International Journal of Innovative Research in Electronics and Communications (IJIREC)
Volume 1, Issue 5, August 2014, PP 22-33
ISSN 2349-4042 (Print) & ISSN 2349-4050 (Online)
www.arcjournals.org

A Lightweight Randomized Low Sampling Compression Technique Verified by GI with Merits over CR and IT Reconstruction Schemes

[1]Bhakthavathsalam R, [1]Gowranga K H, [2]Abhishek Sharma R, Chaitra Suresh [2]Chandan S

[1]Supercomputing Education and Research Centre, Indian Institute of Science, Bangalore, India,
bhaktha@serc.iisc.ernet.in
[2]Department of Computer Science and Engineering, BNM Institute of Technology, Bangalore, India,
rabhisheks@gmail.com

Abstract: *Compressive sensing (CS) is a novel sampling paradigm that samples signals in a much more efficient way than the established Nyquist Sampling Theorem. CS has recently gained a lot of attention due to its exploitation of signal sparsity. This paper gives a brief background on the origins of this idea, reviews the basic mathematical foundations of the sampling theory and compares the different reconstruction schemes. In our work, a signal is generated and sampled using the CS method. Then the original signal is reconstructed using three different reconstruction schemes namely Greedy Iterative (GI), Convex Relaxation (CR), and Iterative Thresholding (IT). The accuracy and the time taken for these three schemes are calculated and compared. It was found that Greedy Iterative took the least time for reconstruction with a lower error rate amongst the three schemes.*

Keywords: *Compressive Sensing, Nyquist Sampling theorem, sparsity, incoherence, orthogonal matching pursuit, basis pursuit, approximate message passing*

1. Introduction

As the modern technology progresses, there is a problem of ever-increasing amounts of data. By now, everyone is aware that most of the data received can be thrown away without almost no perceptual loss. So, can't we just directly measure the part that won't end up being thrown away? The traditional way of sampling i.e. the Nyquist-Shannon sampling theorem states that to restore a signal exactly and uniquely, you need to have sampled with at least twice its frequency. This is true for perfectly band-limited signals. But most real world signals are not perfectly band-limited. When represented in terms of appropriate basis functions, such as trigonometric functions or wavelets, many signals have relatively few non- zero coefficients. In short, they are *sparse* [1]. Compressive Sensing takes advantage of this sparsity in signals to directly acquire just the important information about them and not acquire that part of the data that would eventually just be 'thrown away'. The crucial observation is that one can design efficient sensing or sampling protocols that capture the useful information content embedded in a sparse signal and condense it into a small amount of data. These protocols are non-adaptive and simply require correlating the signal with a small number of fixed waveforms that are incoherent with the sparsifying basis. What is most remarkable about these sampling protocols is that they allow a sensor to very efficiently capture the information in a sparse signal without trying to comprehend that signal.

2. Compressive Sensing

In the traditional signal processing techniques, we uniformly sample data at Nyquist rate, prior to transmission, to generate 'n' samples. These samples are then compressed to 'm' samples; discarding 'n-m' samples which leads to wastage of both time and effort. It is also expensive computationally as it needs more storage space that may later be unused. Our goal is to ensure that the number of samples (i.e. 'm') captured is far less when compared to the traditional method (i.e. 'n') and to show that signal reconstruction is as effective thereby reducing cost, effort and time needed to implement it. Compressive Sensing theory asserts that one can recover certain signals and images from far fewer samples or measurements than traditional methods use. To make this possible, CS relies on two principles: sparsity, which pertains to the signals of interest, and incoherence, which pertains to the sensing modality [2].

2.1 Sparsity

Natural signals such as sound, image or seismic data can be stored in compressed form, in terms of their projection on suitable basis Ψ. When basis is chosen properly, a large number of projection coefficients are zero or small enough to be ignored. If a signal has only 's' non-zero coefficients, it is said to be s-Sparse. If a large number of projection coefficients are small enough to be ignored, then signal is said to be compressible. Sparsity expresses the idea that the "information rate" of a continuous time signal may be much smaller than suggested by its bandwidth, or that a discrete-time signal depends on a number of degrees of freedom, which is comparably much smaller than its (finite) length.

2.2 Incoherence

Incoherence extends the duality between time and frequency and expresses the idea that objects having a sparse representation in Ψ must be spread out in the domain in which they are acquired, just as a Dirac or a spike in the time domain is spread out in the frequency domain. Put differently, incoherence says that unlike the signal of interest, the sampling/sensing waveforms have an extremely dense representation in Ψ. Coherence measures the maximum correlation between any two elements of two different matrices. These two matrices might represent two different basis representation domains.

2.3 Restricted Isometry Property (RIP)

Restricted Isometry Property has been the most widely used tool for analyzing the performance of CS recovery algorithms; a key notion that has proved to be very useful to study the general robustness of CS.

Definition:

For each integer s = 1,2,3…. define the isometric constant δ_s of the matrix ϕ as the smallest number such that,

$$(1 - \delta_s)||x||^2 \leq ||\phi_x||^2 \leq (1 + \delta_s)||x||^2$$

holds for all s-sparse vectors. A vector is said to be s-sparse if it has at most s non-zero entries. Then, the matrix ϕ is said to satisfy the s-restricted isometric property with restricted isometric constant δ_s.We will loosely say that a matrix ϕobeys the RIP of order S if δ_s is not too close to one. When this property holds, ϕ approximately preserves the Euclidean length of S-sparse signals, which in turn implies that S-sparse vectors cannot be in the null space of ϕ. The main point here is that RIP is sufficient to guarantee sparse reconstruction by L1-minimization. It is known that L1-minimization reconstructs every sparse signal precisely when the sensing matrix satisfies the null space property (NSP), and so one way to prove that RIP is sufficient is to show that RIP implies NSP. The following paper bypasses the NSP analysis by giving a direct result for RIP.

3. LITERATURE SURVEY

A literature survey has been done to know the various data compression techniques in vogue. Compression can be either lossy or lossless. Lossless compression reduces bits by identifying and eliminating statistical redundancy. No information is lost in lossless compression. Lossy compression reduces bits by identifying unnecessary information and removing it.

The basic principle that lossless compression algorithms work on is that any non-random file will contain duplicated information that can be condensed using statistical modeling techniques that determine the probability of a character or phrase appearing. These statistical models can then be used to generate codes for specific characters or phrases based on their probability of occurring, and assigning the shortest codes to the most common data.

Given this vast amount of different techniques, there are different ways how to classify compression techniques:

- With respect to the type of data to be compressed.
- In relation with the target application area.
- Based on the fundamental building blocks of the algorithms used.

3.1. History

Starting from 1838 where the Morse code was developed for telegraphy to Huffman's technique of dynamic updating of code words based on accurate data in 1970s, data compression has been an area of interest to most researchers. However in 1977, Abraham Lempel and Jacob Ziv published their groundbreaking LZ77 algorithm, the first algorithm to use a dictionary to compress data. More specifically, LZ77 used a dynamic dictionary oftentimes called a sliding window. Then during mid-1980s, the pioneering work done by Terry Welch led to innovation of Lempel–Ziv–Welch (LZW) algorithm, which later became the most popular algorithm for many general purpose compression systems.

3.2. Nyquist Sampling Theorem

In 1949, Shannon [1][2] presented his famous proof that any band-limited time-varying signal with 'n' Hertz highest frequency component can be perfectly reconstructed by sampling the signal at regular intervals of at-least 1/2n seconds. In traditional signal processing techniques, we uniformly sample data at Nyquist rate, prior to transmission, to generate 'n' samples. These samples are then compressed to 'm' samples; discarding n-m samples. The sampling theorem specifies that to avoid losing information when capturing the signal, the sampling rate must be at least twice the signal bandwidth.

3.3. Compressive Sensing

Compressive Sensing (CS) is an innovative process of acquiring and reconstructing a signal that is sparse or compressible. Around 2004 Emmanuel Candès, Terence Tao and David Donoho discovered important results on the minimum number of data needed to reconstruct an image even though the Nyquist–Shannon criterion would deem the number of data insufficient. Compressive sensing theory asserts that we can recover certain signals from fewer samples than required in Nyquist paradigm. This recovery is exact if signal being sensed has a low information rate (means it is sparse in original or some transform domain). Number of samples needed for exact recovery depends on particular reconstruction algorithm being used. If signal is not sparse, then recovered signal is best reconstruction obtainable from s largest coefficients of signal. CS operates very differently, and performs as "if it were possible to directly acquire just the important information about the object of interest. Capitalizing on this discovery, much of signal processing has moved from the analog to the digital domain and ridden the wave of Moore's law. Digitization has enabled the creation of sensing and processing systems that are more robust, flexible, cheaper and, consequently, more widely used than their analog counterparts. As a result of this success, the amount of data generated by sensing systems has grown from a trickle to a torrent.

4. Proposed System

Any analog signal consists of components at various frequencies. The simplest case is the sine wave, in which all the signal energy is concentrated at one frequency. In practice, analog signals usually have complex waveforms, with components at many frequencies. The highest frequency component in an analog signal determines the bandwidth of that signal. Compressed Sensing or Compressive Sensing is about acquiring and recovering a sparse signal in the most efficient way possible (sub-sampling) with the help of an incoherent projecting basis. Unlike traditional sampling methods, Compressed Sensing provides a new framework for acquiring sparse signals in a multiplexed manner.

The system consists of two main modules; one for sampling the input signal and another for reconstruction. The steps involved in the sampling module are as follows:

1. The user chooses any one of the keys between 0 and 9, for which a signal is generated.

2. The sampling module then takes a certain number of samples of this signal. These samples are written into a file, which is then sent to the reconstruction module.

The step involved in the reconstruction module is as follows:

The reconstruction module attempts the recovery of the signal by using the samples sent to it using one of the 3 algorithms: Orthogonal Matching Pursuit, Approximate Message Passing and Basis Pursuit.

The overall system architecture shown in Fig 4.1 describes how the system works.

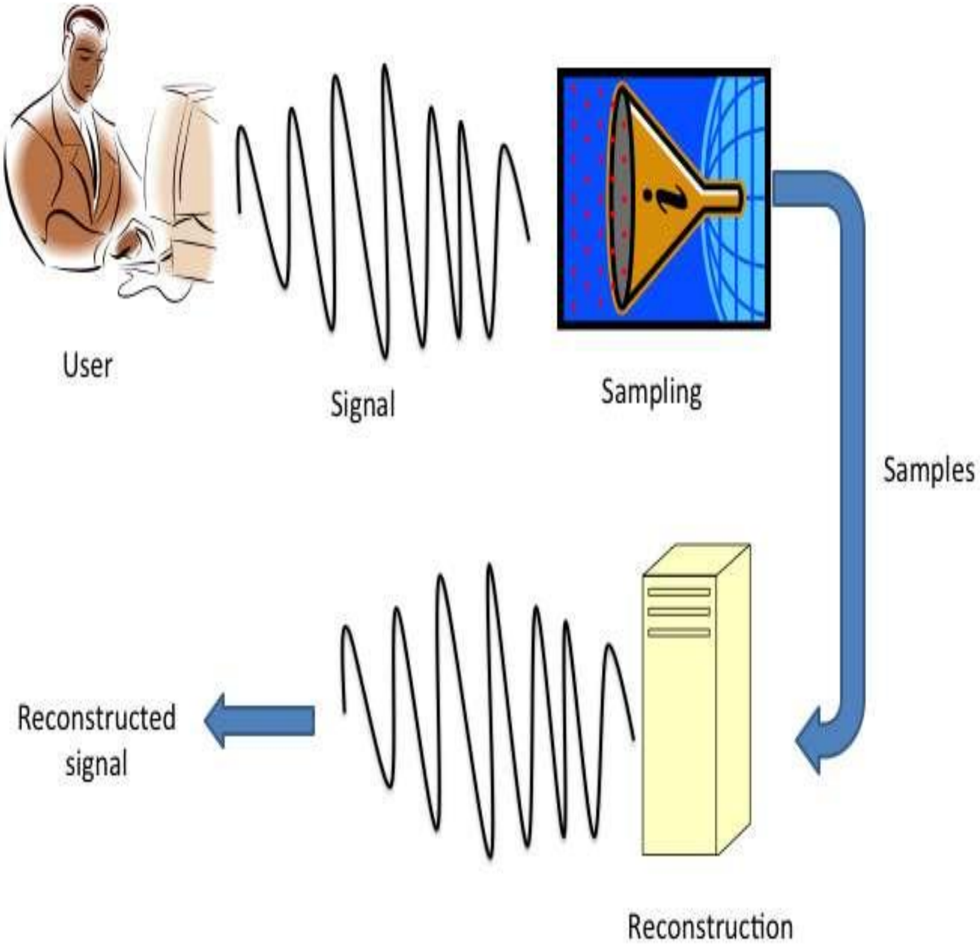

Fig4.1. *System Architecture*

5. METHODOLOGY

The process consists of three steps as stated earlier. The first step is the generation of an analog signal. The user is asked to enter a key between 0 and 9. Each input is associated with two frequencies. The signal generated comprises of two sinusoids. The signal 'f' is generated in our simulation by using

$$f = (\sin(2*pi*f1*t) + \sin(2*pi*f2*t))/2$$

Where f1 and f2 are the two component frequencies obtained from Table 5.1.

Table5.1.

Input	0	1	2	3	4	5	6	7	8	9
f1(Hz)	941	697	697	697	770	770	770	852	852	852
f2(Hz)	1336	1209	1336	1477	1209	1336	1477	1209	1336	1477

The next step is to sample the given analog signal. Any raw signal can be represented as

$$f = \Psi c$$

The signal is then sampled at known time intervals determined by the sampling frequency (F_s)

which is fixed. These samples are obtained by applying a matrix known as the sampling matrix (Φ) to the signal of interest(c). This signal of interest contains all the major coefficients of the given signal, i.e., the non-sparse components. This is an n×1 vector. Vector 'c' is obtained by applying the basis matrix (Ψ) to the vector 'f'. Here, the basis function has been chosen to be the Discrete Cosine Transform (DCT). DCT was chosen as the basis function because it requires lesser terms to represent a typical signal in comparison with other basis functions such as Fast Fourier Transform, Fourier Transform etc. Sampling process can denoted as

$$b = \Phi c$$

Where 'b' is the set of samples obtained.

The final step is the reconstruction of the original signal from the samples obtained [3]. To reconstruct the signal, we must try to recover the coefficients by solving

$$Ax = b$$

where A = ΦΨ. Once we have the coefficients, we can recover the signal itself by computing

$$f = \Psi x$$

Since this is a compression technique, 'A' is rectangular, with many more columns than rows. Computing the coefficients 'x' involves solving an underdetermined system of simultaneous linear equations, Ax = b.

5.1. Orthogonal Matching Pursuit Reconstruction Scheme

Greedy algorithms are well known in computer science literature due to their simplicity while obtaining good, and in some cases, optimal, results. A greedy algorithm is an algorithm that follows the problem solving heuristic of making the locally optimal choice at each stage with the hope of finding a global optimum. In many problems, a greedy strategy does not in general produce an optimal solution, but nonetheless a greedy heuristic may yield locally optimal solutions that approximate a global optimal solution in a reasonable time. Another effective reconstruction scheme is a variant of Greedy Iterative method called as the Orthogonal Matching Pursuit (OMP). It constructs an approximation by going through iteration process [4]. In each iteration, the locally optimum solution is determined by finding the column vector of ***A*** which is most correlated with the residual vector ***r***. Initially the residual vector is equal to the vector that is to be approximated i.e. ***r = b*** and it is adjusted at each iteration to take into account the previously chosen vector. OMP is a stepwise forward selection algorithm and is easy to implement. A key component of OMP is the stopping rule, which depends on the noise structure. In the noiseless case the stopping rule is that the residual becomes zero i.e. $\boldsymbol{r_i} = 0$.

5.1.1. Algorithm

1. Start by setting the residual $r_0 = b$, the time t=0 and index set $U_{0=}\emptyset$
2. Let $u_t = i, where\ a_i$ gives the solution of max $< r_t, a_k >$ where a_k are the row vectors of A.
3. Update the set $U_t\ with\ u_t : U_t = U_{t-1} \cup \{U_t\}$
4. Solve the least-squares problem:

$$\min_{c \in c U_t} ||b - \sum_{j=1}^{t} c(u_j) a_{u_j}||$$

5. Calculate the new residual using c

$$r_{t=} r_{t-1} - \sum_{j=1}^{t} c(u_j) a_{u_j}$$

6. Set $t \leftarrow t + 1$
7. Check stopping criterion (residual=0). If the criterion has not been satisfied then return to step 2.

5.2. Basis Pursuit Reconstruction Scheme

The key to the almost magical reconstruction process is to impose a nonlinear regularization involving the L1 norm. This is a variant of the Convex Relaxation reconstruction scheme [4]. To have theoretical guarantees for the convex relaxation method, one needs to show that the sparse approximation problem is equivalent to its convex relaxation. Known theoretical guarantees work only for random measurements.

In principle, computing this reconstruction should involve counting non-zeros with L0. This is a combinatorial problem whose computational complexity makes it impractical. However, L0 can be replaced by L1 as the two problems have the same solution. The L1 computation is practical because it can be posed as a linear programming problem and solved with the traditional simplex algorithm or modern interior point methods. We use the Primal-Dual Interior Point Algorithm to solve the Basis Pursuit Problem.

5.2.1. Primal Dual Interior Point Algorithm

Primal problem:

$$(P)\ \textit{minimize}\ c^T x$$

$$\textit{subject to}: Ax = b, x \geq 0 \quad (\textit{m equalities}, \textit{n variables})$$

Dual problem:

$$(D)\ \textit{maximize}\ b^T w$$

$$\textit{subject to}: A^T w + s = c, s \geq 0$$

The summary of the derivation is given in the next section.

5.2.2. Derivation summary:

Step1: Remove the inequalities from (P) using a barrier term

$$(PB)\ \textit{minimize}\ c^T x - \Omega \sum_j ln\,(x_j)$$

$$\textit{subject to}: Ax = b, x \geq 0$$

where Ω is a positive barrier parameter.

Step2: State the Lagrange function

The Lagrange function is:

$$L(x, y) := c^T x - \Omega \sum_j ln\,(x_j) - y^T (Ax - b)$$

Where y contains the Lagrange multipliers.

Step3: State the Lagrange optimality conditions

The optimality conditions are:

$$\nabla_x L(x, y) = c - \Omega X^{-1} e - A^T y = 0,$$

$$\nabla_y L(x, y) = Ax - b = 0$$

$$\text{Where } X := diag(x) := \begin{bmatrix} x1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & xn \end{bmatrix}, e = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

Let $s = \Omega X^{-1} e$ hence $Xs = \Omega e$

Equivalent optimality conditions:

$$\text{(O)} \quad Ax = b, x > 0$$

$$A^T y + s = c, s \geq 0$$

$$Xs = \Omega e$$

Step4: Solving the optimality conditions

The nonlinear optimality conditions are solved using Newton's method:

$$\nabla f\left(x^k d_x\right) = f(x^k),$$

$$x^{k+1} = x^k + \alpha d_x$$

Where $\alpha \in (0,1]$ is the step size, solves f(x)=0.

Define:

$$F_\gamma(x,y,s) := \begin{bmatrix} Ax-b \\ A^T y + s - c \\ Xs - \gamma\mu e \end{bmatrix}, \Omega := \gamma\mu = \gamma x^T s/n.$$

Where $\gamma \geq 0$

Given $(x^0, s^0) > 0,$

Then one step of Newton's method applied to

$$F_\gamma(x,y,s) = 0, x, s \geq 0$$

is given by :

$$\nabla F_\gamma(x^0, y^0, s^0)\begin{bmatrix} d_x \\ d_y \\ d_s \end{bmatrix} = -F_\gamma(x^0, y^0, s^0)$$

$$and \begin{bmatrix} x^1 \\ y^1 \\ s^1 \end{bmatrix} := \begin{bmatrix} x^0 \\ y^0 \\ s^0 \end{bmatrix} + \alpha \begin{bmatrix} d_x \\ d_y \\ d_s \end{bmatrix},$$

Where α=0.01

5.2.3. Algorithm

1. Choose (x^0, y^0, s^0) such that $x^0, s^0 > 0.$
2. Choose γ, θ ∈ (0, 1), ε >0
3. k := 0
4. while max (‖$Ax^k - b$‖ ,‖$A^T y^k + s^k - c$‖ ,$(x^k)^T s^k$) ≥ ε
5. μ^k:= $((x^k)^T s^k)$/n
6. Solve:

$$Ad_x = -(Ax^k - b),$$

$$A^T d_y + d_s = -(A^T y^k + s^k - c),$$

$$s^k d_x + X^k d_s = -X^k s^k + \gamma\mu^k e.$$

7. Compute:

$$\alpha^k := \theta \max\{\bar{\alpha}: x^k + \bar{\alpha} d_x \geq 0, s^k + \bar{\alpha} d_s \geq 0, \theta\bar{\alpha} \leq 1\}$$

8. $\left(x^{k+1}; y^{k+1}; s^{k+1}\right) := \left(x^k; y^k; s^k\right) + \alpha^k(d_x; d_y; d_s)$
9. k := k+ 1
10. end while

5.3. Approximate Message Passing Reconstruction Scheme

Another class of algorithms with low computational complexity is the Iterative Thresholding scheme [5]. Approximate Message Passing (AMP) [6] [7] is a variant of this scheme. AMP reconstructs the signal as effectively as L1 while running much faster. The idea behind these

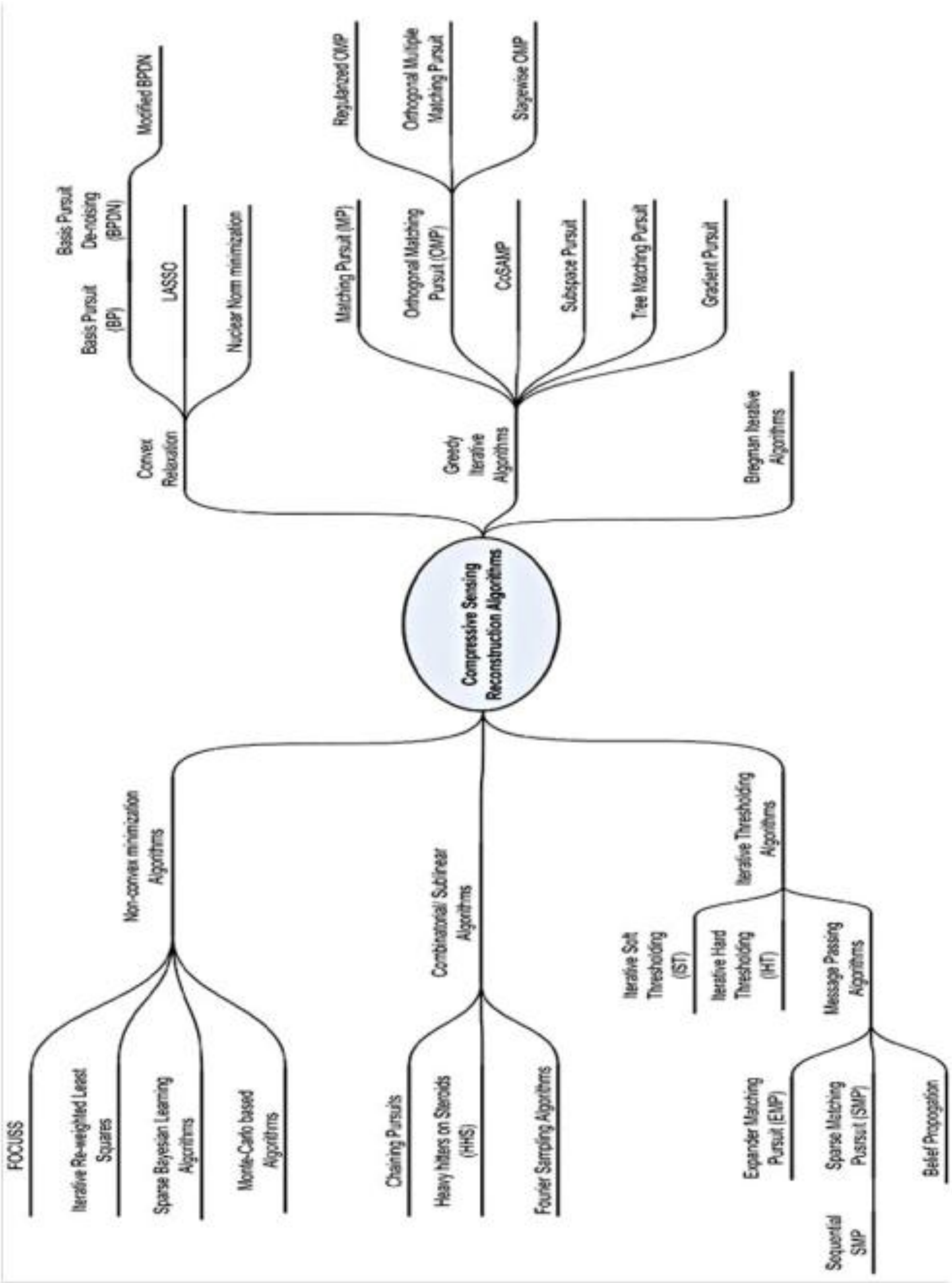

Fig5.2. *Compressed sensing reconstruction algorithms and their classification*

algorithms is that when a signal is represented in terms of a suitable basis, smaller coefficients are set to zeroes while the larger coefficients above a given threshold are possibly shrunk. In every iteration, a residual is calculated along with a new threshold. With every step, these values change and the algorithm breaks when the stopping condition is reached. As in the case of OMP, the stopping rule is dependent on the noise structure. For the noiseless case, the stopping rule is that the residual reaches zero.

5.3.1. Algorithm

1. $x^{t+1} = \eta(P^* z^t + x^t; \tau^t)$.

2. $b - Px^t + \frac{z^{t-1}}{\delta}\langle \eta'(P^* z^{t-1} + x^{t-1}; \tau^{t-1})\rangle$, here the last term is called the Onsager Reaction Term.

3. $\tau^t = \frac{\tau^{t-1}}{\delta}\langle \eta'(P^* z^{t-1} + x^t; \tau^{t-1})\rangle.$

Where,

x^t– Current estimate

P – Measurement matrix

P *- Transpose of A

z^t- Current residual

b- Vector of interest (m x 1)

t- Iteration counter

$\eta -$ Scalar threshold function

τ- Threshold

δ- Under sampling rate

Here,

δ= m/n

τ= 0.001, initially

$$\eta(x;\tau) = \begin{cases} x+\tau, x < -\tau \\ 0, -\tau \le x \le \tau, \\ x-\tau, x > \tau \end{cases}$$

For a vector u=(u(1),u(2)…u(n)),

$$\langle u \rangle \equiv \sum_{i=1}^{n} u(i)/n,$$

$$\eta'(x) = \frac{\partial(\eta(x)}{\partial x}$$

6. RESULTS

Snapshots of the original signal and reconstructed signals are given below. The error rates of the three reconstruction schemes used are also provided for comparison.

6.1. Original Signal

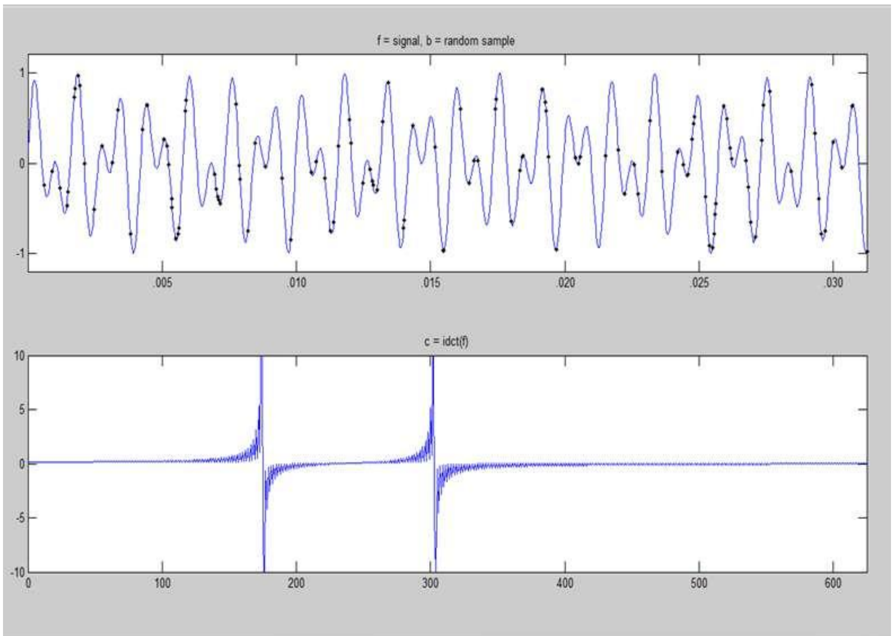

Fig5.1. *Top: Random samples of the original signal generated Bottom: The inverse discrete cosine transform of the signal*

The above figure gives the following details:

1. The graph at the top represents input analog signal with the black dots representing the samples that are taken into consideration.

2. The bottom graph represents the inverse direct cosine transform (IDCT) of the input signal which are represented as two sinusoids.

6.2. OMP Solution

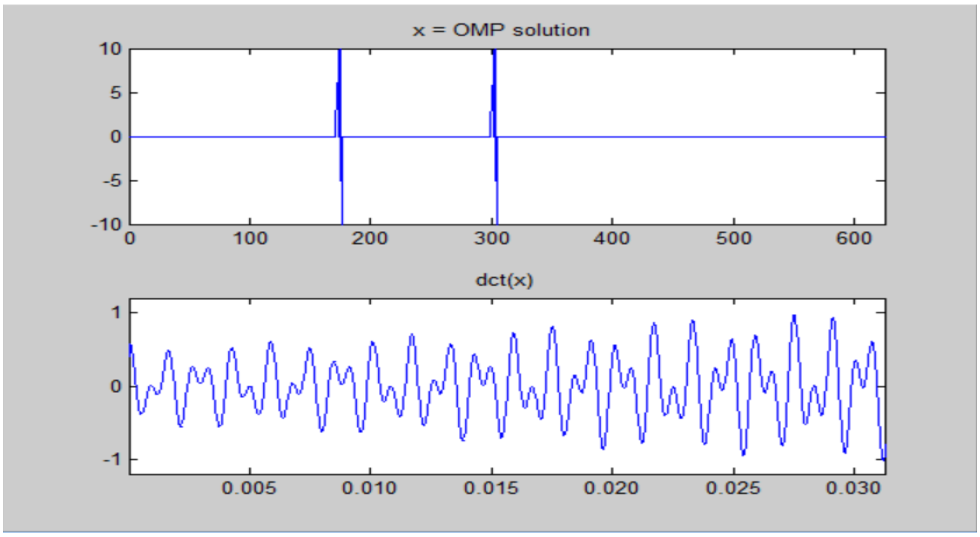

Fig5.2. *Top: OMP solution of the original signal Bottom: Reconstructed signal using OMP*

6.3. Basis Pursuit Solution

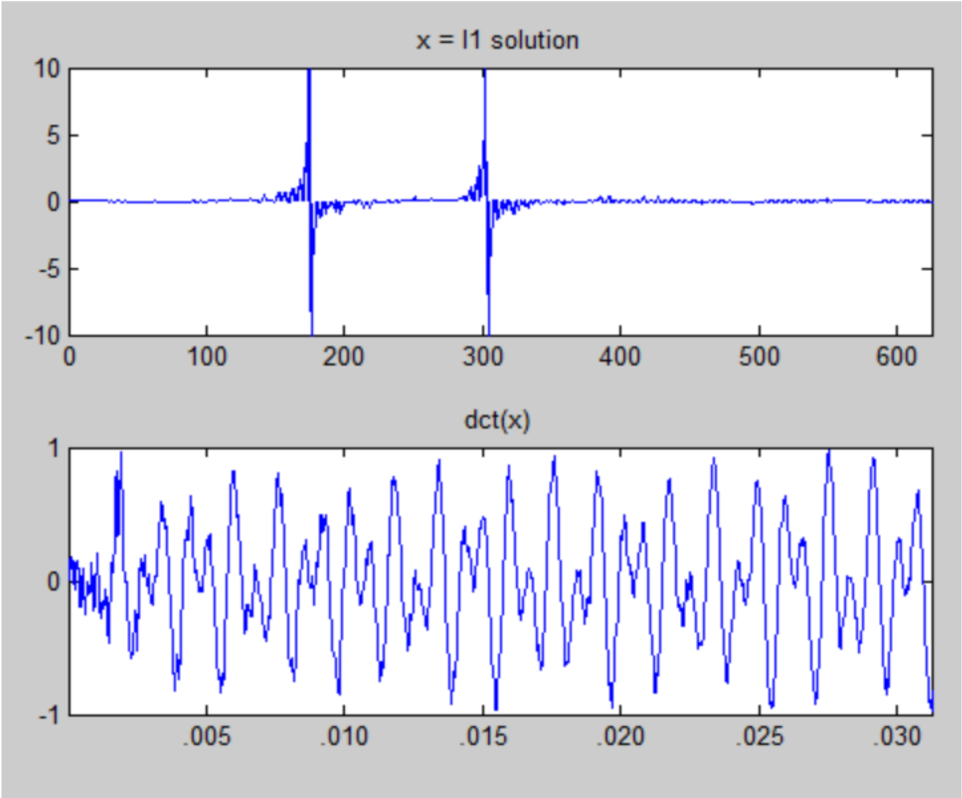

Fig. 5.3

Top: L1 solution of the original signal

Bottom: Reconstructed signal using Basis Pursuit

6.4. AMP Solution

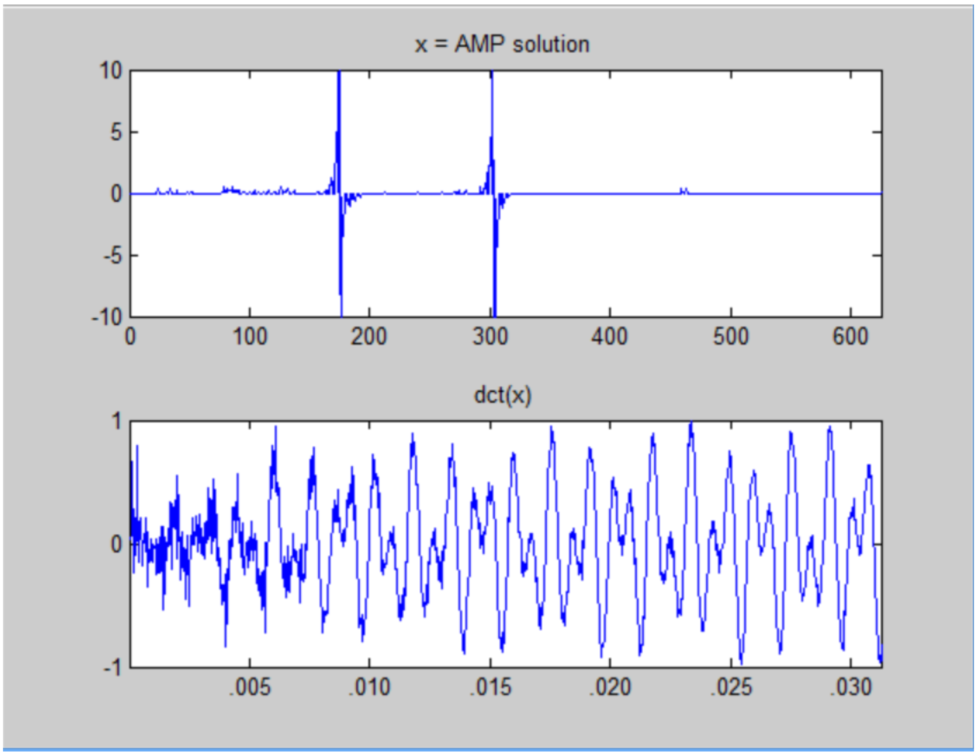

Fig. 5.4

Top: AMP solution of the original signal

Bottom: Reconstructed signal using AMP

6.5. Comparison of Error Rates

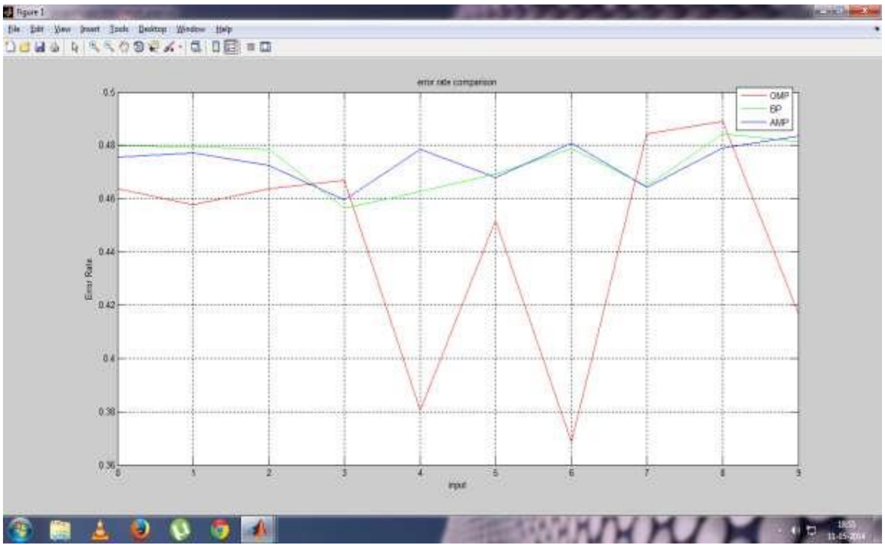

Fig. 5.5 *Comparison of Error Rates*

The error rate was calculated using the following formula:

$$\text{Error rate} = \sum_{i=0}^{n} (x_i - f_i) \times (x_i - f_i)$$

where, x is the solution obtained

f is the original signal

n is the number of samples

Average error rate for

- Orthogonal Matching Pursuit (OMP) is 0.4443
- Basis Pursuit (BP) is 0.4736
- Approximate Message Passing (AMP) is 0.4739

7. Conclusion

The traditional *sample first-then compress* method collects a huge number of samples from the input signal at first but later discards most of the samples. This is due to the fact that the information is contained in fairly few samples. The information rate is found to be far lower than the number of samples collected. It is a well established fact that all real world signals are sparse in some domain (time or frequency). *Compressive Sensing* exploits this property of real world signals. It does so by collecting few samples in the beginning itself. In this work, the technique of compressive sensing is used where an analog signal is sampled at the client side and reconstructed at the server side. The reconstruction was done using the following three techniques: Orthogonal Matching Pursuit, Basis Pursuit and Approximate Message Passing. A comparison was done between the above mentioned algorithms with respect to the following parameters: a) Time taken for signal reconstruction and b) the Error Rate. It was found that OMP algorithm took the least time for reconstruction and had least error rate among the three reconstruction schemes. The system was tested on Windows platform and was found to be performing acceptably.

Acknowledgement

The authors sincerely thank the authorities of Supercomputer Education and Research Center, Indian Institute of Science for the encouragement and support during the entire course of this work.

References

[1] Qaisar, Saad, et al. "Compressive sensing: From theory to applications, a survey." *Communications and Networks, Journal of* 15.5 (2013): 443-456.

[2] Donoho, David L. "Compressed sensing." *Information Theory, IEEE Transactions on* 52.4 (2006): 1289-1306.

[3] Candès, Emmanuel J., and Michael B. Wakin. "An introduction to compressive sampling." *Signal Processing Magazine, IEEE* 25.2 (2008): 21-30.

[4] Breen, Philip. "Algorithms for Sparse Approximation.", *School of Mathematics, University of Edinburgh, Year* 4 (2009).

[5] Blumensath, Thomas, and Mike E. Davies. "Iterative thresholding for sparse approximations." *Journal of Fourier Analysis and Applications* 14.5-6 (2008): 629-654.

[6] Donoho, David L., Arian Maleki, and Andrea Montanari. "Message-passing algorithms for compressed sensing." *Proceedings of the National Academy of Sciences* 106.45 (2009): 18914-18919.

[7] Donoho, David L., Arian Maleki, and Andrea Montanari. "Message passing algorithms for compressed sensing: I. motivation and construction." *Information Theory Workshop (ITW), 2010 IEEE*. IEEE, 2010.