

Spring 5-31-2019

## Blind source separation using dictionary learning over time-varying channels

Anushreya Ghosh  
*New Jersey Institute of Technology*

Follow this and additional works at: <https://digitalcommons.njit.edu/theses>



Part of the [Electrical and Electronics Commons](#)

---

### Recommended Citation

Ghosh, Anushreya, "Blind source separation using dictionary learning over time-varying channels" (2019).  
*Theses*. 1655.  
<https://digitalcommons.njit.edu/theses/1655>

This Thesis is brought to you for free and open access by the Electronic Theses and Dissertations at Digital Commons @ NJIT. It has been accepted for inclusion in Theses by an authorized administrator of Digital Commons @ NJIT. For more information, please contact [digitalcommons@njit.edu](mailto:digitalcommons@njit.edu).

## **Copyright Warning & Restrictions**

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

**Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation**

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

## **ABSTRACT**

### **BLIND SOURCE SEPARATION USING DICTIONARY LEARNING OVER TIME-VARYING CHANNELS**

**by  
Anushreya Ghosh**

Distributed sensors observe radio frequency (RF) sources over flat-fading channels. The activity pattern is sparse and intermittent in the sense that while the number of latent sources may be larger than the number of sensors, only a few of them may be active at any particular time instant. It is further assumed that the source activity is modeled by a Hidden Markov Model. In previous work, the Blind Source Separation (BSS) problem solved for stationary channels using Dictionary Learning (DL). This thesis studies the effect of time-varying channels on the performance of DL algorithms. The performance metric is the probability of detection, where a correct detection is the event that the estimated value of a source exceeds a threshold at a time instant when the true source is active. Using the probability of detection when the channels are stationary as a baseline, it is shown that there is significant degradation for time-varying channels and observation intervals much longer than the time coherence. Detection performance improves when the observation time is approximately equal to the time coherence. Performance is again degraded when the observation is shorter and there is not sufficient information for the DL algorithms to learn from.

**BLIND SOURCE SEPARATION USING  
DICTIONARY LEARNING OVER  
TIME-VARYING CHANNELS**

**by  
Anushreya Ghosh**

**A Thesis  
Submitted to the Faculty of  
New Jersey Institute of Technology  
in Partial Fulfillment of the Requirements for the Degree of  
Master of Science in Electrical Engineering**

**Department of Electrical and Computer Engineering**

**May 2019**

Blank Page

**APPROVAL PAGE**

**BLIND SOURCE SEPARATION USING  
DICTIONARY LEARNING OVER  
TIME-VARYING CHANNELS**

**Anushreya Ghosh**

---

Dr. Alexander M. Haimovich, Thesis Advisor Distinguished Professor of Electrical and Computer Engineering, NJIT	Date
--	------

---

Dr. Ali Abdi, Committee Member Professor of Electrical and Computer Engineering, NJIT	Date
--	------

---

Dr. Joerg Klierer, Committee Member Professor of Electrical and Computer Engineering, NJIT	Date
---	------

## **BIOGRAPHICAL SKETCH**

**Author:** Anushreya Ghosh  
**Degree:** Master of Science in Electrical Engineering  
**Date:** May 2019

### **Undergraduate and Graduate Education:**

- Master of Science in Electrical Engineering  
New Jersey Institute of Technology, Newark, NJ, 2019
- Bachelor of Science in Electronics and Communication Engineering  
Techno India, Salt Lake, Kolkata, India, 2013

**Major:** Electrical Engineering



"The good thing about science is that  
it's true whether or not  
you believe it."

- Neil deGrasse Tyson

## ACKNOWLEDGMENT

This work is made possible because of the guidance of Dr. Alexander Haimovich whose support throughout the research culminated in the fulfillment of the thesis. I am truly grateful for his continued mentoring as my advisor. His keen insight helped shape my work and approach towards engineering research. Special thanks are also due to Dr. Ali Abdi and Dr. Joerg Kliewer for their advice and support as well.

I am grateful to Mr. Annan Dong for all his help in not only introducing me to the world of Dictionary Learning but always having a smile for me all the times I interrupted his work to ask for help. Everyone at CWiP, NJIT also deserve heartiest thanks for being so supportive and the best team anyone could ask for. Ms. Kathleen Bosco went above and beyond helping all of us with any issue that came up in the lab and making us feel like a family.

This thesis would have been impossible to complete if not for Mirana Alam, whose friendship kept me sane during one of the toughest times of my life. Without her dragging me out of the lab when I was being driven crazy at work, I probably would have lost my mind.

Needless to say, a huge shout out goes to my family. I would be nowhere without them making the best version of me I ever could be. And even though, he is too little to understand, I am grateful to Pi for all the pure love he sends my way.

## TABLE OF CONTENTS

Chapter	Page
1 INTRODUCTION.....	1
2 SYSTEM MODEL .....	10
2.1 Signal Generation .....	11
2.2 Channel Generation .....	14
3 THEORY.....	19
3.1 Signal Estimation: LASSO Algorithm.....	21
3.2 Channel Estimation: MDU Algorithm.....	25
3.3 Modification for Time-Varying Channels.....	28
4 PERFORMANCE ANALYSIS.....	31
4.1 Sensitivity to Number of Observations.....	37
4.2 Sensitivity to Doppler Effect.....	44
4.3 Time Coherence and Number of Sampled Observations.....	51
5 CONCLUSION.....	59
6 REFERENCES.....	60

## LIST OF TABLES

Table	Page
4.1 Probability of detection when $P_{fa} = 0.2$ with changing SNR when number of observations = 1000 in a fixed channel.....	38
4.2 Probability of detection when $P_{fa} = 0.2$ with changing SNR when number of observations = 1000 in a time-varying channel.....	40
4.3 Probability of detection when $P_{fa} = 0.2$ with changing SNR when number of observations = 670 in a time-varying channel.....	41
4.4 Probability of detection when $P_{fa} = 0.2$ with changing SNR when number of observations = 330 in a time-varying channel.....	42
4.5 Probability of detection when $P_{fa} = 0.2$ for different Doppler frequencies $f_{m1} = 0.3$ kHz and $f_{m2} = 1$ kHz (SNR = 10 dB).....	49
4.6 Probability of detection when $P_{fa} = 0.2$ for different Doppler frequencies $f_{m1} = 0.3$ kHz and $f_{m2} = 1$ kHz (SNR = 20 dB).....	49
4.7 Probability of detection when $P_{fa} = 0.2$ for different Doppler frequencies $f_{m1} = 0.3$ kHz and $f_{m2} = 1$ kHz (SNR = 30 dB).....	49
4.8 Number of Observations in each individual segment based on maximum Doppler frequency.....	54

## LIST OF FIGURES

Figure	Page
1.1 Basic Communication System with $x(t)$ being transmitted from source through channel $h(t)$ in the presence of noise $z(t)$ . $y(t)$ is received and $\tilde{x}(t)$ is the estimated signal at receiver end.....	1
1.2 $N$ sources transmit signal $X$ over unknown channel $H$ , received by the fusion center for measurements via a total of $M$ sensors.....	2
2.1 State Transition Diagram for a 2-state Markov Model.....	12
2.2 Hidden Markov Model for any source $n$ .....	13
2.3 Doppler Power Spectrum for an unmodulated continuous wave carrier.....	16
2.4 Frequency domain implementation of a Rayleigh fading simulator.....	17
3.1 Functional Block Diagram of proposed algorithm.....	21
4.1 Source Activity Diagram of $N=30$ sources over $T=1000$ time samples. Average of 3 sources are active in each time sample. Average duration of transmission = 50 time samples.....	36
4.2 Power Spectral Density for carrier frequency $f_c = 1$ GHz and maximum Doppler frequency $f_m = 1$ kHz.....	37
4.3 Probability of detection for varying SNR (dB) when probability of false alarm is 0.2 for 1000 observations of a fixed channel.....	38
4.4 Probability of detection for varying SNR (dB) when probability of false alarm is 0.2 for 1000 observations in a time-varying channel.....	39
4.5 Probability of detection for varying SNR (dB) when probability of false alarm is 0.2 for 670 observations in a time-varying channel.....	40
4.6 Probability of detection for varying SNR (dB) when probability of false alarm is 0.2 for 330 observations in a time-varying channel.....	41

## LIST OF FIGURES

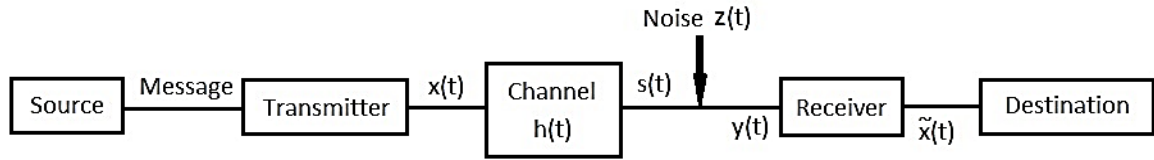
(Continued)

Figure	Page
4.7 Power Spectral Density for carrier frequency $f_c = 1$ GHz and maximum Doppler frequencies $f_{m1} = 0.3$ kHz and $f_{m2} = 1$ kHz.....	45
4.8 Performance for SNR = 10dB when $f_{m1} = 0.3$ kHz.....	46
4.9 Performance for SNR = 10dB when $f_{m2} = 1$ kHz .....	46
4.10 Performance for SNR = 20dB when $f_{m1} = 0.3$ kHz.....	47
4.11 Performance for SNR = 20dB when $f_{m2} = 1$ kHz.....	47
4.12 Performance for SNR = 30dB when $f_{m1} = 0.3$ kHz.....	48
4.13 Performance for SNR = 30dB when $f_{m2} = 1$ kHz.....	48
4.14 Performance for SNR = 30dB and $f_{m1} = 0.3$ kHz when number of segments = 500.....	55
4.15 Performance for SNR = 30dB and $f_{m2} = 1$ kHz when number of segments = 100.....	56

# CHAPTER 1

## INTRODUCTION

A communication system consists of certain components that can be broken into few general categories, mainly comprising of a source sending a message through the transmitter which travels through a channel where it undergoes changes due to the presence of noise and interference before reaching the receiver where the information is decoded to obtain the original message. Such a general communication is shown in figure 1.1 where the sent message is denoted by  $x(t)$ , the channel by  $h(t)$ , noise in the channel by  $z(t)$ , received information as  $y(t)$  and the demodulated message as  $\tilde{x}(t)$ .



**Figure 1.1** Basic Communication System with  $x(t)$  being transmitted from source through channel  $h(t)$  in the presence of noise  $z(t)$ .  $y(t)$  is received and  $\tilde{x}(t)$  is the estimated signal at receiver end.

In a communication system, a received signal can be represented as:

$$y(t) = s(t) + z(t) \quad (1.1)$$

Here,  $s(t)$  is the signal comprising information sent from the source ( $x(t)$ ) mixed with the channel ( $h(t)$ ).

$$s(t) = h(t)x(t) \quad (1.2)$$

The addition of  $z(t)$  is used to denote the presence of Additive White Gaussian Noise (AWGN) with zero mean and variance  $\sigma^2$ .

In the presence of multiple sources and multiple receivers (a Multi-Input Multi-Output (MIMO) system), we can denote equations (1.1) and (1.2) in their matrix forms as:

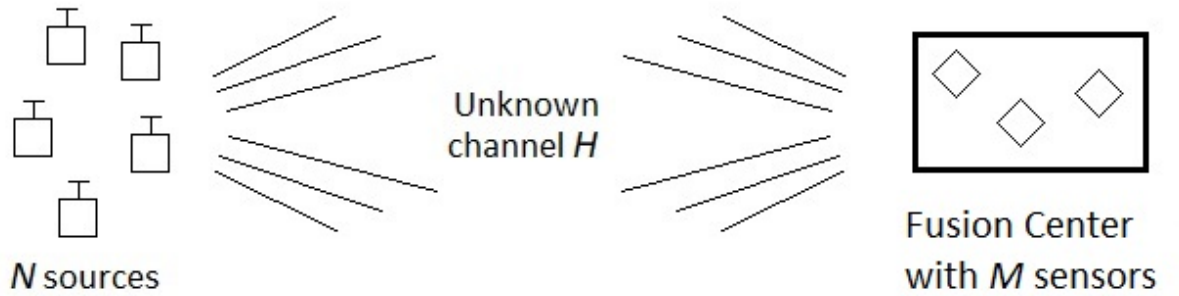
$$Y = S + Z \quad (1.3)$$

$$S = HX \quad (1.4)$$

Therefore,

$$Y = HX + Z \quad (1.5)$$

In this thesis, we address a modified version of this communication system which is depicted in figure 1.2. There are  $N$  sources which transmit the signal matrix  $X$  over an unknown channel of matrix  $H$ . This is then received or observed via  $M$  sensors in the fusion center. The fusion center observes multiple radio sources via noisy sensor measurements over unknown time-varying channels.



**Figure 1.2**  $N$  sources transmit signal  $X$  over unknown channel  $H$ , received by the fusion center for measurements via a total of  $M$  sensors.

Source separation refers to recovering original signals from their mixtures and was first formulated around 1982[1]. In real-life communication systems, there is no prior



information present at the receiver about the source signals or the channels between the transmitter and receiver. These systems are referred to as *blind* and source separation for such systems is known as Blind Source Separation (BSS), which was formulated in 1984 [1]. BSS exploits only the information carried by the received signals.

Our system closely resembles a model of the Internet-of-Things (IoT) system similar to LoRa, Sigfox, or Narrow Band-IoT (NB-IoT) [2, 3]. Keeping this in mind and in an attempt to capture key aspects of IoT systems, our model too has more number of latent sources than the number of sensors. The purpose of these sensors is to separate information from these different sources without any prior knowledge of how they work using different BSS techniques.

For systems which have fewer sources than mixtures, Independent Component Analysis (ICA) [4] is most popularly used as the results have fewer ambiguities. However, in most practical systems, the number of sources is much larger than the number of mixed signals, which leads to underdetermined BSS. In a situation of underdetermined mixing, ICA has a much poorer performance [5]. Other approaches to solve the BSS problem include the Principal Component Analysis (PCA) [6] and Singular Value Decomposition (SVD) [7].

BSS is a heavily applied approach to solve problems in a multitude of fields. In acoustics, it has been used to identify signals from multiple superimposing waves [8] and to provide faithful estimates of the source signals and reduce acoustic noise [9]. The Degenerate Unmixing Estimation Technique (DUET) [10] algorithm in BSS can blindly separate multiple sources given, anechoic mixtures with non-overlapping time-frequency representations, which is true in case of speech [11]. In the field of medical signal

processing, BSS is used for detecting biomedical markers in tests like EKG, ECG and EMG [12]. Research in image processing and analysis has also delved into using BSS for identification from mixtures of images [13] and improving security for speech and image encryption [14]. BSS is also used in speech recognition [15, 16], image extraction [17, 18], and surveillance [19, 20].

Different metrics are used to evaluate the performance of BSS methods depending on the applications. Signal-to-Interference Ratio (SIR) is used for speech recognition [15]. Performance index as mentioned in [17] is used for image feature extraction. The choice of index and approach to BSS problems go hand in hand.

In wireless network, the need for BSS arises in non-collaborative applications in which the signals and channels through which they are received at the sensors are both unknown. ICA has been used in the past to solve BSS problems in wireless networks [21-24] since it provides a good decomposition with only scaling and permutation ambiguities [25]. However, certain assumptions made in the implementation of ICA are that the underlying mixing process has the same number of inputs and outputs, and that all sources are active throughout the observation interval. These assumptions are limiting and not suitable for practical systems similar to IoT systems.

The model depicted in this work has a larger number of sources than sensors ( $M < N$ ), but their activity is sparse and sporadic. The number of active sources at any given time instant is much smaller than the total number of sources [26]. The duration of activity for any source is a small fraction of the overall observation time. Also, a source tends to remain in its current state of activeness for a continuous duration of time. To portray these conditions, a Hidden Markov Model (HMM) [27] [28] is used to determine source activity.

A system is said to be modeled on a Hidden Markov Model when its output is based on a Markov Model with unobserved states [27]. Our model has a HMM dictating the activity of sources: the transmission of information from a source is controlled by the on-off states. These are directed by the transition probabilities as configured in the simulation. It is designed such that an active source tends to remain active for a continuous duration, and overall is active only for a very small fraction of the entire observation period [26], to introduce sparsity in the system.

Signals thus generated are transmitted to the fusion center for measurement via communication paths known as channels. In [30] Shannon describes a channel as *merely the medium used to transmit the signal from transmitter to receiver. It may be a pair of wires, a coaxial cable, a band of radio frequencies, a beam of light, etc.* A channel model is a mathematical way to describe the behavior of the channel. In an ideal case, all information transmitted will reach the center without any modification or attenuation. However, the presence of obstacles in the environment surrounding the transmitters and receivers creates multiple paths that the signals can traverse. The receiver sees multiple incoming signals with variable attenuation, delays and phase shift giving rise to multipath fading. Fading in highly crowded urban regions where more obstacles scatter the transmitted signal can best be described using the Rayleigh distribution – the sum of two Gaussian random variables. Rayleigh Fading is applied when there is no direct line of sight between the transmitter and receiver. Experiments in densely populated Manhattan have shown Rayleigh fading [31].

In mobile communication, the relative motion between sender and receiver causes changes in frequency or wavelength in the signal known as Doppler Shift or Doppler

Effect. When the motion between transmitter and receiver is towards each other, there is an increase in frequency of the signal giving rise to a *blueshift*. When motion is in opposite directions, the frequency at which the signal arrives reduces, causing a *redshift*.

Wireless communication systems face loss in signal strength primarily because of Doppler shift in mobile environments and scattering due to reflections from obstacles in the surroundings. R. H. Clarke modeled the mobile communication channel with Rayleigh Fading [32]. In [33], M. J. Gans introduced a power spectral analysis for Clarke's model to include Doppler. J. I. Smith adapted the Clarke and Gans model for fading for effective computer simulation [34], which is applied in our system model.

Our problem boils down to solving  $X$  from equation (1.5), which is the general equation depicting linear systems. Underdetermined linear systems of equations of the form  $Ax = b$  have infinitely many solutions, when the matrix  $A$  is full rank. The columns of the matrix  $A$  serve as a basis for expressing the observations  $b$ . The set of basis signals that form the matrix  $A$  is called a dictionary. Elements of a dictionary are known as *atoms*. When the dictionary is overcomplete,  $A$  is not unique. This is where the sparsity comes into play because we want to find a sparse vector which has a small number of significant coefficients (i.e., most of constituents are reduced to zero) and reduce the number of atoms to be estimated [35]. Dictionary learning, a mix of machine learning and signal processing, comprises of iterative algorithms aimed to find the dictionary in which some training data admits sparse representations.

Sparse representation problems for which the dictionary  $A$  is unknown require Dictionary Learning in addition to signal recovery [35]. The benefit of using Dictionary learning is that it is capable of *learning* a dictionary adaptively from a set of observations

using an iterative approach. In [36-39], the DL algorithms solve BSS problems for systems that have more sources than sensors and the algorithms are blind to time variability of sources with memories.

Assuming only information about the sparseness of  $x(t)$  at each time  $t$ , a standard approach is to utilize the channel matrix  $H$  as a dictionary to be learned to recover  $X$ . DL techniques approximate the solution of the data-fitting problem where  $\chi$  is the set of matrices with columns containing a limited number of non-zero entries:

$$\min_{H, X \in \chi} \|Y - HX\|^2 \quad (1.6)$$

The Dictionary Learning algorithms used here are Least Absolute Shrinkage and Selection Operator (LASSO) [40] for signal estimation and Multiple Dictionary Update (MDU) [41] for channel estimation. The LASSO is used to minimize the sum of squares subject to sum of absolute value of coefficients being less than a fixed constant [40].

The MDU [41] approach is used to primarily estimate the channel matrix  $H$  for a signal  $X$  as estimated by the LASSO algorithm in an iterative approach. The MDU algorithm minimizes the least square expression subject to the number of limited non-zero components of the dictionary. These two algorithms are used iteratively where an initial dictionary is picked at random [42], fed into the LASSO algorithm. The output from the first stage is then used for the MDU algorithm to estimate the channel. This estimated channel is used by the LASSO again, so on and so forth.

In this thesis, we address the BSS problem in wireless networks depicted in figure 1.2, in which a fusion center observes multiple sources via noisy sensors over time-varying channels. A HMM is used to generate source activity to introduce necessary sparsity in the system. The Clarke and Gans fading model is used to simulate the time-

varying channel with different amounts of Doppler. When the signal arrives at the receiver, LASSO and MDU algorithms are used in tandem to estimate the information.

The performance of the system is charted as a Receiver Operating Characteristic (ROC) Curve which plots the probability of correct detection versus the probability of false alarm. The probability of detection is the ratio of the number of correctly detected active sources and the total number of active sources over  $T$  time samples; and the probability of false alarm is the ratio of the number of incorrectly detected active sources and the total number of inactive sources over the same  $T$  time samples.

The performance of these algorithms are measured for different scenarios. We test the learning approaches in settings of varying Signal-to-Noise Ratios (SNR) to figure out the practical applicability of the same. As Doppler Effect introduces a change in the values of the channel matrix  $H$ , the variation in the data received at the sensors is also high. With higher Doppler frequency values, this variance is higher than it is for lower Doppler frequencies. We also test the algorithms by changing the number of time samples which are observed to figure out if we can control the rate of change of the channel for which the algorithms provide the best trade-off between too many and too few observations. Time coherence is calculated as a function of maximum Doppler frequency to estimate the ideal number of observations for different amounts of Doppler in the system.

The rest of this thesis is arranged as follows:

Chapter 2 talks about the system model in detail. It describes what mathematical models have been used to set up a realistic system, with a set of sources transmitting signals with sparsity in activity and a channel with multipath fading, modeled by the use of

Rayleigh fading, and Doppler Effect. It describes the HMM model used for signal generation and the Clarke and Gans Model for simulating fading channels.

Chapter 3 comprises of the math behind the system design and the algorithms used to decipher the information from the received noisy signals. It includes equations needed to explain how the DL algorithms and how they have been applied to solve our current problem. We discuss LASSO and MDU in detail and the theory behind other algorithms which have been used to formulate the ones we use. We also discuss the changes we suggest to these existing algorithms to improve their performance when subjected to a time-varying system.

Chapter 4 contains the results from the simulation in the form of ROC curves which shows how the performance of the DL algorithms change with varying Doppler, when compared to the performance of a system with time-invariant channel. We also discuss how time coherence can be used to formulate the ideal number of observations to provide the basis of suggested segmentation and subsequently the optimum trade-off between too much and too little variation in the system for the learning algorithms to estimate the transmitted signal.

## CHAPTER 2

### SYSTEM MODEL

Consider a model that includes  $M$  receivers or sensors and  $N$  sources. The number of sources  $N$  is usually larger than the number of sensors  $M$  ( $M < N$ ). All of these  $M$  sensors make up a fusion center which has access to the  $N$  receiver antennae via communication channels. Similar models with fusion centers reflect the architecture of IoT networks [2, 3]. In our model, each model transmits information intermittently and is therefore active only for a small subset of the  $T$  symbol periods over which the sensors collect data.

Assuming all nodes are time-synchronous, the discrete time signals received by the  $M$  sensors over  $T$  symbol periods is given in matrix form as:

$$Y = HX + Z \quad (2.1)$$

where:

$Y = [y(1), y(2), \dots, y(T)]$  is an  $M \times T$  matrix in which the columns represent the  $M \times 1$  received signals  $y(t)$  across all  $T$  symbols  $t = 1, 2, \dots, T$ .

$X = [x(1), x(2), \dots, x(T)]$  is an  $N \times T$  matrix that collects all  $N \times 1$  signals  $x(t)$  transmitted from all  $N$  sources over  $T$  time samples.

$Z = [z(1), z(2), \dots, z(T)]$  consists of independent zero-mean Gaussian noise entries with variance  $\sigma^2$ .

$H$  is an  $M \times N \times T$  matrix denoting the channel information between every pair of source and sensor across all  $T$  symbol periods. The channel is time-varying and changes according with respect to the maximum Doppler frequency of the setup.

Given the intermittent nature of the activity of the sources, the  $t$ -th column vector  $x(t)$  that collects the  $M$  symbols transmitted by the other sources at time  $t$ , is generally



sparse. To put it in other words, only a small subset of these  $x(t)$  entries are non-zero, which makes the vector and by extension the matrix  $X$  sparse.

In cases of wireless communication, there is motion between the sources and the sensors. With the introduction of this motion in our model arises a relative velocity between source and sensor. This created a Doppler Shift in the carrier frequency creating channels that change with time. The maximum Doppler frequency is a function of the carrier wavelength and velocity of motion.  $H$  is represented as a three-dimensional matrix, which has  $T$  pages each containing the channel conditions between each pair of source and sensor at that time instance.

The signals  $y(t)$  for  $t = 1, 2, \dots, T$ , are collected at the fusion center. Based on  $Y$ , the goal of the fusion center is to detect when a source is active and recover the signals  $x(t)$  for  $t = 1, 2, \dots, T$ , or the signal matrix  $X$ , in the absence of data about the channel matrix  $H$ .

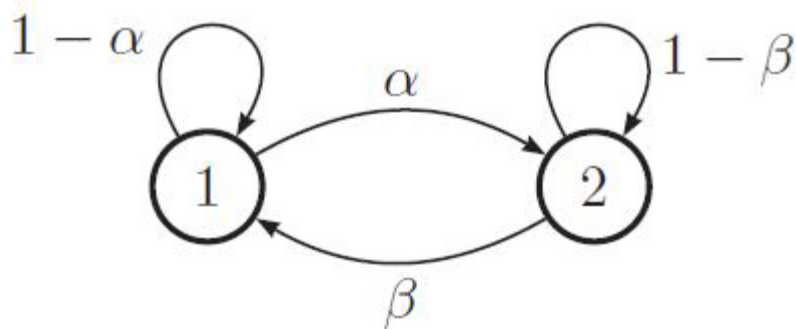
## 2.1 Signal Generation

For each source  $n$ , the activation pattern is noted as a binary sequence  $s_n(t)$ , where the binary state  $s_n(t)$  indicates whether a particular source was active or inactive. When  $s_n$  is zero, the source is considered to be switched off and nothing is transmitted. When  $s_n$  is one, source is active and transmits information.

The signal is generated by using an intermittent and smooth deterministic model. Each source is active only for a small fraction of the entire time duration. The activity pattern tend to be smooth, i.e., a source once switched off tend to stay that way and not

change its state. This causes the sequence  $s_n(t)$  to be smooth as the number of transitions from on state to off state and vice versa is small.

For designing source activity that captures the properties of an IoT system, we use a probabilistic Hidden Markov Model. A binary sequence such as  $s_n(t)$  can be depicted by a Markov Model with two states (on denoted by 1 and off by 0). Figure 2.1 shows the state transition diagram of a Markov Model with two states **1** and **2** with transition probabilities  $\alpha$  and  $\beta$ .



**Figure 2.1** State Transition Diagram for a 2-state Markov Model.

Source: K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, Massachusetts: MIT Press, 2012, pp. - 590.

A stationary finite-state Markov Model is equivalent to a stochastic automaton [28]. It is common to visualize this by drawing a directed graph where nodes represent states and arrows represent legal transitions. The probability of going from state  $i$  to state  $j$  is:

$$A_{ij} = \Pr(X_t = j | X_{t-1} = i) \quad (2.2)$$

The transition probabilities between different states can be shown much clearly in the form of a transition matrix. All rows of this matrix sums to be 1.

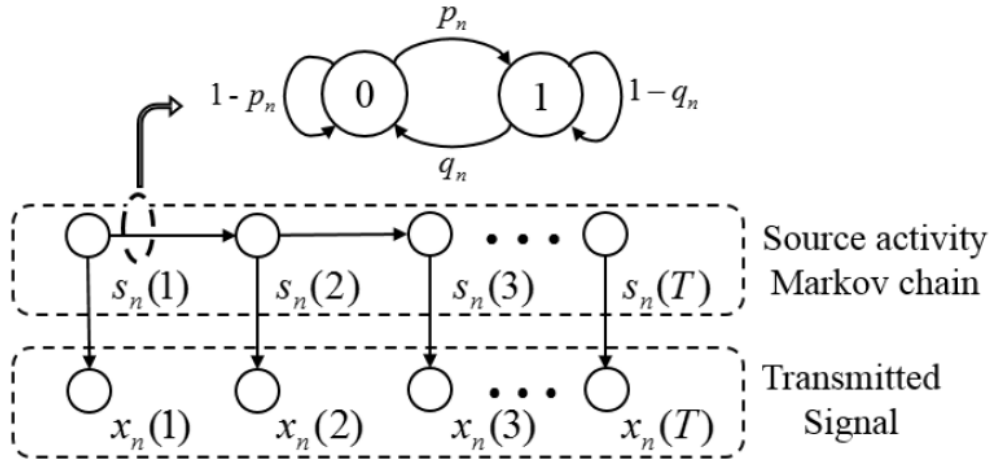
$$A = \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix} \quad (2.3)$$

The transition probabilities of the two-state Markov chain used in our model are defined as:

$$p_n = \Pr(s_n(t) = 1 | s_{n-1}(t) = 0) \quad (2.4)$$

$$q_n = \Pr(s_n(t) = 0 | s_{n-1}(t) = 1) \quad (2.5)$$

A Hidden Markov Model (HMM) consists of a discrete-time discrete-state Markov chain, with hidden states, and an observation model. By controlling the transition probabilities in the Markov Model (transition from OFF to ON is  $p_n$  and from ON to OFF is  $q_n$ ), the duration of transmission can be changed; sparsity can be introduced in the system and controlled with specific calculations.



**Figure 2.2** Hidden Markov Model for any source  $n$ .

Each transition between the binary outputs ( $s_n$ ) of the Markov Model is determined by the Markov chain whose two states are on and off. The binary output ( $s_n$ ) is used to generate the signal ( $x_n$ ) transmitted by the source.

$$x_n(t) \sim f_n(t) \quad \text{when } s_n(t) = 1 \quad (2.6)$$

$$x_n(t) = 0 \quad \text{when } s_n(t) = 0 \quad (2.7)$$

Here,  $f_n$  denotes a distribution on which all independent  $x_n$  samples are modelled. When  $s_n$  is zero, the source is considered to be switched off and nothing is transmitted. For our system,  $f_n$  is Gaussian with zero mean and unit variance.

The signals incoming from the sources at different time instances in real-life appear random to the receiving sensors which have no control over when a source starts transmitting or stops. To design a seemingly random signal, a Hidden Markov Model (HMM) [27] is used.

## 2.2 Channel Generation

A communication channel is the medium used to transmit information from one point to the other. Though wired channels are fast and far more secure than their wireless counterparts, they are usually limited to short distances. In cellular communication, the receivers are often mobile devices and wired communication channels are not possible to connect the sources and sensors. In mobile communication, the relative motion between sender and receiver causes changes in frequency or wavelength in the signal known as Doppler Shift or Doppler Effect. All mobile communications must contend with Doppler frequency shift [49].

Cellular wireless systems experience loss of signal strength due to Doppler shift in mobile environments, and scattering due to reflections from natural and manmade obstructions. Ideally, modeling a channel is calculating all the physical processing effecting a signal from the transmitter to the receiver. For a wireless channel, the envelope of the channel response is modeled to have a Rayleigh distribution. Rayleigh Fading is a reasonable model when there are many objects in the environment that scatter the radio signal before it reaches the receiver. Experiments in densely populated Manhattan have shown Rayleigh fading [31].

By controlling the carrier frequency ( $f_c$ ), and subsequently the wavelength ( $\lambda$ ) and relative velocity ( $v$ ) we can influence the amount of Doppler Effect in the channel. The instantaneous frequency of received component arriving at an angle  $\alpha$  is:

$$f(\alpha) = f_m \cos(\alpha) + f_c \quad (2.8)$$

$$f_m = \frac{v}{\lambda} \quad (2.9)$$

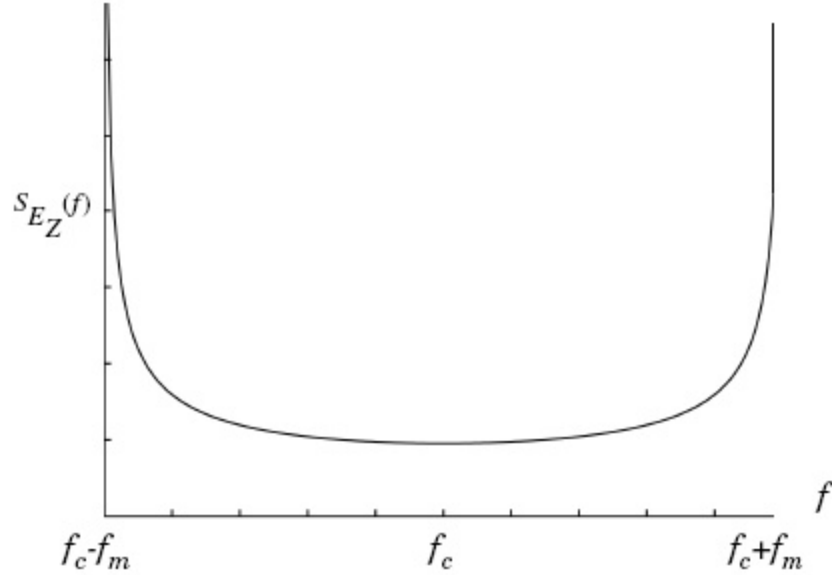
The angle of arrival is assumed to be uniformly distributed over the interval  $(0, 2\pi)$  and Doppler components arriving at exactly  $0^\circ$  and  $180^\circ$  have an infinite power spectral density. This is not a problem since angle of arrival is continuously distributed and probability of components arriving at exactly these angles is negligible [47].

R. H. Clarke modeled the mobile channel as a Rayleigh fading channel [32]. Later, M. J. Gans deduced a spectral model from Clarke's original analysis [33]. In Clarke's model, a vertical quarter wavelength antenna with azimuthal gain pattern of the mobile

antenna as function of angle of arrival,  $G(\alpha) = 1.5$  and a uniform distribution of  $1/2\pi$  over 0 to  $2\pi$ , the output spectrum is given by:

$$S_{E_z}(f) = \frac{1.5}{\pi f_m \sqrt{1 - \left(\frac{f - f_c}{f_m}\right)^2}} \quad (2.10)$$

The Doppler power spectrum we obtain from plotting equation (2.10) with respect to instantaneous frequency generates the following curve.

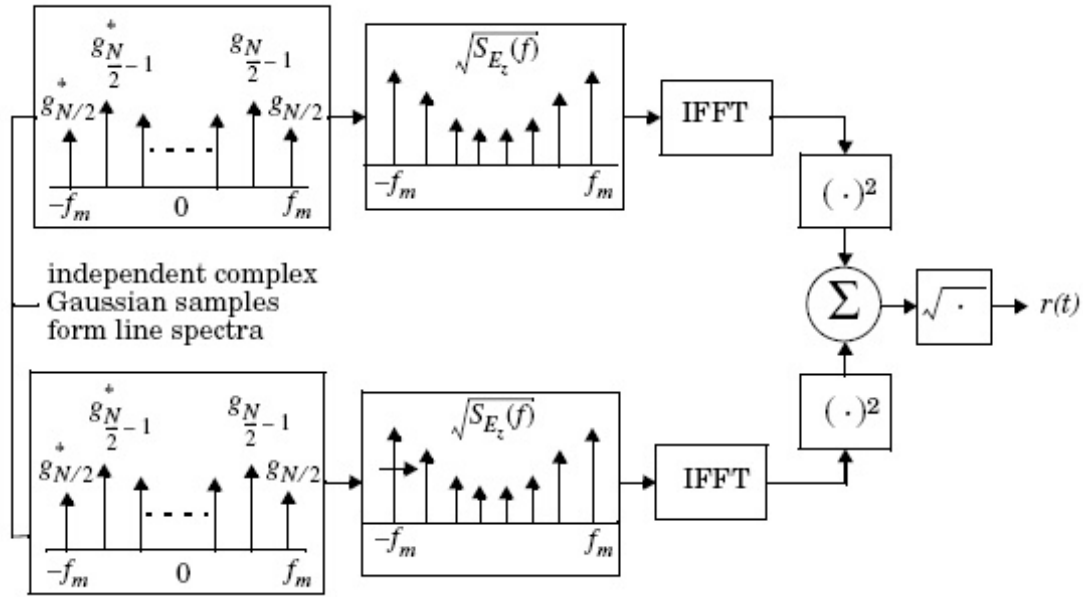


**Figure 2.3** Doppler Power Spectrum for an unmodulated continuous wave carrier.

Source: M. J. Gans, "A power-spectral theory of propagation in the mobile-radio environment," in *IEEE Transactions on Vehicular Technology*, vol. 21, no. 1, pp. 27-38, Feb. 1972.

The spectrum is centered on the carrier frequency and is zero outside the limits of  $f_c \pm f_m$  where  $f_m$  is the maximum Doppler frequency. Each of the arriving components have their own carrier frequency which is slightly offset from the center frequency (as shifts in direction manifest in terms of phase shifts).

The channels connecting every pair of source and sensor in a wireless network may change with time. The possible movement of any or both of the source and sensor introduces Doppler shift in the frequency. John I. Smith simulated the Clarke and Gans model on a computer using the algorithm described below [34] which is applied in our system model to implement said time-varying channels. The following figure demonstrates how to implement in a computer simulation.



**Figure 2.4** Frequency domain implementation of a Rayleigh fading simulator.

**Source:** T. S. Rappaport, Wireless Communications Principles and Practice, 2<sup>nd</sup> ed., Prentice Hall, 2002, pp.-215.

This method uses a complex Gaussian random number generator (noise source) to produce a line spectrum with complex weights in the positive frequency band. The maximum frequency components of this line spectrum is  $f_m$ . The negative frequency components were constructed by conjugating the positive frequency components. The random valued line spectrum is then multiplied with a discrete frequency representation of

$\sqrt{S_{E_z}(f)}$  having the same number of points as the noise source. This is followed by performing IFFT on the sequences thus generated to get two time series. The squares of each signal point in time is added to form one sequence, the square root of which is  $r(t)$ .

$r(t)$  denotes the channel between any given pair of source and sensor for our model, where every value at different time instance  $t$  represents the channel information at that time. We generate multiple such sequences for all possible source and sensor combination. Then arrange it in a three dimensional matrix with every page denoting channel state at any given time, with element  $(i,j)$  corresponding to  $i$ -th sensor and  $j$ -th source. There are  $T$  pages, each containing  $M$  rows and  $N$  columns.



## CHAPTER 3

### DICTIONARY LEARNING ALGORITHMS

Dictionary learning is a mix of machine learning and signal processing and comprises of iterative algorithms aimed to find the dictionary in which some training data admits sparse representations. Our problem boils down to solving for the signal matrix  $X$  which is in the form of any general equation depicting linear systems.

Underdetermined linear systems of equations of the form  $Ax = b$  have infinitely many solutions, when the matrix  $A$  is full rank. The columns of the matrix  $A$  serve as a basis for expressing the observations  $b$ . The set of basis signals that form the matrix  $A$  is called a dictionary. Elements of a dictionary are known as *atoms*. A dictionary can be seen as an over-complete basis such that every vector in the same space can be approximately expressed as linear combinations of elements in this dictionary. When the dictionary is overcomplete,  $A$  is not unique.

Dictionary Learning is the process to find a good over-complete basis in terms of minimum approximation error and sparsest solution given a set of vector. This is where the sparsity comes into play because we want to find a sparse vector which has a small number of significant coefficients and reduce the number of atoms to be estimated [35].

Dictionary learning algorithms use a set of signal examples to identify that will best sparsify them, thereby leading to more effective modeling [41]. Dictionary Learning techniques used here are used to approximate the solution of the data-fitting problem where  $\chi$  is the set of matrices with columns containing a limited number of non-zero entries:

$$\min_{D, X \in \chi} \|Y - DX\|^2 \quad (3.1)$$

This problem is not convex with respect to the pair  $(D, X)$ . Dictionary learning algorithms use an iterative approach where the original information  $X \in \mathcal{X}$  and the dictionary  $D$  are optimized alternately [45]. We attack this problem by iteratively performing a two-stage procedure as described in [42] where the index of iteration is  $k$ . First, the sparse representation is handled in the following form:

$$X^{(k+1)} = \arg \min_X \|Y - D^{(k)} X\|^2 \quad (3.2)$$

Next we perform the dictionary update stage using:

$$D^{(k+1)} = \arg \min_D \|Y - DX^{(k+1)}\|^2 \quad (3.3)$$

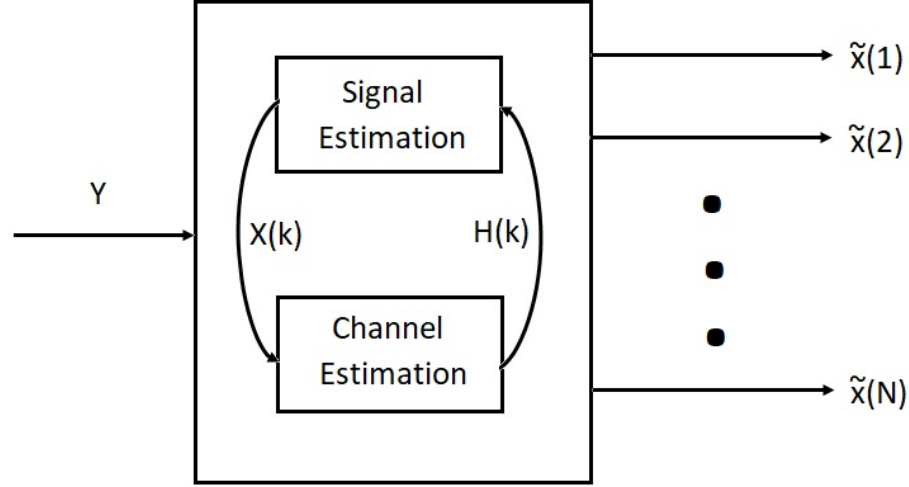
In this section, we review Dictionary Learning techniques which do not need prior information about the memory of the sources. These methods use the fact that the signals  $x(t)$  is sparse at any time  $t$ . Sparse and redundant representation modeling of signals is a very effective way to describe the inner-structure of signal sources [41]. Assuming information only about the sparseness of  $x(t)$  at each time  $t$ , a standard approach is to utilize the channel matrix  $H$  as a *dictionary*, as described above, to be learned to recover signal matrix  $X$ .

In the algorithms we use, we first handle the optimization over the signal  $X$  for a given channel matrix  $H$  and then over the channel  $H$  for a given signal matrix. If the fusion center acts like a master clock synchronizing all other nodes, the problem changes slightly to look like:

$$\min_{H, X \in \mathcal{X}} \|Y - HX\|^2 \quad (3.4)$$

We initiate a random dictionary  $H$  to estimate the signal matrix  $X$  column-by-column using the LASSO algorithm [40]. With this estimated matrix, the MDU

algorithm [41] learns the dictionary  $H$ . Feeding the estimated matrices of these algorithms iteratively into the other helps reach convergence. The functional block diagram of how the learning proceeds is depicted in figure 3.1.



**Figure 3.1** Functional Block Diagram of proposed algorithm.

When the solution to the data fitting problem is reached, we compare the estimated signal matrix with the original sent matrix to plot the probabilities of false alarm and detection which we use as a metric to measure performance of the learning algorithms. DL methods are subject to inherent permutation and sign ambiguities [46]. The scaling ambiguity can be resolved by normalizing the columns of the channel matrix [41-44].

### 3.1 Signal Estimation: LASSO Algorithm

Standard sparse optimization estimators such as orthogonal matching pursuit (OMP) [50] can be used to address the solution of equation (3.1). Similarly, regression analysis can also be used. At its core all regression analysis approaches examine the influence of one or more independent variables on a dependent variable. The Least Absolute Shrinkage and

Selection Operator or the LASSO algorithm [40] is a regression analysis approach that performs both variable selection and regularization to improve interpretability of produced model and also prediction accuracy. LASSO was originally designed to tackle least squares problems and has multiple similarities to ridge regression and subset selection.

Ridge Regression approaches the problem in equation (3.1) by treating it like an ordinary least squares problem and adding a  $\ell_2$  penalty term. The OLS loss function is augmented in such a way that not only the sum of squared residuals is minimized but also penalize the size of parameter estimates, in order to shrink them towards zero.

$$x^{ridge} = \arg \min_x \|y - Dx\|_2^2 + \mu \|x\|_2^2 \quad (3.5)$$

Ridge regression solves the problem column wise using a weight  $\mu$  associated with the  $\ell_2$  penalty term. It could have a better prediction error when compared to linear regression in a variety of scenarios, depending on the choice of the weight  $\mu$ . However, it works best only when a subset of the true coefficients are small or zero. But coefficients are never set to zero exactly, and cannot perform variable selection in the linear model. While this didn't seem to hurt its prediction ability, it is not desirable, especially in cases with a large number of variables.

Controlling the weight, we can change how the algorithm behaves. Setting  $\mu$  to 0 is the same as using the OLS, while the larger its value, the stronger is the coefficients' size penalized. When  $\mu \rightarrow 0$ , the ridge regression estimate is the same as the solution of an ordinary least squares solution. When  $\mu \rightarrow \infty$ , the ridge regression estimate approaches zero, i.e.,  $x^{ridge} \rightarrow 0$ .

LASSO -Least Absolute Shrinkage and Selection Operator- was first formulated by Robert Tibshirani in 1996 [40]. It is a powerful method that performs two main tasks:

regularization and feature selection. LASSO is quite similar to ridge regression conceptually. It also adds a penalty for non-zero coefficients, but unlike ridge regression which penalizes sum of squared coefficients in the form of a  $\ell_2$  penalty with a weight  $\mu$ , LASSO penalizes the sum of their absolute values or the  $\ell_1$  penalty term with a weight  $\lambda$ . As a result, for high values of  $\lambda$ , many coefficients are exactly zeroed under LASSO, which is never the case in ridge regression. The LASSO estimate is defined as:

$$x^{lasso} = \arg \min_x \|y - Dx\|_2^2 + \lambda \|x\|_1 \quad (3.6)$$

The only difference between the LASSO problem and ridge regression is that the latter uses a  $\ell_2$  penalty, while the former uses a  $\ell_1$  penalty. The tuning parameter or the weight  $\lambda$  controls the strength of the penalty:

$$x^{lasso} \rightarrow x^{OLS} \quad \text{when } \lambda \rightarrow 0 \quad (3.7)$$

$$x^{lasso} \rightarrow 0 \quad \text{when } \lambda \rightarrow \infty \quad (3.8)$$

For  $\lambda$  in between these two extremes, we are balancing two ideas: fitting a linear model of  $y$  on  $X$ , and also shrinking the coefficients. But even though these problems look similar, their solutions behave very differently. The nature of the  $\ell_1$  penalty causes some coefficients to be shrunk to exactly zero. This is what makes the LASSO significantly different from the ridge regression approach. It is able to perform variable selection in the linear model which the ridge regression cannot.

As  $\lambda$  increases, more coefficients are set to zero, or in other words, fewer variables are selected, and the non-zero coefficients undergo stricter shrinkage. Reducing coefficients to exactly zero is extremely important when we recall that our model is based on an IoT structure which has intermittent sources and relies heavily on sparsity for its

assembly and that the dictionary learning section of the algorithms provides improved performances for a sparse case.

When the learning proceeds in multiple iterations as shown in figure 3.1, for any fixed iterate  $H^{(k)}$  at the  $k$ -th iteration, estimating the signal  $X$  becomes:

$$X^{(k+1)} = \arg \min_{X \in \mathcal{X}} \|Y - H^{(k)} X\|_2^2 \quad (3.9)$$

Alternatively, we use the LASSO algorithm to solve the following convex problem separately for each  $t$ , where the weight  $\lambda$  is a parameter which is determined as a function of the sparsity of the vector  $x(t)$ :

$$\min_{x(t)} \|y(t) - H^{(k)} x(t)\|_2^2 + \lambda \|x(t)\|_1 \quad (3.10)$$

Here, the vectors  $x(t)$  make up  $X$  which is the signal matrix containing the transmitted information and the channel matrix  $H$  is used as the dictionary in the learning algorithms. LASSO's  $\ell_1$  penalties have powerful statistical and computational advantages. According to the “Bet on Sparsity Principle” as described in [51]: *Assume that the underlying truth is sparse and use a  $\ell_1$  penalty to try to recover it. If you're right, you will do well. If you're wrong— the underlying truth is not sparse—, then no method can do well.*  $\ell_1$  penalties encourage sparsity and simplicity in the solution.  $\ell_1$  penalties are convex and the assumed sparsity can lead to significant computational advantages.

The LASSO is a shrinkage and selection method for linear regression. It minimizes the usual sum of squared errors, with a bound on the sum of the absolute values of the coefficients. It has connections to soft-thresholding of wavelet coefficients, forward stage-wise regression, and boosting methods [28].

The tuning parameter  $\lambda$ , which can be any value between  $[0, \infty)$ , controls the strength of the  $\ell_1$  penalty. The LASSO estimates are generally biased, but have good mean squared error when compared to ridge regression. Additionally, the fact that it sets coefficients to zero is a big advantage for the sake of interpretation, especially for systems such as ours.

### 3.2 Channel Estimation: MDU Algorithm

The previous stage detailed in equation (3.2) is simply an ordinary sparse coding problem in which the sparse representations of all the signals are computed using the current dictionary. The dictionary in equation (3.3) is updated to reduce the representation error of the previous stage.

One of the simplest dictionary learning algorithms is the Method of Optimal Directions (MOD) [43] which primarily finds the unconstrained minimum of equation (3.1) and then projects on the set that contains the dictionary  $D$ . Any standard sparse optimization estimators can be used to approximate the solution of equation (3.1). The MOD dictionary update is performed by the following closed form expression:

$$D^{(k)} = \arg \min_D \|Y - DX^{(k)}\| = YX^T (XX^T)^{-1} \quad (3.11)$$

This is followed by normalizing the columns of the dictionary which is crucial in eliminating errors such as the scaling ambiguity in the channel matrix. If the change in minimization corresponding to each iteration is within a preset threshold, the algorithm stops; otherwise another iteration is applied.

A MOD-like algorithm developed in [41] fixes the support of  $X$  and updating its non-zero entries in their associated row vectors at a time. The support  $\mathbf{S}(X)$  denotes the positions of entries of the matrix  $X$  which are non-zero. This is of special significance when dealing with sparse problems. The problem in equation (3.1) is modified by adding constraints on the support by subjecting it to the support as found in the previous iteration:

$$\begin{aligned} \min_{X,D} \|Y - DX\|_2^2 \\ \text{subject to } \mathbf{S}(X) = \mathbf{S}(X^{(k+1)}) \end{aligned} \quad (3.12)$$

To solve this problem, [41] proposed the usage of alternating minimization of the dictionary  $D$  and the information matrix  $X$ : minimizing equation (3.12) over  $D$  with a fixed  $X$  from equation (3.11) and minimizing equation (3.12) over  $X$  with a fixed  $D$  decouples for each column of  $X$  and results in the following [42]:

$$\forall i : x_i = \arg \min_x \|y_i - Dx\|_2^2 \text{ subject to } \mathbf{S}(x) = \mathbf{S}(x_i^{(k+1)}) \quad (3.13)$$

By defining  $\omega_i = \{j : x_i(j) \neq 0\}$  equation (3.13) undergoes the following changes:

$$\forall i : x_i(\omega_i) \leftarrow (D_i^T D_i)^{-1} D_i^T y_i \quad (3.14)$$

where,  $D_i$  are the columns of  $D$  which have been used in the representation of  $y_i$ . Performing multiple alternations, as few as just three, between equations (3.11) and (3.14) gives improved learning performances which have been shown in [41]. As the dictionary undergoes multiple updates based on the support of  $X$ , this algorithm is known as the Multiple Dictionary Update (MDU) [41].



A change to the conventional MDU has been suggested in [42] which obtains a new dictionary learning problem by defining a first-order series expansion for the matrix-valued function  $F(D, X) = DX$  about a point  $(D_0, X_0)$ .  $D$  and  $X$  can be then rewritten as:

$$D = D_0 + (D - D_0) \text{ and } X = X_0 + (X - X_0) \quad (3.14)$$

Here,  $(D - D_0)$  and  $(X - X_0)$  are small in the sense of the second norm. The primary minimization problem boils down to:

$$\{D^{(k+1)}, X^{(k+1)}\} = \arg \min_{D, X} \|Y + D^{(k)} X^{(k)} - DX^{(k)} - D^{(k)} X\|_2^2 \quad (3.16)$$

Substituting  $D = D^{(k)}$  [42] reduces sparse representation to the general dictionary learning problem. Thus the proposed changes does not affect the sparse representation stage and any sparse coding algorithm used for the same. Substituting  $X \leftarrow X^{(k+1)}$  and introducing  $Z = Y + D^{(k)} X^{(k)} - D^{(k)} X^{(k+1)}$  reduces the problem to:

$$D^{(k+1)} = \arg \min \|Z - DX^{(k)}\|_2^2 \quad (3.17)$$

Approaching this way, sparse representation stays the same as equation (3.14) and dictionary update stage changes as  $D^{(k+1)} = ZX^T (XX^T)^{-1}$  which is followed by the normalizing the columns of the dictionary [42].

The MDU approach estimates the channel matrix  $H$  for a given  $X^{(k+1)}$  by following an iterative method. We denote  $\mathbf{S}(X)$  as the set of indices of the non-zero elements in the vector  $x(t)$ . Also, index  $(j, k)$  denotes the  $j$ -th iteration of the MDU algorithm within the  $k$ -th step of the overall DL alternate optimization scheme. At the  $(j, k)$  iteration, the MDU calculates:

$$H^{(j,k)} = YX^{(j,k)T} (X^{(j,k)} X^{(j,k)T})^{-1} \quad (3.18)$$

$$x^{(j+1,k)}(t) = A^{(k)} (H^{(j,k)T} H^{(j,k)})^{-1} H^{(j,k)T} y(t) \quad (3.19)$$

where for all  $t$ ,  $A^{(k)}$  is a diagonal matrix with elements having indices in  $S(X^{(k+1)}(t))$  equal to one and zero otherwise. The iteration is initialized with  $X^{(1,k)} = X^{(k+1)}$ . So, for a constant sparsity pattern  $S(X^{(k+1)}(t))$ , MDU approximates channel and signal alternatively.

The enhancement proposed in the conventional MDU in [42] substituted at iteration index  $k$ , the received matrix  $Y$  changes as  $Y^{(k)} = Y + H^{(k)} X^{(k)} - H^{(k)} X^{(k+1)}$  and the learning proceeds as in equation (3.17) with subsequent modifications.

### 3.3 Modification for Time-Varying Channels

Dictionary Learning-based blind source separation (BSS) problems have been previously applied for systems where the unknown channels are flat-fading and time-invariant. However, in practical scenarios, communication channels do not remain fixed and unchanging with time. In our work, we applied the algorithms to a system with non-direct line of path between transmitter and receiver, undergoing reflection and scattering, and considered the Doppler Effect on the wave propagation due to the motion of the mobile unit. With these additions, the performance of the same algorithms change.

The spectral broadening caused by the time rate of change of the mobile radio channel causes the correlation in the system to decrease. The performance of the algorithms change with the time rate of change in the channel. This degradation does not occur in a fixed manner. The deterioration of the learning is affected by the amount of change the

channel undergoes in the observation period. When the maximum Doppler frequency for a channel is lowered, and the spectrum appropriately adjusted, the performance improves.

For a fixed Doppler spread, the amount of change we expose our system to can be controlled by changing the duration of observation. By inputting fewer samples spread over a shorter duration of time, the learning improves even with the same Doppler spread. But when the number of samples being observed are lowered beyond a certain limit, there ceases to be sufficient information to perform the learning. When the duration of observation is reduced, a part of the transmitted information is lost, which is highly undesirable.

A method to decide on how many samples are observed such that the learning is not affected by the change in the channel while having sufficient information to learn from has been devised in our work. In time domain, time coherence is used to denote time interval over which channel impulse responses are highly correlated. It is the time domain dual of Doppler spread and is used to characterize the time-varying nature of the frequency dispersiveness of the channel in the time domain. Over a duration of time coherence, the channel appears fixed to the learning algorithms. Time coherence is inversely proportional to the maximum Doppler frequency.

$$T_c \propto \frac{1}{f_m} \quad (3.20)$$

From [48], we further get:

$$T_c = \frac{9}{16\pi f_m} \quad (3.21)$$

The received signal is broken into segments in the time-domain. The duration of each of these segments is smaller than the time coherence. The segments are fed in parts to the algorithm iteratively. The segmentation enforces that the entire signal be used, thus garnering no loss of information.

We change the algorithm to break the received information matrix  $Y$  into desired number of segments in the time domain. In the LASSO step of the entire DL process, we use a number of segments each containing the number of time samples whose time duration is shorter than the time coherence value found using equation (3.21) to estimate signal matrix  $X$ , let's call these segments  $X_1, X_2 \dots X_n$ . Before the process starts again, to estimate these two parameters in tandem as it reaches convergence, we append these segments together and start afresh for the next iteration. Doing this forces the algorithms to learn the information in seemingly fixed channels, thus giving better performance.

## CHAPTER 4

### PERFORMANCE ANALYSIS

As seen in Chapter 2, our system consists of more sources than sensors ( $M < N$ ). The sensors take measurements of incoming signals at every time instant ( $t$ ), to which learning algorithms are applied for estimation of original sent information ( $X$ ). As mentioned in Chapter 3, we use the LASSO [40] algorithm for signal estimation and the MDU [41] algorithm for channel estimation.

The signals incoming from the sources at different time instances appear random to the sensors, having no control over when a source starts transmitting or stops. To design a seemingly random signal, a Hidden Markov Model (HMM) [27] is used. We have:

where  $s_n(t)$  is the binary state which dictates whether a source is switched on or off

$$p_n = \Pr(s_n(t) = 1 | s_{n-1}(t) = 0) \quad (4.1)$$

$$q_n = \Pr(s_n(t) = 0 | s_{n-1}(t) = 1) \quad (4.2)$$

at time  $t$ . By controlling the transition probabilities in the Markov Model (transition from OFF to ON is  $p_n$  and from ON to OFF is  $q_n$ ), the duration of transmission can be changed; sparsity can be introduced in the system and controlled with specific calculations.

The binary output ( $s_n$ ) of the Markov Model is used to generate the signal ( $x_n$ ) transmitted by the source.

$$x_n(t) \sim f_n(t) \quad \text{when } s_n(t) = 1 \quad (4.3)$$

$$x_n(t) = 0 \quad \text{when } s_n(t) = 0 \quad (4.4)$$

Here,  $f_n$  denotes a distribution on which all independent  $x_n$  samples are modelled. When  $s_n$  is zero, the source is considered to be switched off and nothing is transmitted. For our system,  $f_n$  is Gaussian with zero mean and unit variance.

The channels connecting every pair of source and sensor in a wireless network may change with time. The possible movement of any or both of the source and sensor introduces Doppler shift in the frequency. The adapted Clarke and Gans model for fading [32] in the effective computer simulation as shown in [34] is applied in our system model to implement said time-varying channels. By controlling the carrier frequency ( $f_c$ ), and subsequently the wavelength ( $\lambda$ ), and relative velocity ( $v$ ) we can influence the amount of Doppler Effect in the channel. The instantaneous frequency of received component arriving at an angle  $\alpha$  is:

$$f(\alpha) = f_m \cos(\alpha) + f_c \quad (4.5)$$

$$f_m = \frac{v}{\lambda} \quad (4.6)$$

The angle of arrival is assumed to be uniformly distributed over the interval  $(0, 2\pi)$  and Doppler components arriving at exactly  $0^\circ$  and  $180^\circ$  have an infinite power spectral density. This is not a problem since angle of arrival is continuously distributed and probability of components arriving at exactly these angles is negligible.

The spectrum is centered on the carrier frequency and is zero outside the limits of  $f_c \pm f_m$  where  $f_m$  is the maximum Doppler frequency. Each of the arriving components have their own carrier frequency which is slightly offset from the center frequency (as shifts in direction manifest in terms of phase shifts). For the case of a vertical  $\lambda/4$  antenna with

azimuthal gain pattern of the mobile antenna as function of angle of arrival,  $G(\alpha) = 1.5$  and a uniform distribution of  $1/2\pi$  over  $0$  to  $2\pi$ , the output spectrum is given by:

$$S(f) = \frac{1.5}{\pi f_m \sqrt{1 - \left(\frac{f - f_c}{f_m}\right)^2}} \quad (4.7)$$

Assuming only information about the sparseness of  $x(t)$  at each time  $t$ , a standard approach is to utilize the channel matrix  $H$  to recover  $X$ . DL techniques approximate the solution of the data fitting problem where  $\chi$  is the set of matrices with columns containing a limited number of non-zero entries:

$$\min_{H, X \in \chi} \|Y - HX\|^2 \quad (4.8)$$

This problem is not convex with respect to the pair  $(H, X)$ . Dictionary learning algorithms use an iterative approach where the original information  $X \in \chi$  and the channel  $H$  are optimized alternately [45]. In the algorithms we use, we first handle the optimization over the signal  $X$  for a given channel matrix  $H$  and then over the channel  $H$  for a given signal matrix. DL methods are subject to inherent permutation and sign ambiguities [46]. The scaling ambiguity can be resolved by normalizing the columns of the channel matrix [41-44].

For any fixed iterate  $H^{(k)}$  at the  $k$ -th iteration, estimating the signal  $X$  becomes:

$$X^{(k+1)} = \arg \min_{X \in \chi} \|Y - H^{(k)} X\|^2 \quad (4.9)$$

The LASSO [40] algorithm is used to solve the convex problem of signal estimation, separately for each time sample ( $t$ ). The weight ( $\lambda$ ) is a parameter which is determined as a function of the sparsity of the vector  $x(t)$  which can be influenced by choosing appropriate transition probabilities in the HMM. The weight could possible take any non-negative real number.

LASSO solves the signal estimation problem by solving the following convex problem:

$$\min_{X \in \mathcal{X}} \|y(t) - H^{(k)} x(t)\|_2 + \lambda \|x(t)\|_1 \quad \text{for } t=1, \dots, T \quad (4.10)$$

As  $\lambda$  increases, the number of non-zero components in the training set increases as well. The selection of the correct  $\lambda$  thus plays an important role in fitting the least squares regression coefficients in estimating the signal matrix column-by-column (or time sample-by-time sample).

The MDU [41] approach is used to primarily estimate the channel matrix  $H$  for a signal  $X$  as estimated by the LASSO algorithm in an iterative approach. At the  $k$ -th iteration, for a fixed  $X^{(k+1)}$ , the channel estimation step uses the MDU which alternatively estimates the channel and signal in an inner loop of index  $j$  of the MDU algorithm within the  $k^{th}$  step of the Dictionary Learning optimization.

At  $(j, k)$  iteration, MDU computes:

$$H^{(j,k)} = YX^{(j,k)T} (X^{(j,k)} X^{(j,k)T})^{-1} \quad (4.11)$$

$$x^{(j+1,k)}(t) = D^{(k)} (H^{(j,k)T} H^{(j,k)})^{-1} H^{(j,k)T} y(t) \quad (4.12)$$

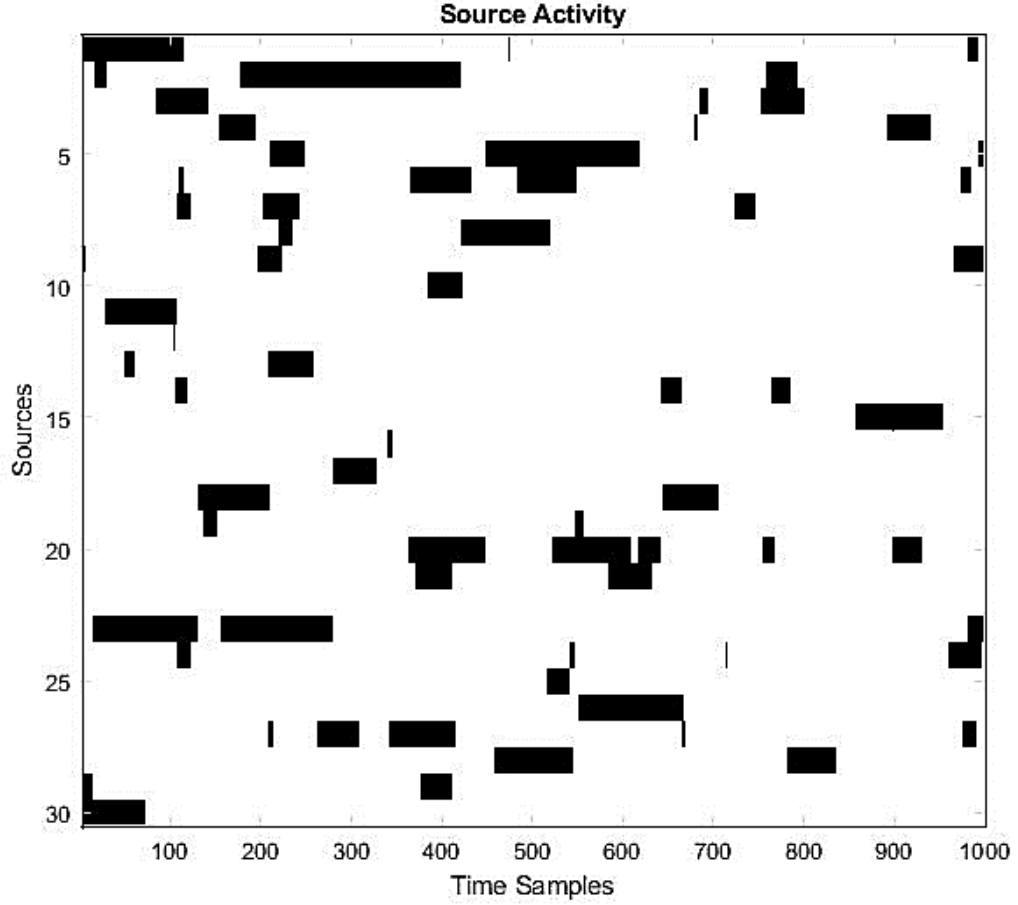


where for all  $t$ ,  $D^{(k)}$  is a diagonal matrix with elements having indices in  $S(x^{(k+1)}(t))$  equal to 1 and otherwise 0.  $S(x)$  is the set of non-zero elements in the vector  $x$ . The iteration is initialized with  $X^{(1,k)} = X^{(k+1)}$ . For a fixed sparsity pattern  $S(x^{(k+1)}(t))$ , MDU alternatively estimates channel and signals.

We present the analysis to obtain insights into the performance of these DL-based estimation algorithms in the presence of time-varying channels. As performance criteria, the probability of false alarm and probability of detection are selected. The probability of detection is the ratio of the number of correctly detected active sources and the total number of active sources over  $T$  time samples; and the probability of false alarm is the ratio of the number of incorrectly detected active sources and the total number of inactive sources over the same  $T$  time samples.

Unless otherwise mentioned, the number of sources  $N=30$ , the number of sensors  $M=20$ . We assume an HMM for the state  $s_n(t)$ , the transition probabilities are  $p_n = 0.0022$  and  $q_n = 0.02$  for all  $N$  sources, such that an average  $N p_n / (p_n + q_n) = 3$  sources are active in each time sample  $t$  and the average duration of transmission is  $1 / q_n = 50$  time samples.

Figure 4.1 depicts the source activity of said system over a 1000 time samples with parameters as defined here. There are 30 sources whose activity is traced over a duration of 1000 time samples. We observe that once switched on, a source tends to remain active for a continuous stretch of time. Also, for most of the observation period, most of the sources are inactive, which is a result of basing the model on a sparse matrix.

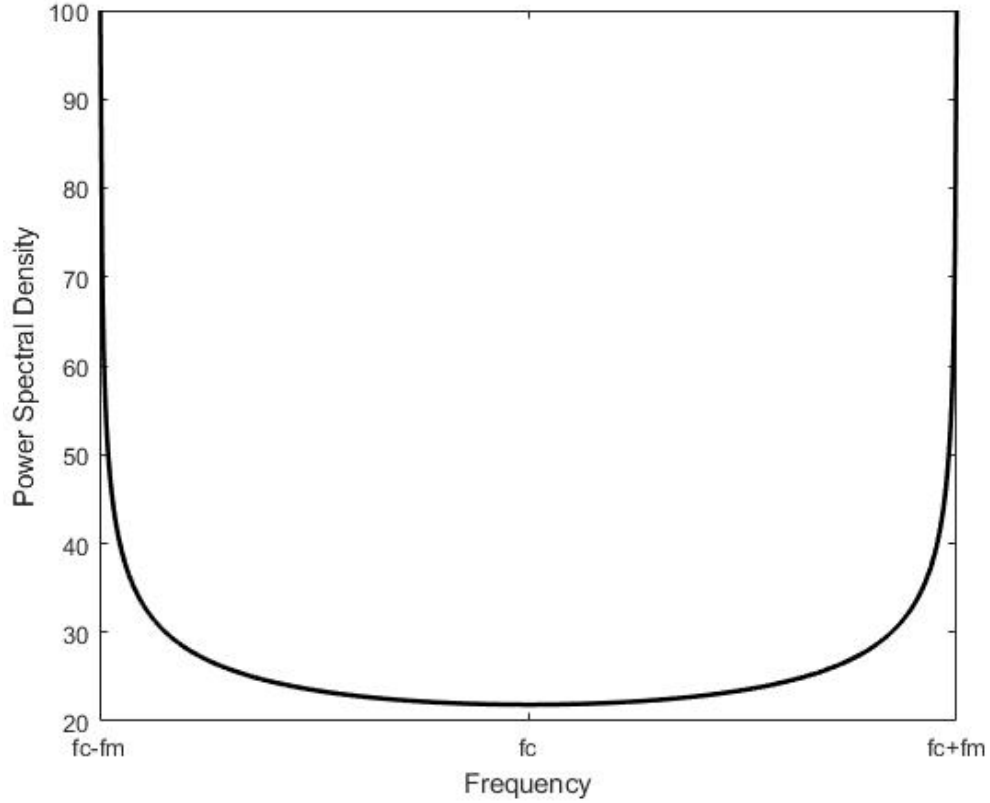


**Figure 4.1** Source Activity Diagram of  $N=30$  sources over  $T=1000$  time samples. Average of 3 sources are active in each time sample. Average duration of transmission = 50 time samples.

In the next part, we discuss the Dictionary Learning algorithms' sensitivity to number of samples being tested for a channel that changes with time. With this change, Doppler is introduced and the performance is compared with respect to a channel which remains fixed over all 1000 time samples. We will also elaborate on the sensitivity to Doppler Effect where we subject the system to different Doppler Shifts and study its effect on the learning algorithms.

#### 4.1 Sensitivity to Number of Observations

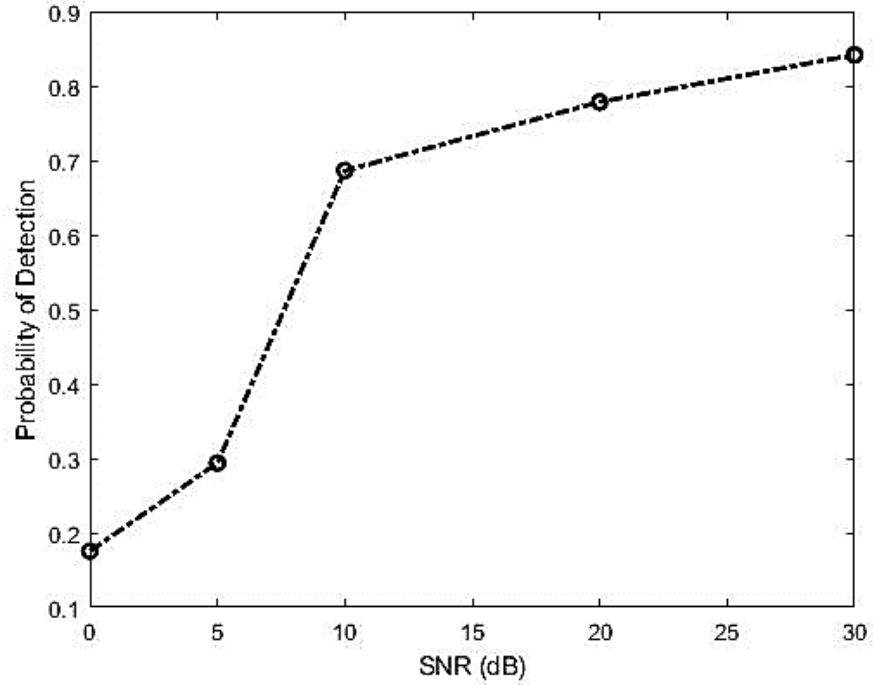
We subject the system to a channel of carrier frequency  $f_c = 1$  GHz and maximum Doppler frequency  $f_m = 1$  kHz. The frequency band of all arriving components increases to  $f_c \pm f_m$ . Figure 4.2 depicts the power spectral density of the resulting signal due to Doppler fading with above conditions.



**Figure 4.2** Power Spectral Density for carrier frequency  $f_c = 1$  GHz and maximum Doppler frequency  $f_m = 1$  kHz.

We plot the probability of detection for a fixed probability of false alarm with respect to varying Signal-to-Noise Ratio (SNR) in a fixed channel that does not change with time as control with 1000 time samples. We observe the performance of the algorithms at 0, 5, 10, 20 and 30 dB, and the probability of detection ( $P_d$ ) corresponding to probability

of false alarm  $P_{fa} = 0.2$ . At different SNR values, for  $P_{fa} = 0.2$ , the results are documented in Table 4.1.



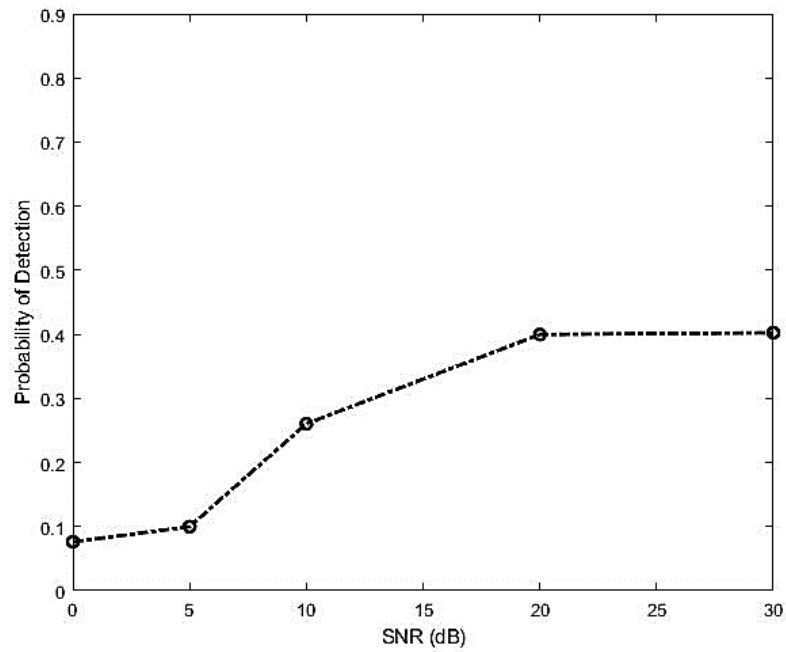
**Figure 4.3** Probability of detection for varying SNR (dB) when probability of false alarm is 0.2 for 1000 observations of a fixed channel.

**Table 4.1** Probability of detection when  $P_{fa} = 0.2$  with changing SNR when number of observations = 1000 in a fixed channel

SNR (dB)	$P_d$
0	0.175
5	0.3
10	0.68
20	0.78
30	0.85

When there is relative motion between the sources and the sensors, the channels between them change with time. The amount of change in the channels can be calculated as a function of the velocity of motion, the wavelength and the angle of arrival. The angle of arrival is uniformly distributed over an interval 0 to  $2\pi$ . The azimuthal gain pattern for given angle or arrival distribution is assumed to be 1.5 and the antenna is a quarter wavelength antenna.

Thus implemented time-varying channel is used to transmit the signal generated using the HMM and 1000 observations are used in the learning algorithm, the probability of detection for SNR = 0, 5, 10, 20, 30dB are plotted at fixed probability of false alarm of 0.2 are tabulated in Table 4.2.

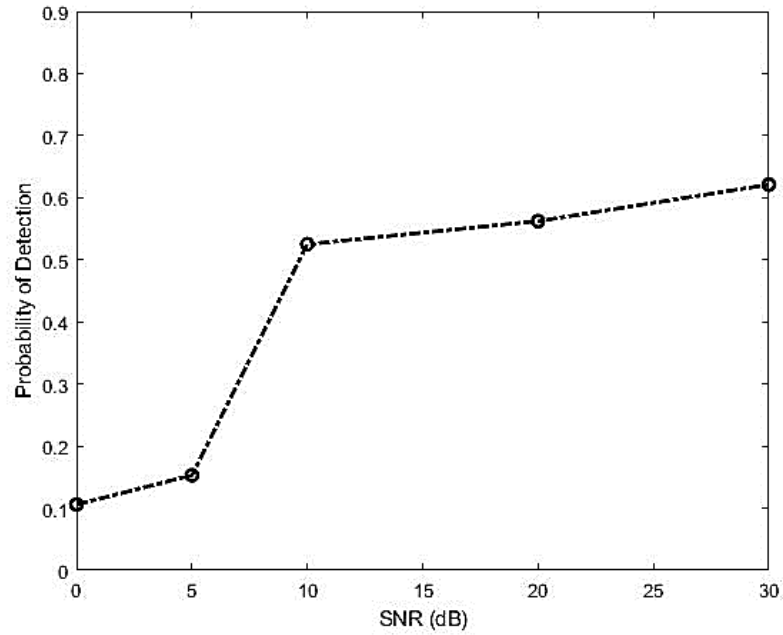


**Figure 4.4** Probability of detection for varying SNR (dB) when probability of false alarm is 0.2 for 1000 observations in a time-varying channel.

**Table 4.2** Probability of detection when  $P_{fa} = 0.2$  with changing SNR when number of observations = 1000 in a time-varying channel

SNR (dB)	$P_d$
0	0.07
5	0.09
10	0.26
20	0.39
30	0.40

Figure 4.5 plots the same parameters when 670 observations of a time-varying channel are fed in to the algorithms. Table 4.3 documents these values for different SNRs.

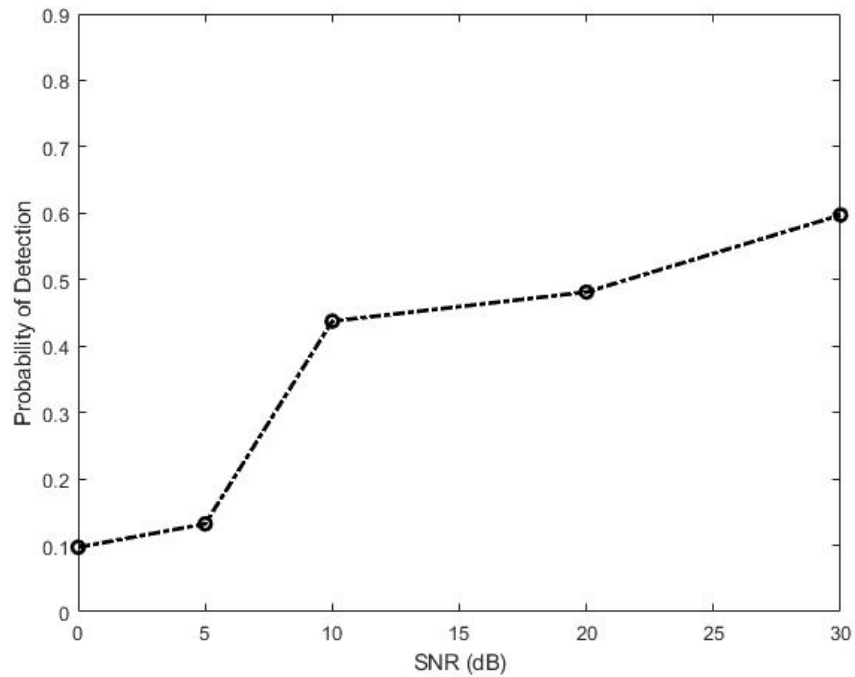


**Figure 4.5** Probability of detection for varying SNR (dB) when probability of false alarm is 0.2 for 670 observations in a time-varying channel.

**Table 4.3** Probability of detection when  $P_{fa} = 0.2$  with changing SNR when number of observations = 670 in a time-varying channel

SNR (dB)	$P_d$
0	0.1
5	0.15
10	0.53
20	0.57
30	0.62

Figure 4.6 plots the same parameters when 330 observations of a time-varying channel are fed in to the algorithms. Table 4.4 documents these values for different SNRs.



**Figure 4.6** Probability of detection for varying SNR (dB) when probability of false alarm is 0.2 for 330 observations in a time-varying channel.

**Table 4.4** Probability of detection when  $P_{fa} = 0.2$  with changing SNR when number of observations = 330 of a time-varying channel

SNR (dB)	$P_d$
0	0.09
5	0.13
10	0.44
20	0.48
30	0.58

We observe certain changes in the performance of the learning algorithms when we change the number of observations over which the channel is sampled. With the change in number of observations, the performance of the algorithms also changes. For a fixed level of Doppler Effect in a channel, the variation in the channel with every passing sample of time increases. The amount of change in the channel increases with increasing number of time instances over which the channel is observed.

When the time-varying channel is observed via a total of 1000 observations, the variation in the channel values is too high and performance degrades very drastically and  $P_d$  is merely 0.4 for  $P_{fa} = 0.2$  even for a SNR as high as 30dB. Reducing the number of observations to  $2/3^{\text{rd}}$  the value, i.e. 670 values, we see improvement in the performance and  $P_d$  goes up to 0.62 for the same conditions. However, following the same process and decreasing the number of observations to  $1/3^{\text{rd}}$  or 330 values doesn't follow the same pattern and  $P_d$  degrades to 0.58 because there are not enough samples for the algorithm to work on.



The channel matrix  $H$  for a channel which is spread over too long a time duration or too many time samples has too high a variance between the elements of the matrix for the Dictionary Learning algorithms to correctly work on the received data to estimate the transmitted information or the signal matrix  $X$ . However, when observed for too little time and very few time samples, there ceases to be enough information for the algorithms to use to estimate the signal matrix  $X$  and the channel matrix  $H$ . The channel matrix is fed back into the iterative DL algorithms which is used to estimate the sent signal or the matrix  $X$ .

We work in the later sections of this chapter to figure out the appropriate number of time samples to observe to reach a trade-off between too many and too few observations and optimal performance. We calculate coherence time for different Doppler frequency values and compare the performances of the algorithms.

Also, for varying SNR values, we notice that performance for different number of observations does not degrade linearly. When SNR = 0dB the performance of the system for any given number of observations is very poor. When SNR is increased to 5dB, performance improves slightly but is still not enough to justify the use of these algorithms to estimate the transmitted signal as probability of detection is comparable to the probability of false alarm, that is to say that any element of the signal matrix can be correctly or incorrectly estimated with the same probability and is highly undesired. The performance is comparatively much poorer for lower values of SNR up to 5dB.

After 5dB, there is a rapid increase from 10dB to 20dB and 30dB. The algorithms provide probability of detection much higher than the probability of false alarm and is similar to any system that can be used in real life scenarios. From SNR = 10dB and higher,

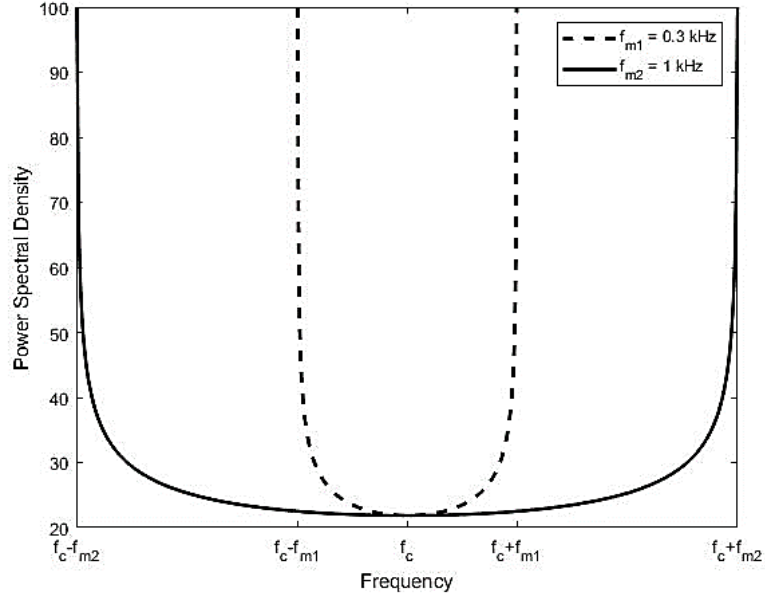
they do not yield as extreme a change in performance as it does for the jump from 5dB to 10dB. The performance for these SNR values are close to each other and improves linearly.

For the next section, we compare the algorithms for the higher SNR values of 10dB, 20dB and 30dB for different Doppler frequencies and discuss how the nature of the curves change on changing these two parameters.

## 4.2 Sensitivity to Doppler Effect

The transmitted signal travels through a channel with carrier frequency  $f_c = 1$  GHz. This channel undergoes movements with two different velocities in such a way that there is Doppler Effect arising from it. The channel is subjected to two different maximum Doppler frequencies  $f_{m1} = 0.3$  kHz and  $f_{m2} = 1$  kHz. Due to the different Doppler frequencies, certain characteristics of the channel changes inherently. The performance corresponding to these frequencies with changing number of observations as noted in the previous section are used to compare the sensitivity of the learning algorithms to Doppler Effect. Figure 4.7 depicts the power spectral density of the resulting signal due to Doppler fading with above conditions.

Here too, the same assumptions, of an average of 3 sources are active during any time instance and average duration of transmission of a source is 50 time samples, are used. In this section, we plot the probabilities of detection and false alarm of the algorithms for different channels for higher SNRs of 10dB, 20dB and 30dB, each with changing number of observations. This is done to better come up with an estimate of how many samples would give the best trade-off. We will see in a later section that it can be calculated using the maximum Doppler frequency.



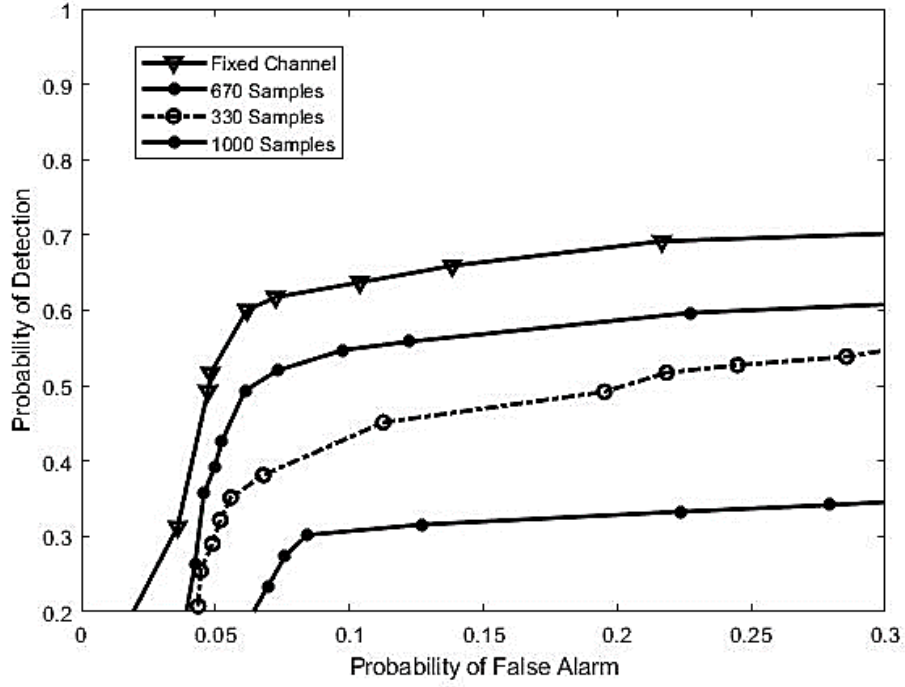
**Figure 4.7** Power Spectral Density for carrier frequency  $f_c = 1$  GHz and maximum Doppler frequencies  $f_{m1} = 0.3$  kHz and  $f_{m2} = 1$  kHz.

The curves correspond to systems with fixed channel not changing with time (which is used as a measure of best-case performance), and different number of observations taken for a time-varying channel which introduces the Doppler Effect in the system for SNRs of 10dB, 20dB and 30dB. These numbers are changed from a 1000 observations to two-thirds or 670 samples and down to one-third or 330 samples over which the system is observed.

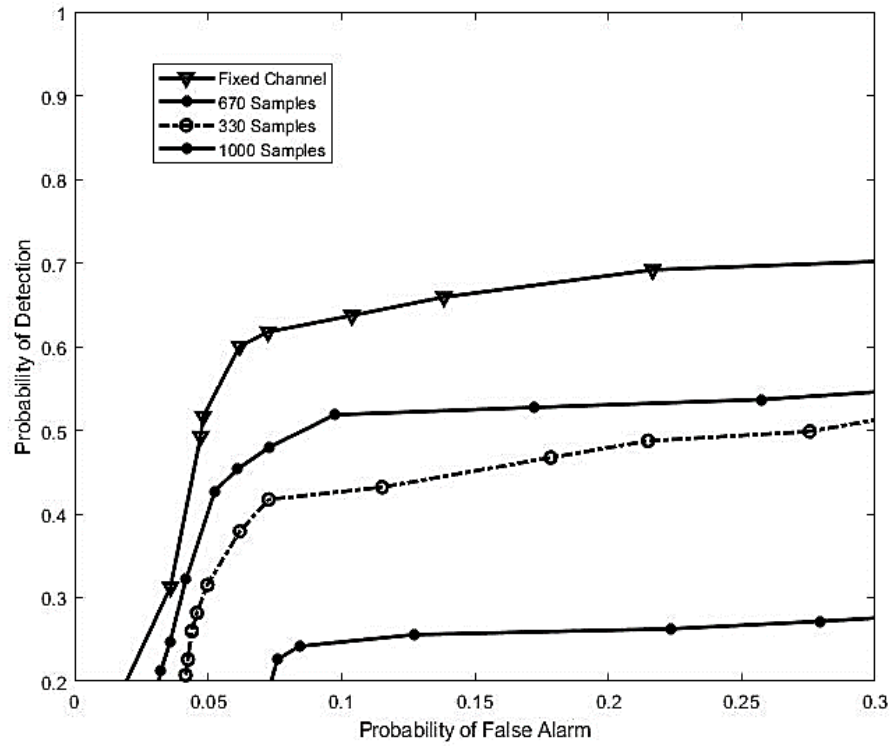
When SNR = 10dB, the performance is traced in figure 4.8 with Doppler frequency  $f_{m1} = 0.3$  kHz, and figure 4.9 does the same with Doppler frequency  $f_{m2} = 1$  kHz.

Figures 4.10 and 4.11 show the performance when SNR = 20dB for  $f_{m1} = 0.3$  kHz and  $f_{m2} = 1$  kHz respectively.

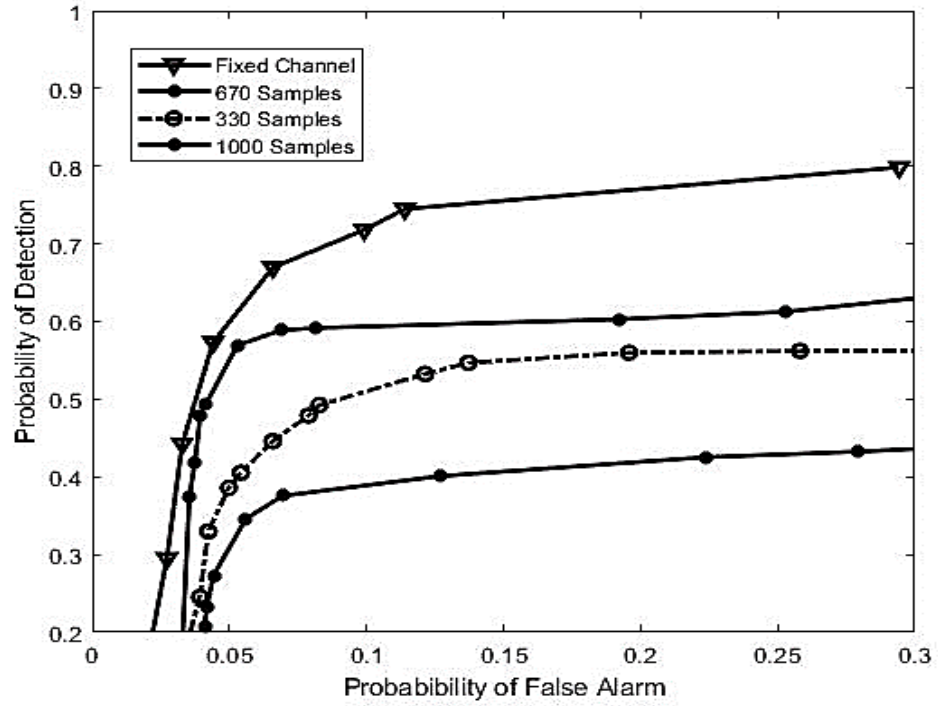
Figures 4.12 and 4.13 plot the performance of the system for  $f_{m1} = 0.3$  kHz and  $f_{m2} = 1$  kHz respectively when SNR = 30dB.



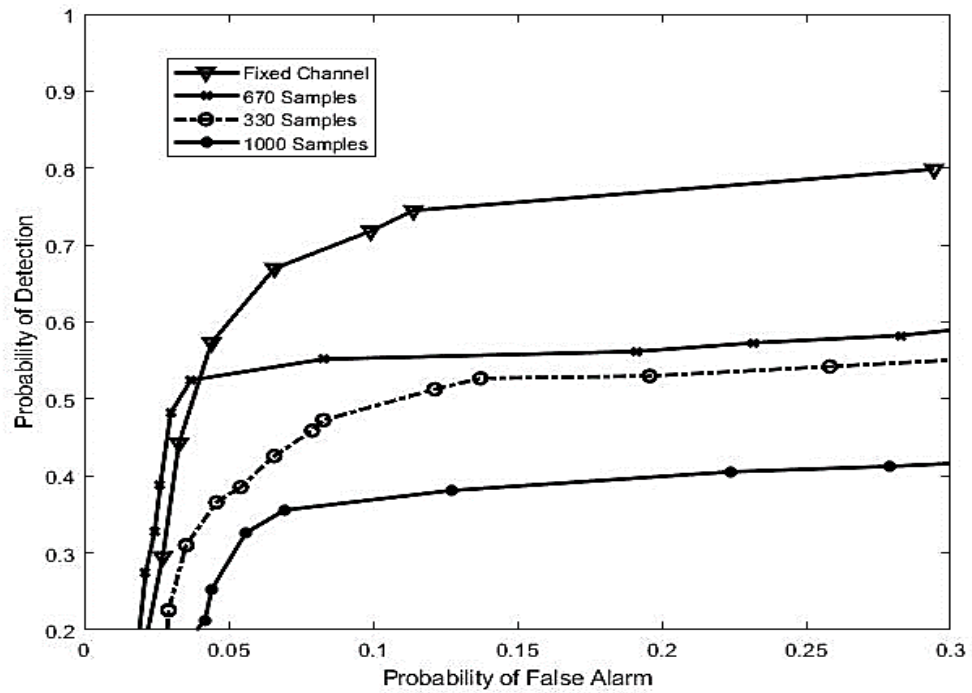
**Figure 4.8** Performance for SNR = 10dB when  $f_{m1} = 0.3$  kHz.



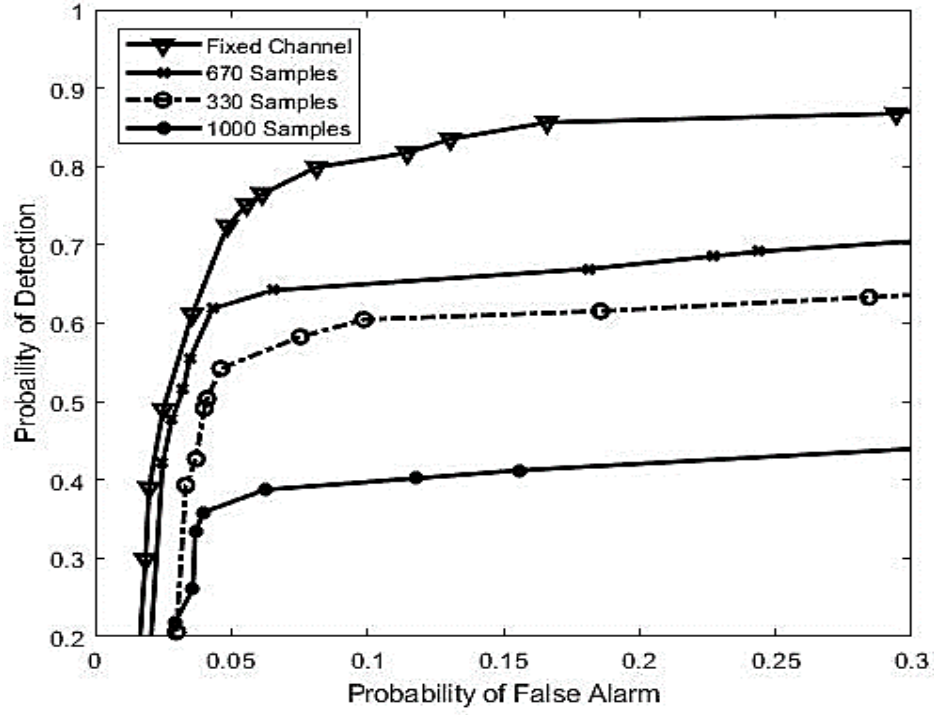
**Figure 4.9** Performance for SNR = 10dB when  $f_{m2} = 1$  kHz.



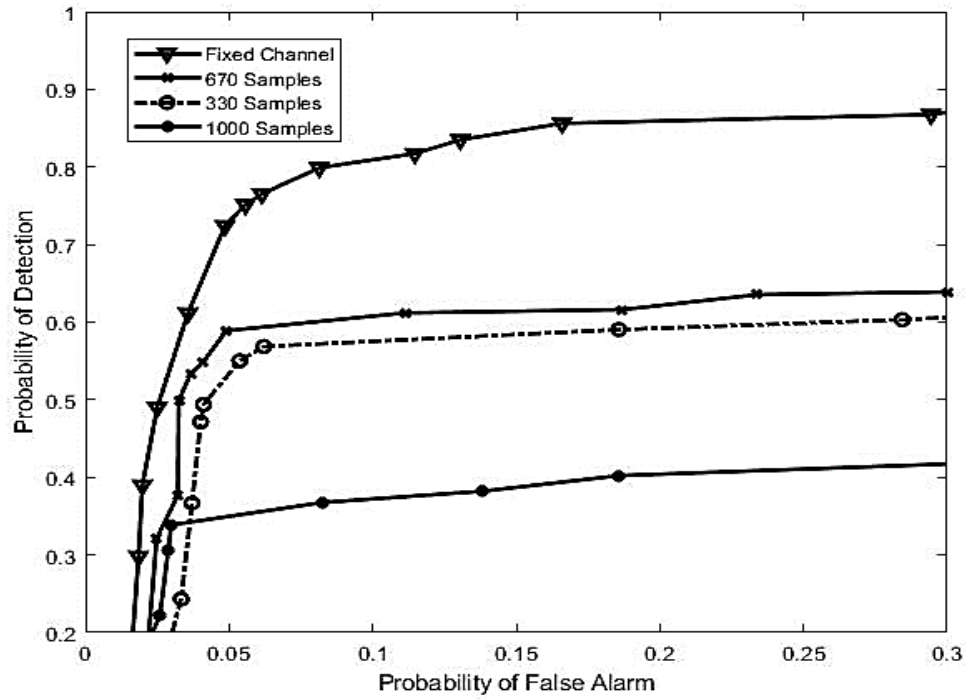
**Figure 4.10** Performance for SNR = 20dB when  $f_{m1} = 0.3$  kHz.



**Figure 4.11** Performance for SNR = 20dB when  $f_{m2} = 1$  kHz.



**Figure 4.12** Performance for SNR = 30dB when  $f_{m1} = 0.3$  kHz.



**Figure 4.13** Performance for SNR = 30dB when  $f_{m2} = 1$  kHz.

**Table 4.5** Probability of detection when  $P_{fa} = 0.2$  for different Doppler frequencies  $f_{m1} = 0.3$  kHz and  $f_{m2} = 1$  kHz (SNR = 10 dB)

Number of Observations	$f_{m1} = 0.3$ kHz	$f_{m2} = 1$ kHz
670	0.59	0.53
330	0.49	0.44
1000	0.32	0.26

**Table 4.6** Probability of detection when  $P_{fa} = 0.2$  for different Doppler frequencies  $f_{m1} = 0.3$  kHz and  $f_{m2} = 1$  kHz (SNR = 20 dB)

Number of Observations	$f_{m1} = 0.3$ kHz	$f_{m2} = 1$ kHz
670	0.61	0.57
330	0.57	0.48
1000	0.42	0.39

**Table 4.7** Probability of detection when  $P_{fa} = 0.2$  for different Doppler frequencies  $f_{m1} = 0.3$  kHz and  $f_{m2} = 1$  kHz (SNR = 30 dB)

Number of Observations	$f_{m1} = 0.3$ kHz	$f_{m2} = 1$ kHz
670	0.67	0.62
330	0.61	0.58
1000	0.43	0.40

We test the system after subjecting it to two different Doppler shifts ( $f_{m1} = 0.3$  kHz and  $f_{m2} = 1$  kHz). It is easily observed that it follows the same trend as seen in section 4.1. The case where we observe the channel through a 1000 samples provides the worst curve in any SNR value. Reducing the number reduces the amount of variation in the channel and improves performance, as is seen when number of observations is decreased to 670.

However, when we keep on reducing the number of observations, we limit the amount of information given to the learning algorithms. These algorithms use the amount of data fed into them to estimate the signal values. When it is reduced beyond a certain limit, in our case 330 samples, performance degrades due to the lack of sufficient information.

The tables 4.5-4.7 show the probabilities of detection for different number of observations in two time-varying channels each with Doppler frequencies  $f_{m1} = 0.3$  kHz and  $f_{m2} = 1$  kHz when probability of false alarm is 0.2 for different SNR values.

With change in number of observations, we notice that performance does not change linearly. A 1000 samples over a fixed channel is used as best case scenario whereas over a time varying channel (with both Doppler frequencies,  $f_{m1}$  and  $f_{m2}$ ), the same number of observations introduces too high a variation in channel parameters to be estimated correctly. When the number of observations is dropped to 670, performance improves but further reducing it to 330 renders the amount of information too little for the learning algorithms to work. On increasing the amount of Doppler in the system, the performance degrades linearly and continues to do so even for values higher than is mentioned in this section.

When Doppler frequency is  $f_{m2} = 1$  kHz, the channel changes too fast and performance deteriorates as seen in section 4.1. For SNR = 10dB and  $P_{fa} = 0.2$  the probability of detection changes from 0.26 for 1000 samples to 0.53 for 670 samples; for 330 samples we have  $P_d = 0.44$ . For SNR = 20dB and  $P_{fa} = 0.2$ , probability of detection increases from 0.39 to 0.57 for 1000 and 670 samples respectively. Lastly, for SNR = 30dB and  $P_{fa} = 0.2$ , detection improves from 0.4 to 0.62.



When Doppler frequency is reduced to  $f_{m1} = 0.3$  kHz, the performance is worse than that of a fixed channel but considerably better than that of  $f_{m2} = 1$  kHz. We see an improvement from  $P_d = 0.26$  to 0.32 for SNR = 10dB and  $P_{fa} = 0.2$  when 1000 samples are observed. For 670 samples, probability of detection goes up from 0.53 to 0.59. When SNR = 20dB, for 670 samples, probability of detection goes up from 0.57 to 0.61 and for 30dB, from 0.62 to 0.67.

From the results in sections 4.1 and 4.2, with respect to Doppler Effect the performance degrades linearly but the same cannot be said for the effect that changing the number of observations has on the system. So we need to come up with a method to predict how many observations should be taken into consideration for different Doppler frequencies. In the next section, we will see how to use the Doppler frequency to calculate how many observations should be sampled for optimal performance. We use the parameter known as time coherence to determine this. The relation between the maximum Doppler frequency and time coherence has been elaborated in the next section.

### **4.3 Time Coherence and Number of Sampled Observations**

Time coherence is a window in time over which the unmodulated carrier envelope remains unchanged [47], or to put it more mathematically, it is the time interval over which channel impulse responses are highly correlated. Because of this high correlation, to the receiver the channel seems to be fixed or unchanged over this time duration. Coherence time is a statistical measure of time duration over which the channel impulse response is essentially invariant and quantifies the similarity of the channel response at different times.

Doppler frequency is a parameter that introduces change with respect to time in a channel. Higher the Doppler Effect, more there is variation in a channel from one time instance to the other, spread over sufficient time instances, the channel completely changes and learning algorithms fail to use information from one iteration to estimate the next.

Doppler spread is a measure of the spectral broadening caused by the time rate of change of the mobile radio channel and is defined as the range of frequencies over which the received Doppler spectrum is non-zero. Coherence time is the time domain dual of Doppler spread and is used to characterize the time-varying nature of the frequency dispersiveness of the channel in the time domain [47]. Time coherence is inversely proportional to the maximum Doppler frequency: one describing over which the channel is unchanged and the other introducing change in said channel.

$$T_c \propto \frac{1}{f_m} \quad (4.13)$$

From [48], we further get:

$$T_c = \frac{9}{16\pi f_m} \quad (4.14)$$

Using these relations, we find out the coherence times for the two different Doppler frequencies we have used in section 4.2. Over these time durations, the channel appear to be fixed to the algorithms and better detection probabilities is displayed. Using equation (4.14), we calculate how many time instances would correspond for both  $f_{m1} = 0.3$  kHz and  $f_{m2} = 1$  kHz. Assuming each time sample in our channel has a duration of  $1\mu s$ , we can formulate how many samples should be fed into the learning algorithms. As long as the number of observations being sampled are less than the coherence time, the algorithms perform similar to the case of a fixed case.

We change the algorithm to break the received information matrix  $Y$  into desired number of segments, in a manner as described above. In the LASSO step of the entire DL process, we use a number of segments each containing the number of time samples as mentioned in equation (4.14) to estimate signal matrix  $X$ , let's call these segments  $X_1, X_2 \dots X_n$ . Before the process starts again, to estimate these two parameters in tandem as it reaches convergence, we append these segments together and start afresh for the next iteration. Doing this forces the algorithms to learn the information in seemingly fixed channels, thus giving better performance.

We see in section 4.1 that with changing number of observations, the performance of the algorithms also changes. In time-varying channels, a 1000 samples introduces too much variation whereas reducing it to 670 improves the performance but further reducing it to 330 deteriorates it again. Similar patterns also hold for the changes implemented in the algorithms in section 4.3. However, these number hold for the smaller runs during which the channel appear fixed, which are dictated by the maximum Doppler frequency.

For intents and purposes of demonstrating these results, the same Doppler frequencies are used as in section 4.2 and corresponding coherence times are calculated. We choose the viable number of time samples depending on how to break all time instances of the signal without losing any data or adding in any redundancy. Table 4.8 shows the coherence times and the number of samples chosen for both  $f_{m1} = 0.3$  kHz and  $f_{m2} = 1$  kHz.

**Table 4.8** Number of Observations in each individual segment based on maximum Doppler frequency

Maximum Doppler Frequency (kHz)	Coherence Time ( $\mu\text{s}$ )	Number of Observations in 1 segment
$f_{m1} = 0.3 \text{ kHz}$	600 $\mu\text{s}$	500
$f_{m2} = 1 \text{ kHz}$	180 $\mu\text{s}$	100

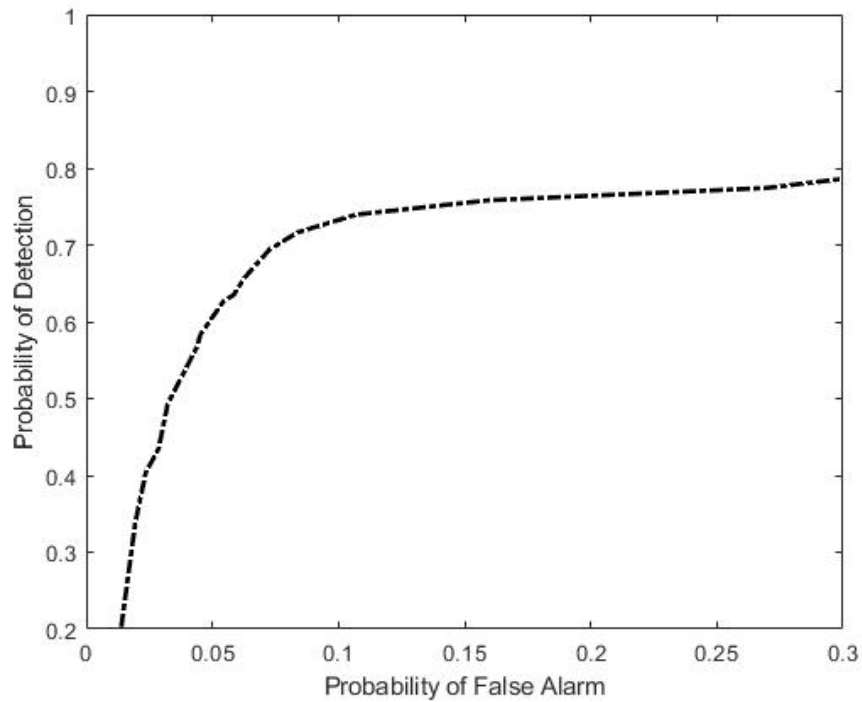
When Doppler frequency is  $f_{m1} = 0.3 \text{ kHz}$ , the coherence time using equation (4.14) equals to 600  $\mu\text{s}$ . The newer method dictates that the total duration be broken into a time smaller than the coherence time, so as make the channel appear fixed. Also, we want these segments to totally encompass all available information, not leaving anything out of consideration and not adding any redundancy either. Therefore, there are two segments over which the entire duration of activity is broken. We are sampling a total of 1000 time samples constituting a duration of 1000  $\mu\text{s}$ . When broken into segments of 500  $\mu\text{s}$ , there are two segments over which the inner loops are run for the learning algorithms.

When the case for  $f_{m2} = 1 \text{ kHz}$  is considered, the coherence time drops to 180  $\mu\text{s}$ . Following the same logic of taking a time smaller than this and breaking the entire duration into a number that evenly breaks it into segments, for this particular Doppler frequency, we choose to break it into ten segments and number of observations in a segment drops to 100. This is much smaller than the 330 samples we have used to estimate the signals and previous results would suggest degradation of performance. However, by introducing the segmented version of the DL algorithms, we see that performance improves from the case

of observing 670 samples, which is significantly better than the performance of observing 330 samples.

In the figures below, the performances of the new algorithm are attached. Figure 4.14 plots the probability of detection versus the probability of false alarm for  $f_{m1} = 0.3$  kHz broken into two segments (each of 500 samples) for the implementation of the new algorithm.

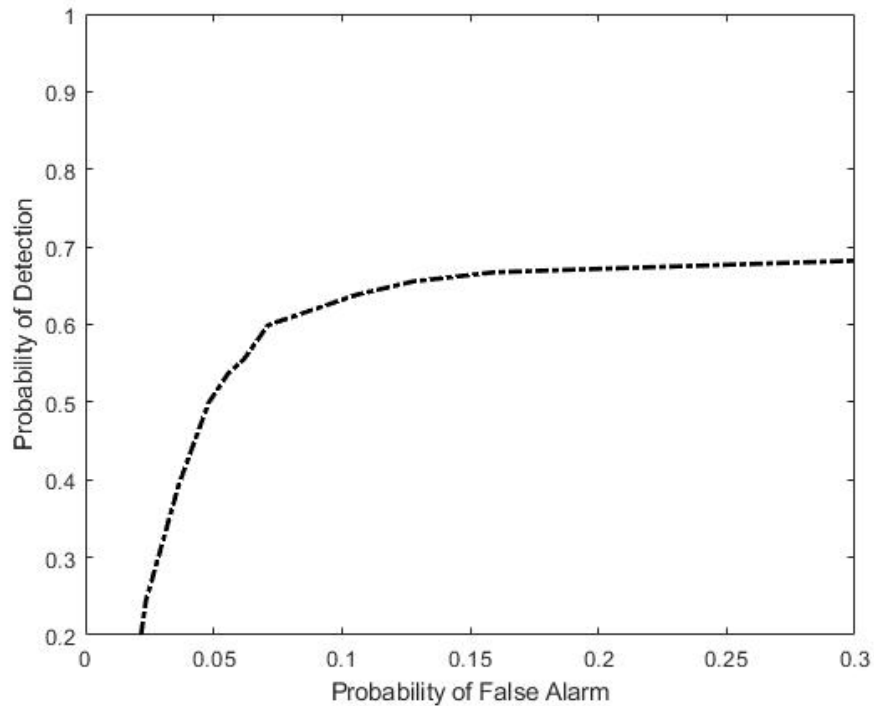
Figure 4.15 plots the same parameters ( $P_d$  versus  $P_{fa}$ ) when  $f_{m2} = 1$  kHz is used and the total duration is divided into ten segments each consisting of 100 time samples. SNR for both is considered to be 30dB.



**Figure 4.14** Performance for SNR = 30dB and  $f_{m1} = 0.3$  kHz when number of segments = 500.

From figure 4.14 it is notable that for probability of false alarm  $P_{fa} = 0.2$ , we have probability of correct detection  $P_d = 0.76$  when there are two segments each of 500  $\mu$ s. It

improves from the best case estimate ( $P_d = 0.67$ ) for same Doppler frequency with the previous implementation of the dictionary learning algorithms. Breaking the algorithm to use  $X_1$ , and  $X_2$  (each consisting of 500 time samples) in each step of learning and then appending before repeating the process till the learning reaches convergence shows a substantial improvement in the learning process. The performance is better than using 670 samples of a time-varying channel, which showed the best performance in section 4.2 for all values of SNRs. We have also shown earlier that reducing number of observed samples from 670 led to deterioration of performance. However, with this new implementation, performance improves drastically and is similar to a scenario when the channel is fixed and not affected by Doppler (for  $P_{fa} = 0.2$  and SNR = 30dB, we had  $P_d = 0.85$ ).



**Figure 4.15** Performance for SNR = 30dB and  $f_{m2} = 1$  kHz when number of segments = 100.

From figure 4.15 we see that for probability of false alarm  $P_{fa} = 0.2$ , we have probability of correct detection  $P_d = 0.68$  when there are ten segments each of 100  $\mu$ s. It improves from the best detection probability for the same probability of false alarm ( $P_d = 0.62$ ) for same Doppler frequency with the previous implementation of the dictionary learning algorithms. Breaking the algorithm to use  $X_1$ ,  $X_2$  and so on, onto  $X_{10}$  (each consisting of 100 time samples) in each step of learning and then appending before repeating the process till the learning reaches convergence shows a substantial improvement in the learning process. The performance is better than using 670 samples of a time-varying channel, which showed the best performance in section 4.2 for all values of SNRs. We have also shown earlier that reducing number of observed samples from 670 to 330 led to deterioration of performance. Following these results, a 100 time samples would provide performances much worse than desired for a practical system. However, with this new implementation, performance improves drastically and is even better to a scenario when the channel is estimated over 670 samples, which in section 4.2 was seen to provide best results.

From both these cases, we see that the new algorithm performs much better for time-varying channels than simply using the prototypical LASSO and MDU algorithms in tandem over the whole time duration during which the system exists. For both maximum Doppler frequencies  $f_{m1} = 0.3$  kHz and  $f_{m2} = 1$  kHz, detection improves by approximately 7% over the traditional way of implementing the learning algorithm.

When number of samples being observed are controlled, there can be a situation where information is too highly uncorrelated to be estimated correctly due to the presence of Doppler Effect. By constraining the number of samples in each inner iteration by

limiting it to a duration smaller than the coherence time, we make the channel appear non time-invariant to the estimation algorithms. Over each individual segment, performance improves and when combined, it is closer to that of a fixed channel when a Doppler frequency of  $f_{m1} = 0.3$  kHz is introduced. When the channel has a maximum Doppler frequency of  $f_{m2} = 1$  kHz, the performance improves from the original method of using the entire matrix for the learning process.

While simply using the LASSO and MDU over the whole duration of signal existence provides results worth applying in practical cases, when the channel is time-varying this method is not feasible due to the much lower detection probabilities for significant probabilities of false alarm. Breaking the total duration in segments controlled by the coherence time and adding an extra estimation step within said algorithms prove much more practically applicable with probabilities of detection improving drastically over constant probabilities of false alarms.



## CONCLUSION

A two-stage Dictionary Learning (DL)-based algorithm has been used to solve the Blind Source Separation (BSS) problem in the presence of radio sources with memory observed over time-varying channels. The sources feature intermittent activity and the number of latent sources may be larger than the total number of sensors.

The communication channels between the sources and the sensors are time-varying. The Doppler Effect due to mobility in wireless communication problems gives rise to deterioration of performance of the proposed learning approach. Controlling the time window over which the system is observed introduces change (for better or worse) in the performance. Using the probability of detection when the channels are stationary as a baseline, it is shown that there is significant degradation for time-varying channels. Over longer time, change in channel increases leading to poorer performance. Over shorter time duration, the information provided to the algorithm is too little to be *learned* from, which again leads to deterioration in detection.

The number of time samples to observe for optimum performance by the algorithm can be found using the coherence time of the channel. When the channel is learned during time windows shorter than coherence time, the algorithm finds the channels to be fixed and learning is greatly improved. However, when the maximum Doppler frequency is too high, the time coherence is too low, and the algorithms work with too few samples and performance deteriorates. However, the deterioration is improved over the total duration being observed without segmentation.

## REFERENCES

1. P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component analysis and Applications*. Cambridge: Academic press, 2010.
2. M. Centenaro, L. Vangelista, A. Zanella and M. Zorzi, “Long-range communications in unlicensed bands: the rising starts in the IoT and smart city scenarios”, *IEEE Wireless Communications*, vol. 23, pp. 60-67, 2016.
3. J. Gozalvez, “New 3GPP standard for IoT”, *IEEE Vehicular Technology Magazine*, vol. 11, pp. 14-20, 2016.
4. A. Hyvärinen and E. Oja, “Independent component analysis: algorithms and applications,” *Neural Networks*, vol. 13, pp. 411–430, 2000.
5. V.G. Reju, S.N. Koh, I.Y. Soon, “An algorithm for mixing matrix estimation in instantaneous blind source separation”, *Signal Process*, vol. 89, no. 9, 1762–1773, 2009.
6. S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometrics and Intelligent Laboratory Systems*, vol. 2, pp. 37–52, 1987.
7. G. H. Golub and C. H. Reinsch, “Singular value decomposition and least squares solutions”, *Numerische Mathematik*, vol. 14, pp. 403–420, 1970.
8. W. F. Xue, J. Chen, J. Q. Li and X. F. Liu, “Acoustical feature extraction of rotating machinery with combined wave superposition and blind source separation”, *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 220, no. 9, pp. 1423–1431, 2006.
9. F. Asano, S. Ikeda, “Evaluation and real-time implementation of blind source separation system using time-delayed decorrelation”, Paper presented at *The 2nd International Workshop on ICA and BSS*, Helsinki, Finland, 2000.
10. S. Rickard, The DUET Blind Source Separation Algorithm. In: Makino S., Sawada H., Lee TW. (eds) *Blind Speech Separation. Signals and Communication Technology*. Springer, Dordrecht, 2007.

11. K. J. Kim, C. S. Jang, J. -. Jeong and S. W. Nam, "Acoustic Echo Cancellation using the DUET Algorithm Based Blind Separation in a Noisy Environment," *TENCON 2006 - 2006 IEEE Region 10 Conference*, Hong Kong, pp. 1-4, 2006.
12. K. H. Knuth, Difficulties applying recent blind source separation techniques to EEG and MEG, *ArXiv e-prints*, Jan 2015.
13. F. P. do Carmo, J. T. de Assis, V. V. Estrela and A. M. Coelho, "Blind signal separation and identification of mixtures of images," *2009 Conference Record of the Forty-Third Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, pp. 337-342, 2009.
14. Q. Lin, F. Yin, H. Liang, Blind Source Separation-Based Encryption of Images and Speeches. In: Wang J., Liao XF., Yi Z. (eds) *Advances in Neural Networks – ISNN 2005*. Lecture Notes in Computer Science, vol 3497. Springer, Berlin, Heidelberg, 2005.
15. S. Aarki, R. Mukai, S. Makino, T. Nishikawa, and H. Saruwatari, "The fundamental limitation of frequency domain blind source separation for convolutive mixtures of speech", *IEEE Transaction on Speech and Audio Processing*, vol. 11, pp. 109-116, 2003.
16. T. Yoshioka, T. Nakatani, M. Miyoshi, and H. G. Okuno, "Blind Separation and Dereverberation of Speech Mixtures by Joint Optimization", *IEEE Transaction on Speech and Audio Processing*, vol. 19, pp. 69-84, 2011.
17. A. Chichocki, and S.-i. Amari, *Adaptive Blind Signal and Image Processing: Learning Algorithms and Applications*. Hoboken, NJ: John Wiley & Sons, 2002.
18. J. Karhunen, A. Hyvärinen, R. Vigario, J. Hurri, and E. Oja, "Applications of neural blind separation to signal and image processing," *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 5, pp. 131–134, 1997.
19. A. Jourjine, S. Rickard, and O. Yilmaz, "Blind separation of disjoint orthogonal signals: demixing N sources from 2 mixtures," *Acoustics, Speech, and Signal Processing, 2000. ICASSP '00. Proceedings. 2000 IEEE International Conference on*, 2000.
20. S. Wehr, I. Kozintsev, R. Lienhart, and W. Kellermann, "Synchronization of acoustic sensors for distributed ad-hoc audio networks and its use for blind source separation," *Multimedia Software Engineering, 2004. Proceedings. IEEE Sixth International Symposium on*, 2004.

21. S. M. Alavi and W. B. Kleijn, "Distributed linear blind source separation over wireless sensor networks with arbitrary connectivity patterns," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, pp. 3171–3175, March 2016.
22. S. S. Ivriigh, S. M.-S. Sadough, and A. A. Ghorashi, "A blind source separation technique for spectrum sensing in cognitive radio networks based on kurtosis metric," *Computer and Knowledge Engineering (ICCCKE), 2011 1st International eConference on*, Oct. 2011.
23. D. Obradovic, N. Madhu, A. Szabo, and C. S. Wong, "Independent component analysis for semiblind signal separation in MIMO mobile frequency selective communication channels," *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, July 2004.
24. S. R. Curnew and J. Ilow, "Blind signal separation in MIMO OFDM system using ICA and fractional sampling," *Signals, Systems and Electronics, 2007. ISSSE '07. International Symposium on*, Aug. 2007.
25. A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, 3rd ed. West Sussex, United Kingdom: John Wiley & Sons, 2004.
26. E.-J. Im, K. Yelick, and R. Vuduc, "Sparsity: Optimization framework for sparse matrix kernels," *The International Journal of High Performance Computing Applications*, vol. 18, 2004.
27. E. Alpaydm, *Introduction to Machine Learning*, 3rd ed., The MIT Press, Cambridge, Massachusetts, 2014
28. K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, Massachusetts: MIT Press, 2012.
29. A.A. Markov, *The Theory of Algorithms*, Trudy Mat. Inst. Steklov., vol. 42, pp. 3-375. Acad. Sci. USSR, Moscow–Leningrad, 1954.
30. C.E. Shannon, W. Weaver, *The Mathematical Theory of Communication*, The University of Illinois Press, 1971.
31. D. Chizhik, J. Ling, P. W. Wolniansky, R. A. Valenzuela, N. Costa, K. Huber, "Multiple-Input–Multiple-Output Measurements and Modeling in Manhattan", *IEEE Journal on Selected Areas in Communications*, vol. 21. No. 3, pp. 21–331, 2003.

32. R. H. Clarke, "A statistical theory of mobile-radio reception", in *The Bell System Technical Journal*, vol. 47, no. 6, pp. 957-1000, 1968.
33. M. J. Gans, "A power-spectral theory of propagation in the mobile-radio environment", in *IEEE Transactions on Vehicular Technology*, vol. 21, no. 1, pp. 27-38, 1972.
34. J. I. Smith. "A Computer Generated Multipath Fading Simulation for Mobile Radio", *IEEE Transactions on Vehicular Technology*, vol. 24, no. 3, pp. 39-40, 1975.
35. I. Todic and P. Frossard, "Dictionary learning," *IEEE Signal Processing Magazine*, vol. 28, pp. 27-38, 2011.
36. H. Ghanbari, H. Zayyani, and E. Yazdian, "Joint DOA Estimation and Array Calibration Using Multiple Parametric Dictionary Learning," *ArXiv e-prints*, Jul. 2017.
37. M. Razaviyayn, M. Sanjabi, and Z.-Q. Luo, "A Stochastic Successive Minimization Method for Nonsmooth Nonconvex Optimization with Applications to Transceiver Design in Wireless Communication Networks," *ArXiv e-prints*, Jul. 2013.
38. H. Raja and W. U. Bajwa, "Cloud K-SVD: A collaborative dictionary learning algorithm for big, distributed data," *IEEE Transactions on Signal Processing*, vol. 64, pp. 173-188, 2015.
39. Y. Ding and B. D. Rao, "Dictionary Learning Based Sparse Channel Representation and Estimation for FDD Massive MIMO Systems," *ArXiv e-prints*, Dec. 2016.
40. R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society*, vol. 58, no. 1, pp. 267-288, 1996.
41. L. N. Smith and M. Elad, "Improving dictionary learning: Multiple dictionary updates and coefficient reuse," *IEEE Signal Process. Lett.*, vol. 20, no. 1, pp. 79-82, 2013.
42. M. Sadeghi, M. Babaie-Zadeh, and C. Jutten, "Dictionary learning for sparse representation: A novel approach," *IEEE Signal Process. Lett.*, vol. 20, no. 12, pp. 1195-1198, 2013.
43. K. Engan, S. O. Aase, and J. Hakon Husoy, "Method of optimal directions for frame design," *Proc, IEEE ICASSP*, vol. 5, pp. 2443-2446, 1999.

44. S. K. Sahoo and A. Makur, “Dictionary Training for Sparse Representation as Generalization of K-Means Clustering”, *IEEE Signal Processing Lett.*, vol. 20, no. 1, pp. 79-82, 2013.
45. J. Chen, Z. J. Towfic, and A. H. Sayed, “Dictionary learning over distributed models,” *IEEE Trans. on Signal Processing*, vol. 63, pp. 1001–1016, 2015.
46. R. Gribonval and K. Schnass, “Dictionary identification- sparse matrix-factorization via  $\ell_1$ -minimization”, *IEEE Trans. on Information Theory*, vol. 56, pp. 3523-3539, 2010.
47. T. S. Rappaport, *Wireless Communications Principles and Practice*, 2<sup>nd</sup> ed., Prentice Hall, 2002.
48. R. Steele, *Mobile Radio Communications*, IEEE Press. 1994.
49. M. Parker, *Digital Signal Processing 101: Everything You Need to Know to Get Started*, Elsevier Science & Technology, 2017.
50. M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*, Springer, 2010.
51. T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2<sup>nd</sup> ed., Springer Series in Statistics, 2016.