were ignored before would again appear in the subsequent iterations. The strategy employed here is to select the atoms in the intersection of the index set (Λ_p) corresponding to the atom ($\mathbf{a}_p$) resulting in the least norm residue and the L chosen atoms (Ω) to be part of the of the index set predicted (Γ). This reduces the number of iterations required for the algorithm to converge without sacrificing on the performance that is observed in LAOMP. The algorithm stops once the size of the support set becomes equal to K. The complete algorithm is shown as Algorithm 2.

In the next section the results of testing the algorithm for various types of inputs are discussed and its performance is compared with that of few other algorithms.

V. Numerical Experiment

A series of computer simulations were performed for testing the algorithm and the performances of the following algorithms were compared: Basis Pursuit, OMP, LAOMP and the proposed algorithm. All the algorithms were implemented in MATLAB and for basis pursuit ℓ_1-magic [9] (a MATLAB implementation of ℓ_1 norm optimization) was used. The analysis of the algorithms was done using the following metrics,

- **Average Cardinality Error** (ACE) [7], $\mathbb{E}[\mathbf{CE}]$, where $\mathbf{CE} = d(\mathbf{x}_i, \hat{\mathbf{x}}_i) = 1 - \frac{|\Gamma_i \cap \hat{\Gamma}_i|}{K}$, where K is the sparsity of the $\mathbf{x}$. Γ_i and $\hat{\Gamma}_i$ denote the support of $\mathbf{x}_i$ and $\hat{\mathbf{x}}_i$ respectively. Hence, $\mathbf{ACE} = \mathbb{E}[\mathbf{CE}] = \frac{1}{T}\sum_{i=1}^{T} d(\mathbf{x}_i, \hat{\mathbf{x}}_i)$, where T is the total number of iterations. The ACE measures the average mismatch between the support set predicted by the algorithm and the actual support set of the vector $\mathbf{x}$.
- **Average Signal to Reconstruction-Error Ratio** (ASRER) [7], $\mathbb{E}[\mathbf{SRER}]$, where $\mathbf{SRER} = \frac{\|\mathbf{x}\|_2}{\|\mathbf{x}-\hat{\mathbf{x}}\|_2}$.
- **Computation time**, the time taken (in seconds) for the algorithm to find the reconstructed vector $\hat{\mathbf{x}}$ averaged over T iterations.

In the experimental setup, the sparse vector $\mathbf{x}$ is generated by sampling the non-zero entries from: (i) Standard normal distribution, $\mathcal{N}(0,1)$ and (ii) the set $\{-1, +1\}$. The over-complete dictionary is constructed with the entries of the matrix filled with independently and identically distributed (i.i.d.) random variables as discussed in Section II, ($a_{i,j} \sim \mathcal{N}(0,1)$). With the values of N and K fixed the performance is measured by varying M. We define fraction of measurements α as, $\alpha = \frac{M}{N}$.

The results of the numerical experiment are discussed in the next section.

VI. Results

In the conducted experiments, the values of N and K were taken to be 500 and 20 respectively. α is varied from 0.1 to 0.2 in steps of 0.01 when non-zero values of $\mathbf{x}$ are taken from $\mathcal{N}(0,1)$, and α is varied from 0.1 to 0.3 when the non-zero values are sampled from the set $\{-1,+1\}$. For each value of α, the over-complete dictionary was generated as described in the previous section. With, T = 500, for each matrix $\mathbf{A}$, 500 vectors ($\mathbf{x}$) of sparsity $K = 20$ were generated. The support of $\mathbf{x}$ was formed by taking a random subset of the set $\{1, 2, \ldots, N\}$. For the measurement of ASRER, Gaussian noise with SMNR = 41 dB was added to the measurements, where SMNR (signal to measurement noise ratio) is defined as in [7], $SMNR = \frac{\mathbb{E}(\|\mathbf{x}\|_2^2)}{\mathbb{E}(\|\mathbf{w}\|_2^2)}$. The performance metrics were computed for each iteration and the values reported are averaged over 500 iterations.

A. Cardinality Error

The figures 1 and 2 show the average cardinality error for the various fraction of measurements considered in the absence of noise, the relative performance of the algorithms are very similar when noise is present. From the results, it can be seen that when $\mathbf{x}_\Gamma \sim \mathcal{N}(0,1)$ BLAOMP outperforms BP, OMP and LAOMP and converges to zero cardinality error much quicker than the other algorithms (around $\alpha = 0.17$). When $\mathbf{x}_\Gamma \sim \{-1,+1\}$ the performance of the proposed algorithm is almost similar to that of LAOMP. The reconstruction accuracy increases, for a fixed value of α, when the value of the look ahead parameter L is increased. With the general trend being, as L increases the fraction of measurements (α) for which the cardinality error drops below a certain value decreases.

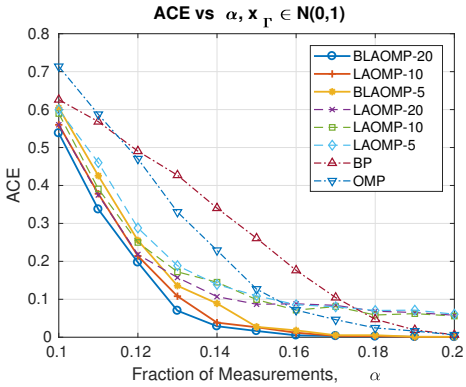

Figure 1: Average CE plotted over Fraction of measurements when the non-zero elements of $\mathbf{x}$ are sampled from Normal distribution, $\mathbf{x}_\Gamma \sim \mathcal{N}(0,1)$ for clean measurements. The number next to the algorithm in LAOMP and BLAOMP denote the value of L.

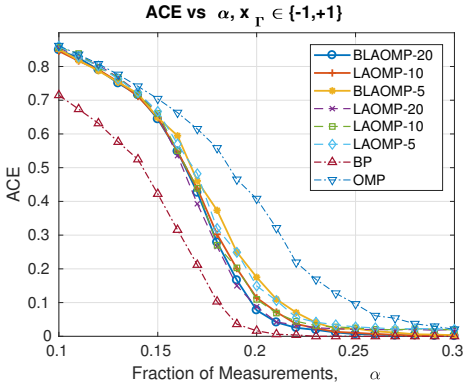

Figure 2: Average CE plotted over Fraction of measurements when the non-zero elements of $\mathbf{x}$ are either +1 or -1 , $\mathbf{x}_\Gamma \sim \{-1,+1\}$ for clean measurements. The number next to the algorithm in LAOMP and BLAOMP denote the value of L.

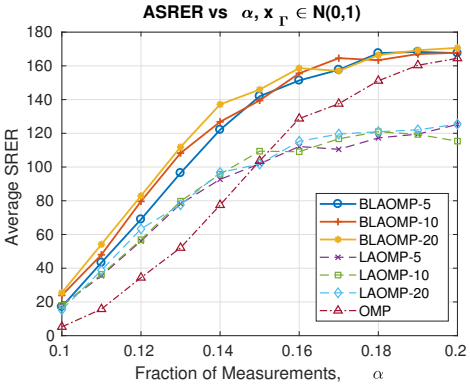

Figure 3: Average SRER plotted over Fraction of measurements when the non-zero elements of $\mathbf{x}$ are sampled from Gaussian distribution, $\mathbf{x}_\Gamma \sim \mathcal{N}(0,1)$, SMNR = 41 dB. The number next to the algorithm in LAOMP and BLAOMP denote the value of L.

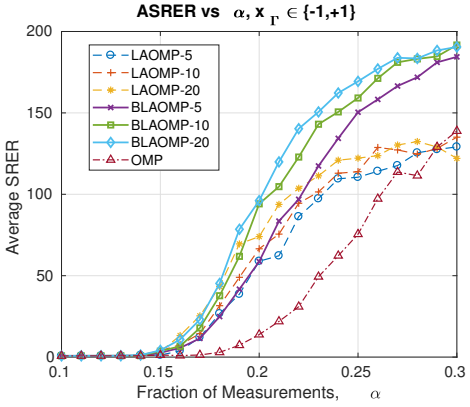

Figure 4: Average SRER plotted over Fraction of measurements when the non-zero elements of $\mathbf{x}$ are sampled from $\{-1,+1\}$, $\mathbf{x}_\Gamma \sim \{-1,+1\}$, SMNR = 41 dB. The number next to the algorithm in LAOMP and BLAOMP denote the value of L.

B. Signal Reconstruction-Error Ratio

Figures 3 and 4 show the relative SRER performances of LAOMP, OMP and the proposed algorithm when SMNR is fixed to be 41 dB and $\mathbf{x}$ being a Gaussian and Rademacher sparse vectors respectively. It is clearly seen that BLAOMP per-

Table I: Comparison of Computation Time (in seconds) between BLAOMP and LAOMP

Algorithm	$\alpha = 0.15$	$\alpha = 0.16$	$\alpha = 0.17$	$\alpha = 0.18$	$\alpha = 0.19$	$\alpha = 0.20$
BLAOMP-20	0.286s	0.269s	0.258s	0.239s	0.234s	0.218s
LAOMP-20	1.487s	1.402s	1.464s	1.491s	1.510s	1.495s
BLAOMP-10	0.192s	0.179s	0.179s	0.225s	0.171s	0.149s
LAOMP-10	0.651s	0.748s	0.758s	0.722s	0.754s	0.713s
BLAOMP-5	0.146s	0.137s	0.136s	0.128s	0.119s	0.115s
LAOMP-5	0.356s	0.350s	0.368s	0.380s	0.380s	0.370s

forms better than the other two for almost all values of α for Gaussian sparse vectors. For Rademacher sparse vectors BLAOMP performs atleast as good as LAOMP untill around $\alpha = 0.2$, after which it overtakes LAOMP interms of SRER reconstruction accuracy. Hence, the proposed algorithm is much more robust to addition of Gaussian noise than LAOMP and OMP.

C. Computation Time

The time taken (in seconds) by the proposed algorithm and LAOMP (implemented on MATLAB 2015a running on a MS Windows 8.1 operating system with 8 GB of RAM and an Intel Core i5 processor) is shown in table I, it can be clearly seen that BLAOMP computes the vector $\hat{\mathbf{x}}$ much faster than LAOMP while also providing an improvement in the reconstruction accuracy when $\mathbf{x} \sim \mathcal{N}(0, 1)$ and maintaining the same level of reconstruction accuracy when $\mathbf{x} \sim \{-1, +1\}$ as discussed in the previous subsection. When the parameter L is increased, the time taken for LAOMP to compute the reconstructed vector increases as the complexity is directly proportional to L. Whereas in BLAOMP, eventhough the computation time increases with increase in L, this increase is much smaller than that observed in LAOMP, as the number of iterations to converge to the result decreases. Even with $L = 20$, BLAOMP takes a much smaller computation than LAOMP (with $L = 5$).

In the spirit of reproducible research, all the MATLAB files used for the numerical experiments are provided in the following Github Repository: https://github.com/muralikgs11/BLAOMP.git

VII. CONCLUSIONS

In this work we proposed a novel sparse signal reconstruction algorithm (BLAOMP) which removes the redundant iterations in LAOMP and thereby the convergence time is reduced. From the results of the numerical experiments it is seen that, Batch-LAOMP is able to achieve an improved performance over LAOMP for Gaussian sparse vectors, and similar performance to LAOMP for Rademacher sparse signal, in a much shorter time.

REFERENCES

[1] S. S. Chen, D. L. Donoho, and M. A. Saunders, Atomic decomposition by basis pursuit, SIAM journal on scientific computing, vol. 20, no. 1, pp. 3361, 1998.

[2] S. G. Mallat, and Z. Zhang, Matching Pursuit With Time-Frequency Dictionaries, IEEE Trans. Signal Processing, vol. 41, no. 12, 1993, pp. 3397.

[3] 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.

[4] S. Chatterjee, D. Sundman, and M. Skoglund, Look Ahead Orthogonal Matching Pursuit, IEEE Int. Conf. Acoustics, Speech, Signal Processing, 2011, pp. 4024.

[5] W. Dei, and O Milenkovic, Subspace Pursuit for Compressed Sensing Signal Reconstruction, IEEE Trans. Information Theory, vol. 55, no. 5, 2009, pp. 2230.

[6] D. Needell, and J.A. Tropp, COSAMP: Iterative signal recovery from incomplete and inaccurate Measurements, Appl. Comput. Harmon. Anal. 26 (2009) pp. 301-321.

[7] S. K. Ambat, S. Chatterjee and K. V. S. Hari, "Fusion of Algorithms for Compressed Sensing," in IEEE Transactions on Signal Processing, vol. 61, no. 14, pp. 3699-3704, July15, 2013.

[8] E.J. Candes, J. Romberg, and T. Tao, Robust Uncertainty Principles: Exact Signal Reconstruction From Highly Incomplete Frequency Information, IEEE Trans. Information Theory, vol. 52, no. 2, Feb 2006.

[9] E.J. Candes, and J. Romberg, $l1$-MAGIC: Recovery of Sparse Signal via Convex programming, A document in l1-Magic toolbox shared online, http://www.acm.caltech.edu/l1magic/, Oct 2005