

## A Sparsity-Independent Greedy Compressive Sensing algorithm for Cognitive Radio: A subjective approach

Charushila Axay Patel<sup>1</sup>, C. H. Vithalani<sup>2</sup> and Preetida V. Jani<sup>3</sup>

<sup>1</sup>BVPIT (DS), Umarakh, Bardoli, Surat, Gujarat Technological University, Ahmedabad, Gujarat, India.

<sup>2</sup>GEC, Rajkot, Gujarat Technological University, Ahmedabad, Gujarat, India.

<sup>3</sup>SVIT, Mumbai University, Mumbai, Maharashtra, India.

### ARTICLE INFO

#### Article history:

Received: 10 March 2017;

Received in revised form:  
23 April 2017;

Accepted: 4 May 2017;

#### Keywords

Compressive Sensing(CS),  
Cognitive Radio (CR),  
Orthogonal Matching Pursuit  
(OMP),  
Karhunen-Loève  
Transform(KLT), Sparsity,  
Measurement Matrix,  
Measurement Vector,  
Sparsity Order,  
Signal Recovery,  
Signal Reconstruction,  
Probability,  
Incoherence,  
Threshold,  
Cross Validation,  
Complexity.

### ABSTRACT

This paper presents a Compressive Sensing (CS) greedy iterative algorithm, based on OMP: Sparsity-Independent-OMP (SI-OMP), for varying sparsity spectral conditions. As CS is one of the most essential techniques used by a Cognitive Radio (CR) for efficient usage of spectrum, it is required to be optimally simple, and, still, swift in working. The complexity here refers to the Number of computations a CR is required to make while using such algorithms and, this also, will in turn affect the effective requirement of hardware and power consumption. The proposed algorithm introduces negligible additional complexity, but enables a significant performance improvement in the reconstruction accuracy for arbitrarily varying spectral conditions. The spectrum here is a function of time and frequency both, exhibiting varying sparsity in both the domains.

© 2017 Elixir All rights reserved.

## I. Introduction

### A. Background

Cognitive radio systems typically involve primary users of the spectrum, who are incumbent licensees and secondary users who seek to use the spectrum opportunistically when the primary users are idle. The introduction of cognitive radios inevitably creates increased interference and thus can degrade the quality-of-service of the primary system. The impact on the primary system, for example in terms of increased interference, must be kept at a minimal level. It is required therefore, that cognitive radios sense and check the spectrum availability, and must be able to detect even very weak primary user signals. Thus, spectrum sensing is one of the most essential functionalities of cognitive radio. [1]

Wideband Spectrum Sensing is the technique, which suggests the spectrum that we have to sense will have the frequency bandwidth more than the coherent bandwidth of the channel. The typical narrowband sensing techniques are limited in the way that they make use of single binary decision and cannot detect individual spectrum opportunity available in the wideband spectrum. [2]

Recently, compressed sensing/compressive sampling (CS) has been considered as a promising technique to improve and implement cognitive radio (CR) systems. In the area of signal processing, compressed sensing is one of the significant technique for extracting and reconstructing a signal by exploring the solution to underdetermine significant linear systems.

The central problem of compressed sensing (CS) is to estimate an unknown signal  $x \in \mathbb{R}^N$  from  $m$  linear measurements  $y = (y_1, \dots, y_m)$  given by  $y = \Phi x + w$ , where,  $x \in \mathbb{R}^N$  is a sparse vector, meaning its number of nonzero components  $K$  is smaller than  $N$ . The support of  $x$  is the locations of the nonzero entries and is sometimes called its *sparsity pattern*. And  $\Phi \in \mathbb{R}^{M \times N}$  is a known measurement matrix,  $y \in \mathbb{R}^M$  represents a vector of measurements and  $w \in \mathbb{R}^M$  is a vector of measurements errors (noise).

Over the years, the OMP (Orthogonal Matching Pursuit) algorithm has long been considered as a heuristic algorithm hard to be analyzed. Recently, however, many efforts have been made to discover the condition of OMP ensuring the exact recovery of sparse signals.

In one direction, studies to identify the recovery condition using probabilistic analyses have been proposed. Tropp and Gilbert showed that when the measurement matrix  $\Phi$  is generated at random and the measurement size is about  $K \log N$ , OMP ensures the accurate recovery of every fixed  $K$ -sparse signal with overwhelming probability. [3]

### Research Gap

It can be seen that existing studies towards implementing compressive sensing on signal processing do subsist with advantages as well as limitations too. However, a closer look into the studies taking place until date was found with a noticeable research gap: [4]

- Less Effective Survey
- Less focus on Reconstruction
- Ambiguity in implementing Sparsity matrix

### B. Motivation for this work

Considering the research gap mentioned above, reconstruction ability of the technique is one aspect we can enhance further in CS using the OMP based scheme.

Besides, there exists a broad spectrum of applications that involve non-Gaussian, heavy-tailed processes, which seems to expand very rapidly. Examples of such applications are: wireless communications, tele-traffic, hydrology, geology, atmospheric noise, economics and image and video processing, etc. This, simply, will lead to a spectrum condition where over-crowding will take place. That means very frequent use and non-use of the spectrum! And this means the sparsity of the spectrum, both in frequency domain and time domain!

In addition, sparsity is a simple but effective model for many real-life signals. For instance, a signal, or, an image may be larger, but when viewed in the right basis (e.g. a wavelet basis), many of the coefficients may be negligible, and so the signal, or, image may be compressible into a file of much smaller size without seriously affecting the information contained in it. In other words, many signals are effectively sparse in the wavelet basis. More complicated models than sparse signals can also be studied, but for simplicity, we will restrict attention to the sparse case here.

Intuitively, if a signal  $x \in \mathbb{R}^n$  is  $S$ -sparse, then it should only have  $S$  degrees of freedom rather than  $n$ . In principle, one should now only need  $S$  measurements or so to reconstruct  $x$ , rather than  $n$ . This is the underlying philosophy of compressive sensing: one only needs a number of measurements proportional to the compressed size of the signal, rather than the uncompressed size.

Moreover, compressed sensing is advantageous whenever signals are sparse in a known basis; measurements (or computation at the sensor end) are expensive; but computations at the receiver end are cheap.

Also, we should be able to estimate the sparsity because it helps address a wide range of issues:

- Modeling assumptions
- The Number of measurements
- The Measurement matrix
- Recovery algorithms

Therefore, the sparsity in frequency and time domain will be the main plausible concern in CS research. When we estimate a signal through a CS system, we assume that the sparsity level of a specific frequency spectrum is already given. This approach can fail when the sparsity assumptions given are invalid due to sparsity varying environment.

With all the advancements of the secondary usage of the spectrum will also increase, making the spectrum denser.

At this time, we will need the algorithm that can work well with varying sparsity levels.

Moreover, we consider the sparsity in time domain as well as one of the important factors to deal with. We have to take into account both deterministic (static), and, dynamic, scenario for change in the sparsity of the signal as we know that the spectrum usage varies with time largely.

Use of an appropriate sparsifying basis and measurement matrix, then, can be instrumental for improving the sensing performance and deal with sparsity variations.

Therefore, the algorithm, that is sparsity-robust and provides a satisfactory signal reconstruction is one, much anticipated development we can look for.

### C. Contribution of this work

As we have discussed above, the sparsity level variations, both in frequency and time domain, motivated us to work out an algorithm that is Sparsity-Independent. We have used a random-sparse spectrum in frequency, which varies randomly in time over a specific period. We are using a presentation basis, which can work well with both the time and frequency domain, thus addressing the sparsity in both time and frequency domain. The algorithm presents an improved performance while not increasing the complexity largely.

Besides, the spectral reconstruction and hence, recovery have also improved, giving a better detecting, or sensing, ability.

We have used the algorithm based on matching pursuit that decomposes any signal into a linear expansion of waveforms that belong to a redundant dictionary of functions.

The functions that are well-localized in both time and frequency, are called time-frequency atoms. Following this line of thought, we are proposing the use of wavelet transforms that offer considerably good time-frequency decomposition and localization of them.

### D. Organization of the paper

In this paper, in section II, we will discuss about the proposed new scheme based on OMP. In section III, we will be discussing the experimental set-up; in section IV we will encounter the performance metrics and results. The implications and conclusions will be discussed in section V. We will get to know the different parametric notations used throughout the paper through the table 1.

**Table 1. Notations.**

Parameters	Description
$x$	Input signal in frequency domain
$y$	Output signal in frequency domain
$w$	Noise in measurements (Measurement Error)
$\hat{x}$	Estimated signal in $\mathbb{R}^d$
$v$	$N \times 1$ dimensional Data Vector for OMP
$K$	Sparsity
$\Phi$	$M \times N$ Sparse Measurement Matrix
$M$	Measurement Vector length
$N$	Signal input length
$\phi_d$	Columns of the measurement matrix
$x_n$	Rows of the measurement matrix or measurement vectors for OMP
$a_m$	$N$ -dimensional approximation of $v$ for OMP
$r_m$	$N$ -dimensional residual for OMP
$\Lambda_0$	index set for OMP
$\lambda_t$	Index for OMP
$\eta$	Error tolerance for OMP

## II. Sparsity-Independent Orthogonal Matching Pursuit

We describe here a sparsity-independent greedy recovery algorithm for  $K$ -sparse signal and provide analyses for

parameter setting, computational complexity, and recovery performance.

The proposed method contains these steps: 1) Deciding the sparsity threshold heuristically, 2) Applying the sparsifying basis to the signal, 3) Applying measurement matrix to the sparsified signal, 4) Applying first iteration of the OMP, 5) Cross Validating the output, 6) Applying further iterations of OMP if required, 7) Checking for the algorithm stoppage criteria, and, 8) Recovering the signal and locating and estimating the K-non-zero frequency components output.

**A. System Model**

Suppose that a total spectrum of x Hz is considered to be shared among a number of primary and secondary users. This can be either an ad-hoc network sharing a total of x Hz spectrum among its nodes or a secondary network of cognitive radios trying to use the licensed spectrum opportunistically for secondary communication. Assume that each node in the ad-hoc network or each cognitive radio needs a bandwidth of B Hz for the communication. Define  $n \triangleq x/B$  to be the number of available channels and denote by  $f_i$  the center frequency of the  $i$ th channel. [5]

Therefore, let our  $x$  be the wide band signal in frequency domain as,

$$x = [x_1, x_2, \dots, x_n]^T \tag{1}$$

where  $n \triangleq x/B = 30$ , and,  $\{x_i\}$  are signal values in frequency domain uniformly sampled over B with spacing  $B/N$ , and indices  $\{i\}$  are related to frequency locations. In noiseless case, if  $|f_i|^2 \neq 0$ , then the spectrum is occupied at the  $i$ -th frequency location. Otherwise, ‘ $i$ ’ represents the unused frequency location that is accessible to secondary users.

**B. Problem Formulation**

As the CS framework depends on the sparsity of a signal majorly in frequency domain, we present here the sparsity of the vector in form of mathematical model.

**Definition**

Sparsity: An  $N$ -dimensional vector,  $x \in C^N$  is said to be  $K$ -sparse if it has  $K$  or fewer non-zero coordinates, i.e.,

$$\|x\|_0 := \lim_{p \rightarrow 0} (\|x\|_p^p + |x_2|^p + \dots + |x_N|^p)^{1/p} \leq K \ll N, \quad x \in C^N \tag{2}$$

where,  $\|\cdot\|_0$  denotes the  $\ell_0$ -norm which just counts the number of non-zero components in the vector and  $p$  is a constant that is traditionally used to parameterize the standard  $\ell_p$ -norm.

In practice, the signals often encountered are not exactly sparse, but are compressible (close to being sparse).

**Definition**

Compressibility: A vector is called compressible if its entries obey a power decay law

$$|x_{(i)}| \leq R_i r^{i-1}, \tag{3}$$

Where,  $|x_{(i)}|$  is the  $i$ th largest value of  $x$ , i.e.,  $(|x_{(1)}| \geq |x_{(2)}| \geq \dots \geq |x_{(N)}|)$ ,  $r > 1$ , and  $R_r$  is a positive constant which depends only on  $r$ .

This means only a few entries of a compressible vector are large while most of them are small. It should be noted that sparse signals are compressible.

Let us have our discrete-time signal  $x \in C^N$ , which we can expand in terms of an  $N \times 1$  orthonormal basis (e.g. wavelet,

Fourier) vectors  $\psi_{x_i}$  with  $i=1, \dots, N$  as

$$X = \sum_{i=1}^N s_i \psi_{x_i} \tag{4}$$

Where,  $s_i$  with  $i=1, \dots, N$  are the entries of coefficient sequence of  $X$ . Alternatively, stacking  $\psi_{x_i}$  for  $i=1, \dots, N$  as columns results in an  $N \times N$  sparsifying basis matrix  $\Psi_x =$

$[\psi_{x_1} \ \psi_{x_2} \ \dots \ \psi_{x_N}]$ . Then  $X$  can be represented in matrix-vector form as

$$X = \Psi_x s. \tag{5}$$

Sensing, i.e., of the time domain signal  $X$  is done by collecting measurements by correlating  $X$  with some sensing vectors  $\phi_i$  (waveforms in case of the continuous-time domain), i.e.,

$$\tilde{x}_i = \langle x, \phi_i \rangle \quad i=1, 2, \dots, M. \tag{6}$$

The approaches including greedy algorithms such as Orthogonal Matching Pursuit, Stage-wise OMP, or Iterative re-weighted algorithms, calculate the support of the signal iteratively, and work for a specific number of measurements

$$M = cK \log(N/K) \tag{7}$$

Where  $c$  is the over-measuring factor ( $c > 0$ , varies between 2 and 20 depending upon the recovery algorithm).

**Theorem**

(Neyman-Pearson approach to the binary hypothesis problem).

$$\text{Max } P_r(H_1; H_1) \tag{8}$$

$$\text{s.t. } P_r(H_1; H_0) = \alpha. \tag{9}$$

where the notation  $P_r(H_i; H_j)$  indicates the probability of deciding hypothesis  $H_i$  when  $H_j$  is true.

The classical Neyman-Pearson approach to binary hypothesis testing suggests maximizing  $P_r(H_1; H_1)$  with an upper-bound constraint on  $P_r(H_1; H_0)$ . The  $P_r(H_1; H_0)$  can be constrained by choosing an appropriate threshold for the decision. We now define the Likelihood Ratio (LR), which is required to understand the hypothesis testing.

**Definition**

Likelihood Ratio: Let  $y \in C^M$  be an observed vector of i.i.d. random variables from a certain distribution. The likelihood ratio is then given by

$$\lambda = \frac{\text{Pr}(y|H_1)}{\text{Pr}(y|H_0)}. \tag{10}$$

The function  $\lambda$  indicates for each value of  $y$  the likelihood of  $H_1$  versus the likelihood of  $H_0$ . The logarithm of the function  $\lambda$  is referred to as the Log-Likelihood Ratio (LLR).

The classical approaches for detection based on CS, recover the signal first by solving the optimization problem given as,  $\min_s \|s\|_p \text{ s.t. } \|\Phi \Psi_x s - \tilde{x}\|_2^2 < \epsilon$ . For example, this could be a simple threshold to just detect the presence or the absence of a signal. We try to avoid this redundant approach and reconstruct the signal and do the detection of the signal. We formulate this problem as Sparse Signal Reconstruction problem. This problem becomes intriguing when only  $M$  observations ( $M < N$ ) are available.

Ideally, we would like to measure all the  $n$  coefficients of  $f$ , but we only get to observe a subset of these and collect the data,

$$Y_k = \langle f, \phi_k \rangle, \quad k \in M, \tag{11}$$

Where,  $M \subset \{1, \dots, n\}$  is a subset of cardinality  $n < N$ .

For sparse signal recovery, among various algorithms used, the greedy iterative Orthogonal Matching Pursuit (OMP) is one of the most widely used recovery techniques.

**C. The Classic Orthogonal Matching Pursuit (OMP)**

As our work is based on classic, greedy CS algorithm, OMP, we will first take a look at this basic algorithm Signal recovery can be considered as a problem dual to sparse approximation. Since  $x$  has only  $K$  nonzero components, the data vector  $v = \Phi x$  is a linear combination of  $m$  columns from  $\Phi$ . In the language of sparse approximation, we say that  $v$  has an  $K$ -term representation over the dictionary  $\Phi$ . Therefore, for recovering sparse signals, we can make use of sparse approximation algorithms.

To identify the ideal signal  $s$ , we need to determine which columns of  $\Phi$  participate in the data vector  $v$ . The idea behind the algorithm is to pick columns in a greedy fashion. At each iteration, we choose the column of  $\Phi$  that is most strongly correlated with the remaining part of  $v$ . Then we subtract off its contribution to  $v$  and iterate on the residual. One hopes that, after  $m$  iterations, the algorithm will have identified the correct set of columns.

The MP (Matching Pursuit) is one of the basic greedy algorithms that find one atom at a time. In OMP, following steps of the algorithm we find the one atom that best matches the signal given the previously found atoms. While in the next step, it finds the next one, to best fit the residual. [3]

### Algorithm 1: OMP for Compressive Sensing

#### Presentation Matrix and Sensing Matrix Application

##### Input:

- $x = wbs$
- $RIC = \delta = \text{delta} = 0.36$

##### Output:

- The  $M \times N$  Measurement Matrix,  $\Theta = \Phi$
- The  $N \times 1$  dimensional data vector  $b$

##### Procedure:

##### Initialize:

- Wavelet Decomposition Level,  $K=1$

##### Decompose:

- Wavelet Decomposition of  $x$ , with  $db1$ , for generating Sparse Representation Matrix,  $\psi$

##### Calculate:

- Measure the sparsity level,  $\text{sprlvl}$ , of input using Gini Index method
- If Calculate the Size of the Dictionary,  $N = \text{length}(C)$ ,  $C = \text{final decomposed signal}$
- Take  $t_n = \text{sprlvl}$
- Calculate the Number of Measurements,  $M = (1 + \text{delta}) * 4 * t_n * \log(N)$

##### Design:

- Design  $M \times N$  Sensing Matrix  $\Phi$
- Find  $N \times 1$  dimensional data vector  $b = \Phi * C$

#### Basic OMP Algorithm

##### Input:

- The  $M \times N$  measurement matrix  $\Theta = \Phi$
- The  $N \times 1$  dimensional data vector  $b$
- The sparsity level  $K$  of the ideal signal
- Maximum number of iterations  $m$
- Error tolerance  $\eta$ .

##### Output:

- An estimate  $\hat{x}$  in  $R^d$  for the ideal signal
- A index set  $\Lambda_m$  containing  $m$  elements from  $\{1, \dots, d\}$
- An  $N$ -dimensional approximation  $a_m$  of the data  $b$
- An  $N$ -dimensional residual  $r_m = b - a_m$

##### Procedure:

##### Initialize:

The index set  $I = \emptyset$  and the residual  $r = b$

- The set of non-zero elements as empty,
- The index set  $\Lambda_0 = \emptyset$ , and,
- Iteration count  $t=1$ .

##### Repeat:

- The following, 'K' times:

##### ➤ Identify

- Find the index  $\lambda_t$  that solves the easy optimization problem,

$$\lambda_t = \arg \max_{j=1, \dots, d} |\langle r_{t-1}, \phi_j \rangle|.$$

- If the maximum occurs for multiple indices, break the tie deterministically.

##### ➤ Update

- Add to the index set and the matrix of chosen atoms:

$$\Lambda_t \leftarrow \Lambda_{t-1}, \quad \lambda_t \text{ and } \Phi_t \leftarrow [\Phi_{t-1}, \phi_{\lambda_t}].$$

We here consider that  $\Phi_0$  is an empty matrix.

- A least square problem is solved to obtain a new signal estimate:

$$x_t \leftarrow \arg \max_x \|b - \Phi_t x\|_2.$$

- Calculate the new approximation of the data and the new residual

$$a_t \leftarrow \Phi_t x_t$$

$$r_t \leftarrow b - a_t.$$

- $t \leftarrow t+1$ , and find the new index  $\lambda_t$ , if  $t < m$ .

The estimate for the ideal signal has nonzero indices at the components listed in  $\Lambda_m$ . The value of the estimate in component  $\lambda_j$  equals the  $j$ th component of  $x_t$ .

- Return if  $t \geq m$ .

OMP, as it promises, gives us an algorithm with less complexity and considerably good recovery performance. With the technique tested for its performance through simulations, we found that the algorithm provides reconstruction accuracy between the two, up to around 0.7. Nevertheless, this is a remarkable performance; still it requires priory information regarding the sparsity level of the signal/ spectrum to be sensed.

In a dynamically varying scenario of secondary spectrum usage, this can practically limit us to first check the sparsity and then get the detection, introducing redundancy in the process. For avoiding the redundant process of sparsity estimation, we present the algorithm, which is Sparsity Information Independent, rather, Sparsity-Independent.

**A. The Solution to Sparse-Spectrum Recovery Problem: The Sparsity-Independent Orthogonal Matching Pursuit (S-I-OMP)**

There are two significant challenges: 1) choosing an appropriate number of sub-Nyquist measurements, and 2) deciding when to terminate the greedy recovery algorithm that reconstructs wideband spectrum.

The classical OMP provides us with a considerably good performance in signal recovery. However, it requires the prior knowledge of the sparsity of the spectrum, which it is supposed to sense. This makes the situation difficult when we have an environment where the sparseness of the signal keeps on changing. We are proposing an OMP-based greedy algorithm which will not require the advanced knowledge of the sparsity of the spectrum.

**The main difference between the classical OMP and SIOMP lies there in four major considerations:**

1. The application of adaptive sparsifying basis or presentation matrix
2. The measurement matrix selection based on incoherence with the sparsifying basis
3. The application of Residue Threshold as algorithm stopping criteria

4. The Cross Validation is applied for overcoming the over-fitting of the noise by terminating the iterations in time avoiding the requirement of priory knowledge of sparsity level.

The sparsity-robust OMP algorithm here, enables a CR to automatically choose the number of measurements while guaranteeing the wideband spectrum recovery with a small predictable recovery error. This is realized by the proposed measurement infrastructure and the validation technique.

The key difference between the classical OMP and the proposed SI-OMP is that the proposed algorithm can find a good spectral estimate by using only a small testing subse Furthermore, the algorithm is able to recover the wideband spectrum without requiring knowledge of the instantaneous spectral sparsity level. Such an algorithm bridges the gap between CS theory and practical spectrum sensing. The proposed recovery algorithm can autonomously adopt a proper number of iterations, therefore solving the problems of under-fitting or noise over-fitting, which commonly exist in most greedy recovery algorithms. The halting criterion plays a crucial role in determining the performance of the Adaptive Compressive Spectrum Sensing framework. To improve the energy efficiency of CRs, we hope that the compressive sampling can come to a halt at the earliest appropriate time such that the current spectral estimate is a good estimate. The residue threshold is also used as a stopping criterion. Moreover, Cross Validation technique is also proposed for limiting the number of iterations carried out by the algorithm, which specifically avoids the need for prior knowledge of sparsity level. Furthermore, before applying the algorithm, we define a specific sparsity threshold before sparsifying basis application and we can convert the sparse spectrum in to the spectrum with a specific sparsity level that our algorithm may handle with efficiently.

Besides, we are using the noiselets and/ or circulant matrix for forming the measurement matrix for taking the advantage of incoherence between the presentation matrix and sensing matrix for better sensing performance.

Sparsity-Robust OMP for Compressive Sensing
Presentation Matrix and Sensing Matrix Application
<p><b>Input:</b></p> <ul style="list-style-type: none"> <li>▪ <math>x=wbs</math></li> <li>▪ Sparsity Threshold, Sprth, heuristically (in terms of Gini Index (GI))</li> <li>▪ <math>RIC=\delta=\Delta=0.36</math></li> </ul> <p><b>Output:</b></p> <ul style="list-style-type: none"> <li>▪ The <math>M \times N</math> Measurement Matrix, <math>\Theta=\Phi</math></li> <li>▪ The <math>N \times 1</math> dimensional data vector <math>b</math></li> </ul> <p><b>Procedure:</b></p> <p>Initialize:</p> <ul style="list-style-type: none"> <li>▪ Wavelet Decomposition Level, <math>K=1</math></li> </ul> <p>Decompose:</p> <ul style="list-style-type: none"> <li>▪ Wavelet Decomposition of <math>x</math>, with db1, for generating Sparse Representation Matrix, <math>\psi</math></li> </ul> <p>Calculate:</p> <ul style="list-style-type: none"> <li>▪ Measure the sparsity level, sprlvl, of output using Gini Index method</li> <li>▪ <b>If</b> sprlvl &lt; sprth, <math>K \leftarrow K+1</math>, and further decomposition of <math>x</math>, <b>Else</b>,</li> <li>▪ Calculate the Size of the Dictionary, <math>N=length(C)</math>, <math>C=</math>final decomposed signal</li> <li>▪ Calculate <math>tn=</math>final sparsity level of decomposed signal</li> <li>▪ Calculate the Number of Measurements, <math>M= (1 + \Delta)*4*tn*\log(N)</math></li> </ul> <p>Design:</p> <ul style="list-style-type: none"> <li>▪ Design <math>M \times N</math> Sensing Matrix <math>\Phi</math> using Noiselets, and/ or Circulant Matrix</li> <li>▪ Find <math>N \times 1</math> dimensional data vector <math>b= \Phi*C</math></li> </ul>
Main Sparsity-Independent OMP-based Algorithm
<b>Input:</b>

- The  $M \times N$  measurement matrix  $\Theta = \Phi$
  - The  $N \times 1$  dimensional data vector  $b$
  - Maximum number of iterations  $m$
  - Error tolerance  $\eta$
  - Cross Validation Matrix  $A_{CV}$
  - $y_{CV} = A_{CV}x + n_{CV}$  (where  $n_{CV}$  is Cross Validation measurement noise)
  - Count 'p' and final count 'q'
  - $\epsilon_{CV}^0 = \|y_{CV}\|_2^2$
- Output:
- An estimate  $\hat{x}$  in  $R^d$  for the ideal signal
  - A index set  $\Lambda_m$  containing  $m$  elements from  $\{1, \dots, d\}$
  - An  $N$ -dimensional approximation  $a_m$  of the data  $b$
  - An  $N$ -dimensional residual  $r_m = b - a_m$
- Procedure:
- Initialize:
- The index set  $I = \emptyset$  and the residual  $r = b$
  - The set of non-zero elements as empty,
  - The index set  $\Lambda_0 = \emptyset$ , and,
  - Iteration count  $t=1$
  - Set Cross Validation Count  $p=1$
  - The threshold for residual,  $R$ , heuristically:  $R=0$  for noiseless channel and  $R$ =energy of noise for AWGN channel
  - $\sigma = 0.001$ , heauristically
- Repeat:
- The following, 'K' times:
    - **Identify**
      - Find the index  $\lambda_t$  that solves the easy optimization problem,
      - $\lambda_t = \arg \max_{j=1, \dots, d} |\langle r_{t-1}, \phi_j \rangle|$ .
      - If the maximum occurs for multiple indices, break the tie deterministically.
    - **Update**
      - Add to the index set and the matrix of chosen atoms:
      - $\Lambda_t \leftarrow \Lambda_{t-1}, \lambda_t$  and  $\Phi_t \leftarrow [\Phi_{t-1}, \phi_{\lambda_t}]$ .
      - We here consider that  $\Phi_0$  is an empty matrix.**
      - A least square problem is solved to obtain a new signal estimate:
      - $x_t \leftarrow \arg \max_x \|b - \Phi_t x\|_2$ .
      - Calculate the new approximation of the data and the new residual
      - $a_t \leftarrow \Phi_t x_t$
      - $r_t \leftarrow b - a_t$ .
      - If  $\epsilon_{CV}^0 > \|A_{CV} \hat{x}^p - y_{CV}\|_2^2$ , then, set new CV residual corresponding to  $\hat{x}^p$ ,
      - $\epsilon_{CV}^p = \|A_{CV} \hat{x}^p - y_{CV}\|_2^2$
      - $p \leftarrow p+1$ , until  $p \geq q$
      - Compute  $\epsilon_{CV} = \arg \min_p \epsilon_{CV}^p$
      - Compute the difference between the recovered signal,  $\hat{x}^p$  and input signal,  $x$ ,
      - $\Delta x^p \triangleq x - \hat{x}^p$ , corresponding to  $p$ .
      - Compute the recovery error,  $\epsilon_x^p \triangleq \|\Delta x^p\|_2^2$ .
      - **Return if  $\epsilon_x^p \leq 0.1\sigma$ , and Output  $\hat{x}^p = \hat{x}^{o_{CV}}$  and terminate the algorithm. Else,**
      - Compare the norm of  $r_t$  with  $R$ .
      - **If  $r_t >$  threshold  $R$ , keep on iterating**
      - $t \leftarrow t+1$ , and find the new index  $\lambda_t$  if  $t < m$ .
      - **If  $t \geq m$  Or the energy of residual  $r_t <$  threshold  $R$ , Return**

### III. The Experimental Setup and Simulations

We carried out simulations for Classical OMP and SI-OMP for signal recovery for spectrum sensing purpose.

We did run the simulations for two types of environment for the proposed SI-OMP: Static scenario in which the frequency-domain sparsity level assumes a specific value and remains static at that level, and, Dynamic scenario in which the frequency-domain sparsity level keeps changing with time over a specific time-period of simulations.

The simulation was carried out on Matlab 2011b, and later, Matlab 2014a version for both the algorithms. The system was running on the processor Intel Core i5, M460, with a 64-bit OS, and 2.53 GHz Clock Speed.

We are constructing a wideband signal, which can be varied randomly in terms of frequencies. We have considered that if the signal changes constantly in time and assumes dynamic nature, then how our algorithm will react.

Here is a signal in the transform domain, which has a fixed number  $k$  of non-zeros, all of equal amplitude, located in random positions. We are then applying the CS framework, and vary  $n$ , the number of samples used in the sensing scheme. The CS matrix  $\Phi$  is constructed, which has its columns drawn independently at random.

These columns come from a uniform distribution found on the unit sphere  $S^{n-1}$  in Euclidean  $n$ -space.

For deterministic experimentation we construct, at baseband, a wideband spectrum range [0 MHz – 60 MHz] containing 30 channels of 2 MHz each of static nature and encode it as  $c = c_1, c_2, \dots, c_n$ ; where  $n = 30$ . Every channel may be possibly occupied by a Primary User (PU) using digital modulation scheme either 16-PSK or 16-QAM. Therefore, the symbol rate will be 2 MHz, number of samples per symbol will be 16, and number of symbols in a frame can be chosen to be 512. Here, we shall consider the Nyquist sampling frequency,  $f_s = 128$  MHz and the sampling number,  $N = 8192$ .

While for the subjective approach, we are generating a spectrum with frequency domain sparsity levels that vary randomly with time. It is a wideband spectrum range [0 GHz – 60 GHz] containing 30 channels of 2 GHz each and define them as  $c = c_1, c_2, \dots, c_n$ ; where  $n = 30$ . Every channel may

**Table 2. Simulation Parameters for Experimental Setup for checking the performances of classical OMP and SI-OMP.**

Parameter	Value
The Wide Band Spectrum Band	60 MHz/ 60 GHz
Number of Channels	30
Band Width of a single channel	2 MHz/ 2 GHz
Modulation Scheme	16-QAM
Number of Samples per Symbol Symbol, M	16
Number of Symbols per Frame	512
Sampling Number, N	8192
Sampling Frequency, Fs	128 MHz/ 121.14 GHz
SNR	5 dBm
<b>Sensing Parameters</b>	
Channel Type	AWGN
Representation Matrix	Daubechies Wavelets (1 <sup>st</sup> -4 <sup>th</sup> Order)
Sensing Matrix	Random Sparse Matrix (constructed using Noiselets and/ or Circulant Matrix) for SI-OMP)

#### IV. The Performance Metrics and Results

The results are depicted here in form of graphical plots for both the algorithms, for different sparsity levels in frequency domain with a static spectrum. These are expressed in terms of the Time taken for simulation (sec) and the reconstruction accuracy between the input and output of the algorithms.

We can know the speed of algorithm with help of the time taken by it for the simulation. This indicates how quickly the algorithm succeeds in recovering the unknown input spectral component when implemented in a cognitive radio working as a secondary user and it indicates how fast the particular algorithm will be out of the two we have considered.

The reconstruction accuracy on the other hand, is used here to show how much the algorithm succeeds in recovering the unknown input spectrum.

The reconstruction accuracy is derived for both the algorithms using the `xcorr` and `mscohere` commands in Matlab. It is derived by taking the root, mean and square of the two outputs obtained by the two commands. The `xcorr` gives the correlation between the two signals (here the input and output spectrums) in terms of cross-correlation between the two random processes. On the other hand, the `mscohere` finds the magnitude squared estimate of the input signals  $x$  and  $y$  (here the input and output spectrums) indicating how well the two signals correspond to each other at each frequencies.

We observed that measuring sparsity by counting nonzeros is inherently limited. While it is a seemingly ideal approach to measuring sparsity, in practice, real signals will not typically have exact zeros anywhere in the transform. Hence, results of the type just shown, while instructive, are of limited practical interest.

be possibly occupied by a Primary User (PU) using digital modulation scheme either 16-PSK or 16-QAM.

The simulation parameters are tabulated in Table 2. Let number of samples per symbol be  $M = 16$  and let  $N =$  Sampling Number = 8192 for Gaussian random matrix for an AWGN channel.

With this regards, the sparsity level is measured by the Gini Index, instead of counting the non-zeros. As the Gini index is one of the most reliable measures for sparsity, we have opted to use it here.

Gini index is used to express the percentage of sparsity that gives its value in true sense. It is one of the most dependable measures for sparsity, [6]

$$GI(x) = 1 - \frac{2 \sum_{k=1}^N \frac{|x_k|}{\|x\|_{l_1}} \left( \frac{N-k+1/2}{N} \right)}{N} \quad (12)$$

Here the vector  $\underline{f} = [f(1), f(2), \dots, f(N)]$  is given and we have re-ordered and represented its elements using  $f_{[k]}$  for  $k = 1, 2, \dots, N$ , where  $|f[1]| \leq |f[2]|, \dots, \leq |f[N]|$ , and  $\|\underline{f}\|_1$  is the  $l_1$  norm of the function  $\underline{f}$ .

The Gini index possesses the values between zero to one. Therefore, percentage representation required multiplication with 100. For us, Sparsity  $K$  is defined in terms of Gini Index in Table 3.

**Table 3. The sparsity orders considered in terms of GINI index and number of active spectrum components.**

Gini Index (%)	Number of Active (Static) Spectrum Components (out of 30)
100	1
79.3103	7
62.069	12
41.3793	18
20.6897	24
0	30

#### Performance Analysis

To evaluate efficiency of the algorithm, it can be taken into account properties of the algorithm (complexity, velocity or speed, memory consumption), the amount of compression, and that how closely the reconstruction resembles the original signal. In this work, we will focus on the complexity of the algorithms, speed, and, a quantification of the difference/ similarity between the original signal and its reconstruction after compression. Moreover, the probability of detection  $P_D$ , probability of missed detection  $P_M$ , and, the probability of false alarm  $P_F$ , or their average across channels, are among the prominent performance metrics used to characterize the wideband sensing performance of these algorithms.

#### The parameters affecting the sensing performance

Before we can analyze the performances of these two algorithms for the static scenario, and compare them, we must get the insight of the experimental parameter considerations used here for performing the sensing.

As we have applied the algorithms to a wideband signal, for compressed sensing, we have represented with sparsifying basis and then sensed by measurement matrix. These two procedures affect the sensing performance of the two algorithms. Therefore, we will have to introduce the basics of the two actions we perform. Let us have a quick look into these two:

#### The Sparsifying Basis (DWT):

For both this algorithms, we sparsify the signal before application to the algorithms. This we have done to make it sure that the signal under consideration is sparse in actual sense.

The Discrete Wavelet Transforms (DWTs) are used as sparsifying basis. It does possess its own excellent time frequency localization property. Usage of DWT in 1D signal corresponds to 1D filter in every dimension. The use of DWT as sparsifying basis enables us to remove the blocking objects. It is useful as a tool in addressing the sparse nature of the spectrum in both frequency and time domain.

The computational complexity of applying DWT basis for sparsification one time is equivalent to  $O(N)$ , where  $N$  is the number of time samples.

### Restricted Isometry Property and the Number of measurements and the Measurement matrix

As an alternative to coherence and to probabilistic analysis, a large number of algorithms within the broader field of CS have been studied using the restricted isometry property (RIP) for the matrix  $\Phi$  [7]. A matrix  $\Phi$  satisfies the RIP of order  $K$  if there exists a constant  $\delta \in (0,1)$  such that,

$$(1 - \delta) \|x\|_2^2 \leq \|\Phi x\|_2^2 \leq (1 + \delta) \|x\|_2^2 \quad (13)$$

holds for all  $x$  such that  $\|x\|_0 \leq K$  [17]. In other words,  $\Phi$  acts as an approximate isometry on the set of vectors that are  $K$ -sparse. Much is known about finding the matrices satisfying the RIP. For example, if we draw a random  $M \times N$  matrix  $\Phi$  whose entries  $\phi_{ij}$  are independent and identically distributed sub-Gaussian random variables, then provided that

$$M = O(K \log(N/K) / \delta^2) \quad (14)$$

with high probability,  $\Phi$  will satisfy the RIP of order  $K$  [17]. When it is satisfied, the RIP for a matrix provides a sufficient condition to guarantee successful sparse recovery using a wide variety of algorithms [7].

The Number of measurements is decided based on the criteria of noise present in the measurement matrix. If the noise is absent, i.e., the measurements are error-free, the number of measurements can be considered to be

$$M \geq (1 + \delta) 4K \log(N) \quad (15)$$

While we expect the noise to be present, the Number of measurements can be,

$$M \geq (1 + \delta) 2K \log(N - K) \quad (16)$$

The RIC (the Restricted Isometry Constant),  $\delta$  here, was taken to be 0.36.

As Tropp and Gilbert have shown that when the number of measurements scales as  $M \geq (1 + \delta) 4K \log(N)$  for some  $\delta > 0$ ,  $A$  has i.i.d. Gaussian entries, and the measurements are noise-free ( $w = 0$ ), the OMP method will recover the correct sparse pattern of  $x$  with a probability that approaches one as  $n$  and  $k \rightarrow \infty$ .

• The number of measurements is defined based on this relation for our proposed CS scheme.

### Incoherence between Sparsifying and Measurement Matrices

One other consideration is the incoherence between the representation (sparsifying) matrix and sensing (measurement) matrix. The higher the incoherence between the two, the better the recovery of the signal will be! For this reason, we have used the Noiselets and the Circulant Matrix as the sensing or measurement matrix.

#### The importance of Incoherence

Fix  $f \in \mathbb{R}^n$  and suppose that the coefficient sequence  $x$  of  $f$  in the basis  $\Psi$  is  $K$ -sparse. Select  $m$  measurements in the  $\Phi$  domain uniformly at random. Then if

$$M \geq C \cdot \mu^2(\Phi, \Psi) \cdot K \cdot \log N \quad (17)$$

Here three things can be stated: (1) The role of the coherence is completely transparent; the smaller the coherence, the

fewer samples are needed, hence our emphasis on low coherence systems in the previous section. (2) One suffers no information loss by measuring just about any set of  $m$  coefficients, which may be far less than the signal size apparently demands. If  $\mu(\Phi, \Psi)$  is equal or close to one; then on the order of  $K \log N$  samples suffice instead of  $n$ . (3) The signal  $x$  can be exactly recovered from our condensed dataset while we do not assume any knowledge about the number of nonzero coordinates of  $x$ , their locations, and their amplitudes which we assume are all completely unknown a priori.

We propose the use of wavelets bases for  $\Psi$ , the presentation matrix, and noiselets for  $\Phi$ , the sensing matrix. Here, the coherence between noiselets and Haar wavelets (or, say, db1) is  $\sqrt{2}$ , and that between noiselets and Daubechies D4 and D8 wavelets is respectively about 2.2 and 2.9 across a very wide range of sample sizes  $n$ . This extends to higher dimensions as well. (Noiselets are also incoherent with spikes to a great extent and incoherent with the Fourier basis.) Our interest in noiselets comes from the fact that, (1) they are incoherent with systems providing sparse representations of image data and other types of data, and (2) they can be associated with very fast algorithms; the run-time for the noiselet transform is  $O(n)$ , and just like the Fourier transform, the noiselet matrix does not need to be stored when it's needed to be applied to a vector. This is of crucial importance for efficient numerical computations without which CS would not be very practical. [8]

### Use of Noiselets for designing the Measurement Matrix [9]

To improve the incoherence between the sparsifying matrix and sensing (measurement matrix), we propose to use the noiselets as the bases for measurement matrix.

❖ The incoherent condition means that the rows  $\phi_j$  of the matrix  $\Phi$  cannot sparsely represent the elements of the sparsity-inducing basis  $\psi_i$ , and vice versa.

❖ Improvement in incoherence between the sparsifying matrix and sensing matrix will improve the compressive sensing performance of the algorithm.

Noiselets can totally spread out the signal energy in the measurement domain and are identified to be maximally incoherent with the Haar wavelet. The mutual incoherence parameter between the noiselet measurement matrix  $\Phi$  and the sparsifying Haar wavelet transform matrix  $\Psi$  is shown to be equal to 1, which is the minimum value possible for the incoherence. Also for Daubechies wavelets, for db2, it is 2.2; and, for db4, it is 2.9. Therefore, theoretically, noiselets are the best suited measurement basis function for CS systems where the wavelet is used as sparsifying transform matrix. [10]

#### Motivation

The motivations behind using noiselets as a measurement matrix in CS are as follows [9]:

- Noiselet basis function is unitary and , therefore, it does not amplify noise as in the case of random encoding.
- Noiselets completely spreads out the signal energy in the measurement domain and are maximally incoherent with wavelets.
- Contrasting the random basis, noiselet basis has conjugate symmetry. Thus, this property of symmetry can be exploited by using the partial Fourier like technique.
- Noiselets are derived in the same way as wavelets, therefore it can be modeled as a multi-scale filter-bank and can be applied in  $O(N \cdot \log(N))$ .

The CS algorithms can successfully recover the sparse signal from smaller number of measurements if the  $\phi_k$  are incoherent with respect to  $\Psi$  — in a general consideration; this means that the  $\phi_k$  are global and diverse in the  $\Psi$  domain.

#### Use of Circulant Matrix for designing sensing matrix

Optimal incoherence is attained by completely random measurement matrices. However, such matrices are often complicated and costly to implement in hardware realizations. Random Toeplitz and circulant matrices can be realized with far more simplicity, (or even naturally) in various applications. [11]

In CS applications, a physical implementation is required for the acquisition of the linear projections  $A\bar{x}$ . In most cases, the use of an i.i.d.

Gaussian random matrix  $A$  is either unfeasible or overly expensive. This motivates the study of easily implementable CS matrices. One of such matrices is the Circulant matrix, which has been shown to be almost as efficient as the Gaussian random matrix for CS encoding/decoding. [11]

$$C = \begin{bmatrix} t_n & t_{n-1} & \dots & t_1 \\ t_1 & t_n & \dots & t_2 \\ \vdots & \vdots & \ddots & \vdots \\ t_{n-1} & t_{n-2} & \dots & t_n \end{bmatrix}$$

Here the  $C$  has its  $i$ th row formed by  $i-1$  times' circulant shifting its first row as shown above.

In various physical domains, it is easy to compute  $T_\Omega \bar{x}$ , where  $T$  is circulant and  $T_\Omega$  is its submatrix that contains a subset of rows of  $T$ . Since the multiplication  $T\bar{x}$  is equivalent to the discrete convolution operation  $h * \bar{x}$  for a certain vector  $h$ ,  $T_\Omega \bar{x}$  is the  $\Omega$ -subsample of the convolution  $h * \bar{x}$ . [11]

Tropp et al. described a random filter for acquiring a signal  $\bar{x}$ : a random vector  $h$ , called a random-tap FIR, is convolved with  $\bar{x}$  followed by down-sampling to yield the compressive measurements  $b$ . In this example, the sensing matrix is the circulant matrix induced by  $h$ . [12]

The use of circulant matrix allows for faster signal recovery.

However, we can see that the use of Circulant matrix for designing the measurement matrix does not assure the sparse outcome. But we can see from various examples that not having the sparse measurement matrix does not affect the basic concept of compressive sensing. [13]

#### The metrics for evaluating the sensing performance of the algorithm:

Here we have focused on the computational complexity of the algorithms, speed, and, a quantification of the difference/ similarity between the original signal and its reconstruction after the application of compressive sensing. In addition, the probability of detection,  $P_D$ , the probability of missed detection,  $P_M$ , and, the probability of false alarm,  $P_F$ ; or, their average across channels, are some of the prominent performance metrics used to characterize the wideband sensing performance of these algorithms.

We will now have a brief analysis of these performance metrics for the algorithms under consideration.

#### Computational Complexity

Normally the computational complexity is equivalent to the number of steps an algorithm takes to solve the problem as a function of the input size.

#### Classical OMP

At each stage, OMP computes residual correlations and solves a least-squares problem for the new solution estimate. OMP builds up the active set one element at a time. Hence, an efficient implementation would necessarily maintain a Cholesky factorization of the active set matrix and update it at each stage, thereby reducing the cost of solving the least-squares system. In total,  $k$  steps of OMP would take at most  $4k^3/3 + knN + O(N)$  flops. Without any sparsity assumptions on the data, OMP takes at most  $n$  steps, thus, its worst-case performance is bounded by  $4n^3/3 + n^2N + O(N)$  operations. [14]

#### Sparsity-Independent OMP (SIOMP):

The main difference between the classical OMP and SIOMP lies there in these major considerations: the application of adaptive sparsifying basis or presentation matrix, the measurement matrix selection based on incoherence with the sparsifying basis, the application of Residue Threshold as algorithm stopping criteria, and, the Cross Validation is applied for overcoming the over-fitting of the noise by terminating the iterations in time.

For an  $n$ -fold cross-validation, each single example occurs exactly  $n-1$  times as a training example. Hence, the time needed to evaluate all tests is reduced by a factor  $n-1$  compared to running the original algorithm  $n$  times. The time needed to assign examples to child nodes is reduced by  $n-1$  if the same test is selected in all folds, otherwise a smaller reduction occurs. Besides this speedup there are no changes in the computational complexity of the algorithm (except for the extra computations involved in, for instance, selecting elements from a two-dimensional array instead of a one-dimensional array). As with one-dimensional array, the computational complexity is around  $O(N)$ , while with two-dimensional array, it will be  $O(N^2)$ . Specifically for cross-validation, the algorithm can be further improved if the employed statistics  $S$ , for any data set  $D$ , can be computed from the corresponding statistics of its subsets in a partition. This holds for all statistics that are essentially sums, since in that case  $S(D) = \hat{a}_i S(D_i)$ . Such statistics could also be called *additive*.

In an  $n$ -fold cross-validation, the data set  $D$  is partitioned into  $n$  sets  $D_i$ , and the training sets  $T_i$  can be defined as  $D - D_i$ . It is then sufficient to compute statistics just for the  $D_i$ ; those for the  $T_i$  can be easily computed from this without further reference to the data (first compute  $S(D) = \hat{a}_i S(D_i)$ ; then  $S(T_i) = S(D) - S(D_i)$ ). Since each example occurs in exactly 1 of the  $D_i$ 's, updating statistics has to be done only  $N$  times instead of  $N(n-1)$  times (with  $N$  the number of examples). [15]

Bearing in mind the above steps required for the SIOMP, we have counted the number of FLOPs needed for the execution of the algorithm. This provides us the idea about the complexity of the algorithm. The total number of FLOPs required by SIOMP are approximately:  $O(N^2) + O(NM) + O(kmN(kN \log_2 N + k^3)) + O(N^2 \log_2 N) + O(N \log_2 N) + O(2NMk) + O(N \log_2 N)$ . [16]

We are taking into consideration the matrix format of the algorithm components, which are applied to recover the sparse signal back. As we already have discussed, the sparse representation and sensing, both take place in form of generation of relevant matrices. Therefore, we are taking the number of rows and columns of the matrices into the usage for finding the relationship for counting the number of FLOPs or the complexity of the algorithm.

Here,  $O(kMN(kN\log_2N+k^3))$  is the complexity of the classical OMP. The other terms indicate here that the complexity count adds up in the new proposed scheme. As the big-Oh notation is a function of variable 'N', the number of measurements, the other terms might be considered as constants and can be ignored. As  $O(N) \subseteq O(N\log N) \subseteq O(N^2) \subseteq O(N^2\log_2N) \subseteq O(N^3)$ , we can omit some terms out of the above expression.

As the Big-Oh notation refers to the worst case performance in terms of the time taken, or, space or memory occupied, by the algorithm, the constants are ignored and only the order of the variable (the 'O' stands for which) is taken into account. Therefore, we can consider the complexity of the proposed algorithm can be approximately  $O(kMN(kN\log_2N+k^3)) + O(N^2\log_2N)$ .

**Table 4. Various cases for simulation of SI-OMP with deterministic approach.**

<b>Case 1:</b>	The Number of Measurements for Cross Validation Matrix = Length of Measurement Vector
	AWGN Channel and Noiseless Measurement Matrix with noiselets and Circulant Matrix
	For Number of Measurements =M (M = round(N/2), N = length (C)), Sparsity Threshold=0.8
<b>Case 2:</b>	The Number of Measurements for Cross Validation Matrix = Length of Measurement Vector
	AWGN Channel and Noiseless Measurement Matrix with noiselets only
	For Number of Measurements =M (M = round(N/2), N = length (C)), Sparsity Threshold=0.8
<b>Case 3:</b>	The Number of Measurements for Cross Validation Matrix = Length of Measurement Vector
	AWGN Channel and Noiseless Measurement Matrix with Circulant Matrix only
	For Number of Measurements =M (M = round(N/2), N = length (C)), Sparsity Threshold=0.8
<b>Case 4:</b>	The Number of Measurements for Cross Validation Matrix = 90
	AWGN Channel and Noiseless Measurement Matrix with Noiselets and Circulant Matrix
	For Number of Measurements =M (M = round(N/2), N = length (C)), Sparsity Threshold=0.8
<b>Case 5:</b>	The Number of Measurements for Cross Validation Matrix = 90
	AWGN Channel and Noiseless Measurement Matrix with Noiselets only
	For Number of Measurements =M (M = round(N/2), N = length (C)), Sparsity Threshold=0.8
<b>Case 6:</b>	The Number of Measurements for Cross Validation Matrix = 90
	AWGN Channel and Noiseless Measurement Matrix with Circulant Matrix only
	For Number of Measurements =M (M = round(N/2), N = length (C)), Sparsity Threshold=0.8
<b>Case 7:</b>	The Number of Measurements for Cross Validation Matrix = 90
	AWGN Channel and Noiseless Measurement Matrix with Noiselets and Circulant Matrix
	For Number of Measurements =M (M = (1+δ)*4*K*log(N), N = length(C), δ=RIC=0.36)), Sparsity Threshold=0.99

$O(N^2) + O(NM) + O(kMN(kN\log_2N+k^3)) + O(N^2\log_2N) + O(N\log_2N) + O(2NMk) + O(N\log_2N) = O(\max(N^2 + NM + kMN(k\log_2N+k^3) + N^2\log_2N + 2N\log_2N + 2NMk))$ . (18)

From this we can find that  $O(\max(N^2 + NM + kMN(k\log_2N+k^3) + N^2\log_2N + 2N\log_2N + 2NMk)) \subseteq O(N^2\log_2N)$ . Therefore, we can conclude that the complexity of the proposed algorithm will be  $O(N^2\log_2N)$ .

Here we can consider that inclusion of Cross Validating concept in the algorithm has not affected the overall complexity of OMP itself in considerable way. The complexity that we have come to derive is affected by entire algorithm including the application of presentation and sensing matrices as well.

Other parameters used as performance metrics for the algorithm are, the Execution Time, or, Run Time; the Reconstruction Accuracy; and, the Reconstruction Error, for the algorithms.

For evaluating the above-mentioned parameters, we have taken these considerations or scenarios into account for simulations:

As we have considered the sparsification or presentation matrix adaptively, the input spectrum is sparsified adaptively based on the input sparsity level and predefined sparsity threshold that we are defining heuristically. Moreover, we are taking into consideration the effect of incoherence between the representation matrix and the sensing matrix. For this reason, we are applying two criteria: one, the Noiselets, and the other, the Circulant matrix. Both do possess a very small amount of coherence with wavelets, which we have used here for representation matrix basis.

Furthermore, we are applying some more algorithm stoppage criteria for the OMP. The residue we obtain is compared with the specified threshold and the number of

iterations is limited with the specific limit applied there. Besides these, we are using the cross validation (CV) technique for limiting the time taken by the OMP for signal recovery.

Based on these considerations, we have defined different seven situations, tabulated in Table 4.

The properties of the Big-Oh notation help us simplify the above expression. Rule for summation and Property 5 can Here, give us the equation in simplified form.

In the first case, we are using both the Noiselets and the Circulant Matrix for generating the Measurement or Sensing Matrix. In the second case, we are using the Noiselets only for getting the Measurement Matrix. In the third case, we are using the Circulant Matrix only to obtain the Measurement Matrix. While in the fourth case, we have used both the Noiselets and Circulant matrix for the above purpose; but with the Cross Validation matrix's number of measurements limited to 90. In the previous three cases, it was kept equal to the length of the measurement vector generated by the main algorithm. In the 5th case we are keeping the number of measurements in CV matrix, equal to 90 but we have used circulant matrix only to define the measurement matrix. While in the 6th case we have used noiseless only with CV matrix's number of measurements equaling 90. The seventh case has the number of measurements kept according to the formula discussed above using both Noiselets and Circulant matrix for designing the Measurement matrix keeping the CV MM's number of measurements equal to 90 again. Here we have experimented with the sparsity threshold keeping it 0.99.

We have made comparison of the results obtained for these different simulation scenarios with SIOMP with that of Classical OMP and Karhunen-Loève Transform (KLT) for Measurement Matrix without noise and with AWGN.

For the subjective approach, we have considered three cases with the construction of measurement matrix: 1. we have used only noiselet for constructing the measurement matrix, 2. we have used only circulant matrix for that, and, 3. we have used both noiselet and circulant matrix for constructing the measurement matrix.

The Execution Time or the Run Time of the algorithms (second):

This parameter defines the speed of the algorithm for solving the given problem. Here, we consider the sensing of the spectrum and detecting the spectral holes as the main objectives; as the device, using these algorithms for spectrum sensing will be a Cognitive Radio device and a non-licensed secondary user (SU).

In the dynamically changing environment of spectrum usage and allotments, it will be necessary to sense the spectrum and detect the opportunities for spectrum reuse, and make decisions quickly, within the least possible time duration. Hence, we consider the Run Time of the algorithm or Execution Time is one of the most important performance metrics. We express this parameter in the unit of seconds. The graphs depicting the results in Figure 1-a here indicate that the proposed Sparsity-Independent OMP (with the deterministic approach) takes runtime more than that taken by the Classical OMP and KLT.

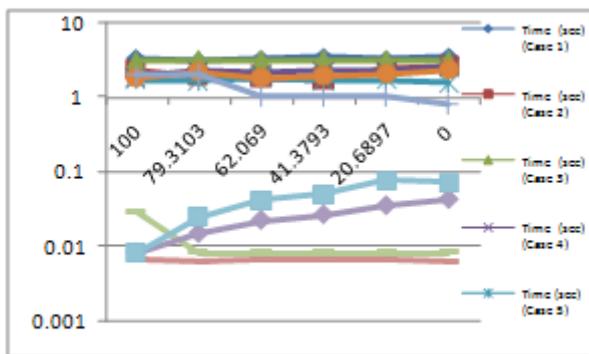


Figure 1a. Execution time for the SI-OMP with deterministic approach and OMP and KLT for various experimental considerations.

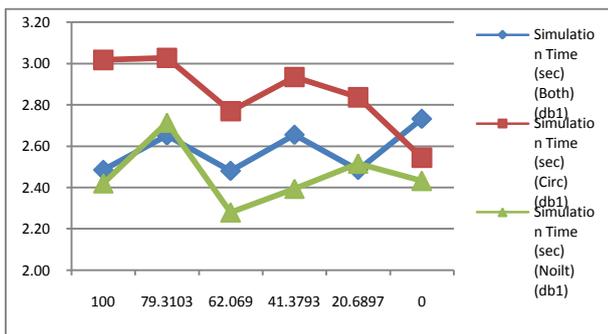


Figure 1b. Execution time for the SI-OMP with subjective approach for various experimental considerations.

The algorithm implemented with both the Noiselets and Circulant matrix used for Measurement Matrix (MM) design, takes the highest time gradually increasing with higher sparsity level (and lower Gini index values). When we have used Circulant Matrix only in Case 3, the time taken by the algorithm is high again but not affected by the sparsity levels.

With Noiselets only used for design of MM (Case 2), the time is quite less and it only goes high when sparsity levels are increased high. In the Case 4, we have used both the Noiselets and Circulant matrix for MM, taking the number of

measurements for Cross Validation (CV) measurement matrix in OMP equal to 90. This has shown a considerable effect as it requires quite a lesser time compared to that with Case 1. Consequently, we have used the same MM for CV in OMP and used only Circulant matrix for main MM design in Case 6. The time does not vary much considerably, but surely, it is lessened compared to the Case 3. On the other hand, with the same CV scenario in OMP, with only Noiselets being used as in Case 5, the time taken is reduced. For all these cases the number of measurements for the main MM, we have taken to be equal to  $N/2$ , where  $N$  is the length of the signal.

In case 7, however, we have used the formula depicted here, using both Noiselets and Circulant matrix for MM design. We can see the time getting reduced with increasing sparsity as well as on an overall account by a considerable amount.

For randomly varying sparsity levels with time, we can see that for all the three cases mentioned above, the execution time remains around 2-3 second for a specific discrete instance of sensing. For any sparsity input level, initially the execution time seems to be similar to that with deterministic approach. The graphs in the Figure 1-b present the resultant discrete simulation time durations for dynamically varying sparsity levels. The highest time around 2.8 to 3.0 seconds, though, seems to be taken by the algorithm with measurement matrix constructed using circulant matrix only, while using db1 as the sparsifying basis. While for the same algorithm with db2 as sparsifying basis, the time is as low as 2.22 seconds compared to that for the others.

**The Reconstruction Accuracy of the Algorithms**

The Reconstruction Accuracy here, is used to show how much the algorithm succeeds in recovering the unknown input spectrum. This in turn can indicate the spectrum detection ability of the algorithm.

The reconstruction accuracy is derived for both the algorithms using the `xcorr` and `mscohere` commands in Matlab. It is derived by taking the root, mean and square of the two outputs obtained by the two commands. The `xcorr` gives the correlation between the two signals (here the input and output spectrums) in terms of cross-correlation between the two random processes. On the other hand, the `mscohere` finds the magnitude squared estimate of the input signals  $x$  and  $y$  (here the input and output spectrums) indicating how well the two signals correspond to each other at each frequencies.

The reconstruction accuracy, or, the measure of similarity between the input and output spectrums to the algorithm, is obtained in a good measure. For almost all the cases for the proposed SI-OMP, the reconstruction accuracy has increased with increasing sparsity levels and it is almost the same.

We obtain the reconstruction accuracy values, which range from 0.95 to 0.99 for all these seven cases.

It shows that the algorithm performance has improved considerably as for Classical OMP it remained around 0.7 (with the MM with AWGN and without any noise) and for KLT (with the MM with AWGN and without any noise), with which, reconstruction accuracy varied drastically from 0.7 to 0.3 at different sparsity levels.

The reconstruction accuracy improvement as depicted in Figure 2-a, shows that the recovery of the spectrum can be done effectively with this proposed SI-OMP and this, in turn, helps to detect the spectral holes efficiently.

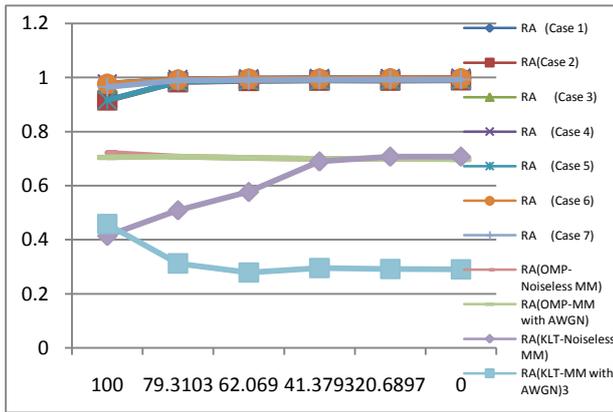


Figure 2a. Reconstruction Accuracy for the SI-OMP (with deterministic approach) and OMP and KLT for various experimental considerations .

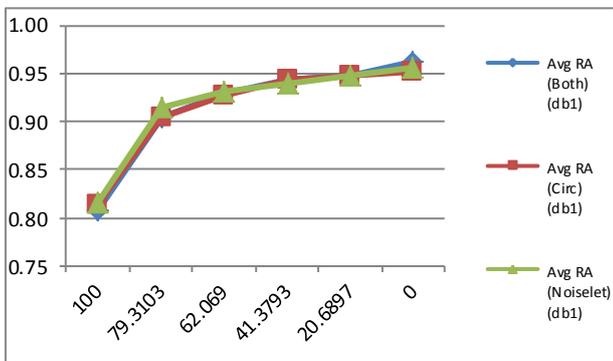


Figure 2b. Reconstruction Accuracy for the SI-OMP with subjective approach for various experimental considerations.

With the sparsity varying randomly with time, we can see the variations of reconstruction accuracy take place between 0.99 to 0.8 approximately. Mostly the values steadying above and around 0.9 while seldom dropping to 0.7 for all the cases mentioned above. This goes the same way for the different input initial sparsity levels. We can see that the reconstruction accuracy is varying in the same fashion for all different considerations, and it highest with the algorithm with measurement matrix constructed using circulant matrix only, while sparsification basis being db2. Figure 2-b shows the resultant graphs here.

The Reconstruction Error (%):

Here the error represents the dissimilarities between the input and output spectrums of the algorithms. It shows that how much these algorithms fall short of giving out the accurate spectrum reconstruction, and, consequently, recovery and it is expressed in percentage.

It will be natural to observe the variation pattern of this parameter be opposite to that of the Reconstruction Accuracy. Reconstruction error, shown in Figures 3-a-b gives us the measure of how much dissimilar the input and output spectrums to the algorithm are! The resultant plots in Figure 3-a, indicate that it is reduced very much considerably with the new SI-OMP, compared with that obtained with Classical OMP and KLT. It also shows that with much lesser Probability of Missed Detection, the spectral conditions can be detected efficiently.

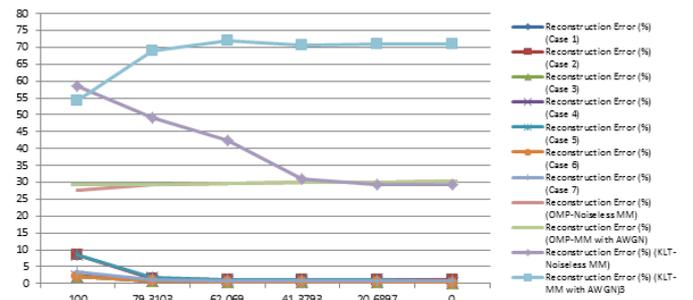


Figure 3a. Reconstruction Error for the SI-OMP (with deterministic approach) and OMP and KLT for various experimental considerations.

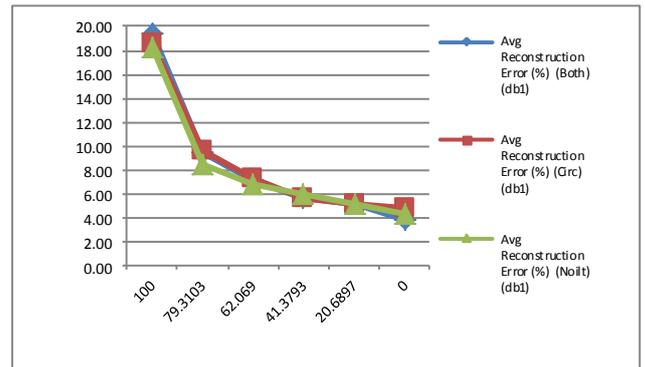


Figure 3b. Reconstruction Error for the SI-OMP with subjective approach for various experimental considerations.

As for the subjective approach, the reconstruction error, as shown in Figure 3-b, varies from 2.0 % to 20 % for all three above-mentioned cases. It normally varies around 5-10 % for most of the times and infrequently shoots up to the values beyond 25 %.

**The Probability of Missed Detection (P<sub>M</sub>) and The Probability of False Alarm (P<sub>F</sub>)**

The Probability of Missed Detection (P<sub>M</sub>) is actually the chance of missing the detection of any existing spectrum component actively present. While the Probability of False Alarm (P<sub>FA</sub>) is the chance where in the sensing process, the CR will get the detection of an active spectrum component even if the said component is not active.

In simplest form, spectrum sensing of a single channel is a binary hypothesis testing problem. Specifically,

$$H_0: y[n] = w[n], n = 1, N \tag{19}$$

$$H_1: y[n] = x[n] + w[n], n = 1, \dots, N. \tag{20}$$

Where, x[n] represents a primary user's signal, w[n] is noise and n represents time. The received signal y[n] is vector, of length L; and n is the sample index.

For simplicity, let 0 and 1 denote the two hypotheses, let the random variable H denote the state of the signal, and let the random variable  $\hat{H}$  denote the sensing decision. Thus, the probability of missed detection and the probability of false alarm are defined as,

$$P_M \triangleq P_r \{ \hat{H} = 0 | H = 1 \} \tag{21}$$

$$P_F \triangleq P_r \{ \hat{H} = 1 | H = 0 \}. \tag{22}$$

Small P<sub>F</sub> is necessary in order to provide possible high throughput in dynamic spectrum access networks, since a false alarm wastes a spectrum opportunity. On the other hand, small P<sub>M</sub> is necessary in order to limit the interference to PUs. A detection algorithm can seek tradeoffs between P<sub>M</sub> and P<sub>F</sub> by varying the detection threshold. [44]

For finding out these probabilities, let us make a simple consideration. Suppose, we have these events being observed during a spectrum sensing/ detection process:

$$A = \{\text{a Signal is ON}\} \quad (23)$$

$$B = \{\text{a Signal is Detected}\} \quad (24)$$

Therefore, the complementary events will be,

$$A^c = \{\text{a Signal is OFF}\} \quad (25)$$

$$B^c = \{\text{a Signal is NOT Detected}\} \quad (26)$$

Therefore, we get:

$$P_D = P(A \cap B) = P(A)P(B|A), \quad (27)$$

$$P_F = P(A^c \cap B) = P(A^c)P(B|A^c), \quad (28)$$

$$\text{And, } P_M = P(B^c \cap A) = P(A)P(B^c|A). \quad (29)$$

Based on these assumptions, the Probability of False alarm,  $P_F$ , and Probability of Missed detection,  $P_M$ , were calculated for both the algorithms.

For detailed understanding, we will have a look at the basic detector theory. We classify the detectors as Classical Detector and Compressed Detector, as we have to find the probabilities giving the measure of performance of a Compressive Sensing algorithm.

#### Classical Detector

We first discuss a classical detector and then described how a compressed detector can be derived using the same approach. Let us say that there are two hypotheses concerning the signal; that it is present in the measurements or it is not. The classical Neyman-Pearson (NP) detector involves a *likelihood ratio test* where the sufficient statistics  $t \equiv h_y$ ,  $x_i$  is compared against a threshold  $\gamma$ . Here  $y$  are the measurements,  $x$  is the signal of interest and  $\gamma$  is set to achieve certain probability of false alarm rate  $P_F \leq \alpha$  for some  $0 \leq \alpha \leq 1$ . It is easy to show that:

Here,

- $P_D = P(B|A)$  = Probability of Detection = Probability of Signal being Detected when Signal is ON.

- $P_M = P(B^c|A) =$  Probability of Missed Detection = Probability of Signal NOT being Detected when Signal is ON.

- $P_F = P(A^c)$  = Probability of False Alarm = Probability of Signal being Detected even when the Signal is OFF.

- $P(B^c|A^c) =$  Probability of Signal NOT being Detected when the Signal is OFF.

$$PD(\alpha) = Q(Q^{-1}(\alpha) - \sqrt{(SNR)}), \quad (34)$$

where  $Q(\cdot)$  is the flipped version of standard Gaussian cumulative distribution function.

**Compressed Detector** This theory can easily be extended to the case when the measurements are made using a compressed sampler. Therefore, we consider the following hypothesis:

$$H_0 : y = \Phi n \quad (30)$$

$$H_1 : y = \Phi(x + n) \quad (31)$$

where  $n \sim N(0, \sigma^2)$  is white Gaussian noise. It is straightforward to show that in this case the sufficient statistics is  $\xi \equiv (y, \Phi x)$ . It can be seen that for some  $\epsilon > 0$ , the probability of false rate is approximately given by the following equation:

$$PD(\alpha) \approx Q(Q^{-1}(\alpha) - \sqrt{(M/N)}\sqrt{SNR}) \quad (32)$$

A false alarm occurs whenever the noise voltage exceeds a defined threshold voltage. It can be given as,

$$P_{fa} = \int_T^\infty f_N(n) dn = \int_T^\infty \frac{n}{\sigma^2} e^{-n^2/2\sigma^2} dn = e^{-T^2/2\sigma^2} \quad (33)$$

As  $T$  is the threshold and we have taken,  $\text{threshold} = \sigma$ ,  $PFA$  (or,  $PF$ )  $= e^{(-1/2)} = 0.606531$ , heuristically.

Comparing equations,  $PD(\alpha) = Q(Q^{-1}(\alpha) - \sqrt{(SNR)})$ , and,  $PD(\alpha) \approx Q(Q^{-1}(\alpha) - \sqrt{(M/N)}\sqrt{SNR})$ , we can see that the performance of the detector will be deteriorated with  $M$  decreasing (as expected), and the rate of performance degradation depends on SNR.

During the simulations, that we carried out for Classical OMP and the proposed SI-OMP, the input and output spectrums were matched. With help of the level of matching between the two, different probabilities related to the detecting performance of the algorithms were calculated. The more the matching, the better the detection we found; therefore giving a higher probability of detection.

$P_{FA}$ , can be set to determine the required threshold of SNR for that specific  $P_{FA}$ . Or else, It was calculated for different SNR specifications using the following commands from MATLAB.

```
noisepow = 1.38e-23*293*db2pow(1)*33e6;
```

```
Ntrial=1000;
```

```
snrthreshold=5;
```

```
noise = sqrt(noisepow/2)*(randn(1000,1)+1j*randn(1000,1));
```

```
threshold=sqrt(noisepow*db2pow(snrthreshold));
```

We can also find the  $\text{snrthreshold}$  by setting the  $P_F$  values heuristically and take the help from the following MATLAB command,

```
snrthresh=npwgnthresh(pfa).
```

It calculates the SNR threshold in decibels for detecting a deterministic signal in white Gaussian noise. The detection uses the Neyman-Pearson decision rule to determine the specified Probability of False Alarm, PFA. This function of Matlab uses a square-law detector.

Based upon above concepts, we have considered various cases for different values of PFA for obtaining the corresponding Probabilities of Detection. From this it is easy to obtain the specific corresponding Probability of Missed Detection,  $P_M$ .

Here we have plotted the results for six various cases mentioned above for our proposed SIOMP in Figure 4. The curves shown here are plotted for sparsity levels versus the probability of detection PD. As we already know theoretically, that the probability of detection increases in direct proportion with the SNR. Therefore, we have not shown the resulting Probability of Detection for our algorithm in that context. Instead, we are showing how the parameter behaves with various values of sparsity levels in frequency domain.

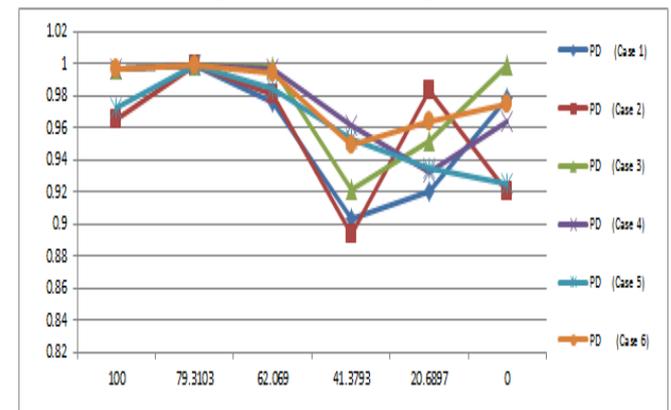


Figure 4. Probability of Detection (PD) for the SI-OMP for various experimental considerations

The results above show that the algorithm performance deteriorates by some amount as the sparsity level increases, i.e., the number of active component present increases.

For case 6, the performance is most consistent one, and, it is found the same way for the case 4, too. At highest sparsity in Gini index with only one active spectral component present,

we have the probability of detection obtained with the proposed algorithm, at its highest value for almost for all the different cases under consideration. With decreasing Gini index levels, upto 62.069 %, it has very small changes in the PD performance. Then after around 41.3793 % of Gini index, we find the PD values vary widely and decrease below 0.9. Specifically this is observed for case 1 and case 2. Then again, it improves and when we reach the highest sparsity levels, the performance is quite improved.

Other performance metrics proposed in the literature include, but are not limited to, the missed detection probability of wideband. It is defined as the probability that not all ON channels are correctly detected, and, the (empirical) probability of detecting a given number of ON channels and so forth. [44]

The output SNR decreases for increasing sparsity for all the cases almost uniformly, except for case 3, where it rises high at the highest sparsity of input signal.

This indicates that the probability of detection will be improved with this algorithm if we use it for Spectrum Sensing.

#### The Implications

The simulations took place for all the algorithms around a thousand times for each.

It is observed, here, that with increasing number of measurements, i.e., the sparsity level,  $M$ , in OMP, the reconstruction accuracy,  $r$ , tends to decrease and simulation time seems to increase. KLT on the other hand takes not much simulation time, but the reconstruction accuracy with it seems to vary drastically. This means the OMP performs better at spectrum sensing and can work better at estimating the spectral holes.

For the SI-OMP, however, the reconstruction accuracy,  $r$ , does not vary significantly and it is already well obtained in the ranges of 0.92 to 0.995 for the deterministic approach. However, with the dynamic sparsity considerations, the reconstruction accuracy value varies between 0.99 and 0.8. The normal range of variation for that remains within 0.85 to 0.95 for most of the times. The simulation times do not vary significantly for both the deterministic and subjective approaches. This means the algorithm can handle the dynamic conditions as well as static conditions.

The results above show that the performance of the OMP is improving with the increasing density of the spectrum, and, even, gives out an appreciable performance for dynamic conditions. It is then evident that the algorithms we discussed above are the enabling tools for efficient spectrum sensing for the CR. Improvisation of these may help development of a sophisticated spectrum sensing CR that will lead towards the better spectrum resource sharing and utilization.

#### IV. Conclusion and Future Work

The results obtained above prove that although quite competent, the KLT is much data dependent and the computational time of the algorithm is high for eigenvector decompositions.

The OMP is also an efficient and impressive algorithm, which is a fast greedy algorithm that iteratively builds up a signal representation by selecting the atom that maximally improves the representation at each iteration. The OMP is easily implemented and it converges quickly. That made it an attractive choice to work on. As we could improvise the signal

recovery performance of the OMP, we can be able to use it for varying sparsity environment.

We found through the above results that the new scheme based on OMP is successfully applicable for sparsity robust environments for compressive spectrum sensing. As it is an extended version of Matching Pursuit, and with appropriate modifications, SI-OMP is well capable of sensing the time-frequency atoms. We have taken a deterministic approach as well as subjective, we have presented the results for both, a static spectrum conditions in terms of time for various levels of sparsity in frequency domain, and frequency-domain sparsity varying with time; and we have found that the proposed Sparsity-Independent OMP has shown improvement in its performance very largely.

In the algorithm, the basis for the presentation matrix also contributes largely in its improved performance. We also wish to state that the presentation matrix, which is formed using discrete wavelet transforms, is capable to handle the time-frequency atom localization problem well and along with the basic ability of OMP itself, it is able to track the time domain variations in the spectrum also the way it has been doing with the static spectrum.

In future, we hope that we can still improve the performance of the algorithm for obtaining uniform performance for both static and dynamic spectral sparsity conditions, and hence, better reconstructing and estimating the spectral opportunities to make the secondary usage of the spectrum more efficient.

#### References

- [1] Axell, E., Leus, G., Larsson, E. G., and Vincent Poor, H., "State-of-the-art and recent advances Spectrum Sensing for Cognitive Radio: State-of-the-art and recent advances", 2012, IEEE signal processing magazine (Print), (29), 3, 101-116.
- [2] Patel, C A, and, Vithalani, C H, "Compressive Spectrum Sensing: An Overview", IJIREC, Volume 1, Issue 6, September 2014, PP 1-10 ISSN 2349-4042 (Print) & ISSN 2349-4050 (Online)
- [3] Tropp, J A, and, Gilbert, A C, "Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit", IEEE Transactions, 2005.
- [4] Lakshminarayana. M., and, Sarvagya, M., "Scaling the Effectiveness of Existing Compressive Sensing in Multimedia Contents", International Journal of Computer Applications (0975 – 8887), Volume 115 – No. 9, April 2015.
- [5] Havary-Nassab, V., Hassan, S., and, Valaee, S., "Compressive detection for wide-band spectrum sensing"
- [6] Zonoobi, D., Kassim, A. A., and, Venkatesh, Y. V., "Gini Index as Sparsity Measure for Signal Reconstruction from Compressive Samples", IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING, VOL. 5, NO. 5, SEP 2011.
- [7] E. Candès and T. Tao, "Decoding by linear programming," IEEE Trans. Inf. Theory, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.
- [8] Candès, E.J., and, Wakin, M. B., "An introduction to compressive sampling", IEEE Signal Processing Magazine, p-p 21-30, March 2008.
- [9] Pawar, K., Egan, G., Zhang, J., "Multichannel Compressive Sensing MRI Using Noiselet Encoding", arXiv:1407.5536v2 [physics.med-ph] 22 Jul 2014
- [10] R. Coifman, F. Geshwind, and Y. Meyer, "Noiselets", Applied and Computational Harmonic Analysis 10, 27–44 (2001)

- [11] Xu, Y., Yin, W., Osher, S., "Learning circulant sensing kernels"
- [12] J. Tropp, M. Wakin, M. Duarte, D. Baron, and R. Baraniuk, "Random filters for compressive sampling and reconstruction," in *Acoustics, Speech and Signal Processing*, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on, vol. 3, may 2006, p. III.
- [13] Hegarty, F., Cathain, P. O., Zhao, Y., "Sparsification of Matrices and Compressed Sensing", arXiv: 1506.03523v3, [cs.IT], 05 Nov., 2015.
- [14] Gibson, J. P., "Complexity & Algorithm Analysis", *Mathematical Foundations*, <http://www-public.it-sudparis.eu/~gibson/Teaching/MAT7003>
- [15] Blockeel, H., and, Struyf, J., "Efficient Algorithms for Decision Tree Cross-validation", *Journal of Machine Learning Research* 3 (2002) 621-650
- [16] Redivo-Zaglia, M., and, Rodriguez, G., "smt: a Matlab structured matrices toolbox", arXiv:0903.5094v1 [math.NA] 30 Mar 2009.

#### Authors' Biographical Information



Charushila Axay Patel did complete her graduation in Ele. & Comm. engineering in 1998, from Bhavnagar University. She worked as a Teaching Faculty in various institutes in Gujarat State, India; and, currently she is working as a Sr. Lecturer at BVPIT (DS), GTU, at Surat, Gujarat. She did obtain her ME degree in Microwave Engg from BIT, Mesra in 2010. She is currently pursuing her PhD at GTU, Gujarat.



Dr. C. H. Vithalani is a prominent researcher in field of Wavelet Transforms. He obtained his PhD from Gujarat University in 2006. He has a wide range of working experience in the field of Electronics & Communication engineering in industries as well as academia. He has guided over 200 students at Graduate level and over 20 students at Post Graduate level. He has presented his research papers at various National and International level conferences. Currently he is working as Department Head of Ele. & Comm. Engg Dept, GEC, Rajkot, Gujarat, India.



Dr. Preetida Jani has a broad range of experience. She obtained her degree of B.E. (Electronics & Communication) from Gujarat University, Ahmadabad, India, M.Eng.Sc (Digital Communication) from University of New South Wales, Sydney, Australia, and, PhD (Electrical Engineering) from University of Technology, Sydney, Australia.

She has worked:

1. As a Senior Research Engineer in Industry Nokia Research Center (Finland) for 10 years
2. As Professor in Sardar Vallabhbhai National Institute of Technology, Surat( For 7 months)
3. As Associate Professor in Dhirubhai Ambani Institute of Information and Communication Technology Gandhinagar, Gujarat, India.(For 2 Years)

Currently she is working as Professor in Department of Electronics & Telecommunication Engineering at SVIT, Mumbai University. Her area of Research Interest ranges from Wireless Communication to Algorithms, and, Computer Networking to Cognitive Radio.