A Novel Framework of Structured Measurement Matrix for Compressed Sensing in Wireless Sensor Networks

By
Edwin Flórez Gómez

A dissertation submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy
in
Computing and Information Science and Engineering

University of Puerto Rico
Mayagüez Campus
2018

Aproved by:

Kejie Lu, PhD President, Graduate Committee	Date
Henrick Mario Ierkic, PhD Member, Graduate Committee	Date
Emmanuel Arzuaga, PhD Member, Graduate Committee	Date
Arturo Portnoy, PhD Member, Graduate Committee	Date
Agustín Rullán, PhD Representative of Graduate Studies	Date
Jaime Seguel, PhD CISE PhD Program Coordinator	Date

Abstract of Dissertation Presented to the Graduate School
of the University of Puerto Rico in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

A Novel Framework of Structured Measurement Matrix for Compressed Sensing in Wireless Sensor Networks

By
Edwin Flórez Gómez
May, 2018

Chair: Kejie Lu
Ph.D Major: Computing and Information Science and Engineering

Wireless Sensor Network (WSN) is a wireless networking technology that can facilitate many important applications in our real life, from environmental monitoring to smart grid, and thus is a key component in the emerging Internet of Things (IoT). To design efficient WSNs, two major issues are (1) the throughput capacity that is the maximal data rate at which a WSN can collect data from a field, and (2) the delay that is the duration from the time a signal is sensed to the time that is a received by the gateway of a WSN, known as the sink. To address these two issues, there are many solutions in the literature and Compressed Sensing (CS) is one of the most promising solutions because it can combine data collection and compression at the same time. By using CS, sensor nodes can collaboratively generate measurements by using a measurement matrix to linearly combine the original signals. In theory, if the original signals are sparse, then they can be reconstructed by well-known convex optimization, using a much smaller number of measurements, which leads to much higher throughput and smaller delay.

In the literature, several CS based data collection schemes for WSNs have been investigated. However, the impact of the measurement matrix has not been fully investigated. On one hand, many researchers focused on the performance of WSN by simply assuming there exists a measurement matrix. On the other hand, some researchers focused on the design of measurement matrix, particularly the structured measurement matrix, without considering the feature of WSN. Therefore, there is still a significant gap between CS and WSN.

In this dissertation, we aim at tackling this challenging issue and we propose a novel framework for structured measurement matrix to improve the performance of data collection in WSN. Specifically, the proposed structured measurement matrix consists of rectangular blocks with non-zero elements. Each of the blocks is used to produce measurements by linearly combining original signals collected from a subset of sensors. Moreover, if two bands are adjacent in the matrix, then the corresponding subset can have intersections, and all such intersections have the same cardinality. In this manner, the measurement matrix is a circular overlapping block diagonal (COB) matrix.

To evaluate the performance of the proposed COB matrix, we first investigate a particular COB case, in which the cardinality of the intersection is one half of the cardinality of a subset of sensors. For this type of matrix, we conduct theoretical analysis to prove that it satisfies the Restricted Isometric Property (RIP), which is widely used to determine the minimal number of measurements that can guarantee the reconstruction of the original signals. The theoretical analysis reveals the impacts of several important factors, including the number of blocks, the sparsity of original signals, and the total number of signals. We also conduct extensive simulation and the numerical results validate the theoretical analysis and demonstrate that the proposed COB matrix outperform existing block diagonal matrix. Based on the understandings from the specific COB case, we generalize the COB in a way such that the size of overlapping can be an arbitrary number. For the generalized COB, we first prove that it also satisfies the RIP with a certain bound for the number of measurement. In addition to the aforementioned factors, we investigate the impact of the size of overlapping. Extensive numerical results show that our analysis again is very accurate.

Finally, we conduct theoretical analysis to evaluate the throughput and delay performance of CS-based WSN with the proposed measurement matrix. In the analysis, we first derive schemes to partition a unit area into equal-sized region, we then develop feasible time division multiple access (TDMA) schemes to facilitate two sensing scenarios in WSN. Using the theoretical analysis, we further analyze the performance of WSN using practical settings,

such as the transmission range, data rate, etc. The numerical results confirm that the proposed COB scheme can improve throughout and delay performance.

Resumen de Disertación Presentado a Escuela Graduada
de la Universidad de Puerto Rico como requisito parcial de los
Requerimientos para el grado de Doctor en Filosofía

Un nuevo modelo de matrices de medición estructuradas para el muestreo con compresión aplicado a redes de sensores inalámbricos

By
Edwin Flórez Gómez
Mayo, 2018

Consejero: Kejie Lu
Ph.D: Ciencias e Ingeniería de la Información y la Computación

Las redes de sensores inalámbricos (WSN, siglas en inglés) es una tecnología que facilita muchas tareas importantes en nuestra vida real, desde el monitoreo del medio ambiente hasta las redes inteligentes, componentes claves en el emergente campo del Internet de las Cosas (IoT, siglas en inglés). Para diseñar WSN eficientes, se deben considerar dos problemas principales (1) la capacidad de rendimiento, que es la velocidad máxima a la que un WSN puede recopilar datos, y (2) el retraso, que es el tiempo desde el momento en que se detecta una señal hasta que es recibida por la puerta de enlace de la WSN, conocido como el sumidero. Existen muchas soluciones en la literatura para abordar estos dos problemas, pero es el muestreo con compresión (CS, siglas en ingés) una de las soluciones más prometedoras, ya que permite combinar la recopilación de datos y la compresión al mismo tiempo. Al usar CS, cada sensor colabora en el sensado mediante el uso de una matriz para combinar linealmente las señales originales. En teoría, si las señales originales tienen muy pocas entradas no cero, entonces pueden ser reconstruidas usando el método conocido como optimización convexa, utilizando un número de mediciones muy pequeño. Esto conduce a un rendimiento mucho mayor y un retraso menor en la WSN.

En la literatura, se han investigado varios esquemas de recopilación de datos basados en CS para WSN. Sin embargo, el impacto de la matriz de medición no se ha investigado por completo. Por un lado, muchos investigadores se centran en el rendimiento de WSN simplemente asumiendo que existe una matriz de medición. Por otro lado, otros investigadores

se centraron en el diseño de la matriz de medición, particularmente la matriz de medición estructurada, sin considerar la característica de WSN. Por lo tanto, todavía hay una brecha significativa entre CS y WSN.

En esta disertación, nuestro objetivo es abordar este desafío y proponemos un nuevo modelo para la matriz de medición estructurada, que mejora el rendimiento de la recopilación de datos en WSN. Específicamente, la matriz estructurada propuesta consiste en bloques rectangulares que contienen elementos distintos de cero. Cada uno de los bloques se usa para producir mediciones que combinan linealmente las señales originales sensadas de un subconjunto de sensores. Además, si dos bandas son adyacentes en la matriz, su subconjunto correspondiente tendrá intersecciones de igual cardinalidad. De esta manera, la matriz de medición propuesta será diagonal circular con bloques solapados (COB, siglas en inglés).

Para evaluar el rendimiento de COB, primero investigamos un caso particular, en la que el tamaño de la intersección es la mitad del tamaño de un subconjunto de sensores. Para este tipo de matriz, se llevó a cabo un análisis teórico para demostrar que satisface la Propiedad Isométrica Restringida (RIP, sigles en inglés), que se usa ampliamente para determinar el número mínimo de mediciones que se necesitan para garantizar la reconstrucción de las señales originales. El análisis teórico revela el impacto de varios factores importantes, tales como, el número de bloques, la esparsidad de las señales originales y el número total de señales. También llevamos a cabo una simulación exhaustiva y los resultados numéricos validan el análisis teórico y demuestran que la matriz COB propuesta, supera a la matriz diagonal de bloques existente.

De lo observado en el caso específico de COB, hemos hecho una generalización, de tal manera que el tamaño de la superposición puede ser elegido. Para COB generalizada, primero demostramos que también satisface RIP con una cierta cuota para el número de medidas. Además de los factores antes mencionados, investigamos el impacto del tamaño de la superposición. Los extensos resultados numéricos muestran que nuestro análisis nuevamente es muy preciso.

Finalmente, realizamos un análisis teórico para evaluar la capacidad de rendimiento y el retraso en WSN basado en CS con la matriz de medición propuesta. En el análisis, primero derivamos esquemas para dividir un área específica en regiones de igual tamaño, luego desarrollamos esquemas de acceso múltiple por división de tiempo (TDMA, siglas en inglés) para facilitar dos escenarios de muestreo en WSN. Utilizando el análisis teórico, analizamos con mas detalle el rendimiento de WSN utilizando valores en la red, tales como, el rango de transmisión, la velocidad de datos, etc. Los resultados numéricos confirman que el esquema de COB propuesto puede mejorar la capacidad de rendimiento y los retrasos en la red.

To my wife Angy and my children Juanes and Salomé.

Also to my parents Cristobal and Ruth.

I cannot possibly imagine how happy they will be.

Acknowledgments

I would like to express my gratitude to my advisor for his academic, financial and emotional support. His confidence and persistence, feeling that I could accomplish what was proposed, made this investigation possible. His words at the right time along with his wise way of conducting research, has made me see how great and important this process is for a researcher.

I thank the technical staff of the Department of Science and Computer Engineering who in an agile and appropriate manner responded to requests for computation resources in this project.

I also want to thank those people who in some way or another helped during the process, the staff and professors of the CISE Department, Mr. Luis Irrizarry, Juan Valera, Rafael Aparicio, Hiva Samadiam and many others, thank you very much for helping to make this happen possible.

This work was supported in part by National Science Foundation (NSF) under grants CNS-0922996 and CNS-1730325.

Contents

List of Figures

List of Tables

List of Algorithms

Nomenclature

$[N]$	The set of whole numbers $1, 2, 3, \ldots, N$
δ_S	Restricted Isometry Constant (RIC) of a signal or vector with number of nonzero compoments S
$\delta_S(A, U)$	Restricted Isometry Constant (RIC) of a signal or vector with number of nonzero compoments S represented in the orthobasis U
$\mathbb{E}$	The expectation operator
$\mathbb{E}^p\|Z\|$	The *absolute moments* of the random variable Z, here defined by $(\mathbb{E}\|Z\|^p)^{1/p}$ for $p \geq 1$
$\mathbb{E}_\psi$	The expectation operator from a group of random variables
$\mathcal{C}(\mathcal{S}, \|\cdot\|, r)$	A *cover* for the set $\mathcal{S}$ at resolution r and with respect to the metric $\|\cdot\|$. A set is a cover if for every $x \in \mathcal{S}$ there exits x' in the cover such that $\|x - x'\| \leq r$
$\mathcal{N}(\mathcal{S}, \|\cdot\|, r)$	The *covering number* of $\mathcal{S}$ at resolution r with respect to the norm $\|\cdot\|$. Defined as the minimum cardinality of all covers of $\mathcal{S}$
$\mu(U)$	The coherence of a matrix $U_{N\times N}$, defined as $\sqrt{N} \max_{p,q\in[N]} \|U(p,q)\|$
$\|\cdot\|$	A classical norm definition
$\|\mathbf{x}\|_0$	The number of nonzero elements of the vector $\mathbf{x}$, it is not a norm.
$\|\mathbf{x}\|_\infty$	The maximum absolute entry of $\mathbf{x}$
$\|\mathbf{x}\|_p$	The l_p-norm for $1 \leq p < \infty$
$\|A\|_2$	The spectral matrix norm
$\|A\|_{\max}$	The largest entry of the matrix A in magnitude
$\|A\|_F$	The Frobenius matrix norm
$\|Z\|_{\psi_2}$	The sub-Gaussian norm, defined by $\sup_{p\geq 1} \frac{1}{\sqrt{p}} \mathbb{E}^p\|Z\|$
ϕ_i	A block in a diagonal matrix

Ψ	A Distinct Block Diagonal matrix (DBD)
$\sigma_s(\mathbf{x})_p$	The l_p-error of the best s-term approximation to a vector $\mathbf{x}$ (EBTA)
$\text{supp}(\mathbf{x})$	The support of a vector $\mathbf{x}$
$\mathbf{x}^*$	The nonincreasing rearrangement of the vector $\mathbf{x}$
$\widetilde{N}$	The size of the signal o vector related to block diagonal matrices
Ξ	A Repeated Block Diagonal matrix (RBD)
a	Some description
$a \lesssim b$	A semi-order when exists a absolute constant C such that $a \leq Cb$
$a \lesssim_\tau b$	A semi-order when exists a absolute constant C depends on some parameter τ such that $a \leq C_\tau b$
B_p^N	A l_p-ball
J	The number of blocks in a diagonal block matrix
N	The size of the signal o vector
U	A orthobasis U
U^*	The conjugate transpose of U
Z	A random variable
$z \circledast x$	The circular convolution between signals $\mathbf{z}$ and $\mathbf{x}$
DAS/DAQ	*Data acquisition* is the process of sampling signals that measure real world physical conditions and converting the resulting samples into digital numeric values that can be manipulated by a computer.
Rank(A)	The rank of a matrix A
RIP	Restricted Isometry Property

Chapter 1

Introduction

1.1 About Compressed Sensing

Although many of the mathematical bases for *Compressed Sensing* (CS) had already been found, it was professor Emmanuel Candès who started the recent studies. At the beginning of 2004, Candès was trying to take away the noise that he added in the famous Shepp-Logan Phantom image (similar to the one shown in Fig. 1.1). He decided to apply the technique called l_1 minimization to see if it could reduce a bit the simulated streaks in fuzzy images that occur when a Magnetic Resonance Imaging (MRI) is not given the necessary time to complete a scan. To his surprise, after the algorithm gave him the output, he observed that the returned image was almost perfect in all details. Justin Romberg was a postdoc engaged with Candès and they immediately began to work on answering why it happened. They explained the situation to professor Terrence Tao and later, in June 2004, they submitted a paper [1], which is considered as one of the two CS starting points. The other one was submitted by Donoho [2] in September 18 of the same year, who was Candès' doctoral advisor and had been doing research for a long time on similar situation. These pioneers published

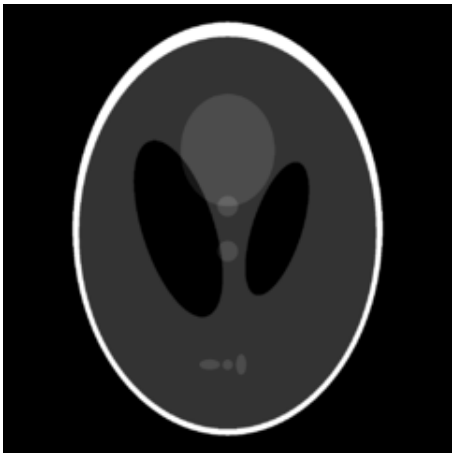


Figure 1.1: Shepp-Logan Phantom image[1]

several papers in CS, and [3] is the most cited one.

We can see CS as a signal processing technique to get data and compress it at the same time. Its main property is to make possible to reconstruct the signal efficiently by finding solutions to under-determined linear systems.

An *underdetermined system of linear equations*, i.e., more unknown that equations, can be represented as $Ax = y$ where the size of A is $\widetilde{M} \times \widetilde{N}$, $x \in \mathbb{C}^{\widetilde{N}}$, $y \in \mathbb{C}^{\widetilde{M}}$ and $\widetilde{M} < \widetilde{N}$. By basic linear algebra we know that this type of system can be inconsistent or consistent with infinitely many solutions. A vector x with at least S entries nonzero is called S-sparse. Compressed sensing is related with reconstruct an S-sparse vector (or signal) $x \in \mathbb{C}^{\widetilde{N}}$ from $Ax = y$ where $A \in \mathbb{C}^{\widetilde{M} \times \widetilde{N}}$ is called the measurement matrix and the system is underdetermined. In this case, it is clear that the system is consistent, i.e. there exist a signal that was already sensing by A and represented by sample y, and therefore should has solution. The problem is deciding which solution to select. The assumption of sparsity will help to find the original vector x.

As mentioned before, the emerging of CS looks like by chance. However, it has attracted significant attention because CS can help to manage the huge explosion of data collected

[1]Image taken from Wikipedia entry, copyright code: CC BY-SA 4.0.

by a variety of sensors with high dimensionality. In an article of The Economist magazine published in 2010 with title "The data deluge", it is mentioned that "Everywhere we look, the quantity of information in the world is soaring". One simple example is the resolution of digital cameras, which increases from 2 mega-pixel to 50 mega-pixel in a decade and keeps increasing.

1.1.1 Compressed Sensing Problem

Formally the CS problem is: to find the vector $x \in \mathbb{C}^N$ that is the unique S-sparse solution of $Az = y$ with $y = Ax$, that is, $\{z \in \mathbb{C}^N : Az = Ax, \|z\|_0 \leq s\} = \{x\}$.

It is equivalent to: finding the vector x that can be reconstructed as the unique solution of

$$\underset{z \in \mathbb{C}^N}{\text{minimize}} \|z\|_0 \quad \text{subject to } Az = y. \tag{P$_0$}$$

There are two related cases to solve the compressed sensing problem, first is how to design a stable measurement matrix such that the salient information in any s-sparse or compressible vector is not damaged by the dimensional reduction from $\mathbf{x} \in \mathbb{C}^N$ to $\mathbf{y} \in \mathbb{C}^M$. Another case is the reconstruction algorithm to recover x from a small number of measurement, i.e, $M \ll N$. In this dissertation we are focus in first case.

In both cases, let S be an index set. It can be proved that the set of $(2s) \times N$ matrices such that $\det(A) = 0$ form some $S \subset [N]$ with $\text{card}(S) \leq 2s$ has Lebesgue measure zero; hence, most $(2s) \times N$ matrices allow the reconstruction of every s-sparse vector $\mathbf{x} \in \mathbb{C}^N$ from $\mathbf{y} = A\mathbf{x} \in \mathbb{C}^{2s}$.

Even having a reconstruction procedure, in practice, solving (P$_0$) is not feasible. This is because a minimizer has sparsity at most s, the straightforward approach for finding it consists

of solving every rectangular system $A_S\mathbf{u} = \mathbf{y}$, or rather every square system $A_S^* A_S \mathbf{u} = A_S^* \mathbf{y}$, for $\mathbf{u} \in \mathbb{C}^S$ where S runs through all the possible subsets of $[N]$ with size s. The number $\binom{N}{s}$ of these subsets if too large therefore this approach is completely unpractical. For example, for $N = 1000$ and $s = 10$, we have to solve $\binom{1000}{10} \geq \left(\frac{1000}{10}\right) = 10^{20}$ linear system of size 10×10, but solving those system in 10^{-10} seconds, the time to solve (P_0) would be 10^{10} seconds, i.e., more than 300 years. In general, was proved by Natarajan in [4] that solving (P_0) using any possible approach is NP-hard.

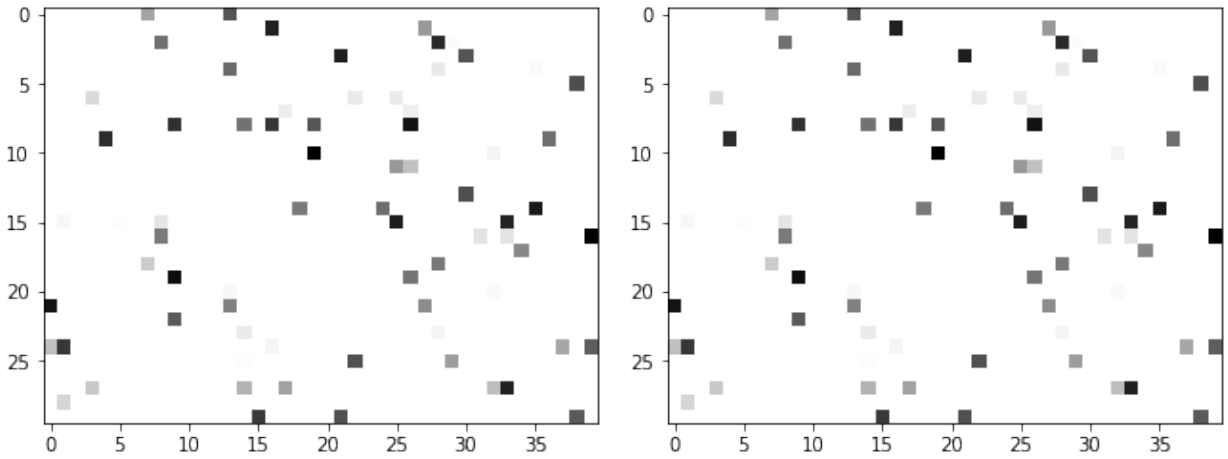
There exist another procedure without use the straightforward approach which include a partial Fourier matrix and is based on the Prony method [5]. This was know long before the development of compressive sensing. But still this method hides some important drawbacks and it is not stable with respect to sparsity defects nor is it robust with respect to measurement errors. It was proved in [6] that any *stable scheme* for s-sparse reconstruction requires at least

$$M \approx cs \log(eN/s) \tag{1.1}$$

linear measurements, where $c > 0$ is a constant depending on the stability requirement.

To illustrate the process with our own example, we create a grayscale random 30×40 pixel image showing in figure 1.2 on left. The image can be represented as a signal $x \in \mathbb{R}^N$ where $N = 1200$. We select a 6% of signal sparsity, i.e., 72 non-white pixels. When we sample it with a linear operator represented by a random 300×1200 matrix Φ and get $y \in \mathbb{R}^{300}$ as result, i.e., do $y = \Phi x$. And after give the sensing matrix Φ and sensed signal y to the l_1 solver program, it give us back the recover signal x^*. It search in $\mathbb{R}^N$ space amount the all possible solutions for the best one using the *important* property of x, be sparse. The error reconstruction is $\|x - x^*\|_2 \leq 7.9 \times 10^{-6}$, i.e., we can said, *vuala, x has been exactly recovered*. The reconstruction figure is showing on right.

In this example, we choose 300 samples. However, with a lower sample value $\widetilde{M} \geq 72 \cdot$



(a) Original random grayscale image (b) Recovered random gayscale image

Figure 1.2: Compressed Sensing Example

$\log 1200 \approx 221.70$, CS should also work as demonstrated in [7] and [2], which prove the bound in Eq. (1.1).

There are several applications where CS made a real impact and it is still being investigated very actively. One of the best known is single pixel camera that had its first appearance in [8], then it was improved in [9] and more recently in 2017 it has had a significant improvement in [10] using the technique called Compressive Ultrafast Sensing (CUS).

After more than 10 years of research using CS, there are already very good notes like [11] ,[12], [13] and [14] that can be used to introduce the topic. In addition, some books as [15] which focus in all mathematical aspect of CS. Or [16] that present the theory in a more friendly way and explain several applications including benefices and difficulties. There exists a book dedicated to Wireless Networks [17] that shows how to incorporate efficiently the idea of CS over assorted wireless network scenarios addressing all issues from an engineering perspective. A book by the Springer publishing house that will be released by 2019 called "Compressed Sensing: Applications to Communication and Digital Signal Processing" which focus in generalization of CS to more structured signal models could be an important resource in the developed of our future research.

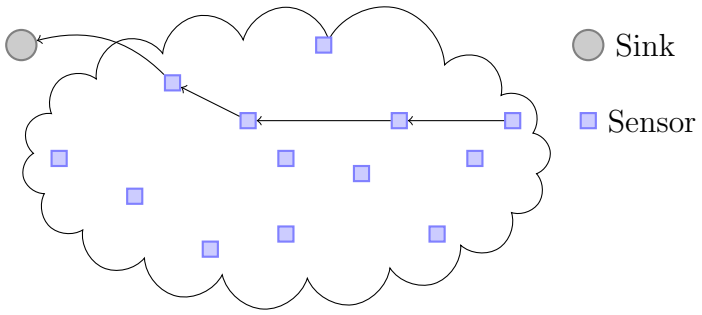


Figure 1.3: A base representation of a Wireless Sensor Network (WSN)

1.2 CS in Wireless Sensor Networks

Wireless Sensor Network (WSN) is a key component in the emerging Internet of Things (IoT). This technology can facilitate many important applications in real life, from environmental monitoring to smart grid. In this study, we consider a WSN that consists of one static sink and a total of N (or $\widetilde{N}$) static sensors as in figure 1.3. To design efficient WSNs, we aim to apply CS technology to address two major issues: (1) the throughput capacity that is the maximal data rate at which a WSN can collect data from a field, and (2) the delay that is the duration from the time a signal is sensed to the time that the signal is a received by the sink.

In the rest of this section, we will first discuss the data collection in WSN, and then we will elaborate on the proposed scheme for applying CS in WSN.

1.2.1 Data collection models

In a WSN, a *data collection model* specifies the way that sensors send their reading information to the sink, and it is considered the most essential issue that determines the performance of WSN. For the throughput and delay of wireless network, there are already many existing studies. To this end, one of the first studies is [18], in which the authors analyzed the ca-

pacity of two types of wireless networks: arbitrary networks and random networks, and they developed the theoretical bounds for capacity. Besides the capacity analysis, some studies, e.g., [19], investigated the connectivity issues in randomly deployed wireless networks.

For the performance of WSN, one pioneer work was done in [20], in which the authors analyzed the transport capacity of a data-gathering in WSN under one-to-one and many-to-one communication pattern. They discovered that the per-node throughput is bounded by

$$\lambda = \frac{W}{(2+\Delta)^2 N},$$

where W is the maximum transmission rate of a sensor and Δ represents the impact of interference.

There are numerous approaches for improve the performance of WSN. For example, in [21] and [22], the authors proposed to use mobile nodes, i.e., mobile sink and mobile relays, to improve the performance. For instance, the authors in [21] and [22] showed that the theoretical upper bound $\lambda = \frac{W}{N}$ can be achieved if the speed of sink is sufficiently large.

To further improve the throughput and delay performance, CS based data gathering schemes have been proposed. To explain how CS can help to improve the performance, let us first consider the traditional way to collect data from sensors to the sink through multi-hop relay without CS. To this end, a simple example is showed in Figure 1.4. In this example, we assume that there are three sensors and one sink, and that sensor s_1 sends one message x_1 to sensor s_2, s_2 has to forward message x_1 and send its own message x_2 to sensor s_3, and finally, the sink receives x_1, x_2 and x_3 messages from sensor s_3. In a scenario where there are N sensors s_i, the sink will receive all N messages from the sensor that is nearest to the sink.

To take advantage of CS in WSNs, it is necessary that the sink will receive M coded messages y_j for $j = 1, \ldots M$, where $M < N$. To collect data in WSN using CS, the first complete design was proposed in [23], where the authors proposed a Compressive Data Gathering (CDG)

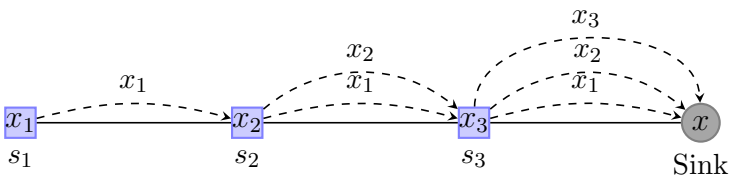


Figure 1.4: Traditional data collection

scheme, which is illustrated in Figure 1.5. Basically, the sink will received M weighted sums of all data x_i from nodes s_i. For example, for the first one weighted sum y_1, sensor s_1 will multiplies its data x_1 with a random coefficient ϕ_{11} and send that product to the next sensor s_2; sensor s_2 will do the same with its own data x_2 by multiplying it with ϕ_{12} and send the sum $\phi_{11}x_1 + \phi_{12}x_2$ to sensor s_3; finally, sensor s_3 also do the same for its own signal and then send $\phi_{11}x_1 + \phi_{12}x_2 + \phi_{13}x_3$ to the sink.

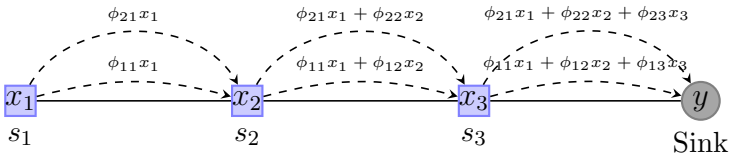


Figure 1.5: CS data collection

Let us assume that the sink can reconstruct $N = 3$ original signals if it receives $M = 2$ weighted sums y_1 and y_2, which are

$$y_1 = \phi_{11}x_1 + \phi_{12}x_2 + \phi_{13}x_3$$

$$y_2 = \phi_{21}x_1 + \phi_{22}x_2 + \phi_{23}x_3.$$

This example shows that CDG is better than the traditional data gathering scheme because the sink only needs to receive two messages in CDG, while it needs to receive three message in the traditional data gathering scheme.

1.2.2 The measurement matrices for CS in WSNs

For the example we just discussed, we can represent the previous situation as $y = \Phi x$, where $x = [x_1, x_2, x_3]^T$, $y = [y_1, y_2]^T$ and

$$\Phi = \begin{bmatrix} \phi_{11} & \phi_{12} & \phi_{13} \\ \phi_{21} & \phi_{22} & \phi_{23} \end{bmatrix}, \tag{1.2}$$

which is a full nonzero *unstructured random matrix*.

We can also observe from the example that, although sensor s_1 will send more messages in CDG than traditional model in Figure 1.4, sensor s_3 will send less. In a general chain topology, if $M \ll N$, then the sensor that connects to the sink will send fewer messages, which will increase the throughput and increase the network lifetime significantly. In addition, CDG can also reduce the total number of transmitted messages. In theory, CDG needs to send a total of $\Theta(MN)$ messages, while the traditional scheme needs to send $\Theta(N^2)$ messages in *overall complexity*.

In practice, sensors usually are deployed on a two-dimensional area. Moreover, the data collection typically follows a *tree topology*. Figure 1.6 shows an example in which there are four regions, each of which has one sub-tree for data collection.

Based on this tree topology, a full measurement matrix (like Eq. (1.2)) can be used with CDG. However, if this approach is used, an *important observation* is that each of the B branches or children of the sink has to send M measurements of the original signals and therefore the sink will receive a total of $BM > N$ messages, an obviously undesirable situation.

To avoid such a situation, [23] and most following studies assume that CS applies only to individual sub-trees. In Figure 1.6, there are 4 sub-trees, so for each one of them, the data collection presented in Figure 1.5 can be applied. In this way, the corresponding measurement

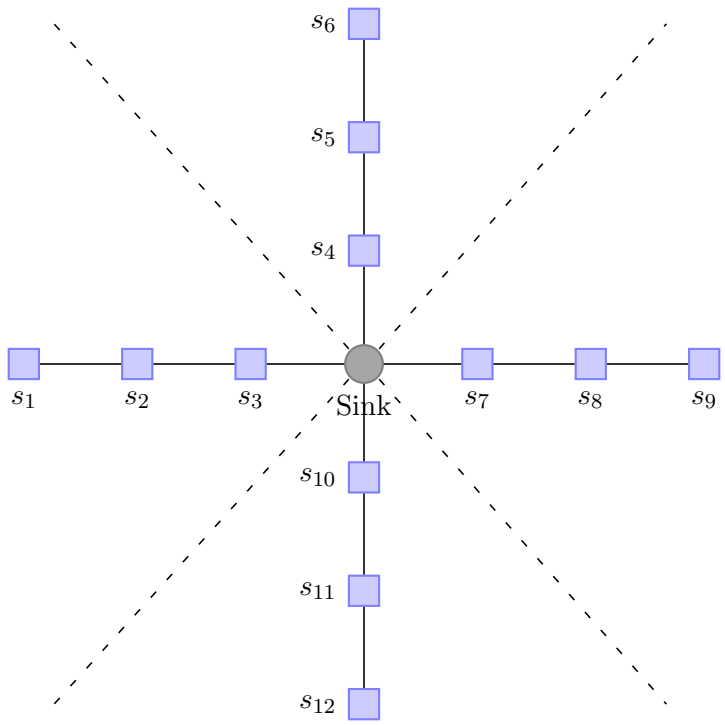


Figure 1.6: WSN two-dimensional area

matrix is not a full matrix but a *structured matrix*. Specifically, the matrix is a *block diagonal* (BD) matrix.

Before introduce our new matrix, let us see again the two-dimensional example, each fourth regions around the sink collect M data messages, then we will have a measurement matrix having a *Block Diagonal* (BD) structure

$$\begin{bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \\ \mathbf{y}_3 \\ \mathbf{y}_4 \end{bmatrix} = \begin{bmatrix} \Phi_1 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \Phi_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \Phi_3 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \Phi_4 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \\ \mathbf{x}_4 \end{bmatrix}, \tag{1.3}$$

where $\mathbf{y}_i = \Phi_i \mathbf{x}_i$ for $i = 1, \ldots, 4$ represents data sending by nodes in each of the four regions to the sink. In general, if a WSN sink has J children, then the BD matrix will have J blocks, each of which has a size $M \times N$, and the BD matrix has a size of $\widetilde{M} \times \widetilde{N}$, where $\widetilde{N} = JN$

and $\widetilde{M} = JM$.

1.2.3 Motivation

In the literature, the CDG has inspired many further investigation on CS-based data gathering in WSN. In [24], the authors considered sparsity in time and space domains for real dataset and they proposed a model using low-rank matrices. In [25], the authors investigated signals that follow a power-law decay model and they also proposed a new data gathering scheme that explores the spatio-temporal sparsity and they showed that the lifetime improvements can be up to two times over CDG. Their proposal is supported when a statistical characterization of the signal is available and thus a Bayesian inference [26] can complement conventional CS methods. Although there are many existing studies, to the best of our knowledge, none of them considers how to improve WSN performance by using new structured measurement matrix other than the BD matrix.

Intuitively, if the BD matrix is used, each block will be used to capture the correlations of sensor data collected from that region by linear equations. By exploiting the correlation, particularly the sparsity (or compressible) feature, we may be able to reconstruct N signals using M ($M < N$) measurements. Although this method is viable, it does not have a direct way to capture the correlations of data collected from different regions.

This observation has motivated us to study new structure for measurement matrix such that the correlation can be represented by linear equations. In addition, as we discussed before, a proper measurement matrix for CS needs to satisfy certain property to effectively recover the signals. This step was assumed in the CDG [23] and following studies, and later it was considered in [27] where a formal proof was presented. This connection also motivates us to carry out a comprehensive study to not only propose a new measurement matrix, but also formally prove that it satisfies certain property and demonstrate that the new matrix can

lead to better performance in WSN.

In this dissertation, we propose a novel framework for measurement matrix. Specifically, the proposed matrix can improve CDG by redefining the blocks such that adjacent blocks overlap. The main idea of the new matrix is shown in Figure 1.7, where region 2 on the right overlaps with regions 1 and 3 on the left and so on. Corresponding to this new partitioning scheme, we show how a BD matrix with 4 blocks can be transformed to an overlapped matrix with 8 block in Figure 1.8.

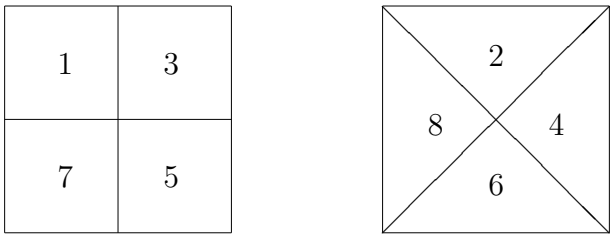


Figure 1.7: The partitioning of a square field.

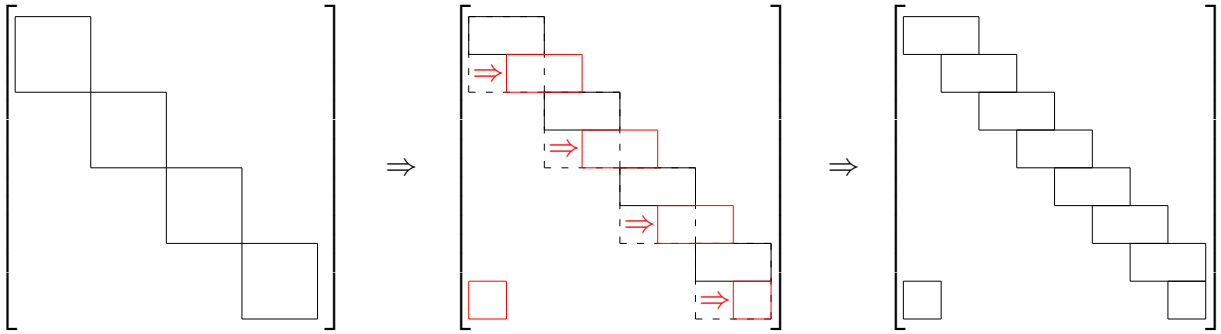


Figure 1.8: From BD to COB matrix

From this example, we can observe that some correlations of signals in region 1 and 3 can be directly represented by linear equations for region 2. Consequently, it is possible to use the new matrix to achieve a better performance than the performance of BD, in terms of the number of measurements M that are needed to recover the signal.

1.3 Objectives

The main objective in this research is to corroborate the observed hypothesis, i.e., it is possible to improve the performance of BD, in terms of the number of measurements M needed to recover the signal, by using the proposed matrix with overlapping blocks. To achieve this goal, we will develop formal proof to show that the proposed matrix satisfies the Restricted Isometric Property (RIP) and we plan to verify the proof through numerical experiments.

In our study, the second objective is to conduct theoretical analysis to evaluate the throughput and delay performance of CS-based WSN with the proposed measurement matrix.

1.4 Methodology

To prove the RIP property we are trying to use similar techniques developed in [27] and [28]. To reach that, we will analyze the material presented by Foucart & Rauhut in the book [15]. We will also study another book called "Compressed Sensing, Theory and Applications" [16]. Finally, we will investigate [29], which introduces a mathematical view of CS focusing specifically on recovery algorithms using l_1-minimization and structured random matrices.

At the same time, we will doing some experiments to figure out under which parameters the proposed matrix satisfies the RIP and show the differences comparing with the BD matrix presented in [27].

1.5 Thesis Outline

Chapter 2 includes a literature review of structure matrices and their theoretical bases that allow them to be used in CS. A table with the summary of all matrices in the review, the bound in the hypothesis of the theorem that makes it eligible to comply with RIP, as well as their respective references are presented in section 2.4. Some conclusions to the structure matrices review is presented in section 2.5.

Chapter 3 presents definition of the proposed cyclic overlapping block diagonal structure matrix in section 3.1. The whole section 3.2 gives a detailed list of steps to reach one of our main results, the property that guarantees our matrix can be used in CS. An important lower bound to the number of measurement needed to recover signals using our matrix is also presented in this section. Finally, in section 3.3, we use numerical experiments with empirical signal recovery through a minimizer program, to find the minimal number of measurements needed to recover signal using our proposed matrix. Those experiments validated each factor in the lower bound.

Chapter 4 describes all about the generalization of our new matrix and creates the propose Framework for CS. Section 4.1 gives the definition an their relations with others structures matrices. It includes figures where can observed the behavior when the structure becomes an unstructured matrix. Section 4.2 present the theoretical analyzes that guarantee its use in CS. Last section 4.3, is used to do numerical experiment with the framework.

Chapter 5 aims a way to apply CS in Wireless Sensor Networks using our proposed framework. In sections 5.3 and 5.4 the model is presented with its data collection scheme that applies CS technique as well as its partition and schedule analyses. In section 5.5, numerical experiments of scheme performance are included as another main result. Those results shows improvements on network throughput and delays with a standing performance over the other

model.

Finally, Chapter 6 presents the conclusions of this work as well as indicates possible paths for future research.

Chapter 2

Literature Review

In this chapter, we will present a survey of structured matrices for CS measurement. Over the past decade, CS has been used in a lot of application areas, including wireless channels, analog sampling hardware, sensor networks and optical imaging. In CS, the traditional measurement matrix is called the *unstructured matrix*, in which every entry is a non-zero random variable. To improve the performance of CS, various *structured matrices* have been proposed.

Before we start, let us present some definitions and properties from linear algebra and matrix analysis to get a better understanding of the review.

2.1 Definitions and properties

Throughout the dissertation, we refer to the set of positive integers up to N as $[N]$, and we let $\text{vec}(A)$, where A is an $M \times N$ matrix, represent the vectorization of a matrix, i.e., a linear transformation which converts the matrix into a column vector with size $MN \times 1$ by

stacking the columns of the matrix as

$$\mathrm{vec}(A) = [a_{11}, a_{21}, \ldots, a_{M1}, a_{12}, a_{22}, \ldots, a_{M2}, a_{1N}, a_{2N} \ldots, a_{MN}]^T. \tag{2.1}$$

Definition 1. *A nonnegative function $\|\cdot\| : X \to [0, \infty)$, where $X = \mathbb{R}^N$ or $X = \mathbb{C}^N$, is called a* norm *if*

1. *(Definiteness) $\|x\| = 0$ if and only if $x = 0$.*
2. *(Homogeneity) $\|\lambda x\| = |\lambda| \, \|x\|$ for all scalars λ and all vectors $x \in X$.*
3. *(Triangle inequality) $\|x + y\| \leq \|x\| + \|y\|$ for all vectors $x, y \in X$.*

A function holding 1 and 2 but 3 is replaced by the weaker quasi-triangle inequality

$$\|x + y\| \leq C \left(\|x\| + \|y\| \right),$$

for some constant $C > 1$, it is called a *quasinorm*. The smallest C is called its *quasinorm constant*.

Definition 2. *The l_p-norm (or simply p-norm) for $x \in \mathbb{R}^N$ or $x \in \mathbb{C}^N$ with $1 \leq p < \infty$ is equal to*

$$\|\mathbf{x}\|_p := \left(\sum_{j=1}^{N} |x_j|^p \right)^{1/p},$$

and for $p = \infty$

$$\|x\|_\infty := \max_{j \in [N]} |x_j|.$$

For $0 < p < 1$, $\|\mathbf{x}\|_p$ defines a quasinorm with constant $C = 2^{1/p-1}$.

2.1.1 Sparsity and Compressibility

The *sparsity* concept in CS is critical, it is the essential measure of signal complexity, and roughly speaking plays the same role in CS that *bandwidth* in the classical Shannon-Nyquist theory.

Definition 3. *The* support *of a vector* $\mathbf{x} \in \mathbb{C}^N$ *is the index set of its non-zero entries,*

$$\text{supp}(\mathbf{x}) = \{j \in [N] : x_j \neq 0\}.$$

Definition 4. *A vector* $\mathbf{x} \in \mathbb{C}^N$ *is called* S-sparse *if at most* S *of its entries are nonzero. In term of the support is,*

$$\|\mathbf{x}\|_0 := \text{card}(\text{supp}(\mathbf{x})) \leq S.$$

The cardinality of a set A is represented as $\text{card}(A)$. The expression $\|\mathbf{x}\|_0$ is used abusively because is not a norm or quasinorm. We can see it like a limit as p decrease to zero of the pth power of l_p-quasinorm of $\mathbf{x}$.

In real work the concept of sparsity is hard to get, therefore, let's introduce the weaker concept of *compressibility*, where we can consider signals almost S-sparse. To do that, we first introduce one way to measure if a vector $\mathbf{x}$ is nearly S-sparse with the following error.

Definition 5. *For* $p > 0$*, the* l_p-error of the best S-term approximation *(EBTA) to a vector* $\mathbf{x} \in \mathbb{C}^N$ *is defined by*

$$\sigma_S(\mathbf{x})_p := \inf\{\|\mathbf{x} - \mathbf{z}\|_p, \ \mathbf{z} \in \mathbb{C}^N \text{ is } S\text{-sparse}\}.$$

Note: The infimum may not be unique, is independently of $p > 0$ and $\sigma_s(\mathbf{x})_p$ is reached by an S-sparse vector $\mathbf{z} \in \mathbb{C}^N$ where its nonzero entries are equal to the S largest absolute

entries of $\mathbf{x}$.

Proposition 1. *For any $q > p > 0$ and any $\mathbf{x} \in \mathbb{C}^N$,*

$$\sigma_S(\mathbf{x})_p \leq \frac{1}{S^{1/p-1/q}} \|\mathbf{x}\|_p . \tag{2.2}$$

The proof needs a useful and natural definition based on the note given above.

Definition 6. *The* nonincreasing rearrangement *of the vector $\mathbf{x} \in \mathbb{C}^N$ is the vector $\mathbf{x}^* \in \mathbb{R}^N$ for which*

$$x_1^* \geq x_2^* \geq \cdots \geq x_N^* \geq 0$$

and there is a permutation $\pi : [N] \to [N]$ with $x^*_{\pi(j)} = |x_{\pi(j)}|$ for all $j \in [N]$.

Proof. If $x^* \in \mathbb{R}^N_+$ is the nonincreasing rearrangement of $\mathbf{x} \in \mathbb{C}^N$, we have

$$\begin{aligned}
\delta_s(\mathbf{x})_q^q = \sum_{j=s+1}^{N} (x_j^*)^q &= \sum_{j=s+1}^{N} (x_j^*)^{q-p}(x_j^*)^p \\
&\leq (x_s^*)^{q-p} \sum_{j=s+1}^{N} (x_j^*)^p \\
&\leq \left(\frac{1}{s} \sum_{j=1}^{s} (x_j^*)^p \right)^{\frac{q-p}{p}} \left(\sum_{j=s+1}^{N} (x_j^*)^p \right) \\
&\leq \left(\frac{1}{s} \|\mathbf{x}\|_p^p \right)^{\frac{q-p}{p}} \|\mathbf{x}\|_p^p \\
&= \frac{1}{s^{\frac{q}{p}-1}} \|\mathbf{x}\|_p^{q-p} \|\mathbf{x}\|_p^p . \\
\delta_s(\mathbf{x})_q^q &\leq \frac{1}{s^{\frac{q}{p}-1}} \|\mathbf{x}\|_p^q \\
\delta_s(\mathbf{x})_q &\leq \left(\frac{1}{s^{\frac{q}{p}-1}} \right)^{1/q} \|\mathbf{x}\|_p \\
\delta_s(\mathbf{x})_q &\leq \frac{1}{s^{1/p-1/q}} \|\mathbf{x}\|_p .
\end{aligned}$$

□

For now, we can call $\mathbf{x} \in \mathbb{C}^N$ a *compressible* vector if the EBTA decays quickly as S. From previous proposition and EBTA definition, compressible vectors belongs to the unit l_p-ball for some small $p > 0$,

$$B_p^N := \{\mathbf{z} \in \mathbb{C}^N : \|\mathbf{z}\|_p \leq 1\}.$$

In particular, the non-convex ball B_p^N for $p < 1$ serve as good models for compressible vectors.

2.1.2 Some Probability Theory

A probability space is represented by $(\Omega, \Sigma, \mathbb{P})$ where Σ denotes an algebra on the sample space Ω and $\mathbb{P}$ a probability measure on (Ω, Σ). The probability if an event $B \in \Sigma$ es denoted by

$$\mathbb{P}(B) = \int_B d\mathbb{P}(\omega) = \int_\Omega I_B(\omega) d\mathbb{P}(\omega). \tag{2.3}$$

Boole's inequality, also known as the *union bound*, says that for any finite or countable set of events, the probability that at least one of the events happens is no greater than the sum of the probabilities of the individual events. Formally, for a countable set of events $A_1, A_2, A_3, \ldots$, we have

$$\mathbb{P}\left(\bigcup_i A_i\right) \leq \sum_i \mathbb{P}(A_i). \tag{2.4}$$

The *expectation* or mean of a random variable X is denoted by

$$\mathbb{E}X = \int_\Omega X(\omega) d\mathbb{P}(\omega). \tag{2.5}$$

The values $\mathbb{E}X^p$ for integer p are called *moments* of X, while $\mathbb{E}|X|^p$, for real-valued $p > 0$,

are called *absolute moments*. The quantity

$$\mathbb{E}(X - \mathbb{E}X)^2 = \mathbb{E}X^2 - (\mathbb{E}X)^2 \tag{2.6}$$

is called *variance*.

For $1 \leq p < \infty$, $(\mathbb{E}|X|^p)^{1/p}$ defines a norm on the $L_p(\Omega, \mathbb{P})$-space.

The binomial distribution is the discrete probability distribution counting the number of successes in a sequence of N independent experiments where the probability of each individual success is p. If X has the binomial distribution, then

$$\mathbb{P}(X = k) = \binom{N}{k} p^k (1-p)^{N-k}. \tag{2.7}$$

The expectation of X is given by $\mathbb{E}X = pN$. If pN is an integer, then the median $M = M(X)$ coincides with the expectation $M(X) = pN$.

A normally distributed random variable or *Gaussian random variable* X has probability density function

$$\psi(t) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right). \tag{2.8}$$

The mean of a Gaussian random variable is $\mathbb{E}X = \mu$ and its variance $\mathbb{E}(X-\mu)^2 = \sigma^2$.

The Bernoulli distribution is a discrete probability distribution having two possible outcomes labelled by $n = 0$ and $n = 1$ in which $n = 1$ ("success") occurs with probability p and $n = 0$ ("failure") occurs with probability $q = 1 - p$, where $0 < p < 1$. Its probability density function is

$$P(n) = \begin{cases} 1-p & \text{for } n = 0 \\ p & \text{for } n = 1, \end{cases}$$

or, $P(n) = p^n(1-p)^n$.

Rademacher distribution is a discrete probability distribution where a random variate X has a 50% chance of being either $+1$ or -1. Its probability mass function is

$$f(k) = \begin{cases} 1/2 & \text{if } k = -1, \\ 1/2 & \text{if } k = +1, \\ 0 & \text{otherwise.} \end{cases}$$

It can be also written as a probability density function, in terms of the Dirac delta function, as $f(k) = \frac{1}{2}(\delta(k-1) + \delta(k+1))$.

A random variable Z is sub-Gaussian if its norm,

$$\|Z\|_{\psi_2} \coloneqq \sup_{p \leq 1} \frac{1}{\sqrt{p}} \mathbb{E}^p |Z|,$$

is finite.

2.1.3 Other CS related language

The property called *RIP and U-RIP* defined below is one of the mayor contribution in CS theory in [7], it basically helps to show that a linear transformation behaves as an isometric.

Definition 7. *The Restricted Isometry Constant (RIC) of a matrix $A \in \mathbb{R}^{m \times N}$ is defined as the smallest positive number δ_S for which*

$$(1 - \delta_S) \|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_S) \|x\|_2^2 \text{ for all } x \text{ with } \|x\|_0 \leq S. \tag{2.9}$$

Many interesting signals, audio, video, MRIs images or pictures in general are sparse or

compressive, i.e., have a low-dimensional geometric structure and can be represented by a properly base with much lower dimension that the original one. That can be exploited to design efficient signal acquisition and recovery methods.

The CS theory let sensing and compressing the signal directly in one step. It is well know now that sparse signals, i.e., signals with few nonzero entries, can be recovery with high probability from a small number of linear measurements using convex optimization problem but just if the measurement linear process system satisfies RIP. Basically, RIP helps to figure out that the measurement transformation process can preserve the geometry property from the signal space to the new measurement space.

If the signal is not directly sparse and it can be represented by a basis then the property has a little change. Let be $x \in \mathbb{R}^N$ and U an orthonormal basis such that $x = U\alpha$ were α is a sparse signal, then we will work with $U^*x = \alpha$. The following alternative RIP property is presented

Definition 8. *Let U denote an orthobasis for $\mathbb{C}^N$. The RIC of a matrix $A \in \mathbb{R}^{M\times N}$ in the basis U, is defined as the smallest positive number δ_S for which*

$$(1-\delta_S)\left\|x\right\|_2^2 \leq \left\|Ax\right\|_2^2 \leq (1+\delta_S)\left\|x\right\|_2^2 \text{ for all } x \text{ with } \left\|U^*x\right\|_0 \leq S.$$

A matrix A satisfies the restricted isometry property if δ_S is small for reasonably large S. The meaning of small δ_S and large value S, we will see later in chapter 3 where RIP for our matrix is proved.

The will use the notion of *semi-order* in this dissertation, i.e., $a \lesssim b$ means that there is an absolute constant C_1 such that $a \leq C_1 b$. If C_1 depends on some parameter c, we will write as $a \lesssim_c b$. Also $a \gtrsim b$ and $a \gtrsim_c b$ are defined similarly.

A set $\mathcal{C}(\mathcal{S}, \left\|\cdot\right\|, r)$ is called a *cover* for the set $\mathcal{S}$ at resolution r with respect to the metric

$\|\cdot\|$ if for every $x \in \mathcal{S}$, there exists $x' \in \mathcal{C}(\mathcal{S}, \|\cdot\|, r)$ such that $\|x - x'\| \leq r$. The minimum cardinality of all such covers is called the *covering number* of $\mathcal{S}$ at resolution r and with respect to the norm $\|\cdot\|$, denoted by $\mathcal{N}(\mathcal{S}, \|\cdot\|, r)$.

One way to analyze the recovery algorithms in Compressive Sensing (CS) involves a value that measures how good is the measurement matrix. That value is called *Coherence*, and in general, if it is smaller, better performance in the algorithm. One of the first paper with the concept was [30]. They called a measure of the *mutual coherence* of two bases. That concept come from the fact of choose a suitable base to the original signal through the simple concept in linear algebra, change of basis. The coherence is related with the angle between any two vectors, roughly speaking, it is the cosine of the minimum angle between any two columns of the basis. More precisely, the mutual coherence is the maximum absolute value of the inner product between any two normalized columns.

Definition 9. *Let Φ and Ψ be orthonormal bases for $\mathbb{R}^N$ the* mutual coherence *between Φ and Ψ is defined as*

$$\mu(\Phi, \Psi) = \sup_{\phi_k \in \Phi, \psi_j \in \Psi} |\langle \phi_k, \psi_j \rangle|, \tag{2.10}$$

where ϕ_k and ψ_j represented matrix columns.

If two bases have a very small value they are *mutually incoherent*. It is obvious that value is between 0 and 1. If two bases have element in common, the measure will be 1.

Lemma 1. *For any pair of orthonormal bases, ϕ_1 and ϕ_2 of $\mathbb{R}^N$*

$$\mu(\phi_1, \phi_2) \geq 1/\sqrt{N}.$$

Proof. Because ϕ_1 and ϕ_2 are orthonormal bases, we have that $\phi_1^T \phi_1 = \phi_1 \phi_1^T = I$ and $\phi_2^T \phi_2 = \phi_1 \phi_1^T = I$. That implies that the matrix $\phi_1^T \phi_2$ is an orthonormal matrix, i.e., $(\phi_1^T \phi_2)^T \phi_1^T \phi_2 = \phi_2^T (\phi_1 \phi_1^T) \phi_2 = I$. The sum of squares of entries in an orthonormal matrix

is N; the average squared entry is, therefore, $1/N$; the maximum entry is therefore at least $1/\sqrt{N}$. □

In [11], the two orthobases Φ and Ψ for $\mathbb{R}^N$ are used for sensing and signal representation respectively.

Definition 10. *The coherence between the sensing basis* Φ *and the representation basis* Ψ *is*

$$\mu(\Phi, \Psi) = \sqrt{n} \max_{1 \leq k,j \leq n} |\langle \phi_k, \psi_j \rangle|. \tag{2.11}$$

The coherence measures the largest correlation between any two elements of Φ and Ψ. If Φ and Ψ contain correlated elements, is large, otherwise is small. In this case, $\mu(\Phi, \Psi) \in [1, \sqrt{n}]$.

The main result in [27] use the coherence to bound the number of measurements for BD matrices based in the basis in which the signal have a sparse expansion.

Definition 11. *Let* $U \in \mathbb{C}^{\widetilde{N} \times \widetilde{N}}$ *be a matrix with* l_2*-normalized columns* $\mathbf{a}_1, \ldots, \mathbf{a}_N$*, i.e.,* $\|\mathbf{a}_i\| = 1$ *for all* $i \in [N]$*. The* coherence $\mu = \mu(A)$ *of the matrix* A *is defined as*

$$\mu(U) := \sqrt{\widetilde{N}} \max_{p,q \in [N]} |U(p, q)|, \tag{2.12}$$

where $U(p, q)$is the (p, q)th entry of U. Also, we can see μ as the similarity between U and the canonical basis as (2.11). If $\{u_{\widetilde{n}}\}$ and $\{e_{\widetilde{n}}\}$, $\widetilde{n} \in [\widetilde{N}]$, denote the columns of U and of the canonical basis for $\mathbb{C}^{\widetilde{N}}$, respectively, then

$$\mu(U) := \sqrt{\widetilde{N}} \max_{p,q \in [\widetilde{N}]} |\langle \mathbf{u}_p, \mathbf{e}_q \rangle|. \tag{2.13}$$

2.2 Structured Random Matrices

2.2.1 Fourier Matrices

A *Fourier Matrix* can be defined as $W = \left(\frac{w^{jk}}{\sqrt{N}}\right)$ for $j, k = 0, 1, 2, \ldots, N-1$, and normalized by $1/\sqrt{N}$ to make it a unitary. or equivalently:

$$W = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ 1 & w & w^2 & w^3 & \cdots & w^{N-1} \\ 1 & w^2 & w^4 & w^6 & \cdots & w^{2(N-1)} \\ 1 & w^3 & w^6 & w^9 & \cdots & w^{3(N-1)} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & w^{N-1} & w^{2(N-1)} & w^{3(N-1)} & \cdots & w^{(N-1)(N-1)} \end{bmatrix}$$

where $w = e^{-2\pi i/N}$, is a primitive Nth root of unity which $i = \sqrt{-1}$.

2.2.2 Circulant Matrices

A very special and important type of matrices found in CS are those calls *Circular Matrices*, "is one in which the components of a vector in $\mathbb{R}^N$ or $\mathbb{C}^N$ are repeated again and again, but with a shift in position" [31]. Special because its structure and important because they are *diagonalized by a discrete Fourier transform.*

$$\begin{pmatrix} c_0 & c_{n-1} & \cdots & c_2 & c_1 \\ c_1 & c_0 & c_{n-1} & & c_2 \\ \vdots & c_1 & c_0 & \ddots & \vdots \\ c_{n-2} & & \vdots & \vdots & c_{n-1} \\ c_{n-1} & c_{n-2} & \cdots & c_1 & c_0 \end{pmatrix} \tag{2.14}$$

The *Circular Convolution* operation is defined as

Definition 12. *Let be* $\mathbf{x}, \mathbf{z} \in \mathbb{C}^N$ *by*

$$(z \circledast x)_j := \sum_{k=0}^{N-1} z_{j \ominus k} x_k, \text{ for } j = 0, \ldots, N-1,$$

where $j \ominus k = j - k \mod N$ *is the cyclic subtraction.*

2.3 Structured Matrices and its RIP bounds

The simple way that originally CS was presented, complicated mathematical techniques and tools were needed for its analysis. Some years later, the developed of probability bounds analysis and together with the relation of some theories already found and somewhat forgotten in mathematics, make the way to analyze the RIP property much more accessible and practical. Even better, now it is possible to solve some real-world signal acquisition challenges where structure matrices are used, saving time and resources to researches proving that their model satisfies CS requirements.

A good point to start with this survey is a review made in [32] where they show four available alternatives for structured CS matrices. They present and cite references to known performance guarantees, as well as areas of application where those structures arose.

Sub-Sampled Incoherent Bases (SIB)

The first one is for *sub-sampled incoherent bases* where the key concept of the coherence of a frame (the linear algebra frames) is extended to pairs of orthonormal bases. The theorems that guarantee which are:

Theorem 1. *Let $x = \Psi\theta$ be a S-sparse signal in Ψ with support $\Omega \subset [N]$, $|\Omega| = S$, and with entries having signs chosen uniformly at random. Choose a set $\Gamma \subseteq [N]$ uniformly at random for the set of observed measurements, with $M = |\Gamma|$. Suppose that*

$$M \geq C \cdot S \cdot \mu^2(\Phi, \Psi) \cdot \log(N/\delta) \text{ and } M \geq C' \log^2(N/\delta),$$

for fixed values of $\delta < 1, C$, and C'. Then with probability at least $1 - \delta$, θ is the solution of the L1 minimization problem or basis pursuit (BP).

The extended result for *compressible signals* is

Theorem 2. *Choose a subset $\Gamma \subseteq [N]$ for the set of observed measurements, with $M = |\Gamma|$. Suppose that*

$$M \geq C \cdot S \cdot t \cdot \mu(\Phi, \Psi) \cdot \sqrt{N} \cdot \log^2 S \cdot \log(t \cdot S \log N),$$

for a fixed value of C. Then with probability at least $1 - 5e^{-t}$ the matrix $\Phi^T\Psi$ has the RIP with constant $\delta_{2S} \leq 1/2$.

The very important applications are magnetic resonance imaging (MRI), and tomographic imaging and optical microscopy. Another is used in new acquisition hardware that can obtain projections of the signal against a class of vectors. They are a class of single pixel imagers based on optical modulators and the random sampling ADC.

Structurally Subsampled Matrices (SSM)

Over many applications when measurements are obtained by the hardware, they do not correspond to a sensed coefficients in a particular transform. However, what does happen is that observations are linear combinations of multiple coefficients of the signal, this giving rise to a type of structured matrix. The supported theorem is:

Theorem 3. *Let $\overline{\Phi}$ be a structurally sub-sampled matrix of size $M \times N$ obtained from the basis $\mathbf{U}$ and the $p \times N$ mixing matrix $\mathbb{R} = \mathbf{SM}$ via randomized sub-sampling. Then for each integer $S > 2$, any $z > 1$ and $\delta \in (0, 1)$, there exist absolute positive constants c_1, c_2 such that if*

$$M \geq c_1 \cdot z \cdot S \cdot \mu^2(\mathbf{U}, \Psi) \cdot N \cdot \log^2 S \cdot \log^3 N,$$

the the matrix $\overline{\Phi}$ has the (s, δ)-RIP with probability at least

$$1 - 20 \max\{e^{-c_2 \delta^2 z}, N^{-1}\}.$$

The compressive ADCs with the first step as the random demodulator (RD) employs this structurally.

Structurally Circulant Matrices (SCM)

Applications in communications were the first one in CS use circulant or more general Toeplitz matrices. Some are channel estimation and multi-user detection. Compared with generic CS matrices, sub-sampled circulant matrices have a very small degree of freedom because the matrix can be represented by only one vector.

Theorem 4. *Let Φ be a subsampled circulant matrix whose distinct entries are independent random variables following a Rademacher distribution, and $\mathbf{R}$ is an arbitrary $M \times N$ identity*

submatrix. Furthermore, let δ *be the smallest value for which RIP holds for all* $x \in \sum_S$. *Then for* $\delta_0 \in (0,1)$ *we have* $\mathbb{E}\{\delta \leq \delta_0\}$ *provided that*

$$M \geq C \max\{\delta_0^{-1} S^{3/2} \log^{3/2} N, \delta_0^{-2} S \cdot \log^2 S \cdot \log^2 N\}$$

where $C > 0$ *is a universal constant. Furthermore, for* $0 \leq \lambda \leq 1$,

$$\mathbb{P}\left(\delta_S \geq \mathbb{E}\{\delta\} + \gamma\right) \leq e^{\gamma^2/\sigma^2}, \text{ where } \sigma^2 = C' \frac{S}{M} \log^2 S \cdot \log N,$$

for a universal constant $C' > 0$.

Separable Matrices (SM)

For very large signal, the *separable matrices* will give a nice opportunity to apply CS and measurement it adequately. Applications coming from hyper-cube sampling from multidimensional data are using this structure. The single pixel camera extension to hyper-spectral imaging and the transform imager, an imaging hardware architecture that implements a separable CS matrix, are examples that use this CS structure.

The usual way to represent the matrix is using Kronecker products where the similarity between blocks made it easy to obtain bounds.

Lemma 2. *Let* Φ_d, $1 \leq d \leq D$, *be matrices that have the* (S, δ_d)*-RIP,* $1 \leq d \leq D$, *respectively. Then* $\overline{\Phi} = \bigotimes_{i=1}^{D} \Phi_i$, *has the* (S, δ)*-RIP, with*

$$\delta \leq \prod_{d=1}^{D} (1 + \delta_d) - 1.$$

Start to think about structured matrices was a natural process to follow after original CS developed. An important step in signal processing is data acquisition, choosing an efficient

and universal one will help to save time and computer capabilities. Romberg in [7], one author of the original paper about CS, presented in 2009 a result using an original structure matrix where convolving the signal of interest with a random pulse and then randomly subsampling, give him the opportunity to create a new framework for CS. He describes the two scenarios where researches can works, measurement by *random waveforms* and *random sampling from an incoherent orthobasis*. His proposal used the last one.

When random sampling from an incoherent orthobasis is used, two basis are involved, one for sampling and another for the representation on the signal. Therefore, it is necessary to quantify this relationship since the number of measurements will depend on that. As it is mentioned in its name process, the *mutual coherence* defined in 9 will be used and appears in the number of measurements as an important factor

$$M \gtrsim \mu^2 \cdot S \cdot \log N. \tag{2.15}$$

Note that to reach the standard CS result $M \gtrsim S \cdot \log N$, the mutual coherence should be close to 1, i.e., the basis will be *mutual incoherence*.

The steps of the random convolution process are summarized as

$$H = \sqrt{N} F^* \Sigma F, \tag{2.16}$$

where F is a discrete Fourier matrix and Σ has a special definition of a diagonal matrix, see [33] for details. The construction made H orthogonal and it can interpret convolution with a pulse as a transformation into a *random orthobasis*.

$$H^*H = \left(\sqrt{N} F^* \Sigma F\right) \left(\sqrt{N} F^* \Sigma F\right)^* = N^{-1} F^* \Sigma^* F F^* \Sigma F = N \cdot I,$$

which is getting because $FF^* = F^*F = N \cdot I$ and $\Sigma^*\Sigma = I$.

Next, two methods are included to compress, named, *sampling at random locations* and *Randomly Pre-Modulated Summation* (RPMS) which give him the following two results. Let us call the whole process as *Random Convolution and Compress* (RCC).

Theorem 5. *Let Ψ be an arbitrary orthonormal signal representation. Fix a support set Γ of size $|\Gamma| = S$ in the Ψ domain, and choose a sign sequence z on Γ uniformly at random. Let α_0 be a set of Ψ domain coefficients supported on Γ with signs z, and take $x_0 = \Psi\alpha_0$ as the signal to be acquired. Create a random convolution matrix H as described above, and choose a set of sample locations Ω of size $|\Omega| = M$ uniformly at random with*

$$M \geq C_0 \cdot S \cdot \log(N/\delta) \tag{2.17}$$

and $M \geq C_1 \log^3(N/\delta)$, where C_0 and C_1 are known constants. Set $\Phi = R_\Omega H$, where R_Ω is the restriction operator over Ω. Then, given the set of samples on Ω of the convolution Hx_0, $y = \Phi x_0$, the convex l_1-minimization will recover α_0 (and hence x_0) exactly with probability exceeding $1 - \delta$.

Theorem 6. *Let Ψ, Γ, α_0, x_0, and H be as in Theorem 5. Create a random pre-moduled summation matrix $P\Omega$ as described in Section 1.2.1 in [33] that outputs a number of samples M with*

$$M \geq C_0 \cdot S \cdot \log^2(N/\delta) \tag{2.18}$$

and $M \geq C_1 \log^4(N/\delta)$, where C_0 and C_1 are known constants. Set $\Phi = P\Theta H$. Then, given the measurements $y = \Phi x_0$, the convex l_1-minimization will recover α_0 (and hence x_0) exactly with probability exceeding $1 - \delta$.

What can be seen from this article is a *very important theoretical strategy to sensing* since this random convolution construction is *universal* and allows *fast computations*. On this theoretical basis in the same article, they propose two important topics in which it can be applied and show their description of their use, those are *Radar Imaging* and *Fourier Optics*.

The next one is where I start knowing about compressed sensing, and more specifically, when it involves structure matrices. Two related types of structured measurement matrices were described there having a diagonal block structure. One is when the blocks are different called *Distinct Block Diagonal* (DBD), and the other when blocks are equal called *Repeated Block Diagonal* (RBD). In both cases, each block on the diagonal is itself a sub-Gaussian random matrix satisfying RIP. The number of measurements in each case depends on the signal basis. A very important issue about these types of matrices is that, for specific basis cases, perform nearly as well as dense Gaussian random matrices without being affected by the amount of non-zero elements.

For the structure of the matrix, let us use J, N and M positive integers and set $\widetilde{M} := JM$ and $\widetilde{N} := JN$. The model vector $\mathbf{x} \in \mathbb{C}^{\widetilde{N}}$ is partitioned into J blocks of length N, i.e, $\mathbf{x} = [x_1^T, \ldots, x_J^T]^T$ where $x_j \in \mathbb{C}^N$, $j \in [J]$. For each j, they suppose that a linear operator $\Phi_j : \mathbb{C}^N \to \mathbb{C}^M$ collects the measurements $y_j = \Phi_j x_j$. Concatenating all of the measurements into a vector $\mathbf{y} \in \mathbb{C}^{\widetilde{M}}$, the overall measurement operator relating $\mathbf{y}$ to $\mathbf{x}$, i.e. $\mathbf{y} = \Psi\mathbf{x}$, will have a block diagonal structure as it is shown in the following system:

$$\underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_J \end{bmatrix}}_{\mathbf{y}:\, \widetilde{M}\times 1} = \underbrace{\begin{bmatrix} \Phi_1 & & & \\ & \Phi_2 & & \\ & & \ddots & \\ & & & \Phi_J \end{bmatrix}}_{\Psi:\, \widetilde{M}\times\widetilde{N}} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_J \end{bmatrix}}_{\mathbf{x}:\, \widetilde{N}\times 1}. \tag{2.19}$$

Their main result shows that the number of measurements depend on the base in which the vector is sparse. They showed that those matrices perform nearly as well as dense Gaussian random matrices.

Formally, let $\Psi \in \mathbb{C}^{\widetilde{M}\times\widetilde{N}}$ denote a matrix as defined in (2.19). And let each block Φ_j for $j \in [J]$ be a different unstructured random matrix populated with i.i.d. sub-Gaussian

random variables having mean zero, standard derivation $1/\sqrt{M}$, and sub-Gaussian norm $\tau/\sqrt{M}$, for some $\tau > 0$.

Theorem 7. *Let U denote an orthobasis for $\mathbb{C}^{\widetilde{N}}$ and set*

$$\widetilde{\mu}(U) := \min\left(\sqrt{J}, \mu(U)\right), \text{ where } \mu(U) = \sqrt{\widetilde{N}} \max_{p,q\in[\widetilde{N}]} |U(p,q)|.$$

If $S \gtrsim 1$ and

$$\widetilde{M} \gtrsim_\tau \delta^{-2} \cdot \widetilde{\mu}^2(U) \cdot S \cdot \log^2 S \cdot \log^2 \widetilde{N},$$

then $\delta_S(\Psi, U) \leq \delta < 1$, except with a probability of at most $O(\widetilde{N}^{-\log \widetilde{N} \log^2 S})$.

The applications for this type of structure matrices are principally to represent *acquisition systems* with architectural constraints that prevent global data aggregation. An important case, which was the one that attracts our attention, is *distributed sensing systems* where communication and environmental constraints limit the dependence of each sensor to only a subset of the data. Another one could be, in *streaming applications* where signals have data rates that necessitate operating on local signal blocks rather than on the entire signal simultaneously.

Another interesting structure quite similar to previous block diagonal is one proposed by Castorena in [28] called *Banded Random Matrix* (BRM). That type of matrix structure appears in the literature by Wigner mentioned in 1955. The formal definition used in the paper is taken from one of the most used and classic text book in computer science in matrix computations [34]. The structure is depicted in Figure 2.1.

The matrices are defined as $\widetilde{\Phi} \in \mathbb{C}^{M\times N}$ with entries obtained from a random probability distribution with a centered band and zeros elsewhere. There are two main parameters that define the band, $b_v \in [M]$ and $b_h \in [N]$, the vertical and horizontal bands respectively. Furthermore, the parameter $b_s \in [b_h]$ is important, which determines the amount of shifting

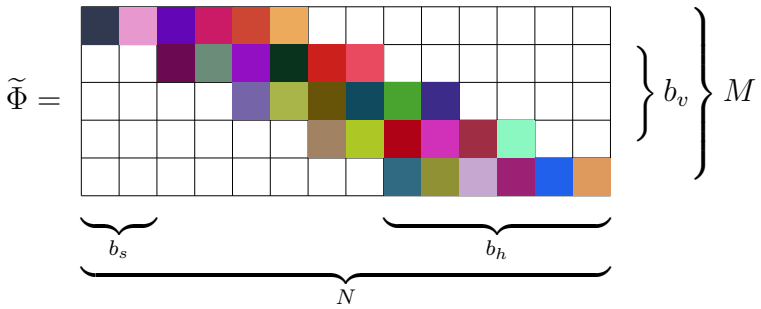


Figure 2.1: A banded random matrix structure of dimension $M \times N$.

present between subsequent rows. From this parameter, it is understood that $b_v = b_h/b_s \in \mathbb{Z}$. The main RIP result is presented in Theorem 8.

Theorem 8. *A Banded Random Matrix $\widetilde{\Phi} \in \mathbb{C}^{M\times N}$ with band parameters $b_h \in [N]$, $b_s \in [b_h]$ and $b_v = b_h/b_s$, satisfies the RIP with constant $\delta > \delta_S$ and with probability at least $1 - 4\exp(-c\delta^2 b_v/\beta)$ for $c > 0$ if the number of measurements satisfies*

$$M \geq \lceil C \cdot S \cdot \beta(\log(N/S) + 1) + N/b_s \rceil - 1. \tag{2.20}$$

Here, the constant $C > 0$ is small and depends upon the parameter $\delta \in (0, 1)$, on the sub-Gaussian norm of its entries, and on c. Furthermore β is given by

$$\beta = \begin{cases} 1 & \text{for } S = 1, \text{ or } S \leq b_s, b_h \\ 1 - \frac{(S-b_s)(S+b_s)}{3b_h(S-1)} & \text{for } b_s \leq S \leq b_h \\ \frac{b_s^2+3S(b_h-1)-b_h^2}{3S(S-1)} & \text{for } S > b_h. \end{cases}$$

The application of this matrix is concerned with CS sampling architectures models, such as random demodulator, the parallel non-interleaved, the random sampler, periodic non-uniform sampling, coded aperture camera and coded exposure camera. Some more information about all this process and their references, can be found over the paper.

A framework to construct fast and efficient sensing matrices for practical applications in CS were developed in [35]. The matrix in the framework, called *Structurally Random Matrix* (SRM), is going to be the product of three matrices

$$\Phi = \sqrt{\frac{N}{M}} DFR. \tag{2.21}$$

Each matrix in this composition represents a step in the process to apply CS and help improve the performance on recovering algorithms. From right to left, R has the responsibility to do a pre-randomization, either flipping its sample sign on its signal's elements or uniformly permuting its sample locations.

The flipping process is doing with a *diagonal matrix of Bernoulli* random variables where flip the sign's signal *locally*. The uniformly random permutation step will use a *random permutation matrix* that scrambles the signal's sample locations globally.

The matrix F is going to take the role of doing a fast transformation, it is well known the fast calculations using a Fourier matrix, but it can be visualized like any other that does it fast. The last one D does the sub-sampling, it will pick M measurements randomly from the already transformed coefficients. The theoretical result presented with this framework is backed by the following theorem.

Theorem 9. *With probability at least of* $1 - \delta$*, the proposed framework employing the local randomizer can reconstruct* S*-sparse signals exactly if the number of measurements is*

$$M \gtrsim \frac{N}{B} \cdot S \cdot \log\left(\frac{N}{\delta}\right) \quad \text{and} \quad M \gtrsim \frac{N}{B} \log^3\left(\frac{N}{\delta}\right), \tag{2.22}$$

where $1 \leq B \leq N$ *come from* F *as the maximal absolute magnitude of the entries in the order of* $\mathcal{O}\left(\frac{1}{\sqrt{B}}\right)$*, i.e,*

$$\max_{1 \leq i,j, \leq N} |F_{i,j}| = \frac{c}{\sqrt{B}} \tag{2.23}$$

for some positive constant c.

If F is a dense and uniform matrix (e.g., DCT or normalized WHT), the sufficient conditions becomes

$$M \gtrsim S \cdot \log\left(\frac{N}{\delta}\right) \quad \textit{and} \quad M \gtrsim \log^3\left(\frac{N}{\delta}\right). \tag{2.24}$$

The proposed framework will perform as other frameworks where optimal performance is guaranteed.

The way that we are proving COB and its generalization was inspired by the work of Eftekhari et al. in [27], but actually, the main result there to prove RIP for BD was using a theorem established by Krahmer et al. in [36]. The theorem was used to improve measurement bounds obtaining RIP for two structure matrices named *partial random circulant matrices* (PRC) and *time-frequency structured random* (TFS) matrices. The recovery property probability is supported by Theorems 10 and 11 respectively.

To describe PRC, let $H = H_z \in \mathbb{C}^{N\times N}$ be a circulant matrix associated with z given by $Hz = z \circledast x$ and with entries $H_{jk} = z_{j\ominus k}$, where $\circledast$ and $\ominus$ are operations defined in 12. And let be $\Omega \subset [N]$ with $|\Omega| = M$ and $R_\Omega : \mathbb{C}^N \to \mathbb{C}^M$ the operator that restricts a vector $x \in \mathbb{C}^N$ to its entries in Ω. The associated *partial random circulant matrix* is given by $\Phi = M^{-1/2} R_\Omega H_\epsilon$ and acts on vectors $x \in \mathbb{C}^N$ via

$$\Phi x = \frac{1}{\sqrt{M}} R_\Omega(\epsilon \circledast x), \tag{2.25}$$

where ϵ is a Radamacher vector. The matrix Φ can be see as a circulant matrix generated by a Rademacher vector, where the rows outside Ω are removed.

And now to describe (TFS), let us define the translation and modulation operator on $\mathbb{C}^M$ as $(Th)_j = h_{j\ominus 1}$ and $(Mh)_j = e^{2\pi ij/M} h_j = w^j h_j$, where $w = e^{2\pi/M}$ and $\ominus$ denotes cyclic

subtraction module M. The time-frequency shift are given by

$$\pi(\lambda) = M^l T^k, \quad \lambda = (k, l) \in \mathbb{Z}_M^2 = \{0, \dots, m-1\}^2. \tag{2.26}$$

For $h \in \mathbb{C}^M \setminus \{0\}$ the system $\{\pi(\lambda)h : \lambda \in \mathbb{Z}_M^2\}$ is called *Gabor system* and the $M \times M^2$ matrix Ψ_h whose columns are the vectors $\pi(\lambda)h$ for $\lambda \in \mathbb{Z}_M^2$ is called a *Gabor synthesis matrix*,

$$\Psi_h = [\pi(\lambda)h]_{\lambda \in \mathbb{Z}_M^2} \in \mathbb{C}^{M \times M^2}. \tag{2.27}$$

Theorem 10. *Let $\Phi \in \mathbb{R}^{M \times N}$ be a draw of a partial random circulant matrix generated by a Rademacher vector ϵ. If*

$$M \gtrsim \delta^{-2} \cdot S \cdot \log^2 S \cdot \log^2 N, \tag{2.28}$$

then with probability at least $1 - N^{-\log N \cdot \log^2 S}$, the RIC of Φ satisfies $\delta_S \leq \delta$.

Theorem 11. *Let ϵ be a Rademacher vector and consider the Gabor synthesis matrix $\Psi_h \in \mathbb{C}^{M \times M^2}$ generated by $h = \frac{1}{\sqrt{M}}\epsilon$. If*

$$M \gtrsim \delta^{-2} \cdot S \cdot \log^2 S \cdot \log^2 M, \tag{2.29}$$

then with probability at least $1 - M^{-\log M \cdot \log^2 S}$, then RIC of Ψ_h satisfies $\delta_s \leq \delta$.

The applications for PRC are system identification, radar and cameras with coded aperture. Meanwhile, applications for TFS or Gabor synthesis matrices include operator identification (channel estimation in wireless communications), radar and sonar. See the paper for references over those applications.

The last paper for now, we hope to work with more structures later, was selected because it was published recently, April 2018, and motivated us to continue our research. In addition, its model signals with corrupted measurements not discussed until now.

Two structured matrices are presented, one called *Randomly Modulated Unit-Norm Tight Frames* (MUTF) and the second one was already discussed using *Randomly Subsampled Orthonormal System*. We show here just the first one and referred the reader to the paper for the other one.

The matrix in MUTF can be written as $A = UD\widetilde{B}$, where $U \in \mathbb{C}^{M\times\widetilde{N}}$ is a UTF with $\mu(U) \sim 1/\sqrt{M}$, $D = \text{diag}(\xi)$ is a diagonal matrix with ξ being a length-$\widetilde{N}$ random vector with independent, zero mean, unit-variance, and L-subgaussian entries, and $\widetilde{B} \in \mathbb{C}^{\widetilde{N}\times N}$, $\widetilde{N} \geq N$, represent a column-wise or orhonormal matrix. i.e., $\widetilde{B}^*\widetilde{B} = I$.

The result that guarantees recovery signals with high probabilities is Theorem 12.

Theorem 12. *Suppose* $y = Ax^* + z^* + w$ *with* $\Theta = [A,\, I] \in \mathbb{C}^{M\times(N+M)}$, $A = UD\widetilde{B}$ *and* $\mu(U) \sim 1/\sqrt{M}$. *If, for* $\delta \in (0, 1)$,

$$M \gtrsim \delta^{-2} \cdot S \cdot \widetilde{N} \cdot \mu^2(\widetilde{B}) \cdot \log^2 S \cdot \log^2 \widetilde{N}, \tag{2.30}$$

$$M \gtrsim \delta^{-2} \cdot K \cdot \log^2 K \cdot \log^2 \widetilde{N}, \tag{2.31}$$

where K *is the sparsity of* z, *then with probability at least* $1 - 2\widetilde{N}^{-\log^2 S\cdot\log\widetilde{N}}$, *the* (S, K)*-RIP constant of* Θ *satisfies* $\delta_{s,k} \leq \delta$.

Applications of MUTF arise where the compressed measurement may be corrupted by impulse noise like sensor networks and error correction in joint source channel coding.

2.4 Summary

Table 2.1 shows part of the research summary in structured CS matrices. The table will give the opportunity to see their measurement bounds M and the reference. In addition, it will help to see improvements or similar results over related matrices. On the other hand, it

shows the high and updated research activity about structures matrices.

Some papers give name to its matrix, while others do not. We have used one that seems us to be better related to the matrix and we have awarded it in order to follow the sequence.

Name	**Matrix**	$M \gtrsim$ **bound**	**Reference**
SIM	$\overline{\Phi}^T$ $\Phi^T\Psi$	$S\mu^2(\Phi,\Psi)\log(N/\delta)$ & $\log^2(N/\delta)$ $CSt\mu(\Phi,\Psi)\sqrt{N}\log^2 S\log(tS\log N)$	Duarte et al. 2011 [32]
SSM	$\Phi = \mathbf{RU}, \overline{\Phi}^T$	$zS\mu^2(\mathbf{U},\Psi)N\log^2 S\log^3 N$	
SCM	$\Phi = \mathbf{RU}$	$\max\{\delta_0^{-1}S^{3/2}\log^{3/2} N,$ $\delta_0^{-2}S\log^2 S\log^2 N\}$	
SM	$\overline{\Phi} = \bigotimes_{i=1}^{D}\Phi_i$	$\delta \leq \prod_{d=1}^{D}(1+\delta_d)-1$	
RCC	$H = \sqrt{N}F^*\Sigma F$ $\Phi = R_\Omega H$	$S\log(N/\delta)$ & $C_1\log^3(N/\delta)$	Romberg 2009 [33]
RPMS	$\Phi = P\Theta H$	$S\log^2(N/\delta)$ & $C_1\log^4(N/\delta)$	
DBD	$\Psi = I_J \otimes \Phi_j$	$\delta^{-2}\widetilde{\mu}^2(U)S\log^2 S\log^2\widetilde{N}$	Eftekhari et al. 2015 [27]
RBD	$\Psi = I_J \otimes \Phi$	$\delta^{-2}\gamma^2(U)S\log^2 S\log^2\widetilde{N}$	
BRM	$\widetilde{\Phi}$ $\widetilde{\Phi}F$	$\lceil S\beta(\log(N/S)+1)+N/b_s\rceil - 1$ $C_bS[\log(N/S)+1]$	Castorena & Creusere 2014 [28]
SRM	$\Phi = \sqrt{\frac{N}{M}}DFR$ $F = \mathrm{DTC}$	$\frac{N}{B}S\log\left(\frac{N}{\delta}\right)$ &$\frac{N}{B}\log^3\left(\frac{N}{\delta}\right)$ $S\log\left(\frac{N}{\delta}\right)$ &$\log^3\left(\frac{N}{\delta}\right)$	Thong et al. 2012. [35]
PRC TFS	$\Phi = M^{-1/2}R_\Omega H_\epsilon$ $\Psi_h = [\pi(\lambda)h]_{\lambda\in\mathbb{Z}_M^2}$	$\delta^{-2}S\log^2 S\log^2 N$ $\delta^{-2}S\log^2 S\log^2 M$	Krahmer et al. 2014. [36]
MUTF	$A = UD\widetilde{B}$ $\Theta = [A,\ I]$	$\delta^{-2}S\widetilde{N}\mu^2(\widetilde{B})\log^2 S\log^2\widetilde{N}$ & $\delta^{-2}K\log^2 K\log^2\widetilde{N}$	Zhang et al. 2017. [37]

Table 2.1: The summary of structured matrices, M bounds and references.

2.5 Conclusions

We can see in this short survey that most approaches are to apply CS in applications where it is impractical to include the full measurement matrix because the size of the signals is very large and recovery calculation will take a long time. The objective is to take advantage of the structure of the application to define the matrix. Although there exist already several contributions on this direction and when more areas of applications get involve of CS, more specific structures will require, therefore, a complete classification of this structures will be needed. We hope in future works do a more complete one.

Under special circumstances structure matrices will performs similar as unstructured matrices, but usually it takes more measurement in order to reach RIP.

A reference discuss the most used structure matrices, Fourier and Toelipz, is presented by Rauhut in notes [29]. Those notes later becomes a chapter in book [15]. The idea of do a complete survey will help the researches shows ways to how prove RIP for structures matrices and also check if their structure have been already prove.

Chapter 3

The COB Matrix and Its RIP

In this chapter, we will first introduce a new type of measurement matrix, namely, the C*yclic* O*verlapping* B*lock diagonal* (COB) matrix. Then, we will prove that the COB matrix satisfies the RIP property. And finally, we will demonstrate the performance of the COB matrix and compare it with the BD matrix.

3.1 The COB Matrix

For the definition of the proposed measurement matrix Ψ, let $\Phi_{j1}, \Phi_{j2} \in \mathbb{C}^{M\times N}$ for $j \in [J]$ be matrices populated with i.i.d. sub-Gaussian random variables having mean zero, standard deviation $1/\sqrt{2M}$ and sub-Gaussian norm τ. In this way, blocks in Ψ are $M \times 2N$ matrices $\Phi_j = [\Phi_{j1}\ \Phi_{j2}]$, i.e., blocks are formed with the concatenation of Φ_{j1} and Φ_{j2}, its structure will be as equation (3.1)

$$\Psi = \begin{bmatrix} [\Phi_{11} & \Phi_{12}] & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & [\Phi_{21} & \Phi_{22}] & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & [\Phi_{31} & \Phi_{32}] & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & [\Phi_{(J-1)1} & \Phi_{(J-1)2}] \\ \Phi_{J2}] & \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & [\Phi_{J1} \end{bmatrix} \tag{3.1}$$

The symbol $\mathbf{0}$ represents a block of zeros, a very important advantage over full nonzero matrices called before *unstructured matrices*. *Structures matrices* becomes in CS a new branch of research because less computation over this type of matrices gives good performance to the recover algorithms. In addition, the architectural sample devices does not fit for unstructured matrices.

3.2 The RIP of the COB Matrix

In this section, we are going to prove that the proposed COB matrix (3.1) satisfies the U-RIP property (8). The general idea is use a result from [27] which uses a result from [36] that also took the main idea from [38]. The prove is based in *Random Process* or *Stochastic Process* where the study of the supremum or more precisely find upper and lower bounds for these suprema constitutes the main task of that area. This field is a useful and highly studied math area today that has applications in many disciplines including sciences such as biology, chemistry, ecology, neuroscience, and physics as well as technology and engineering fields such as image processing, signal processing, information theory, computer science, cryptography and telecommunications.[1]

[1] Areas taken from Wikipedia definition of Stochastic Process

3.2.1 The Main Result

Shows that a matrix satisfies RIP is synonymous with guarantee that a signal can be recovered using well-known algorithms. Our main result written above has a classical Compressed Sensing form, i.e., assuming that the signal has some sparsity $S \gtrsim 1$ and assuming too some lower bound for the number of measurements $\widetilde{M}$, then shows that RIC δ_S is small enough with *high probability*. The S is the sparsity of the signal that can be represented by a proper orthobasis U, i.e., for a signal $x = U\alpha$ in $\mathbb{C}^{\widetilde{N}}$, where $\alpha \in \Omega_S$ is an element in the set of all S-sparse signals with unit norm. The *main result* in this chapter is presented in the following theorem.

Theorem 13. *Let U denote an orthobasis for $\mathbb{C}^{\widetilde{N}}$, $\Psi \in \mathbb{C}^{\widetilde{M}\times\widetilde{N}}$ a J blocks COB matrix as defined above* (3.1) *and set*

$$\widetilde{\mu^*} := \min\left\{\sqrt{\frac{J}{2S}}, \mu(U)\right\}, \qquad \textit{where} \qquad \mu(U) = \sqrt{\widetilde{N}} \max_{p,q\in[\widetilde{N}]} |U(p,q)|.$$

If $S \gtrsim 1$ and

$$\frac{\widetilde{M}}{\log \widetilde{M}} \gtrsim \delta^{-2} \cdot \widetilde{\mu^*}^2 \cdot S \cdot \log^2 S \log \widetilde{N},$$

for some constant c, then $\delta_S(\Psi, U) \leq \delta < 1$, except with a very low probability of at most $O\left(\widetilde{N}^{-\log \widetilde{M} \log^2 S}\right)$.

Proof. Go to 3.2.7 in this section. □

A special case use later in the experiments is when the signal is represented over the *canonical base*, in that case we got the following property.

Corollary 1. *Let $x \in \mathbb{C}^{\widetilde{N}}$ be a signal over the canonical base, and $\Psi \in \mathbb{C}^{\widetilde{M}\times\widetilde{N}}$ a J blocks*

COB matrix as defined above (3.1). *If* $S \gtrsim 1$ *and*

$$\frac{\widetilde{M}}{\log \widetilde{M}} \gtrsim \delta^{-2} \cdot J \cdot \log^2 S \log \widetilde{N},$$

then $\delta_S \leq \delta < 1$, *except with a very low probability of at most* $O\Big(\widetilde{N}^{-\log \widetilde{N} \log^2 S}\Big)$.

Our results show that the number of rows (measurements) needed depends linearly of the number of blocks J and poly-logarithm with the sparsity S and the size of the signal $\widetilde{N}$.

The next table 3.1 shows the different bounds for the number of measurements for structured matrices BD, COB and compare the bound with unstructured sub-Gaussian matrices, a well known result in the CS area.

Matrix	**Bounds**	**Reference**
Unstructured	$\widetilde{M} \gtrsim \delta^{-2} \cdot S \cdot \log(\widetilde{N})$	Candès et al. 2006. [7]
BD	$\widetilde{M} \gtrsim \delta^{-2} \cdot \widetilde{\mu}^2(U) \cdot S \cdot \log^2 S \cdot \log^2 \widetilde{N}$	Eftekhari et al. 2015. [27]
COB	$\frac{\widetilde{M}}{\log \widetilde{M}} \gtrsim \delta^{-2} \cdot J \cdot \log^2 S \cdot \log \widetilde{N}$	our result, 2018

Table 3.1: The measurement bounds for unstructured matrices, and BD and COB structures matrices.

3.2.2 The Outline of the Proof

Using the U-RIP definition 8, we consider that a matrix A satisfies the restricted isometry property if $0 < \delta_S \ll 1$ *for reasonably large* S. How big S can be, we will see it later.

Our result follows the idea of Eftekhari's paper [27] which has as its main ingredient the following theorem from [36]:

Theorem 14 (Krahmer et al. [36])**.** *Let* $\mathcal{A} \subset \mathbb{C}^{\overline{M} \times \overline{N}}$ *be a set of matrices, and let* ε *be*

a random vector whose entries are i.i.d., zero-mean, unit-variance random variables with sub-Gaussian norm τ. Set

$$d_F(\mathcal{A}) := \sup_{A\in\mathcal{A}} \|A\|_F,$$
$$d_2(\mathcal{A}) := \sup_{A\in\mathcal{A}} \|A\|_2,$$

and

$$E := \gamma_2(\mathcal{A}, \|\cdot\|_2)\big(\gamma_2(\mathcal{A}, \|\cdot\|_2) + d_F(\mathcal{A})\big) + d_F(\mathcal{A})d_2(\mathcal{A}),$$
$$V := d_2(\mathcal{A})\big(\gamma_2(\mathcal{A}, \|\cdot\|_2) + d_F(\mathcal{A})\big),$$
$$U := d_2^2(\mathcal{A}).$$

Then, for $t > 0$, it holds that

$$\log P\left\{\sup_{A\in\mathcal{A}} \left|\|A\varepsilon\|_2^2 - \mathbb{E}\|A\varepsilon\|_2^2\right| \gtrsim_\tau E + t\right\} \lesssim_\tau -\min\left(\frac{t^2}{V^2}, \frac{t}{U}\right). \tag{3.2}$$

At first glance, it looks like there is no relation prove RIP for our COB matrix with this theorem, but if we explain some details and doing mainly a transformation, it will be clear.

The steps that we need to prove RIP for our COB matrix Ψ (3.1) are:

1. **RIC**: write RIC as $\delta_S = \sup_{\alpha\in\Omega_S} \left|\|\Psi \cdot x(\alpha)\|_2^2 - 1\right|$ and check that $\mathbb{E}\{\|\Psi \cdot x(\alpha)\|_2^2\} = 1$ verifying that $\mathbb{E}\{\Psi^* \cdot \Psi\} = I$.

2. **Transformation**: build $X_j(\alpha)$, $A(\alpha)$, and $\mathcal{A}$. Check that $\|\Psi \cdot x(\alpha)\|_2 =_{\text{i.d.}} \|A(\alpha) \cdot \varepsilon\|_2$.

3. **Calculation 1**: values $d_F(\mathcal{A})$ and $d_2(\mathcal{A})$.

4. **Calculation 2**: value $\gamma_2(\mathcal{A}, \|\cdot\|_2)$

5. **Applying Theorem**: put everything together to get $\widetilde{M}$, i.e., the number of measurements/rows for Ψ and shows that $\delta_S < \delta$, for any $\delta < 1$, i.e., get RIP for Ψ.

In the rest of this section, we will follow those steps. First we present a formal definition of the RIP in Section 3.2.3, in section 3.2.4 we will transform the matrix to get the random vector ε that Theorems 14 uses. In sections 3.2.5 and 3.2.6 we calculate the values E, V and U and finally in section 3.2.7 in order to apply Theorem 14, we will put everything together and prove our main result Theorem 13.

3.2.3 The RIP of Measurement Matrix

Defining the set of all S-sparse signals with unit norm as

$$\Omega_S := \left\{ \alpha \in \mathbb{C}^{\tilde{N}} : \|\alpha\|_0 \leq S, \|\alpha\|_2 = 1 \right\},$$

from the U-RIP property

$$(1 - \delta_S)\|x\|_2^2 \leq \|\Psi x\|_2^2 \leq (1 + \delta_S)\|x\|_2^2, \text{ for all } x \text{ with } \|U^* x\|_0 \leq S,$$

and using the fact that the signal can be represented in another base, i.e., $x = U\alpha$, we get

$$\|x\|_2^2 = \|U\alpha\|_2^2 = (U\alpha)^* U\alpha = \alpha^*(U^*U)\alpha = 1.$$

Therefore, the RIC can be written as

$$(1 - \delta_S) \leq \|\Psi x\|_2^2 \leq (1 + \delta_S)$$

$$-\delta_S \leq \|\Psi x\|_2^2 - 1 \leq \delta_S$$

$$\delta_S \leq \left| \|\Psi x\|_2^2 - 1 \right|,$$

i.e.,

$$\delta_S = \sup_{\alpha \in \Omega_S} \left| \|\Psi \cdot x(\alpha)\|_2^2 - 1 \right|. \tag{3.3}$$

The following expectation is also necessary in order to apply the *Chaos Theorem* 14

$$\begin{aligned} \mathbb{E}\{\|\Psi \cdot x(\alpha)\|_2^2\} &= \mathbb{E}\left\{ \left(\Psi U \alpha\right)^* \Psi U \alpha \right\} \\ &= \alpha^* U^* \mathbb{E}\{\Psi^* \Psi\} U \alpha \\ &= 1, \end{aligned}$$

which is reaching because the $\mathbb{E}\{\Psi^*\Psi\} = I$. The prove of this fact is getting in the following lemma.

Lemma 3. *Let $A \in \mathbb{R}^{M \times N}$ be a random matrix populated with i.i.d. sub-Gaussian random variables having mean zero, standard deviation $1/\sqrt{M}$. Then*

$$\mathbb{E}\{A^* A\} = I. \tag{3.4}$$

Proof. Because A^*A is a N-square matrix and we want to prove that its expectation is the identity matrix, we will check two types of elements, in and outside the diagonal.

Each diagonal element is $\|\mathbf{a_i}\|_2^2 = \langle \mathbf{a}_i, \mathbf{a}_i \rangle$, where $\mathbf{a}_i$ for $i \in [N]$ represent columns of A. Then,

$$\begin{aligned}
\mathbb{E}\left\{\langle \mathbf{a}_i, \mathbf{a}_i \rangle\right\} &= \mathbb{E}\left\{\sum_{k=1}^{M} a_{ki}^2\right\} \\
&= \sum_{k=1}^{M} \mathbb{E}\left\{a_{ki}^2\right\} \\
&= \sum_{k=1}^{M} \left(\mathrm{Var}(a_{ki}) + \mathbb{E}\{a_{ki}\}^2\right) \\
&= \sum_{i=1}^{M} \frac{1}{M} = 1.
\end{aligned}$$

For other entries, i.e, where $i \neq j$,

$$\begin{aligned}
\mathbb{E}\{\langle \mathbf{a}_i, \mathbf{a}_j \rangle\} &= \mathbb{E}\left\{\sum_{k=1}^{M} a_{ki} \cdot a_{kj}\right\} \\
&= \sum_{k=1}^{M} \mathbb{E}\{a_{ki} \cdot a_{kj}\} \\
&= \sum_{k=1}^{M} \mathbb{E}\{a_{ki}\} \cdot \mathbb{E}\{a_{kj}\} = 0.
\end{aligned}$$

The last line is zero by hypothesis, i.e., $\mathbb{E}\{a_{ij}\} = 0$ for all entries in A and the random variables are independent. □

For our case, Ψ involved two unstructured $\phi_{M\times N}$ matrices, because we choose standard deviation $1/\sqrt{2M}$ in each Φ (see definition 3.1), using a similar prove to the Lemma 3, the result is also getting for Ψ, i.e, $\mathbb{E}\{\Psi^*\Psi\} = I$.

3.2.4 Transformation

In order to apply the *Chaos Theorem*, we will convert Ψ to a vector ε, and convert $x(\alpha)$ to a matrix $A(\alpha)$. Before giving the formal definitions of the transformation, let us illustrate the conversion in the following example, where Ψ is a 6×6 matrix.

$$\Psi x(\alpha) = \begin{bmatrix} \phi_1 & \phi_2 & \phi_3 & \phi_4 & 0 & 0 \\ \phi_5 & \phi_6 & \phi_7 & \phi_8 & 0 & 0 \\ 0 & 0 & \phi_9 & \phi_{10} & \phi_{11} & \phi_{12} \\ 0 & 0 & \phi_{13} & \phi_{14} & \phi_{15} & \phi_{16} \\ \phi_{19} & \phi_{20} & 0 & 0 & \phi_{17} & \phi_{18} \\ \phi_{23} & \phi_{24} & 0 & 0 & \phi_{21} & \phi_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} \phi_1 x_1 + \phi_2 x_2 + \phi_3 x_3 + \phi_4 x_4 \\ \phi_5 x_1 + \phi_6 x_2 + \phi_7 x_3 + \phi_8 x_4 \\ \phi_9 x_3 + \phi_{10} x_4 + \phi_{11} x_5 + \phi_{12} x_6 \\ \phi_{13} x_3 + \phi_{14} x_4 + \phi_{15} x_5 + \phi_{16} x_6 \\ \phi_{17} x_5 + \phi_{18} x_6 + \phi_{19} x_1 + \phi_{20} x_2 \\ \phi_{21} x_5 + \phi_{22} x_6 + \phi_{23} x_1 + \phi_{24} x_2 \end{bmatrix}.$$

Becomes, $A(\alpha)\varepsilon =$

$$\frac{1}{\sqrt{2M}}\begin{bmatrix} x_1 & x_2 & x_3 & x_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_1 & x_2 & x_3 & x_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_3 & x_4 & x_5 & x_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_3 & x_4 & x_5 & x_6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & & 0 & 0 & 0 & 0 & 0 & 0 & x_5 & x_6 & x_1 & x_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & x_5 & x_6 \end{bmatrix} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \\ \epsilon_7 \\ \epsilon_8 \\ \epsilon_9 \\ \epsilon_{10} \\ \epsilon_{11} \\ \epsilon_{12} \\ \epsilon_{13} \\ \epsilon_{14} \\ \epsilon_{15} \\ \epsilon_{16} \\ \epsilon_{17} \\ \epsilon_{18} \\ \epsilon_{19} \\ \epsilon_{20} \\ \epsilon_{21} \\ \epsilon_{22} \\ \epsilon_{23} \\ \epsilon_{24} \end{bmatrix} \tag{3.5}$$

$$= \frac{1}{\sqrt{2M}} \begin{bmatrix} \epsilon_1 x_1 + \epsilon_2 x_2 + \epsilon_3 x_3 + \epsilon_4 x_4 \\ \epsilon_5 x_1 + \epsilon_6 x_2 + \epsilon_7 x_3 + \epsilon_8 x_4 \\ \epsilon_9 x_3 + \epsilon_{10} x_4 + \epsilon_{11} x_5 + \epsilon_{12} x_6 \\ \epsilon_{13} x_3 + \epsilon_{14} x_4 + \epsilon_{15} x_5 + \epsilon_{16} x_6 \\ \epsilon_{17} x_5 + \epsilon_{18} x_6 + \epsilon_{19} x_1 + \epsilon_{20} x_2 \\ \epsilon_{21} x_5 + \epsilon_{22} x_6 + \epsilon_{23} x_1 + \epsilon_{24} x_2 \end{bmatrix},$$

where $\phi_i = \frac{1}{\sqrt{2M}}\epsilon_i$, for $i \in [24]$.

Note that the rows of non-zero blocks in Ψ formed a long column vector ε and the vector $x(\alpha)$ with a common factor from elements of Ψ formed a big row block diagonal matrix $A(\alpha) = \frac{1}{\sqrt{2M}}X$. Using this transformation is easy to see that $\|\Psi \cdot x(\alpha)\|_2 =_{\text{i.d.}} \|A(\alpha) \cdot \varepsilon\|_2$. The i.d. means that the random variables on both sides have the same distribution.

We did that transformation because inside the probability in equation (3.2) of Theorem 14

the inequality,

$$\underbrace{\sup_{A\in\mathcal{A}}\left|\|A\varepsilon\|_2^2-\mathbb{E}\|A\varepsilon\|_2^2\right|}_{\delta_S}\gtrsim_\tau \underbrace{E+t}_{\delta}, \tag{3.6}$$

will be later $\delta_S \gtrsim_\tau \delta$, where by previous section

$$\delta_S=\sup_{\alpha\in\Omega_S}\left|\|\Psi\cdot x(\alpha)\|_2^2-1\right|,$$

then from Ψ and $x(\alpha)$ we have built the set of matrices $\mathcal{A}$ and vector ε needed in (3.6).

Note a important fact about $A(\alpha)A^*(\alpha)=\frac{1}{2M}\times$

$$\begin{bmatrix}
\|x_1(\alpha)\|_2^2+\|x_2(\alpha)\|_2^2 & 0 & 0 & 0 & 0 & 0\\
0 & \|x_1(\alpha)\|_2^2+\|x_2(\alpha)\|_2^2 & 0 & 0 & 0 & 0\\
0 & 0 & \|x_2(\alpha)\|_2^2+\|x_3(\alpha)\|_2^2 & 0 & 0 & 0\\
0 & 0 & 0 & \|x_2(\alpha)\|_2^2+\|x_3(\alpha)\|_2^2 & 0 & 0\\
0 & 0 & 0 & 0 & \|x_3(\alpha)\|_2^2+\|x_1(\alpha)\|_2^2 & 0\\
0 & 0 & 0 & 0 & 0 & \|x_3(\alpha)\|_2^2+\|x_1(\alpha)\|_2^2
\end{bmatrix},$$

it is 6×6 diagonal matrix with entries related with sub-vectors from $x(\alpha)$. Later we use this fact to calculate a norm which has the following property

$$\|A(\alpha)\|^2=\|A(\alpha)A^*(\alpha)\|=\|A^*(\alpha)A(\alpha)\|. \tag{3.7}$$

Let us explain in more detail the transformation example.

A. $\Psi\to\varepsilon$

The first block

$$\Phi_1=\begin{bmatrix}\Phi_{11} & \Phi_{12}\end{bmatrix}=\begin{bmatrix}\phi_1 & \phi_2 & \phi_3 & \phi_4\\ \phi_5 & \phi_6 & \phi_7 & \phi_8\end{bmatrix}$$

is a matrix populated with i.i.d. Bernoulli random variables having mean zero, standard

deviation $1/\sqrt{2M}$. Using the columns of Φ_1^*, is transformed to a vector as follows

$$\text{vec}(\Phi_1^*) = \begin{bmatrix} \phi_1 \\ \phi_2 \\ \phi_3 \\ \phi_4 \\ \phi_5 \\ \phi_6 \\ \phi_7 \\ \phi_8 \end{bmatrix} = \frac{1}{\sqrt{2M}} \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \\ \epsilon_5 \\ \epsilon_6 \\ \epsilon_7 \\ \epsilon_8 \end{bmatrix} = \frac{1}{\sqrt{2M}} \varepsilon_1$$

where ε_1 is populated with i.i.d. Bernoulli random variables having mean zero, a unit standard deviation. The same for other blocks of Ψ.

B. $x(\alpha) \rightarrow A(\alpha)$

The matrix A is building from sub-blocks of x, i.e.,

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} x_1(\alpha) \\ x_2(\alpha) \\ x_3(\alpha) \end{bmatrix}$$

where $x_1(\alpha) = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, $x_2(\alpha) = \begin{bmatrix} x_3 \\ x_4 \end{bmatrix}$ and $x_3(\alpha) = \begin{bmatrix} x_5 \\ x_6 \end{bmatrix}$. Then, let us use

$$X_1(\alpha) = \begin{bmatrix} x_1(\alpha)^T\ x_2(\alpha)^T & \\ & x_1(\alpha)^T\ x_2(\alpha)^T \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_1 & x_2 & x_3 & x_4 \end{bmatrix},$$

$$X_2(\alpha) = \begin{bmatrix} x_2(\alpha)^T\ x_3(\alpha)^T & \\ & x_2(\alpha)^T\ x_3(\alpha)^T \end{bmatrix} = \begin{bmatrix} x_3 & x_4 & x_5 & x_6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_3 & x_4 & x_5 & x_6 \end{bmatrix},$$

$$X_3(\alpha) = \begin{bmatrix} x_3(\alpha)^T\ x_1(\alpha)^T & \\ & x_3(\alpha)^T\ x_1(\alpha)^T \end{bmatrix} = \begin{bmatrix} x_5 & x_6 & x_1 & x_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & x_5 & x_6 & x_1 & x_2 \end{bmatrix}.$$

Therefore, the linear map $A : \Omega_S \to \mathbb{C}^{6\times 3\cdot 2\cdot 4}$ is defined as

$$A(\alpha) = \frac{1}{\sqrt{2M}} \begin{bmatrix} X_1(\alpha) & & \\ & X_2(\alpha) & \\ & & X_3(\alpha) \end{bmatrix}_{6\times 24}$$

as shown explicitly in the above example in equation (3.5).

C. The index of the random process

The transformation is going to take vectors of size $\widetilde{N}$ from S-sparse unitary space Ω_S and create a set of matrices $\mathcal{A}$. In stochastic process the set is called *the index set of the random process.*

For $\alpha \in \mathbb{C}^{\widetilde{N}}$, set $x(\alpha) = U\alpha$, where U denote an orthobasis and define $x_j(\alpha) = x_j(\alpha, U) \in \mathbb{C}^N, j \in [J]$ such that

$$x(\alpha) = [x_1(\alpha)^T, x_2(\alpha)^T, \ldots, x_J(\alpha)^T]^T. \tag{3.8}$$

And if $X_j \in \mathbb{C}^{M\times 2MN}, i \in [J]$ is

$$X_j(\alpha) = \begin{bmatrix} x_j^*(\alpha)\ x_{j'}^*(\alpha) & & & \\ & x_j^*(\alpha)\ x_{j'}^*(\alpha) & & \\ & & \ddots & \\ & & & x_j^*(\alpha)\ x_{j'}^*(\alpha) \end{bmatrix} \tag{3.9}$$

where

$$j' = \begin{cases} j+1 & \text{if } i < J \\ 1 & \text{if } j = J \end{cases}. \tag{3.10}$$

Therefore,

$$A(\alpha) = \frac{1}{\sqrt{2M}} \begin{bmatrix} X_1(\alpha) & & & \\ & X_2(\alpha) & & \\ & & \ddots & \\ & & & X_J(\alpha) \end{bmatrix}_{\widetilde{M}\times 2JMN} \tag{3.11}$$

The index set of the random process will be

$$\mathcal{A} := \{A(\alpha) : \alpha \in \Omega_S\}. \tag{3.12}$$

3.2.5 Calculation 1

Two of the three values that we have to calculate in order to apply the *Chaos Theorem* use norms over the index of the random process $\mathcal{A}$, i.e.,

$$d_F(\mathcal{A}) = \sup_{A\in\mathcal{A}} \|A\|_F,$$

$$d_2(\mathcal{A}) = \sup_{A\in\mathcal{A}} \|A\|_2.$$

where $A : \Omega_S \to \mathbb{C}^{\widetilde{M}\times M\widetilde{N}}$ is the linear map defined above and A an element of $\mathcal{A}$.

The first one used *Frobenius norm*, defined as the square root of the sum of the absolute squares of its elements,

$$\|A\|_F = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}|a_{i,j}|^2},$$

and the other is called the *spectral norm* and can be defined as the square root of the maximum eigenvalue of A^*A or the *natural norm induced* from 2-norm vector, i.e.,

$$\begin{aligned}\|A\|_2 &= (\text{maximum eigenvalue of } A^*A)^{1/2} \qquad \text{or}\\ &= \max_{|x|_2\neq 0}\frac{|Ax|_2}{|x|_2}.\end{aligned}$$

Intuitively, we can think of $\|A\|_2$ as the maximum "scale", by which the matrix A can "stretch" a vector.

Calculating $d_F(\mathcal{A})$

To get the Frobenius norm, check that A is one row block diagonal with elements from $x(\alpha)$. Each row has different "consecutively" sub-vectors $x_j(\alpha)$ appearing in two sub-matrix X_j and for all sub-matrix, each row is repeated M times for $j = 1, \ldots, J$. Therefore,

$$\begin{aligned}d_F(\mathcal{A}) &= \sup_{A(\alpha)\in\mathcal{A}}\|A(\alpha)\|_F\\ &= \sup_{\alpha\in\Omega_S}\frac{1}{\sqrt{2M}}\sqrt{2M\|x_1(\alpha)\|_2^2 + 2M\|x_2(\alpha)\|_2^2 + \cdots + 2M\|x_J(\alpha)\|_2^2}\\ &= \sup_{\alpha\in\Omega_S}\|x(\alpha)\|_2\\ &= \sup_{\alpha\in\Omega_S}\|U\alpha\|_2 = \sup_{\alpha\in\Omega_S}\|\alpha\|_2 = 1. \qquad (3.13)\end{aligned}$$

Calculating $d_2(\mathcal{A})$

To do this calculation, $d_2(\mathcal{A}) = \sup_{A(\alpha)\in\mathcal{A}}\|A(\alpha)\|_2$, let us work first with $\|A(\alpha)\|_2$ using the property (3.7) mentioned above

$$\begin{aligned}
\|A(\alpha)\|_2 &= \|A(\alpha)A(\alpha)^*\|_2^{1/2} \\
&= \frac{1}{\sqrt{2M}} \max_{j\in[J]} \sqrt{\|x_j(\alpha)\|_2^2 + \|x_{j'}(\alpha)\|_2^2} \\
&\leq \frac{1}{\sqrt{2M}} \sqrt{2 \max_{j\in[J]} \|x_j(\alpha)\|_2^2} \\
&= \frac{1}{\sqrt{M}} \max_{j\in[J]} \sqrt{\|x_j(\alpha)\|_2^2} \\
&= \frac{1}{\sqrt{M}} \max_{j\in[J]} \|x_j(\alpha)\|_2 \\
&= \frac{1}{\sqrt{M}} \max_{i\in[J]} \|U_j\alpha\|_2,
\end{aligned} \tag{3.14}$$

from each rectangular block in U, called U_j, $U_j\alpha$ is the j-subvector of x, called $x_j(\alpha) = U_j\alpha$. Let be $u_{j,n}, j \in [J]$ and $n \in [N]$, the $((j-1)N+n)$th row of U. Then, continuing the calculation

$$\begin{aligned}
\|A(\alpha)\|_2 &\leq \frac{1}{\sqrt{M}} \max_{i\in[J]} \|U_j\alpha\|_2 \\
&\leq \frac{1}{\sqrt{M}} \left(N \max_{j\in[J],n\in[N]} \langle u_{j,n}, \alpha\rangle^2 \right)^{1/2} \\
&= \frac{1}{\sqrt{M}} \sqrt{N} \max_{j\in[J],n\in[N]} |\langle u_{j,n}, \alpha\rangle| \\
&\leq \frac{1}{\sqrt{JM}} \left(\sqrt{JN} \max_{j\in[J],n\in[N]} \|u_{j,n}\|_\infty \right) \|\alpha\|_1 \\
&= \frac{\mu}{\sqrt{\widetilde{M}}} \|\alpha\|_1,
\end{aligned} \tag{3.15}$$

where $\mu = \mu(U)$, the biggest absolute value entry of U times $\sqrt{\widetilde{N}}$, called the *coherence of* U.

If we bounded $\max_{i\in[J]}\|x_j\|_2$ in (3.14) with its whole vector, then

$$\begin{aligned}\|A(\alpha)\|_2 &\leq \frac{1}{\sqrt{2M}}\max_{i\in[J]}\|x_j\|_2\\ &\leq \frac{1}{\sqrt{2M}}\|x\|_2\\ &= \frac{1}{\sqrt{2M}} = \frac{\sqrt{J/2}}{\sqrt{\widetilde{M}}}.\end{aligned} \tag{3.16}$$

Then, from (3.15) and (3.16), the two calculated bounds

$$\begin{aligned}\|A(\alpha)\|_2 &\leq \frac{\mu}{\sqrt{\widetilde{M}}}\|\alpha\|_1,\\ \|A(\alpha)\|_2 &\leq \frac{\sqrt{J/2}}{\sqrt{\widetilde{M}}},\end{aligned}$$

finally, if we use that $\|\alpha\|_1 \leq S$ and $\sqrt{\frac{J}{2}} = \sqrt{\frac{J}{2S}}\sqrt{S}$, we got

$$\begin{aligned}\|A(\alpha)\|_2 &\leq \frac{1}{\sqrt{\widetilde{M}}}\min\left(\mu\|\alpha\|_1, \sqrt{J/2}\right),\\ &\leq \frac{1}{\sqrt{\widetilde{M}}}\min\left(\mu\sqrt{S}, \sqrt{\frac{J}{2S}}\sqrt{S}\right)\\ &= \sqrt{\frac{S}{\widetilde{M}}}\min\left(\mu, \sqrt{\frac{J}{2S}}\right)\end{aligned} \tag{3.17}$$

Therefore, applying sup, the bound for $d_2(\mathcal{A})$ is

$$\begin{aligned}d_2(\mathcal{A}) &= \sup_{A(\alpha)\in\mathcal{A}}\|A(\alpha)\|_2\\ &\leq \sup_{\alpha\in\Omega_S}\sqrt{\frac{S}{\widetilde{M}}}\min\left(\mu, \sqrt{\frac{J}{2S}}\right)\\ &= \widetilde{\mu^*}\sqrt{\frac{S}{\widetilde{M}}},\end{aligned} \tag{3.18}$$

where $\widetilde{\mu^*} = \min\left(\mu, \sqrt{\frac{J}{2S}}\right)$ and $\mu(U) = \sqrt{\widetilde{N}}\max_{p,q\in[\widetilde{N}]}|U(p,q)|$.

3.2.6 Calculation 2

The last value involved in *Chaos Theorem 14* is $\gamma_2(\mathcal{A}, \|\cdot\|)$, a geometry property of $\mathcal{A}$. It is part of a family of functionals which is the core of the *generic chaining method* developed and described in a book written in 1931 by A. Kolmogorov. In 1967, R. Dudley developed a method to bound it using *covering number*, although it provides a very efficient bound for Gaussian processes, unfortunately, in some cases the bound is not tight and some "parasitic" logarithmic factor will be present [2]. In 1985, X. Fernique and Talagrand, found the precise relationship between the "size" of a Gaussian process and the "size" of this metric and provided the *missing understanding* in the case of these processes. The book written for Talagrand is an attempt to extend this result to other processes.

The bound for the functional γ_2

The bound by Dudley is given in the next lemma,

Lemma 4. *(See Talagrand's book) Let $\mathcal{A}$ be a set as defined above, then*

$$\gamma_2(\mathcal{A}, \|\cdot\|) \lesssim \int_0^\infty \log^{\frac{1}{2}}\left(\mathcal{N}(\mathcal{A}, \|\cdot\|, v)\right) dv.$$

The covering number $\mathcal{N}$ is defined as the minimal cardinality of all covers $\mathcal{C}$ of $\mathcal{A}$. A set $\mathcal{C}(\mathcal{A}, \|\cdot\|, r)$ is called *a cover* for $\mathcal{A}$ at resolution r and with respect to the metric $\|\cdot\|$ if for every $x \in \mathcal{A}$, there exists $x' \in \mathcal{C}(\mathcal{A}, \|\cdot\|, r)$ such that $\|x - x'\| \leq r$.

The covering number for $\mathcal{A}$, should be the same as the covering number of Ω_S, the set of all

[2] Taken from Talagrand's book

unitary S-sparse vectors. This is because the definition of $\mathcal{A}$ and its metrics, therefore

$$\gamma_2(\mathcal{A}, \|\cdot\|) \lesssim \int_0^\infty \log^{\frac{1}{2}} \mathcal{N}\big(\Omega_S, \|\cdot\|_A, v\big) dv.$$

Using the trivial identity $\mathcal{N}(c\mathcal{S}, \|\cdot\|, r) = \mathcal{N}(\mathcal{S}, \|\cdot\|, r/c)$, we got

$$\begin{aligned}
\gamma_2(\mathcal{A}, \|\cdot\|) &\lesssim \int_0^\infty \log^{\frac{1}{2}} \mathcal{N}\left(\Omega_S, \|\cdot\|_A, v\right) dv \\
&= \int_0^\infty \log^{\frac{1}{2}} \mathcal{N}\left(\frac{\sqrt{S}\Omega_S}{\sqrt{S}}, \|\cdot\|_A, v\right) dv \\
&= \int_0^\infty \log^{\frac{1}{2}} \mathcal{N}\left(\frac{\Omega_S}{\sqrt{S}}, \|\cdot\|_A, \frac{v}{\sqrt{S}}\right) dv.
\end{aligned}$$

By using a simple "change of variable" in the integral by substitution is transformed to

$$\gamma_2(\mathcal{A}, \|\cdot\|) \lesssim \sqrt{S} \int_0^\infty \log^{\frac{1}{2}} \mathcal{N}\left(\frac{\Omega_S}{\sqrt{S}}, \|\cdot\|_A, v\right) dv. \tag{3.19}$$

Before trying to bound the covering number over $\Omega/\sqrt{S}$, let us do an example to gain a better understanding of it.

Example. *Let us consider the real line $\mathbb{R}$ and the absolute value $|\cdot|$ as a metric space and define $\mathcal{A}$ as the set of real numbers whose absolute value is at most k, i.e., $\mathcal{A} = \{x \in \mathbb{R} : |x| \leq k\}$. The covering of $\left\lceil \frac{2k}{r} \right\rceil$ intervals covering the interval $[-k, k]$, therefore*

$$\mathcal{N}(\mathcal{A}, |\cdot|, r) \leq \frac{2k}{r}.$$

Note that $\mathcal{N}(\mathcal{A}, |\cdot|, r) = 1$ for $r \geq 2k$, i.e, exist just one set in the cover of $\mathcal{A}$.

Our task is trying to find a bound for $\mathcal{N}\left(\frac{\Omega_S}{\sqrt{S}}, \|\cdot\|_A, v\right)$ in terms of the independent variable v, i.e., the radius of the elements in the cover, to bound the integral. The first to note is

that it should exist $v_{\max}$ such that

$$\int_0^{\infty} \log^{\frac{1}{2}} \mathcal{N}\left(\frac{\Omega_S}{\sqrt{S}}, \|\cdot\|_A, v\right) dv = \int_0^{v_{\max}} \log^{\frac{1}{2}} \mathcal{N}\left(\frac{\Omega_S}{\sqrt{S}}, \|\cdot\|_A, v\right) dv, \tag{3.20}$$

this is because for $v \geq v_{\max}$, $\mathcal{N}\left(\frac{\Omega_S}{\sqrt{S}}, \|\cdot\|_A, v\right) = 1$ and therefore the log is zero.

The function $\mathcal{N}\left(\frac{\Omega_S}{\sqrt{S}}, \|\cdot\|_A, v\right)$ is one that for small radios its values are "high" and for large radios its value is close to 1, and exactly 1 for values greater than one $v_{\max}$.

To do that, let us consider a similar lemma from Eftekhari's paper [27]. The idea is presented in Figure 3.1, what we want is find two decreasing functions f and g. To approximate the integral (3.19), the values v_0 and $v_{\max}$ divide the integral in two parts, thereby finding two bounds for those integrals. Note again that after $v_{\max}$ value, the covering number will be one and using the log, it is zero and is not going to contribute to the integral.

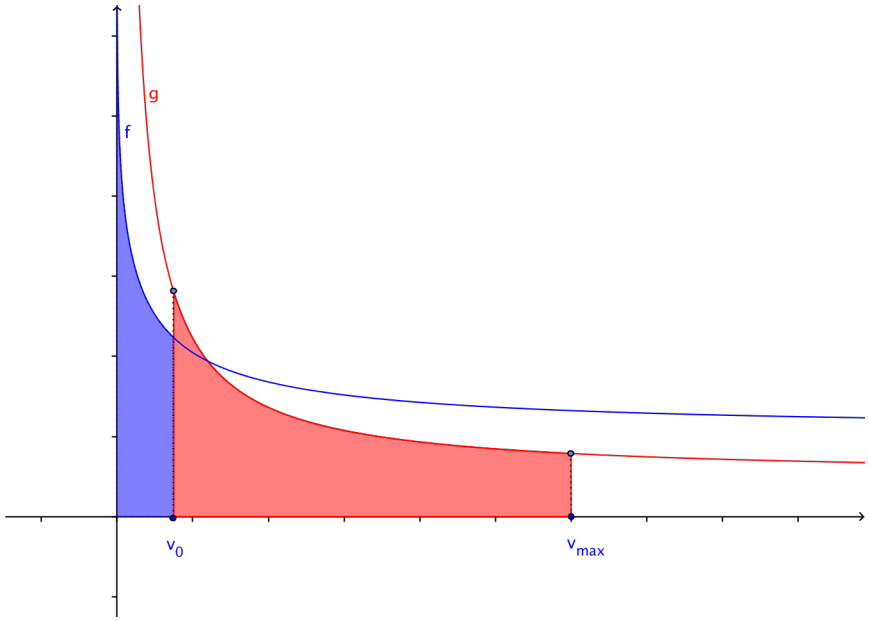


Figure 3.1: Idea of bounds for Covering Lemma

Lemma 5 (The Covering Lemma for $\Omega_S/\sqrt{S}$)**.** *Consider a norm* $\|\cdot\|_A$ *on* $\mathbb{C}^{\tilde{N}}$*, for every* $\alpha \in \mathbb{C}^{\tilde{N}}$*, satisfies*

$$\|\alpha\|_A = \|A(\alpha)\|_2 \leq \frac{\kappa}{\sqrt{\widetilde{M}}},$$

for some linear map $A(\cdot) : \mathbb{C}^{\widetilde{N}} \to \mathbb{C}^{N'}$ *with rank of at most* $\widetilde{M}$ *and some* $\kappa > 0$ *and integer* N'. *Then, for* $0 < v < \kappa/\sqrt{\widetilde{M}}$ *and* $\widetilde{M} \gtrsim 1$, *we have that*

$$\log\left(\mathcal{N}\left(\frac{\Omega_S}{\sqrt{S}}, \|\cdot\|_A, v\right)\right) \lesssim \min\left\{S\log\widetilde{N} + S\log\left(1 + \frac{2\kappa}{v\sqrt{\widetilde{M}}}\right), \frac{\kappa^2}{v^2\widetilde{M}}\log\widetilde{M}\log\widetilde{N}\right\}.$$

When $v \geq \kappa/\sqrt{\widetilde{M}}$, *we have* $\mathcal{N}\left(\frac{\Omega_S}{\sqrt{S}}, \|\cdot\|_A, v\right) = 1$.

Proof. Before end this section. □

Because the variable v is taking values like $\kappa/\sqrt{\widetilde{M}}$ and usually $\kappa = \widetilde{\mu^*} = \sqrt{\frac{J}{2S}}$, then Bound left $\to S\log\widetilde{N}$ and Bound right $\to \left(\frac{\log\widetilde{N}}{\widetilde{M}}\right)^2$ which means that for small values of v the left one is better but for larger values the right one must be chosen.

The norm $\|\cdot\|_A$ satisfies the hypothesis of Lemma 5 with $\kappa = \widetilde{\mu^*}$ and the linear map defined above $A(\cdot)$, then applying it for the values of v in $0 < v_0 \leq \widetilde{\mu^*}/\sqrt{\widetilde{M}}$ and a v_0 value setting later, we have

$$\begin{aligned}
\int_0^\infty \log^{\frac{1}{2}}\mathcal{N}\left(\frac{\Omega_S}{\sqrt{S}}, \|\cdot\|_A, v\right)dv &= \int_0^{v_0}\log^{\frac{1}{2}}\mathcal{N}\left(\frac{\Omega_S}{\sqrt{S}}, \|\cdot\|_A, v\right)dv + \int_{v_0}^{\frac{\widetilde{\mu^*}}{\sqrt{\widetilde{M}}}}\log^{\frac{1}{2}}\mathcal{N}\left(\frac{\Omega_S}{\sqrt{S}}, \|\cdot\|_A, v\right)dv \\
&\lesssim \int_0^{v_0}\log^{\frac{1}{2}}\left(\sqrt{S\log\widetilde{N}} + \sqrt{S\log\left(1 + \frac{2\widetilde{\mu^*}}{v\sqrt{\widetilde{M}}}\right)}\right)dv + \sqrt{\log\widetilde{M}\log\widetilde{N}}\int_{v_0}^{\frac{\widetilde{\mu^*}}{\sqrt{\widetilde{M}}}}\frac{\widetilde{\mu^*}}{v\sqrt{\widetilde{M}}}dv \\
&\lesssim v_0\sqrt{S\log\widetilde{N}} + v_0\sqrt{S\log\left(1 + \frac{2\widetilde{\mu^*}}{v_0\sqrt{\widetilde{M}}}\right)} + \frac{\widetilde{\mu^*}}{\sqrt{\widetilde{M}}}\sqrt{\log\widetilde{M}\log\widetilde{N}}\log\left(\frac{\widetilde{\mu^*}}{v_0\sqrt{\widetilde{M}}}\right),
\end{aligned}$$

choosing the value $v_0 = \widetilde{\mu^*}/\sqrt{S\widetilde{M}}$ the right hand side is

$$\begin{aligned}
&\lesssim \frac{\widetilde{\mu^*}}{\sqrt{\widetilde{M}}}\sqrt{\log\widetilde{N}} + \frac{\widetilde{\mu^*}}{\sqrt{\widetilde{M}}}\sqrt{\log(1 + 2\sqrt{S})} + \frac{\widetilde{\mu^*}}{\sqrt{\widetilde{M}}}\log S\sqrt{\log\widetilde{M}\log\widetilde{N}} \\
&\lesssim \frac{\widetilde{\mu^*}}{\sqrt{\widetilde{M}}}\log S\sqrt{\log\widetilde{M}\log\widetilde{N}}.
\end{aligned}$$

Adding the $\sqrt{S}$ factor from (3.19), the final bound is

$$\gamma_2(\mathcal{A}, \|\cdot\|) \lesssim \sqrt{S} \int_0^\infty \log^{\frac{1}{2}} \mathcal{N}\left(\frac{\Omega_S}{\sqrt{S}}, \|\cdot\|_A, v\right) dv \lesssim \widetilde{\mu^*} \sqrt{\frac{S}{\widetilde{M}}} \log S \sqrt{\log \widetilde{M} \log \widetilde{N}}. \tag{3.21}$$

3.2.7 Applying The Chaos Theorem

Remembering that a matrix satisfies RIP if δ_S is small for reasonably large S. Moreover, in the property

$$(1 - \delta_S)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1 + \delta_S)\|x\|_2^2 \text{ for all } x \text{ with } \|U^* x\|_0 = \|\alpha\|_0 \leq S.$$

δ_S was converted to

$$\delta_S = \sup_{\alpha \in \Omega_S} \left| \|\Psi \cdot x(\alpha)\|_2^2 - 1 \right|.$$

Therefore, taking a $\delta < 1$, the objective is show that $\delta_S < \delta$ for the matrix Ψ, the already calculated bounded values,

$$\begin{aligned} d_F(\mathcal{A}) &= 1 \\ d_2(\mathcal{A}) &\leq \widetilde{\mu^*} \sqrt{\frac{S}{\widetilde{M}}} \\ \gamma_2(\mathcal{A}, \|\cdot\|_2) &\lesssim \widetilde{\mu^*} \sqrt{\frac{S}{\widetilde{M}}} \log S \sqrt{\log \widetilde{M} \log \widetilde{N}} \end{aligned}$$

allow us to bound the values E, V and U. This finally enable us apply the *Chaos Theorem 14* to get the main goal of prove RIP.

The E value

Let us compute the first one

$$\begin{aligned} E &= \gamma_2(\mathcal{A}, \|A\cdot\|_2)\big(\gamma_2(\mathcal{A}, \|\cdot\|_2 + d_F(\mathcal{A})\big) + d_F(\mathcal{A})d_2(\mathcal{A}), \\ &\lesssim \widetilde{\mu^*}\sqrt{\frac{S}{\widetilde{M}}}\log S\sqrt{\log\widetilde{M}\log\widetilde{N}}\left(\widetilde{\mu^*}\sqrt{\frac{S}{\widetilde{M}}}\log S\sqrt{\log\widetilde{M}\log\widetilde{N}}+1\right) + \widetilde{\mu^*}\sqrt{\frac{S}{\widetilde{M}}}, \end{aligned}$$

for a number $\delta < 1$, let us assume that $\widetilde{\mu^*}\sqrt{\frac{S}{\widetilde{M}}}\log S\sqrt{\log\widetilde{M}\log\widetilde{N}} \lesssim \delta$. It is from this assertion, but rewriting it as $\widetilde{M} \gtrsim_\tau \delta^{-2}\widetilde{\mu^*}^2 S\log^2 S\log\widetilde{M}\log\widetilde{N}$, and taken $S \gtrsim 1$ where we get the number of measurements, the bound now is

$$\begin{aligned} E &\lesssim \delta(\delta+1) + \frac{\delta}{\log S\sqrt{\log\widetilde{M}\log\widetilde{N}}} \\ &\leq 2\delta + \frac{\delta}{\log S\sqrt{\log\widetilde{M}\log\widetilde{N}}} \\ &\lesssim \delta \end{aligned}$$

The V value

For the second one, assuming the same as before for δ but rewritten as

$$\widetilde{\mu^*}\sqrt{\frac{S}{M}} \lesssim_\tau \frac{\delta}{\log S\sqrt{\log\widetilde{M}\log\widetilde{N}}},$$

we have that

$$
\begin{aligned}
V &= d_2(\mathcal{A})\big(\gamma_2(\mathcal{A}, \|\cdot\| + d_f(\mathcal{A})\big), \\
&\lesssim \widetilde{\mu^*}\sqrt{\frac{S}{\widetilde{M}}}\left(\widetilde{\mu^*}\sqrt{\frac{S}{\widetilde{M}}}\log S\sqrt{\log\widetilde{M}\log\widetilde{N}} + 1\right) \\
&= \widetilde{\mu^*}^2\frac{S}{\widetilde{M}}\log S\sqrt{\log\widetilde{M}\log\widetilde{N}} + \widetilde{\mu^*}\sqrt{\frac{S}{\widetilde{M}}} \\
&\lesssim_\tau \frac{\delta^2}{\log S\sqrt{\log\widetilde{M}\log\widetilde{N}}} + \frac{\delta}{\log S\sqrt{\log\widetilde{M}\log\widetilde{N}}} \\
&\lesssim_\tau \frac{\delta}{\log S\sqrt{\log\widetilde{M}\log\widetilde{N}}}(\delta + 1) \\
&\lesssim \frac{\delta}{\log S\sqrt{\log\widetilde{M}\log\widetilde{N}}}.
\end{aligned}
$$

The U value

Finally, for the last one rewritten δ as in V, we got

$$
\begin{aligned}
U &= d_2^2(\mathcal{A}) \\
&\leq \left(\widetilde{\mu^*}\sqrt{\frac{S}{\widetilde{M}}}\right)^2 \\
&\lesssim_\tau \frac{\delta^2}{\log^2 S\log\widetilde{M}\log\widetilde{N}}.
\end{aligned}
$$

Putting it all together

From *Chaos Theorem 14*, for a $t > 0$, it holds that

$$
\log P\left\{\sup_{A\in\mathcal{A}}\left|\|A\varepsilon\|_2^2 - \mathbb{E}\|A\varepsilon\|_2^2\right| \gtrsim_\tau E + t\right\} \lesssim_\tau -\min\left(\frac{t^2}{V^2}, \frac{t}{U}\right),
$$

then using all the previous calculations

$$\left|\|A\varepsilon\|_2^2 - \mathbb{E}\|A\varepsilon\|_2^2\right| \to \left|\|A(\alpha)\varepsilon\|_2^2 - 1\right| = \left|\|\Psi x(\alpha)\|_2^2 - 1\right|$$

$$\delta_S = \sup_{\alpha\in\Omega_S} \left|\|\Psi \cdot x(\alpha)\|_2^2 - 1\right|$$

$$E \lesssim \delta$$

$$V \lesssim \frac{\delta}{\log S\sqrt{\log \widetilde{M}\log \widetilde{N}}}$$

$$U \lesssim_\tau \frac{\delta^2}{\log^2 S\log \widetilde{M}\log \widetilde{N}},$$

we got that

$$\log P\left\{\sup_{\alpha\in\Omega_S}\left|\|\Psi x(\alpha)\|_2^2 - 1\right| \gtrsim_\tau \delta + t\right\} \lesssim_\tau -\min\left\{\frac{t^2}{\left(\frac{\delta}{\log S\sqrt{\log \widetilde{M}\log \widetilde{N}}}\right)^2}, \frac{t}{\frac{\delta^2}{\log^2 S\log \widetilde{M}\log \widetilde{N}}}\right\}$$

$$= -\min\left\{\delta^{-2}t^2\log^2 S\log \widetilde{M}\log \widetilde{N}, \delta^{-2}t\log^2 S\log \widetilde{M}\log \widetilde{N}\right\}$$

and changing $t = \delta$, then

$$\log P\left\{\sup_{\alpha\in\Omega_S}\left|\|\Psi x(\alpha)\|_2^2 - 1\right| \gtrsim_\tau \delta\right\} \lesssim_\tau -\min\left\{\log^2 S\log \widetilde{M}\log \widetilde{N}, \delta^{-1}\log^2 S\log \widetilde{M}\log \widetilde{N}\right\},$$

and if we assume again that $S \gtrsim 1$, and redefined δ to eliminate the factor depending of τ inside the probability. In addition, using the fact that if $0 < \delta < 1$ then $\delta^{-1} > 1$ we finally got that

$$\log P\left\{\sup_{\alpha\in\Omega_S}\left|\|\Psi x(\alpha)\|_2^2 - 1\right| \gtrsim \delta\right\} \lesssim_\tau -\log^2 S\log \widetilde{M}\log \widetilde{N}$$

or

$$P\left\{\sup_{\alpha\in\Omega_S}\left|\|\Psi x(\alpha)\|_2^2-1\right|\gtrsim\delta\right\}\lesssim_\tau 10^{-\log^2 S\log\widetilde{M}\log\widetilde{N}}$$
$$=\left(10^{\log\widetilde{N}}\right)^{-\log^2 S\log\widetilde{M}},$$

which means

$$P\{\delta_S\gtrsim\delta\}\lesssim_\tau \widetilde{N}^{-\log^2 S\log\widetilde{M}}.$$

The last equation shows that the probability that δ_S is greater than δ is very low, therefore we have proved that the matrix Ψ satisfies RIP with high probability, this finally proves Theorem 13. □

Details about the *Covering Lemma 5*

The calculation of the bound for γ_2 has a a lot of details and techniques that lemma 5 hides. Before ending this section, we are going to give some details about its prove, which we called previously *The Covering Lemma 5*.

The covering number of a set $\mathcal{A}$ at resolution v with respect to the norm $\|\cdot\|$, denoted by $\mathcal{N}(\mathcal{A},\|\cdot\|,v)$ is the minimal cardinality of all possible covers. To bound the covering number of $\frac{\Omega_S}{\sqrt{S}}$, let us use unit balls with different norms as elements on covers, i.e.,

$$\mathcal{B}_A^{\widetilde{N}}=\{\beta\in\mathbb{C}^{\widetilde{N}}:\|\alpha\|_A\leq 1\}$$
$$\mathcal{B}_1^{\widetilde{N}}=\{\beta\in\mathbb{C}^{\widetilde{N}}:\|\alpha\|_1\leq 1\}$$
$$\mathcal{B}_2^{\widetilde{N}}=\{\beta\in\mathbb{C}^{\widetilde{N}}:\|\alpha\|_2\leq 1\},$$

also, for $T\subseteq[\widetilde{N}]$, where $\#T=S$, the unit balls in the S-dimensional subspace of $\mathbb{C}^{\widetilde{N}}$

spanned by $\{e_{\widetilde{n}}\}$, $\widetilde{n} \in T$ as

$$\mathcal{B}_A^T = \{\beta \in \mathbb{C}^S : \|\alpha\|_A \leq 1\}$$
$$\mathcal{B}_1^T = \{\beta \in \mathbb{C}^S : \|\alpha\|_1 \leq 1\}$$
$$\mathcal{B}_2^T = \{\beta \in \mathbb{C}^S : \|\alpha\|_2 \leq 1\}.$$

We can check first that $\mathcal{B}_2^T$ is related with $\mathcal{B}_A^T$ as is presented in the next lemma.

Lemma 6. *For any support $T \subset [\widetilde{N}]$ with $|T| = S$, we have*

$$\frac{\mathcal{B}_2^T}{\sqrt{S}} \subseteq \frac{\kappa}{\sqrt{\widetilde{M}}}\mathcal{B}_A^T. \tag{3.22}$$

Proof. From the set of all signal in $\widetilde{N}$-dimensional complex space with unit norm, i.e.,

$$\Omega_S = \left\{\alpha \in \mathbb{C}^{\widetilde{N}} : \|\alpha\|_2 = 1, \|\alpha\|_0 \leq S\right\},$$

and norm property $\|\alpha\|_1 = \sqrt{\widetilde{N}}\|\alpha\|_2$ or $\|\alpha\|_1 = \sqrt{S}\|\alpha\|_2$ because in the support T, $|T| = S$, we have that if $\alpha \in \frac{\Omega_S}{\sqrt{S}}$, then $\|\alpha\| \leq 1$, i.e.,

$$\frac{\Omega_S}{\sqrt{S}} = \left\{\alpha \in \mathbb{C}^{\widetilde{N}} : \|\alpha\|_2 = \frac{1}{\sqrt{S}}, \|\alpha\|_0 \leq S, \|\alpha\|_1 \leq 1\right\},$$

then used the hypothesis of the *Covering Lemma 5* we have

$$\|\alpha\|_A \leq \frac{\kappa}{\sqrt{\widetilde{M}}}. \tag{3.23}$$

Let us consider any β on the surface of ball $\frac{\mathcal{B}_2^T}{\sqrt{S}}$, i.e,. $\|\beta\|_2 = \frac{1}{\sqrt{S}}$ and therefore $\beta \in \frac{\Omega_S}{\sqrt{S}}$, i.e., $\|\beta\|_2 \leq 1$. Consequently, according to (3.23), $\|\beta\|_A \leq \frac{\kappa}{\sqrt{\widetilde{M}}}$, and therefore $\beta \in \frac{\kappa}{\sqrt{\widetilde{M}}}\mathcal{B}_A^T$.

Next, for every $\beta \in \frac{\mathcal{B}_2^T}{\sqrt{S}}$, $\|\beta\|_2 = \frac{r}{\sqrt{S}}$, where $r \leq 1$, we can find $\beta' = \frac{1}{r}\beta$ that is on the surface

of ball $\frac{\mathcal{B}_2^T}{\sqrt{S}}$. Then, $\|\beta'\| \leq \frac{\kappa}{\sqrt{\widetilde{M}}}$, which means $\|\beta\|_A \leq \frac{r\kappa}{\sqrt{\widetilde{M}}} \leq \frac{\kappa}{\sqrt{\widetilde{M}}}$, so $\beta \in \frac{\kappa}{\sqrt{\widetilde{M}}}\mathcal{B}_A^T$. □

The set $\frac{\Omega_S}{\sqrt{S}}$ can be represented now by unions of balls, $\frac{\mathcal{B}_2^T}{\sqrt{S}}$ over the supports T, as well as by lemma 6 with $\frac{\kappa}{\sqrt{\widetilde{M}}}\mathcal{B}_A^T$, i.e.,

$$\frac{\Omega_S}{\sqrt{S}} = \bigcup_{|T|=S} \frac{\mathcal{B}_2^T}{\sqrt{S}} \subseteq \bigcup_{|T|=S} \frac{\kappa}{\sqrt{\widetilde{M}}}\mathcal{B}_A^T. \tag{3.24}$$

Before continuing, note that the set $\frac{\Omega_S}{\sqrt{S}}$ can be covered with just one ball $\frac{\kappa}{\sqrt{\widetilde{M}}}\mathcal{B}_A^{\widetilde{N}}$. Therefore, $\mathcal{N}\left(\Omega_S/\sqrt{S}, \|\cdot\|_A, v\right) = 1$, if $v \geq \kappa/\sqrt{\widetilde{M}}$. This proves the last part of *Covering Lemma 5.*

Now, from (3.24), if $v \leq \kappa/\sqrt{\widetilde{M}}$ then

$$\begin{aligned}\mathcal{N}\left(\frac{\Omega_S}{\sqrt{S}}, \|\cdot\|_A, v\right) &\leq \mathcal{N}\left(\bigcup_{|T|=S} \frac{\kappa}{\sqrt{\widetilde{M}}}\mathcal{B}_A^T, \|\cdot\|_A, v\right) \\ &\leq \binom{\widetilde{N}}{S}\mathcal{N}\left(\frac{\kappa}{\sqrt{\widetilde{M}}}\mathcal{B}_A^T, \|\cdot\|_A, v\right),\end{aligned}$$

using again the trivial identity $\mathcal{N}(c\mathcal{S}, \|\cdot\|, r) = \mathcal{N}(\mathcal{S}, \|\cdot\|, r/c)$ we get,

$$\binom{\widetilde{N}}{S}\mathcal{N}\left(\mathcal{B}_A^T, \|\cdot\|_A, v\kappa^{-1}\sqrt{\widetilde{M}}\right).$$

Using a similar result from [39] in Lemma 5.2 we have $\mathcal{N}\left(\mathcal{B}_A^T, \|\cdot\|_A, r\right) \leq (1 + 2/r)^{2S}$ for $r > 0$. Then,

$$\mathcal{N}\left(\frac{\Omega_S}{\sqrt{S}}, \|\cdot\|_A, v\right) \leq \binom{\widetilde{N}}{S}\left(1 + \frac{2\kappa}{v\sqrt{\widetilde{M}}}\right)^{2S} \leq \left(\frac{e\widetilde{N}}{S}\right)^S \left(1 + \frac{2\kappa}{v\sqrt{\widetilde{M}}}\right)^{2S},$$

when $\widetilde{N} \gtrsim 1$ we have,

$$\log \mathcal{N}\left(\frac{\Omega_S}{\sqrt{S}}, \|\cdot\|_A, v\right) \leq S \log\left(\frac{e\widetilde{N}}{S}\right) + 2S \log\left(1 + \frac{2\kappa}{v\sqrt{\widetilde{M}}}\right)$$
$$\lesssim S \log \widetilde{N} + S \log\left(1 + \frac{2\kappa}{v\sqrt{\widetilde{M}}}\right). \tag{3.25}$$

We discussed in the paragraph after the *Covering Lemma 5* that this bound is good for small values of v, but for larger values they just proposed another one. To find that bound let us use again that the set $\frac{\Omega_S}{\sqrt{S}}$ can be covered with just one ball, but in this case, $\frac{\Omega_S}{\sqrt{S}} \subseteq \mathcal{B}_1^{\widetilde{N}}$, then our task is now to find a cover for this ball because

$$\mathcal{N}\left(\frac{\Omega_S}{\sqrt{S}}, \|\cdot\|_A, v\right) \leq \mathcal{N}\left(\mathcal{B}_1^{\widetilde{N}}, \|\cdot\|_A, v\right). \tag{3.26}$$

The last proposition on this section is going to be a lemma with that bound. For the prove we refer the reader to [27] on Appendix F.

Lemma 7. *Let $B_{1,\widetilde{N}}$ denote the l_1-ball in $\mathbb{R}^{\widetilde{N}}$, and consider the norm $\|\cdot\|_A$ on $\mathbb{C}^{\widetilde{N}}$, which satisfies the hypothesis of* Covering Lemma 5. *Naturally, $\|\cdot\|_A$ induces a norm on $\mathbb{R}^{\widetilde{N}} \subset \mathbb{C}^{\widetilde{N}}$ For $v > 0$ and $\widetilde{M} \gtrsim 1$, it holds that*

$$\log\left(\mathcal{N}(B_{1,\widetilde{N}}, \|\cdot\|_A, v)\right) \lesssim \frac{\kappa^2}{v^2 \widetilde{M}} \log \widetilde{M} \log \widetilde{N}.$$

The previous lemma bound $B_{1,\widetilde{N}}$ where its elements are in $\mathbb{R}^{\widetilde{N}}$. The strategy uses this two times to bound $\mathcal{B}_1^{\widetilde{N}}$ where its elements are in $\mathbb{C}^{\widetilde{N}}$, i.e.,

$$\mathcal{N}\left(\mathcal{B}_1^{\widetilde{N}}, \|\cdot\|_A, v\right) \leq \left(\mathcal{N}\left(B_{1,\widetilde{N}}, \|\cdot\|_A, v/2\right)\right)^2,$$

then using lemma 7, finally we got the another bound for bigger values of v.

$$\log \mathcal{N}\left(\frac{\Omega_S}{\sqrt{S}}, \|\cdot\|_A, v\right) \lesssim \frac{\kappa^2}{v^2 \widetilde{M}} \log \widetilde{M} \log \widetilde{N}. \tag{3.27}$$

3.3 Overview of Numerical Experiments

In this section, we will support our result with numerical experiments. To be able to do so, we use cutting edge good tools. They were very important because in many cases they allowed us to obtain and see the results easily; we will show those results below. The main objectives taking in account for the experiments were:

1. validate the theoretical analysis, and
2. demonstrate the advantages of COB over BD.

An ideal model to validate the theorem should create several COB matrices and for each of them, while we are varying the sparsity S, the number of measurements $\widetilde{M}$ and the number of blocks J, we get the ratio of how many RIC values are below of a fixed threshold overall number of COB matrices tested. But, as it has been suspected for some time that evaluating RIP is a NP hard problem in general, Tillmann and Pfetsch in [40] confirm that conjecture and therefore made that scenario impossible to realize. There exists some research, see for example [41] and [42], looking for alternative or heuristic algorithms to approximate RIC but have some limitations. We may consider this option for future works.

Instead of checking whether the RIC holds, we will run experiments using empirical signal recovery through the minimizer problem l_1 from Python package CVXPY to find the minimal number of measurement $\widetilde{M}$ so that the program starts to exactly recovered signals effectively validating the factors in the bound of the Theorem 13 for COB.

The rest of this section is organized as what follows. First, in 3.3.1 we will explain the simulation program and settings. Next in 3.4.1 we will show using three cases of signal sparsity the performance of BD and COB over several numbers of blocks. It will allow us to validate the J factor bound in our theoretical result. We also included some graphs that will help us assert how much performance is improved using COB over BD. In part 3.4.2, we focus on the way that the sparsity influences the number of samples needed to recover the signal. For the last factor in 3.4.3, we also show how relevant is the size of the signal, a typical result in CS usually said is above a logarithm factor. The last part of this section shows a important analysis about the worst signal sparsity case in 3.5, again it can be seen that COB improves BD.

3.3.1 The Simulation Program and Settings

The first attempt to show the natural improvement of COB over BD was made using a UPRM institutional Matlab © version and a l_1-MAGIC collection of routines for solving the convex optimization focus in compressive sampling. The l_1-magic solver was performed good but it was coded for Matlab. Therefore, we searched for new ones and over one of the more popular programming languages today, we found CVXPY, ConVeX optimization problems in Python. CVXPY is a Python-embedded modeling language for convex optimization problems. It allows to express problems in a natural way that follows the math, rather than in the restrictive standard form required by solvers.[3]

The Evaluator

Before describing the evaluator, we will show how we create random matrices. It is well know in CS theory that random matrices with random variables type such as Gaussian, Bernoulli

[3]Definition taken from its website.

or Rademacher are examples of the more general case called sub-Gaussian. We implement two cases, Gaussian and Bernoulli and use those to construct the BD and COB matrices.

The function 1 return a random Gaussian matrix having mean zero and standard deviation $1/\sqrt{m}$. The *rand* function creates an array of the given shape and populate it with random samples from a uniform distribution over $[0, 1)$.

Function 1 Psi_Gaussian(m, n)

Input: $m, n > 0$
Output: $A_{m \times n}$
1: $A \leftarrow (1/\sqrt{m}) \cdot \text{rand}(m, n)$
2: **return** A

The function 2 builds a random Bernoulli matrix with equal mean and standard deviation as Gaussian. We use the same probability for $+1$ and -1, i.e., we could also have called it a Rademacher matrix.

Function 2 Psi_Bernoulli(m, n)

Input: $m, n > 0$
Output: $A_{m \times n}$
1: $A \leftarrow (1/\sqrt{m})\big(2(\text{rand}(m, n) >= 0.5) - 1\big)$
2: **return** A

The function 3 will build BD, a J different block diagonal matrix where each block has size $M \times N$. It use the function *block_diagonal* to create a block diagonal matrix from provided arrays, the for-loop add the others $J - 1$ blocks.

Function 3 Psi_BD_Bernoulli(M, N, J)

Input: $M, N \mod J = 0$
Output: $A_{M \times N}$
1: $A \leftarrow$ Psi_Bernoulli(m,n)
2: **for** $j = 1 : J - 1$ **do**
3: $A \leftarrow$block_diag(A, Psi_Bernoulli(m, n))
4: **end for**
5: **return** A

To build COB matrix in function 4, the idea is quite natural, create two BD matrices but rolling the second one n spaces and next add together to get the diagonal overlap. Note that before returning it is important to normalize the values to get the standard deviation $1/\sqrt{M}$.

Function 4 Psi_COB_Bernoulli(M, N, J)

Input: $M, N \mod J = 0$
Output: $A_{M\times N}$
1: $A_1 \leftarrow$ Psi_BD_Bernoulli(M,N)
2: $A_2 \leftarrow$ Psi_BD_Bernoulli(M,N)
3: $n = N/J$
4: $A_2 \leftarrow \text{Roll}(A_2, n)$
5: $A \leftarrow A_1 + A_2$
6: **return** $A/\sqrt{2}$

We are dealing with different variables in the experiment and we cannot vary all at the same time. Some of them we are not going to use here but our code will support other types of future experiments like signals in different base or other optimizer different than L_1. The pseudo-code in 5 will describe a class used for those purposes.

Some functions to the optimizer over library CVXPY, are showing in functions 6 and 7. If we need another type of optimizer it is easy just to create functions similar to those.

The list of functions in 8 are tools to save, get or print graphs. The function *getMaxM* will read from files the maximum measurement $\widetilde{M}$ value used to reconstruct the signal over each experiment. All those values are taken from saved files by function *save_data*.

The main procedure to run the experiments is showing in function 9, where we use a while loop to search the minimal number of measurements $\widetilde{M}$ value to recover signal. The while loop will stop if the l_2 norm is less than a tolerence value, otherwise incrementing $\widetilde{M}$ using *m_step*. The value *m_step* must be able to be divided by J. Next, we use two nested loops, one for each signal given in X as a column and trying to recover those in another loop for

Class 5 CS_Reconstructor(Matrix_Generator, Optimizer, N, j, r, Orthobase)

Input: Matrix_Generator, Optimizer, N, j, Orthobase
Output: t, L_1, L_2, x_c

```
if Orthobase = 1 then
    U = Fourier basis
else if Orthobase = 2 then
    U = Convolution Matrix
else
    U = Canonical base
end if

def Evaluate(x, M)
A = Matrix_Generator(M, N, j)
if Orthobase = 0 then
    y = Ax
else
    y = A(Ux)
end if
t_i ← get current time
x_c = Optimizer(y, A, N){obtain reconstructed signal}
t_f ← get current time
t = t_f − t_i
L_1, L_2 = L1_L2(x, x_c, N)
return t, L_1, L_2, x_c
```

Function 6 CS_Optimizer_LASSO(y, A, N)

Input: y, A, N
Output: x_c

```
lasso ← Lasso(alpha = 0.00001)
lasso.fit(A, y)
x_c ← lasso.coef_
return x_c
```

Function 7 CS_Optimizer_L1(y, A, N)

Input: y, A, N
Output: x_c

```
vx ← cvx.Variable(N)
objective ← cvx.Minimize(cvx.norm(vx, 1))
constraints ← [A · vx == y]
prob ← cvx.Problem(objective, constraints)
result ← prob.solve(verbose=False)
x_c ← vx.value
return x_c
```

Tool functions 8 Other functions

save_data(List_M, List_L1, List_L2, k, j, method, matrix, typeExp)

save_raw_data(OP_time, OP_L1, OP_L2, k, j, M, method, matrix, typeExp)

print_figure(fig, x, xc)

get_data(str)

get_raw_data(str)

mean_and_ci(data, confidence=0.95)

Get_5_Lists(strBegin, strEnd, k, j, N)

getMaxM(strBegin, strEnd, k, j)

each of the different P COB matrices. The function *save_raw_data* will save in a text file for each of the $S \cdot P$ recover intent, the time spending in the process, the l_1 and l_2 error values. When the while loop will finish, the function *save_data* will record in a new text file the M value, and the average l_1 and l_2 errors of all $S \cdot P$ tries. All files generated will have names including the N, M, K and J values for better identification.

Function 9 CS_Evaluate(tE, N, S, P, X, opt, MG, k, j, M_i, m_step, M_end, Print_Fig, U)

Input: several parameters
Output: *.txt files in folder tE, one for each recover x signal, and another with its averages.

```
M ← best M to start, J should divides M.
Opt ← choosing optimizer
MG ← selecting matrix
Reconstructor ← CS_Reconstructor(MG, Opt, N, j, r, U)
Normalize the signals X

while (M ≤ M_end)&(L2 > 0.01) do
    for i = 1 : S do
        for p = 1 : P do
            t, L1, L2, xc ← Reconstructor.Evaluate(x, M)
            Get time, L1 and L2 norms
        end for
    end for
    save_raw_data(OP_time, OP_L1, OP_L2, k, j, r, M, Str_method, Str_matrix, typeExp)

    Get L1 and L2 averages
    M ← M + m_step
end while
save_data(List_M, List_L1, List_L2, k, j, r, Str_method, Str_matrix, typeExp)
```

Settings

As mentioned previously, we start using Matlab © and l_1-MAGIC, but because we could had some licensing issues and also l_1-MAGIC was coded some time ago, we decides given a different try with other options. The very flexible and popular programming language Python was the election, first because there exist nice tools to do research interactively as is in many commercial software, the tool called *notebooks* tools is the one in which we are interested.

Jupyter Notebook is a web application over Python in which you can create and share documents easily that contain live code, equations, visualizations as well as text. The Jupyter Notebook is one of the ideal tools used today by data science researches around the world. The name "Jupyter" is a loose acronym meaning Julia, Python, and R. These programming languages were the first target languages of the Jupyter application, but nowadays, the notebook technology also supports many other languages.

Jupyter Notebook became really popular due to its versatility and ease of use, that is why the giant Google decided to incorporate it into their projects and create *Colaboratory*, which is a Google research project created to help disseminate machine learning education and research. It is a Jupyter notebook environment that requires no setup to use and runs entirely in the cloud over private virtual machines with the possibility to run with GPUs. Colaboratory notebooks are stored in Google Drive and can be shared just as you would with Google Docs or Sheets making teamwork really interesting. Colaboratory is also free to use.

The use of Matlab running in the university which give sometimes access troubles, low computing capabilities and the short learning curve in Jupyter Notebook, were all considered and we decided to make the jump to the Jupyter notebooks app.

The impressive amount of good software already implemented and freely available has allowed us to save a lot of time in our tests. We use a library, as was mentioned at the beginning of the section, to solve optimization problems called CVXPY.

One of the main objectives is to show that COB performance is better than BD. Because we are dealing with several variables in our experiments, we first fix some of them and obtain data using the aforementioned Evaluator. The theoretically bounds that we want to corroborate are

$$\begin{aligned} \text{BD} &\longrightarrow M \geq C_1 \delta^{-2} \widetilde{\mu}^{-2} S \log^2 S \log^2 \widetilde{N} \\ \text{COB} &\longrightarrow \frac{M}{\log M} \geq C_2 \delta^{-2} J \log^2 S \log \widetilde{N}. \end{aligned}$$

By default, we are generating sparse signals in canonical basis. Moreover, first we will choose S indices uniformly in $[N]$ for non-zero signal elements. In addition, we include and analyze the worst case for the sparsity of the signal, i.e., when its sparsity is non-uniformly and it start at a random position. In case that position of the sparsity is bigger than the signal size, we do a cyclic position for that.

For all experiments in this section we select $\widetilde{N} = 1680 = 2^4 \times 3 \times 5 \times 7$, which has 40 divisors 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14, 15, 16, 20, 21, 24, 28, 30, 35, 40, 42, 48, 56, 60, 70, 80, 84, 105, 112, 120, 140, 168, 210, 240, 280, 336, 420, 560, 840, 1680. We can not use the whole the divisors set, in some cases we select some of them to present the results.

The function called to get data written in files as mentioned before is:

CS_Evaluate(N, S, P, X, opt, MG, k, j, M_i, m_s, r, M_end, Print_F, Orthobasis=0)

where parameters and their options are:

N: the number of elements in each signal vector

S: the number of signal vectors to be created

P: the number of measurement matrices to test

X: S columns signals

opt: Optimizer

(1) CS_Optimizer_LASSO,

(2) CS_Optimizer_L1,

(3) CS_Optimizer_L1e

MG: Matrix generator

(1) Psi_BD_Bernoulli,

(2) Psi_COB_Bernoull,

(3) Psi_rCOB_Bernoulli

k: the sparsity, default=100

J: the number of bands, it should divide N

M_init: the initial number of samples to measure

r: the number of overlapped blocks

m_step: the M step

M_end: Final M

Print_F: Show the original and last recover signal

Othobasis: The orthobasis to transform the signal

(1) Fourier basis,

(2) Convolution Matrix,

(Other) Canonical

The following tests were made:

#	$\widetilde{N}$	**S**	**P**	**X**	**opt**	**MG**	**k**	**J**	**M_i**	**m_s**
1	1680	10	5		2	1 & 2	50, 100, 150	4, 5, 6, 7, 8, 10, 12, 14, 15, 16, 20, 21, 24, 28, 30, 35, 40, 42, 48, 56, 60, 70, 80, 84, 105, 112, 120, 140, 168, 210, 240, 280, 336, 420, 560, 840	100	10
2	1680	10	5		2	1 & 2	20, 50, 80, 100, 120, 150, 180, 200, 250	10, 20, 40	100	10
3	800, 1200, 1600, 2000, 2400	10	5		2	1 & 2	100	10, 20, 40, 80	100	10

With values at #1 experiment presented as a row in the table, we want to show the impact of the number of blocks as we will discuss in the next section 3.4.1. These experiments were executed in two different runs completely in Google Colaboratory and it took about 26 hours of execution runs. The amount of output txt files was 4294 and a total size of 17.7 MB.

The second one was prepared to show the impact of sparsity and we will present the results in section 3.4.2. Here we got 1730 txt files with a total size of 7.2 MB also running in Google Colaboratory and took almost 6 hours to run. In case of the third, it was programmed to see the impact in the size of the signal. We selected few values because choosing $\widetilde{N}$ very larger made the recover signal process spend more time, In addition, these values must have several divisors to choose J.

In all three experiments we include differences of minimal number of measurements $\widetilde{M}$ between COB and BD to compare the performance. Those differences are taken in account a normalized J, i.e, taking the same number of nonzero elements for both matrices. It will show that COB matrices, over the same conditions, need less measurements to recover the signal that BD matrices as expected. Furthermore, we include $\frac{minM_{BD}-minM_{COB}}{minM_{BD}}$, i.e, a ratio of improvement with the minimal number of measurements differences between BD and COB. It helps us assert that using COB matrices to sense a signal, and over the similar conditions on the matrices, COB will improve around 20% to BD matrices.

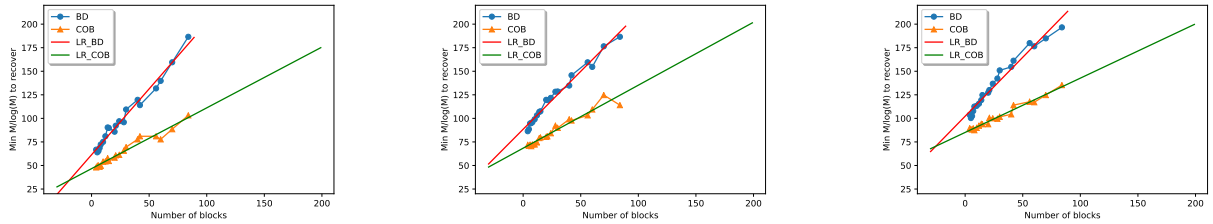
3.4 Performance of COB with uniform random signal

3.4.1 The Impact of the Number of Blocks

If we consider the theoretical result, the number of blocks should affect the number of measurements about a linear factor. This was confirmed in the experiments and can be seen in figure 3.2. Note that COB is improving the performance of BD in the number of measurements. If we consider the slope as a parameter to show that, we can claim that the performance of COB will be about 20% better than BD since one is twice the other. This improvement will also be visualized in the next section.

Validating J factor bound in Theorem 13

According to the lineal regression in figures 3.2, it is easy to see that we are validating the linear factor of J in the number of measurements needed to recover signals. Another observation is about the y-intercept is not zero. This means that our bound is working without constant values that do not depend of $\widetilde{N}, S$ or J. This is because we have discarded values that do not dominate in the demonstration leaving behind linear or logarithmic factors of only J, S, and $\widetilde{N}$.



(a) $S = 50$ (b) $S = 100$ (c) $S = 150$

Figure 3.2: The minimum $M/\log(M)$ versus J using three different sparsity

The linear equations in table 3.4, give us the possibility to assert that the slope of BD is twice the slope of COB, then an improvement of 20% can be archived if COB is used.

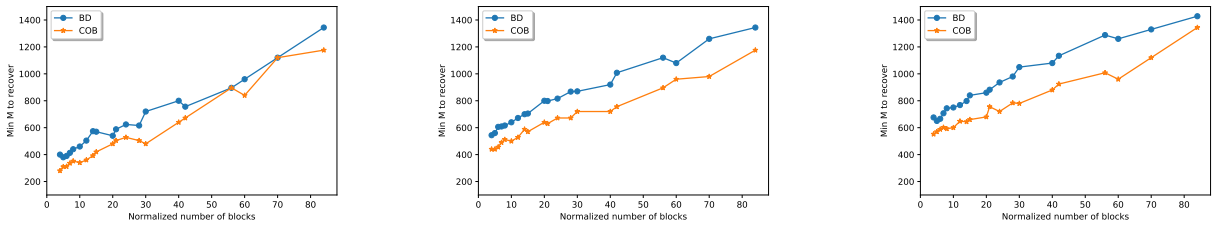
Sparsity	**LR equations**	R^2**-Coef. of determination**
50	$y_{BD} = 1.399x_{BD} + 61.349$	0.9778
	$y_{COB} = 0.6451x_{COB} + 46.6383$	0.9575
100	$y_{BD} = 1.229x_{BD} + 88.453$	0.9779
	$y_{COB} = 0.6684x_{COB} + 68.3714$	0.9434
150	$y_{BD} = 1.250x_{BD} + 102.286$	0.9661
	$y_{COB} = 0.5760x_{COB} + 85.0773$	0.9767

Table 3.4: Linear regression BD vs COB using sparsity

COB vs. BD

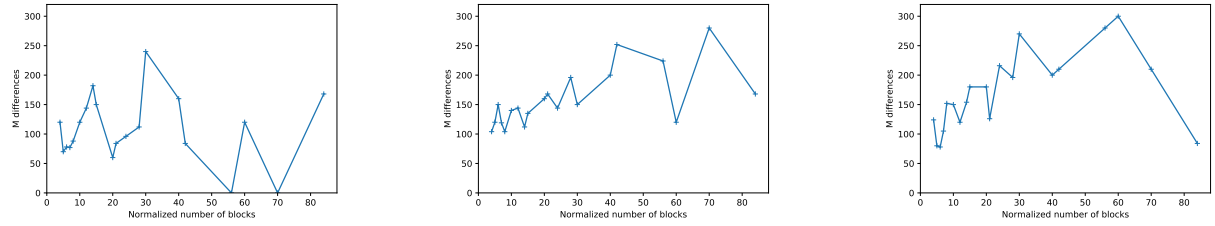
The amount of measurements needed to recover the signal using COB is less than BD, our experiments are showing this claim clearly. For example, in figure 3.3 we normalize the number of nonzero entries in matrix COB to be according to BD. We show three different sparsity in the signal, i.e., for 50, 100, 150, and when the number of blocks is increasing, the numbers of measurements needed is always less in each different sparsity figure.

We include figures 3.4 and 3.5 where we show the differences and normalized difference respectively, between the number of measurements in *average* needed to recover the signal for different blocks in BD and COB. In 3.4 we can see that these differences seem constant when the number of blocks increases. In addition, it can also deduced clearly from figure 3.5 that the improvement is about 20%.



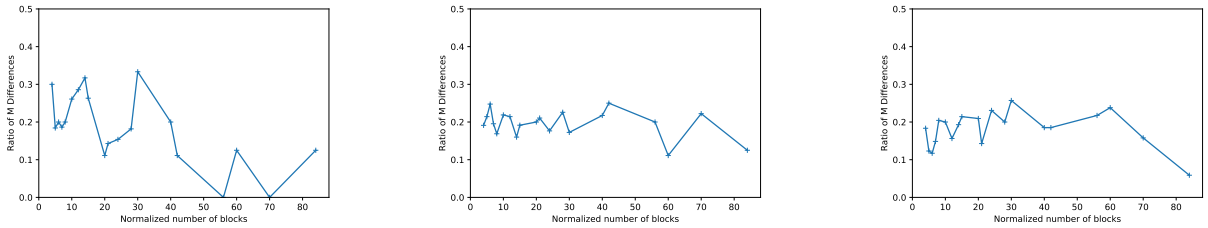
(a) $k = 50$ (b) $k = 100$ (c) $k = 150$

Figure 3.3: The minimum M versus normalized J using three different sparsity



(a) $k = 50$ (b) $k = 100$ (c) $k = 150$

Figure 3.4: M differences versus normalized J using three different sparsity



(a) $k = 50$ (b) $k = 100$ (c) $k = 150$

Figure 3.5: M differences ratio versus normalized J using three different sparsity

3.4.2 The Impact of The Sparsity

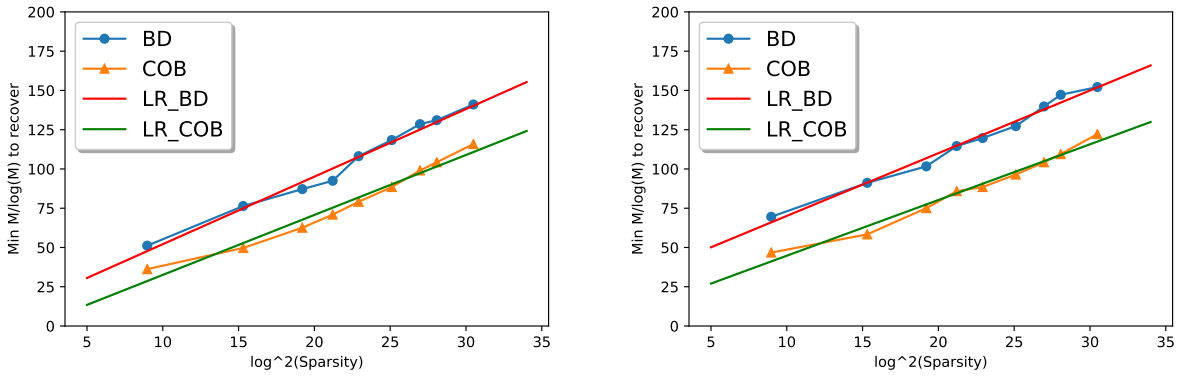
To show the impact of the sparsity we select two fixed block sizes and let the number of nonzero elements on testing signals variate. The sparsity range in respect to the signal size $\widetilde{N} = 1680$ was $[20, 250]$, i.e., about between 1% and 15% of signal sparsity.

Validating the S factor in Theorem 13

As found in figure 3.6, after take the $\log^2$ over the all S values tested, we confirm the factor in $\widetilde{M}$ bound on our COB result. Although the two matrices do not have the same number of nonzero elements, we can again claim the better performance of COB over BD. The linear regressions are shown to certify the logarithm factor as well the lower slope (see table 3.5) for COB over BD.

Blocks	**LR equations**	R^2-Coef. of determination
10	$y_{BD} = 4.3014x_{BD} + 9.0715$	0.9838
	$y_{COB} = 3.8185x_{COB} - 5.6479$	0.9710
20	$y_{BD} = 3.9934x_{BD} + 30.1582$	0.9862
	$y_{COB} = 3.5476x_{COB} + 9.2784$	0.9791

Table 3.5: Linear regression

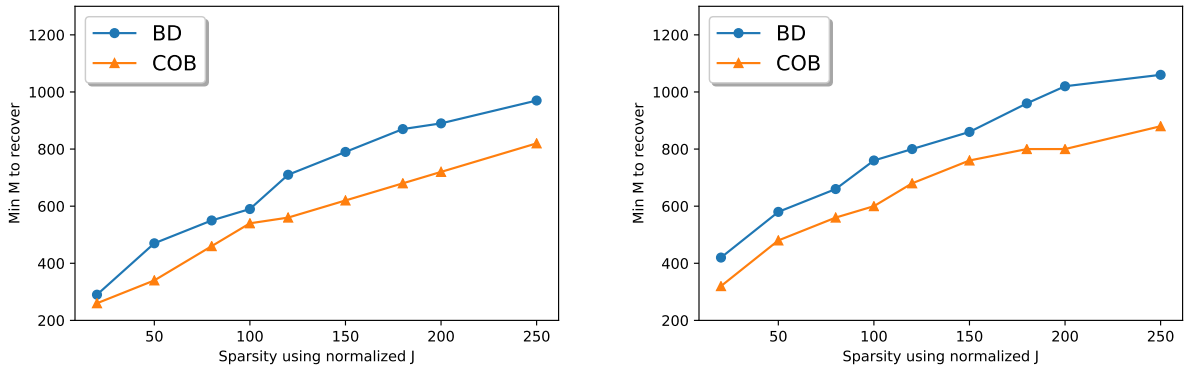


(a) $J = 10$ (b) $J = 20$

Figure 3.6: The minimum $M/\log(M)$ versus Sparsity using two different number of blocks

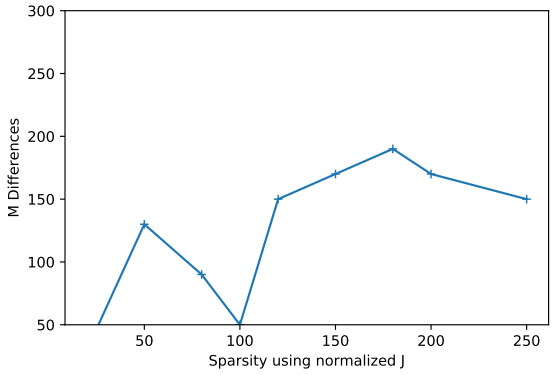
COB vs. BD

Figures 3.7, 3.8 and 3.9 use the same number of nonzero elements over both matrices and it can confirm now that COB improves BD. To quantify this improvement, a ratio was graphed in 3.9 which makes it clear again that the improvement is about 20%.

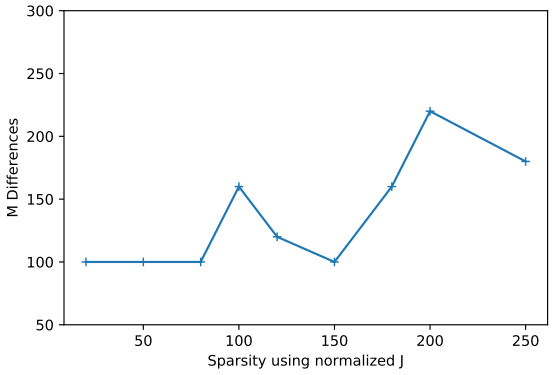


(a) $J = 10$ (b) $J = 20$

Figure 3.7: The minimum $M/\log(M)$ versus normalized J using two different number of blocks

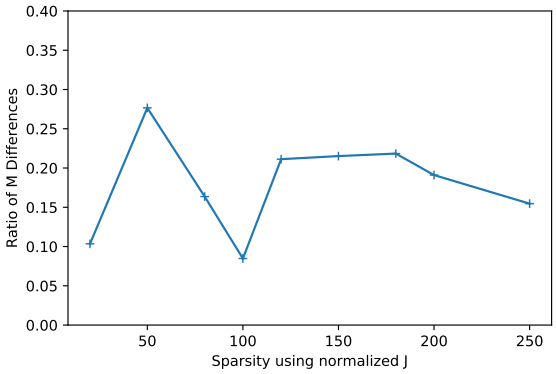


(a) $J' = 10$

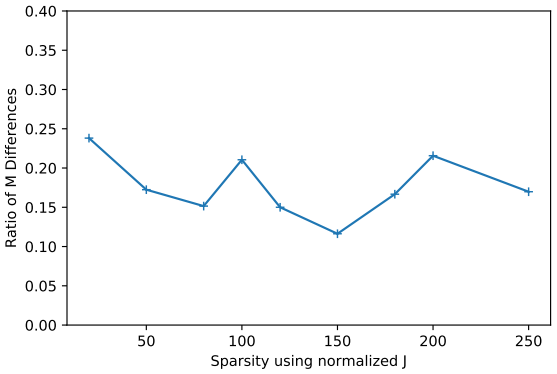


(b) $J' = 20$

Figure 3.8: M differences versus Sparsity using two different normalized J



(a) Normalized $J = 10$



(b) Normalized $J = 20$

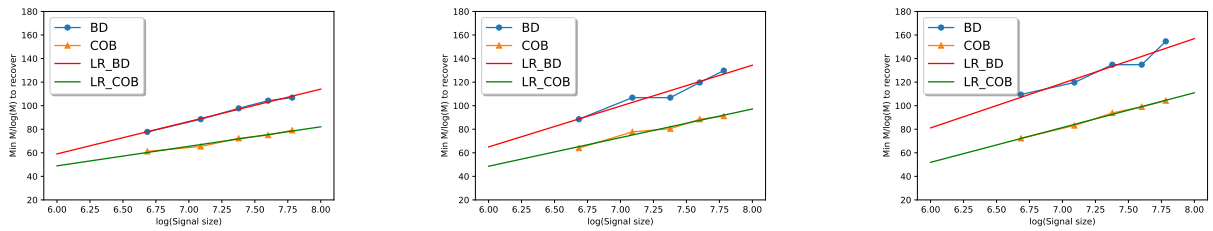
Figure 3.9: M differences ratio versus Sparsity using two different normalized J

3.4.3 The Impact of Signal Size

In this case, for three fixed block sizes, we variate the signal size to validate the bound. The style of previous sections is preserved. First we show the shape of the factor and next we normalized the blocks on both matrices to ratify that COB performs better than BD.

Validating the $\widetilde{N}$ factor on Theorem 13

From theoretical results we know that the influence of $\widetilde{N}$ is about a logarithmic factor. Then, after take the log to each test $\widetilde{N}$ value over normalized J, the figure 3.10 validate it. The linear regression over both matrices and using the slope help to visualize the improvements.



(a) $J = 10$ (b) $J = 20$ (c) $J = 40$

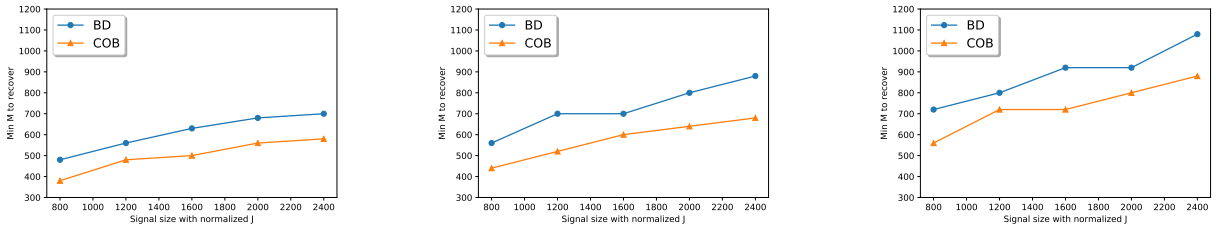
Figure 3.10: The minimum $M/\log(M)$ vs the log(signal size) using three different number of blocks

Signal size	**LR equations**	R^2-Coef. of determination
10	$y_{BD} = 27.5158x_{BD} - 106.0469$	0.9932
	$y_{COB} = 16.5817x_{COB} - 50.5781$	0.9797
20	$y_{BD} = 34.6989x_{BD} - 143.2206$	0.9369
	$y_{COB} = 24.3212x_{COB} - 973580$	0.9718
40	$y_{BD} = 37.8929x_{BD} - 146.2236$	0.9159
	$y_{COB} = 29.5358x_{COB} - 125.3190$	0.9962

Table 3.6: Linear regression impact of signal size

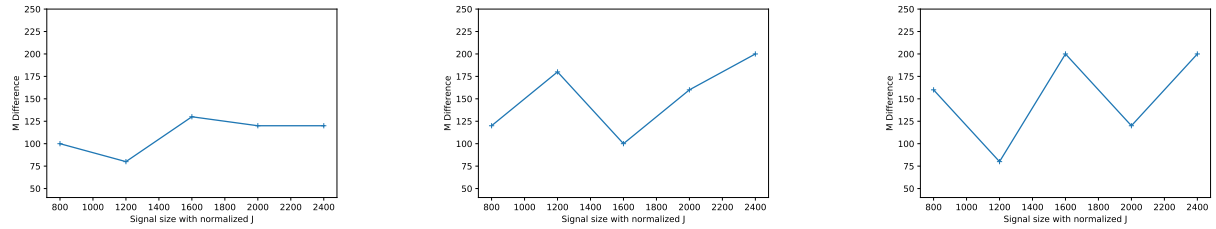
COB vs. BD

As we did with the factors J and S, we now normalize J to let both matrices compete in equal circumstances. Figures 3.11, 3.12 and 3.13 will shows clearly the improvements. If we want to see the approximate 20% of improvements again, check graph 3.13.



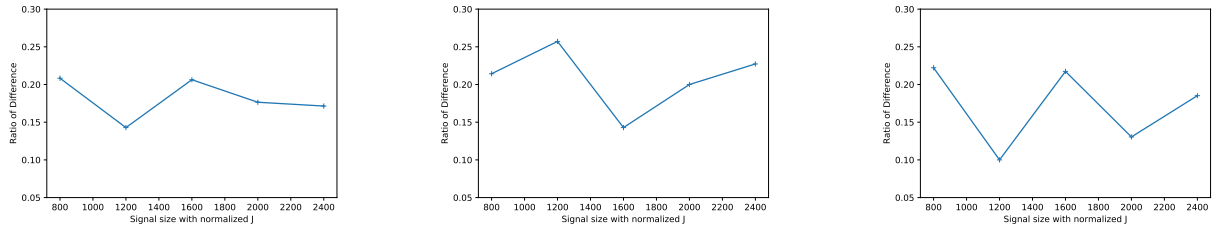
(a) $J = 10$ (b) $J = 20$ (c) $J = 40$

Figure 3.11: The minimum $M/\log(M)$ vs $\widetilde{N}$ using three different normalized J



(a) $J = 10$ (b) $J = 20$ (c) $J = 40$

Figure 3.12: Difference between $M/\log(M)$ versus $\widetilde{N}$ using three different normalized J.



(a) $J = 10$ (b) $J = 20$ (c) $J = 40$

Figure 3.13: Difference ratio $M/\log(M)$ versus $\widetilde{N}$ using three different normalized J

3.5 Performance of COB with random bursty signal

All signals created previously have a independent and identically distributed variables elements on its sparsity. For signal size $\widetilde{N} = 1680$, as we did before, we do here experiments with 50, 100, and 150 nonzero elements, i.e., the signal sparsity. The worst case for the random signal is that all its non-zero elements are concentrated starting in some unknown random position. We are also experimented with this type of signals and the results show that although more measurements of the signal are needed, we still have better results with COB than with BD. For small number of blocks J, we observe same pattern as before, a linear increase, see figure 3.15 and figure 3.16 when both matrices have the same nonzero values quantity. For large J's, the number of measurements are converging to $\widetilde{N}$ in less intensity for COB. For very large J values, the signal cannot be even reconstructed.

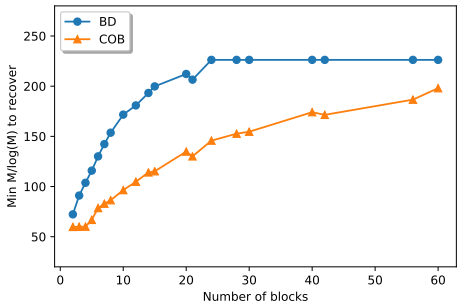


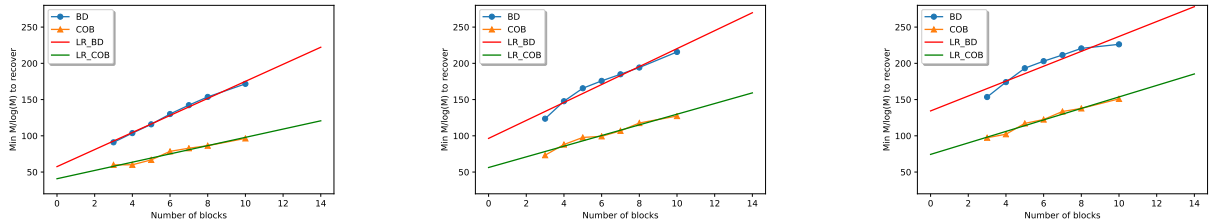
Figure 3.14: The minimum $M/\log(M)$ vs J worst signal case

3.5.1 The Impact of the Number of Blocks

For this case where random signal is concentrated, and consider the theoretical result as we did before, the number of blocks should affect the number of measurements about a linear factor. This was confirmed again with numerical experiments and is presented below.

Validating the J factor on Theorem 13

The random sparsity concentrate of the signal is not affected the results and still COB has advantage over BD for small J values. We can observe in graph 3.15 that, using same values as before in 3.2 were the sparsity is identically distributed, similar improvements over small J values it keep getting. This is observed in the slope in the linear regressions which is twice for BD over COB.



(a) $S = 50$ (b) $S = 100$ (c) $S = 150$

Figure 3.15: The minimum $M/\log(M)$ vs J worst signal case

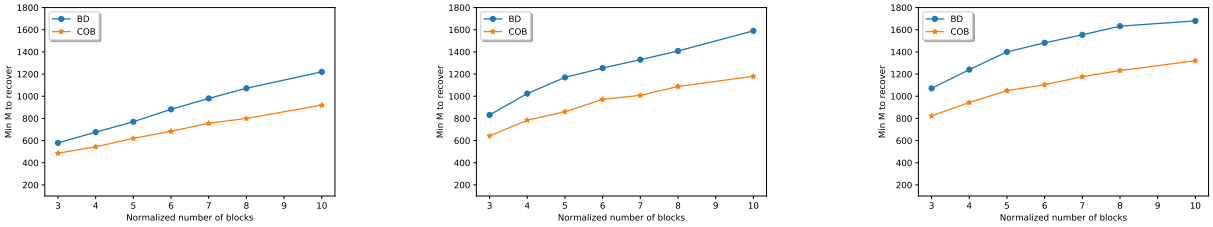
The linear regression values Table 3.7 shows that in terms of quantity, the slope is close to double in BD that in COB. As we asserted before, we can consider a 20% of improvements with COB that it is confirm in graph 3.18 where ratio differences were drawn.

Sparsity	**LR equations**	R^2**-Coef. of determination**
50	$y_{BD} = 11.7654x_{BD} + 57.4777$	0.9938
	$y_{COB} = 5.7084x_{COB} + 40.7791$	0.9640
100	$y_{BD} = 12.3845x_{BD} + 96.4295$	0.9628
	$y_{COB} = 7.3556x_{COB} + 56.2486$	0.9652
150	$y_{BD} = 15.1716_{BD} + 107.4568$	0.9502
	$y_{COB} = 8.8805x_{COB} + 69.3582$	0.9853

Table 3.7: Linear regression BD vs COB worst case

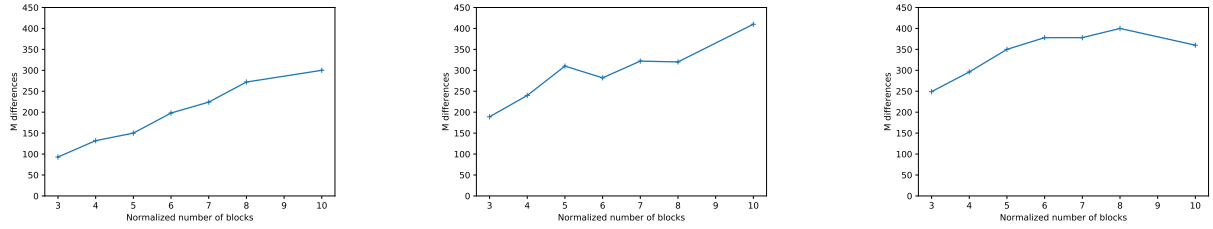
3.5.2 COB vs. BD

If we consider normalized J, i.e., made both BD and COB matrix with equal nonzero entries, a concentrated signal that represent the worst case scenario to represent the sensors data, we can observed better performance with COB, see figure 3.16. Reaffirming improvements of 20% with COB again can be observed through the figure 3.18.



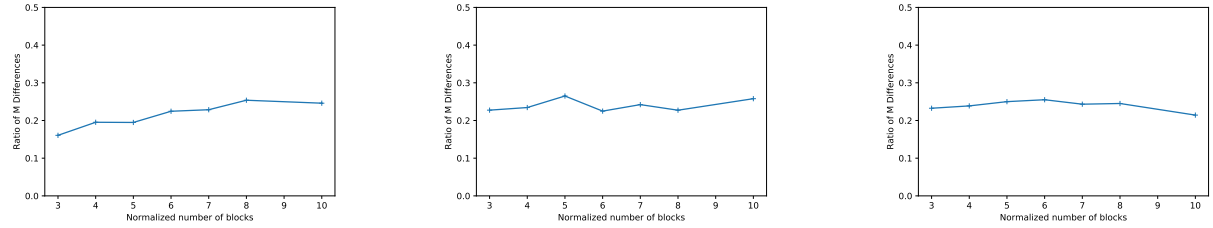
(a) $S = 50$ (b) $S = 100$ (c) $S = 150$

Figure 3.16: The minimum M versus normalized J worst signal case



(a) $S = 50$ (b) $S = 100$ (c) $S = 150$

Figure 3.17: M differences versus normalized J worst signal case



(a) $S = 50$ (b) $S = 100$ (c) $S = 150$

Figure 3.18: M differences ratio versus normalized J worst signal case

Chapter 4

The COB generalization

In COB we overlap one block and considering the BD result, it was very natural think that we can get the RIP and improve the number of measurements needed to recover the signal using COB matrices. We show that in chapter 3. Now we can think also naturally that overlapping r blocks we should improve more but with the disadvantage that the calculations made to recover the signal would take longer. Although under certain circumstances it might be better to pay the cost. In this chapter, we will generalize the measurement matrix COB, now called rCOB, and also prove that this matrix satisfies RIP. To extend our result we shows over which circumstances rCOB performs good, we believe that it should work better when the number of blocks increases. When this happens BD and COB presented some instability. Numerical result are likewise include at the end of the chapter.

4.1 The rCOB Matrix

Similarly to COB's definition, let $\Phi_{j1}, \Phi_{j2}, \ldots, \Phi_{j(r+1)} \in \mathbb{C}^{M\times N}$ for $j \in [J]$ be matrices populated with i.i.d. sub-Gaussian random variables having mean zero, standard deviation

$1/\sqrt{(r+1)M}$ and sub-Gaussian norm τ. Now the rectangular blocks are formed by $M \times (r+1)N$ matrices $\Phi_j = [\Phi_{j1}\ \Phi_{j2} \cdots \Phi_{j(r+1)}]$, i.e with the concatenation of Φ_{jk} for $i \in [J]$ and $k \in [r+1]$.

The generalization is summarize in the following table:

r**COB**				
BD		**COB**		**Full**
↓		$r = 1$		↓
0	$\leq$	r	$\leq$	$J-1$

Table 4.1: rCOB generalization range

and its structure will be similar to

$$\Psi = \begin{bmatrix} [\Phi_{11} & \Phi_{12} & \cdots & \Phi_{1(r+1)}] & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & [\Phi_{21} & \Phi_{22} & \cdots & \Phi_{2(r+1)}] & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & [\Phi_{31} & \Phi_{32} & \cdots & \Phi_{3(r+1)}] & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \ddots & \cdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & [\Phi_{(J-3)1} & \Phi_{(J-3)2} & \cdots & \Phi_{(J-3)(r+1)}] \\ \Phi_{(J-2)(r+1)}] & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & [\Phi_{(J-2)1} & \cdots & \Phi_{(J-2)r} \\ \Phi_{(J-1)r} & \Phi_{(J-1)(r+1)}] & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & [\Phi_{(J-1)1} & \cdots \\ \cdots & \Phi_{Jr} & \Phi_{J(r+1)}] & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & [\Phi_{J1} \end{bmatrix} \tag{4.1}$$

The symbol $\mathbf{0}$ represent block of zeros, in this case the matrix could be unstructured or structured. The number of non-zero is increasing when r go to $J-1$. On figure 4.1 we can see the shape when $r = 3$ and for $N = 20, M = 10$ and $J = 10$, of course its size is 200×100. There, we add BD and COB matrices to compare and visualize better the form of the matrices in which we have focused in this thesis. The color black represent the blocks Φ_{jk} and white part the zeros.

In order to better see the impact of r on the shape of the matrix and how it changes its quantity of non-zero elements we have added the Figure 4.2, where $r = 8$ for the same values

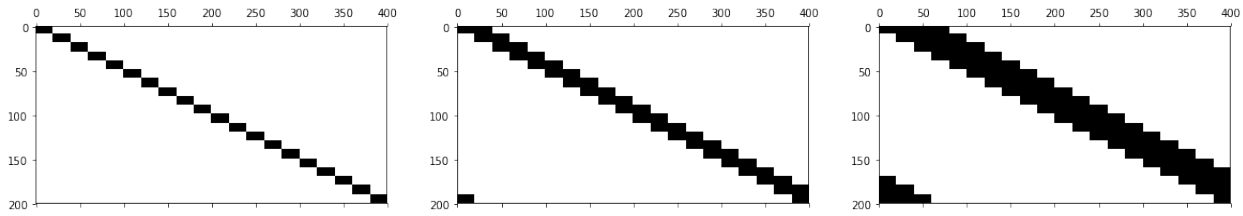


Figure 4.1: BD, COB and 3COB for $N = 20, M = 10$ and $J = 10$

of the Figure 4.1.

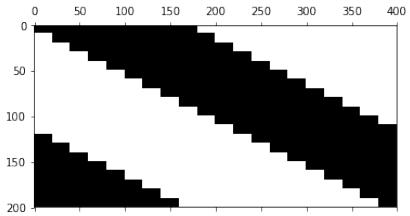


Figure 4.2: 8COB for $N = 20, M = 10$ and $J = 10$

The following equations describe for BD, COB and rCOB its non-zero ratio elements quantity, i.e.,

$$\text{BD}: \qquad \frac{JMN}{\widetilde{M}\widetilde{N}} = \frac{1}{J}, \tag{4.2}$$

$$\text{COB}: \qquad \frac{2JMN}{\widetilde{M}\widetilde{N}} = \frac{2}{J}, \tag{4.3}$$

$$r\text{COB}: \qquad \frac{(r+1)JMN}{\widetilde{M}\widetilde{N}} = \frac{r+1}{J}. \tag{4.4}$$

It is clear that the matrix sparsity is related with J, when it grows the matrix will be more sparse and the calculations will be much faster for recover algorithm. Although, as we have seen the number of samples M depends linearly on J, then increasing J will also increase M and we lost the main property of CS, sub-sample the signal. For each application we have to be carefully choosing the number of blocks.

4.2 RIP for rCOB

The process to follow here is similar to COB. The steps are:

1. **RIC**: write RIC as $\delta_S = \sup_{\alpha\in\Omega_S} \left| \|\Psi \cdot x(\alpha)\|_2^2 - 1 \right|$ and check that $\mathbb{E}\{\|\Psi \cdot x(\alpha)\|_2^2\} = 1$ verifying that $\mathbb{E}\{\Psi^* \cdot \Psi\} = I$.
2. **Transformation**: build $X_j(\alpha), A(\alpha)$, and $\mathcal{A}$. Check $\|\Psi \cdot x(\alpha)\|_2 =_{\text{i.d.}} \|A(\alpha) \cdot \varepsilon\|_2$.
3. **Calculation 1**: values $d_F(\mathcal{A})$ and $d_2(\mathcal{A})$.
4. **Calculation 2**: value $\gamma_2(\mathcal{A}, \|\cdot\|_2)$
5. **Applying Theorem**: put everything together to get $\widetilde{M}$, i.e., the number of measurements/rows for Ψ and shows that $\delta_S < \delta$, i.e., get RIP for Ψ'.

We will present in next section the main theorem for rCOB without prove. The rest of the section will shows the steps to prove it.

4.2.1 The measurement Theorem

The following theorem supports our *novel framework* for structures matrices. It generalizes the BD matrix and our COB matrix, when $r = 0$ it reach BD, for $r = 1$ is COB and for $r = J - 1$ will be a full structure matrix.

Theorem 15. *Let U denote an orthobasis for $\mathbb{C}^{\widetilde{N}}$, $\Psi \in \mathbb{C}^{\widetilde{M}\times\widetilde{N}}$ a J blocks rCOB matrix as defined above* (4.1) *and set*

$$\widetilde{\mu} := \min\left\{\sqrt{J/(rS+S)}, \mu(U)\right\}, \qquad \textit{where} \qquad \mu(U) = \sqrt{\widetilde{N}} \max_{p,q\in[\widetilde{N}]} |U(p,q)|.$$

If $S \gtrsim 1$ *and*

$$\frac{\widetilde{M}}{\log \widetilde{M}} \gtrsim \delta^{-2} \cdot \widetilde{\mu}^2 \cdot S \cdot \log^2 S \cdot \log \widetilde{N},$$

then $\delta_S(\Psi, U) \leq \delta < 1$*, except with a very low probability of at most* $O\left(\widetilde{N}^{-\log \widetilde{M} \log^2 S}\right)$.

Once again we make the case for the canonical basis where the number of measurements will be linear now with value $\frac{J}{r+1}$.

Corollary 2. *Let* $x \in \mathbb{C}^{\widetilde{N}}$ *be a signal, and* $\Psi \in \mathbb{C}^{\widetilde{M} \times \widetilde{N}}$ *a* J *blocks* r*COB measurement matrix as defined above* (4.1)*. If* $S \gtrsim 1$ *and*

$$\frac{\widetilde{M}}{\log \widetilde{M}} \gtrsim \delta^{-2} \cdot \frac{J}{r+1} \cdot \log^2 S \cdot \log \widetilde{N}, \tag{4.5}$$

then $\delta_S \leq \delta < 1$*, except with a very low probability of at most* $O\left(\widetilde{N}^{-\log \widetilde{M} \log^2 S}\right)$.

4.2.2 RIC for rCOB

We can represent δ_S for rCOB in the same way that we did for COB in (3.3), i.e.,

$$\delta_S = \sup_{\alpha \in \Omega_S} \left| \|\Psi \cdot x(\alpha)\|_2^2 - 1 \right|. \tag{4.6}$$

To check that $\mathbb{E}\{\Psi^* \cdot \Psi\} = I$, note that $\Psi^*\Psi$ involved $r+1$ unstructured $\phi_{M \times N}$ matrices, because we choose standard deviation $1/\sqrt{(r+1)M}$ in each ϕ (see definition 4.1), using a similar prove in Lemma 3, the result is also getting for rCOB.

4.2.3 The transformation

As in COB, similar transformation is required to get *index set of the random process* for rCOB. For $\alpha \in \mathbb{C}^{\widetilde{N}}$, set $x(\alpha) = U\alpha$ and define $x_j(\alpha)$ as in COB such that

$$x(\alpha) = [x_1(\alpha)^T, x_2(\alpha)^T, \ldots, x_J(\alpha)^T]^T.$$

To define the sub-matrix $X_j \in \mathbb{C}^{M\times(r+1)MN}$ with $j \in [J]$, let us use $[(x_k^*(\alpha))_j]$ as a notation for a row vector to all $k \in \mathrm{idx}_j$ where idx_j is define later. Therefore,

$$X_j(\alpha) = \begin{bmatrix} [(x_k^*(\alpha))_j] & & & \\ & [(x_k^*(\alpha))_j] & & \\ & & \ddots & \\ & & & [(x_k^*(\alpha))_j] \end{bmatrix}.$$

To define the *ordered list of index* idx_j with fixed size $r+1$ as follow

$$\mathrm{idx}_j = \begin{cases} [j, j+1, \cdots, j+r] & \text{if } j+r \le J \\ \underbrace{[j, j+1, \cdots, r+j]}_{\text{size } J-j+1} \cup \underbrace{[1, 2, \cdots, r+j-J]}_{\text{size } r+j-J} & \text{if } r+j \ge J \end{cases}. \tag{4.7}$$

Note that (4.7) is consistent with (3.10) in COB, where $r = 1$ and with BD if $r = 0$.

Therefore, a matrix A here is similar as in COB, but could has to many columns as r grows.

It has the following form:

$$A(\alpha) = \frac{1}{\sqrt{(r+1)M}} \begin{bmatrix} X_1(\alpha) & & & \\ & X_2(\alpha) & & \\ & & \ddots & \\ & & & X_J(\alpha) \end{bmatrix}_{\widetilde{M}\times(r+1)JMN} \tag{4.8}$$

The index set of the random process will be

$$\mathcal{A} := \{A(\alpha) : \alpha \in \Omega_S\}. \tag{4.9}$$

Some notes about the transformation:

1. The size of matrix and vector multiplication corresponded, i.e.,

$$y_{\widetilde{M}\times 1} = \Psi_{\widetilde{M}\times\widetilde{N}} \cdot x_{\widetilde{N}\times 1}(\alpha)$$
$$y'_{\widetilde{M}\times 1} = A(\alpha)_{\widetilde{M}\times(r+1)M\widetilde{N}} \cdot \varepsilon_{(r+1)M\widetilde{N}\times 1}$$

   Actually, it can be proved that $y = y'$ or equally

$$\|\Psi \cdot x(\alpha)\|_2 =_{\text{i.d.}} \|A(\alpha) \cdot \varepsilon\|_2 .$$

2. The result of $A(\alpha) \cdot A^*(\alpha)$ is M-square diagonal matrix for $j = 1, \ldots, J$ with the following structure

$$\frac{1}{(r+1)M} \begin{bmatrix} \sum_{k\in \text{ idx}_1} \|x_k(\alpha)\|_2^2 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \sum_{k\in \text{ idx}_J} \|x_k(\alpha)\|_2^2 \end{bmatrix} \tag{4.10}$$

This will be helpful later to calculate spectral norm of $A(\alpha)$.

4.2.4 Remaining steps

The two values $d_F(\mathcal{A})$ and $d_2(\mathcal{A})$ for rCOB are also similar to COB. For the first one, check that A is diagonal composed of single-row blocks with elements from $x(\alpha)$. Each row has different "consecutively" sub-vectors $x_j(\alpha)$, and for all j, each of them will appear in $r+1$ blocks. In addition, each sub-matrix X_j repeat M times a row, therefore

$$\begin{aligned} d_F(\mathcal{A}) &= \sup_{A(\alpha)\in\mathcal{A}} \|A(\alpha)\|_F \\ &= \sup_{\alpha\in\Omega_S} \frac{1}{\sqrt{(r+1)M}} \sqrt{(r+1)M\|x_1(\alpha)\|_2^2 + \cdots + (r+1)M\|x_J(\alpha)\|_2^2} \\ &= \sup_{\alpha\in\Omega_S} \|x(\alpha)\|_2 \\ &= \sup_{\alpha\in\Omega_S} \|U\alpha\|_2 = \sup_{\alpha\in\Omega_S} \|\alpha\|_2 = 1. \end{aligned} \tag{4.11}$$

This result and the next one let us to think about the transformation and its properties. Off course, the transformation matrix has repeated elements from x, therefore will be natural that $d_F(\mathcal{A})$ and $d_2(\mathcal{A})$ has values related with x. Following this, for $d_2(\mathcal{A})$ we can claim that

$$\|A(\alpha)\|_2 \leq \sqrt{\frac{S}{\widetilde{M}}} \min\left(\mu, \sqrt{\frac{J}{r+1}}\right) = \widetilde{\mu}\sqrt{\frac{S}{\widetilde{M}}},$$

where we used same deduction of (3.17) and property (4.10). Therefore,

$$d_2(\mathcal{A}) = \sup_{A(\alpha)\in\mathcal{A}} \|A(\alpha)\|_2 \leq \widetilde{\mu}\sqrt{\frac{S}{\widetilde{M}}}, \tag{4.12}$$

where $\widetilde{\mu} = \min\left(\mu, \sqrt{\frac{J}{(r+1)S}}\right)$ and $\mu(U) = \sqrt{\widetilde{N}}\max_{p,q\in[\widetilde{N}]}|U(p,q)|$.

Here we shows the most difficult calculation to be able to apply the chaos theorem. But the deduction made in (3.21) applying the Covering Lemma 5 can be extended to this general matrix rCOB, we present without prove that

$$\gamma_2(\mathcal{A}, \|\cdot\|) \lesssim \sqrt{S} \int_0^\infty \log^{\frac{1}{2}} \mathcal{N}\left(\frac{\Omega_S}{\sqrt{S}}, \|\cdot\|_A, v\right) dv \lesssim \widetilde{\mu}\sqrt{\frac{S}{\widetilde{M}}} \log S \sqrt{\log \widetilde{M} \log \widetilde{N}}. \tag{4.13}$$

From (4.11), (4.12) and (4.13) and applying Theorem 14 we prove the Theorem 15. □

4.3 Numerical Experiments

In this section, like we did before with COB, we will support our theoretical results with numerical experiments. First we will validate the theoretical analysis, and next shows under what conditions using rCOB offers better performance than using BD. The next table shows the parameters passed to the solver function 9. For all r values, we run the solver for each J in the table. We left $\widetilde{N}$ and k always constant.

#	**N**	**S**	**P**	**X**	**opt**	**MG**	**k**	**J**	**M_i**	**r**	**m_s**
1	1680	5	5		2	3	100	4, 5, 6, 7, 8, 10, 12, 14, 15, 16, 20, 21, 24, 28, 30, 35, 40, 42, 48, 56, 60, 70, 80	380	0, 1, 2, 3, 4, 5, 8, 10, 15, 20, 25	10

Table 4.2: Values to test rCOB

The implementation code function is presented in algorithm 10. To build rCOB matrices is done through the function 10, a similar idea from COB is used but now adding $J-1$ BD matrices rolling the second one to a factor of $i \cdot n$ each time before added. Again there we have to normalize before returning.

Function 10 Psi_rCOB_Bernoulli(M, N, J, r)

Input: $M, N \mod J = 0 \quad \& \quad 0 \leq r \leq J$
Output: $A_{M \times N}$
1: $n = N/J$
2: **for** $i = 1 : r + 1$ **do**
3: $\quad A \leftarrow A + \text{Roll}(\text{Psi\_BD\_Bernoulli}(M, N, J), i \cdot n)$
4: **end for**
5: $x \leftarrow \sqrt{r + 1}$
6: **return** A/x

To tests all this values on table 4.2 took 18 hours running and generate more than 1000 text files. With this experiments we expect to see that rCOB could start to recover signals over the factor range presented in equation 4.5. From that equation, the number of blocks do a linearly with J over the number of overlapped blocks, i.e., $r + 1$.

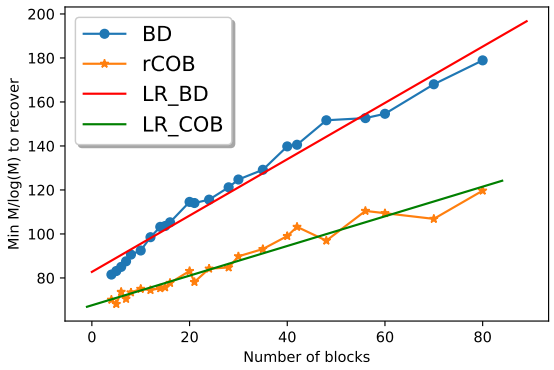
4.3.1 Validating the theoretical results

On Corollary 2, the only different factor over the number of measurements is J. Therefore we prepare experiments using rCOB matrix over several block (J) varying the parameter r. For r-values 0 and 1 the matrix is BD and COB respectively. On figures 4.3 to 4.4 we present the results.

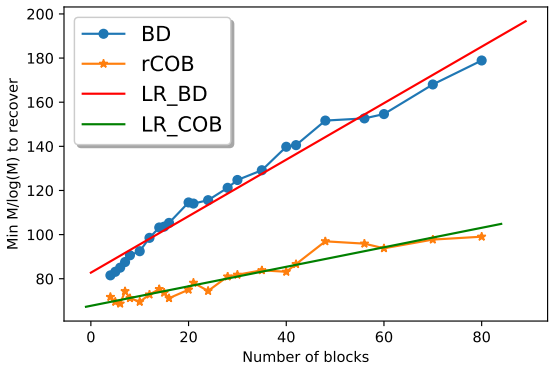
We observe that rCOB is consistently recovering signals linearly when the number of blocks is increasing. But, graphs shows clearly that the linear factor it is much lower than in BD as those blocks are influence by parameter r.

4.3.2 BD vs rCOB

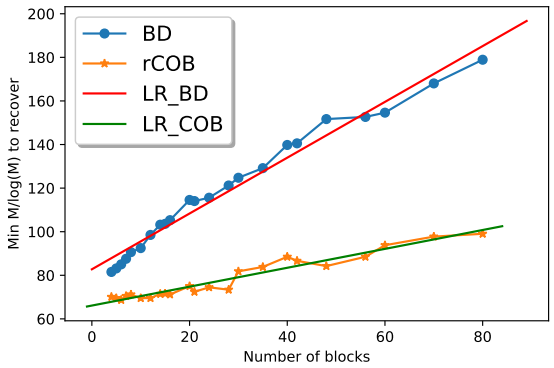
If we consider normalize J, i.e., use the same number of nonzero values in BD and rCOB for each r, still we can observe that rCOB is using less measurements as BD. See figure 4.5 to one example.



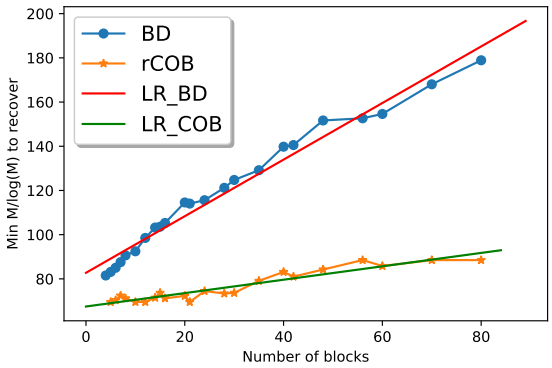
(a) $r = 1$



(b) $r = 2$

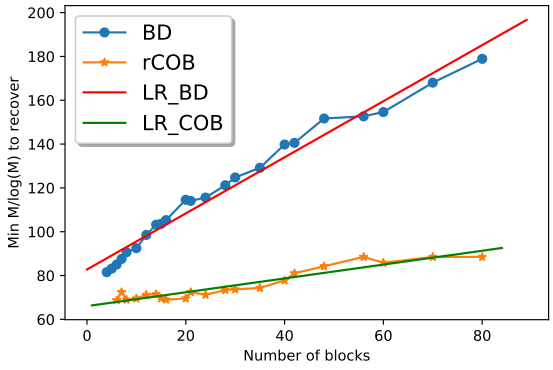


(c) $r = 3$

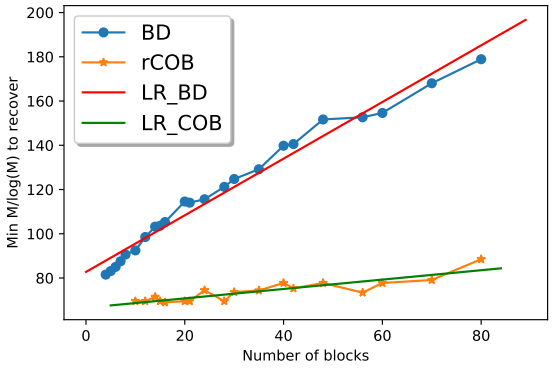


(d) $r = 4$

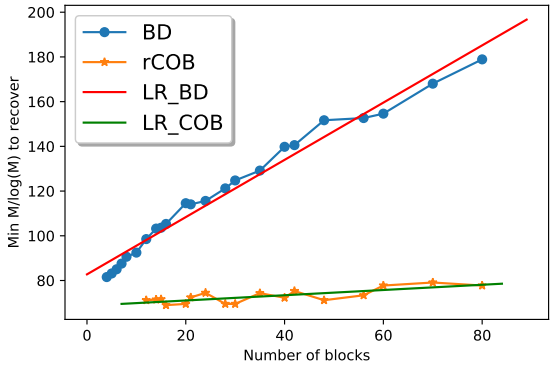
Figure 4.3: The minimum $M/\log(M)$ versus J for rCOB with $r = 1, 2, 3, 4$



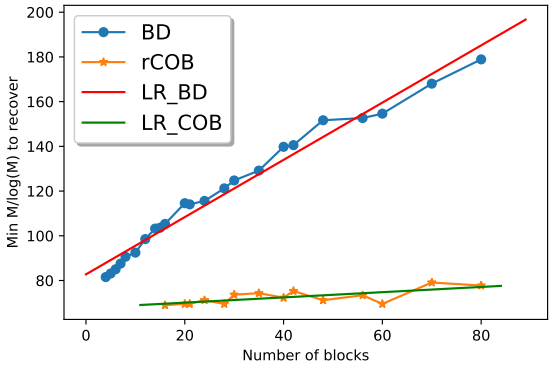
(a) $r = 5$



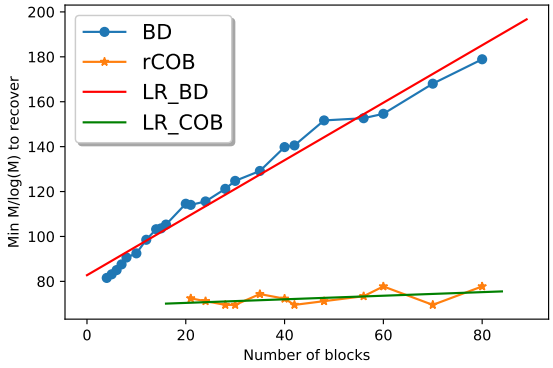
(b) $r = 8$



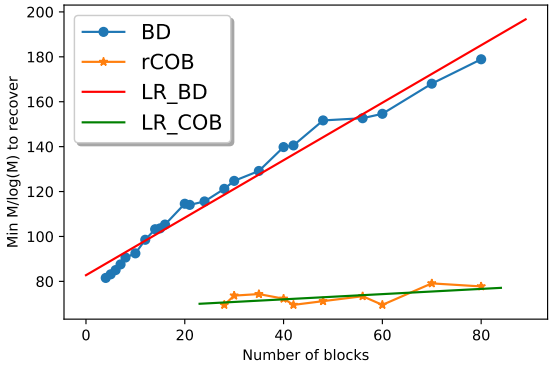
(c) $r = 10$



(d) $r = 15$

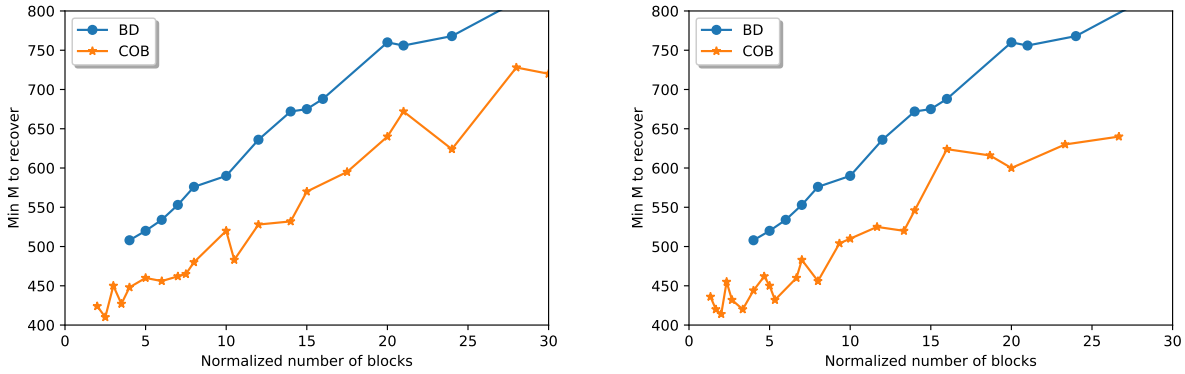


(e) $r = 20$



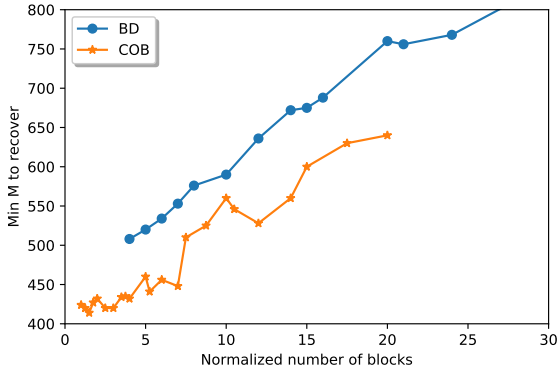
(f) $r = 25$

Figure 4.4: The minimum $M/\log(M)$ versus J for rCOB with $r = 5, 8, 10, 15, 20, 25$

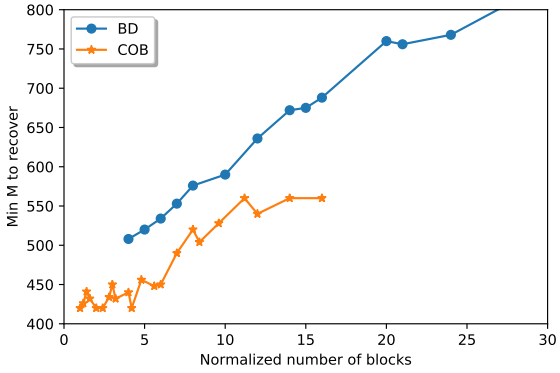


(a) $r = 1$

(b) $r = 2$

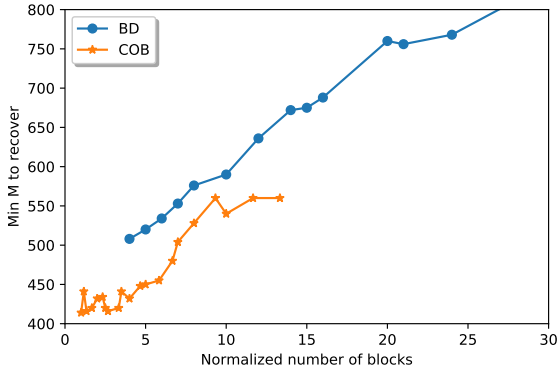


(c) $r = 3$

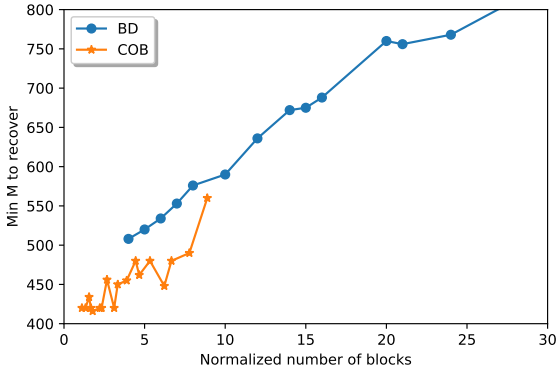


(d) $r = 4$

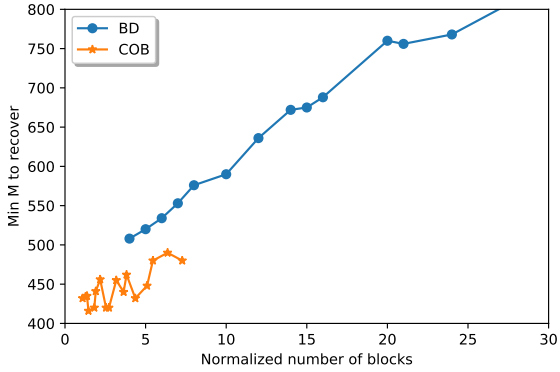
Figure 4.5: rCOB vs BD over normalized J



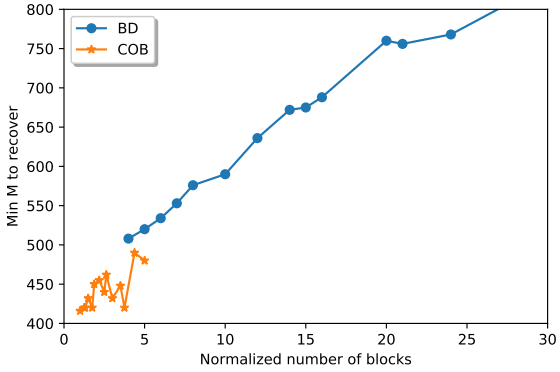
(a) $r = 5$



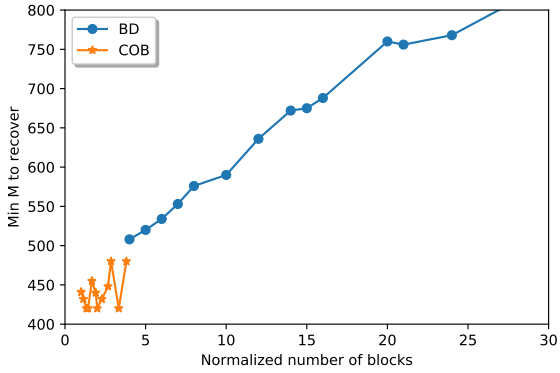
(b) $r = 8$



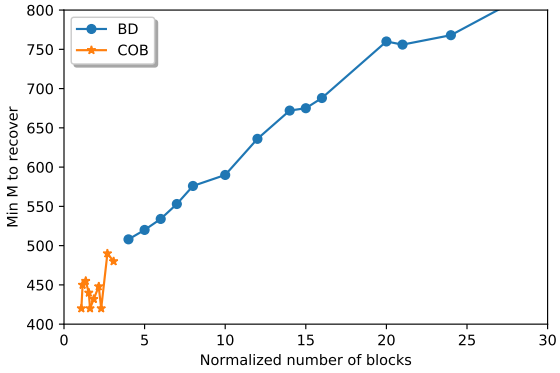
(c) $r = 10$



(d) $r = 15$



(e) $r = 20$



(f) $r = 25$

Figure 4.6: rCOB vs BD over normalized J for r=20 and 25.

The impact of r should be proportional to $\frac{J}{r+1}$, the figure 4.7 will shows that. If we take the BD and rCOB linear regression slope, the theoretical ratio should be $\frac{1}{r+1}$ as it is show also numerically.

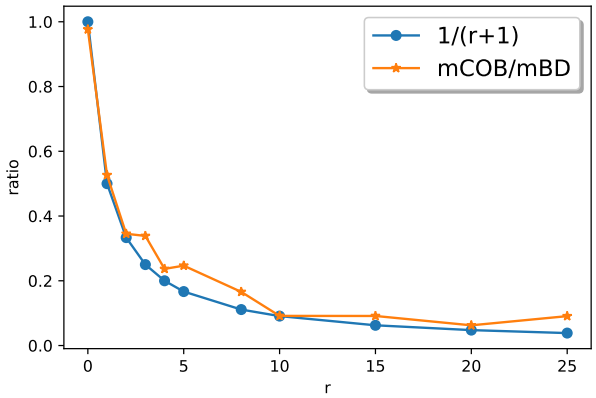


Figure 4.7: Slope ratio rCOB/BD vs r

If $r = 1$, we are improving 20% the number of measurement in BD using COB, as we mention in previous chapter. The slopes observed here by the linear regressions in table 4.3 shows those improvements at least where r is small.

r	**LR equations**	R^2**-Coef. of determination**
1	$y_{BD} = 1.2802x_{BD} + 82.7476$	0.9762
	$y_{COB} = 0.64741x_{COB} + 67.5694$	0.9549
2	$y_{BD} = 1.2802x_{BD} + 82.7476$	0.9762
	$y_{COB} = 0.4419x_{COB} + 67.7321$	0.9283
3	$y_{BD} = 1.2802x_{BD} + 82.7476$	0.9762
	$y_{COB} = 0.4334x_{COB} + 66.1106$	0.9444
4	$y_{BD} = 1.2802x_{BD} + 82.7476$	0.9762
	$y_{COB} = 0.3031x_{COB} + 67.4967$	0.9028
5	$y_{BD} = 1.2802x_{BD} + 82.7476$	0.9762
	$y_{COB} = 0.3158x_{COB} + 66.0416$	0.9054
8	$y_{BD} = 1.2802x_{BD} + 82.7476$	0.9762
	$y_{COB} = 0.2121x_{COB} + 66.5506$	0.7843
10	$y_{BD} = 1.2802x_{BD} + 82.7476$	0.9762
	$y_{COB} = 0.1172x_{COB} + 68.7185$	0.6085
15	$y_{BD} = 1.2802x_{BD} + 82.7476$	0.9762
	$y_{COB} = 0.1169x_{COB} + 67.7633$	0.5119
20	$y_{BD} = 1.2802x_{BD} + 82.7476$	0.9762
	$y_{COB} = 0.0799x_{COB} + 68.8139$	0.2527
25	$y_{BD} = 1.2802x_{BD} + 82.7476$	0.9762
	$y_{COB} = 0.1162x_{COB} + 67.3478$	0.3577

Table 4.3: Linear regression for BD vs rCOB

Chapter 5

The Performance of Wireless Sensor Networks with COB-based CS

5.1 Introduction

In this chapter, we will investigate the performance of *wireless sensor network* (WSN) that utilizes *compressed sensing* (CS) with the proposed *cyclic overlapping block diagonal* (COB) matrix. To demonstrate the potentials of COB, we will also study the performance of WSN without CS and the performance of WSN that uses CS with *block diagonal* (BD) matrix.

The rest of this chapter is organized as what follows. First, in Section 5.2, we will review related work and discuss the novelties and contributions of our study. We then present the system model in Section 5.3, including the network model and the CS-based data collection model. In Section 5.4, we elaborate on the analytical model to evaluate the throughput and delay performance of the WSN with CS.

5.2 The Related Work

In the literature, several CS based data collection schemes for WSNs have been investigated. However, the impact of the measurement matrix has not been fully investigated. On the one hand, many researchers focused on the performance of WSN by simply assuming there exists a measurement matrix. On the other hand, some researchers focused on the design of measurement matrix, particularly the structured measurement matrix, without considering the feature of WSN. Therefore, there is still a significant gap between CS and WSN.

To collect data in WSN using CS, the first complete design was proposed in [23], where the authors proposed a Compressive Data Gathering (CDG) scheme, which was illustrated in Figure 1.5. The CDG basic scheme has been improved and has inspired many further investigation on CS-based data gathering in WSN. For example in [24], the authors considered sparsity in time and space domains for real dataset and they proposed a model using low-rank matrices recovery, a related CS theory using incomplete linear measurement. Rather than recover a sparse vector, the process aim at recovering a matrix from incomplete information where sparsity is replaced by the assumption that matrix has low rank. This setup also appears for example in consumer taste prediction.

In [25], the authors investigated signals that follow a power-law decay model and proposed a new data gathering scheme that explores the spatial-temporal sparsity. They showed that the lifetime improvements can be up to two times over CDG because they reduce the number of measurements. To evaluate it and assert that what they propose is better, real data sets where used and compared with the traditional scheme and CGD. Their proposal is supported when a statistical characterization of the signal is available and thus a Bayesian inference [26] can complement conventional CS methods. An extension of this data gathering scheme for Mobile Sensor Networks is presented in [43]

Although using the proposed methods, the amount of messages sent to the sink is improved and therefore a better performance in the network is supposed, together with a better distribution of the energy consumed by each sensor, there are few works investigating the performance of the data gathering with CS in term of capacity and delay. In [44] is presented a the fundamental limitation of the data gathering with CS for single-sink and multi-sink random networks under protocol interference model, in term of capacity and delay. For the single-sink case, they presented a scheme different that [23] where the assumption on the relatively uniform characteristics of data sparsity distribution in each sub-tree is not considered. They prove that the capacity gain and delay bounds were archived over the traditional scheme.

Although there are many existing studies, to the best of our knowledge, none of them are consider how to improve WSN performance by using new structured measurement matrix other than the BD matrix.

5.3 The System Model

In this section, we present a system model to facilitate our investigation. Specifically, we first introduce the WSN model. We then discuss the data collection scheme that applies the CS technique with the proposed COB matrix.

5.3.1 The WSN Model

In our study, we consider a WSN that consists of one sink and a total of $\widetilde{N}$ static sensors. To simplify the discussions, we assume that all sensors are deployed in a *unit area* and the sink is located at the *center* of the unit area.

We consider that each sensor can sense environmental data. We also assume that each sensor has wireless communication module such that it can communicate with all neighboring sensors that are within a fixed transmission range r. Our model satisfies the standard connectivity as was presented in [19]. To enable CS, we further assume that each sensor has sufficient storage and processing capabilities.

5.3.2 The CS-Based Data Collection Model

In this study, we assume that all sensor nodes measure environmental data periodically. For each period, the sink node shall receive $\widetilde{M}$ measurements to reconstruct $\widetilde{N}$ original sensing data. To collect data in WSN, we assume that a tree-based routing scheme is used, in which a tree rooted at the sink is constructed after the WSN is deployed. In this tree, each sensor will receive data packets from its children nodes and send packets to its parent node.

To apply CS, we consider that each sensor node can utilize *two transmission modes*. In the first mode, a sensor can send its own data to its parent or forward the data from its descendant nodes to its parent. In the second mode, a sensor will use the measurement matrix and transmit measurement data to its parent. To illustrate these two modes, we consider a simple WSN in Figure 5.1.

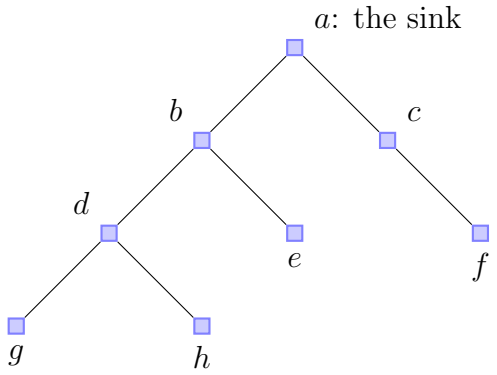


Figure 5.1: A routed tree for data collection.

In this example, there is one sink (node a) and there are seven sensors (nodes b to h). We

assume that at a certain time epoch, all sensors measure environmental data, denoted as $X = [x_b, x_c, \cdots, x_h]^T$.

$$\Phi = \begin{bmatrix} \phi_{1b} & \phi_{1c} & \phi_{1d} & \phi_{1e} & \phi_{1f} & \phi_{1g} & \phi_{1h} \\ \phi_{2b} & \phi_{2c} & \phi_{2d} & \phi_{2e} & \phi_{2f} & \phi_{2g} & \phi_{2h} \end{bmatrix} \tag{5.1}$$

Next, they will use a measurement matrix Φ defined in Eq. (5.1) to collect two measurements in vector $Y = \Phi X$, which is shown in Eq. (5.2):

$$Y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \phi_{1b} & \phi_{1c} & \phi_{1d} & \phi_{1e} & \phi_{1f} & \phi_{1g} & \phi_{1h} \\ \phi_{2b} & \phi_{2c} & \phi_{2d} & \phi_{2e} & \phi_{2f} & \phi_{2g} & \phi_{2h} \end{bmatrix} \begin{bmatrix} x_b \\ x_c \\ x_d \\ x_e \\ x_f \\ x_g \\ x_h \end{bmatrix}. \tag{5.2}$$

To utilize the measurement matrix, we set all leaf nodes, i.e., nodes e, f, g, h, to the first mode, which means that they will only send their own measurement data, i.e., x_e, x_f, x_g, x_h, respectively. We set all internal vertices, i.e., nodes b, c, d to the second mode. In this manner, each of them must wait for the measurement data from all descendants, and then append its own data and send 2 measurements to its parent node.

For example, node c shall first wait for the measurement data x_f from its child node f, and then send two measurement data to the sink a: (1) $\phi_{1c}x_c + \phi_{1f}x_f$ and (2) $\phi_{2c}x_c + \phi_{2f}x_f$.

On the other hand, node b shall first wait for the measurement data from all its descendants, including x_e from node e, and two measurement data from node d: (1) $\phi_{1d}x_d + \phi_{1g}x_g + \phi_{1h}x_h$

and (2) $\phi_{2d}x_d + \phi_{2g}x_g + \phi_{2h}x_h$. After that, node b will send two measurements to the sink: (1) $\phi_{1b}x_b + \phi_{1d}x_d + \phi_{1e}x_e + \phi_{1g}x_g + \phi_{1h}x_h$ and (2) $\phi_{2b}x_b + \phi_{2d}x_d + \phi_{2e}x_e + \phi_{2g}x_g + \phi_{2h}x_h$.

Finally, we note that the objective for choosing one mode is to minimize the number of packets it shall transmit. In other words, a node shall choose mode one if and only if the number of nodes in the sub-tree rooted at this node is smaller than the number of measurements $\widetilde{M}$. For example, the sub-tree rooted at node d has three nodes. Thus, node d shall use mode one if $\widetilde{M} > 3$.

5.4 The Analytical Model

In this section, we first specify necessary assumptions for the analysis, as well as notations. We then explain the outline of the analysis, including two major issues: (1) node partitioning, and (2) scheduling.

5.4.1 Assumptions and Notations

To facilitate the analysis, we make the following assumptions.

- The sensing field is a unit square.
- All $\widetilde{N}$ nodes are uniformly distributed in the sensing field.
- The unit square is partitioned into $C \times C$ squares, namely, cells, where C is an integer and it is chosen in a way such that one cell has at least one node. We further define $c = \frac{1}{C}$.
- $d(i, j)$ means the Euclidean distance between node i and node j.

- Node i can successfully receive a message from node j if and only if (1) $d(i,j) \leq r$, and (2) $d(i,k) > (1+\Delta)r$, where node k is a node that transmits at the same time and Δ represents the impact of interference. In the literature, such a model is known as a protocol model.

- One node in a cell can send messages to any node in four neighboring cells (up, down, left, right). This assumption implies that the transmission range $r \geq \sqrt{5}c$.

- Due to the complexity of the analysis, we consider five measurement matrices: (1) full matrix (i.e., BD matrix with $J = 1$), denoted as BD1, (2) BD matrix with $J = 2$, denoted as BD2, (3) BD matrix with $J = 4$, denoted as BD4, (4) COB matrix with $J = 4$, denoted as COB4, (5) COB matrix with $J = 8$, denoted as COB8.

- A *time division multiple access* (TDMA) scheme is used to determine when a sensor can send and receive a message. Moreover, in the TDMA scheme, the time horizon is partitioned into equal-sized time slots. The duration of a slot is t_s seconds, which can be used to send one message.

- The transmission rate is fixed to W bits/second.

5.4.2 Overview of the Analysis

To analyze the throughput and delay performance of a WSN that utilizes CS with BD or COB, there are two major issues: (1) how to partition the sensing field such that we can apply CS with BD or COB, and (2) how to design a feasible time schedule for each node to send messages. Next, we will explain how to address these two questions one-by-one. Based on them, we will evaluate the throughput and delay of WSN with CS.

5.4.3 Field Partitioning for BD and COB

As explained previously, we consider only five measurement matrices. Next, we first explain the most simple case, BD1, in Figure 5.2. In this figure, the picture to the left-hand-side represents a unit area that is partitioned into 9×9 cells[1], and the smaller picture to the right hand side represents the shape of the measurement matrix. Since BD1 is used, the measurement matrix is full, represented by a single yellow block in the matrix. Correspondingly, we note that the field consists of all yellow cells, except the cell at the center, which contains the sink.

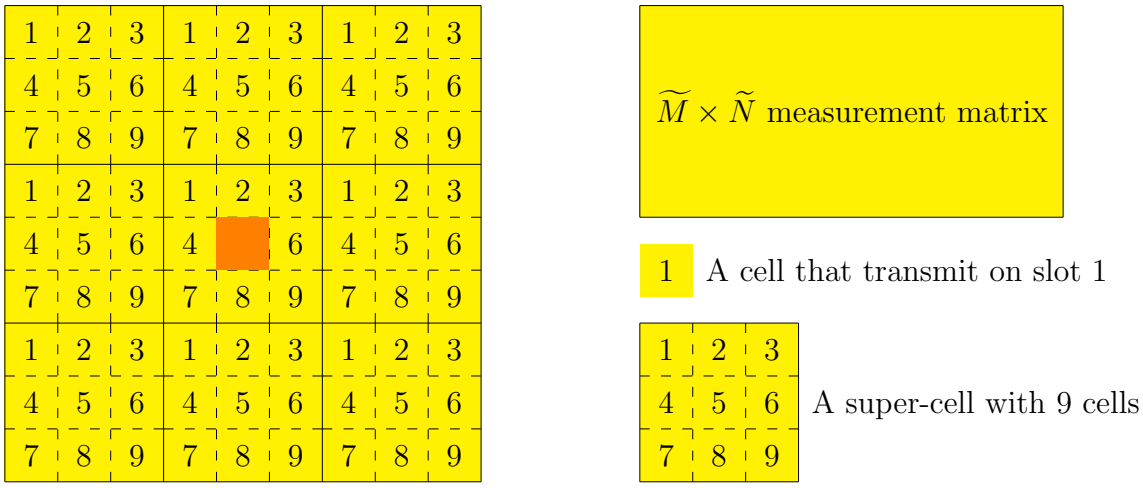


Figure 5.2: An example of field partition for BD1.

In Figure 5.3, we illustrate the field partitioning when BD2 is used. Since BD2 is used, the measurement matrix has two blocks, yellow and green. Correspondingly, we note that the field consists of two parts, yellow and green, and they have the same number of cells.

In Figure 5.4, we illustrate the field partitioning when BD4 is used. Since BD4 is used, the measurement matrix has four blocks, yellow, green, pink, and cyan. Correspondingly, we

[1]Here we note that each cell has a number. These numbers are related to the TDMA schedule that will be explained in the next subsection.

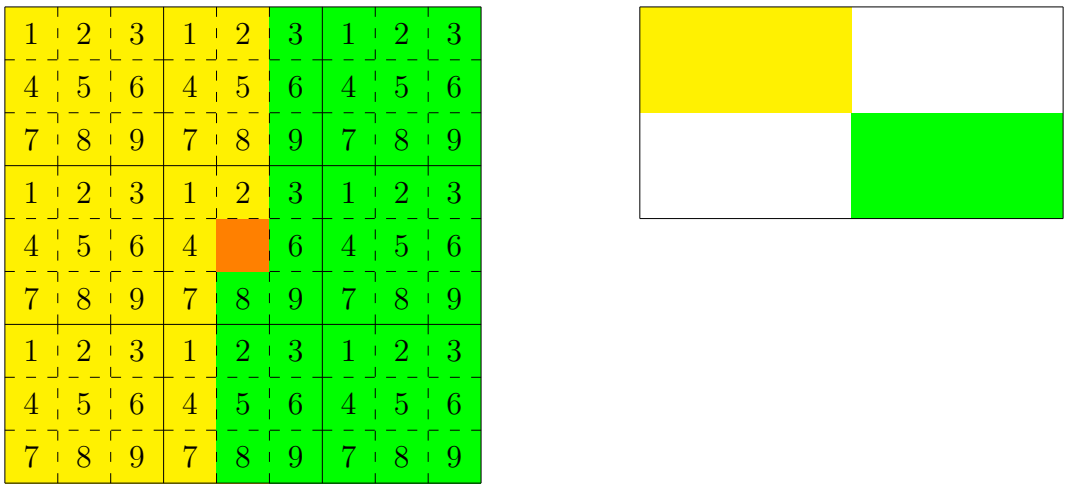


Figure 5.3: An example of field partition for BD2.

note that the field consists of four parts, represented by the same four colors, and they have the same number of cells.

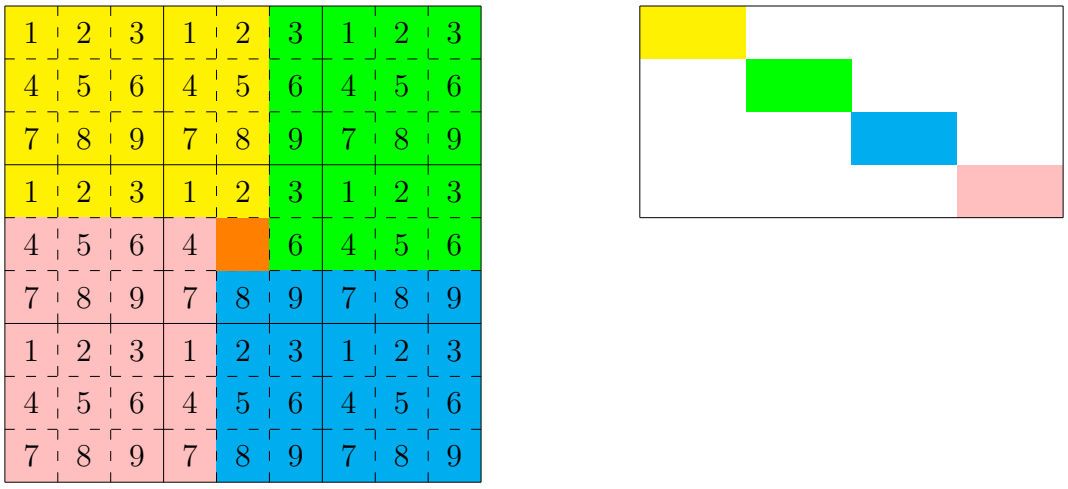


Figure 5.4: An example of field partition for BD4.

In the previous example, we have demonstrated the field partitioning when BD is used. We can observe that, since blocks have no overlaps, the parts also have no overlaps. By comparison, when COB is used, we need to find partitions with overlaps. In Figure 5.5,

we illustrate the field partitioning when COB4 is used. Similar to BD4, COB4 has four blocks in the measurement matrix: yellow, green, pink, and cyan. To enable overlapping, we design two partition schemes in the left-hand-side of Figure 5.5, one for yellow and green, and another for pink and cyan. The first one is the same as the partition for BD2, i.e., left and right, while the second one uses a different method, i.e., up and down. In this manner, we can observe that the size of four regions are the same, one region is overlapping with two regions and the overlapping parts have the same size.

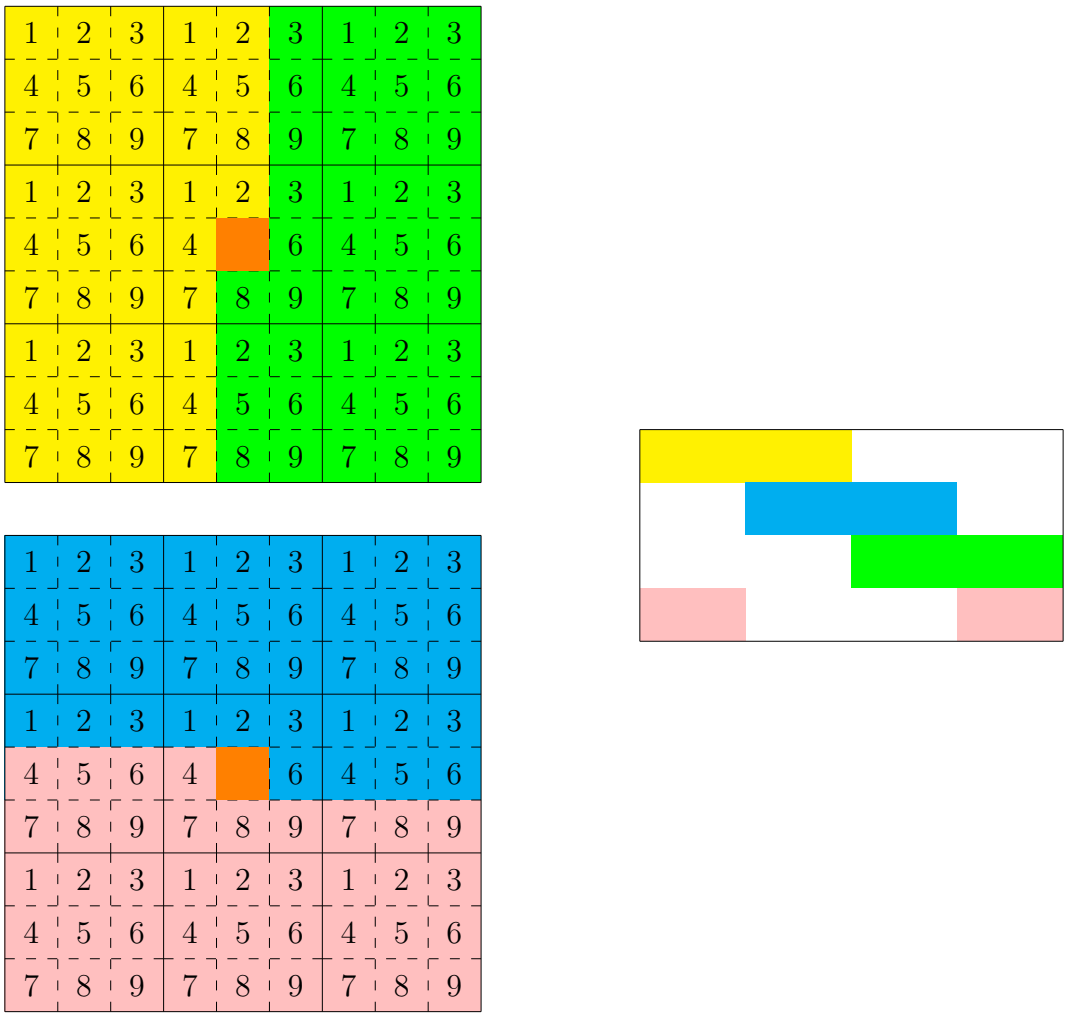


Figure 5.5: An example of field partition for COB4.

Finally, in Figure 5.6, we illustrate how to partition the field for COB8. In this case, there

are eight blocks in the measurement matrix. Similar to those for COB4, we also develop two partitioning maps to define eight regions, in which the first one is the same of the partitioning for BD4.

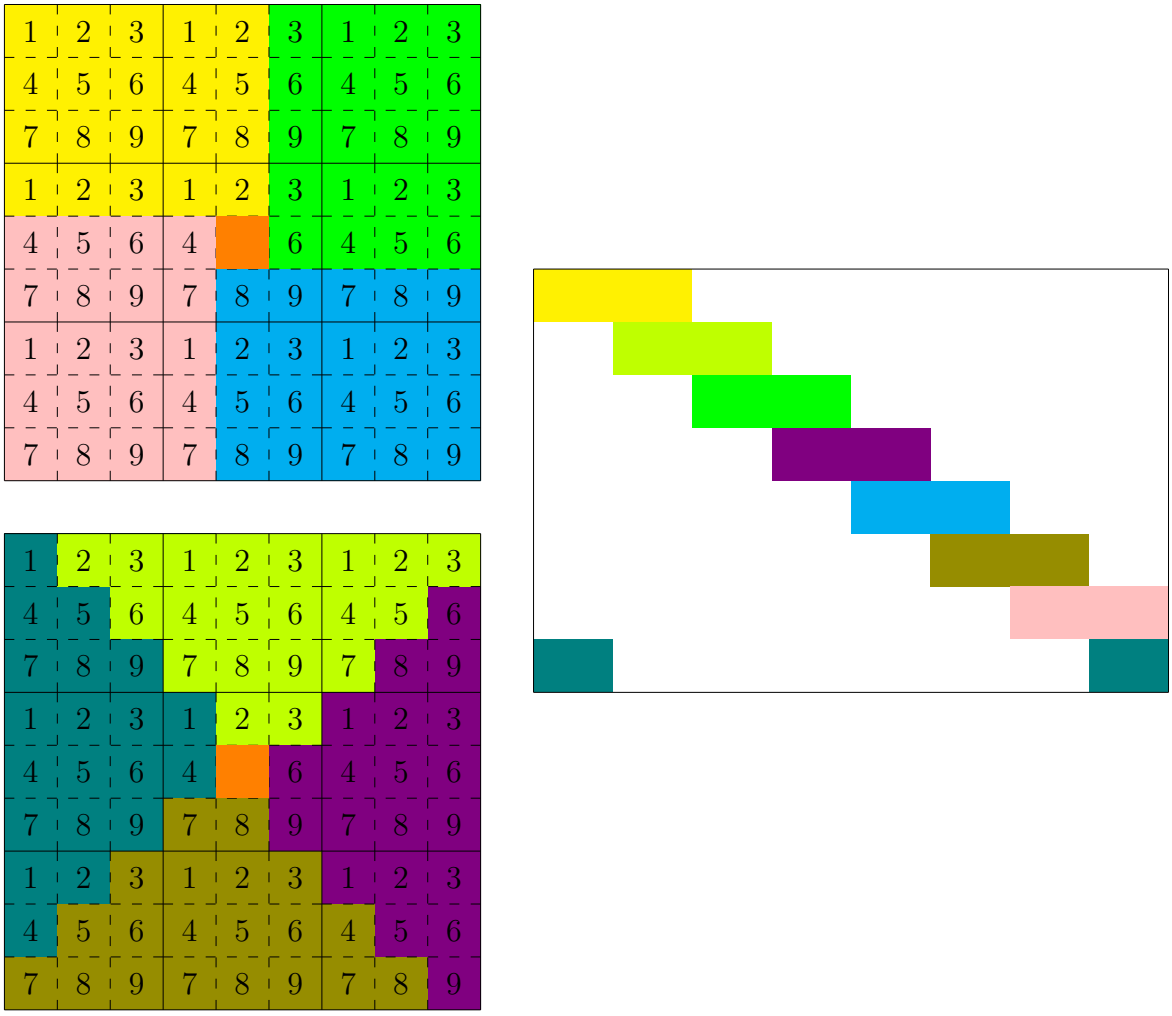


Figure 5.6: An example of field partition for COB8.

5.4.4 Feasible Scheduling Scheme

In this subsection, we discuss how to design a feasible TDMA scheduling scheme to allow each node to transmit and receive messages. Specifically, we first define two types of timing

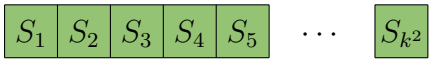


Figure 5.7: The frame of k^2 time slots

format: (1) frame, and (2) super-frame. We then discuss how to design a feasible scheduling scheme.

1. Frame: We apply a typical approach [44] to define a *frame* as a sequence of k^2 time slots, as shown in Figure 5.8. In this manner, all cells with index i (as shown in previous figures) can use time slot S_i in a frame to transmit. Moreover, parameter k is chosen in a way such that the co-channel interference, i.e., interference generated by other cell with index i, is limited according to the protocol model.

2. Super-frame: We consider that a super-frame consists of a number of frames. Since there are different WSN applications that can be accommodated by different scheduling schemes, here we define three types of frames in a super-frame: (1) local frame, (2) transit frame, and (3) idle frame. A local frame is used to enable transmissions within a cell. Meanwhile, a transit frame is used to transmit a message from one cell to an adjacent cell. In addition, an idle frame is a frame that has no transmission.

Next, we discuss design criteria and constraints.

1. Design criteria and constraints for frame

   To utilize the frame, we consider that a *group* is composed of $k \times k$ cells in a square region. For example, in Figure 5.2, we consider $k = 3$ and there are a total of 9 groups in the field. In this figure, the number in a cell represents the index of time slot allocated to the cell.

Here we can observe that the same index is assigned to multiple cells, which means these are using the same time slot and thus one node in each of these cells can transmit at the same time. Consequently, the minimal distance between a receiver and an interfering node is $(k-2)c$. To satisfy the interference constraint specified previously, we can derive

$$(k-2)c > (1+\Delta)r \geq (1+\Delta)\sqrt{5}c. \tag{5.3}$$

This analysis leads to the constraint for choosing k:

$$k > 2 + (1+\Delta)\sqrt{5}. \tag{5.4}$$

2. Design criteria and constraints for super-frame

   - The flow constraint in the network level: In the network level, the flow constraint means that, in a super-frame, $\widetilde{N}$ sensors can generate $\widetilde{N}$ messages and the sink can receive $\widetilde{M}$ messages.
   - The flow constraint for sensor nodes on mode 1: A node on mode 1 will send its own data or forward messages without appending its own data. In this case, the flow constrain means that the number of messages sent by a node is the same as the number of messages it generates plus the number of messages it receives in a super-frame.
   - The flow constraint for sensor nodes on mode 2: If a node is on mode 2, the node will generate and send m messages in one super-frame, where m depends on the measurement matrix.
   - The timing constraint for forwarding: If a node is on mode 1, then it must be scheduled to forward a message after it receives it. If a node is on mode 2, then it must be scheduled to send a measurement message after it receives all messages that are necessary to construct the measurement message. In addition, the design

shall lead the minimal delay.

5.4.5 Performance of a Throughput-Maximized WSN

In this subsection, we investigate the throughput and delay performance of a WSN that aims at maximizing the throughput of the WSN. In what follows, we first discuss the performance of BD, and then elaborate on the performance of COB.

When BD is used, we will use the field partitioning schemes described in Section 5.4.3. Since the objective is to maximize throughput, we consider that a super-frame consists of only local and transit frames. Specifically, we design a super-frame that consists of g local frames and m transit frames, as shown in Figure 5.8.

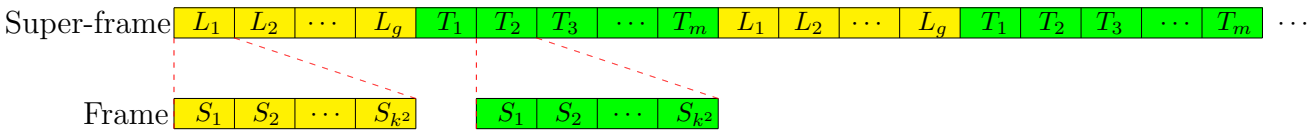


Figure 5.8: Super-frame format for BD in WSN that maximizes throughput.

To maximize data collection, we consider that all sensors sense the environmental data at the beginning of each super-frame, and that the time horizon is partitioned into super-frames without any gap (i.e., idle period).

According to the field partitioning scheme, we can form a *cluster* in each cell and we can choose one node in a cluster as the *cluster head* that forwards all messages to neighboring cell. In this way, we can choose g as the maximum number of nodes in a cell minus one. In our analysis, g can be estimated by nc^2. By using g frames, all sensors that are not cluster head can operate on mode 1 and send their original messages to cluster heads.

To satisfy the flow constraint in the network level, we let $m = \frac{\widetilde{M}}{J}$ for BD. This is feasible because, for BD matrices with $J \in \{1, 2, 4\}$, the sink can collect one measurement from

each region in each frame. Specifically, in Figure 5.2, the sink can receive four measurement messages in slots 2, 4, 6, 8 in each frame. Since these four messages are used to construct one measurement, the sink can receive one measurement in each frame. Similarly, in Figure 5.3, the sink can receive four measurement messages in slots 2, 4, 6, 8 in each frame. However, messages received in slots 2 and 4 can be used to construct a measurement for the yellow region, while messages received in slots 6 and 8 can be used to construct a measurement for the green region. Finally, in Figure 5.4, we can see that the sink can receive four measurement messages in slots 2, 4, 6, 8, where each of them is measurement from one region.

In our design, we consider that all cluster heads are operating on mode 2. To satisfy the flow constraint for nodes in mode 2 and the timing constraint, we consider that frame T_i ($1 \leq i \leq m$) will be used by all cluster heads that have the largest hop distance (denoted as h) to the sink to send the i-th measurement. After receiving these messages, cluster heads whose hop distance to the sink is $h-1$ will use the next available transit frame to send the i-th measurement. Specifically, when $i < m$, these messages can be sent in frame T_{i+1} if the same super-frame. However, when $i = m$, the messages will be sent in T_1 of the next super-frame. In a similar manner, all cluster heads with hop distance $h-2$ to 1 can be scheduled to transmit m measurement messages.

Based on the description above, we can see that (1) the duration of a super-frame is $k^2(g+m)t_s$, (2) the sink can receive $\widetilde{M}$ messages in each super-frame, (3) the amount of bits in a measurement message is Wt_s, and (4) the sink can then reconstruct $\widetilde{N}$ original signals. Therefore, the achievable per-node *throughput* $\lambda_{BD(J)}$ is

$$\lambda_{BD(J)} = \frac{Wt_s\widetilde{M}}{k^2(g+m)t_s} \cdot \frac{\widetilde{N}}{\widetilde{M}} \cdot \frac{1}{\widetilde{N}} = \frac{W}{k^2(g+m)} = \frac{W}{k^2(g+\frac{\widetilde{M}}{J})} \tag{5.5}$$

According to the scheduling scheme, all sensors collect environmental data at the beginning of each frame. So we consider the delay as the duration from the epoch that the data are

collect to the time that the last measurement message is received by the sink. To understand the delay, we can first see that it is bounded by the delay of the last measurement, i.e., the m-th measurement. Clearly, the first message for the m-th measurement is send in the m-th frame in the same super-frame that the data is collected. This message shall be forwarded $h-1$ times. Since a message can advance m hops in a super-frame, we need to use $\lceil \frac{h-1}{m} \rceil$ super-frames, which means that we need to wait for $\left(\lceil \frac{h-1}{m} \rceil + 1\right) g$ local frames. In addition, we need $m+h-1$ transit frames to deliver all measurements. Consequently, the total delay $\tau_{BD(J)}$ is

$$\tau_{BD(J)} = \left[\left(\left\lceil \frac{h-1}{m} \right\rceil + 1\right) g + (m+h-1)\right] k^2 t_s. \tag{5.6}$$

When COB is used, we will use the field partitioning schemes described in Section 5.4.3. Since to the design for BD, we consider that a super-frame consists of only local and transit frames. Specifically, we design a super-frame that consists of g local frames and $2m$ transit frames, as shown in Figure 5.8, where m denotes the number of measurements for each region. Notice that we need $2m$ transit frames because we use two field partitioning schemes to facilitate COB.

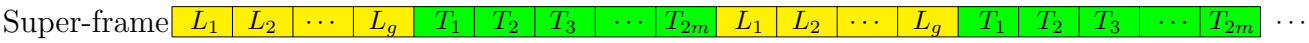


Figure 5.9: Super-frame format for COB in WSN that maximizes throughput.

Since the scheduling for COB is very similar to that of BD, we can derive the throughput and delay as follows.

$$\lambda_{COB(J)} = \frac{W t_s \widetilde{M}}{k^2(g+2m)t_s} \cdot \frac{\widetilde{N}}{\widetilde{M}} \cdot \frac{1}{\widetilde{N}} = \frac{W}{k^2(g+2m)} = \frac{W}{k^2(g+\frac{2\widetilde{M}}{J})} \tag{5.7}$$

$$\tau_{COB(J)} = \left[\left(\left\lceil \frac{h-1}{m} \right\rceil + 1\right) g + 2(m+h-1)\right] k^2 t_s. \tag{5.8}$$

5.4.6 Performance of a Low-duty-cycle WSN

In practice, many WSNs are deployed to support low duty-cycle monitoring applications. For example, in a smart grid, power consumption can be sent once every 15 minutes. To facilitate such applications, we design scheduling schemes for BD and COB that are slightly different to the ones we discussed in Section 5.4.5.

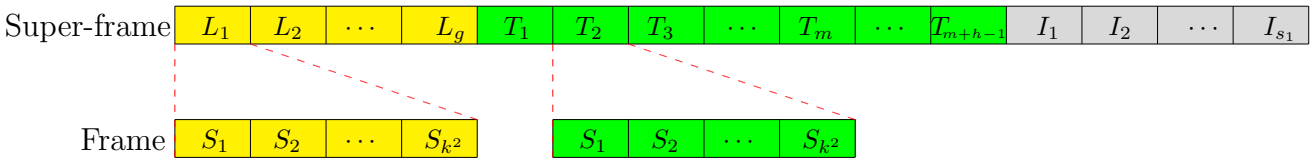


Figure 5.10: Super-frame format for BD in low-duty-cycle WSN.

As shown in Figure 5.10, we design a super-frame format that consists of local frames, transit frames, and idle frames. Compared to the format in Figure 5.8, another difference is that there are $m+h-1$ transit frames, which are used to deliver all measurements. Consequently, we can derive the delay as follows.

$$\tau'_{BD(J)} = [g + (m + h - 1)]k^2 t_s. \tag{5.9}$$

In a similar way, we design super-frame format for COB in Figure 5.11. The delay performance is:

$$\tau'_{COB(J)} = [g + 2(m + h - 1)]k^2 t_s. \tag{5.10}$$

5.5 Numerical Results

In this section, we investigate the performance of WSN using practical settings.

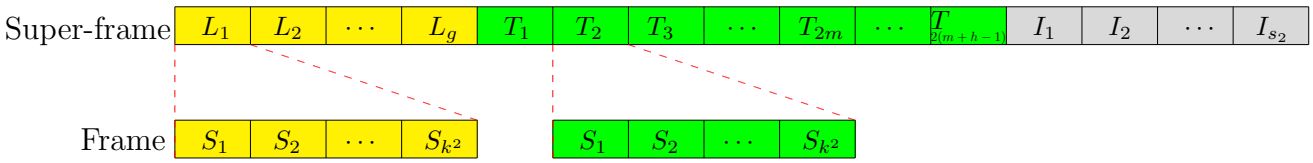


Figure 5.11: Super-frame format for COB in low-duty-cycle WSN.

Parameter	**Value**
Area	$200\ m \times 200\ m = 1\ unit \times 1\ unit$
Unit length	$200\ m$
$1\ m$	$0.005\ units$
Transmission range r	$20m = 0.1\ units$
$\widetilde{N}$	1680
minimal C	$\left\lceil \frac{\sqrt{5}}{r} \right\rceil = 23$
c	$\frac{1}{C} = 0.043478261\ units$
g	$\lceil nc^2 - 1 \rceil = 3$
h	$h = C - 1 = 22$
W	$40,000\ bps$
t_s	$10\ ms$

Table 5.1: Simulation settings.

As shown in Table 5.1, we consider that a total of $\widetilde{N} = 1680$ sensors are uniformly deployed in a $200m \times 200m$ square area. We then consider $200m$ as a unit. Next, we use the parameters of sensor node defined in [liu12throughput], including the transmission range, data rate. Particularly, we use the transmission range ($20meters$), to derive the minimal $C = 23$ and the corresponding side length of a cell c.

Next, in Table 5.2, we choose the minimal k that satisfies the interference constraint corresponding to different Δ.

Δ	$k = \lceil 2 + (1 + \Delta)\sqrt{5} \rceil$
0.5	6
1	7
1.5	8

Table 5.2: TDMA parameter k corresponding to Δ

5.5.1 Using a random uniform-distributed sparse signal

Based on the above numerical results, we show the throughput and delay performance of throughput-maximized WSN that utilize different BD and COB schemes in Table 5.3 when $\Delta = 0.5$, Table 5.4 when $\Delta = 1$, and Table 5.5 when $\Delta = 1.5$. In all experiments, we can observe that COB8 has the best throughput and delay performance.

Matrix	$\widetilde{M}$	**Throughput 1 (bps)**	**Delay 1 (seconds)**	**Delay 2 (seconds)**
BD1 (Full)	430	2.566076469	164.52	163.44
BD2	480	4.572473708	96.12	95.04
COB4	436	5.027652086	95.76	94.68
BD4	544	7.993605116	58.68	57.6
COB8	440	9.832841691	56.88	55.8

Table 5.3: Throughput and delay performance when $\Delta = 0.5$

Matrix	$\widetilde{M}$	**Throughput 1 (bps)**	**Delay 1 (seconds)**	**Delay 2 (seconds)**
BD1 (Full)	430	1.885280671	223.93	222.46
BD2	480	3.359368439	130.83	129.36
COB4	436	3.693785206	130.34	128.87
BD4	544	5.872852738	79.87	78.4
COB8	440	7.224128589	77.42	75.95

Table 5.4: Throughput and delay performance when $\Delta = 1$

Matrix	$\widetilde{M}$	Throughput 1 (bps)	Delay 1 (seconds)	Delay 2 (seconds)
BD1 (Full)	430	1.443418014	292.48	290.56
BD2	480	2.572016461	170.88	168.96
COB4	436	2.828054299	170.24	168.32
BD4	544	4.496402878	104.32	102.4
COB8	440	5.530973451	101.12	99.2

Table 5.5: Throughput and delay performance when $\Delta = 1.5$

5.5.2 Using a random non-uniform-distributed sparse signal

This section will shows results when the signal has the worst case scenario in its sparsity, i.e., values are not randomly distributed, on the contrary, their sparsity is concentrated from a certain start point. We will use the same parameters as in the previous section, including the size of the signal $\widetilde{N} = 1680$ with a sparsity of $S = 100$. In this case the numbers of measurements $\widetilde{M}$ get worst compared with previous section when the number of blocks increase.

We show the throughput and delay performance of throughput-maximized WSN that utilize different BD and COB schemes in Table 5.6 when $\Delta = 0.5$, Table 5.7 when $\Delta = 1$, and Table 5.8 when $\Delta = 1.5$. Even for the worst case, in all experiments, we can observe that COB8 still has the best throughput and delay performance.

Matrix	$\widetilde{M}$	Throughput 1 (bps)	Delay 1 (seconds)	Delay 2 (seconds)
BD1 (Full)	430	2.566076469	164.52	163.44
BD2	670	3.28731098	130.32	129.24
COB4	556	3.954132068	117.36	116.28
BD4	1024	4.29000429	101.28	100.8
COB8	784	5.58347292	87.84	86.76

Table 5.6: Throughput and delay performance when $\Delta = 0.5$ worst case

Matrix	$\widetilde{M}$	**Throughput 1 (bps)**	**Delay 1 (seconds)**	**Delay 2 (seconds)**
BD1 (Full)	430	1.885280671	223.93	175.91
BD2	670	2.41516725	177.38	129.36
COB4	556	2.905076621	159.74	158.27
BD4	1024	3.151839887	138.67	137.2
COB8	784	4.10214337	119.56	118.09

Table 5.7: Throughput and delay performance when $\Delta = 1$ worst case

Matrix	$\widetilde{M}$	**Throughput 1 (bps)**	**Delay 1 (seconds)**	**Delay 2 (seconds)**
BD1 (Full)	430	1.443418014	292.48	290.56
BD2	670	1.849112426	231.68	229.76
COB4	556	2.224199288	208.64	206.72
BD4	1024	2.413127413	181.12	179.2
COB8	784	3.140703518	156.16	154.24

Table 5.8: Throughput and delay performance when $\Delta = 1.5$ worst case

Chapter 6

Conclusions and future work

6.1 Conclusions

In this dissertation, we have presented a comprehensive study on structured measurement matrix for *compressed sensing* (CS) to improve the performance of data collection in wireless sensor network (WSN). In the literature, CS technique was first applied in WSN in a scheme, called *compressive data gathering* (CDG), where distributed sensing signals are assumed to be sparse or compressible. Compared to the traditional data collection scheme that requires the sink to received a message from every sensor, CDG can significantly improve the performance of data collection by exploiting CS because the sink only needs to receive much fewer measurement messages. Nevertheless, to enjoy the benefit of CS, CDG and most following work assume that measurements are generated in individual regions, where there is no overlap between adjacent regions. In this manner, the measurement matrix is a *block diagonal* matrix (BD) and there are some previous work investigate the restricted isometric property (RIP) of BD matrix. Although these studies are important, we notice that non-overlapping blocks cannot explicitly represent the correlation of signals in adjacent regions.

In this study, we envision that the performance of data collection in WSN can be improved if we design new measurement matrix in which the correlation of signals in adjacent regions can be directly exploited.

Specifically, we first proposed a novel family of measurement matrices for two-dimensional WSN with the following key features: (1) a matrix consists of multiple blocks of non-zero entries, (2) each block is used to generate measurements for a region in WSN, and (3) each region is overlapping with two adjacent regions. In this way, the measurement matrix has a cyclic overlapping block diagonal (COB) structure, so we named it as COB matrix. For the COB matrix, we first developed a rigorous proof to show that it satisfies RIP when the total number of measurements exceeds a certain threshold, i.e., a lower bound. Compared to existing bounds, our bound can more accurately reveal the impacts of important design parameters, including the number of blocks, the sparsity of signal, and the number of sensor nodes. To validate the theoretical analysis, we conducted extensive numerical experiments to test the performance of signal recovery in various scenarios. These results confirm that our theoretical analysis can accurately capture the impacts of factors involved on the number of measurements, and that COB outperforms BD in virtually all scenarios.

The promising results of COB motivate us to developed a novel framework for measurement matrix used in WSN for data collection. With the new generalized COB framework, signals collected by sensors in nearby area can be use to generate measurements so that the correlation of these signals can be better represented. For the generalized COB, we also theoretically proved that it satisfy RIP, and we conducted numerical experiments to studied the impacts of various factors.

Finally, we evaluated the throughput and delay performance of CS-based WSN with the proposed measurement matrix, in which we used practical settings, such as the transmission range of sensor, the deployment of sensor, etc. The theoretical and numerical result confirm that the proposed COB schemes can improve throughout and delay performance over

previous data collection models in WSN.

6.2 Future work

The work achieved so far has left good results when CS in WSN is applied through the COB matrix in term of throughput and delay performance over the BD matrix, but there are some aspects that are in progress, such as

- investigating more non-uniform random signals to represent different scenarios, and
- applying the COB matrix in real WSN.

Finally, we will investigate the COB matrix in others applications, where correlations in large-scale scenarios exist, such as in imaging processing.

Bibliography

[1] Emmanuel J. Candes and Terence Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, dec 2005.

[2] David L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, apr 2006.

[3] Emmanuel J. Candès, Justin K. Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.

[4] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM J. Comput.*, 24(2):227–234, April 1995.

[5] R. Prony. Essai expérimental et analytique sur les lois de la dilatabilité des fluides élastiques et sur celles de la force expansive de la vapeur de l'eau et de la vapeur de l'alkool, à différentes températures. *J. École Polytechnique*, 1:24–76, 1795.

[6] Simon Foucart, Alain Pajor, Holger Rauhut, and Tino Ullrich. The Gelfand widths of l_p-balls for $0 < p \leq 1$. *Journal of Complexity*, 26(6):629–640, 2010.

[7] Emmanuel J. Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, 2006.

[8] Michael B. Wakin Marco F. Duarte Dror Baron Shriram Sarvotham Kevin F. Kelly Richard G. Baraniuk Dharmpal Takhar, Jason N. Laska. A new compressive imaging camera architecture using optical-domain compression. *Proc.SPIE*, 6065:6065 – 6065 – 10, 2006.

[9] G. Huang, H. Jiang, K. Matthews, and P. Wilford. Lensless imaging by compressive sensing. In *2013 IEEE International Conference on Image Processing*, pages 2101–2105, Sept 2013.

[10] G. Satat, M. Tancik, and R. Raskar. Lensless imaging with compressive ultrafast sensing. *IEEE Transactions on Computational Imaging*, 3(3):398–407, Sept 2017.

[11] Emmanuel J. Candès and Michael B. Wakin. An introduction to compressive sampling: A sensing/sampling paradigm that goes against the common knowledge in data acquisition. *IEEE Signal Processing Magazine*, 25(2):21–30, mar 2008.

[12] R. G. Baraniuk. Compressive sensing [lecture notes]. *IEEE Signal Processing Magazine*, 24(4):118–121, July 2007.

[13] Dana Mackenzie. Compressed sensing makes every pixel count. *What's Happening in the Mathematical Sciences*, 7(9), 2010.

[14] Jordan Ellenberg. Fill in the blanks: Using math to turn lo-res datasets into hi-res samples. *WIRED*, 2010.

[15] Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. Birkhäuser Basel, 2013.

[16] Yonina and at al. *Compressed Sensing: Theory and Applications*. Cambridge University Press, 2012.

[17] Zhu Han, Husheng Li, and Wotao Yin. *Compressive Sensing for Wireless Networks*. Cambridge University Press, 2013.

[18] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, Mar 2000.

[19] Feng Xue and P. R. Kumar. The number of neighbors needed for connectivity of wireless networks. *Wirel. Netw.*, 10(2):169–181, March 2004.

[20] Enrique J. Duarte-Melo and Mingyan Liu. Data-gathering wireless sensor networks: Organization and capacity. *Comput. Netw.*, 43(4):519–537, November 2003.

[21] W. Liu, K. Lu, J. Wang, G. Xing, and L. Huang. Performance analysis of wireless sensor networks with mobile sinks. *IEEE Transactions on Vehicular Technology*, 61(6):2777–2788, July 2012.

[22] W. Liu, K. Lu, J. Wang, L. Huang, and D. O. Wu. On the throughput capacity of wireless sensor networks with mobile relays. *IEEE Transactions on Vehicular Technology*, 61(4):1801–1809, May 2012.

[23] Chong Luo, Feng Wu, Jun Sun, and Chang Wen Chen. Compressive data gathering for large-scale wireless sensor networks. In *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking*, MobiCom '09, pages 145–156, New York, NY, USA, 2009. ACM.

[24] J. Cheng, Q. Ye, H. Jiang, D. Wang, and C. Wang. Stcdg: An efficient data gathering algorithm based on matrix completion for wireless sensor networks. *IEEE Transactions on Wireless Communications*, 12(2):850–861, February 2013.

[25] X. Y. Liu, Y. Zhu, L. Kong, C. Liu, Y. Gu, A. V. Vasilakos, and M. Y. Wu. CDC: Compressive data collection for wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 26(8):2188–2197, Aug 2015.

[26] D. Baron, S. Sarvotham, and R. G. Baraniuk. Bayesian compressive sensing via belief propagation. *IEEE Transactions on Signal Processing*, 58(1):269–280, Jan 2010.

[27] Armin Eftekhari, Han Lun Yap, Christopher J. Rozell, and Michael B. Wakin. The restricted isometry property for random block diagonal matrices. *Applied and Computational Harmonic Analysis*, 38(1):1–31, 2015.

[28] Juan Castorena and Charles D. Creusere. The restricted isometry property for banded random matrices. *IEEE Transactions on Signal Processing*, 62(19):5073–5084, 2014.

[29] Holger Rauhut. Compressive sensing and structured random matrices, 2011.

[30] D. L. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. *IEEE Transactions on Information Theory*, 47(7):2845–2862, Nov 2001.

[31] Philip J Davis. *Circulant Matrices*. New York: Chelsea, 2 edition, 1994.

[32] Marco F. Duarte and Yonina C. Eldar. Structured compressed sensing: From theory to applications. *IEEE Transactions on Signal Processing*, 59(9):4053–4085, jun 2011.

[33] Justin Romberg. Compressive sensing by random convolution. *SIAM J. Img. Sci.*, 2(4):1098–1128, November 2009.

[34] Gene H. Golub and Charles F. Van Loan. *Matrix Computations (3rd Ed.)*. Johns Hopkins University Press, Baltimore, MD, USA, 1996.

[35] Thong T. Do; Lu Gan; Nam H. Nguyen; Trac D. Tran. Fast and efficient compressive sensing using structurally random matrices. *IEEE Transactions on Signal Processing*, 60(1):139 – 154, oct 2012.

[36] Felix Krahmer, Shahar Mendelson, and Holger Rauhut. Suprema of chaos processes and the restricted isometry property. *Communications on Pure and Applied Mathematics*, 67(11):1877–1904, 2014.

[37] Peng Zhang, Lu Gan, Cong Ling, and Sumei Sun. Uniform recovery bounds for structured random matrices in corrupted compressed sensing. *CoRR*, abs/1706.09087, 2017.

[38] M. Talagrand. *Upper and Lower Bounds for Stochastic Processes: Modern Methods and Classical Problems*. Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge / A Series of Modern Surveys in Mathematics. Springer Berlin Heidelberg, 2014.

[39] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. In Yonina C. Eldar and GittaEditors Kutyniok, editors, *Compressed Sensing: Theory and Applications*, page 210–268. Cambridge University Press, 2012.

[40] A. M. Tillmann and M. E. Pfetsch. The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing. *IEEE Transactions on Information Theory*, 60(2):1248–1259, Feb 2014.

[41] Charles Dossal, Gabriel Peyré, and Jalal Fadili. A numerical exploration of compressed sampling recovery. *Linear Algebra and its Applications*, 432(7):1663 – 1679, 2010.

[42] Michel Journée, Yurii Nesterov, Peter Richtárik, and Rodolphe Sepulchre. Generalized power method for sparse principal component analysis. *J. Mach. Learn. Res.*, 11:517–553, March 2010.

[43] Haifeng Zheng, Jiayin Li, Xinxin Feng, Wenzhong Guo, Zhonghui Chen, and Neal Xiong. Spatial-temporal data collection with compressive sensing in mobile sensor networks. *Sensors*, 17(11), 2017.

[44] H. Zheng, S. Xiao, X. Wang, X. Tian, and M. Guizani. Capacity and delay analysis for data gathering with compressive sensing in wireless sensor networks. *IEEE Transactions on Wireless Communications*, 12(2):917–927, February 2013.