

IMPROVED DIRECTION OF ARRIVAL
ESTIMATION FOR ULTRA-WIDEBAND
CHANNELS

BY

AWADALLA SALAHELDIN AWADALLA SALIH

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

Electrical Engineering

January 2017

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **Awadalla SalahEldin Awadalla Salih** under the direction of his thesis advisor and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfilment of the requirements for the degree of **MASTER OF SCIENCE** in **ELECTRICAL ENGINEERING**.

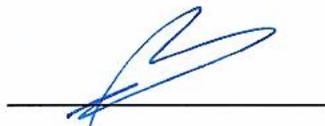
Thesis Committee



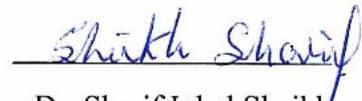
Dr. Ali Ahmad Al-Shaikhi
Department Chairman



Dr. Wajih Abu-Al-Saud
Advisor



Dr. Salam Adel Zummo
Dean of Graduate Studies



Dr. Sharif Iqbal Sheikh
Member

5/6/17

Date



Dr. Samir Al-Ghadban
Member

I would like to dedicate my work to my loving parents.

ACKNOWLEDGEMENTS

In the name of Allah, the Most Beneficent, the Most Merciful

All praise be to Allah (The One and The Only Creator of everything) for His limitless blessings. May Allah bestow peace and His choicest blessings on His last prophet, Muhammad (Peace Be Upon Him), his family (May Allah be pleased with them), his companions (May Allah be pleased with them) and his followers.

I would like to express my profound gratitude and appreciation to my thesis committee chairman and adviser, Dr. Wajih Abu-Al-Saud, whose expertise, understanding, patience, and guidance, added considerably to my graduate experience. I appreciate his vast knowledge and skill in many areas, including but not limited to, communication systems, signal processing and above all mathematics. He is the one professor who truly made a difference in my life. It was under his tutelage that I developed a focus and became interested in research. He provided me with direction, intellectual support and became more of a mentor than a professor. It was through his persistence, understanding, kindness and overwhelming support that I completed my degree.

I would like to thank the other members of my committee Dr. Sharif Iqbal Sheikh and Dr. Samir Al-Ghadban for their assistance and for taking out time from their busy schedule to serve on my thesis committee.

I would also like to thank my family for the support they provided me throughout my entire life and in particular, I must acknowledge my father and my mother, without their prayers, love and encouragement, I would not have accomplished anything worthwhile in my life.

I would also like to thank all my friends at KFUPM for the support and venting of frustration during the period of my stay at KFUPM, which helped enrich the experience. I express special gratitude to those who helped me for their guidance and exchange of knowledge, ideas and skills.

In addition, I acknowledge that this research would not have been possible without the support and assistance provided by the of King Fahd University of Petroleum and Minerals and its caring staff, and to mention especially professors and faculty members of the Electrical Engineering Department.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iv
Table of Contents	vi
List of Algorithms/Tables.....	ix
List of Figures	x
List of Abbreviations.....	xi
Thesis Abstract	xiii
ملخص الرسالة.....	xv
Chapter One: Introduction.....	1
1.1 Preface	1
1.2 Motivation	2
1.2.1 Ultra-Wideband Channels	2
1.2.2 Compressive Sensing.....	3
1.2.3 Direction Finding.....	4
1.3 Problem Statement.....	5
1.4 Summary	6
1.5 Layout.....	6
Chapter Two: Literature Review	7
2.1 Introduction	7
2.2 Ultra-Wideband (UWB) Channel Model	8
2.2.1 Channel Characterization	8
2.2.2 Saleh-Valenzuela (S-V) Channel Model	10
2.2.3 Signal Model	12
2.2.4 Sparsity of the UWB Channel	14
2.3 Compressive Sensing.....	16

2.3.1	Definition.....	16
2.3.2	Sparse Signal Recovery.....	18
2.3.3	Matching Pursuit (MP) Algorithm.....	19
2.3.4	Compressive Sampling Matching Pursuit (CoSaMP) Algorithm.....	22
2.4	Direction Finding.....	23
2.4.1	Direction of Arrival Estimation Algorithms.....	23
2.4.2	Mathematical Formulation.....	25
2.4.3	Information Theoretic Detection.....	32
2.4.4	Review of CS based DF methods.....	34
2.5	K-Means Clustering Algorithm.....	35
2.6	Summary.....	37
Chapter Three: Methodology.....		39
3.1	Proposed Method Description.....	39
3.1.1	Channel Impulse Response.....	39
3.1.2	Compressive Sensing.....	40
3.1.3	Direction of Arrival Estimation.....	41
3.1.4	Overall System Description.....	42
3.2	UWB Channel Model.....	46
3.2.1	Intel-based Model.....	46
3.2.2	Ray Tracing Model.....	48
3.3	Compressive Sensing.....	52
3.3.1	Compression and Expansion.....	52
3.3.2	Minimum Square Error Comparison of CS Algorithms.....	53
3.4	Direction Finding.....	54
3.4.1	Algorithms Implementation.....	54
3.4.2	Angle-Delay Domain Clustering.....	56
3.4.3	DOA Estimation Metrics.....	57

Chapter Four: Results	59
4.1 Review	59
4.2 UWB Channel Generation	61
4.2.1 Ray-Tracing Model	61
4.2.2 Intel-based Model	64
4.2.3 CIR Models Comparison	66
4.3 Compression and Expansion of CIRs	68
4.4 Direction Finding Simulation	72
4.5 Angle of Arrival Estimation Method	73
4.6 Method Test Scenarios	77
4.7 Comparison to Conventional DF	80
Chapter Five: Conclusion	83
5.1 Conclusions	83
5.2 Future Work and Recommendations	84
References	86
Appendix : Code Listing	92
Vitae	103

LIST OF ALOGRITHMS/TABLES

Algorithm 2.1: Orthogonal Matching Pursuit.....	21
Algorithm 2.2: Compressive Sampling Orthogonal Matching Pursuit.....	23
Algorithm 2.3: <i>k</i> -Means Clustering.....	36
Table 4.1: Intel-based Channel Model Parameters.....	65

LIST OF FIGURES

Figure 2.1: Low-energy density and high-energy density wireless systems [9].....	9
Figure 2.2: An illustration of a channel impulse response (CIR).	10
Figure 2.3: S-V channel model [11].	12
Figure 2.4: UWB Gaussian-like Pulse.....	13
Figure 2.5: (a) UWB CIR, (b) Pulse propagates through channel [13].	14
Figure 2.6: Effect of UWB channel indoor propagation in residential environment (LOS) [13]..	16
Figure 2.7: CS mathematical framework [14].	17
Figure 2.8: Different methods for DOA estimation.	24
Figure 2.9: 4-element ULA receiving signal from a single emitter at direction θ	26
Figure 2.10: Information detection criteria's comparison (AIC and MDL) [27].	33
Figure 2.11: K-Means Algorithm flowchart and pseudo code.	36
Figure 3.1: 1×2 DOA Estimation for MPCs arriving symmetrically at receiver element.	42
Figure 3.2: Summarized block diagram of the proposed method.....	43
Figure 3.3: Typical Ray Tracing Methodology.....	48
Figure 3.4: Example of Ray Tracing Model Operation.....	50
Figure 3.5: Sample PDPs and CIRs obtained from different channel models.....	51
Figure 3.6: CS compression and Expansion for CIR realization.....	53
Figure 3.7: MUSIC algorithm flowchart.....	55
Figure 4.1: Block diagram of the procedure.....	59
Figure 4.2: Ray-Tracing Model operation snapshot.....	62
Figure 4.3: CIR obtained from Ray-Tracing simulation with the aforementioned parameters.....	63
Figure 4.4: Angle of arrivals of different MPCs.....	64
Figure 4.5: A sample CIR generated by Intel UWB Model simulation.....	66
Figure 4.6: CIR Models used and their outputs.....	67
Figure 4.7: Reconstruction of Sparse CIR after compression using MP Algorithms.....	69
Figure 4.8: MSE-based noise performance comparison for CS algorithms.....	70
Figure 4.9: MSE-based compression ratio comparison for CS algorithms.	71
Figure 4.10: MUSIC spectrum simulation.....	73
Figure 4.11: Different ray-arrival patterns.....	74
Figure 4.12: Angle-Delay Domain clustering for DOA estimates.....	76
Figure 4.13: Line of Sight (LOS) scenario in outdoor setup.....	78
Figure 4.14: Non-Line of Sight (NLOS) scenario in indoor setup.....	79
Figure 4.15: Angle-Delay Domain Clustering DF Method	81
Figure 4.16: Conventional DF Method (MUSIC Spectrum)	82

LIST OF ABBREVIATIONS

ADC	Analog-to-Digital Conversion
UWB	Ultra-Wideband
FCC	Federal Communication Commission
CS	Compressive Sensing (Sampling)
DF	Direction Finding
DOA	Direction of Arrival
AOA	Angle of Arrival
IEEE	Institute of Electrical and Electronics Engineers
SNR	Signal-to-Noise Ratio
CIR	Channel Impulse Response
LOS	Line of Sight
NLOS	Non-Line of Sight
PSD	Power spectral Density
WLAN	Wireless Local Area Network
GPS	Global Positioning System
RF	Radio Frequency
IC	Integrated Circuit
SOC	System on Chip
FIR	Finite Impulse Response
S-V	Saleh-Valenzuela
IR-UWB	Impulse-Radio Ultra-Wideband
CM _x	Channel Mode x
RIP	Restricted Isometry Property
LASSO	Least Absolute Shrinkage and Selection Operator
MP	Matching Pursuit
OMP	Orthogonal Matching Pursuit
StOMP	Stagewise Orthogonal Matching Pursuit
ROMP	Regularized Orthogonal Matching Pursuit
CoSaMP	Compressive Sampling Matching Pursuit
ML	Maximum Likelihood

MUSIC	MUltiple Signal Classification
ESPRIT	Estimation of Signal Parameters via Rotational Invariance Techniques
ULA	Uniform Linear Array
SVD	Singular Value Decomposition
EVD	Eigen Value Decomposition
LS	Least-Square
TLS	Total Least-Square
AIC	Akaike Information Criterion
MDL	Minimum Description Length
Wi-Fi	Wireless Fidelity
SIMO	Single Input Single Output
LMS	Least Mean Square
RLS	Recursive least squares
MPC	Multipath Component
ESV	Extended Saleh-Valenzuela
DOD	Direction of Departure
TX	Transmitter
RX	Receiver
MSE	Minimum Square Error
RMSE	Root Minimum Square Error
PDP	Power Delay Profile
MVDR	Minimum variance distortion-less response
LCMV	Linear constraint minimum variance
GEESE	GEneralized Eigenvalues utilizing Signal subspace Eigenvectors

THESIS ABSTRACT

NAME: Awadalla SalahEIDin Awadalla Salih

TITLE OF STUDY: Improved Direction of Arrival Estimation for
Ultra-Wideband Channels

MAJOR FIELD: Electrical Engineering

DATE OF DEGREE: January 2017

By combining the techniques of Compressive Sensing (CS) and utilizing an optimized version of a current DOA estimation algorithms, a framework of a new method for DF applicable for UWB channels is proposed. Computer Simulations are carried out in this research to verify the system operation.

The first part of this research focuses on the application of CS extraction techniques to obtain a high-resolution Channel Impulse Response (CIR) of an UWB SIMO transmission environment. Expansion is achieved using Compressive Sampling Matching Pursuit (CoSaMP) algorithm that utilizes a linear solution for the under-fitted data model to find an estimation of the compressed data samples. CS framework enables the DOA estimator to work on a sample-by-sample basis while relaxing the need for a high sample rate ADCs.

A new DF method is deduced by combing conventional algebraic algorithm for DF (ESPRIT) and clustering the space of DOA estimates found in the Angle-Delay domain. Clustering is achieved using a modified K-means algorithm to include the incoming rays' respective powers per index delay. ESPRIT algorithm is deployed in a sample-by-sample over the antenna array samples by using the extracted CIRs vectors instead of the received signal strengths.

Various functional blocks of the new DF method are described and their simulation results under UWB assumptions are stated and discussed. This method provides an additional amount of details over the conventional DF methods in characterizing UWB channels by describing DOA of different rays and their respective time of arrivals (delays). A comparison of performance of this method with the performance of the conventional MUSIC DOA algorithm is conducted.

ملخص الرسالة

الاسم الكامل: عوض الله صلاح الدين عوض الله صالح

عنوان الرسالة: طريقة محسنة لايجاد الأتجاه للقنوات فائقة الاتساع

التخصص: هندسة كهربائية

تاريخ الدرجة العلمية: يناير 2017

عن طريق الجمع بين تقنيات التحسس المضغوط (Compressive Sensing) و استخدام نسخ محسنة من خوارزميات تقدير زوايا الوصول (Direction of Arrival)، تقدم هذه الرسالة طريقة جديدة يمكن تطبيقها على قنوات النطاق العريض جدا (Ultra-Wideband). تقدم الرسالة نتائج المحاكاة الحاسوبية للطريقة المقترحة للتحقق من كيفية عمل هذا النظام.

الجزء الأول من هذا البحث يركز على استعمال تقنيات استخلاص (CS) للحصول على الاستجابة النبضية للقناة بدقة عالية للقنوات ذات النطاق العريض جدا (UWB) للأنظمة وحيدة المداخل و متعددة المخارج (SIMO). تتم عملية التوسيع باستعمال خوارزمية (CoSaMP) التي تستعمل حلول خطية باستخدام بيانات منقوصة للحصول على تقدير البيانات المضغوطة. التحسس المضغوط يمكن نظام تقدير زوايا الوصول للعمل على العينات المفردة و يخفف الحاجة الى محولات اشارات متصلة الى رقمية (ADC) ذات سرعة عالية.

في هذه الرسالة تعرض طريقة جديدة لتحديد الاتجاهات عن طريق الدمج بين الخوارزميات الجبرية التقليدية لتحديد الاتجاهات (ESPRIT) و تجميع تقديرات زوايا الوصول (DOA) في نطاق الزوايا-التأخير. التجميع يتم باستخدام خوارزمية محسنة من خوارزمية المتوسطات (K-Means) لتضمين طاقة الأشعة الواصلة لكل معامل تأخير.

تمت في هذه الرسالة مناقشة و استعراض نتائج مختلف المراحل لتنفيذ هذه الطريقة بالمحاكاة الحاسوبية تحت افتراض قنوات (UWB). الطريقة المقترحة تقدم معلومات اضافية مقارنة بالطرق التقليدية لتحديد الاتجاه (Direction Finding) في توصيف قنوات (UWB) بتحديد زوايا الوصول (DOA) لمختلف الأشعة و أوقات الوصول (Time of Arrival) المقابلة. أجريت مقارنة بين أداء هذه الطريقة و أداء الطرق التقليدية مثل خوارزمية تقدير زوايا الوصول (MUSIC).

CHAPTER 1

INTRODUCTION

1.1 Preface

Wireless communication is the transfer of information without the use of wires or cables, allowing the system the freedom to be mobile. Wireless systems have developed significantly in the past years and played an extremely important role in our civilization; connecting people via tele-services. The request for communications among people is growing rapidly; in response, more connectivity, more services, and higher quality are the base requirements.

The design prospective of the physical layer of such systems, is to develop schemes and methodologies that increase the information rate and improve the efficiency of a communication system under the strict conditions of the wireless environment.

Short-range wireless technology plays a key role in scenarios where everybody and everything is connected by different types of communication links. While most of human to human information exchanges are still by voice, a rapid increase in data transfers is observed in other types of links as expressed by the rising need for location-aware applications and video transfer capability within the home and office environments.

1.2 Motivation

1.2.1 Ultra-Wideband Channels

Ultra-Wideband (UWB) communication technology became a powerful technology for wireless communication systems. It acts in low-power, short-range wireless networks. UWB technology is helping people to be free from wires. It enables wireless connection between multiple devices for transmission of video, audio and bandwidth demanding communication technologies.

UWB technology started in the 1960s, when it was mostly a radar and military technology [1]. UWB technology has presence today in wide and diverse areas, with applications in high and low-data-rate communications for short-distance and long-distance transmission. Such applications include (but not limited to): Wireless (sensor) networks, Medical Computerized Imaging systems and Vehicular radar systems.

The new trend is to allow UWB waveforms to coexist with other narrowband systems (the US FCC approved such coexistence); which resulted in a great growth in attention given by academic/industry/standardization organizations to UWB technology. In 2002, the United States FCC allowed unrestricted utilization of the band (from 3.1 – 10.6 GHz), which forms a large portion of spectrum.

With current advancements in switching circuitry, more attention towards UWB technology has followed. These advancements derived a spread for the technology from military to consumer sectors.

The transmission of data through UWB channels results in a number of looked-for characteristics. The key fact in UWB communication is the use of low energy transmitted

signals that occupy a large bandwidth (though, a desired feature “frequency fading immunity” is guaranteed). The rich multipath diversity nature is presented by the large number of propagation paths existing in a UWB channel.

UWB transceivers are usually used to carry information in form of radio impulses, hence the name Impulse Radio Ultra Wideband (IR-UWB). The transmission of short-time radio impulses enables the receivers to extract channel information for further processing. Channel Impulse Responses (CIR) are assumed to have enough information to characterize the channel between the transmitter and the receiver.

Compressive Sensing techniques allow compressing the data in the sampling stage (which is known to greatly reduce the number of digital samples required to reconstruct certain *sparse* signals). For that reason, CS is useful where analog-to-digital conversion (ADC) faces a physical design limitation, and hence the usefulness in UWB systems.

1.2.2 Compressive Sensing

Signal acquisition is a major area in signal processing. The well-established Sampling theorems are responsible of easing the transition from continuous to discrete-time worlds. The most popular theorem in data acquisition and digitalization of information is credited to Shannon [3], and it states that the rate in which the signals are sampled should be “at least” twice the maximum frequency component present in a continuous-time signal in order to perfectly represent the signal in the discrete domain.

However, the authors in [4] laid out the foundation of Compressive Sampling, also known as Compressed Sensing (CS). The assumptions made are to ease the reproduction of signals

by using far fewer samples of measurements than conventional methods of sampling (with some constraints).

CS is potentially applicable in areas where the reconstruction of signals from a limited set of linear measurements is possible. Application of CS includes wide range of consumer, medical and military applications, such as Data Acquisition and Compression, Analog-to-Information Conversion and Channel Coding, Computerized Tomography (both optical and wave-induced) and Target Detection or Radar Positioning.

1.2.3 Direction Finding

Many methods are developed to enhance wireless communication systems, among them, are techniques to provide devices with the sense of location and orientation. Localization algorithms are well established and documented in many researches.

In smart antenna technology, a Direction Finding (DF) algorithm is usually incorporated to develop systems that provide accurate location information for wireless services [5]. The existing work in this area is mainly intended to conclude with methods to improve the precision and reduce the complexity of the algorithms for Direction of Arrival/Angle of Arrival (DOA/AOA) estimation.

The increase of demand for the wireless technology services have spread into many areas. All of the upcoming usage scenarios can be considered as reasons for determining the direction of arrival of incoming signals in wireless systems, such as smart mobile antenna and sensor networks, environmental monitoring and public security, seismology and search and rescue and strategy and defense operations.

1.3 Problem Statement

With the wide range of advantages, the UWB channel is to be chosen for the reasons stated earlier [Section 1.2.1]. The system is designed to assume a channel that supports transmission of high data rates. Common systems have a physical layer more or less described by the Industrial standard IEEE 802.15.4a-2007 [6].

This research makes use of the emerging techniques used to ease processing signals which have relatively large bandwidth (hence require high sampling rates “according to Nyquist Sampling Theorem”) to reduce the hardware requirements and relax communication system design constraints. The technique called the Matching Pursuit (MP) Algorithm [7] “and its modified variants” is implemented to achieve compressive sampling.

The global aim of this work is to develop a CS-friendly modified DOA estimation technique using extracted channel impulse responses to better work with UWB channels. The proposed DF method provides higher accuracy of DOA estimation in situation where the conventional DF methods would fall short. The results would discuss the profit of this technique over the other conventional methods available in the literature. The key aspects of this project are:

- Formulation and realization of a modified DF technique for UWB channels utilizing CS framework.
- Presenting a verification of the technique using two different channel models.
- Reduction in algorithms complexity by processing receiver extracted propagation information (channel impulse responses) rather than considering incoming signals.
- Improving estimation accuracy in low SNR environments of operation.

1.4 Summary

The framework of CS provides robust tools for reducing the number of measurement required to summarize sparse signals. For that reason, CS is useful in systems where the analog-to-digital conversion process is critical, for example UWB systems.

The improvement of DOA estimation techniques, led by the emerging CS method, would make the problem of DF less demanding. Although, the assumption that justifies the use of CS tools is directly related to the level of sparsity of the CIRs being considered.

1.5 Layout

The layout of this thesis is as following:

- **Chapter 2** presents a literature review of the three main themes of the research (UWB channels, Compressive Sensing and Direction Finding). The related mathematical and statistical frameworks, assumptions and equations are presented.
- **Chapter 3** describes the scope of the work to be carried out to justify the claimed conclusions, how the simulation is done and the environment setup are also stated.
- **Chapter 4** shows number of results and graphs produced during the research and a discussion of the findings.
- **Chapter 5** states the conclusions and possible future research opportunities related to this work.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

In this chapter, a survey about the existing foundation for the topic in hand is presented. The main aspects in relation to this work are Ultra-Wideband (UWB) channel characteristics and modelling, Compressive Sensing (CS) algorithms, Direction Finding (DF) methodologies and Data Clustering and Analysis in the Angle-Delay domain.

The proposed method is a DF technique for UWB system that utilizes CS framework and data clustering algorithms to provide more insight of the DOAs of multiple signals and their paths. CS provides a handful of tools to deal with high data rate applications and thus relaxing the need for high sampling rate Analog-to-Digital Convertors (ADC) at the receiver, while enables finding high resolution measurements.

DOA estimation is done through algebraic algorithms that are currently found in the literature (namely ESPRIT and Root-MUSIC). Although, the difference here is the proposed method considers high resolution time-domain channel impulse responses (CIRs) that are obtained by processing the received signal via a CS algorithm (namely CoSaMP), which will be discussed later.

Clustering a group of data samples in a multi-dimensional space provides a collective answer for the probable DOAs in the Angle-Delay Domain.

2.2 Ultra-Wideband (UWB) Channel Model

2.2.1 Channel Characterization

The FCC defines UWB systems as having 500 MHz bandwidth or more. The UWB bandwidth is restricted by 10 dB below the peak power of co-existing narrowband systems [8]. The fractional bandwidth (B_f) is defined by $B_f = 2 \frac{f_H - f_L}{f_H + f_L}$, where f_L and f_H define the lower and upper bounds of the signal spectrum, respectively. The value of $B_f \leq 0.2$ would specify an UWB channel transmission.

UWB Systems have several favorable characteristics over other communication systems, such as:

- Wireless transmission at high throughput: large bandwidth (several GHz), can provide more than 500 Mb/s data rate (low-range “10 m”).
- High precision ranging: Fine pulses duration (sub-nanosecond), fine time-domain resolution, which offers sub-centimeter accuracy for transmitter localization applications.
- Reasonable path loss: propagating waves penetrate obstacles (in line-of-sight (LOS) and non-line-of-sight (NLOS) communications).
- Robust to fading: immunity to multipath fading, ability to resolve multipath components, RAKE receiver advantage.
- Security: very low power spectral density (PSD) (power levels are below the noise floor), not easy for unauthorized detection.

- Coexistence: coexists with the bandwidth of cellular systems, wireless local area networks (WLAN), global positioning systems (GPS), and other wireless systems.
- Low cost transceiver hardware design: low power signals, integrated RF and baseband in a single SoC IC.

The main reasons for UWB to emerge as a solid contestant for future communication systems is the support for high data rates and the low power required to transmit and receive data. These features give UWB systems a coexistence feature with other currently deployed wireless systems (Such as Wireless LAN “WLAN”). Figure 2.1 illustrate some of these advantages over other high-energy, comparatively low-data rate wireless systems.

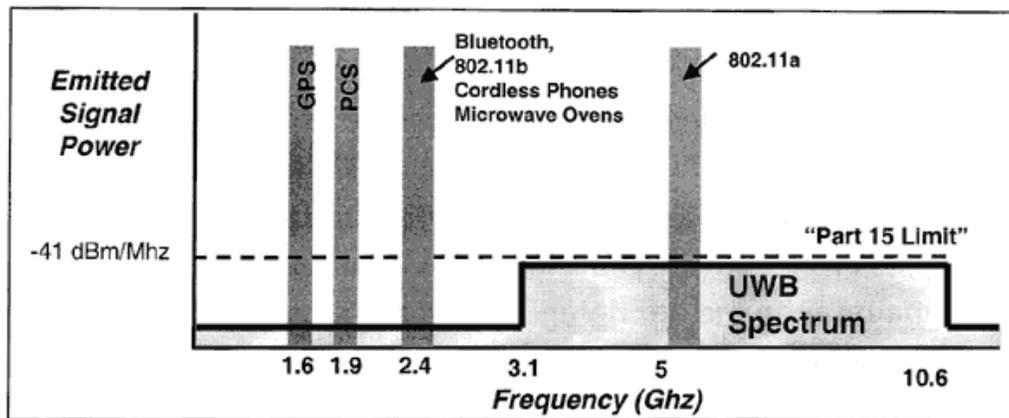


Figure 2.1: Low-energy density and high-energy density wireless systems [9].

2.2.2 Saleh-Valenzuela (S-V) Channel Model

Due to multipath fading, the IEEE 802.11b standard model is based on the statement that the impulse response is made of finite impulse response filter (FIR) taps that have sampling interval equal to the reciprocal of the ray arrival rate [6]. While the channel impulse response (CIR) of the UWB channel is based on the measurements-based finding of Saleh-Valenzuela (S-V) [10]. A Two-Poisson model is deployed for the arrival rate of clusters and rays within each cluster. Figure 2.2 illustrates the concept of clustered CIR by giving a Power-delay Profile (PDP) example.

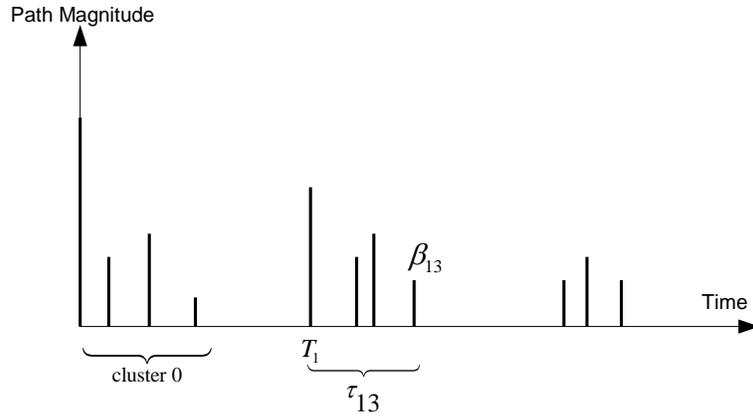


Figure 2.2: An illustration of a channel impulse response (CIR).

The terminology of the S-V channel model is as follows:

T_l = Arrival time of the first path of the l^{th} cluster,

$\tau_{k,l}$ = Delay of the k^{th} path within the l^{th} cluster relative to the first arrival time,

Γ = Power decay rate of clusters,

γ_l = Power decay rate of rays in the l^{th} cluster,

Λ = Cluster arrival rate,

λ = Ray arrival rate within each cluster,

β_{kl} = Amplitude of the k^{th} ray in the l^{th} cluster.

By definition, $\tau_{k,l} = T_l$. The PDFs of T_l and $\tau_{k,l}$ depends on the rates Λ and λ , and given by the Poisson distributions:

$$p(T_l|T_{l-1}) = \Lambda \exp[-\Lambda(T_l - T_{l-1})], \quad l > 0 \quad (2.1)$$

$$p(\tau_{k,l}|\tau_{(k-1),l}) = \lambda \exp[-\lambda(\tau_{k,l} - \tau_{(k-1),l})], \quad k > 0 \quad (2.2)$$

respectively. The amplitude of the k^{th} path within the l^{th} cluster denoted $\beta_{k,l}$ and it is

Rayleigh distributed with a mean $\overline{\beta^2}_{k,l}$ written as:

$$\overline{\beta^2}_{k,l} = \overline{\beta^2(0,0)} \exp[-T_l/\Gamma] \exp[-\tau_{k,l}/\lambda] \quad (2.3)$$

where $\overline{\beta^2(0,0)}$ is the average power of the first arrival of the first cluster (the very first ray). An illustration of the assumed model is given by Figure 2.3.

Simulations done in [6] concluded that the assumption of using S-V model for simulating the CIR of UWB channels is a practical technique. It gives high matching percentages when compared to real channel measurement data.

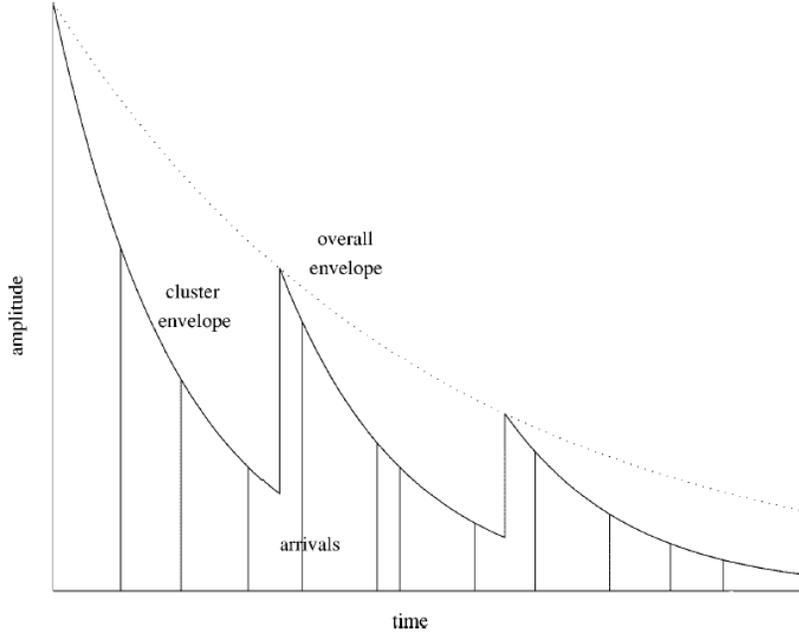


Figure 2.3: S-V channel model [11].

2.2.3 Signal Model

UWB signals transmit data by low power level signals (under the thermal noise floor) and experiences a dense multipath phenomenon. To simplify the design of UWB systems, the pulse shape is selected accordingly. That designed pulse affects severely the overall system performance.

The UWB transmitted signal is modeled as

$$s(t) = \sum_{k=0}^{\infty} \sum_{j=0}^{N-1} a_k p(t - jT_f - kT_{sym}) \quad (2.4)$$

where the data $a_k \in \pm 1$ is the k^{th} transmitted bit, T_{sym} is the symbol duration and $T_f = T_{sym}/N$ is the pulse repetition period and N is the number of symbols. The pulse $p(t)$ in

Impulse-Radio UWB (IR-UWB) is either Gaussian, monocycle or poly-cycle pulse [12].

An example of a Gaussian-like pulse is shown in Figure 2.4.

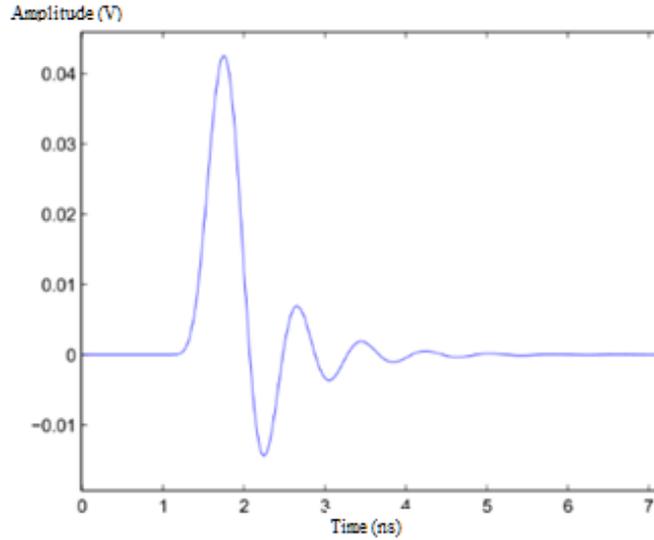


Figure 2.4: UWB Gaussian-like Pulse.

The signal $s(t)$ travels through an $(L - 1)$ -path fading channel. The channel impulse response (CIR) is:

$$h(t) = \sum_{l=0}^{L-1} \alpha_l \delta(t - \tau_l) \quad (2.5)$$

where the received pulse is assumed to exhibit different fading coefficient (α_l) from each l -path, whose delay is τ_l . We assume that $\tau_0 \leq \tau_1 \leq \dots \leq \tau_{L-1}$. Figure 2.5 shows the resulting signal power levels when a pulse propagates through a specific channel.

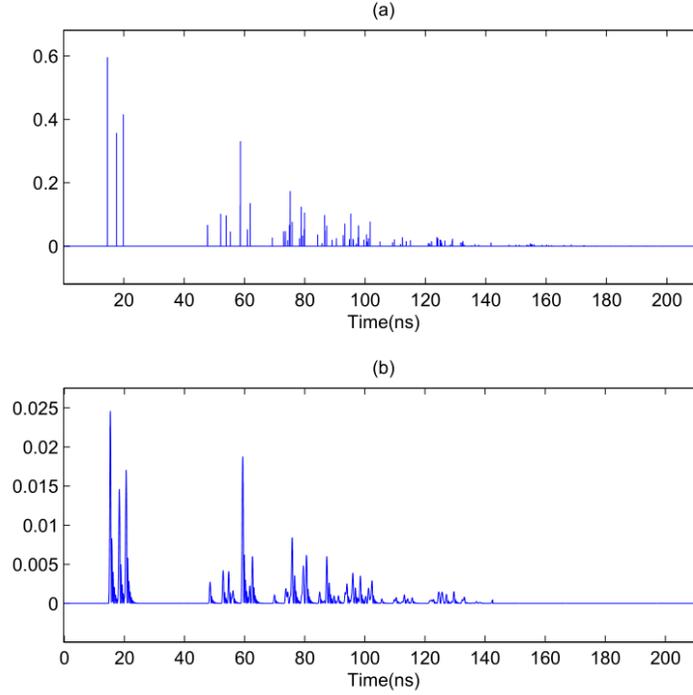


Figure 2.5: (a) UWB CIR, (b) Pulse propagates through channel [13].

The received waveform is assumed to be

$$r(t) = s(t) * h(t) = \sum_{k=0}^{\infty} \sum_{j=0}^{N-1} \sum_{l=0}^{L-1} a_k p(t - jT_f + kT_{sym} - \tau_l) + w(t) \quad (2.6)$$

where $w(t)$ is the Gaussian noise defined as the two-sided power spectral density of $N_o/2$.

2.2.4 Sparsity of the UWB Channel

Sparsity is expressed in the fact that the information rate of a continuous time signal may be much smaller than suggested by its bandwidth, or that discrete-time signal depends on a number of degrees of freedom which is comparably much smaller than its length.

- **Definition:** The signal \mathbf{x} is said to be sparse in the basis Ψ if it can be sufficiently represented using only a small number $K \ll M$ of atoms from Ψ .

$$\|\Psi\mathbf{x}\|_{l_0} \leq K \quad (2.7)$$

where $\|f\|_{l_0}$ is the zero norm (i.e.: non-zero coefficients count in the signal f).

Many signals are sparse if they are expressed in a convenient basis. The implication of sparsity is that one can discard the part of the coefficients without much perceptual loss. Thus, there is no need to spend a lot of power into capturing all the samples of a sparse signal in all coordinates when most of them are zero anyway. Such process requires not only the knowledge of some coefficients of “real-signal” but also the locations of the significant pieces of information. Fortunately, these pieces tend to be clustered. As an example of sparse-clustered model, it has been shown that many physical channels tend to be distributed as clusters within respective channel spreads [6].

A wireless channel can be assumed to be a sparse channel, where the delay spread could be very large, but the number of major paths is normally very small compared to the window length. We can start with the assumption that the short duration signal (high frequency UWB pulses) propagate through the multipath channels, the received signal continues to be sparse in some domain, and thus the use of compressed sensing is possible.

To reinforce the assumption of received signal sparsity, we consider the 8th order Butterworth pulse usually implemented in UWB systems as the data bearer having a length of $T_p = 1ns$. Figure 2.6 shows the received signal per frame for a UWB channel that models an indoor residential environment with Line-Of-Sight (LOS) scenario (according to the IEEE 802.15.4a channel model CM1 [6]) in the absence of noise.

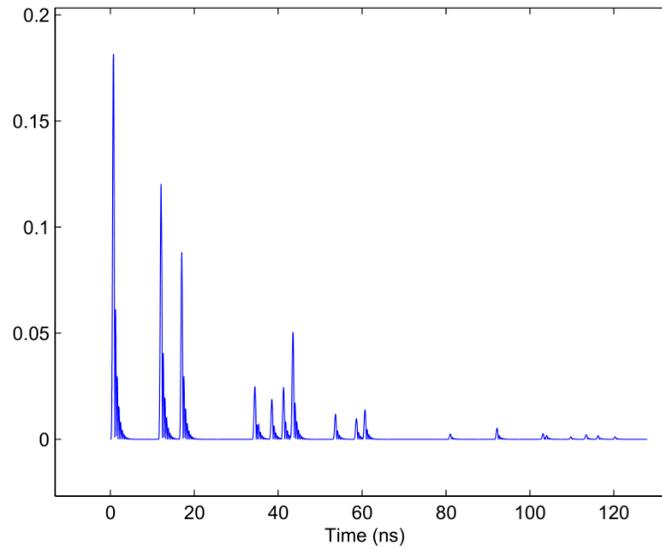


Figure 2.6: Effect of UWB channel indoor propagation in residential environment (LOS) [13].

As is depicted in Figure 2.6 from [13], the received signal is made of groups of spread out clusters of the transmitted signal. It has relatively longer inter-clusters time intervals where the amplitude of the received signal is zero or a negligible value. In this particular example, the signal plotted in Figure 2.6 has 12,672 taps, of which 9,765 (77%) have amplitude lower than 10^{-5} [13].

2.3 Compressive Sensing

2.3.1 Definition

Compressive Sensing (CS) is a modern concept founded on the theoretical results of signal recovery with random basis samples. The notable result of CS discloses that a signal, f , with a large number of samples that is M -sparse in some basis Ψ , can be precisely

reconstructed with high probability using a lower number of random projections “compressed samples” of the signal onto a random function Φ that is incoherent with Ψ .

By applying CS methodologies, the sampling rate can be minimized to sub-Nyquist rate.

Considering the linear measurements model for the signal \mathbf{y}

$$\mathbf{y} = \Phi \mathbf{x} = \Phi \Psi \theta \quad (2.8)$$

where $\Phi \Psi$ form the effective measurements matrix for estimating the K -sparse vector θ .

Matrix Φ is called the measurement kernel, and it has lower rank N than the signal “ \mathbf{x} ” rank

which is equal to M . The $N \times M$ matrix Φ is a projection plane for the signal \mathbf{x} . Figure 2.7

summarizes the mathematical relation between CS entities.

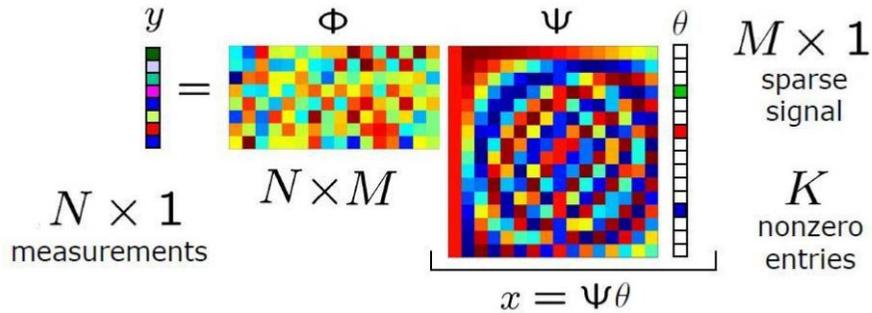


Figure 2.7: CS mathematical framework [14].

The maximum value amongst inner product of the orthonormal basis and the orthonormal measurement matrix is usually described by the coherence measure

$$\mu(\Phi, \Psi) = \max_{\substack{1 \leq k \leq N \\ 1 \leq j \leq M}} |\langle \phi_k, \psi_j \rangle| \quad (2.9)$$

The classical sampling scheme correspond to $\mu = 1$ and $\phi_k(t) = \delta(t - k)$ and Ψ is the Fourier basis $\psi_j(t) = e^{i2\pi jt/n}$. It is known that any random matrix is greatly incoherent with any fixed matrix Ψ .

2.3.2 Sparse Signal Recovery

The main issue in CS is the design of a convenient measurement matrix (to avoid information damage in dimension conversion) and designing the corresponding reconstruction algorithm to recover the sparse vector θ from only N compressed measurement samples vector y .

The most common criteria for benchmarking the quality of a CS measurement matrix is the Restricted Isometry Property (RIP) [13]. This property is summarized as follows

$$(1 - \delta_s) \|\theta\|_{l_2}^2 \leq \|\Phi\Psi\theta\|_{l_2}^2 \leq (1 + \delta_s) \|\theta\|_{l_2}^2 \quad (2.10)$$

for a positive constant δ_s . The $\|\cdot\|_{l_2}^2$ is the l_2 -norm of the vector. The RIP property should be fulfilled when choosing Φ in order to successfully recover the signal. It is known that by selecting Φ to be a random matrix RIP can be satisfied with a high probability. Gaussian, Bernoulli and partial Fourier matrices satisfies the RIP [15].

If the RIP is met, then an accurate signal reconstruction problem procedure the linear programming of the form

$$\min_{\theta^* \in \mathbb{R}^M} \|\theta^*\|_{l_1}^2 \quad \text{subject to} \quad \mathbf{y} = \Phi\Psi\theta^* \quad (2.11)$$

where $\|\theta\|_{l_1} = \sum_{i=1}^M |\theta_i|$ is the l_1 -norm of the reconstructed signal.

The RIP tells us about how much success would be in the process of sparse signal reconstruction. Nonetheless the process itself consists of solving the Euclidean norm using least norm procedures. An analytical solution exists and it is in the form of:

$$\theta^* = (\Phi\Psi)^H((\Phi\Psi)(\Phi\Psi)^H)^{-1}\mathbf{y} \quad (2.12)$$

which give poor results and the solution is almost never sparse. By using the prior information of that the solution is sparse (θ is K -sparse), the result can be improved by solving for the vector with least non-zero entries. This is named l_0 -norm:

$$\min_{\theta^* \in \mathbb{R}^M} \|\theta^*\|_{l_0}^2 \quad \text{subject to} \quad \mathbf{y} = \Phi\Psi\theta^* \quad (2.13)$$

However, an exhaustive search problem is created and feasibility is reduced. A relaxation of this problem is presented by [16] and [17]. It is shown that by Gaussian measurements it can be exactly reconstruct K -sparse vector and closely approximate compressible vectors with high probability using l_1 optimization shown in Equation 2.11.

The l_1 -minimization is a convex problem and can be solved using modern optimization techniques [18]. This problem is usually called Least Absolute Shrinkage and Selection Operator (LASSO). In the next sections, a review is provided for the various CS algorithms and their procedures.

2.3.3 Matching Pursuit (MP) Algorithm

Sparse signal reconstruction is commonly achieved by methods that can be divided into two classifications. Algorithms that solve the linear optimization by minimizing the

residual error as a function of the gain and the delay location of all dominant paths [19] are the first class. The second being the algorithms that choose sequentially the most important taps of the sampled CIR. The Greedy algorithms (such as the Matching Pursuit and its orthogonal version) are the most popular from the second class [20]. In our case, we are focusing on the second class, because we are interested in finding DOA of the most dominant paths (or taps) of the FIR model of the channel.

The Matching Pursuits (MP) algorithm tries to build up the sparse signal iteratively by selecting the atom that maximizes the representation at each iteration. Orthogonal Matching Pursuit (OMP) solves the problem by projecting the signal onto the subspace spanned by the selected atoms (with no atoms selected twice). The result is optimal in the least squares sense (require fewer steps to converge, although computationally demanding).

In this work, we will focus on OMP algorithm family, because of its simplicity of implementation and the fast computation time and the convenient results produced [21]. Algorithm 2.1 summarizes the steps required to implement the OMP algorithm required to solve the signal reconstruction problem. It produces an estimation (\hat{x}) for the sparse signal extracted from a set of linearly sampled data points.

Algorithm 2.1: Orthogonal Matching Pursuit

Inputs: CS Matrix A , measurement vector y

Initialize: $\hat{x}_0 = 0$, $r_0 = y$, $\Lambda_0 = \Phi$

for $i = 1$; $i = i + 1$ until stopping criterion is met **do**

$g_i \leftarrow A^T r_{i-1}$ {Form signal estimate from residual}

$\Lambda_i \leftarrow \Lambda_{i-1} \cup \text{supp}(H_1(g_i))$ {Add largest residual entry to support}

$\hat{x}_i|_{\Lambda_i} \leftarrow A_{\Lambda_i}^H y$, $\hat{x}_i|_{\Lambda_i^c} \leftarrow 0$ {Update signal estimate}

$r_i \leftarrow y - A\hat{x}_i$ {Update measurement residual}

end for

Output: Sparse representation \hat{x}

The algorithm takes the compressed measurements vector (the residual) as well as the compressing matrix as inputs (y and A respectively), and by solving for the highest value support and subtracting its effects (as in most greedy algorithms), the residual (r) is updated in each iteration until a stopping criteria is met.

Another variant of the Greedy MP family of algorithms is the Stagewise Orthogonal Matching Pursuit (StOMP) [22] and Regularized Orthogonal Matching Pursuit (ROMP) [23]. Each one of these algorithms has its positives and shortcomings as well. Some are extremely fast although require relatively large number of samples to produce a continent estimation error. Some are computationally expensive though operate with comparatively low number of acquired samples. The author of the Compressive Sampling Matching Pursuit (CoSaMP) proves [24] that by modifying the OMP to combine features from both categories results in a combinatorial algorithm that guarantees speed and provide lower error bounds.

2.3.4 Compressive Sampling Matching Pursuit (CoSaMP) Algorithm

The Compressive Sampling Matching Pursuit (CoSaMP) algorithm described in last section requires the same three inputs (inherited from its mother algorithm, the OMP) as well as an additional input, being the sparsity of the approximation to be produced.

After setting the initial residual to the trivial compressed signal, the algorithm repeats these five steps until a halting criterion is triggered [25]:

1. Identification: Find the residual from the current samples and locate the largest components.
2. Support Merger: Merging the newly identified components with the components that appear in the current approximation.
3. Estimation: Solving a least-squares problem to find the target signal.
4. Pruning: Produce a new solution by retaining only the largest entries in the estimation from step 3.
5. Sample Update: Subtract the effect of the current iteration approximation from the residual.

The Pseudocode for CoSaMP is listed in Algorithm 2.2. Comprehensive details about how this algorithm performs is included in [25].

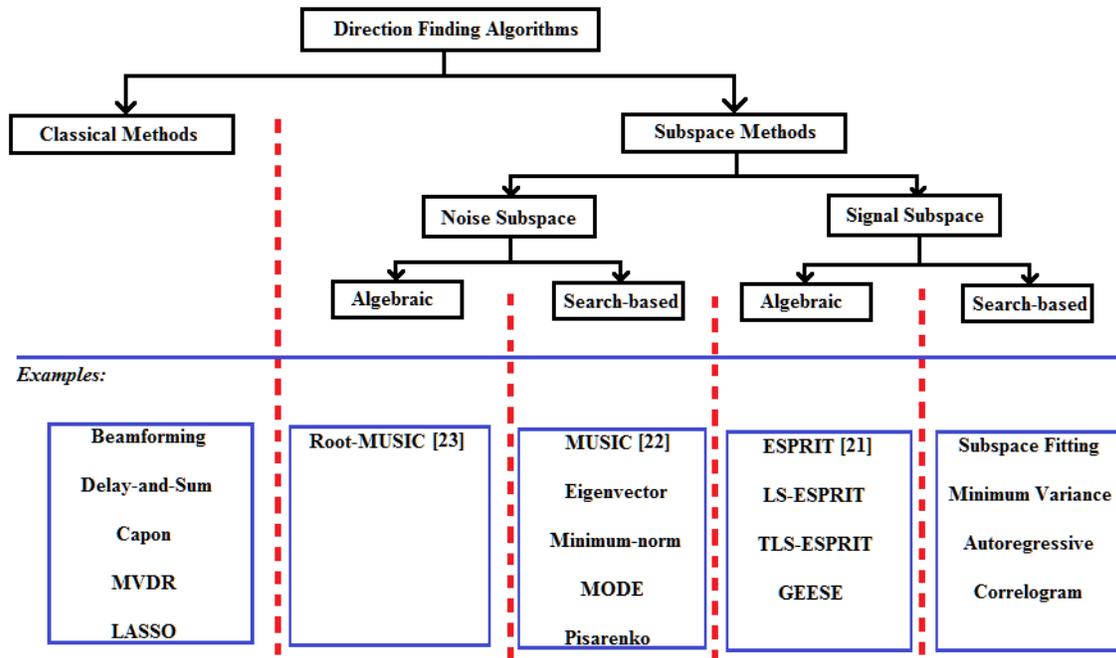


Figure 2.8: Different methods for DOA estimation.

Regarding signal subspace techniques, only the signal projection is considered, while in noise subspace based techniques, noise projection is used. Subspace approaches are categorized into search-based and algebraic-based methods. For search-based techniques, the response of the array to a single source is to be known in advance (analytically, as a function of arrival angle, or found via calibration). On the other hand, algebraic techniques don't involve a search process, and result in the DOA estimates in angles directly.

MULTiple Signal Classification (MUSIC) algorithm is a search-based noise subspace method used to determine the DOA of a narrowband source using an array of sensors [29]. A famous algorithm based on signal subspace method implemented as a DF solution is Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) [28]. In Root-MUSIC [30], we use antenna array that is assumed to be uniformly-spaced and linear,

in which the search process required after using MUSIC is exchanged by a root finding solution.

The next sections briefly review the formulation of these three algorithms (MUSIC, ESPRIT, Root-MUSIC). In section 3.4, the methodology followed to apply those three DOA estimation techniques to UWB sparse channels is presented. The aim is to benchmark the performance of our new method against those existing algorithms.

2.4.2 Mathematical Formulation

Consider a 6-element uniform linear array (ULA) receiving signal from a single emitter source at direction θ as shown in Figure 2.9.

The received signal is assumed to be:

$$\mathbf{x}(t) = \begin{bmatrix} s(t - \tau_1)e^{j\omega_c(t-\tau_1)} \\ s(t - \tau_2)e^{j\omega_c(t-\tau_2)} \\ \vdots \\ s(t - \tau_M)e^{j\omega_c(t-\tau_M)} \end{bmatrix} \quad (2.14)$$

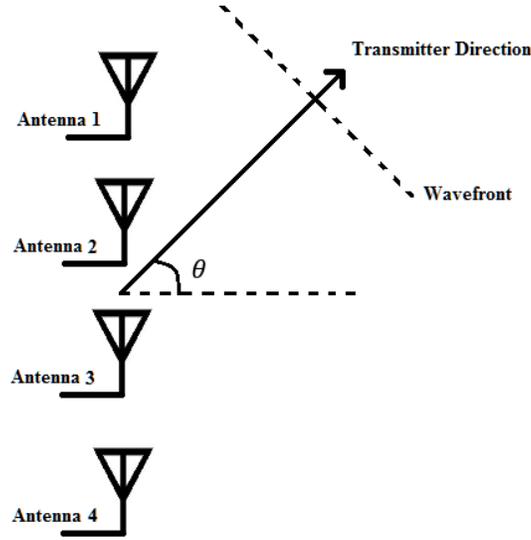


Figure 2.9: 4-element ULA receiving signal from a single emitter at direction θ .

where $\mathbf{x}(t)$ is the received RF signals vector and τ_M are the path delays resulting from relative direction compared to the receiver antenna arrays. In case of linear arrays $\tau_M = \tau_0 - (d_m/c) \sin \theta$, and τ_0 being the path delay from the array central point to the emitter, d_m are the lengths of paths from the array elements to middle point, c denoted the speed of light, and θ is the required angle to be found. The unknown random phase term $e^{-j\omega_c\tau_0}$ is equal for all elements. The baseband received signal is as follow:

$$\hat{\mathbf{x}}(t) = \begin{bmatrix} s(t - \tau_1)e^{j\omega_c\tau_1} \\ s(t - \tau_2)e^{j\omega_c\tau_2} \\ \vdots \\ s(t - \tau_M)e^{j\omega_c\tau_M} \end{bmatrix} \quad (2.15)$$

Let λ denote the wavelengths. The “array manifold” $\mathbf{a}(\theta)$ (or usually it is called the steering vector) is the vector that has a unit amplitude defined by

$$\mathbf{a}(\theta) = \begin{bmatrix} \exp\{j2\pi d_1 \sin(\theta/\lambda)\} \\ \exp\{j2\pi d_2 \sin(\theta/\lambda)\} \\ \vdots \\ \exp\{j2\pi d_M \sin(\theta/\lambda)\} \end{bmatrix} \quad (2.16)$$

Assuming that $s(t)$ is normalized (unit power value), the channel model for the received signal is formed as:

$$\mathbf{y}(t) = \sqrt{\text{SNR}} s(t) \mathbf{a}(\theta_0) + \mathbf{v}(t) \quad (2.17)$$

where $\mathbf{v}(t)$ is an i.i.d complex Gaussian noise signal that has zero mean and unit variance.

By definition θ_0 is the source direction, and SNR is the received-signal-to-noise ratio.

Direction finding algorithms are usually implemented in digital systems, so the discrete-time version of the signal model can be written as: $\mathbf{y}[k] = \sqrt{\text{SNR}} s[k] \mathbf{a}(\theta_0) + \mathbf{v}[k]$, where k is the index of the sample.

2.4.2.1 MUSIC Algorithm

The algorithm is developed assuming that there are P signals that are arriving on the array, given by the discrete-time vector form summation:

$$y[k] = \sum_{p=1}^P \alpha_p[k] \mathbf{a}(\theta_p) + \mathbf{v}[k], \quad k = 1, \dots, K \quad (2.18)$$

where θ_p are the signal directions, $\alpha_p[k]$ are uncorrelated received signal complex amplitudes, and $\mathbf{v}[k]$ is a complex vector representing Gaussian noise process with zero-

mean and unit-variance. The covariance of $\mathbf{y}[k]$ vector is found by $\mathbf{R}_y = E\{\mathbf{y}[k]\mathbf{y}[k]^H\}$, and it has the form given by

$$\mathbf{R}_y = \sum_{p=1}^P \sigma_p^2 \mathbf{a}(\theta_p) \mathbf{a}(\theta_p)^H + \mathbf{I} \quad (2.19)$$

where $\sigma_p^2 = E\{|\alpha_p[k]|^2\}$ is the p th signal SNR, \mathbf{I} is the identity matrix of size $p \times p$, and H is the Hermitian operator. Let $\mathbf{R}_y = \mathbf{U}\mathbf{\Sigma}\mathbf{U}^H$ be the singular-value decomposition (SVD) of the matrix \mathbf{R}_y , where $\mathbf{\Sigma} = \text{diag}\{[\sigma_1^2 + 1, \dots, \sigma_p^2 + 1, 1, \dots, 1]\}$ is a diagonal matrix of the singular values of \mathbf{R}_y . The matrix \mathbf{U} can be portioned as $\mathbf{U} = [\mathbf{U}_s, \mathbf{U}_n]$, where the $M \times P$ matrix \mathbf{U}_s “the signal subspace” resembles the vector corresponding to the largest P singular values, and the matrix \mathbf{U}_n “the noise subspace” forms the singular vector associated with the remaining $M - P$ smallest singular values. The two subspaces are orthogonal (\mathbf{U} is unitary), i.e.: $\mathbf{U}_s^H \mathbf{U}_n = \mathbf{0}$. It follows that $\mathbf{a}^H(\theta) \mathbf{U}_n = \mathbf{0}$ for $\theta = \theta_1, \dots, \theta_p$.

The spectrum $S(\theta) = \frac{1}{|\mathbf{a}^H(\theta) \mathbf{U}_n|^2}$ will contain peaks (very high amplitudes in the case of no noise assumption) at the signals directions $\theta_1, \dots, \theta_p$. Although, it is usual that the covariance matrix \mathbf{R}_y is not available at the receiver. Thus, estimation of the covariance matrix using the available data is required to find the sample covariance matrix $\hat{\mathbf{R}}_y = \sum_{k=1}^K \mathbf{y}[k]\mathbf{y}[k]^H$.

The SVD of the sample covariance matrix is $\hat{\mathbf{R}}_y = \hat{\mathbf{U}} \hat{\mathbf{\Sigma}} \hat{\mathbf{U}}^H$, and the corresponding noise subspace is found by portioning $\hat{\mathbf{U}}$ as in $\hat{\mathbf{U}} = [\hat{\mathbf{U}}_s, \hat{\mathbf{U}}_n]$. The spectrum $\hat{S}(\theta)$ is called the MUSIC spectrum, and is given by

$$\hat{S}(\theta) = \frac{1}{|\mathbf{a}^H(\theta) \hat{\mathbf{U}}_n|^2} \quad \text{or} \quad \hat{S}(\theta) = \frac{\mathbf{a}^H(\theta) \mathbf{a}(\theta)}{|\mathbf{a}^H(\theta) \hat{\mathbf{U}}_n|^2} \quad (2.20)$$

Orthogonality between $\mathbf{a}(\theta)$ and $\hat{\mathbf{U}}_n$ will minimize the denominator and hence will give rise to peaks in the MUSIC spectrum. By computing and plotting the MUSIC spectrum over the whole range of θ , the DOA's can be estimated by locating the peaks in the spectrum. A one-dimension search algorithm for the largest P peaks in the MUSIC spectrum is required to get the DOA's as an angle (i.e.: a numerical form).

2.4.2.2 ESPRIT Algorithm

ESPRIT algorithm is implemented by using two identical and identically oriented sub-arrays (where one of them represents a shifted replica of the other), and the displacement vector between the two subarrays is assumed to be known in advance (though, it is geometry independent). The two steering vectors for the two sub-arrays are assumed to be

$$\mathbf{a}_1(\theta) = \mathbf{J}_1 \mathbf{a}(\theta), \quad \mathbf{a}_2(\theta) = \mathbf{J}_2 \mathbf{a}(\theta) \quad (2.21)$$

where \mathbf{J}_1 and \mathbf{J}_2 are two matrices defined for selection as having in each row only one entry that is one and the remaining are set to zeros. In linear organized arrays, the steering vectors for the two sub-arrays are related as $\mathbf{a}_1(\theta) = \mathbf{a}_2(\theta) e^{j(\frac{2\pi}{\lambda})d_x \sin \theta}$, where d_x is the displacement between the two sub-arrays. Similarly, the steering matrices for the two sub-arrays are related as $\mathbf{A}_1(\theta) = \mathbf{A}_2(\theta) \mathbf{Q}(\theta)$, where

$$\mathbf{Q}(\theta) \triangleq \text{diag} \left\{ e^{j(\frac{2\pi}{\lambda})d_x \sin \theta_1}, \dots, e^{j(\frac{2\pi}{\lambda})d_x \sin \theta_L} \right\} \quad (2.22)$$

For sources with incoherent signals, the entries of the columns of the signal sub-space eigenvectors (\mathbf{E}_S) and the steering matrix ($\mathbf{A}(\theta)$) extends the same subspace. Thus $\mathbf{E}_S = \mathbf{A}(\theta) \mathbf{T}$ where \mathbf{T} is nonsingular. It can be shown that $\mathbf{E}_{S1} = \mathbf{A}_2 \mathbf{Q} \mathbf{T} = \mathbf{E}_{S2} \mathbf{T}^{-1} \mathbf{Q} \mathbf{T}$, where $\mathbf{E}_{S2} = \mathbf{J}_2 \mathbf{E}_S$. The matrices $\mathbf{\Psi} = \mathbf{T}^{-1} \mathbf{Q} \mathbf{T}$ and \mathbf{Q} would be having equal singular values. Thus, the algorithm will be operating by two steps. First, an estimate of $\mathbf{\Psi}$ is to be found. Second, the AOAs are calculated from the singular values of $\mathbf{\Psi}$.

The least-squares (LS) solution to estimate $\mathbf{\Psi}$ is $\hat{\mathbf{\Psi}} = \arg \min_{\mathbf{\Psi}} \|\hat{\mathbf{E}}_{S1} - \hat{\mathbf{E}}_{S2} \mathbf{\Psi}\|_F = \hat{\mathbf{E}}_{S2}^\dagger \hat{\mathbf{E}}_{S1}$, where $(\cdot)^\dagger$ denotes the pseudo-inverse. Although the matrices $\hat{\mathbf{E}}_{S1}$ and $\hat{\mathbf{E}}_{S2}$ may contain errors in their estimate, so a solution might be using the total least-square (TLS) approach. The TLS solves the problem

$$\min_{\mathbf{\Psi}, \mathbf{Z}_1, \mathbf{Z}_2} \|\mathbf{Z}_1, \mathbf{Z}_2\|_F \quad \text{s. t.} \quad \hat{\mathbf{E}}_{S1} + \mathbf{Z}_1 = (\hat{\mathbf{E}}_{S2} + \mathbf{Z}_2) \mathbf{\Psi} \quad (2.23)$$

which leads to $\hat{\mathbf{\Psi}} = -\mathbf{M}_{22} \mathbf{M}_{12}^{-1}$ [31], where \mathbf{M}_{12} and \mathbf{M}_{22} are $P \times P$ matrices computed from the eigenvalue decomposition of

$$\begin{pmatrix} \hat{\mathbf{E}}_{S1}^H \\ \hat{\mathbf{E}}_{S2}^H \end{pmatrix} \begin{pmatrix} \hat{\mathbf{E}}_{S1} & \hat{\mathbf{E}}_{S2} \end{pmatrix} = \begin{pmatrix} \mathbf{M}_{11} & \mathbf{M}_{12} \\ \mathbf{M}_{21} & \mathbf{M}_{22} \end{pmatrix} \tilde{\mathbf{\Lambda}} \begin{pmatrix} \mathbf{M}_{11}^H & \mathbf{M}_{21}^H \\ \mathbf{M}_{12}^H & \mathbf{M}_{22}^H \end{pmatrix} \quad (2.24)$$

and $\tilde{\mathbf{\Lambda}} = \text{diag}\{\tilde{\lambda}_1, \dots, \tilde{\lambda}_{2P}\}$ is the sorted eigenvectors. TLS-ESPRIT is shown to be better than LS-ESPRIT in fully calibrated sensor arrays.

The estimated DOAs are then calculated as

$$\theta_i = \sin^{-1}[-\tilde{\lambda}_i / 2\pi d_x] \quad , \quad i = 1, \dots, P \quad (2.25)$$

2.4.2.3 Root-MUSIC

By considering the steering vector $\mathbf{a}(\theta)$ to be written as:

$$\mathbf{a}(\theta) = \begin{bmatrix} \exp\left\{-j\frac{N-1}{2}\frac{2\pi}{\lambda}d_x \sin\theta\right\} \\ \exp\left\{-j\frac{N-3}{2}\frac{2\pi}{\lambda}d_x \sin\theta\right\} \\ \vdots \\ \exp\left\{-j\frac{N-1}{2}\frac{2\pi}{\lambda}d_x \sin\theta\right\} \end{bmatrix} = \begin{bmatrix} z^{-\frac{N-1}{2}} \\ z^{-\frac{N-3}{2}} \\ \vdots \\ z^{-\frac{N-1}{2}} \end{bmatrix} \triangleq \mathbf{a}(z) \quad (2.26)$$

where $z \triangleq \exp\{j(2\pi/\lambda)d_x \sin\theta\}$, and d_x is the spacing between the array elements. The denominator of the spectrum found by MUSIC algorithm would then be written as

$$f(\theta) = \mathbf{a}^H(\theta) \widehat{\mathbf{E}}_N \widehat{\mathbf{E}}_N^H \mathbf{a}(\theta) = \mathbf{a}^T \left(\frac{1}{z}\right) \widehat{\mathbf{E}}_N \widehat{\mathbf{E}}_N^H \mathbf{a}(z) \triangleq f(z) \quad (2.27)$$

There are $(N - 1)$ complex conjugate reciprocal roots for the polynomial $f(z)$. In noise-free case $f(z)$ would have P sets of dual roots $z_i = \exp\{j(2\pi/\lambda)d_x \sin\theta_i\}$, $i = 1, \dots, P$, and there are $2(N - P - 1)$ more “noise” roots “neglected”. In case of noisy operation, the roots positions are slightly shifted although can be estimated from the closest roots to unit circle in $f(z)$.

Thus, the Root-MUSIC algorithm tries to find all possible roots of $f(z)$ and estimates signal DOAs from the P highest-magnitude roots which correspond to the signal subspace inside the unit circle.

2.4.3 Information Theoretic Detection

Usually, the receiver side does not know how many signals to estimate (i.e.: P is unknown). The Akaike Information Criterion (AIC) [32] and Minimum Description Length (MDL) [33] are the two-common information theoretic approaches used to estimate the number of signals. They are widely used methods for estimating the size of smallest model that is represented by a given set of data.

The eigenvalues represent the noise-subspace which are assumed to be equal to σ ; as the noise is white (spectrally flat, uncorrelated by assumption). The signal-subspace eigenvalues are comparatively larger than σ . Thus, the smallest number of eigenvectors that reflects the correlation between the antenna elements is to be treated as the number of signals found in such model (set of data).

In [34] a formulation of the AIC and MDL is presented, and it has been shown that the two approaches are applicable for cases where the noise eigenvalues are roughly equal and the signal is having higher power (basically a reasonable SNR values).

The two criteria are based on Kullback-Leibler information measure, and are described as a function of the parameter m where $0 \leq m \leq N - 1$ as:

$$1. \quad AIC(M) = -2K(N - M) \ln[Q(M)] + 2M(2N - M) \quad (2.28)$$

$$2. \quad MDL(M) = -K(N - M) \ln[Q(M)] + \left(\frac{1}{2}\right)M(2N - M) \ln(K) \quad (2.29)$$

where

K = number of vectors used to find the eigenvalues,

M = number of signals that are incident on the array,

N = number of antenna array elements.

$$Q(M) = \frac{(\lambda_{M+1} \cdot \lambda_{M+2} \cdots \lambda_N)^{\frac{1}{N-M}}}{\frac{1}{N-M} (\lambda_{M+1} \cdot \lambda_{M+2} \cdots \lambda_N)} \quad (2.30)$$

The probability of error is shown as a function of the SNR in Figure 2.10 (adapted from [34]) for both AIC and MDL for comparison.

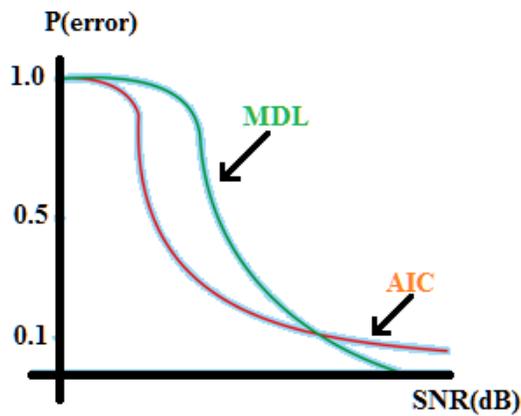


Figure 2.10: Information detection criteria's comparison (AIC and MDL) [27].

In low power applications (specially the case of UWB systems), the need is for techniques that perform better in low SNR environment. In low SNR values, the MDL criteria gives high probability when used to find M than that results from using the AIC, hence it is more favorable to be utilized. Although, in high SNR values the AIC outperforms the MDL.

2.4.4 Review of CS based DF methods

Here we present a review of some of the previous research work done regarding combining compressive sensing with DOA estimation. Most of the research is focused on acoustic signals “narrowband signals” to exploit the sparsity constrain. The goal is to reduce the SVD calculation cost and get higher resolution signals using low sampling rates. The l_1 -SVD is an algorithm developed to achieve that.

The authors of [35] present a sparse-based source localization method in which the samples of the array manifold are assumed to be sparse and signal reconstruction is made possible by utilizing an l_1 -norm based penalties. The proposed method uses subspace based DOA algorithms (Root-MUSIC and ESPRIT). Their method uses an l_1 -SVD calculation technique to summarize the projection of parallel streams. Their scheme is applicable for both narrowband and wideband systems through a resolution refinement method and a regularized parameter λ for the l_1 -SVD algorithm.

Another l_1 -SVD implementation for acoustic source localization appears in [36]. The method used is for acoustic signal localization with passive antenna arrays, which is a non-parametric technique. Super resolution signals are obtained using non-quadratic regularization penalty function to the sparse audio signals. Assumptions made are for both correlated or uncorrelated, narrowband or wideband channels. Beamforming, Capon and MUSIC algorithms are proven to increase resolution in the given assumptions.

An angle domain sparse bearing localization method is developed in [37]. Again, acoustic signals which are assumed to be modeled as a sparse vector in the angle space are processed

using an l_1 -norm optimization problem solution. Here, a quantized compressed microphone data is combined (sparse signal reconstruction) with angle domain analysis to localize the transmitter.

As [31] concluded, the nonconventional methods for DOA estimation utilizing the two algorithms of MUSIC and ESPRIT are not the best-fit for estimating the DOAs in sparse channels, because only small number of the significant channel paths are resolvable, i.e. captured with high time-domain resolution by the receiver antennas configuration (inter-element spacing and array arrangement). The need for other DF techniques for UWB systems is because such CIRs have a very large number of multipath components that should be resolved efficiently. Therefore, these methods can be utilized in combination with the framework of compressive sensing. That is true when the realization of real multipath channels is to only be considered “effectively” sparse.

2.5 K-Means Clustering Algorithm

k -means clustering (Lloyd's algorithm [38]) is an iterative algorithm used for data-partitioning. It assigns n observation points to one of k clusters and gives their centroids. The number of centroids is assumed to be known before the algorithm begins to group the points into clusters.

It is a commonly known simple statistical algorithm that works by alternating between two steps until convergence. It starts with randomly assigning points to k clusters. The assignment step (in which each sample point is assigned to the cluster with the closest

mean), and the update step (in which it calculates the new means and sets them as centroids). It reaches convergence when no sample point is reassigned in the previous step.

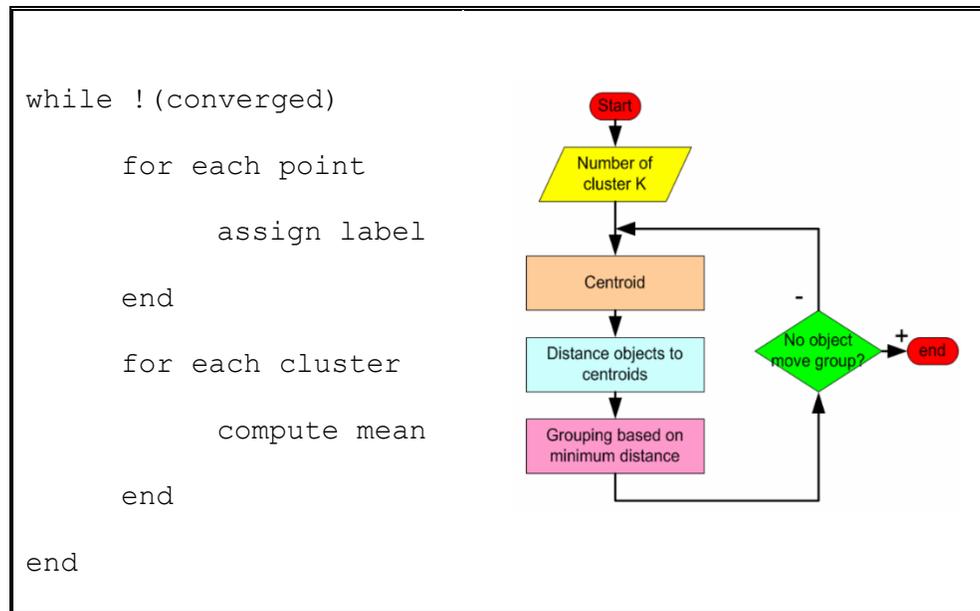


Figure 2.11: K-Means Algorithm flowchart and pseudo code.

K-Means treats each observation in the data set as an object having a location in a two-dimensional space. The algorithm is implemented efficiently to work with both online and offline clustering and data analysis routines. It makes use of data vectorization and matrix multiplication between sparse matrices.

Algorithm 2.3: *k*-Means Clustering

Inputs: 2D sample Data X , number of clusters k , maximum number of iterations t

Initialize: k Centroids, Labels

repeat

 Compute point-to-cluster distances of all observations to each centroid

 Assign *label* to each observation with the closest centroid

 Update *Centroids* by computing the average in each *lable*

Until (no re-assignments are made) OR ($i = t$)

Output: Labels

Another alternative to achieve the goal of data clustering is the use of a member of the Expectation-Maximization family of algorithms. Their implementation depends on finding a maximum likelihood of a posteriori estimate. The application of such methods is rather complex for the use of clustering for direction of arrival estimation. Although it may introduce enhancements that requires investigation.

2.6 Summary

Ultra-Wideband systems have a great role in enhancing communication links to utilize higher data rates. The problem of minimizing multiple sources interference is achieved by beamforming, and hence the need for precise, yet practical, direction finding approaches. The testing of such systems can be done using computer simulation, in which a Saleh-Valenzuela and Ray-Tracing models can be setup for sparse UWB channel realization.

In Compressive sensing, the number of samples acquired is much smaller than the number of samples required to fully represent a signal. This leads to a relaxed sampling rate constraints and to a minimized use of ADCs resources. Signal reconstruction from the compressed version is found by using a solution of a simplex convex optimization problem; in many cases with fast iterative algorithms that can be done effectively. As it was said before, the sparsest solution can be obtained with l_0 -norm optimization. Unfortunately, this optimization problem is impractical under computational constraints. The problem is relaxed using l_1 -norm.

One of the Matching Pursuit family of algorithms (a modified variant of the orthogonal one, CoSaMP) is selected to be used for sparse signal recovery. The role of CS is to show

when the sensing procedure is done in a non-redundant fashion for the received signal, the sampling rates may be reduced by factors where the information contained in the signal remains intact.

Conventional method for Direction Finding are well established. The emerging demand of source localization solutions in UWB has yet to be studied in subsequent work. Combining DOA estimation procedures with some statistical methods for sparse UWB systems is subject of this research.

This research gives attention to the design of the various blocks that enables the DF system to function, and verifies the system operation via computer simulation. This report sheds the light on these topics and highlights point where further investigation is recommended.

CHAPTER THREE

METHODOLOGY

3.1 Proposed Method Description

3.1.1 Channel Impulse Response

The assumed system is proposed to operate in UWB communication systems. Impulse Radio UWB (IR-UWB) is a special case where short-interval low-power impulses are transmitted rather than a modulated RF signal. The resulting signal (as well as the channel impulse response) is assumed to be sparse in nature (Section 2.2.4). The channel affecting UWB-IR transmission is considered to have sparse coefficients, so as the modulated-UWB transmission.

The receiver for such system is required to have an impractically high sampling rate that is a subject of hardware and design cost limitations. Other UWB systems, such as the 802.11 Wi-Fi protocol, assumes a bandwidth in terms of tens of megahertz. According to Nyquist sampling theory, the receiver should at least sample the received signal at twice the largest frequency component (when dealing with incoming amplitude signals, rather than channel impulse responses), which gives rise to the use of compressive sensing in dealing with such systems to relax the high sampling constraints (further though reducing power consumed by the samplers of portable wireless systems).

With the aid of compressive sensing framework, the signal processing aspects of UWB signals are becoming increasingly feasible. The ability of the receiver to acquire “sparse in nature” signals and deal with “sparse” channels makes the possibility high for enhancing current signal processing techniques for both narrowband and wideband communication systems.

Channel Impulse Responses (CIRs) are to be synthesized using one of the two methods described in Chapter 3. The model produced by the UWB measurements done by Intel is utilized as the first source of CIR snapshots. The second model is derived from a ray-tracing methodology. Each model has its set of parameters and settings; and both to be considered as a simulation setup for system operation.

3.1.2 Compressive Sensing

Here we define two complementing processes, the compression (i.e.: simulating the inability of the receiver to sample at high sampling rate) and the expansion (i.e.: signal reconstruction from sparse linear measurements). Based on the assumption that the signal is in much higher bandwidth (typical UWB scenarios), the receiver samples at lower sampling rate, and given the assumption that the channel is sparse, reconstruction algorithms can be used to get a higher resolution channel impulse response for later DOA estimation subsystem.

A method is developed to solve the problem of CIR reconstruction from its compressed samples; the CoSaMP algorithm (with Gaussian sampling matrix). The choice of using CoSaMP is based on the conventional compressive sensing approach discussed in Section

3.3.2. The aim is to get a finer resolution signal (and hence a finer resolution CIR) that can resolve DOAs within fine time resolution.

Although, the resolution of the CIR obtained would imply the success of such methods. Special signals “pilots” are usually transmitted within the data signal to excite the specific portions of spectrum of the channel; and by reproducing the same pilot signal at the receiver, the problem of channel identification becomes an easier task. The ability to grasp a clearer information about the CIR is subject to the capability of the designed pilot signal to excite the channel bandwidth (i.e.: the bandwidth of the transmitted signal is sufficiently wide, which is commonly the case for UWB systems).

3.1.3 Direction of Arrival Estimation

Methods of enhancing the performance of direction finding techniques are varying between simplifying algorithms workflows, increasing accuracy of detection for multiple adjacent signals in low-SNR environments, reducing calculations and orders of operations and relaxing implementation and system design complexities.

Most of the conventional algorithms for DOA estimation utilize subspace based projection analysis, although current research is emerging for using the channel impulse response (CIR) acquired by the receiver for DOA estimation and receiver localization. In such systems, the receiver could use compressive sensing tools to get a finer resolution of the CIR. Linear channel identification to obtain the CIR is currently deployed via adaptive algorithms [37].

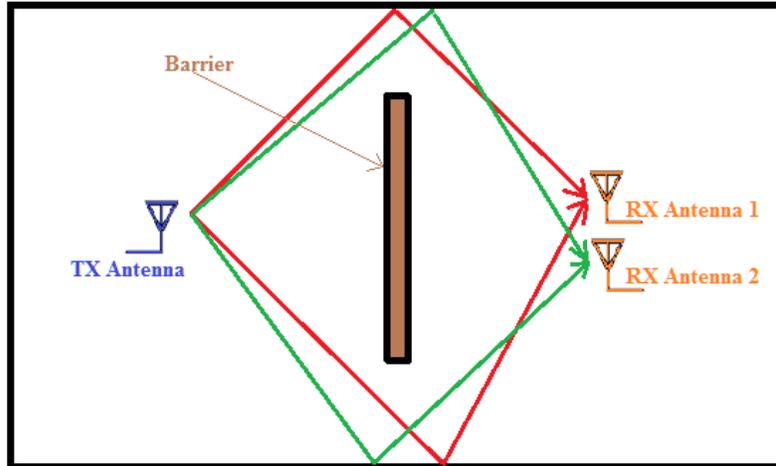


Figure 3.1: 1×2 DOA Estimation for MPCs arriving symmetrically at receiver element.

When deploying conventional DF techniques; in some scenarios (illustrated in Figure 3.1 for the case of 1×2 SIMO channel) the estimated DOA angle might be a combinatorial (from multiple symmetric paths that overlap in time domain) which results in relatively inaccurate estimation. By looking at the multipath components (MPCs) arriving at each antenna, the correct DOA angle is estimated based on the CIR of each receiver element. This justifies the need to consider a high resolution of amplitudes and phases of the arrived MPCs rather than using a signal (or noise) subspace projection technique.

3.1.4 Overall System Description

The block diagram in Figure 3.2 summarizes the proposed method for DOA estimation under UWB channel conditions. A transmitter consisting of a conventional quadrature modulation scheme is deployed. A Sparse UWB multipath channel realization is generated

using the concept of Section 3.2, and is used to “corrupt” the signal. Additive white Gaussian noise is added so the system SNR performance could be studied.

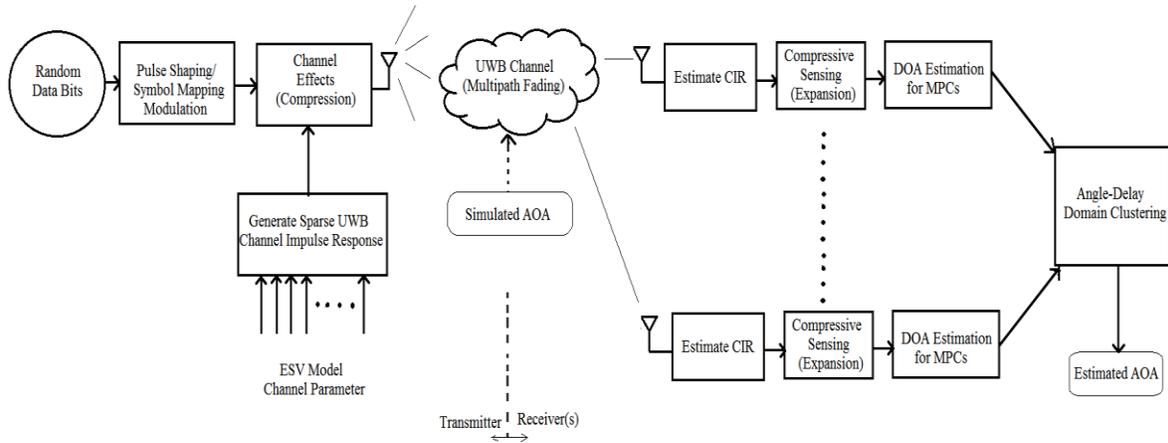


Figure 3.02: Summarized block diagram of the proposed method.

The proposed system contains an antenna array of m uniformly spaced receiving elements is capturing the signal passed through the channel. Compression is the method to simulate the receiver incapable samplers “low rate samplers”. The extraction of the channel impulse response is an essential step to base DOA estimate on the angular domain “rather than using the received signal vector”. It is implemented via conventional channel identification utilizing adaptive algorithms (i.e.: LMS or RLS). The Expansion is the process of ‘zooming-in’ the CIR to resolve MPCs that are lumped together due to the lack of high sampling rate ADCs.

The DOA estimation is done in the angular domain (delay bins versus estimated individual DOAs) by associating MPC with their respective ‘images’ through the m receiving element. A one-dimensional search procedure is deployed to match incoming MPC

received by different antennas. Using these MPC associations' along with the relative ray arrival delays, a distinct DOA can be estimated as per delay-bin basis.

As a DOA detection technique, the factors that affects system performance vary between SNR levels, receiver element inter-spacing distance, estimation errors result from detection algorithm used, channel parameters and deployment case. Due to the presence of noise in real life scenarios and CIR extraction errors results from reconstruction algorithms, direction estimations will become noisy and estimates tend to scatter around the actual DOA of each incoming ray.

A majority classification method is used to get a collective final DOA estimate per arrived ray as the main goal of this method. Clustering DOA estimation in the Angle-Delay domain [39] is a common method for reporting probable angles and the degree of confidence for the estimated directions for later purposes (i.e.: transmit beamforming). Statistical method devised from the K-means Algorithm are deployed for such goals.

In the following sections, a detailed procedure for each step is presented and illustrated. The first part of the channel modelling is composed of the channel models used to synthesize UWB channels and behaviors. The methods stated in Section 3.2.1 are carried out to achieve a channel impulse response (CIR) that corresponds to the Intel suggested model.

The second part of the channel modelling is the procedure to simulate a ray-tracing model of an assumed premise that is composed of a transmitter element, a multiple-antenna receiver element and a barrier object. Channel models extracted from this method are set

in UWB scenarios. Both models are used as a starting point to get channel impulse responses that the later system will work on as inputs.

After defining a channel model, a justification of utilizing compressive sensing (CS) in the proposed DOA estimation is presented. It is assumed that the receiver is unable to get a fine resolution of the channel state parameters (amplitudes and phases) in presence of a highly sampled (wideband) transmitted signal that exceeds its sampling frequency; rather to mention that there will be 2 or more antenna/sampling elements that are required to devise the direction of the incoming signal (which gives rise to receiver's cost and power/portability design concerns).

Estimating DOA from CIRs is carried out in a sample-by-sample basis (rather than considering a window of samples of the received signal as done in conventional DF methods). The samples are assumed to have a sufficient time-domain resolution (number of samples per delay bin) that allows for accurately resolving DOAs for distinct MPCs. CS expansion is the step accounted for producing high resolutions.

Numerical DOA estimation techniques (fast algebraic methods, Figure 2.8) act an important role in the overall system operation and efficiency. Variants of ESPRIT and Root-MUSIC algorithms (Section 2.4.2.2 and 2.4.2.3) are selected as the main algorithms for the purpose of estimating angles of individual samples because of their simple implementation and reasonable computation efficiency. Although, modified version (such as Unitary-ESPRIT) can be considered to replace them.

A post-processing step for the individual samples' DOA estimates is developed to accomplish a collective decision that can be used as a beamforming reference. A modified

clustering method (a statistical technique described in Section 2.5) based on K-Means algorithm is established to form a final DOA estimate.

Due to the broad number of techniques that are utilized to design this method, the number of parameters to adjust are fairly large (highly flexible). Investigation is carried out for some key aspects and their performance limits are discussed in the Chapter 4.

3.2 UWB Channel Model

3.2.1 Intel-based Model

Based on the discussion presented on Section 2.2.2, an extended model is proposed to simulate the UWB channel effect with high degree of matching real channels measured in [6]. The channel model is based on the S-V model with the additional extension discussed below. Due to the fact that UWB channels have rich multipath propagation effects, the received UWB signal is composed of multiple echoes of the transmitted signal, which suffer individually from different path attenuations and delays.

The used model is the one suggested by the IEEE 802.15.3a and 4a working groups for UWB communication channels. In narrowband communication channels, Rayleigh random processes are deployed to simulate the fading coefficients, but in UWB channels fading is simulated as log-normal random process [6]. An Extended Saleh-Valenzuela (ESV) channel model is used for simulating channel impulse response and obtaining their power delay profile (PDP).

With reference to the notation listed in Section 2.2.2, the deployed channel model coefficients are defined as $\alpha_{k,l} = p_{k,l}\beta_{k,l}$, where $p_{k,l}$ is an equiprobable ± 1 . And $\beta_{k,l}$ is modeled as a log-normal fading term, as $20 \log_{10} \beta_{k,l} \propto \text{Normal}(\mu_{k,l}, \sigma^2)$, or $|\beta_{k,l}| = 10^{n/20}$ where $n \propto \text{Normal}(\mu_l, \sigma^2)$, and $E[\beta_{k,l}] = \Omega_0 e^{-T_l/\Gamma} e^{-\tau_{k,l}/\gamma}$, where T_l is the excess delay of the l^{th} tap and Ω_0 is the mean power of the first path of the first cluster, and the ray arrival time μ_l is given by:

$$\mu_l = \frac{10 \ln(\Omega_0) - 10 (T_l/\Gamma) - 10 (\tau_{k,l}/\gamma)}{\ln(10)} - \frac{\sigma^2 \ln(10)}{20} \quad (3.1)$$

The model used to generate an UWB CIR is verified to fit measurements in both LOS and NLOS scenarios. The fitting is done via matching the values of the mean excess delay, RMS delay, and mean path number from the measurements and the simulation [6].

There are four channel sub-models presented by the standard set by the IEEE in the 802.15.3a work-group to fulfill measurement results, namely channel modes (CM) [1, 2, 3 and 4], for different channel characteristics. They differ in line-of-sight (LOS) availability, delay and the transmitter-receiver separation range [40].

Figure 3.4 – “Intel-Tracing Model” shows a snapshot of the realization of a channel impulse response (CIR) as well as its respective power delay profile generated by the simulation of Intel model. The amplitude/power are normalized with respect to the maximum arrived ray voltage/gain-coefficient at the RX.

3.2.2 Ray Tracing Model:

Ray-Tracing is a system to calculate paths of electromagnetic waves and model their propagation effects throughout an assumed premise. The appropriate system should include the phenomena of line-of-sight (LOS) loss “free space propagation”, reflection, diffraction and diffusion.

The graph on Figure 3.3 illustrates the idea behind ray-tracing. Although the two phenomena in question are the LOS attenuation, and the reflection of radio waves when they hit objects and their direction of departure (DOD) changes as seen by the receiving element compared to the line of sight DOD.

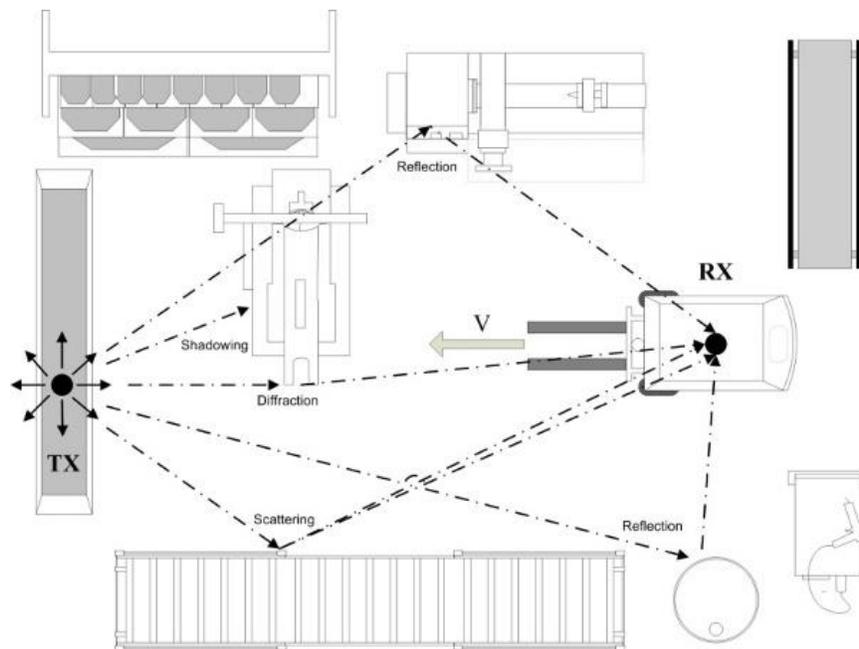


Figure 3.3: Typical Ray Tracing Methodology.

The set of assumptions that are based on propagation theory, and are made to achieve a close-to-real room CIR realization for UWB channel is detailed. The input parameters considered are: the room geometry (size and/or dimension), barrier location, TX-RX separation, detection bin size (time-domain resolution), receiver antenna inter-element spacing, frequency of operation and utilized bandwidth, and reflection coefficients of walls/barrier/partition that depends on their materials. The transmitter is assumed to be an omni-directional antenna which radiates in 360 degrees' circle (2D propagation is considered in our model). Regarding the propagation environment, it is assumed to be homogenous; i.e., there are no medium discontinuities (the wave is traveling through air). Reflections only occur when the electromagnetic signal hits either a partition face, or the room walls.

Figure 3.4 shows the operation of a ray-tracing model in a room environment. The more the objects to be simulated in the premises (chairs, tables, ...etc); the more the complexity level of tracing, and hence the closer the model results to actual scenarios. Although the focus is on getting a sample CIR that has multiple reflections and spread over a reasonable time scale (in terms of hundreds of nanoseconds).

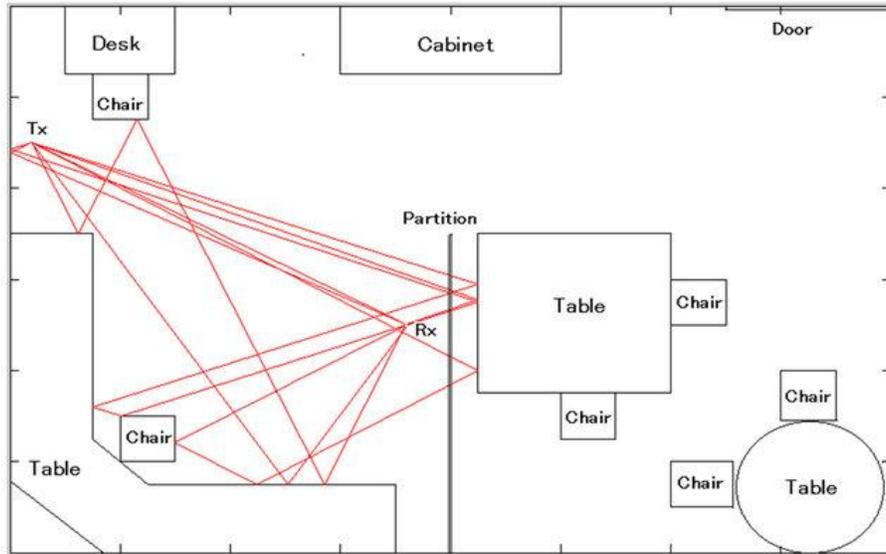


Figure 3.4: Example of Ray Tracing Model Operation.

When the ray hits one of the receiver antenna (RX) elements (i.e.: sensed), it's magnitude and total path length (which is later translated to its total time of travel "delay") are recorded in the corresponding delay bin (the CIR is assumed to be as a discrete-time vector quantity, each amplitude/phase sample correspond to a delay bin). The DOA of that individual ray is saved for later comparisons.

The fact that two or more rays may arrive at the same bin has been considered, and a separate DOA is noted for each incoming ray. The rays arrive at the same delay bin are added in terms of power and averaged in terms of DOAs. The highest power ray will dominate the DOA estimation at that delay bin, because the estimates will be weighted by their respective powers.

Outputs of the Ray-Tracing model are CIRs (by the number of the receiver antenna elements), DOAs of each ray, path lengths of different rays and paths delays. To compute

the CIR per antenna element using the computed received amplitudes, phases of each sample are obtained from path delays and incorporated in a complex-valued noise-free CIR. A simulation of a Gaussian channel noise is added to the noise-free samples to result in a CIR of the current scenario.

By varying the location of TX and RX as well as the partition properties, the channel realization (like the one sampled in Figure 3.5 – “Ray-Tracing Model”) obtained is consequently changed, thus achieving random simulated snaps of different CIRs. By controlling the parameters above and the presence (geometry) of the barrier, availability of a LOS component is a matter of chance (random geometric re-location of the transmitter and the receiver).

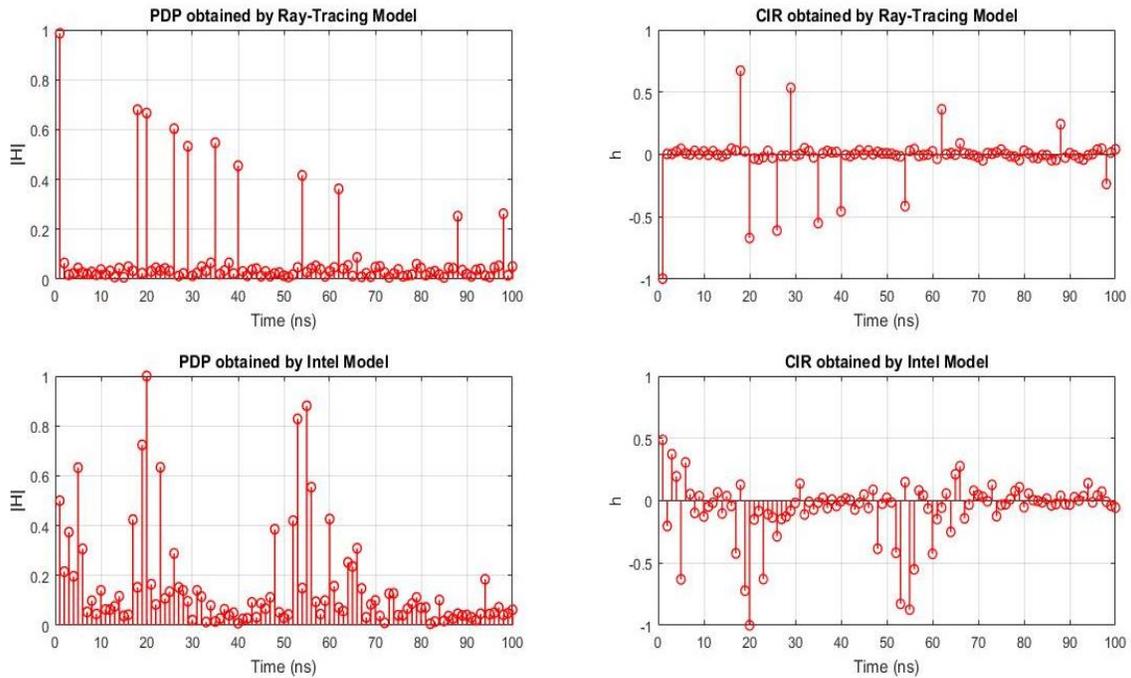


Figure 3.5: Sample PDPs and CIRs obtained from different channel models.

3.3 Compressive Sensing

3.3.1 Compression and Expansion

In Compressive sensing framework, compression of CIRs is done through multiplying the presumably-sparse vector (channel realization) by a Gaussian Random sensing kernel. This operation simulates the receiver inability of detecting fine resolution CIRs with the presence of the wideband signal sensed by its antenna elements (zooming out the delay axis). Sensing noise is the error that results from the imperfections of the receive elements structure, and assumed to be Gaussian with i.i.d samples.

The compression matrix used is a normalized Gaussian matrix that has orthogonal columns. Other matrices should be conditioned by satisfying the RIP constraint. Possible choices are random matrices that have entries chosen according to sub-Gaussian, Bernoulli or Partial Fourier distribution. The RIP property is summarized mathematically as:

$$(1 - \delta_s) \|\theta\|_{l_2}^2 \leq \|\Phi\Psi\theta\|_{l_2}^2 \leq (1 + \delta_s) \|\theta\|_{l_2}^2 \quad (3.2)$$

Signal Reconstruction problem (designated by CS Expansion process in Figure 3.6) is solved using compressive sensing framework. To recover (i.e., zoom in the delay axis) the sparse CIR with an acceptable degree of error (MSE level), a linear programming algorithm is used. Matching Pursuit (MP) algorithm variants are investigated for this purpose. The performance of the reconstruction step is subject to the choice of sensing matrix structure in the (CS Compression) step, as well as the desired level of MSE error and compression ratios. A more reliable implementation of the MP concept (Section 2.3.3) is achieved by Compressive Sampling Matching Pursuit (CoSaMP) algorithm (Section 2.3.4).

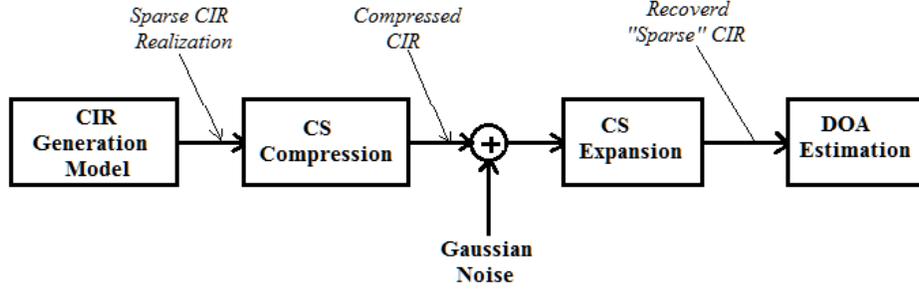


Figure 3.6: CS compression and Expansion for CIR realization.

3.3.2 Minimum Square Error Comparison of CS Algorithms

How to choose the best algorithm for UWB sparse channel is an issue of complexity versus performance and design. An MSE-based criterion for comparison between MP algorithms is deployed. The performance depends on the noise level as well as the channel simulation parameter. The MSE error is calculated as:

$$MSE = \frac{1}{L} \sum_{i=1}^L [h_{actual}(i) - h_{recov}(i)]^2 \quad (3.3)$$

where

$L \equiv$ length of the CIR vector “no. of delay bins”.

$h_{actual} \equiv$ Actual CIR before compressing.

$h_{recov} \equiv$ Recovered CIR after expansion.

The selection of the CoSaMP algorithm is done based on having the minimum MSE error in the expansion step compared to the actual CIR produced by the Intel and Ray-Tracing model. The setup that validates this procedure is presented in the results part of this thesis (Section 4.3).

3.4 Direction Finding

3.4.1 Algorithms Implementation

Given the received signal vectors (n of them, each corresponds to a different antenna element), the algorithms MUSIC, ESPRIT and Root-MUSIC work by obtaining an estimate for the received signal correlation matrix \mathbf{R}_y . The accuracy of such methods depends on the window size (the number of samples captured to estimate \mathbf{R}_y). In other words, DOA estimation resolution is a function of the bin size and the window length (number of samples captured). That is the more number of samples, the better is the estimate of \mathbf{R}_y to ensemble the true covariance matrix.

Eigenvalue decomposition is an essential step into projecting the signal eigenvalues in the signal/noise spaces. Given \mathbf{U}_s “the signal space”, \mathbf{U}_n “the noise subspace”, $\mathbf{\Sigma}$ “the signal eigenvalues”, MUSIC, ESPRIT and Root-MUSIC algorithms do calculate a DOA estimate. The difference is the first algorithm “MUSIC” produces a spectrum with high peaks at DOAs of the incoming signals, while ESPRIT uses a Total-Least Squares solution to numerically estimate the angles of arrivals. The third algorithm looks at the problem as a polynomial roots solution.

The number of incoming rays (P) is estimated using an information theoretic detection method. Namely, Akaike Information Criterion (AIC) and Minimum Description Length (MDL) (Section 2.4.3) are implemented to resolve this question. Although both measures are similar and straightforward, an efficient eigenvalue decomposition does affects their performance. Thus P is given by:

$$P = -2K(N - M) \ln[Q(M)] + 2M(2N - M) \quad (3.4)$$

in case of using the AIC metric, or:

$$P = -K(N - M) \ln[Q(M)] + \left(\frac{1}{2}\right)M(2N - M) \ln(K) \quad (3.5)$$

when using the MDL metric.

Figure 3.7 summarizes the steps for MUSIC algorithm. For the analytical DOA estimation methods (ESPRIT and Root-MUSIC) there's no need to calculate the spectrum or a search step, the angle of arrivals is given by Section 2.4.2 as:

$$\theta_i = \sin^{-1}[-\tilde{\lambda}_i/2\pi d_x] \quad , \quad i = 1, \dots, P \quad (3.6)$$

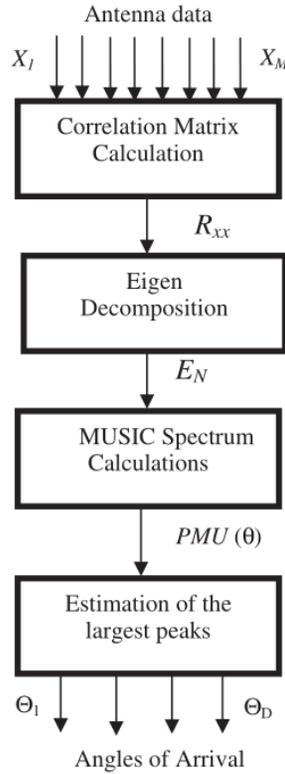


Figure 3.7: MUSIC algorithm flowchart.

3.4.2 Angle-Delay Domain Clustering

The technique of obtaining a cumulative DOA estimate is done through clustering individually estimated DOAs in the Angle-Delay domain. Individually estimated DOAs are the results of running one of the conventional algorithms for finding the AOA on a bin-by-bin basis. Estimates of DOAs are then viewed as a two-dimensional space; angles axis and their respective delay bins on the other axis. The points of that space are shown to be scattered and condensed around the actual rays' DOAs.

A post-processing step for clustering “grouping” DOAs is made possible by the use of the K-means algorithm (Section 2.5). A modified technique is used when clustering the points in the angle-delay domain. The procedure of representing different DOA estimates with different number of points is done as follows: the estimates obtained using low power are represented by fewer points, and vice versa (i.e.: the more the received power the more the confidence on the estimates that are based on them).

This method leads to treating noisy received bins as noisy DOA estimates, so that actual received rays are favored in the clustering step. The way of judging whether a favored estimate is reliable or not depends on the definition of “received high power”. There are several ways of defining that, some of them are: by averaging all the powers across all the receiver antennas in each bin, finding the weighted product sum, selecting the maximum power to represent that bin power, using the variance or the sum of square roots (i.e., sum of the received voltage).

The clustering step is supposed to find centroids (the average cluster angle) of the DOA clusters which should be compared against the actual ray AOA. The performance of this process is subject to the performance of the estimation algorithm used (i.e., ESPRIT or Root-MUSIC). Other factors are based on the channel structures (density and geometry of reflectors) and CIR resolution captured at the receiver. The choice of which method to use on treating high powers is subject to more investigation and analysis.

3.4.3 DOA Estimation Metrics

The most looked-for features of a Direction-finding algorithm is having high accuracy and resolution when applied using small aperture arrays (small antenna footprint) and yet using the minimal number of antenna elements, and performing in lowest possible SNR environment, though having a reasonable cost/performance implementation.

The root mean square is used as a metric to compare the performance of DF algorithms. The RMS Error (RMSE) in DOA estimation is computed as the square root of the average squared errors over repeated DOA estimation trials. It is defined by

$$RMSE = \sqrt{\frac{1}{T} \sum_{t=1}^T \frac{1}{P} \sum_{p=1}^P (\theta_p - \hat{\theta}_p)^2} \quad (3.7)$$

where,

$T \equiv$ Number of repeated DOA estimation trails over which the error is averaged,

$P \equiv$ Number of incident rays available at the receiver elements (estimated using AIC or MDL),

$\theta_p \equiv$ The actual angle of arrival of the p th ray as generated in the simulation model,

$\hat{\theta}_p \equiv$ The estimated angle of arrival of the p th ray as estimated by the clustering DOA in the angle-delay domain.

CHAPTER FOUR

RESULTS

4.1 Review

In this chapter, a review of the simulation setup is provided. A set of functional blocks is detailed in order to illustrate the various steps (Figure 4.1) that represent the verification of the DOA estimation method and the parameters considered in validating its performance are described.

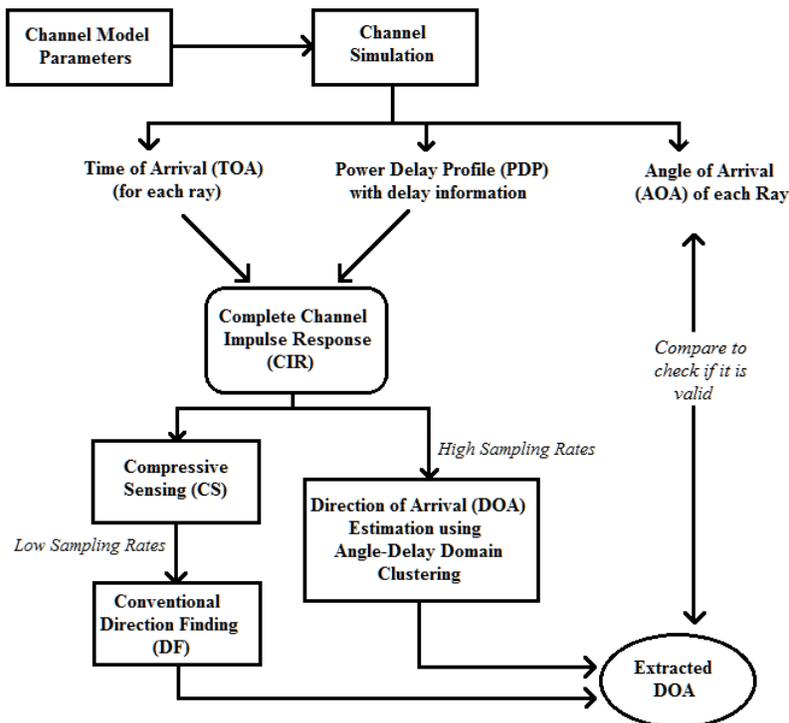


Figure 4.1: Block diagram of the procedure.

In the subsequent sections, the different parts and the overall system performance of the proposed DOA estimation technique is listed and discussed. The technique described in chapter 3 is a way of estimating the DOA of an incoming signal in an UWB transmission environment. Performance measures, results, and graphs that illustrate the different steps are listed in the succeeding sections. The tool used for testing and verifying the direction-finding system is Matlab.

The core of the pre-described technique involves the algorithms/techniques: UWB Channel generation (Intel model and Ray-Tracing model), CIR Compressing, CIR Expansion (with CoSaMP reconstruction), DOA estimation (ESPRIT or Root-MUSIC), Angle-Delay Domain Clustering (K-Means), Performance metrics (MSE).

There are two channel models deployed to test such scenarios. The Ray-Tracing channel model and the model derived from the measurements done by Intel are simulated to obtain CIRs of different channel realizations.

The later discussion involves the use of compressive sensing in dealing with the high bandwidth transmission. The receiver is assumed to be incapable of sampling at high sampling rates dictated by the bandwidth of the transmitted signal. The efficiency of such method is debated and its results are expressed.

Conventional DOA estimation numerical methods (ESPRIT/Root-MUSIC) are utilized in a bin-by-bin basis to find intermediate DOA estimates in each delay-bin to form a constellation of points in the Angle-Delay space. The following sections deal with the problem of clustering DOA estimates in the Angle-Delay domain.

The final DOA measures are assumed to be the centroids of the concentrated distinct estimates. Readings from the simulations are compared to the actual values of ray DOAs stated by the models for verification. Comparisons are facilitated using the RMSE metric.

4.2 UWB Channel Generation

4.2.1 Ray-Tracing Model

Simulations to get different CIRs to mimic real life UWB channels are done using the basic knowledge of radio frequency propagation phenomenon. The methodology described in Section 3.2.2 dictates that a realization of a power delay profile (PDP) along with individual rays DOAs would give sufficient information to deal with the DOA estimation problem.

A scenario where there are 4 receiving antenna elements is used to give further explanation of the system operation. Figure 4.2 shows a snapshot of the tracing operation of this scenario where a ray (purple line) is registered to hit the first and the third antennas.

The setup parameters are as follows: frequency of operation is set to 3 GHz, TX sweep angle (scanning range) is set to 360 degrees, maximum allowed number of reflections is 16 reflection per ray, room dimensions are $10 \times 8 \text{ m}^2$, barrier is set to be relatively at the middle of the room and inclined towards TX side, minimum hit/miss detection angle is set to be 0.5 degrees, bin size is 10 cm, reflection coefficient is same across the room boarders and the barrier and is set to 0.75 and for purpose of illustration the antenna interspacing distance is set to 20 cm.

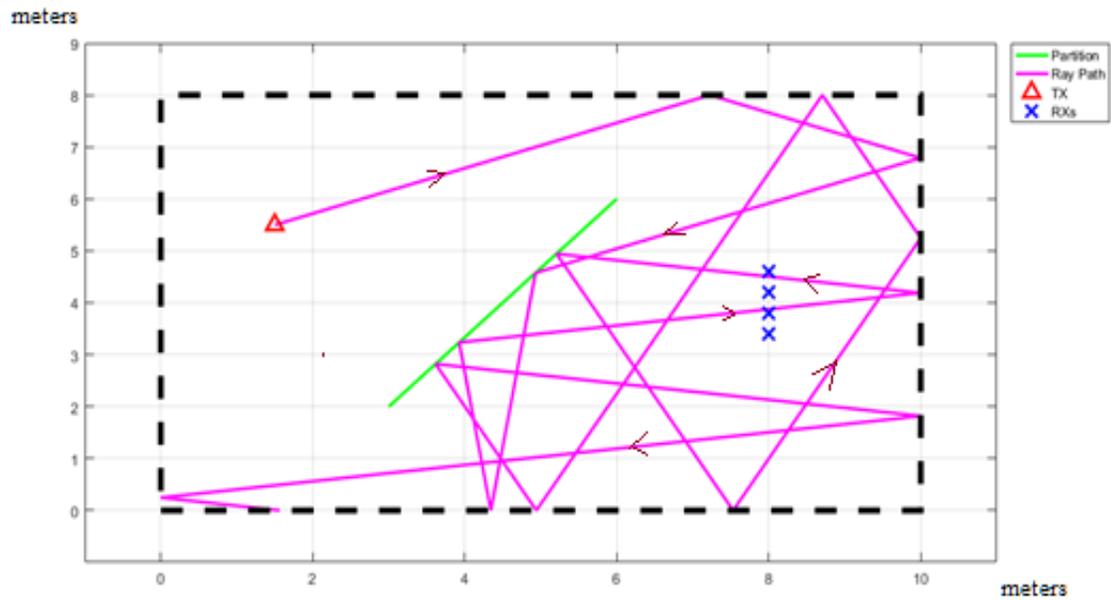


Figure 4.2: Ray-Tracing Model operation snapshot.

The resulting PDPs of each receiving antenna of the scenario defined in Figure 4.2 with respect to the time domain is plotted in Figure 4.3. The power levels are normalized and scaled to 10 dB with respect to the highest received power across the 4 antennas (the LOS component), the signal power is set to 10 dB above the noise level. The length of the CIR in ns is set to 700 ns; as the powers of later received rays tends to fade out due to multiple consecutive reflections (the lengthier path traveled by the EM wave the more its power drops due to free space loss). Multipath Components (MPCs) are shown to be sparse and have a nature of arriving times at the receiver antenna elements as clusters.

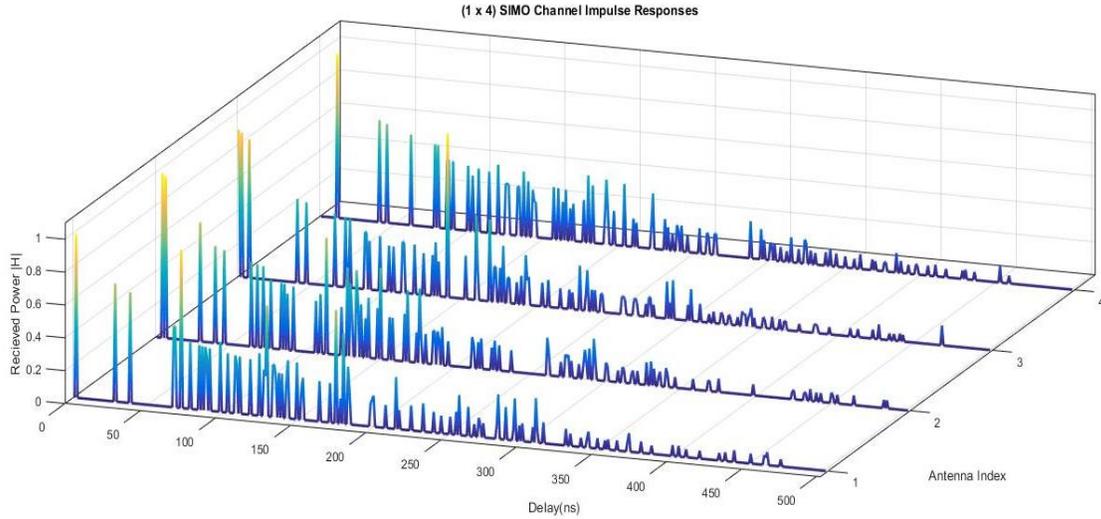


Figure 4.3: CIR obtained from Ray-Tracing simulation with the aforementioned parameters.

Each ray hits one of the receiving elements, several variables associated with that ray are recorded: the sweep angle in which it has been recorded to hit an RX antenna, the path length traveled through multiple reflections, whether two or more rays arrive at the same time or not (and their number), and most conveniently, their DOA resulted from the last reflector. These DOA's are used for comparison with computed DOAs.

The graph on Figure 4.4 shows the angle of arrival (AOA) generated by the simulation per ray for the same previously described scenario. Each group of rays that falls more or less in the same delay bin (slightly before or after) are shown to have similar AOA (although not exactly equal). The fact that RX interelement distance is small compared to the actual path the ray travels accounts for the main idea of direction detection with multiple antennas. It is to be mentioned that the maximum delay shown is reduced to 250 ns for convenience.

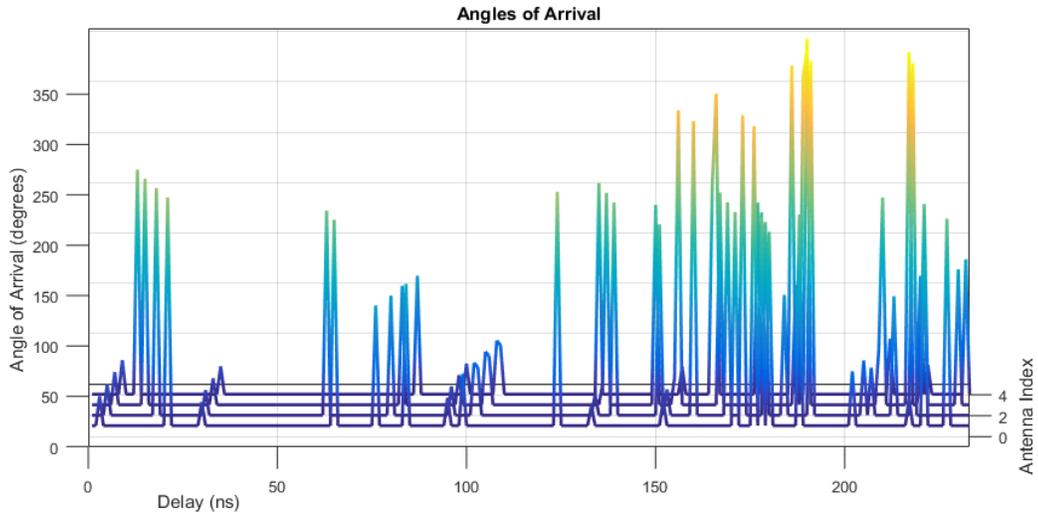


Figure 4.4: Angle of arrivals of different MPCs.

4.2.2 Intel-based Model

The model accepts channel parameters and produces a channel impulse response according to the Extended Saleh-Valenzuela channel model. The control of the model to produce a sparse channel impulse response is done through the suggested value made by Intel and their measurement results recorded in [6]. The parameters that controls the model to produce different CIR in terms of number of clusters and arrival times are summarized by the code snippet in Table 4.1.

Table 4.1: Intel-based Channel Model Parameters.

```
% Extended Saleh-Valenzuela Model:
% Returns the impulse response of the indoor multipath channel
% as modeled by Intel in IEEE 802.15-02/279r0-SG3a and is
% based on the Saleh-Valenzuela model with lognormal fading.

% Outputs:
%   CIR           :   Simulated Channel Impulse Response.
%   t             :   Arrival time of each ray.
% Inputs:
%   LAMBDA        :   Cluster arrival rate(GHz) (avg # of
clusters per ns)
%   lambda        :   Ray arrival rate(GHz) (avg # of rays per
ns)
%   GAMMA         :   Cluster decay factor (ns)
%   gamma         :   Ray decay factor (ns)
%   std_shadow   *:   Standard deviation of log-normal shadowing
%   LOSflag       *:   Flag to specify Line Of Sight component
availability
%   T             *:   Time resolution (ns) .
%   MaxLength     *:   Maximum length of the impulse response
% marked with (*):   Optional Input.
```

A sample simulated channel snapshot is shown in Figure 4.5. The channel length is set to 200 taps (coefficients) and the model produces 5 clusters of MPCs (for each channel of the 4) that forms the assumed sparse channel. The simulated UWB in this case is a NLOS channel and the SNR value is set to again 10 dB. As previous, the power is normalized across the 4 antennas and scaled to 10. The graph shows a single PDP, where the MPCs are plotted against the time axis. The arrival time of each ray is random, and it is shown to follow a Poisson distributed arrival time “as assumed by the S-V model”).

The values of model parameters used to generate these results are listed below:

```
LAMBDA = 1/21.5;   lambda = 4.5;      GAMMA = 50;      gamma = 2.6;
sigma1_dB = 4.8;   sigma2_dB = 4.8;  std_shadow = 2;  LOSflag = 0;
```

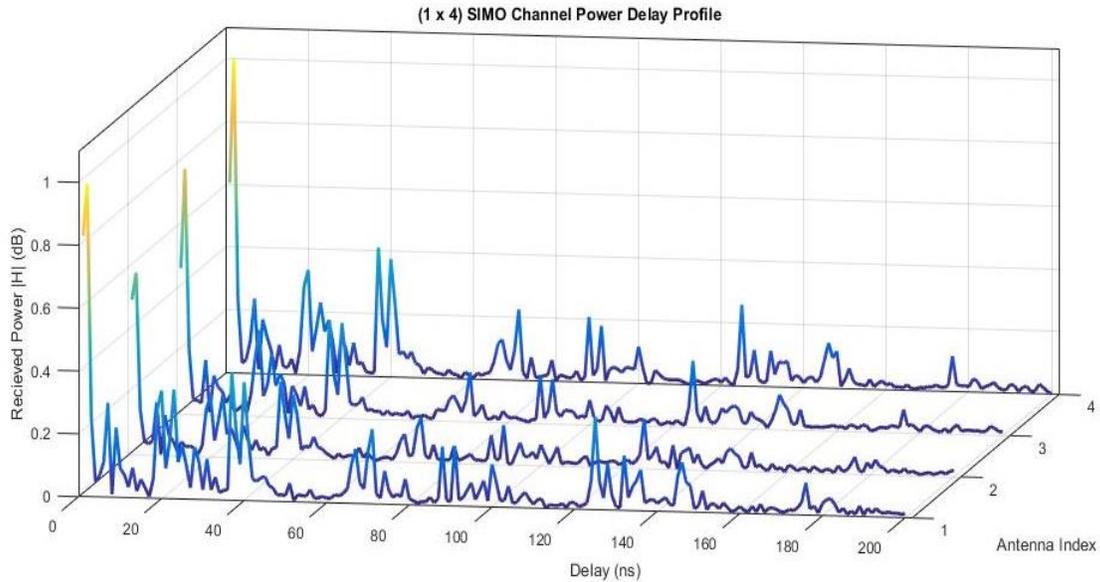


Figure 4.5: A sample CIR generated by Intel UWB Model simulation.

The CIR obtained by this model lacks the DOA information of individual rays. Although through PDP clustering (a way of identifying separate clusters), each cluster is given a reference DOA to compare with the computed DOA. PDP clustering is done through observing concentrated powers and associating each cluster with its center arrival time (arrival times of individual rays are obtained from the model as secondary outputs).

4.2.3 CIR Models Comparison

Both models discussed in Section 4.2.1 and 4.2.2 generate channel impulse responses (CIRs) to simulate UWB channels. The power-delay profile (PDP) of each one is

considered to be a vector of samples (delay bins) that correspond to different ray arrival times. Figure 3.5 shows a sample PDP and CIR outputs of those models.

The model based on ray-tracing (Section 4.2.1) gives the reference DOAs for each individual ray, while the model based on Intel lacks an intrinsic DOA information. Although, the second model can be controlled by certain parameters (Table 4.1) to produce a specific number of clusters, as well as the mean time of arrival between the clusters. Figure 4.6 illustrates the use of both models in the subsequent DOA estimation method.

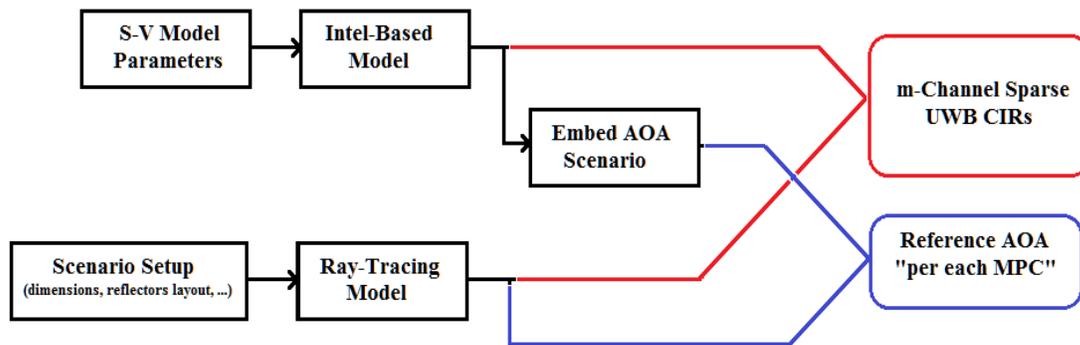


Figure 4.6: CIR Models used and their outputs.

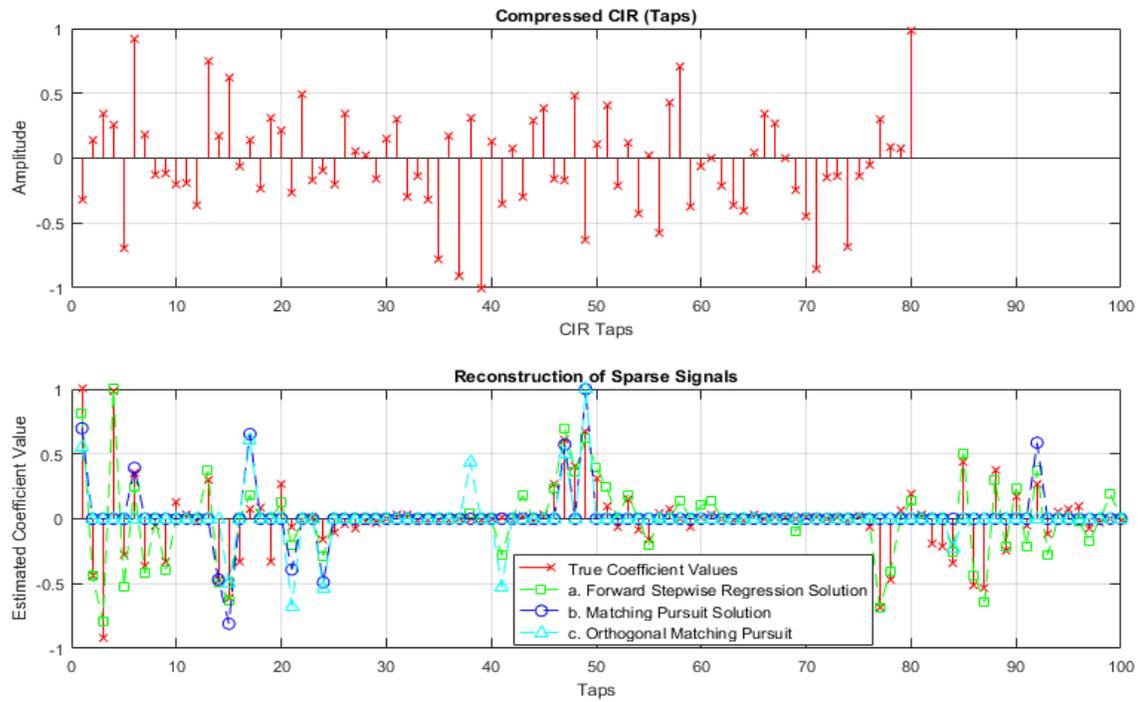
The first model (Ray-Tracing) is suitable for simulating a CIRs of an environment without previous knowledge of cluster/ray arrival rates and shadowing coefficients, while the second model (Intel) is best suitable for simulating general UWB transmission environments. The first model gives a complete scenario of the DOAs of individual rays, while in the second model, a step of embedding an Angle of Arrival (AOA) details is necessary.

4.3 Compression and Expansion of CIRs

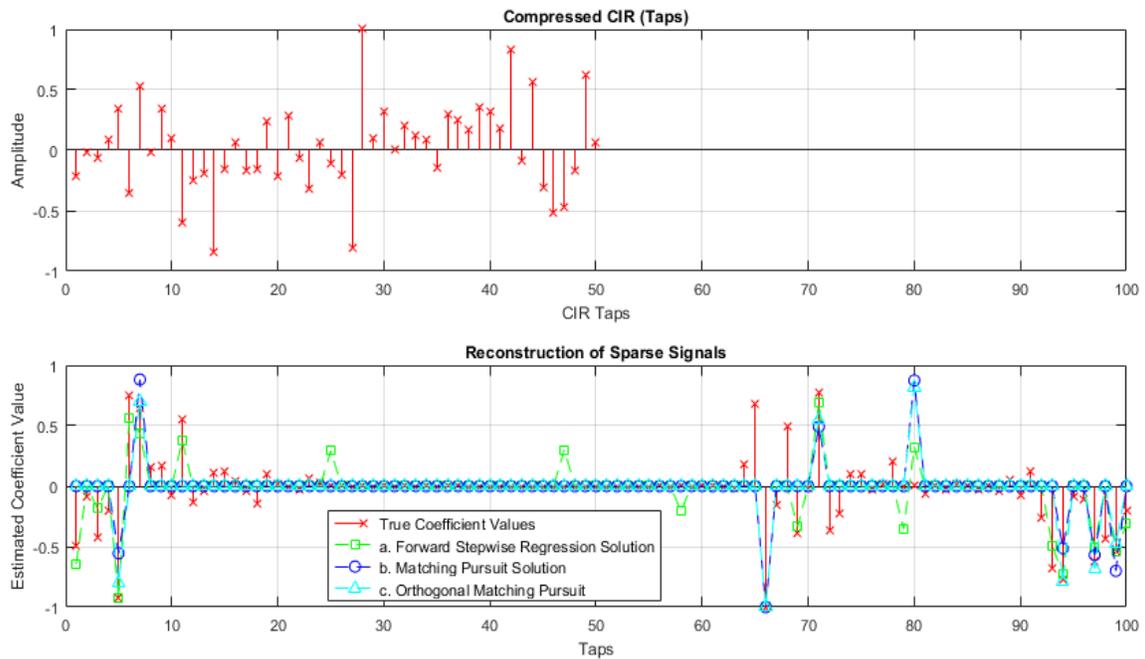
Compressive sensing is implemented with help of *SparseLab* 2.1 library for Matlab. The compression is done at the transmitting side to simulate the receiving of a compressed vector of the signal. After extracting the CIR from the incoming signal (assuming the data transmitted is common between the two sides “a pilot signal”) by means of channel identification, the receiver tries to estimate the “full-length” CIR by applying an Expansion procedure (sparse vector reconstruction from a limited set of measurements).

In Figure 4.7 (a), a channel of 100 taps length is compressed using a random Gaussian compressing matrix. The compression ratio (CR) is set to accommodate for 80% of the true number of the samples (in this case, only 80 taps are considered by the receiving end as the effective CIR). Matching Pursuit algorithms (three of them, namely: the conventional MP, Forward Stage-wise Regression and Compressive Sampling Matching Pursuits (CoSaMP)) are deployed to get a solution as a “zoom-in” process that is done by the receiver by solving the under-fitted $Ax = b$ linear equation using one of the compressive sensing sparse CIR reconstruction techniques.

Figure 4.7 (b) shows another compression and expansion simulation. In this scenario, the compression ratio is set to 50% and the uncompressed channel length is 100 taps. As the compression ratio increases (less percentage of samples are used to represent the compressed CIR), the reconstruction algorithm produces more errors (higher MSE value). The other effect to be considered is the noise level (SNR). The following discussion illustrates those factors that affects the use of CS in obtaining finer time-domain resolution CIRs from the compressed representation.



(a) $CR = 80\%$,



(b) $CR = 50\%$

Figure 4.7: Reconstruction of Sparse CIR after compression using MP Algorithms.

The algorithms tested are Forward Stage-wise Regression (FSR), MP and CoSaMP. An MSE criteria is then utilized for comparison purpose. Figure 4.8 shows Monte-Carlo simulation results ran for 300 simulated CIR snapshots per 1 dB SNR value, generated by using the information described in Section 4.2 to compare the MSE efficiency. The CIR length is kept fixed at 100 delay bins. First, the compression ratio is set to 60% (i.e., the three algorithms (FSR, MP and CoSaMP) try to reconstruct the original CIR from only 60 randomly sampled vectors) second it is set to 40%, and last it is set to 20% to illustrate the effect of varying the compression ratio on the expansion's error (MSE value). Section 3.3.2 summarizes the procedure for MSE comparison between several CS algorithms. The choice of using CoSaMP algorithm is justified by having the minimum error values for all SNR iterations as Figure 4.8 illustrates.

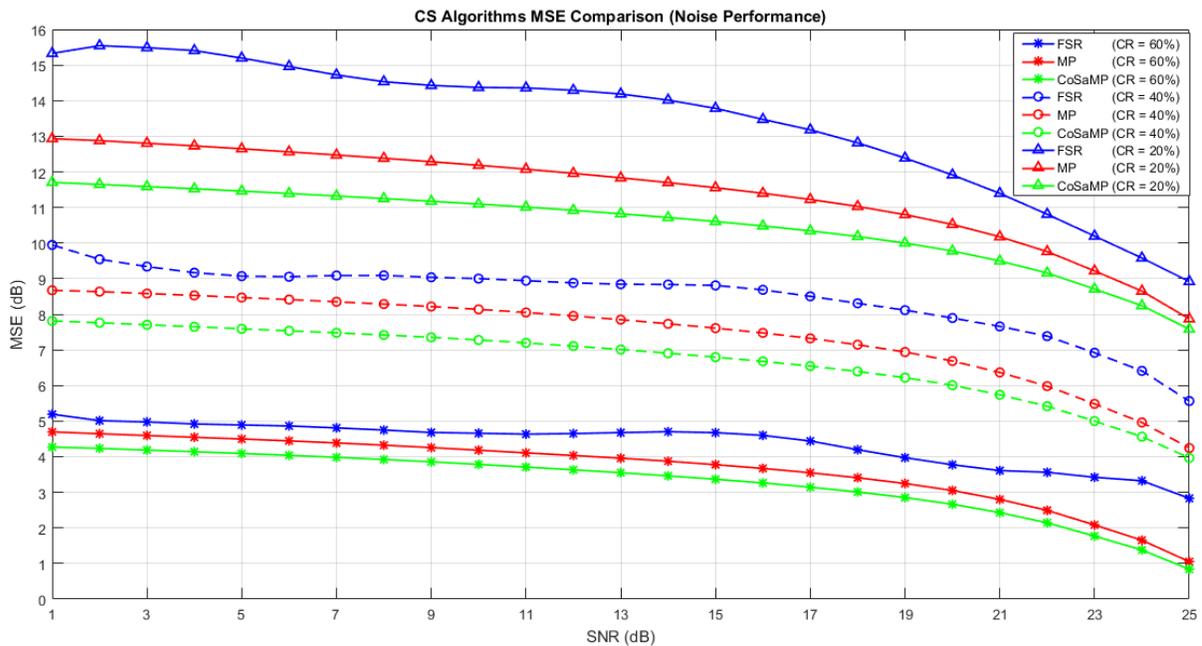


Figure 4.8: MSE-based noise performance comparison for CS algorithms.

The effect of compression is tested by varying the compression ratio under fixed SNR value of 10 dB. Another Monte-Carlo simulation ran to compare the performance of the three algorithms (FSR, MP and CoSaMP) by changing the compression ratio between 10% to 90% (where 10% means compressing the CIR to 10 random samples from 100 delay bins available in the original CIR) and calculating the reconstruction MSE error for 200 different CIR realization for each compression ratio value. Figure 4.9 states that the CoSaMP algorithm has the minimum error values for all compression ratios.

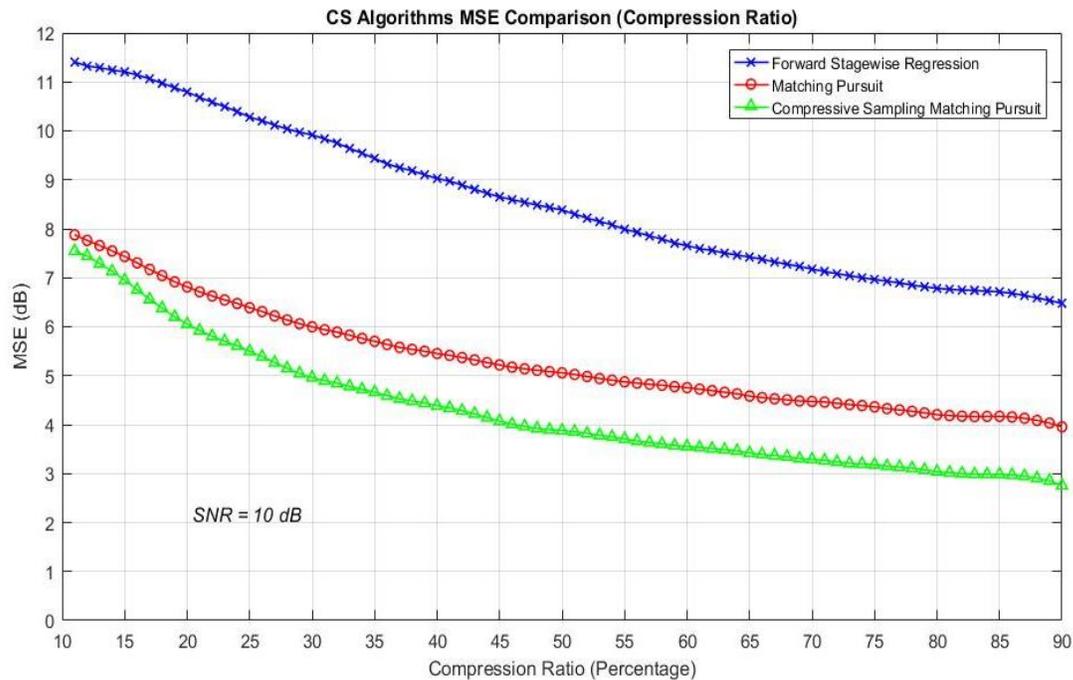


Figure 4.9: MSE-based compression ratio comparison for CS algorithms.

As expected in the review of the MP and the CoSaMP algorithms presented in Section 2.3.4, the introduction of an orthogonal solution at each iterative step is better in terms of

MSE performance (as dictated by Figure 4.8 and 4.9). Therefore, the use of CoSaMP in expanding the CIR at the RX side is approved under such constraints (and when compression is done at the TX through multiplying with a random orthogonal Gaussian sensing matrix that is already known at the RX side).

4.4 Direction Finding Simulation

The conventional method of finding the direction of arrival are used to compare with the proposed method results. The numerical DOA estimation in the proposed method is carried out using ESPRIT and Root-MUSIC.

MUSIC algorithm outputs a spectrum for probable DOAs. The spectrum gives a degree of confidence described with peak powers. ESPRIT and Root-MUSIC gives a numerical value for the DOAs as they don't require a search "for peaks" step.

A simulation run for the conventional DOA algorithms with the following parameters is carried out:

No. of signal to be detected $P = 1$, No. of receiver antennas $M = 4$,
Signal-to-Noise Ratio $SNR = 10$, No. of samples (window size) $K = 3000$
Simulated Angle of Departure $AOD = -20^\circ$

MUSIC spectrum result is given in Figure 4.10. ESPRIT algorithm simulation is performed (using TLS) and it yields -19.90° as the direction of arrival estimation, while Root-MUSIC result came to be -19.91° .

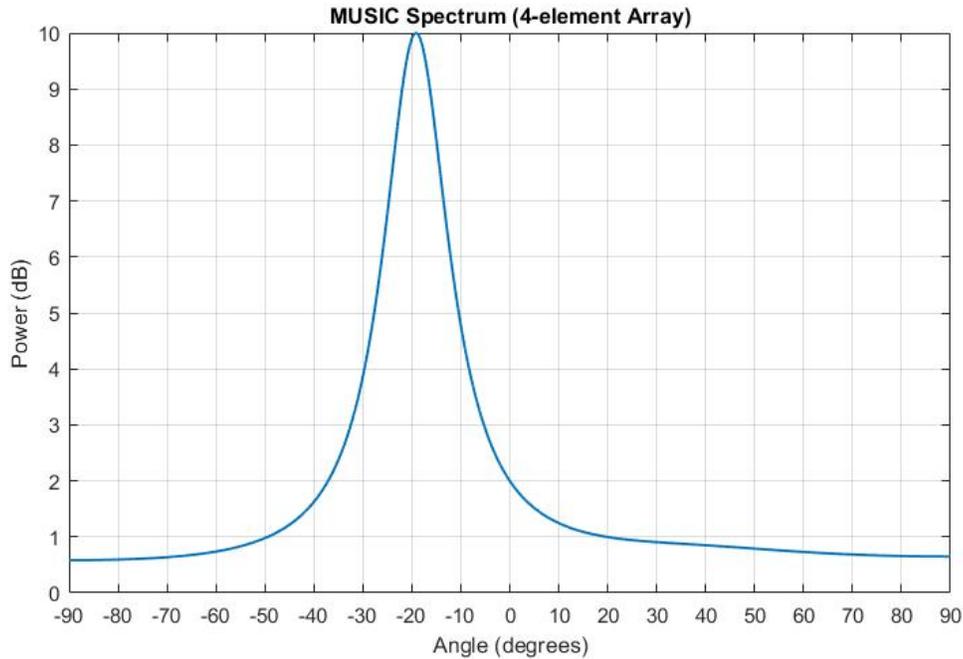


Figure 4.10: MUSIC spectrum simulation.

4.5 Angle of Arrival Estimation Method

The core task is to find angles of arrivals based on each incoming ray as a single angle estimate. The main aspect that differentiates this method from conventional DF techniques is that it estimates the direction from CIRs rather than received signals vectors. Individual Estimates are done by grouping m (no. of receiver antenna) CIR coefficient tabs in a vector “ χ ”. Then ESPRIT estimator estimates the angle by using the eigenvalue λ obtained from the correlation matrix as

$$\theta = \sin^{-1} \left[-\frac{\lambda}{2\pi d} \right] \quad (4.1)$$

A step for aligning each incoming ray across the m considered CIR coefficients is achieved by associating the MPCs coefficients based on the minimum Euclidean-distances between their samples. The less this distance measure indicated the high probability that two or more MPCs (across the m antenna elements) are due to a specific reflector (thus having nearly equal magnitude and phase). Figure 4.11 shows an illustration of the considered ray arrival patterns.

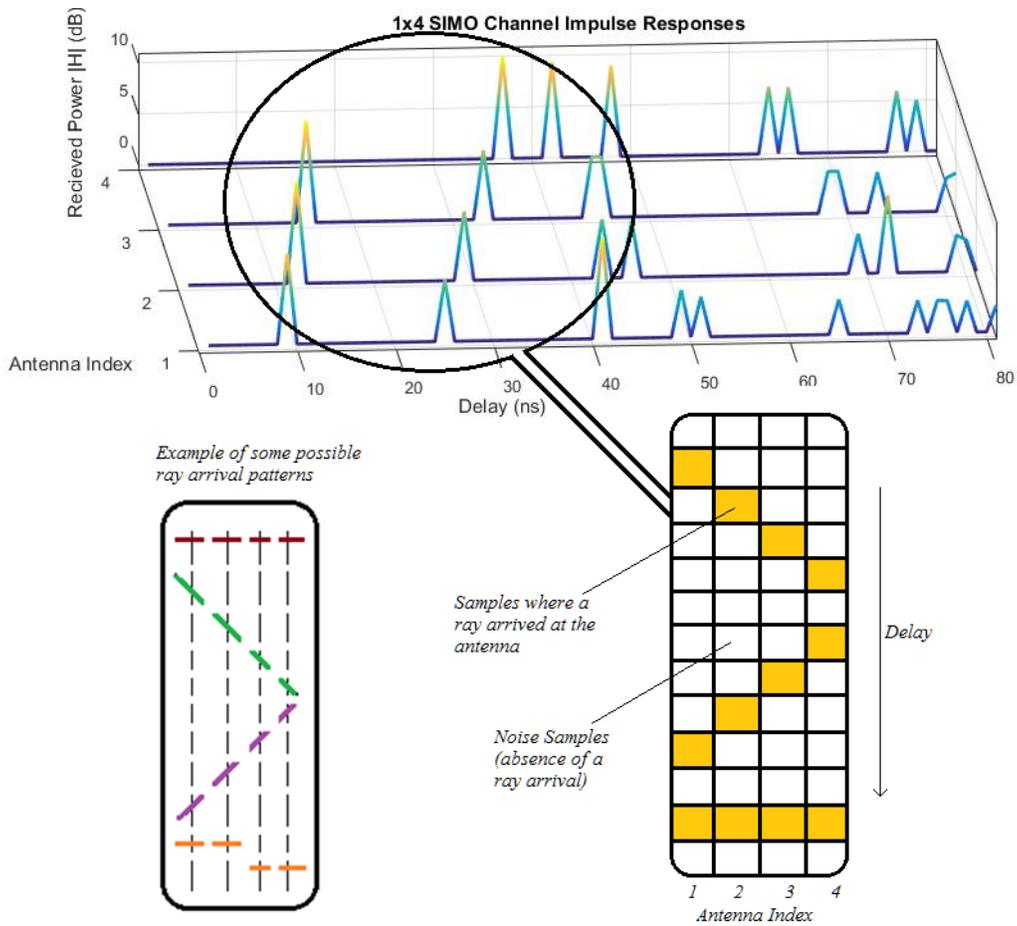


Figure 4.11: Different ray-arrival patterns.

The minimum distance measure is to associate different rays together to make a single DOA estimation. The Euclidian distance between a ray-bin and an i.i.d noise sample is considered the highest, compared to the distance between 2 ray-bins arrived due to the same reflector (having more or less similar amplitude and phase, rather than random values). The measure of how likely each two ray samples correspond to a unique reflector is given by:

$$Prob. of alignment = \sum_{i \neq j}^m \min (|h_i - h_j|^2) \quad (4.2)$$

The estimation done for each “aligned” bins is performed extensively through all possible coefficients in a pairwise fashion (instead of taking all the m together at once) producing several intermediate DOA estimates per delay bin. By averaging the possible variations based on the power of the received component (weighting the estimates by power as a measure of confidence in their estimated AOAs), final DOA estimate is obtained to represent the angle axis in the Angle-Delay domain. The distance d does change through the possible combinations (i.e.: distance from antenna 1 to 2 is not equal to the distance from antenna 1 to 3).

High power CIR coefficient are more probable to be considered as (actual) rays. Noisy bins have a nature of random DOA estimates (because they are random Gaussian samples). Although the “noisy” estimates would be influenced by nearby (in the delay domain) “actual” rays’ coefficients. Thus, estimates tend to scatter around the actual AOA of each specific ray.

Figure 4.12 explains the idea of Angle-Delay clustering. The simulated CIRs are obtained by the procedure described in Section 4.2.1 (based on Ray-Tracing channel model). Four CIRs of length of 130 ns is considered with a noise level of 10 dB. A number of 6 incoming rays are aligned together to be fed to an ESPRIT angle estimator. Different combinations of DOA estimates are averaged according to their powers. The reference AOAs for the 6 rays are produced by the model and equal to -69.27° , 25.70° , 75.61° , 19.75° , -32.23° and 29.16° with respect to arrival time.

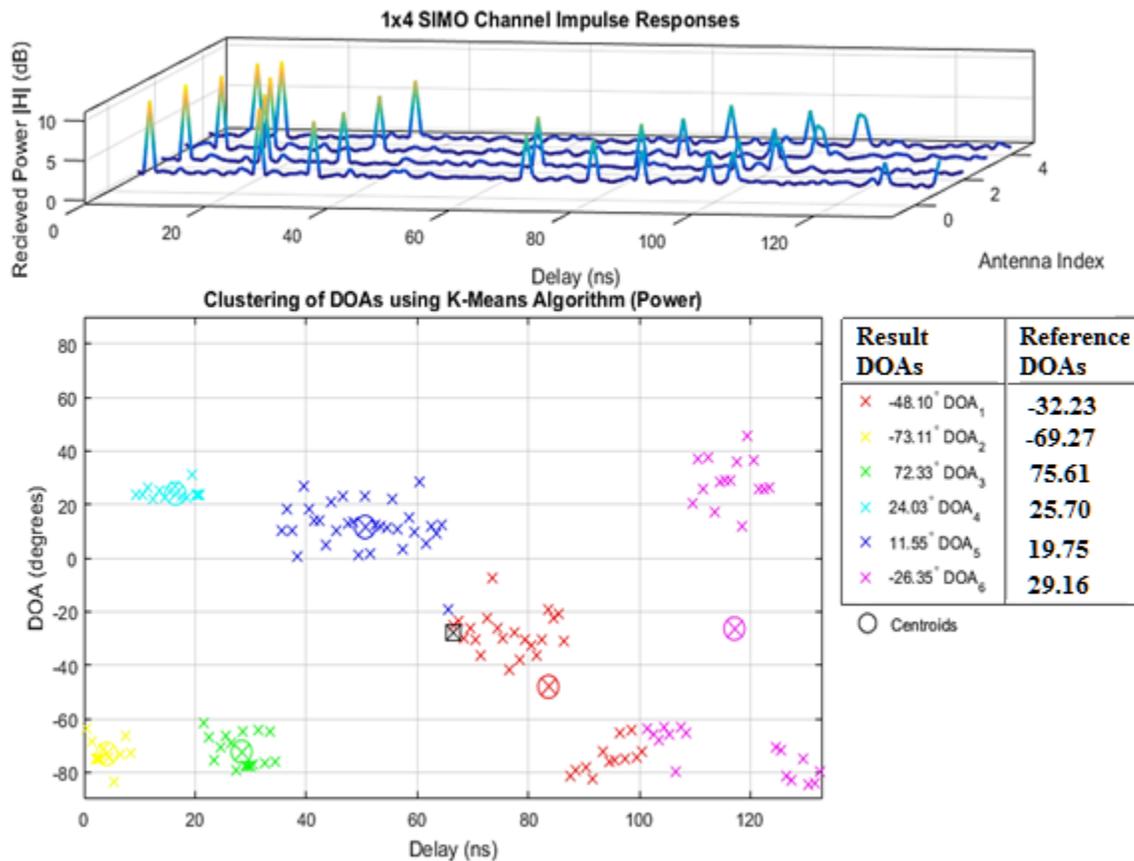


Figure 4.12: Angle-Delay Domain clustering for DOA estimates.

The clustering of DOA estimate is carried out by a modified k-means algorithm. The average power for each bin (across all the four CIRs) is used to replicate the DOA estimates in the Angle-Delay domain. The highest mean power is represented with 5 co-located points and the lower is represented only once. Centroids of the 6 clusters came to match expected outcomes. The group of bins from 0 to 40 ns are having high power (compared to noise level), thus, distinct clusters and clear centroids verification (although a little skewed by additive noise samples). The less the received energy the difficult to distinguish between actual received MPCs and noise samples, thus, scattered clusters with off-located centroids (purple cluster in Figure 4.12).

4.6 Method Test Scenarios

Since the proposed DOA estimation method is intended for various UWB channel structures, the method is applied to those combinations: Large/small area premises, LOS/NLOS. The first aspect (the dimension of the area occupied by the TX and RX) is investigated to verify the system operation in outdoor/indoor environments. Although UWB communications systems are mainly designed for short range indoor transmission links, by testing the “long-range” scenario, the extreme boundary of short-range systems is put into consideration. Long-range scenarios correspond to a closed-premises with a comparatively large area; hence stretched TX-RX distance “as in outdoor links”.

The first setup is for the case where the LOS component is not present in the CIR, and the TX-RX separation is set to simulate the outdoor case. Figure 4.13 shows 4 CIRs obtained by the Intel-based model, in which the room size is set to $40 \times 20 \text{ m}^2$. The rest of the

parameters (i.e.: bin size, RX antenna interspacing distance) are set to the previous values for fair comparison.

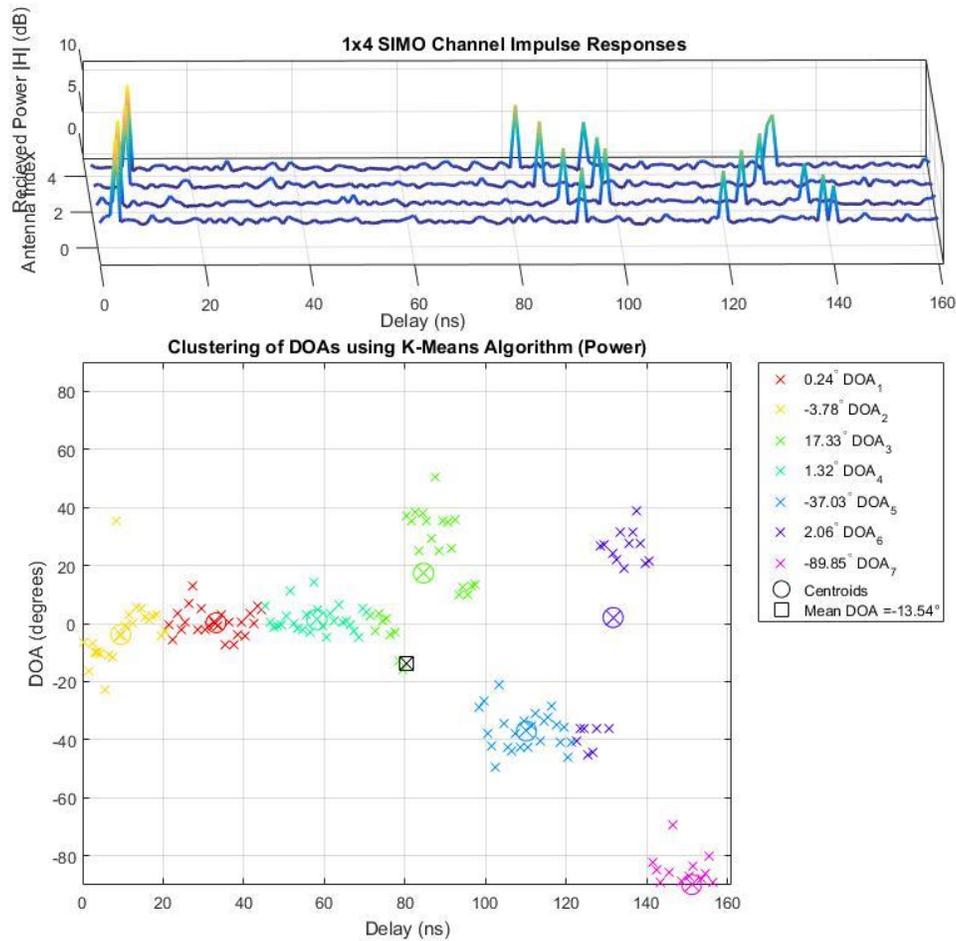


Figure 4.13: Line of Sight (LOS) scenario in outdoor setup.

The availability of a high power received ray (the first 10 ns), that is isolated from interfering rays from different directions would imply the detection of the TX Angle of Departure (DOD) with respect to the RX. The reason behind that is the LOS path is the shortest distance traveled by the EM wave and received by the RX antennas with AOAs that are slightly different (in this case, the 4 antennas are simulated to have 10.14°, 11.14°

12.14° and 13.14° regarding the LOS component AOA “designated by the yellow cluster of angles in Figure 4.13”).

The second scenario under examination is the case of an indoor channel with the room dimensions kept at $8 \times 5 \text{ m}^2$. The availability of a partition in between the TX and the RX is set to simulate an NLOS channel. Again, the rest of the model parameters are kept as previous. Figure 4.14 shows a sample of the system operation.

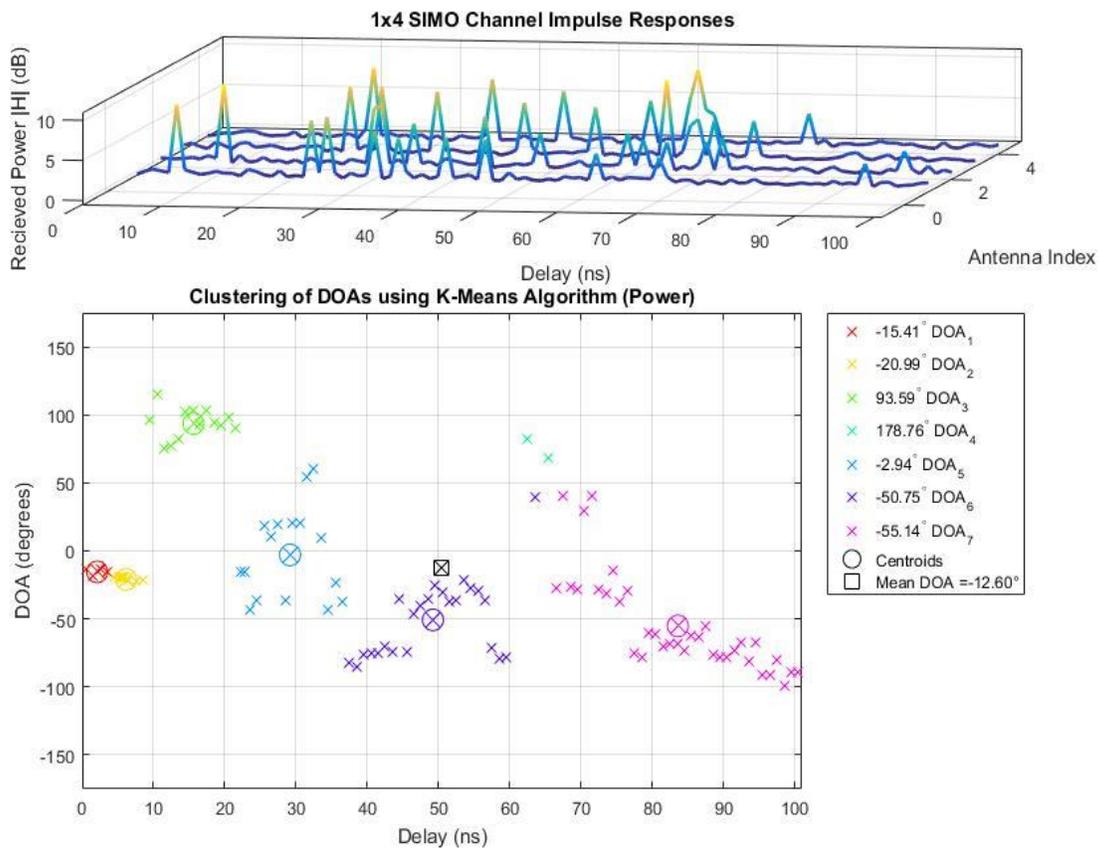


Figure 4.14: Non-Line of Sight (NLOS) scenario in indoor setup.

Due to the large amount of MPCs arriving close to each other (because of the multiple reflections from walls and the middle partition), the CIR loses its cluster-nature. Such CIRs suffers from a reduced sparsity level (the CIR become non-sparse) compared to the outdoor LOS case. Because of more rays that arrived in close time-bins have different AOAs, the Angle-Delay domain clustering of DOAs become a challenging process. In Figure 4.13, the points within a single cluster have large span across the angle axis, thus resulting in erroneous detection of centroids in MPCs-dense bins (i.e.: 20 to 60 ns).

4.7 Comparison to Conventional DF

The method of Angle-Delay Domain Clustering has some advantages when applied to find the DOA of multiple incoming rays in UWB channels scenarios compared to conventional DF methods. The amount of information extracted is per delay bin; in contrast with the collective answer nature of the direction estimate found when applying MUSIC, ESPRIT or Root-MUSIC.

In Angle-Delay Domain Clustering method, the estimates of the directions are found after acquiring the CIRs of the UWB using the receiver's antenna elements array followed by a CS expansion step to compensate for the high sampling rate requirement. A pre-processing block in which the rays are aligned in their most probably respective delay bins is used. Then a sample-by-sample basis ESPRIT estimator is applied extensively in multiple co-located samples in the delay axis to find multiple probable DOAs for each delay bin. Afterwards, clustering the space of DOA estimates with regards to their arrived average power to reflect strong ray's emphasis over noise sample.

In MUSIC method, the DOA estimator uses eigenvalues decomposition technique over a block of received samples from the antenna array. It treated the incoming signal as a vector of samples and computes the DOA from it directly (regardless of the CIR or the resolution of the received sample vector).

The amount of information extracted using the Angle-Delay Domain Clustering method is large compared to when using MUSIC to estimate the DOA of incoming signals (the number of directions found is per sample rather than per block of received signal vector). It can be shown in Figure 4.15 that by averaging the DOAs estimated per bin over the delay axis, the collective answer of MUSIC method of finding the DOA can be obtained.

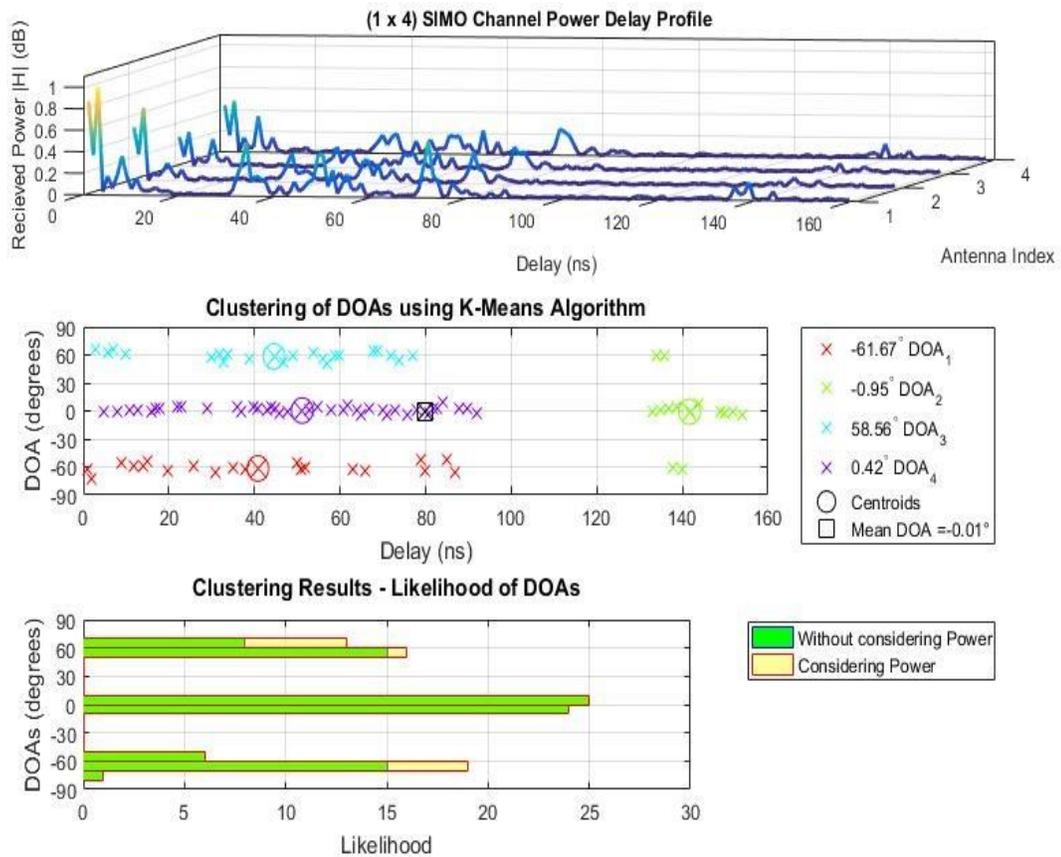


Figure 4.15: Angle-Delay Domain Clustering DF Method.

Figure 4.16 simulates 3 incoming signals (at AOAs of -60° , 0° and 60°) and their respective MUSIC spectrum. The windows size (K) affects the sharpness on the peaks of the estimated DOAs using a search tool operating in MUSIC spectrum. Figure 4.16 shows 2 different runs using $K = 20$ and $K = 500$ samples to illustrate the fact that when MUSIC is used to find DOAs it should have sufficient number of symbols to properly estimate received signal correlation matrix. The same information that is shown in Figure 4.15 when simulating the same scenario and using the Angle-Delay Domain Clustering method and then averaging the DOAs over the delay axis. The latter method uses a window of 20 symbols (5 delay bins across the 4 antennas to estimate 1 DOA point).

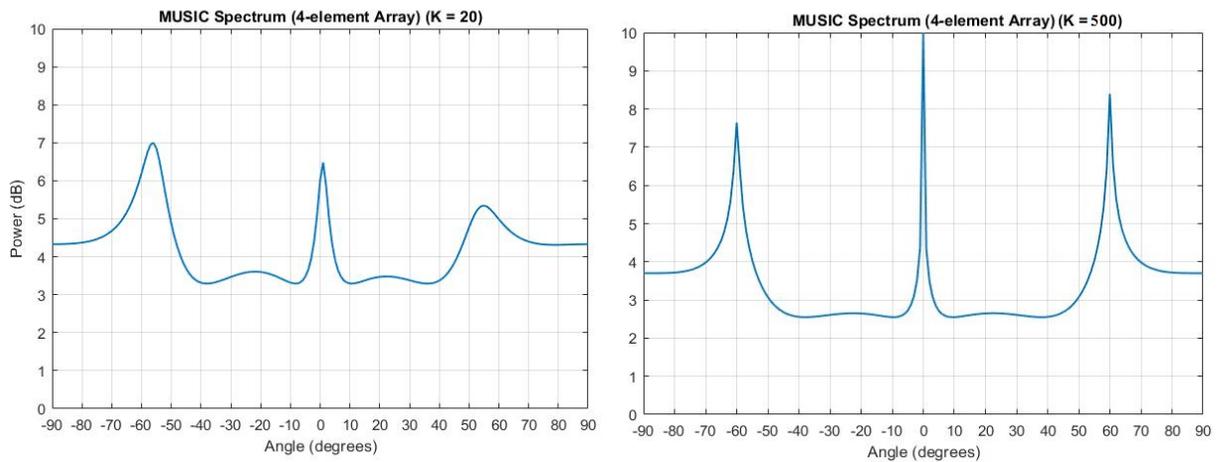


Figure 4.16: Conventional DF Method (MUSIC Spectrum).

CHAPTER FIVE

CONCLUSION

5.1 Conclusions

The modified technique for estimating the direction of arrival in Ultra-Wideband channels is formulated and verified. The method makes use of several functional blocks (algorithms) interconnected to reach an enhanced DOA measure at the receiver side for purposes of tracking and/or localization application.

UWB Channel is realized via two different models. The first model deployed is adopted from the model obtained by the measurements done by Intel in the IEEE standard 802.15-02. The second model is derived from basic propagation theory and depends on the idea of Ray-Tracing. Both models are used to verify various system components, and their simulations are documented.

The framework of Compressive sensing is utilized as a solution for obtaining a fine-resolution channel impulse response at the receiver, having lower sampling rates. A selection of Compressive Sampling Matching Pursuit algorithm is favoured in solving the CIR reconstruction from randomly sampled data. Such decision is justified by aid of an MSE comparison done to multiple competing algorithms.

Clustering DOA estimates from individual rays is achieved by applying a modified data analysis algorithm. The use of the K-means technique supports a collective answer from

the different estimates scattered in the Angle-Delay domain. The outcome is probable angles of the received signal directions with respect to the layout of a certain number of reflectors.

Computer software simulations are carried to describe the operations done by this method to attain its aim. Results are shown to match expected outcomes from the methodology and agreed to the mathematical formulations presented in the literature.

5.2 Future Work and Recommendations

The method presented in this project forms a relatively new idea of estimation the DOA in UWB systems. Different algorithms and techniques are utilized (this accounts for its implementation flexibility). Thus, the concept requires further investigation on selecting different algorithms implementation for each step of its operation.

Various number of recommendations can be listed. Regarding the ground of which the system is tested, different channel models can be explored. Scenarios where the TX and RX are closely separated or relatively far away (by covering a wide range of “both LOS and NLOS” environments). Channel with severe interference levels and/or strong fading characteristics. Also, adapting the method to medium-wideband and narrowband communication channels is recommended.

The model of Ray-Tracing can be improved by adding a complete layout of a room where several reflectors contribute to the multipath nature of the channel. Specific objects layout

scenarios (such as a living room, lecture halls and meeting rooms) would add more realization for the system operation.

In the part related to CS, optimization of the expansion process by considering more different algorithms implementations would benefit the flexibility of the system. Performance of such algorithms should be tested with different sensing matrices structure.

As for the detection of angles and the clustering in the Angle-Delay domain step, the suggestions are: investigation of high accuracy DF methods in resolving individual MPCs AOA's (enhancements from the kind of complexity reduction and execution time optimization is consider promising for the overall system speed and accuracy), different clustering techniques (that use adaptive methods for identifying the number of clusters) that is accustomed to a specific channel nature (distribution of clusters of MPCs and rays with in the clusters). The addition of other performance metrics (such as the standard deviation of dispersion of individual DOAs within each group of estimates with respect to the delay domain) would consolidate the reason behind this method popularity.

REFERENCES

- [1] M. Win and R. Scholtz, “Ultra-Wide Bandwidth Time-hopping Spread Spectrum Impulse Radio for Wireless Multiple-access Communications,” *IEEE Transactions on Communications*, vol. 48, pp. 679–689, April, 2000.
- [2] S. S. Chen, D. L. Donoho, and M. A. Saunders, “Atomic Decomposition by Basis Pursuit,” *SIAM Journal on Scientific Computing*, vol. 20, no. 1, pp. 33–61, 1999.
- [3] M. Unser, “Sampling: 50 Years After Shannon,” *Proceedings of the IEEE*, vol. 88, no. 4, pp. 569–587, 2000.
- [4] D. Donoho and J. Tanner, “Precise Undersampling Theorems,” *Proceedings of the IEEE*, vol. 98, no. 6, pp. 913–924, May, 2010.
- [5] Z.Chen, G. Gokeda, Y.Yu, “Introduction to Direction of Arrival Estimation”, Artech House, 2010.
- [6] “802.15.4a-2007: IEEE Standard for Information Technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - specific requirement. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs),” 2007.
- [7] J. A. Tropp, “Greed is Good: Algorithmic Results for Sparse Approximation,” *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2231–2242, Oct, 2004.

- [8] F. M. Naini, R. Gribonval, L. Jacques, and P. Vandergheynst, "Compressive sampling of pulse trains: Spread the spectrum!" 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 2877–2880, Apr. 2009.
- [9] Arun Agarwal, Kabita Agarwal. The Next Generation Mobile Wireless Cellular Networks – 4G and Beyond. American Journal of Electrical and Electronic Engineering. 2014; 2(3):92-97.
- [10] A. Saleh and R. Valenzuela, "A Statistical Model for Indoor Multipath Propagation," IEEE JSAC, Vol. SAC-5, No. 2, Feb. 1987, pp. 128-137.
- [11] Q. H. Spencer, B. D. Jeffs, M. A. Jense, A. L. Swindlehurst, "Modeling the Statistical Time and Angle of Arrival Characteristics of an Indoor Multipath Channel," IEEE JSAC, Vol. 18, No. 3, March 2000.
- [12] Huanan Yu and Shuxu Guo. Compressed Sensing: Ultra-Wideband Channel Estimation Based on FIR Filtering Matrix, Ultra Wideband - Current Status and Future Trends, Dr. Mohammad Matin (Ed), 2012.
- [13] E. L. Targarona, "Sparse Channel Estimation based on Compressed Sensing Theory for UWB systems," M.Sc. Thesis, MERIT TSC UPC, Barcelona, November 2010.
- [14] Y. Won and S. Yoon, "Compressive sensing-based channel bandwidth improvement in optical wireless orthogonal frequency division multiplexing link using visible light emitting diode," Opt. Express 22, 19990-19999, 2014.

- [15] E. J. Candes and T. Tao, "Near Optimal Signal Recovery from Random Projections: Universal Encoding Strategies," *IEEE Transactions on Information Theory*, vol. 52, no. 12, pp. 5406–5425, Dec, 2006.
- [16] E. J. Candes, "Compressive Sampling," *Proceedings of the International Congress of Mathematicians (ICM)*, vol. 3, pp. 1433–1452, 2006.
- [17] E. J. Candes, J. Romberg, and T. Tao, "Stable Signal Recovery from Incomplete and Inaccurate Measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207–1223, 2006.
- [18] D. Donoho, "Compressed Sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, Apr, 2006.
- [19] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
- [20] A. Rontogiannis and K. Berberidis, "Bandwidth Efficient Transmission through Sparse Channel using Parametric Channel-Estimation-Based DFE," *Proc. Inst. Electr. Eng. Commun.*, vol. 152, no. 2, pp. 251–256, 2005.
- [21] S. G. Mallat and Z. Zhang, "Matching Pursuits with Time-Frequency Dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, Dec, 1993.
- [22] J. A. Tropp and A. C. Gilbert, "Signal Recovery from Random Measurements Via Orthogonal Matching Pursuit," *IEEE Transactions on Information Theory*, vol. 53, no. 12, pp. 4655–4666, Dec, 2007.

- [23] D. L. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck. Sparse solution of underdetermined linear equations by stagewise Orthogonal Matching Pursuit (StOMP). Submitted for publication, 2007.
- [24] D. Needell and R. Vershynin. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. Submitted for publication, October 2007.
- [25] D. Needell, J.A. Tropp, CoSaMP: Iterative signal recovery from incomplete and inaccurate samples, ACM Technical Report 2008-01, California Institute of Technology, Pasadena, 2008.
- [26] Liu Xuebin, Wei Gang, and Ji Fei, “A Design of Dual Uniform Linear Array Based on the Method of DOA Estimation with FFT,” Journal of South China University of Technology: Natural Science Edition, vol. 32, pp. 5-10, April 2004.
- [27] Yang Xueya, Chen Baixiao, and Zhu Gen-sheng, “Novel method for DOA estimation of coherent signals by eigen-subspace iteration,” Journal of Xidian University: Natural Science Edition, vol. 37, pp. 242-247, April 2010.
- [28] R. Roy, “ESPRIT-Estimation of signal Parameters via Rotational Invariance Techniques,” Ph.D. dissertation, Stanford Univ., 1987.
- [29] R. O. Schmidt, “Multiple emitter location and signal parameter estimation” IEEE Transactions on Antennas and Propagation, vol. AP-34, No. 3, pp. 276-280, March 1986.
- [30] Friedlander, B., 1993. The Root-MUSIC algorithm for direction finding with interpolated arrays. Signal Process. 30 (1), 15–29.

- [31] Classical and Modern Direction-of-Arrival Estimation, July 2009, Academic Press.
- [32] Akaike, H., "A new look at the statistical model identification." IEEE Transactions on Automatic Control, Dec. 1974, Vol. 19, No. 6, pp. 716-723.
- [33] J. Rissanen, Paper: Modeling by shortest data description, Automatica (Journal of IFAC), v.14 n.5, p.465-471, Sept. 1978.
- [34] S. Haykin, J. P. Reilly, V. Kezys and E. Vertatschitsch,, "Some aspects of array signal processing," Radar and Signal Processing, IEE Proceedings F, vol.139, no.1, pp.1-26, Feb 1992.
- [35] D. Malioutov, M. Cetin and A. S. Willsky, "A sparse signal reconstruction perspective for source localization with sensor arrays", IEEE Trans. Signal Process., vol. 53, no. 8, pp. 3010-3022, 2005.
- [36] Joseph Lardies, Hua Ma, Marc Berthillier. Source localization using a sparse representation of sensor measurements. Societe Francaise d'Acoustique. Acoustics 2012, Apr 2012, Nantes, France. 2012.
- [37] V. Cevher, A.C. Gurbuz, James H. McClellan, and R. Chellappa, "Compressive wireless arrays for bearing estimation," in submitted to ICASSP, 2008.
- [38] Lloyd, Stuart P. "Least Squares Quantization in PCM." IEEE Transactions on Information Theory. Vol. 28, 1982, pp. 129–137.
- [39] D. Shutin, "Clustering wireless channel impulse responses in angular delay domain", IEEE 5th Workshop on Signal Processing Advances in Wireless Communications, pp. 253-257, Jul. 2004.

[40] Ahmed Syed Faraz, “A Compressed Sensing Approach To Channel Estimation For Impulse-Radio Ultra-Wideband (IR-UWB) Communication”, M.S. Thesis, King Fahad University Of Petroleum And Minerals, Saudi Arabia, 2011.

APPENDIX

CODE LISTING

1. Ray_Tracing.m

```
clear all; close all;
MaxLength = 10000;
Bin_Size = 10; % cm
Left = 0; Right = 10; Bottom = 0; Top = 8;
Bx = [3 6]; % Don't make line exactly vertical
By = [2 6]; % Make it have a high but finite slope
Reflections = 16; RC = 0.75; Th = 0.01;
Angle_Th = 0.6; % 0.3 to 0.5
Tx = 1.5; Ty = 5.5;
R1x = 8; R1y = 4.6;
R2x = 8; R2y = 4.2;
R3x = 8; R3y = 3.8;
R4x = 8; R4y = 3.4;
Reflections1 = zeros(MaxLength,50); Reflections2 = zeros(MaxLength,50); Reflections3 =
zeros(MaxLength,50); Reflections4 = zeros(MaxLength,50);
Path_Length1 = zeros(MaxLength,50); Path_Length2 = zeros(MaxLength,50); Path_Length3 =
zeros(MaxLength,50); Path_Length4 = zeros(MaxLength,50);
DoA1 = zeros(MaxLength,50); DoA2 = zeros(MaxLength,50); DoA3 =
zeros(MaxLength,50); DoA4 = zeros(MaxLength,50);
At_Theta1 = zeros(MaxLength,50); At_Theta2 = zeros(MaxLength,50); At_Theta3 =
zeros(MaxLength,50); At_Theta4 = zeros(MaxLength,50);
C1 = zeros(MaxLength,1); C2 = zeros(MaxLength,1); C3 =
zeros(MaxLength,1); C4 = zeros(MaxLength,1);
for theta = 0.09:0.05:360

    Line = zeros(Reflections,2);
    Path = 0;
    if (theta > 0) && (theta <= 90)
        angle = theta;
        q = 1;
    elseif (theta > 90) && (theta <= 180)
        angle = 180 - theta;
        q = 2;
    elseif (theta > 180) && (theta <= 270)
        angle = theta - 180;
        q = 3;
    elseif (theta > 270) && (theta <= 360)
        angle = 360 - theta;
        q = 4;
    end

    Line(1,:) = [Tx Ty];

    BI = 0;

    for i = 2:Reflections
        if (q == 1)
            phi = angle;
            L = [Line(i-1,1)+10*cosd(phi) Line(i-1,2)+10*sind(phi)];
            M1 = (L(2) - Line(i-1,2))/(L(1) - Line(i-1,1));
            c1 = -M1*Line(i-1,1) + Line(i-1,2);
            M2 = (By(2) - By(1))/(Bx(2) - Bx(1));
            c2 = -M2*Bx(1) + By(1);
            IP = ([-M1 1; -M2 1]\[c1 c2]');
        end
    end
end
```

```

        if (IP(1) > min(Bx)) && (IP(1) < max(Bx)) && (IP(2) > min(By)) && (IP(2) <
max(By)) && (IP(1) > min([Line(i-1,1) L(1)])) && (IP(1) < max([Line(i-1,1) L(1)])) &&
(IP(2) > min([Line(i-1,2) L(2)])) && (IP(2) < max([Line(i-1,2) L(2)])) && (BI == 0)
        BI = 1;
        Line(i,:) = IP;
        phi = mod(2*atand((By(2)-By(1))/(Bx(2)-Bx(1)))-phi + 360,360);
        if phi > 270
            angle = 360 - phi;
            q = 4;
        elseif phi > 180
            angle = phi - 180;
            q = 3;
        elseif phi > 90
            angle = 180 - phi;
            q = 2;
        else
            angle = phi;
            q = 1;
        end
    else
        BI = 0;
        LE1 = [Right Line(i-1,2)+tand(angle)*(Right-Line(i-1,1))];
        LE2 = [Line(i-1,1)+1/tand(angle)*(Top-Line(i-1,2)) Top];
        if ((LE1(1) - Line(i-1,1))^2 + (LE1(2) - Line(i-1,2))^2 <= ((LE2(1) -
Line(i-1,1))^2 + (LE2(2) - Line(i-1,2))^2)
            Line(i,:) = LE1;
            q = 2;
        else
            Line(i,:) = LE2;
            q = 4;
        end
    end
elseif (q == 2)
    phi = 180 - angle;
    L = [Line(i-1,1)+10*cosd(phi) Line(i-1,2)+10*sind(phi)];
    M1 = (L(2) - Line(i-1,2))/(L(1) - Line(i-1,1));
    c1 = -M1*Line(i-1,1) + Line(i-1,2);
    M2 = (By(2) - By(1))/(Bx(2) - Bx(1));
    c2 = -M2*Bx(1) + By(1);
    IP = ([-M1 1; -M2 1]\[c1 c2]');
    if (IP(1) > min(Bx)) && (IP(1) < max(Bx)) && (IP(2) > min(By)) && (IP(2) <
max(By)) && (IP(1) > min([Line(i-1,1) L(1)])) && (IP(1) < max([Line(i-1,1) L(1)])) &&
(IP(2) > min([Line(i-1,2) L(2)])) && (IP(2) < max([Line(i-1,2) L(2)])) && (BI == 0)
        BI = 1;
        Line(i,:) = IP;
        phi = mod(2*atand((By(2)-By(1))/(Bx(2)-Bx(1)))-phi + 360,360);
        if phi > 270
            angle = 360 - phi;
            q = 4;
        elseif phi > 180
            angle = phi - 180;
            q = 3;
        elseif phi > 90
            angle = 180 - phi;
            q = 2;
        else
            angle = phi;
            q = 1;
        end
    else
        BI = 0;
        LE1 = [Left Line(i-1,2)+tand(angle)*(Line(i-1,1)-Left)];
        LE2 = [Line(i-1,1)-1/tand(angle)*(Top-Line(i-1,2)) Top];
        if ((LE1(1) - Line(i-1,1))^2 + (LE1(2) - Line(i-1,2))^2 <= ((LE2(1) -
Line(i-1,1))^2 + (LE2(2) - Line(i-1,2))^2)
            Line(i,:) = LE1;
            q = 1;
        else
            Line(i,:) = LE2;
            q = 3;
        end
    end
end

```

```

end
elseif (q == 3)
    phi = 180 + angle;
    L = [Line(i-1,1)+10*cosd(phi) Line(i-1,2)+10*sind(phi)];
    M1 = (L(2) - Line(i-1,2))/(L(1) - Line(i-1,1));
    c1 = -M1*Line(i-1,1) + Line(i-1,2);
    M2 = (By(2) - By(1))/(Bx(2) - Bx(1));
    c2 = -M2*Bx(1) + By(1);
    IP = ([-M1 1; -M2 1]\[c1 c2]')';
    if (IP(1) > min(Bx)) && (IP(1) < max(Bx)) && (IP(2) > min(By)) && (IP(2) <
max(By)) && (IP(1) > min([Line(i-1,1) L(1)])) && (IP(1) < max([Line(i-1,1) L(1)])) &&
(IP(2) > min([Line(i-1,2) L(2)])) && (IP(2) < max([Line(i-1,2) L(2)])) && (BI == 0)
        BI = 1;
        Line(i,:) = IP;
        phi = mod(2*atand((By(2)-By(1))/(Bx(2)-Bx(1)))-phi + 360,360);
        if phi > 270
            angle = 360 - phi;
            q = 4;
        elseif phi > 180
            angle = phi - 180;
            q = 3;
        elseif phi > 90
            angle = 180 - phi;
            q = 2;
        else
            angle = phi;
            q = 1;
        end
    end
else
    BI = 0;
    LE1 = [Left Line(i-1,2)-tand(angle)*(Line(i-1,1)-Left)];
    LE2 = [Line(i-1,1)-1/tand(angle)*(Line(i-1,2)-Bottom) Bottom];
    if ((LE1(1) - Line(i-1,1))^2 + (LE1(2) - Line(i-1,2))^2 <= ((LE2(1) -
Line(i-1,1))^2 + (LE2(2) - Line(i-1,2))^2)
        Line(i,:) = LE1;
        q = 4;
    else
        Line(i,:) = LE2;
        q = 2;
    end
end
end
elseif (q == 4)
    phi = -angle;
    L = [Line(i-1,1)+10*cosd(phi) Line(i-1,2)+10*sind(phi)];
    M1 = (L(2) - Line(i-1,2))/(L(1) - Line(i-1,1));
    c1 = -M1*Line(i-1,1) + Line(i-1,2);
    M2 = (By(2) - By(1))/(Bx(2) - Bx(1));
    c2 = -M2*Bx(1) + By(1);
    IP = ([-M1 1; -M2 1]\[c1 c2]')';
    if (IP(1) > min(Bx)) && (IP(1) < max(Bx)) && (IP(2) > min(By)) && (IP(2) <
max(By)) && (IP(1) > min([Line(i-1,1) L(1)])) && (IP(1) < max([Line(i-1,1) L(1)])) &&
(IP(2) > min([Line(i-1,2) L(2)])) && (IP(2) < max([Line(i-1,2) L(2)])) && (BI == 0)
        BI = 1;
        Line(i,:) = IP;
        phi = mod(2*atand((By(2)-By(1))/(Bx(2)-Bx(1)))-phi + 360,360);
        if phi > 270
            angle = 360 - phi;
            q = 4;
        elseif phi > 180
            angle = phi - 180;
            q = 3;
        elseif phi > 90
            angle = 180 - phi;
            q = 2;
        else
            angle = phi;
            q = 1;
        end
    end
else
    BI = 0;
    LE1 = [Right Line(i-1,2)-tand(angle)*(Right-Line(i-1,1))];

```

```

LE2 = [Line(i-1,1)+1/tand(angle)*(Line(i-1,2)-Bottom) Bottom];
if ((LE1(1) - Line(i-1,1))^2 + (LE1(2) - Line(i-1,2))^2) <= ((LE2(1) -
Line(i-1,1))^2 + (LE2(2) - Line(i-1,2))^2)
    Line(i,:) = LE1;
    q = 3;
else
    Line(i,:) = LE2;
    q = 1;
end
end
end

LL = sqrt((Line(i,2) - Line(i-1,2))^2 + (Line(i,1) - Line(i-1,1))^2);

if ((abs((Line(i,2) - Line(i-1,2))*R1x - (Line(i,1) - Line(i-1,1))*R1y +
Line(i,1)*Line(i-1,2) - Line(i,2)*Line(i-1,1))/LL < Th) && (R1x > min(Line(i-
1,1),Line(i,1))) && (R1x < max(Line(i-1,1),Line(i,1))) && (R1y > min(Line(i-
1,2),Line(i,2))) && (R1y < max(Line(i-1,2),Line(i,2))))
    Bin = floor((100/Bin_Size)*(Path+sqrt((R1x - Line(i-1,1))^2 + (R1y -
Line(i-1,2))^2)));
    if C1(Bin) == 0
        C1(Bin) = 1;
        Reflections1(Bin,1) = i - 2;
        Path_Length1(Bin,1) = Path+sqrt((R1x - Line(i-1,1))^2 + (R1y - Line(i-
1,2))^2);
        DoA1(Bin,1) = mod(180/pi*atan2(Line(i-1,2)-Line(i,2), Line(i-1,1)-
Line(i,1)),360);
        At_Thetal(Bin,1) = theta;
    else
        Different = 1;
        for z = 1:C1(Bin)
            if abs(At_Thetal(Bin,z) - theta) < Angle_Th
                Different = 0;
            end
        end
        if Different == 1
            C1(Bin) = C1(Bin) + 1;
            Reflections1(Bin,C1(Bin)) = i - 2;
            Path_Length1(Bin,C1(Bin)) = Path+sqrt((R1x - Line(i-1,1))^2 + (R1y
- Line(i-1,2))^2);
            DoA1(Bin,C1(Bin)) = mod(180/pi*atan2(Line(i-1,2)-Line(i,2), Line(i-
1,1)-Line(i,1)),360);
            At_Thetal(Bin,C1(Bin)) = theta;
        end
    end
end

if ((abs((Line(i,2) - Line(i-1,2))*R2x - (Line(i,1) - Line(i-1,1))*R2y +
Line(i,1)*Line(i-1,2) - Line(i,2)*Line(i-1,1))/LL < Th) && (R2x > min(Line(i-
1,1),Line(i,1))) && (R2x < max(Line(i-1,1),Line(i,1))) && (R2y > min(Line(i-
1,2),Line(i,2))) && (R2y < max(Line(i-1,2),Line(i,2))))
    Bin = floor((100/Bin_Size)*(Path+sqrt((R2x - Line(i-1,1))^2 + (R2y -
Line(i-1,2))^2)));
    if C2(Bin) == 0
        C2(Bin) = 1;
        Reflections2(Bin,1) = i - 2;
        Path_Length2(Bin,1) = Path+sqrt((R2x - Line(i-1,1))^2 + (R2y - Line(i-
1,2))^2);
        DoA2(Bin,1) = mod(180/pi*atan2(Line(i-1,2)-Line(i,2), Line(i-1,1)-
Line(i,1)),360);
        At_Theta2(Bin,1) = theta;
    else
        Different = 1;
        for z = 1:C2(Bin)
            if abs(At_Theta2(Bin,z) - theta) < Angle_Th
                Different = 0;
            end
        end
        if Different == 1
            C2(Bin) = C2(Bin) + 1;
            Reflections2(Bin,C2(Bin)) = i - 2;

```

```

- Line(i-1,2))^2);
    Path_Length2(Bin,C2(Bin)) = Path+sqrt((R2x - Line(i-1,1))^2 + (R2y
- Line(i-1,2))^2);
    DoA2(Bin,C2(Bin)) = mod(180/pi*atan2(Line(i-1,2)-Line(i,2), Line(i-
1,1)-Line(i,1)),360);
    At_Theta2(Bin,C2(Bin)) = theta;
    end
end
end

    if ((abs((Line(i,2) - Line(i-1,2))*R3x - (Line(i,1) - Line(i-1,1))*R3y +
Line(i,1)*Line(i-1,2) - Line(i,2)*Line(i-1,1))/LL < Th) && (R3x > min(Line(i-
1,1),Line(i,1))) && (R3x < max(Line(i-1,1),Line(i,1))) && (R3y > min(Line(i-
1,2),Line(i,2))) && (R3y < max(Line(i-1,2),Line(i,2))))
    Bin = floor((100/Bin_Size)*(Path+sqrt((R3x - Line(i-1,1))^2 + (R3y -
Line(i-1,2))^2)));
    if C3(Bin) == 0
        C3(Bin) = 1;
        Reflections3(Bin,1) = i - 2;
        Path_Length3(Bin,1) = Path+sqrt((R3x - Line(i-1,1))^2 + (R3y - Line(i-
1,2))^2);
        DoA3(Bin,1) = mod(180/pi*atan2(Line(i-1,2)-Line(i,2), Line(i-1,1)-
Line(i,1)),360);
        At_Theta3(Bin,1) = theta;
    else
        Different = 1;
        for z = 1:C3(Bin)
            if abs(At_Theta3(Bin,z) - theta) < Angle_Th
                Different = 0;
            end
        end
        if Different == 1
            C3(Bin) = C3(Bin) + 1;
            Reflections3(Bin,C3(Bin)) = i - 2;
            Path_Length3(Bin,C3(Bin)) = Path+sqrt((R3x - Line(i-1,1))^2 + (R3y
- Line(i-1,2))^2);
            DoA3(Bin,C3(Bin)) = mod(180/pi*atan2(Line(i-1,2)-Line(i,2), Line(i-
1,1)-Line(i,1)),360);
            At_Theta3(Bin,C3(Bin)) = theta;
        end
    end
end

    if ((abs((Line(i,2) - Line(i-1,2))*R4x - (Line(i,1) - Line(i-1,1))*R4y +
Line(i,1)*Line(i-1,2) - Line(i,2)*Line(i-1,1))/LL < Th) && (R4x > min(Line(i-
1,1),Line(i,1))) && (R4x < max(Line(i-1,1),Line(i,1))) && (R4y > min(Line(i-
1,2),Line(i,2))) && (R4y < max(Line(i-1,2),Line(i,2))))
    Bin = floor((100/Bin_Size)*(Path+sqrt((R4x - Line(i-1,1))^2 + (R4y -
Line(i-1,2))^2)));
    if C4(Bin) == 0
        C4(Bin) = 1;
        Reflections4(Bin,1) = i - 2;
        Path_Length4(Bin,1) = Path+sqrt((R4x - Line(i-1,1))^2 + (R4y - Line(i-
1,2))^2);
        DoA4(Bin,1) = mod(180/pi*atan2(Line(i-1,2)-Line(i,2), Line(i-1,1)-
Line(i,1)),360);
        At_Theta4(Bin,1) = theta;
    else
        Different = 1;
        for z = 1:C4(Bin)
            if abs(At_Theta4(Bin,z) - theta) < Angle_Th
                Different = 0;
            end
        end
        if Different == 1
            C4(Bin) = C4(Bin) + 1;
            Reflections4(Bin,C4(Bin)) = i - 2;
            Path_Length4(Bin,C4(Bin)) = Path+sqrt((R4x - Line(i-1,1))^2 + (R4y
- Line(i-1,2))^2);
            DoA4(Bin,C4(Bin)) = mod(180/pi*atan2(Line(i-1,2)-Line(i,2), Line(i-
1,1)-Line(i,1)),360);
            At_Theta4(Bin,C4(Bin)) = theta;
        end
    end
end

```

```

        end
    end
end

    Path = Path + LL;
end

    plot([Bx(1) Bx(2)], [By(1) By(2)], 'g', Line(:,1), Line(:,2), 'm', Tx, Ty, 'r^', [R1x R2x
R3x R4x], [R1y R2y R3y R4y], 'xb', 'LineWidth', 2, 'MarkerSize', 10);
    box on; grid on;
    axis([-1 11 -1 9]);
    rectangle('Position', [Left Bottom Right Top], 'LineWidth', 4, 'LineStyle', '--');
    legend('Partition', 'Ray Path', 'TX', 'RXs'); legend('Location', 'NorthEastOutside');
    drawnow;
    theta
end

```

2. Angle_Delay_Domain_Clustering.m

```

WindowsLength = 1;
d = 0.4;
SNR = 70;
Effective_Range = 118:(250);
% Channel Impulse Responses for the 4 Antennas
MaxLength = 10000;
CIR1 = zeros(1, MaxLength); CIR2 = CIR1; CIR3 = CIR1; CIR4 = CIR1;
f = 3e8; lambda = 3e8/f;
Theta1 = Path_Length1./lambda; Theta2 = Path_Length2./lambda; Theta3 =
Path_Length3./lambda; Theta4 = Path_Length4./lambda;
for Ant_ID=1:MaxLength
    for Ray_ID=1:C1(Ant_ID)
        CIR1(Ant_ID) = CIR1(Ant_ID) + sqrt((RC)^Reflections1(Ant_ID, Ray_ID) *
1/(Path_Length1(Ant_ID, Ray_ID)^2)*exp(1i*deg2rad(Theta1(Ant_ID, Ray_ID))));
    end
    for Ray_ID=1:C2(Ant_ID)
        CIR2(Ant_ID) = CIR2(Ant_ID) + sqrt((RC)^Reflections2(Ant_ID, Ray_ID) *
1/(Path_Length2(Ant_ID, Ray_ID)^2)*exp(1i*deg2rad(Theta2(Ant_ID, Ray_ID))));
    end
    for Ray_ID=1:C3(Ant_ID)
        CIR3(Ant_ID) = CIR3(Ant_ID) + sqrt((RC)^Reflections3(Ant_ID, Ray_ID) *
1/(Path_Length3(Ant_ID, Ray_ID)^2)*exp(1i*deg2rad(Theta3(Ant_ID, Ray_ID))));
    end
    for Ray_ID=1:C4(Ant_ID)
        CIR4(Ant_ID) = CIR4(Ant_ID) + sqrt((RC)^Reflections4(Ant_ID, Ray_ID) *
1/(Path_Length4(Ant_ID, Ray_ID)^2)*exp(1i*deg2rad(Theta4(Ant_ID, Ray_ID))));
    end
end
m = 4;
No_of_Ant = m;
time = (1:MaxLength)/Bin_Size/3e8; time = time(Effective_Range);
h1 = CIR1(Effective_Range); h2 = CIR2(Effective_Range); h3 = CIR3(Effective_Range); h4
= CIR4(Effective_Range);
h1 = h1./max(abs(h1)); h2 = h2./max(abs(h2)); h3 = h3./max(abs(h3)); h4 =
h4./max(abs(h4));
h = [h1; h2; h3; h4]';
Noisy_h = h + (1/(0.35*SNR))*(randn(size(h))+1i*randn(size(h)));
clf, subplot(3,1,1);
xMat = time; xMatLinear = repmat((1:No_of_Ant), length(Effective_Range), 1); yMat =
repmat((1:length(Effective_Range)), No_of_Ant, 1); zMat = abs(Noisy_h);
surface(xMatLinear, yMat, 10.*zMat, 'LineStyle', '-
', 'MeshStyle', 'column', 'FaceColor', 'none', 'EdgeColor', 'interp', 'LineWidth', 2);
xlabel('Antenna Index'), ylabel('Delay (ns)'), zlabel('Recieved Power |H| (dB)');
axis([-1 No_of_Ant+1 0 length(Effective_Range) -0.5 +11*max(max(abs(h)))]); axis
manual;
%ax = gca; ax.PlotBoxAspectRatio = [1 1.2 0.4];
grid on, rotate3d on; view(-90, 60); set(gca, 'YDir', 'reverse');
leg txt = sprintf('1x%d SIMO Channel Impulse Responses', No_of_Ant);

```

```

title(leg_txt);set(gca,'fontsize',9); box on; view(-80,40); %legend('Antenna CIR');
set(gca,'fontsize',10); box on;
CC1 = C1(Effective_Range); CC2 = C2(Effective_Range); CC3 = C3(Effective_Range); CC4 =
C4(Effective_Range);
idx1=find(CC1>0); idx2=find(CC2>0); idx3=find(CC3>0); idx4=find(CC4>0);
idx = 1:min([length(idx1) length(idx2) length(idx3) length(idx4)]);
DDoA1 = DoA1(Effective_Range); DDoA2 = DoA2(Effective_Range); DDoA3 =
DoA3(Effective_Range); DDoA4 = DoA4(Effective_Range);

Res_DoA1 = 50.*(mean(DoA1'))'; Res_DoA1 = Res_DoA1(Effective_Range);
Res_DoA2 = 50.*(mean(DoA2'))'; Res_DoA2 = Res_DoA2(Effective_Range);
Res_DoA3 = 50.*(mean(DoA3'))'; Res_DoA3 = Res_DoA3(Effective_Range);
Res_DoA4 = 50.*(mean(DoA4'))'; Res_DoA4 = Res_DoA4(Effective_Range);
Res_DoA = [Res_DoA1 Res_DoA2 Res_DoA3 Res_DoA4];
Ref_DoA = [Res_DoA1(idx1(idx)) Res_DoA2(idx2(idx)) Res_DoA3(idx3(idx))
Res_DoA4(idx4(idx))];
All_h = [(h1(idx1(idx)))' (h2(idx2(idx)))' (h3(idx3(idx)))' (h4(idx4(idx)))'];
Ampls = abs(All_h);
Phass = rad2deg([phase(h1(idx1(idx)))' phase(h2(idx2(idx)))' phase(h3(idx3(idx)))'
phase(h4(idx4(idx)))']);
www = 2;
for i=1:length(All_h)-www
    for j=1:3
        dist = zeros(1,www); r=0;
        for k=0:www
            Re1 = real(All_h(i,j)); Imagin1 = imag(All_h(i,j));
            Re2 = real(All_h(i+k,j+1)); Imagin2 = imag(All_h(i+k,j+1));
            r=r+1; dist(r) = sqrt((Re2-Re1)^2 + (Imagin2-Imagin1)^2);
        end
        taken = i + find( dist == min(dist) );
        idxidxidx(i,j) = taken;
    end
end
Associations = [(1:length(All_h)-www)' idxidxidx];
Phass_New = Phass(Associations); Ampls_New = Ampls(Associations); All_h_New =
All_h(Associations);
Final_h = zeros(size(h));
for o=1:numel(Associations)/length(Associations)
    Final_h(idx1(o),:) = All_h_New(o,:);
end
MaxLength = length(Effective_Range);
% Unitary ESPRIT for h(i,:)
G = Final_h'; cont = 0;
SpaceX = 1:MaxLength; SpaceY = NaN(1,MaxLength); SpaceZ = zeros(1,MaxLength); jk=1;
pl=1;
for k=0:WindowsLength:(MaxLength-WindowsLength) % (k+1:k+WindowsLength)
    g = G(:,(k+1:k+WindowsLength));
    R=(1/m)*(g*g'); Est_DOA=ESPRIT(R,d,1,1); SpaceY(k+1)=Est_DOA;
    if (SpaceY(k+1)~=0)
        jk = k + 1;
        Required_DoAs(pl) = SpaceY(k+1);
        pl = pl + 1;
    end
    if (SpaceY(k+1)==0)
        SpaceY(k+1) = Required_DoAs(pl) + (1/(SNR))*270*randn();
        cont = cont + 1;
    end
    SpaceZ(k+1) = mean(var(abs(g)));
    %sum(sqrt(abs(g)
    %mean(abs(g)
    %max(abs(g)
    %var(abs(g)
end
SpaceY = 2*SpaceY-90;
No_of_DOAs_Expected = length(Required_DoAs);
%% 2D Clustering:
%Extract the non-NaN elements of SpaceY
indx = find(~isnan(SpaceY));
True_SpaceX = SpaceX(indx); True_SpaceY = SpaceY(indx); True_SpaceZ = SpaceZ(indx);
% No of discarded sample is: (MaxLength-length(indx))
No_of_DOAs_Expected = 6;%3*AIC(R,MaxLength,1);

```

```

Space2D = [True_SpaceX; True_SpaceY]';
[Clusters_IDs, Centroids] = kMeansCluster(Space2D, No_of_DOAs_Expected);

subplot(3,1,2);
IDX = Clusters_IDs; cent_count = max(IDX); Colors = hsv(cent_count); Legends =
{};
for i=0:max(IDX)
    Xi = Space2D(IDX==i,:);
    if (i~=0)
        Legends{end+1} = strcat(num2str(Centroids(i,2),'%-03.2f '), '^{\circ}');
    end
    if ~isempty(Xi)
        plot(Xi(:,1),Xi(:,2),'x','MarkerSize',7,'Color',Colors(i,:));
    end
    hold on;
end
plot(NaN,'ok','MarkerSize',10); plot(NaN,'sk','MarkerSize',11);
Legends{end+1} = 'Centroids';
Legends{end+1} = strcat('Mean DOA = ',sprintf('%3.2f%c',mean(True_SpaceY),char(176)));
legend(Legends); legend('Location','NorthEastOutside');
% Draw the center of the cluster
for i=1:No_of_DOAs_Expected
    x_cent = Centroids(i); y_cent = Centroids(i+No_of_DOAs_Expected);
    Colors = hsv(length(Centroids)); Color = Colors(i,:);
    plot(x_cent,y_cent,'o','MarkerSize',12,'Color',Color);
    plot(x_cent,y_cent,'x','MarkerSize',12,'Color',Color);
end
plot((MaxLength/2),floor(mean(True_SpaceY)),'xk','MarkerSize',10);
plot((MaxLength/2),floor(mean(True_SpaceY)),'sk','MarkerSize',11);
axis([0 MaxLength -180 180]),xlabel('Delay (ns)'),ylabel('DOA (degrees)');
title('Clustering of DOAs using K-Means Algorithm'); grid on; hold off; box on;

Confidence = 100*((1 - (length(indx)/MaxLength))/WindowsLength);
%% Alternative 2D Clustering (considering Power as points density):
Pow = SpaceZ; Pow = 10.* (Pow./max(max(Pow))); ZoomRes = 5; MaxLen = length(Pow);
Neu_SpaceY = NaN(1,ZoomRes*MaxLen);
for i=1:length(Pow)
    if ((Pow(i)>=00) && (Pow(i)<02))
        temp_SpaceY = [ NaN NaN SpaceY(i) NaN NaN ];
    end
    if ((Pow(i)>=02) && (Pow(i)<04))
        temp_SpaceY = [ NaN SpaceY(i) NaN SpaceY(i) NaN ];
    end
    if ((Pow(i)>=04) && (Pow(i)<06))
        temp_SpaceY = [ NaN SpaceY(i) SpaceY(i) SpaceY(i) NaN ];
    end
    if ((Pow(i)>=06) && (Pow(i)<08))
        temp_SpaceY = [SpaceY(i) SpaceY(i) NaN SpaceY(i) SpaceY(i)];
    end
    if ((Pow(i)>=08) && (Pow(i)<100))
        temp_SpaceY = [SpaceY(i) SpaceY(i) SpaceY(i) SpaceY(i) SpaceY(i)];
    end
    if (isnan(Pow(i)))
        temp_SpaceY = [ NaN NaN NaN NaN NaN ];
    end
    Neu_SpaceY(1+ZoomRes*(i-1):i*ZoomRes) = temp_SpaceY;
end
Neu_SpaceX = (1:length(Neu_SpaceY)); Neu_indx = find(~isnan(Neu_SpaceY));
Neu_SpaceX = Neu_SpaceX(Neu_indx); Neu_SpaceY = Neu_SpaceY(Neu_indx);
Space2D_Extended = [Neu_SpaceX; Neu_SpaceY]';
[Clusters_IDs_Neu, Centroids_Neu] =
kMeansCluster(Space2D_Extended, No_of_DOAs_Expected);
MaxLen = MaxLength*ZoomRes;

subplot(3,1,[2 3])
IDX = Clusters_IDs_Neu; cent_count = max(IDX); Colors = hsv(cent_count); Legends =
{};
for i=0:max(IDX)
    Xi = Space2D_Extended(IDX==i,:);
    if (i~=0)

```

```

        Legends{end+1} = strcat(num2str(Centroids_Neu(i,2),'%-03.2f '),'^{\circ}
DOA_',num2str(i));
    end
    if ~isempty(Xi)
        plot(Xi(:,1)/ZoomRes,Xi(:,2),'x','MarkerSize',7,'Color',Colors(i,:));
    end
    hold on;
end
plot(NaN,'ok','MarkerSize',10); plot(NaN,'sk','MarkerSize',11);
Legends{end+1} = 'Centroids';
Legends{end+1} = strcat('Mean DOA = ',sprintf('%3.2f%c',mean(Neu_SpaceY),char(176)));
legend(Legends); legend('Location','NorthEastOutside');
% Draw the center of the cluster
for i=1:No_of_DOAs_Expected
    x_cent = Centroids_Neu(i); y_cent = Centroids_Neu(i+No_of_DOAs_Expected);
    Colors = hsv(length(Centroids_Neu)); Color = Colors(i,:);
    plot(x_cent/ZoomRes,y_cent,'o','MarkerSize',12,'Color',Color);
    plot(x_cent/ZoomRes,y_cent,'x','MarkerSize',12,'Color',Color);
end
plot((MaxLen/2)/ZoomRes,floor(mean(Neu_SpaceY)),'xk','MarkerSize',10);
plot((MaxLen/2)/ZoomRes,floor(mean(Neu_SpaceY)),'sk','MarkerSize',11);
axis([0 MaxLen/ZoomRes -90 90],xlabel('Delay (ns)'),ylabel('DOA (degrees)');
title('Clustering of DOAs using K-Means Algorithm (Power)'); grid on; hold off; box on;

```

3. Sparse_CIR_Reconstruction.m

```

clear all;
p = 50; % Channel Length
n = 40; % Number of observations (n/p is the compression ratio)
z = randn(n,1); % AWGN
No = 1; % Noise variance
zn = z*sqrt(No); % Noise level drawn from N(0,No)
X = randn(n,p);
for j = 1:p
    X(:,j) = X(:,j) ./ norm(X(:,j)); %orthogonalize columns
end
% Up to here, X is a "Uniform Spherical Ensemble" matrix (compression matrix)
LAMBDA = 0.0267; lambda = 2;
GAMMA = 16; gamma = 1.2;
sigma1_dB = 4.8; sigma2_dB = 4.8;
std_shadow = 1; LOSflag = 1;
T = 1; MaxLength = p;
[b, t] = ESV(LAMBDA,lambda,GAMMA,gamma,sigma1_dB,sigma2_dB,std_shadow,LOSflag,T,p);
b = 1000.*b';
% CS-ing
y = X*b + zn;
% Reconstruction Solver:
[bhatstep, ~, ~] = Solve_Stepwise_FDR(X,y,0);
[bhatMP, ~, ~] = Solve_MP(X, y, p, norm(zn), 0);
[bhatOMP, ~, ~] = Solve_O_MP(X, y, p, norm(zn), 0);
% fprintf('MSE for Stepwise Regression is (dB) ');
MSE1 = 10*log10( sum( abs(b-bhatstep).^2 )/p );
% fprintf('MSE for Matching Pursuit is (dB)');
MSE2 = 10*log10( sum( abs(b-bhatMP).^2 )/p );
% fprintf('MSE for Orthogonal Matching Pursuit is (dB)');
MSE3 = 10*log10( sum( abs(b-bhatOMP).^2 )/p );
% Plot Results
y = y./max(abs(y));
b = b./max(abs(b));
bhatstep = bhatstep./max(abs(bhatstep));
bhatMP = bhatMP./max(abs(bhatMP));
bhatOMP = bhatOMP./max(abs(bhatOMP));
subplot(2,1,1);
stem(y,'xr'),xlabel('CIR Taps'),ylabel('Amplitude');
axis([0 MaxLength -1 +1]), grid on;
title('Compressed CIR (Taps)');set(gca,'fontsize',8);
subplot(2,1,2);

```

```

stem(1:p,b,'xr')
hold on; plot(1:p,bhatstep,'s--g'); plot(1:p,bhatMP,'o--b'); plot(1:p,bhatOMP,'^--c');
hold off;
xlabel('Taps'),ylabel('Estimated Coefficient Value');
legend('True Coefficient Values','a. Forward Stepwise Regression Solution','b. Matching Pursuit Solution','c. Orthogonal Matching Pursuit');
title('Reconstruction of Sparse Signals');
axis([0 MaxLength -1 +1]), grid on; set(gca,'fontsize',8);

```

4. Direction_Finding.m

```

clear all;
%% Constants
Word_Length = 8; % No. of bits to generate (Frame length)
Tp = 0.1;
No_of_Samples = 3000;
RF_Freq = 0.3; % in MHz
Resolution = 0.1;
Ts = No_of_Samples/(Tp*Word_Length);
d = 0.25; %Inter-element seperation
M = 3; p = 2;
No = 1; SNR = 10; uu = 5;
% S-V Channel Model Simulation Parameters
N = 3; % Number of simulated channels
NLOS = 0; % No LOS component
b002 = 1; % Power of 1st ray of 1st cluster
sigma_x = 1; % Standard deviation of log-normal shadowing
LAMBDA_Cluster = 0.0233;
lambda_Ray = 1.4;
GAMMA_Cluster = 8;
gamma_Ray = 4;
L = 3; % Channel Length (No. of taps)
%% Tx Side:
%Baseband_Data = rand(1,Word_Length) > 0.5;
Baseband_Data = [1 0 1 0 1 0 1 0];
% Modulation:
Symbols = (2*Baseband_Data) - 1; % BPSK Mapping
Data_Samples = repmat(Symbols(:),1,Ts)';
Data_Samples = Data_Samples(:)'; % Up Conversion
Carrier = cos(2*pi*RF_Freq*[0:Resolution:No_of_Samples-Resolution]);
X = Carrier .* Data_Samples;
%% Channel Response
% 1. AWGN:
Noise = sqrt(No/2) * (randn(1,No_of_Samples/Tp) + 1i*randn(1,No_of_Samples/Tp));
AWGN = (SNR*No) * X + Noise;
% 2. S-V Channel Model:
[h,t,t0,np] =
SV_model_ct(LAMBDA_Cluster,lambda_Ray,GAMMA_Cluster,gamma_Ray,N,b002,sigma_x,NLOS);
H(1,:) = h(1:L,1); H(2,:) = h(1:L,2); H(3,:) = h(1:L,3);
% Limit all channels to L tap only
Y = zeros(M, (No_of_Samples/Tp)+(L-1));
for i=1:M
    Y(i,:) = conv(AWGN,H(i,:));
end
%% Rx Side:
%Y = Y(:,1:Ts); % Take one symbol only
% Direction Finding
K = length(Y(i,:));
R = (1/K) * (Y * Y');
[U ,SIGMA] = eig(R);
[SIGMA ,I] = sort(diag(SIGMA),1,'descend');
U_sorted = U(:,I);
Us = U_sorted(:,1:p); % Signal Supspace
Un = U_sorted(:,p+1:M); % Noise Supspace
% MUSIC:
Resolution = 0.1; angles = (-90:Resolution:90);
al = exp(-1i*2*pi*d*(0:M-1)'*sin((angles(:).')*pi/180));

```

```

MUSIC = zeros(1,length(a1));
for k=1:length(angles)
%Compute MUSIC "spectrum"
MUSIC(k) = uu.* (a1(:,k)'*a1(:,k))/(a1(:,k)'*(Un*Un')*a1(:,k));
%MUSIC(k) = uu.* 1/(a1(:,k)'*(Un*Un')*a1(:,k));
end
% Plot MUSIC Spectrum
plot(angles,abs(MUSIC));
grid on , axis([-90,90,0,max(abs(MUSIC))]) , xlabel('Angle in degrees');
title(sprintf('MUSIC Spectrum for %d signals (%d-element Array)',p,M));
% ESPRIT:
Es1 = Us(1:(end-1),:); Es2 = Us(2:end,:);
[V,sigma] = eig( [Es1';Es2'] * [Es1 Es2] );
[~,I] = sort(abs(diag(sigma)),'descend');
V = V(:,I); M12 = V(1:p,p+1:end); M22 = V(p+1:end,p+1:end);
EPSI = -(M12/M22);
lambdas = angle(eig(EPSI));
lambdas_clear = lambdas(lambdas<1 & lambdas>-1);
if (isempty(lambdas_clear))
    lambdas_clear = 0;
end
thetas = zeros(1,length(lambdas_clear)); Estimated_AOAs_ESPRIT = thetas;
for i=1:length(lambdas_clear)
    if (lambdas_clear(i)<0)
        thetas(i) = -lambdas_clear(i)/(2*pi*d);
        Estimated_AOAs_ESPRIT(i) = asind(thetas(i));
    else
        thetas(i) = lambdas_clear(i)/(2*pi*d);
        Estimated_AOAs_ESPRIT(i) = - asind(thetas(i));
    end
end
Estimated_AOAs_ESPRIT

```

Vitae

- Name: Awadalla SalahEIDin Awadalla Salih
- Nationality: Sudanese
- Date of Birth: February 13, 1991
- E-mail: awadalla333@hotmail.com
- Permanent Address: 64, 1, Shambat North, Khartoum Bahri, Khartoum, Sudan.

- Received Bachelor of Science (B. Sc.) with First Class in Electrical Engineering (Communication Engineering) from University of Khartoum, Khartoum, Sudan in October 2012.

- Joined King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia as Graduate Scholar in August 2014.

- Completed Master of Science (M. Sc.) in Electrical Engineering (First Class) in January 2017.

- Interested in Signal Processing Algorithms for Digital Communications Systems, Modulation and Coding, Efficient Algorithms for Wireless Data Transmission, UWB Systems, Mobile Communications Software Defined Radio and DSP/FPGA research areas.