A Theoretical Framework for Sparse Recovery Algorithm Design and Evaluation in Compressed Sensing Perspective

A thesis submitted
in partial fulfillment for the award of the degree of

Doctor of Philosophy

by

Vivekanand V.

Supervised by
Deepak Mishra, PhD.
Professor

Department of Avionics
Indian Institute of Space Science and Technology
Thiruvananthapuram, INDIA. 695 547
October 2023

Certificate

This is to certify that the thesis titled **A Theoretical Framework for Sparse Recovery Algorithm Design and Evaluation in Compressed Sensing Perspective** submitted by **Vivekanand V.**, to Indian Institute of Space Science and Technology, Thiruvananthapuram, in partial fulfillment of the requirements for the award of the degree of **Doctor of Philosophy** is a bona fide record of the original work carried out by him under my supervision. The contents of this thesis, in full or in parts have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Deepak Mishra, PhD
Research Supervisor
Professor, Department of Avionics
Indian Institute of Space Science and Technology
Thiruvananthapuram

Prof. N. Selvaganesan, PhD
Chairman Doctoral Committee
Professor and Head, Department of Avionics
Indian Institute of Space Science and Technology

Place : Thiruvananthapuram
Date : 04 October 2023

This Thesis has been examined by this Doctoral Committee

Professor N. Selvaganesan, PhD.......................................
Chairman Doctoral Committee
Department of Avionics
Indian Institute of Space Science and Technology
Thiruvananthapuram

Professor Kunal Narayan Chaudhury, PhD ....
Department of Electrical Engineering
Indian Institute of Science
Bangalore

Professor Nicholas Sabu, PhD...
Department of Mathematics
Indian Institute of Space Science and Technology
Thiruvananthapuram

Professor Lakshmi Narayanan, PhD ....
Department of Avionics
Indian Institute of Space Science and Technology
Thiruvananthapuram

Professor Deepak Mishra, PhD ....
Research Supervisor
Department of Avionics
Indian Institute of Space Science and Technology
Thiruvananthapuram

Declaration

I declare that the thesis titled **A Theoretical Framework for Sparse Recovery Algorithm Design and Evaluation in Compressed Sensing Perspective** submitted to Indian Institute of Space Science and Technology, Thiruvananthapuram, in partial fulfillment of the requirements for the award of the degree of **Doctor of Philosophy** is a record of the original work carried out by me under the supervision of **Prof. Deepak Mishra,** and has not formed the basis for the award of any other degree, diploma, associateship, fellowship or titles in this or any other Institution or University of higher learning. In keeping with the ethical practice in reporting the scientific information, due acknowledgment have been made wherever the finding of others have been cited.

Vivekanand V
SC 17 D 026
Indian Institute of Space Science and Technology
Thiruvananthapuram

Place : Thiruvananthapuram
Date : 04 October 2023

Acknowledgments

This work is completed in time because of the whole hearted support from my wife Geetha N. and my daughter Niranjana V.

I sincerely express gratitude to my research supervisor Professor Deepak Mishra for his invaluable suggestions, encouragements and guidance extended to me during my work.

I thankfully acknowledge Professor Manoj B.S. for the suggestions and guidance given to me as the first chairman of the doctoral committee. I am grateful to the doctoral committee members Professor Kunal Narayan Chaudhury, Professor Nicholas Sabu and Professor Lakshmi Narayanan for their suggestions for improvements during my research work. I am grateful to the current doctoral committee chairman and the head of the department Professor N. Selvaganesan for his reviews on my work for timely completion.

I am grateful to the editors of Elsevier and Springer Journals, and anonymous reviewers of these Journals who have helped me to improve the quality of my publications. I thank the staffs of Library at Indian Institute of Space Science and Technology, Thiruvananthapuram and Vikram Sarabhai Space Centre Thiruvananthapuram for their support.

I am thankful to the Director, Indian Institute of Space Science and Technology, Thiruvananthapuram and the Director, Vikram Sarabhai Space Centre Thiruvananthapuram for supporting my research work. I am grateful to the scientific temper and heritage in Vikram Sarabhai Space Centre which encouraged me to pursue this research work.

I dedicate this work in loving memory of my mother.

Vivekanand V

Place : Thiruvananthapuram
Date : 04 October 2023

Abstract

Compressed sensing is a well established signal acquisition method. This involves sampling of correlated and integrated signal at reduced sampling rate. The compressed sensed signals are not direct time domain representations, hence the reconstruction of the original signal involve function minimization methods or matrix minimization methods. Large numbers of sparse signal reconstruction algorithms are developed in the recent time. The availability of large numbers of reconstruction algorithms create dilemma in choosing a particular method for a specific reconstruction application. The recovery algorithms are generally compared in terms of computational complexity, computational time, probability of recovery and recovery precision. Typically absolute Mean Squared Error (MSE) and relative MSE are used to compare the recovery precision of various sparse recovery algorithms. However, these two metric alone are not sufficient to assess all algorithms. The research work presented in this thesis starts with a novel algorithm evaluation strategy by ranking the algorithms based on the observable similarity between the original and the reconstructed signal.

The thesis presents four consequential developments: first is the description of a novel method for analysis and ranking of sparse recovery algorithms. Second a frame work for improving the accuracy of sparse signal recovery algorithms using iterative residue estimation, proximal projection and segmented thresholding and the development of two new recovery methods using the proposed frame work. Third, the evaluation of an IoT based computing platform for the implementation of the proposed sparse reconstruction algorithms. Fourth the implementation of the proposed algorithm in real-time networked data acquisition scenario.

The signal reconstruction from the compressed sensed data need iterative methods since the sparse measurement matrix is analytically non invertible. The iterative thresholding and ℓ_0 function minimization are of special interest as these two operations provide sparse solution. However these methods need an inverse operation

corresponding to the measurement matrix for estimating the reconstruction error. The pseudo-inverse of the measurement matrix is used in general for this purpose. In the second part of the work, a sparse signal recovery framework using an approximate inverse matrix $\mathbf{Q}$ and iterative segment thresholding of ℓ_0 and ℓ_1 norm with residue addition is presented. Two recovery algorithms are developed using this framework. The ℓ_0 based method is later developed into a basis function dictionary based network for sparse signal recovery. The proposed framework enables the user to experiment with different inverse matrices to achieve better sparse signal recovery efficiency and implement the algorithm in computationally efficient way. Also, the proposed framework is used in the development of a cascade computing network for sparse signal recovery.

In the third part, the functional evaluation of an IoT based computing platform for implementation of the proposed sparse reconstruction algorithm is presented. The Beagleboard is used as prototyping and product development platform, however the full computational and networking capability of its ARM processor AM3358 and programmable real time unit are not fully utilized due to the bandwidth limitations of the networking device used in the board. The network performance evaluation of the board is performed experimentally and compared with the real time requirements of a networked commanding and data acquisition system. The feasibility of using this board for real time applications with a response latency of $< 20ms$ is studied. The observations from the timing analysis indicate that the timing constraints need to be implemented on the system for getting real-time performance.

The work presented in this thesis concludes by implementation of the proposed sparse recovery algorithm on the IoT computing platform, for realization of a networked system for acquisition and reconstruction of naturally sparse events. The naturally sparse event acquired here is the surge in ground potential. A common reason for electronic measurement anomaly is the inadvertent rise in ground potential with respect to measurement ground. The ground potential rise happens during current leakage to the ground from lightning or from power grid and leads to catastrophic failure unless appropriate preventive action is taken to isolate the sensitive measurement systems. A networked system for compressed sensing and transmission of the ground potential measurement values to a remote data monitoring station is demonstrated using the proposed method and platform. The limited processing power of such devices is not sufficient enough to run high computation intensive routing algorithms. Hence a lightweight routing algorithm for this purpose is also proposed. The discussion on the reliability of such systems is presented for completeness. The multipath route discovery strategy presented here reconfigures the network to an optimal configuration with respect to energy dissipation and node distribution.

In brief, the work presented in this thesis begins with analysis of various sparse signal reconstruction algorithms, then proposes a novel metric for ranking these algorithms using the signal similarity and probability. Based on the salient features of various sparse signal reconstruction algorithm, a framework for improving the performance of these algorithm is presented. This framework is used in the development of a function dictionary based computing network for sparse signal reconstruction. To implement the proposed algorithms, an IoT based computing platform is selected and evaluated to confirm that it meets the computational requirements. A distributed data acquisition system for measurement and reconstruction of sparse signal using the proposed algorithm is presented. Additionally a low power data routing algorithm for the IoT based system is also developed to support the data communication.

Contents

1 Introduction **13**

1.1 Sparse Signals 14

2 New Metric for Sparse Recovery Algorithm Evaluation **19**

2.1 Introduction 19

2.2 Sparse Signal Measurement and Recovery 21

2.3 Recovery Performance Measures 27

2.3.1 Sparse Recovery Limit 28

2.4 Performance Evaluation 31

2.4.1 Results 32

2.4.2 Observation 37

2.5 Chapter Summary 38

3 Framework for Segmented Threshold Based Sparse Signal Recovery **47**

3.1 Introduction 47

3.1.1 Constraints in Computing Platform and Algorithms 49

3.2 Optimization Based Sparse Signal Recovery 51

3.2.1 Improving Sparse Signal Recovery using Residue 53

3.2.2 Improving Sparse Solution through Proximal Projection 56

3.2.3 Segmented Thresholding as Proximity Operator 57

3.2.4 Error due to Segmented Threshold 58

3.3 Framework for Sparse Signal Recovery using Residue Projection and Thresholding 60

3.3.1 Segmented Threshold Residue Projection for ℓ_1 Minimisation 61

3.3.2 Segmented Threshold Residue Projection for ℓ_0 Minimization 64

3.4 Chapter Summary 70

4 Function Dictionary Based Network Implementation for Sparse Signal Recovery 71

4.1 Simulation and Evaluation 76

4.1.1 Influence of Scale Factors α_0, β and q on SNR 78

4.2 Sparse Recovery using the Proposed ℓ_1 Based ISTRP Algorithm 81

4.3 Reconstruction of Images using the Proposed STXEL0 Algorithm 86

4.3.1 Convergence and Execution Time 91

4.4 Chapter Summary 97

5 Evaluation of AM3358 Board for Networked Sparse Signal Acquisition 99

5.1 Introduction 99

5.2 Platform Constraints 99

5.2.1 Effect of Software Threads on Throughput 103

5.2.2 Effect of Network Switch on Throughput 103

5.2.3 Bandwidth Limitations of the Board 104

5.3 Timing Analysis 105

5.3.1 Communication Bus Load 107

5.3.2 Computational Complexity and Power Dissipation 108

5.4 Chapter Summary 110

6 Distributed Measurement of Naturally Sparse Events **111**
6.1 Introduction . . . 111
6.2 Ground Potential Measurement . . . 116
6.2.1 Sparse Measurement of Ground Potential . . . 117
6.3 Wireless Sensor Network . . . 119
6.3.1 Sensor Network State Parameters . . . 123
6.3.2 Node Weight Scaling Parameters . . . 125
6.3.3 Failure Detection and Routing Switching . . . 125
6.3.4 Network Analysis . . . 126
6.3.5 Power Dissipation . . . 129
6.3.6 Reliability of Multipath Routing . . . 131
6.3.7 Reliability Estimation by Testing . . . 134
6.3.8 System Evaluation . . . 136
6.4 Chapter summary . . . 139

7 Research Summary **141**

8 Future Directions **145**

9 Publications Related to Thesis **147**

List of Figures

2-1 The graphical representation of obtaining K_{max} from $Sm(\hat{\mathbf{x}}, \mathbf{x})$ vs sparsity K curve. 29

2-2 Spares pulse signal sampled from i.i.d Gaussian noisy measurements and reconstructed using various algorithms. 34

2-3 The relative MSE in sparse vector reconstruction using ℓ_0, ℓ_p and ℓ_1 based minimization algorithms from perturbed measurements. 35

2-4 The probability of reconstruction of sparse vector using ℓ_0, ℓ_p and ℓ_1 based minimization algorithms from perturbed measurements. 35

2-5 The probability of reconstruction of sparse vector using ℓ_0, ℓ_p and ℓ_1 based minimization algorithms for various noise levels. 36

2-6 The relative MSE in sparse vector reconstruction using ℓ_0, ℓ_p and ℓ_1 based minimization algorithms for various noise levels. 36

2-7 The variation of similarity measure $Sm(\hat{\mathbf{x}}, \mathbf{x})$ vs sparsity K of greedy ℓ_1 and thresholding methods. 39

2-8 The variation of similarity measure $Sm(\hat{\mathbf{x}}, \mathbf{x})$ vs sparsity K of Lagrangian, ℓ_0, ℓ_p and ℓ_1 methods. 40

2-9 The similarity measure $Sm(\hat{\mathbf{x}}, \mathbf{x})$ vs sparsity K of the reconstructed signals recovered using the projected gradient methods and Bayesian method. 41

2-10 The variation of similarity measure $Sm(\hat{\mathbf{x}}, \mathbf{x})$ vs noise perturbation forLagrangian, ℓ_0, ℓ_p and ℓ_1 methods. 42

3-1 The mapping between sparse vector $\mathbf{x}$, measurement vector $\mathbf{y}$ and reconstructed vector $\hat{\mathbf{x}}$. 54

3-2 The gray scale representation of $\mathbf{QA}$ matrix. 56

3-3 The segmented threshold function $T\lambda(x)$ used as the proximal projection operator. 58

3-4 Schematic representation of the residue computation for minimizing the projection error. 61

3-5 The approximation of ℓ_0 norm function using polynomial $1 - 1/(1 + x^q/(\sigma/k)^q)$. 66

4-1 The dictionary of ℓ_0 gradient minimization basis functions generated from $F(x) = \frac{x^{q-1}}{1+px^q}$ for various values of q and σ. 72

4-2 The network architecture of the basis function dictionary based cascade network for sparse signal recovery developed from STXEL0 algorithm 74

4-3 The ℓ_0 minimization basis functions generated from $F(x) = \frac{x^{q-1}}{1+px^q}$. . 75

4-4 Simulation setup for algorithm evaluation. 77

4-5 SNR of the STXEL0 recovered signal for various values of residue scale factor β. 79

4-6 SNR of the STXEL0 recovered signal for various values of the gradient scale factor α. 79

4-7 SNR of the STXEL0 recovered signal for different values $q = 2, 4, 6$ and 8 when $\beta = 0.25$. 80

4-8 Comparison of electric field mill signal reconstructed using ISTRP algorithm and other optimization and thresholding based algorithms . 82

4-9 SNR of recovered sparse pulse signal when reconstructed from noisy ($-33dB$) measurements. 88

4-10 Probability of exact support reconstruction of the sparse pulse signal reconstructed from noisy ($-33dB$) measurements. 89

4-11 The house-image reconstructed from compressed measurements . . . 92

4-12 The Barbara-image reconstructed from compressed measurements . . 93

4-13 The starfish-images reconstructed from compressed measurements . . 94
4-14 The airplane-images reconstructed from compressed measurements . . 95
4-15 The convergence of internal parameters of the two proposed algorithms ISTRP and STXEL0. 96
4-16 The convergence of reconstruction error of the two proposed algorithms ISTRP and STXEL0. 96

5-1 The plot of data loss after 150 frames when inter message gap is < 0.7 ms. 100
5-2 The data loss between the boards in Rx and Tx mode when inter message gap is < 2.0 ms. 101
5-3 The change in communication efficiency when inter message gap is increased from 1 us to 2 ms. 102
5-4 The data loss plot when the board receive the data through multiple software threads. 102
5-5 The communication frame format of the networked data acquisition and control modules 104
5-6 The configuration of the data acquisition and control network. 106
5-7 The relation between control latency vs. number of nodes, network bus load vs. data frame size. 107
5-8 The variation in average power dissipation versus inter message gap during network data acquisition. 109

6-1 The profile of simulated ground potential rise and measurement using differential probes. 118
6-2 Sparse measurement and reconstruction of ground potential signal . . 119
6-3 The routing architecture of the distributed sparse measurement system 120
6-4 The graphical representation of the normalized link weight computation function. 124
6-5 Protocol timing diagram of data aggregation and forwarding node. . . 128
6-6 Network communication timing diagram. 130

6-7 The energy dissipation and node attrition rate of routing algorithms. 132
6-8 The comparison of distribution of the WSN nodes established by LEACH and RCAMR routing algorithms. 133
6-9 The analytical value of the reliability and estimated depletion of the residual energy. 137
6-10 The frame loss in the data acquisition node established using the R-CAMR algorithm for various inter frame delays. 138

List of Tables

2.1 **Performance comparison of sparse signal reconstruction algorithms in terms of conventional measures MSE, $relativeMSE$, Support Error and Execution time.** 43

2.2 **Performance comparison of sparse signal reconstruction algorithms in terms of conventional measures MSE, $relativeMSE$, Support Error and Execution time.** 44

2.3 **Comparison of sparse recovery algorithms in terms of ξ-metric computed for different similarity index $Sm(.)$** 45

2.4 **Comparison of sparse recovery algorithms in terms of ξ-metric computed for different similarity index $Sm(.)$** 46

4.1 **Comparison of sparse recovery algorithms in terms of SNR of the recovered sparse signal.** 83

4.2 **Comparison of sparse recovery algorithms in terms of SNR of the recovered sparse pulse.** 84

4.3 **SNR of the electric field mill signal recovered using various algorithms** 85

4.4 **Recovery performance the algorithms using various inverse matrices on low sampled Images** 85

4.5 **SNR of the image reconstructed using various algorithms** . . 90

5.1 **Realtime latency and number of control nodes** 108

5.2 **Message length and bus load** 109

6.1 **Parameter usage, cost function and complexity of routing algorithms** 115

Symbols

List of Symbols

N Sparse signal vector length.

M Measurement vector length.

K Number of non-zero elements in the vector.

(k) $k-$th iterate.

$\mathbf{x} \in \mathbb{R}^N$ Sparse vector of length N.

$\mathbf{y} \in \mathbb{R}^M$ Compressed measurement of length M.

x_i The i-the element of vector $\mathbf{x}$.

$x_i(k)$ i-the element of vector $\mathbf{x}$ in k-th iteration.

$\hat{\mathbf{x}}$ The estimated sparse vector.

$Supp(\mathbf{x})$ Non zero index locations of sparse vector $\mathbf{x}$.

$Sm(\mathbf{x}, \hat{\mathbf{x}})$ A measure of similarity between signals $\mathbf{x}$ and $\hat{\mathbf{x}}$.

ε Finite constant.

δ_K Isometric constant for K-sparse vector.

$\mathbf{A}$ The sparse measurement matrix

$\mathbf{A}^\dagger$ The pseudo inverse of $\mathbf{A}$

$\mathbf{A}^T$ Transpose of $\mathbf{A}$

$\mathbf{A}_\Omega$ Sub matrix with columns index by set Ω

$diag(\mathbf{A})$ The diagonal elements of $\mathbf{A}$

$\mathbf{Q}$ The approximate inverse of $\mathbf{A}$

ξ The metric for representing sparse recovery limit.

q The polynomial exponent

α_0, β, λ The regularization constants

$a, c, p, \alpha_k, \sigma_k, \gamma_k, \mu_k$ The scalar values for regularization

$\ell_0(\mathbf{x}), \ell_p(\mathbf{x}), \ell_1(\mathbf{x}), \|\mathbf{x}\|$ The norms of $\mathbf{x}$.

$\mathcal{N}(0,1)$ The normal distribution of 0 mean and variance 1.

$\text{sign}(x)$ The sign of the scalar x.

$\nabla f(x)$ The gradient of the function $f(x)$.

$\Theta(x)$ The value of x after thresholding.

$\mathcal{D}_g$ The domain of the function $g(x)$.

N_k The identifier for k-the node.

Ω^k The set of upstream neighboring nodes of N_k.

Φ^k The set of downstream nodes transmitting data to N_k.

$\mathbf{W}^k$ The parameter weight matrix of the node N_k .

t_d The time delay between transmissions.

BW The network communication bandwidth.

G_A The specimen antenna gain

d_{GI} The network hop distance to the data collection node.

$P_s(i)$ The observed signal strength of the node N_i.

$F(i)$ The data frame size of the node N_i.

$L(i)$ The upstream communication bandwidth of the node N_i.

$b(i)$ The available backup power of the node N_i.

$Ch(i)$ The channel number used by the node N_i for communication.

$\mathcal{R}(k)$ The network route from node N_k to N_0

$\mathcal{R}^N(k)$ The N multi-path network route from node N_k to N_0

$\mathcal{P}(t_k), \mathcal{P}(r_k), \mathcal{P}(i_k)$ The probability of transmission, reception and idling state.

E_{TX}, E_{RX} The power dissipation during data transmission and reception.

k_B The Boltzmann constant

$R(t)$ The reliability of node as function of time.

$mttf$ The mean time to failure

T_0, T_a The operational and testing temperature.

Chapter 1

Introduction

The motivations for this research begins with the requirement of developing a data acquisition method for naturally sparse signals. The primary survey of the literature points toward the compressed sensing method as the optimal method for sparse signal acquisition. However, the survey reveals a myriad of sparse signal reconstruction methods. Naturally, the scientific temper demands which is the best algorithm for sparse signal recovery, if at all the sparse signal is acquired. This leads to a study of various sparse signal reconstruction methods. But a comprehensive comparison of all of the parse signal recovery algorithms is not possible with the current evaluation metrics. This leads to the proposal of an empirical formula for collective evaluation of sparse signal recovery algorithms. From this analysis, one particular method based on iterative thresholding and proximal projection is found to be interesting. Also, the original requirement needs a hardware platform for implementation. As soon as the algorithm is selected, two new questions sprout up. First, the selection of a cost effective computing platform for implementing the method; and second, the possibility of developing a computationally optimal sparse recovery method. The first query is answered through a product survey of embedded and IoT based computing platforms available at the current time. The TI AM3358 SOC processor based IoT platform board was selected and evaluated independently to assess its capabilities. The quest for a computationally optimal sparse recovery algorithm leads to the development of a framework for algorithm development. And this results in the development of segmented threshold residue projection based norm minimization methods. Once the hardware and algorithms are ready, the sparse signal acquisition and reconstruction system is developed. A resource-constrained routing algorithm is eventually developed to use the same system for the acquisition of sparse signals from spatially distributed nodes. The thesis presents the contributions made in various fields of computation, optimization and networking before presenting the system developed for the acquisition of naturally occurring sparse events.

1.1 Sparse Signals

The signal acquisition and data compression remained as two distinct processes till the idea of compressed sensing was proposed in [1]. This process exploits the sparse nature of the signal to sample it with fewer samples compared to the limit defined by Nyquist Shannon sampling theorem. The process applies only for signals with sparse nature. However if the signal is not apparently sparse, it can be made sparse through basis transformation. The acquisition of such sparse signal is carried out through the second transformation using a rank deficient measurement matrix. The measurements reduction is the consequent of this transformation. This new framework for sparse signal acquisition is used in smart sensor design, where signals are under-sampled without losing the information content. A scenario where this becomes an effective data acquisition technique is when communication bandwidth is limited. Different from classical methods, where the Nyquist rate sampled signals are compressed prior to transmission, this compressed sensing technique optimizes the acquisition process and reduces the number samples. Signals like Synthetic Aperture Radar, where fine resolution signal characteristics can be captured only at high sampling rate, which may not be possible always, because of engineering or timing constraints, compressed sensing based acquisition and recovery techniques are utilized to capture these signals at sampling rate well below the recommended Nyquist rate [2].

The sampling theorem states that to avoid information loss and aliasing during sampling of any signal, sample the signal at least two times faster than the signal's bandwidth. In many emerging applications like magnetic resonance imaging, synthetic aperture radar, abundance of data generated by the data acquisition systems due to high Nyquist sampling rate, demands compression prior to store or transmission, to save communication bandwidth. Compressed sensing combines the sampling and compression into a single process, and uses non-adaptive linear projections to preserve the time domain characteristics of the signal; and the signal is later reconstructed from these projections using optimization techniques [3]. The transformation of compressed sensing from an algebra theory to a practical technique for sparse signal processing began with the work of Donoho, Candes, Romberg, Tao and et al [2], [4], [6], [7], [8].

A sparse signal $\mathbf{x} \in \mathbb{R}^N$ with K number of non zero values ($K \ll N$) is transformed into a projected space using a measurement matrix $\mathbb{A} \in \mathbb{R}^{M \times N}$ and then the signal is acquired with lesser number of samples M ($M < N$) as given in (1.1) [1].

$$\mathbf{y} = \mathbf{A}\mathbf{x}, \quad \mathbf{x} \in \mathbb{R}^N, \mathbf{A} \in \mathbb{R}^{M \times N}, \mathbf{y} \in \mathbb{R}^M \quad \|\mathbf{x}\|_0 = K \ll N, \tag{1.1}$$

The properties of a proper measurement matrix is described in [4]. The original sparse signal is recovered using the optimization method defined as (1.2)

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \,, \text{ subject to } \mathbf{y} - \mathbf{A}\mathbf{x} = 0 \tag{1.2}$$

Considering the computational limitation, if a finite error in reconstruction is tolerated, the sparse recovery problem can be represented as (1.3) [5].

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \,, \text{ subject to } \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \leq \varepsilon \tag{1.3}$$

The direct estimation of $\mathbf{x}$ through minimizing the ℓ_0 norm and satisfying the reconstruction criterion is Non-Polynomial time hard (NP-hard) problem. The reconstruction is also possible if ℓ_1 norm is used in place of ℓ_0 norm. However all matrices of type $A \in \mathbb{R}^{M \times N}$ do not qualify as measurement matrix. The reconstruction is guaranteed only if the measurement matrix satisfies the condition known as restricted isometric property as given in (1.4)

$$(1-\delta_K)\left\|\mathbf{x}\right\|_2^2 \leq \left\|\mathbf{A}\mathbf{x}\right\|_2^2 \leq (1+\delta_K)\left\|\mathbf{x}\right\|_2^2, \ \forall \mathbf{x} \in \Sigma_K \tag{1.4}$$

where, δ_K is a finite constant defined as restricted isometric constant [6] and $\mathbf{\Sigma}_X$ is the set of all K-sparse vectors. This is the upper bound for the difference in ℓ_2 norm of the measurement $\mathbf{y} = \mathbf{A}\mathbf{x}$ with respect to ℓ_2 norm of the original signal $\mathbf{x}$. This measurement restriction is a consequence of the null space requirement. The null space of the selected measurement matrix $\mathbf{A}$ should not contain any $2K$ sparse vectors, if the matrix is used to take measurements of the signal with sparsity $\leq K$ (1.5). This is a necessary condition for the sparse recovery.

$$Null(\mathbf{A}) \cap \mathbf{\Sigma}_{2K} = \emptyset \tag{1.5}$$

where $\mathbf{\Sigma}_{2K}$ is set of all $2K$ sparse signals. The probability of signal recovery can be estimated if the number of measurements M satisfies (1.6), as shown in [7].

$$if \quad M \geq \frac{C}{\delta^2} K \left(\ln(\frac{N}{K}) + \ln(\frac{2}{\epsilon}) \right) \quad then \quad p(\hat{\mathbf{x}} := \mathbf{x}) = 1 - \epsilon \tag{1.6}$$

where $p(\hat{\mathbf{x}} := \mathbf{x})$ is the probability of recovery, ϵ is a finite value, $\hat{\mathbf{x}}$ is the recovered sparse signal, $C > 0$ is a positive constant and $\delta > \delta_K$ is a finite constant [7]. The restrictions given in (1.4) and (1.5) are satisfied when columns of the measurement matrix are selected from independent and identical distribution (i.i.d.) of random vectors. The Bernoulli (1.7) and Gaussian (1.8) type random matrices satisfy these properties and are used in general as measurement matrix [4].

$$\mathbf{A} = \{a_{ji}\}, i = 1 \ldots N, j = 1 \ldots M, a_{ji} = \pm 1/\sqrt{M} \tag{1.7}$$

$$\mathbf{A} = \{a_{ji}\}, i = 1 \ldots N, j = 1 \ldots M, a_{ji} \in \mathcal{N}(0, 1/M) \tag{1.8}$$

Alternate way to confirm the validity of measurement matrix is described in [8]. The minimum number of *Linearly Dependent Columns* of $\mathbf{A}$ or alternatively indicated as the $spark(\mathbf{A})$ of the measurement matrix is within the range $[2K, M+1]$. Its value is estimated from the statement (1.9), where $\mu(\mathbf{A})$ is the column wise coherence defined as (1.10) and A_i, A_j are columns of the matrix $\mathbf{A}$.

$$spark(\mathbf{A}) \geq 1 + \frac{1}{\mu(\mathbf{A})}, \quad 2K \leq spark(\mathbf{A}) \leq M+1 \tag{1.9}$$

$$\mu(\mathbf{A}) = \max \frac{|\langle A_i, A_j \rangle|}{\sqrt{|\langle A_i, A_i \rangle \langle A_j, A_j \rangle|}}, \quad 1 \leq i, j \leq N, i \neq j \tag{1.10}$$

The original sparse vector is reconstructed from the measurement vector using the generalized approach given in (1.2). Considering the problem (1.2) the unknown $\mathbf{x}$ has two parts (i) the support of the vector $\mathbf{x}$, ($supp(\mathbf{x}) = \{i : \mathbf{x}_i \neq 0\}$) and (ii) the non-zero values over these supported locations. Two alternatives for finding the solution of the above problem are

1. Greedy Algorithms: Here the focus is given for finding the signal support, the non-zero values of $\mathbf{x}$ at the supported locations are easily determined by least squares method. These methods build up an approximation one step at a time by making locally optimal choices at each step.

2. Relaxation Methods: The second method do not consider the signal support and takes the unknown $\mathbf{x}$ as a vector ($\mathbf{x} \in \mathbb{R}^N$). By smooth approximation of the objective norm function $\|\mathbf{x}\|_0$, solution of $min \|\mathbf{x}\|_0$ is obtained using continuous function optimization techniques. These techniques solve a convex optimization problem whose minimizer approximates the target vector $\mathbf{x}$ [3].

Recent research on compressed sensing has widened the scope of its application from biomedical to astronomy [9], [10]. For each application domain unique recovery algorithms are designed specifically to meet the requirements. The CS recovery algorithms use theoretical foundation from various techniques of function optimization, regression, iterative methods, machine learning and artificial neural networks. The objective of developing new sparse recovery algorithms has changed from general algorithm, which deals with all kind of data to application specific algorithm taking advantage of the inherent features of the signals under consideration. As more and more signal restrictions are imposed for each application specific algorithm, the performance comparison of these becomes increasingly difficult, just by evaluating the results.

Large numbers of sparse signal reconstruction algorithms are developed in the recent time. The availability of large numbers of reconstruction algorithms create dilemma in choosing a particular method for a specific reconstruction application. The recovery algorithms are generally compared in terms of computational complexity, computational time, probability of recovery and recovery precision. Typically absolute Mean Squared Error (MSE) and relative MSE are used to compare the recovery precision of various sparse recovery algorithms. However, these two metric alone may not qualify to assess all algorithms. This thesis begins with an algorithm evaluation strategy by ranking the algorithms based on an observable similarity between the original and reconstructed signal.

The research work presented in this thesis has 5 major chapters. The chapter 2 presents a novel method for analysis and ranking of sparse recovery algorithms. The chapter 3 presents a frame work for improving the accuracy of sparse signal recovery using iterative residue estimation, proximal projection and segmented thresholding. Also, the development of two sparse signal recovery methods based on the proposed frame work. The chapter 4 presents the function dictionary based implementation of the proposed sparse recovery algorithm for low profile computing platforms. The chapter 5 presents the capability evaluation of an IoT based computing platform for implementation of the proposed sparse reconstruction algorithm. The chapter 6 presents the implementation of the proposed algorithm for real-time distributed data acquisition of naturally sparse signals. Also presents an energy efficient routing method for networked data collection. The research summary and conclusion is given in Chapter 7. The future directions for the work is given in Chapter 8, followed by list of publications related to this thesis.

Chapter 2

New Metric for Sparse Recovery Algorithm Evaluation

The sparse signal recovery is of great interest in compressed sensed data recovery. Many sparse recovery algorithms were developed in the last decade. However, selection of the appropriate recovery algorithm is an important matter of concern in many applications. The recovery algorithms are generally compared in terms of computational complexity, computational time, probability of recovery and recovery precision. Typically absolute Mean Squared Error (MSE) and relative MSE are used to compare the recovery precision of various sparse recovery algorithms. However, these two metric alone are not sufficient to assess all algorithms. This chapter presents an algorithm evaluation strategy by ranking the algorithms concerning an observable similarity between the original and the reconstructed signal. A recovery similarity measure and an empirically defined factor for comparing the performance of sparse recovery algorithm is proposed and described.

2.1 Introduction

Sparse estimation algorithms are extensively used for sparse discriminant analysis and classification of high-dimensional data [11]. Any sparse recovery algorithm should possess specific vital characteristics. Moreover, while using any recovery algorithm for any specific application, the algorithms should satisfy the prescribed selection criteria. In general, the performance metrics for signal processing algorithms are given in terms of the data throughput, real-time computation latency, processor load, and power dissipation [12]. However, considering the sparse recovery algorithms in compressed sensing recovery perspective, the measure of interest is the ability to reconstruct the original signal perfectly. The compressed sensing is used as a signal acquisition method when the engineering limitations do not support the conventional Nyquist sampling rate. Also, sparse recovery algorithms are used in cases where the measurements are taken at a reduced sampling rate, and there is a requirement to reconstruct the original signal in its most acceptable form [13]. In these applications, if the priority is on the signal recovery precision than the complexity or the computational time of the algorithm, the mean squared error of the recovered signal is used. However, there can be instances when MSE of the recovered signal using first algorithm is low compared to the second, but the signal appears more similar to the original in the second case. This kind of observation is due to the averaging effect of the MSE measure. The performance analysis of every new algorithm introduced is given in terms of the lists of error measures, like mean squared error, relative error and SNR. Some of the standard metrics that are used to assess recovery algorithm are CPU time vs. vector length, CPU time vs. sparsity of vector, and MSE vs. measurement sample number. etc [14]. In some other applications MSE vs. sparsity of vector, covariance of original and recovered signal vs. measurement sample number, covariance of original and recovered signal vs. sparsity and the phase transition diagram K/M vs. M/N [15], where N is the length of the sampled signal, K is the sparsity of the signal and M is the number of measurements are commonly used. These performance evaluation graphs of the algorithms are left to the reader's interpretation. It is becoming increasingly challenging to compare the algorithms just by interpreting the results given in graphs and tables unless the user does simulations of the algorithm.

Here a method to rank the sparse recovery algorithm in a logical order in terms of signal reconstruction efficiency is presented. The estimation of the performance measure proposed here is done in two steps. In the first step, a measure of similarity of the recovered signal with the original signal is determined, and the second step determines an unique value from this similarity measure. This chapter is organized as follows, section 3.2 gives an overview of compressed sensing and sparse recovery algorithms. The performance measure of sparse recovery algorithms is given in section 3.3. The data set and the simulation details are given in section 3.4, followed by the chapter summary.

2.2 Sparse Signal Measurement and Recovery

Compressed Sensing: Given a sparse signal $\mathbf{x} \in \mathbb{R}^N$ with sparsity $\ell_0(\mathbf{x}) = K << N$ or a compressible signal $\mathbf{x} \in \mathbb{R}^N$ can be measured with lesser measurements $M < N$ compared to Nyquist rate as given in (1.1) using a suitable measurement matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$. The original signal can be faithfully reconstructed (1.2) from the measurement $\mathbf{y} \in \mathbb{R}^M$ if the measurement matrix $\mathbf{A}$ satisfies the condition (1.4). Also, the measurement matrix $\mathbf{A}$ should have null space property: The kernel of the matrix $\mathbf{A}$ should not have any signal with $2K$ sparsity when the measurement is performed for a signals with sparsity $\leq K$ (2.1).

$$\ker(\mathbf{A}) \cap \{\mathbf{z} \in \mathbb{R}^N : \|\mathbf{z}\|_0 = 2K\} = \emptyset \tag{2.1}$$

where $\ker(\mathbf{A})$ over the field $\mathbb{R}$ is a linear subspace of $\mathbb{R}^N$, which always contains the zero vector such that $\mathbf{Az} = \mathbf{0}, \quad \forall \mathbf{z} \in \mathbb{R}^N$. The measurement matrix with columns selected from independent identical distribution (iid) Gaussian random vectors or Bernoulli random vectors satisfy the above conditions (1.4) and (2.1). The sparse signal can be recovered from the compressed measurements using the convex optimization given in (2.2).

$$\mathbf{x}^* = arg \min_{\mathbf{x} \in \mathbb{R}^N} J(\mathbf{x}) \text{ s.t. } \|(\mathbf{y} - \mathbf{Ax})\| \leq \epsilon \tag{2.2}$$

where $J(\mathbf{x})$ is sparsity inducing function like ℓ_0 norm and ϵ is the maximum error acceptable in signal reconstruction. There are varieties of sparse signal reconstruction algorithms developed based on this idea. The sparse recovery algorithms can be classified as function minimization based methods, greedy matrix minimization based methods or Bayesian estimation based methods. The function minimization based methods use ℓ_0 or ℓ_p, $p \in [0, 1]$ or ℓ_1 norm as sparseness inducing function and the error estimate in ℓ_2 norm constrains the solution.

As the focus of discussion is on the performance measure of sparse recovery algorithms, 24 algorithms from 8 different categories are selected for analysis. The salient features of those selected algorithms are briefly discussed here.

Set 1: Initially the greedy matrix minimization based ℓ_1 solutions were used for sparse signal recovery. The operations in these algorithms can be written as (2.3) [16], where Ω is the index of non zero values of vector $\mathbf{x}$. Ω is updated in every iteration and a sub matrix $\mathbf{A}_\Omega$ of $\mathbf{A}$ is formed by selecting columns indexed by Ω.

$$\{s\} = \arg\max_j [\mathbf{A}^\dagger(\mathbf{y} - \mathbf{A}\mathbf{x}^k)]_j, \quad \Omega = \Omega \cup \{s\},$$

$$\mathbf{x}_i^{k+1} = \mathbf{A}_\Omega^\dagger \mathbf{y}, i \in \Omega, \quad \mathbf{x}_i^{k+1} = 0, i \notin \Omega, \tag{2.3}$$

The matrix $\mathbf{A}^\dagger$ indicates pseudo inverse $(\mathbf{A^T A})^{\mathbf{-1}}\mathbf{A^T}$ and $\mathbf{A}_\Omega^\dagger$ indicates pseudo inverse of $\mathbf{A}_\Omega$. The initial value for iteration ($\mathbf{x}^0$) is set as zero vector. The iteration is done for maximum possible sparsity K. Many variations of this basic idea are developed into different algorithms, such as few listed below.

- Orthogonal matching pursuit (OMP) [17].
- Generalized orthogonal matching pursuit (GOMP) [18].
- Compressive sampling matching pursuit (CoSAMP) [19].
- L1 regularized least square (L1LS).
- Your algorithm for L1 (YALL) [20].

Set 2: ℓ_1 minimization with thresholding gives sparser solution. Few methods based on this idea is listed below. For evaluation, the threshold of these algorithms are set to $\tau = 10^{-3}$.

- Backtracking iterative hard threshold (BIHT) [21].

- Fast iterative shrinkage thresholding (FISTA) [22].

Set 3: The classical Lagrangian constrained minimization based sparse recovery methods were later used such as few listed below.

- Prime Augmented Lagrangian Multiplier (PALM) [23].

- Dual Augmented Lagrangian Multiplier (DALM) [24].

- Primal and Dual augmented Lagrangian.

- Radial basis function approximation sparse recovery algorithm (RASR) [25].

In RASR ℓ_0 function is approximated using inverted Gaussian bell function and the ℓ_2 norm of the reconstruction error is used as the constraint as described in **algorithm-1**. For evaluation, the parameter is set as $\mu_0 = 0.05$. The exit condition is set to $\|\mathbf{x}_{k+1} - \mathbf{x}_k\|_2 \leq 10^{-4}$.

Algorithm 1
RASR: radial basis approximation sparse recovery, a Lagrangian constrained minimization based method

Require: $A \in \mathbb{R}^{M \times N}$, $\mathbf{y}$, δ, L,
Task: $\min_{\mathbf{x}} \|\mathbf{x}\|_0$ subject to $\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 = 0$
Initialization: $x_0 = \mathbf{A}^{\dagger}\mathbf{y}$, $\sigma_0 = 2\max\{x_0\}$
for $k = 1 \dots L$ **do**
$x_i(k) = x_i(k) - \mu_0 x_i(k) \exp\left(\frac{-x_i^2(k)}{2\sigma_k^2}\right)$
  minimize ℓ_2 error
  for $j = 1 \dots M$ **do**
$x(k+1) = x(k) + \frac{A_j^T}{\|A_j\|^2}\left(y_j - A_j x(k)\right)$
  end for
$x(k) = x(k+1)$, $\sigma_{k+1} = \sigma_k \delta$
end for
Output: $x(k)$

Set 4: ℓ_0 based optimization was infeasible as the function is not differentiable at 0. Later, sparse recovery using function approximations of ℓ_0 norm were used such as few listed below.

- Smooth L0 (SL0) [26].
- X-L0 E-L0 sparse recovery (XEL0) [27].

The ℓ_0 function of the sparse vector is computed using inverted Gaussian function $\ell_0(x) = 1 - exp(-x^2/2\sigma)$, where σ controls the level of approximation. The error $\mathbf{y} - \mathbf{Ax}$ is taken as the minimization constraint. For evaluation the values are set as $\sigma_{min} = 10^{-8}$ and σ decrease factor as $\delta = 0.95$. The ℓ_0 minimization method is given in **algorithm 2**.

Algorithm 2
Smooth L0, ℓ_0 minimization based method

Require: $\mathbf{A}$, $\mathbf{y}$, L , μ
- **Task:** Solve: $\min_{\mathbf{x}} \|\mathbf{x}\|_0$ subject to $\mathbf{y} = \mathbf{Ax}$
- **Initialization:** $x(0) = \mathbf{A}^{\dagger}\mathbf{y}$
- **for** $\sigma > \sigma_{min}$ **do**
  - **for** $j = 1 \ldots L$ **do**
    - $x_i(k) = x_i(k) - \mu_0 x_i(k) \exp\left(\frac{-x_i^2(k)}{2\sigma_k^2}\right)$
    - $x(k+1) = x(k) - A^{\dagger}(Ax(k) - y)$
  - **end for**
  - $x(k) = x(k+1)$, $\sigma_{k+1} = \sigma_k \delta$
- **end for**
- **Output:** $x(k)$

Set 5: Iterative re-weighted least square (IRLS) [28] is a ℓ_p norm based method generally taken as the bench mark algorithm for comparison. The method is described in **algorithm 3**.

Algorithm 3
Iterative reweighted least square, ℓ_p minimization based method

Require: $\mathbf{A}$, $\mathbf{y}$
- **Task:** Solve: $\min_{\mathbf{x}} \|\mathbf{x}\|_p$ subject to $\mathbf{y} = \mathbf{Ax}$
- **Initialization:** $\mathbf{W} = \mathbf{I}$
- **for** $k = 1 \ldots L$ **do**
  - $\mathbf{x} = \mathbf{WWA^T}(\mathbf{AWWA^T})^{\dagger}\mathbf{y}$
  - $\mathbf{W} = diag(\mathbf{x}_i^p), i = 1 \ldots N$
- **end for**
- **Output:** $\mathbf{x}$

Set 6: In ℓ_1 function minimization based method, basis pursuit [29] is widely used as the benchmark algorithm, where the indexes of the maximum support values are estimated and the sparse signal is reconstructed using the pseudo inverse of the submatrix of $\mathbf{A}$ indexed using the estimated locations. The algorithm is described in **algorithm 4**. The ℓ_1 minimization methods used here for comparison are listed below.

- Basis pursuit (BP) [29].
- Lasso [30].
- homotopy [31].

Algorithm 4

The basis pursuit, the ℓ_1 minimization based algorithm

Require: $\mathbf{A}$, $\mathbf{y}$, τ

Task: Solve: $\min_{\mathbf{x}} \|\mathbf{x}\|_0$ subject to $\mathbf{y} = \mathbf{Ax}$

Initialization: $r = y, \Omega = \emptyset$

for $r^T r > \tau$ **do**

  $\Omega = \Omega \cup index(\max(A^T r))$

  $r = y - A_\Omega A_\Omega^\dagger y$

end for

$x = A_\Omega^\dagger y$

Output: $x(k)$

Set 7: The Expectation maximization Gaussian mixture approximate message passing (EGAmp) [32] is taken as a candidate Bayesian method. The format of the Bayesian estimation based algorithms can be written as (2.4) [33].

$$\mathbf{x}^* = \max_{\mathbf{x} \in \mathbb{R}^N} p(\mathbf{x} \mid \mathbf{y}) = \min_{\mathbf{x} \in \mathbb{R}^N} (-\log p(\mathbf{y} \mid \mathbf{x}) - \log p(\mathbf{x})) \tag{2.4}$$

Set 8: The proximal projection based methods are latest addition to the sparse signal recovery. A candidate algorithm Iterative proximal projection with smoothly clipped absolute deviation (IPP-scad) [37] is described in **algorithm 5**. Other projection methods used here for performance comparison are listed below.

- Smoothly clipped absolute deviation (SCAD) [34], [35].

- Successive concave sparse approximation(SCSA) [36].
- Iterative sparsification projection with SL0 (ISP-SL0) [37].
- Iterative sparsification projection with Imat (ISP-Imat) [37].
- Iterative proximal projection with hard thresholding (IPP-hrd) [37].
- Iterative proximal projection with mcp (IPP-mcp) [37].
- Iterative proximal projection with SCAD (IPP-scad) [37].

Algorithm 5
Iterative proximal projection based algorithm

Require: $\mathbf{A}$, $\mathbf{y}$, L , δ, θ, τ, γ, $\mathbf{M} = (I + \gamma A^T A)^{-1}$
Task: Solve: $\min_{\mathbf{x}} \|\mathbf{x}\|_0$ subject to $\mathbf{y} = \mathbf{Ax}$
Initialization: $x_0 = \mathbf{A}^{\dagger}\mathbf{y}, \quad v = 0$
while $\theta > \tau$ **do**
  for $j = 1 \ldots L$ **do**
    $x_k = Threhsold(\theta, x_k + w(x_k - x_{k-1}))$
    $e = y - Ax_k$
    while $\|e\|^2 - \varepsilon^2 > \tau$ **do**
      $e = e + \frac{1}{\gamma} v$
      $if : \|e\| > \varepsilon ? : e = \frac{e}{\|e\|}\varepsilon$
      $x_{k+1} = M(x_k + \gamma A^T(y - e + \frac{1}{\gamma} v))$
      $e = y - Ax_{k+1}$
      $v = v - \gamma(e_{old} - e)$
    end while
    $x_k = x_{k+1}$
  end for
  $\theta = \delta\theta$
end while
Output: x_k

Further discussions on the signal recovery algorithms are limited to recovery performance evaluation and the limitations of the conventional performance evaluation metric. In the following section a method to improve the performance comparison is discussed.

2.3 Recovery Performance Measures

Based on the general idea discussed, large number of sparse signal recovery algorithms have been designed and published. In all the cases, the theoretical explanation and the proof of sparse recovery algorithms are followed by the experimental evidence and graphs illustrating the advantages over the existing algorithms. The metric used for quantifying the advantages of the algorithms are given in terms of the mean squared error (2.5) or relative mean squared error (2.6).

$$MSE = \frac{1}{N}\sum_{i=i}^{N}(\mathbf{x}_i - \hat{\mathbf{x}}_i)^2 \tag{2.5}$$

$$relMSE(\hat{\mathbf{x}}, \mathbf{x}) = \frac{\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2}{\|\mathbf{x}\|_2^2} \tag{2.6}$$

where, $\mathbf{x}$ is the original sparse signal and $\hat{\mathbf{x}}$ is the signal recovered from the measurement $\mathbf{y}$. In some evaluations the SNR of the recovered signal is computed using (2.7) is used.

$$SNR = 10\log\left(\frac{\|\hat{\mathbf{x}}\|_2^2}{\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2}\right) \tag{2.7}$$

where, the signal recovery error $\|\mathbf{x} - \hat{\mathbf{x}}\|_2$ is considered as the noise. The support recovery error is computed as (2.8) and the the probability of exact signal recovery is evaluated as (2.9).

$$SupError(\hat{\mathbf{x}}, \mathbf{x}) = 1 - \frac{\|supp(\mathbf{x}) \cap supp(\hat{\mathbf{x}})\|_0}{\max(\|\mathbf{x}\|_0, \|\hat{\mathbf{x}}\|_0)} \tag{2.8}$$

$$p(\hat{\mathbf{x}} := \mathbf{x}) = \frac{\|supp(\mathbf{x}) \cap supp(\hat{\mathbf{x}})\|_0}{\max(\|\mathbf{x}\|_0, \|\hat{\mathbf{x}}\|_0)} \tag{2.9}$$

where $supp(.)$ gives the location index set of the non zero elements and $\|.\|_0$ is the number of non zero elements. The complexity of sparse signal recovery algorithms are compared in terms of the number of iterations or the total computational time. To analyse the robustness of the sparse recovery algorithms in presence of noise, the compressed sensed measurement $\mathbf{y} = \mathbf{A}\mathbf{x}$ is perturbed with Gaussian noise $\mathbf{w}_n$

$\mathcal{N}(0, 0.1)$ of relative strength varying from -40dB to -10dB ($n_L = 0.01$ to 0.20). The signal is recovered from the noisy measurement $\mathbf{y}_n$ (2.10) using various algorithms.

$$\mathbf{y}_n = \mathbf{A}\mathbf{x} + n_L \frac{\|\mathbf{A}\mathbf{x}\|}{\|\mathbf{w}_n\|}\mathbf{w}_n \tag{2.10}$$

A detailed study on the effect of noise in the recovery of compressed measurements is presented in [38]. Similarly, the perturbation analysis of the sparse recovery is given in [39]. From these studies, it is observed that the relative absolute error (2.6) is close to zero for greedy sparse recovery algorithms, but the probability of exact recovery (2.9) is unacceptable. Moreover, it is not easy to arrive at a conclusive comparison of algorithms from this list of different performance measures. And, the ranking of the algorithms depends upon the type of analysis performed. It is inevitable to have a generalized performance comparison for all types of sparse recovery algorithms. A novel metric for generalized comparison of various types of sparse recovery algorithms is discussed in the following section.

2.3.1 Sparse Recovery Limit

An empirical function $Sm(\hat{\mathbf{x}}, \mathbf{x})$ (2.11) using (2.6) the relative MSE ($relMSE(\hat{\mathbf{x}}, \mathbf{x})$) and the exact support recovery probability $p(\hat{\mathbf{x}} := \mathbf{x})$ (2.9) is proposed. This is designated as the signal similarity measure between the original signal and the reconstructed signal in terms of the relative error and probability of recovery. The function is formed in this way so that the resulting value is always less than unity $Sm(\mathbf{x}, \hat{\mathbf{x}}) \in [0, 1]$.

$$Sm(\hat{\mathbf{x}}, \mathbf{x}) = \frac{1}{2}\begin{cases} p(\hat{\mathbf{x}} := \mathbf{x}) + relMSE(\hat{\mathbf{x}}, \mathbf{x}) + 1, & e \leq 1 \\ p(\hat{\mathbf{x}} := \mathbf{x}) + \frac{1}{relMSE(\hat{\mathbf{x}}, \mathbf{x}) + 1}, & e > 1 \end{cases} \tag{2.11}$$

The variation of $Sm(.)$ with sparsity K is expected to be distinct for every algorithms, so that a relative comparison is possible. However, the plot of this function $Sm(\hat{\mathbf{x}}, \mathbf{x})$ does not directly give a numerical value for the performance comparison and this graph needs to be interpreted to obtain a quantitative estimate for the comparison. The analysis should be inclusive of all the performance measures, to get a comprehensive ranking. Factors used in the performance measure should also include the ratio of maximum measurable sparsity to the sparse vector size (K_{max}/N) [4] and the ratio of number of measurements to the vector size (M/N). A quantifiable value from the above signal similarity measure graph is obtained as the maximum value of the sparsity K_{max} when the similarity $Sm(\hat{\mathbf{x}}, \mathbf{x})$ reduces to 0.9 or any other value (c) as desired. A graphical representation of obtaining K_{max} from $Sm(\hat{\mathbf{x}}, \mathbf{x})$ graph is shown in Figure 2-1, where K_{max} is the maximum sparsity for which the recovered signal $\hat{\mathbf{x}}$ has a visual similarity index of $c = 0.8$ or more with respect to the original signal $\mathbf{x}$.

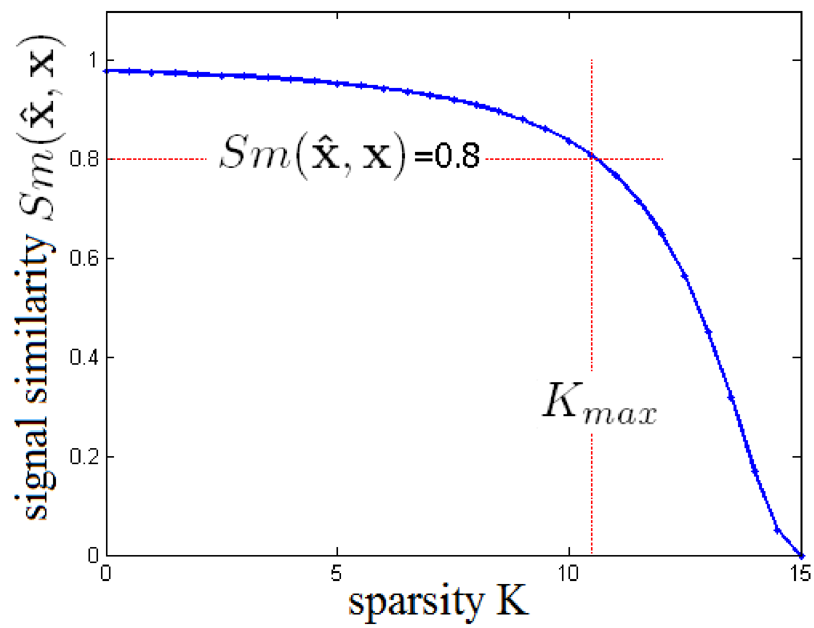


Figure 2-1: The graphical representation of obtaining K_{max} from $Sm(\hat{\mathbf{x}}, \mathbf{x})$ vs sparsity K curve.

To have a relative comparison across all sparse vector size and to maintain the proposed numerical measure within a bound, the normalized value of the measurement size M with respect to the sparse vector size (M/N) and the normalized value K_{max}/M is taken as reference. The value $K_{max} = 0$ is assumed if the graph of $Sm(\hat{\mathbf{x}}, \mathbf{x})$ is below the required mark. Using these two values a new comprehensive metric designated as **sparse recovery limit** of the algorithm ξ is defined (2.12).

$$\text{Sparse Recovery Limit} \qquad \xi|_c = \frac{1}{2}(1 - \frac{M}{N} + \frac{K_{max}}{M})\Big|_{Sm(\hat{\mathbf{x}},\mathbf{x})=c} \tag{2.12}$$

where c is the minimum value of the signal similarity $Sm(\hat{\mathbf{x}}, \mathbf{x})$ (2.11) as indicated and the factor $1 - \frac{M}{N}$ is included to limit the maximum value of ξ to 1. This value ξ is a measure of limit of recovery performance. This measure gives a comprehensive comparison of the absolute ability of the algorithm in sparse signal reconstruction, since it incorporates all of the otherwise individual performance measures; and has the following salient features.

1. The probability of exact signal recovery $p(\hat{\mathbf{x}} := \mathbf{x})$.
2. The relative MSE.
3. The sparse vector size N.
4. The measurement vector size M.
5. The maximum sparsity supported K_{max}.
6. The data similarity between original and reconstructed signal $Sm(\hat{\mathbf{x}}, \mathbf{x})$.
7. The user defined threshold limit for comparison c.
8. Inherent normalization of all measures.
9. Clearly defined measure bound $0 < \xi < 1$.
10. The value $\xi = 1$ indicates the best achievable sparse recovery algorithm.

2.4 Performance Evaluation

Two types of synthetic sparse signals are generated for algorithm simulation. i) The sparse signals with non zero values at random locations and ii) Locally sparse pulse. Null vectors of length 50 are generated using MATLAB function and then made sparse by adding random values at random locations. The number of locations where random values are inserted are increased from 1 to 15, to simulate different sparsity level ($K = 1 \dots 15$). For generating the locally sparse pulse, the null vectors of length 50 are generated and are transformed into sparse pulse by adding random values at K consecutive locations. As in the earlier case the number of locations are increased from 1 to 15. The random measurement matrix $\mathbf{A}$ of size 30×50 are generated by arranging 50 columns of i.i.d. Gaussian random vectors of length 30. The sparse vectors ($\mathbf{x}$) are compressed sensed using the measurement matrix ($\mathbf{y} = \mathbf{A}\mathbf{x}$). The sparse vectors are then reconstructed using the 24 different sparse recovery algorithms selected from different categories. The parameter values used in the different algorithms are given identical values regarding their significance for a fair comparison. The performance measure computed are absolute mean squared error given in (2.5), relative reconstruction error given in (2.6), the probability of exact support recovery given in (2.9), signal to noise ratio in the reconstructed signal given in (2.7), and the computational time in addition to the proposed signal similarity measure $Sm(.)$ given in (2.11) and the overall measure the sparse recovery limit of the algorithm ξ given in (2.12). The compressed measurement and reconstruction were performed 50 times using different measurement matrices for every sparse vector, and the averaged results are presented. The simulations were done using $MATLAB$ running on Intel-i3 1.9 GHz dual-core processor with 12 GB RAM and 64 bit MS Windows 8 operating system.

2.4.1 Results

The performance measure of the algorithms compared is given in Table 2.1 and Table 2.2 for the sparse pulse and sparse spike signals with sparsity $K = 5$. A careful reading of the values of relative error and the support recovery error suggests that Orthogonal matching pursuit [17], Generalized orthogonal matching pursuit, Basis pursuit [29], and Lasso homotopy methods [30] are the best as these algorithms give reconstruction error in the order of 10^{-30} with no support error for the two types of sparse signals simulated. These measurements are averaged over entire span of the sparse signal. However, there are advanced reconstruction methods listed in the same table. But these algorithms are not giving the specific numbers in terms of the reconstruction errors alone. Furthermore, when noise is simulated, the relative error in the reconstructed signal varies significantly.

The sparse pulse signals recovered using various algorithms in presence of acquisition noise are shown in Figure 2-2, where the signals are compared in terms of SNR (2.7), but the inference is not conclusive. The $relMSE(\hat{\mathbf{x}}, \mathbf{x})$ (2.6) for some of the algorithms compared is given in Figure 2-3. It is difficult to conclude which of these is the best one without looking at the probability of signal recovery graph given in Figure 2-4 and Figure 2-5. However, there is no characteristic difference between the graphical results obtained. The results are indistinguishable from one another for the candidate algorithms from different classes (ℓ_0, ℓ_p, ℓ_1) studied. Also, the inference from noise perturbation vs relative MSE as shown in Figure 2-6 does not provide conclusive remark. So for a generalized comparison of the performance measures the new metric sparse recovery limit (2.12) for a given signal similarity measure ($\xi \mid_{Sm(\hat{\mathbf{x}},\mathbf{x})=c}$) is used.

The signal similarity measure $Sm(\hat{\mathbf{x}}, \mathbf{x})$ is computed using the proposed method (2.11). The data set used is random sparse signal and the sparsity is varied from $K = 1 \dots 15$. The setup configuration used in the simulation are as discussed earlier. The graphs of $Sm(\hat{\mathbf{x}}, \mathbf{x})$ obtained for 24 algorithms are given in Figures 2-7, 2-8 and 2-9. The graphs are generated for different measurement noise levels with relative strength 0.01, 0.02 and 0.03. Evidently, the characteristics of these function curves are different for different classes of algorithms. As expected, the signal similarity measure is reduced significantly in classical optimization methods like PALM and DALM (Figure 2-7 and Figure 2-8). The signal similarity measure graphs of ℓ_0, ℓ_p and ℓ_1 methods show similar characteristic curves (SL0, IRLS, BP in Figure 2-8). The signal similarity measure graph of projected gradient methods are distinct from other classes of algorithms and the measure is maintained high for large sparsity numbers as shown in Figure 2-9.

The performance comparison of the algorithms is done by computing the sparse recovery limit of the algorithm ξ from the $Sm(\hat{\mathbf{x}}, \mathbf{x})$ graphs using the proposed method (2.12). The ξ-metric is computed for $Sm(\hat{\mathbf{x}}, \mathbf{x}) = 1$, $Sm(\hat{\mathbf{x}}, \mathbf{x}) \geq 0.9$ and $Sm(\hat{\mathbf{x}}, \mathbf{x}) \geq 0.8$. Also, the values obtained for the noisy and noise free cases are given in Table 2.3 and Table 2.4. The tables clearly show that many of the algorithms have poor value $\xi = 0.2$ when the measurements are noisy. Interestingly the smoothly clipped absolute deviation improves the signal recovery when noise is added to the measurement (SCAD in Figure 2-8). Amongst all the algorithms compared, the recently published iterative proximal projection [37] shows the best sparse recovery limit value ($\xi \mid_{Sm=0.9} = 0.416$). The variation of signal similarity measure of the reconstructed signal when the measurement is perturbed with noise of relative strength -40 dB to -10 dB is shown in Figure 2-10. It can also be seen that the signal similarity graph of ℓ_0, ℓ_p and ℓ_1 based algorithms have same profile, when the measurements are noisy.

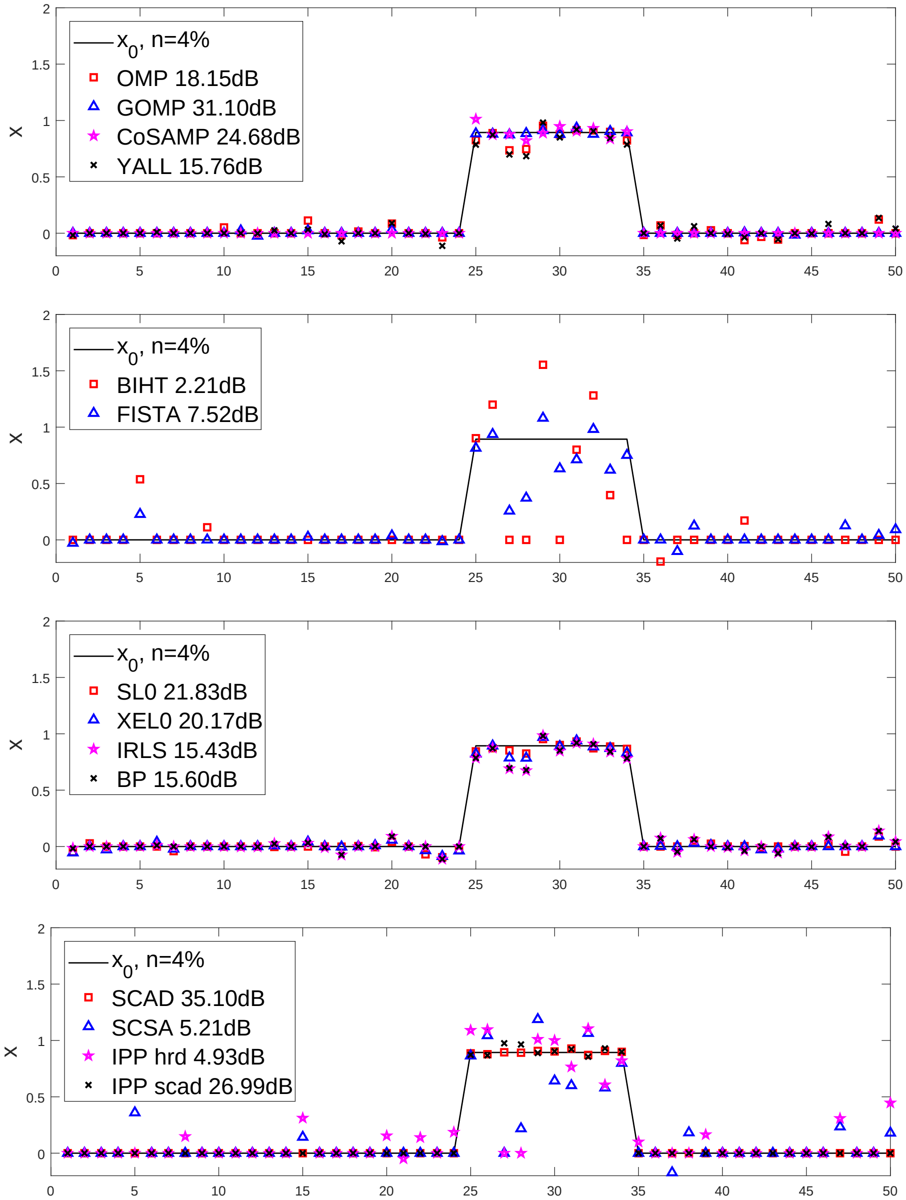


Figure 2-2: Spares pulse signal sampled from i.i.d Gaussian noisy measurements and reconstructed using various algorithms.

Note: Sparse pulse signal $\mathbf{x}_0$ *(length* $N = 50$*, sparsity* $K = 10$*,* $\max \mid \mathbf{x}_0 \mid = 1$*) sampled using* $M = 30$ *i.i.d Gaussian noisy measurements (noise level* $n = 4\%$*). The signal is then recovered using various algorithms. The quality of reconstruction is measured in SNR. Based on the computed SNR of the reconstructed signal, it can be seen that Smoothly Clipped Absolute Deviation (SCAD) based thresholding gives better signal reconstruction when the measurements (*$M = 30$*) are noisy.*

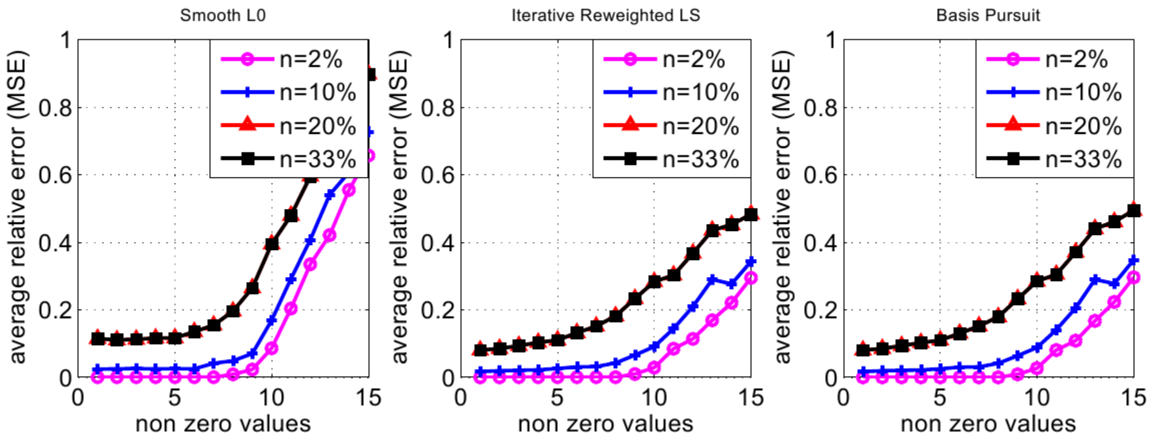


Figure 2-3: The relative MSE in sparse vector reconstruction using ℓ_0, ℓ_p and ℓ_1 based minimization algorithms from perturbed measurements.

Note: The sparse vector (length $N = 50$, sparsity $K = 1$) is measured with $M = 30$ samples and then reconstructed using ℓ_0 (SL0), ℓ_p (IRLS) and ℓ_1(Basis Pursuit) minimization algorithms. The relative MSE in signal reconstruction for these algorithms, when the measurement is perturbed with Gaussian noise of relative strength $n = 2\%$, $n = 10\%$, $n = 20\%$ and $n = 33\%$ are shown in the corresponding graphs.

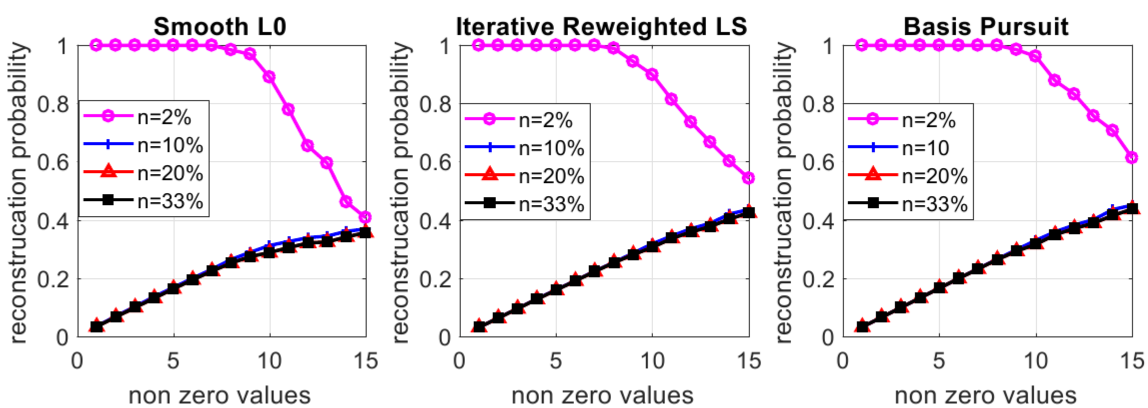


Figure 2-4: The probability of reconstruction of sparse vector using ℓ_0, ℓ_p and ℓ_1 based minimization algorithms from perturbed measurements.

Note: The sparse vector (length $N = 50$, sparsity $K = 1$) is measured with $M = 30$ samples and then reconstructed using ℓ_0 (SL0), ℓ_p (IRLS) and ℓ_1(Basis Pursuit) minimization algorithms. The probability of exact signal reconstruction for these algorithms, when the measurement is perturbed with Gaussian noise of relative strength $n = 2\%$, $n = 10\%$, $n = 20\%$ and $n = 33\%$ are shown in the corresponding graphs.

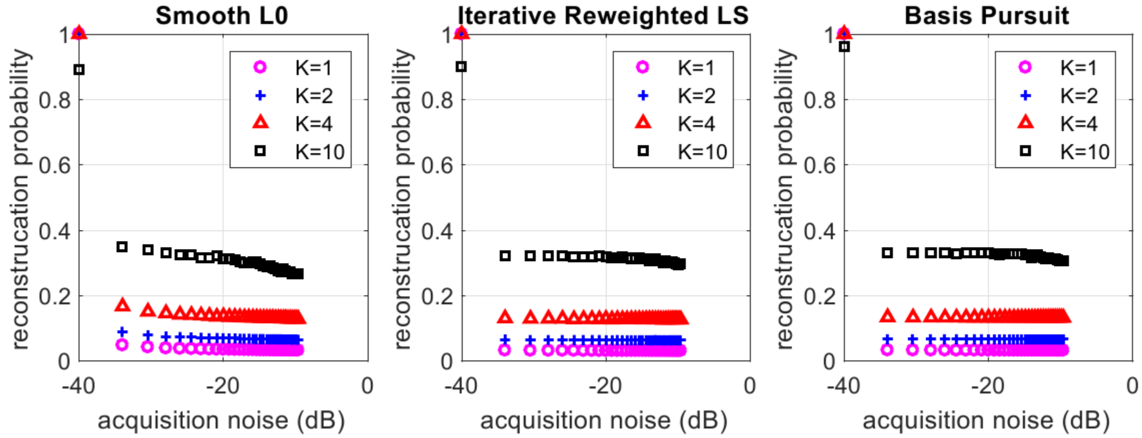


Figure 2-5: The probability of reconstruction of sparse vector using ℓ_0, ℓ_p and ℓ_1 based minimization algorithms for various noise levels.

Note: The sparse vector (length $N = 50$, sparsity $K = 1, 2, 4, 10$) is measured with $M = 30$ samples and then reconstructed using ℓ_0 (SL0), ℓ_p (IRLS) and ℓ_1(Basis Pursuit) minimization algorithms. The probability of exact signal reconstruction for these algorithms, when the measurement is perturbed with Gaussian noise of relative strength $-40dB$ to $-10dB$ is shown.

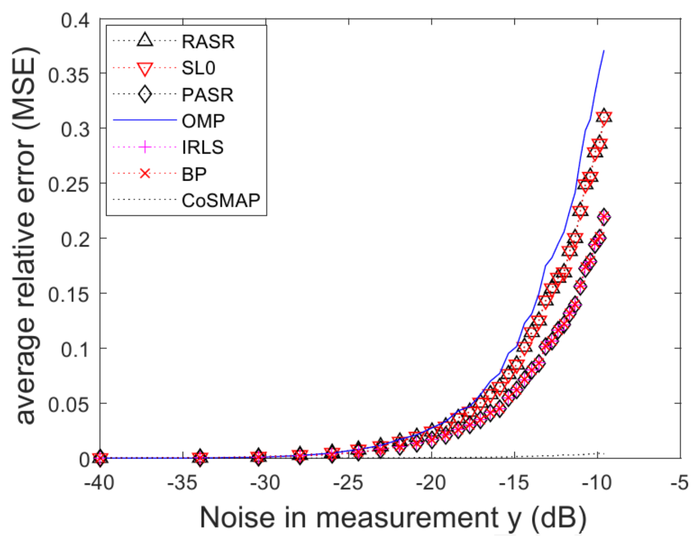


Figure 2-6: The relative MSE in sparse vector reconstruction using ℓ_0, ℓ_p and ℓ_1 based minimization algorithms for various noise levels.

Note: The sparse vector ($N = 50$, sparsity $K = 1$) is measured with $M = 30$ samples and then reconstructed using few candidate algorithms discussed. The relative MSE in signal reconstruction when measurement is perturbed with Gaussian noise of relative strength $-40dB$ to $-10dB$ is shown.

2.4.2 Observation

The results presented in Table 2.3 and Table 2.4 can be classified into four cases, namely case 1: noise free random sparse signal recovery, case 2: noisy random sparse signal recovery, case 3: noise free discrete pulse recovery, and case 4: noisy discrete pulse recovery. Prima facie, all the cases considered the highest value for the sparse recovery limit of the algorithm ξ is achieved by the iterative proximal projection algorithm with smoothly clipped absolute deviation thresholding function [37]. In fact, this is the ground truth known even before obtaining the new metric. This observation confirms the validity of the proposed metric. It can also be observed that the values in case 4 are lesser compared to the corresponding values in other 3 cases. In fact, the pulse recovery in presence of measurement noise (case-4) is most challenging for all algorithms. In the noisy cases (case 2 and 4), the value of sparse recovery limit ξ is almost same for most of the algorithms. The reality is that the performance of many of the sparse recovery algorithms are impeded by the noise in measurement.

After analyzing the data given in Table 2.3 and Table 2.4 critically, it can be seen that most of the algorithms compared give sparse recovery limit in the range ($0.20 \leq \xi \leq 0.35$) when the measurements are noisy; And the projected gradient based methods give better performance ($0.33 \leq \xi \leq 0.45$). It can be inferred directly from this comparison of sparse recovery limit ξ of the algorithm that the projected gradient methods give relatively better signal reconstruction performance when the measurements are noisy. However, this is claimed in the paper [37] but, a quantitative measure is not known until this new measure is used for comparison. It is also noted that this kind of direct inference of relative performance of the sparse recovery algorithms cannot be done by analyzing the conventional measures given in Table 2.1 and Table 2.2.

2.5 Chapter Summary

Many sparse signal recovery algorithms have been proposed in the last two decades. Furthermore, many newer algorithms have been developed, and research in this area continues. Moreover, these algorithms are presented with their particular merits and claim. The plethora of sparse recovery algorithms with different characteristics creates a dilemma while choosing the suitable one for a particular application. The conventional metric used to compare the sparse signal recovery algorithms are relative MSE and probability of support recovery. These two metrics are not standalone and need to be interpreted together. Here a method to evaluate the sparse signal recovery performance of the algorithms by using these two metrics to compute signal similarity between the original signal and the reconstructed signal is proposed. This chapter present two performance characterization functions indicated as signal similarity measure $Sm(\hat{\mathbf{x}}, \mathbf{x})$ and sparse recovery limit ξ of the algorithm to relatively compare the performance of sparse signal recovery algorithms. The computation time of the algorithms needs to be compared separately and is not included in the empirical function. The algorithms are evaluated using sparse vectors, and the performance comparison of algorithms on images is not studied. In short, the proposed method of comparison simplifies the interpretation of performance measures of the sparse recovery algorithms; and incorporates the relative MSE, the probability of exact support recovery and use the (K/M) ratio and the (M/N) ratio indirectly to generate a numerical figure of merit. It is shown experimentally that the proposed method gives a quantifiable performance measure. This new performance metric is computed for 24 algorithms from 8 different categories, and is shown numerically that the recently published projected gradient-based algorithms perform better. Following the inspiration gained from the analysis of different classes of algorithms and the advantages of thresholding and projected gradient methods, a new generalised method of sparse recovery algorithm design is proposed in the next chapter.

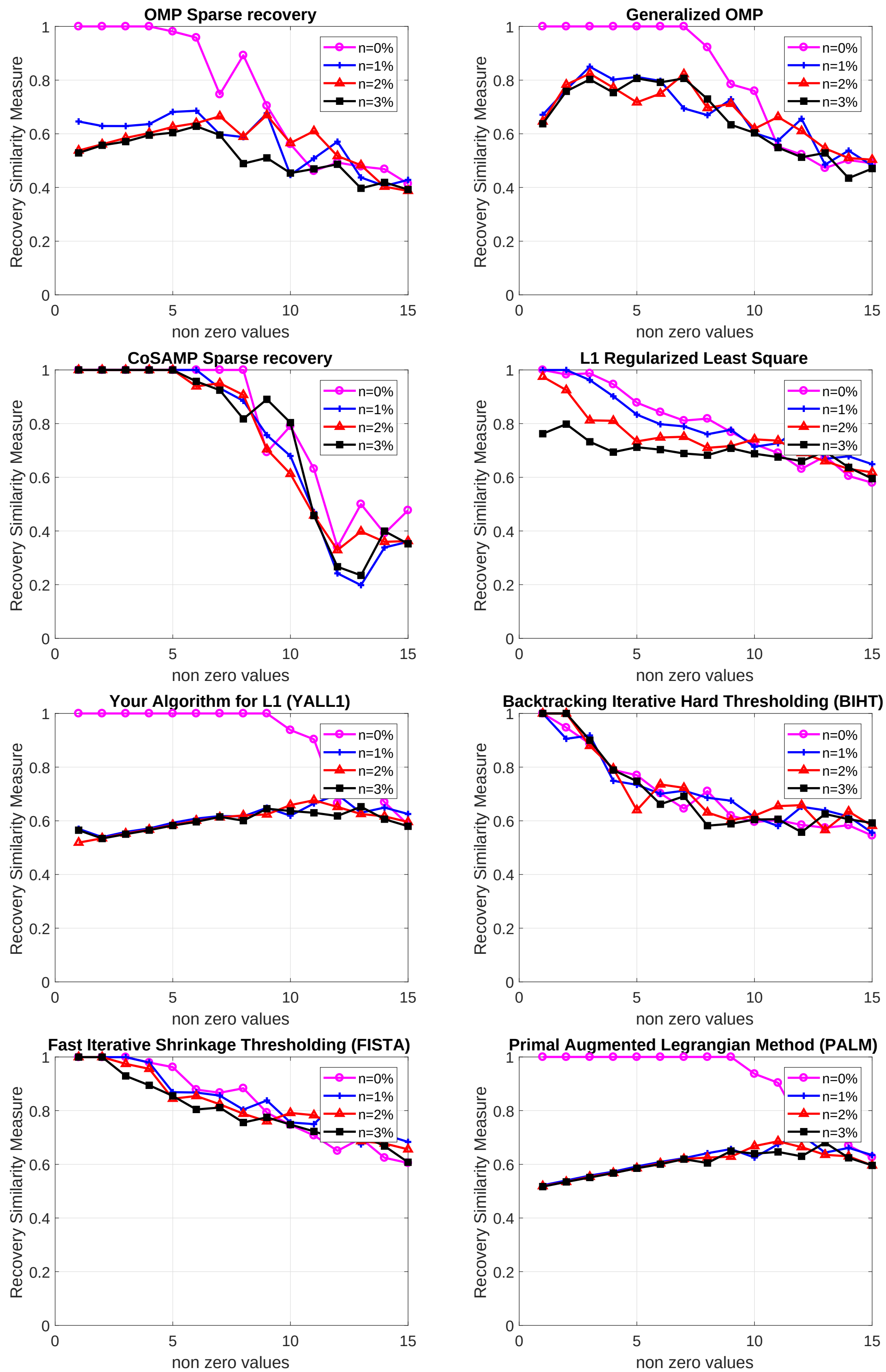


Figure 2-7: The variation of similarity measure $Sm(\hat{\mathbf{x}}, \mathbf{x})$ vs sparsity K of greedy ℓ_1 and thresholding methods.

Note: The similarity measure $Sm(\hat{\mathbf{x}}, \mathbf{x})$ of signals recovered using OMP, GOMP, CoSAMP, L1LS and YALL (greedy ℓ_1), BIHT, FISTA (threshold) and PALM.

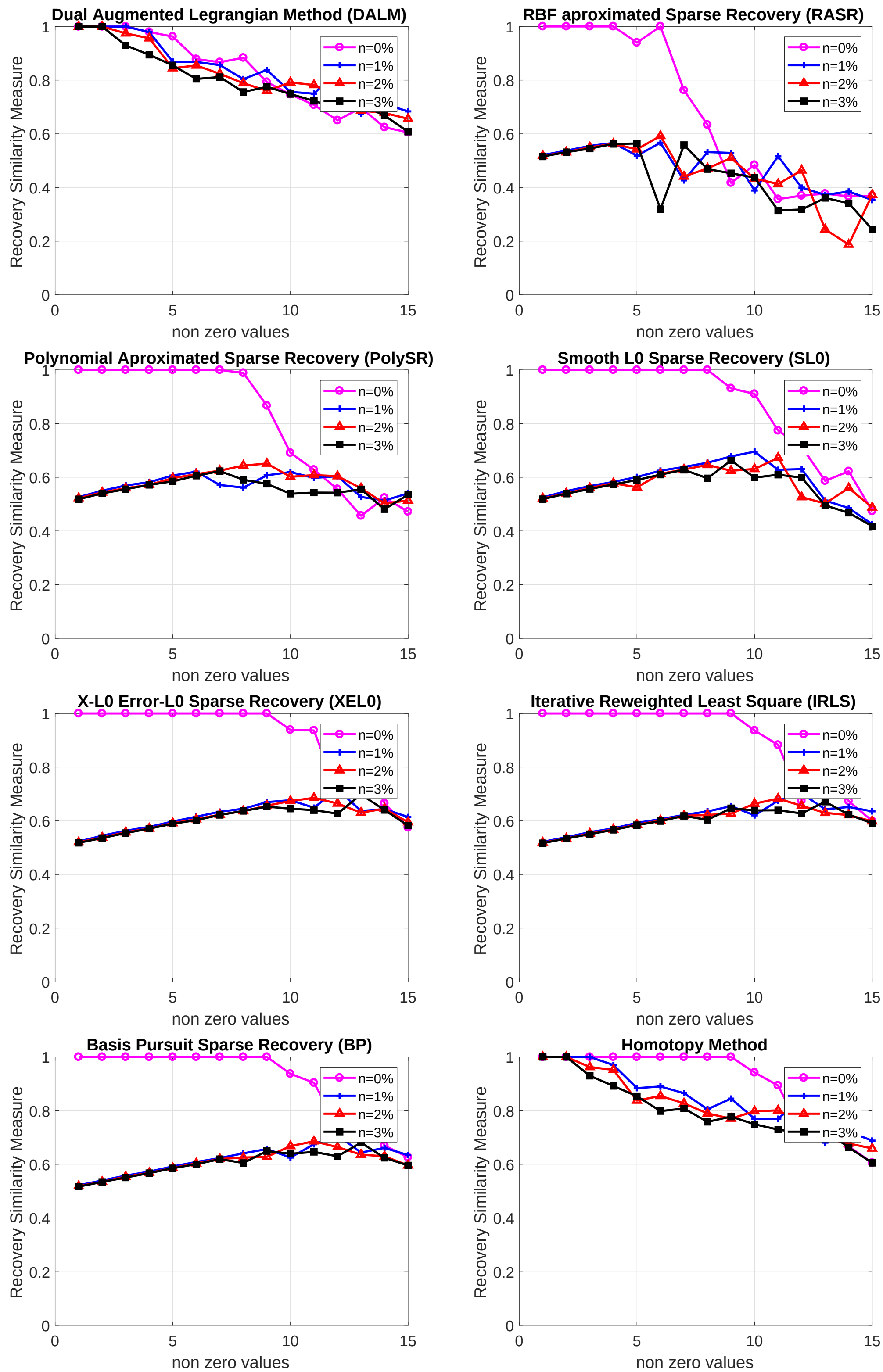


Figure 2-8: The variation of similarity measure $Sm(\hat{\mathbf{x}}, \mathbf{x})$ vs sparsity K of Lagrangian, ℓ_0, ℓ_p and ℓ_1 methods.

Note: The similarity measure $Sm(\hat{\mathbf{x}}, \mathbf{x})$ of signals recovered using DALM, RASR, PolySR (Lagrangian), SL0, XEL0 (ℓ_0), IRLS (ℓ_p), BP and Homotopy (ℓ_1).

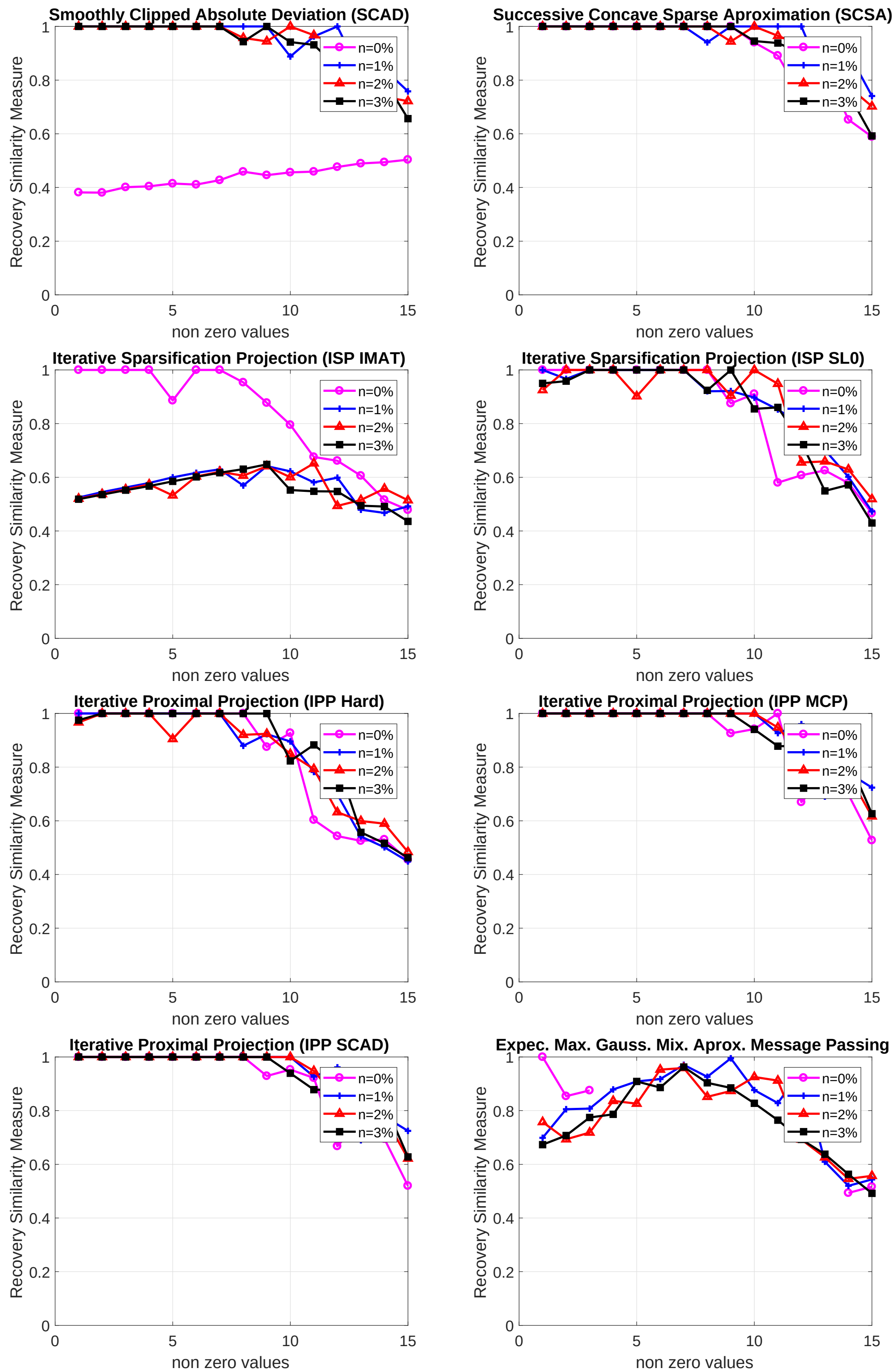


Figure 2-9: The similarity measure $Sm(\hat{\mathbf{x}}, \mathbf{x})$ vs sparsity K of the reconstructed signals recovered using the projected gradient methods and Bayesian method.

Note: The $Sm(\hat{\mathbf{x}}, \mathbf{x})$ *measure is close to* 1 *for perfect recovery with negligible reconstruction error.*

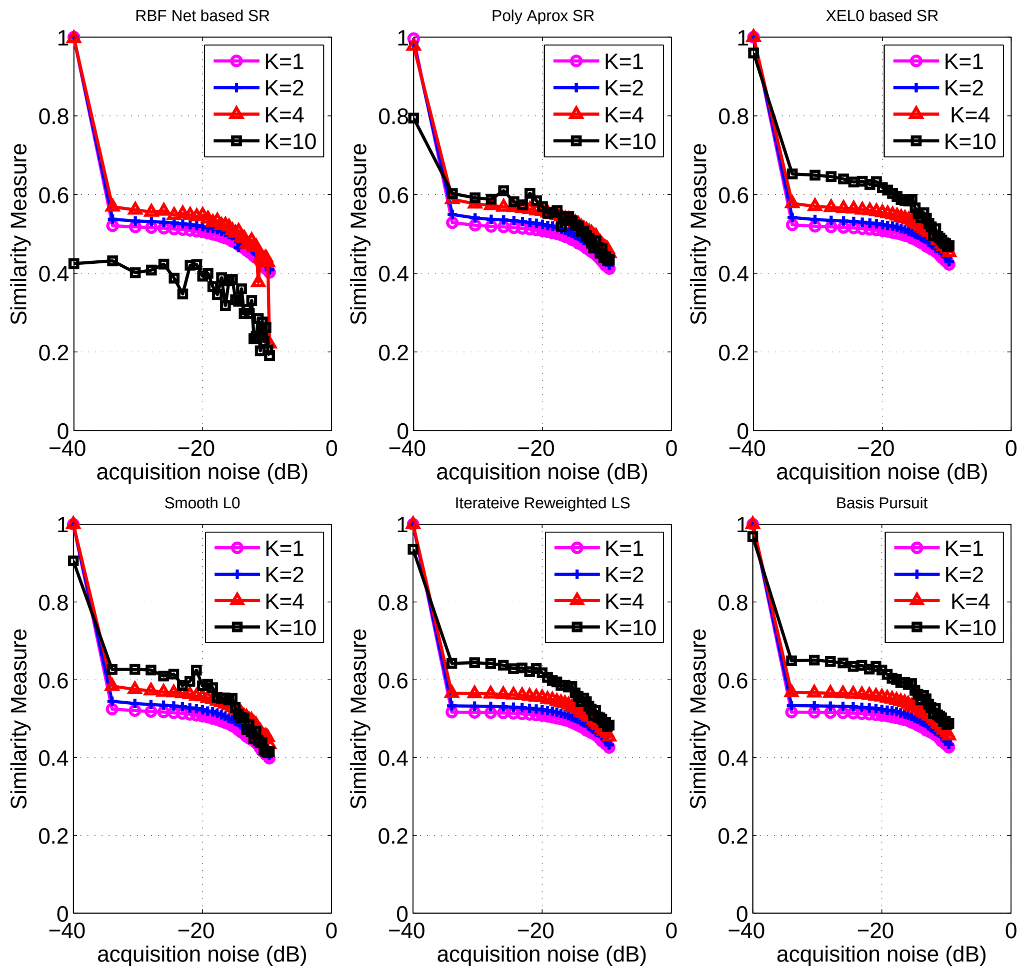


Figure 2-10: The variation of similarity measure $Sm(\hat{\mathbf{x}}, \mathbf{x})$ vs noise perturbation for-Lagrangian, ℓ_0, ℓ_p and ℓ_1 methods.

Note: The similarity measure $Sm(\hat{\mathbf{x}}, \mathbf{x})$ vs acquisition noise of the reconstructed signals recovered using classical Lagrangian method RBF Network based sparse recovery and Polynomial Approximation based sparse recovery; the ℓ_0 method SL0 and X-EL0, the ℓ_p method IRLS and the ℓ_1 method basis pursuit. The sparsity simulated are $K = 1, 2, 4, 10$. The measurement is perturbed with Gaussian noise of strength $-40dB$ to $-10dB$. The signal recovery similarity measure of the ℓ_0, ℓ_p and ℓ_1 based methods shows similar characteristic curve in presence of noise. The graph is shown here for illustration. The ξ-metric is computed from $Sm(\hat{\mathbf{x}}, \mathbf{x})$ vs K graph.

Table 2.1: **Performance comparison of sparse signal reconstruction algorithms in terms of conventional measures MSE, $relativeMSE$, Support Error and Execution time.**

Method	MSE	Relative MSE	Supp Error	Time (ms)
OMP	1.41×10^{-3}	6.64×10^{-31}	0.03	0.0065
GOMP	2.57×10^{-32}	1.20×10^{-31}	0.0	0.0007
CoSAMP	$\mathbf{1.90}\times10^{-32}$	$\mathbf{9.38}\times10^{-32}$	0.0	0.0110
L1 LS	1.03×10^{-4}	4.95×10^{-4}	0.24	0.0323
YALL	9.03×10^{-8}	4.07×10^{-7}	0.0	0.0070
BIHT	5.24×10^{-2}	2.51×10^{-1}	0.26	0.0426
FISTA	3.83×10^{-4}	1.84×10^{-3}	0.07	0.0221
PALM	5.02×10^{-11}	2.38×10^{-10}	0.0	0.5053
DALM	3.83×10^{-4}	1.84×10^{-3}	0.07	0.0508
RASR	8.17×10^{-3}	3.13×10^{-2}	0.09	0.0508
PolySR	1.13×10^{-7}	5.27×10^{-7}	0.0	0.0464
SL0	5.44×10^{-17}	2.61×10^{-16}	0.0	0.0082
XEL0	1.03×10^{-9}	4.90×10^{-9}	0.0	0.0054
IRLS	2.84×10^{-30}	1.36×10^{-29}	0.0	0.0556
BP	1.44×10^{-30}	6.41×10^{-30}	0.0	0.0246
Homtop	$\mathbf{7.20}\times10^{-33}$	$\mathbf{4.17}\times10^{-32}$	0.0	0.0039
EGAmp	1.77×10^{-7}	1.47×10^{-5}	0.04	0.053
SCAD	7.97×10^{-2}	3.73×10^{-1}	0.89	0.0063
SCSA	7.00×10^{-13}	3.76×10^{-12}	0.0	0.0102
ISP Imat	2.99×10^{-2}	1.54×10^{-1}	0.09	0.0015
ISP SL0	2.88×10^{-11}	1.45×10^{-10}	0.0	0.0235
IPP hrd	2.87×10^{-10}	1.37×10^{-9}	0.0	0.0230
IPP mcp	3.09×10^{-10}	1.48×10^{-9}	0.0	0.0262
IPP scad	2.94×10^{-10}	1.41×10^{-9}	0.0	0.0297

Data type used for simulation : sparse spike signal.

Table 2.2: **Performance comparison of sparse signal reconstruction algorithms in terms of conventional measures** MSE, $relativeMSE$, **Support Error and Execution time.**

Method	MSE	Relative MSE	Supp Error	Time (ms)
OMP	8.15×10^{-33}	1.07×10^{-30}	0.0	0.002
GOMP	$\mathbf{1.92}\times10^{-33}$	$\mathbf{1.33}\times10^{-31}$	0.0	0.000
CoSAMP	1.01×10^{-3}	8.34×10^{-2}	0.06	0.011
L1 LS	8.67×10^{-5}	2.63×10^{-2}	0.10	0.026
YALL	1.34×10^{-9}	1.36×10^{-7}	0.0	0.006
BIHT	4.90×10^{-3}	2.64×10^{-1}	0.22	0.042
FISTA	3.38×10^{-4}	1.04×10^{-1}	0.05	0.021
PALM	1.64×10^{-11}	1.45×10^{-9}	0.0	0.484
DALM	3.38×10^{-4}	1.04×10^{-1}	0.05	0.048
RASR	2.72×10^{-16}	6.16×10^{-15}	0.0	0.045
PoySR	1.42×10^{-9}	3.22×10^{-8}	0.0	0.042
SL0	5.54×10^{-17}	2.41×10^{-14}	0.0	0.007
XEL0	9.54×10^{-10}	4.18×10^{-7}	0.0	0.005
IRLS	9.74×10^{-10}	8.05×10^{-8}	0.0	0.059
BP	1.85×10^{-31}	9.18×10^{-30}	0.0	0.024
Homtop	$\mathbf{5.28}\times10^{-34}$	$\mathbf{2.93}\times10^{-32}$	0.0	0.002
EGAmp	1.77×10^{-7}	1.47×10^{-5}	0.04	0.053
SCAD	4.50×10^{-3}	3.14×10^{-1}	0.89	0.006
SCSA	1.12×10^{-14}	1.01×10^{-12}	0.0	0.011
ISP Imat	$\mathbf{3.61}\times10^{-34}$	$\mathbf{3.27}\times10^{-32}$	0.0	0.001
ISP SL0	4.18×10^{-11}	1.76×10^{-8}	0.0	0.024
IPP hrd	2.99×10^{-10}	1.28×10^{-7}	0.0	0.019
IPP mcp	2.96×10^{-10}	9.73×10^{-8}	0.0	0.024
IPP scad	2.95×10^{-10}	9.86×10^{-8}	0.0	0.025

Data type used for simulation : sparse pulse signal.

Table 2.3: **Comparison of sparse recovery algorithms in terms of ξ-metric computed for different similarity index** $Sm(.)$

		case-1			case-2		
	Noise	nil	nil	nil	0.01	0.01	0.01
	$Sm(.)$	1.0	≥0.9	≥0.8	1.0	≥0.9	≥0.8
genre	method	$\xi\ \vert_{sm=1}$	$\xi\ \vert_{sm\geq 0.9}$	$\xi\ \vert_{sm\geq 0.8}$	$\xi\ \vert_{sm=1}$	$\xi\ \vert_{sm\geq 0.9}$	$\xi\ \vert_{sm\geq 0.8}$
Greedy	OMP	0.2833	0.3167	0.3167	0.2000	0.2000	0.2000
ℓ_1	GOMP	0.3333	0.3500	0.3500	0.2000	0.2000	0.2000
	CoSAMP	0.3500	0.3500	0.3500	0.3167	0.3333	0.3500
	L1-LS	0.2333	0.2833	0.3500	0.2500	0.2667	0.3000
	YALL1	0.3667	0.3833	0.4000	0.2000	0.2000	0.2000
Thresh.	FISTA	0.2667	0.3000	0.3500	0.2667	0.2833	0.3333
	BIHT	0.2333	0.2500	0.2667	0.2333	0.2333	0.2667
Lagran.	PALM	0.3667	0.3833	0.4000	0.2000	0.2000	0.2000
	DALM	0.2667	0.3000	0.3500	0.2667	0.2833	0.3333
	RASR	0.2833	0.3167	0.3167	0.2000	0.2000	0.2000
	PolySR	0.3333	0.3500	0.3667	0.2000	0.2000	0.2000
ℓ_0	SL0	0.3500	0.3833	0.3833	0.2000	0.2000	0.2000
	XEL0	0.3667	0.4000	0.4000	0.2000	0.2000	0.2000
ℓ_p	IRLS	0.3667	0.3833	0.4000	0.2000	0.2000	0.2000
ℓ_1	BP	0.3667	0.3833	0.4000	0.2000	0.2000	0.2000
	Homtop	0.3667	0.3833	0.4000	0.2667	0.2833	0.3333
Bayes	EGAmp	0.2333	0.2333	0.4333	0.2000	0.2000	0.2000
Projected	SCAD	0.2000	0.2000	0.2000	0.3667	0.3667	0.4500
Grad.	SCSA	**0.3667**	**0.3833**	**0.4000**	**0.3333**	**0.4167**	**0.4167**
	ISP Imat	0.2833	0.2833	0.3667	0.2000	0.2000	0.2000
	ISP SL0	0.3500	0.3500	0.3833	0.2333	0.3667	0.4000
	IPP hrd	0.3500	0.3500	0.3833	0.3333	0.3333	0.3833
	IPP mcp	**0.3500**	**0.4000**	**0.4000**	**0.3833**	**0.4167**	**0.4167**
	IPP scad	**0.3500**	**0.4000**	**0.4000**	**0.3833**	**0.4167**	**0.4167**

Data type used for simulation : sparse spike signal. Case 1: noise free measurement. Case 2: 1.0% noise added to the measurement.
Higher value of ξ indicates closeness of reconstructed signal to the original signal. The highlighted values indicate the best values obtained for both noisy and noise free case considered. Sparse signal reconstruction using iterative proximal projection based algorithms give highest closeness to original signal.

Table 2.4: **Comparison of sparse recovery algorithms in terms of ξ-metric computed for different similarity index $Sm(.)$**

		case-3			case-4		
	Noise	nil	nil	nil	0.01	0.01	0.01
	$Sm(.)$	1.0	≥0.9	≥0.8	1.0	≥0.9	≥0.8
genre	method	$\xi\ \|_{sm=1}$	$\xi\ \|_{sm\geq 0.9}$	$\xi\ \|_{sm\geq 0.8}$	$\xi\ \|_{sm=1}$	$\xi\ \|_{sm\geq 0.9}$	$\xi\ \|_{sm\geq 0.8}$
Greedy	OMP	0.2667	0.3167	0.3667	0.2333	0.2500	0.3167
ℓ_1	GOMP	0.3667	0.3667	0.3667	0.2000	0.2000	0.3500
	CoSAMP	0.3000	0.3000	0.3500	0.2833	0.2833	0.3500
	L1-LS	0.2000	0.2667	0.3000	0.2000	0.3000	0.3167
	YALL1	0.4000	0.4167	0.4167	0.2000	0.2000	0.2000
Thresh.	BIHT	0.2500	0.2667	0.2667	0.2333	0.2333	0.2500
	FISTA	0.2000	0.2000	0.3000	0.2000	0.2000	0.3000
Lagran.	PALM	0.4167	0.4167	0.4167	0.2000	0.2000	0.2000
	DALM	0.2000	0.2000	0.3000	0.2000	0.2000	0.3000
	RASR	0.3167	0.3167	0.3167	0.2000	0.2000	0.2000
	PolySR	0.3500	0.3500	0.3833	0.2000	0.2000	0.2000
ℓ_0	SL0	0.3667	0.3667	0.4000	0.2000	0.2000	0.2000
	XEL0	0.4167	0.4167	0.4167	0.2000	0.2000	0.2000
ℓ_p	IRLS	0.3833	0.4167	0.4167	0.2000	0.2000	0.2000
ℓ_1	BP	0.4167	0.4167	0.4167	0.2000	0.2000	0.2000
	Homotp	0.2667	0.2667	0.3667	0.2500	0.3000	0.3167
Bayes	EGAmp	0.2333	0.4000	0.4167	0.2000	0.2000	0.2000
Projected	SCAD	0.2000	0.2000	0.2000	0.3167	0.3167	0.4000
Grad.	SCSA	**0.4167**	**0.4167**	**0.4167**	**0.3667**	**0.3667**	**0.4000**
	ISP Imat	0.3667	0.3667	0.3667	0.2000	0.2000	0.2000
	ISP SL0	0.3667	0.3667	0.4000	0.2000	0.3333	0.3667
	IPP hrd	0.3667	0.3667	0.4000	0.3500	0.3500	0.3500
	IPP mcp	**0.4167**	**0.4167**	**0.4167**	**0.3667**	**0.3667**	**0.4000**
	IPP scad	**0.4167**	**0.4167**	**0.4167**	**0.3667**	**0.3667**	**0.4000**

Data type used for simulation: sparse pulse signal. Case 3: noise free measurement. Case 4: 1.0% noise added to the measurement.
Higher value of ξ indicates closeness of reconstructed signal to the original signal. The highlighted values indicate the best values obtained for both noisy and noise free case considered. Sparse signal reconstruction using iterative proximal projection based algorithms give highest closeness to original signal.

Chapter 3

Framework for Segmented Threshold Based Sparse Signal Recovery

Signal reconstruction from compressed sensed data need iterative methods since the sparse measurement matrix is analytically non invertible. The iterative thresholding and ℓ_0 function minimization are of special interest as these two operations provide sparse solution. However these methods need an inverse operation corresponding to the measurement matrix for estimating the reconstruction error. The pseudo-inverse of the measurement matrix is used in general for this purpose. A sparse signal recovery framework using an approximate inverse matrix $\mathbf{Q}$ and iterative segment thresholding of ℓ_0 and ℓ_1 norm with residue addition is presented in this chapter. Two recovery algorithms are developed using this framework. The ℓ_0 based method is later developed to a basis function dictionary based network for sparse signal recovery. The proposed framework enables the users experiment with different inverse matrix to achieve better sparse signal recovery efficiency and implement in the algorithm in computationally efficient way.

3.1 Introduction

Sparse signals with limited number of nonzero values can be measured with lesser number of samples compared to Nyquist rate using Compressed Sensed (CS) from a linear projection space [1]. Signal sampling in CS based sparse signal acquisition combines the acquisition and compression into a single process, there by able to reduce the sample count. Instead of sampling $\mathbf{x}$, the correlated and integrated signal $\mathbf{y} = \mathbf{A}\mathbf{x}$ is sampled, where $\mathbf{A}$ is the measurement matrix. This type of data acquisition is used in synthetic aperture radar, magnetic resonance imaging and computed tomography, where the high sampling requirement for fine resolution is not achievable due to physical constraints [2]. The compressed measurements need to be converted back to its original form. The realtime recovery of sparse signal is used in applications like channel estimation, where the average signal property is sufficient [40]. Different algorithms based on greedy matrix minimization [41], least squared error minimization, neural network and ℓ_0, ℓ_p and ℓ_1 function approximation are available for sparse signal recovery. The ideal ℓ_0 minimization problem is non polynomial time hard in terms of computation hence alternative methods are used. The initially developed greedy matrix minimization methods for ℓ_1 solution like Orthogonal matching pursuit (OMP) [17], Generalized orthogonal matching pursuit (GOMP) [18], Compressive sampling matching pursuit (CoSAMP) [19], and the later developments like L1 regularized least square (L1LS) and Your algorithm for L1 (YALL) [20] are successful in their demonstrated data set. The ℓ_1 function minimization based methods, basis pursuit [29] and homotopy [31] are widely used as the benchmark algorithms. These algorithms perform better on sparse data with high correlation. The basis pursuit denoising (BPDN)[42] relaxes the recovery condition with an acceptable error term ϵ. This recovery condition is modified in Dantzig selector [43] with ℓ_∞ norm. The least absolute shrinkage and selection operator (LASSO) algorithm [30] also solves the ℓ_1 approximation of ℓ_0 problem.

The classical Lagrangian constrained minimization based sparse recovery methods were used later such as Prime augmented Lagrangian multiplier (PALM) [23], Dual augmented Lagrangian multiplier (DALM) [24] and Radial basis function approximation sparse recovery algorithm (RASR) [25]. These classical methods performs well in defined noise levels. These algorithms are used here for the performance comparison.F Initially, the ℓ_0 based optimization was not attempted as the function is not differentiable at 0. Later, sparse recovery using function approximation of ℓ_0 norm was used in Smooth L0 (SL0) [26]. Iterative re-weighted least square (IRLS) [28] is a ℓ_p norm $p \in (0, 1)$ based method generally taken as the bench mark algorithm for many applications. ℓ_p function approximation based sparse signal recovery is a recent addition to this class [44]. A combination of ℓ_p and ℓ_1 norm minimization for sparse recovery is described in [45]. The methods based on the ℓ_1 minimization with thresholding like the Backtracking iterative hard threshold (BIHT) [21] and the Fast iterative shrinkage thresholding (FISTA) [22] induce highly sparse solutions, when the threshold is set to $\tau = 10^{-3}$. The proximal projection based methods are new addition to the sparse signal recovery. The variation of this method like Smoothly clipped absolute deviation (SCAD) [34], [35], Successive concave sparsity approximation [36] , Iterative sparsification projection with SL0 (ISP-SL0) [37], Iterative proximal projection with hard thresholding (IPP-hrd) [37], Iterative proximal projection with mcp (IPP-mcp) [37], and Iterative proximal projection with SCAD (IPP-scad) [37] give promising results. The Expectation maximization Gaussian mixture approximate message passing (EGAmp) [32] is taken as a candidate Bayesian method [33]. Comparative evaluation of various sparse recovery algorithms can be found in [46].

3.1.1 Constraints in Computing Platform and Algorithms

All the sparse recovery algorithms discussed above need heavy matrix computation and iterative function minimization. These are well established methods and work well in desktop computing platforms with MATLAB. But when it comes to implementation there are constrains on the computing platform. These algorithms are herculean task for platforms like IoT devices. However, the IoT based compressed sensing and sparse recovery are gaining momentum [47]. The evaluation AM3358 processor based IoT platform board for the implementation of networked data acquisition system is given in [48]. In scenarios where the measurements are sparse and distributed, like in ground potential rise measurements or electric field measurements, compressed sensing based signal acquisition can be used to reduce the bandwidth requirement for data collection [49]. The use of this approach in power quality analysis is described in [50] and agricultural environment monitoring is given in [51]. Two aspects need to be considered in such field applications. The systems may get damaged due to environmental condition hence needs to be replaceable with lesser cost. Second, the unit should have sufficient processing capability to handle the computational requirements. One way to meet both conditions is to use low cost computing platform and develop computationally efficient signal processing algorithm.

Considering the implementation constraints there is a need to design efficient sparse recovery algorithm. Here a discussion about spare signal recovery algorithms, their constraints and a general framework for designing efficient algorithms using sparsity promoting functions is presented. The rank deficient sparse measurement matrix is non invertible and hence other methods are used for the recovery of sparse data. One such method is the approximation of ℓ_0 norm function and its minimization. However an appropriate inverse matrix for the measurement matrix $\mathbf{A}$ is needed. In general sparse recovery algorithms use pseudo inverse $A^\dagger$ or A^T as the approximate inverse matrix. A framework for using any approximate inverse matrix $\mathbf{Q}$ in the sparse recovery algorithm is described here. The proposed framework is a combination and generalization of the approaches given in smooth ℓ_0 (SL0) [26], ℓ_0-zero attraction projection (ℓ_0-ZAP) [52], radial basis function cascade network for sparse signal recovery (RASR) [25] and the iterative proximal projection algorithm (IPP) [37]. The algorithm framework discussed here uses iterative method for the estimation of residue, segmented thresholding of the residue and the projection of residue to obtain optimal solution. Based on this framework two sparse reconstruction algorithms are proposed. First one is iterative segmented thresholding of residue for ℓ_1 minimization and the second is the iterative segmented thresholding of polynomial basis function approximation of ℓ_0 minimization for sparse recovery. A computing network architecture for implementation of the second algorithm namely the polynomial basis function dictionary based cascade network for sparse signal recovery is presented in section 4.3.3. The network is implemented with multiply and accumulate unit (MACC), ℓ_0-gradient approximation polynomial lookup table and segmented threshold function. Similar 3 layer implementation of the neural network using floor function, exponential and step function is described in [53]. Higher degree Lagrange polynomial based optimization of neural network is given in [54]. This chapter is arranged as follows: the methods for improving the performance of function minimization based sparse recovery algorithm are given in section 4.2. The section 4.3 presents the proposed sparse recovery frame work. The simulation and performance comparison with other seven classes of algorithms are given in section 4.4, followed by the chapter conclusion.

3.2 Optimization Based Sparse Signal Recovery

The unknown $\mathbf{x}$ given in the minimization problem (3.1) has two features; the support of $\mathbf{x}$ and the non zero values at the support locations, where Σ_K is the set of all K sparse vectors.

$$\mathbf{x}^* = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \,, \text{ subject to } \mathbf{y} - \mathbf{A}\mathbf{x} = 0 \tag{3.1}$$

$$\exists \mathbf{A} \ni \mathbf{y} = \mathbf{A}\mathbf{x}, \quad (1-\delta_K) \|\mathbf{x}\|_2^2 \leq \|\mathbf{A}\mathbf{x}\|_2^2 \leq (1+\delta_K) \|\mathbf{x}\|_2^2 \,, \quad \forall \mathbf{x} \in \Sigma_K \subset \mathbb{R}^N, \mathbf{A} \in \mathbb{R}^{M\times N}$$

Two alternatives for finding the solution to this problem are (i) greedy algorithms: where the support is determined first, then the non-zero values of $\mathbf{x}$ are determined by least square method, and (ii) relaxation methods: where, $\mathbf{x} \in \mathbb{R}^N$ is considered as a signal and the objective function $\|\mathbf{x}\|_0$ is approximated using a continuous differentiable function. The algorithms like SL0[26] and IPP[37] use ℓ_0 minimization with ℓ_2 of the error to optimize the solution. The ℓ_0 norm for error estimation is not explored in many sparse recovery algorithms. The ℓ_0 approximation of the signal $\mathbf{x}$ and the ℓ_0 approximation of the error $\mathbf{e}$ with segmented threshold is used here to develop a framework for sparse recovery algorithm. The points addressed in the development of this new framework are as follows.

- As the measurement matrix is not invertible, the best approximate-inverse of this matrix that can be used in the minimization algorithm needs to be found.
- The solution can be constrained using proximal projection.
- The thresholding induces sparse solution and hence an optimal thresholding method needs to be determined

Since the calculation of the inverse of the rank deficient measurement matrix is not feasible, a framework method for evaluating with various inverse matrices for the given problem is proposed. This is achieved by developing a computational expression using arbitrary approximate inverse matrix $\mathbf{Q}$. Two objective function minimization routes are used one through ℓ_1 minimization and second through ℓ_0 minimization. Any combinations of the sparsity inducing functions described in (3.2) can be used for sparse recovery, where $e(\mathbf{x}) = \mathbf{y} - \mathbf{A}\mathbf{x}$, $\mathbf{x}$ is the current solution, $\mathbf{Q}$ is the inverse matrix and λ is a regularization constant.

$$f(\mathbf{x}) = \begin{cases} case\ 1 : \ell_0(\mathbf{x}) - \lambda \|\mathbf{Q}e(\mathbf{x})\|_0 \\ case\ 2 : \ell_0(\mathbf{x}) - \lambda \|\mathbf{Q}e(\mathbf{x})\|_q^q \quad q \in [1, 2] \\ case\ 3 : \ell_p(\mathbf{x}) - \lambda \|\mathbf{Q}e(\mathbf{x})\|_q^q \quad q \in [1, 2], 0 < p < 1, \\ case\ 4 : l_1(\mathbf{x}) - \lambda \|\mathbf{Q}e(\mathbf{x})\|_2^2 \end{cases} \tag{3.2}$$

The first part of function induces sparse solution and second part constraining the error so that the solution has minimum support error with respect to the original signal. Ideally this method results in solution with exact support recovery. If error still persists, that will be in the magnitude part of the solution. In the first case the iterate and the reconstruction error are made sparse using ℓ_0 norm. In the later case, the iterate is made sparse using ℓ_0 norm and the error is minimized through p-norm, with $1 \leq p \leq 2$. The typical example of the case-3 algorithm is IRLS [28], where p-norm of signal and the mean squared error (MSE) of the recovered signal are minimized. Many of the algorithms in this class use only MSE of the error instead of any other norm. The framework for improving the sparse signal reconstruction is developed using the following concepts.

- Iterative residue estimation is used to compensate for the inaccuracy in the inverse transformation.
- Translating the estimated residue to the solution space.
- Use segmented threshold as proximity operator for inducing sparse solution.

In the following sections these concepts are described and used in the development of basis function dictionary based network for sparse signal recovery.

3.2.1 Improving Sparse Signal Recovery using Residue

The compressed measurement is not an 1-to-1 map. The sparse signal in $\mathbb{R}^N$ space is measured in $\mathbb{R}^M$ space using non invertible measurement matrix $\mathbf{A}$. The inverse mapping function $\mathbf{Q}$ will induce a finite error with respect to the original signal ($\|\mathbf{x} - \hat{\mathbf{x}}\| \neq 0$). The process involved in this operation using the measurement matrix $\mathbf{A} : \mathbb{R}^N \to \mathbb{R}^M$ is illustrated in Figure 3-1; where, $\mathbf{x}$ is transformed to measurement $\mathbf{y}$. The original signal is estimated from the measurement by defining an inverse function $\mathbf{Q} : \mathbb{R}^M \to \mathbb{R}^N$, which maps $\mathbf{y}$ back to $\hat{\mathbf{x}}$. As the measurement matrix $\mathbf{A}$ is not invertible in general and the selected inverse function $\mathbf{Q}$ may not give exact recovery. The difference $\delta\mathbf{x}(k)$ between the iterative estimate $\mathbf{x}(k)$ and the forward followed by the inverse operation of the estimate will be finite non zero value (3.3).

$$\delta\mathbf{x}(k) = \mathbf{x}(k) - \mathbf{QAx}(k) \tag{3.3}$$

where, k indicates the iteration. It is possible to estimate a residue $\mathbf{r}(k) \in \mathbb{R}^M$ in measurement space as given in (3.4), such that it compensates the error.

$$\delta\mathbf{x}(k) = \mathbf{Qr}(k) \tag{3.4}$$

This residue is taken as function (f_r) of the difference between the original measurement $\mathbf{y}$ and the measurement mapped from the current iterate $\mathbf{Ax}(k)$, the inverse function $\mathbf{Q}$ and the measurement matrix $\mathbf{A}$.

$$\mathbf{r}(k) = f_r(\mathbf{y} - \mathbf{y}(k), \mathbf{Q}, \mathbf{A}) \tag{3.5}$$

where, $\mathbf{y}(k) = \mathbf{Ax}(k)$. This residue value should reduce if the iteration converges to the solution. Using (3.3) and (3.4) the projected residue is expressed as (3.6) and a

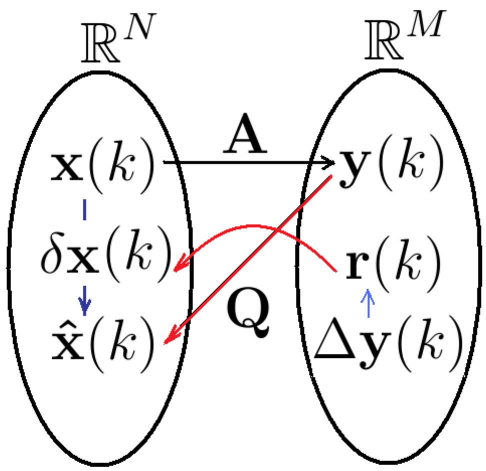


Figure 3-1: The mapping between sparse vector $\mathbf{x}$, measurement vector $\mathbf{y}$ and reconstructed vector $\hat{\mathbf{x}}$.

Note: Sparse measurement is not an 1-to-1 map. The sparse signal from $\mathbb{R}^N$ *is measured in* $\mathbb{R}^M$ *using a non invertible measurement mapping function* $\mathbf{A}$*. The best possible inverse mapping function* $\mathbf{Q}$ *results in some error with respect to the original signal* ($\|\mathbf{x} - \hat{\mathbf{x}}\| \neq 0$)*. There is a residue* $\mathbf{r}(k) \in \mathbb{R}^M$ *which when projected using* $\mathbf{Q}$ *compensates the error in the iterate* $\mathbf{x}(k)$*. The accuracy of reconstruction depends on how accurately this residue* $\mathbf{r}(k)$ *is estimated.*

small change in the residue is expressed as (3.7).

$$(\mathbf{I} - \mathbf{QA})\mathbf{x}(k) = \mathbf{Qr}(k) \tag{3.6}$$

$$(\mathbf{I} - \mathbf{QA})d\mathbf{x} = \mathbf{Q}d\mathbf{r} \tag{3.7}$$

Considering $d\mathbf{x} = \mathbf{x}(k) - \mathbf{x}(k-1)$, the finite change in residue required to eliminate the inverse operation error is written as (3.8).

$$d\mathbf{r} = (\mathbf{Q}^{\dagger} - \mathbf{A})d\mathbf{x} \tag{3.8}$$

where $\mathbf{Q}^{\dagger}$ is the pseudo inverse of $\mathbf{Q}$ and $\mathbf{x}(k-1)$ is the previous iterate. The residue update and its inverse operation in solution space is written as (3.9) and (3.10)

$$\mathbf{r}(k+1) = \mathbf{r}(k) + (\mathbf{Q}^{\dagger} - \mathbf{A})d\mathbf{x} \tag{3.9}$$

$$\mathbf{Qr}(k+1) = \mathbf{Qr}(k) + (\mathbf{I} - \mathbf{QA})d\mathbf{x} \tag{3.10}$$

Considering (3.2) as the Lagrangian formulation of the problem, the signal estimate is updated as (3.11), using the projected residue and the gradient of the Lagrangian function; where $\alpha < 1$ is a scale factor and $\nabla f(\mathbf{x}(k))$ the gradient.

$$\mathbf{x}(k+1) = \mathbf{x}(k) - \alpha\nabla f(\mathbf{x}(k)) + \alpha\mathbf{Qr}(k) + \hat{\sigma}d\mathbf{x} \tag{3.11}$$

$\hat{\sigma} = \alpha(\mathbf{I} - \mathbf{QA})$ is the scalar approximation of the weak diagonal $\mathbf{I} - \mathbf{QA}$ matrix with low condition number. The element vise magnitude representation of $\mathbf{QA}$ matrix is shown in Figure 3-2. Ideally, if the inverse operation is perfect the matrix $\mathbf{QA}$ will be a diagonal matrix. This scalar is used to change the search direction during optimization. For example if the Lagrangian is $L = \|\mathbf{x}\|_1 - \frac{\lambda}{2}\|\mathbf{Qe}\|_2^2$, the update can be written as (3.12).

$$\mathbf{x}(k+1) = \mathbf{x}(k) - \alpha(\hat{1} - 2\lambda(\mathbf{QA})^{\mathbf{T}}\mathbf{Qe}) + \alpha\mathbf{Qr}(k) + \hat{\sigma}d\mathbf{x} \tag{3.12}$$

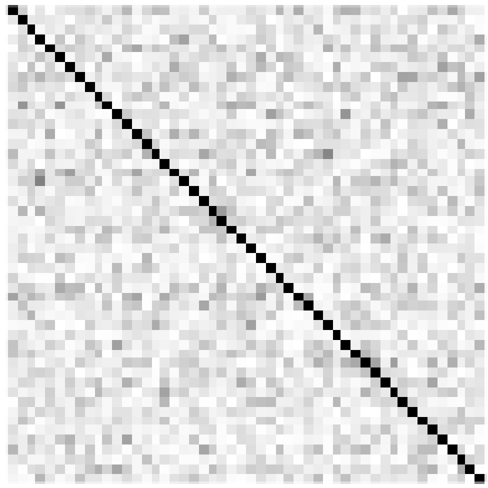


Figure 3-2: The gray scale representation of **QA** matrix.

Note: The dark spots show the non zero values. The matrix will be diagonal if the inverse transformation is perfect.

3.2.2 Improving Sparse Solution through Proximal Projection

It is found from the preliminary study that the iterative proximal projection based methods give better reconstruction of the sparse signals. The use of proximal projection method is explored in the framework for improving the sparse recovery performance. The proximal function minimization is written in general as (3.13)

$$\mathbf{x}^* = arg \min_{\mathbf{x}\in\mathbb{R}^N} f(\mathbf{x}) \quad s.t. \quad \text{Proximal to } g(\mathbf{x}) \tag{3.13}$$

In the present scenario $f(\mathbf{x})$ is sparsity inducing function and $g(\mathbf{x})$ is a continuous differentiable function. The functions which induce sparse solution have two parts; (1) a p-norm computation $l_p(\mathbf{x})$ and (2) error minimization, generalized as (3.14).

$$f(\mathbf{x}) = l_p(\mathbf{x}) + h(|\mathbf{y} - \mathbf{A}\mathbf{x}|) \quad 0 \leq p \leq 1 \tag{3.14}$$

where $h(|\mathbf{y} - \mathbf{Ax}|)$ is a function for error minimization, is taken as scaled error $\alpha\|\mathbf{y} - \mathbf{Ax}\|_2^2$ or as projected value of error $\|\mathbf{Q}(\mathbf{y} - \mathbf{Ax})\|_p$. Given a current solution point, $\mathbf{x}(k+1)$, the proximal point estimated using the function $g(\mathbf{x})$ is defined as (3.15)

$$\text{Proximal to } g : g_\mu(\mathbf{x}(k+1)) = arg \min_{\mathbf{x}\in\mathcal{D}_g} \frac{1}{2}\|\mathbf{x} - \mathbf{x}(k+1)\|_2^2 + \mu . g(\mathbf{x}(k+1)) \quad (3.15)$$

where $\mathcal{D}_g$ denotes the domain of the function $g(\mathbf{x})$, μ is a scalar and the factor $\|\mathbf{x} - \mathbf{x}(k+1)\|_2^2$ constraints the deviation from the current solution. The proximal solution is written as

$$\hat{\mathbf{x}} = g_\mu(\mathbf{x}(k+1)) \quad (3.16)$$

3.2.3 Segmented Thresholding as Proximity Operator

The computational precision introduce limit cycle oscillations in solution making it hard to reach ideal zero, when the iteration approaches the minimum. To improve the sparseness of the solution a thresholding function is used as proximal operator. The user configurable thresholding function $g(\mathbf{x}, \mu\ \hat{a})$ is defined using linear segments as given in (3.17) $\forall x \subset \mathbf{x}$.

$$g(\mathbf{x}, \mu, \hat{a}) = \begin{cases} 0 & 0 \leq |x| \leq a_1\mu \\ \text{sign}(x)(\frac{a_2-1}{a_2-a_1}|x| - a_1\mu) & a_1\mu < |x| \leq a_2\mu \\ \text{sign}(x)\max(|x| - \mu, 0) & a_2\mu < |x| \leq a_3\mu \\ \frac{(a-1)x - \text{sign}(x)a\mu}{a-2} & a_3\mu < |x| \leq a\mu \\ x & |x| > a\mu \end{cases} \quad (3.17)$$

where $\text{sign}(x)$ gives the sign of the variable. The linear segments slope and range are controlled using the parameters $\hat{a} = [a_1, a_2, a_3, a]$ $(a_1 < a_2 < a_3 \leq a)$ and μ a finite positive constant. The graph of this function is given in Figure 3-3. The non linear thresholding is effective for $|x| \leq a\mu$ and for values $|x| > a\mu$ the transfer function is linear. A similar approach of thresholding used in [37] where only two segments for

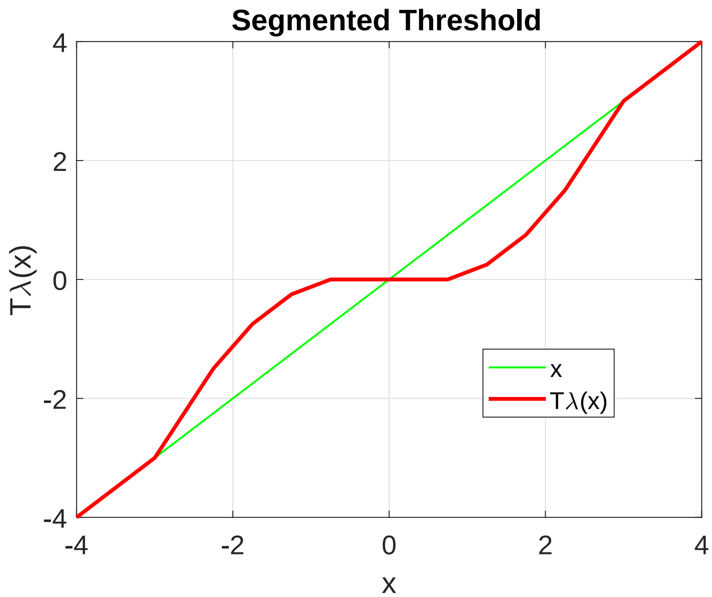


Figure 3-3: The segmented threshold function $T\lambda(x)$ used as the proximal projection operator.

Note: The segmented threshold function $T\lambda(x)$ is used as the proximal projection operator for the proposed segmented threshold algorithm. For small values the vector is soft thresholded and later a non linear transfer function is created using linear segments which reach unity gain for $|x| > a\mu$. The graph is shown for the case $a = 3\ \lambda = 1$. The parameter used is $\hat{a} = [0.75, 1.25, 2, 3]$. $T\lambda(x)$ represents $g(\mathbf{x}, \mu, \hat{a})$ described in the text.

soft thresholding is used. In this framework a configurable threshold function (3.17) is defined and used.

3.2.4 Error due to Segmented Threshold

The thresholding of the signal $\mathbf{x}$ into $\mathbf{x}_T$ (3.18) makes the signal sparser by decreasing the number of non zero coefficients.

$$\mathbf{x}_T = \{x_j : \quad \forall x_j \geq \theta : x_j = x_j \text{ else } x_j = 0\} \tag{3.18}$$

where θ is the threshold limit value. The thresholding introduces an error $e_T(\mathbf{x})$.

$$e_T(\mathbf{x})_p = \min_{\mathbf{x}_T} \|\mathbf{x} - \mathbf{x}_T\|_p \quad \text{s.t. } \|\mathbf{x}_T\|_0 \leq \|\mathbf{x}\|_0 \tag{3.19}$$

The upper limit of error is determined by arranging the elements of $\mathbf{x}$ and $\mathbf{x}_T$ in non increasing order as $\mathbf{x}^{'}$ and $\mathbf{x}_T^{'}$. When threshold is chosen to limit the maximum number of elements to K, the non zero elements beyond K in the rearranged $\mathbf{x}_T$ contribute to the error as given in (3.20).

$$e_T(\mathbf{x})_p^p = \sum_{k=1}^{N} |x_k - x_{Tk}|^p = \sum_{k=1}^{N} |x_k^{'} - x_{Tk}^{'}|^p = \sum_{k=K+1}^{N} |x_k^{'}|^p \tag{3.20}$$

Using quasi p-norm $\|\mathbf{x}\|_{p,\infty}^p = \max_k(k|x_k^{'}|^p), \ k = 1 \ldots N$, the error in the threshold approximation is written as (3.21).

$$e_T(\mathbf{x})_p^p \leq (\frac{1}{K^2} - \frac{1}{N^2})\|\mathbf{x}\|_{p,\infty}^p \tag{3.21}$$

Considering the upper limit of quasi-p norm $\|\mathbf{x}\|_{p,\infty}^p \leq \|\mathbf{x}\|_p^p$, the relative noise in solution due to the segmented thresholding is computed as (3.22). The thresholding error can be approximated as (3.23) for highly sparse signals.

$$\text{Threshold Noise} = 10\log\left((\frac{1}{K^2} - \frac{1}{N^2})\frac{Ka^2\mu^2}{\|\mathbf{x}\|_2^2}\right) \tag{3.22}$$

$$\text{Threshold Noise} \leq -20\log(\frac{\sqrt{K}\|\mathbf{x}\|_2}{\mu a}) \tag{3.23}$$

The error introduced depends on the sparsity K and thresholding limit μa alone for normalized vector of large length ($K \ll N$).

3.3 Framework for Sparse Signal Recovery using Residue Projection and Thresholding

Using the concepts described in the section earlier, a framework for designing algorithms for sparse signal recovery is presented here. The process starts with selection of an approximate inverse matrix $\mathbf{Q}$ corresponding to the measurement matrix $\mathbf{A}$. Then select a starting point for iteration by setting $\mathbf{x}(0)$ and the initial residue $\mathbf{r}(0)$. The incremental change in the residue is computed as $d\mathbf{r} = (\mathbf{Q}^{\dagger} - \mathbf{A})d\mathbf{x}$. The residue is projected to solution space and augmented with Lagrangian gradient minimization. In the later step the segmented thresholding is used as the proximal solution operator.

The schematic description of the minimization framework is given in Figure 3-4 and the algorithmic representation is given in Algorithm-6. Using this framework two sparse recovery algorithms are developed; namely iterative segmented threshold residue projection for ℓ_1 minimization and iterative segmented threshold residue projection for ℓ_0 minimization. The second algorithm is further modified as a basis function dictionary based network for sparse signal recovery. The performance enhancement with respect to the existing benchmark algorithm is evaluated in the experimental evaluation section.

Algorithm 6
Framework for Sparse Recovery through Residue Projection and Thresholding

Require: $\mathbf{y}, \mathbf{A}, \mathbf{Q}, p, q, \mu, \alpha, \hat{a}, g_\mu$
Task: obtain sparse $\mathbf{x}$
Initialization: Select inverse matrix $\mathbf{Q}$ for the given $\mathbf{A}$
Define Error $\mathbf{e}(\mathbf{x}) = (\mathbf{y} - \mathbf{A}\mathbf{x})$, $\hat{\sigma} = \alpha(\mathbf{I} - \mathbf{Q}\mathbf{A})$
Thresholding function $g_\mu(\mathbf{x}, \mu, \hat{a})$, $\mu_0 = \mu$
Define Objective $f(\mathbf{x}) = l_p(\mathbf{x}) - \lambda \|\mathbf{Q}\mathbf{e}(\mathbf{x})\|_q^q \quad 0 \le p \le 1, \quad 0 \le q < \infty$
compute : $\nabla f(\mathbf{x})$
while $\mu_k > \mu_{min}$ **do**
  $\hat{\mathbf{x}}(k) = \mathbf{x}(k) - \alpha \nabla f(\mathbf{x}(k)) + \alpha \mathbf{Q}\mathbf{r}(k) + \hat{\sigma} d\mathbf{x}$
  $\mathbf{x}(k+1) = g_\mu(\hat{\mathbf{x}}(k), \mu, \hat{a})$
  $\mathbf{x}(k) = \mathbf{x}(k+1)$
  reduce: α, μ
end while
Output: sparse signal $\mathbf{x}(k)$

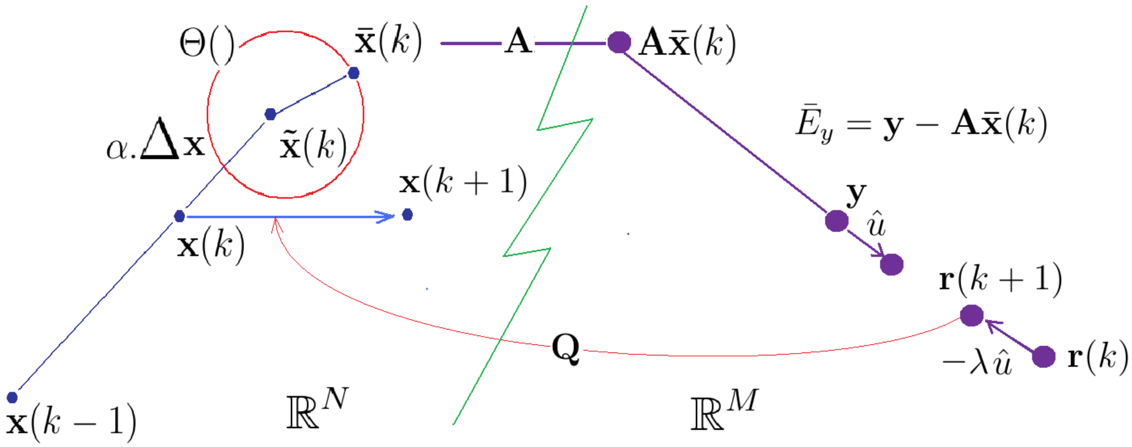


Figure 3-4: Schematic representation of the residue computation for minimizing the projection error.

Note: Residue computation for minimizing the error. From the previous 2 iterates, the current difference $\Delta\mathbf{x}$ *is determined. This value along with thresholding function* $\theta()$ *determines an intermediate value* $\bar{\mathbf{x}}(k)$*. The error in measurement* $\bar{E}_y$ *due to non optimal solution is determined. The updated residue* $\mathbf{r}(k+1)$ *is computed as a function of this error. The new residue is projected back using* $\mathbf{Q}$ *to update the iterate.*

3.3.1 Segmented Threshold Residue Projection for ℓ_1 Minimisation

Taking the sparsity inducing function described in case : 4 of (3.2) and using the residue mapping and the segmented thresholding a new algorithm is proposed based on ℓ_1 minimization. The function used for the Lagrangian based minimization is defined as (3.24)

$$f(\mathbf{x}) = \varepsilon\|\mathbf{x}\|_1 + \frac{\lambda}{2}\|\mathbf{Q}(\mathbf{y} - \mathbf{A}\mathbf{x})\|_2^2 \tag{3.24}$$

where ε and λ are regularization constants and $\mathbf{Q}$ is the inverse transformation matrix. The segmented thresholding function $g(\mathbf{x}, \mu, \hat{a})$ (3.17) is used as proximal operator. The i-th element of the Lagrangian function gradient, in the k-th iteration is given in (3.25) where $\mathbf{e}(k) = \mathbf{y} - \mathbf{A}\mathbf{x}(k)$.

$$\nabla f(\mathbf{x}(k)_i) = \varepsilon - \lambda[(\mathbf{Q}\mathbf{A})^{\mathbf{T}}\mathbf{Q}\mathbf{e}(k)]_i, \quad i = 1 \dots N \tag{3.25}$$

Using the Lagrangian minimization, the residue mapping and the segmented thresholding described in the framework, the iterate solution update is written as (3.26) and (3.27).

$$\hat{\mathbf{x}}(k) = \mathbf{x}(k) + \alpha_k \left(\lambda(\mathbf{QA})^{\mathbf{T}}\mathbf{Q}\mathbf{e}(k) - \varepsilon\hat{1}\right) + \alpha_k\mathbf{Q}\mathbf{r}(k) + \hat{\sigma}d\mathbf{x} \tag{3.26}$$

$$\mathbf{x}(k+1) = g(\mathbf{x}, \mu, \hat{a}) \tag{3.27}$$

where $d\mathbf{x} = \mathbf{x}(k) - \mathbf{x}(k-1)$. The scale factor α_k is decreased in every iteration and $\lambda = 1$ is set in the proceeding discussion. The residue is updated using new error $\mathbf{e}(k+1)$ and the finite change in residue estimated from (3.8).

$$\mathbf{r}(k+1) = \mathbf{r}(k) + \alpha_k\mathbf{e}(k+1) - \varepsilon\alpha_k d\mathbf{r} \tag{3.28}$$

where $\varepsilon < 1$ is a finite scale factor and $\mathbf{e}(k+1) = \mathbf{y} - \mathbf{A}\mathbf{x}(k+1)$. After the iteration the scale factor α_k and the threshold determination factor μ_k are updated as (3.29), where $\delta < 1$ is the value reduction factor.

$$\alpha_{k+1} = \delta\alpha_k, \quad \mu_{k+1} = \delta\mu_k \tag{3.29}$$

The algorithm is identified here as iterative segmented threshold residue projection (ISTRP). The computations steps in the proposed method is given in Algorithm-7.

Parameter Initialization for ISTRP: The initial value for the iteration is taken as $\mathbf{x}(0) = \mathbf{A}^{\dagger}\mathbf{y}$ and initial residue as $\mathbf{r}(0) = \varepsilon(\mathbf{y} - \mathbf{A}\mathbf{x}(0))$. The inverse of the measurement matrix $\mathbf{Q}$ is arbitrarily taken as (3.30) for the evaluation of this algorithm (3.30).

$$\mathbf{Q} = \left(\mathbf{I} - \frac{1}{2}\left\langle \frac{\mathbf{A}^{\mathbf{T}}\mathbf{A}}{diag(\mathbf{A}^{\mathbf{T}}\mathbf{A})}\right\rangle\right)^{-1}\mathbf{A}^{\dagger} \tag{3.30}$$

where the operator $\langle\rangle$ is defined to performs the column vise division with the corresponding element of denominator vector. The $diag(\mathbf{A}^{\mathbf{T}}\mathbf{A})$ represents the diagonal vector of $\mathbf{A}^{\mathbf{T}}\mathbf{A}$. The regularization constants are set as (3.31) and the segmented

Algorithm 7

Iterative segmented threshold residue projection (ISTRP)

Require: $\mathbf{y}, \mathbf{A}, \mathbf{Q}, \mu_{min}, \delta_T, \alpha, \lambda, \hat{a}$

Task: $\min \varepsilon \left\|\mathbf{x}\right\|_1 + \frac{1}{2}\mathbf{Q}\|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2$ s.t. Proximal to $\mathbf{g}(\mathbf{x}, \mu_k, \hat{a})$

Initialization: $\mathbf{P} = (\mathbf{QA})^{\mathbf{T}}\mathbf{Q}, \quad \mathbf{J} = \mathbf{Q}^{\dagger} - \mathbf{A}, \quad \hat{\sigma} = \mathbf{I} - \mathbf{QA}$

$\mathbf{x}(0) = \mathbf{x}(1) = \mathbf{A}^{\dagger}\mathbf{y}, \quad \mathbf{r}(1) = \varepsilon(\mathbf{y} - \mathbf{A}\mathbf{x}(\mathbf{0})), \quad \mu_1 = 3\max(\mathbf{x}(1))$

while $(\mu_k > \mu_{min})$ **do**

    while 1...N **do**

        $d\mathbf{x} = \mathbf{x}(k) - \mathbf{x}(k-1)$

        $\tilde{\mathbf{x}}(k) = \mathbf{x}(k) + \hat{\sigma} d\mathbf{x}$

        $\bar{\mathbf{x}}(k) = \mathbf{g}(\tilde{\mathbf{x}}(k), \mu_k, \hat{a})$

        $\mathbf{e}(k) = \mathbf{y} - \mathbf{A}\bar{\mathbf{x}}(k),$

        while $\|\mathbf{e}(k)\|_2 > \varepsilon$ **do**

            $\mathbf{x}(k+1) = \bar{\mathbf{x}}(k) + \mathbf{Q}\mathbf{r}(k) + \alpha\lambda\mathbf{P}\mathbf{e}(k) - \alpha\varepsilon\hat{1}$

            $\mathbf{e}(k+1) = \mathbf{y} - \mathbf{A}\mathbf{x}(k+1)$

            residue:

            $d\mathbf{x} = \mathbf{x}(k+1) - \mathbf{x}(k)$

            $d\mathbf{r} = \mathbf{J}d\mathbf{x}$

            update:

            $\mathbf{r}(k+1) = \mathbf{r}(k) + \alpha(\mathbf{e}(k+1) - \varepsilon d\mathbf{r})$

            $\mathbf{e}(k) = \mathbf{e}(k+1)$

            $\mathbf{r}(k) = \mathbf{r}(k+1)$

        end while

        $\mathbf{x}(k-1) = \mathbf{x}(k)$

        $\mathbf{x}(k) = \mathbf{x}(k+1)$

    end while

    $\mu_{k+1} = \mu_k\delta_T, \quad k = k+1$

end while

Output: $\mathbf{x}(k)$

threshold function is set using the parameters given in (3.32).

$$\lambda = 1, \quad \alpha = 0.2, \quad \varepsilon = 0.0001 \tag{3.31}$$

$$\hat{a} = [0.75, 1.24, 1.90, 2], \quad \mu_0 = 3 \max |\mathbf{x}(0)_i| \tag{3.32}$$

After every iteration the threshold level is reduced as (3.33) and the iteration is continued till the threshold reaches the minimum μ_{min} .

$$\delta = 0.9, \quad \mu_{min} = 1.0 \times 10^{-15} \tag{3.33}$$

3.3.2 Segmented Threshold Residue Projection for ℓ_0 Minimization

The first case of (3.2) is considered for ℓ_0 minimization and the Lagrangian function is defined as (3.34).

$$f(\mathbf{x}) = \|\mathbf{x}\|_0 + \lambda \|\mathbf{Qe}\|_0 \tag{3.34}$$

The ℓ_0 norm of the signal $\mathbf{x}$ and the ℓ_0 norm of the projected error is considered for minimization. The proximal solution is found using segmented thresholding function. The logic for using the ℓ_0 norm of the projected error ($\|\mathbf{Qe}\|_0$) is that: for the algorithm to converge, the error in reconstruction should be in the magnitude part of the sparse vector only. Since the ℓ_0 norm is not a differentiable function, the radial basis function is used to approximate the ℓ_0 function in [26], here the polynomial function is used to approximate the ℓ_0 norm as given in (3.35), where, q is an even and σ is a scalar value.

$$\|\mathbf{x}\|_0 := \lim_{\sigma \to 0} \sum_{i=1}^{N} 1 - \frac{1}{1 + \frac{x^q}{(\sigma/a)^q}}, \qquad \forall \mathbf{x} \in \mathbb{R}^N, \tag{3.35}$$

The Figure 3-5 show the function plots of this approximated ℓ_0 norm for various values of σ and q. In addition to approximating the ℓ_0 norm, this function (3.35) performs the thresholding also. The parameter σ determines the level of ℓ_0 approximation and the scalar value a determines the thresholding profile. The value σ_k is reduced in every iteration till σ_{min} is reached, to change the ℓ_0 approximation from coarse to fine solution. This function approaches ℓ_0 norm when $\sigma/a \to 0$. The thresholding changes from soft to had when q is increased. The optimization problem for ℓ_0 minimization is defined as (3.36), where $\Theta(\mathbf{x}, a, \mu_k)$ is the thresholding operator.

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ s.t. } \|\mathbf{Q}(\mathbf{y} - \mathbf{A}\mathbf{x})\|_0 \leq \varepsilon \quad \text{and proximal to } \Theta(\mathbf{x}, a, \mu_k) \tag{3.36}$$

The recovery error is computed as the projected value of the difference between measurement $\mathbf{y}$ and the measurement corresponding to the reconstructed signal; $\mathbf{e} = \mathbf{Q}(\mathbf{y} - \mathbf{A}\mathbf{x})$. Applying the polynomial approximation to the Lagrangian problem defined in (3.34) it is represented as (3.37).

$$L := N - \sum_{i=1}^{N} (\frac{1}{1 + (\frac{a x_i}{\sigma_k})^q}) - \lambda \left(N - \sum_{i=1}^{N} (\frac{1}{1 + (\frac{a e_i}{\sigma_k})^q}) \right) \tag{3.37}$$

The Lagrangian gradient is approximated as (3.38), by setting $\gamma_k = q(a/\sigma_k)^q$ and considering only the lower order terms of the polynomial derivative.

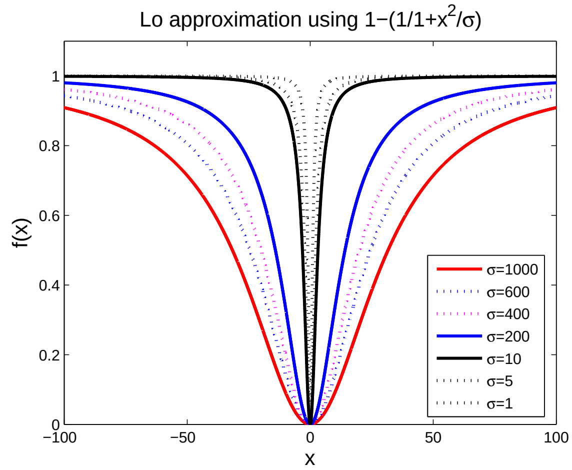
$$\frac{\partial L}{\partial x_i} \approx \frac{\gamma_k x_i^{q-1}}{1 + \frac{2}{q}\gamma_k x_i^q} + \lambda \mathbf{Q}\mathbf{A} \frac{\gamma_k e_i^{q-1}}{1 + \frac{2}{q}\gamma_k e_i^q} \tag{3.38}$$

where $x_i \subset \mathbf{x}$, $e_i \subset \mathbf{e}$ and $\frac{\partial e_i}{\partial x_i} = -(\mathbf{Q}\mathbf{A})_i$. The new iterate point $(\hat{\mathbf{x}}(k))$ is computed as (3.39), by adding projected error $\mathbf{e}(k)$ and the Lagrangian gradient to the $\mathbf{x}(k)$.

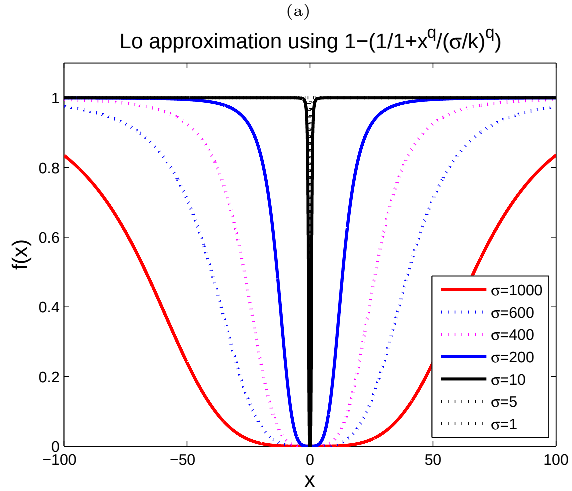
$$\hat{\mathbf{x}}(k) = \mathbf{x}(k) + \beta \mathbf{e}(k) - \alpha_k \nabla L(k) \tag{3.39}$$

where β is the scale factor for error correction. The gradient scale factor α_k is represented in terms of σ_k, a and q as (3.40), where α_0 is a positive constant and $\gamma_k = q(a/\sigma_k)^q$

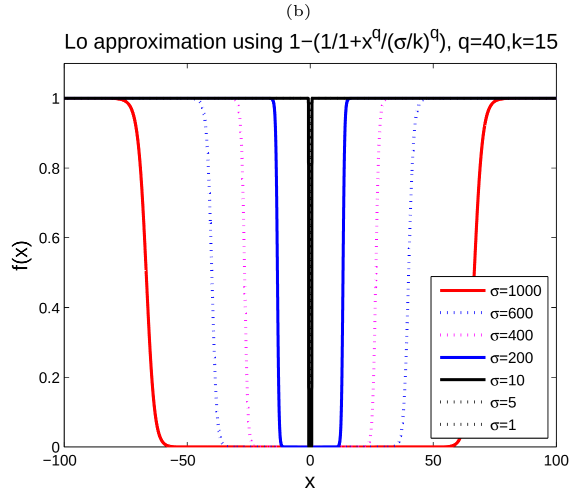
$$\alpha_k = \frac{q}{\gamma_k} \alpha_0 \tag{3.40}$$



(a)



(b)



(c)

Figure 3-5: The approximation of ℓ_0 norm function using polynomial $1 - 1/(1 + x^q/(\sigma/k)^q)$.

Note: The plot of ℓ_0 *norm approximation polynomial* $1-1/(1+x^q/(\sigma/k)^q)$ *for various values of* q *and* k*. When* $\sigma \to 0$ *all non zero values are given support 1. (a):* $q = 2, k = 1$*, (b):* $q = 4, k = 15$*, (c):* $q = 40, k = 15$

The iterative update equation (3.39) is modified as (3.41) after removing higher order terms from the gradient.

$$\hat{\mathbf{x}}_i(k) = \mathbf{x}_i(k) + \beta \mathbf{e}_i(k) - \alpha_k\left(\frac{\gamma_k x_i^{q-1}}{1 + \frac{2}{q}\gamma_k x_i^q} + \lambda \mathbf{Q}\mathbf{A}\frac{\gamma_k e_i^{q-1}}{1 + \frac{2}{q}\gamma_k e_i^q}\right) \tag{3.41}$$

The slope of the ℓ_0 approximation function $(1/1 + (\frac{ax_i}{\sigma_k})^q)$ near origin is increased in each iteration as (3.42) where $0 < \delta < 1$ determines the rate of convergence of σ_k.

$$\sigma_{k+1} = \sigma_k \delta \tag{3.42}$$

The initial value σ_0 is set in the range $(0, 3]$. The iterative update is simplified to (3.43), where $\Delta\mathbf{x}^k$ and $\Delta\mathbf{e}^k$ (3.44) are small changes in each iteration.

$$\hat{\mathbf{x}}(k) = \mathbf{x}(k) + \beta \mathbf{e}(k) - q\alpha_0(\Delta\mathbf{x}^k + \Delta\mathbf{e}^k) \tag{3.43}$$

$$\Delta\mathbf{x}_i^k \approx \frac{\mathbf{x}_i(k)^{q-1}}{1 + \frac{2}{q}\gamma_k \mathbf{x}_i(k)^q} \qquad \Delta\mathbf{e}_i^k \approx \lambda \mathbf{Q}\mathbf{A}\frac{\mathbf{e}_i(k)^{q-1}}{1 + \frac{2}{q}\gamma_k \mathbf{e}_i(k)^q} \tag{3.44}$$

The solution is made sparser (3.45) by applying the segmented thresholding.

$$\mathbf{x}(k+1) = g(\hat{\mathbf{x}}(k), \hat{a}, \mu_k) \tag{3.45}$$

Iteration Limit: The small amplitude values at the unsupported indexes appear in the reconstruction process when σ reaches lower limit and the incremental update is comparable to the noise figure x_n. The error residue and the gradient update become negligible at this final stage of solution convergence. Equating (3.43) to 0 gives the limiting condition (3.46) and the value of γ_k is obtained as (3.47)

$$|x_n| = q\alpha_0 \frac{x_n^{q-1}}{1 + \frac{2}{q}\gamma_k x_n^q} \tag{3.46}$$

$$\gamma_k = (q\alpha_0 x_n^{-2} - x_n^{-q})q/2 \tag{3.47}$$

since α_0 is positive, the following conditions (3.48), (3.49) and (3.50) are obtained as limiting condition for the iteration.

$$\alpha_0 > \frac{1}{q x_n^{q-2}} \tag{3.48}$$

$$\gamma_{max} > (c-1) x_n^{-q} q/2 \tag{3.49}$$

$$\sigma_{min} < a \left(\frac{2 q x_n^2}{\alpha_0 (1 - 1/c)} \right)^{1/q} \tag{3.50}$$

where $c > 1$ is a finite constant. The iterations (3.41) is stopped when σ_k reaches the minimum value σ_{min}. The ℓ_0 minimization for sparse signals based on this method is referred here as Segmented Threshold X-L0 E-L0 (STXEL0) algorithm. A special case of this algorithm using the inverse matrix $\mathbf{Q} = \mathbf{A}^{\dagger}$ and $q = 2$ is considered here for evaluation and the update equation is written as (3.51), where $i = 1 \dots N$.

$$\mathbf{x}(k+1) = g\left(\mathbf{x}(k) + \beta \mathbf{e}(k) - \alpha_0 \left[\frac{\mathbf{x}_i(k)}{1 + \frac{\mathbf{x}_i^2(k)}{\sigma_k}} + \lambda \frac{\mathbf{e}_i(k)}{1 + \frac{\mathbf{e}_i^2(k)}{\sigma_k}} \right], \hat{a}, \sigma_k \right) \tag{3.51}$$

where $g(.)$ is the thresholding function, $\hat{a}$ is the parameter for segmented threshold, $\sigma_{k+1} = \delta \sigma_k$ and $\delta < 1$. The values of σ_{min} is determined as (3.52).

$$\sigma_{min} = \max(\hat{a}) x_n / \sqrt{\alpha_0 (1 - 1/c)} \tag{3.52}$$

If the gradient of the polynomial is replaced with gradient of the Gaussian function $1 - exp(-x_n^2 / 2\sigma_{min}^2)$ for finite magnitude elements near zero, the limiting values of σ_{min} can be determined as (3.53).

$$\sigma_{min} = \frac{x_n}{\sqrt{2 \ln(\alpha_0)}} \tag{3.53}$$

Remark 1: A possible alternative condition for ℓ_0 minimization problem is given in (3.54), where $\mathbf{x}_{opt}$ is taken as an optimal solution since the prior information about the ideal solution $\mathbf{x}_{org}$ is not available.

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \; s.t. \; \|\mathbf{x}_{opt} - \hat{\mathbf{x}}\|_0 \leq \varepsilon \tag{3.54}$$

The Lagrangian condition gives same solution as above when $\hat{\mathbf{x}}$ is defined as $\mathbf{A}^{\dagger}\mathbf{A}\mathbf{x}$. Even when the problem is defined as $\min_{\mathbf{x}} \|\mathbf{x}\|_0 \, s.t. \, \|\mathbf{x} - \hat{\mathbf{x}}\|_0 \leq \varepsilon$ or with a different subjective condition $\min_{\mathbf{x}} \|\mathbf{x}\|_0 \, s.t. \, \left\|\mathbf{A}^{\dagger}(\mathbf{y} - \hat{\mathbf{y}})\right\|_0 \leq \varepsilon$, the solution do not change significantly and hence the problem defined in (3.36) is considered optimal. The implementation of segmented threshold ℓ_0 minimization method for sparse recovery is described in Algorithm-8.

Algorithm 8

Segmented Threshold X-L0 E-L0 Algorithm Description

Require: $\mathbf{y}, \mathbf{A}, \mathbf{Q}, \hat{a}, \alpha, \beta, \sigma_{min}, \mu_{min}, \delta_\alpha, \delta_T, \gamma$

Task: $\min \|\mathbf{x}\|_0$ s.t: $\|\mathbf{Q}(\mathbf{y} - \mathbf{A}\mathbf{x})\|_0 \leq \varepsilon$, proximal to $\mathbf{g}(\mathbf{x}, \mu_k, \hat{a})$

Initialization: $\mathbf{Q} = (\mathbf{I} + \gamma \mathbf{A}^{\mathbf{T}}\mathbf{A})^{-1}\mathbf{A}^{\dagger}, \quad \hat{a} = [0.75, 1.25, 1.99, 2]$

$\alpha = 0.7, \quad \sigma_{min} = 10^{-8}, \quad \delta_\sigma = 0.95, \quad \gamma = 0.1$

$\beta = 1.1, \quad \mu_{min} = 10^{-15}, \quad \delta_T = 0.90, \quad \lambda = 1, \quad \varepsilon = eps$

$\mathbf{x}(1) = \mathbf{Q}\mathbf{y}, \quad \sigma_1 = 2max(|\mathbf{x}(1)|), \quad \mu_1 = 3max(|\mathbf{x}(1)|)$

$\mathbf{e}(1) = \mathbf{Q}(\mathbf{A}\mathbf{x}(1) - \mathbf{y})$

while $(\sigma_k > \sigma_{min})$ **do**

    while $1 \ldots N$ **do**

        $d\mathbf{x}_i(k) = \mathbf{x}_i(k)/(1 + \mathbf{x}_i(k)^2/\sigma_k)$

        $d\mathbf{e}_i(k) = \lambda \mathbf{e}_i(k)/(1 + \mathbf{e}_i(k)^2/\sigma_k)$

        $\bar{\mathbf{x}}(k) = \mathbf{x}(k) - \alpha_k(d\mathbf{x}(k) + d\mathbf{e}(k))$

        $\hat{\mathbf{x}}(k) = \mathbf{g}(\bar{\mathbf{x}}(k), \mu_k, \hat{a})$

        $\mathbf{e}(k) = \beta \mathbf{Q}(\mathbf{A}\hat{\mathbf{x}}(k) - \mathbf{y})$

        $\mathbf{x}(k+1) = \hat{\mathbf{x}}(k) - \mathbf{e}(k)$

    end while

    $\sigma_{k+1} = \delta_\sigma \sigma_k$

    $\mu_{k+1} = \delta_T \mu_k$

    $\alpha_{k+1} = \delta_T \alpha_k$

    $k = k + 1$

end while

Output: $\mathbf{x}(k)$

The initial value $\mathbf{x}(0) = \mathbf{A}^{\dagger}\mathbf{y}$ is selected for evaluation. The σ_0 is set as $2 \times \max_{x_i}\{\mathbf{x}(0)\}$. The stopping criterion is set as $(\sigma_k < \sigma_{min})$. The computational time is determined by δ and σ_{min}. The error term $\mathbf{e}(\mathbf{x})$ is computed in every step to improve the recovery precision. The optimal value of α and β is determined experimentally and, is described later in the next chapter.

3.4 Chapter Summary

The development of sparse recovery algorithms have adopted concepts from wider areas of mathematics and engineering and is ever improving. However there is a fundamental constraint in the problem. The sparse measurement matrix is generally non invertible and hence other methods like iteration need to be done for the reconstruction of the original data. This involves function optimization process using ℓ_0 or ℓ_1 or ℓ_p based objective functions. Also an appropriate inverse matrix for the measurement matrix $\mathbf{A}$ is needed. Many of the currently available algorithms use $A^{\dagger}$ or A^T as the approximate inverse matrix. A general framework using arbitrary inverse matrix $\mathbf{Q}$ for the development of sparse recovery algorithms is proposed. This provides analytical tool for experimenting with various approximations of the matrix inverse for sparse signal recovery. The method is used in the development of two improved algorithms based on ℓ_1 and ℓ_0 minimization, residue estimation and segmented thresholding techniques. The first is based on iterative segmented thresholding of ℓ_1 residue with the inverse operation. The second is based on segmented thresholding of polynomial approximation of ℓ_0 function. The logic for selecting residue and minimizing it for arriving at optimal sparse solution is described. A range alterable segmented thresholding function is proposed and used in the final stage of the iteration. The next chapter presents the cascaded computational network implementation of the proposed ℓ_0 minimization based STXEL0 algorithm for real time use.

Chapter 4

Function Dictionary Based Network Implementation for Sparse Signal Recovery

The segmented thresholded X-L0 E-L0 algorithm described in (3.43) is configured as a cascade network consisting of three parts. One part for error computation, the second part for ℓ_0 minimization using a dictionary of approximation functions obtained from the gradient of the function $1 - 1/(1 + \frac{x^q}{(\sigma/a)^q})$ and the third part is the segmented thresholding function. In the first part the vector $\mathbf{x}(k)$ to be minimized is set as the weights of the network. The rows of the measurement matrix $(\mathbf{A}_i)$ are taken as inputs to the network. The element vise weighted inputs are added with scale factor (-1) to the corresponding element of the measurement $\mathbf{y}_j$. The resulting scalar value is taken as input for the multiply accumulate unit (MAC), where each column of the inverse matrix $\mathbf{Q}_j$ is weighted with this scalar and accumulated. The multiply and accumulate process is continued for M times and the resulting vector is stored in the output register. In the second stage of the network, x_i's $(i = 1 \ldots N)$, and the output of the first stage are ℓ_0 minimized using a set of dictionary functions. The modified values are scaled and accumulated to generate the output of the second stage. The ℓ_0 minimization is performed using the dictionary of basis functions computed using (4.1), derived from the gradient of the ℓ_0 approximation; where q is an even number and $p = 2(\frac{a}{\sigma})^q$.

$$F(x) = \frac{x^{q-1}}{1 + px^q} \tag{4.1}$$

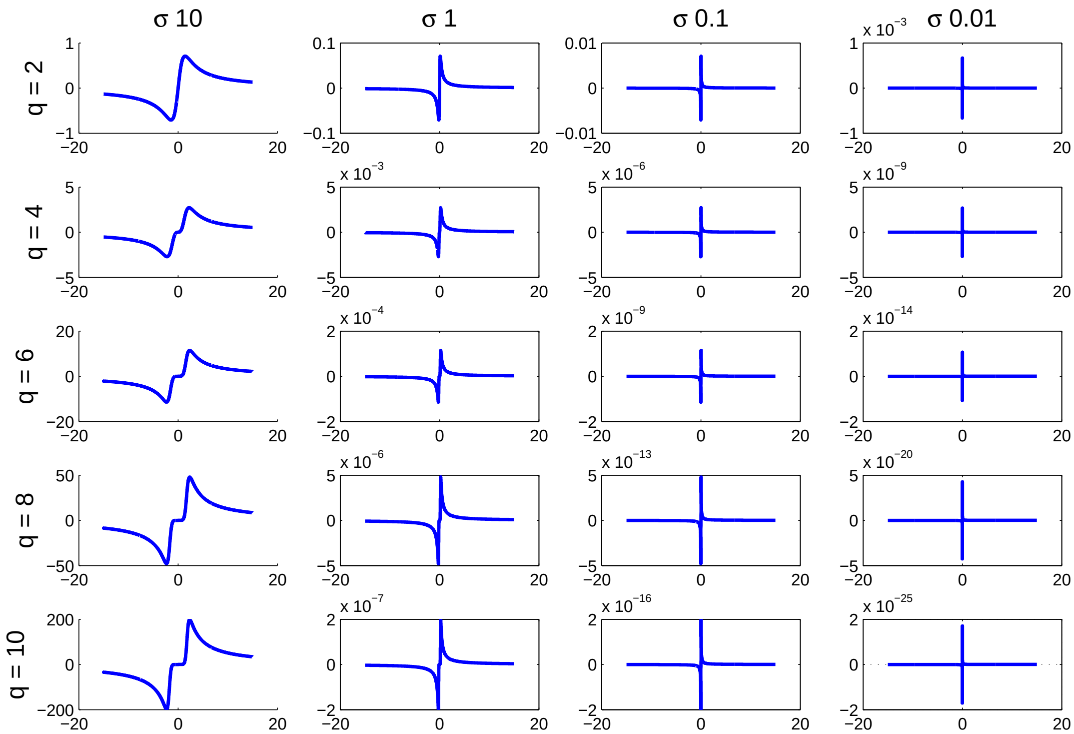


Figure 4-1: The dictionary of ℓ_0 gradient minimization basis functions generated from $F(x) = \frac{x^{q-1}}{1+px^q}$ for various values of q and σ.

Note: It can be seen from the figure that as σ approach 0, the gradient based corrections are applied only to values near zero. Values close to zero are corrected heavily compared to higher values.

The output is thresholded using the segmented thresholding function. The dictionary of ℓ_0 gradient minimization basis functions generated for various values of q and σ are shown in Figure 4-1. Based on the architecture described a network schematic for implementation of the polynomial basis function dictionary based cascade network for sparse signal recovery is illustrated in Figure 4-2. After every iteration the basis function of the second stage network is changed to next function table corresponding to lower σ value. The process is continued till the basis function table corresponding to $\sigma_{min} = x_n/\sqrt{2\ln(\alpha_0)}$ is selected.

The basis functions corresponding to various values of threshold upper limit a and threshold profile q are generated and stored in RAM. The magnitude plot of this dictionary matrices are shown in Figure 4-3. The computational complexity of this architecture can be estimated as (4.2), where n is the number of iterations.

$$N.Ops = n \times [N(2M + 3)mul + N(M + 4)add] \tag{4.2}$$

Based on the initial value chosen for iteration and the exit condition selected, the number of iteration can be approximated as (4.3) for σ_k decrease factor $\delta_\sigma = 0.9$; where $|x_0|$ is the initial value chosen for iteration. A detailed discussion on the value approximation rate of the neural network with polynomially decaying activation function is given in [55].

$$n \approx 20 \log \left(\frac{\max |x_0|}{\sigma_{min}} \right) \tag{4.3}$$

The number of processor operations can be simplified as (4.4) if the data is normalized and all operations are performed using MAC unit.

$$N.Ops \approx 40MN \log \left(\frac{1}{\sigma_{min}} \right) \tag{4.4}$$

The computational load on the real-time recovery of sparse signal from compressed measurements acquired at F_s frames per second rate is approximated as (4.5), where eps is the computational machine precision.

$$N.Ops/sec \approx 40MNF_s \log \left(\frac{1}{eps} \right) \tag{4.5}$$

Considering $eps = 10^{-5}$, matrix size 30×50 and frame rate of 20, the processing load is expected to be $\approx 6.0\ MFLOPS$. Due to smaller throughput requirement, this network can be implemented on conventional low profile computational platforms. The evaluation of IoT platform board $AM3358$ for the implementation of networked data acquisition system is given in [48].

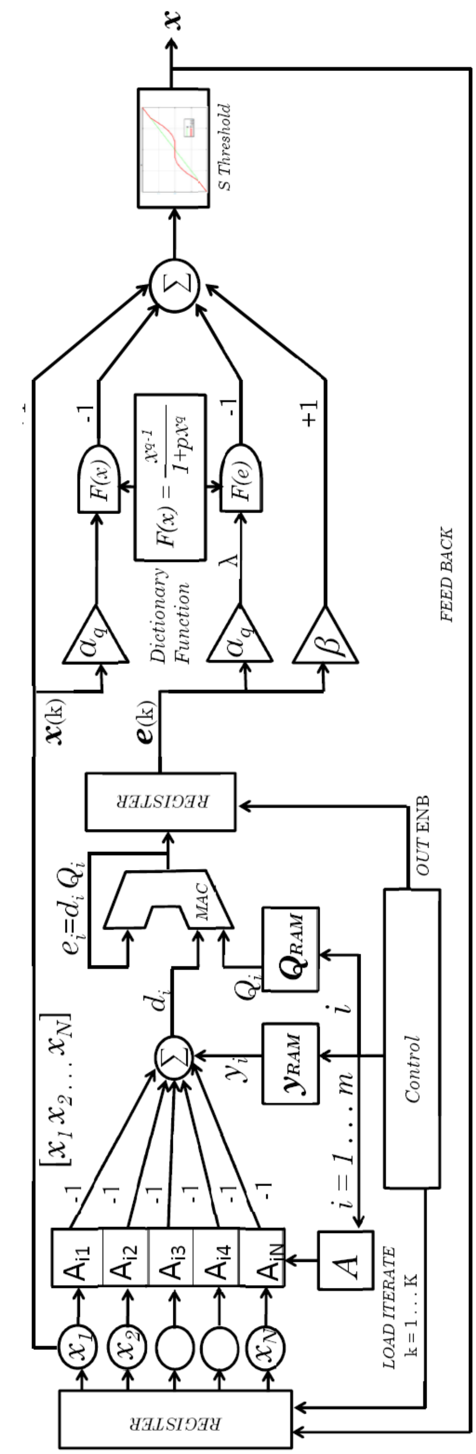
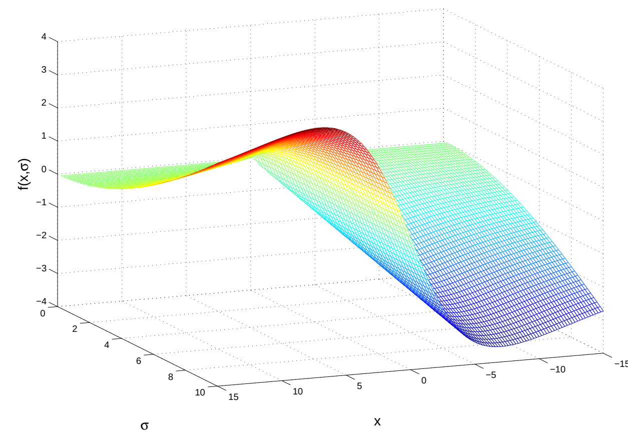
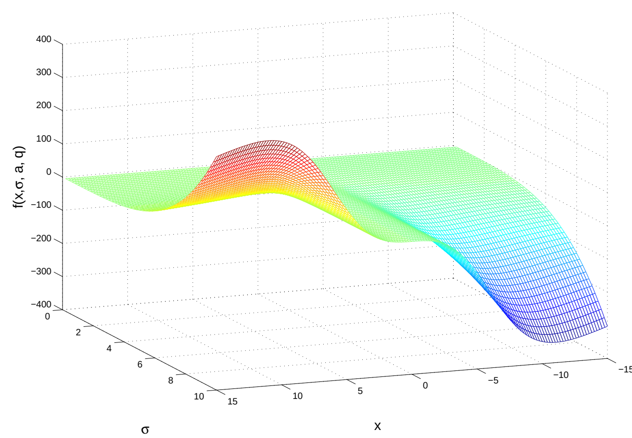


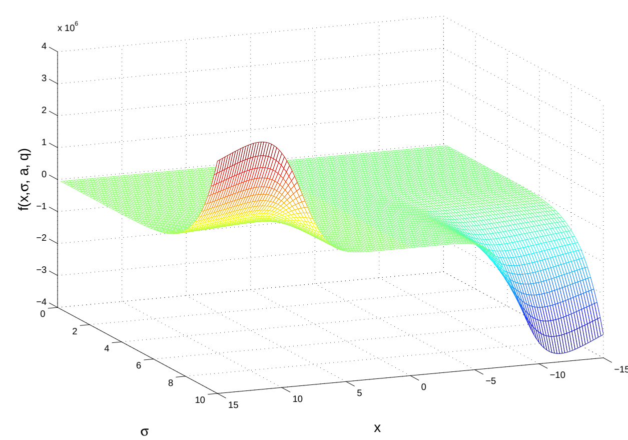
Figure 4-2: The network architecture of the basis function dictionary based cascade network for sparse signal recovery developed from STXEL0 algorithm



(a)



(b)



(c)

Figure 4-3: The ℓ_0 minimization basis functions generated from $F(x) = \frac{x^{q-1}}{1+px^q}$.

Note: The functions are generated for various threshold limit a and the exponent q. The input to the function table are $-15 < x_i < 15$ and $0 < \sigma < 10$. (a) The ℓ_0 gradient minimization basis function set for $a = 5$, $q = 2$. (b) The basis function set for $a = 5$, $q = 4$. (c) The basis function set for $a = 5$, $q = 8$.

4.1 Simulation and Evaluation

The algorithms are tested in various sparse signal recovery scenarios. Three types of sparse signals are used in the testing: (1) gray scale images, (2) sparse spike signals with sparsity between 1 to 15 and (3) sparse pulse signal with pulse width varied from 1 to 15. Benchmark images of size 256×256 pixels are segmented into 16×16 pixels with no overlapping and used in the image acquisition and recovery process. DCT matrix of size 16×16 is used as the basis transformation matrix for converting the image into sparse data. Each image segment is transformed to a vector of size 256 and then compressed sensed into a vector of size 100 using a measurement matrix of size 100×256. The measurement matrix are generated from normal distribution $\mathcal{N}(0, 1)$. The measurements are simulated using (4.6), where n_L is the relative noise strength and $\mathbf{w}_n = \mathcal{N}(0, 0.01)$ is white Gaussian noise.

$$\hat{\mathbf{y}} = \mathbf{Ax} + n_L \frac{\|\mathbf{ADx}\|}{\|\mathbf{w}_n\|} \mathbf{w}_n \tag{4.6}$$

The measurement $\mathbf{y} = \mathbf{Ax}$ is perturbed with noise of relative strength $-60dB$ to $-20dB$ ($n_L = 0.001$ to 0.1). Individual image blocks are reconstructed from the perturbed measurements $\hat{\mathbf{y}}$'s and combined together to form the complete image. The throughput estimated is 102.4 $MFLOPS$ for 20 frames per second reconstruction rate. The schematic representation of process involved in the image acquisition and reconstruction is illustrated in Figure 4-4. In the second simulation scenario, the sparse vectors of length 50 are selected from normal distribution $\mathcal{N}(0, 1)$. The elements of the measurement matrix of size 30×50 are taken from normal distribution $\mathcal{N}(0, 1)$ and the columns are later normalized. The reconstruction quality is estimated in terms of average SNR (4.7) and the probability of exact signal recovery (4.8), where $\hat{\mathbf{x}}$ is the reconstructed signal.

$$avg.SNR = 10 \log \left(\frac{\|\mathbf{x} - \hat{\mathbf{x}}\|^2}{\|\mathbf{x}\|^2} \right) \tag{4.7}$$

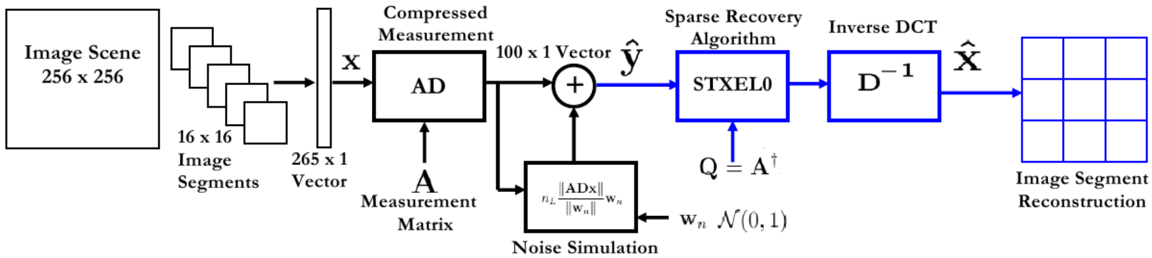


Figure 4-4: Simulation setup for algorithm evaluation.

Note: The image scene is captured as segmented 16×16 *block and converted to* 265 *vector. This vector is acquired as* 100 *compressed measurements using an augmented measurement matrix* **AD***, which is the product of chosen measurement matrix* **A** *and the DCT matrix* **D***. Sparse recovery algorithms are used to reconstruct the sparse components. The image segment is recovered using Inverse-DCT of the sparse components.*

The probability of exact signal recovery is estimated by comparing the support indexes of the non zero value in the reconstructed signal and the original signal.

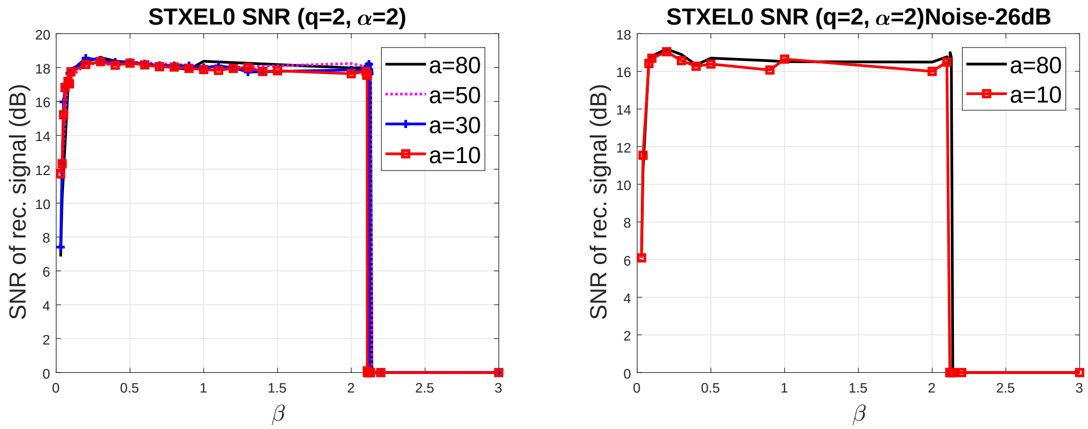
$$RecoveryProb. = 1 - \frac{\max(\|\mathbf{x}\|_0, \|\hat{\mathbf{x}}\|_0) - \|\mathbf{x} \cap \hat{\mathbf{x}}\|_0}{\max(\|\mathbf{x}\|_0, \|\hat{\mathbf{x}}\|_0)} \tag{4.8}$$

where, the operator $\cap$ determine the support locations common to the original signal $\mathbf{x}$ and the reconstructed signal $\hat{\mathbf{x}}$. The computational complexity of the algorithms are measured in terms of the algorithm execution time, relative to the benchmark algorithm SL0 [26]. All the simulations are performed in MATLAB, running in 64-bit MS Windows-8 OS on Intel i3 dual core 1.9 GHz processor with 12 GB RAM. The recovery performance are compared with benchmark sparse recovery algorithms and recently published iterative proximal projection smoothly clipped absolute deviation IPPSCAD [37]. The initial value for the iteration is set as $\mathbf{x}(0) = 2 \times \mathbf{A}^\dagger\mathbf{y}$. The basis functions for STXEL0 algorithm is generated for the parameter set $q = 2$ and $a = 10, 30, 50, 80$; where a is the last element of $\hat{a}$. The approximate inverse is taken as $\mathbf{Q} = \mathbf{A}^\dagger$. The segmented threshold is generated using the parameter set $\hat{a} = [0.74, 1.25, 1.9, 3.0]$. The gradient scale factor $\alpha_0 = 0.7$ and residue scale factor $\beta = 0.25$ are experimentally determined. The effect of α_0 and β in the convergence of STXEL0 algorithm is discussed in the following subsection. The stopping criterion is set as $\sigma_{min} = 10^{-8}$ and the σ_k reduction factor is set as $\delta_\sigma = 0.95$

4.1.1 Influence of Scale Factors α_0, β and q on SNR

The effect of the gradient scale factor α_0 and residue scale factor β are studied through simulations. The noise perturbation in the measurement is set to the minimum $-60dB$. To study the effect of β on the reconstruction performance, the α_0 is set to 2.0. The simulations are performed with threshold limit $a = 10, 30, 50, 80$ and the value of β is varied from 0.03 to 3.0. The images of size 256×256 from MATLAB image processing repository are used in the simulation. The images are segmented as discussed earlier. 10 Nos of image-simulations for each value of β is performed and the SNR of reconstructed images are averaged. The SNR increases as β increases from 0.03 and value reaches maxima near $\beta = 0.25$. Segmented reconstruction fails when β is increased above 2.121. The simulations were repeated for threshold limit $a = 30$. The SNR variation with respect to β follows the same profile as earlier. The SNR peak is also observed near $\beta = 0.25$. The SNR for various values of β is shown in Figure 4-5. The experiments are repeated for noisy case. The noise perturbation is increased to $-26dB$, however the SNR variation with β shows the similar profile. SNR reaches maxima near $\beta = 0.25$ and the reconstruction fails when β is increased above 2.121. The observation is consistent for noisy and noise free reconstruction cases.

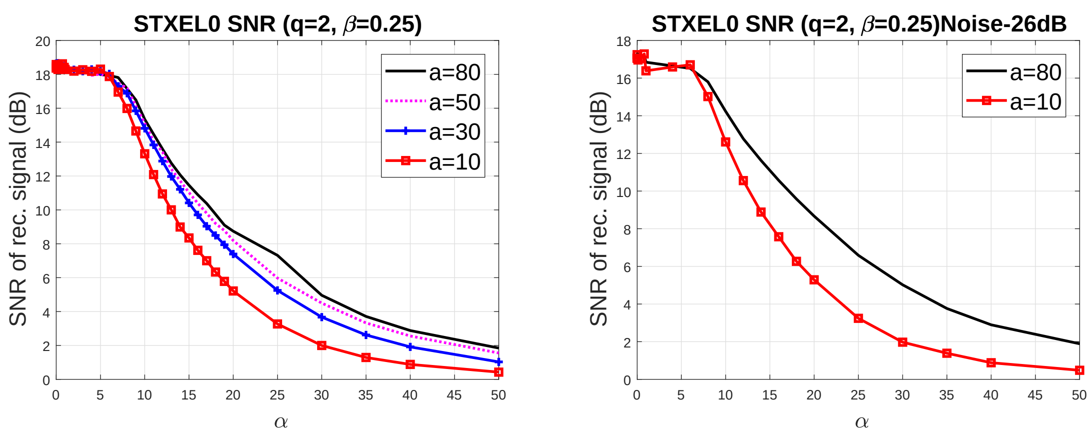
To study the effect of the gradient scale factor α_0 on SNR of the recovered signal, β is set to 0.25 and the degree of ℓ_0 approximation polynomial is set as $q = 2$. The α_0 is varied from 0.1 to 50 and the SNR of the recovered signal is determined. The experiment is repeated for threshold limits $a = 10, 30, 50, 80$, without noise perturbation. The variation of SNR is found to be minimal till α_0 is varied from 0.1 to 5. The SNR reduces when $\alpha_0 > 5.0$. However, when noise perturbation is increased to $-26dB$, SNR of the recovered signal reduces when $\alpha_0 > 2.0$. The observed variation of SNR with respect to change in α_0 is shown in Figure 4-6.



(a) (b)

Figure 4-5: SNR of the STXEL0 recovered signal for various values of residue scale factor β.

Note: (a) SNR during noise free case: β *is varied from* 0 *to* 3, $\alpha_0 = 2.0$ *and* $q = 2$. *(b) SNR when the measurement is perturbed with Gaussian noise of relative strength* $-26dB$ *and* β *is varied form* 0 *to* 3 *while* $\alpha_0 = 2.0$ *and* $q = 2$. *Consistently, SNR peaks near* $\beta = 0.25$ *and reconstruction fails for* $\beta > 2.121$.



(a) (b)

Figure 4-6: SNR of the STXEL0 recovered signal for various values of the gradient scale factor α.

Note: (a) SNR in noise free case: α *changed from* 0 *to* 50, $\beta = 0.25$ *and* $q = 2$. *The SNR decreases when* $\alpha_0 > 5$. *(b) SNR when the measurement is perturbed with Gaussian noise of relative strength* $-26dB$ *and* α *changed from* 0 *to* 50 *while* $\beta = 0.25$ *and* $q = 2$. *SNR decrease when* $\alpha_0 > 2$.

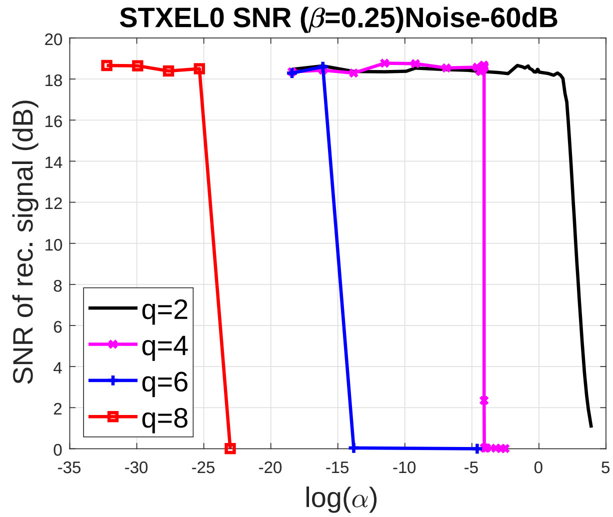


Figure 4-7: SNR of the STXEL0 recovered signal for different values $q = 2, 4, 6$ and 8 when $\beta = 0.25$.

Note: The measurement is perturbed with white Gaussian noise of relative strength $-60dB$

The effect of degree of polynomial basis function in the performance of the STXEL0 is analysed. Image reconstruction performance is used in the simulation. The measurements are perturbed with white Gaussian noise of relative strength $-60dB$. The reconstruction is performed using basis functions generated using $q = 2, 4, 6$, and 8. The parameters β is set to 0.25 and the gradient scale factor α_0 is varied from 0.1 to 50. The optimal value of α_0 is found to differ with respect to the q. The SNR variation with respect to α_0 for various values of q is shown in Figure 4-7. The optimal value of α_0 is found to be ≈ 2.0 for $q = 2$. In higher degree ℓ_0 approximations with $q = 4$, $q = 6$ and $q = 8$, the optimal value of α_0 is found to be of the order 10^{-4}, 10^{-16} and 10^{-26} respectively. The implementation of higher order polynomial versions of STXEL0 algorithm ($q = 6$ and $q = 8$) is infeasible in low profile devices, since the gradient scale factor α_0 corresponding these are negligibly small compared to the computational precision. The optimal values $\alpha_0 = 2.0$, $\beta = 0.25$ and $q = 2$, obtained from this simulation are used in the proceeding algorithm evaluations.

4.2 Sparse Recovery using the Proposed ℓ_1 Based ISTRP Algorithm

Two types of sparse signals of length 50 are generated for the simulation. Sparse spikes having 2 to 10 non zero elements and sparse pulses with 2 to 10 continuous sample wide. These sparse signals are measured using normalized *i.i.d.* measurement matrix of size 30×50 and then recovered using various algorithms. The simulations are performed 100 times for each signal, using different measurement matrix and the average measures from the reconstruction are recorded. Seven different classes of algorithms are simulated and compared with the proposed method. The results are tabulated in Table 4.1 and Table 4.2 The SNR of recovery and execution time are taken into consideration. It can be found from the result that the proposed ℓ_1 based projection and thresholding method give finite improvement over the existing methods. To evaluate the algorithm in more realistic scenario electric field mill signal acquired during lightning is used in the simulations. The lighting flash from cumulonimbus clouds span for 50 to 200 ms with a discharge peak near 10 μs and the voltage transients persists for duration typically < 200 μs. In the case of periodic sampling this signal should be sampled with frequency > 5000 Hz. However, most of the time the measurements are zero or a constant bias value. The data equivalent to 1000 Nyquist samples are acquired in 600 samples per frame. The processor throughput estimated for recovery of the signal from this frame is 600 $MFLOPS$. The signal reconstructed using various benchmark algorithms and the proposed method are shown in Figure 4-8 and the results are summarized in Table 4.3. The analysis of the results shows that the iterative segmented threshold residue mapping gives an improvement of $0.04dB$ over the benchmark algorithm iterative proximal projection-SACD [37], indicating the possibility of improvement using segmented thresholding. The second proposed ℓ_0 minimization with segmented thresholding method is evaluated in image recovery scenario.

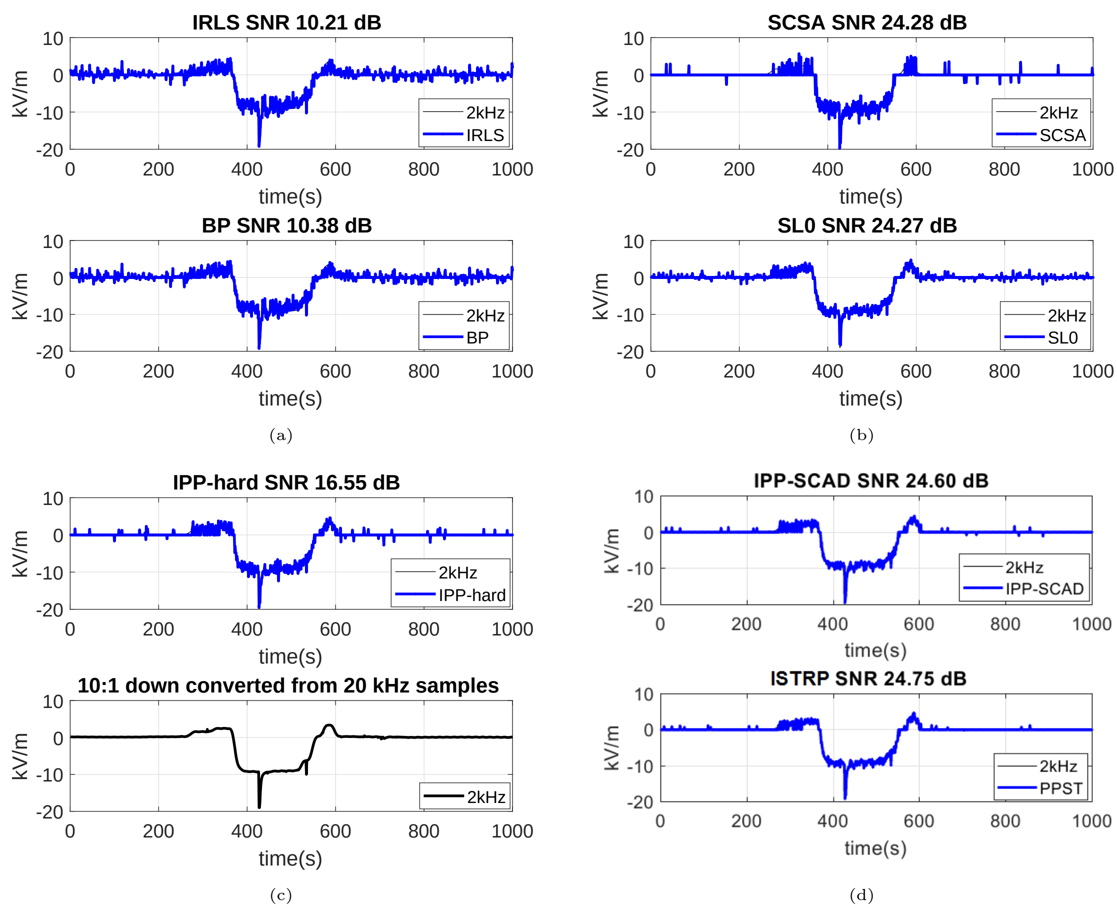


Figure 4-8: Comparison of electric field mill signal reconstructed using ISTRP algorithm and other optimization and thresholding based algorithms

Note: Electric field mill signal during lighting acquired at 20 kHz sampling rate with 10:1 down conversion. The same signal is acquired using compressed sensing and reconstructed using various algorithms. The function optimization based algorithms (SCSA, SL0) perform better compared to the ℓ_1 *and* ℓ_p *based methods (BP, IRLS). The function optimization and projection with thresholding algorithms (IPP Hard threshold and IPP SCAD threshold) give the promising results. ISTRP (Algorithm-7) gives 0.04dB advantage over the existing iterative proximal projection algorithm IPPSCAD.*

Table 4.1: **Comparison of sparse recovery algorithms in terms of SNR of the recovered sparse signal.**

	Noise		-26	(dB)			-33	(dB)				nil		Time
	sparsity	10	8	4	2	10	8	4	2	10	8	4	2	SL0
Genre	method													
Greedy	OMP	5.9	10.9	29.1	31.0	3.6	32.0	34.1	38.6	3.2	14.4	30.5	31.7	1.1
ℓ_1	GOMP	8.9	28.7	36.4	43.4	4.9	11.5	41.5	51.1	11.6	10.4	31.7	32.1	0.2
	COSAMP	9.4	30.1	37.6	46.8	5.0	35.3	43.7	52.1	6.5	5.9	31.6	32.1	1.9
	L1 LS	8.8	25.0	33.7	41.5	20.3	16.6	38.4	43.4	12.8	18.5	39.7	44.5	5.6
	YALL	9.0	22.8	30.1	32.8	22.6	16.4	34.6	39.7	14.0	17.4	68.9	76.4	1.0
ℓ_1	BP	9.1	22.8	30.1	32.7	23.8	16.5	34.6	39.6	13.9	27.4	30.5	31.5	2.2
	Homtop	8.5	28.3	34.7	42.6	29.3	16.8	45.9	53.8	13.0	15.6	31.7	32.7	1.2
Lagrangian	PALM	9.1	22.8	30.1	32.7	23.8	16.5	34.6	39.6	13.9	10.0	10.0	10.9	58.8
	DALM	8.5	22.6	31.9	37.9	16.9	16.4	33.9	37.7	10.9	14.5	33.9	38.6	5.7
	RASR	4.6	4.3	29.1	31.9	4.9	8.0	34.5	38.2	2.5	14.0	13.7	13.8	6.0
ℓ_0	SL0	6.5	24.1	29.0	30.8	10.2	31.5	32.4	37.5	12.2	16.1.	16.0	16.0	1.0
ℓ_p	IRLS	9.2	22.6	30.1	32.7	22.0	16.5	34.5	39.6	13.9	28.9	30.8	31.7	7.1
Threshold	FISTA	8.5	22.6	31.9	37.9	16.9	16.4	33.9	37.7	10.9	14.5	33.9	38.6	2.4
	BIHT	4.6	6.5	10.6	23.4	4.4	6.0	11.1	53.8	4.6	5.7	12.1	32.1	4.3
Bayes	EGAmp	8.3	34.1	37.4	39.0	13.2	16.9	43.5	44.9	14.2	22.1	56.5	66.1	6.8
Projected	ISP SL0	7.5	32.9	38.2	45.7	9.6	38.7	46.7	53.8	9.5	91.5	92.4	94.8	2.7
Threshold	IPP hrd	6.9	30.4	38.4	41.2	7.3	35.9	42.7	50.3	6.9	13.4	92.1	95.2	2.7
	IPP mcp	10.3	32.2	37.2	41.7	34.5	37.6	44.4	51.1	90.6	91.5	93.6	95.8	3.4
	IPP scad	10.3	34.3	39.2	48.6	34.4	36.4	43.8	53.0	91.1	91.7	94.0	95.9	3.9
	ISTRP	10.3	34.6	39.7	48.7	34.4	37.0	44.0	53.2	91.5	91.8	94.1	95.7	3.5

Notes: sparse spike signals of length 50 with 2,4,8 and 10 non zero elements are used in the simulation. The sparse signals are acquired using i.i.d. measurement matrix of size 30×50 *and recovered using the indicated methods. EGAmp indicates Expectation Maximization Gaussian approximate message passing. The proposed* ℓ_0 *based ISTRP method performs slightly better compared to other.*

Table 4.2: **Comparison of sparse recovery algorithms in terms of SNR of the recovered sparse pulse.**

	Noise		-26	(dB)			-33	(dB)				nil		Time
	sparsity	10	8	4	2	10	8	4	2	10	8	4	2	SL0
Genre	method													
Greedy	OMP	4.7	10.8	27.9	31.3	3.8	7.3	34.4	39.6	3.4	8.2	31.0	31.7	1.0
ℓ_1	GOMP	7.2	15.4	36.9	43.1	11.6	30.9	42.3	49.0	3.7	17.5	31.7	31.7	0.2
	COSAMP	3.0	11.6	38.4	45.9	0.8	13.2	43.4	53.6	2.3	6.5	31.6	31.9	2.1
	L1 LS	12.7	20.5	28.9	34.5	15.6	22.4	28.5	34.6	10.3	22.6	31.0	34.7	4.7
	YALL	13.0	22.6	28.9	31.9	22.8	29.4	33.6	37.9	12.0	31.6	71.1	58.2	1.0
ℓ_1	BP	13.0	22.5	28.9	31.8	24.7	29.3	33.6	37.8	17.9	31.7	29.5	30.8	2.4
	Homtop	13.0	28.2	38.3	45.8	13.8	30.2	44.2	48.0	10.5	25.8	38.7	32.0	1.0
Lagrangian	PALM	13.0	22.5	28.9	31.8	24.7	29.3	33.5	37.8	17.9	31.7	9.8	10.2	57.3
	DALM	11.1	15.7	23.7	28.9	11.9	17.1	22.8	28.8	8.2	17.7	25.0	28.7	6.4
	RASR	4.6	12.3	28.4	30.9	7.9	10.2	33.6	37.1	5.2	7.2	14.2	13.7	5.9
ℓ_0	SL0	8.3	13.4	27.4	29.7	10.8	20.8	32.1	36.1	11.5	15.2	15.1	15.1	1.0
ℓ_p	IRLS	13.4	21.9	28.9	31.7	22.1	29.2	33.5	37.9	15.3	32.2	29.7	30.8	7.5
Threshold	FISTA	11.1	15.7	23.7	28.9	11.9	17.1	22.8	28.8	8.2	17.7	25.0	28.7	2.6
	BIHT	3.2	4.8	8.6	47.2	4.2	5.6	9.2	54.1	2.9	4.5	11.4	31.8	5.0
Bayes	EGAmp	14.8	38.8	36.5	38.8	43.8	35.5	41.9	43.8	14.6	35.5	48.2	52.9	6.6
Projected	ISP SL0	6.9	13.4	39.1	45.3	25.2	20.3	46.7	54.7	9.1	29.8	93.8	98.0	2.8
Threshold	IPP hrd	6.8	31.2	37.4	45.2	9.4	18.0	43.5	54.0	8.5	29.7	85.0	86.1	2.7
	IPP mcp	11.3	31.0	38.1	45.6	34.1	38.6	45.1	54.0	11.4	31.4	84.9	86.6	3.3
	IPP scad	12.1	30.4	36.9	45.0	34.5	37.7	42.7	49.2	11.4	31.5	83.8	83.5	4.1
	ISTRP	12.2	30.7	36.9	45.2	34.7	37.8	44.2	54.8	11.4	31.5	83.8	83.9	3.4

Sparse signals of length 50 with pulse of width 2,4,8, and 10 samples are used in the simulation. The sparse signals are acquired using i.i.d. measurement matrix of size 30×50 *and recovered using the indicated methods. EGAmp indicates Expectation Maximization Gaussian approximate message passing. The proposed* ℓ_0 *based ISTRP method performs slightly better compared to other.*

Table 4.3: **SNR of the electric field mill signal recovered using various algorithms**

Genre	Algorithm	SNR dB	Rel. Error	No. Iter	Algorithm	SNR dB	Rel. Error	No Iter
Greedy	OMP	3.95	0.4029	568	L1 LS	10.22	0.0950	796
ℓ_1	GOMP	1.44	0.7178	364	YALL	10.35	0.0922	364
	COSAMP	0.05	0.9877	62	EmGmAmp	24.60	0.0035	364
ℓ_1	BP	10.38	0.0917	1	Homtop	10.18	0.0958	796
Lagrangian	PALM	10.38	0.0917	5000	DALM	10.14	0.0969	5000
ℓ_0	SL0	24.27	0.0037	379	IRLS	10.21	0.0953	101
Threshold	FISTA	10.13	0.0969	1000	BIHT	0.05	0.9885	200
Projected	SCAD	3.93	0.4043	364	ISP SL0	22.83	0.0052	364
Gradient	IPP Hrd	16.55	0.0221	364	IPP MCP	24.62	0.0033	364
	IPP SCAD	24.60	0.0035	364	ISTRP	24.75	0.0033	364

Table 4.4: **Recovery performance the algorithms using various inverse matrices on low sampled Images**

λ	0.6	0.4	0.2	0.18	0.9	0.6	0.4	0.2	0.18	0.6	0.4	0.2	0.18
SNR			(dB)					(dB)				(dB)	
$\lambda\mathbf{A}^T$	7.75	7.76	7.44	7.51	7.75	7.81	7.82	7.76	7.83	8.09	8.06	8.03	8.05
XEL0($Q1$)	7.64	7.51	7.51	7.56	7.76	7.89	7.80	7.70	7.79	8.18	8.05	7.99	8.01
XEL0($Q2$)		7.49	*nc*				*nc*				8.07		
STXEL0 ($Q1$)	9.19	*nc*	*nc*	*nc*	9.35	*nc*	*nc*	*nc*		9.53	*nc*	*nc*	*nc*
STXEL0 ($\lambda\mathbf{A}^{\dagger}$)	9.43	9.47	9.53	9.63	9.51	9.53	9.69	9.79	9.65	9.54	9.63	9.85	9.75

$Q1 = (\mathbf{I} + \lambda\mathbf{A^TA})^{-1}\mathbf{A^T}$, $Q2 = \lambda(\mathbf{I} + \mathbf{A^TA})^{-1}\mathbf{A^T}$, *nc : non converging*, *STXEL0 ($Q1$): Segmented Threshold XEL0 with $Q1$ inverse.* *STXEL0 ($\lambda\mathbf{A}^{\dagger}$): Segmented Threshold XEL0 with pseudo inverse.*

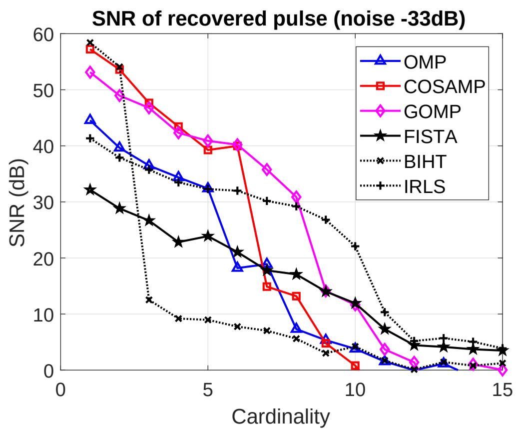
4.3 Reconstruction of Images using the Proposed STXEL0 Algorithm

The benchmark images from MATLAB image processing repository are used in the evaluation. The STXEL0 basis function network architecture supports arbitrary inverse matrix. Three types of inverse operations are used in the evaluation namely $\mathbf{Q} = \lambda\mathbf{A}^T$, $\mathbf{Q} = \lambda\mathbf{A}^\dagger$, $\mathbf{Q} = (\mathbf{I} + \lambda\mathbf{A^T A})^{\mathbf{-1}}\mathbf{A^T}$ and $\mathbf{Q} = \lambda(\mathbf{I} + \mathbf{A^T A})^{\mathbf{-1}}\mathbf{A^T}$. The SNR of the recovered images obtained using these methods are given in Table 4.4. The algorithm fail to converge for some combinations of inverse matrix and input noise level. It is observed that when $Q = \lambda\mathbf{A}^\dagger$ the algorithm performs well and converges for wider range $\lambda > 0.18$. This inverse matrix is used in the further evaluations.

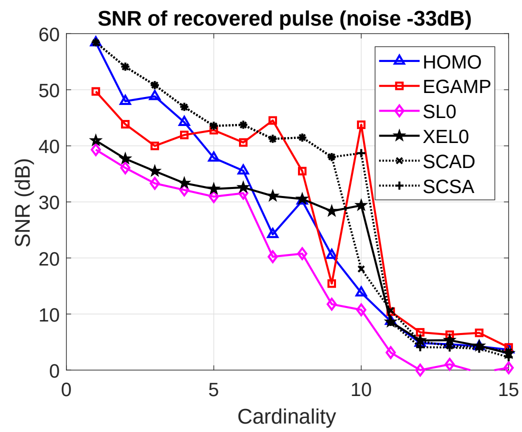
The experimental evaluation the SNR of the recovered pulse signal obtained from the proposed method and existing methods are shown in Figure 4-9, for comparison. The probability of support recovery of these algorithms are shown in Figure 4-10. The experiments are repeated 50 times and the average SNR value of the reconstructed sparse signals are plotted. The results are recorded when the measurements are perturbed with $-33dB$ white noise. It can be seen from the SNR values that all algorithms performs well in reconstruction of the spike signal when measurements are noise free. When the noise level is $-26dB$, SNR of $10dB$ is achieved in the reconstruction of high-sparse signals. ($K = 10$).

The SNR of benchmark images recovered using various algorithms are given in Table 4.5. The comparison of recovered images using various algorithms are shown in Figure 4-11 to Figure 4-14. The ℓ_0 gradient basis function dictionary based signal and error minimization network performs better compared to other benchmark algorithms, in terms of the SNR of recovered signal and the time for computation. SL0 execution time is taken as benchmark for comparison. The computation time of other algorithms are expressed as multiples of SL0 recovery time. The computations time is recorded in the last column of the performance comparison Table 4.5. In case of image recovery, the algorithm performance varies with scenario. Algorithms like

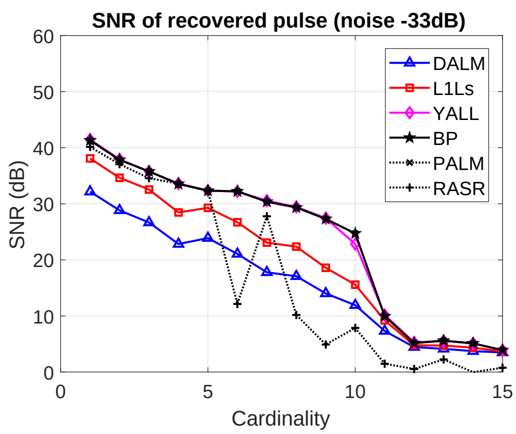
YALL (ℓ_1), IRLS (ℓ_p), and FISTA ($Thresholding$) give performance measures comparable to the proposed methods. These algorithms performs in par with proposed method in image reconstruction scenario, but, with larger processing demand.



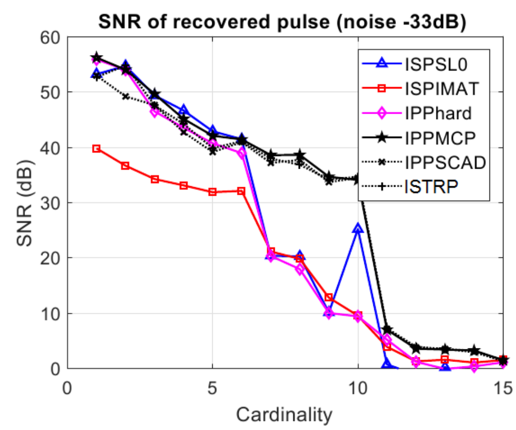
(a)



(b)



(c)



(d)

Figure 4-9: SNR of recovered sparse pulse signal when reconstructed from noisy $(-33dB)$ measurements.

Note: XEL0 represent the STXEL0 algorithms without segmented thresholding. ISTRP represent proposed ℓ_1 based minimization method with segmented thresholding.

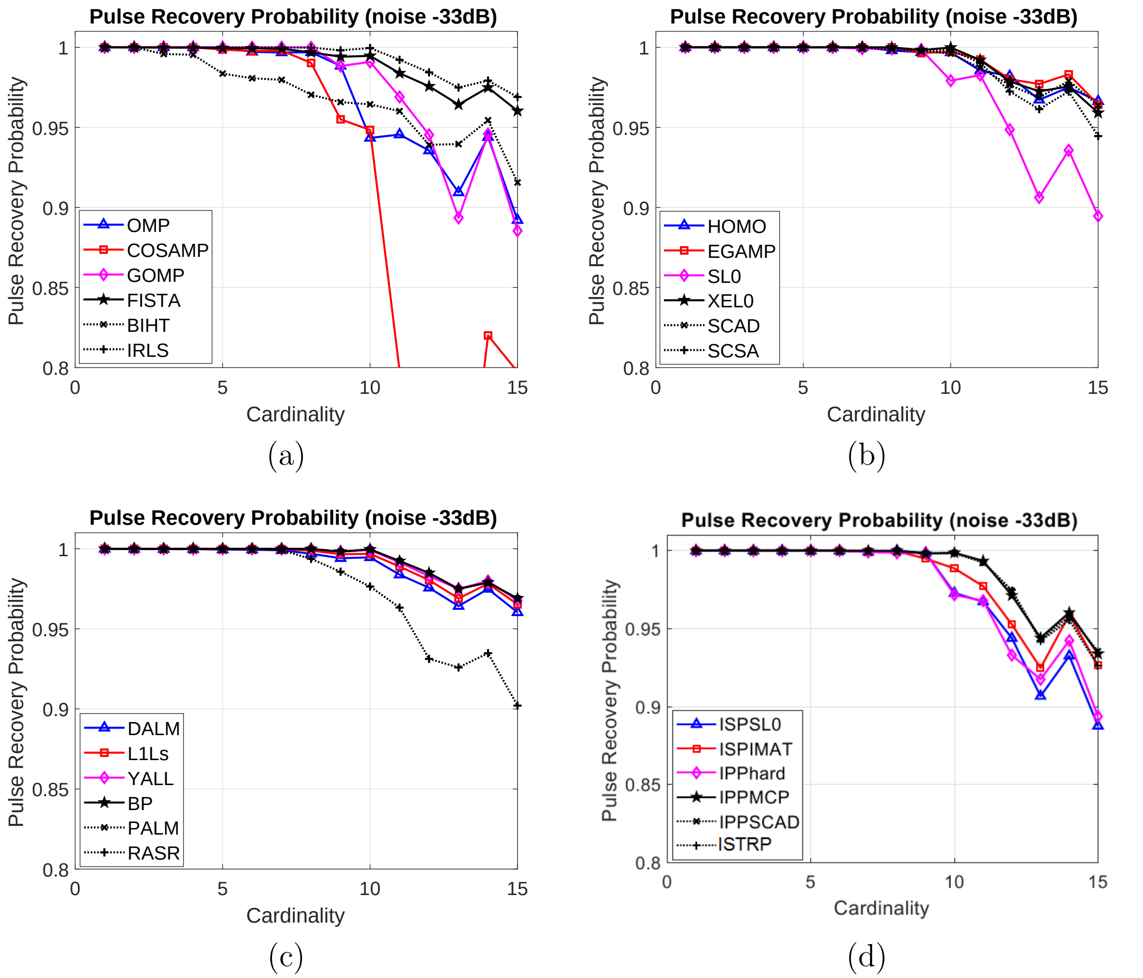


Figure 4-10: Probability of exact support reconstruction of the sparse pulse signal reconstructed from noisy ($-33dB$) measurements.

Table 4.5: **SNR of the image reconstructed using various algorithms**

	Noise	-26dB				-33dB				nil				Time
	Algorithm	I1	I2	I3	I4	I1	I2	I3	I4	I1	I2	I3	I4	
Greedy	OMP	15.1	14.6	16.4	6.9	17.8	15.8	18.8	7.0	18.8	16.1	19.8	6.9	14.7
ℓ_1	COSAMP	13.5	12.0	15.4	6.0	13.5	12.0	15.4	6.0	13.5	12.0	15.4	6.0	0.2
	GOMP	18.4	16.4	19.6	7.5	19.1	16.8	20.4	7.5	19.2	16.6	20.3	7.2	0.2
	L1 LS	16.1	15.2	16.1	9.4	16.8	16.4	19.1	9.4	18.9	17.3	14.9	9.2	40.7
	YALL	18.1	17.6	19.7	9.5	21.0	19.0	21.8	9.7	21.9	19.4	22.9	9.7	0.5
ℓ_1	BP	-2.0	-2.6	-2.4	-2.5	-2.0	-2.3	-2.1	-2.2	-2.0	-2.3	-2.4	-2.0	9.1
	Homtop	18.1	17.5	19.6	9.4	21.0	18.9	21.8	9.7	21.9	19.4	23.2	9.6	1.8
Lagrangian	DALM	18.1	17.5	19.6	9.4	21.0	18.9	21.8	9.7	21.9	19.4	23.2	9.6	1.1
	PALM	18.1	17.5	19.6	9.4	21.0	18.9	21.8	9.7	21.9	19.4	23.2	9.6	18.8
ℓ_0	SL0	15.3	15.1	16.9	7.2	18.7	16.5	19.4	7.4	19.8	17.2	20.7	7.4	1.0
ℓ_p	IRLS	18.1	17.5	19.6	9.4	20.9	18.9	21.8	9.7	21.9	19.4	22.9	9.6	40.0
Threshold	FISTA	18.0	17.5	19.6	9.4	20.9	18.9	21.7	9.7	21.9	19.4	22.9	9.6	1.9
	BIHT	13.5	12.0	15.4	6.0	13.5	12.0	15.4	6.0	13.4	12.0	15.4	6.0	0.8
Bayes	EGAmp	18.6	16.8	20.0	8.8	18.8	16.7	20.2	8.8	18.9	16.8	20.4	8.5	7.4
	SCAD	18.3	16.9	20.1	8.2	20.1	17.8	21.1	8.0	0.2	-0.2	-0.2	0.1	4.4
	SCSA	18.9	17.1	20.3	8.2	20.0	17.6	21.2	8.4	21.9	19.4	23.1	9.6	2.0
Projected	ISP IMAT	16.7	16.4	18.1	8.4	19.9	17.9	20.6	8.4	21.0	18.5	22.1	8.4	0.2
Gradient	IPP SCAD	18.3	16.5	19.7	7.6	19.6	17.2	20.6	7.9	20.6	17.9	21.5	8.1	3.4
Threshold	ISTRP	17.8	16.2	19.3	7.6	19.4	17.1	20.4	7.9	20.6	17.9	21.5	8.0	2.3
	XEL0	17.7	17.3	19.3	9.0	20.8	18.8	21.6	9.3	21.9	19.3	23.0	9.3	0.4

*Notes: Images recovered from noisy measurements of 16 $\times$ 16 blocks compressed sensed data. I*1 *Barbera, I*2 *Starfish, I*3 *Airplane, I*4 *SAR ships. Measurement matrix used is :* **AD**. *Reconstruction matrix used is :* **A**. *Image recovery done using* $\mathbf{x} = \mathbf{D^{-1}z}$ *where* $\mathbf{D}$ *is 16 $\times$ 16 DCT matrix.*

4.3.1 Convergence and Execution Time

The convergence condition of the algorithm using ℓ_0 thresholding method is described in (3.48) and (3.50). The convergence of the proposed ℓ_1 and the ℓ_0 based thresholding methods are studied by analyzing the variation of the internal parameters of the algorithms. The parameter studied are variation from the original signal $|\mathbf{x}(k)-\mathbf{x}_{org}|$, the internal residue $\mathbf{r}(k)$, projected residue $\mathbf{Qr}(k)$, the Lagrangian function gradient, projected measurement $\mathbf{Ax}(k)$, delta change in $d\mathbf{x}$ and $d\mathbf{r}$. All parameters internal to the algorithms shows decreasing tread indicating convergence. The Figure 4-15 shows the internal parameter convergence. The convergence of the error is evaluated by computing the error in every iteration as shown in Figure 4-16(a). The ℓ_1 based thresholding algorithm IPPSCAD and ISTRP follows same profile. When the data is sparse spike or sparse pulse, it is found that the proposed ℓ_1 thresholding method ISTRP take comparable convergence time. However, for image reconstruction it is $1.5\times$ times faster compared to other algorithm with similar reconstruction performance. The algorithms with performance same as the proposed methods in terms of SNR of recovery are YALL, DALM, Homotopy and FISTA, but, at the expense of computational time. These methods have the computational load of $1.2\times$, $3\times$, $4\times$ and $5\times$ times respectively compared to the proposed STXEL0 method. To study the effectiveness of thresholding on the new algorithm, simulations are carried out with and with out thresholding. The convergence profile obtained is shown in Figure 4-16(b). The initial relative error is large for ℓ_1 based methods and reduce to 0.1 with in 50 iterations. The initial relative error in the ℓ_0 based methods are half of ℓ_1 based methods. However the error reduction to 0.1 takes the same number of iterations. When thresholding is not applied XEL0 takes nearly double the iterations to converge.

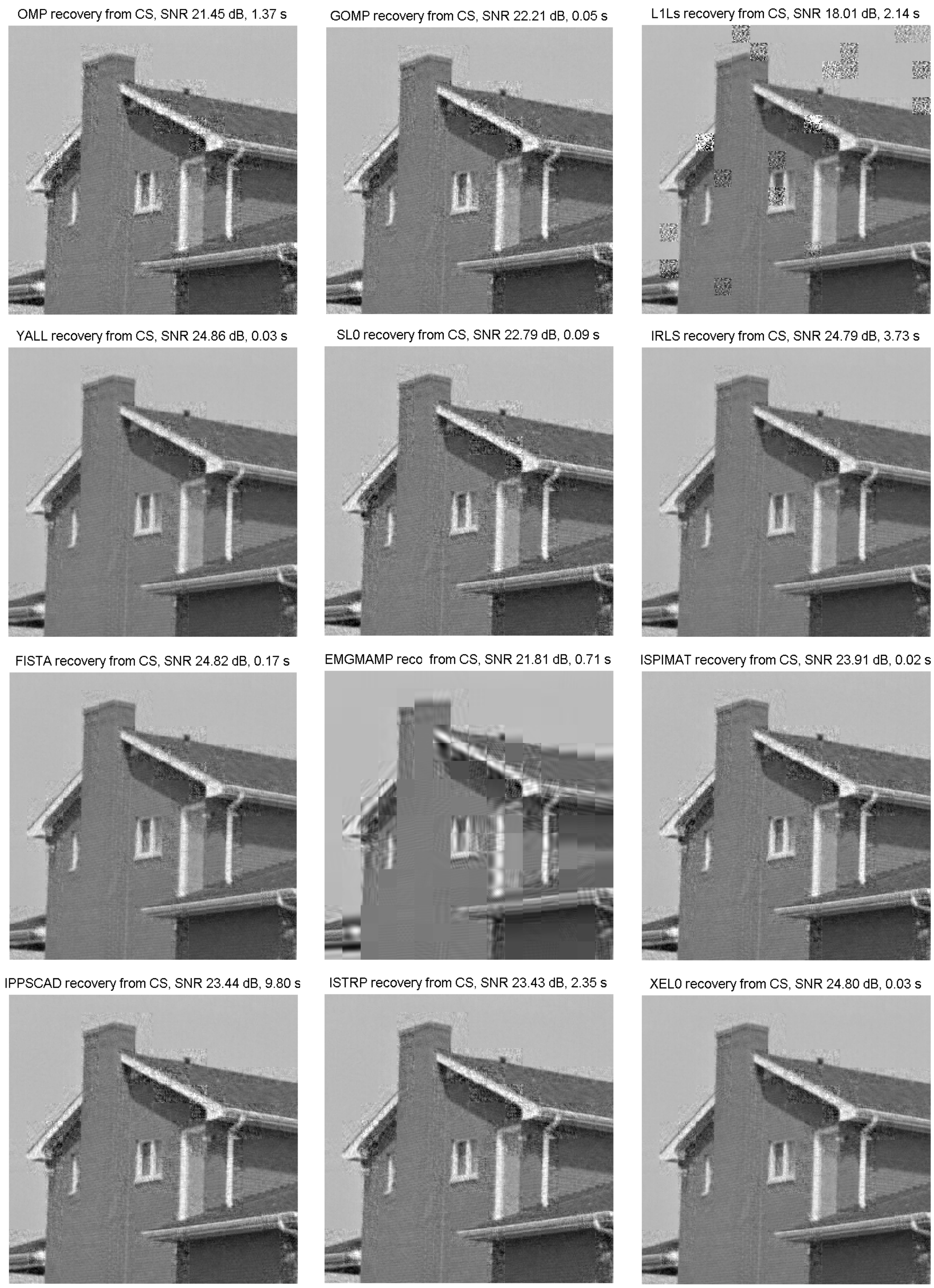


Figure 4-11: The house-image reconstructed from compressed measurements

Note: Image is captured through noise free compressed sensing of 16×16 *blocks of image scene. The reconstructed image using various algorithms are shown. The SNR achieved and the algorithm execution time are given in title of each image.*

Figure 4-12: The Barbara-image reconstructed from compressed measurements

Note: Image is captured through noise free compressed sensing of 16×16 *blocks of image scene. The reconstructed image using various algorithms are shown. The SNR achieved and the algorithm execution time are given in title of each image.*

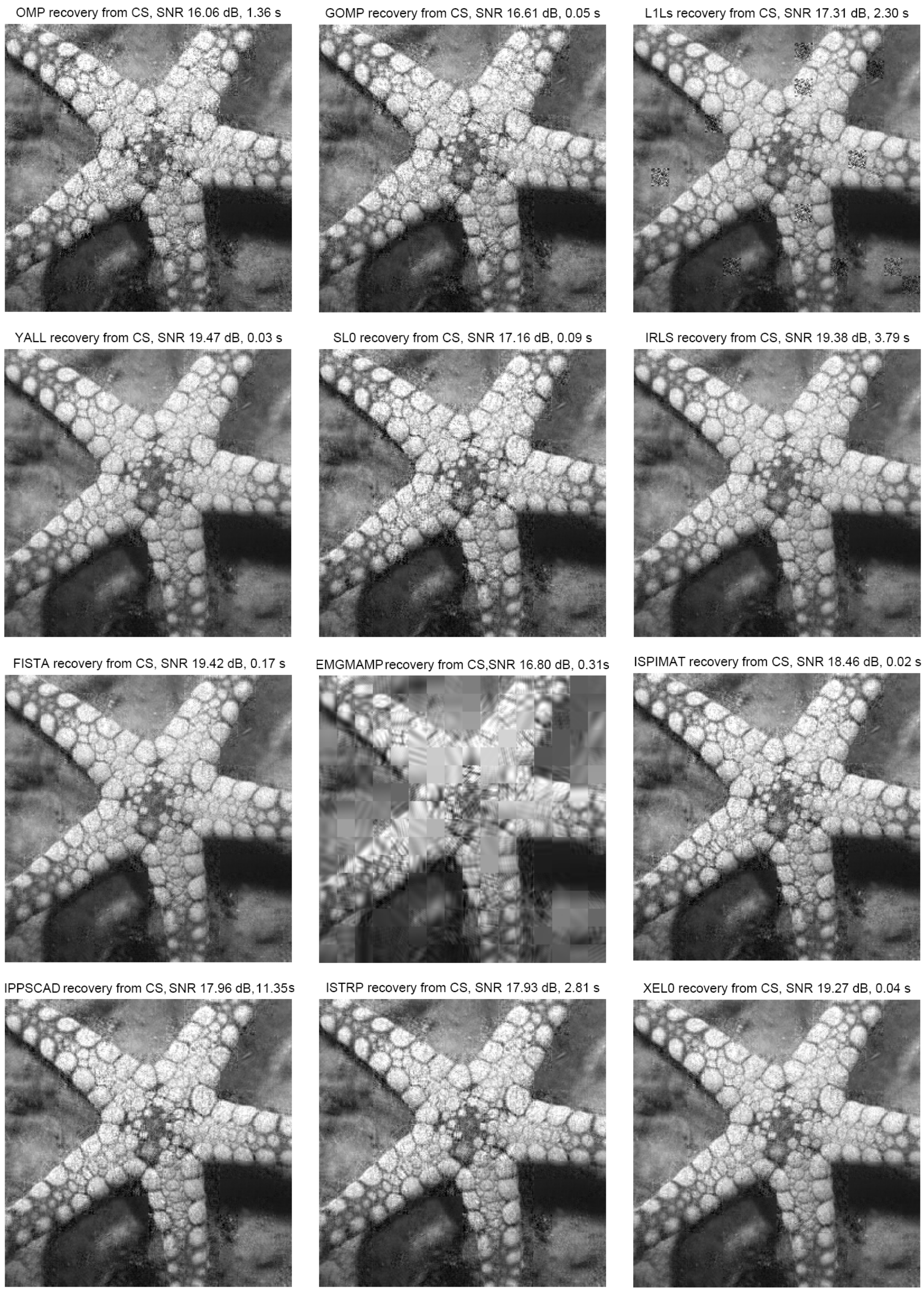


Figure 4-13: The starfish-images reconstructed from compressed measurements

Note: Image is captured through noise free compressed sensing of 16×16 *blocks of image scene. The reconstructed image using various algorithms are shown. The SNR achieved and the algorithm execution time are given in title of each image.*

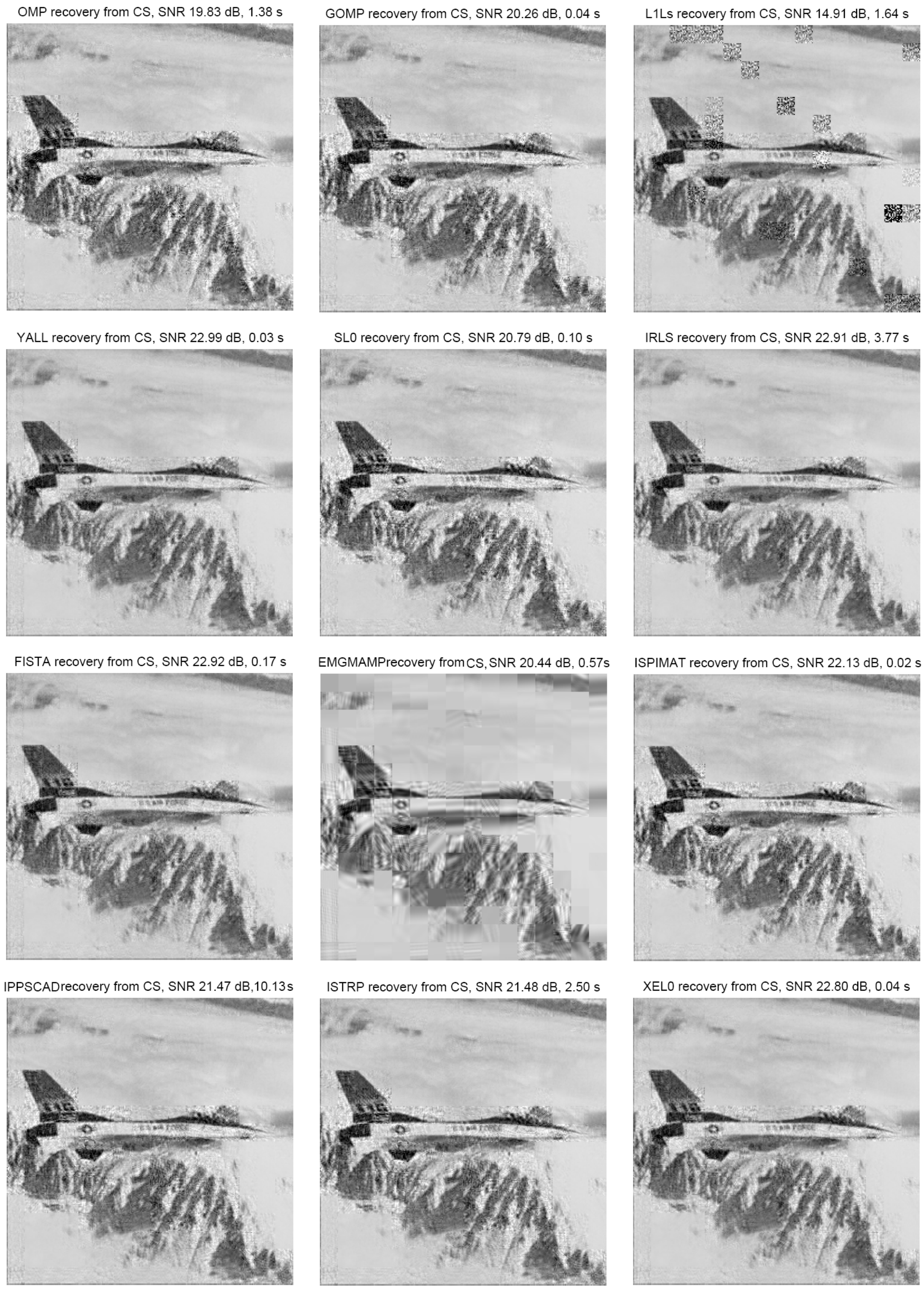


Figure 4-14: The airplane-images reconstructed from compressed measurements

Note: Image is captured through noise free compressed sensing of 16×16 *blocks of image scene. The reconstructed image using various algorithms are shown. The SNR achieved and the algorithm execution time are given in title of each image.*

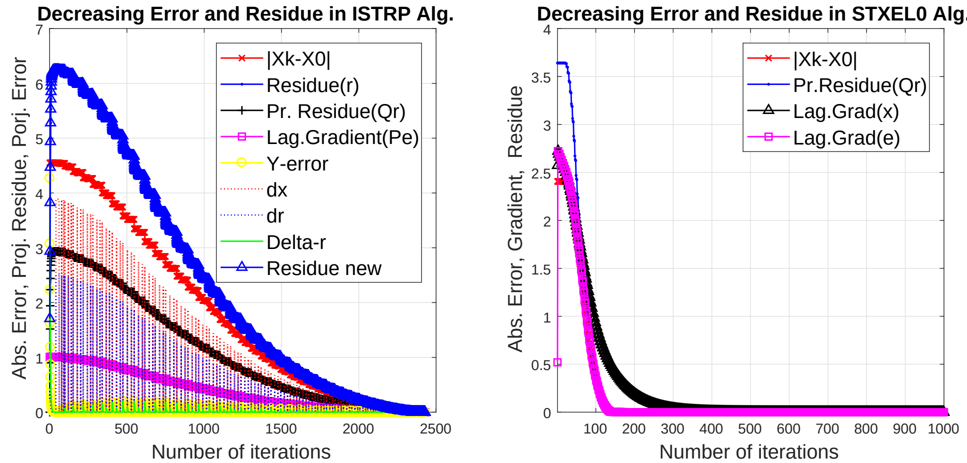
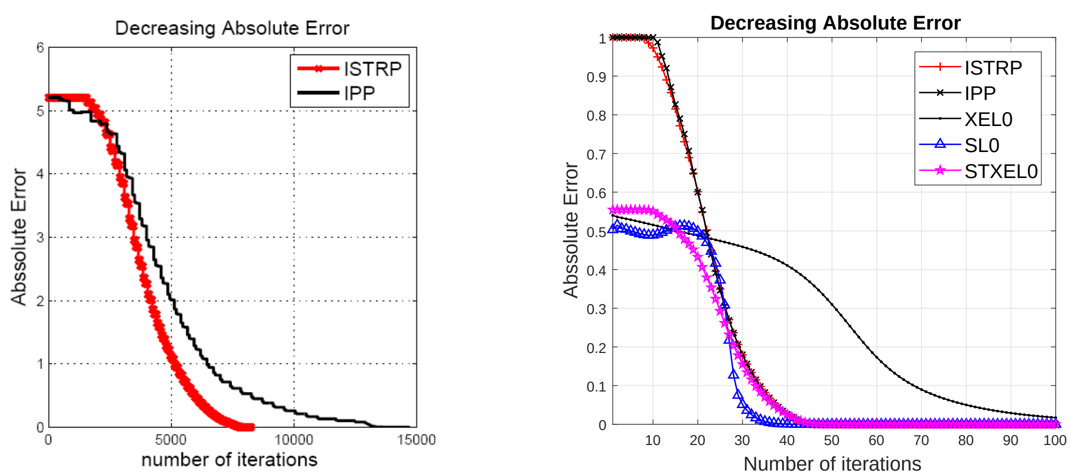


Figure 4-15: The convergence of internal parameters of the two proposed algorithms ISTRP and STXEL0.

Note: All of the internal parameters used inside the algorithms show decreasing trend with iteration, indicating convergence of the solution. Compared to the proposed ℓ_1 *based method ISTRP, the* ℓ_0 *based method converges* $4\times$ *times faster.*



(a) (b)

Figure 4-16: The convergence of reconstruction error of the two proposed algorithms ISTRP and STXEL0.

Note: (a) Decreasing absolute error in reconstruction with iteration. The ℓ_1 *based thresholding algorithm IPPSCAD and ISTRP follows same profile. (b) Decreasing relative error of the proposed algorithm with thresholding (STXEL0) and without thresholding (XEL0).*

4.4 Chapter Summary

The ℓ_0 minimization based sparse signal recovery method STXEL0 presented in the previous chapter is reformulated into a cascaded computational network, to enable the implementation on low profile computing platforms for real time use. Here the gradient minimization functions are defined for various values of algorithm parameter and stored in RAM to reduce processing load. The computational complexity of the hardware implementation is described in terms of the basic MAC units. The computational precision limits of the algorithms is described and evaluated. The optimal value of the algorithm regularization parameters are determined experimentally. The magnitude change in the internal variables of the algorithms are studied to confirm the convergence. Extensive experimental evaluation of the algorithms are done and the results are compared with the seven different classes of existing methods. The advantage of the proposed methods are presented. The algorithm is tested with various signal reconstruction scenarios.

The analysis shows that the ℓ_0 minimization based STXEL0 algorithm gives better SNR in the reconstruction of images with lesser processing time. Generally ℓ_0 methods are avoided during problem definition, since an analytical solution is difficult to arrive. The ℓ_0 problem is approximated with a polynomial and the solution is obtained more efficiently. In short the methods described here enable the users to experimentally determine optimal inverse matrix for the specific sparse recovery problem and estimate the computational load required for the implementation. The theoretical analysis of convergence guarantee for any arbitrary inverse matrix is not discussed. However, if arbitrary matrix is selected as the inverse, the convergence is influenced by the value of regularization parameter. The proposed method enables implementation of the sparse recovery algorithm with basic MAC units and function tables. However, the proposed algorithms are optimized for low profile computing devices, these were simulated and tested using MATLAB on 64-bit MS Windows OS in Intel i3 dual core 1.9 GHz processor with 12 GB RAM. The evaluation of a low profile computing platform for implementation of this algorithm is presented in the next chapter.

Chapter 5

Evaluation of AM3358 Board for Networked Sparse Signal Acquisition

5.1 Introduction

The single board computers are extensively used in rapid prototyping and product development [56], [57], [58], [59]. This chapter presents the result and analysis of a feasibility study on Beaglebone black single board computer (beagle board) for realtime data acquisition and commanding application. This board has $TIAM3358$ ARM processor and $LAN8710A$ IEEE802.3 10/100Base-T/TX transceiver with a reduced media independent interface. The TCP/IP protocol stack is implemented on the Debian Linux kernel and the MAC protocol is implemented on a dedicated eMAC unit. The sections of this chapter is presented as follows: the section 2 describes the platform constraints of the board [60]. The section 3 gives the analysis and feasibility of using the board as a realtime data acquisition system. [61], [62]. The summary of analysis and scope of improvement are given in the chapter summary.

5.2 Platform Constraints

IoT platforms are used in data acquisition and control applications [63], [64]. The performance evaluation of Ethernet protocol for realtime application is given in [65]. The processor and protocol stack of this board supports 1000Base-T (1Gb Ethernet)

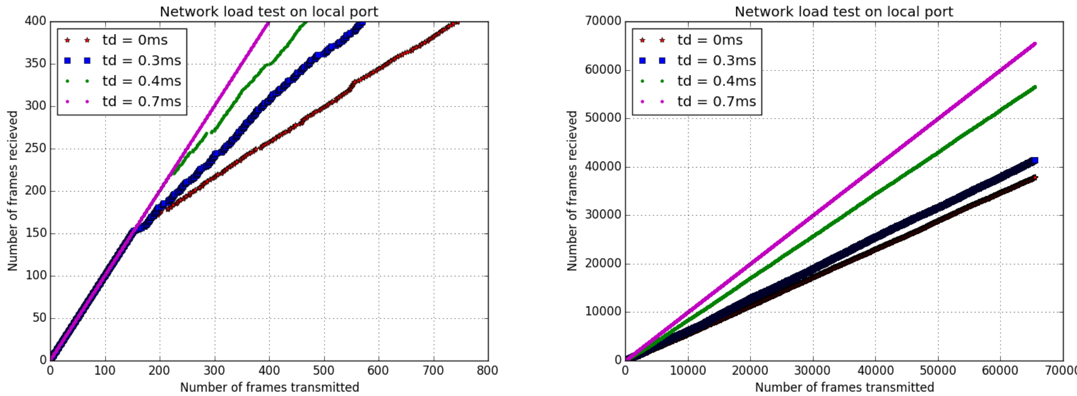


Figure 5-1: The plot of data loss after 150 frames when inter message gap is < 0.7 ms.

with 2016 byte frames size and allows the servicing of core interrupts by the Cortex A8 or programmable realtime unit [66]. However, the network interface hardware in the board restricts the speed to 10Base-T or 100Base-Tx. In both cases the bit duration is 8 ns corresponding to 125 $Mbps$ signaling rate and the $4/5B$ bit encoding of 100BaseTx reduces the effective bit rate to 100 $Mbps$. The 5 level voltage signaling (1.0, 0.5, 0.0, -0.5, -1.0 v) of 1000BaseT for representing the logic values 00, 01, 10, 11 respectively excluding 0v gives signaling rate of 250 $Mbps$ and its $8/9B$ bit encoding reduces the effective bit rate to 222 $Mbps$. The 4 pair half duplex mode can achieve 888 $Mbps$.

The UDP evaluation program for sending and receiving 1k frames from a given port is written in C and compiled using native gcc. Two sockets are created for transmission and reception. The network communication is implemented using $sendto$ and $recvfrom$ functions. To verify that all the transmitted frames are received without error and in sequence, the frames are designated with source ID, destination ID and frame sequence number at byte locations 4, 6 and 16 respectively in the Tx frame buffer; 65536 such frames are transmitted in a burst. Checking the command and command complement words at location 12 and 14 of the buffer confirms the validity of the frame In the receiver board the frame number and sequence in which it is received are recorded. The process is repeated 100 times to get an average estimate.

The data frames are transmitted from the board in 0, 200, 500, 700 μs time gaps

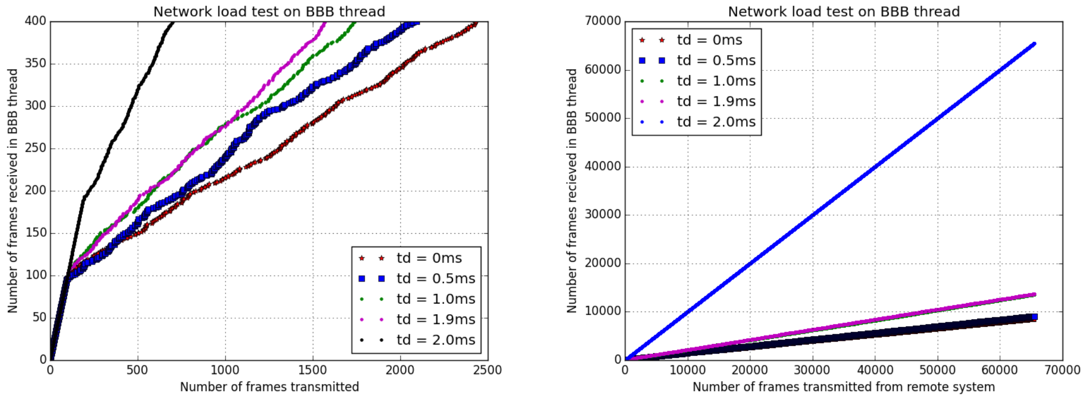
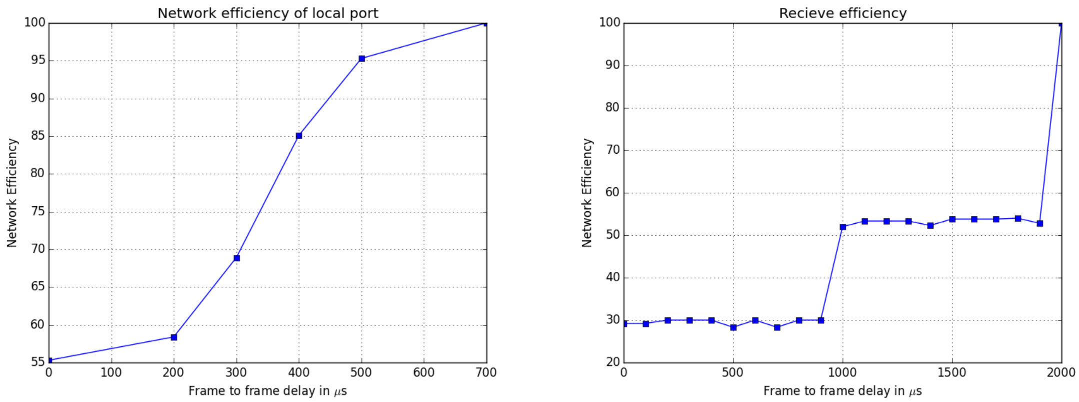


Figure 5-2: The data loss between the boards in Rx and Tx mode when inter message gap is $< 2.0\ ms$.

and the frames are received using the same board in loop-back mode. The plot of frame number versus receive sequence count is shown in figure 5-1. This gives a measure of number of frames transmitted and received by the same board for various inter messages gaps. After the reception of 150 frames the buffer over flow occurs and the receiver misses the frames. However, if the inter message gap $> 0.7\ ms$ no data loss is observed. The buffer overflow happens due to the limited buffer size (200k) of the board [67]. From this analysis it is found that 100% loss less communication is achieved when the inter message gap is 700 μs and hence the useful bandwidth is 11.7028 $Mbps$ $(1024 \times 8/700\ \mu s)$ even if the network interface operating in 100Base-Tx mode.

The communication integrity between two similar boards is shown in figure 5-2. In this case if the inter message gap is $> 2\ ms$ no data loss is observed. The maximum transmit bandwidth observed is 4.096 $Mbps$ $(1024 \times 8/2000\ \mu s)$. The test is repeated with direct interface and interface using network switch; in both the cases the maximum receive bandwidth available is same. The percentage of date received by the board in loop back mode and remote data reception mode through network switch for various inter message gap between 0 to 2000 μs is shown in figure 5-3(a) and figure 5-3(b).



(a) (a)

Figure 5-3: The change in communication efficiency when inter message gap is increased from 1 *us* to 2 *ms*.

Note: (a) The percentage of date received in loop back mode. (b) The percentage of date received in remote reception mode.

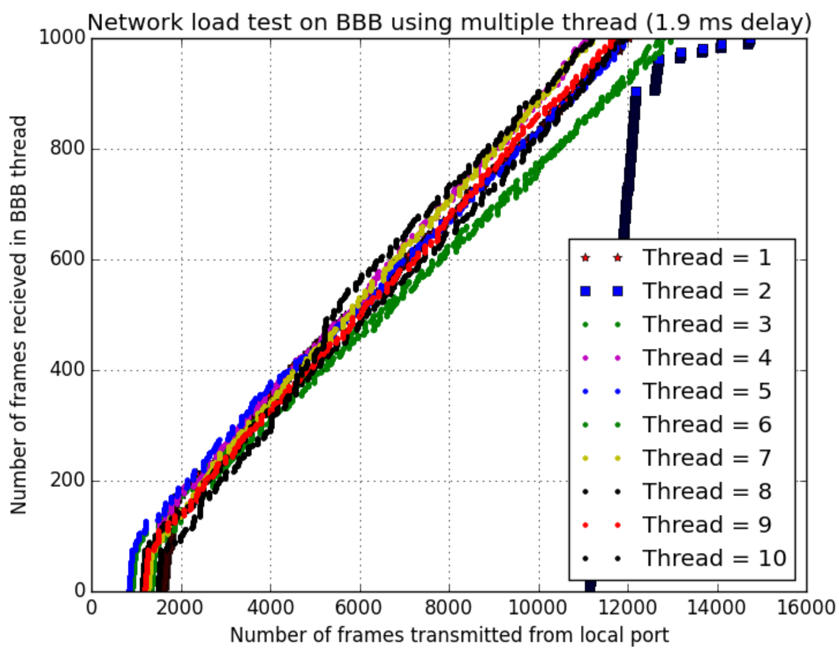


Figure 5-4: The data loss plot when the board receive the data through multiple software threads.

Note: Multiple threads usage: the number of frames received by each thread is reduces approximately by the same factor as number of threads

5.2.1 Effect of Software Threads on Throughput

The board is configured to receive data through dedicated threads. No improvement in communication integrity is observed when inter message gap is < 2.0 ms as shown in Figure 5-4. The experiment is repeated with 10 data sockets ports and each socket is programmed to receive the data using individual threads. The inter message gap is varied between 0 ms to 2.0 ms. It is observed that irrespective of delay or the number of threads the communication bandwidth is limited by the capability of interface hardware. When inter message gap is 1.9 ms only 10% of the total frames are received by each thread and total number of frame received remains same ($10\% \times 10 = 100\%$). Hence no improvement can be achieved using multiple threads. It is observed that only one thread is receiving the complete data frame in sequence. The communication bandwidth is not related to the processing power of the PRU, but limited by usable communication bandwidth of the interface hardware.

5.2.2 Effect of Network Switch on Throughput

The network is configured with 32 units of the beagle board and an industrial computer capable of network speed negotiation up to 1Gbps. The communication protocol is developed over UDP/IP with no retransmission even if acknowledgment frame is not received. The data frame is buffered using 200kb circular buffer inside the network interface chip. This buffer is emptied after PRU has completed the frame processing. When the PRU is not performing the processing at the rate at which the data is received the buffer gets overwritten by the new data received and eventually results in the data frames loss. Also, the network traffic is monitored using wireshark application and found that all data frames are send to the board. The data log of the network switch also confirms this. The loss of data is more when the two boards are connected directly, since the virtual buffer created by the network switch is absent in that case.

5.2.3 Bandwidth Limitations of the Board

To evaluate the realtime performance a process control network with 32 commanding and data acquisition modules is considered. The commands to the modules are send through Ethernet. The modules acquire the data and transmit back to control console. As the available bandwidth is 4.096 $Mbps$ the frame rate possible is 500 frames per second ($4.096M/8$). This corresponds to 2 ms command periodicity when frame size is 1024 byte long. Commanding all 32 boards using 64 bytes command and 256 bytes data acknowledgment frame require 10240 bytes per cycle. The communication format is shown in Figure 5-5. In bus topology network with no active switching system this communication load results in 50 frames per second or 20 ms periodicity. This gives a realizable communication bandwidth of $50Hz$ and control bandwidth of 10Hz. Considering the transmission bandwidth of 4.5223 $Mbps$, the achievable channel sampling rate is $4.41K$ samples per second if there are 64 simultaneous sampling ADC channels with 16 bit resolution. In general the achievable networked control bandwidth can be written as (5.1), where BW_{net} is the network bandwidth, L_{cmd} is command frame length, L_{dat} is the data frame length and N is total number of data acquisition and control modules.

$$BW_{ctrl} = \frac{BW_{net}}{8(L_{cmd} + L_{dat})N} \tag{5.1}$$

Figure 5-5: The communication frame format of the networked data acquisition and control modules

Note: Frame format of the networked control modules with 32 units. First 64 bytes corresponds to command frame followed by 256 bytes acknowledgment frame contains status and acquired data.

5.3 Timing Analysis

From the experimental evaluation it is estimated that the board has 4.5223 $Mbps$ transmission and 4.096 $Mbps$ reception bandwidth. The reduction in the reception bandwidth is due to the processing overhead involved in frame validation process. The programmable boards with Ethernet interface are used in networked data acquisition and control applications [68]. The feasibility of implementing the realtime data acquisition and commanding system using beagle board is discussed here. The analysis of timing requirements for the networked control system is described in [69],[70] and the time delay compensation scheme is given in [71]. The configuration of data acquisition network used in the evaluation is shown in Figure 5-6. The functional requirement specifications are summarized as :

- Receive system commands from the supervisor module and configure the data acquisition system for the specified sampling rate.
- Periodically acquire and store the parameters in local memory.
- Send the status message with in 12 μs of command reception.
- Performs high-speed sampling of the requested analog channel.
- Acquire parameters of maximum 30 measurement channels within 20 ms.

However, the data acquisition should complete before 20 ms, to yield time to process the system commands. The maximum number of words involved in the system commanding is 4 words plus network frame header. While maintaining the status response latency of 12 μs, the maximum time required to complete a system command is $4 \times 20\ \mu s + 12\ \mu s$. The net time required to command all the 30 sub-systems is 2.760 ms and hence there is a time margin of 17.24 ms. After providing a time margin of 1 ms the remaining 16.24 ms can be equally divided for data acquisition from 30 sub-systems. This gives 541 μs for data acquisition from each sub-system. Within this time maximum 27 words can be transmitted. But, considering the data

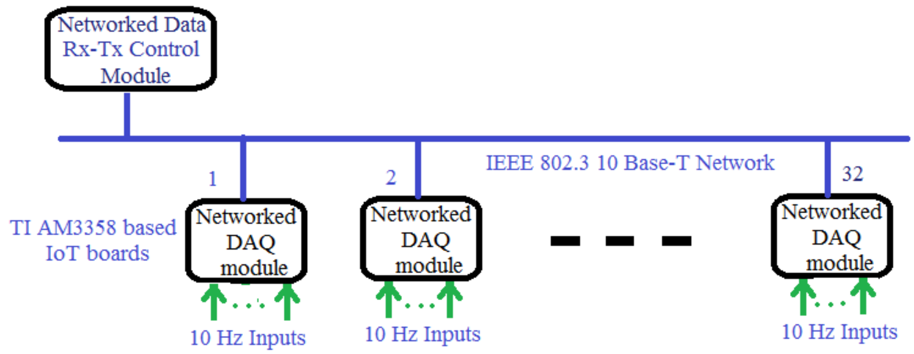


Figure 5-6: The configuration of the data acquisition and control network.

request command, status response and 12 μs response latency the maximum 24 word communication is possible. The 24 data word limit is sufficient for sending the system related information back to control console. Considering all this timing constrains discussed, it is feasible to complete the commanding and data acquisition within one cycle period of 20 ms.

Algorithm 9 Networked Data Acquisition and Commanding

Require: T_{cycle}, ID, $[CMD, SID]$

Task: receive and process control command
Initialization: $nodeID$ = ID
while SID == Ctrl Console **do**
  $receive$: [Network data]
  $verify$: DID == $nodeID$
  $validate$: CMD xor $C\bar{M}D$ == 0xFFFF
  $send$: Status to Ctrl Console
  get : command angle θ
  $send$: θ to DSP
end while
while $time > T_{cycle}$ **do**
  $format$: data response
  $send$: Data to Ctrl Console
  $wait$: T_{cycle}
  if link fail **then**
    Do $configure$: salvage mode
  end if
end while
Output: command angle θ

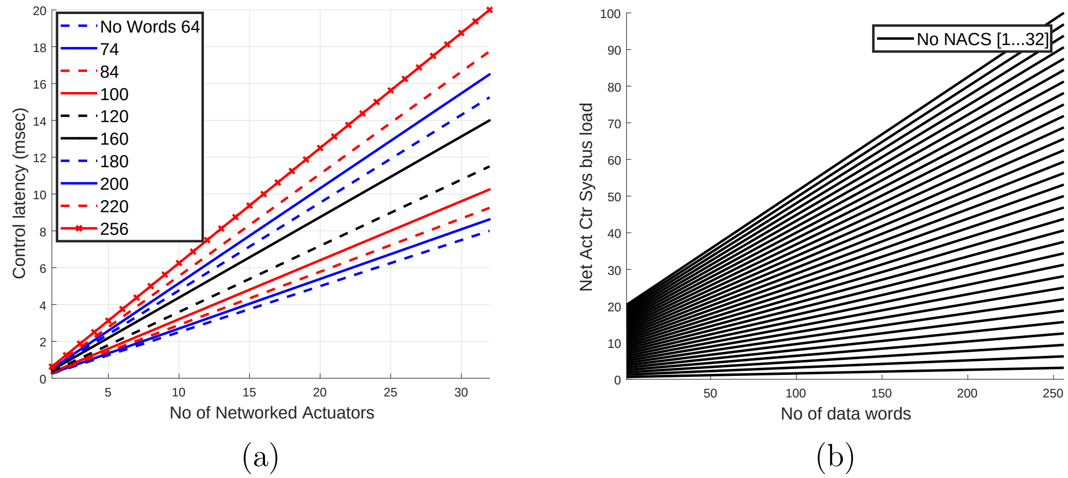


Figure 5-7: The relation between control latency vs. number of nodes, network bus load vs. data frame size.

Note: (a) The number of sub-systems that can be controlled using the node vs feasible control latency of the node. (b) The number of data words that can be handled in one frame vs bus load on the network.

Based on the timing analysis discussed, a protocol for commanding the beagle board is given in algorithm-9. The cycle timing analysis of realtime protocol over Ethernet is given in [72]. The command to the board consists of command word and 2 data words; the first 5 bits of the data word are the controller address repeating the same address as in the first command word. The remaining 11 bits specify the sub-system control word. The second word is the complement of the first. Additional 26 bytes network header and 32 extra bytes are included to make the frame 64 bytes long. In response to the command frame the board sends status word response within 12 μs and executes the actuation. The status frame consists of 26 bytes frame header, 24 words status data, 128 bytes analog data corresponding to 64 ADC channels and 54 extra bytes to make the frame 256 bytes long.

5.3.1 Communication Bus Load

In command overload situation the supervisor system send commands to all the 30 subsystems in 20 ms interval, this limiting case is considered to estimate the maximum load scenario. After issuing command the control console waits for the status data

Table 5.1: **Realtime latency and number of control nodes**

Control latency	No Words in Data Msg	Max No of nodes
2 ms	64	7
4 ms	74	15
8 ms	100	25
16 ms	200	31

from the node. The beagle board responds within 12 μs by transmitting command word followed by 64 data words containing information about the sub-system. This total communication takes $30 \times 8 \times 320/4.096\ MHz = 18.75\ ms$ leaving 1.25 ms margin. Based on this worst-case timing the bus is free for $(100 \times 1.25/20) = 6.26\%$ of the cycle interval. The communication load can be determined using equation (5.2) where BW_{net} is the network bandwidth in bits per second unit, T_p is minor cycle period and other parameters are as described earlier.

$$\text{bus load} = \frac{8(L_{cmd} + L_{dat})N}{BW_{net} \times T_p\ ms} 100 \tag{5.2}$$

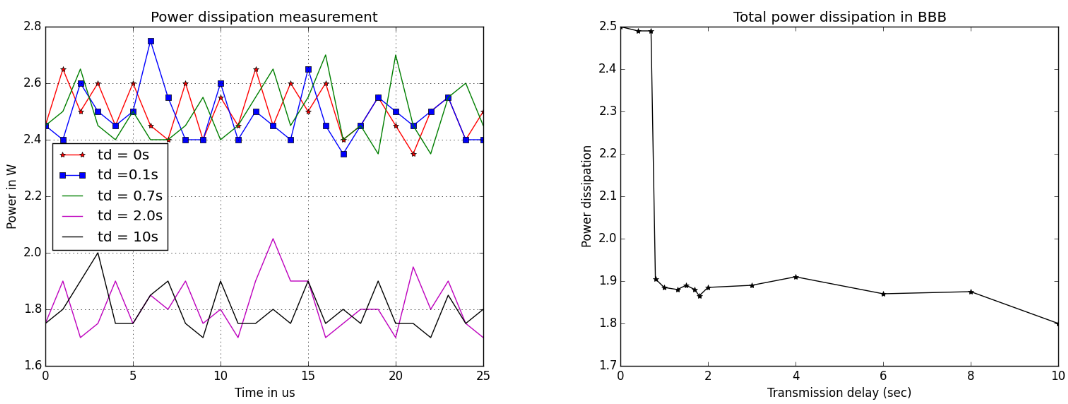
The response latency of the network for various frame size and number of nodes is shown graphically in Figure 5-7(a) and is summarized in Table 5.1. The change in bus load when number of nodes are increased from 1 to 32 and the number of data words are increased from 2 to 256 is given in Figure 5-7(b) and summarized in Table 5.2. Under any given constraints the number of nodes and the number of data words are limited. However, if the control latency is $> 20\ ms$, 32 control nodes with 256 byte data words and 64 byte command can be used in the network. If the latency is limited to 10 ms there will be corresponding reduction in communication bandwidth.

5.3.2 Computational Complexity and Power Dissipation

The computational complexity of the networked data acquisition and control algorithm is determined from the types of instructions used in the algorithm implementation.

Table 5.2: **Message length and bus load**

No. Act	No. Rx CMD	No.Tx Data	Net time	BW load
30	64	64	7.85 ms	38.0%
30	64	100	9.84 ms	48.0%
30	64	200	15.74 ms	82.2%
30	64	256	19.24 ms	94.2%



(a) (b)

Figure 5-8: The variation in average power dissipation versus inter message gap during network data acquisition.

Note: (a) Transient variation in power dissipation per node during network data acquisition. (b) Average power dissipation versus inter message gap.

The processor instructions used in the implementation of the algorithm are classified into 4 types; register instructions, ALU instructions, branch instruction and float instructions. The word comparison is implemented using 2 MOV, 1 CMP and 1 JMP instructions. The data communication uses the required number of MOV instructions. The computational complexity in the implementation of the algorithm can be written as $(L_{cmd} + L_{dat} + 8)MOV + 3ALU + 3JMP + FLOP$. Assuming all integer operations takes the same processor load and the floating-point operation uses 5 INT operations, the computational complexity can be written as (5.3).

$$\text{Number of instructions } = L_{cmd} + L_{dat} + 19 \; ops \tag{5.3}$$

This empirical equation includes the commanding and communication process only and the complexity of the DSP based controller algorithm is not discussed. The operating current of the board is measured during transmission and reception. The transient variation in power dissipation when the board is receiving 1024 byte network frame is given in Figure 5-8(a). The average power dissipation when the inter message gap is varied between 0s to 10s is given in Figure 5-8(b). At maximum transmission load the peak average power dissipation measured is $2.5 watts$ and at steady state the power dissipation measured is $1.8 watts$.

5.4 Chapter Summary

In summary the AM3358 processor based beagle board can be used for networked realtime applications with the following timing constraints: (i) Realtime system response tolerance is $> 20\ ms$. (ii) The data receive speed is limited to 4.096 $Mbps$. (iii) The individual network command periodicity is $> 2\ ms$ for command frame of size 1024 bytes long. (iv) The data acquisition and transmission bandwidth is limited to 4.5223 $Mbps$. The realtime network performance can be achieved under the above constraints as the AM3358 processor of the board has predictable performance. The dedicated use of threads for data reception process does not improve network bandwidth, And if higher communication bandwidth is required, the network interface chip or module can be replaced to $1Gbps$ capable device. In short the beagle board based networked data acquisition system can work in 20 ms realtime periodicity and with $10Hz$ output bandwidth if the number nodes are < 32. In summary the AM3358 based processor board can be used for realtime application with strict timing specifications. An implementation of the distributed sparse data acquisition system is presented in the next chapter, which use the data communication protocol described in this chapter and X-L0 E-L0 algorithm described in the previous chapter for reconstruction of the sparse data.

Chapter 6

Distributed Measurement of Naturally Sparse Events

6.1 Introduction

Events with naturally sparse measurement signature are lightning induced ground potential rise, earthquake or tsunami triggered seismic events, hurricane and flood triggered water saturation and land slides, solar flare induced electric and magnetic field variations. These events are rare and result in extreme inconveniences. However there are large number of widely distributed binary sensor systems for detecting such extreme events, in many case these warning systems are signal threshold based detection and the measurement are detected when it is large and some times the damage starts when the events are occurring. All of these events generates typical precursor signals, however these signals are often missed due the lower acquisition rate, not due to the low sampling rate as extremely fast signal sampling systems are available for measurement and storage of all of signals. The event missing depends on how the sampled data is acquired and this needs no explanation that these rare events are not possible to measure and acquire from the data analysis lab. The sensors need to be widely distributed in event prone remote locations. It is easy to establish high sampling rate system in every location to capture the typical precursor signals, however bringing these signals to the labs for real time analysis is important. The solution to this problem is simple, use high bandwidth communication systems. But it is not cost effective and impractical. An alternative solution is to compress the data and transmit through low bandwidth communication framework. But there exist a method of measuring and compressing at the same time and it is widely studied and used in various applications like synthetic aperture radar signal measurements and MRI. One way to increase the chance of detection is to spawn expendable low cost measurement systems widely and establish a communication network for acquiring these signals.

The Internet of things based measurement system provides the low cost expendable measurement and processing platform. This chapter discusses about the networking that can be implemented using these units for transmitting the acquired sparse signals using the concept of wireless sensor networking systems. There exist advanced systems and algorithms for all the processes that are explored in the chapter, what is being discussed is how to optimize system to a minimum level so that it is implementable using expendable IoT platforms. The contents of this chapter is organized into three sections, the measurement of sparse events, distributed acquisition and reliability. The accurate measurement of the ground potential voltage at distributed locations inside a plant is necessary for correcting the offset in the corresponding ground referenced measurements for better data interpretation; especially in industrial process control applications. This offset is transient and exist only for a short duration $< 200\mu s$ and in most of the time the value is zero. The sampling rate has to be high to capture these transient characteristics in the signal. Considering the large number of such distributed measurements the data generated will also be enormous. The following four problems need to be addressed in this scenario:

- The measurement of transient signal.
- The handling of large amount of data.
- The transmission of the acquired signal to the data processing station
- The implementation of the system.

Considering the implementation of the system, it needs to be low cost and expendable, because, even if high reliable components are used in the fabrication of these devices, the degradation is eminent as long as they are left outside. Hence the low cost IoT based devices are selected in such scenario. But, such devices have processing capabilities limitation that constrains the implementation of data acquisition, data handling and the communication processes. The capability of one such IoT data processing board is discussed in the earlier chapter. The sparse measurement method is adopted here to solve the first two problems as described in section 2. Also the processing requirement is relatively low compared to conventional high speed data acquisition and compression process, as the sparse measurement combines the data acquisition and compression into a simpler process of matrix multiplication. This gives triple advantage of high speed acquisition, data compression and simplified computation. Also, the processing can be done in the limited resources of IoT devices.

Considering the transmission of measured data to data processing station, the use of copper cables is infeasible as the communication cables become electrically polarized during lightning and causes damage to the devices interfaced to it. In such case the measurement system will be the first one to fail. However, wireless sensor network can be used in this scenario as these are independent, easily deployable, network scalable and implementable using IoT devices [73],[74]. A trivial solution is to implement a data collection node with multiple data acquisition and transmission nodes. But this is limited by the communication range of the underlying physical layer. To spread-out the data acquisition units in a wider area a routing mechanism need to be implemented for data acquisition and transmission nodes; and at the same time implementable using IoT devices with limited capabilities. Some of the algorithms available for such routing are Dijikstra's algorithm (DA), ad hoc on demand distance vector routing (AODV) [75], ad hoc on demand multipath distance vector routing (AOMDV) [76], secure multipath load balancing-AODV [77]. The low energy adaptive clustering hierarchy (LEACH) [78] with self-organization and adaptive clustering feature is the base of many of the power aware routing algorithms. The algorithm is compared with other power aware routing algorithms like Energy efficient clustering (DEEC) [79], Immune cooperative particle swarm optimization (ICPSO) [80], Extended stable election protocol (ESEP) [81] and Energy-efficient, delay-aware and lifetime-balancing data collection protocol (EDAL) [82]. Other energy conservation options in WSN are spectrum sensing and channel allocation [83] and Efficient energy-aware routing with redundancy elimination [84], but in these the route configuration is fixed and not randomly deployable. The implementations of these algorithms demand high computational power as these algorithms are designed for high efficiency routing applications; but, if the implementation platform is a low cost IoT processor the cost of computation is a major factor. A comparison of these algorithms is given in Table 6.1.

A similar work on combining compressed sensing and MAC protocol design is presented in [85]. A custom protocol development scheme for adaptive multipath load balancing scheme based on disjoint links found from path vacant ratio is given in [77]. A node distribution strategy maximizing the coverage is described in [86]. A survey of multipath routing protocols and its classifications is given in the paper [87]. A comparison and evaluation-metric for multipath routing algorithms can be seen in [88] and a survey of Bluetooth multi-hop networks including low energy mesh networks is given in [89]. Power dissipation can be further reduced using radio duty cycling protocol [90] or straight line routing protocol [91]; but these are not considered as they need platform hardware change. The geographical energy aware routing is not considered as this requires GPS based triangulation method for route discovery. Based on various wireless sensor network schemes studied in [92] it is observed that IEEE 802.11 based solutions are suitable in such applications. A general guideline for design and analysis of sensing network can be found in [93] and a frame work for MAC protocol modeling is given in [94]. The reliability of the network is estimated using accelerated testing concept described in [95]. An empirical expression for reliability estimate using Eyring model is described in [96]. The quantitative estimation of the reliability of the proposed network is given in section 3 and 7 of this chapter. The chapter summary is given in section 4.

Table 6.1: **Parameter usage, cost function and complexity of routing algorithms**

Routing	d	T	L	E_r	P	W	arch.	route.	complexity	cost function
AOMDV (2001) [76]	Y	Y	Y	-	-	-	tree	dist. vec.	$\mathcal{O}(rn \log n)$	$\min \sum d$
LEACH (2002) [78]	Y	-	-	Y	-	-	cluster	acc	$\mathcal{O}(n \log n)$	$\min \sum -E_r$
DEEC (2006) [79]	Y	-	-	Y	-	Y	cluster	acc	$\mathcal{O}(n \log n)$	$\min \sum d - E_r$
ICPSO (2011) [80]	Y	Y	-	Y	-	-	cluster	PSO	$\mathcal{O}(rn \log n)$	$\max \sum E_r / \{f_1(E_r) + f_1(d) + f_1(T_{pd})\}$
ESEP (2012) [81]	Y	Y	-	Y	-	Y	cluster	acc	$\mathcal{O}(n^2 \log n)$	$\min \sum d + F - E_r$
CAA (2014) [93]	Y	Y	Y	-	Y	Y	tree	acc	$\mathcal{O}(r\ n^2)$	$\max \mathcal{T}(\tau_{ij}, \bar{W}\Omega^k)$
SM-AODV (2014) [77]	Y	Y	-	Y	-	Y	ODV	QoS	$\mathcal{O}(n \times n)$	$\max PDrate$
C EDAL (2015) [82]	Y	Y	Y	Y	Y	Y	T graph	heuristic	$\mathcal{O}(m^2 n \log n)$	$\min \sum_k \sum_i \sum_j c_{ij} x_{ijk}$
D EDAL (2015)[82]	Y	Y	Y	Y	Y	Y	ACop	geographic	$\mathcal{O}(n^2 \log n^2)$	$\sum c_j + (n-k) \sum_i^k \frac{W_{ij}}{e_i}$
RCAMP (*proposed*)[97]	Y	Y	Y	Y	Y	Y	cluster	sync	$\mathcal{O}(rn \log n)$	$\max(e^{f_1(d)} + e^{f_2(E_r)} + e^{f_1(P)})$

Notes: d-hop distance, T-propagation delay, L-link speed, E_r-residual energy, P-Tx power or bandwidth, W-link weight. n-number of nodes, r-number of routes, $f()$-some function, c_{ij}, x_{ijk} - algorithm specific parameters, Ω^k-cluster set. dv-distance vector, acc-accumulation, heur-heuristic, Tgrph-Topology graph, PD-packet delivery, $\mathcal{T}$-throughput. ACop-Ant colony optimisation, PSO-particle swam optimisation.

6.2 Ground Potential Measurement

The lighting flash from 1 to 10 km altitude cumulonimbus clouds span for $50-200ms$ with a discharge peak near $10\mu s$. The ground potential rise due to lightning current leakage to ground is expressed as $V_r = \frac{I\rho}{2\pi r}$, where ρ is the local surface resistivity of earth in Ωm, I is the leakage current, r is the radial distance from the lighting strike point [98]. The figure 6-1(a) shows the peak time value of the simulated potential during electrical discharge. The lightning induced ground potential is simulated using matlab model[1] and the transients are acquired using simulink model. The simulink schematic is similar to one used in [99] for wide band signal acquisition for spectral estimation. The transient nature of this signal can be observed from figure 6-2(a). In ground potential measurement the reference ground is infeasible hence two electrical probes at a distance apart measure the relative potential difference. The probes located in radial direction measure large potential difference compared to electrodes along the tangent to the equi-potential field as shown in the figure 6-1(b). The Schlumberger resistance technique is used to calibrate the measurement device. The differential potential developed can be written as (6.1) where d is the spacing between electrodes.

$$V_d = \frac{I\rho}{\pi r}\left(\frac{1}{1+r/d}\right) \tag{6.1}$$

As this transient voltage exist only for a short duration $< 200\mu s$; based on sampling theorem this signal should be acquired least at $5KHz$. Due to sparse nature of the event most of the time the measurement is zero. However to get the transient characteristics the sampling rate cannot be compromised. Considering the large number of such distributed measurement units the data generated will be enormous. 1000 such measurement nodes will generate $5000K$ samples per second, that needs a bandwidth of $\geq 50Mbps$ at data collection node. This data requirement is more than the specifications of IoT devices. As discussed earlier the compressed sensing technique can combine the data acquisition and compression into a simpler process of matrix mul-

[1] data taken form soil science society of America

tiplication and can be done in any low profile computational units, hence the sparse measurement method is adopted here to reduce the sampling rate.

6.2.1 Sparse Measurement of Ground Potential

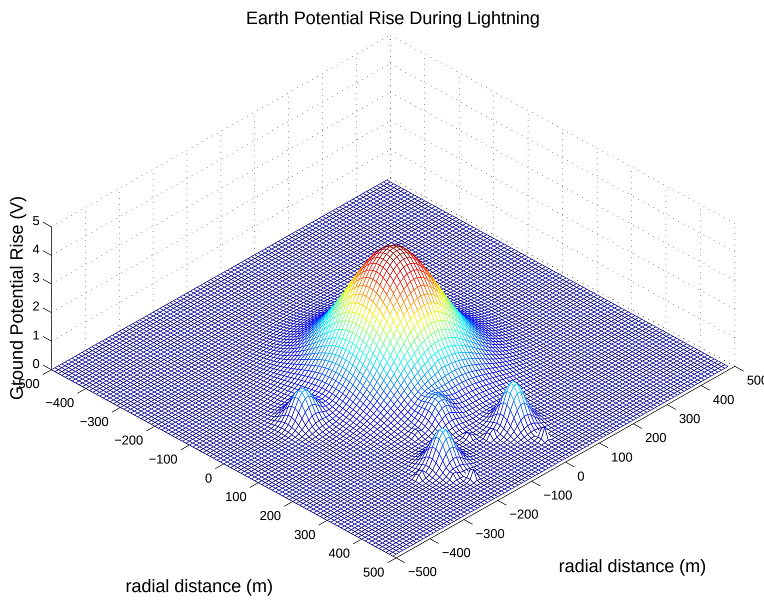
Compressed Sensing is a signal measurement and compression technique for sparse signals, where the analog to digital conversion can be done at sufficiently smaller sampling frequency compared to the Nyquist sampling rate [1]. For any arbitrary signal $\mathbf{x} \in \mathbb{R}^N$ with K-sparse representation $\mathbf{z} \in \mathbb{R}^N$ ($\|\mathbf{z}\|_0 = K$) in some basis $\mathbf{B}$ and with $\mathbf{x} \to \mathbf{z}$ transformation given by $\mathbf{x} = \mathbf{B}\mathbf{z}, (\mathbf{B} \in \mathbb{R}^{N\times N})$ the theory states that the signal $\mathbf{x}$ can be measured as $\mathbf{y}$ with $M \ll N$ samples from a linear sparse-projection space $\mathbf{B}$ using a measurement matrix $\mathbf{A}$ (6.2).

$$\mathbf{y} = \mathbf{A}\mathbf{B}^{-1}\mathbf{x}, \quad \mathbf{A} \in \mathbb{R}^{M\times N}, \|\mathbf{z}\|_0 = K < M \ll N \tag{6.2}$$

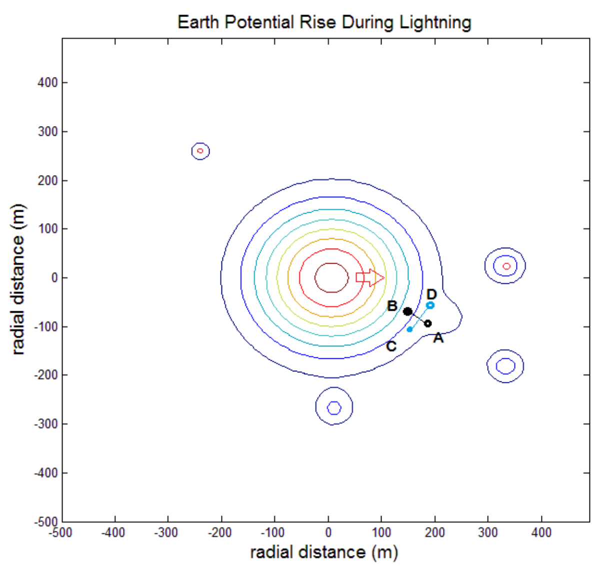
where $\mathbf{B}^{-1} \in \mathbb{R}^{N\times N}$ is the transformation matrix to convert $\mathbf{x}$ to sparse $\mathbf{z}$. If the signal to be acquired $\mathbf{x}$ is sparse, $\|\mathbf{x}\|_0 = K \ll N$ then $\mathbf{B} = \mathbf{I}$. As the transient signal acquired is in compressed form, it needs to be reverted back to its original form prior to use. The voltage profile is reconstructed from the measurement using ℓ_0 minimization given in (6.3) with small error $\varepsilon \approx 0$.

$$\mathbf{x}^* = \mathbf{B}\mathbf{z}^*,\ \mathbf{z}^* = \arg\min_{\mathbf{z}} \|\mathbf{z}\|_0\,,\ \text{s.t.}\ \|\mathbf{y} - \mathbf{A}\mathbf{z}\|_2 \le \varepsilon \tag{6.3}$$

This optimization involves large computation, however this processing is not meant to be done in IoT processor board. The signal reconstruction and analysis is done in a remote supervisory console. The ℓ_0 minimization is nondeterministic in polynomial time (NP) hard problem in terms of function computation because of the combinatorial search required, it is very large even for smaller vectors. However, modified ℓ_0 function approximation methods like Segmented Threshold X-L0 E-L0 (STXEL0) [100], the algorithm presented in Chapter 4 or radial basis function based sparse recovery [25] does some alternate ways to minimize the computation require-



(a)



(b)

Figure 6-1: The profile of simulated ground potential rise and measurement using differential probes.

Note: (a) Simulated ground potential using $3.2Amp$ $10ms$ *DC discharge pulse causes proximately* $5V$ *ground potential rise along* $5m$ *radius in wet Clayey sand with* $50\Omega m$ *resistivity. (b) The contour of ground potential rises. The voltage sensed by the isolated differential probes [A,B] in the radial direction to the field is high compared to the measurement by the probes [C,D] in the tangential direction to the field.*

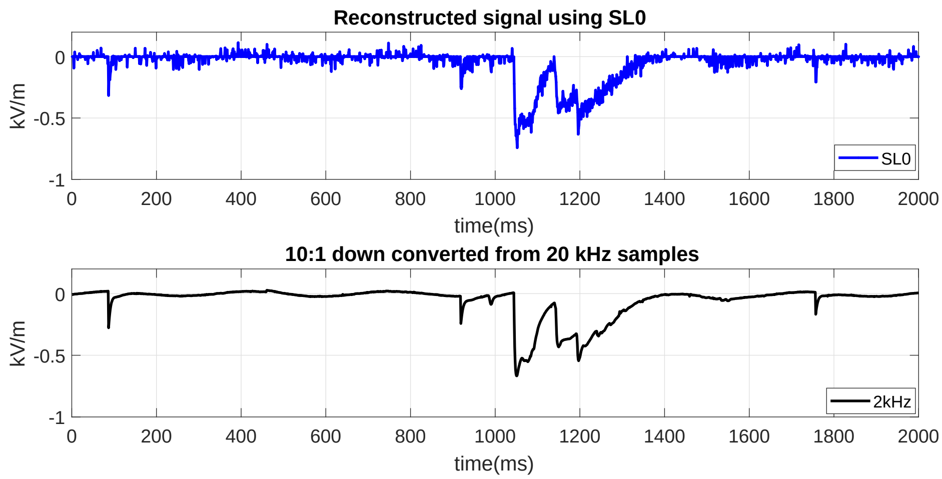


Figure 6-2: Sparse measurement and reconstruction of ground potential signal

Note: Comparison of sparse ground potential variation signal reconstructed from compressed samples using SL0 algorithm and the corresponding 10:1 down converted samples from 20 KHz base band measurement

ment. The computation in STXEL0 algorithm without segmentation is recapitulated as minimize $\|\mathbf{z}\|_0$ and $\|\mathbf{e}\|_0$ by iteration using polynomial approximation subject to the recovery error $\mathbf{e} = \mathbf{A}^{\dagger}(\mathbf{y} - \mathbf{Az}) < \varepsilon$ and scale down the function gradient in every iteration with smaller scale factor $\alpha_k = \alpha_0 \sigma_k$. Here $\mathbf{z}$ is the sparse representation of the compressible signal $\mathbf{x}$. The initial value used is $\mathbf{z}(0) = \mathbf{A}^{\dagger}\mathbf{y}$. The figure 6-2 shows the voltage transients recovered using smooth ℓ_0 [26] minimization method.

6.3 Wireless Sensor Network

Having acquired the data it needs to be transmitted to the monitoring station. The network considered here consists of 3 types of nodes, the nodes with direct interface to data processing station (N0 nodes), the nodes with data routing and data acquisition functions (APQ nodes) and the nodes with data acquisition and transmission functions (AQ nodes). The strength of the network depends on the effectiveness of the logic build into its routing algorithm. Routing algorithms in general use clustering or forwarding schemes, clustering is computationally demanding and is suitable for high speed low data frame size networks. The frame forwarding scheme is optimal

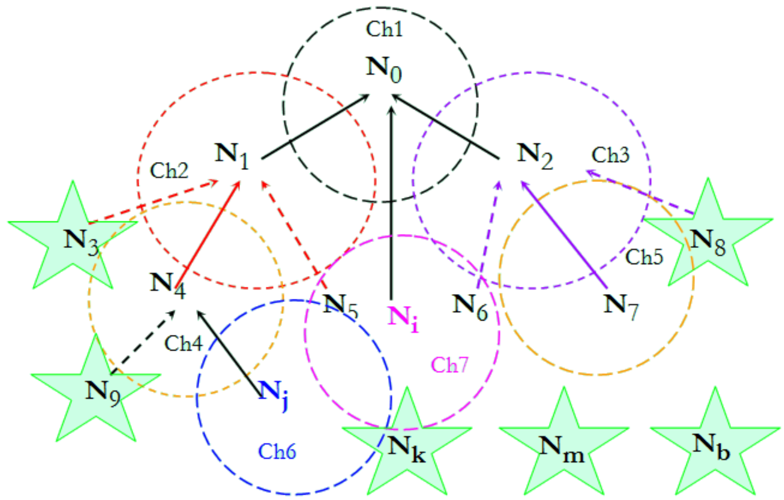


Figure 6-3: The routing architecture of the distributed sparse measurement system

Note: N_k represents the node which is trying to establish communication with one of its neighbourhood $\{N_i, i \in \Omega^k\}$. The link weight is computed as function of $d_{GI}(i)$, $P_S(i)$, $F(i)$, $L(i)$ and $b(i)$.

for low data rate networks. A simpler protocol is needed for low profile IoT boards. The table I shows the computational complexity of some of the currently available algorithms. Here we try to develop a light-weight routing algorithm. Four factors are considered as most essential; namely energy efficiency, data communication reliability, data security and its maintainability. While considering reliability as primary concern the multipath routing is optimal since it reduces the system unavailability due to node failures. Considering the data synchronization issues of multipath routing, the stand-by redundant multipath routing technique is effective. Here every node maintains a list of priority routes selected based on link cost estimation. These links will be used in later time if a failure is observed in the active link. The parameters used in the algorithm design are: number of nodes in the upstream path up to N0 node d_{GI}, link speed L, bandwidth BW, battery energy level b / residual energy E_r, received signal power P_S and the data frame size F. The table I shows the parameter usage of some of the currently available algorithms.

The proposed routing strategy considers the above mentioned resources as constrained while determining the optimal route [101]. The network graph referred here is shown in figure 6-3. The algorithm first identifies its neighbouring access point nodes (APQ) or N0 node which satisfies the network discovery process described as

$$\Omega^k = \{i \dots\} \quad | \quad Pwd = f(SSID(i)), \quad \forall i \in \mathbb{N}_k \tag{6.4}$$

where $\mathbb{N}_k$ the enumeration index of all possible neighbouring APQs, Ω^k is the set of nodes such that the password Pwd can be decoded from the service set identifier (SSID). To identify all nodes as part of this WSN, the SSID is given the format $[SYS]_3.[YYMM]_4.[SEN]_3.[NUM]_6$ where each field is of the specified bytes long and SYS: system identifier, SEN: sensor type, YY SS: year and month of installation and NUM: sensor number, like [MET1611DFV000822]. After boot up the nodes scan all the WiFi channels for the compliance of the connection format and determines neighbouring APQs [102]. The node N_k then generates the parameter matrix $\mathbf{W}^k$ from the information available in the beacon frame of the observable adjacent nodes $i \dots j \in \Omega^k$.

$$\mathbf{W}^k = \begin{bmatrix} d_{GI}(i) & F(i) & P_S(i) & L(i) & b(i) & Ch(i) \\ . & . & . & . & . & . \\ d_{GI}(j) & F(j) & P_S(j) & L(j) & b(j) & Ch(j) \end{bmatrix}^T \subset \mathbb{R}^{6\times\kappa} \quad i \in \Omega^k, \quad \kappa = |\Omega^k|_0 \tag{6.5}$$

where (i) indicates the node N_i, $d_{GI}(i) \in (1, N_{max})$ the hop distance, N_{max} is the maximum links to reach the data processing node, $P_S(i)$ is the observed signal strength of the node N_i, $P_{min} \leq P_S(i) \leq P_{max} \leq 0$, P_{min}, P_{max} are the minimum and maximum transmission power and $0 < b(i) \leq 1$ is the energy reserve. The operator $|.|_0$ finds the number elements in the set and κ is the total number of neighbourhood nodes of N_k. These parameters are used in the weight computation as exponential functions for accommodating wider range. The upper limit of bandwidth $L(i)$ is accommodated using saturating function $(1 - e^{-\sigma L(i)})$. The frame size $F(i)$ include the node's own data frame size and frame size of other nodes for whichN_i functions as router. A trivial routing from node N_k to N_i ($\mathcal{R}(k)$) is to define a vector weight function $w(\mathbb{R}^{6\times\kappa} \rightarrow \mathbb{R}^{\kappa})$

and select a node with largest link weight (6.7).

$$i = \mathcal{R}(k) : \forall k \ \exists i \in \Omega^k \ \ni \ i = \max_i \{w(\mathbf{W}^k)\} \tag{6.6}$$

$$w(\mathbf{W}_i^k) = \alpha e^{-\lambda d_{GI}(i)} + \beta e^{\gamma P_S(i)} + \mu \left(1 - e^{-\sigma L(i)}\right) + \delta e^{-\epsilon F(i)} + \zeta b(i) e^{\xi b(i)} \tag{6.7}$$

where $(\alpha \ \lambda)$, $(\beta \ \gamma)$, $(\mu \ \sigma)$, $(\delta \ \epsilon)$ and $(\zeta \ \xi)$ are function normalization constants quantified in later section. The computation is further reduced by taking the inter dependency of the parameters. If the nodes generate same amount of data the cumulative frame size $F(i)$ increases as distance $d_{GI}(i)$ to N_0 node reduces ($F(i) \propto 1/d_{GI}(i)$). Hence these factors can be combined together. Similarly the maximum possible link speed depends on the carrier power as $L_{(k \to i)} = B_W \log(1 + \frac{P_S(i)}{P_N})$, where P_N is the ambient noise power [103]. Considering these dependencies the link weight function is simplified as (6.8) after setting $\sigma_0 = \sigma B_W$.

$$w(\mathbf{W}_i^k) = \mu + \alpha e^{-\lambda d_{GI}(i)} + \delta e^{-\epsilon / d_{GI}(i)} + \beta e^{\gamma P_S(i)} - \mu e^{-\sigma_0 \log(1+SNR)} + \zeta b(i) e^{\xi b(i)} \tag{6.8}$$

The transmitted signal power $P_T(i)$ of the node N_i attenuates to $P_S(i)$ when it reaches the node N_k and this is related as $P_S(i) = G_T P_T(i) \frac{1}{4\pi |\mathbf{r}|^2} \frac{\lambda_W^2}{4\pi} G_R$, where G_T and G_R are the transmitter and receiver antenna gain, λ_W is the wavelength and $\mathbf{r}$ is the distance between nodes. If all nodes transmit the beacon frame at $P_T(i) = P_{max}$ power; and if the antenna gain is same (G_A) for all nodes the power ratio $\eta = 10 \log(P_T(i)/P_S(i))$ can be written as (6.9).

$$\eta = 20 \left(\log(\frac{4\pi |\mathbf{r}|}{\lambda_W}) - \log(G_A) \right) = 10 \log \frac{P_{max}}{P_S(i)} \tag{6.9}$$

If the node N_k detects that the signal strength of the beacon frame $P_S(i) > P_{sen}$ (sensitivity of the node), then decreases the link power to a minimum enough to meet the node N_i's sensitivity. From the relation $G_A P_{Tmin} \frac{1}{4\pi |\mathbf{r}|^2} \frac{\lambda_W^2}{4\pi} G_A = P_{sen}$ the minimum transmission power required is given in (6.10).

$$P_{Tmin}|_{(k\to i)} = 10\log(P_{sen}\frac{P_{max}}{P_S(i)})\ dBi \tag{6.10}$$

The access point mode beacon broadcast from N_k is maintained at P_{max}. The data frame size is automatically reduced using compression feature of sparse measurement described in the previous section. This is also included in link weight function as increase in energy level parameter: $b(i) = b_i + c(i)$, where $0 < c(i) < 1$ represents the compression rate. The link weight function is changed as (6.11) after setting $\mu = \alpha$.

$$\begin{aligned} w(\mathbf{W}_i^k) = \alpha(1 \quad + & \quad e^{-\lambda d_{GI}(i)} + e^{-\epsilon/d_{GI}(i)} + b(i)e^{b(i)-1} \\ + & \quad e^{\gamma P_S(i)+1} - e^{-\sigma_0 \log(1+SNR)}) \in \mathbb{R}^{\kappa} \end{aligned} \tag{6.11}$$

This function can be implemented as 4 lookup tables corresponding to each variable. The network routing starts from the node proximal to N0 node. The APQ nodes connected to the N0 node, broadcast the beacon frame with the information $\{\ d_{GI}(0) = 0,$ $L(0) = L_0,\ F(0) = F_0,\ b(0) = 1\ \}$. If the node N_k gets the beacon broadcast from multiple nodes like Ni, Nj etc, the routing algorithm $\mathcal{R}(.)$ determines the optimal node using (6.11) and establishes the connection.

6.3.1 Sensor Network State Parameters

After establishing the upstream connection, the node then defines its current state S_k (6.12) using the available information, where $Ch(k)$ is the WiFi channel, T_{pdi} is the propagation delay and $L(k)$ is the $(N_k \to N_i)$ link speed of the upstream link. $P_S(k)$ is the received power and $\mathbf{R}(k)$ is the route table from the node to N_0 node.

$$\mathbf{S}_k \quad = \quad [d_{GI}(k)\ P_S(k)\ L(k)\ F(k)\ b(k)\ Ch(k)\ T_{pdi}\ \mathbf{R}(k)\ w(\mathbf{W}_i^k)\ P_{Tmin}]^T \tag{6.12}$$

The following parameters are updated with the information obtained from the upstream node: $d_{GI}(k) = d_{GI}(i) + 1$, $F(k) = F(i) + F_{k0}$. where F_{k0} is its own data

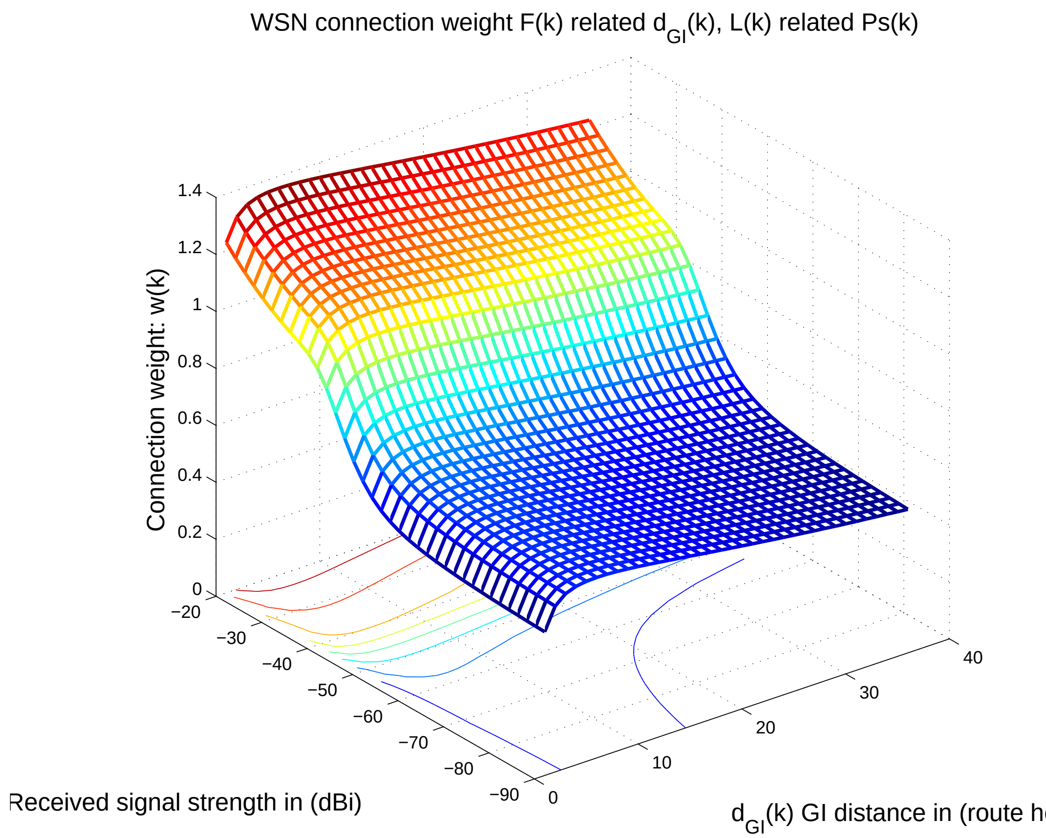


Figure 6-4: The graphical representation of the normalized link weight computation function.

Note: The link speed is taken as function of signal strength $L(i) = B\log(1+\frac{P_S(i)}{P_N})$ *and the frame size is taken as function of hop distance* $(F(i) = 1/\epsilon d_{GI})$. *The combined link weight function is computed for* $B = 2000kbps$, $P_S(k) \in [-90 - 20]dBi$, $P_N = 0.001Watts$, *and* $d_{GI}(i) \in [1 \ldots 36]$.

frame size and $b(k) = b_k$. These values determine the quality of link through this node. The node N_k then starts the transmission of its beacon frame containing state parameters in access point mode through its selected channel. The state parameters S_k is also send to its upstream node. When the other downstream node N_m connects to N_k the frame size is updated as $F(k) = F(i) + F_{k0} + F_{m0}$. As new node connects to N_m the frame size of N_k gets updated again as $F(k) = F(i) + F_{k0} + F(m)$. This new information is updated through the beacon frame. Other nodes connected to N_k can switch to any alternate node if the link becomes suboptimal for it. This adaptability strengthens the network structure.

6.3.2 Node Weight Scaling Parameters

The values of sensitivity and scaling parameters in the link weight function (6.11) are selected to keep $w(\mathbf{W}_i^k) < 4$ as given in (6.13)

$$\lambda = \frac{1}{\lceil\{d_{GI}(i)\}\rceil}, \gamma = \left|\frac{1}{\lceil\{P_S(i)\}\rceil}\right|, \sigma = \frac{1}{\lceil\{L(i)\}\rceil}$$

$$\epsilon = 1/100, \qquad \xi = 1, \qquad \sigma_0 = \sigma B_W$$

$$\alpha = 1/e, \quad \beta = 1, \quad \mu = 1/e, \quad \delta = 1/e, \quad \zeta = 1/e^2 \tag{6.13}$$

Using these values the link weight function is modified as (6.14). The graphical representation of this normalized weight is shown in figure 6-4. It is interesting to note that the node proximal to the data processing unit is not always energy efficient when the entire WSN is considered as a single entity.

$$w(\mathbf{W}_i^k) = \frac{1}{e}(1+e^{-\lambda d_{GI}(i)}+e^{-\epsilon/d_{GI}(i)}+b(i)e^{\xi b(i)-1}+e^{\gamma P_S(i)+1}-e^{-\sigma_0 \log(1+SNR)}) \in \mathbb{R}^\kappa \tag{6.14}$$

6.3.3 Failure Detection and Routing Switching

The data unavailability due to intermediate node failure is avoided by switching to alternate link when no-response is obtained within T_{pd} time. To determine new link the route logic is modified to N-point routing algorithm $\mathcal{R}^N(k)$ this gives a set of indexes $V^k \subset \Omega^k$ ordered according to its weight.

$$V^k = \mathcal{R}^N(k) : \forall N_k \quad \exists V^k \subset \Omega^k \quad \text{s.t.}$$

$$V^k = \text{N Link Index (Sort}\{w(\mathbf{W}^k)\} \in \mathbb{R}^\kappa) \tag{6.15}$$

where $|V^k|_0 = N < |\Omega^k|_0 = \kappa$. The algorithm selects M links for multipath routing. The route table information (6.16) for data routing from an AQ node at the boundary of the network (N_b) to data collection node N_0 ($N_b \rightarrow \{N_k\}_K \rightarrow N_0$) is collectively obtained from the route table available in the state parameters $\mathbf{S}_k$ of the upstream nodes.

$$\mathbf{R}(b) = \{k_1 \ldots k_K\} \subset \Omega \quad k_n = \mathcal{R}(k_{n-1}), k_0 = b, \quad n = 1 \ldots K, \quad k_K = 0 \quad (6.16)$$

where $\Omega = \bigcup_{k=0}^{K-1} \Omega^k$. The route table $\mathbf{R}(b)$ is included in the system parameter $\mathbf{S}_k$ to avoid cyclic routing and if the node finds its own number that path is avoided. Based on the routing strategy discussion above the *resource constrained adaptive multipath routing* (RCAMR) for autonomous sensor network is described in Algorithm-10. The timing diagram of the communication protocol is given in figure 6-5.

6.3.4 Network Analysis

According to the communication scheme discussed the access point node control the data transmission. The protocol timing of communication between upstream node N_0 and downstream node N_1 is illustrated in figure 6-5 with respect to the N_1 node. The node senses the upstream communication link for t_{sense} duration and if the channel is available, transmits its load of data to the upstream nod. This communication exists for t_{tx} duration. Upon successful communication the acknowledgment frame is received from the upstream node and the channel is left in sense mode. Other nodes for which N_0 is upstream, transmits their data. The entire upstream communication extends up to $t_{txcycle}$ duration depending on the number of nodes connected to N_0 node. When all nodes have transmitted data the N_0 node broadcasts the beacon frame containing the state parameter information S_0 for t_{bcx} duration. All the nodes are programmed to remain in idle mode for t_{idle} duration after the beacon frame reception, during this time the network is available for other nodes waiting to establish connection. The upstream communication cycle continues after this idle time. In the downstream the communication happens in a different channel $Ch2$. The N_1 node in access point mode listens to transmission from its downstream nodes. The data is received in succession from its n downstream nodes. For every successful reception the N_1 node transmits ack frame. This communication lasts for $t_{rxcycle}$. After reception,

Algorithm 10
Resource Constrained Adaptive Multipath Routing

Require: SSID,
Task: determine fault tolerant multipath $\{k \rightarrow i\}_N$ and forward data
Initialization: (6.13)
if GI=true **then** *exit* **else**
  scan: all WSN channel SSID
  decrypt: SSID $\rightarrow$ PWD
for SSID **do**
  connect : node N_k with $[SSID, PWD]$
  if connected **then**
    get: $\mathbf{S}_k$ update Ω^k, $\mathbf{W}^k$: *disconnect*
  end if
    scan: next
end for
  generate: $\mathbf{W}^k$ using (6.5)
for $i \in \Omega^k$ **do**
  compute: $w(i)$ using (6.7)
  sort: $\mathbf{W}^k$ *get* : V^k *st* $|V^k| = N$
end for
if $|V^k| > N$ **then**
  prune: $|V^k| \rightarrow N$, $j = 1$
  $configure(i)$: link $i = \{V^k\}_i$
end if
while route **do**
  compute: P_{Tmin} using (6.10)
  connect: N_i using [SSID, PWD]
  (Thread Tx)
    send: $\mathbf{S}_k$ and [data] to N_i
    wait T_{pd} : **if** $NO - ACK$: $configure(i + 1)$
    wait $RxBcx$: **if** $NO - BCX$: $configure(i + 1)$
    wait T_{idle}: *continue*
  (Thread Rx)
    receive $\mathbf{S}_m$ and [data] from N_m
    send: ACK
    wait: $T_{rxcycle}$
    timeout t_{sense}: *broadcast* $\mathbf{S}_k$ in AP frame
end while
if *receive*: $\mathbf{S}_m$ **then**
  update: $\mathbf{S}_k$,
  send: $\mathbf{S}_k$ to N_i
end if
if $linkfail$: **then Do** $configure(i + 1)$
Output: link $k \rightarrow i$

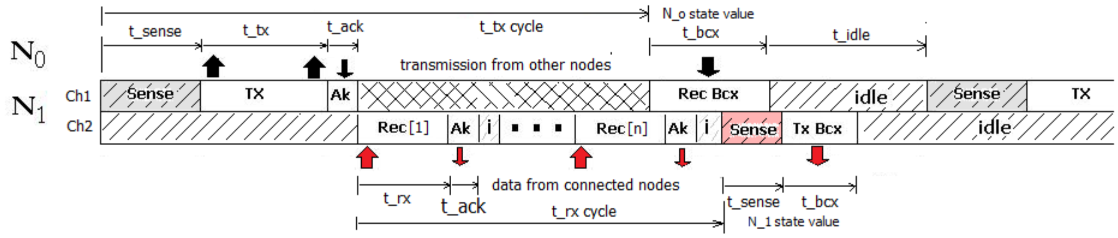


Figure 6-5: Protocol timing diagram of data aggregation and forwarding node.

the node senses the channel for t_{sense} duration to confirm the channel silence and the node then transmits the beacon frame containing its state parameter S_1. As earlier, all nodes in the network remains in idle mode for t_{idle} duration and during this time the network is available for other nodes waiting to establish communication with N_1. The following definitions are made to have a clear understanding of the communication protocol. The data aggregation network established by the node N_k is synonymously called as network N_k, Ω^k: set of upstream nodes of N_k, Φ^k: set of downstream nodes for which N_k is the access point, $\mathcal{P}(N_l)$ transmission probability of node N_l in the network N_k, $\mathcal{P}(c_k|N_l)$ conditional probability of collision when N_l transmits. The probability of collision in the network N_k can be written as $\mathcal{P}(c_k) = \sum_{l\in\Phi^k}\mathcal{P}(c_k|N_l)\mathcal{P}(N_l)$. However, the conditional probability of collision is difficult to estimate for every node. The probability of any transmission in network N_k can be expressed as (6.17)

$$\mathcal{P}(t_k) = 1 - \prod_{l\in\Phi^k}(1-\mathcal{P}(N_l)) \tag{6.17}$$

The probability of collision free transmission happening from any downstream node N_m to N_k can be written as $\mathcal{P}(N_m)[\prod_{l\in\{\Phi^k-m\}}(1-\mathcal{P}(N_l))]/[1-\prod_{l\in\Phi^k}(1-\mathcal{P}(N_l))]$. When the node N_k is working reliably, the probability of successful reception by N_k is same as probability of successful transmission from all the nodes in downstream network as given in (6.18)

$$\mathcal{P}(r_k) = \sum_{m\in\Phi^k}\frac{\mathcal{P}(N_m)\quad\prod_{l\in\{\Phi^k-m\}}(1-\mathcal{P}(N_l))}{1-\prod_{l\in\Phi^k}(1-\mathcal{P}(N_l))} \tag{6.18}$$

Using these 2 expressions the collision probability can be expressed as $\mathcal{P}(c_k) = \mathcal{P}(t_k)(1 - \mathcal{P}(r_k))$. The probability that the network N_k is in the idling state can be written as $\mathcal{P}(i_k) = \prod_{l\in\Phi^k}(1 - \mathcal{P}(N_l)) = 1 - \mathcal{P}(t_k)$.

If t_c is the average collision time per cycle observed in the network, the total channel usage time for one communication cycle using the protocol can be written as (6.19).

$$\begin{aligned} T_{proto} &= (t_{idle} + t_{sense})\mathcal{P}(i_k) \\ &\qquad + (|\Phi^k|_0(t_{rx} + t_{ack}) + t_{bcx})\mathcal{P}(t_k)\mathcal{P}(r_k) + t_c\mathcal{P}(c_k) \end{aligned} \tag{6.19}$$

where $|.|_0$ gives number of elements in the set. The effective time utilized by all the downstream nodes ($N_l, l \in \Phi^k$) for real data communication including basic Headers (H), short interframe space (S_{SIF}) and acknowledgment (A_k) is computed as (6.20).

$$\begin{aligned} T_{data} &= |\Phi^k|_0(\frac{H + S_{SIF} + A_k}{L(k)} + 2T_{pd}) \\ &\qquad + \sum_{m\in\Phi^k}\frac{F(m)}{L(k)}\mathcal{P}(N_m)\prod_{l\in\{\Phi^k - m\}}(1 - \mathcal{P}(N_l)) \end{aligned} \tag{6.20}$$

where $L(k)$ is the link speed and T_{pd} is the propagation delay of the network. $F(m)$ is the frame size of the N_m node. The throughput of the network N_k can be computed as

$$\mathcal{T}_k = \frac{T_{data}}{T_{proto}} \tag{6.21}$$

The state transition diagram of the protocol and the timing diagram of the communication protocol using the described algorithm are given in figure 6-6. The communication channels are colour coded as the network described in figure 6-3.

6.3.5 Power Dissipation

The factors that determine the practicality of WSN routing algorithms are energy dissipation, number of nodes retained and the distribution of nodes after certain amount of routing cycle. The power dissipation of the routing algorithm is computed

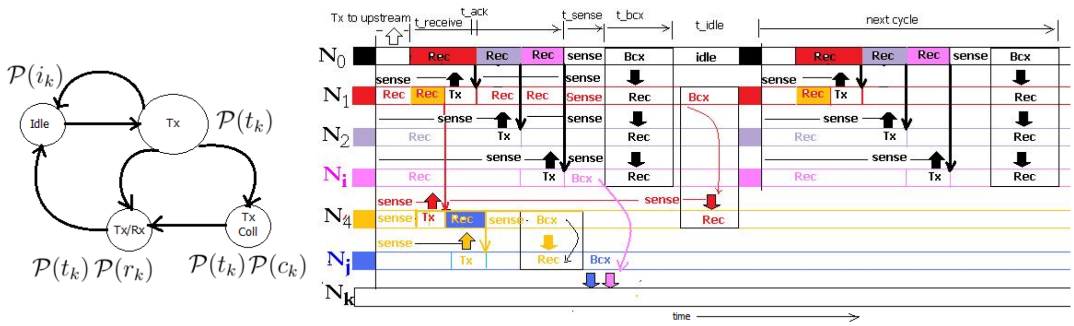


Figure 6-6: Network communication timing diagram.

using the following expressions (6.22).

$$E_{RX} = (E_{el} + E_{DA})F(i)$$

$$E_{TX} = (E_{el} + E_{amp}d_{GI}^2)F(i)$$

$$E_{CH} = (E_{el} + E_{da} + E_{amp}d_{GI}^2)F(i) \tag{6.22}$$

The parameters are set as follows: initial energy $E_0 = 1J$, bit processing $E_{el} = 50nJ$, data aggregation $E_{da} = 5nJ$, RF amplifier system $E_{amp} = 100pJ$, frame size $F(i) = 4000$ and number of nodes $N = 100$. The probability that the node function as cluster head is $p = 0.5$. The algorithms used for comparison are LEACH[78], enhanced SEP [81], and DEEC[79]. The simulation is run till 50% of nodes get depleted of its energy. The figure 6-7 shows the energy consumption in the network established using various algorithms. The energy dissipation of the proposed algorithm varies during multipath route switching when the number of nodes start falling below 95%. After 50% of the nodes get depleted the power consumption in the algorithms varies considerably. With the same initial condition, number of active nodes falls to 50% in 600 to 1000 routing cycles for the algorithms compared while the proposed RCAMR algorithm takes 2.8 times longer for 50% power depletion. The node attrition rate of these clustering and routing algorithms is shown in figure 6-7(b). It is observed that when the nodes use RCAMR algorithm there is considerable increase in the lifetime of nodes and hence the overall network. The advantage of the proposed algorithm can be seen when analysing the distribution of the nodes in the network after half of the nodes get depleted. The node distribution at 50% energy level is shown in figure 6-8. The RCAMR algorithm retains the overall node distribution of the network for longer duration, at the expense of finite increase in energy consumption. This signature can be seen in the energy graph figure 6-7(b), while network using other algorithms are at the end of its lifetime the proposed algorithm reconfigures the network to remain active. This algorithm retains the network 1.5 times longer than extended stable election protocol. this is the consequence of progressive reconfiguration of the routes to minimize the energy consumption of the entire network Hence in network established by RCAMR the distribution of the measurement is maintained for longer duration and is suitable for applications where this distribution of the measurements are necessary.

6.3.6 Reliability of Multipath Routing

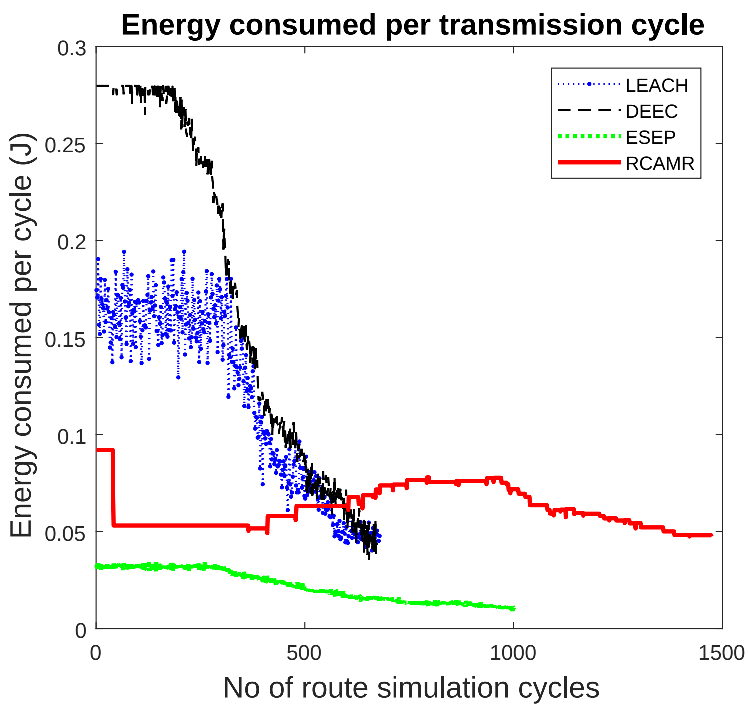
The reliability of the network with K routing nodes in series from the boundary node N_b to the sink node N_0 ($N_b \to \{N_k\}_K \to N_0$) can be computed as (6.23).

$$R(t) = \prod_{k \in \mathbf{R}(b)} e^{-\varepsilon_k t}, \quad b \in \partial\Omega \subset \Omega, \quad |\mathbf{R}(b)|_0 = K, t \in [0, \infty] \tag{6.23}$$

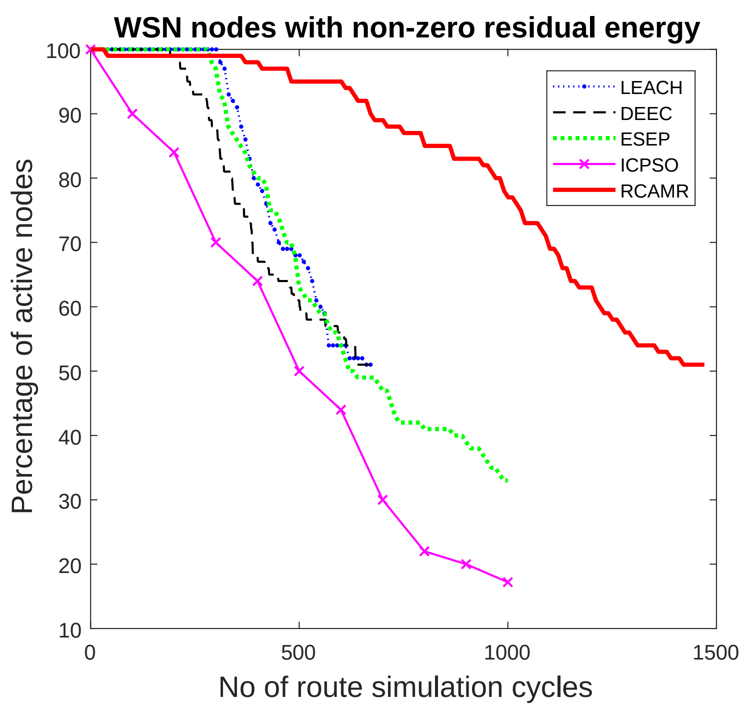
where $\mathbf{R}(b)$ is the route table described in (6.16), Ω is set of all nodes, $\partial\Omega$ is boundary set, t is the operational duration, $\varepsilon_k = 1/t_{Fk}$ and t_{Fk} is finite mean time to failure of the node N_k. The fault tolerance is achieved using M links ($V_i^k, i = 1 \ldots M$) selected by $\mathcal{R}^N(k)$ for multipath routing. The reliability of this configuration is given in (6.24) assuming the measure of reliability $e^{-\varepsilon_k t}$ is identical for all nodes.

$$R(t) = \prod_{k \in \mathbf{R}(b)} \sum_{m=M}^{N} \binom{N}{m} \quad e^{-m\varepsilon_k t} \quad (1 - e^{-\varepsilon_k t})^{N-m} \tag{6.24}$$

Every node N_k maintains N number of routes leading to the sink node N_0 connected through to the $K - 1$ links. Considering nodes with different reliability measures $\Lambda_k = \{\varepsilon_k^i \subset \mathbb{R} \mid i \in V^k, k \in \mathbf{R}(b)\}$. After sorting elements of Λ_k in ascending order corresponding to nodes with higher to lower reliability, the set can be split into two



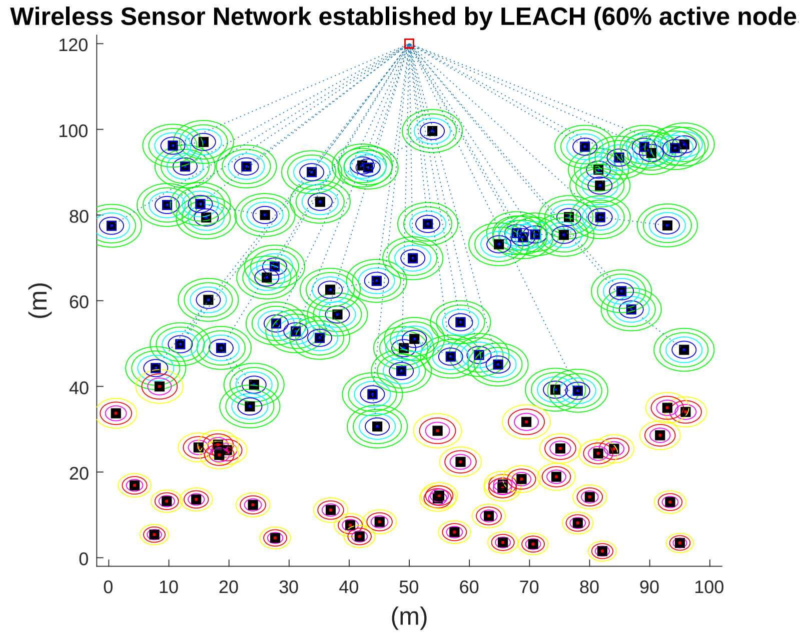
(a)



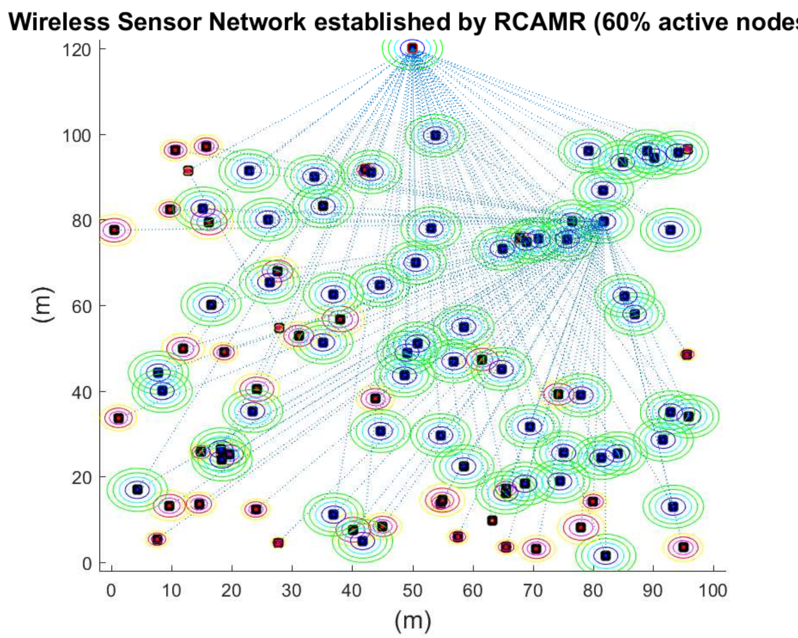
(b)

Figure 6-7: The energy dissipation and node attrition rate of routing algorithms.

Note: (a) Energy dissipated per transmission for various clustering algorithms. (b) The attrition rate of nodes for various clustering and routing algorithms.



(a)



(b)

Figure 6-8: The comparison of distribution of the WSN nodes established by LEACH and RCAMR routing algorithms.

Note: The green circles indicate the active and red circles indicate the energy depleted nodes. The radius of the circle indicates the duration of active existence. The network is shown after half of the total nodes are depleted of its energy. (a) The network established by LEACH algorithm: the farthest nodes get depleted fast and the network becomes localized. (b) The overall distribution of the active nodes is maintained by RCAMR even after 50% of the nodes is depleted.

parts (i) Λ_k^m with m elements (ii) $\Lambda_k^{\bar{m}}$ with $\bar{m} = N - m$ elements. The reliability of this redundant routing network can be written as (6.25).

$$R(t) = \prod_{k \in \mathbf{R}(b)} \sum_{m=M}^{N} \binom{N}{m} \prod_{\varepsilon_k \in \Lambda_k^m} e^{-\varepsilon_k t} \prod_{\varepsilon_k \in \Lambda_k^{\bar{m}}} (1 - e^{-\varepsilon_k t}) \tag{6.25}$$

6.3.7 Reliability Estimation by Testing

The network reliability evaluation based on the measurements obtained from the accelerated degradation testing is presented here. The processes that affect the system reliability are (i) thermo-mechanical (TM) stress induced failure like PCB warping, track breaking or solder disconnection, (ii) electrical (EL) stress induced failures like electro migration or track burn out and (iii) thermo-environmental (TE) stress induced failures like dendrite formation on PCB. Considering these effects the reliability of the network can be estimated through accelerated degradation testing of a sample node. Using the Eyring model [96] the mean time to failure ($mttf$) can be written as (6.26).

$$t_{F(TM)} = \hat{t}_{F(TM)} \left(\frac{T_a}{T_0}\right)^{\beta_T} \times \exp\left(\frac{E}{k_B}(\frac{1}{T_0} - \frac{1}{T_a}) + \sum_{i=1}^{i=Z} A_i \Delta S_i + B_i \Delta S_i^T\right) \tag{6.26}$$

$$\Delta S_i = S_{i0} - S_{ia} \quad \Delta S_i^T = \frac{S_{i0}}{T_0} - \frac{S_{ia}}{T_a}$$

where, $\mathbf{S}$ is stress, T_a is accelerated testing temperature, T_0 is the operational temperature, $\hat{t}_{F(.)}$ is the estimated time to failure during accelerated testing, β_T is the power of temperature interaction, A_i and B_i are empirical constants determining the stress interaction, ΔS_i and ΔS_i^T are the difference in stress and stress per temperature rise experienced in operational temperature and accelerated test temperature. k_B is the Boltzmann constant and E is the activation energy of the PCB track metal. The time to failure due to thermo electrical stress induced electro migration is given in (6.27), where J_0 and J_a are the current densities at the dendrite formation point at operational and accelerated testing temperatures.

$$t_{F(EL)} = \hat{t}_{F(EL)}(J_a/J_0) \tag{6.27}$$

Time to failure due to thermo environmental dendrite growth $t_{F(TE)}$ due to high humid environment is given in (6.28), where R_{H0} is the ambient relative humidity, V_0 the maximum electric field experienced between PCB tracks, and D_0 is the average distance between PCB tracks.

$$t_{F(TE)} = \hat{t}_{F(TE)}(\frac{R_{Ha}}{R_{H0}})(\frac{V_a}{V_0})(\frac{D_0}{D_a})e^{\frac{E}{k_B}(\frac{1}{T_0}-\frac{1}{T_a})} \tag{6.28}$$

Using the estimated $mttf$ the reliability computed in (6.24) and (6.25) is modified as (6.29) where $t_F = \min\{t_{F(TM)}, t_{F(EL)}, t_{F(TE)}\}$

$$R(t)_{net} = e^{-t/t_F}R(t) \tag{6.29}$$

The combinatorial term $\binom{N}{m}$ can be avoided using the *Local Theorem of DeMoivre–Laplace*, which defines the probability of m instances of N events as follows.

$$\lim_{N\to\infty} \sqrt{pqN}\; P_N(m) = \frac{1}{\sqrt{2\pi}}e^{(-\alpha^2/2)},$$
$$\alpha = \frac{m-pN}{\sqrt{pqN}}, \quad \{0 < \alpha \ll \infty\} \tag{6.30}$$

where $p = e^{-\varepsilon_k t}$ is the probability of functioning of a unit, $q = (1-p)$. and $P_N(m)$ is the probability that m instance of N units function correctly. Using (6.30) the reliability of the redundant routing network can be expressed as (6.31).

$$R(t)_{net} \approx e^{-t/t_F} \prod_{k\in\mathbf{R}(b)} \frac{1}{\sqrt{2\pi pqN}} \sum_{m=M}^{N} e^{-\frac{(m-pN)^2}{2pqN}} \tag{6.31}$$

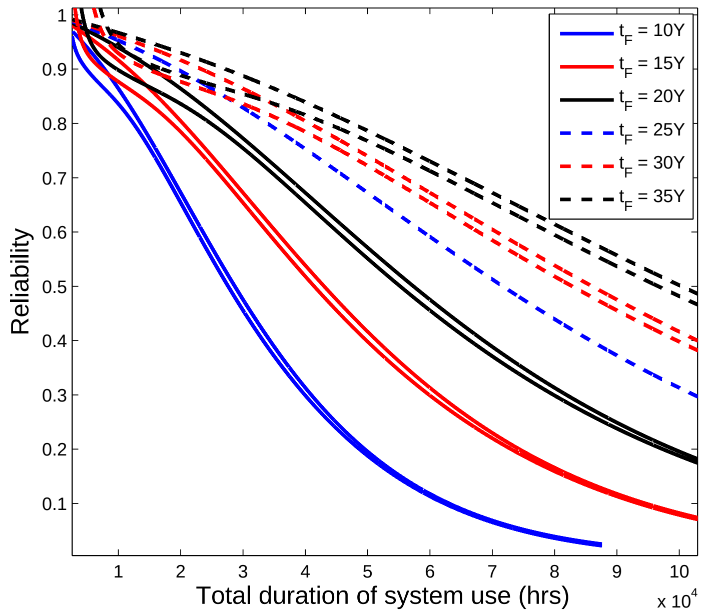
For a simple multipath network with 9 nodes, one source, one sink and 7 routing nodes with 2 out of 3 redundancy, the reliability expression is computed as (6.32) and is shown in figure 6-9.

$$R(t)_{net} \approx e^{-t/t_F} \frac{1}{\sqrt{6\pi pq}} \sum_{m=2}^{3} e^{-\frac{(m-3p)^2}{6pq}} \tag{6.32}$$

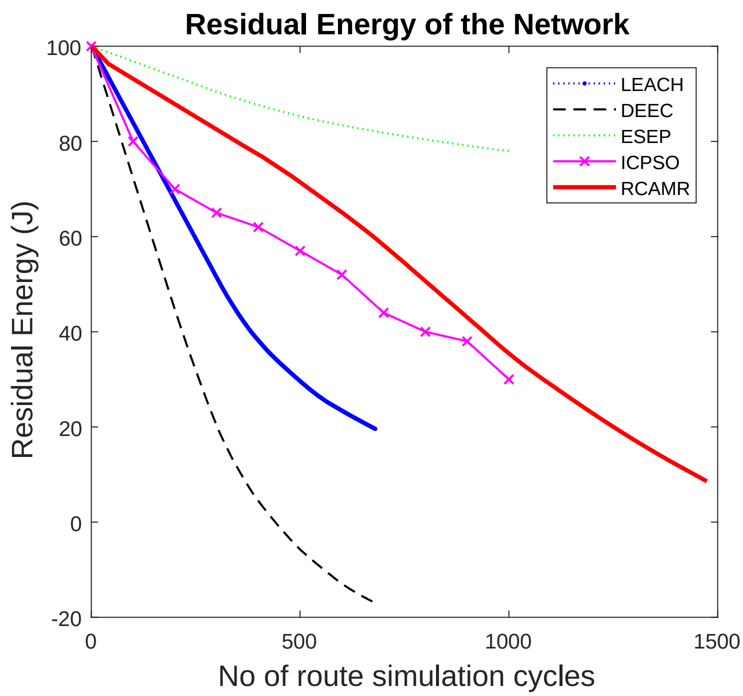
The figure also shows the computed reliability value from DeMoivre-Laplace theorem and the value obtained from the binomial theorem; for estimation purpose the reliability can be computed using (6.31). It is observed from figure 6-9 that as the design constraint for the node becomes stringent (higher t_F values), the reliability of the network increases. The reliability of the network during initial 7 year period of operation is > 0.75. The low-cost IoT processor based devices can be used for parameter monitoring, if it is planned to be replaced in every 3 to 4 years, to have reasonably high confidence level on the WSN. Even if high reliable components are used in the fabrication of these devices, as long as these devices are left outside they will degrade. Hence it is logical to use low cost IoT processors based WSN devices with a plan for regular replacement or additions, also, considering the cost of the plant maintenance the cost involved in the short time usage of low cost IoT based devices for sensor network will be negligible.

6.3.8 System Evaluation

The features of IoT platform processor considered are the cost, availability of development environments, OS and file system support, programming modularity and availability. The WSN nodes with compression features are created using TI CC3200 WiFi processor based boards with transmitter power $P_{max} = 17.3dBm$ and receiver sensitivity $P_{sen} = -90dBm$. The minimum transmitter power for this node is $P_{Tmin} = -72.7dBm - (P_S(i)dBm)$. The data acquisition node with routing feature is implemented using TI AM3358 ARM processor based board. The analysis of network realtime capability of this board is presented in Chapter 5. The link weight (6.11) is not computed in realtime, but, stored as 3 function table for the variables d_{GI}, P_S and b. A simple WSN is established using 1 sink node, 3 data acquisition and routing (APQ) nodes and 1 data acquisition (AQ) node. The nodes uses RCAMR algorithm



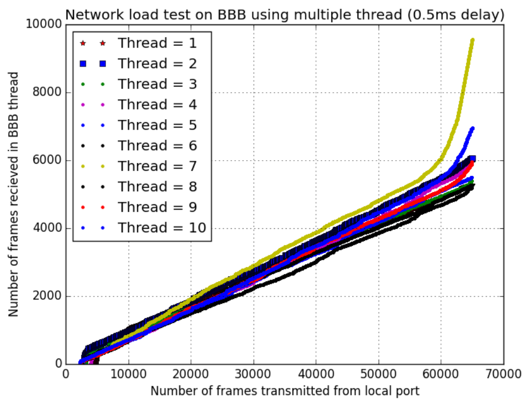
(a)



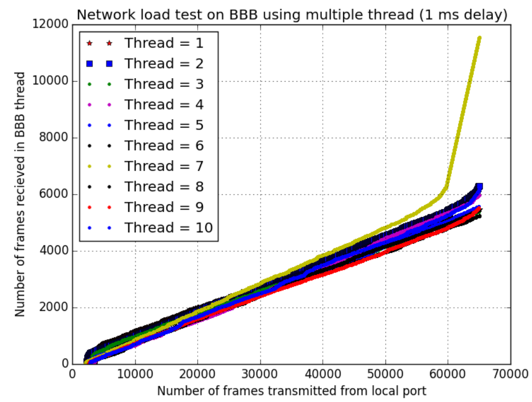
(b)

Figure 6-9: The analytical value of the reliability and estimated depletion of the residual energy.

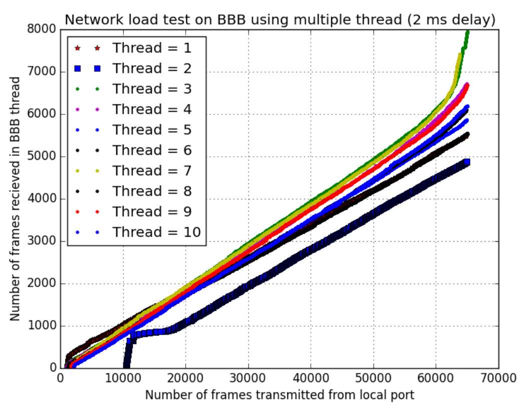
Note: (a)The analytical value of reliability obtained from DeMoivre-Laplace theorem (6.31) and binomial theorem (6.24). The system designed with mttf $t_F > 25$ *years has reliability* ≥ 0.75 *for an initial period of 4.5 years. (b) The depletion in residual energy of the network after 1000 simulated routing cycles for various algorithms.*



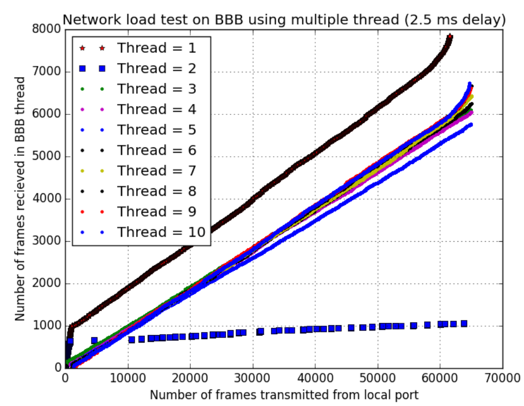
(a)



(b)



(c)



(d)

Figure 6-10: The frame loss in the data acquisition node established using the R-CAMR algorithm for various inter frame delays.

Note: Number of frame received to number of frames transmitted for various inter frame delay, 10 units of TI CC3200 WiFi processor based data acquisition units are programmed to transmit data frame continuously with inter frame delay of 0.5, 1.0, 2.0 *and* 2.5*ms. The TI AM3358 ARM processor based beaglebone black (BBB) board is programmed to receive the data frame using RCAMR algorithm. The loss less reception of data is achieved when there is a minimum inter frame interval of* 2.5*ms for every data transmission nodes.*

and are programmed to generate 256 bytes of data and transmit in every fixed interval. The data handling capacity for various frame to frame delays is tested and is shown figure 6-10. 100% data delivery is obtained when there is minimum interframe delay of 700μ sec. The corresponding power dissipation is also shown in the figure 6-10.

6.4 Chapter summary

This chapter discusses about the resource constrained wireless sensor networking system for distributed parameter measurement using autonomous routing capability and link fault tolerance. The data processing unit of the sensor node is implemented using low cost IoT platform and the sparse measurement is used for transient voltage acquisition. The sparse measurement based acquisition system gives the advantage of having high transient signal sampling capability at reduced data bandwidth. The high sampling is needed for capturing transient changes in the signal. And the low bandwidth is a desirable feature for the autonomous data routing network implemented using low power WiFi enabled devices. The relaxation based recovery algorithm is used for reconstruction of the transient signal at the data processing station. To make the signal sparse, the thresholding of the signal is done prior to acquisition. The ground potential voltage acquired is transmitted to the central data processing system through a network of wireless sensors nodes, which also function as the routing nodes. Computationally minimalistic routing algorithm is designed and incorporated into IoT based wireless sensor network nodes. The power dissipation is minimized by exploring possible options like adaptive RF power, considering the next routing node's sensitivity, routing to node with higher energy backup and data compression capabilities. The link weight computation is stored as look-up table. This adaptive wireless sensor networks can be deployed randomly or orderly in vast area to gather spacial and temporal information. The routing algorithm presented here has fast route discovery and adaptive capabilities. From the analysis it is found that the optimal connection node is 3-5 links away from the data processing node. The WSN scheme for ground potential monitoring presented here has the following advantages (1) lower computational requirement, (2) compatible with low profile IoT platforms, (3) easily deployable, scalable and expendable, (4) failure tolerant autonomous routing capability, (5) reliable and (6) maintains node distribution even in power depleted phase. A general model for evaluating the reliability of the multipath resource constrained routing algorithm is also presented. The actual reliability value are calculated from mean time to failure t_F determined from accelerated degradation testing of a node. A reliability model for heterogeneous redundant routing network is also discussed. To summarize, the chapter presents a lightweight routing algorithm for IoT based wireless sensor network established for distributed sparse measurement of transients in ground voltage potential.

Chapter 7

Research Summary

The compressed sensing is a data acquisition method which supports low frequency sampling of sparse signals. As the compressed sensed signals are not direct time domain representations, the signals need to be reconstructed to its original form using reconstruction algorithms. Large numbers of sparse signal reconstruction algorithms are developed in the recent time. These algorithms are presented with their unique merits. The plethora of sparse recovery algorithms with different characteristics creates a dilemma while choosing the suitable one for the given application. The conventional metric used to compare the sparse signal recovery algorithms are relative MSE and probability of support recovery. These two metrics need to be analysed together. A method to evaluate the performance of the algorithms by using these two metrics to compute a signal similarity measure between the original signal and the reconstructed signal is proposed. Two performance characterization functions namely signal similarity measure $Sm(\hat{\mathbf{x}}, \mathbf{x})$ and sparse recovery limit ξ are proposed. This measure use the relative MSE, the probability of exact support recovery, the (K/M) ratio and the (M/N) ratio to generate a numerical figure of merit. This new metric is evaluated for 24 algorithms from 8 different categories and is experimentally shown that the proposed method gives a quantifiable performance comparison.

There is a fundamental constraint in the sparse reconstruction problem. The sparse measurement matrix is analytically non invertible, hence other methods like iteration, thresholding and function minimization are adopted for signal reconstruction. This includes function minimization of ℓ_0 or ℓ_1 or ℓ_p based objective functions. An inverse operator for the measurement matrix $\mathbf{A}$ is also needed. Many of the current algorithms use $A^\dagger$ or A^T as this inverse operator. A general framework using arbitrary inverse matrix $\mathbf{Q}$ for the development of sparse recovery algorithms is proposed. This algorithm framework enable the developers to simulate the signal reconstruction using various pseudo inverse matrices and improve the reconstruction performance. The method is used in the development of two improved algorithms based on ℓ_1 and ℓ_0 minimization. The former is based on iterative segmented thresholding of ℓ_1 residue with the inverse operation. The later is based on segmented thresholding of polynomial approximation of ℓ_0 function. The logic for selecting residue and minimizing it for arriving at optimal sparse solution is described. A range alterable segmented thresholding function is proposed and used in the final stage of the iteration.

The ℓ_0 minimization based sparse signal recovery method STXEL0 is reconfigured to a cascaded computational network, to enable the algorithm implementation on low profile computing platforms for real time use. The gradient minimization functions are evaluated for various values of the algorithm parameter and are stored in RAM as a library of polynomial function tables, to reduce processing load. The computational complexity of the hardware implementation is described in terms of the basic MAC units. The computational precision of the algorithm is evaluated and the optimal value of the algorithm regularization parameters are estimated experimentally. The convergence of the iteration is verified by the continuous reduction in the internal error estimates of the algorithm. Experimental evaluation of the algorithms are carried out and the results are compared with the seven different classes of methods. The analysis of the results shows that the ℓ_0 minimization based STXEL0 algorithm gives better SNR in the reconstruction of images with lesser processing time. The theoretical analysis of convergence guarantee for any arbitrary inverse matrix is not discussed. However, if arbitrary matrix is selected as the inverse, the convergence is influenced by the value of the regularization parameter. The proposed architecture of the algorithm supports the implementation of the STXEL0 algorithm with basic MAC units and function tables, for real time sparse recovery.

To evaluate the sparse recovery algorithm in real time application a distributed data acquisition system for the measurement and recovery of the sparse signals using IoT based processing board is envisaged. To implement this system the AM3358 processor based beagle board is selected and evaluated to confirm that it meets the computational requirements of the algorithm. The computing platform is evaluated under various network configurations and the results are analyzed. It is found from the evaluation that this ARM processor AM3358 based platform has predictable performance and can handle real time data acquisition process if the throughput latency requirement is $> 20ms$. The performance remains consistent if the data acquisition and transmission bandwidth is < 4.5223 $Mbps$ and the data receive speed is < 4.096 $Mbps$. Also, for a distributed data acquisition system with 32 nodes designed with this board, the node control-command latency requirement should be > 2 ms if the command frame size is ≤ 1024 bytes. The use of software threading do not improve the data throughput and if higher communication bandwidth is required, the network interface chip or module should be replaced to $1Gbps$ capable device. In short the networked data acquisition system implemented using AM3358 processor based beagle board can work in 20 ms realtime periodicity and with $10Hz$ output bandwidth if the number nodes are < 32.

The evaluated computing platform is then used in the development of a distributed data acquisition system for the measurement and recovery of transients in ground potential. The sparse measurement method is used for transient voltage acquisition. To make the signal sparse, the thresholding of the signal is done prior to acquisition. The inadvertent rise in ground potential with respect to the measurement ground can damage the system hence it is designed to be expendable and the low cost IoT platform is selected for this system implementation. The data processing unit of the sensor node is implemented using the evaluated board. The higher sampling is necessary for capturing the transients changes in the signal and the low bandwidth is a desirable for the data routing network. The sparse measurement based acquisition system gives the advantage of having high transient signal sampling capability at reduced data bandwidth. To establish reliable data communication a resource constrained wireless sensor networking system with autonomous routing capability and link fault tolerance is developed. The proposed STXEL0 algorithm is used in the recovery of the transient signal from compressed sensed data. The acquired data is transmitted to the central data processing system through a network of wireless sensors nodes, which also function as the routing nodes. A computationally minimalistic routing algorithm is also designed and incorporated into the IoT based wireless sensor network nodes. The power dissipation is minimized by exploring possible options like adaptive RF power, routing through node with higher energy backup and data compression capabilities. The link weight computation function is stored as look-up table, minimize computation. This adaptive wireless sensor networks can be deployed randomly or orderly in vast area to gather spacial and temporal information. The routing algorithm developed has fast route discovery and adaptive capabilities. A general scheme for evaluating the reliability of multipath resource constrained routing is also presented. In short a lightweight routing algorithm for IoT based wireless sensor network established for distributed sparse measurement of transients in ground voltage potential is illustrated.

In summary this thesis introduces a novel metric for sparse signal recovery algorithm evaluation and presents a framework for developing better sparse recovery algorithms. The proposed ℓ_0 norm based sparse signal recovery method is then used in a distributed system for acquisition of naturally sparse signals. To enable the implementation of the system a low cost IoT device is selected and evaluated to determine its capabilities. Additionally a low power optimal routing algorithm is proposed for establishing this distributed sparse signal sensor network.

Chapter 8

Future Directions

The proposed framework for the development of sparse signal recovery algorithm is used for developing two new algorithms based on ℓ_0 and ℓ_1 minimization. However many different variations of ℓ_p norm based algorithms can be tried using this framework. Similarly, the framework supports evaluation of sparse recovery algorithm with any empirical approximation of the inverse of the measurement matrix. The selection of inverse of the measurement matrix is limited only by the imagination of the user. The segmented thresholding function can be further improved. From the evaluation of may sparse recovery algorithms it is observed that the residue projection and thresholding based methods shows better signal reconstruction. Further scope for improvements exists in the estimation of residue, projection of residue and thresholding. These area can be explored further. In the implementations side various computing modules and networking schemes can be adopted. This work involve analysis, theoretical development, experimental evaluation and networking, the further improvement can be directed towards the latest development in any of this fields.

Chapter 9

Publications Related to Thesis

[1] Vivekanand V. and Deepak Mishra, Framework for Segmented Threshold ℓ_0 Approximation Based Computing Network for Sparse Signal Recovery,
Elsevier Neural Networks (2023).
https://doi.org/10.1016/j.neunet.2023.03.005

[2] Vivekanand V. and Deepak Mishra, Feasibility of using AM3358 beagle board for networked realtime signal acquisition,
Elsevier Internet of Things (2020).
https://doi.org/10.1016/j.iot.2020.100199

[3] Vivekanand V. and Deepak Mishra, Expendable and Distributed Measurement Scheme for Acquisition of Naturally Sparse Events,
Springer Wireless Personal Communications (2022).
https://doi.org/10.1007/s11277-022-10151-z

[4] Vivekanand V. and Deepak Mishra, Performance measure for sparse recovery algorithms in compressed sensing perspective,
Springer International Journal of Data Science and Analytics (2022).
https://doi.org/10.1007/s41060-022-00357-6

Bibliography

[1] D. L. Donoho, Compressed sensing, IEEE Trans. Inf. Theory, 52, 1289-1306, (2006).

[2] E. Candes, J. Romberg, T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Comm. Pure. Appl. Math, 59 (2006) 1207-1223.

[3] Michael Elad, Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing, Springer (2010).

[4] R. DeVore, Deterministic constructions of compressed sensing matrices, J. Complex 23, 918-925, (2007).

[5] S. Foucart, H. Rauhut, A Mathematical Introduction to Compressive Sensing, Applied and Numerical Harmonic Analysis, Springer Science+Business Media New York, (2013).

[6] E. Candes, The restricted isometry property and its implications for compressed sensing, Elsevier Comptes Rendus Math., C. R. Acad. Sci. Paris, Series I, 346, 589-592, (2008).

[7] E. Candes, J Romberg and Terence Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information, IEEE Trans. Inf. Theory, 52, 489-509, (2006)

[8] E. Candes and J. Romberg, Sparsity and incoherence in compressive sampling, Inverse Problems, 23, 969-985, (2007)

[9] Jerome Bobin, Jean Luk Starck, Roland Ottensamer, Compressed sensing in astronomy, IEEE J. Sel. Top. Signal Process. 2 (2008) 718-726.

[10] Y. Zhang, J. Jiang, G. Zhang, Compression of Remotely Sensed Astronomical Image Using Wavelet-Based Compressed Sensing in Deep Space Exploration. Remote Sensing. 13(2) 288 (2021). https://doi.org/10.3390/rs13020288

[11] Qing Maia, Hui Zou, Sparse semiparametric discriminant analysis, J. Multivariate Analysis 135, 175-188, (2015) .

[12] Z. Zhang, Yong Xu, J. Yang, Xuelong Li and D. Zhang, A survey of sparse representation: algorithms and applications , IEEE Access, (2015).

[13] X. Yuan, R. Haimi-Cohen, Image Compression Based on Compressive Sensing: End-to-End Comparison With JPEG, IEEE Transactions on Multimedia, 22, 2889-2904, (2020).

[14] E. Crespo Marques, N. Maciel, L. Naviner, H. Cai and J. Yang, A Review of Sparse Recovery Algorithms, IEEE Access, 7, 1300-1322, (2019).

[15] Steffen Peter, Algorithms for Robust and Fast Sparse Recovery, New approaches towards the noise folding problems and the big data challenge, Thesis, Technical University of Munich (2016)

[16] J. D. Blanchard, M. Cermak, D. Hanle and Y. Jing, Greedy Algorithms for Joint Sparse Recovery, IEEE Trans. Signal Process. 62, 1694-1704, (2014).

[17] J. A. Tropp, A. C. Gilbert , Signal recovery from random measurements via orthogonal matching pursuit, IEEE Trans. Inf. Theory, 53, 4655-4666, (2007).

[18] Tong, Z., Wang, F., Hu, C. et al. Preconditioned generalized orthogonal matching pursuit. EURASIP J. Adv. Signal Process. 21 (2020).

[19] D. Needell and J.A. Tropp, CoSaMP: Iterative signal recovery from incomplete and inaccurate samples, Appl. Comput. Harmonic Anal. 26, 301-321, (2008).

[20] Zhang, Yin., User's Guide for YALL1: Your ALgorithms for L1 Optimization (2009).

[21] Y. Hai-Rong, F. Hong, C. Zhang, W. Sui, Iterative Hard Thresholding Algorithm Based on Backtracking, Acta Automatica Sinica. 37, 276-282, (2011).

[22] A. Beck and M. Teboulle, A fast iterative shrinkage thresholding algorithm for linear inverse problems, SIAM J. Imaging Sciences, 2, 183-202, (2009).

[23] Gill, P.E. and Robinson, D.P, A primal-dual augmented Lagrangian. Comput Optim Appl 51, 1âĂŞ25, (2012).

[24] Stephen Becker, JÃľrÃťme Bobin, and Emmanuel J. CandÃÍs, NESTA: A Fast and Accurate First-Order Method for Sparse Recovery, SIAM J. Imaging Sci. 4(1), 1-39, (2011)

[25] Vidya L., Vivekanand V., Shyam Kumar U., Deepak Mishra, RBF-network based sparse signal recovery algorithm for compressed sensing reconstruction, Elsevier Neural Network, 63, 66-78, (2015).

[26] G. H. Mohimani, M. B. Zadeh and C. Jutten, A fast approach for overcomplete sparse decomposition based on smoothed L0 norm, IEEE Trans. Signal Process. 57, 289-301, (2009).

[27] Vivekanand V. and Vidya L., Compressed sensing recovery based on polynomial approximated l_0 minimization of signal and error: XEL0, in Proc. IEEE SPCOM Conf. Bangalore, India, (2014).

[28] I. Daubechies, R. DeVore, M. Fornasier, et.al. Iterative re-weighted least squares minimization for sparse recovery, Comm. Pure. Appl. Math. 63, 1-38, (2010).

[29] S. B. Chen, D. L. Donoho, M. A. Saunders, Atomic decomposition by basis pursuit, SIAM J Sci. Comput. 20, 33-61, (1998).

[30] R. Tibshirani, Regression shrinkage and selection via the lasso, J. Royal. Statist. Soc. B. 58, 267-288, (1996).

[31] M. S. Asif and J. Romberg, Fast and accurate algorithm for reweighted ℓ_1 norm minimization, IEEE Trans. Signal Process. 61, 5905-5916, (2012).

[32] Vila, Jeremy and Schniter, Philip, Expectation-Maximization Gaussian-Mixture Approximate Message Passing, IEEE Trans. Signal Processing, 61, 4658-4672, (2013).

[33] Geethu Joseph, Chandra Murthy, A Non-iterative Online Bayesian Algorithm for the Recovery of Temporally Correlated Sparse Vectors, IEEE Trans. Signal Process. 65, 5510-5525, (2017).

[34] J. Fan, R. Li, Variable selection via nonconcave penalized likelihood and its oracle properties, J. Amer. Statistical Assoc. 96, 1348-1360, (2001) .

[35] A. Mehranian, H. Saligheh Rad, M. R. Ay, A. Rahmim and H. Zaidi, Smoothly clipped absolute deviation (SCAD) regularization for compressed sensing MRI using an augmented Lagrangian scheme, IEEE NSS/MIC CA, USA , 3646-3653, (2012).

[36] M. M. Mohammadi, A. Koochakzadeh, M.B. Zadeh, M. Jansson and C. R. Rojas Successive concave sparsity approximation for compressed sensing, IEEE Trans. Signal Process. 64, 5657-5671, (2016).

[37] F. Ghayem, M. Sadeghi, M. Babaie-Zadeh, S. Chatterjee, M. Skoglund and C. Jutten, Sparse signal recovery using iterative proximal projection, IEEE Trans. Signal Process. 66, 879-894, (2018).

[38] E. A. Castro and Y. C. Eldar, Noise Folding in Compressed Sensing, IEEE Sig. Process. Lett. 18, 478-481, (2011).

[39] M. A. Herman and T. Strohmer, General Deviants: An Analysis of Perturbations in Compressed Sensing, IEEE J. Sel. Top. Signal Process. (2010).

[40] Z. Zhang, Yong Xu, J. Yang, Xuelong Li and D. Zhang, A survey of sparse representation: algorithms and applications , IEEE Access, (2015).

[41] J. D. Blanchard, M. Cermak, D. Hanle and Y. Jing, Greedy Algorithms for Joint Sparse Recovery, IEEE Trans. Signal Process. 62 (2014) 1694-1704.

[42] D.L. Donoho and M. Elad. On the stability of basis pursuit in the presence of noise , Signal Process. 86 (2006) 511-532.

[43] G.M. James, P. Radchenko, and J. Lv, DASSO: connection between Dantzig selector and lasso, J. Royal. Statist. Soc. B. 71 (2009) 127-142.

[44] Y. Zhao, X.Liao, X. He, R.Tang and W. Deng, Smoothing inertial neurodynamic approach for sparse signal reconstruction via Lp-norm minimization, Elsevier Neural Networks, 140 (2021) 100-112.

[45] Z.Zhou and J.You, A new nonconvex sparse recovery method for compressing sensing, Frontiers in App. Math. and Statistics, (2019)

[46] Vivekanand V. and Deepak Mishra, Performance measure for sparse recovery algorithms in compressed sensing perspective, Springer Int Journal of Data Science and Analytics (2022).

[47] A. Kaur, P. Kumar, An energy-efficient data sensing technique using compressive sensing for IoT-based systems, Futuristic Trends in Networks and Computing Technologies, Springer LNEE, vol. 936, (2022) 339-348.

[48] Vivekanand V. and Deepak Mishra, Feasibility of using AM3358 beagle board for networked realtime signal acquisition, Elsevier Internet of Things, (2020)

[49] M. Amarlingam, P. K. Mishra, P. Rajalakshmi, S. S. Channappayya, C.S. Sastry, Novel light weight compressed data aggregation using sparse measurements for IoT networks, Elsevier J. Network and Computer App. 121 (2018) 119-134.

[50] Y. Niu, T. Yang, F. Yang, X. Feng, P. Zhang and W. Li, Harmonic analysis in distributed power system based on IoT and dynamic compressed sensing, Elsevier Energy Reports 8 (2022) 2363-2375.

[51] M. Melek and A. Khattab, Joint sparse recovery in precision agriculture WSN and IoT applications, IEEE 7th World Forum on IoT (2021) 506-511.

[52] J. Jin, Y. Gu, S. Mei, A stochastic gradient approach on compressive sensing signal reconstruction based on adaptive filtering framework, IEEE J. Sel. Top. Signal Process. 4 (2010) 409-420.

[53] Z. Shen, H.Yang and S.Zhang, Neural network approximation: Three hidden layers are enough, Elsevier Neural Networks 141 (2021) 160-173.

[54] G. Wang, D.Yu and P.Zhou, Neural network interpolation operators optimized by Lagrange polynomial, Elsevier Neural Networks 153 (2022) 179-191

[55] J. W. Siegel, J. Xu, Approximation rates for neural networks with general activation functions, Elsevier Neural Networks 128 (2020) 313-321.

[56] M. Saari, A. M. bin Baharudin and S. Hyrynsalmi, Survey of prototyping solutions utilizing Raspberry Pi, in Proc. IEEE Conf. MIPRO, Opatija, Croatia, (2017) 991-994.

[57] A. Elrayes, M. H. Ali, A. Zakaria, M. H. Ismail, Smart airport foreign object debris detection rover using LiDAR technology, Internet of Things, 5 (2019) 1-11.

[58] D. Bruneo, S. Distefano, M. Giacobbe, A. L. Minnolo, et al, An IoT service ecosystem for Smart Cities: The #SmartME project, Internet of Things, 5 (2019) 12-33

[59] S. Cleva, L. Pivetta, P. Sigalotti, BeagleBone for embedded control system applications, in Proc. ICALEPCS, San Francisco, USA, (2013).

[60] G. Lencse and S Repas Method for benchmarking single board computers for building a mini supercomputer for simulation of telecommunication systems, in Pro. IEEE TSP Conf. Prague, Czech Republic, (2015).

[61] F. Figueroa, R.R. Selmic, D. Jethwa, Intelligent Actuator Control in On-Board ISHM Concept , in Proc. IEEE Networking Sensing and Control Conf. (2007).

[62] D. Jethwa, R. Selmic, F. Figueroa, Real-time implementation of intelligent actuator control with a transducer health monitoring capability, in Proc. IEEE Control and Automation Conf. (2008).

[63] M. Sridharan, R. Devi, C.S. Dharshini, M. Bhavadarani, IoT based performance monitoring and control in counter flow double pipe heat exchanger, Internet of Things, 5 (2019) 34-40.

[64] O. O. Olakanmi, M. S. Benyeogor, Internet based tele-autonomous vehicle system with beyond line-of-sight capability for remote sensing and monitoring, Internet of Things, 5 (2019) 97-115.

[65] F. Li Lian, J.R. Moyne, and D. M. Tilbury, Performance Evaluation of Control Networks: Ethernet, ControlNet, and DeviceNet, IEEE Control Systems Magazine (2001).

[66] R. Birkett, Enhancing Real-time Capabilities with the PRU, TI application note Sitara Proc, (2014).

[67] A. Bhargava and M. G. Hluchyj, Frame loss due to buffer overflows in fast packet networks, in Proc. IEEE INFOCOM , San Francisco, CA, USA, 1 (1990) 132-139.

[68] A. Ramadhan, M. A. Murti and L. V. Yovita, Implementation of networked control systems using programmable controller based Ethernet network, in Proc. IEEE Conf. ICoICT, Bandung, Indonesia, (2013) 22-27.

[69] J. Chen, S. Meng and J. Sun, Stability Analysis of Networked Control Systems With Aperiodic Sampling and Time-Varying Delay, in IEEE Trans. Cybernetics 47 (2017) 2312-2320.

[70] F. Kheirandish, M. Ekramian and M. Ataei, Stability analysis of networked control systems with time varying delays, in Proc. ICEE, Tehran, Iran, (2017) 633-638.

[71] H. Zhi-qing and Z. Yan-xin, Time-delay compensation for Networked Control Systems based on threshold autoregressive modeling, in Proc. Int. Conf. Test Measurement., Hong Kong,China, (2009) 402-408.

[72] J. Robert, J. P. Georges, E. Rondeau and T. Divoux, Minimum Cycle Time Analysis of Ethernet Based Real-Time Protocols, Int. Report, Universite de Lorraine, CNRS, CRAN, UMR, Cedex, France.

[73] S. Morosi, M. Dolfi, E. Del Re, E. Masi, I. Colzi, S. Mancuso, F. Francini, R. Magliacani, A. Valgimigli and L. Masini, A WSN for ground-level ozone monitoring based on plant electrical activity analysis, in Proc. IWCMC Conf. Dubrovnik, Croatia, 715-720, (2015)

[74] J.K.T. Caballa, G.D. Abrajano, M.L.C. Guico, C.S.Pineda, N.J.C.Libatique and G.L.Tangonan, Nationwide 5GHz-fixed wireless network for prototype rain alarm system, in Proc. ISSNIP Conf. Singapore, 1-5, (2015)

[75] C. E. Perkins, E. M. Royer, Ad-hoc on-demand distance vector routing, in IEEE Workshop. Mobile Comp. Sys. App. WMCSA, 90-100, (1999)

[76] M. K. Marina, S. R. Das, On-demand multipath distance vector routing in ad hoc networks, in IEEE Int. Conf. Net. Proto. 14-23, (2001)

[77] S. Li, S. Zhao, X. Wang, K. Zhang and L. Li, Adaptive and secure load balancing routing protocol for service oriented wireless sensor networks, IEEE Syst. J. 8, 858-867, (2014)

[78] W.B. Heinzelman, A.P. Chandrakasan and H. Balakrishnan, An application-specific protocol architecture for wireless microsensor networks, in IEEE Trans. Wireless Commun. 1, 660-670, (2002)

[79] Li Qing, Qingxin Zhu, Mingwen Wang, Design of a distributed energy efficient clustering algorithm for heterogeneous wireless sensor networks, Computer Communications, 29 2230-2237, (2006)

[80] Y. Hu, Y. Ding, and K. Hao, An immune cooperative particle swarm optimization algorithm for fault-tolerant routing optimization in heterogeneous wireless sensor networks, J. Math. Problems Eng., 19, (2011)

[81] M.M. Islam, M.A. Matin and T.K. Mondol, Extended stable election protocol (SEP) for three-level hierarchical clustered heterogeneous WSN, in Porc. IET Conf., London, UK, (2012)

[82] Y. Yao, Q. Cao and A. V. Vasilakos, EDAL: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for heterogeneous wireless sensor networks, IEEE/ACM Trans. Netw. 23, 810-823, (2015)

[83] L. T. Tan, H. Y. Kong and V. N. Q. Bao, Projected Barzilai-Borwein methods applied to distributed compressive spectrum sensing, in Proc. DySPAN Conf. (2010).

[84] N. Omidvar, D. H. K. Tsang, M. R. Pakravan and V. K. N. Lau, Efficient energy-aware routing with redundancy elimination', IEEE J. Sel. Areas Commun. 33, 2815-2824, (2015)

[85] L. T. Tan and L. B. Le, Joint data compression and MAC protocol design for smartgrids with renewable energy, Wirel. Commun. Mob. Comput. 16, 2590 - 2604, (2016).

[86] C. Zygowski and A. Jaekel, Optimal path planning strategies for monitoring coverage holes in Wireless Sensor Networks, Elsevier Ad Hoc Networks, 96, 101990, (2020)

[87] M.Z. Hasan, H. Al Rizzo and F. Al Turjman, A survey on multipath routing protocols for QoS assurances in real-time wireless multimedia sensor networks, IEEE Commun. Surveys Tuts. 19, 1424-1456, (2017)

[88] K. Sha , J. Gehlot and R. Greve, Multipath routing techniques in wireless sensor networks: a survey, Springer Wireless Pers. Commun. 70, 807-829, (2013)

[89] N. Todtenberg and R. Kraemer, A survey on Bluetooth multi-hop networks, Elsevier Ad Hoc Networks , 93, 101922, (2019)

[90] M.P. Uwase, M. Bezunartea, J. Tiberghien, J. M. Dricot and K. Steenhaut, Experimental comparison of radio duty cycling protocols for wireless sensor networks, IEEE Sensors J. 17, 6474-6482, (2017)

[91] H.H. Liu, J.J. Su and C.Fu Chou, On energy efficient straight line routing protocol for wireless sensor networks, IEEE System J. 11, 2374-2382, (2017)

[92] Q.Wang and J.Jiang, Comparative examination on architecture and protocol of industrial wireless sensor network standards, IEEE Commun. Surveys Tuts. 18, 2197-2219, (2016)

[93] L. T. Tan and L. B. Le, General Analytical Framework for Cooperative Sensing and Access Trade off Optimization, in (2014)

[94] F. A. Aoudia, M. Gautier, M. Magno, O. Berder and L. Benini, A generic framework for modelling MAC protocols in wireless sensor networks, IEEE/ACM Trans. Netw. 25, 1489-1500, (2017)

[95] The Institute of Electrical Engineers of Japan, IEEJ Report No 772.

[96] L. A. Escobar and W. Q. Meeker, A review of accelerated test models', Inst. Math. Stat. Statistical Science, 21, 552-577, (2006)

[97] Vivekanand V. and Deepak Mishra, Expendable and Distributed Measurement Scheme for Acquisition of Naturally Sparse Events, Springer Wireless Personal Communications (2022). https://doi.org/10.1007/s11277-022-10151-z

[98] A. Raizera, W. Valente Jr., V. L. Coelhoc, Development of a new methodology for measurements of earth resistance, touch and step voltages within urban substations, Electric Power Sys. Res. J. 153, 111-118, (2017)

[99] L. T. Tan and H. Y. Kong, A novel and efficient mixed-signal compressed sensing for wide-band cognitive radio, in Proc. IFOST Conf. (2010).

[100] Vivekanand V. and Deepak Mishra, Framework for Segmented Threshold ℓ_0 Approximation Based Computing Network for Sparse Signal Recovery, Elsevier Neural Networks (2023).

[101] Vivekanand V. and Thushara V. T., Ultra resource constrained adaptive multipath routing for meteorological sensor networks, in Proc. ANTS Conf. Bangalore, India, (2016).

[102] Deebak B.D. and F. Al. Turjmanb, A hybrid secure routing and monitoring mechanism in IoT-based wireless sensor networks, Elsevier Ad Hoc Networks, 97, 102022, (2020)

[103] F. Qin. X.Dai and J.E.Mitchell, Effective SNR estimation for wireless sensor network using Kalman filter, Elsevier Ad Hoc Networks, 11, 944-958, (2013)