APPROVAL SHEET

Title of Thesis: Detection Performance and Computational Complexity of Radar Compressive Sensing for Noisy Signals

Name of Candidate: Asmita Korde
Master of Science, 2013

Thesis and Abstract Approved: Tinoosh Mohsenin

Dr. Tinoosh Mohsenin, Department of
Computer Science and
Electrical Engineering

Date Approved: 7/26/2013

Curriculum Vitae

Name: Asmita Korde.

Permanent Address: 3311 Hollow Court, Ellicott City, MD 21043.

Degree and date to be conferred: Master of Science, August, 2013.

Date of Birth: October 18, 1989.

Place of Birth: Gujarat, India.

Secondary Education: Mount Hebron High School, Ellicott City, MD.

Collegiate institutions attended:

University of Maryland Baltimore County, M.S Electrical Engineering, 2013.

University of Maryland Baltimore County, B.S, 2011.
Major: Computer Engineering.
Minor: Mathematics.

Professional publications:

Korde, A., Bradley, D., Mohsenin, T. (2013, May). Detection performance of radar compressive sensing in noisy environments. In SPIE Defense, Security, and Sensing (pp. 87170L-87170L). International Society for Optics and Photonics.

Professional positions held:

Graduate Co-Op, NASA Goddard Space Flight Center. (January 2011 - Present).

Student Intern, NASA Goddard Space Flight Center. (June 2010 - August 2010).

Undergraduate Research Assistant, Mathematics Department, UMBC. (September 2008- January 2009, June 2009- August 2009).

Testing Intern, LandAmerica. (June 2008 - August 2008).

ABSTRACT

Title of dissertation: Detection Performance and Computational Complexity of Radar Compressive Sensing for Noisy Signals

Asmita Korde, Master of Science, 2013

Dissertation directed by: Dr. Tinoosh Mohsenin
Department of Computer Science and Electrical Engineering

In recent years, compressive sensing has received a lot of attention due to its ability to reduce the sampling bandwidth, yet reproduce a good reconstructed signal back. Compressive sensing is a new theory of sampling which allows the reconstruction of a sparse signal by sampling at a much lower rate than the Nyquist rate. This concept can be applied to several imaging and detection techniques. In this thesis, we explore the use of compressive sensing for radar applications. By using this technique in radar, the use of matched filter can be eliminated and high rate sampling can be replaced with low rate sampling. We analyze compressive sensing in the context of radar by applying varying factors such as noise and different measurement matrices. Different reconstruction algorithms are compared by generating Receiver Operating Characteristic (ROC) curves to determine their detection performance, which in turn are also compared against a traditional radar system. Computational complexity and MATLAB run time are also measured for the different algorithms. We also propose an algorithm called Simplified Orthogonal Matching Pursuit, which works

well in noisy environments and has a very low computational complexity.

Detection Performance and Computational Complexity of Radar Compressive Sensing for Noisy Signals

by

Asmita Korde

Thesis submitted to the Faculty of the Graduate School
of the University of Maryland in partial fulfillment
of the requirements for the degree of
Master of Science
2013

© Copyright Asmita Korde 2013

DEDICATION

I would like to dedicate this thesis to my family. My parents and sister have been very supporting and encouraging throughout my undergraduate and graduate career. They have helped me stay focused and determined. Without their unconditional love and support I would not have been able to come this far.

ACKNOWLEDGMENTS

I owe gratitude to all those who have made this thesis possible. Firstly, I would like to thank my advisor, Dr. Tinoosh Mohsenin, for her guidance and support. Her experience and knowledge has helped me in my research work and I would like to thank her for giving me this opportunity to work with her. I would also like to thank my committee members, Dr. Joel Morris, Dr. Mohamed Younis, and Dr. Tulay Adali for all their feedback and guidance. Their feedback has been very valuable to me and my research work.

I would also like to thank Dr. Arian Maleki from Columbia University for all his guidance on generating Receiver Operating Characteristic (ROC) curves and helping me understand the algorithms. Dr. LaBerge has also given me very valuable feedback, so I would like to thank him as well.

My bosses/co-workers at NASA Goddard Space Flight Center, Damon Bradley and Dr. Mark Wong, have also helped me a lot throughout my thesis work. Thank you to both of you for all your feedback and help.

Finally, I would like to thank all my family and friends for their support. Without their encouragement I would not have been able to accomplish this.

TABLE OF CONTENTS

List of Tables

List of Figures

Chapter 1

INTRODUCTION

1.1 Motivation

In today's digital age, high-resolution sensing systems are becoming more and more popular. Attaining high-resolution requires sampling at high Nyquist rate leading to huge amounts of data. Many times, sampling at such high rates can be too costly or sometimes, even impossible. Hence, acquisition and processing of signals in such applications is an ongoing challenge. In the recent years, compressive sensing (CS) has emerged as a new technology to reconstruct sparse signals.

In traditional methods, a signal has to be sampled at the Nyquist rate in order to reconstruct the signal without aliasing. However, if a signal is sparse in some domain, then obtaining Nyquist rate samples are wasteful. An alternative to Nyquist rate sampling would be to under-sample. Down-sampling or under-sampling a signal can lead to a loss of information especially if the signal is sparse. A new approach to this type of problem is compressive sensing. Instead of obtaining a few samples of the signal, a few linear combinations are

sampled. Hence, one measurement consists of a collective sum [1] [2] [3]. Using fewer measurements, a signal can be reconstructed almost perfectly under certain conditions.

1.2 Contributions

Most research in Compressive sensing radar so far has mainly focused on reconstruction of the target image profiles via CS techniques. In this thesis, we show the detection performance of radar compressive sensing in different noise scenarios. The comparison of detection performance of different reconstruction algorithms is a novel comparison essential for determining which reconstruction algorithm needs to be used in a certain circumstance. In addition, we also propose a simplified version of Orthogonal Matching Pursuit and show that this algorithm works comparatively very well in noisy environments. This algorithm is also very low cost and can run 39 times faster than L1Ls in MATLAB. Detection performance of CS algorithms are also compared to that of an ideal traditional radar system. Moreover, computational complexity and MATLAB run time for the different algorithms are also calculated. A small study on the incoherence of the measurement matrix is also done and is used to prove the theoretical requirements of having an incoherent measurement matrix.

1.3 Organization of Thesis

The goal of this thesis is to compare and analyze the detection performance and computational complexities of different CS based reconstruction algorithms in different noise

models. To begin with, we first provide background on some CS techniques as well as its application to radar. In the following chapter, we provide a brief explanation on the reconstruction algorithms we used. These algorithms include L1 regularized least squares, Iterative Hard Thresholding (IHT), Orthogonal Matching Pursuit (OMP), Compressive Sampling Matching Pursuit (CoSaMP), and our proposed algorithm, Simplified Orthogonal Matching Pursuit. In Chapter 4 we compare and analyze the detection performance of the reconstruction algorithms in different noise models. Then in Chapter 5 we perform analysis and comparison of computational complexities of the reconstruction algorithms. Finally, in Chapter 6, we summarize the results, conclude the thesis and provide possible research paths for future work.

Chapter 2

OVERVIEW OF RADAR COMPRESSIVE SENSING

2.1 Traditional Radar

A simple radar works by transmitting a signal to a target profile and then receiving a signal reflected back from the target area. Usually a radar pulse, which is a short burst of radio wave, is transmitted. Depending on the reflectivity of the objects present in the scene, a fraction of the wave will be reflected back; this is the received signal. Hence, if an object is present, at least some part of the transmitted wave will be reflected back when it hits that object [4]. A pictorial representation is shown in figure 2.1. Radar can be used for various purposes such as target presence detection, target speed detection, image mapping, etc. In this thesis, we will use radar for target presence detection. A block diagram of traditional radar signal processing is shown in figure 2.2. In all our experiments, we used a raster scan to simulate a traditional radar. This would correspond to an ideal radar system, where the received samples would directly correspond to the sampled target scene.

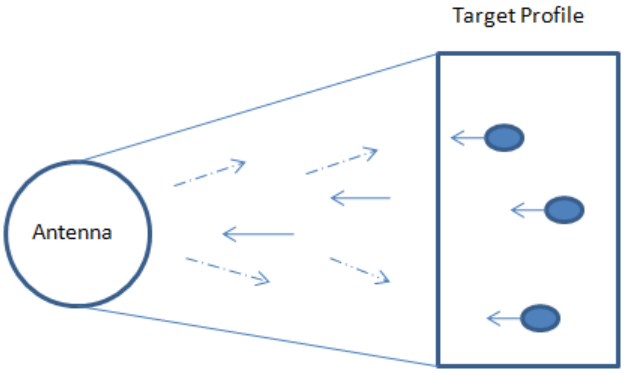

FIG. 2.1. Transmitted signal is transmitted from the antenna and the received signal is part of the transmitted signal that gets reflected from the targets and hits the antenna. This idealized system does not have any noise added.

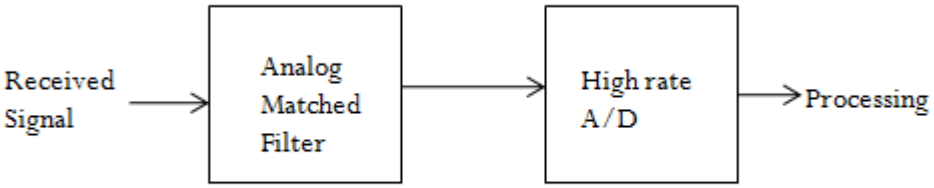

FIG. 2.2. Traditional Radar

2.2 Compressive Sensing (CS) Theory

Before describing the mathematics behind CS theory, we will define some of the concepts required for CS.

2.2.1 Sparsity

A signal, x, is sparse when it can be exactly represented by a linear combination of only a few elements [3]. In simpler terms, it means the signal, x, has at most k nonzero elements.

A signal that is not sparse in its original domain can be sparsified in another transform domain. For example, a sine wave is not sparse in the time domain, but it is sparse in the Fourier domain. In real-world applications, signals are rarely truly sparse; usually they are compressible or weakly sparse. A compressible signal has coefficients, $|c|_i$, whose rate of decay is very high [5]. In other words, we can neglect a small fraction of the coefficients without losing much information. Hence, the signal, x, can be well approximated as a k-sparse signal. In our simulations, for simplicity in modeling, we use the signal, x, as k-sparse rather than *approximately* k-sparse.

2.2.2 Measurement (Sensing) Matrix

For a signal, x, of length, N, and $M << N$, a measurements matrix, ϕ, is an MxN matrix that maps $\mathbb{R}^N$ into $\mathbb{R}^M$ [3]. Given $x \in \mathbb{R}^N$, we want to acquire M linear measurements. This relationship using ϕ can be represented as

$$y = \phi x \tag{2.1}$$

where ϕ needs to be chosen such that the restricted isometry property (RIP) of order

$2k$ is satisfied, where k represents the sparsity of the signal, x. The ϕ satisfies RIP bound for an order of k if

$$(1 - \delta_k)\|x\|_2^2 \leq \|\phi x\|_2^2 \leq (1 + \delta_k)\|x\|_2^2 \tag{2.2}$$

where $\delta_k \in (0, 1)$ [3] [6] This bound establishes successful recovery of a sparse signal when the measurements are noiseless. The bounds need to be stricter for noisy measurements.

In general, since random matrices are incoherent with any basis, this bound is satisfied and hence, they can be used as good measurement matrices. This idea is further explored in Section 2.6.

2.2.3 CS Mathematics

Let us assume x to be a k-sparse signal of length N. Let ϕ be the measurement matrix multiplied with the original signal, x. The ϕ must be incoherent with the basis of the sparse signal, x. If x is not sparse in its original bases, it can be transformed to another domain in which the signal is sparse. Then the measurement matrix has to be uncorrelated with the signal in the transformed domain [1]. The size of ϕ is MxN, where $M<<N$ and represents the number of measurements. y is a M length vector containing the measurements obtained by the multiplication of ϕ and x. Equation 2.2 can be modified for a signal that is not sparse in its original sampling domain and that needs to be transformed to a sparse basis, Ψ, to induce sparsity, i.e.,

$$y = \phi\Psi x = \Phi x \tag{2.3}$$

In our simulations, we assume x to be sparse in the spatial domain itself. Thus, no transformation takes place and we can directly use equation 2.2 Given this setting, y and ϕ are known. The problem lies in inversely solving for a good estimate of x. When noise is added, either to the original signal, x, or the observations y, this task of solving for x becomes more difficult. Several reconstruction algorithms can be used, and later in this thesis, these reconstruction algorithms are compared against each other for the different noise models.

2.3 Radar CS

In Radar compressive sensing, the transmitted signal must be incoherent with some basis. We can create projections of this incoherent signal on to a sparse target reflectivity profile. By generating only a few (less than the Nyquist rate) projections, we can reconstruct the target reflectivity profile. By using CS technique in radar, the use of matched filter can be eliminated and sampling can be done at a low rate [7] [8] [9]. Although data acquisition is a much efficient process in CS based radar, the reconstruction of the signal can be computationally complex. Hence, the key is to use an algorithm which balances between good detection performance and low computational cost. The basic block diagram of a CS radar system is shown in figure 2.3. Radar based CS can be very beneficial in satellite

or space communication, where data transmission is limited due to bandwidth. By using radar CS, we can reduce the amount of data transmitted and get a good reconstruction of the signal back. Since reconstruction can be done on the ground, power can be saved for on board communication.

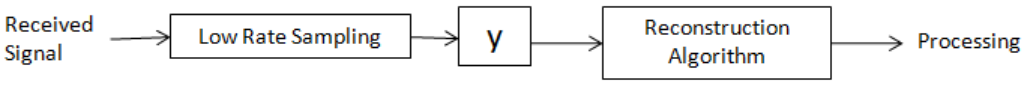

FIG. 2.3. CS Radar

We will describe the transition from traditional radar to compressive sensing radar to better understand how compressive sensing can be applied to radar. First let us consider a traditional radar 1-D signal. Let $u(r)$ describe a target model. If $tx(t)$ is the transmitted signal, then the received signal, $rx(t)$, will be a linear convolution of the target model, $u(r)$, and the transmitted signal, $tx(t)$ as shown below.

$$rx(t) = A \int tx(t-\tau)u(\tau)d\tau \tag{2.4}$$

where A is the attenuation and $t = \frac{2r}{c}$. For simplicity, we assume $A = 1$.

For compressive sensing, $u(\tau)$ needs to be sparse in some basis. We can generate from N Nyquist rate samples from the target reflectivity profile. Let us call this sampled

reflectivity profile, $x(n)$, where $n = 1, 2, ..., N$. The radar transmits an N length signal, p(n) for $n = 1, 2, ...N$, via $tx(t)$. We can assume p(n) consists of bernoulli random values. The received signal given by equation (2.5) can be sampled every $D\Delta$ seconds instead of ever Δ seconds, where $D = \lfloor N/M \rfloor$ and $M < N$. This gives us M samples for $m = 1, 2, ..., M$. Hence, equation 2.5 can now be seen as:

$$y(m) = rx(t) \mid_{t=mD\Delta} = \int_0^{N\Delta} tx(mD\Delta - \tau)u(\tau)d\tau \tag{2.5}$$

$$y(m) = \Sigma_{n=1}^{N} p(mD - n)x(n) \tag{2.6}$$

Let $p(mD - n)$ for $m = 1, 2....M$. Now we have an equation which resembles the compressive sensing problem seen in equation 2.2. $p(mD - n)$ corresponds to a random filter [7]. However, $p(mD - n)$ can be replaced with any random or incoherent vector to obtain the same results; having a random filter is not necessary.

2.4 Notations

The ϕ is the matrix containing rows of the transmitted signal transmitted at a time. If the random filter vector, $p(mD - n)$, is one transmitted signal of length N,

$$\phi = \begin{bmatrix} p(D-n) \\ p(2D-n) \\ ..... \\ p(MD-n) \end{bmatrix}$$

Here ϕ does not have to always correspond to a random filter. For example, if p(n) is an N length signal of Bernoulli random numbers, then ϕ will contain M rows of N length Bernoulli sequences, where each row is a signal transmitted via $tx(t) = p(\lceil t/D\Delta \rceil)$.

x is the target scene or the target reflectivity profile. x is sampled from a continuous target profile, so that x has N Nyquist-rate samples. We are trying to reconstruct this signal.

y is the low-rate sampled received signal. However, since we sample it at a rate lower than the Nyquist rate, we cannot directly achieve samples from y which correspond to x. A reconstruction algorithm needs to be applied to solve for this under determined system of equations.

2.5 Receiver Operating Characteristic (ROC) Curves

ROCs are used to determine the effectiveness of a radar detector. They are generated by plotting probability of detection versus probability of false alarm. Once CS techniques are performed on the original signal, we can derive the estimate by a reconstruction algorithm. For each algorithm a different parameter is changed in order to change the sparsity of the signal. The estimates are compared to the original target profile and probability of

detection and false are obtained [10] [11].

Detection:

$\widehat{x(i)} \neq 0$ and $x(i) \neq 0$

where $\widehat{x}$ is the estimated signal

and i is the i^{th} element of the signal

Probability of Detection:

$$pD = \frac{number\ of\ Detected}{number\ of\ x(i) \neq 0}$$

False Alarm:

$\widehat{x(i)} \neq 0$ and $x(i) = 0$

Probability of False Alarm:

$$pFA = \frac{number\ of\ false\ alarm}{number\ of\ N - x(i) \neq 0}$$

$x(i) \neq 0$ can also be referred to as targets

Monte Carlo simulations are performed, and the average values of probability of false alarm and probability of detection are plotted in the ROC. In order to determine the ROC, we took 1000 Monte Carlo simulations for all our simulations.

2.6 Coherence Between Measurement Matrix and Original Signal

The matrix projected onto the sparse target profile must be incoherent with the basis of the sparse signal. Coherence is the largest correlation between any two elements of the

two signals. Hence, we want measurement matrices which are the most uncorrelated with the sparse signal. Random matrices are universally incoherent, and hence, make a good basis for the transmitted signal, or the measurement matrix [1]. In addition, if the signal is sparse in time domain, Fourier basis can also be used for the measurement matrix. In this section,we compare normal random (Gaussian) matrix, Bernoulli random matrix and Fourier matrix. Gaussian matrix give normal random variables, Bernoulli gives ± 1, and Fourier matrix gives a Fourier transform of the indices of the size of the matrix. 1000 Monte Carlo simulations are performed to find the maximum coherence between the measurement matrix and a sparse signal of length 64 with 3 targets of the same amplitude. The average coherence of all three measurement matrices are listed in Table 2.1.

Table 2.1. Coherence of Measurement Matrix

Measurement Matrix	Average Coherence
Gaussian	0.6436
Bernoulli	0.6172
Fourier	0.6212

In order for CS to give good results, the measurement matrix as incoherent as possible with the sparse signal. From Table 2.1 we can see that all three of them are highly uncorrelated since their correlation values are close to 0. A maximum coherence for this specific signal, x, would be 3.

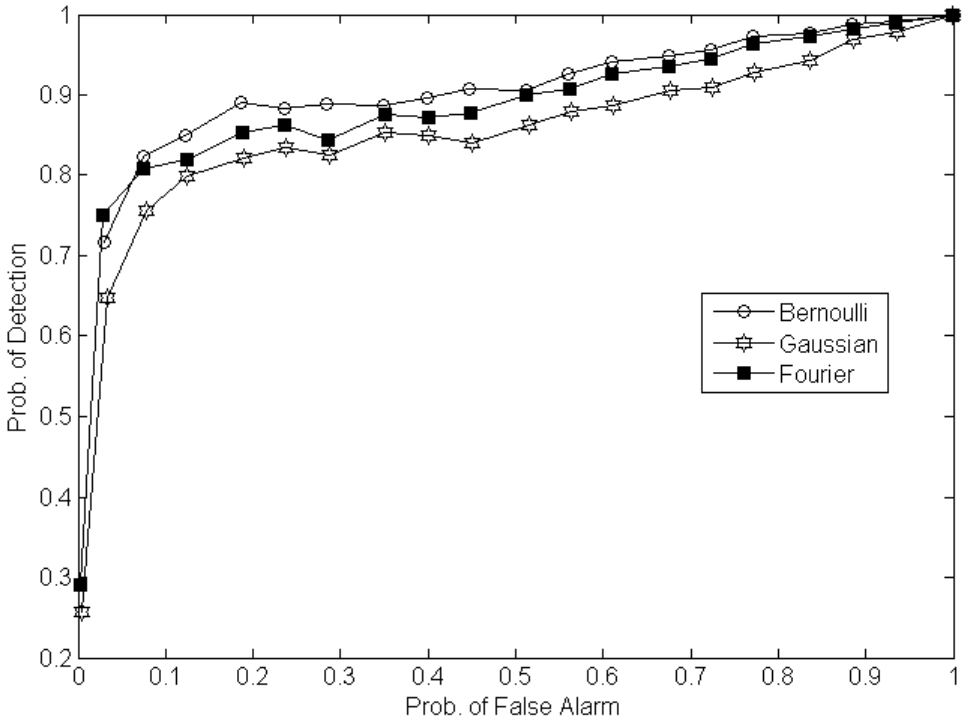

FIG. 2.4. Comparison of measurement matrices for a 64 length signal with 3 targets and AWGN of SNR 15dB.

In Fig. 2.4, we compare the performance of the three measurement matrices listed in Table 2.1. The 64 length signal with AWGN of SNR 15dB is reconstructed using OMP algorithm discussed in Chapter 3. From Table 2.1, it can be noted that Bernoulli measurement matrix is the most uncorrelated with the original target profile, followed by Fourier and lastly by Gaussian. From Fig. 2.4, we can see that the best ROC curve is given by Bernoulli measurement matrix, followed by Fourier. Since Gaussian measurement matrix was the most correlated out of the three measurement matrices, the ROC curve performed

slightly worse than the other two. Hence, we can see a direct relation between correlation and detection performance. In conjunction with Table 2.1, we can show that lower correlation of the measurement matrix basis and the sparse signal basis leads to a better reconstruction of the signal, and in turn, better detection in terms of radar.

Chapter 3

RECONSTRUCTION ALGORITHMS

Once low-rate samples are obtained from the receiver, reconstruction algorithm are used to inversely solve for the original signal. We used several greedy algorithms as well as a convex relaxation algorithm [12]. Some of the commonly used reconstruction algorithms are discussed below and are used to analyze detection performance.

3.1 Normalized Iterative Hard Thresholding(IHT)

Normalized IHT is an iterative algorithm that uses a nonlinear operator to reduce the L_0 norm at each iteration. The algorithm is described in detail in [13], which is based on the IHT algorithm described in [14]. This algorithm solves for:

$$x_t = \min_x \|y - \phi_t x\|_2^2 + \lambda |x|_0 \tag{3.1}$$

$x_0 = 0$

$g = \phi^T(y - \phi x^t)$

Γ^n : Support set of x^t

Step size: $\mu = \frac{g_{\Gamma^n}^T g_{\Gamma^n}}{g_{\Gamma^n}^T \phi_{\Gamma^n}^T \phi_{\Gamma^n} g_{\Gamma^n}}$

$x^{t+1} = H_k(x^t + \mu\phi^T(y - \phi x^t))$

Where H_k is the non linear operator, which sets all but the k max elements of $(x^t + \mu\phi^T(y - \phi x^T))$ to 0.

From here on in the paper, we will refer to Normalized IHT as simply IHT. We will use the package provided in [14] for the simulations.

3.2 Orthogonal Matching Pursuit (OMP)

OMP is a greedy algorithm- at each iteration, a column of ϕ is chosen which most strongly correlates with y. Then, the amount of contribution that the column provides to y is subtracted off. Further iterations are performed on this residual. After L iterations, the correct set of columns would be determined [15]. In this algorithm, we assume the desired sparsity to be k. The algorithm is shown below:

1. Initialize $R_0 = y$, $\phi_0 = \oslash$, $\Lambda_0 = \oslash$, $\Theta_0 = \oslash$ and $t = 0$

2. Find index $\lambda_t = max_{j=1...n}$ subject to $| < \phi_j R_{t-1} > |$

3. Update $\Lambda_t = \Lambda_{t-1} \cup \lambda_t$

4. Update $\Theta_t = [\Theta_{t-1}\ \phi_{\Lambda_t}]$

5. Solve the Least Squares Problem: $x_t = \min_x \|y - \Theta_t x\|_2^2$

6. Calculate new approximation: $\alpha_t = \Theta_t\ x_t$

7. Calculate new residual: $R_t = y - \alpha_t$

8. Increment t, and repeat from step 2 if $t < k$

Once all the iterations are completed we hope to find the correct sparse signal x.

3.3 Compressive Sampling Matching Pursuit (CoSaMP)

CoSaMP is also based on OMP. This algorithm is described in more detail in [16]. It uses the columns of ϕ that produce the maximum dot product with y. The algorithm is shown below:

1. Initialize $R_0 = y$, $x_0 = 0$, $\Theta_0 = \oslash$, and $t = 0$

2. Identify $2k$ largest components: $\Omega = \ supp_{2k\ j=1...n}$ subject to $| < \phi_j R > |$

3. Merge the supports: $\Theta_t = [\Theta_{t-1}\ \Omega]$

4. Solve the Least Squares Problem: $a_t = \min_a \|y - \Theta_t a\|_2^2$

5. Prune to retain k largest coefficients: $x_t = a_{ts}$

6. Calculate new residual: $R_t = y - \alpha_t$

7. Increment t, and repeat from step 2 if stopping criteria is not met, else $x = x_t$

The MATLAB code for this algorithm is a modified version of [17].

3.4 Simplified Orthogonal Matching Pursuit

We propose a simplified version of the OMP algorithm. Instead of choosing the columns of the new Θ by iterating through k times (where k corresponds to the sparsity), all the

columns are chosen in one iteration itself.

The algorithm is described below:

1. Initialize $R_0 = y$, $\phi_0 = \oslash$, $\Lambda_0 = \oslash$

2. Find index set $\lambda = supp_{k\ j=1...n}$ subject to $| < \phi_j y > |$

4. Let $\Theta = [\phi_{\lambda_{1,..,k}}]$

5. Solve the Least Squares Problem: $x = \min_x \|y - \Theta x\|_2^2$

6. Calculate new approximation: $\alpha = \Theta\ x$

7. Calculate new residual: $R = y - \alpha$

The main difference between OMP and Simplified OMP is that Θ is not updated each time, but is formed all at once.

3.5 L_1 Regularized Least Squares (L1Ls)

3.5.1 Sparse solutions from L_1 Regularized Least Squares (L1Ls)

Before we state the algorithm, we will give a brief description on why this algorithm gives sparse solutions. For all the previous algorithms, the sparsity value is a direct output of the number of nonzero outcomes we want from the algorithm. This is not exactly the case for L_1 Regularized Least Squares (L1Ls). We give an explanation on obtaining sparse solutions below.

A circle (or sphere in 3D) represents the L2 norm level set, and a square (or octahedron in 3D) represents the L1 norm level set. We want to minimize the L2 and L1 norm. We

should keep in mind that minimizing the L2 norm is the same as minimizing the square of the L2 norm. So for the sake of simplicity, we can consider just the L2 norm. In order to minimize the sum of L1 and L2 norms, the tangent between the sphere and the octahedron needs to be found. If the octahedron is much smaller than the sphere, the tangent is likely to be a vertex, and if the tangent is a vertex, then we can zero out atleast one of the coefficients. This shows that L1 regularized least squares can give a sparse solution. By increasing and decreasing the size of the octahedron in comparison with the sphere, we are changing the sparsity of the solutions. The regularization parameter, λ, is used to determine how much weight needs to be given to the octahedron. If we have a large, λ, we would have to make the L1 term really small. By making the L1 really small, we are reducing the size of the octahedron. So, for large λ, we need a small L1 term, hence, giving more sparse solutions. While for a small λ, we need a large L1 term, hence, giving less sparse solutions. Thus, λ, can range from zero to infinity, providing most sparse to least sparse solutions. Sparse solutions do not have to be exactly zero; they can be very close to zero. Because of the way L1Ls obtains sparsity through the regularization parameters, L1Ls gives solutions which are approximately zero. Due to this, a threshold needs to be set to determine if the approximately zero value should be counted as a zero value or a nonzero value; in other words, a target or not a target. The ROC curves change depending on the threshold.

3.5.2 Algorithm

L1 regularized Least Squares solves for

$$min \parallel Ax - y \parallel_2^2 + \lambda \parallel x \parallel_1 \tag{3.2}$$

where λ is a regularization parameter. By changing the value of λ we can change the desired sparsity of the reconstructed signal. In order to solve for (3.22), truncated newton interior point method is used. The procedure shown below can be used to solve the newton's method.

1. Transform the problem into a convex quadratic problem. In our case, (3.22) would be transformed into

$$min \parallel Ax - y \parallel_2^2 + \lambda \sum_{i=1}^{n} u_i \tag{3.3}$$

subject to

$$-u_i \leq x \leq u_i \; i = 1, 2, \ldots, n \tag{3.4}$$

where

$x \in \mathbb{R}^n$ and $u \in \mathbb{R}^n$

2. Solve the Newton system to compute the search direction:

$$\Delta x = -\nabla f(x) / \nabla^2 f(x) \tag{3.5}$$

for $f(x) = \|Ax - y\|_2^2 + \lambda\|u\|_1$ The Newton step can be solved by preconditioned conjugate gradient method. To find the exact solution, cholesky factorization can be used. However, an approximate solution can be found by the conjugate gradient method. To speed up computation even more, preconditioned conjugate gradient method is used.

3. Use a backline search algorithm to determine the step size.

4. Update the iterate.

5. Calculate dual feasible point.

6. Evaluate duality gap.

7. Apply a stopping criteria using the duality gap. If stopping criteria is not met, repeat from step 2.

This algorithm is described more in detail in [18]. We will use the accompanied software package provided in [19] perform the simulations.

Chapter 4

DETECTION PERFORMANCE

In this section we compare reconstruction algorithms for a 64-length signal with 3 targets of equal amplitude of value 1. We use 50% of the measurements, i.e., 32 measurements to reconstruct the signal. 1000 Monte Carlo simulations are performed to attain the results shown later in this Chapter. Since the same signal was used in Section 2.5, we found that Bernoulli measurement matrix works the best for this signal. Hence, Bernoulli measurement matrix was used for all algorithms for all our simulations. In order to show the detection performance, ROC curves were used that plot probability of detection vs false alarm.

We compared the CS reconstruction algorithms to that of a traditional radar system. We used a raster scan for the traditional radar. A raster scan does a direct acquisition of the sampled target scene, and hence, was the best detection a radar system could do for each case.

4.1 Detection performance of Noisy Observations (Output Noise Model)

Many times the observations, y, may contain noise. This can be due to shot noise or some other form of measurement noise. We can model this by:

$$y = Ax + n \tag{4.1}$$

In this case noise, n, is the output noise [20]. We use n as a white Gaussian noise vector. Here we assume that the signal, x, is noiseless. This model is commonly investigated in radar applications as well as in compressive sensing. In the recent years, CS algorithms have been studied to determine reconstruction of signals in the presence of observation noise. Here, we extend that to detection performance, comparing different reconstruction algorithms.

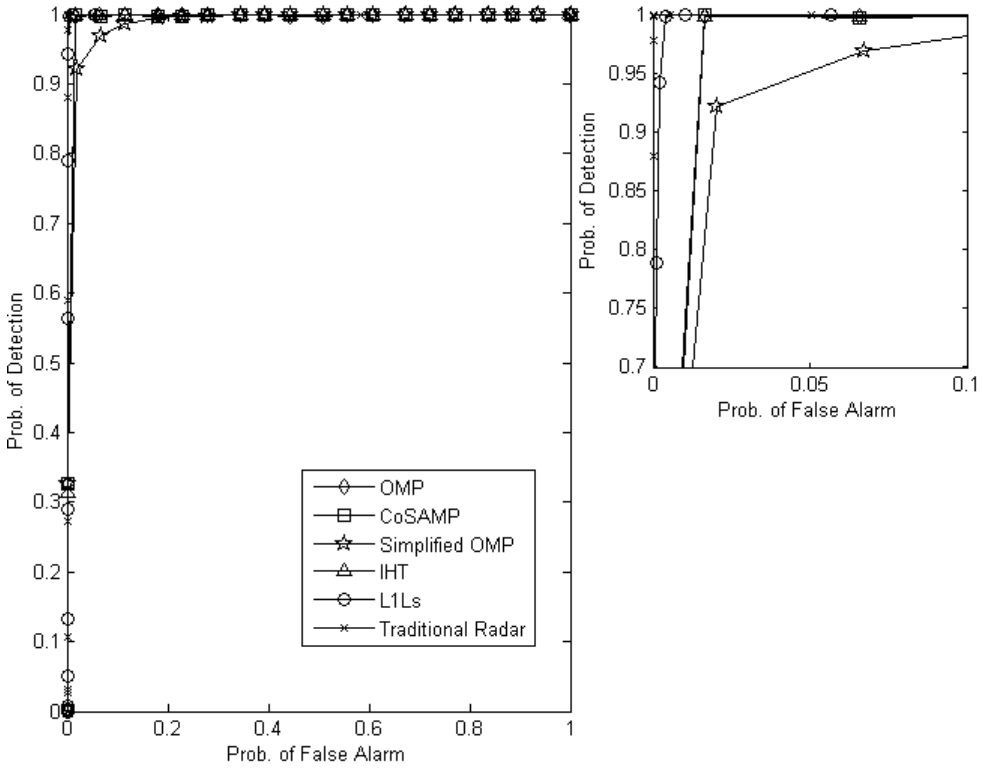

FIG. 4.1. Comparison of different reconstruction algorithms in 25dB SNR. Right: Zoomed in version of the plot on the left

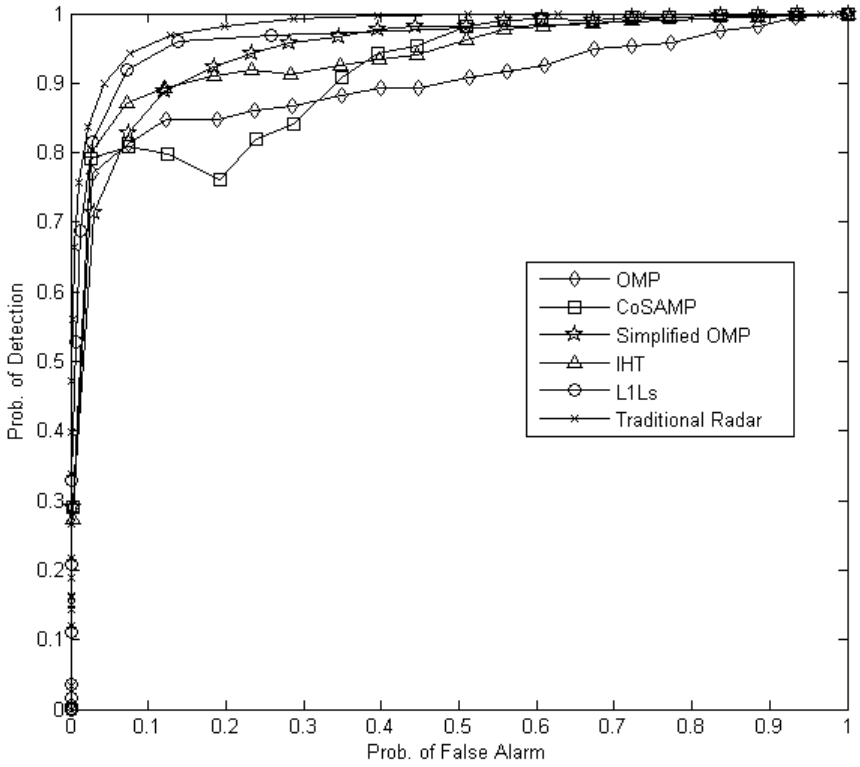

FIG. 4.2. Comparison of different reconstruction algorithms in 15dB SNR

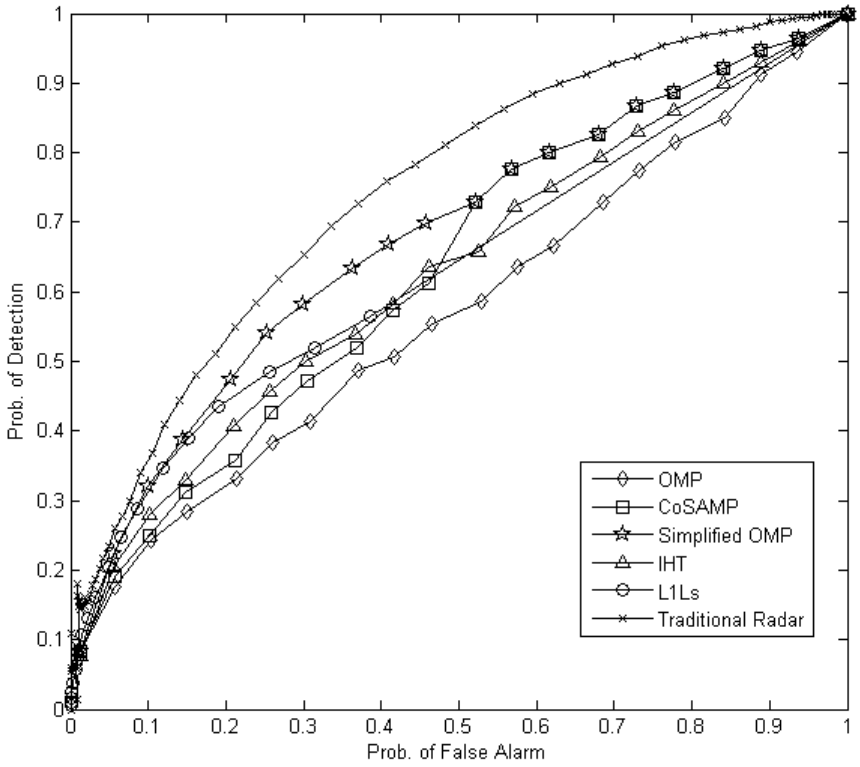

FIG. 4.3. Comparison of different reconstruction algorithms in 5dB SNR

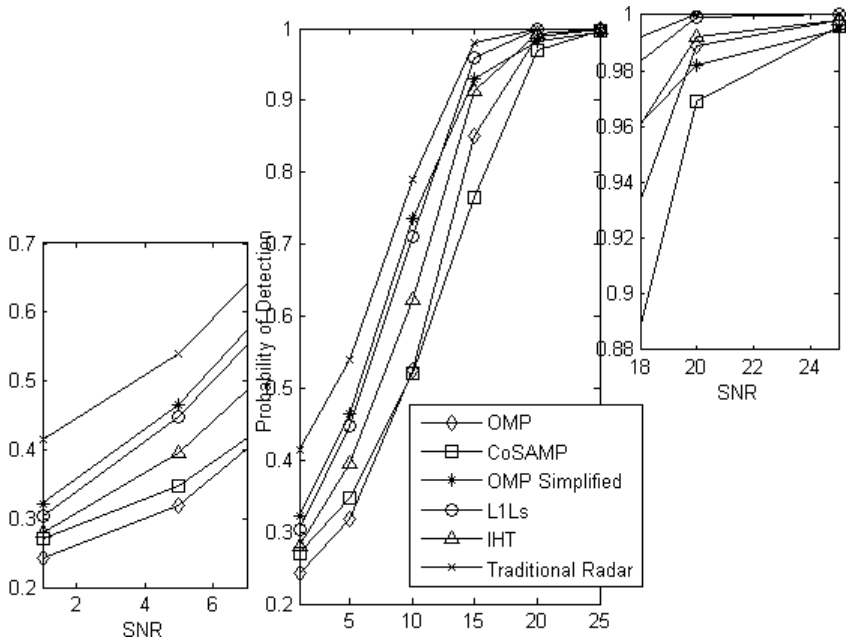

FIG. 4.4. Detection performance vs SNR for CFAR of 0.2. Left: zoomed in version for the lower SNR. Right: zoomed in version for the higher SNR

In this case, the traditional radar system did not perform as well for low SNR. Hence, the disparity between the CS and traditional radar system is not much. This noise model is contaminated with more noise (as compared to the noise model in Section 4.2) since the noise is added to the observations. We can note that in low SNR, OMP Simplified and CoSAMP work better than IHT and L1Ls. From [13] we can see that CoSAMP works better than IHT for some sparsities when the target coefficients are uniformly distributed. In our case, all the target coefficients have a reflectivity value of 1 and thus, are uniformly distributed. CoSAMP works better than IHT and L1Ls only when sparsity is over estimated for the lower SNRs. For CoSAMP, when k increases to more than half the length of the

signal, i.e, we assume the sparsity to be higher than half the signal length, CoSAMP takes in k largest dot product elements- making the performance of CoSAMP almost equal to that of Simplified OMP. We can classify CFAR of 0.2 as low. For low SNR and high-estimated sparsity, OMP Simplified worked the best.

4.2 Detection Performance of Noisy Target Profiles (Input Noise Model)

In most CS applications, signals are assumed to be noiseless, and the measurements are noisy (Section 4.1). However, in many practical applications, the signal is subject to some pre-measurement noise such as clutter, background noise etc. [21] [22]. The input noise can be modeled by many different probability density functions. For example, the input noise can be due to compound Gaussian clutter, non-Gaussian clutter, etc. For simplicity and to mainly focus on detection performance of reconstruction algorithms rather than input noise modeling, we assume the input noise to be Gaussian. This noise model corresponds to a technique known as noise folding, and only in the recent years it has emerged as a field of study in CS [23]. In this section we assume that the measurement noise is negligible, and the signal noise is significant. We do not include any measurement noise. Here we use Gaussian noise as the input noise or the signal noise. The following model is used:

$$x' = x + n \tag{4.2}$$

where x represents the original target profile, and n is i.i.d Gaussian noise.

$$y = Ax' \tag{4.3}$$

$$y = A(x + n) \tag{4.4}$$

$$y = Ax + An \tag{4.5}$$

This model has a noise reduction performance [24] and can be seen as an input noise model [20].

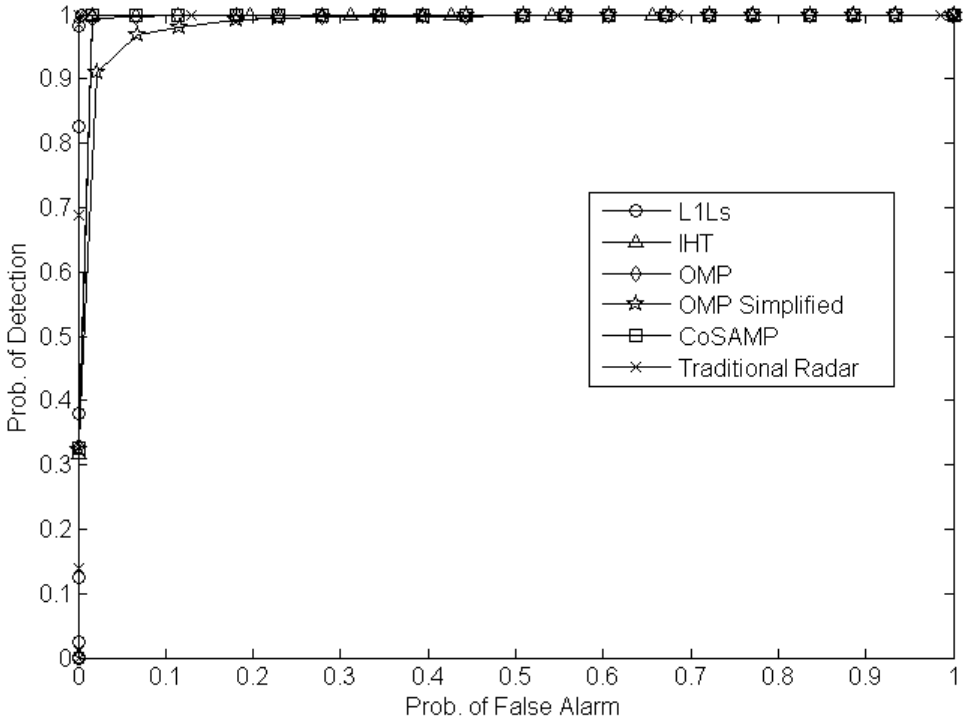

FIG. 4.5. Comparison of different reconstruction algorithms for signal with AWGN of SNR 25dB

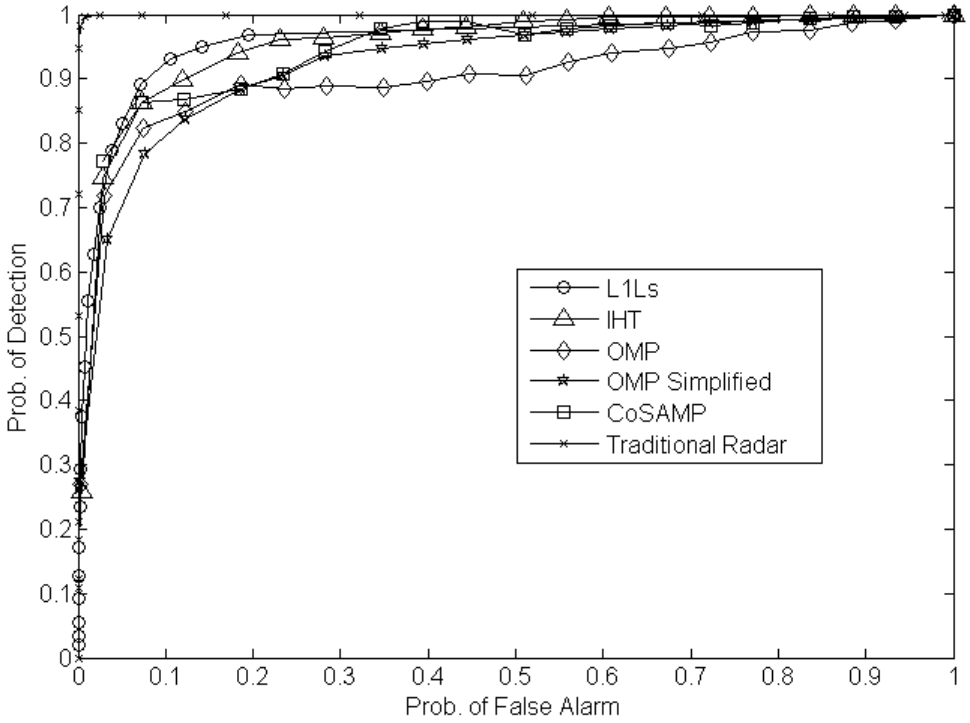

FIG. 4.6. Comparison of different reconstruction algorithms for signal with AWGN of SNR 15dB

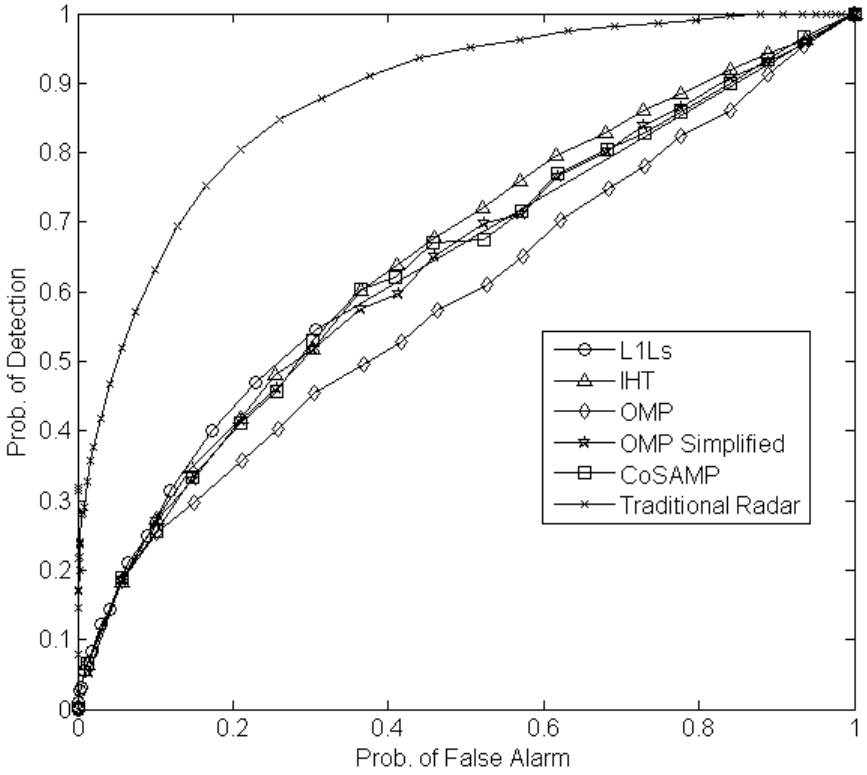

FIG. 4.7. Comparison of different reconstruction algorithms for signal with AWGN of SNR 5dB

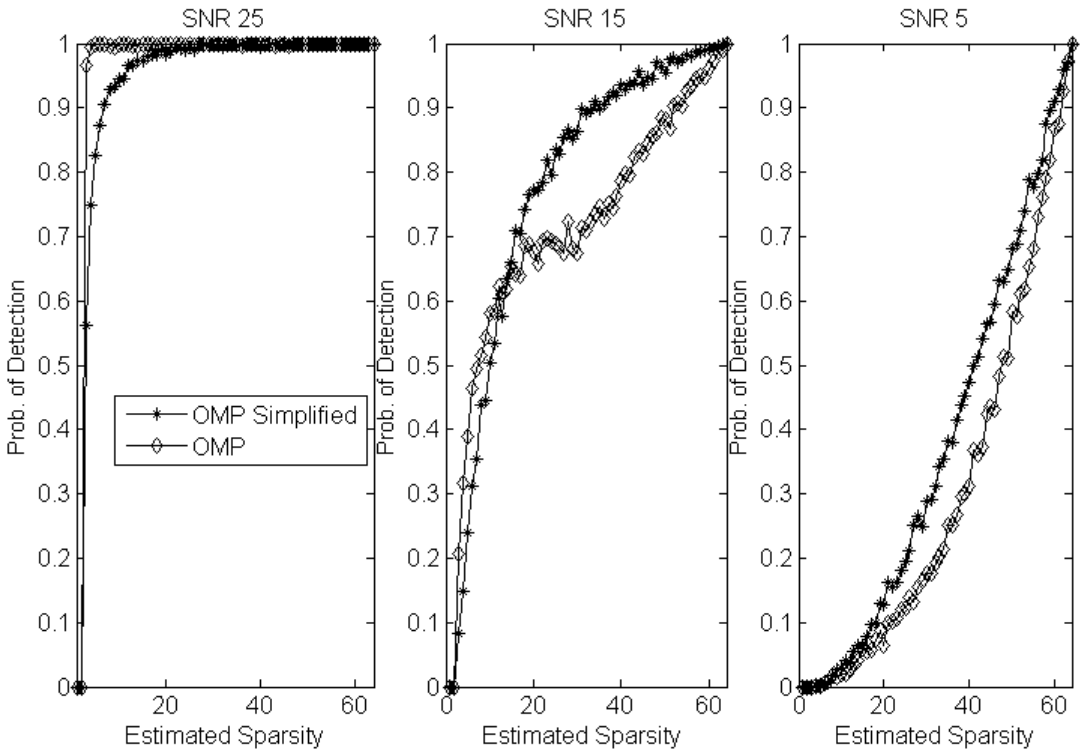

FIG. 4.8. Comparison of detection performance OMP and OMP Simplified for estimated sparsity. Left to right: SNR 25dB, SNR 15dB, SNR 5dB

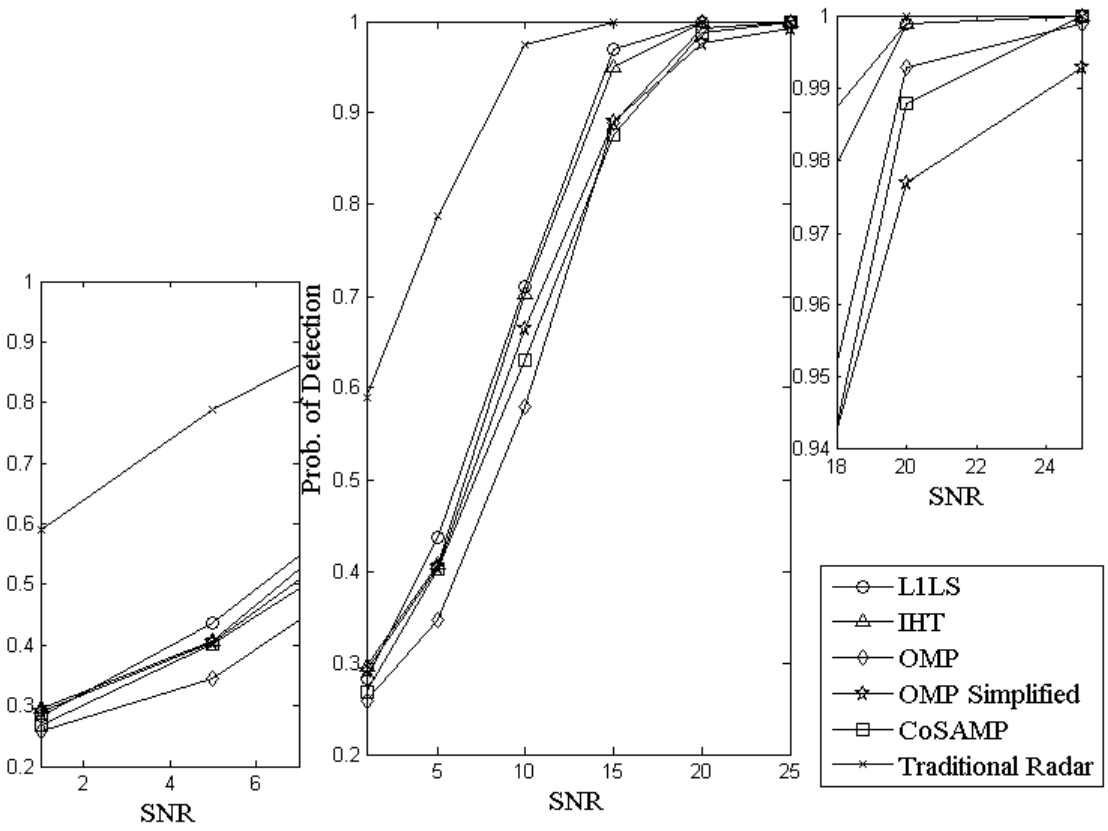

FIG. 4.9. Detection performance vs SNR for CFAR of 0.2. Left: zoomed in version for the lower SNR. Right: zoomed in version for the higher SNR

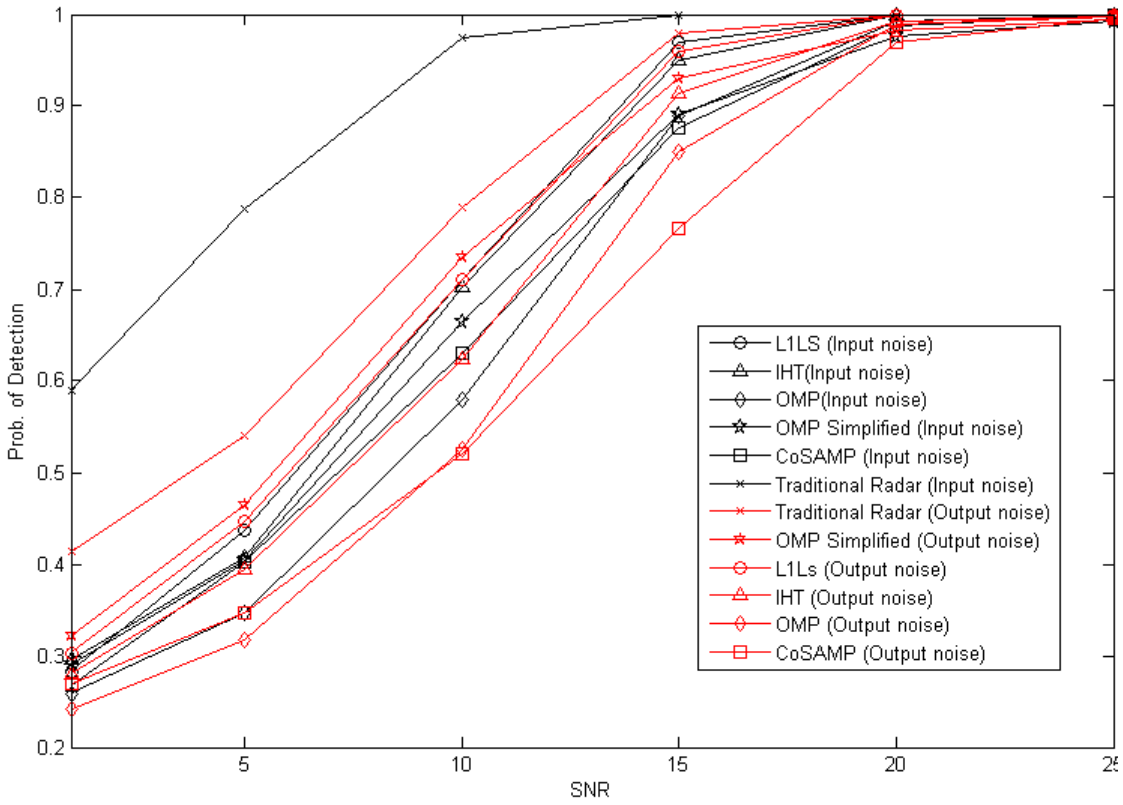

FIG. 4.10. Comparison of Input noise model (2) and output noise model (1)

Furthermore, probability of detection was determined for varying SNR given a constant probability of false alarm of 0.2 for a 64- length target profile with 3 targets (Fig. 4.9). A very interesting find from the Fig. 4.6, 4.7, 4.8, and 4.9 is that OMP Simplified works better than OMP when the signal, x', is noisy and when sparsity is over estimated. L1 Regularized least squares and IHT worked better as compared to the variants of OMP. OMP Simplified worked better in low SNR as compared to OMP and vice versa.

L1 Regularized Least Squares and IHT work almost equally well. We should note that IHT

can recover only very sparse signals, while L1 minimization technique works well for even moderately sparse signals [16]. L1 Regularized Least Squares and IHT can be computationally expensive. On the other hand, the computational complexity of OMP is low [25] [26].Simplified OMP is a faster version of OMP since it does not have to go through k iterations and perform the dot product each time. By doing so, we can risk the accuracy of detection in moderately low noise signals (Fig. 4.5, 4.6 and 4.8). When the SNR of x' is low, OMP simplified works as well as or even better than CoSAMP (Fig. 4.6, 4.7 and 4.9).

From Fig. 4.6, we can note that OMP and OMP Simplified cross over at a point. This shows that the estimated sparsity (under or over estimating sparsity) can make a difference in deciding if OMP outperforms OMP Simplified or vice versa. In Fig. 4.8, we show the effect of estimated sparsity and detection between OMP and OMP Simplified. For high SNR, such as SNR 25 shown in Fig. 4.8, OMP always has a higher detection performance until they even out at about a sparsity level of 30, where the detection is almost perfect. However, for lower SNR signals, OMP Simplified outperforms OMP if the sparsity is over estimated. For the plot of source signal with AWGN of 15dB in Fig. 4.8, OMP Simplified works better than OMP after the estimated sparsity level of about 13 and for SNR 5dB, after a sparsity level of 9. Hence, over estimating the sparsity of noisy signals can lead to better detection performance by OMP Simplified as compared to OMP. However, overestimating sparsity can also lead to more false alarms [27].

By using this model, the noise gets reduced because it is also multiplied by the measurement matrix (Fig. 4.10). In a way, the noise gets partially filtered out [24]. Thus, the

ideal traditional radar ROC curve using the original 64-length signal, x, also had very good ROC curves. This shows that the observations have reduced noise as compared to adding noise directly to the observations as we seen in the previous section. However, we can see quite a disparity between the ideal radar detection system and the CS reconstruction algorithms. This is because we used a raster scan for ideal radar detection. By doing so, the measurement matrix contains a lot of zero terms, which in turn, cancelled out a lot of the noise elements of the noise vector. If a different transmitted signal would have been used, the disparity would probably not have been as great. Although, the detection of both is comparatively higher in this noise model than the previous one, CS does not perform as well for the lower SNR in this noise model.

4.3 Detection Performance in Background Noise and Measurement Noise

Many times, both input and output noise are present. A signal is likely to be corrupted due to some background noise and the observations at the receiver end are likely to be corrupt because of observation noise such as shot noise. We use input and output noise to be Gaussian. From [21] we know that by adding both input and output noise, there is a large noise increase in the compressed measurements. The following model is used [28]:

$$y = A(x + n1) + n2 \tag{4.6}$$

where $n1$ and $n2$ are both i.i.d Gaussian. We perform detection performance simulations when $n1$ is 20dB and when $n2$ is 15dB and when both $n1$ and $n2$ are 15dB.

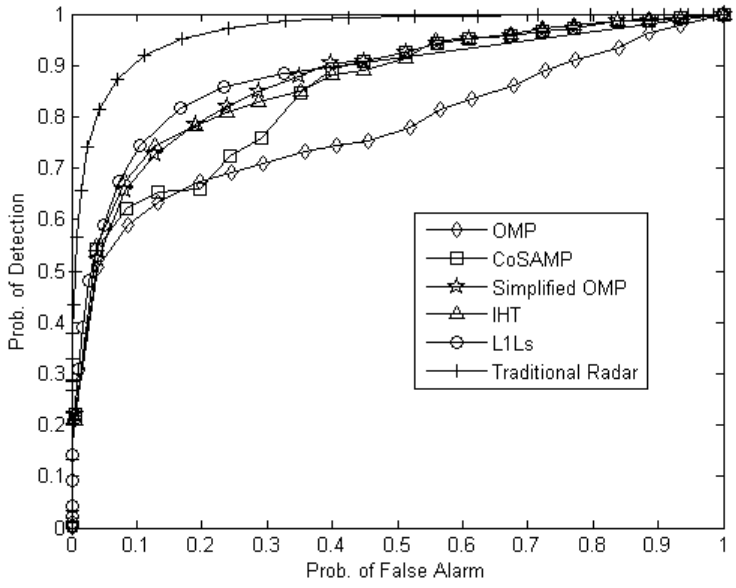

FIG. 4.11. Detection performance for input noise of 15dB and output noise of 15dB

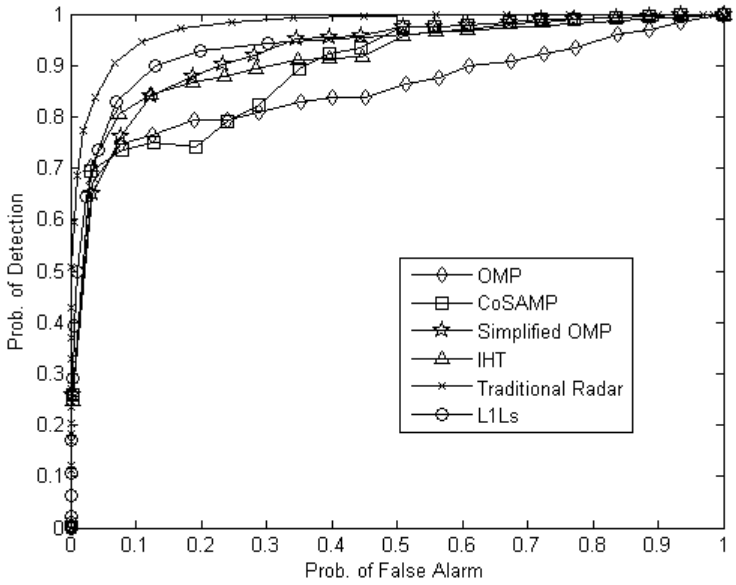

FIG. 4.12. Detection performance for input noise of 20dB and output noise of 15dB

If the source noise is increased (Fig. 4.11), CS detection performs much worse than traditional radar. For better comparison, L1Ls reconstruction for both the noise models is shown below. The noise models used in section 4.1, 4.2, and 4.3 will be referred as noise model 1, 2 and 3 respectively.

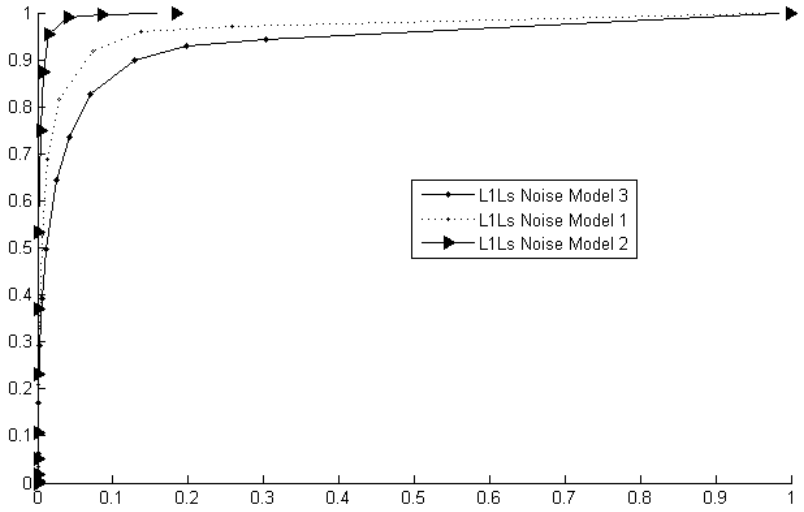

FIG. 4.13. Detection performance comparison of L1Ls for noise model 1, model 2 and model 3 with input noise of 20dB and output noise of 15dB

Here we can see that in the presence of internal and external noise the detection performance of all the algorithms deteriorates.

Chapter 5

TIMING ANALYSIS

5.1 Computational Complexity

In this section we compute the computational complexity of each algorithm We assume that $k \ll M \ll N$, where k is the estimated sparsity, M are the number of measurements, and N is the length of the signal.

5.1.1 Orthogonal Matching Pursuit (OMP)

The computational complexity for each step is explained below:

1. $< \phi R >$

This step requires matrix multiplication of ϕ, which is MxN matrix with R, which is a 1xM vector. This has a computational complexity of $O(MN)$.

2. Maximum of $< \phi R >$, which gives a Nx1 vector.

Hence, it has a computational complexity of $O(N)$.

3. The next step we need to consider is solving the least squares problem. We can solve the least squares problem by:

$$minimize \|y - \Theta x\|_2^2$$

$$\|y - \Theta x\|_2^2 = (y - \Theta x)^T (y - \Theta x)$$

$$= y^T y - y^T \Theta x - y \Theta^T x^T + \Theta^T x^T \Theta x$$

$$= y^T y - 2y \Theta^T x^T + \Theta^T x^T \Theta x$$

In order to minimize, we can take the derivative with respect to x and set it to zero.

$$0 = -2\Theta^T y + \Theta^T x^T \Theta + \Theta^T \Theta x$$

$$0 = -2\Theta^T y + \Theta x \Theta^T + \Theta \Theta^T x$$

$$0 = -2\Theta^T y + 2\Theta \Theta^T x$$

$$2\Theta^T y = 2\Theta \Theta^T x$$

$$\Theta^T y = \Theta \Theta^T x$$

$$x = (\Theta^T \Theta)^{-1} \Theta^T y$$

At each iteration, i, Θ has i columns of size M. Hence, the new matrix, Θ, is of size ixM. Doing a $(\Theta^T \Theta)$ gives a ixi resulting matrix. Thus, a cost of $O(iM)$. The cost of inverting

this ixi matrix by Gauss-Jordan elimination is $O(i^3)$. The cost of $\Theta^T y$ is $O(i^2)$.

4. Calculate the approximation: $\alpha_t = \Theta_t \; x_t$

Θ is of size ixM and x is of size 1xi. This gives a computational complexity of $O(iM)$.

5. Compute new residual: $R_t = y - \alpha_t$

y and α_t are $Mx1$ matricies. The subtraction at each iteration will take M computations, hence, $O(M)$.

Therefore, the total cost per iteration will be $O(MN)$. If the signal is k-sparse, this algorithm will be iterated k times, giving a total computation complexity of $O(kMN)$ [29].

5.1.2 Compressive Sampling Matching Pursuit (CoSAMP)

The calculation of computational complexity for CoSAMP is similar to that of OMP.

1. Compute $| < \phi_j R > |$. This has $O(MN)$ complexity.

2. Find maximum $2k$ elements. This requires a sorting of the vector, and with N elements the best sorting algorithm gives a computational complexity is $O(M)$.

3. Solve the least squares problem: $argmin_x \; \|y - \Theta_t x\|_2^2$

This step is similar to Step 3 from the previous OMP section. However, instead of i, at each iteration there are $2k$ columns of length N. Θ, is of size $2k$xM . The total cost given the inversion as well would be $O((2k)^2)$.

4. Prune to retain the maximum k elements:

This would require to sort through the 1x$2k$ vector of estimated x. Thus, giving a

computational cost of $O(2k)$.

5. Calculate the approximation: $\alpha_t = \Theta_t \; x_t$

Θ is of size kxM and x is of size 1xk. This gives a computational complexity of $O(kM)$.

6. Compute new residual: $R_t = y - \alpha_t$

y and α_t are $Mx1$ matrices. The subtraction at each iteration will take M computations, hence, $O(M)$.

These steps are iterated over t times until a stopping criteria is reached. We set the maximum number of iterations to be N. So, we can guarantee $t \leq N$. The total computational cost is $O(tMN)$.

5.1.3 Simplified Orthogonal Matching Pursuit

Simplified OMP follows the same steps as OMP, except it is not iterative. Hence, the total cost for Simplified OMP will be $O(MN)$.

5.1.4 Normalized Iterative Hard Thresholding (IHT)

This algorithm is solved by: $x^{t+1} = H_k(x^t + \mu\phi^T(y - \phi x^t))$.

To find the total computational complexity, we will break down the equation into parts.

1. ϕx^t :

ϕ is an MxN matrix and x^t is an Nx1 vector. This gives a computational cost of $O(MN)$.

2. $y - \phi x^t$ has $O(M)$ complexity since it is a subtraction of 2 1xM vectors.

3. $\phi^T(y - \phi x^t)$: This requires a multiplication of an NxM matrix with a Mx1 vector.

Hence, this steps costs $O(MN)$.

4. Computing the step size, $\mu = \frac{g_{\Gamma^n}^T g_{\Gamma^n}}{g_{\Gamma^n}^T \phi_{\Gamma^n}^T \phi_{\Gamma^n} g_{\Gamma^n}}$

$g_{\Gamma^n}^T g_{\Gamma^n}$ requires $O(N)$.

Breaking the denominator into 3 matrix multiplications results in:

a. $g_{\Gamma^n}^T \phi_{\Gamma^n}^T$: $O(MN)$

b. $g_{\Gamma^n}^T \phi_{\Gamma^n}^T \phi_{\Gamma^n}$: $O(MN)$

c. $g_{\Gamma^n}^T \phi_{\Gamma^n}^T \phi_{\Gamma^n} g_{\Gamma^n}$: $O(N)$

5. $x^t + \mu\phi^T(y - \phi x^t)$:

Addition of x^t , which is a Nx1 vector, requires $O(N)$.

6. The final step is to set all but the k max elements to zero. This requires sorting of the Nx1 vector, which has a cost of $O(N)$, and then finally setting the elements to zero, which again requires $O(N)$.

The total cost is $O(4MN)$ per iteration. So, for t iterations, the cost is $O(4tMN)$.

5.1.5 L1 Regularized Least Squares (L1Ls)

Computing the dual feasible point and duality gap can be computationally expensive. However, due to the log barrier bounds placed, calculating the exact computational complexity is very difficult. Let us use, $O(j)$ for the total computational complexity those steps. The value of j can vary at each iteration. Since we used the truncated newton interior point method, preconditioned conjugate gradient (PCG) is used to solve the newton system. This has a computational complexity of $O(MN)$ [18]. For each iteration

t, s PCG iterations are computed, where s is some variable value, which can change per iteration, depending on the Newton system and the stopping criteria for the PCG algorithm. Let us call the largest s, S. Thus, the approximate total computational complexity would be $O(tSMN + tj)$.

The computational complexities of the different reconstruction algorithms are summarized in the table 5.

Table 5.1. Computational Complexity

Reconstruction Algorithm	Computational Complexity
L1 Ls	$O(tSMN + tj)$
IHT	$O(4tMN)$
OMP	$O(kMN)$
CoSAMP	$O(tMN)$
Simplified OMP	$O(MN)$

Note that the t for each algorithm would be different since the number of iterations each algorithms go through differ according to various parameters; some of these parameters can be set by the user, but may not necessarily be directly related to the number of iterations. For each iteration L1Ls goes through, the algorithm also has to go through several more iterations (s) to solve the newton system. Hence, the total number of iterations can be the highest in comparison to the other algorithms. t

for CoSAMP is set to be always $\leq N$, while t for IHT and t combined with s for L1Ls can be $> N$.

5.2 MATLAB Simulation Timing

We performed MATLAB timing analysis for each of the algorithms. MATLAB simulations were performed on a Core(TM) i5-2430M CPU at 2.4GHz. First, the average time for the varying sparsity levels (1-64) was calculated. Then the average of that over 1000 Monte Carlo simulations was calculated. The same target profile with SNR of 15dB (Section 4.2) was used.

Table 5.2. MATLAB Simulation Timings

Reconstruction Algorithm	Timing	Speed Up
L1 Ls	0.1064	1
IHT	0.1036	1.027
OMP	0.0318	3.346
CoSAMP	0.0343	3.102
Simplified OMP	0.0027	39.41

Comparatively, we can see from table 5.1 and 5.2 that OMP and its variants have lower computational complexity [25] [26]

Chapter 6

CONCLUSION

6.1 Results Summary

This thesis analyzes the use of compressive sensing radar. The detection performance of different reconstruction algorithms are analyzed and also compared with that of a traditional radar system. In addition, the computational complexities of different CS reconstruction algorithms are analyzed as well.

6.1.1 Traditional Radar vs CS Radar

In this thesis we analyzed when the benefits of CS outweigh traditional radar. As seen from the detection performance plots, compressive sensing does not match up to an ideal radar system when noise is present. However, the detection performance of certain algorithms in the presence of noise is fairly close to the ideal performance when measurement noise is added, i.e, for the noise model $y = \phi x + n$. When the signal, x, itself is noisy, i.e, for the noise model $y = \phi(x + n)$, an idea radar system has a much better detection per-

formance. Since in most cases noise is added as part of observation, compressive sensing is definitely beneficial; the detection performance of ideal radar system and compressive sensing is very close. As a general statement, in regards to either noise models, as the SNR increases, detection performance of CS matches that of an ideal system fairly well. Hence, by using only half or less samples (50% in our simulations) in low noise, high SNR environments, detection performance is as good as that of an ideal system. Thus, for high SNR environments, compressive sensing definitely is more advantageous.

6.1.2 Comparison of reconstruction algorithms

In terms of different reconstruction algorithms, depending on the noise model, the detection performance differs. For the noise model described in equation 4.2, L1Ls and IHT have the best detection performance in both low SNR and high SNR signals. Also, for high-estimated sparsity, OMP Simplified worked better than OMP and as well as CoSAMP in high SNR signals. This outcome differs when the observation noise is noted (Section 4.1). In this case due to the greater impact of noise, OMP Simplified, which chooses the maximum k elements, had the best detection performance in the presence of noise. However, OMP Simplified does not work as well in high SNR environments. IHT does not work well when the signal is not very sparse; hence, in the first noise model (Section 4.1), when the impact of noise was greater than the noise folding model, IHT did not work well. In low noise environments, we can directly see an inverse relationship between computational complexity and detection performance. However, in very noisy environments, this is not

necessarily true. Instead, the least computationally complex algorithm worked the best in very noisy systems.

Based on the detection performance of the various algorithms in the different noise models, the best reconstruction algorithms to use in the different scenarios are listed in Table 6.1. We chose the algorithms that had the best detection probability for a low false alarm probability.

Table 6.1. Best Reconstruction Algorithm to Use in Terms of Good Detection and Low False Alarms for Given Circumstances

Circumstance	Reconstruction Algorithm
Source is less noisy	L1Ls, IHT, CoSAMP, OMP
Source is moderately noisy	L1Ls, IHT
Source is very noisy	L1Ls, IHT, Simplified OMP, CoSAMP
measurements are less noisy	L1Ls, IHT, CoSAMP, OMP
measurements are moderately noisy	L1Ls, IHT, Simplified OMP
measurements are very noisy	L1Ls, Simplified OMP

6.1.3 Comparison of Different Noise Models

The detection performance of the input noise model is better than that of the output noise model when the same amount of noise is added for a traditional radar and slightly better for CS based radar except for OMP Simplified algorithm. Because the measurement

matrix gets multiplied with the noise vector as well, noise reduction takes place in the input noise model. In noise model 3, both input and output noise is present, making the overall detection performance compared to the previous two models slightly worse. Our proposed algorithm, OMP Simplified, worked well for noisy source signals, and exceptionally well for high measurement noise systems, but did not work as well for low noise systems or source signals.

6.1.4 Computational Complexity

L1Ls is a convex optimization problem and uses Newton's method. Hence, the computational complexity is the highest. This was followed by IHT and finally by the variants of OMP. The algorithm that we proposed in this thesis had the least computational complexity and the best performance for noisy measurements. OMP Simplified had the lowest computational complexity- the MATLAB runtime was 39 times faster than L1Ls. Thus, in terms of computational complexity, OMP Simplified is definitely the best algorithm out of the ones we compared.

6.2 Conclusions on the Use of CS Reconstruction Algorithms

In this section we discuss the advantages and disadvantages of using each algorithm.

L1Ls

L1Ls can be used when there is comparatively plenty of processing time and power available. Detection is very good in almost all circumstances. It is robust to noise, so can

be used for noisy situations as well. When estimating the sparsity of the signal to be low enough, i.e for less probability of false alarms, detection in noisy environments can be as good as an ideal radar system.

IHT

IHT works almost as well as L1Ls when noise is low. When source (input) noise is present it works well also, but it does not work as well when measurement noise is present. As the impact of noise gets greater, the signal becomes less sparse, and IHT does not work as well for less sparse signals. Computational complexity of IHT is slightly lower than L1Ls.

OMP

Detection performance is moderately good, but not as good as IHT or L1Ls for low noise signals. Out of all the algorithms, OMP had the worst performance when noise was high. However, as a tradeoff, computational cost is significantly low. Hence, when faster processing is required at the expense of detection, OMP can be used.

CoSaMP

CoSaMP works well for low noise environments. It also works well when the source signal is noisy. Due to the algorithm acquiring $2k$ elements and then narrowing it down to k elements after solving the least squares problem, CoSaMP tends to not work for certain sparsity values, especially when $k \sim N/4$. Similar to OMP, computational cost is low.

Simplified OMP

This algorithm works very well and better than the rest when high measurement noise is present. It also works well when the source is noisy. However, Simplified OMP does not

work as well when noise is low. A very important advantage of Simplified OMP is its extremely low computational cost; the MATLAB runtime was about 39 times faster than L1Ls.

6.3 Future Work

There are several ways this work could be extended. On a broad level, 1D radar CS work performed in this thesis could be extended to 2D or 3D. Another general extension to other applications such as LIDAR, hyperspectral, etc. could be studied more in depth since the use of CS in these fields is fairly new. The single pixel camera based on compressive sensing has been studied and implemented for optical imaging. An extension of a single pixel camera for the hyperspectral domain can bring game-changing advancement in the hyperspectral imaging field. In the case of hyperspectral, the single pixel camera is replaced with an optical spectrum analyzer. Global LIDAR imaging from satellites can also require tremendous amount of data downlink to cover complete imaging of the earth. Similar to other applications, data bandwidth limitation can be a major problem. Hence, compressive sensing can greatly benefit LIDAR applications. LIDAR imaging can also produce 3D images using micro-mirror arrays for single pixel cameras. A 1D LIDAR detection would be similar to the 1D radar detection done in this thesis. In addition, specific aspects of this thesis that can be directly extended include sparsity increment, reflectivity profiles, non-gaussian clutter, and the number of measurements required. Changing the sparsity:measurements:length of signal ratios could change the detection performance and

would be interesting to see how each algorithm reacts to such changes. The reflectivity profile of the targets in this thesis was uniformly distributed. Generally clutter can be modeled by various distributions. Performing detection performance for a specific application with a known model for clutter could change the overall detection performance of all the algorithms. Moreover, changes in the number of measurements used could change detection performance. Theoretical measurement bounds have been studied. However, applying these bounds to certain practical circumstances for detection purposes could also be an interesting study. Simplified OMP is similar to iterative thresholding with inversion algorithm (ITI). The main difference is that in ITI elements can get into and out of the active set [30], where as in simplified OMP, because it is not iterative, once the elements for the active set are chosen, they are used to determine the reconstructed, estimated values. ITI also does thresholding on the dot product, while OMP Simplified does not perform any sort of thresholding; k columns of ϕ are directly chosen without any prior thresholding. A comparison between the two will be beneficial in understanding if one algorithm outweighs the other, and if so, in which all circumstances. Furthermore, Hardware implementation of these algorithms is also a current research area. Due to the computational complexity of some of the algorithms such as L1Ls, hardware implementations and hardware complexity have not been greatly studied. Future work could also focus on implementing lower computational cost algorithms in hardware.

Bibliography

[1] E. J. Candès, "Compressive sampling," in *Proceedings o the International Congress of Mathematicians: Madrid, August 22-30, 2006: invited lectures*, 2006, pp. 1433–1452.

[2] E. J. Candès and M. B. Wakin, "An introduction to compressive sampling," *Signal Processing Magazine, IEEE*, vol. 25, no. 2, pp. 21–30, 2008.

[3] Y. Eldar and K. Gitta, *Compressed Sensing Theory and Applications*. New York, USA: Cambridge University Press, 2012.

[4] B. R. Mahafza, *Radar Systems Analysis and Design Using MATLAB*. Florida, USA: Chapman Hall/CRC, 2005.

[5] F. M. Jean-Luc Starck and J. M. Fadili, *Sparse Image and Signal Processing*. New York, USA: Cambridge University Press, 2010.

[6] E. J. Candès, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9, pp. 589–592, 2008.

[7] R. Baraniuk and P. Steeghs, "Compressive radar imaging," in *Radar Conference, 2007 IEEE*. IEEE, 2007, pp. 128–133.

[8] M. Herman and T. Strohmer, "Compressed sensing radar," in *Radar Conference, 2008. RADAR'08. IEEE*. IEEE, 2008, pp. 1–6.

[9] M. A. Herman and T. Strohmer, "High-resolution radar via compressed sensing," *Signal Processing, IEEE Transactions on*, vol. 57, no. 6, pp. 2275–2284, 2009.

[10] L. Anitori, M. Otten, and P. Hoogeboom, "Detection performance of compressive sensing applied to radar," in *Radar Conference (RADAR), 2011 IEEE*. IEEE, 2011, pp. 200–205.

[11] ——, "False alarm probability estimation for compressive sensing radar," in *Radar Conference (RADAR), 2011 IEEE*. IEEE, 2011, pp. 206–211.

[12] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 948–958, 2010.

[13] T. Blumensath and M. E. Davies, "Iterative hard thresholding for compressed sensing," *Applied and Computational Harmonic Analysis*, vol. 27, no. 3, pp. 265–274, 2009.

[14] T. Blumensath, "Sparsify," http://users.fmrib.ox.ac.uk/tblumens/sparsify/sparsify.html, 2000.

[15] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *Information Theory, IEEE Transactions on*, vol. 53, no. 12, pp. 4655–4666, 2007.

[16] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[17] D. Mary, "Pursuits in the null space," http://media.aau.dk/null_space_pursuits/2011/08/16/cosamp.m, 2011.

[18] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale l1-regularized least squares," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 1, no. 4, pp. 606–617, 2007.

[19] K. Koh, S. Kim, and S. Boyd, "l1 ls: A simple matlab solver for l1-regularized least squares problems," 2007.

[20] S. Aeron, V. Saligrama, and M. Zhao, "Information theoretic bounds for compressed sensing," *Information Theory, IEEE Transactions on*, vol. 56, no. 10, pp. 5111–5130, 2010.

[21] E. Arias-Castro and Y. C. Eldar, "Noise folding in compressed sensing," *Signal Processing Letters, IEEE*, vol. 18, no. 8, pp. 478–481, 2011.

[22] Y. Wiaux, G. Puy, and P. Vandergheynst, "Compressed sensing reconstruction of a string signal from interferometric observations of the cosmic microwave background," *Monthly Notices of the Royal Astronomical Society*, vol. 402, no. 4, pp. 2626–2636, 2010.

[23] J. Treichler, M. Davenport, and R. Baraniuk, "Application of compressive sensing to the design of wideband signal acquisition receivers," *US/Australia Joint Work. Defense Apps. of Signal Processing (DASP), Lihue, Hawaii*, 2009.

[24] N. Wan-zheng, W. Hai-yan, W. Xuan, and Y. Fu-zhou, "The analysis of noise reduction performance in compressed sensing," in *Signal Processing, Communications and Computing (ICSPCC), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–5.

[25] J. L. Stanislaus and T. Mohsenin, "High performance compressive sensing reconstruction hardware with qrd process," in *Circuits and Systems (ISCAS), 2012 IEEE International Symposium on*. IEEE, 2012, pp. 29–32.

[26] ——, "Low-complexity fpga implementation of compressive sensing reconstruction," in *International Conference on Computing, Networking and Communications*, 2013.

[27] A. Korde, D. Bradley, and T. Mohsenin, "Detection performance of radar compressive sensing in noisy environments," in *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics, 2013, pp. 87 170L–87 170L.

[28] P. B. Tuuk and S. L. Marple, "Compressed sensing radar amid noise and clutter," in *Signals, Systems and Computers (ASILOMAR), 2012 Conference Record of the Forty Sixth Asilomar Conference on*. IEEE, 2012, pp. 446–450.

[29] B. Mailhé, R. Gribonval, P. Vandergheynst, and F. Bimbot, "Fast orthogonal sparse approximation algorithms over local dictionaries," *Signal Processing*, vol. 91, no. 12, pp. 2822–2835, 2011.

[30] A. Maleki, "Coherence analysis of iterative thresholding algorithms," in *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*. IEEE, 2009, pp. 236–243.