R.V. COLLEGE OF ENGINEERING, BENGALURU–560059(Autonomous Institution Affiliated to VTU, Belagavi)

Hardware Implementation of Compressive Sensing and Reconstruction Algorithm

MINOR PROJECT REPORT

Submitted by,

ADITYA MUKUNDAN 1RV14EC009
ANJANI PRAKASH 1RV14EC025
BHANUTEJ P RAVILLA 1RV14EC033
ISHAN RANJAN 1RV14EC057

Under the guidance of

Mrs. Sujata D. Badiger
Assistant Professor,
Dept. of ECE,
R.V. College of Engineering.

in partial fulfilment for the award of degree

of

Bachelor of Engineering

in

ELECTRONICS & COMMUNICATION ENGINEERING
2017-2018

R.V. COLLEGE OF ENGINEERING, BENGALURU– 560059
(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

CERTIFICATE

This is to Certify that the minor project work titled, "**HARDWARE IMPLEMENTATION OF COMPRESSIVE SENSING AND RECONSTRUCTION ALGORITHM**", carried out by **Aditya Mukundan**(1RV14EC009),**Anjani Prakash**(1RV14EC025),**Bhanutej P Ravilla**(1RV14EC033) and **Ishan Ranjan**(1RV14EC057), in partial fulfilment for the award of the degree of **Bachelor of Engineering in Electronics and Communication Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2017-2018. It is certified that all the corrections/suggestions indicated for the internal assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies academic requirements in respect of minor project work prescribed by the institution for the said degree.

17/11/17

Signature of Guide

U. B. V

Signature of Head of the Department
Dr.B.V.Uma

External Viva

Name of Examiners	**Signature with Date**
1	
2	

R.V. COLLEGE OF ENGINEERING, BENGALURU – 560059
(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

DECLARATION

We, **Aditya Mukundan(1RV14EC009)**,**Anjani Prakash(1RV14EC025)**,**Bhanutej P Ravilla(1RV14EC033)** and **Ishan Ranjan(1RV14EC057)**, students of seventh semester B.E., Department of Electronics and Communication Engineering, R.V. College of Engineering, Bengaluru-560059, hereby declare that the minor project titled **"HARDWARE IMPLEMENTATION OF COMPRESSIVE SENSING AND RECONSTRUCTION ALGORITHM"** has been carried out by us and submitted in partial fulfilment for the award of degree in **Bachelor of Engineering** in **Electronics and Communication Engineering**, during the year **2017-2018.** Further we declare that the content of the dissertation has not been submitted previously by anybody for the award of any degree or diploma to any other university.

We also declare that any Intellectual Property Rights generated out of this project carried out at R.V.C.E. will be the property of R.V. College of Engineering, Bengaluru and we will be one of the authors of the same.

Place: Bengaluru

Date:

Name 17-11-2017

Name		**Signature**
Aditya Mukundan	**(1RV14EC009)**	Adi
Anjani Prakash	**(1RV14EC025)**	Anjani Prakash
Bhanutej P Ravilla	**(1RV14EC033)**	Bhanutej
Ishan Ranjan	**(1RV14EC057)**	Ishan

ACKNOWLEDGEMENT

As a part of process in the completion of the minor project titled "**Hardware Implementation of Compressive Sensing and Reconstruction Algorithm**", we would like to extend our gratitude to our guide, **Mrs.Sujata D. Badiger**, Assistant Professor, Dept. of Electronics and Communication Engineering for her unyielding support and guidance throughout the course of the project.

Weare in debt with much gratitude to the project committee, **Dr. M Uttara Kumari** and **Mrs.Rohini Hallikar ,**committee members **Mr. K.S. Shushrutha** and **Mrs.Shilpa G D** Department of Electronics and Communication Engineering, for their patient guidance, encouragement, timely support and assistance during the entire study of the project.

We take this opportunity to thank **Dr.B.V.Uma**, Professor and HOD, Electronics & Communication Engineering, for the cooperation, advice and her timely support.

We would also like to thank **Dr.K.N.Subramanya,** Principal, R.V.College of Engineering, for providing us with the platform to setup and implement our project with a smoothly implemented chain of facilities.

We would like to thank the **faculty members** and **non-teaching** staff of the Department of Electronics and Communication Engineering, RVCE for their constant support.

We like to thank our **parents and friends** for their constant moral and financial support.

ABSTRACT

Due to the advancement of Internet of Things there has been an exponential increase in the amount of data generated .Signal processing techniques needs to be efficiently utilised to handle such data. An early breakthrough in signal processing was the Nyquist–Shannon sampling theorem. It states that if the signal's highest frequency is less than half of the sampling rate, then the signal can be reconstructed perfectly by means of interpolation. The major drawback of this is that the sampling rate needs to be very high if the original signal contains higher frequencies to avoid aliasing. Compressive sensing is a signal processing technique for efficiently acquiring and reconstructing a signal, by finding solutions to underdetermined linear systems. This is based on the principle that, through optimization, the sparsity of a signal can be exploited to recover it from far fewer samples than required by the Shannon-Nyquist sampling theorem.

The main objectives are to reduce the number of samples to be transmitted and development of an efficient recovery algorithm. Compressive sensing is an evolving technique for data acquisition that promises sampling a sparse signal from a far fewer measurements than its dimension. Compressive sensing enables a potentially large reduction in the sampling and computation cost for sensing signals that have sparse representation. The signal having sparse representation can be recovered from small set of linear, non-adaptive measurements. This can be achieved by applying reconstruction algorithms of normalizations and greedy methods. The recovered signals are classified by an envelope detector.

In this project, recovery of original signal after compression is demonstrated by an OMP reconstruction algorithm. The original signal is sparsed by an incoherent sparse matrix to compress the signal that is to reduce the number of samples. The compressed signal is then recovered by Orthogonal Matching Pursuit Algorithm. The recovered signals are classified on the basis of their envelope. The algorithm is implemented on ARM processor. The results obtained were accurate upto ±0.02.

CONTENTS

List of Figures

List of Tables

List of symbols, abbreviation and nomenclature

CS – COMPRESSIVE SENSING

MP – MATCHING PURSUIT

OMP – ORTHOGONAL MATCHING PURSUIT

IDE – INTEGRATED DEVELOPMENT ENVIRONMENT

RIP – RESTRICTED ISOMETRY PROPERTY

AMP – APPROXIMATE MESSAGE PASSING

TV – TOTAL VARIATION

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The age of "big data" created by the digital infrastructure of our world is now expanding. Some people may or may not have come across that term, but regardless, everyone is a part of it and is contributing to it. It refers to the sheer amount of data produced by our sensor systems that include anything from simple temperature sensors in homes to the multi-megapixel cameras everyone carries around in their pockets. A couple of decades ago the largest bottleneck for digital systems was the sensors itself. In general the signal processing and information storage infrastructure was far superior to data acquisition. However, recent advances in sensor hardware have exponentially increased the data produced and the price of owning such sensors has dropped drastically making it possible for everyone to own them. Soon, we are going to face a problem of having too much data to deal with, which might seem advantageous at first, but is most probably going to overwhelm the communication, signal processing and storage infrastructure. The data deluge has become particularly difficult to manage as we have reached the diffraction limit in lithography for microfabrication of storage devices. Even with innovative advances in these techniques to keep up with Moore's law, the rate of production of storage media is lagging severely to the rate of data acquisition and the gap is expanding. With this scenario it is quite clear that steps need to be taken such that information is not lost due to lack of storage. Loss of data might not seem like a critical problem when it concerns civilian infrastructure, but it can be a major problem for applications in military reconnaissance and large scientific experiments such as the ones performed at CERN and astronomical observatories.

Compressive sensing (CS) is an emerging and revolutionary technology where the signal of interest is efficiently acquired from very few non-zero coefficients which mean CS strongly relies on sparsity of the signal.CS is a technique that allows going beyond the Nyquist-Shannon sampling.It is very simple and efficient technique that provides both sampling and compression. Most of the signal acquisition systems have been designed based on the Nyquist Shannon sampling theorem. According to sampling theorem, an encoded signal can be reconstructed exactly if it is sampled at a frequency called sampling frequency that is at least twice the maximum frequency present in the signal. The main drawback of Nyquist sampling theorem is that it leads to large number of samples and most of them are probably not required for its reconstruction. Also in some systems, increasing the sampling rate beyond certain point make them expensive. The importance of CS is that it helps in sampling the signal below Nyquist rate. Hence CS is essential as far as storage and transmission are concerned. It also allows reconstruction of the signal at a frequency much less than the sampling frequency. Hence number of samples required for reconstructing the signal would decrease. Reducing the number of measurements will reduce the time and cost of signal acquisition. CS has applications in many fields. The prominent examples include magnetic resonance imaging (MRI), image acquisition, wireless communication and radar

Considering the traditional process in which the data needs to be acquired, compressed and then stored. In Figure 1.1 process where the data is acquired through a sensor network, digitized, compressed and then stored.

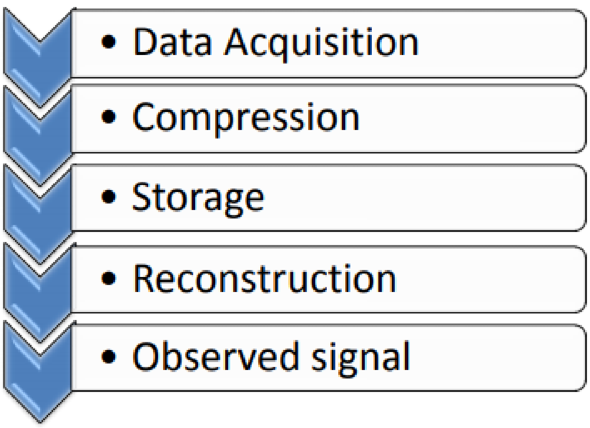

Fig 1.1 Conventional Way of acquiring data, storing it and reconstructing it

When one wants to view the data, it needs to decompress the stored compressed data to reconstruct the original analog data. Compression is the step where one throws out a lot of the data (regardless of any information loss) and store only certain important coefficients in the data that will help reconstruct it using proper algorithms. For example, JPEG is an algorithm for the compression and reconstruction of digital images. Currently, compression of the acquired data is the best way to make the storage systems more efficient. The process of first acquiring a huge amount of data and then compressing it seems quite wasteful. It would be more efficient if one could collect compressed data to begin with. That is the modus operandi on which this project is based and called “compressive sampling” or “compressive sensing”. For several decades, the collection of information, which one call “sampling,” has been governed by the Shannon-NyquistWhittaker Sampling theorem. This theorem states that perfect reconstruction of a signal is possible only when the sampling frequency is greater than twice the maximum frequency of a bandlimited signal; furthermore, by the fundamental theorem of linear algebra, the number of collected samples (measurements) of a discrete finite-dimensional signal should be at least as large as its length (its dimension), to ensure perfect reconstruction. Almost every piece of technology today is designed with these constraints.

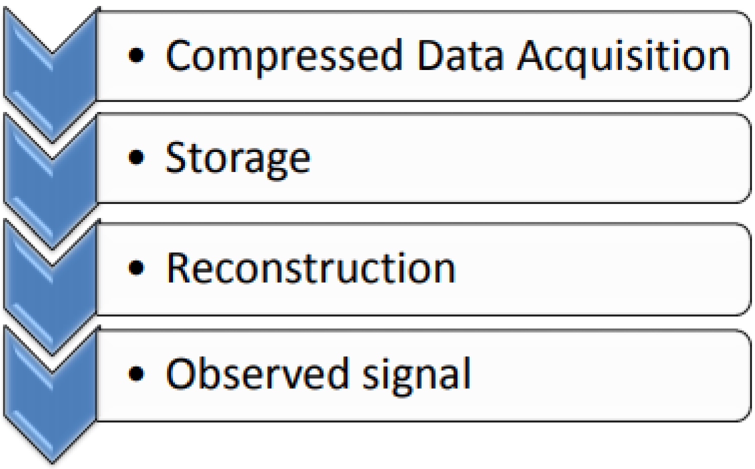

Fig 1.2 Compressed Sensing Paradigm

Compressed sensing (CS) is a new sensing paradigm where we try to combine the compression step and the sensing step for a more efficient system. Acquiring compressed data implies that one would be sampling data in violation of the Shannon-Nyquist Whittaker sampling theorem since it would be taking fewer measurements than are required for perfect reconstruction. However, that idea is precisely what needs to be explored, by taking compressed measurements to accurately reconstruct the object under certain critical assumptions that are trivial in practice.

1.2 LITERATURE REVIEW

Avi Septimus and Raphel Steinberg published a paper [1] about OMP (Orthogonal Matching Pursuit used to reconstruct the signal. Compressed signal is mathematically modeled as $y=\Phi x$, where x is i/p signal, y is number samples and Φ is sensing matrix. Minimum number of samples required to reconstruct is represented with the formula O (m.log N). This algorithm mainly consists of two stages. In the first stage number of sparse (m) is identified where as in second stage value of reconstructed signal is calculated using some iteration as per the algorithm. Using VHDL code matrix inverse unit is designed which is required to implement one of the concept used in OMP. This block is incorporated in OMP unit. This unit will take y and Φ as a input to reconstruct signal x. Results of OMP algorithm is compared with GPU implementation which is done on other processor, which indicates that OMP hardware implementation require almost negligible time.

- Need to develop consumer product using OMP unit.
- To replace existing ADC in real time application.

Guoyan Li, Junhua and Qingzeng Song [2] published paper used CG (Conjugate Gradient) algorithm Both AMP & OMP method reconstruction has a high precision, large complex calculations and speed is very less. To improve above parameter parallel and pipelined processor mapping is proposed. CG algorithm stars at interior part of rising or falling edge of the signal and later proceeds with downwards. This iterative method is called as inferior point method. CG algorithm contains some calculations based on multiplication & division on vectors. Multiplication part is build using processor's MAC unit, whereas division is implemented through software. Both parts are combined and it is called as CG processor. This processor implements internal point method whereas remaining part of the algorithm is implemented in other hardware. Software experiment is implemented DSP Builder 11.1 tools. Paper is concluded with following future work

- Need to develop more & more parallel unit to improve complexity.

Patric Machler, Lin Bai, Hubert and Michel published paper [3] on compressed sensing. CS is mathematically modeled as y=Φx, where x is i/p signal, Φ is non-adaptive linear projections. Recovering original signal is done by solving y=Φx = y=Φψs,(ψs is wavelet transform), under certain conditions. OMP uses the concept of adding best fitting element in each steps of algorithm. AMP uses thresholding concept to refine the estimation in step as per the algorithm. OMP consist of matrix multiplication as well as least square optimization operations. To reduce the complexity of operations, large number of parallel multiplier is instantiated using MAC units and it configured as vector multiplication unit (VMU). For this VMU parallel data (Fixed floating point arithmetic) is supplied sequential circuits such as RAM, ROM, REG, SRR and MAX. AMP uses modified version of VMU, which is operated in two modes that is parallel and serial mode. Both algorithm uses problem size of 256x1024. These are implemented on Xilinux vertex-6 (XC6VLX240T). FPGA is tested in PC through UART interface by taking Lena image of size 256x256 pixels as a test input. These FPGA results are compared with MATLAB reconstruction, which states that OMP is 4000 & AMP is 5000 times faster than software.

Mohammed M. Abo-Zahhad, Aziza I. Hussein and Abdelfatah M. Mohamed published a paper on the various algorithms [4].In this paper the author has discussed about the various algorithm available for reconstruction. This paper highlights the need for an efficient algorithm for recovery of signals

Emmanuel J Candes and Michel B. Wakin published a paper [6] on the mathematical model involved is introduced .Conditions for Signal Recovery, the incoherence property and Restricted Isometric Property are introduced in this paper. Need to develop a matrix which satisfies both the conditions

1.3 MOTIVATION

Recent advances in Internet of Things have exponentially increased the data generated and the price of bandwidth paid to transmit the same is very high. The world is going to face a problem of having too much data to deal with, which might seem advantageous at first, but is most probably going to overwhelm the communication, signal processing and storage infrastructure. To put the idea in perspective by looking at some numbers from the past decade. In 2010, the amount of data generated was in the ballpark of 10^{20} bytes and in the last 5 years we have produced over 90% of the data generated ever and this amount is expected to increase over tenfold in the next 10-15 years. In 2007, we reached a critical milestone where we produced more data than we had storage available in the world; and in 2011, we produced twice as much data as we had storage available and the rate is growing exponentially.

The data deluge has become particularly difficult to manage as we have reached the diffraction limit in lithography for micro-fabrication of storage devices. Even with innovative advances in these techniques to keep up with Moore's law, the rate of production of storage media is lagging severely to the rate of data acquisition and the gap is expanding. With this scenario it is quite clear that steps need to be taken such that information is not lost due to lack of storage. Loss of data might not seem like a critical problem when it concerns civilian infrastructure, but it can be a major problem for applications in military reconnaissance and large scientific experiments

The process of first acquiring a huge amount of data and then compressing it seems quite wasteful. It would be more efficient if we could collect compressed data to begin with.

1.4 PROBLEM DEFINITION

Nyquist-rate sampling completely describes a signal by exploiting its bandlimitedness. The goal is to reduce the number of measurements required to completely describe a signal by exploiting its compressibility. The twist is that the measurements are not point samples but sparse vectors of the original signal. The compressed signal needs to be reconstructed back to the original signal.

1.5 OBJECTIVE

The main objective of this mini project involves the design and implementation of compressing sensing algorithms in simulation and on an ARM controller and comparison of the performance parameters of the said algorithms. Finally, an efficient real time algorithm is implemented on the hardware.

- To develop a stable measurement matrix such that the salient information in any K-sparse or compressible signal is not damaged by the dimensionality reduction from $x \in RN$ to $y \in RM$ and
- To develop reconstruction algorithm to recover x from only $M \approx K$ measurements.
- To Classify the signals of compressive sensing reconstruction algorithm using envelope detectors.

1.6 METHODOLOGY

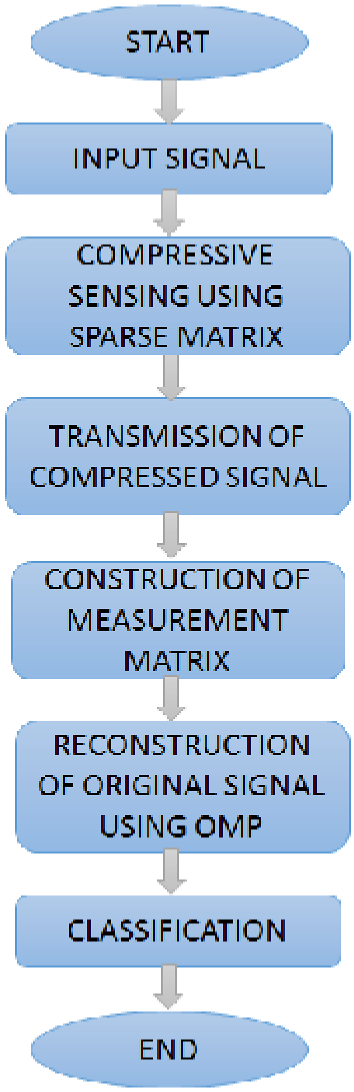

Fig 1.3 Flowchart of methodology

1.7 ORGANISATION OF THE REPORT

Chapter 2 discusses the compressive sensing paradigm and some important assumptions such as sparsity that make the process possible.

Chapter 3 provides the overview of the design and the reconstruction of the compressively sampled data and the algorithms used.

Chapter 4 discusses the implementation of experimental setup on an ARM processor to create an efficient system for reconstruction.

Chapter 5 includes the simulation results and the results obtained on hardware.

Chapter 6 end the report with conclusions and future scope.

CHAPTER 2

THEORY AND FUNDAMENTALS OF COMPRESSIVE SENSING

CHAPTER 2

THEORY AND FUNDAMENTALS OF COMPRESSIVE SENSING

In this chapter the theoretical and mathematical concepts behind the project are introduced. It contains the fundamentals of the Shanon-Nyquist Theorem and Compressive Sensing.

2.1 SHANON-NYQUIST THEOREM

Sampling is a process of converting a signal into a numeric sequence (a function of discrete time and/or space). Shannon's version of the theorem states:

If a function x(t) contains no frequencies higher than B hertz, it is completely determined by giving its ordinates at a series of points spaced 1/(2B) seconds apart.

A sufficient sample-rate is therefore 2B samples/second, or anything larger. Equivalently, for a given sample rate fs(Fig2.1) perfect reconstruction is guaranteed possible for a bandlimit B < fs/2.

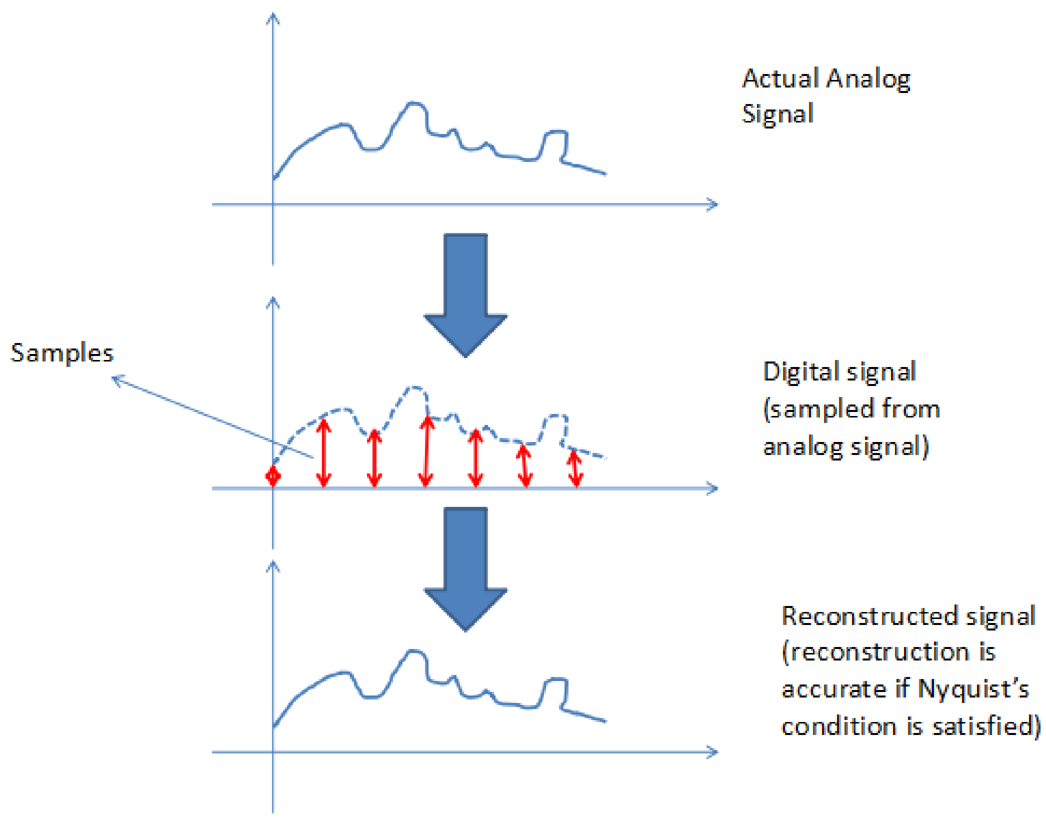

Fig 2.1 Shanon-Nyquist Theorem [5]

When the bandlimit is too high (or there is no bandlimit), the reconstruction exhibits imperfections known as aliasing. Modern statements of the theorem are sometimes careful to explicitly state that x(t) must contain no sinusoidal component at exactly frequency B, or that B must be strictly less than ½ the sample rate. The two thresholds, 2B and fs/2 are respectively called the Nyquist rate and Nyquist frequency. And respectively, they are attributes of x(t) and of the sampling equipment. The condition described by these inequalities is called the Nyquist criterion. The theorem is also applicable to functions of other domains, such as space, in the case of a digitized image. The only change, in the case of other domains, is the units of measure applied to t, fs, and B.The normalized sinc function: $\sin(\pi x) / (\pi x)$ ... showing the central peak at x= 0, and zero-crossings at the other integer values of x. The symbol T = 1/fs is customarily used to represent the interval between samples and is called the sample period or sampling interval. And the samples of function x(t) are commonly denoted by x[n] = x(nT) , for all integer values of n. A mathematically ideal way to interpolate the sequence involves the use of sinc functions. Each sample in the sequence is replaced by a sinc function, centered on the time axis at the original location of the sample, nT, with the amplitude of the sinc function scaled to the sample value, x[n]. Subsequently, the sinc functions are summed into a continuous function.

A mathematically equivalent method is to convolve one sinc function with a series of Dirac delta pulses, weighted by the sample values. Neither method is numerically practical. Instead, some type of approximation of the sinc functions, finite in length, is used. The imperfections attributable to the approximation are known as interpolation error.Sampling an analog signal with maximum frequency B at a rate less than or equal to 2B causes an artifact called aliasing.

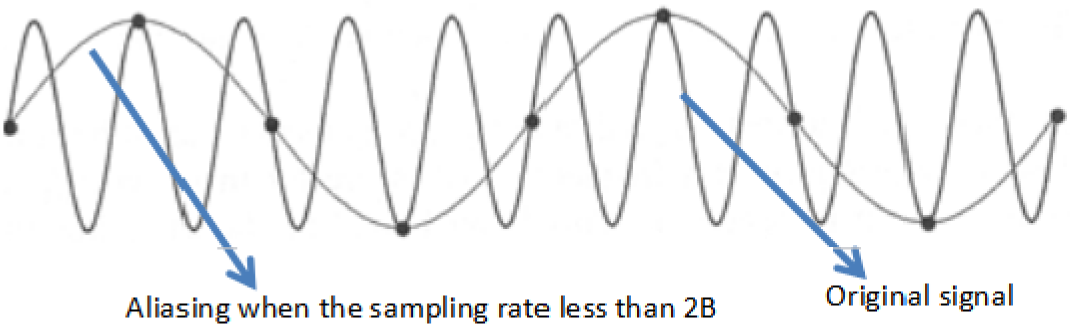

Fig 2.2 Aliasing

Practical digital-to-analog converters produce neither scaled and delayed sinc functions, nor ideal Dirac pulses. Instead they produce a piecewise-constant sequence of scaled and delayed rectangular pulses (the zero-order hold), usually followed by an "anti-imaging filter" to clean up spurious high-frequency content.The following are the limitations of Shanon-Nyquist Theorem- The samples need to be uniformly spaced (there are extensions for non-uniformly spaced samples, with the equivalent of Nyquist rate being an average sampling rate).The sampling rate needs to be very high if the original signal contains higher frequencies (to avoid aliasing).Does not account for several nice properties of naturally occurring signals (except for band-limitedness).

2.2 PRINCIPLES OF COMPRESSIVE SENSING

Compressed sensing (also known as compressive sensing, compressive sampling, or sparse sampling) is a signal processing technique for efficiently acquiring and reconstructing a signal, by finding solutions to underdetermined linear systems. This is based on the principle that, through optimization, the sparsity of a signal can be exploited to recover it from far fewer samples than required by the Shannon-Nyquist sampling theorem. There are two conditions under which recovery is possible. The first one is sparsity which requires the signal to be sparse in some domain. The second one is incoherence which is applied through the isometric property which is sufficient for sparse signals.

A common goal of the engineering field of signal processing is to reconstruct a signal from a series of sampling measurements. In general, this task is impossible because there is no way to reconstruct a signal during the times that the signal is not measured. Nevertheless, with prior knowledge or assumptions about the signal, it turns out to be possible to perfectly reconstruct a signal from a series of measurements. Over time, engineers have improved their understanding of which assumptions are practical and how they can be generalized.

An early breakthrough in signal processing was the Nyquist–Shannon sampling theorem. It states that if the signal's highest frequency is less than half of the sampling rate, then the signal can be reconstructed perfectly by means of sinc interpolation. The main idea is that with prior knowledge about constraints on the signal’s frequencies, fewer samples are needed to reconstruct the signal.

Around 2004, Emmanuel Candès, Terence Tao, and David Donoho proved that given knowledge about a signal's sparsity, the signal may be reconstructed with even fewer samples than the sampling theorem requires. This idea is the basis of compressed sensing.

2.3. MATHEMATICAL MODEL

2.3.1 Underdetermined linear system

An underdetermined system of linear equations has more unknowns than equations and generally has an infinite number of solutions. The figure below shows such an equation system $y = D x$, $y=Dx$ where we want to find a solution for x. .

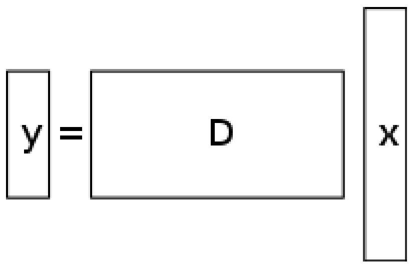

Fig 2.3 System Equation

In order to choose a solution to such a system, one must impose extra constraints or conditions (such as smoothness) as appropriate. In compressed sensing, one adds the constraint of sparsity, allowing only solutions which have a small number of nonzero coefficients. Not all underdetermined systems of linear equations have a sparse solution. However, if there is a unique sparse solution to the underdetermined system, then the compressed sensing framework allows the recovery of that solution.

Normed vector spaces

Throughout this book, we will treat signals as real-valued functions having domains that are either continuous or discrete, and either infinite or finite. Theseassumptions will be made clear as necessary in each chapter. We will typically beconcerned with normed vector spaces, i.e.vector spaces endowed with a norm.In the case of a discrete, nite domain, one can view our signals as vectors in an n-dimensional Euclidean space, denoted by Rn. When dealing with vectors in Rn, we will make frequent use of the `p norms, which are defined for

$$\|x\|_p = \begin{cases} \left(\sum_{i=1}^{n} |x_i|^p\right)^{\frac{1}{p}}, & p \in [1, \infty); \\ \max\limits_{i=1,2,\ldots,n} |x_i|, & p = \infty. \end{cases} \qquad \ldots\ldots(2.1)$$

2.3.2 Solution / reconstruction method

Compressed sensing takes advantage of the redundancy in many interesting signals—they are not pure noise. In particular, many signals are sparse, that is, they contain many coefficients close to or equal to zero, when represented in some domain. This is the same insight used in many forms of lossy compression.

Compressed sensing typically starts with taking a weighted linear combination of samples also called compressive measurements in a basis different from the basis in which the signal is known to be sparse. The number of these compressive measurements can be small and still contain nearly all the useful information. Therefore, the task of converting the image back into the intended domain involves solving an underdetermined matrix equation since the number of compressive measurements taken is smaller than the number of pixels in the full image. However, adding the constraint that the initial signal is sparse enables one to solve this underdetermined system of linear equations.

The least-squares solution to such problems is to minimize the L2 norm—that is, minimize the amount of energy in the system. This is usually simple mathematically (involving only a matrix multiplication by the pseudo-inverse of the basis sampled in). However, this leads to poor results for many practical applications, for which the unknown coefficients have nonzero energy.

To enforce the sparsity constraint when solving for the underdetermined system of linear equations, one can minimize the number of nonzero components of the solution. The function counting the number of non-zero components of a vector was called the L^0 or L0 "norm" by David Donoho .This equivalence result allows one to solve the L1 problem, which is easier than the L2 problem. Finding the candidate with the smallest L1, L1 norm can be expressed relatively easily as a linear program, for which efficient solution methods already exist. When measurements may contain a finite amount of noise, basis pursuit denoising is preferred over linear programming, since it preserves sparsity in the face of noise and can be solved faster than an exact linear program.

2.3.3 Sparsity Model

A signal is said to be sparse if majority of its components are zero. A mathematical definition can be expressed using the concept of, the support of a vector x, which is the index set of its nonzero entries given by,

$$(x) := \{j\epsilon\ [N]: xj \neq 0\}. \qquad \text{.........(2.2)}$$

The vector x is called s-sparse if at most s of its entries are nonzero, i.e.,

$$||x||0 := (sup(x)) \leq s. \qquad \text{.........(2.3)}$$

Here the operator $card$ is the cardinality of the support of x, i.e., it counts the non-zero indices of x which are defined by $supp$. $||x||0$ is the 0 norm of the signal x, where the norm in general is defined by

$\|x\|p := (\sum|x|\ )\ 1\ p$. ………(2.4)

In reality, many real world signals can be approximated well by sparse signals in some basis.

The measurement matrix Φ is incoherent (poorly correlated) with the signal basis matrix Ψ, i.e. the sensing waveforms have a very dense representation in the signal basis.

Many signals have sparse representations in standard orthonormal bases

$$f = \Psi\theta = \sum_{k=1}^{n} \Psi_k \theta_k ,$$

$$f \in R^n , \theta \in R^n , \Psi \in R^{n\times n} , \Psi^T \Psi = I , \qquad \text{…….(2.5)}$$

$$\|\theta\|_0 << n$$

2.4 Existing approaches of Reconstruction

2.4.1 Iteratively reweighted $l_1$minimization

In the CS reconstruction models using constrained minimization l1, larger coefficients are penalized heavily in the l 1 l1 norm. It was proposed to have a weighted formulation of l 1 minimization designed to more democratically penalize nonzero coefficients. An iterative algorithm is used for constructing the appropriate weights. Each iteration requires solving one l1 minimization problem by finding the local minimum of a concave penalty function that more closely resembles the l0 norm. An additional parameter, usually to avoid any sharp transitions in the penalty function curve, is introduced into the iterative equation to ensure stability and so that a zero estimate in one iteration does not necessarily lead to a zero estimate in the next iteration. The method essentially involves using the current solution for computing the weights to be used in the next iteration.

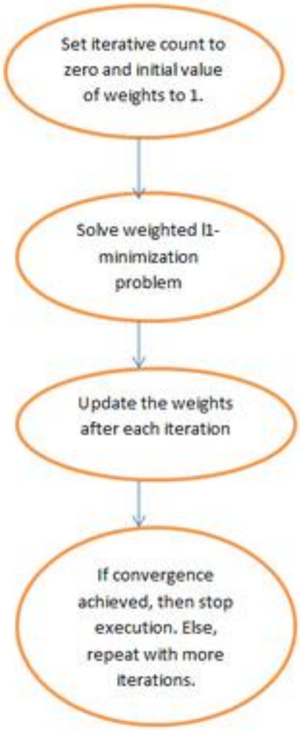

Fig 2.4 Iteratively reweighted l1 minimization method for CS

Early iterations may find inaccurate sample estimates, however this method will down-sample these at a later stage to give more weight to the smaller non-zero signal estimates. One of the disadvantages is the need for defining a valid starting point as a global minimum might not be obtained every time due to the concavity of the function. Another disadvantage is that this method tends to uniformly penalize the image gradient irrespective of the underlying image structures. This causes over-smoothing of edges, especially those of low contrast regions, subsequently leading to loss of low contrast information. The advantages of this method include: reduction of the sampling rate for sparse signals; reconstruction of the image while being robust to the removal of noise and other artefacts; and use of very few iterations. This can also help in recovering images with sparse gradients.

In the figure 2.4 P1 refers to the first-step of the iterative reconstruction process, of the projection matrix P of the fan-beam geometry, which is constrained by the data fidelity term. This may contain noise and artifacts as no regularization is performed. The minimization of P1 is solved through the conjugate gradient least squares method. P2 refers to the second step of the iterative reconstruction process wherein it utilizes the edge-preserving total variation regularization term to remove noise and artifacts, and thus improve the quality of the reconstructed image/signal. The minimization of P2 is done through a simple gradient descent method. Convergence is determined by testing, after each iteration, for image positivity, by checking iff^{k-1}=0for the case when $f^{k-1} < 0$

2.4.2 Edge-preserving total variation (TV) based compressed sensing

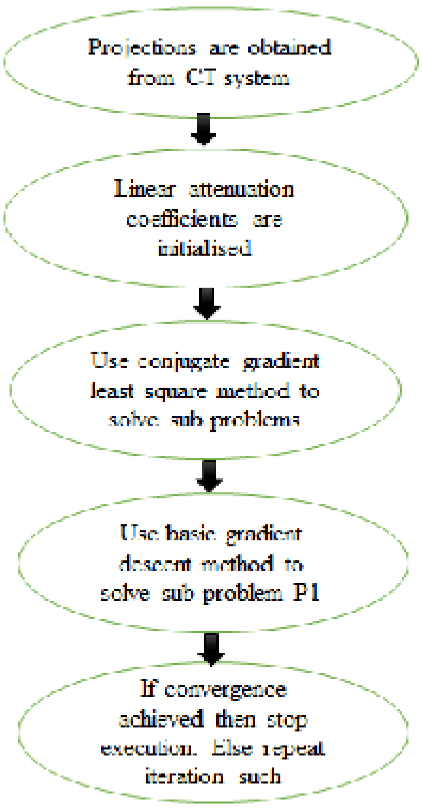

Fig 2.5 Flow diagram figure for edge preserving total variation method for compressed sensing

This is an iterative CT reconstruction algorithm with edge-preserving TV regularization to reconstruct CT images from highly undersampled data obtained at low dose CT through low current levels (milliampere). In order to reduce the imaging dose, one of the approaches used is to reduce the number of x-ray projections acquired by the scanner detectors. However, this insufficient projection data which is used to reconstruct the CT image can cause streaking artifacts. Furthermore, using these insufficient projections in standard TV algorithms end up making the problem under-determined and thus leading to infinitely many possible solutions. In this method, an additional penalty weighted function is assigned to the original TV norm. This allows for easier detection of sharp discontinuities in intensity in the images and thereby adapt the weight to store the recovered edge information during the process of signal/image reconstruction. The parameter σ controls the amount of smoothing applied to the pixels at the edges to differentiate them from the non-edge pixels. The value of σ is changed adaptively based on the values of the histogram of the gradient magnitude so that a certain percentage of pixels have gradient values larger than σ . The edge-preserving total variation term, thus, becomes sparser and this speeds up the implementation. A two-step iteration process known as forward-backward splitting algorithm is used. The optimization problem is split into two sub-problems which are then solved with the conjugate gradient least squares method and the simple gradient descent method respectively. The method is stopped when the desired convergence has been achieved or if the maximum number of iterations is reached.

Some of the disadvantages of this method are the absence of smaller structures in the reconstructed image and degradation of image resolution. This edge preserving TV algorithm, however, requires fewer iterations than the conventional TV algorithm. Analyzing the horizontal and vertical intensity profiles of the reconstructed images, it can be seen that there are sharp jumps at edge points and negligible, minor fluctuation at non-edge points. Thus, this method leads to low relative error and higher correlation as compared to the TV method. It also effectively suppresses and removes any form of image noise and image artifacts such as streaking.

2.4.3 Greedy approach

Greedy algorithm mainly relies on iterative approximation of signal coefficients and support. The algorithm computes the support of sparse signal iteratively. Once the support of the signal is computed correctly, then the pseudo-inverse of the measurement matrix can be used to reconstruct the signal, Main advantage of using this approach is for its speed of computation. Orthogonal matching pursuit (OMP) is one of the greedy algorithms used here.

Chapter 3

IMPLEMENTATION OF SPARSE MATRIX AND RECONSTRUCTION ALGORITHM

Chapter 3

IMPLEMENTATION OF SPARSE MATRIX AND RECONSTRUCTION ALGORITHM

3.1. Solving minimization

Several different schemes have been proposed for solving the l1-minimization. However, we chose to compare the performance of only three well-known algorithms: Matching Pursuit (MP), Orthogonal Matching Pursuit (OMP), and Approximate Message Passing (AMP). We chose MP since it is known for its speed and relative simplicity, and we chose OMP and AMP since they are known for their performance. In general, there is a trade-off between performance and complexity of the algorithm. Here, MP has the lowest complexity while the OMP and AMP are more complex and have longer running time for each iteration. For example, OMP requires matrix inversion, and AMP requires square-root computation and a significant number of matrix multiplications in each iteration which make them more complex to implement in hardware compared to MP.

On the other hand, OMP and AMP require a smaller number of iterations to reconstruct the signal, and they also have better performance. Among these algorithms, AMP has the best performance as it performs the same as l1 minimization. However, AMP has higher complexity and longer running time compared to the other two algorithms which make it unsuitable for some applications which require high speed/throughput reconstruction. For example, in the AIC system used in a cognitive radio system, MP might be a better option as this application requires fast reconstruction process. In addition, it does not require a very precise reconstruction, and it has higher tolerance to the reconstruction error. Therefore, the performance of these algorithms should be compared in terms of both speed and quality of reconstruction in practical environments to decide which one would be a better choice for a specific application.

3.2 Reconstruction Algorithms

3.2.1 Conditions for Signal Recovery

The signal should satisfy the following two conditions for signal recovery

Restricted Isometry Property

An important consideration for robust compressive sampling is the ability of the measurement matrix to preserve the important pieces of information from the signal of interest in the inner product. This property is verified by applying something known as the Restricted Isometry Property (RIP) of the measurement matrix. RIP is defined in terms of the Restricted Isometry Constant (RIC) for every sparse vector x, denoted by $\boldsymbol{\delta}$ such that,

$$(1-\delta)||x||2\ 2 \leq ||\boldsymbol{A}x||2\ 2 \leq (1+\delta)\ ||x||2 \qquad \text{………(3.1)}$$

Literally speaking, this inequality is NP-hard in nature and cannot be computed easily and the RIC is unbounded. However, for certain random matrices it has been shown that the RIC can remain bounded. RIC is generally in the range of 0 to 1 for random matrices. The closer the RIC is to 0 the better the property of the matrix to hold the information about the sparse vector x in the inner product. RIP ensures that all subsets of s columns from the measurement matrix are orthogonal and that the sparse signal is not in the null space of the matrix i.e. it does not have a trivial solution where the inner product Ax is 0.

Coherence

When fewer measurements than required by the Nyquist sampling theorem are being taken, it is worthwhile to put a bound on the minimum number of measurements required to accurately reconstruct the desired signal. The number of measurements required for robust sampling and the ability to reconstruct from those measurements depends on the coherence of the measurement matrix. Coherence is a property that measures the maximum correlation between the elements of two matrices. The two matrices can be the measurement matrix and the matrix representing the basis in which the signal is sparse. It is important for the two matrices to have low coherence for compressive sampling to work. If the measurement matrix is given by φ and the representation basis matrix is given by ∂ then the coherence is given by,

$$\mu\,(\varphi, \partial) = \sqrt{n}\ \max\ 1 \leq k, \leq n\ |\langle \varphi k, \partial j \rangle| \qquad \ldots\ldots(3.2)$$

Here we take the inner product of the vectors represented by φk and ∂j to find the correlation between the two matrices. The representation bases are common signal representations such as the pixel basis, Fourier basis, wavelet basis, discrete cosine basis etc. It is worthwhile to have measurement matrices that can have low coherence with any representation basis. This property is fulfilled by random matrices such as Gaussian, Bernoulli and pseudorandom binary matrices. In the next chapter, we will use particular convolutions of pseudorandom binary matrices for compressive laser ranging as they are incoherent with any basis and provide the best results for compressive sampling in an unknown basis, which in our case will be the range domain

3.3 Algorithms

Greedy algorithms solve the reconstruction problem by iteration. The algorithm looks for columns in the measurement matrix in greedy fashion, which correlate with the measurement vector iteratively. Each iteration minimizes the least-square error and subtracts the residual from the measurement vector for the next iteration. The process continues until the exact columns corresponding to the measurement vector are identified. Matching pursuit algorithms are the most widely used greedy algorithms for sparse approximation. For this project the OMP (Orthogonal Matching Pursuit) and CoSaMP (Compressive Sampling Matching Pursuit) were used which are derived from the Matching Pursuit algorithm developed by Mallat and Zhang in 1993.25

Orthogonal Matching Pursuit. Orthogonal Matching Pursuit (OMP) was developed my Mallat and Zhang in 1993 and demonstrated for compressive sensing by Gilbert and Tropp in 2007.OMP iteratively constructs an approximation by calculating a locally optimal solution at each step. The locally optimal solutions add up to form the final globally optimal solution that is an approximation of the original signal x. The basic premise is to find columns in A that most closely correspond to the measurement vector y. Since x is sparse, the measurement y is formed from the linear combination of the columns in A that correspond to the non-zero elements in x. The algorithms take the measurement vector y, the measurement matrix A and the sparsity level s as the inputs. The measurement vector y is considered as a residual and is updated every iteration. The residual is updated by subtracting the contribution of the largest column from A at each iteration to give the largest columns in A that correspond to the non-zero elements in the vector x. The algorithm is described in Table 1.

Table 1 OMP Algorithm[26,27]

OMP(A, y, s)
Input: Measurement matrix A, measurement vector y, sparsity level s Output: s-sparse approximation a of the signal
$a^0 \leftarrow 0$ $v \leftarrow y$ $k \leftarrow 0$ $repeat$ $k \leftarrow k+1$ $z \leftarrow A * v$ $\Omega \leftarrow argmax(z)$ $T \leftarrow \Omega \cup supp(a^{k-1})$ $b\|_T \leftarrow A_T^{\dagger} y$ $a^k \leftarrow b_s$ $v \leftarrow u - Aa^k$

The algorithm begins by initializing the approximation a to 0, the residual v to the measurement vector y and the stopping condition k to 0. The algorithm then goes into a loop to iteratively approximate the signal until the stopping condition is met, which for this project was the sparsity s of the signal. An iteration starts off by taking the inner product between the measurement matrix A and the residual v, and finding the largest correlation between the two, Ω. The new value is combined with the vector containing the values from earlier iterations in the support vector T. A least-square approximation of the signal is then calculated using the columns with indices corresponding to the support vector T and is stored in b which is then assigned to the indices of the non-zero values in the signal approximation a. Finally, the residual from the current iteration is subtracted from the total residual before the iteration is run again. The recovered approximation to the signal x is based on its sparsity and the number of measurements taken. As the sparsity increases, so do the amount of measurements required for an accurate reconstruction, a characteristic seen in every recovery algorithm. The number of measurements required for accurate reconstruction is given by

$$m = slog(n), \qquad \text{.......(3.3)}$$

Where s is the sparsity and n is the length of the unknown signal.

3.4 Matching Pursuit

Matching Pursuit (MP) is an iterative greedy algorithm that finds the sparse solution x subject to y = ΦΨx, where Ψ is the basis matrix, Φ is the sampling 66 matrix, and y is measurements. A flow chart of the algorithm is shown in, where rt denotes the residual/error at t-th iteration. MP iteratively improves its estimate of the signal by choosing the column of the matrix A (where A = ΦΨ) that has the highest correlation with the residual rt . Then, it subtracts the correlated column from the approximation error/residual and then iterates the procedure on the newly obtained approximation error. The algorithm stops if the norm of the residual falls below a threshold, or if the number of iterations (k) reaches to the limit L. Note that even if we perform M iterations of MP, it is not guaranteed that we will obtain an error of zero, though the asymptotical convergence of MP for k → ∞ has been proven.

Let Φ denote the dictionary of atoms as a N-by-M matrix with M>N. If the complete, redundant dictionary forms a frame for the signal space, you can obtain the minimum L2 norm expansion coefficient vector by using the frame operator.

$$\Phi^{\dagger}=\Phi^*(\Phi\,\Phi^*)^{-1} \qquad \ldots\ldots(3.4)$$

However, the coefficient vector returned by the frame operator does not preserve sparsity. If the signal is sparse in the dictionary, the expansion coefficients obtained with the canonical frame operator generally do not reflect that sparsity. Sparsity of your signal in the dictionary is a trait that you typically want to preserve. Matching pursuit addresses sparsity preservation directly.

Matching pursuit is a greedy algorithm that computes the best nonlinear approximation to a signal in a complete, redundant dictionary. Matching pursuit builds a sequence of sparse approximations to the signal stepwise. Let Φ= {φk} denote a dictionary of unit-norm atoms. Let f be your signal.

1. Start by defining $R^0 f = f$

2. Begin the matching pursuit by selecting the atom from the dictionary that maximizes the absolute value of the inner product with $R^0 f$ = f. Denote that atom by φ_p.

3. Form the residual $R^1 f$ by subtracting the orthogonal projection of R0f onto the space spanned by φp.

$$R^1 f = R^0 f - \langle R^0 f, \phi_p \rangle \phi_p$$

4. Iterate by repeating steps 2 and 3 on the residual.

$$R^{m+1} f = R^m f - \langle R^m f, \phi_k \rangle \phi_k$$

5. Stop the algorithm when you reach some specified stopping criterion.

In nonorthogonal (or basic) matching pursuit, the dictionary atoms are not mutually orthogonal vectors. Therefore, subtracting subsequent residuals from the previous one can introduce components that are not orthogonal to the span of previously included atoms.

3.5 Orthogonal Matching Pursuit

The selection criterion of the OMP algorithm is the same as the MP algorithm. The main difference between MP and OMP is in their projection step. At each iteration of MP, the measurements are projected on only selected column of matrix A, and, hence, the residual may not be orthogonal to the subspace span by Ai unless the columns of A are orthogonal to each other. In contrast to MP, at each iteration of OMP, the measurements y are projected on the range of all previously selected columns of matrix A, and, as a result, the newly derived residual is orthogonal not only to the immediately selected column, but also to all the columns selected at previous iterations.

As a consequence, once a column is selected, it is never selected again in subsequent iterations. Approximate Message Passing Approximate message passing (AMP) has been recently proposed as an effective algorithm to reconstruct a sparse signal from a small number of incoherent linear measurements. It has been shown that this algorithm performs exactly the same as l1 minimization while it is running fast. AMP flowchart where zt and xt are residual and the estimate of the signal at time t (t-th iteration), respectively.

Note that

$$< u >= P/N \ i=1 \ u(i)/N \text{ , and } \delta = M/N. \qquad \ldots\ldots.(3.5)$$

Here,ηt() is a threshold function. Many threshold policies have been proposed for AMP. However, we can simply set the threshold to the magnitude of the M-th largest coefficient in absolute value.

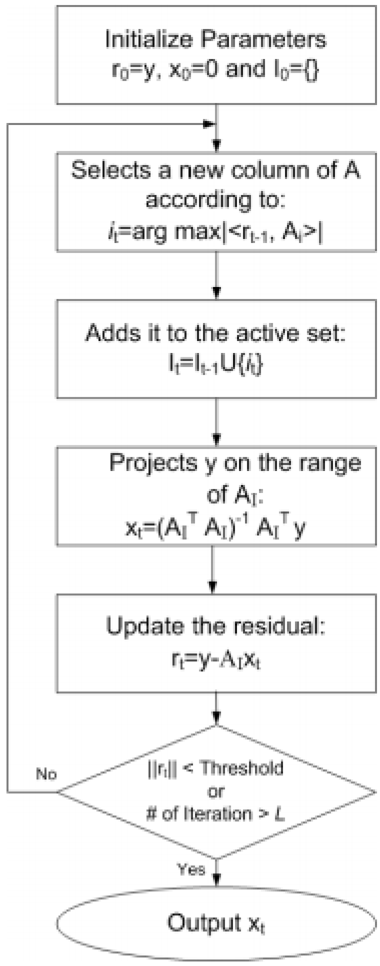

Fig 3.1 OMP Flowchart

The algorithm for CoSaMP is quite similar to the OMP algorithm. The algorithm starts off by initializing the values of the signal approximation, the residual and the stopping condition. The inner product of the measurement matrix and the residual is calculated for each iteration. The next step differs from OMP, where instead of searching for the largest correlation between the measurement matrix and the residual, the algorithm searches for $2s$ largest components in the inner product. The largest components are merged with the approximation from the previous iteration and a least squares approximation of the signals is calculated and the most relevant k values are saved in the approximation. The signal approximation is updated and the weight of these values is subtracted from the residual before going to the next iteration. CoSaMP provides better bounds on its results because it searches for $2s$ components instead of one largest component, improving the bounds on Restricted Isometry Constant (RIC, seen in the next section) and making it more robust than OMP.

3.6 Time-sparse Application

In this section, we evaluate the performance of MP, OMP and AMP algorithms when signal of interest is time-sparse (i.e. the signal is sparse in time domain). To illustrate the performance comparison results, random signals of length N = 1000 with S non-zero coefficients are generated by drawing on a uniform random distribution over [-1,1] to assign the sign and magnitude, and over [1,1000] to assign the position of each non-zero value in the S-sparse signal. The generated signal is corrupted by white Gaussian noise and is quantized to Q bits. Then, the quantized signal is multiplied by a sampling matrix to generate CS measurements. Finally, the CS measurements are used to reconstruct the signal. We first evaluate the performance of the algorithms in terms of probability of reconstruction failure.

Here, we set the failure threshold to P RD of -2dB which corresponds to ENOB of 8 bits. In other words, a reconstruction is called a failure if the reconstructed signal has P RD > −2dB (i.e., resolution of the reconstructed signal is lower, OMP Figure 4-4: signal sparsity versus probability of reconstruction failure for MP, AMP and OMP when the signal of interest is sparse in time domain (N = 1000, M = 50, Failure threshold set to P RD = −2dB). performs better than AMP for input signal with sparsity level (S) higher than 5. Although, here, OMP algorithm performs better that AMP for this set-up and some range of sparsity, its performance might become worse than AMP for some other cases (i.1e. different block size, signal and etc).

However, since MP is less computational complex, it can still be a good choice for applications which require fast reconstruction but does not require very low probability of failure. Finally, it should be noted that the performance of the algorithms can be improved by increasing the number of measurements, M.

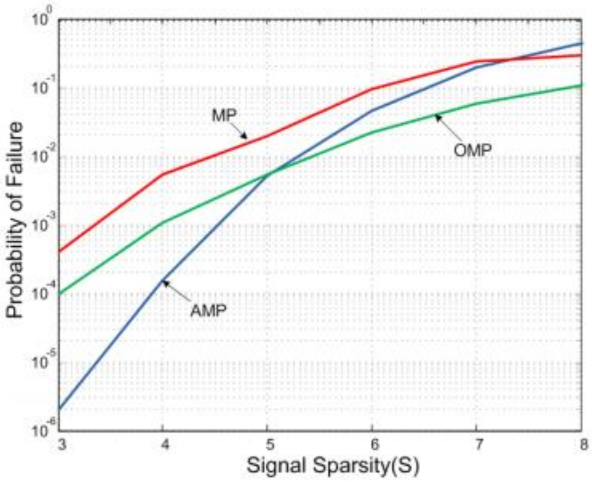

Fig 3.2 Signal Sparsity vs. Probability of Failure[5]

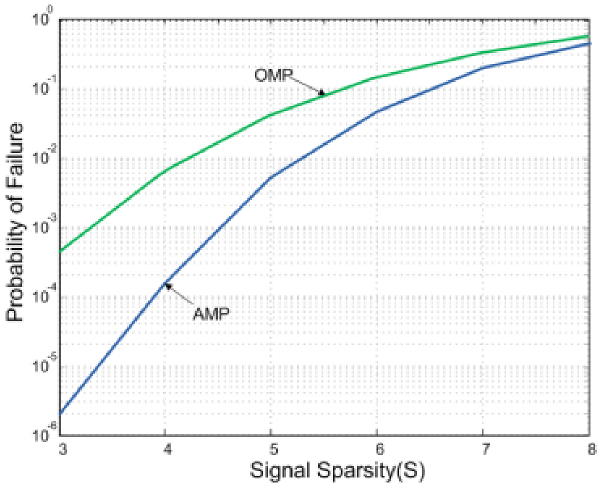

Fig 3.3 Signal Sparsity for OMP[5]

Required Number of Measurements

As it was mentioned, increasing M can improve the performance of the algorithms.

Hence, the other factor which needs to be considered to evaluate the performance of algorithms is the number of measurements, M, required to meet a desired probability of failure. Figure 3.4 shows the required M to achieve the probability of failure of 1% and 0.1%. As it is shown, there is no significant difference in the required M for OMP versus AMP, while MP requires slightly higher number of measurements to achieve the same probability of failure as the other two algorithms.

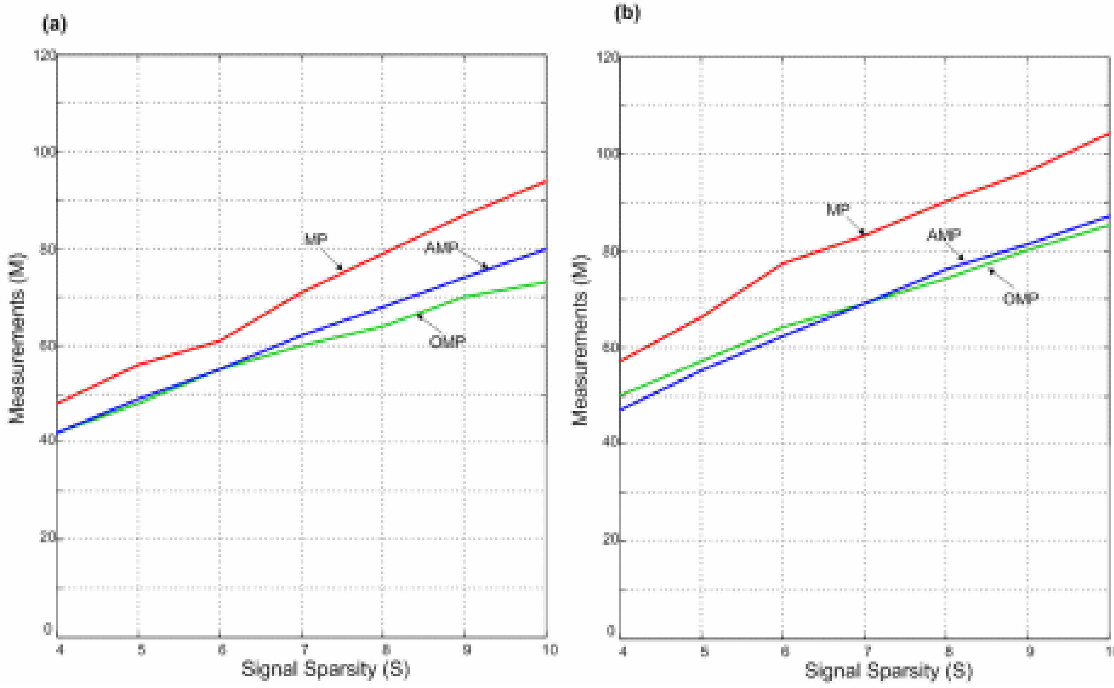

Fig 3.4 Signal Sparsity vs Measurements [5]

3.7 Frequency-sparse Application

In this section, we perform the same evaluation. However, instead, we assume the signal of interest is frequency-sparse (i.e. the signal is sparse in the frequency domain). One example of these applications is AIC used in cognitive radio. To illustrate the performance comparison results, we use the same signal model explained number of measurements curves for the frequency-sparse signal. We see similar observations as the time-sparse signal in the performance of the algorithms.

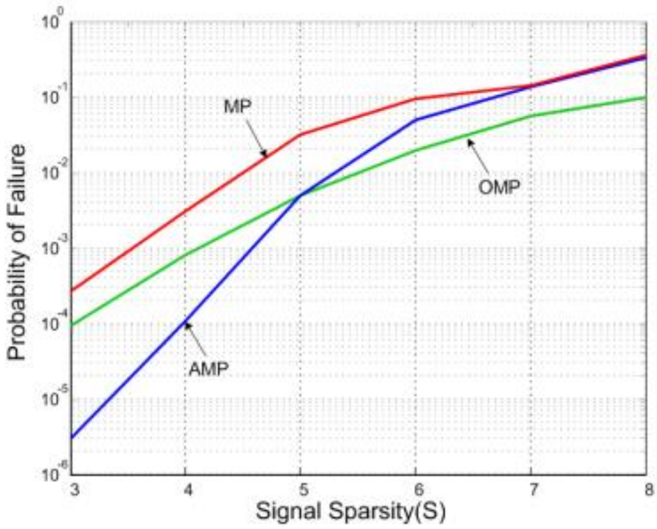

Fig 3.5 Comparison of probability of failure[5]

3.8 Evaluation Summary

In conclusion, depending on the application, one algorithm may be more suitable than the other. For example, for an application which require fast reconstruction but does not need very low probability of failure, MP may be a more suitable choice. On the other hand, AMP can provide higher performance at the cost of reconstruction time and higher complexity.

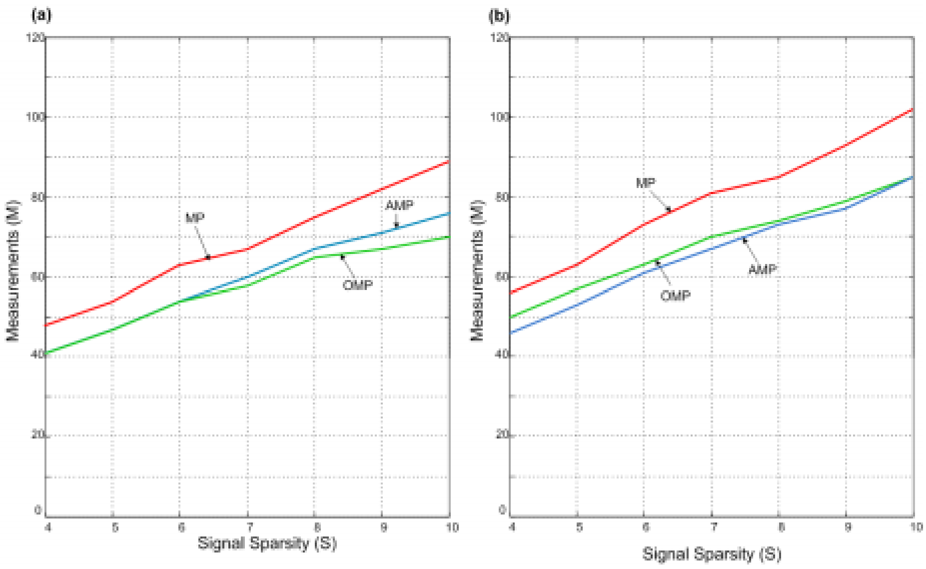

Fig 3.6 Comparison of Measurements

CHAPTER 4

HARDWARE IMPLEMENTATION

CHAPTER 4

HARDWARE IMPLEMENTATION

4.1 HARDWARE PLATFORM

The hardware unit selected for the implementation of the kalman filter is the ARM microcontroller - TM4C123GXL. The Tiva™ C Series TM4C123G LaunchPad Evaluation Board (EK-TM4C123GXL) is a low-cost evaluation platform for ARM® Cortex™-M4F-based microcontrollers. The Tiva C Series LaunchPad design highlights the TM4C123GH6PMI microcontroller USB 2.0 device interface, hibernation module, and motion control pulse-width modulator (MC PWM) module. The Tiva C Series LaunchPad also features programmable user buttons and an RGB LED for custom applications. The stackable headers of the Tiva C Series TM4C123G LaunchPad BoosterPack XL interface demonstrate how easy it is to expand the functionality of the Tiva C Series LaunchPad when interfacing to other peripherals on many existing BoosterPack add-on boards as well as future products. Figure 1-1 shows a photo of the Tiva C Series LaunchPad.

4.2 FEATURES OF THE MICROCONTROLLER

• Tiva TM4C123GH6PMI microcontroller

• Motion control PWM

• USB micro-A and micro-B connector for USB device, host, and on-the-go (OTG) connectivity

• RGB user LED

• Two user switches (application/wake)

• Available I/O brought out to headers on a 0.1-in (2.54-mm) grid

• On-board ICDI

• Switch-selectable power sources:

– ICDI

– USB device

• Reset switch

• Preloaded RGB quickstart application

• Supported by TivaWare for C Series software including the USB library and the peripheral driver library

• Tiva C Series TM4C123G LaunchPad BoosterPack XL Interface, which features stackable headers toexpand the capabilities of the Tiva C Series LaunchPad development platform

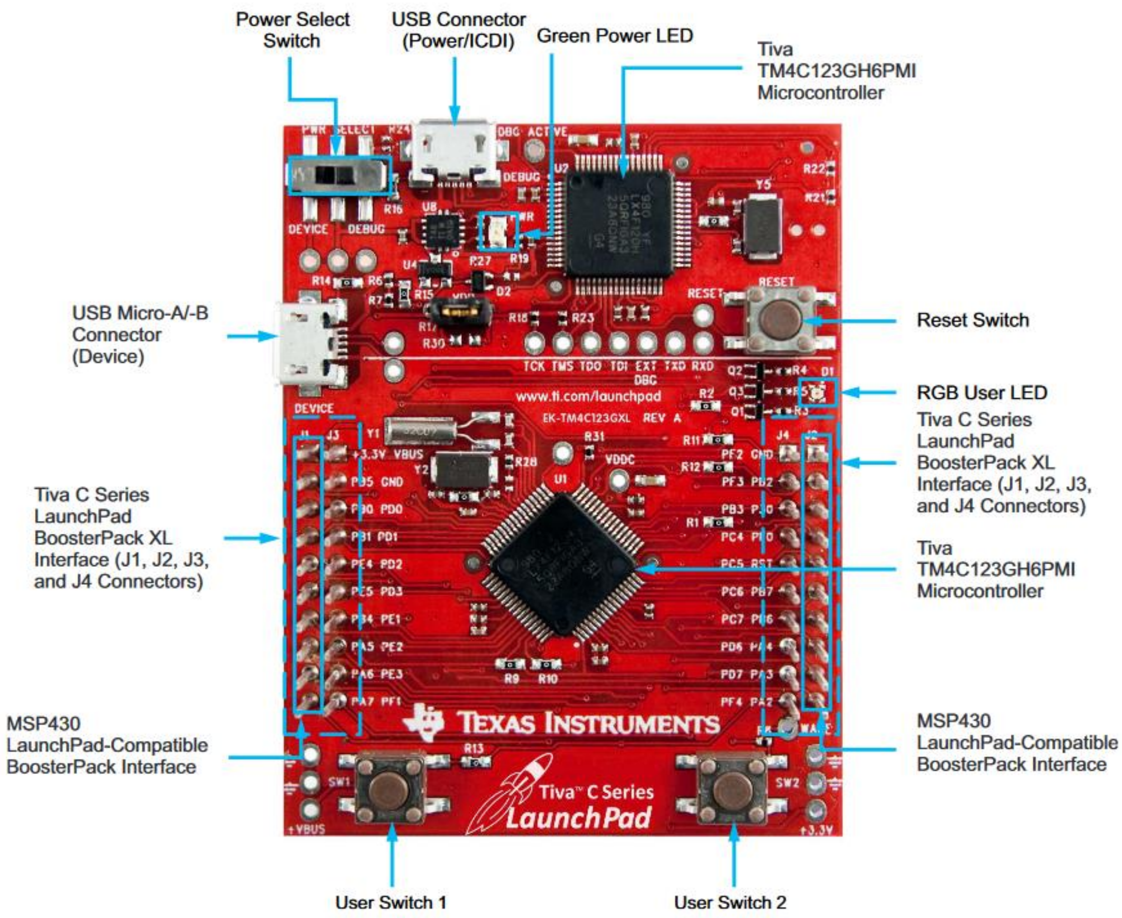

FIG 4.1 : Tiva C Series TM4C123G LaunchPad Evaluation Board

4.3 SIGNAL RECOVERY USING A CORE PROCESSOR

Implementation of Recovery Algorithms in C Implementation of the compressive ranging setup on the ARM is mainly divided into two parts; generating multi-gigabit patterns and processing the range using recovery algorithms. The first part of the implementation was demonstrated in Chapter 4, where ranging results were successfully obtained from patterns and a trigger generated by the ARM. The second part of the implementation was demonstrated using a soft-core NiosII processor on an Altera Stratix V ARM board. The board is a transceiver signal integrity development kit, which has plenty of hardwired transceiver outputs to SMA ports. Altera provides IP for the soft-core processor and the transceivers that were used for the setup. ARMs gave us the option of doing computation in hardware as well as in software. Processing done in hardware is generally harder to implement since real-time troubleshooting is difficult and the code is not very modular since hardware redesign takes time. Although hardware computations are much faster than software calculations, instantaneous calculations are not a requirement for the processing part of the system.

The decision was made to go with software computation and the code for the processing was written in C, making the implementation of certain numerical algorithms relatively simple compared to hardware languages such as Verilog or VHDL. ARM cortex 4 is a 32-bit RISC processor, which has the option to do multiplication in hardware, making some of the computations faster. It provides three options with increasing computational power. The most computationally efficient processor was used, since most of the ARM capacity was used for the processing. However, less memory was provided as on-chip memory with the ARM, so the code had to be compact and memory efficient. The memory needed to accommodate both the patterns to be used as well as the essential matrices required in the recovery algorithms. To provide memory-efficient computation, the C code was coded from scratch, without importing any libraries for complex linear algebra calculations. The code was made modular towards the processing side with all the variables and matrix dimensions kept variable and entered at the beginning of the processing step, since it is generally inconvenient to write a program that can deal with a matrix of arbitrary dimensions.

In C, the use of pointers makes this process relatively easy and all the matrices in this demonstration were described using double pointers to keep the code consistent as well as modular. The algorithm implemented in this setup was the CoSaMP algorithm as described in chapter 3, although the program was designed such that any similar algorithms, such as OMP, STOMP and ROMP, could be implemented easily with minimal restructuring of the code. The biggest challenge of the program was the implementation of the MoorePenrose pseudoinverse of a matrix that is central to solving an underdetermined linear system.

The common method used to implement the function is done using singular value decomposition (SVD), which factorizes the target matrix into three matrices of the form M = UΣV*, where V* is the transpose of the matrix V. U and V are transposed and multiplied in a certain order with the inverse matrix Σ+ to obtain the pseudoinverse. Σ is a diagonal matrix and the inverse is obtained trivially by taking a reciprocal of the diagonal values. The Moore-Penrose pseudoinverse of the matrix M is given by M+ = VΣ+U*. Functions for multiplying and transposing these matrices also were required as were functions for allocating and de-allocating memory used for these matrices. To keep the code memory efficient, memory allocated for matrices was freed as soon as the computation on that matrix was finished. Some of the computations also involved matrix and vector multiplication. The program was divided into three major code files with one containing the main body, one containing the function for the pseudoinverse and one containing all the functions for matrix calculations. The file containing the matrix operations contained the functions for matrix and vector multiplications of all forms and sizes, along with the functions for matrix and vector memory allocations and sorting of vectors and matrices.

The code containing the pseudoinverse function was implemented separately, which included the complex function for SVD. The code for the SVD was based on the algorithm and code provided in the book Numerical Recipes in C by Teukolsky, Vetterling and Flannery.39 The main file of the program was the implementation of the CoSaMP algorithm which asked for the known matrix from the user, the measurements and the dimensions of the known matrix. The three major files along with the board support package for the ARM were the software part of the implementation of the recovery algorithm. The hardware for the processor was designed and implemented with 3MB of memory using the IP provided by ARM programming software provided with the ARM board. The processor communicates with the computer through a USB-Blaster cable. The Energy Integrated Development Environment (IDE) package was used to create the board support package for the ARM and compile the C code. The program went through several iterations of debugging and troubleshooting before successfully processing the accurate range.

4.4 SPECIFICATIONS

Parameter	Value
Board supply voltage	4.75 V_{DC} to 5.25 V_{DC} from one of the following sources: • Debugger (ICDI) USB Micro-B cable (connected to a PC) • USB Device Micro-B cable (connected to a PC)
Dimensions	2.0 in x 2.25 in x 0.425 in (5.0 cm x 5.715 cm x 10.795 mm) (L x W x H)
Break-out power output	• 3.3 V_{DC} (300 mA max) • 5.0 V_{DC} (depends on 3.3 V_{DC} usage, 23 mA to 323 mA)
RoHS status	Compliant

Fig 4.2 : Specifications of TM4C123GXL

4.5 BASIC BLOCK DIAGRAM

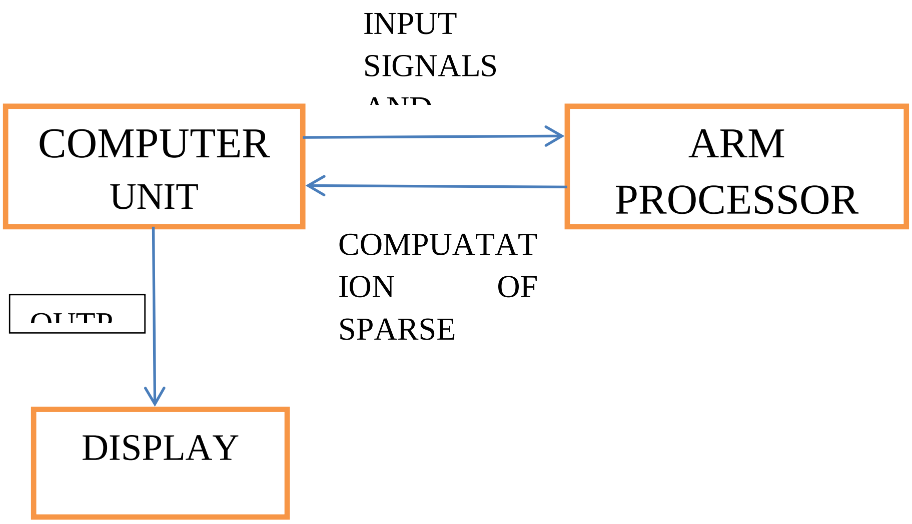

Fig 4.3 Basic Block Diagram for Hardware implementation

CHAPTER 5

RESULTS

CHAPTER 5

RESULTS

The algorithm was tested on various platforms.First the code was simulated on MATLAB where an audio signal was given as input and the following outputs where obtained.The input signal had 9000 samples which was compressed and reconstructed back to the original signal.

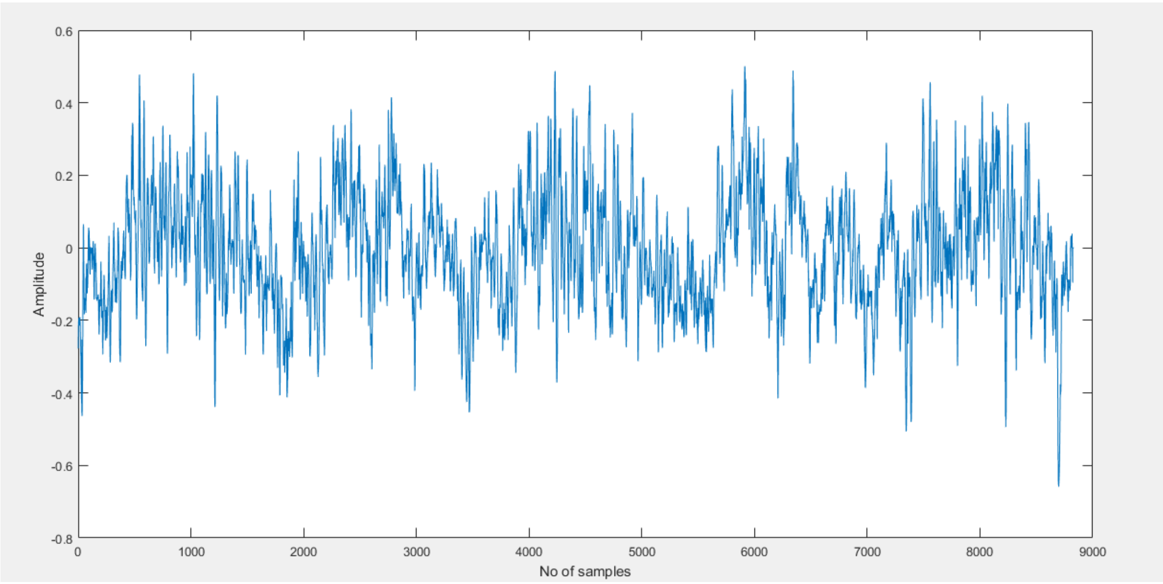

Fig 5.1 Original Signal of 9000 samples

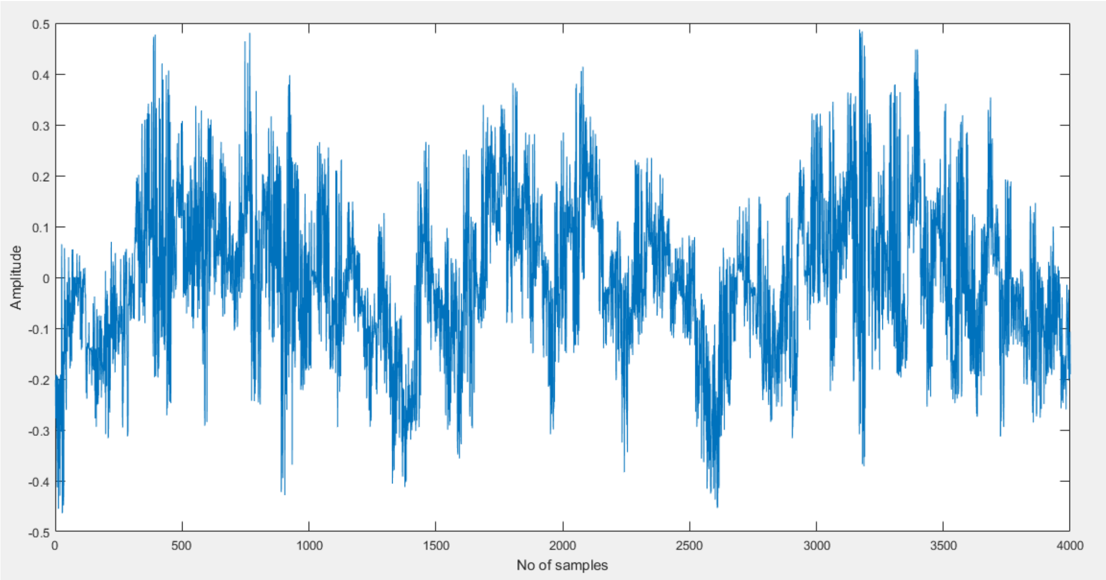

Fig 5.2 Compressed Matrix to 4000 samples

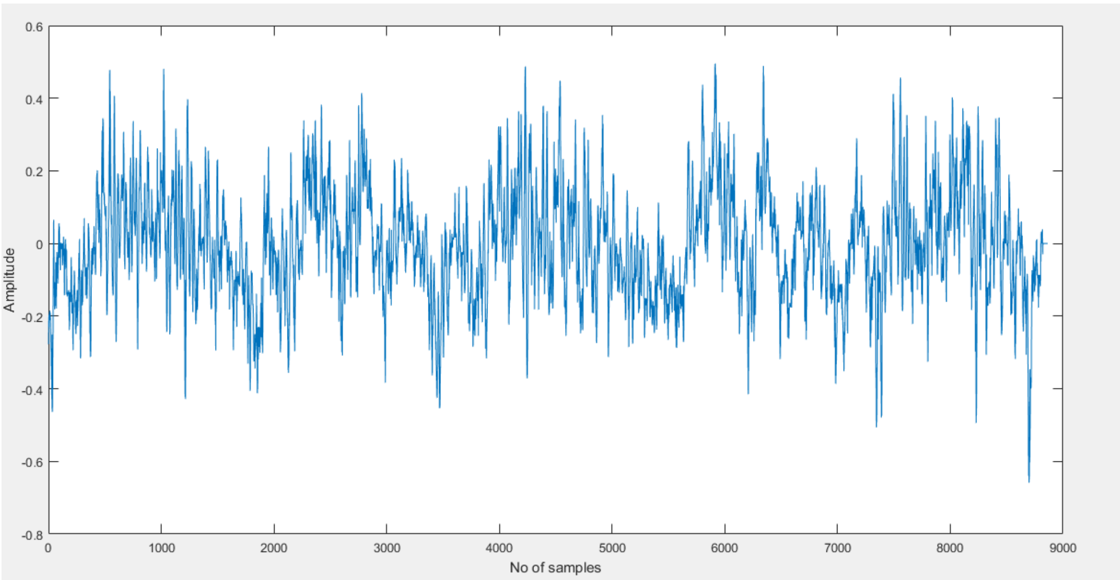

Fig 5.3 Reconstructed Signal

Simulation Results on the system

```
0.000000 0.190900 0.289500 0.218300 0.000000 -0.244700 -0.363900 -0.269500 0.000
000
```

Fig 5.4: Input vector

```
sparse matrix is :
1 3 2 0 0 2 1 0 1
0 2 3 0 2 1 0 1 1
-2 2 4 4 0 -2 2 2 1
3 0 3 3 1 1 0 0 0
4 2 -3 4 2 2 1 0 0
5 -2 0 0 1 0 0 0 1
6 -2 2 1 1 1 0 0 1
7 2 2 2 -2 1 0 1 2
8 -3 2 1 1 1 1 1 -2
```

Fig 5.5: Sparse matrix

```
The Inverse is :
0.086197 -0.0255572 -0.0478792 0.235262 -0.0493889 0.480805 -0.500395 0.0623652
0.0589504
0.202013 0.0381021 -0.0805176 0.491733 -0.0740474 0.796549 -1.06973 0.0898634 0.
0330697
0.109921 -0.0050683 0.00316319 0.244968 -0.136377 0.154421 -0.179403 -0.0213875
0.0201294
-0.156506 -0.0624371 0.0602444 0.017793 0.107189 -0.331704 0.321819 0.0131201 -0
.0711718
0.0306973 0.181057 -0.0278936 0.248922 0.0207764 0.547376 -0.463444 -0.129583 0.
00431344
-0.149533 0.0981308 -0.0186916 -0.742991 0.214953 -1.45794 1.53738 -0.0327103 -0
.0280374
0.444644 -0.297807 0.214234 -0.272646 0.0363048 0.0948958 0.167685 -0.221244 0.0
905823
-0.587994 0.455823 0.11977 -1.03145 0.222646 -1.33487 1.24123 0.178828 0.125809

-0.0576564 0.0229331 0.106254 -0.441768 0.0780733 -0.358303 0.825953 -0.00251618
 -0.232926
```

Fig 5.6: Pseudo Inverse

```
Result matrix is
0.000000 0.706825 1.000000 0.707951 0.001593 -0.705698 -0.999997 -0.709075 -0.0
3186
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.000000 0.00000

0.000000 -1.413650 -1.999999 -1.415902 -0.003186 1.411395 1.999994 1.418150 0.0
6371
0.000000 2.120475 2.999999 2.123853 0.004778 -2.117093 -2.999991 -2.127225 -0.0
9557
0.000000 2.827301 3.999999 2.831804 0.006371 -2.822791 -3.999989 -2.836300 -0.0
2742
0.000000 3.534126 4.999999 3.539754 0.007964 -3.528488 -4.999986 -3.545375 -0.0
5928
0.000000 4.240951 5.999998 4.247705 0.009557 -4.234186 -5.999983 -4.254449 -0.0
9113
0.000000 4.947776 6.999998 4.955656 0.011150 -4.939884 -6.999980 -4.963524 -0.0
2299
```

Figure 5.7: Measurement matrix

```
reconstructed matrix is
0.050000 0.224445 0.314546 0.249483 0.050000 -0.173608 -0.282533 -0.196270 0.050
000
```

Fig 5.8: Reconstructed signal

The input vector was passed through a compressed sensing matrix and the number of samples send where reduced.The value of the reconstructed signal was under acceptable limits.Further we sampled two audio signals in Matlab and generated a vector for further processing.

The Hardware takes 2 input audio vectors and compresses it according to the sparse matrix. The OMP algorithm is used for reconstruction of the received matrix.

Following results were obtained.

```
The pseudoinverse is :
0.04  -0.18  0.05  0.04  0.02  0.02  0.01  0.06  0.03
0.22  -0.31  0.05  0.04  -0.04  -0.08  -0.19  0.18  0.03
0.10  -0.33  0.15  -0.02  -0.05  -0.57  0.52  0.05  -0.02
-0.47  0.54  -0.06  0.32  0.18  0.33  -0.01  -0.29  -0.12
0.09  0.55  -0.26  0.18  -0.22  1.11  -0.96  -0.14  0.12
0.23  0.30  -0.28  -0.31  -0.10  -0.21  -0.05  0.13  0.12
0.93  -3.46  1.50  -1.34  0.76  -4.74  3.95  0.69  -0.22
-0.35  2.77  -1.12  0.33  -0.74  3.62  -3.64  -0.08  0.55
0.15  -0.25  0.14  -0.32  0.04  -0.34  0.51  0.18  -0.21
Result matrix is
0.00  0.19  0.29  0.22  0.00  -0.24  -0.36
0.00  0.00  0.00  0.00  0.00  0.00  0.00
0.00  -0.38  -0.58  -0.44  0.00  0.49  0.73
0.00  0.57  0.87  0.65  0.00  -0.73  -1.09
0.00  0.76  1.16  0.87  0.00  -0.98  -1.46
0.00  0.95  1.45  1.09  0.00  -1.22  -1.82
0.00  1.15  1.74  1.31  0.00  -1.47  -2.18
reconstructed matrix is
0.00  0.18  0.28  0.21  0.00  -0.23  -0.35  -0.26  0.00
```

```
The pseudoinverse is :
0.04  -0.18  0.05  0.04  0.02  0.02  0.01  0.06  0.03
0.22  -0.31  0.05  0.04  -0.04  -0.08  -0.19  0.18  0.03
0.10  -0.33  0.15  -0.02  -0.05  -0.57  0.52  0.05  -0.02
-0.47  0.54  -0.06  0.32  0.18  0.33  -0.01  -0.29  -0.12
0.09  0.55  -0.26  0.18  -0.22  1.11  -0.96  -0.14  0.12
0.23  0.30  -0.28  -0.31  -0.10  -0.21  -0.05  0.13  0.12
0.93  -3.46  1.50  -1.34  0.76  -4.74  3.95  0.69  -0.22
-0.35  2.77  -1.12  0.33  -0.74  3.62  -3.64  -0.08  0.55
0.15  -0.25  0.14  -0.32  0.04  -0.34  0.51  0.18  -0.21
1
Result matrix is
0.00  0.19  0.29  0.22  0.00  -0.24  -0.36
0.00  0.00  0.00  0.00  0.00  0.00  0.00
0.00  -0.38  -0.58  -0.44  0.00  0.49  0.73
0.00  0.57  0.87  0.65  0.00  -0.73  -1.09
0.00  0.76  1.16  0.87  0.00  -0.98  -1.46
0.00  0.95  1.45  1.09  0.00  -1.22  -1.82
0.00  1.15  1.74  1.31  0.00  -1.47  -2.18
reconstructed matrix is
0.00  0.18  0.28  0.21  0.00  -0.23  -0.35  -0.26  0.00
0.18
Signal 1 received
```

Fig 5.9 Addition of envelope detector to recognize the signals

For the percentage of error, the four signals were compressed and reconstructed by the respective algorithm. The percentage of error was calculated by averaging the first 40 samples of both the original and reconstructed signals.

Comparison of the error obtained on the system and Hardware for the initial vrctor vailues:

Table 5.1

Comparison of the error

Simulation Error(%)	Hardware Error(%)
8.05	5.29
7.85	4.95
7.79	5.78
6.99	5.15

CHAPTER 6

CONCLUSION

CHAPTER 6

CONCLUSION

Compressive sensing can be efficiently used in speech processing technique as it will reduce the storage capacity and increase the data rate of the signal. CS technique can be used as a substitute for the traditional Nyquist Shannon Sampling as it uses less than half of the measurements to reconstruct the signal. Experiments were carried out for various threshold windows. For the threshold window of "- 0.02 to +0.02", better reconstruction of audio signal is possible. Various performance parameters are measured which describe exact reconstruction of the signal. For proper reconstruction Orthogonal Matching Pursuit algorithm is used. The project classifies between two signals by envelope detector at the receiver end. For future work investigating the performance of a multi-sensor CS system for audio signals, and finding a scheme based on recovery algorithms to provide improved performance over standard algorithms for a wide range of audio signals, and also provide the possibility of location detection.

The major use of this project will be in the field of IOT in which a large amount of data needs to be transmitted over the channel. Applying Compressive Sensing Techniques to these process can greatly increase the efficiency by reducing the amount of data required to be transmitted. Another field in which compressive sensing can be used is the Data fusion of signals from various sensors.It can be used to classify the various signals fused together.

REFERENCES

[1] Compressed Sensing Reconstruction of an Audio signal using OMP Shwetha A. Gangannawar and Saroja V. Siddmal ,International Journal of Advanced Computer Research ISSN (Print): 2249-7277 ISSN (Online): 2277-7970 Volume-5 Issue-18 March-2015

[2] David L. Donoho, “Compressed Sensing,” IEEE Trans. Information Theory, vol. 52, no. 4, pp.1289-1306, Apr. 2006.

[3] Emmanuel J. Candes and Michel B.Wakin “Compressive Sampling” IEEE signal processing Magazine 21 March 2008.

[4] Mohammed M. Abo-Zahhad, Aziza I. Hussein ,Compressive Sensing Algorithms for Signal Processing Applications, Int. J. Communications, Network and System Sciences, 2015

[5] Emmanuel Candes and Justin Romberg ,Sparsity and Incoherence in Compressive Sampling, , Submitted manuscript, June 2006.

[6] M. Davenport, P. Boufounos, M. Wakin, and R. Baraniuk. “Signal processing with compressive measurements”. IEEE J. Select. Top. Signal Processing, 4(2):4458211; 460, 2010.

[7] A. Septimus and R. Steinberg, “Compressive sampling hardware reconstruction,” in Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on, 2010, pp. 3316–3319.

[8] Richard Baraniuk, Mark Davenport, Ronald DeVore, Michael Wakin “A Simple Proof of the Restricted Isometric Property for Random Matrices”. Published online: 15 January 2008 © Springer Science+Business Media, LLC 2008.