

國立臺灣大學電機資訊學院電信工程學研究所

碩士論文

Graduate Institute of Communication Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

基於感測器錯誤與陣列對稱性的稀疏陣列穩健性分析

Robustness Analysis of Sparse Arrays with Sensor Failures
and Array Symmetry

陳崇瀚

Chung-Han Chen

指導教授: 劉俊麟 博士

Advisor: Chun-Lin Liu, Ph.D.

中華民國 111 年 8 月

August, 2022

國立臺灣大學碩士學位論文
口試委員會審定書

MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

基於感測器錯誤與陣列對稱性的稀疏陣列穩健性分析

Robustness Analysis of Sparse Arrays with Sensor Failures and
Array Symmetry

本論文係陳崇瀚 (R09942126) 在國立臺灣大學電信工程學研究所完成之碩士學位論文，於民國 111 年 8 月 26 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Graduate Institute of Communication Engineering on 26 August 2022, have examined a Master's thesis entitled above presented by Chung-Han Chen (R09942126) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

劉俊麟

(指導教授 Advisor)

馮世邁

薛柏育

系主任/所長 Director:

劉錫增





致謝

回想起兩年前，抱著期待與緊張的心情進入實驗室，對一切都還很陌生，如今即將完成自己的研究，著實令人感動。這一路走來，成長了許多，從課堂中學到的理論基礎、論文中學到的研究方法、還有做過的各種實驗模擬，都使我一點一滴地進步。

能夠走到今天，有許多要感謝的人。首先要感謝指導教授，劉俊麟教授。即使很忙碌，也依然每週都與實驗室的同學個別 meeting。在與教授討論的過程中，從教授身上學到對學術研究嚴謹的態度，並也激發出許多想法。在論文撰寫方面，很感謝教授的多方建議，也因此我才能完成這份研究。

接著要感謝實驗室一起努力的其他三位同學，在遇到困難時互相協助，遇到研究與課業上的問題時互相討論。雖然只相處短短的兩年，但也謝謝各位使我的研究所生活更加的豐富。這段時間的辛苦，即將有個成果，大家共同的努力，使我們能夠一起走到今天。

最後要感謝我的家人們，在學習的這條路上不斷地給予支持，默默地成為重要的力量。一直以來心中都充滿著感謝，也要將完成碩士論文的這份感動與喜悅與你們一起分享。





摘要

在一維稀疏陣列中，多個感測器以不同的間距被擺放在一直線上。稀疏陣列能夠用 $O(N)$ 個感測器分辨出 $O(N^2)$ 個不相關訊號源，原因是差異協列中間之連續片段有 $O(N^2)$ 的長度，而差異協列定義為陣列中任兩個感測器位置的差所形成的集合。基於差異協列的角度估計器如協同陣列多信號分類，可以使用在差異協列上的資料來估計訊號源角度。經驗上來說，差異協列容易被感測器錯誤所影響，感測器錯誤會隨機發生並造成感測器無法準確地接收信號，因此錯誤的感測器會從陣列中被移除。一旦有感測器被移除，稀疏陣列便不保證能擁有辨別 $O(N^2)$ 個不相關訊號源的優點。

傳統均勻線性陣列比稀疏陣列更穩健，但其最多只能辨別 $N - 1$ 個不相關訊號源。一個強化稀疏陣列穩健性的方法是將陣列做對稱。本篇論文提出一些關於對稱陣列的性質，例如，被研究過的「廣義 1-脆弱性」之上下限。此外，我們證明將互質陣列做對稱後，它可以達到廣義 1-脆弱性的下限。

如果每個感測器失敗的機率為 p ，且互相獨立，那差異協列中間連續片段的期望值可以用機率質量函數的觀點被推導出來。此期望值是變數 p 的一元多項式。除了能量化一個陣列的穩健性之外，此期望值還適合去比較不同陣列的效能。最後透過模擬的結果，可以展現出在感測器錯誤影響下，對稱陣列的優勢。

關鍵字：稀疏陣列、差異協列、感測器錯誤、穩健性、對稱陣列、期望值





Abstract

In one-dimensional sparse arrays, multiple sensors are placed on a line with different intervals. Sparse arrays are able to distinguish $\mathcal{O}(N^2)$ uncorrelated sources with $\mathcal{O}(N)$ sensors. The reason is that the difference coarray, defined as the differences between any two sensors of an array, has a central uniform linear array (ULA) segment of length $\mathcal{O}(N^2)$. The coarray-based angle estimators such as coarray Multiple Signal Classification (MUSIC) can use the data on the difference coarray to estimate the source directions. Empirically, difference coarrays are easily influenced by sensor failures. They will occur randomly and cause the sensors not to receive the signals accurately. Therefore, the faulty sensors will be removed from the array. Once the sensors are removed, sparse arrays are not guaranteed to have the advantage of identifying the $\mathcal{O}(N^2)$ uncorrelated sources.

Traditional ULA are known to be more robust than sparse arrays, but they can only resolve at most $N - 1$ uncorrelated sources. A method that can enhance the robustness of sparse arrays is to symmetrize the array. This thesis advances some properties related to the

symmetrical arrays. For instance, the upper bound and the lower bound of the generalized 1-fragility are studied. Additionally, we prove that coprime arrays can achieve the lower bound of the generalized 1-fragility after symmetrizing them.



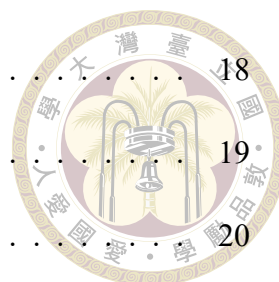
If each sensor fails independently with probability p , then the expected value of the size of the central ULA segment in the difference coarray can be derived from the view of the probability mass function (PMF). The expected value is an unary polynomial with the variable p . Besides quantifying the robustness of an array, the expected value is suitable to compare the performance of different arrays. Finally, the benefits of symmetrical arrays under the influence of sensor failures will be shown through the simulation results.

Keywords: sparse arrays, difference coarrays, sensor failures, robustness, symmetrical arrays, expected value



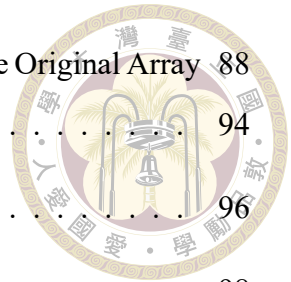
Contents

	Page
口試委員會審定書	i
致謝	iii
摘要	v
Abstract	vii
Contents	ix
List of Figures	xiii
List of Tables	xvii
Chapter 1 Introduction	1
1.1 Overview and Motivation	1
1.2 Outline of The Thesis	4
1.3 Notation	5
Chapter 2 Preliminaries	7
2.1 Data Model of Array Signal Processing	7
2.2 Difference Coarray and Weight Function	12
2.3 Review of Sparse Arrays	14
2.3.1 Minimum Redundancy Arrays (MRA)	15
2.3.2 Nested Arrays	16



2.3.3	Coprime Arrays	18
2.3.4	Uniform Linear Arrays (ULA)	19
2.3.5	Summary	20
2.4	Sensor Failure	21
2.4.1	Importance Function	22
2.5	Existing Robustness Metrics	25
2.6	Coarray-Based DOA Estimation Algorithm	29
Chapter 3	Symmetrical Arrays and Proposed Properties	35
3.1	Symmetrical Array Generation	36
3.2	Relationship between Sets and Relationship between Size of Sets	39
3.3	Properties of Importance Function and Generalized 1-fragility	45
3.4	Robustness of Symmetrical Coprime Arrays	52
3.5	Numerical Comparison	57
Chapter 4	Array Analysis with Random Sensor Failures	63
4.1	The Factors Impacting MSE	64
4.1.1	Size of \bar{U} versus MSE	64
4.1.2	Sensor Failure Probability versus MSE	66
4.2	Derivation and Comparison	70
4.2.1	Expected Value of \bar{U} Size	70
4.2.2	Proposed Robustness Metric	75
4.2.3	Numerical Results	77
4.2.3.1	The Expected Value and The Corresponding Approximation	77
4.2.3.2	Comparison of Different Arrays	84

4.2.3.3	The Symmetrical Array Compare with The Original Array	88
4.2.3.4	Random Source Direction	94
4.3	ULA Analysis	96
4.3.1	ULA with N sensors and ULA with $N + 1$ sensors	98
4.3.2	ULA with N sensors and ULA with $N + 2$ sensors	100
4.3.3	Numerical Results	102
4.4	Concluding Remarks	108
Chapter 5	Conclusion and Future Work	109
References		111

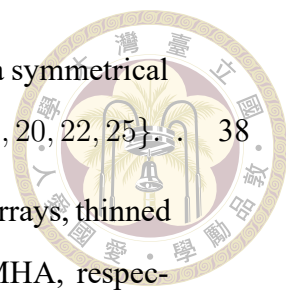






List of Figures

2.1	The system model for array signal processing.	8
2.2	The position of sensor M_0, M_1 and point source Q on the coordinate plane.	9
2.3	An illustration of array geometry \mathbb{S} , difference coarray \mathbb{D} , and the central ULA segment \mathbb{U} . The the dots are elements and the crosses are empty space. The array geometry we consider here is the coprime array with $M = 2$ and $N = 3$	12
2.4	Array geometry of the MRA with 6 sensors, and the corresponding weight function.	16
2.5	Array geometry of the nested array with $N_1 = N_2 = 3$, and the corresponding weight function.	17
2.6	Array geometry of the coprime array with $M = 2, N = 3$, and the corresponding weight function.	19
2.7	Array geometry of ULA with 6 sensors, and the corresponding weight function.	20
2.8	Two different importance functions $\mathcal{I}_{ess}(\mathbb{A})$ and $\mathcal{I}_{\mathbb{U}}(\mathbb{A})$ of every sensor with respect to (a) MRA with 6 sensors, (b) nested array with $N_1 = N_2 = 3$, (c) coprime array with $M = 2, N = 3$ and (d) ULA with 6 sensors.	24
2.9	Generalized k -fragility of four arrays (same as Figure 2.8) with two different importance functions, (a) $\mathcal{I}_{ess}(\mathbb{A})$ and (b) $\mathcal{I}_{\mathbb{U}}(\mathbb{A})$	27
2.10	P_c of four different arrays (same as Figure 2.8). The curves are depicted based on (2.31).	28
3.1	The relationship between the sets $\mathbb{S}, \mathbb{S}^r, \mathbb{S}_o, \mathbb{S}_e$ and \mathbb{S}_o^r	37



3.2 An illustration of the sets $\mathbb{S}_e, \mathbb{S}_o$ and \mathbb{S}_o^r . Here we consider a symmetrical coprime array, where $\tilde{\mathbb{S}} = \{0, 3, 5, 6, 9, 10, 12, 13, 15, 16, 19, 20, 22, 25\}$ 38

3.3 $\mathcal{D}(1, \mathcal{I})$ for six different types of arrays. They are coprime arrays, thinned coprime arrays, nested arrays, MRA, MISC arrays and MHA, respectively. The circles with the same color are the arrays generated by different parameters under the same array definition. 58

3.4 The comparison of P_c of \mathbb{S} and $\tilde{\mathbb{S}}$. We consider the array \mathbb{S} in Example 3.5.1 and plot the curves based on (2.31). 61

4.1 Estimation error versus size of $\bar{\mathbb{U}}$. $\bar{\mathbb{U}}$ is obtained from $\bar{\mathbb{S}} = \mathbb{S} \setminus \mathbb{A}$, where $\mathbb{S} = \{0, 1, 2, 5, 8, 9, 12, 15, 16, 17\}$ and $\mathbb{A} = \{n_1, n_2, n_3 \mid n_1 \neq n_2, n_1 \neq n_3, n_2 \neq n_3 \text{ and } n_1, n_2, n_3 \in \mathbb{S}\}$. SNR is 0 dB and the number of snapshots is 500. There is one source with $\theta = \pi/4$. Each point is averaged from 300 Monte-Carlo runs. The dotted line is the least square approximation of these points based on (4.4) to (4.7). 66

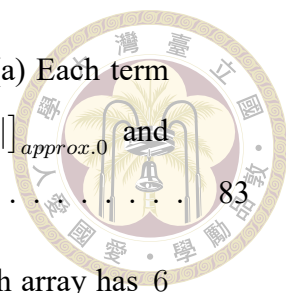
4.2 Estimation error versus probability of failure for each sensor, p . Here we consider four arrays in Table 4.1. SNR is 0 dB and the number of snapshots is 500. There is one source with $\theta = \pi/4$. Each point is averaged from $5 \cdot 10^5$ Monte-Carlo runs. 69

4.3 PMF of $|\bar{\mathbb{U}}|$ with different failure probability p . Here we consider the coprime array with $M = 2$ and $N = 3$. For each p , the number of trials is 10^5 , and the PMF is obtained by counting the occurrence of every $|\bar{\mathbb{U}}|$ through Algorithm 2. 79

4.4 (Continued from Figure 4.3). 80

4.5 The comparison of the expected value obtained from the statistical PMF and Eq.(4.11). 81

4.6 Here we consider the MRA with 6 sensors in Table 4.1. (a) Each term of k in $\mathbb{E}[|\bar{\mathbb{U}}|]$. (b) The approximations of $\mathbb{E}[|\bar{\mathbb{U}}|]$, $\mathbb{E}[|\bar{\mathbb{U}}|]_{approx.0}$ and $\mathbb{E}[|\bar{\mathbb{U}}|]_{approx.1}$ defined in (4.21). 82



4.7 Here we consider the ULA with 6 sensors in Table 4.1. (a) Each term of k in $\mathbb{E}[|\bar{\mathbf{U}}|]$. (b) The approximations of $\mathbb{E}[|\bar{\mathbf{U}}|]$, $\mathbb{E}[|\bar{\mathbf{U}}|]_{approx.0}$ and $\mathbb{E}[|\bar{\mathbf{U}}|]_{approx.1}$ defined in (4.21). 83

4.8 The comparison of $p_{90\%}$ and $0.9 \cdot |\mathbf{U}|$ of four arrays. Each array has 6 sensors and the array geometries are listed in Table 4.1. 87

4.9 The comparison of $p_{90\%}$ and $0.9 \cdot |\mathbf{U}|$ of four arrays. Each array has 10 sensors and (4.32) shows the array geometries. 87

4.10 The comparison of \mathbb{S} and $\tilde{\mathbb{S}}$ of the MRA in Table 4.2. 89

4.11 The comparison of \mathbb{S} and $\tilde{\mathbb{S}}$ of the nested array with $N_1 = 3$ and $N_2 = 3$ in Table 4.2. 89

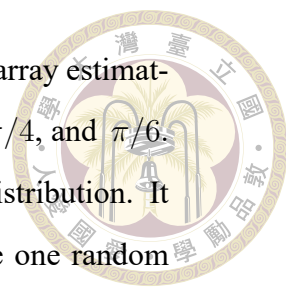
4.12 The comparison of \mathbb{S} and $\tilde{\mathbb{S}}$ of the coprime array with $M = 2$ and $N = 3$ in Table 4.2. 89

4.13 The estimation error comparison of the MRA \mathbb{S} and its symmetrical version $\tilde{\mathbb{S}}$. We estimate one source with $\theta = \pi/4$. SNR is 0 dB and the number of snapshots is 500. Each point is averaged from $5 \cdot 10^5$ Monte-Carlo runs. 91

4.14 The estimation error comparison of the nested array \mathbb{S} and its symmetrical version $\tilde{\mathbb{S}}$. We estimate one source with $\theta = \pi/4$. SNR is 0 dB and the number of snapshots is 500. Each point is averaged from $5 \cdot 10^5$ Monte-Carlo runs. 91

4.15 The estimation error comparison of the coprime array \mathbb{S} and its symmetrical version $\tilde{\mathbb{S}}$. We estimate one source with $\theta = \pi/4$. SNR is 0 dB and the number of snapshots is 500. Each point is averaged from $5 \cdot 10^5$ Monte-Carlo runs. 92

4.16 The comparison of \mathbb{S} and $\tilde{\mathbb{S}}$ of the array in (4.34). (a) The expected value curve based on (4.11). (b) Robustness metrics $p_{90\%}$ and performance metrics $0.9 \cdot |\mathbf{U}|$ 93



4.17 The estimation error comparison of the symmetrical nested array estimating different source angles. The fixed angle include $\pi/3$, $\pi/4$, and $\pi/6$. The random angle is $\theta \in [-\pi/2.5, \pi/2.5]$ with uniform distribution. It will be regenerated for every \bar{S} , so every \bar{S} will estimate one random source angle and perform 100 Monte-Carlo runs. There are 1000 \bar{S} so that each point is averaged from 10^5 Monte-Carlo runs. We set SNR to 0 dB and the number of snapshots to 500. 95

4.18 A schematic diagram we compare the ULA with different number of sensors. 97

4.19 The comparison of ULA 7 and ULA 8 of each k item in Table 4.3. . . . 104

4.20 The comparison of ULA 6 and ULA 8 of each k item in Table 4.4. . . . 105

4.21 The comparisons of $\mathbb{E} [|\bar{U}|]$ of (a) ULA with 7 sensors and 8 sensors.
(b) ULA with 6 sensors and 8 sensors. 106

4.22 The comparisons of $p_{90\%}$ and $0.9 \cdot |\bar{U}|$ of three ULA. They have 6 sensors, 7 sensors and 8 sensors, respectively. 107

4.23 The estimation error comparison of the three ULA. They have 6 sensors, 7 sensors and 8 sensors, respectively. We estimate one source with $\theta = \pi/4$. SNR is 0 dB and the number of snapshots is 500. Each point is averaged from $5 \cdot 10^5$ Monte-Carlo runs. 107



List of Tables

2.1	Array summary of this section.	21
3.1	The importance function \mathcal{I}_{ess} of each sensor in \mathbb{S} and $\tilde{\mathbb{S}}$	60
3.2	The importance function $\mathcal{I}_{\mathbb{U}}$ of each sensor in \mathbb{S} and $\tilde{\mathbb{S}}$	60
4.1	The comparison of four arrays, each of them has 6 sensors. The second column is their corresponding $ \bar{\mathbb{U}} $ when $p = 0$. The third and fourth column are their generalized 1-fragility with respect to \mathcal{I}_{ess} and $\mathcal{I}_{\mathbb{U}}$, respectively.	67
4.2	Symmetrical arrays compare with original arrays (Here \mathbb{S} are the arrays in Table 4.1 and $\tilde{\mathbb{S}}$ are generated from (3.4) and (3.5)).	88
4.3	All the components of $\mathbb{E} [\bar{\mathbb{U}}]$ of the ULA with N sensors and $N + 1$ sensors.	98
4.4	All the components of $\mathbb{E} [\bar{\mathbb{U}}]$ of the ULA with N sensors and $N + 2$ sensors.	100

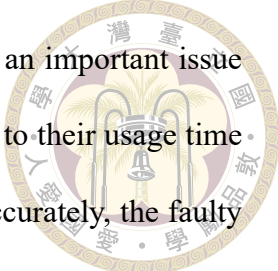




Chapter 1 Introduction

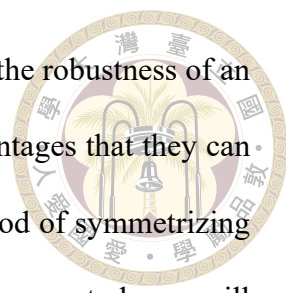
1.1 Overview and Motivation

Array signal processing is widely used in many fields such as radar [1, 2], communications [3] and acoustics [4]. Sensors in an array are used for receiving signals emitted from the sources in the environment. Through these received data, we can extract the interested information of sources by appropriate algorithms. Direction-Of-Arrival (DOA) estimation is one of the application in array signal processing, which let us obtain the angle information of the sources. Additionally, sparse arrays have been extensively discussed in recent years [5, 6], since they are able to distinguish $\mathcal{O}(N^2)$ uncorrelated sources with $\mathcal{O}(N)$ sensors. The reason is that the difference coarray, defined as the differences between any two sensors of an array, has a central uniform linear array (ULA) segment of length $\mathcal{O}(N^2)$. Then the algorithms such as coarray Multiple Signal Classification (MUSIC) [7] and coarray Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) [8], can utilize the data on the difference coarray to estimate the source directions. They are based on the orthogonality of the signal subspace and the noise subspace. A large ULA segment in the difference coarray generally increase the performance [6, 9, 10]. Hence, the size of the ULA segment is an important metric in designing sparse arrays.



However, if we consider the system reliability [11, 12], there is an important issue called “sensor failures”. Every sensor may fail randomly according to their usage time [11, 12]. Since it could cause the sensors not to receive the signals accurately, the faulty sensors will be removed from the array. Therefore, the size of the ULA segment in the difference coarray may decrease accordingly. This condition usually happens when the array has $\mathcal{O}(N^2)$ -long central ULA segment in the difference coarray such as minimum hole arrays (MHA) [13], minimum redundancy arrays (MRA) [14] and nested arrays [6]. The methods in the literature are mainly divided into two aspects to deal with the issue of sensor failures. First, creating new algorithms that are useful under the circumstance of sensor failures. Second, analyzing the robustness of different array configurations. In the first case, several approaches have been explored. For example, the authors in [15] provided a signal model with sensor failures and a corresponding method to estimate the covariance matrix. Also, the authors in [16] proposed a learning algorithm, minimal resource allocation network (MRAN), for the DOA estimation with sensor failures. On the other hand, in the second case, the authors in [17] proposed the robustness metric “fragility” to quantify the robustness of an array and they further analyzed the robustness of different sparse arrays based on the fragility in [18]. Moreover, [19] extended the fragility to “generalized k -fragility”. This definition is more general because we can define any indicators if four designated conditions are met. Under these robustness metrics, an array is more robust if the measured value closing to 0.

Empirically, sparse arrays are not robust [17], especially for the arrays which have a large difference coarray size. The authors in [20] proposed a new array geometry called robust MRA (RMRA). This array own the maximum size of the hole-free difference coarray, like the MRA. Furthermore, it is as robust as the ULA. However, the RMRA is computa-



tionally expensive to solve in general. In our study, we will enhance the robustness of an array by symmetrizing them. Symmetrical arrays also have the advantages that they can advance the performance of the DOA estimation [21, 22]. The method of symmetrizing the asymmetrical sparse array has been mentioned in [23]. However, in our study, we will further decompose the sparse array into two subarrays before symmetrizing. This action will help us to do a more in-depth analysis of the array. For example, we can provide some properties related to the generalized 1-fragility of the symmetrized array. These properties will be studied in this thesis. Moreover, we can obtain the concept that how to design robust arrays through these properties.

Since sensor failures would occur randomly [11, 12], we assume that the sensors in an array have a certain probability, p , of failure and each sensor fails independently. In [24], the authors derived an expression, P_c , for the probability that the difference coarray changes due to the sensor failure probability p . As the usage time increases, the probability p will also go up [11, 12]. Therefore, the benefit of P_c is that it can be used for comparing the robustness of arrays based on random sensor failures. However, P_c has a shortcoming that it only depends on whether the difference coarray changes. Nevertheless, the key of DOA estimation performance is the ULA segment size in the difference coarray [6, 10]. Therefore, we will combine the ULA segment size and the sensor failure probability to generate a new metric. It is a expected value of the random variable “the size of central ULA segment in the difference coarray”. Also, it is a unary polynomial with the variable p . Besides defining the robustness of an array, it can also compare the performance of different arrays. If we consider a two-dimensional point with the proposed expected value expression, then it is possible to simultaneously evaluate the robustness and the performance of different arrays.

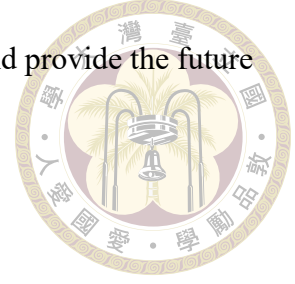
Here the main contributions of this thesis are listed.

1. We propose some properties of symmetrical sparse arrays, including the size relationship between difference coarrays, the upper bound and the lower bound of generalized 1-fragility of symmetrical arrays, and the condition of improving the robustness if an array is symmetrized.
2. We prove that the symmetrical coprime arrays can reach the lower bound of generalized 1-fragility. Namely, they are the most robust arrays based on the definition of generalized 1-fragility.
3. We propose the expected value of the random variable “the size of central ULA segment in the difference coarray” based on sensor failure probability p . We will use it for defining a robustness metric and also comparing the estimation performance of different arrays.
4. Through point 3, we further discuss the performance of the uniform linear arrays (ULA) with different number of sensors based on random sensor failures.
5. Through the simulation results, we show that it can improve the robustness and the performance at the same time after symmetrizing the array.

1.2 Outline of The Thesis

Chapter 2 reviews some critical prior knowledge, making us understand the following study comprehensively. Chapter 3 advances some useful properties related to symmetrical arrays, along with discussions and proofs. In Chapter 4, we will detail our proposed robustness metric of sparse arrays and demonstrate the advantages of symmetrical arrays

by experiments. Finally, in Chapter 5, we will conclude this thesis and provide the future direction that can be researched.



1.3 Notation

This section defines the notations that are used in this thesis. Scalars and vectors are represented as lower case (such as a) and lower case with bold symbol (such as \mathbf{a}), respectively. Moreover, matrices and sets are represented as upper case with bold symbol (such as \mathbf{A}) and blackboard boldface (such as \mathbb{A}), respectively. Additionally, \mathbb{A}^+ represents the nonnegative part of \mathbb{A} . The intersection and the union of two sets \mathbb{A} and \mathbb{B} are denoted by $\mathbb{A} \cap \mathbb{B}$ and $\mathbb{A} \cup \mathbb{B}$, respectively. The relative complement of a set \mathbb{A} with respect to a set \mathbb{B} is written as

$$\mathbb{B} \setminus \mathbb{A} = \{x \in \mathbb{B} \mid x \notin \mathbb{A}\}. \quad (1.1)$$

We use $\mathbb{E}[\cdot]$ to represent the expectation operator. For a full column rank matrix \mathbf{A} , the pseudo inverse of \mathbf{A} and the orthogonal projection onto the null space \mathbf{A}^H are defined as follows.

$$\mathbf{A}^\dagger = (\mathbf{A}\mathbf{A}^H)^{-1}\mathbf{A}^H, \quad \Pi_{\mathbf{A}}^\perp = \mathbf{I} - \mathbf{A}\mathbf{A}^\dagger. \quad (1.2)$$

For $\mathbf{A} \in \mathbb{C}^{M \times N}$, the Kronecker product between \mathbf{A} and the matrix \mathbf{B} is defined as

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} [\mathbf{A}]_{1,1} \mathbf{B} & [\mathbf{A}]_{1,2} \mathbf{B} & \cdots & [\mathbf{A}]_{1,N} \mathbf{B} \\ [\mathbf{A}]_{2,1} \mathbf{B} & [\mathbf{A}]_{2,2} \mathbf{B} & \cdots & [\mathbf{A}]_{2,N} \mathbf{B} \\ \vdots & \vdots & \ddots & \vdots \\ [\mathbf{A}]_{M,1} \mathbf{B} & [\mathbf{A}]_{M,2} \mathbf{B} & \cdots & [\mathbf{A}]_{M,N} \mathbf{B} \end{bmatrix}. \quad (1.3)$$

Assume that $\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{bmatrix}$ and $\mathbf{B} = \begin{bmatrix} \mathbf{b}_1 & \mathbf{b}_2 & \cdots & \mathbf{b}_n \end{bmatrix}$, the Khatri-Rao product between \mathbf{A} and \mathbf{B} is defined as

$$\mathbf{A} \odot \mathbf{B} = \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 & \mathbf{a}_2 \otimes \mathbf{b}_2 & \cdots & \mathbf{a}_n \otimes \mathbf{b}_n \end{bmatrix}. \quad (1.4)$$

On the other hand, we use the row vectors to represent the the matrices \mathbf{A} and \mathbf{B} , where

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_n \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \vdots \\ \mathbf{b}_n \end{bmatrix}. \quad (1.5)$$

The Face-splitting product, which also called the transposed Khatri-Rao product, is defined as

$$\mathbf{A} \bullet \mathbf{B} = \begin{bmatrix} \mathbf{a}_1 \otimes \mathbf{b}_1 \\ \mathbf{a}_2 \otimes \mathbf{b}_2 \\ \vdots \\ \mathbf{a}_n \otimes \mathbf{b}_n \end{bmatrix}. \quad (1.6)$$

Finally we define the bracket notation [10, Def.2]. Let $\mathbf{x}(t)$ be the signal column vector received by the array \mathbb{S} , where \mathbb{S} is an integer set. The square bracket $[\mathbf{x}(t)]_i$ is the i -th component of $\mathbf{x}(t)$, and the triangular bracket $\langle \mathbf{x}(t) \rangle_n$ for $n \in \mathbb{S}$ is the signal received by the sensor n . These bracket notations extending to covariance matrices $\mathbf{A} = \mathbb{E} [\mathbf{x}(t)\mathbf{x}^H(t)]$ is represented as $[\mathbf{A}]_{i,j} = \mathbb{E} [[\mathbf{x}(t)]_i[\mathbf{x}(t)]_j^*]$ and $\langle \mathbf{A} \rangle_{n_1, n_2} = \mathbb{E} [\langle \mathbf{x}(t) \rangle_{n_1} \langle \mathbf{x}(t) \rangle_{n_2}^*]$ for $n_1, n_2 \in \mathbb{S}$.



Chapter 2 Preliminaries

In this chapter, Section 2.1 will introduce the data model of array signal processing. The definition and use of the difference coarray and the weight function will be mentioned in Section 2.2. Several common sparse arrays will be introduced in Section 2.3. Sensor failure is an important issue considered by this thesis. It will be presented in Section 2.4. Regarding how to quantify the robustness of an array, some definitions have been proposed in the previous studies [17, 19], and they will be introduced in Section 2.5. Finally, in Section 2.6, we will introduce how to use the difference coarray to implement DOA estimation [3, 10]. Coarray-based MUSIC is the main algorithm used by this thesis.

2.1 Data Model of Array Signal Processing

In this section, [3] is taken as a reference. Figure 2.1 illustrates the model of one-dimensional array signal processing. Red solid circles represent the physical sensors that used for receiving signals. All sensors are placed on the x -axis, which consist of the sensor array. Multiple sources emit propagating waves from different directions in the environment. After the sensor array receives the signals, we can get the information of interest through appropriate algorithms such as the source angle and the source distance.

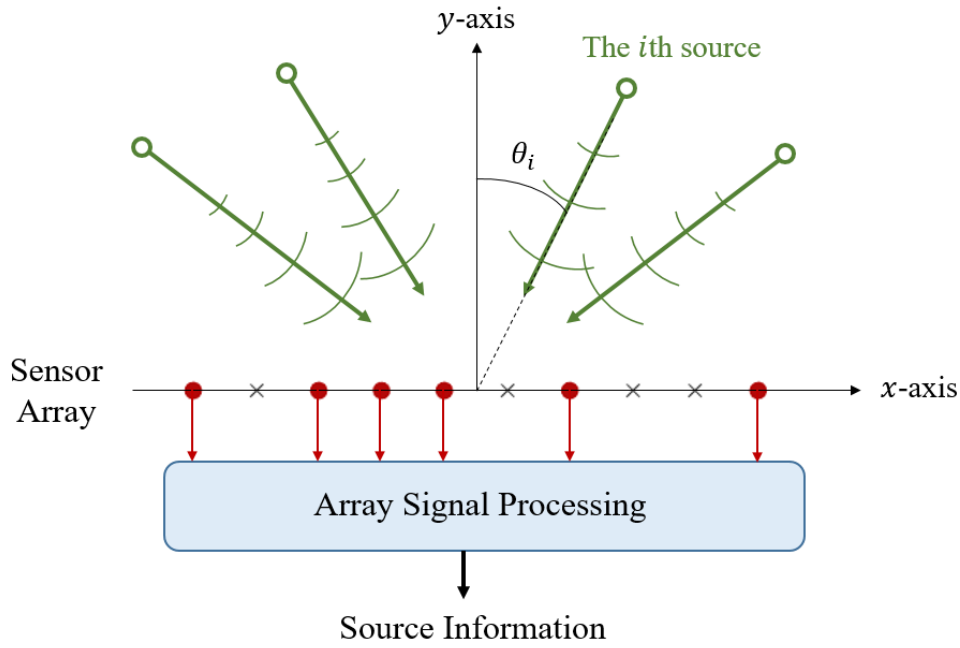


Figure 2.1: The system model for array signal processing.

Now suppose that there are two sensors M_0 and M_1 , and the distance between them is d (as shown in Figure 2.2). There is a point source at position Q which is at a distance of R from M_0 , so the distance from M_1 to Q is $\overline{M_1Q} = [(d - R \sin \theta)^2 + (R \cos \theta)^2]^{1/2}$. Assume that the source emits the waveform $s_Q(t) = Ae^{j2\pi ft}$, propagation velocity is c and wavelength $\lambda = c/f$. Also, there is no path loss. Then the waveform at M_0 is

$$s_{M_0}(t) = s_Q\left(t - \frac{\overline{M_0Q}}{c}\right) = Ae^{j2\pi f\left(t - \frac{\overline{M_0Q}}{c}\right)}. \quad (2.1)$$

We need to assume that the point source Q is at far-field, $d \ll R$, so the wavefront can be approximated by plane waves. Thus, at M_1 , the waveform becomes

$$s_{M_1}(t) = Ae^{j2\pi f\left(t - \frac{\overline{M_1Q}}{c}\right)} = Ae^{j2\pi f\left(t - \frac{\overline{M_0Q}}{c}\right)} e^{j\frac{2\pi}{\lambda}(\overline{M_0Q} - \overline{M_1Q})} = s_{M_0}(t) e^{j\frac{2\pi}{\lambda}(\overline{M_0Q} - \overline{M_1Q})}. \quad (2.2)$$

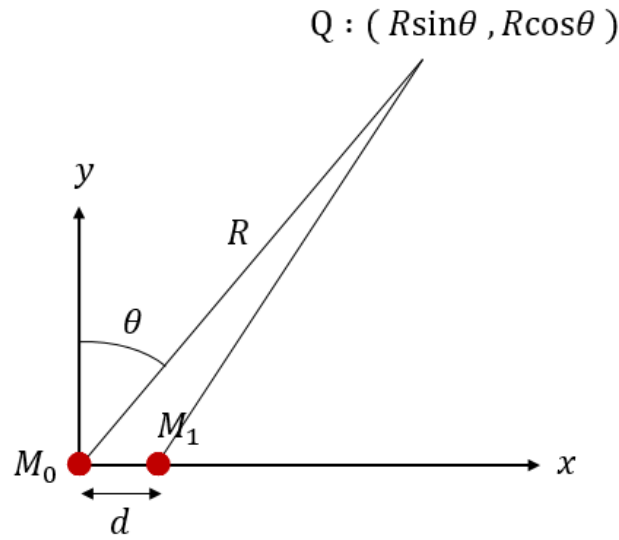


Figure 2.2: The position of sensor M_0 , M_1 and point source Q on the coordinate plane.

There is a phase difference between the signals received by these two different sensors, and the phase depends on $\overline{M_0Q} - \overline{M_1Q}$. It can be simplified as follows.

$$\begin{aligned}
 \overline{M_0Q} - \overline{M_1Q} &= R - [(d - R \sin \theta)^2 + (R \cos \theta)^2]^{\frac{1}{2}} \\
 &= R \left\{ 1 - \left[\left(\sin \theta - \frac{d}{R} \right)^2 + (\cos \theta)^2 \right]^{\frac{1}{2}} \right\} \\
 &= R \left\{ 1 - \left[1 - \frac{2d \sin \theta}{R} + \left(\frac{d}{R} \right)^2 \right]^{\frac{1}{2}} \right\} \\
 &\approx R \left\{ 1 - \left[1 + \frac{1}{2} \left(-\frac{2d \sin \theta}{R} + \left(\frac{d}{R} \right)^2 \right) \right] \right\} \tag{2.3} \\
 &\approx R \left\{ 1 - \left[1 + \frac{1}{2} \left(-\frac{2d \sin \theta}{R} \right) \right] \right\} \\
 &= R \cdot \frac{1}{2} \cdot \frac{2d \sin \theta}{R} \\
 &= d \sin \theta.
 \end{aligned}$$

Therefore, we can change (2.2) to

$$s_{M_1}(t) = s_{M_0}(t) e^{j2\pi \left(\frac{d}{\lambda} \sin \theta \right)}. \tag{2.4}$$

Eq.(2.4) demonstrates that the signal received by the sensors M_0 and M_1 is related to the source angle θ and the wavelength λ . Note that the distance d cannot exceed $\lambda/2$ because of spatial ambiguity [25]. Hence, in this thesis, d is always set as

$$d = \frac{\lambda}{2}. \quad (2.5)$$

In the previous description, we only used two sensors. The same concept can be generalized to an array with N sensors which are located at

$$M_0 : (d_0, 0), \quad M_1 : (d_1, 0), \quad \dots, \quad M_{N-1} : (d_{N-1}, 0). \quad (2.6)$$

The signal received by the array is represented by a vector $\mathbf{x}(t)$ which contains the noise term $\mathbf{n}(t)$. They are denoted by

$$\mathbf{x}(t) = \begin{bmatrix} x_{M_0}(t) \\ x_{M_1}(t) \\ \vdots \\ x_{M_{N-1}}(t) \end{bmatrix}, \quad \mathbf{n}(t) = \begin{bmatrix} n_{M_0}(t) \\ n_{M_1}(t) \\ \vdots \\ n_{M_{N-1}}(t) \end{bmatrix}. \quad (2.7)$$

Therefore, we can get the signal data model $\mathbf{x}(t)$ received by the array,

$$\mathbf{x}(t) = \begin{bmatrix} x_{M_0}(t) \\ x_{M_1}(t) \\ \vdots \\ x_{M_{N-1}}(t) \end{bmatrix} = s_{M_0}(t) \begin{bmatrix} e^{j2\pi(\frac{d_0}{\lambda} \sin \theta)} \\ e^{j2\pi(\frac{d_1}{\lambda} \sin \theta)} \\ \vdots \\ e^{j2\pi(\frac{d_{N-1}}{\lambda} \sin \theta)} \end{bmatrix} + \begin{bmatrix} n_{M_0}(t) \\ n_{M_1}(t) \\ \vdots \\ n_{M_{N-1}}(t) \end{bmatrix} = s_{M_0}(t) \mathbf{a}(\theta) + \mathbf{n}(t), \quad (2.8)$$

where $\mathbf{a}(\theta)$ is called the steering vector.

In reality, it is possible that multiple sources exist at the same time, so here we as-

sume that there are D sources: $s_1(t), \dots, s_D(t)$. The incident angle of the i th source is denoted by θ_i , and these angles satisfy $-\frac{\pi}{2} \leq \theta_i \leq \frac{\pi}{2}$. All of the sources are monochromatic, which means all of them have the same frequency f . Then the data model for the array output vector $\mathbf{x}(t)$ can be represented as

$$\mathbf{x}(t) = \begin{bmatrix} \mathbf{a}(\theta_1) & \dots & \mathbf{a}(\theta_D) \end{bmatrix} \begin{bmatrix} s_1(t) \\ \vdots \\ s_D(t) \end{bmatrix} + \mathbf{n}(t) = \mathbf{A}(\boldsymbol{\theta})\mathbf{s}(t) + \mathbf{n}(t), \quad (2.9)$$

where $\mathbf{A}(\boldsymbol{\theta})$ is called the array manifold matrix. From (2.9), we can find that the time and angle information of the sources are separated. This property is favorable for signal processing algorithms. In this thesis, we will use a integer set \mathbb{S} to represent the array geometry. Assume that the sensor locations belong to a uniform grid of distance d . That is, they can be modeled by nd , where $n \in \mathbb{S}$. Finally, we look at an example.

Example 2.1.1. Here we consider the array geometry in Figure 2.3, $\mathbb{S} = \{0, 2, 3, 4, 6, 9\}$. There are D monochromatic sources with the same wavelength λ , which emit the signal to the array from far-field. If the spacing d is equal to $\lambda/2$, then the output signals of each sensor can be expressed as

$$\begin{bmatrix} \langle \mathbf{x}(t) \rangle_0 \\ \langle \mathbf{x}(t) \rangle_2 \\ \langle \mathbf{x}(t) \rangle_3 \\ \langle \mathbf{x}(t) \rangle_4 \\ \langle \mathbf{x}(t) \rangle_6 \\ \langle \mathbf{x}(t) \rangle_9 \end{bmatrix} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ e^{j\pi \sin \theta_1 \cdot 2} & e^{j\pi \sin \theta_2 \cdot 2} & \dots & e^{j\pi \sin \theta_D \cdot 2} \\ e^{j\pi \sin \theta_1 \cdot 3} & e^{j\pi \sin \theta_2 \cdot 3} & \dots & e^{j\pi \sin \theta_D \cdot 3} \\ e^{j\pi \sin \theta_1 \cdot 4} & e^{j\pi \sin \theta_2 \cdot 4} & \dots & e^{j\pi \sin \theta_D \cdot 4} \\ e^{j\pi \sin \theta_1 \cdot 6} & e^{j\pi \sin \theta_2 \cdot 6} & \dots & e^{j\pi \sin \theta_D \cdot 6} \\ e^{j\pi \sin \theta_1 \cdot 9} & e^{j\pi \sin \theta_2 \cdot 9} & \dots & e^{j\pi \sin \theta_D \cdot 9} \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_D(t) \end{bmatrix} + \begin{bmatrix} \langle \mathbf{n}(t) \rangle_0 \\ \langle \mathbf{n}(t) \rangle_2 \\ \langle \mathbf{n}(t) \rangle_3 \\ \langle \mathbf{n}(t) \rangle_4 \\ \langle \mathbf{n}(t) \rangle_6 \\ \langle \mathbf{n}(t) \rangle_9 \end{bmatrix}. \quad (2.10)$$

The triangular bracket was mentioned in Section 1.3.

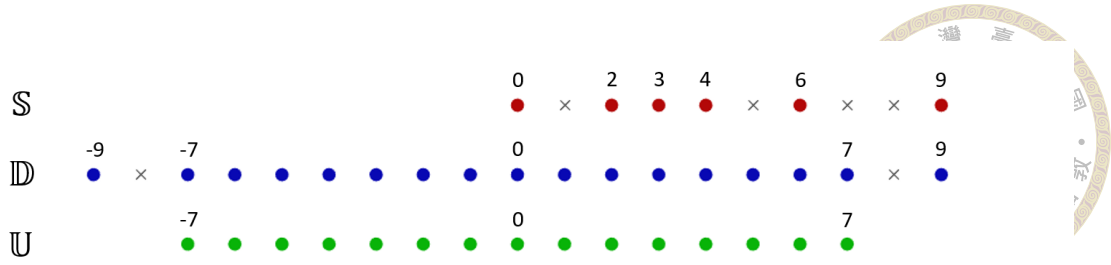


Figure 2.3: An illustration of array geometry \mathbb{S} , difference coarray \mathbb{D} , and the central ULA segment \mathbb{U} . The the dots are elements and the crosses are empty space. The array geometry we consider here is the coprime array with $M = 2$ and $N = 3$.

2.2 Difference Coarray and Weight Function

In sparse array design related papers, difference coarray plays a fundamental and important role, since the signals received by the array \mathbb{S} can be converted to the statistical signals on the difference coarray \mathbb{D} [7, 10]. The definition of the difference coarray is shown as follows.

Definition 2.2.1. [17, Def.1] The difference coarray \mathbb{D} of an array \mathbb{S} is defined as a set generated by the differences between the sensor locations. Namely,

$$\mathbb{D} = \{n_1 - n_2 \mid n_1, n_2 \in \mathbb{S}\}. \quad (2.11)$$

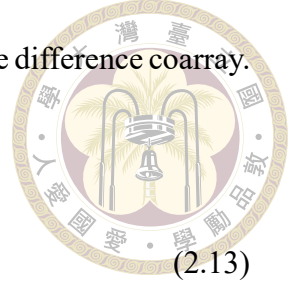
We take the coprime array with $M = 2$ and $N = 3$ as an example. Figure 2.3 illustrates the array geometry (red points) and the corresponding difference coarray (blue points). It can be found that there exists holes, -8 and 8 , in the difference coarray. Therefore, the set \mathbb{U} (green points) is defined as the central ULA segment in the difference coarray. It can be denoted by

$$\mathbb{U} = \{m \mid \{0, 1, \dots, |m|\} \subseteq \mathbb{D}\}. \quad (2.12)$$

If $\mathbb{D} = \mathbb{U}$, then we can say that the difference coarray is a hole-free difference coarray.

Furthermore, the set of holes is defined as

$$\mathbb{H} = \{m \mid \min(\mathbb{D}) \leq m \leq \max(\mathbb{D}), m \notin \mathbb{D}\}. \quad (2.13)$$



In addition, difference coarray has two special cases [18], the self difference and the cross difference. They are defined as follows.

Definition 2.2.2. The self difference of a set \mathbb{Q} is denoted by

$$SD(\mathbb{Q}) = \{m_1 - m_2 \mid m_1, m_2 \in \mathbb{Q}\}, \quad (2.14)$$

and the cross difference between two sets \mathbb{Q}_1 and \mathbb{Q}_2 is denoted by

$$CD(\mathbb{Q}_1, \mathbb{Q}_2) = \{q_1 - q_2 \mid q_1 \in \mathbb{Q}_1, q_2 \in \mathbb{Q}_2\}. \quad (2.15)$$

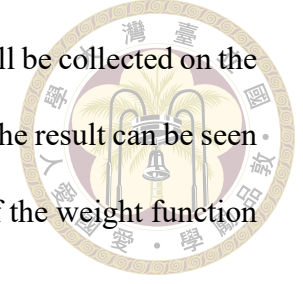
Among several commonly used coarray based DOA estimators such as coarray MUSIC [7] and coarray ESPRIT [8], the data measured by the difference coarray does not be used completely. In fact, only the data on \mathbb{U} will be used. Therefore, the central ULA segment in the difference coarray \mathbb{U} is a focus of the analysis in this thesis.

In Definition 2.2.1, the difference coarray \mathbb{D} is an integer set. However, it is possible that multiple sensor pairs in \mathbb{S} generate the same difference. Thus, here we define the weight function.

Definition 2.2.3. [17, Def.2] The weight function $w(m)$ of an array \mathbb{S} is defined as the number of sensor pairs with coarray index m . That is,

$$w(m) = |\{(n_1, n_2) \in \mathbb{S}^2 \mid n_1 - n_2 = m\}|. \quad (2.16)$$

When the weight function is larger, there are more signal data will be collected on the coarray index m . It also could impact the estimation performance. The result can be seen through the experiments in Chapter 4. Here we list four properties of the weight function which are provided by Pal [6].



Corollary 2.2.1. An array with N sensors is considered. Following properties of weight function are related to its difference coarray \mathbb{D} .

1. $w(0) = N$.
2. $1 \leq w(m) \leq N - 1 \quad \forall m \in \mathbb{D} \setminus \{0\}$.
3. $w(m) = w(-m) \quad \forall m \in \mathbb{D}$.
4. $\sum_{m \in \mathbb{D}, m \neq 0} w(m) = N(N - 1)$.

We can use the view of permutation to explain Property 4 in Corollary 2.2.1. The sum of all occurrences of differences except for $m = 0$, is equal to $N!/(N - 2)!$. This is all possible permutations of two elements from an array set \mathbb{S} with N sensors. According to this property, we can know the maximum degrees of freedom that can be obtained from a difference coarray. Therefore, if an array geometry is properly designed, it is possible that we get $O(N^2)$ degrees of freedom using only $O(N)$ physical sensors. The example arrays will be introduced in the next section. Also, we will see their weight functions with figures.

2.3 Review of Sparse Arrays

In this section, we will review four different sparse arrays that will be discussed in Chapter 3 and Chapter 4.



2.3.1 Minimum Redundancy Arrays (MRA)

MRA were first proposed by Moffet [14]. The purpose is to minimize the redundancy R of the array, where R is defined as

$$R = \frac{\binom{|S|}{2}}{(|U| - 1)/2} = \frac{\binom{|S|}{2}}{\max(U)}. \quad (2.17)$$

The definition of MRA is shown below.

Definition 2.3.1. The MRA with N physical sensors can be defined as [14]

$$S_{\text{MRA}} = \arg \max_{\mathbb{S}} |\mathbb{D}| \quad \text{subject to } |\mathbb{S}| = N, \quad \mathbb{D} = \mathbb{U}. \quad (2.18)$$

This equation indicates that MRA have the largest hole-free difference coarray for a given number of sensors. Furthermore, the corresponding \mathbb{D} and \mathbb{U} can be denoted by

$$\mathbb{D}_{\text{MRA}} = \mathbb{U}_{\text{MRA}} = \{0, \pm 1, \pm 2, \dots, \pm (\max(S_{\text{MRA}}) - \min(S_{\text{MRA}}))\}. \quad (2.19)$$

However, the disadvantage is that when the number of sensors increases, the complexity of solving this optimization problem will also increase. That will cause complicated execution. Here we look at an example of MRA.

Example 2.3.1. A MRA with 6 elements [14] is considered. $S_{\text{MRA}} = \{0, 1, 4, 5, 11, 13\}$. Figure 2.4 illustrates its geometry and the weight function. It can be observed that except for $w(\pm 1) = w(\pm 4) = 2$, all the weight functions at the other coarray index m are equal to 1. That means only one sensor pair in \mathbb{S} can generate this difference, and that is the reason why MRA can minimize the redundancy.

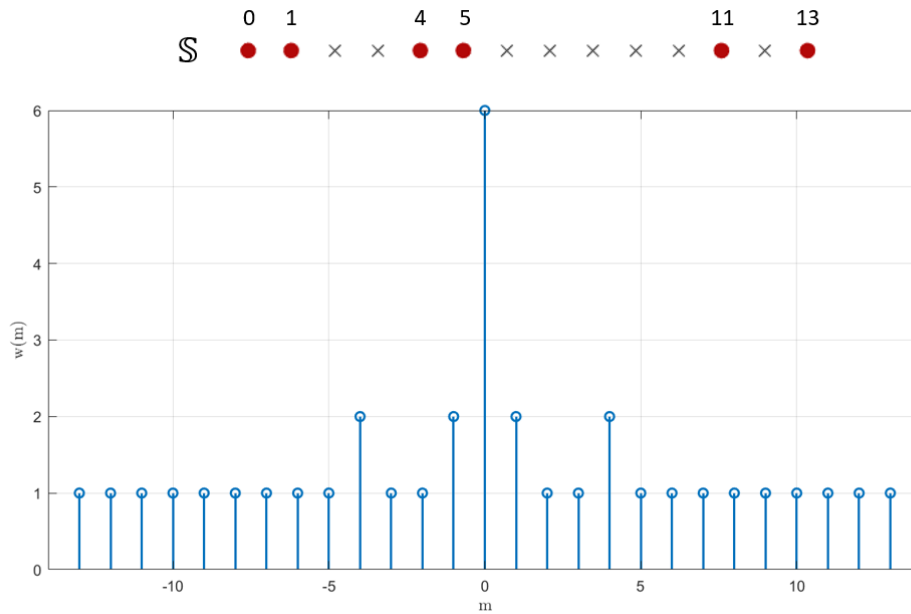
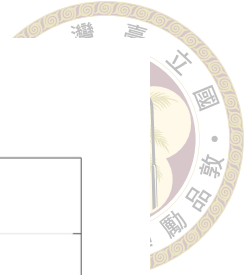


Figure 2.4: Array geometry of the MRA with 6 sensors, and the corresponding weight function.

2.3.2 Nested Arrays

Unlike MRA, the sensor locations of nested arrays can be represented as a closed form [6]. It does not require a lot of computation to obtain the array geometry. Also, the configuration is easily scalable. We can obtain the array geometry as long as the parameters N_1 and N_2 are given. The definition of nested arrays is shown below.

Definition 2.3.2. Assume that N_1 and N_2 are positive integers, the sensor locations of nested arrays [6] are

$$\mathbb{S}_{\text{nested}} = \mathbb{G}_1 \cup \mathbb{G}_2, \tag{2.20}$$

$$\text{where } \mathbb{G}_1 = \{1, 2, \dots, N_1\}, \quad \mathbb{G}_2 = \{n(N_1 + 1) \mid n = 1, 2, \dots, N_2\}.$$

The number of sensors is $N_1 + N_2$. Based on (2.20), the geometry of nested arrays is composed of a dense ULA, \mathbb{G}_1 , with spacing 1 and a sparse ULA, \mathbb{G}_2 , with spacing

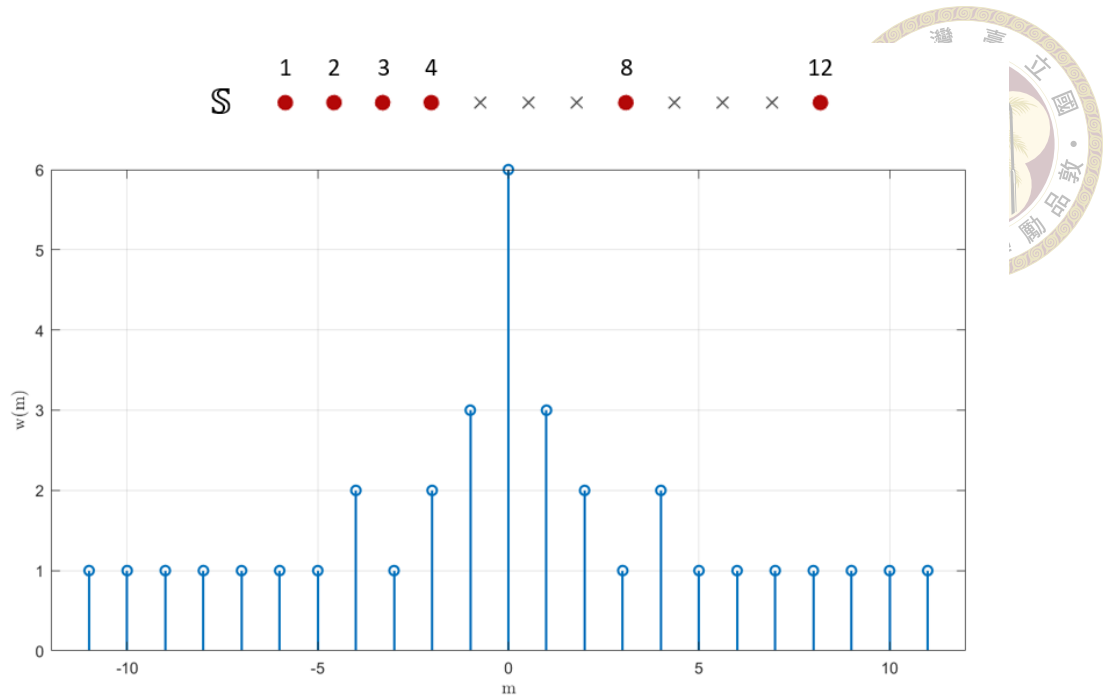


Figure 2.5: Array geometry of the nested array with $N_1 = N_2 = 3$, and the corresponding weight function.

$N_1 + 1$. The corresponding \mathbb{D} and \mathbb{U} are equal, and they can be represented with N_1 and N_2 , like

$$\mathbb{D}_{\text{nested}} = \mathbb{U}_{\text{nested}} = \{0, \pm 1, \pm 2, \dots, \pm(N_2(N_1 + 1) - 1)\}. \quad (2.21)$$

Given N sensors, if both N_1 and N_2 are approximately equal to $N/2$, the size of the difference coarray can be shown that $|\mathbb{D}_{\text{nested}}| = \mathcal{O}(N^2)$ [6]. Thus, nested arrays are possible to identify $\mathcal{O}(N^2)$ uncorrelated sources with $\mathcal{O}(N)$ physical sensors. This property is similar to MRA. Here we look at an example of nested arrays.

Example 2.3.2. We consider the nested array with $N_1 = N_2 = 3$. The sensor locations can be obtained based on (2.20) that $\mathbb{S}_{\text{nested}} = \{1, 2, 3, 4, 8, 12\}$. Its weight function is shown in Figure 2.5. Since there is a dense part in the nested array, it can be observed that the weight function at $m = 1$ and $m = -1$ are higher than the other weight function except for $m = 0$.



2.3.3 Coprime Arrays

The sensor locations of coprime arrays can also be expressed as a closed form with two coprime integers M and N . They were first proposed in [5]. In recent research about sparse array signal processing, coprime arrays have obtained great interest [9, 26, 27]. The definition of coprime arrays is shown below.

Definition 2.3.3. Assume that M and N are two positive and coprime integers, the sensor locations of coprime arrays [5] are

$$\mathbb{S}_{\text{coprime}} = \mathbb{F}_1 \cup \mathbb{F}_2, \tag{2.22}$$

where $\mathbb{F}_1 = \{0, M, 2M, \dots, (N - 1)M\}$, $\mathbb{F}_2 = \{0, N, 2N, \dots, (2M - 1)N\}$.

The number of sensors is $N + 2M - 1$. Based on (2.22), the geometry of coprimes arrays is composed of two sparse ULA with spacing M and N , respectively. Although $\mathbb{D}_{\text{coprime}}$ is not hole-free, its central ULA segment can be represented as a closed form [5], where

$$\mathbb{U}_{\text{coprime}} = \{0, \pm 1, \dots, \pm(MN + M - 1)\}. \tag{2.23}$$

Thus, there must exist holes at $\pm(MN + M)$. Coprime arrays can provide $\mathcal{O}(MN)$ degrees of freedom and only need $\mathcal{O}(M + N)$ physical sensors. Here we look at an example.

Example 2.3.3. We can use the parameters $M = 2$ and $N = 3$ to obtain a coprime array with 6 sensors. The sensor locations are denoted by $\mathbb{S}_{\text{coprime}} = \{0, 2, 3, 4, 6, 9\}$. In Figure 2.6, the geometry of $\mathbb{U}_{\text{coprime}}$ is equal to $\{-7, \dots, 0, \dots, 7\}$, which consists with Eq.(2.23).

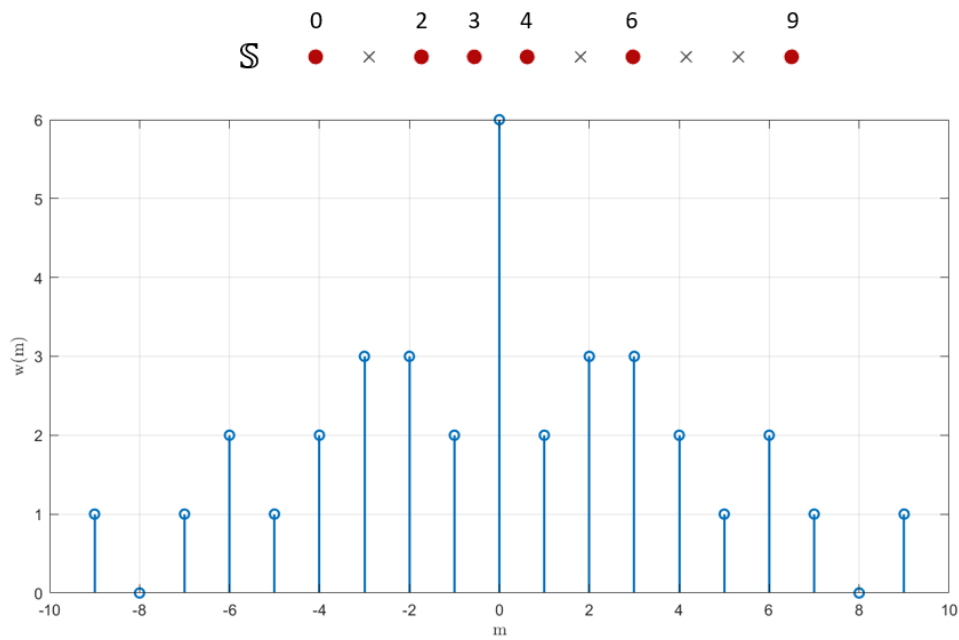
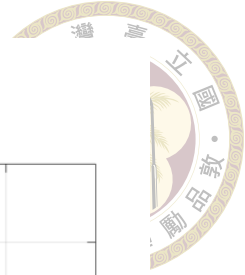


Figure 2.6: Array geometry of the coprime array with $M = 2, N = 3$, and the corresponding weight function.

2.3.4 Uniform Linear Arrays (ULA)

ULA are one of the most widely used arrays. The ULA with N sensors are defined as [3]

$$\mathbb{S}_{\text{ULA}} = \{0, 1, \dots, N - 1\}. \quad (2.24)$$

Moreover, the difference coarray of ULA can be expressed as

$$\mathbb{D}_{\text{ULA}} = \{0, \pm 1, \dots, \pm (N - 1)\}, \quad (2.25)$$

and $\mathbb{U}_{\text{ULA}} = \mathbb{D}_{\text{ULA}}$. This property indicates that ULA can identify at most $N - 1$ uncorrelated sources with N physical sensors. Figure 2.7 is an example of ULA with 6 sensors. We can find that the weight function of the ULA has two properties:

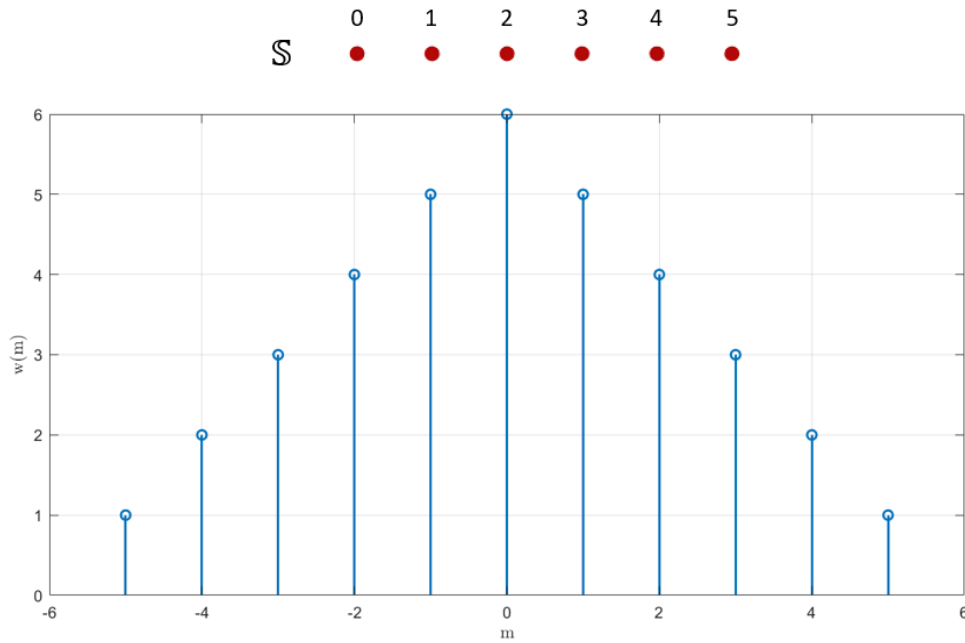
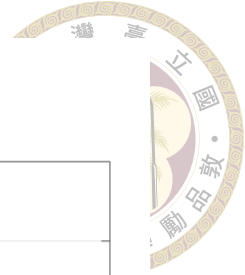
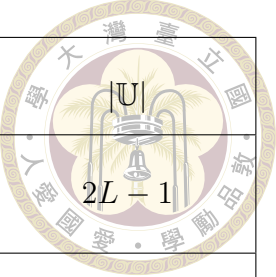


Figure 2.7: Array geometry of ULA with 6 sensors, and the corresponding weight function.

1. $w(\pm m) = N - m$, for $0 \leq m \leq N - 1$.
2. $w(m) - 1 = w(m + 1)$, for $0 \leq m \leq N - 2$.

2.3.5 Summary

Finally, we use Table 2.1 to make the summary. This table contains the size of the array, the size of the difference coarray and the size of central ULA segment in the difference coarray. We denote these values by the given parameters. For more details of these arrays, the authors in [18] had a lot of discussion and analysis.



Array	Description	$ \mathbb{S} $	$ \mathbb{D} $	$ \mathbb{U} $
(a)	MRA with N sensors, $\min(\mathbb{S}_{MRA})$ $+\max(\mathbb{S}_{MRA}) = L$	N	$2L - 1$	$2L - 1$
(b)	Nested array with N_1 and N_2	$N_1 + N_2$	$2N_2(N_1 + 1) - 1$	$2N_2(N_1 + 1) - 1$
(c)	Coprime array with M and N	$2M + N - 1$	\times	$2M(N + 1) - 1$
(d)	ULA N sensors	N	$2N - 1$	$2N - 1$

Table 2.1: Array summary of this section.

2.4 Sensor Failure

Most electronic devices demonstrate a constant failure rate during their useful lifetime [11, 12]. If applying this concept to our study on sparse arrays, every sensor may fail randomly. We assume that the sensor cannot receive the signals accurately. Therefore, the faulty sensors will be removed from the array. This issue may lead to a decrease in the estimation performance because the difference coarrays are possible to be changed after removing sensors. Thus, this is an important issue we need to consider when implementing the DOA estimation with sparse arrays.

In the following, we will introduce two definitions that are proposed in [17] to identify the set of faulty sensors.

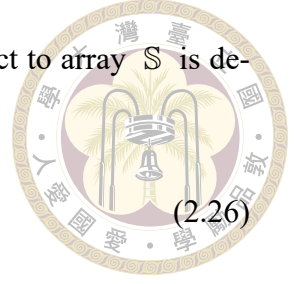
Definition 2.4.1. [17, Def.4] A subarray \mathbb{A} of array \mathbb{S} is said to be k -essential with respect to \mathbb{S} if the following properties are satisfied.

1. The size of \mathbb{A} is equal to k .
2. The difference coarray changes when \mathbb{A} is removed from \mathbb{S} .

Definition 2.4.2. [17, Def.5] The k -essential family \mathcal{E}_k with respect to array \mathbb{S} is defined as

$$\mathcal{E}_k = \{\mathbb{A} \mid \mathbb{A} \text{ is } k\text{-essential with respect to } \mathbb{S}\}, \quad (2.26)$$

where $k \in \{1, 2, \dots, |\mathbb{S}|\}$.



2.4.1 Importance Function

From the description above, we know that the sensors in an array have a certain probability of causing defects as the usage time increases. Therefore, we must remove the faulty sensors to keep the array running. The lack of different sensors in the array has different effects, so the author in [19] proposed the importance function to quantify the impact of the removed sensors (or subsets). We can define any indicator ourselves, as long as four properties are met. The definition of the importance function is shown as follows.

Definition 2.4.3. [19, Def.4] A function \mathcal{I} is said to be an importance function with respect to the array \mathbb{S} if it can satisfy all the following properties.

1. $0 \leq \mathcal{I}(\mathbb{A}) \leq 1$ for all $\mathbb{A} \subseteq \mathbb{S}$.
2. $\mathcal{I}(\emptyset) = 0$, where \emptyset is the empty set.
3. $\mathcal{I}(\mathbb{S}) = 1$.
4. \mathcal{I} is monotone. That is, if $\mathbb{A} \subseteq \mathbb{B} \subseteq \mathbb{S}$, then $\mathcal{I}(\mathbb{A}) \leq \mathcal{I}(\mathbb{B})$.

If $\mathcal{I}(\mathbb{A}) = 1$, then we can say that the subset \mathbb{A} is the most important. Large importance will cause significant impact on the array. Conversely, if $\mathcal{I}(\mathbb{A}) = 0$, then

the subset \mathbb{A} is the least important. It will not cause any impact on the array when it is removed.



Two different examples of the importance function are provided in [19]. One is related to the k -essentialness property and the other one is related to the size of \mathbb{U} . First, the importance function corresponds to k -essentialness property is defined as

$$\mathcal{I}_{ess}(\mathbb{A}) = \begin{cases} 1, & \text{if } \mathbb{A} \text{ is } |\mathbb{A}|\text{-essential} \\ 0, & \text{otherwise} \end{cases}, \quad (2.27)$$

where $\mathbb{A} \subseteq \mathbb{S}$.

Second, the importance function corresponds to the size \mathbb{U} is defined as

$$\mathcal{I}_{\mathbb{U}}(\mathbb{A}) = 1 - \frac{|\bar{\mathbb{U}}|}{|\mathbb{U}|}, \quad (2.28)$$

where $\mathbb{A} \subseteq \mathbb{S}$. The set $\bar{\mathbb{U}}$ corresponds to $\bar{\mathbb{D}}$, where $\bar{\mathbb{D}}$ is the difference coarray of $\bar{\mathbb{S}} = \mathbb{S} \setminus \mathbb{A}$.

Both of two indicators satisfy the properties of Definition 2.4.3, whose details are shown in [19]. Figure 2.8 shows two different importance functions with four different arrays. We can find a property in Figure 2.8 that these two importance functions satisfy $\mathcal{I}_{\mathbb{U}}(\mathbb{A}) \leq \mathcal{I}_{ess}(\mathbb{A})$. Based on (2.27), the importance function $\mathcal{I}_{ess}(\mathbb{A})$ is either 1 or 0. If $\mathcal{I}_{ess}(\mathbb{A}) = 1$, then the inequality holds true due to Property 1 in Definition 2.4.3 that $\mathcal{I}_{\mathbb{U}}(\mathbb{A})$ must be less than 1. Furthermore, if $\mathcal{I}_{ess}(\mathbb{A}) = 0$, then both the difference coarray and the corresponding ULA segment will not be changed when \mathbb{A} is removed. It means that $\bar{\mathbb{U}} = \mathbb{U}$. According to (2.28), $\mathcal{I}_{\mathbb{U}}(\mathbb{A}) = 0$. Thus, $\mathcal{I}_{\mathbb{U}}(\mathbb{A}) \leq \mathcal{I}_{ess}(\mathbb{A})$.

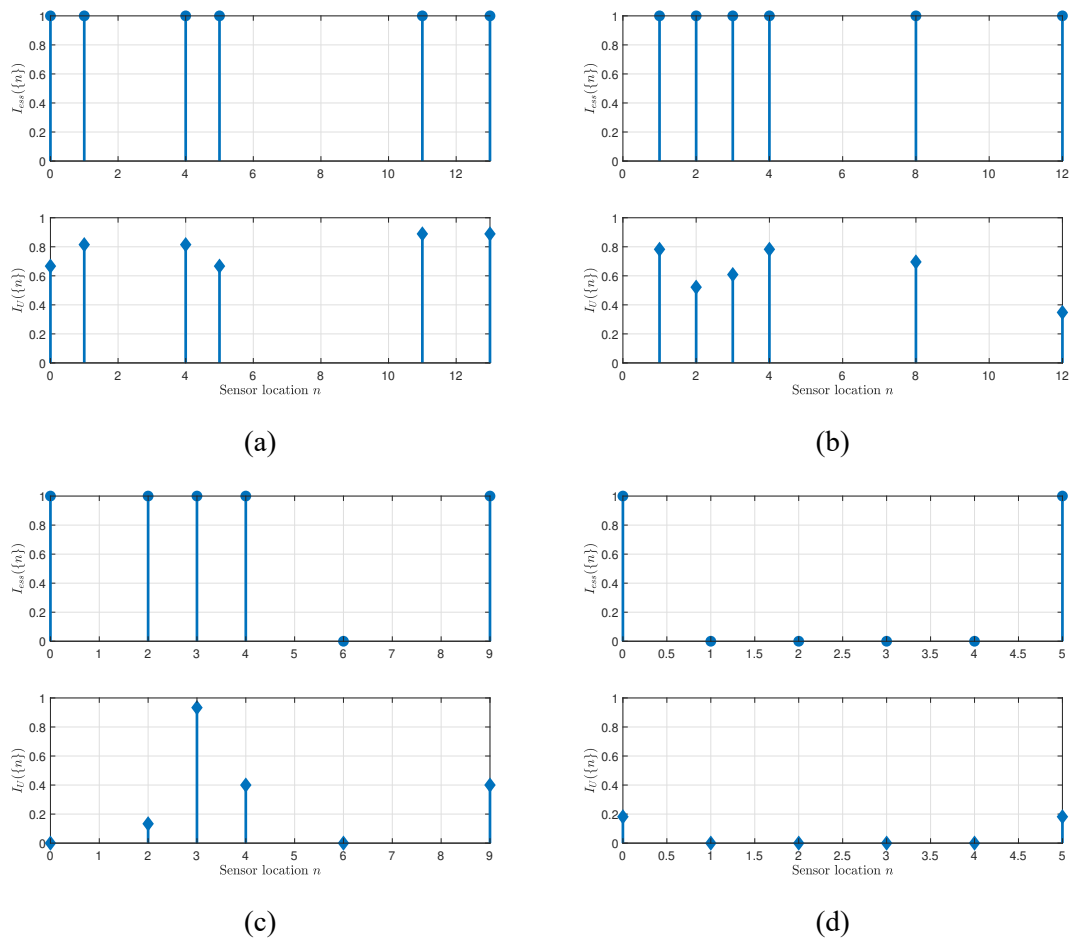


Figure 2.8: Two different importance functions $\mathcal{I}_{ess}(\mathbb{A})$ and $\mathcal{I}_U(\mathbb{A})$ of every sensor with respect to (a) MRA with 6 sensors, (b) nested array with $N_1 = N_2 = 3$, (c) coprime array with $M = 2, N = 3$ and (d) ULA with 6 sensors.



2.5 Existing Robustness Metrics

In this section, we will introduce a metric called generalized k -fragility. It was proposed in [19] to quantify the whole array robustness with a specified importance function. The definition is shown below.

Definition 2.5.1. [19, Def.6] The generalized k -fragility $\mathcal{F}_k(\mathbb{S}, \mathcal{I})$ with respect to the array \mathbb{S} and the importance function \mathcal{I} is defined as

$$\mathcal{F}_k(\mathbb{S}, \mathcal{I}) = \sum_{\mathbb{A} \subseteq \mathbb{S}, |\mathbb{A}|=k} \frac{\mathcal{I}(\mathbb{A})}{\binom{|\mathbb{S}|}{k}}, \quad (2.29)$$

for $k = 0, 1, \dots, |\mathbb{S}|$.

An array is said to be more robust if $\mathcal{F}_k(\mathbb{S}, \mathcal{I})$ is close to 0, and less robust if $\mathcal{F}_k(\mathbb{S}, \mathcal{I})$ is close to 1. From (2.29), we know that every subset \mathbb{A} with size $|\mathbb{A}| = k$ in \mathbb{S} will be considered once when calculating $\mathcal{F}_k(\mathbb{S}, \mathcal{I})$. If all the importance $\mathcal{I}(\mathbb{A})$ are close to 1, then the answer on the right-hand side of (2.29) will also be close to 1.

Several properties of $\mathcal{F}_k(\mathbb{S}, \mathcal{I})$ are provided in [19], and they are shown as follows.

1. $0 \leq \mathcal{F}_k(\mathbb{S}, \mathcal{I}) \leq 1$ for $k = 0, 1, \dots, |\mathbb{S}|$.
2. $\mathcal{F}_0(\mathbb{S}, \mathcal{I}) = 0$ and $\mathcal{F}_{|\mathbb{S}|}(\mathbb{S}, \mathcal{I}) = 1$.
3. $\mathcal{F}_k(\mathbb{S}, \mathcal{I})$ is an increasing function of k .

Note that these properties are suitable for any importance functions \mathcal{I} .

Now we use $\mathcal{F}_k(\mathbb{S}, \mathcal{I}_{ess})$ and $\mathcal{F}_k(\mathbb{S}, \mathcal{I}_{\cup})$ to compare four arrays which were introduced in Section 2.3. Figure 2.9(a) shows the result of $\mathcal{F}_k(\mathbb{S}, \mathcal{I}_{ess})$. It can be observed

that the ULA is the most robust array, followed by the coprime array, and then the MRA and the nested array. In particular, the curves of the MRA and the nested array fit perfectly. It means the robustness of them is equal no matter which k we consider based on $\mathcal{F}_k(\mathbb{S}, \mathcal{J}_{ess})$. The reason is that both of MRA and nested arrays are maximally economic arrays [23]. On the other hand, if we choose \mathcal{J}_U as the importance function (Figure 2.9 (b)), the nested array is more robust than the MRA. In fact, this relationship is more consistent with the simulation performance of the DOA estimation, and we will show it in Chapter 4.

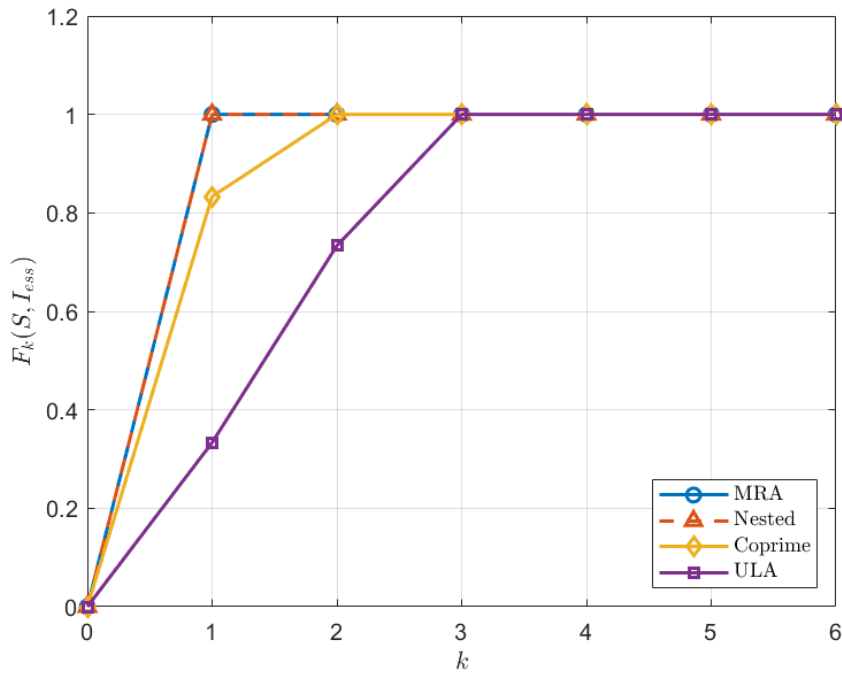
In Section 2.4, we mentioned that sensor failures could occur randomly. Thus, we can assume that each sensor fails independently with probability p . After removing the faulty sensors, we use $\bar{\mathbb{S}}$ and $\bar{\mathbb{D}}$ to denote the array and its corresponding difference coarray, respectively. There is a robustness definition based on the probability p , which was proposed in [24]. It can be defined as

$$P_c = \Pr[\bar{\mathbb{D}} \neq \mathbb{D}]. \quad (2.30)$$

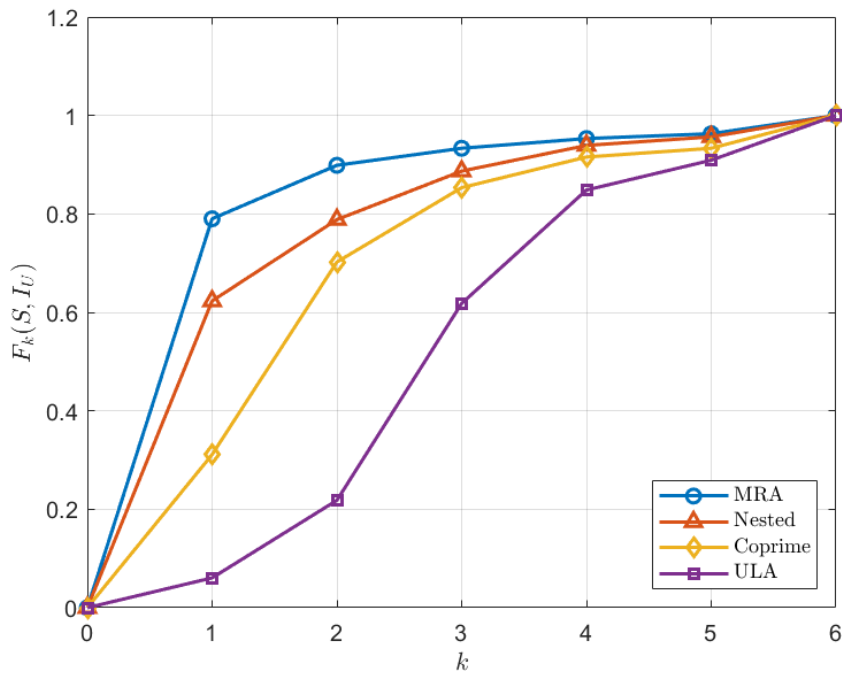
Eq.(2.30) denotes the probability that the difference coarray changes when faulty sensors are removed. An array is more robust when P_c close to 0, since its difference coarray does not be impacted easily by sensor failures.

Here the set \mathbb{A} , composed of faulty sensors, are stochastic because of the failure probability of each sensor, p . The following is the advantages of P_c [24].

1. We do not need to know the information of the number of faulty sensors.
2. The parameter p can be designed based on the budget because it is related to the quality and the cost of the sensing device.



(a)



(b)

Figure 2.9: Generalized k -fragility of four arrays (same as Figure 2.8) with two different importance functions, (a) $\mathcal{I}_{ess}(\mathbb{A})$ and (b) $\mathcal{I}_U(\mathbb{A})$.

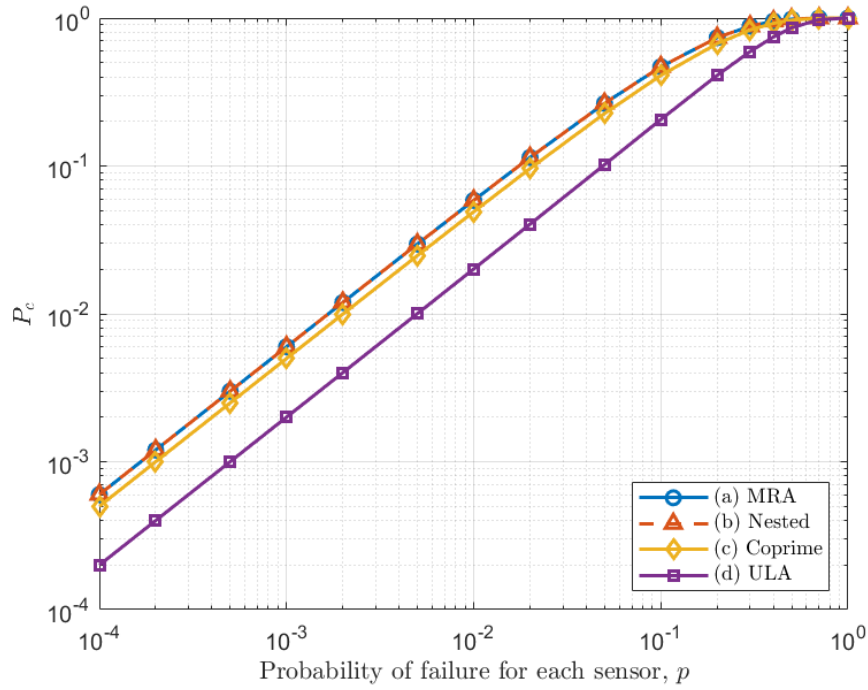
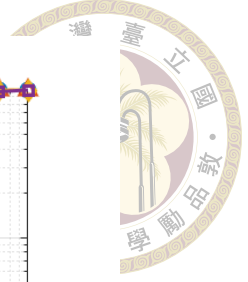


Figure 2.10: P_c of four different arrays (same as Figure 2.8). The curves are depicted based on (2.31).

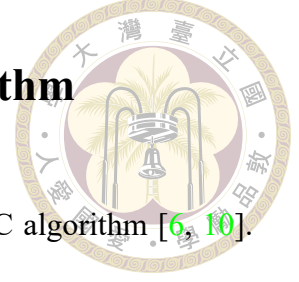
The authors in [24] presented a closed-form relationship between P_c and $\mathcal{F}_k(\mathbb{S}, \mathcal{I}_{ess})$, like the following equation.

$$P_c = \sum_{k=1}^{|\mathbb{S}|} |\mathcal{E}_k| p^k (1-p)^{|\mathbb{S}|-k} = \sum_{k=1}^{|\mathbb{S}|} \binom{|\mathbb{S}|}{k} \mathcal{F}_k(\mathbb{S}, \mathcal{I}_{ess}) p^k (1-p)^{|\mathbb{S}|-k} \quad (2.31)$$

Also, we compare the same four arrays as Figure 2.8. In Figure 2.10, we plot the curves of P_c through (2.31). It can be found that the curves of the MRA and the nested array still overlap. They are the least robust arrays among these arrays based on P_c . The ULA is the most robust array due to the smallest probability of difference coarray changes.

However, the robustness metrics related to the size of \mathbb{U} and the sensor failure probability p has not been developed yet. Therefore, we will propose a new metric that related to them at the same time, and also derive a closed-form relationship with $\mathcal{F}_k(\mathbb{S}, \mathcal{I}_{\mathbb{U}})$ in Chapter 4.

2.6 Coarray-Based DOA Estimation Algorithm



The DOA estimator we use in this thesis is the coarray MUSIC algorithm [6, 10].

Hence, we will review the details of the algorithm in this section.

From (2.9), we know the array output can be denoted by $\mathbf{x}(t) = \mathbf{A}(\boldsymbol{\theta})\mathbf{s}(t) + \mathbf{n}(t)$, where $\mathbf{x}(t), \mathbf{n}(t) \in \mathbb{C}^{|\mathcal{S}|}$. For the derivation of the algorithm, there are some statistical assumptions that must be met. The signal term $\mathbf{s}(t)$ and the noise term $\mathbf{n}(t)$ are zero-mean and uncorrelated random vectors. These relations mean that

$$\mathbb{E}[\mathbf{s}(t)] = \mathbf{0}, \quad \mathbb{E}[\mathbf{n}(t)] = \mathbf{0}, \quad \mathbb{E}[\mathbf{s}(t)\mathbf{n}^H(t)] = \mathbf{O}_{|\mathcal{S}| \times |\mathcal{S}|}, \quad (2.32)$$

$$\mathbb{E}[\mathbf{s}(t)\mathbf{s}^H(t)] = \mathbf{P} = \begin{bmatrix} p_1 & 0 & \cdots & 0 \\ 0 & p_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & p_D \end{bmatrix}, \quad \mathbb{E}[\mathbf{n}(t)\mathbf{n}^H(t)] = p_n \mathbf{I}, \quad (2.33)$$

where p_i, p_n are the source power and the noise power, respectively.

The essence of coarray-based algorithms is to convert the data to their second-order statistics. Then the covariance matrix of $\mathbf{x}(t)$ is defined as

$$\begin{aligned} \mathbf{R}_S &= \mathbb{E} [\mathbf{x}(t)\mathbf{x}^H(t)] \\ &= \mathbb{E} [(\mathbf{A}(\boldsymbol{\theta})\mathbf{s}(t) + \mathbf{n}(t))(\mathbf{A}(\boldsymbol{\theta})\mathbf{s}(t) + \mathbf{n}(t))^H] \\ &= \mathbf{A}(\boldsymbol{\theta}) \mathbb{E} [\mathbf{s}(t)\mathbf{s}^H(t)] \mathbf{A}^H(\boldsymbol{\theta}) + \mathbb{E} [\mathbf{n}(t)\mathbf{n}^H(t)] \\ &= \mathbf{A}(\boldsymbol{\theta})\mathbf{P}\mathbf{A}^H(\boldsymbol{\theta}) + p_n \mathbf{I}. \end{aligned} \quad (2.34)$$

All of the entries in $\mathbf{A}(\boldsymbol{\theta})\mathbf{P}\mathbf{A}^H(\boldsymbol{\theta})$ can be obtained from

$$\langle \mathbf{A}(\boldsymbol{\theta})\mathbf{P}\mathbf{A}^H(\boldsymbol{\theta}) \rangle_{n_1, n_2} = \sum_{n_1, n_2 \in \mathbb{S}} p_i e^{j\pi \sin \theta_i (n_1 - n_2)}. \quad (2.35)$$



These elements can be seen as the factors associated with the difference coarray. Thus, by vectoring $\mathbf{R}_{\mathbb{S}}$, we can get the autocorrelation vector defined on the difference coarray, like

$$\mathbf{r} = (\mathbf{A}^* \odot \mathbf{A})\mathbf{p} + p_n \mathbf{i}, \quad (2.36)$$

where $\mathbf{p} = [p_1, p_2, \dots, p_D]^T$ and $\mathbf{i} = \text{vec}(\mathbf{I})$. Eq.(2.36) can be regarded as the output on the difference coarray. If we carefully select the rows of \mathbf{r} , then we can obtain the measurement vector on \mathbb{U} by a selection matrix \mathbf{F} [7], where

$$\mathbf{F}_{m, p+(q-1)N} = \begin{cases} \frac{1}{w(m-u)} & , \Delta_{pq} = m - u \\ 0 & , \text{otherwise} \end{cases}. \quad (2.37)$$

for $m = 1, 2, \dots, 2u - 1$, $p = 1, 2, \dots, N$, $q = 1, 2, \dots, N$.

Among (2.37), N is the number of physical sensors, $w(\cdot)$ is the weight function defined in Definition 2.2.3, $u = \max(\mathbb{U}) + 1$, and the (m, n) -th element of $\mathbf{R}_{\mathbb{S}}$ is associated with the difference $s_m - s_n = \Delta_{mn}$, where $s_m, s_n \in \mathbb{S}$. Therefore, the measurement vector on \mathbb{U} can be represented as

$$\mathbf{x}_{\mathbb{U}} = \mathbf{F}\mathbf{r}. \quad (2.38)$$

In the MUSIC algorithm, it is necessary to perform eigen-decomposition on a positive semidefinite covariance matrix. Thus, a spatially smoothed matrix \mathbf{R}_{u1} was proposed to fit the positive semidefinite property[6], and perform the MUSIC algorithm on the dif-

ference coarray. \mathbf{R}_{u1} can be constructed by

$$\mathbf{R}_{u1} = \frac{1}{u} \sum_{i=1}^u \mathbf{z}_i \mathbf{z}_i^H. \quad (2.39)$$



where $\mathbf{z}_k = \Gamma_k \mathbf{x}_U$ and $\Gamma_k = [\mathbf{0}_{u \times (k-1)} \quad \mathbf{I}_{u \times u} \quad \mathbf{0}_{u \times (u-k)}]$ for $k = 1, 2, \dots, u$. Moreover, another positive semidefinite matrix \mathbf{R}_{u2} was proposed in [10], and \mathbf{R}_{u2} can be constructed by

$$\mathbf{R}_{u2} = [\mathbf{z}_u \quad \mathbf{z}_{u-1} \quad \cdots \quad \mathbf{z}_1]. \quad (2.40)$$

The two matrices \mathbf{R}_{u1} and \mathbf{R}_{u2} have the relationship that $\mathbf{R}_{u1} = \mathbf{R}_{u2}^2 / u$. The equation means that \mathbf{R}_{u1} and \mathbf{R}_{u2} share the same eigenspace. Here we use \mathbf{R}_{u2} to perform the MUSIC algorithm. \mathbf{R}_{u2} can also be represented as

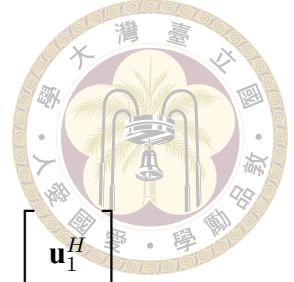
$$\mathbf{R}_{u2} = \mathbf{A}_U \mathbf{P} \mathbf{A}_U^H + p_n \mathbf{I}, \quad (2.41)$$

where $\mathbf{A}_U = [\mathbf{a}_U(\theta_1) \quad \mathbf{a}_U(\theta_2) \quad \cdots \quad \mathbf{a}_U(\theta_D)]_{u \times D}$ and $\mathbf{a}_U(\theta_i) = [1 \quad e^{j\pi \sin \theta_i} \quad \cdots \quad e^{j(u-1)\pi \sin \theta_i}]^T$.

We begin by decomposing $\mathbf{A}_U \mathbf{P} \mathbf{A}_U^H$ in (2.41) into the eigenvalues μ_j and the eigenvectors \mathbf{u}_j , where

$$\mathbf{A}_U \mathbf{P} \mathbf{A}_U^H = [\mathbf{u}_1 \quad \cdots \quad \mathbf{u}_D] \begin{bmatrix} \mu_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mu_D \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^H \\ \vdots \\ \mathbf{u}_D^H \end{bmatrix}. \quad (2.42)$$

The eigenvalues satisfy $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_D > 0$, and the eigenvectors are orthonormal. Assume \mathbf{A}_U is full-rank that $\text{rank}(\mathbf{A}_U) = D$. We can construct the other $u - D$ orthonormal eigenvectors $[\mathbf{u}_{D+1} \quad \cdots \quad \mathbf{u}_u]$ to operate eigen-decomposition on the matrix \mathbf{R}_{u2} in (2.41). Therefore,



$$\mathbf{A}_U \mathbf{P} \mathbf{A}_U^H + p_n \mathbf{I} =$$

$$\begin{bmatrix} \mathbf{u}_1 & \cdots & \mathbf{u}_D & \mathbf{u}_{D+1} & \cdots & \mathbf{u}_u \end{bmatrix}
 \begin{bmatrix}
 \mu_1 + p_n & \cdots & 0 & 0 & \cdots & 0 \\
 \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
 0 & \cdots & \mu_D + p_n & 0 & \cdots & 0 \\
 0 & \cdots & 0 & p_n & \cdots & 0 \\
 \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\
 0 & \cdots & 0 & 0 & \cdots & p_n
 \end{bmatrix}
 \begin{bmatrix}
 \mathbf{u}_1^H \\
 \vdots \\
 \mathbf{u}_D^H \\
 \mathbf{u}_{D+1}^H \\
 \vdots \\
 \mathbf{u}_u^H
 \end{bmatrix}.$$
(2.43)

Through these eigenvectors, we can separate the space \mathbb{C}^u into a signal subspace and a noise subspace. The first D eigenvectors $\{\mathbf{u}_1, \cdots, \mathbf{u}_D\}$, which corresponds to the D larger eigenvalues $\{\mu_1 + p_n, \cdots, \mu_D + p_n\}$, consist of the bases of the signal subspace. Moreover, the remaining $u - D$ eigenvectors $\{\mathbf{u}_{D+1}, \cdots, \mathbf{u}_u\}$, which corresponds to the $u - D$ smaller eigenvalues $\{p_n, \cdots, p_n\}$, consist of the bases of the noise subspace. We can represent the subspace as a matrix that composed of their corresponding bases, like the following form.

$$\mathbf{U}_s = [\mathbf{u}_1 \ \mathbf{u}_2 \ \cdots \ \mathbf{u}_D] \in \mathbb{C}^{u \times D}, \quad \text{for the signal subspace,}$$
(2.44)

$$\mathbf{U}_n = [\mathbf{u}_{D+1} \ \mathbf{u}_{D+2} \ \cdots \ \mathbf{u}_u] \in \mathbb{C}^{u \times (u-D)}, \quad \text{for the noise subspace.}$$

Then through the orthonormal property, multiplying the noise bases vector to (2.42) will obtain a zero vector. Besides that, since \mathbf{A}_U is full-rank and \mathbf{P} is positive definite, we can make a conclusion that

$$\mathbf{U}_n^H \mathbf{a}_U(\theta_i) = \mathbf{0}, \quad \text{for } i = 1, 2, \cdots, D.$$
(2.45)

Based on (2.45), the MUSIC spectrum can be defined as

$$P_{MUSIC}(\theta) = \frac{1}{\|\mathbf{U}_n^H \mathbf{a}_U(\theta)\|_2^2}, \quad \text{for } -\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}. \quad (2.46)$$



Now we add the tilde notation to distinguish the actually received data. The signal measured at the array is denoted by $\tilde{\mathbf{x}}_S$ and the estimated covariance matrix can be obtained by K snapshots, where

$$\tilde{\mathbf{R}}_S = \frac{1}{K} \sum_{t=1}^K \tilde{\mathbf{x}}(t) \tilde{\mathbf{x}}^H(t). \quad (2.47)$$

We replace the measurement \mathbf{x}_U with the finite snapshot version $\tilde{\mathbf{x}}_U$, which is defined as follows [10].

$$\langle \tilde{\mathbf{x}}_U \rangle_m = \frac{1}{w(m)} = \sum_{n_1 - n_2 = m} \langle \tilde{\mathbf{R}}_S \rangle_{n_1, n_2}, \quad (2.48)$$

for all $m \in U$. It was proved that the MUSIC spectrum can be computed directly from the Toeplitz matrix $\tilde{\mathbf{R}}_{u2}$ [10], where

$$\tilde{\mathbf{R}}_{u2} = \begin{bmatrix} [\tilde{\mathbf{x}}_U]_u & [\tilde{\mathbf{x}}_U]_{u-1} & \cdots & [\tilde{\mathbf{x}}_U]_1 \\ [\tilde{\mathbf{x}}_U]_{u+1} & [\tilde{\mathbf{x}}_U]_u & \cdots & [\tilde{\mathbf{x}}_U]_2 \\ \vdots & \vdots & \ddots & \vdots \\ [\tilde{\mathbf{x}}_U]_{2u-1} & [\tilde{\mathbf{x}}_U]_{2u-2} & \cdots & [\tilde{\mathbf{x}}_U]_u \end{bmatrix}. \quad (2.49)$$

Also, it was proved in [10] that the computational complexity of performing the coarray MUSIC algorithm from $\tilde{\mathbf{R}}_{u2}$ is lower than that from $\tilde{\mathbf{R}}_{u1}$.





Chapter 3 Symmetrical Arrays and Proposed Properties

As we mentioned in Chapter 1, symmetrical arrays have the advantage of improving the performance. Moreover, symmetrical arrays are more robust than asymmetrical arrays. The reasons will be discussed in Chapter 3 and Chapter 4. In Section 3.1, we will introduce how to construct the symmetrical array $\tilde{\mathcal{S}}$ through the original array \mathcal{S} . The corresponding difference coarray and the ULA segment in the difference coarray are denoted by $\tilde{\mathcal{D}}$ and $\tilde{\mathcal{U}}$, respectively. Thus, in Section 3.2, we will present the relationship between these sets and the size relationship between them. Several importance function properties related to the symmetrical arrays will be mentioned in Section 3.3. Through these properties, we can give the lower bound and the upper bound of the generalized 1-fragility of symmetrical arrays. Additionally, in Section 3.4, we will prove that symmetrical coprime arrays can reach the lower bound of the generalized 1-fragility, which is as robust as ULA. Finally, in Section 3.5, we will provide some examples of the comparison between \mathcal{S} and $\tilde{\mathcal{S}}$ through the numerical results.



3.1 Symmetrical Array Generation

We know that every function $f(x)$ can be uniquely decomposed into the sum of the even part $f_e(x)$ and the odd part $f_o(x)$. That is, $f(x) = f_e(x) + f_o(x)$, where

$$f_e(x) = \frac{f(x) + f(-x)}{2} \quad \text{and} \quad f_o(x) = \frac{f(x) - f(-x)}{2}. \quad (3.1)$$

For arrays, we can also decompose them into a similar form. It is a union of two subarrays. Assume that the sensor array is denoted by \mathbb{S} , then it can be represented as the union of the even part \mathbb{S}_e and the odd part \mathbb{S}_o , like the following form.

$$\mathbb{S} = \mathbb{S}_e \cup \mathbb{S}_o \quad \text{and} \quad \mathbb{S}_e \cap \mathbb{S}_o = \emptyset. \quad (3.2)$$

Before explaining the method of decomposition, we need to define the reverse set, the even subset and the odd subset.

Definition 3.1.1. \mathbb{Q}^r is said to be the reverse set of a set \mathbb{Q} , if

$$\mathbb{Q}^r = \{\min(\mathbb{Q}) + \max(\mathbb{Q}) - n \mid n \in \mathbb{Q}\}. \quad (3.3)$$

Definition 3.1.2. \mathbb{S}_e is the even subset of \mathbb{S} , where $\mathbb{S}_e = \mathbb{S} \cap \mathbb{S}^r$, and \mathbb{S}_o is the odd subset of \mathbb{S} , where $\mathbb{S}_o = \mathbb{S} \setminus \mathbb{S}_e$.

Figure 3.1 shows the relationship of the sets \mathbb{S} , \mathbb{S}^r , \mathbb{S}_o , \mathbb{S}_e , and \mathbb{S}_o^r . Note that three sets \mathbb{S}_e , \mathbb{S}_o , and \mathbb{S}_o^r are mutually exclusive. Then we can get

$$\mathbb{S} = \mathbb{S}_o \cup \mathbb{S}_e, \quad \mathbb{S}^r = \mathbb{S}_e \cup \mathbb{S}_o^r. \quad (3.4)$$

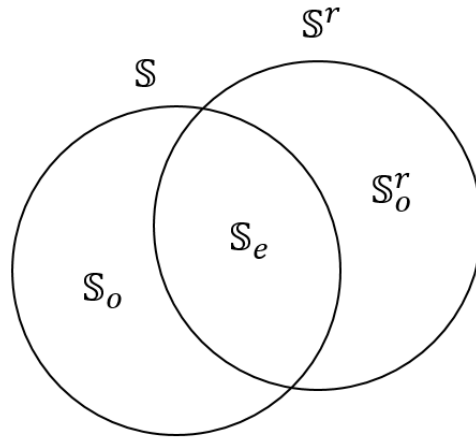


Figure 3.1: The relationship between the sets \mathbb{S} , \mathbb{S}^r , \mathbb{S}_o , \mathbb{S}_e and \mathbb{S}_o^r .

Now the symmetrical array with respect to the original array \mathbb{S} can be easily constructed. We denote the symmetrical array by $\tilde{\mathbb{S}}$, where

$$\tilde{\mathbb{S}} = \mathbb{S}_e \cup \mathbb{S}_o \cup \mathbb{S}_o^r. \quad (3.5)$$

An example is shown in Figure 3.2. We consider the coprime array with $M = 3$ and $N = 5$, so $\mathbb{S} = \{0, 3, 5, 6, 9, 10, 12, 15, 20, 25\}$. From the discussion above, we can get $\mathbb{S}^r = \{0, 5, 10, 13, 15, 16, 19, 20, 22, 25\}$, $\mathbb{S}_e = \mathbb{S} \cap \mathbb{S}^r = \{0, 5, 10, 15, 20, 25\}$ and $\mathbb{S}_o = \mathbb{S} \setminus \mathbb{S}_e = \{3, 6, 9, 12\}$. Finally, if we construct the reverse set of \mathbb{S}_o , then we can get the symmetrical array $\tilde{\mathbb{S}} = \{0, 3, 5, 6, 9, 10, 12, 13, 15, 16, 19, 20, 22, 15\}$.

In the following discussion, two symbols related to the array \mathbb{S} will be used. Thus, here we need to define them.

Definition 3.1.3. If the size of the \mathbb{S}_e of the array \mathbb{S} is odd, then the sensor in the middle of the array \mathbb{S} is defined as

$$\text{middle}(\mathbb{S}) = \frac{\max(\mathbb{S}) + \min(\mathbb{S})}{2}. \quad (3.6)$$

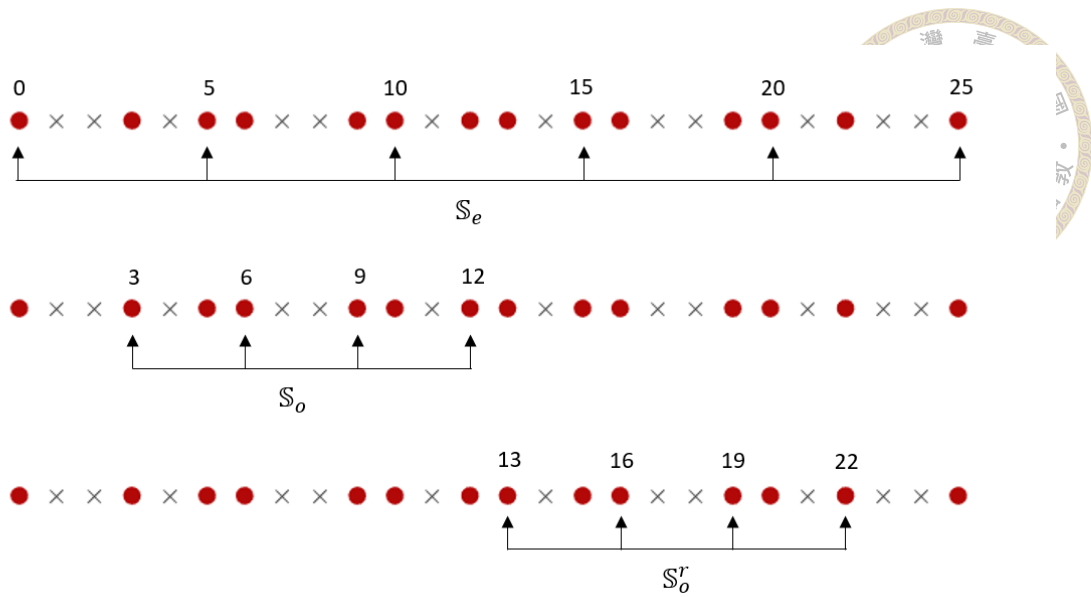
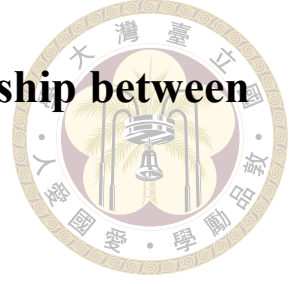


Figure 3.2: An illustration of the sets \mathbb{S}_e , \mathbb{S}_o and \mathbb{S}_o^r . Here we consider a symmetrical coprime array, where $\tilde{\mathbb{S}} = \{0, 3, 5, 6, 9, 10, 12, 13, 15, 16, 19, 20, 22, 25\}$.

Definition 3.1.4. The aperture of an array \mathbb{S} is defined as

$$\text{aperture}(\mathbb{S}) = \max(\mathbb{S}) - \min(\mathbb{S}). \quad (3.7)$$

Actually, \mathbb{S} and $\tilde{\mathbb{S}}$ have the same aperture. According to (3.4) and (3.5), it can be observed that $\mathbb{S} \subseteq \tilde{\mathbb{S}}$.



3.2 Relationship between Sets and Relationship between Size of Sets

There will be a corresponding difference coarray after symmetrizing the array \mathbb{S} . We use $\tilde{\mathbb{D}}$ to represent the difference coarray of $\tilde{\mathbb{S}}$. Also, we use $\tilde{\mathbb{U}}$ to represent the central ULA segment in $\tilde{\mathbb{D}}$. In order to discuss the properties conveniently, we split \mathbb{D} and $\tilde{\mathbb{D}}$ into several different subsets by (2.14) and (2.15).

$$\begin{aligned}\mathbb{D} &= SD(\mathbb{S}_o) \cup SD(\mathbb{S}_e) \cup CD(\mathbb{S}_o, \mathbb{S}_e). \\ \tilde{\mathbb{D}} &= SD(\mathbb{S}_o) \cup SD(\mathbb{S}_e) \cup SD(\mathbb{S}_o^r) \cup CD(\mathbb{S}_o, \mathbb{S}_e) \cup CD(\mathbb{S}_e, \mathbb{S}_o^r) \cup CD(\mathbb{S}_o, \mathbb{S}_o^r).\end{aligned}\tag{3.8}$$

The following properties will illustrate the relationship between \mathbb{D} and $\tilde{\mathbb{D}}$, also \mathbb{U} and $\tilde{\mathbb{U}}$. They help us to know more about the difference coarray of symmetrical arrays, and they contribute to the discussion of the generalized 1-fragility which will be illustrated in the next section.

Proposition 3.2.1. $\mathbb{D} \subseteq \tilde{\mathbb{D}}$.

Proof. Let $m \in \mathbb{D}$, there exist $n_1, n_2 \in \mathbb{S}$ such that $n_1 - n_2 = m$. Since $\mathbb{S} \subseteq \tilde{\mathbb{S}}$, we have $n_1, n_2 \in \tilde{\mathbb{S}}$ implying $m \in \tilde{\mathbb{D}}$. Then $\mathbb{D} \subseteq \tilde{\mathbb{D}}$. \square

From Proposition 3.2.1, we can know that $|\mathbb{D}| \leq |\tilde{\mathbb{D}}|$. Moreover, there is a necessary and sufficient condition related to this inequality:

$$CD(\mathbb{S}_o, \mathbb{S}_o^r) \subseteq \mathbb{D} \quad \text{if and only if} \quad \mathbb{D} = \tilde{\mathbb{D}}.\tag{3.9}$$

Equivalently, if $CD(\mathbb{S}_o, \mathbb{S}_o^r) \subseteq \mathbb{D}$, then $|\mathbb{D}| = |\tilde{\mathbb{D}}|$. The proof is shown as follows.



Proof. $CD(\mathbb{S}_o, \mathbb{S}_o^r) \subseteq \mathbb{D}$ if and only if $\mathbb{D} = \tilde{\mathbb{D}}$.

1. If $CD(\mathbb{S}_o, \mathbb{S}_o^r) \subseteq \mathbb{D} = SD(\mathbb{S}_o) \cup SD(\mathbb{S}_e) \cup CD(\mathbb{S}_o, \mathbb{S}_e)$, then

$$\begin{aligned} \tilde{\mathbb{D}} &= SD(\mathbb{S}_o) \cup SD(\mathbb{S}_e) \cup SD(\mathbb{S}_o^r) \cup CD(\mathbb{S}_o, \mathbb{S}_e) \cup CD(\mathbb{S}_e, \mathbb{S}_o^r) \cup CD(\mathbb{S}_o, \mathbb{S}_o^r) \\ &= SD(\mathbb{S}_o) \cup SD(\mathbb{S}_e) \cup CD(\mathbb{S}_o, \mathbb{S}_e) \\ &= \mathbb{D}, \end{aligned} \tag{3.10}$$

since $SD(\mathbb{S}_o) = SD(\mathbb{S}_o^r)$ and $CD(\mathbb{S}_o, \mathbb{S}_e) = CD(\mathbb{S}_e, \mathbb{S}_o^r)$, which are proved in Lemma 3.2.1 and Lemma 3.2.2.

2. If $\mathbb{D} = \tilde{\mathbb{D}}$, where

$$\tilde{\mathbb{D}} = SD(\mathbb{S}_o) \cup SD(\mathbb{S}_e) \cup SD(\mathbb{S}_o^r) \cup CD(\mathbb{S}_o, \mathbb{S}_e) \cup CD(\mathbb{S}_e, \mathbb{S}_o^r) \cup CD(\mathbb{S}_o, \mathbb{S}_o^r), \tag{3.11}$$

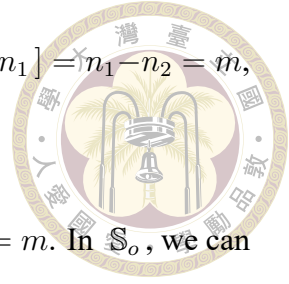
then $CD(\mathbb{S}_o, \mathbb{S}_o^r) \subseteq \mathbb{D} = SD(\mathbb{S}_o) \cup SD(\mathbb{S}_e) \cup CD(\mathbb{S}_o, \mathbb{S}_e) = \mathbb{D}$, since $SD(\mathbb{S}_o) = SD(\mathbb{S}_o^r)$ and $CD(\mathbb{S}_o, \mathbb{S}_e) = CD(\mathbb{S}_e, \mathbb{S}_o^r)$, which are proved in Lemma 3.2.1 and Lemma 3.2.2.

By 1 and 2, Eq.(3.9) can be proved. □

Lemma 3.2.1. $SD(\mathbb{S}_o) = SD(\mathbb{S}_o^r)$, where \mathbb{S}_o and \mathbb{S}_o^r are defined in Definition 3.1.2.

Proof. $SD(\mathbb{S}_o) = SD(\mathbb{S}_o^r)$. The proof is divided into two parts, $SD(\mathbb{S}_o) \subseteq SD(\mathbb{S}_o^r)$ and $SD(\mathbb{S}_o^r) \subseteq SD(\mathbb{S}_o)$.

1. Let $m \in SD(\mathbb{S}_o)$, there exist $n_1, n_2 \in \mathbb{S}_o$ such that $n_1 - n_2 = m$. In \mathbb{S}_o^r , we can also find two numbers which are associated with n_1 and n_2 . That is, there exist $m_1, m_2 \in \mathbb{S}_o^r$, where $m_1 = \min(\mathbb{S}) + \max(\mathbb{S}) - n_1$ and $m_2 = \min(\mathbb{S}) + \max(\mathbb{S}) - n_2$.



Since $m_2 - m_1 = [\min(\mathbb{S}) + \max(\mathbb{S}) - n_2] - [\min(\mathbb{S}) + \max(\mathbb{S}) - n_1] = n_1 - n_2 = m$,
 $m \in SD(\mathbb{S}_o^r)$. Then we can obtain $SD(\mathbb{S}_o) \subseteq SD(\mathbb{S}_o^r)$.

2. Let $m \in SD(\mathbb{S}_o^r)$, there exist $n_1, n_2 \in \mathbb{S}_o^r$ such that $n_1 - n_2 = m$. In \mathbb{S}_o , we can also find two numbers which are associated with n_1 and n_2 . That is, there exist $m_1, m_2 \in \mathbb{S}_o$, where $m_1 = \min(\mathbb{S}) + \max(\mathbb{S}) - n_1$ and $m_2 = \min(\mathbb{S}) + \max(\mathbb{S}) - n_2$. Since $m_2 - m_1 = [\min(\mathbb{S}) + \max(\mathbb{S}) - n_2] - [\min(\mathbb{S}) + \max(\mathbb{S}) - n_1] = n_1 - n_2 = m$, $m \in SD(\mathbb{S}_o)$. Then we can obtain $SD(\mathbb{S}_o^r) \subseteq SD(\mathbb{S}_o)$.

By 1 and 2, Lemma 3.2.1 holds true. □

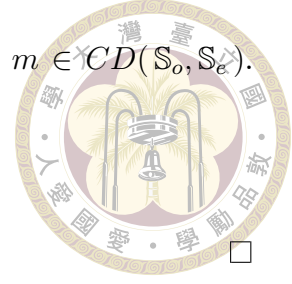
Lemma 3.2.2. $CD(\mathbb{S}_o, \mathbb{S}_e) = CD(\mathbb{S}_e, \mathbb{S}_o^r)$, where $\mathbb{S}_o, \mathbb{S}_e$ and \mathbb{S}_o^r are defined in Definition 3.1.2.

Proof. $CD(\mathbb{S}_o, \mathbb{S}_e) = CD(\mathbb{S}_e, \mathbb{S}_o^r)$. The proof is divided into two parts, $CD(\mathbb{S}_o, \mathbb{S}_e) \subseteq CD(\mathbb{S}_e, \mathbb{S}_o^r)$ and $CD(\mathbb{S}_e, \mathbb{S}_o^r) \subseteq CD(\mathbb{S}_o, \mathbb{S}_e)$.

1. Let $m \in CD(\mathbb{S}_o, \mathbb{S}_e)$, there exist $n_1 \in \mathbb{S}_o$ and $n_2 \in \mathbb{S}_e$ such that $n_1 - n_2 = m$. In \mathbb{S}_o^r , we can find a number which is associated with n_1 . That is, there exists $m_1 \in \mathbb{S}_o^r$, where $m_1 = \min(\mathbb{S}) + \max(\mathbb{S}) - n_1$. The elements in \mathbb{S}_e are symmetric, so if we take an element m_2 that $m_2 = \min(\mathbb{S}) + \max(\mathbb{S}) - n_2$, then m_2 is also in \mathbb{S}_e . Since $m_2 - m_1 = [\min(\mathbb{S}) + \max(\mathbb{S}) - n_2] - [\min(\mathbb{S}) + \max(\mathbb{S}) - n_1] = n_1 - n_2 = m$, $m \in CD(\mathbb{S}_e, \mathbb{S}_o^r)$. Then we can obtain $CD(\mathbb{S}_o, \mathbb{S}_e) \subseteq CD(\mathbb{S}_e, \mathbb{S}_o^r)$.
2. Let $m \in CD(\mathbb{S}_e, \mathbb{S}_o^r)$, there exist $n_1 \in \mathbb{S}_o^r$ and $n_2 \in \mathbb{S}_e$ such that $n_1 - n_2 = m$. In \mathbb{S}_o , we can find a number which is associated with n_1 . That is, there exists $m_1 \in \mathbb{S}_o$ where $m_1 = \min(\mathbb{S}) + \max(\mathbb{S}) - n_1$. If we take an element m_2 that $m_2 = \min(\mathbb{S}) + \max(\mathbb{S}) - n_2$, then m_2 is also in \mathbb{S}_e . Since $m_2 - m_1 = [\min(\mathbb{S}) +$

$$\max(\mathbb{S}) - n_2] - [\min(\mathbb{S}) + \max(\mathbb{S}) - n_1] = n_1 - n_2 = m, \quad m \in CD(\mathbb{S}_o, \mathbb{S}_e).$$

Then we can obtain $CD(\mathbb{S}_e, \mathbb{S}_o^r) \subseteq CD(\mathbb{S}_o, \mathbb{S}_e)$.



By 1 and 2, Lemma 3.2.2 holds true. \square

Proposition 3.2.2. $|\mathbb{D}| \leq |\tilde{\mathbb{D}}| \leq |\mathbb{D}| + |\mathbb{S}_o|(1 + |\mathbb{S}_o|)$.

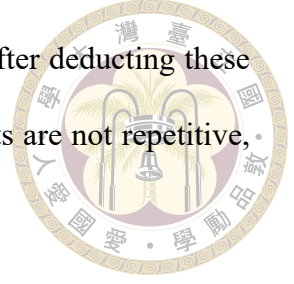
Proposition 3.2.1 illustrates the condition that $|\tilde{\mathbb{D}}|$ reaches its lower bound: $|\mathbb{D}|$. Thus, here we will focus on the right inequality of Proposition 3.2.2, i.e., the upper bound of $|\tilde{\mathbb{D}}|$. There are two sufficient conditions that $|\tilde{\mathbb{D}}|$ can reach its upper bound. Note that both of the conditions need to be satisfied.

1. $\mathbb{D} \cap CD(\mathbb{S}_o, \mathbb{S}_o^r) = \emptyset$.
2. $|CD(\mathbb{S}_o, \mathbb{S}_o^r)|$ reaches its upper bound $|\mathbb{S}_o|(1 + |\mathbb{S}_o|)$.

According to (3.8), we know that $\tilde{\mathbb{D}} = \mathbb{D} \cup CD(\mathbb{S}_o, \mathbb{S}_o^r)$. Therefore, the size of $\tilde{\mathbb{D}}$, $|\tilde{\mathbb{D}}| = |\mathbb{D}| + |CD(\mathbb{S}_o, \mathbb{S}_o^r)| - |\mathbb{D} \cap CD(\mathbb{S}_o, \mathbb{S}_o^r)|$. If the first condition is met, namely $\mathbb{D} \cap CD(\mathbb{S}_o, \mathbb{S}_o^r) = \emptyset$, then $|\tilde{\mathbb{D}}| = |\mathbb{D}| + |CD(\mathbb{S}_o, \mathbb{S}_o^r)|$. Next, it can be demonstrated that the upper bound of $|CD(\mathbb{S}_o, \mathbb{S}_o^r)|$ is equal to $|\mathbb{S}_o|(1 + |\mathbb{S}_o|)$. First, $CD(\mathbb{S}_o, \mathbb{S}_o^r)$ is composed of the differences between two sensors which are located in \mathbb{S}_o and \mathbb{S}_o^r , respectively. Both \mathbb{S}_o and \mathbb{S}_o^r have $|\mathbb{S}_o|$ numbers, so there are $2 \times |\mathbb{S}_o|^2$ differences can be generated. Based on Definition 3.1.1, we know the relationship between \mathbb{S}_o and \mathbb{S}_o^r is

$$\mathbb{S}_o^r = \{\min(\mathbb{S}_o) + \max(\mathbb{S}_o) - n \mid n \in \mathbb{S}_o\}. \quad (3.12)$$

Now we pick two elements, $\min(\mathbb{S}_o) + \max(\mathbb{S}_o) - n_1$ in \mathbb{S}_o^r and n_2 in \mathbb{S}_o . The differences between them are $\pm[\min(\mathbb{S}_o) + \max(\mathbb{S}_o) - n_1 - n_2]$. We can find that swapping n_1 and n_2 will not change the differences. But now the element we pick in \mathbb{S}_o^r is



$\min(\mathbb{S}_o) + \max(\mathbb{S}_o) - n_2$ and the element we pick in \mathbb{S}_o is n_1 . After deducting these repetitive elements in $|CD(\mathbb{S}_o, \mathbb{S}_o^r)|$, if all of the remaining elements are not repetitive, then $|CD(\mathbb{S}_o, \mathbb{S}_o^r)|$ can reach its upper bound $|\mathbb{S}_o|(1 + |\mathbb{S}_o|)$.

Here we look at an example that $|\tilde{\mathbb{D}}| = |\mathbb{D}| + |\mathbb{S}_o|(1 + |\mathbb{S}_o|)$.

Example 3.2.1. We consider the array $\mathbb{S} = \{0, 1, 5, 6, 10, 15\}$. By (3.2) and (3.3), we can get the other sets, where $\mathbb{S}_e = \{0, 5, 10, 15\}$, $\mathbb{S}_o = \{1, 6\}$ and $\mathbb{S}_o^r = \{9, 14\}$. First, it can be found that $\mathbb{D} \cap CD(\mathbb{S}_o, \mathbb{S}_o^r) = \emptyset$, since $\mathbb{D} = \{0, \pm 1, \pm 4, \pm 5, \pm 6, \pm 9, \pm 10, \pm 15, \pm 16\}$ and $CD(\mathbb{S}_o, \mathbb{S}_o^r) = \{\pm 3, \pm 8, \pm 13\}$. Second, $|CD(\mathbb{S}_o, \mathbb{S}_o^r)|$ reaches its upper bound that $|CD(\mathbb{S}_o, \mathbb{S}_o^r)| = 6 = |\mathbb{S}_o|(1 + |\mathbb{S}_o|)$. Finally, we can examine the size of $|\tilde{\mathbb{D}}|$, and find that $|\tilde{\mathbb{D}}| = 23$, which is equal to $|\mathbb{D}| + |\mathbb{S}_o|(1 + |\mathbb{S}_o|)$.

Proposition 3.2.3. $\mathbb{U} \subseteq \tilde{\mathbb{U}}$.

Proof. Based on (2.12), $\mathbb{U} = \{m \mid \{0, 1, \dots, |m|\} \subseteq \mathbb{D}\}$. There exist $n_1, n_2 \in \mathbb{S}$ such that $n_1 - n_2 = m$. Since $\mathbb{S} \subseteq \tilde{\mathbb{S}}$, we have $n_1, n_2 \in \tilde{\mathbb{S}}$ implying $m \in \tilde{\mathbb{U}}$. Then $\mathbb{U} \subseteq \tilde{\mathbb{U}}$. □

With this relationship, we know that $|\mathbb{U}| \leq |\tilde{\mathbb{U}}|$. Now let us discuss the condition that the equality holds true.

1. If the original difference coarray \mathbb{D} is hole-free, then $|\mathbb{U}| = |\tilde{\mathbb{U}}|$. Since $|\mathbb{D}| = |\tilde{\mathbb{D}}|$, no new elements will be generated after symmetrizing the array \mathbb{S} .
2. Assume that the original difference coarray \mathbb{D} is not hole-free.
 - (a) If $CD(\mathbb{S}_o, \mathbb{S}_o^r) \cap \{\pm \min(\mathbb{H}^+)\} \neq \emptyset$, then $|\mathbb{U}| < |\tilde{\mathbb{U}}|$.
 - (b) If $CD(\mathbb{S}_o, \mathbb{S}_o^r) \cap \{\pm \min(\mathbb{H}^+)\} = \emptyset$, then $|\mathbb{U}| = |\tilde{\mathbb{U}}|$.

Here the set \mathbb{H} was defined in (2.13). The definition of \mathbb{U} can also be expressed as this form: $\mathbb{U} = \{m \mid -\min(\mathbb{H}^+) < m < \min(\mathbb{H}^+)\}$. If there exist elements in $\tilde{\mathbb{D}}$ fill the holes $\{\pm\min(\mathbb{H}^+)\}$, then the length of \mathbb{U} will become larger, so $|\mathbb{U}| < |\tilde{\mathbb{U}}|$. Conversely, if no new elements can fill the holes $\{\pm\min(\mathbb{H}^+)\}$, then the length of \mathbb{U} will still be the same, so $|\mathbb{U}| = |\tilde{\mathbb{U}}|$.

Proposition 3.2.4. 1. If $\mathbb{D} = \tilde{\mathbb{D}}$, then $\mathbb{U} = \tilde{\mathbb{U}}$. 2. If $\mathbb{U} = \tilde{\mathbb{U}}$ and $|\mathbb{H}^+| \in \{0, 1\}$, then $\mathbb{D} = \tilde{\mathbb{D}}$.

For the first statement, if two difference coarrays are equal, then the central ULA segments of them will be equal as well. But conversely, for the second statement, $\mathbb{U} = \tilde{\mathbb{U}}$ does not mean that \mathbb{D} is definitely equal to $\tilde{\mathbb{D}}$. It is possible to happen that $\tilde{\mathbb{D}}$ have the elements which are not belong to \mathbb{D} . However, as long as we add another condition that $|\mathbb{H}^+| \in \{0, 1\}$, \mathbb{D} will be equal to $\tilde{\mathbb{D}}$. The reasons can be expressed as follows. $|\mathbb{H}^+| = 0$ means \mathbb{D} is a hole-free difference coarray, and $\mathbb{U} = \mathbb{D}$. Under the premise, $\tilde{\mathbb{D}}$ is also a hole-free difference coarray, so $\mathbb{D} = \tilde{\mathbb{D}}$. Also, $|\mathbb{H}^+| = 1$ means there is a hole in \mathbb{D}^+ . After symmetrizing the array, new elements in the difference coarray, $CD(\mathbb{S}_o, \mathbb{S}_o^r)$, probably fill this hole. Nevertheless, if this hole is filled, the assumption $\mathbb{U} = \tilde{\mathbb{U}}$ will be violated, so definitely there is a same hole in the $\tilde{\mathbb{D}}^+$. Then we can obtain $\mathbb{D} = \tilde{\mathbb{D}}$.



3.3 Properties of Importance Function and Generalized 1-fragility

In Chapter 2, we introduced the importance function and the generalized k -fragility. The importance function is a metric that can quantify the influence on the difference coarray of sensors or subsets. Therefore, some properties of the importance function will arise after symmetrizing the array \mathbb{S} . Furthermore, according to these proposed properties of the importance function, we can give the range of the generalized 1-fragility. For the convenience of discussion, here we define a set \mathbb{A}^r based on Definition 3.1.1, where

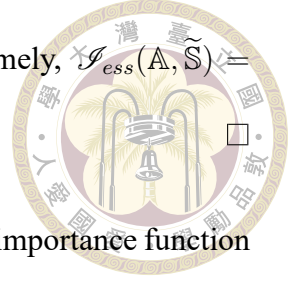
$$\mathbb{A}^r = \{a' = \min(\mathbb{S}) + \max(\mathbb{S}) - a \mid a \in \mathbb{A}\}. \quad (3.13)$$

Proposition 3.3.1. $\mathcal{I}(\mathbb{A}, \tilde{\mathbb{S}}) = \mathcal{I}(\mathbb{A}^r, \tilde{\mathbb{S}})$, for $\forall \mathbb{A}$, $\forall \mathbb{S}$, and two types of the importance function \mathcal{I}_{ess} and \mathcal{I}_U . Here we add the array $\tilde{\mathbb{S}}$ to the bracket of the importance function to represent that \mathbb{A} is removed from $\tilde{\mathbb{S}}$.

Proof. Assume that $\tilde{\mathbb{D}}_{\mathbb{A}}$ is the difference coarray of the array $\tilde{\mathbb{S}} \setminus \mathbb{A}$, and $\tilde{\mathbb{D}}_{\mathbb{A}^r}$ is the difference coarray of the array $\tilde{\mathbb{S}} \setminus \mathbb{A}^r$.

$$\begin{aligned} \tilde{\mathbb{D}}_{\mathbb{A}} &= \{m_1 - m_2 \mid \forall m_1, m_2 \in \tilde{\mathbb{S}} \setminus \mathbb{A}\} \\ &= \{(\min(\mathbb{S}) + \max(\mathbb{S}) - m_1) - (\min(\mathbb{S}) + \max(\mathbb{S}) - m_2) \mid \forall m_1, m_2 \in \tilde{\mathbb{S}} \setminus \mathbb{A}\} \\ &= \{p_1 - p_2 \mid \forall p_1, p_2 \in \tilde{\mathbb{S}} \setminus \mathbb{A}^r\} \\ &= \tilde{\mathbb{D}}_{\mathbb{A}^r} \end{aligned} \quad (3.14)$$

According to this derivation, we know that both removing the subset \mathbb{A} and removing the subset \mathbb{A}^r have the same difference coarray. Hence, regardless of $\mathcal{I}_{ess}(\mathbb{A})$ or $\mathcal{I}_U(\mathbb{A})$



we consider, the value of the importance function will be equal. Namely, $\mathcal{I}_{ess}(\mathbb{A}, \tilde{\mathbb{S}}) = \mathcal{I}_{ess}(\mathbb{A}^r, \tilde{\mathbb{S}})$ and $\mathcal{I}_{\mathbb{U}}(\mathbb{A}, \tilde{\mathbb{S}}) = \mathcal{I}_{\mathbb{U}}(\mathbb{A}^r, \tilde{\mathbb{S}})$.

Proposition 3.3.2. We consider the subset \mathbb{A} of the array \mathbb{S} and the importance function \mathcal{I}_{ess} . Four properties of $\mathcal{I}_{ess}(\mathbb{A}, \tilde{\mathbb{S}})$ are shown as follows. The relationship between a and \mathbb{A} was defined in (3.13).

1. Assume that $|\mathbb{A}| = 1$, $\mathbb{A} \subseteq \mathbb{S}_o$ and $\mathcal{I}_{ess}(\mathbb{A}, \mathbb{S}) = 1$.
If $\pm(a' - a) \in \mathbb{D}$, then $\mathcal{I}_{ess}(\mathbb{A}, \tilde{\mathbb{S}}) = 0$ and $\mathcal{I}_{ess}(\mathbb{A}^r, \tilde{\mathbb{S}}) = 0$.
2. Assume that $|\mathbb{A}| = 1$, $\mathbb{A} \subseteq \mathbb{S}_o$ and $\mathcal{I}_{ess}(\mathbb{A}, \mathbb{S}) = 0$.
If $\pm(a' - a) \in \mathbb{D}$, then $\mathcal{I}_{ess}(\mathbb{A}, \tilde{\mathbb{S}}) = 0$ and $\mathcal{I}_{ess}(\mathbb{A}^r, \tilde{\mathbb{S}}) = 0$.
3. Assume that $|\mathbb{A}| = 1$, $\mathbb{A} \subseteq \mathbb{S}_e \setminus \{\text{middle}(\mathbb{S}_e)\}$ and $\mathcal{I}_{ess}(\mathbb{A}, \mathbb{S}) = 1$.
 - (a) If $w(\pm(a' - a)) > 1$, then $\mathcal{I}_{ess}(\mathbb{A}, \tilde{\mathbb{S}}) = 0$ and $\mathcal{I}_{ess}(\mathbb{A}^r, \tilde{\mathbb{S}}) = 0$.
 - (b) If $w(\pm(a' - a)) = 1$, and $\pm(a' - a) \in CD(\mathbb{S}_o, \mathbb{S}_o^r)$, then $\mathcal{I}_{ess}(\mathbb{A}, \tilde{\mathbb{S}}) = 0$ and $\mathcal{I}_{ess}(\mathbb{A}^r, \tilde{\mathbb{S}}) = 0$.
 - (c) If $w(\pm(a' - a)) = 1$, and $\pm(a' - a) \notin CD(\mathbb{S}_o, \mathbb{S}_o^r)$, then $\mathcal{I}_{ess}(\mathbb{A}, \tilde{\mathbb{S}}) = 1$ and $\mathcal{I}_{ess}(\mathbb{A}^r, \tilde{\mathbb{S}}) = 1$.

Note that the weight function here is corresponding to the original array \mathbb{S} .

4. Assume that $|\mathbb{A}| = k$, $\mathbb{A} \subseteq \mathbb{S}_e$. If $\mathcal{I}_{ess}(\mathbb{A}, \tilde{\mathbb{S}}) = 0$, then $\mathcal{I}_{ess}(\mathbb{A}', \tilde{\mathbb{S}}) = 0$.

Property 1 and Property 2 can be explained together. Note that $a \in \mathbb{S}_o$ and $a' \in \mathbb{S}_o^r$. After symmetrizing the array \mathbb{S} , except for the difference $\pm(a' - a)$, all the other differences which are related to a can be found at least one alternative pair. Since $SD(\mathbb{S}_o) = SD(\mathbb{S}_o^r)$ in Lemma 3.2.1 and $CD(\mathbb{S}_o, \mathbb{S}_e) = CD(\mathbb{S}_e, \mathbb{S}_o^r)$ in Lemma 3.2.2, removing

a or a' from $\tilde{\mathbb{S}}$ will not reduce the elements in the set $\tilde{\mathbb{D}} \setminus CD(\mathbb{S}_o, \mathbb{S}_o^r)$. In addition, as we discussed in Proposition 3.2.2, there must exist repetitive elements when computing $CD(\mathbb{S}_o, \mathbb{S}_o^r)$. Thus, except for $\pm(a' - a)$, the other elements in $CD(\mathbb{S}_o, \mathbb{S}_o^r)$ will also not disappear after removing a or a' from $\tilde{\mathbb{S}}$. Note that the pair (a, a') only belongs to $\tilde{\mathbb{S}}$. If we want to let the equations $\mathcal{J}_{ess}(\mathbb{A}, \tilde{\mathbb{S}}) = 0$ and $\mathcal{J}_{ess}(\mathbb{A}^r, \tilde{\mathbb{S}}) = 0$ be true, the only sufficient condition is $\pm(a' - a) \in \mathbb{D}$.

In Property 3, both a and a' are belong to \mathbb{S}_e , so \mathbb{D} has the difference $\pm(a' - a)$. For (a), $w(\pm(a' - a)) > 1$ means that we can find another sensor pair in \mathbb{S} to generate the difference $\pm(a' - a)$. Here we assume that two sensors b_1 and b_2 are in the array \mathbb{S} . They are different from the sensor pair (a, a') and they can generate the difference $\pm(a' - a)$. If $b_1, b_2 \in \mathbb{S}_o$, then removing a or a' will not change $\tilde{\mathbb{D}}$ because of $SD(\mathbb{S}_o) = SD(\mathbb{S}_o^r)$ in Lemma 3.2.1. Also, if $b_1 \in \mathbb{S}_o$ and $b_2 \in \mathbb{S}_e$, then removing a or a' will not change $\tilde{\mathbb{D}}$ because of $CD(\mathbb{S}_o, \mathbb{S}_e) = CD(\mathbb{S}_e, \mathbb{S}_o^r)$ in Lemma 3.2.2. The final case, both b_1 and b_2 are in \mathbb{S}_e . The difference $\pm(a' - a)$ will also not disappear because of the symmetry of \mathbb{S}_e . Therefore, we can obtain $\mathcal{J}_{ess}(\mathbb{A}, \tilde{\mathbb{S}}) = 0$ and $\mathcal{J}_{ess}(\mathbb{A}^r, \tilde{\mathbb{S}}) = 0$. For (b), if $w(\pm(a' - a)) = 1$ and $\pm(a' - a) \in CD(\mathbb{S}_o, \mathbb{S}_o^r)$, the same result with (a) can be obtained. According to Figure 3.1, we know that \mathbb{S}_o , \mathbb{S}_e and \mathbb{S}_o^r are mutually exclusive. Removing a or a' does not change $CD(\mathbb{S}_o, \mathbb{S}_o^r)$. Thus, $\mathcal{J}_{ess}(\mathbb{A}, \tilde{\mathbb{S}}) = 0$ and $\mathcal{J}_{ess}(\mathbb{A}^r, \tilde{\mathbb{S}}) = 0$. However, for (c), if $w(\pm(a' - a)) = 1$ and $\pm(a' - a) \notin CD(\mathbb{S}_o, \mathbb{S}_o^r)$, only the sensor pair (a, a') in $\tilde{\mathbb{S}}$ can generate the difference $\pm(a' - a)$. Thus, $\mathcal{J}_{ess}(\mathbb{A}, \tilde{\mathbb{S}}) = 1$ and $\mathcal{J}_{ess}(\mathbb{A}^r, \tilde{\mathbb{S}}) = 1$.

In Property 4, arbitrary sensors in \mathbb{S}_e are removed. If the removal of \mathbb{A} does not change the difference coarray $\tilde{\mathbb{D}}$, then removing the corresponding \mathbb{A}^r will not change $\tilde{\mathbb{D}}$ as well. Here we can take Proposition 3.3.1 as the reference.



Proposition 3.3.3. Based on $\mathbb{D} = \tilde{\mathbb{D}}$, the generalized 1-fragility with respect to the symmetrical array $\tilde{\mathbb{S}}$ and two importance functions \mathcal{J}_{ess} and $\mathcal{J}_{\mathbb{U}}$ have a certain range. They are shown as follows.

1. $\frac{2}{|\tilde{\mathbb{S}}|} \leq \mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{J}_{ess}) \leq \frac{|\mathbb{S}_e|}{|\tilde{\mathbb{S}}|}$.
2. $0 \leq \mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{J}_{\mathbb{U}}) \leq \frac{|\mathbb{S}_e|}{|\tilde{\mathbb{S}}|} - \frac{|\mathbb{S}_e|}{|\tilde{\mathbb{U}}| \times |\tilde{\mathbb{S}}|}$.

For the first case, through Definition 2.5.1, $\mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{J}_{ess})$ is defined as

$$\mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{J}_{ess}) = \sum_{\substack{\mathbb{A} \subseteq \tilde{\mathbb{S}}, |\mathbb{A}|=1}} \frac{\mathcal{J}_{ess}(\mathbb{A})}{\binom{|\tilde{\mathbb{S}}|}{1}}. \quad (3.15)$$

According to (3.5), $\tilde{\mathbb{S}}$ can be decomposed into three subsets. Hence, $\mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{J}_{ess})$ can be represented as the addition of three parts, like

$$\mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{J}_{ess}) = \sum_{\substack{\mathbb{A} \subseteq \mathbb{S}_o \subseteq \tilde{\mathbb{S}} \\ |\mathbb{A}|=1}} \frac{\mathcal{J}_{ess}(\mathbb{A})}{|\tilde{\mathbb{S}}|} + \sum_{\substack{\mathbb{A} \subseteq \mathbb{S}_e \subseteq \tilde{\mathbb{S}} \\ |\mathbb{A}|=1}} \frac{\mathcal{J}_{ess}(\mathbb{A})}{|\tilde{\mathbb{S}}|} + \sum_{\substack{\mathbb{A} \subseteq \mathbb{S}_o^r \subseteq \tilde{\mathbb{S}} \\ |\mathbb{A}|=1}} \frac{\mathcal{J}_{ess}(\mathbb{A})}{|\tilde{\mathbb{S}}|}. \quad (3.16)$$

Here we use the Proposition 3.3.2 to simplify (3.16). First, we consider the sensors in \mathbb{S}_o . Property 1, and 2 in Proposition 3.3.2 tell us that if $\pm(a' - a) \in \mathbb{D}$, then both a and a' can be removed from $\tilde{\mathbb{S}}$ and that does not impact $\tilde{\mathbb{D}}$. Since we have already assumed that $\mathbb{D} = \tilde{\mathbb{D}}$, every $\pm(a' - a)$ which is generated by different a are definitely in \mathbb{D} . Then we can get $\mathcal{J}_{ess}(\mathbb{A}, \tilde{\mathbb{S}}) = \mathcal{J}_{ess}(\mathbb{A}', \tilde{\mathbb{S}}) = 0$, where $\mathbb{A} \subseteq \mathbb{S}_o$, $|\mathbb{A}| = 1$. Therefore, the following two summations are equal to 0.

$$\sum_{\substack{\mathbb{A} \subseteq \mathbb{S}_o \subseteq \tilde{\mathbb{S}} \\ |\mathbb{A}|=1}} \frac{\mathcal{J}_{ess}(\mathbb{A})}{|\tilde{\mathbb{S}}|} = 0 \quad \text{and} \quad \sum_{\substack{\mathbb{A} \subseteq \mathbb{S}_o^r \subseteq \tilde{\mathbb{S}} \\ |\mathbb{A}|=1}} \frac{\mathcal{J}_{ess}(\mathbb{A})}{|\tilde{\mathbb{S}}|} = 0. \quad (3.17)$$

Eq.(3.16) can be rewrote as

$$\mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{I}_{ess}) = \sum_{\substack{\mathbb{A} \subseteq \mathbb{S}_e \subseteq \tilde{\mathbb{S}} \\ |\mathbb{A}|=1}} \frac{\mathcal{I}_{ess}(\mathbb{A})}{|\tilde{\mathbb{S}}|}. \quad (3.18)$$

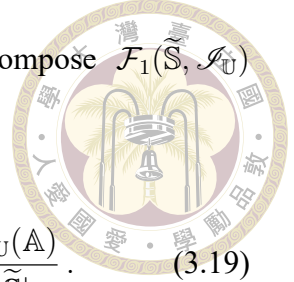


Eq.(3.18) indicates that $\mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{I}_{ess})$ is only related to the essentialness property of the sensors in \mathbb{S}_e . The least robust case of (3.18) is that all of the considered \mathbb{A} are 1-essential. Then $\mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{I}_{ess})$ will reach its upper bound $|\mathbb{S}_e| / |\tilde{\mathbb{S}}|$. Here we look at an example.

Example 3.3.1. We consider a MRA that $\mathbb{S} = \{0, 1, 3, 6, 13, 20, 27, 31, 35, 36\}$. Its difference coarray is hole-free, so $\mathbb{D} = \tilde{\mathbb{D}}$. The even subset $\mathbb{S}_e = \{0, 1, 35, 36\}$, and the symmetrical array $\tilde{\mathbb{S}} = \{0, 1, 3, 5, 6, 9, 13, 16, 20, 23, 27, 30, 31, 33, 35, 36\}$. If checking the essentialness for every sensor in \mathbb{S}_e based on $\tilde{\mathbb{S}}$, we can find that all of them are 1-essential. Thus, $\mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{I}_{ess}) = 4/16$, which is equal to the upper bound.

On the other hand, the most robust case of (3.18) is that all of the considered \mathbb{A} can be removed from $\tilde{\mathbb{S}}$ except for two sensors $\min(\mathbb{S}_e)$ and $\max(\mathbb{S}_e)$. Then $\mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{I}_{ess})$ will reach its lower bound $2 / |\tilde{\mathbb{S}}|$. Here we look at an example.

Example 3.3.2. An array $\mathbb{S} = \{0, 1, 3, 5, 8, 9, 10, 12\}$ is considered. The difference coarray is hole-free so that $\mathbb{D} = \tilde{\mathbb{D}}$. The even subset $\mathbb{S}_e = \{0, 3, 9, 12\}$, and the symmetrical array $\tilde{\mathbb{S}} = \{0, 1, 2, 3, 4, 5, 7, 8, 9, 11, 12\}$. For all the sensors in \mathbb{S}_e , it can be found that only the sensors $\min(\mathbb{S}_e)$ and $\max(\mathbb{S}_e)$ are 1-essential with respect to $\tilde{\mathbb{S}}$. Thus, $\mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{I}_{ess}) = 2/10$, which is equal to the lower bound.



For the second case of Proposition 3.3.3, we can similarly decompose $\mathcal{F}_1(\tilde{\mathcal{S}}, \mathcal{I}_{\tilde{\mathcal{U}}})$ into the addition of three parts:

$$\mathcal{F}_1(\tilde{\mathcal{S}}, \mathcal{I}_{\tilde{\mathcal{U}}}) = \sum_{\substack{\mathbb{A} \subseteq \mathcal{S}_o \subseteq \tilde{\mathcal{S}} \\ |\mathbb{A}|=1}} \frac{\mathcal{I}_{\tilde{\mathcal{U}}}(\mathbb{A})}{|\tilde{\mathcal{S}}|} + \sum_{\substack{\mathbb{A} \subseteq \mathcal{S}_e \subseteq \tilde{\mathcal{S}} \\ |\mathbb{A}|=1}} \frac{\mathcal{I}_{\tilde{\mathcal{U}}}(\mathbb{A})}{|\tilde{\mathcal{S}}|} + \sum_{\substack{\mathbb{A} \subseteq \mathcal{S}_o^r \subseteq \tilde{\mathcal{S}} \\ |\mathbb{A}|=1}} \frac{\mathcal{I}_{\tilde{\mathcal{U}}}(\mathbb{A})}{|\tilde{\mathcal{S}}|}. \quad (3.19)$$

We have already known that $\mathcal{I}_{ess}(\mathbb{A}, \tilde{\mathcal{S}}) = \mathcal{I}_{ess}(\mathbb{A}', \tilde{\mathcal{S}}) = 0$, where $\mathbb{A} \subseteq \mathcal{S}_o$ and $|\mathbb{A}| = 1$. Thus, these \mathbb{A} can be removed from $\tilde{\mathcal{S}}$ and does not change $\tilde{\mathcal{U}}$. So that the equation $\mathcal{I}_{\tilde{\mathcal{U}}}(\mathbb{A}, \tilde{\mathcal{S}}) = \mathcal{I}_{\tilde{\mathcal{U}}}(\mathbb{A}', \tilde{\mathcal{S}}) = 0$ also holds true. Then we can rewrite (3.19) as

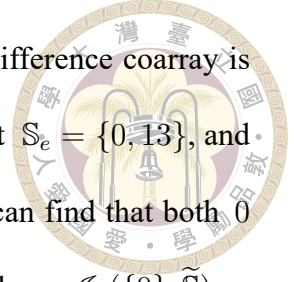
$$\mathcal{F}_1(\tilde{\mathcal{S}}, \mathcal{I}_{\tilde{\mathcal{U}}}) = \sum_{\substack{\mathbb{A} \subseteq \mathcal{S}_e \subseteq \tilde{\mathcal{S}} \\ |\mathbb{A}|=1}} \frac{\mathcal{I}_{\tilde{\mathcal{U}}}(\mathbb{A})}{|\tilde{\mathcal{S}}|}. \quad (3.20)$$

The least robust case of (3.20) is that the remaining $\tilde{\mathcal{U}}$ only have one element $\{0\}$ no matter which sensor we remove from \mathcal{S}_e , where $\mathcal{S}_e \subseteq \tilde{\mathcal{S}}$. In this case, the importance function of all the sensors in \mathcal{S}_e are $\mathcal{I}_{\tilde{\mathcal{U}}}(\mathbb{A}) = 1 - 1/|\tilde{\mathcal{U}}|$. If adding all these values, then the result is

$$\sum_{\substack{\mathbb{A} \subseteq \mathcal{S}_e \subseteq \tilde{\mathcal{S}} \\ |\mathbb{A}|=1}} \frac{\mathcal{I}_{\tilde{\mathcal{U}}}(\mathbb{A})}{|\tilde{\mathcal{S}}|} = \frac{|\mathcal{S}_e|(1 - 1/|\tilde{\mathcal{U}}|)}{|\tilde{\mathcal{S}}|} = \frac{|\mathcal{S}_e|}{|\tilde{\mathcal{S}}|} - \frac{|\mathcal{S}_e|}{|\tilde{\mathcal{U}}| \times |\tilde{\mathcal{S}}|}. \quad (3.21)$$

This case usually happens when $\tilde{\mathcal{U}} = \{0\}$.

As for the most robust case of (3.20), if $\tilde{\mathcal{U}}$ will not be changed after removing any sensors from \mathcal{S}_e , then $\mathcal{I}_{\tilde{\mathcal{U}}}(\mathbb{A}) = 0$ for all sensors in \mathcal{S}_e . This case only happens when $\tilde{\mathcal{D}}$ is not a hole-free difference coarray. Because if $\tilde{\mathcal{D}}$ is hole-free, then removing $\min(\mathcal{S}_e)$ or $\max(\mathcal{S}_e)$ must change $\tilde{\mathcal{U}}$. Let us look at an example that $\mathcal{F}_1(\tilde{\mathcal{S}}, \mathcal{I}_{\tilde{\mathcal{U}}}) = 0$.



Example 3.3.3. We consider the array $\mathbb{S} = \{0, 2, 6, 8, 9, 13\}$. Its difference coarray is not hole-free, but it still fits our assumption $\tilde{\mathbb{D}} = \mathbb{D}$. The even subset $\mathbb{S}_e = \{0, 13\}$, and the symmetrical array $\tilde{\mathbb{S}} = \{0, 2, 4, 5, 6, 7, 8, 9, 11, 13\}$. In \mathbb{S}_e , we can find that both 0 and 13 will not change $\tilde{\mathbb{U}}$ after removing one of them from $\tilde{\mathbb{S}}$. Thus, $\mathcal{I}_{\mathbb{U}}(\{0\}, \tilde{\mathbb{S}}) = \mathcal{I}_{\mathbb{U}}(\{13\}, \tilde{\mathbb{S}}) = 0$. According to (3.20), $\mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{I}_{\mathbb{U}}) = 0$, which is equal to the lower bound.

If we take the $\mathcal{F}_1(\mathbb{S}, \mathcal{I}_{ess})$ and $\mathcal{F}_1(\mathbb{S}, \mathcal{I}_{\mathbb{U}})$ as the robustness metrics, it is worth nothing that $\tilde{\mathbb{S}}$ is more robust than \mathbb{S} . The relationship can be proved based on the assumption $\mathbb{D} = \tilde{\mathbb{D}}$. Similar to (3.16) and (3.19), we also decompose the generalized 1-fragility of \mathbb{S} into two parts:

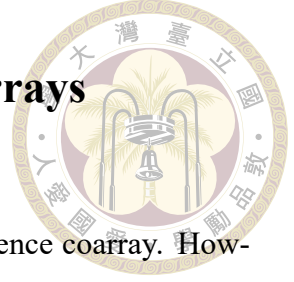
$$\mathcal{F}_1(\mathbb{S}, \mathcal{I}_{ess}) = \sum_{\substack{\mathbb{A} \subseteq \mathbb{S}_o \subseteq \mathbb{S} \\ |\mathbb{A}|=1}} \frac{\mathcal{I}_{ess}(\mathbb{A})}{|\mathbb{S}|} + \sum_{\substack{\mathbb{A} \subseteq \mathbb{S}_e \subseteq \mathbb{S} \\ |\mathbb{A}|=1}} \frac{\mathcal{I}_{ess}(\mathbb{A})}{|\mathbb{S}|}. \quad (3.22)$$

$$\mathcal{F}_1(\mathbb{S}, \mathcal{I}_{\mathbb{U}}) = \sum_{\substack{\mathbb{A} \subseteq \mathbb{S}_o \subseteq \mathbb{S} \\ |\mathbb{A}|=1}} \frac{\mathcal{I}_{\mathbb{U}}(\mathbb{A})}{|\mathbb{S}|} + \sum_{\substack{\mathbb{A} \subseteq \mathbb{S}_e \subseteq \mathbb{S} \\ |\mathbb{A}|=1}} \frac{\mathcal{I}_{\mathbb{U}}(\mathbb{A})}{|\mathbb{S}|}. \quad (3.23)$$

Both (3.16) and (3.22) have one term which considers the sensors in \mathbb{S}_e . We know that the number of sensors $|\mathbb{S}| \leq |\tilde{\mathbb{S}}|$, so definitely

$$\sum_{\substack{\mathbb{A} \subseteq \mathbb{S}_e \subseteq \mathbb{S} \\ |\mathbb{A}|=1}} \frac{\mathcal{I}_{ess}(\mathbb{A})}{|\mathbb{S}|} \geq \sum_{\substack{\mathbb{A} \subseteq \mathbb{S}_e \subseteq \tilde{\mathbb{S}} \\ |\mathbb{A}|=1}} \frac{\mathcal{I}_{ess}(\mathbb{A})}{|\tilde{\mathbb{S}}|}. \quad (3.24)$$

Therefore, based on (3.17) and (3.24), we can obtain $\mathcal{F}_1(\mathbb{S}, \mathcal{I}_{ess}) > \mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{I}_{ess})$. For the same reason, the inequality of generalized 1-fragility with respect to \mathbb{U} is available. That is, $\mathcal{F}_1(\mathbb{S}, \mathcal{I}_{\mathbb{U}}) > \mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{I}_{\mathbb{U}})$. In the next section, we will show that the symmetrical coprime arrays can reach the lower bound of $\mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{I}_{ess})$ and $\mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{I}_{\mathbb{U}})$.



3.4 Robustness of Symmetrical Coprime Arrays

Generally, $\mathbb{D} = \tilde{\mathbb{D}}$ only happens when \mathbb{D} is a hole-free difference coarray. However, as we mentioned in Section 2.3, the difference coarray of coprime arrays is not hole-free. Nevertheless, in this section, we will prove that $\mathbb{D}_{\text{coprime}} = \tilde{\mathbb{D}}_{\text{coprime}}$ even if $\mathbb{D}_{\text{coprime}}$ is not hole-free. With this result, we will prove that the generalized 1-fragility of symmetrical coprime arrays can reach the lower bound:

$$\mathcal{F}_1(\tilde{\mathbb{S}}_{\text{coprime}}, \mathcal{I}_{\text{ess}}) = \frac{2}{|\tilde{\mathbb{S}}_{\text{coprime}}|} \quad \text{and} \quad \mathcal{F}_1(\tilde{\mathbb{S}}_{\text{coprime}}, \mathcal{I}_{\text{U}}) = 0. \quad (3.25)$$

Proposition 3.4.1. After symmetrizing the coprime arrays, the difference coarray will not be changed. Namely, $\mathbb{D}_{\text{coprime}} = \tilde{\mathbb{D}}_{\text{coprime}}$.

We will explain $CD(\mathbb{S}_o, \mathbb{S}_o^r) \subseteq \mathbb{D}$, and then through the proof of (3.9), we can obtain Proposition 3.4.1.

First, we build a $|\mathbb{S}_o| \times |\mathbb{S}_o|$ matrix \mathbf{C} to store half of the elements in the set $CD(\mathbb{S}_o, \mathbb{S}_o^r)$. Assume that $\min(\mathbb{S}) + \max(\mathbb{S}) = Q$, the elements in \mathbf{C} is defined as

$$[\mathbf{C}]_{i,j} = Q - n_i - n_j, \quad (3.26)$$

where n_i and n_j are the sensors in \mathbb{S}_o . According to (2.22) and Definition 3.1.2, we know that the size of the \mathbb{S}_o of coprime arrays is equal to $(N - 1)M$. The sensors in \mathbb{S}_o can be denoted by $\mathbb{S}_o = \{n_1, n_2, \dots, n_{(N-1)M}\}$, and they satisfy $n_1 < n_2 < \dots < n_{(N-1)M}$. Because of the symmetric property of difference coarrays in Definition 2.2.1, another half of the elements in the set $CD(\mathbb{S}_o, \mathbb{S}_o^r)$ is $-\mathbf{C}$. The matrix \mathbf{C} can be used in not only

coprime arrays but also any other arrays. For coprime arrays, \mathbf{C} is equal to

$$\begin{bmatrix} (2M-1)N-2M & (2M-1)N-3M & \cdots & (2M-1)N-NM \\ (2M-1)N-3M & (2M-1)N-4M & \cdots & (2M-1)N-(N+1)M \\ \vdots & \vdots & \ddots & \vdots \\ (2M-1)N-NM & (2M-1)N-(N+1)M & \cdots & (2M-1)N-2(N-1)M \end{bmatrix} \quad (3.27)$$

This matrix can be observed that it is a Hankel matrix. Hence, there are many repetitive elements in the matrix. If each element is considered once only, then the set which is composed of these elements can be denoted as

$$\{(2M-1)N-m \mid m \in \{2M, 3M, \dots, NM, (N+1)M, \dots, 2(N-1)M\}\}. \quad (3.28)$$

Now we can combine the other part differences in $-\mathbf{C}$ so that all the elements in $CD(\mathbb{S}_o, \mathbb{S}_o^r)$ of coprime arrays are

$$\{\pm[(2M-1)N-m] \mid m \in \{2M, 3M, \dots, NM, (N+1)M, \dots, 2(N-1)M\}\}. \quad (3.29)$$

If all the elements in (3.29) can be found in $\mathbb{D}_{\text{coprime}}$, then the equation $\mathbb{D}_{\text{coprime}} = \widetilde{\mathbb{D}}_{\text{coprime}}$ holds true. The following is our discussion.

Picking the element $(2M-1)N$ in \mathbb{S}_e and the element in

$$\{2M, 3M, \dots, (N-1)M\} \in \mathbb{S}_o \quad (3.30)$$

can generate the differences

$$\{(2M-1)N-2M, (2M-1)N-3M, \dots, (2M-1)N-(N-1)M\}. \quad (3.31)$$



Thus, (3.31) belongs to $CD(\mathbb{S}_o, \mathbb{S}_e)$. Also, the difference between $(M-1)N \in \mathbb{S}_e$ and $0 \in \mathbb{S}_e$ is

$$(M-1)N = (2M-1)N - NM, \tag{3.32}$$

which is just the element in (3.28). It means that $(2M-1)N - NM \in SD(\mathbb{S}_e)$. Moreover, through (3.32), the element $(M-1)N$ and the element in

$$\{M, 2M, \dots, (N-2)M\} \in \mathbb{S}_o \tag{3.33}$$

can generate the difference

$$\{(2M-1)N - (N+1)M, (2M-1)N - (N+2)M, \dots, (2M-1)N - 2(N-1)M\}. \tag{3.34}$$

Thus, (3.34) also belongs to $CD(\mathbb{S}_o, \mathbb{S}_e)$, and they complement the remaining elements in (3.28). Due to the symmetric property of difference coarrays, (2.11), all of the elements in (3.29) can be generated at the same time. Therefore, we have explained $CD(\mathbb{S}_o, \mathbb{S}_o^r) \subseteq \mathbb{D}$. Through the proof of (3.9), $\mathbb{D}_{\text{coprime}} = \tilde{\mathbb{D}}_{\text{coprime}}$ can be obtained.

Proposition 3.4.2. The generalized 1-fragility with respect to \mathcal{J}_{ess} of symmetrical coprime arrays can reach the lower bound and also can be represented as a closed form.

Namely, $\mathcal{F}_1(\tilde{\mathbb{S}}_{\text{coprime}}, \mathcal{J}_{ess}) = \frac{2}{|\tilde{\mathbb{S}}_{\text{coprime}}|} = \frac{1}{N+M-1}$.

According to (3.18), here we need to focus on the elements in \mathbb{S}_e . Removing an element in \mathbb{S}_e from $\tilde{\mathbb{S}}$ will affect the sets $SD(\mathbb{S}_e)$, $CD(\mathbb{S}_o, \mathbb{S}_e)$ and $CD(\mathbb{S}_e, \mathbb{S}_o^r)$. However, $CD(\mathbb{S}_o, \mathbb{S}_e)$ and $CD(\mathbb{S}_e, \mathbb{S}_o^r)$ would complement each other due to Lemma 3.2.2. It is not possible that $CD(\mathbb{S}_o, \mathbb{S}_e)$ and $CD(\mathbb{S}_e, \mathbb{S}_o^r)$ reduce the difference at the same time, because $CD(\mathbb{S}_o, \mathbb{S}_e) = CD(\mathbb{S}_e, \mathbb{S}_o^r)$ in Lemma 3.2.2 and the size of \mathbb{S}_e of

coprime arrays is definitely an even number. If recalling the sensor locations of coprime arrays, $\mathbb{S}_e = \{0, N, 2N, \dots, (2M - 1)N\}$, the size of \mathbb{S}_e is $(2M - 1) - 0 + 1 = 2M$, which is an even number. Moreover, \mathbb{S}_e of coprime arrays is an ULA with a spacing of N . Except for the sensors “min(\mathbb{S})” and “max(\mathbb{S})”, all the other sensors in ULA are inessential [18]. In conclusion, we can get

$$\mathcal{F}_1(\tilde{\mathbb{S}}_{\text{coprime}}, \mathcal{I}_{\text{ess}}) = \sum_{\substack{\mathbb{A} \subseteq \mathbb{S}_e \subseteq \tilde{\mathbb{S}} \\ |\mathbb{A}|=1}} \frac{\mathcal{I}_{\text{ess}}(\mathbb{A})}{|\tilde{\mathbb{S}}_{\text{coprime}}|} = \frac{2}{|\tilde{\mathbb{S}}_{\text{coprime}}|}. \quad (3.35)$$

From Section 2.3.2, we know the size of $\mathbb{S}_{\text{coprime}}$ is $N + 2M - 1$. Therefore, $\tilde{\mathbb{S}}_{\text{coprime}}$ can also be represented as a closed form with parameters M and N .

$$\begin{aligned} |\tilde{\mathbb{S}}_{\text{coprime}}| &= |\mathbb{S}| + |\mathbb{S}_e^r| \\ &= (N + 2M - 1) + (N - 1) \\ &= 2N + 2M - 2. \end{aligned} \quad (3.36)$$

Then we can further simplify (3.35) to the following form.

$$\mathcal{F}_1(\tilde{\mathbb{S}}_{\text{coprime}}, \mathcal{I}_{\text{ess}}) = \frac{2}{|\tilde{\mathbb{S}}_{\text{coprime}}|} = \frac{2}{2(N + M - 1)} = \frac{1}{N + M - 1}. \quad (3.37)$$

Proposition 3.4.3. The generalized 1-fragility with respect to $\mathcal{I}_{\mathbb{U}}$ of symmetrical coprime arrays can reach the lower bound, zero. Namely, $\mathcal{F}_1(\tilde{\mathbb{S}}_{\text{coprime}}, \mathcal{I}_{\mathbb{U}}) = 0$.

In Proposition 3.2.4, we mentioned a property that if $\mathbb{D} = \tilde{\mathbb{D}}$, then $\mathbb{U} = \tilde{\mathbb{U}}$. Thus, for coprime arrays,

$$\mathbb{U}_{\text{coprime}} = \tilde{\mathbb{U}}_{\text{coprime}} = \{0, \pm 1, \dots, \pm(MN + M - 1)\}. \quad (3.38)$$

Also, from Proposition 3.4.2, we know that only the sensors 0 and $(2M - 1)N$ in $\tilde{\mathcal{S}}_{\text{coprime}}$ are 1-essential. This indicates that removing one sensor from $\tilde{\mathcal{S}}_{\text{coprime}} \setminus \{0, (2M - 1)N\}$ does not change $\tilde{\mathcal{D}}_{\text{coprime}}$ and $\tilde{\mathcal{U}}_{\text{coprime}}$. Then we can get the following conclusion through Eq.(2.28).

$$\mathcal{I}_{\mathcal{U}}(\mathbb{A}, \tilde{\mathcal{S}}_{\text{coprime}}) = 0, \quad \text{for } |\mathbb{A}| = 1, \mathbb{A} \in \tilde{\mathcal{S}}_{\text{coprime}} \setminus \{0, (2M - 1)N\}. \quad (3.39)$$

The reason why 0 and $(2M - 1)N$ are 1-essential is because the difference between them, $(2M - 1)N$, will disappear after removing one of them. Through (3.14), we know that the difference coarray of removing 0 from $\tilde{\mathcal{S}}$ is the same as the difference coarray of removing $(2M - 1)N$ from $\tilde{\mathcal{S}}$. Therefore, the missing difference $(2M - 1)N$ does not change $\tilde{\mathcal{U}}_{\text{coprime}}$ in (3.38). Then we can rewrite (3.39) as

$$\mathcal{I}_{\mathcal{U}}(\mathbb{A}, \tilde{\mathcal{S}}_{\text{coprime}}) = 0, \quad \text{for } |\mathbb{A}| = 1, \mathbb{A} \in \tilde{\mathcal{S}}_{\text{coprime}}. \quad (3.40)$$

According to (3.40), the generalized 1-fragility with respect to $\mathcal{I}_{\mathcal{U}}$ of symmetrical coprime arrays is

$$\mathcal{F}_1(\tilde{\mathcal{S}}_{\text{coprime}}, \mathcal{I}_{\mathcal{U}}) = \sum_{\substack{\mathbb{A} \subseteq \mathcal{S}_e \subseteq \tilde{\mathcal{S}} \\ |\mathbb{A}|=1}} \frac{\mathcal{I}_{\mathcal{U}}(\mathbb{A})}{|\tilde{\mathcal{S}}_{\text{coprime}}|} = 0. \quad (3.41)$$



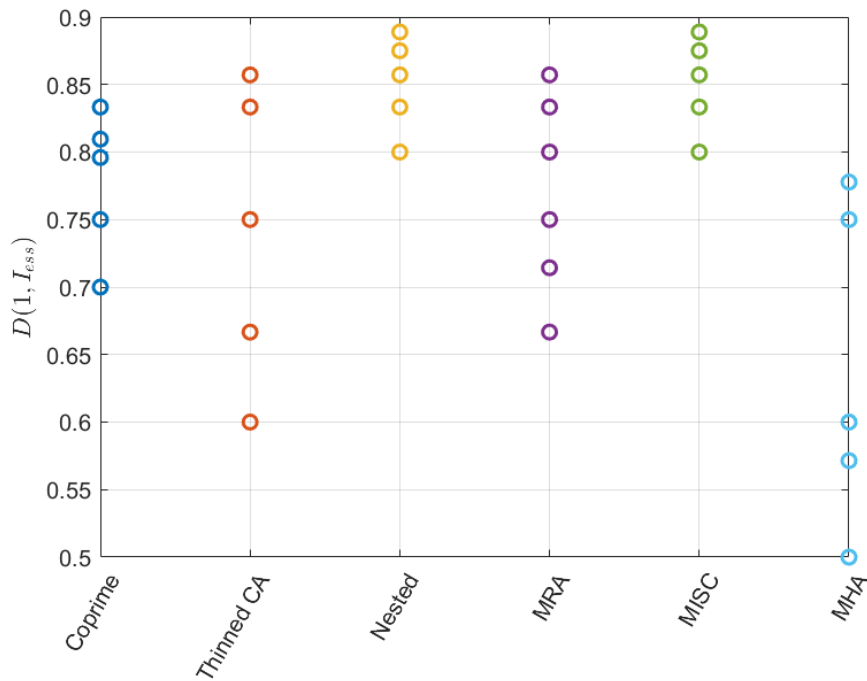
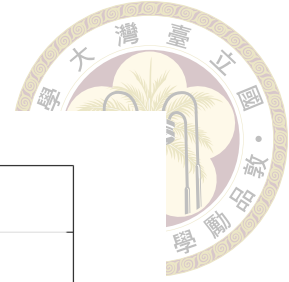
3.5 Numerical Comparison

Among several common metrics for measuring the robustness of an array, such as $\mathcal{F}_1(\mathbb{S}, \mathcal{I}_{ess})$, $\mathcal{F}_1(\mathbb{S}, \mathcal{I}_U)$, and P_c , we can intuitively think that the symmetrical array is more robust than its original array, since more sensors are placed in the same aperture. As the content we discussed in Section 3.3 and Section 3.4, if the same faulty sensors are removed from \mathbb{S} and $\tilde{\mathbb{S}}$, the impact on $\tilde{\mathbb{S}}$ is relatively small. Now we compare different arrays through an experiment. Besides the arrays we introduced in Section 2.3, we also consider other arrays in the following comparisons, including thinned coprime arrays [28], MISC arrays [29] and minimum hole arrays (MHA) [13]. Here we define an indicator that can measure the increasing degree of robustness after symmetrizing the array.

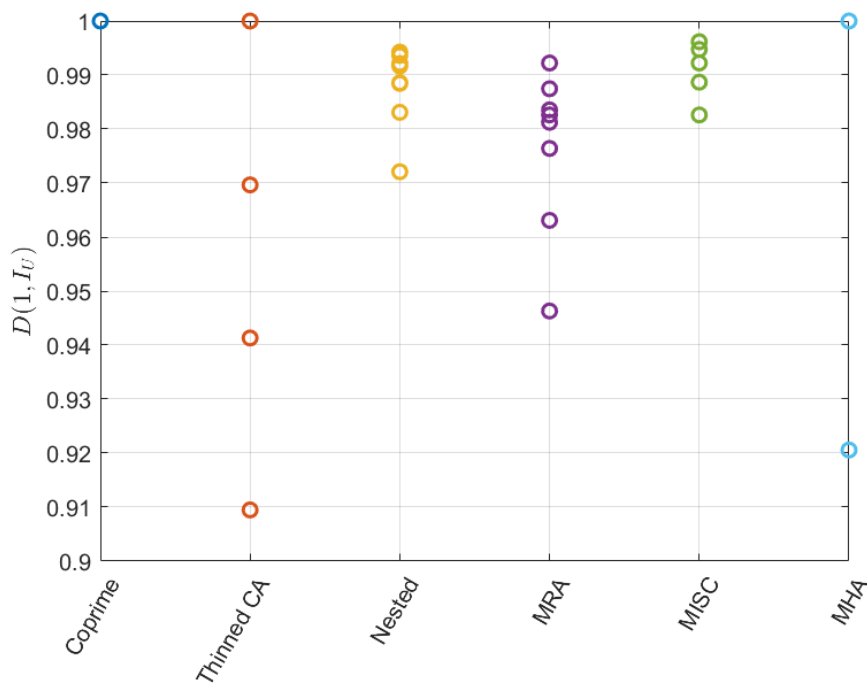
Definition 3.5.1. Given the array \mathbb{S} , the importance function \mathcal{I} and the value of k , then the increasing degree of robustness after symmetrizing the array is defined as

$$\mathcal{D}(k, \mathcal{I}) = \frac{\mathcal{F}_k(\mathbb{S}, \mathcal{I}) - \mathcal{F}_k(\tilde{\mathbb{S}}, \mathcal{I})}{\mathcal{F}_k(\mathbb{S}, \mathcal{I})}. \quad (3.42)$$

The comparison results are shown in Figure 3.3. We compare the values of $\mathcal{D}(1, \mathcal{I})$ of different arrays. Each vertical line in a plot represent the same type of arrays, and the circles with the same color are the arrays generated by different parameters under the same array definition. By (3.42), if $\mathcal{F}_k(\mathbb{S}, \mathcal{I}) > \mathcal{F}_k(\tilde{\mathbb{S}}, \mathcal{I})$, then $\mathcal{D}(k, \mathcal{I})$ will be greater than 0. Therefore, it can be observed in Figure 3.3 that all of the points for both $\mathcal{D}(1, \mathcal{I}_{ess})$ and $\mathcal{D}(1, \mathcal{I}_U)$ are greater than 0. For these arrays, we can say that symmetrizing them will increase the robustness based on generalized 1-fragility.

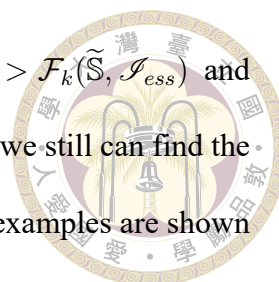


(a) $\mathcal{I} = \mathcal{I}_{ess}$.



(b) $\mathcal{I} = \mathcal{I}_U$.

Figure 3.3: $D(1, \mathcal{I})$ for six different types of arrays. They are coprime arrays, thinned coprime arrays, nested arrays, MRA, MISC arrays and MHA, respectively. The circles with the same color are the arrays generated by different parameters under the same array definition.



From the discussion below (3.24), we only sure that $\mathcal{F}_k(\mathbb{S}, \mathcal{I}_{ess}) > \mathcal{F}_k(\tilde{\mathbb{S}}, \mathcal{I}_{ess})$ and $\mathcal{F}_k(\mathbb{S}, \mathcal{I}_U) > \mathcal{F}_k(\tilde{\mathbb{S}}, \mathcal{I}_U)$ based on $\mathbb{D} = \tilde{\mathbb{D}}$. Nevertheless, if $\mathbb{D} \neq \tilde{\mathbb{D}}$, we still can find the special cases that $\mathcal{D}(k, \mathcal{I}) < 0$ with respect to \mathcal{I}_{ess} and \mathcal{I}_U . The examples are shown as follows.

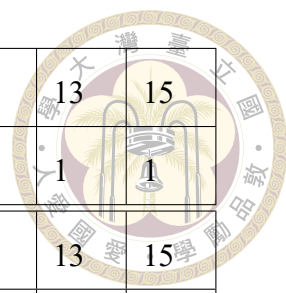
Example 3.5.1. We consider the array $\mathbb{S} = \{0, 3, 5, 8, 10, 13, 15\}$. Its corresponding symmetrical array is $\tilde{\mathbb{S}} = \{0, 2, 3, 5, 7, 8, 10, 12, 13, 15\}$. Table 3.1 shows the importance function \mathcal{I}_{ess} of each sensor in \mathbb{S} and $\tilde{\mathbb{S}}$. If summing up all the importance values based on the same array \mathbb{S} or $\tilde{\mathbb{S}}$, we can get the result $\mathcal{F}_1(\mathbb{S}, \mathcal{I}_{ess}) = 4/7$ and $\mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{I}_{ess}) = 6/10$. Because

$$\frac{4}{7} = 0.5714 < \frac{6}{10} = 0.6, \quad (3.43)$$

the generalized 1-fragility with respect to \mathcal{I}_{ess} is larger after symmetrizing the array.

Example 3.5.2. The same array $\mathbb{S} = \{0, 3, 5, 8, 10, 13, 15\}$ is considered. Table 3.2 shows the importance function \mathcal{I}_U of each sensor in \mathbb{S} and $\tilde{\mathbb{S}}$. If summing up them based on the same array \mathbb{S} or $\tilde{\mathbb{S}}$, then we can get the result that $\mathcal{F}_1(\mathbb{S}, \mathcal{I}_U) = 0$ and $\mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{I}_U) = 32/27$. It indicates the generalized 1-fragility with respect to \mathcal{I}_U is larger after symmetrizing the array.

For P_c , another robustness definition we introduced in Section 2.5, lower probability $\Pr[\tilde{\mathbb{D}} \neq \mathbb{D}]$ represents the array is more robust. We compare the same \mathbb{S} and $\tilde{\mathbb{S}}$ as Example 3.5.1. In Figure 3.4, we plot the curves of P_c based on (2.31) for \mathbb{S} and $\tilde{\mathbb{S}}$, respectively. We can find that no matter which p we take, P_c of $\tilde{\mathbb{S}}$ is higher than P_c of \mathbb{S} . That means $\tilde{\mathbb{S}}$ is less robust than \mathbb{S} based on the definition of P_c .



$n \in \mathbb{S}$	0	×	3	5	×	8	10	×	13	15
$\mathcal{I}_{ess}(\{n\})$	1	×	1	0	×	0	0	×	1	1
$n \in \tilde{\mathbb{S}}$	0	2	3	5	7	8	10	12	13	15
$\mathcal{I}_{ess}(\{n\})$	1	1	1	0	0	0	0	1	1	1

Table 3.1: The importance function \mathcal{I}_{ess} of each sensor in \mathbb{S} and $\tilde{\mathbb{S}}$.

$n \in \mathbb{S}$	0	×	3	5	×	8	10	×	13	15
$\mathcal{I}_{\mathbb{U}}(\{n\})$	0	×	0	0	×	0	0	×	0	0
$n \in \tilde{\mathbb{S}}$	0	2	3	5	7	8	10	12	13	15
$\mathcal{I}_{\mathbb{U}}(\{n\})$	0	$\frac{6}{27}$	$\frac{10}{27}$	0	0	0	0	$\frac{6}{27}$	$\frac{10}{27}$	0

Table 3.2: The importance function $\mathcal{I}_{\mathbb{U}}$ of each sensor in \mathbb{S} and $\tilde{\mathbb{S}}$.

In the next chapter, we will present a different view to mitigate this condition. Although it cannot be guaranteed that $\tilde{\mathbb{S}}$ is more robust than \mathbb{S} , different insights can be obtained through the observations of the size of \mathbb{U} . Moreover, we will illustrate the benefits of symmetrizing the arrays through more experiments.

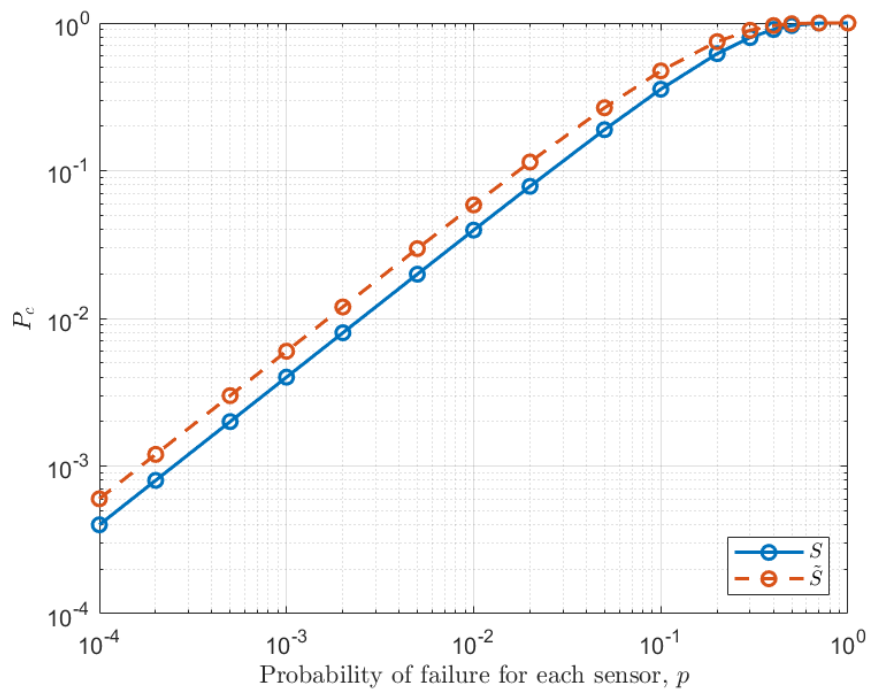


Figure 3.4: The comparison of P_c of \mathbb{S} and $\tilde{\mathbb{S}}$. We consider the array \mathbb{S} in Example 3.5.1 and plot the curves based on (2.31).





Chapter 4 Array Analysis with Random Sensor Failures

In Section 2.6, we introduced the DOA estimation algorithm, coarray MUSIC. It can utilize the data on $\bar{\mathbb{U}}$ to estimate the source directions. Therefore, larger size of $\bar{\mathbb{U}}$ indicates that more virtual sensors are used. In this Chapter, we will consider two factors, $|\bar{\mathbb{U}}|$ and p , at the same time. Both of them would impact the performance of the DOA estimation. Note that $\bar{\mathbb{U}}$ is the corresponding central ULA segment in the difference coarray of the sensor array $\bar{\mathbb{S}}$, where $\bar{\mathbb{S}} = \mathbb{S} \setminus \mathbb{A}$. Also, p is the probability of failure for each sensor. In Section 4.1, we will discuss the relationship between $|\bar{\mathbb{U}}|$ and the MSE, also p and the MSE, through the experiments. Actually, $|\bar{\mathbb{U}}|$ is a random variable because of the sensor failure probability p . Thus, in Section 4.2, we will derive the expected value expression of $|\bar{\mathbb{U}}|$ based on p . Furthermore, we will provide a robustness metric which related to $|\bar{\mathbb{U}}|$ and p at the same time. We can get a two-dimensional point with this robustness metric as x -axis and a performance metric as y -axis. That can be used for comparing different arrays. Therefore, in Section 4.2.3, we will have some numerical results to compare the arrays, verify the expression we have derived, and see the benefits of the symmetrical arrays. Finally, in Section 4.3, we will focus on the analysis of the ULA with different number of sensors.



4.1 The Factors Impacting MSE

In Chapter 4, we will use the coarray MUSIC to perform many experiments. Therefore, here we define the empirical MSE to evaluate the simulation performance.

$$\text{MSE} = \frac{1}{DL} \sum_{l=1}^L \sum_{i=1}^D \left(\hat{\theta}_i^{(l)} - \theta_i^{(l)} \right)^2, \quad (4.1)$$

where D is the number of sources and L is the number of trials.

4.1.1 Size of $\bar{\mathbb{U}}$ versus MSE

We consider an array with 10 sensors that $\mathbb{S} = \{0, 1, 2, 5, 8, 9, 12, 15, 16, 17\}$. Assume that we have already known there are 3 faulty sensors in \mathbb{S} . Thus, here the set \mathbb{A} which consist of the faulty sensors can be denoted by

$$\mathbb{A} = \{n_1, n_2, n_3 \mid n_1 \neq n_2, n_1 \neq n_3, n_2 \neq n_3 \text{ and } n_1, n_2, n_3 \in \mathbb{S}\}. \quad (4.2)$$

Every case of \mathbb{A} will be considered once so that there are $\binom{10}{3} = 120$ different \mathbb{A} in total. Then we will use the $\bar{\mathbb{S}}$ to perform the DOA estimation, so there are 120 different $\bar{\mathbb{S}}$ as well, since $\bar{\mathbb{S}} = \mathbb{S} \setminus \mathbb{A}$. Moreover, $\bar{\mathbb{U}}$ is obtained from $\bar{\mathbb{S}}$. Here SNR is 0 dB and the number of snapshots is 500. For each case, we will do 300 Monte-Carlo runs and then take the average. The result is shown as Figure 4.1 and we have two observations:

1. For two arrays with the same number of sensors and size of $\bar{\mathbb{U}}$, the estimation results will be different. The reason is that the weight functions of them are different.
2. There is a banded inverse relationship between the size of $\bar{\mathbb{U}}$ and the MSE.

To find a more specific relationship between the size of \bar{U} and the MSE, we can fit these points to a straight line by the least square approximation. Assume that these points are

$$(\log_{10} x_1, \log_{10} y_1), (\log_{10} x_2, \log_{10} y_2), \dots, (\log_{10} x_{120}, \log_{10} y_{120}). \quad (4.3)$$

The straight line in Figure 4.1 can be expressed as

$$\log_{10} y = a + b \log_{10} x. \quad (4.4)$$

If substitute the points in (4.3) into (4.4), we can obtain that $\log_{10} y_i = a + b \log_{10} x_i$ for $i = 1, 2, \dots, 120$. Hence, we need to solve

$$\underbrace{\begin{bmatrix} 1 & \log_{10} x_1 \\ 1 & \log_{10} x_2 \\ \vdots & \vdots \\ 1 & \log_{10} x_{120} \end{bmatrix}}_{\mathbf{D}} \underbrace{\begin{bmatrix} a \\ b \end{bmatrix}}_{\mathbf{e}} \approx \underbrace{\begin{bmatrix} \log_{10} y_1 \\ \log_{10} y_2 \\ \vdots \\ \log_{10} y_{120} \end{bmatrix}}_{\mathbf{e}}. \quad (4.5)$$

Since \mathbf{D} is full column rank, (a, b) can be obtained by the least square approximation, which is shown as follows.

$$\begin{bmatrix} a \\ b \end{bmatrix} = (\mathbf{D}^T \mathbf{D})^{-1} \mathbf{D}^T \mathbf{e}. \quad (4.6)$$

For the data points in Figure 4.1, a is equal to -2.3620 and b is equal to -2.4966 , so we can get

$$\begin{aligned} \log_{10} \text{MSE} &\approx -2.3620 - 2.4966 \log_{10} |\bar{U}| \\ &\rightarrow \text{MSE} \approx 10^{-2.3620} \cdot |\bar{U}|^{-2.4966}. \end{aligned} \quad (4.7)$$

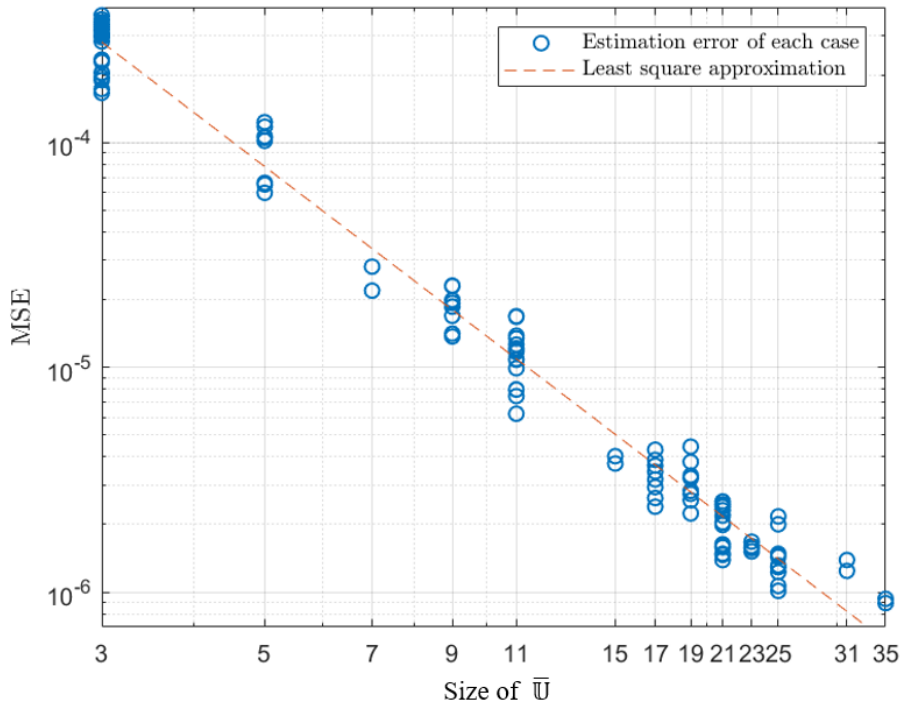
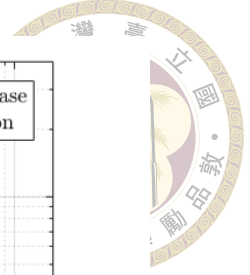


Figure 4.1: Estimation error versus size of \bar{U} . \bar{U} is obtained from $\bar{S} = S \setminus A$, where $S = \{0, 1, 2, 5, 8, 9, 12, 15, 16, 17\}$ and $A = \{n_1, n_2, n_3 \mid n_1 \neq n_2, n_1 \neq n_3, n_2 \neq n_3 \text{ and } n_1, n_2, n_3 \in S\}$. SNR is 0 dB and the number of snapshots is 500. There is one source with $\theta = \pi/4$. Each point is averaged from 300 Monte-Carlo runs. The dotted line is the least square approximation of these points based on (4.4) to (4.7).

Therefore, we can generally assume that an array S which has large size of \bar{U} can obtain better estimation performance. The advantage of using single indicator $|\bar{U}|$ as performance metric is that it can simplify the comparisons of arrays. Furthermore, we can consider random sensor failures at the same time.

4.1.2 Sensor Failure Probability versus MSE

As we mentioned in Section 2.4, sensor failure probability p increases with the usage time of the array. For different values of p , the estimation performance of different arrays would be changed. Therefore, we conduct an experiment to see the relationship between the failure probability for each sensor, p , and the estimation performance, MSE. Here we consider four arrays which have introduced in Section 2.3. Table 4.1 shows their \bar{U} size


Arrays	Size of $\bar{\mathcal{U}}$, when $p = 0$	$\mathcal{F}_1(\mathcal{S}, \mathcal{I}_{ess})$	$\mathcal{F}_1(\mathcal{S}, \mathcal{I}_{\mathcal{U}})$
(a) MRA $\mathcal{S} = \{0, 1, 4, 5, 11, 13\}$	27	1	0.7901
(b) Nested Array $\mathcal{S} = \{1, 2, 3, 4, 8, 12\}$	23	1	0.6232
(c) Coprime Array $\mathcal{S} = \{0, 2, 3, 4, 6, 9\}$	15	0.8333	0.3111
(d) ULA $\mathcal{S} = \{0, 1, 2, 3, 4, 5\}$	11	0.3333	0.0606

Table 4.1: The comparison of four arrays, each of them has 6 sensors. The second column is their corresponding $|\bar{\mathcal{U}}|$ when $p = 0$. The third and fourth column are their generalized 1-fragility with respect to \mathcal{I}_{ess} and $\mathcal{I}_{\mathcal{U}}$, respectively.

when $p = 0$, namely their original \mathcal{U} size. Also, we have their generalized 1-fragility with respect to two importance functions, \mathcal{I}_{ess} and $\mathcal{I}_{\mathcal{U}}$.

Assume that there is one source $\theta = \pi/4$. SNR is 0 dB and the number of snapshots is 500. We consider 11 different values of p , where $p \in \{10^{-4}, 2 \cdot 10^{-4}, 5 \cdot 10^{-4}, 10^{-3}, 2 \cdot 10^{-3}, 5 \cdot 10^{-3}, 10^{-2}, 2 \cdot 10^{-2}, 5 \cdot 10^{-2}, 0.1, 0.2\}$. In the beginning, random faulty sensors will be removed from the array \mathcal{S} with a fixed probability p . The set \mathcal{A} is composed of these faulty sensors. Then we will use the $\bar{\mathcal{S}}$, defined as $\mathcal{S} \setminus \mathcal{A}$, to do 100 Monte-Carlo runs. Next, the random faulty sensors will be regenerated for another 100 Monte-Carlo runs. This action will be repeated 5000 times. That is, each data point is averaged from $5 \cdot 10^5$ Monte-Carlo runs. The results are shown in Figure 4.2.

Figure 4.2 can be divided into four regions. In Region (I), the MRA possess the smallest MSE. The nested array own the second smallest MSE, then the coprime array, and finally the ULA. They are sorted in order of the size of \mathcal{U} . Since p is sufficiently small, few sensors will be removed in each run. Thus, $|\bar{\mathcal{U}}| \approx |\mathcal{U}|$, and more virtual



sensors in the MRA are used. In Region (II), the nested array possess the smallest MSE. Although the size of $\mathbb{U}_{\text{nested}}$ is smaller than the size of \mathbb{U}_{MRA} , the estimation performance is possible to be different when the value of p increases. It can be observed in Table 4.1 that $\mathcal{F}_1(\mathbb{S}, \mathcal{I}_{\mathbb{U}})$ of the nested array is smaller than $\mathcal{F}_1(\mathbb{S}, \mathcal{I}_{\mathbb{U}})$ of the MRA. Thus, we can deduce that the robustness has helped to improve the estimation performance. As the value of p continuously go up, in Region (III), the coprime array possess the smallest MSE. Its $\mathcal{F}_1(\mathbb{S}, \mathcal{I}_{\mathbb{U}})$ is smaller than $\mathcal{F}_1(\mathbb{S}, \mathcal{I}_{\mathbb{U}})$ of the nested array and the MRA. Therefore, the least MSE of the coprime array itself is sustained longer. Finally, in Region (IV), p is quite large. The estimation performance is sorted of the robustness metric $\mathcal{F}_1(\mathbb{S}, \mathcal{I}_{\mathbb{U}})$ instead of the size of \mathbb{U} . Therefore, the ULA own the least MSE, and the MSE of the MRA is the highest. In particular, in Table 4.1, we can observe that the value of $\mathcal{F}_1(\mathbb{S}, \mathcal{I}_{\text{ess}})$ of the MRA and the nested array are equal. It is not consistent with the simulation result. Hence, the robustness metrics should be designed based on the size of \mathbb{U} . Moreover, both of the robustness and the size of \mathbb{U} need to be considered when designing array geometries.

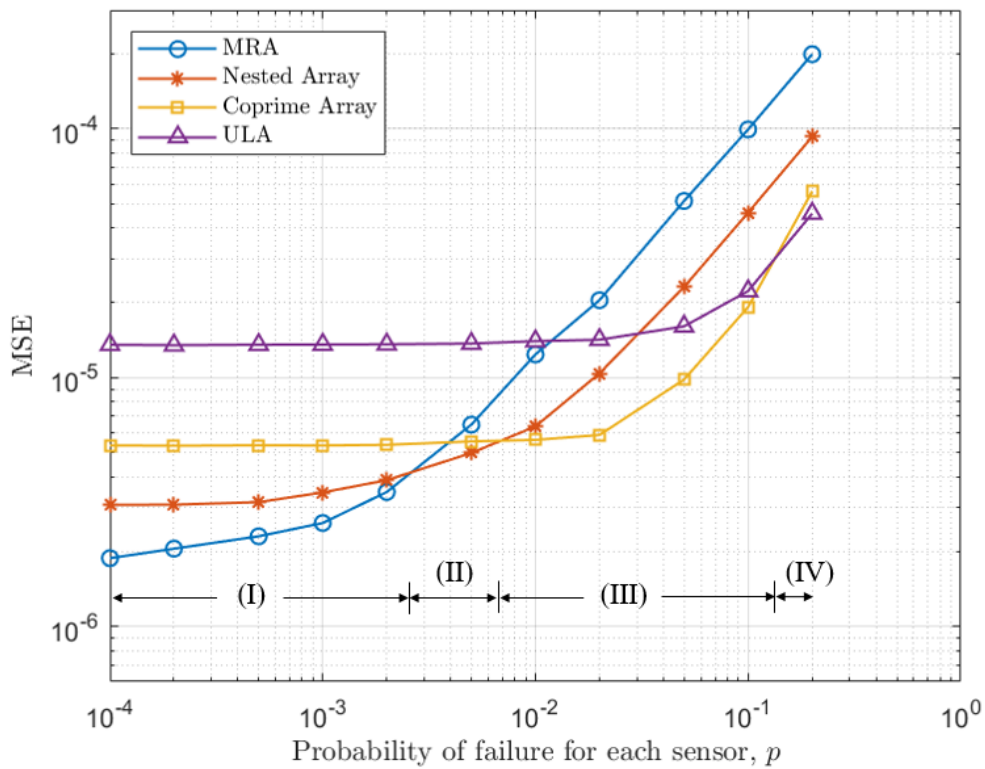


Figure 4.2: Estimation error versus probability of failure for each sensor, p . Here we consider four arrays in Table 4.1. SNR is 0 dB and the number of snapshots is 500. There is one source with $\theta = \pi/4$. Each point is averaged from $5 \cdot 10^5$ Monte-Carlo runs.



4.2 Derivation and Comparison

4.2.1 Expected Value of \bar{U} Size

In this section, we will derive the expression of $\mathbb{E} [|\bar{U}|]$. It combines two factors impacting the estimation MSE, which we introduced in Section 4.1.1 and Section 4.1.2. Here the set \mathbb{A} that consists of the faulty sensors is stochastic because every sensor in the array \mathbb{S} fails independently with the same probability. Also, the set $\bar{\mathbb{S}}$, defined as $\mathbb{S} \setminus \mathbb{A}$, and the corresponding \bar{U} are stochastic as well. The size of \bar{U} now become a random variable, which maps \bar{U} to a real number. The definition is defined as follows.

$$|\bar{U}| = \{u \mid u = |\bar{U}|\}, \bar{U} \text{ denotes the central ULA segment in the difference coarray of } \bar{\mathbb{S}} = \mathbb{S} \setminus \mathbb{A}, \text{ where } \mathbb{A} \text{ is a subset of } \mathbb{S}\}. \quad (4.8)$$

Since the set \bar{U} generated by different \mathbb{A} may be different, here we separate all \mathbb{A} sets with different symbols. They can be written as \mathbb{A}_k^j , where $k = 0, 1, \dots, |\mathbb{S}|$ is the number of elements in \mathbb{A} , and $j = 1, 2, \dots, \binom{|\mathbb{S}|}{k}$ is the index of \mathbb{A}_k . Also, we use $\bar{\mathbb{S}}_k^j$ to separate different cases that $\mathbb{S} \setminus \mathbb{A}_k^j$, and the corresponding central ULA segment in the difference coarray is denoted by \bar{U}_k^j . Note that \mathbb{A}_k^j , $\bar{\mathbb{S}}_k^j$, and \bar{U}_k^j are deterministic. Now let us look at the probability of $\mathbb{A} = \mathbb{A}_k^j$. This probability is inspired by [24, Eq.32].

If every sensor in the array \mathbb{S} fails independently with the same probability p , then

$$\begin{aligned}
 \Pr(\mathbb{A} = \mathbb{A}_k^j) &= \Pr\left(\left(\bigcap_{n_1 \in \mathbb{A}_k^j} (n_1 \text{ fails})\right) \cap \left(\bigcap_{n_2 \in \bar{\mathbb{S}}_k^j} (n_2 \text{ fails})^c\right)\right) \\
 &= \left(\prod_{n_1 \in \mathbb{A}_k^j} \Pr(n_1 \text{ fails})\right) \cdot \left(\prod_{n_2 \in \bar{\mathbb{S}}_k^j} [1 - \Pr(n_2 \text{ fails})]\right) \quad (4.9) \\
 &= p^k (1 - p)^{|\mathbb{S}| - k}.
 \end{aligned}$$

Eq.(4.9) is also the probability of $\bar{\mathbb{S}} = \bar{\mathbb{S}}_k^j$, so

$$\Pr(\mathbb{A} = \mathbb{A}_k^j) = \Pr(\bar{\mathbb{S}} = \bar{\mathbb{S}}_k^j) = p^k (1 - p)^{|\mathbb{S}| - k}. \quad (4.10)$$

For every array $\bar{\mathbb{S}}_k^j$, the corresponding central ULA segment in the difference coarray, $\bar{\mathbb{U}}_k^j$, can be obtained based on (2.12). Therefore, the probability that every $\bar{\mathbb{U}}_k^j$ happens is also equal to $p^k (1 - p)^{|\mathbb{S}| - k}$. We calculate the size of all $\bar{\mathbb{U}}_k^j$. If every $|\bar{\mathbb{U}}_k^j|$ are multiplied their corresponding probability, and we sum up all of them, then we can obtain the expected value of $|\bar{\mathbb{U}}|$. The expression is shown as follows.

$$\mathbb{E} [|\bar{\mathbb{U}}|] = \sum_{k=0}^{|\mathbb{S}|} \sum_{j=1}^{\binom{|\mathbb{S}|}{k}} |\bar{\mathbb{U}}_k^j| p^k (1 - p)^{|\mathbb{S}| - k}. \quad (4.11)$$

In (4.11), $\sum_{j=1}^{\binom{|\mathbb{S}|}{k}} |\bar{\mathbb{U}}_k^j|$ purely depends on the array configuration. Thus, it is possible to increase $\mathbb{E} [|\bar{\mathbb{U}}|]$ by designing the array geometry. Furthermore, $p^k (1 - p)^{|\mathbb{S}| - k}$ depends on the probability p , the summation index k , and the size of the array $|\mathbb{S}|$. For a fixed array configuration, we can try to sustain the largest $\mathbb{E} [|\bar{\mathbb{U}}|]$ by controlling p or $|\mathbb{S}|$. In Section 2.5, we mentioned that p is affected by the sensing device. On the other hand, the size of $|\mathbb{S}|$ can be changed by adding other sensors to the array.



Next, we will present a closed-form relationship between $\mathbb{E} [|\bar{\mathcal{U}}|]$ and generalized k -fragility $\mathcal{F}_k(\mathbb{S}, \mathcal{S}_{\mathcal{U}})$ which was introduced in Section 2.5. First, we rewrite (4.11) by multiplying $|\mathcal{U}| / |\bar{\mathcal{U}}|$, and we can get

$$\mathbb{E} [|\bar{\mathcal{U}}|] = \sum_{k=0}^{|\mathbb{S}|} p^k (1-p)^{|\mathbb{S}|-k} \sum_{j=1}^{\binom{|\mathbb{S}|}{k}} |\bar{\mathcal{U}}_k^j| = |\mathcal{U}| \sum_{k=0}^{|\mathbb{S}|} p^k (1-p)^{|\mathbb{S}|-k} \left(\frac{1}{|\bar{\mathcal{U}}|} \sum_{j=1}^{\binom{|\mathbb{S}|}{k}} |\bar{\mathcal{U}}_k^j| \right). \quad (4.12)$$

If adding $\binom{|\mathbb{S}|}{k} - \binom{|\mathbb{S}|}{k}$ to the parentheses, (4.12) will become

$$\mathbb{E} [|\bar{\mathcal{U}}|] = |\mathcal{U}| \sum_{k=0}^{|\mathbb{S}|} p^k (1-p)^{|\mathbb{S}|-k} \left(\binom{|\mathbb{S}|}{k} - \left(\binom{|\mathbb{S}|}{k} - \frac{1}{|\bar{\mathcal{U}}|} \sum_{j=1}^{\binom{|\mathbb{S}|}{k}} |\bar{\mathcal{U}}_k^j| \right) \right). \quad (4.13)$$

Then we move $1/|\bar{\mathcal{U}}|$ and $\binom{|\mathbb{S}|}{k}$ to the summation of j , so (4.13) become

$$\mathbb{E} [|\bar{\mathcal{U}}|] = |\mathcal{U}| \sum_{k=0}^{|\mathbb{S}|} p^k (1-p)^{|\mathbb{S}|-k} \left(\binom{|\mathbb{S}|}{k} - \sum_{j=1}^{\binom{|\mathbb{S}|}{k}} 1 - \frac{|\bar{\mathcal{U}}_k^j|}{|\bar{\mathcal{U}}|} \right). \quad (4.14)$$

Then we multiply $\binom{|\mathbb{S}|}{k} / \binom{|\mathbb{S}|}{k}$ to (4.14), so (4.14) become

$$\mathbb{E} [|\bar{\mathcal{U}}|] = |\mathcal{U}| \sum_{k=0}^{|\mathbb{S}|} \binom{|\mathbb{S}|}{k} p^k (1-p)^{|\mathbb{S}|-k} \left(1 - \sum_{j=1}^{\binom{|\mathbb{S}|}{k}} \frac{1 - \frac{|\bar{\mathcal{U}}_k^j|}{|\bar{\mathcal{U}}|}}{\binom{|\mathbb{S}|}{k}} \right). \quad (4.15)$$

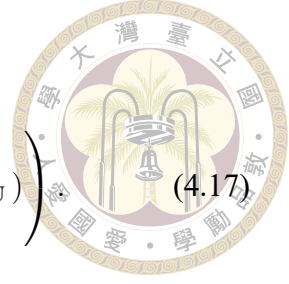
In Definition 2.5.1, we have defined $\mathcal{F}_k(\mathbb{S}, \mathcal{S}_{\mathcal{U}}) = \sum_{\mathbb{A} \subseteq \mathbb{S}, |\mathbb{A}|=k} \mathcal{S}_{\mathcal{U}}(\mathbb{A}) / \binom{|\mathbb{S}|}{k}$, and $\mathcal{S}_{\mathcal{U}}(\mathbb{A})$ is equal to $1 - |\bar{\mathcal{U}}| / |\mathcal{U}|$ based on (2.28). Thus, we can rewrite (4.15) as

$$\mathbb{E} [|\bar{\mathcal{U}}|] = |\mathcal{U}| \sum_{k=0}^{|\mathbb{S}|} \binom{|\mathbb{S}|}{k} p^k (1-p)^{|\mathbb{S}|-k} (1 - \mathcal{F}_k(\mathbb{S}, \mathcal{S}_{\mathcal{U}})). \quad (4.16)$$

Finally, we know that $\sum_{k=0}^{|\mathbb{S}|} \binom{|\mathbb{S}|}{k} p^k (1-p)^{|\mathbb{S}|-k} = 1$, because it is the CDF of the binomial distribution. Therefore, we can obtain the relationship between $\mathbb{E} [|\bar{\mathcal{U}}|]$ and $\mathcal{F}_k(\mathbb{S}, \mathcal{S}_{\mathcal{U}})$

like the following form.

$$\mathbb{E} [|\bar{\mathcal{U}}|] = |\mathcal{U}| \left(1 - \sum_{k=0}^{|\mathcal{S}|} \binom{|\mathcal{S}|}{k} p^k (1-p)^{|\mathcal{S}|-k} \mathcal{F}_k(\mathcal{S}, \mathcal{S}_{\mathcal{U}}) \right) \quad (4.17)$$

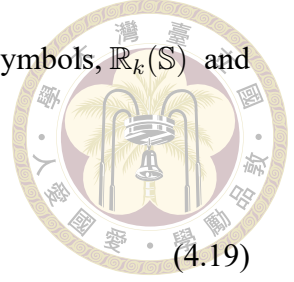


We can notice the content in the parentheses of (4.17). Its range is

$$0 \leq \left(1 - \sum_{k=0}^{|\mathcal{S}|} \binom{|\mathcal{S}|}{k} p^k (1-p)^{|\mathcal{S}|-k} \mathcal{F}_k(\mathcal{S}, \mathcal{S}_{\mathcal{U}}) \right) \leq 1. \quad (4.18)$$

As we mentioned in Section 2.5, the value of $\mathcal{F}_k(\mathcal{S}, \mathcal{S})$ is between 0 and 1. Here $\mathcal{F}_k(\mathcal{S}, \mathcal{S})$ of each k will be multiplied a probability of the binomial distribution, so the summation of k is range from 0 to 1. That is the reason we can obtain the inequality of (4.18). An array is more robust if its $\mathcal{F}_k(\mathcal{S}, \mathcal{S}_{\mathcal{U}})$ is close to 0. Hence, for a fixed p , the computed value in the parentheses of (4.18) is close to 1 if the array is more robust. Also, the value of $\mathbb{E} [|\bar{\mathcal{U}}|]$ will close to $|\mathcal{U}|$ according to (4.17). It indicates that a robust array is more likely to maintain its original $|\mathcal{U}|$, even though it is under the influence of p .

However, the complexity of computing $\mathbb{E} [|\bar{\mathcal{U}}|]$ based on (4.11) is $\mathcal{O}(2^{|\mathcal{S}|})$, since we need to compute $\binom{|\mathcal{S}|}{k}$ values of $|\bar{\mathcal{U}}_k^j|$ for each k and sum up all of them. It will become computationally expensive for large $|\mathcal{S}|$. If we consider two terms $k = 0$ and $k = 1$, the complexity will become $\mathcal{O}(|\mathcal{S}|)$. Furthermore, if we only consider one term $k = 0$, then the complexity is $\mathcal{O}(1)$. It greatly reduce the complexity. But the approximation is not suitable for any arrays, and we will further explain through the numerical results in Section 4.2.3.



For the convenience of the following discussion, we define two symbols, $\mathbb{R}_k(\mathbb{S})$ and $\mathbb{P}_k(p, \mathbb{S})$, where

$$\mathbb{R}_k(\mathbb{S}) = \sum_{j=1}^{\binom{|\mathbb{S}|}{k}} |\bar{\mathbb{U}}_k^j|, \quad (4.19)$$

$$\mathbb{P}_k(p, \mathbb{S}) = \mathbb{R}_k(\mathbb{S}) p^k (1-p)^{|\mathbb{S}|-k}.$$

Therefore, $\mathbb{E} [|\bar{\mathbb{U}}|]$ of (4.11) also can be represented as

$$\mathbb{E} [|\bar{\mathbb{U}}|] = \sum_{k=0}^{|\mathbb{S}|} \mathbb{R}_k(\mathbb{S}) p^k (1-p)^{|\mathbb{S}|-k} = \sum_{k=0}^{|\mathbb{S}|} \mathbb{P}_k(p, \mathbb{S}). \quad (4.20)$$

Now we can define the approximations of $\mathbb{E} [|\bar{\mathbb{U}}|]$ depending on the summation of k in (4.20). They are expressed as follows.

$$\begin{aligned} \mathbb{E} [|\bar{\mathbb{U}}|]_{approx.0} &= \sum_{k=0}^0 \mathbb{P}_k(p, \mathbb{S}), \\ \mathbb{E} [|\bar{\mathbb{U}}|]_{approx.1} &= \sum_{k=0}^1 \mathbb{P}_k(p, \mathbb{S}), \\ \mathbb{E} [|\bar{\mathbb{U}}|]_{approx.2} &= \sum_{k=0}^2 \mathbb{P}_k(p, \mathbb{S}), \\ &\vdots \\ \mathbb{E} [|\bar{\mathbb{U}}|]_{approx.(|\mathbb{S}|-1)} &= \sum_{k=0}^{|\mathbb{S}|-1} \mathbb{P}_k(p, \mathbb{S}), \\ \mathbb{E} [|\bar{\mathbb{U}}|]_{approx.|\mathbb{S}|} &= \sum_{k=0}^{|\mathbb{S}|} \mathbb{P}_k(p, \mathbb{S}) = \mathbb{E} [|\bar{\mathbb{U}}|]. \end{aligned} \quad (4.21)$$

From the top to the bottom of (4.21), the value of these approximations continuously accumulate. We know that

$$\mathbb{P}_0(p, \mathbb{S}) \leq \mathbb{P}_0(p, \mathbb{S}) + \mathbb{P}_1(p, \mathbb{S}) \leq \mathbb{P}_0(p, \mathbb{S}) + \mathbb{P}_1(p, \mathbb{S}) + \mathbb{P}_2(p, \mathbb{S}) \leq \dots \quad (4.22)$$

Hence, the approximations in (4.21) have the following relationship.

$$\mathbb{E} [|\bar{\mathbf{U}}|]_{approx.0} \leq \mathbb{E} [|\bar{\mathbf{U}}|]_{approx.1} \leq \dots \leq \mathbb{E} [|\bar{\mathbf{U}}|]_{approx.(|\mathbb{S}|-1)} \leq \mathbb{E} [|\mathbf{U}|]. \quad (4.23)$$



4.2.2 Proposed Robustness Metric

Here we define a robustness metric based on $\mathbb{E} [|\bar{\mathbf{U}}|]$. The advantage is that it is related to p and $|\bar{\mathbf{U}}|$ at the same time. The definition is shown below.

Definition 4.2.1. The robustness metric of the array \mathbb{S} is defined as $p_{90\%}$, where $p_{90\%}$ is the value of p when $\mathbb{E} [|\bar{\mathbf{U}}|]$ decreasing to $0.9 \cdot |\mathbf{U}|$.

Different arrays have different size of $|\mathbf{U}|$, and this metric $p_{90\%}$ take the $|\mathbf{U}|$ of the array itself as the reference point. It can be regarded as the usage time of the array when the value of $\mathbb{E} [|\bar{\mathbf{U}}|]$ is between $|\mathbf{U}|$ and $0.9 \cdot |\mathbf{U}|$. An array is more robust if the value of $p_{90\%}$ close to 1, and less robust if the value of $p_{90\%}$ close to 0. We can solve the value of $p_{90\%}$ through the following equation with the variable p .

$$\sum_{k=0}^{|\mathbb{S}|} p^k (1-p)^{|\mathbb{S}|-k} \sum_{j=1}^{\binom{|\mathbb{S}|}{k}} |\bar{\mathbf{U}}_k^j| = 0.9 \cdot |\mathbf{U}|. \quad (4.24)$$

This is a unary polynomial equation with up to $|\mathbb{S}|$ -th power. It can be written as

$$c_0 + c_1 p + c_2 p^2 + \dots + c_{|\mathbb{S}|-1} p^{|\mathbb{S}|-1} + c_{|\mathbb{S}|} p^{|\mathbb{S}|} = 0.9 \cdot |\mathbf{U}|. \quad (4.25)$$

Algorithm 1 shows how to get all the coefficients c_i in (4.25). We just follow up the steps of Algorithm 1, and then we can get a vector $[c_0, c_1, c_2, \dots, c_{|\mathbb{S}|-1}, c_{|\mathbb{S}|}]$. In fact, c_0

is equal to $|\mathbb{U}|$, so the equation we actually need to solve is

$$0.1 \cdot |\mathbb{U}| + c_1 p + c_2 p^2 + \cdots + c_{|\mathbb{S}|-1} p^{|\mathbb{S}|-1} + c_{|\mathbb{S}|} p^{|\mathbb{S}|} = 0. \quad (4.26)$$



The answer of (4.26) can be easily obtained through a MATLAB function: *roots*.

Since $p_{90\%}$ is the intersection of p of $\mathbb{E} [|\bar{\mathbb{U}}|]$ and $0.9 \cdot |\mathbb{U}|$, there is only a real solution of p between 0 and 1. The solution of p is $p_{90\%}$.

Algorithm 1 Generate the coefficients of all k -th power of the variable p of $\mathbb{E} [|\bar{\mathbb{U}}|]$.

- 1: build a vector $\mathbf{h} = [1, -1]$
 - 2: build a vector $\mathbf{a} = \mathbf{h} * \mathbf{h}$
 - 3: build a column vector \mathbf{b} , where $\mathbf{b} \in \mathbb{N}^{|\mathbb{S}|}$
 - 4: build a matrix \mathbf{B} , where $\mathbf{B} \in \mathbb{Z}^{|\mathbb{S}| \times (|\mathbb{S}|+1)}$
 - 5: assign $\mathbf{B}_{|\mathbb{S}|, |\mathbb{S}| : |\mathbb{S}|+1} = \mathbf{h}$
 - 6: assign $\mathbf{B}_{|\mathbb{S}|-1, |\mathbb{S}|-1 : |\mathbb{S}|+1} = \mathbf{a}$
 - 7: **for** $k = 0, 1, \dots, |\mathbb{S}| - 1$ **do**
 - 8: $\mathbf{b}_{k+1} = \mathbb{R}_k(\mathbb{S})$
 - 9: **if** k is equal to 0 or 1 **then**
 - 10: **continue**
 - 11: **else**
 - 12: $\mathbf{a} = \mathbf{a} * \mathbf{h}$
 - 13: $\mathbf{B}_{|\mathbb{S}|-k, |\mathbb{S}|-k : |\mathbb{S}|+1} = \mathbf{a}$
 - 14: **end if**
 - 15: **end for**
 - 16: Sum up all the values of each column of the matrix $\mathbf{b} \bullet \mathbf{B}$
-

In Algorithm 1, we use the convolution operator between vectors. For digital signal processing, the convolution of two signals $f[n]$ and $g[n]$ is

$$y[n] = f[n] * g[n] = \sum_m f[m]g[n - m]. \quad (4.27)$$

If $f[n] = \sum_{j=0}^{M_1-1} f_j \delta[n - j]$ and $g[n] = \sum_{j=0}^{M_2-1} g_j \delta[n - j]$, then the output

$$y[n] = \sum_{j=0}^{M_1+M_2-1} y_j \delta[n - j]. \quad (4.28)$$

Assume that the vector $\mathbf{f} = [f_0, f_1, \dots, f_{M_1-1}]$ and the vector $\mathbf{g} = [g_0, g_1, \dots, g_{M_2-1}]$, the convolution between them is $\mathbf{f} * \mathbf{g}$. The answer is also a vector, where

$$\mathbf{y} = \mathbf{f} * \mathbf{g} = [y_0, y_1, \dots, y_{M_1+M_2-1}]. \quad (4.29)$$

If \mathbf{f} and \mathbf{g} are vectors of polynomial coefficients, the convolution of them is equivalent to multiplying the two polynomials [30]. That is the reason we use vector convolution in the algorithm, since we have this term $p^k(1-p)^{|\mathbb{S}|-k}$ in (4.24).

4.2.3 Numerical Results

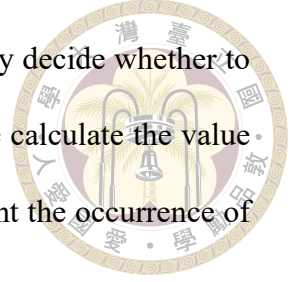
4.2.3.1 The Expected Value and The Corresponding Approximation

First, we compare the expected value obtained from the statistical PMF with the theoretical expected value we derived in (4.11). We consider the coprime array with $M = 2$ and $N = 3$. Then we generate the PMF of $|\bar{\mathbb{U}}|$ through Algorithm 2. There are 16 different p values, each p will be run Algorithm 2 once and then generate a PMF plot. The results are shown in Figure 4.3 and Figure 4.4.

Algorithm 2 PMF of $|\bar{\mathbb{U}}|$ generation.

- 1: Given an array with N sensors, $\mathbb{S} = \{n_1, n_2, \dots, n_N\}$
 - 2: Fix p and the number of trials
 - 3: **for** $iteration = 1, 2, \dots, \text{number of trials}$ **do**
 - 4: **for** n_1, n_2, \dots, n_N **do**
 - 5: $c = \text{single uniformly distributed random number in the interval } (0, 1)$
 - 6: **if** $c \leq p$ **then**
 - 7: Remove the current sensor from \mathbb{S}
 - 8: **end if**
 - 9: **end for**
 - 10: Calculate $|\bar{\mathbb{U}}|$ of the current array $\bar{\mathbb{S}}$
 - 11: **end for**
 - 12: Counting the occurrence of every $|\bar{\mathbb{U}}|$, and dividing all of them by the number of trials
 - 13: Plot the PMF
-

In Algorithm 2, we set the number of trials to 10^5 , and randomly decide whether to remove the sensor based on the probability p in each trial. Then we calculate the value of $|\bar{\mathcal{U}}|$. After getting all the values of $|\bar{\mathcal{U}}|$ in each trial, we will count the occurrence of every $|\bar{\mathcal{U}}|$ to obtain the final PMF.



In Figure 4.3 and Figure 4.4, we can observe several features. When $p = 0$, all sensors work normally, and all \mathbb{A} sets are empty sets. The PMF is like a delta function at 15 (the size of \mathcal{U}). The PMF will gradually concentrate to the left when p is increasing. Finally, when $p = 1$, the PMF is like a delta function at 0.

Now we can take the expected value of each plot in Figure 4.3 and Figure 4.4. They are shown as the blue curve in Figure 4.5. Moreover, we plot the theoretical expected value based on (4.11) with the same 16 values of p . They are shown as the red curve in Figure 4.5. It can be seen that two curves are completely overlapped. That is, the expression of the expected value $\mathbb{E}[|\bar{\mathcal{U}}|]$ can be confirmed. Note that this expression is suitable for any array configurations.

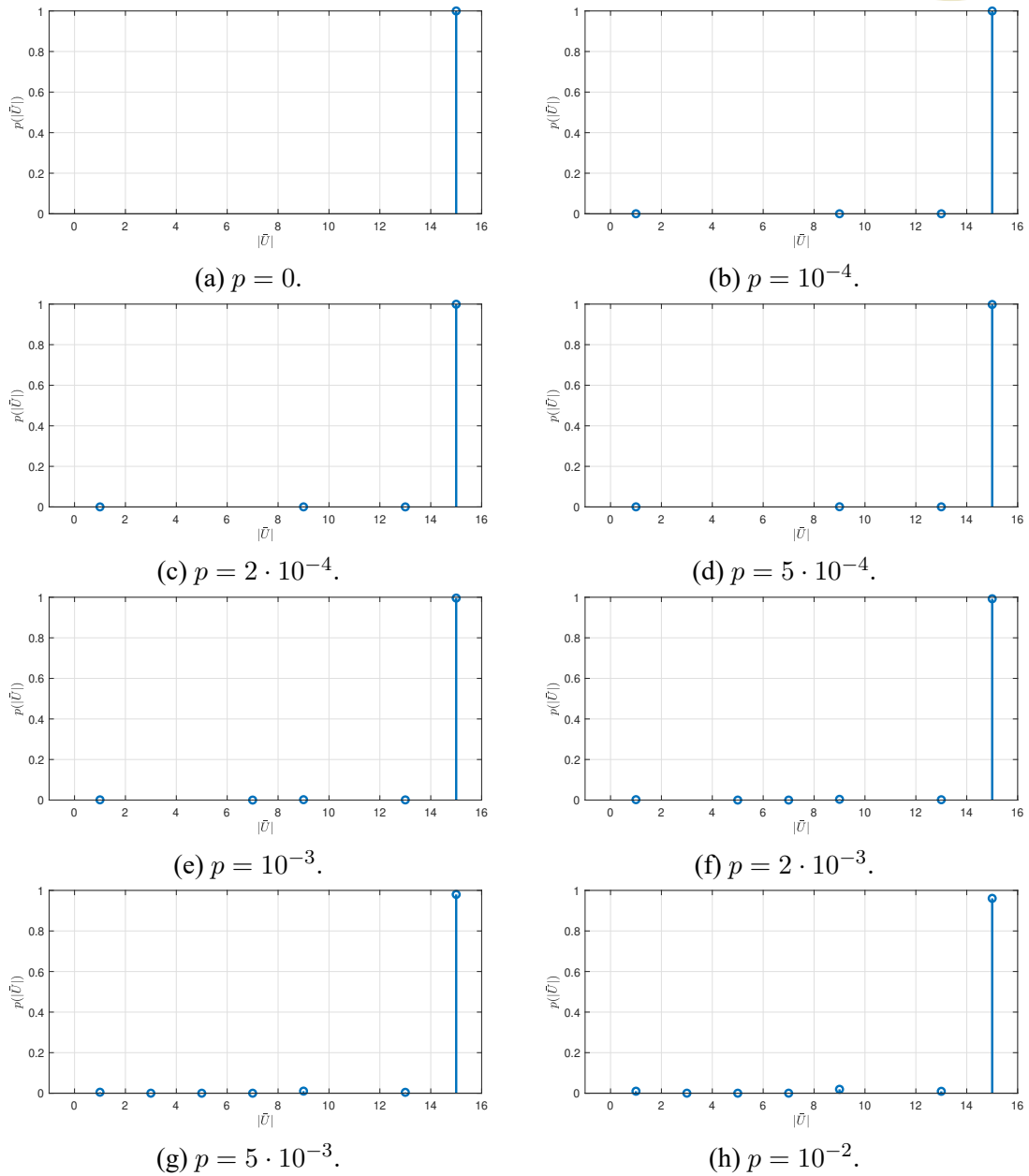
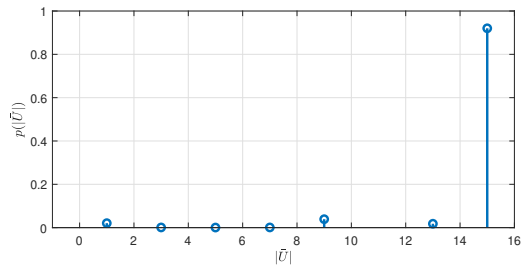
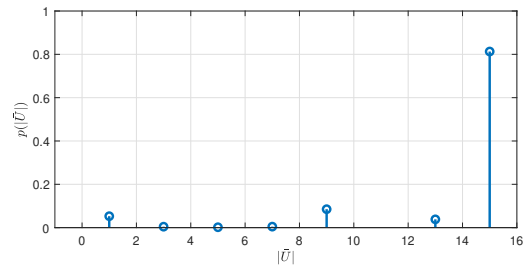


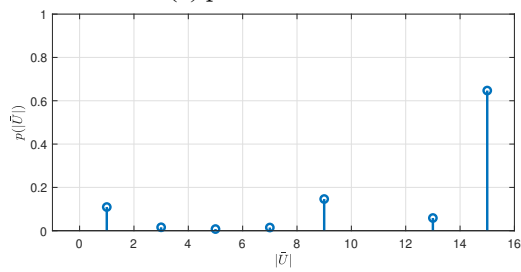
Figure 4.3: PMF of $|\bar{U}|$ with different failure probability p . Here we consider the coprime array with $M = 2$ and $N = 3$. For each p , the number of trials is 10^5 , and the PMF is obtained by counting the occurrence of every $|\bar{U}|$ through Algorithm 2.



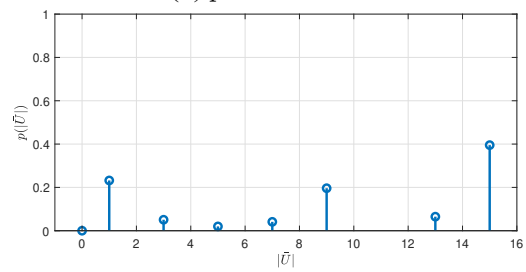
(a) $p = 2 \cdot 10^{-2}$.



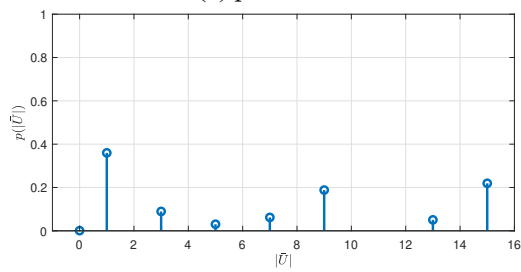
(b) $p = 5 \cdot 10^{-2}$.



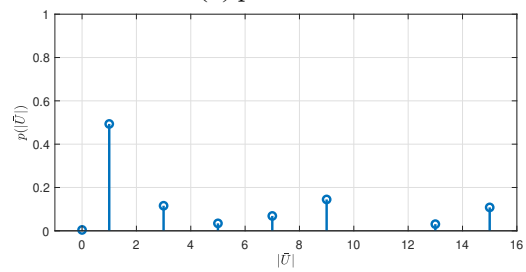
(c) $p = 0.1$.



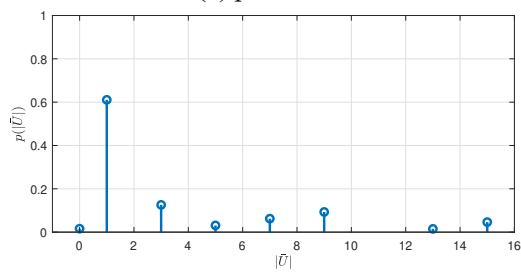
(d) $p = 0.2$.



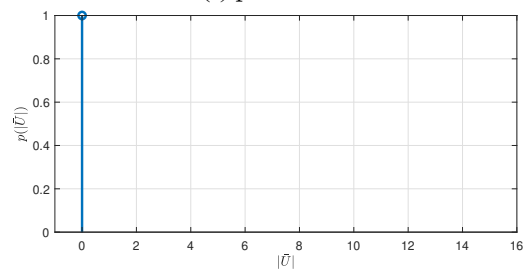
(e) $p = 0.3$.



(f) $p = 0.4$.



(g) $p = 0.5$.



(h) $p = 1$.

Figure 4.4: (Continued from Figure 4.3).

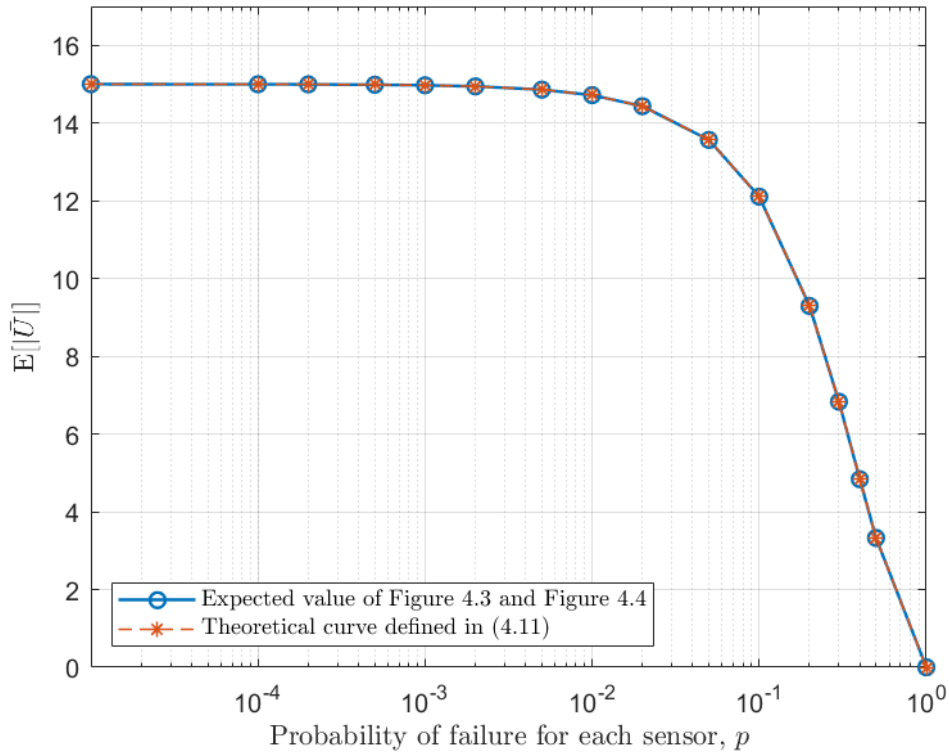
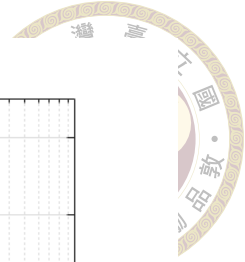
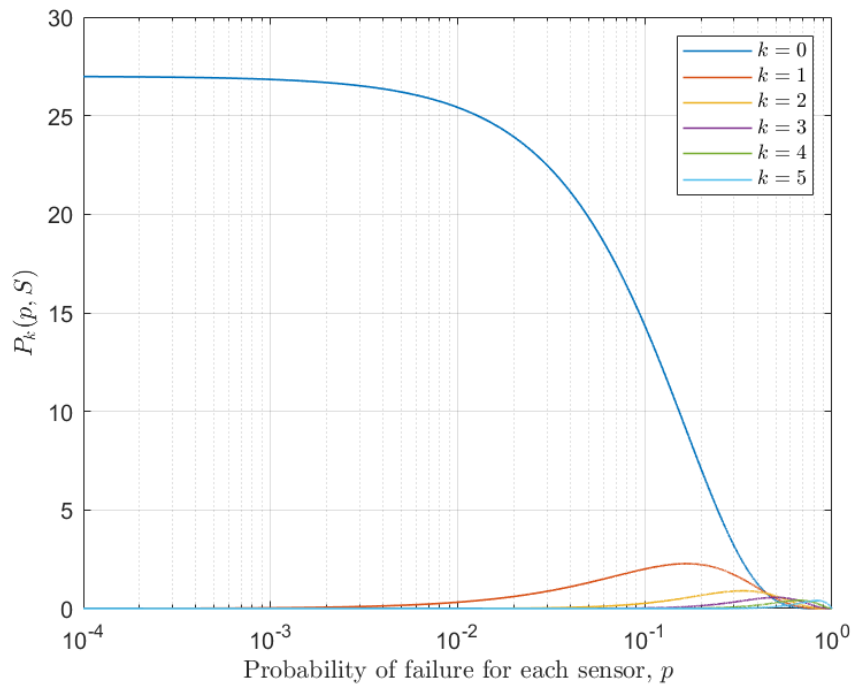
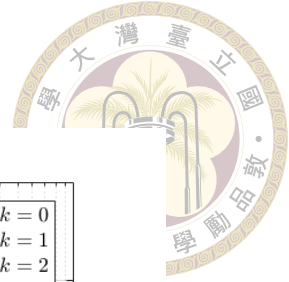


Figure 4.5: The comparison of the expected value obtained from the statistical PMF and Eq.(4.11).

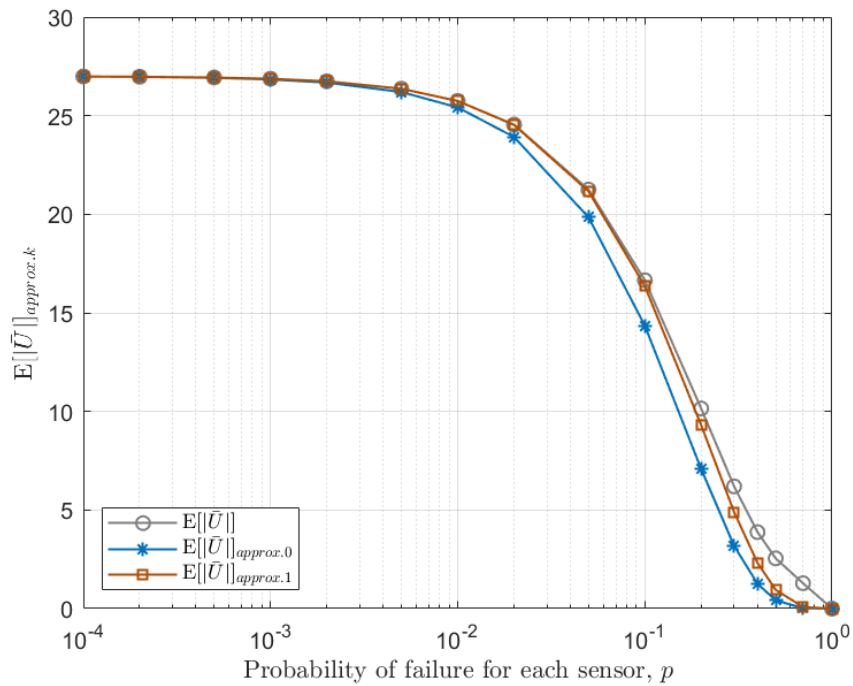
For the approximation of $\mathbb{E} [|\bar{U}|]$, we have two examples. One is the MRA and the other one is the ULA. Both of them has 6 sensors, and the array geometry is the same as Table 4.1. Figure 4.6 shows the approximation of the MRA. In this case, $k = 0$ term accounts for a large proportion of $\mathbb{E} [|\bar{U}|]$ (Figure 4.6(a)). This $k = 0$ term can be denoted by

$$\mathbb{E} [|\bar{U}|]_{approx.0} = |\mathbb{U}| \cdot (1 - p)^{|\mathbb{S}|}. \quad (4.30)$$

In Figure 4.6(b), the blue line (approximation of $k = 0$) is close to the gray line (original expected value) in the interval $p = 10^{-4} \sim 5 \cdot 10^{-3}$. If we consider two terms $k = 0$ and $k = 1$, the closeness is more obvious. The brown line is close to the gray line in the interval $p = 10^{-4} \sim 0.05$.

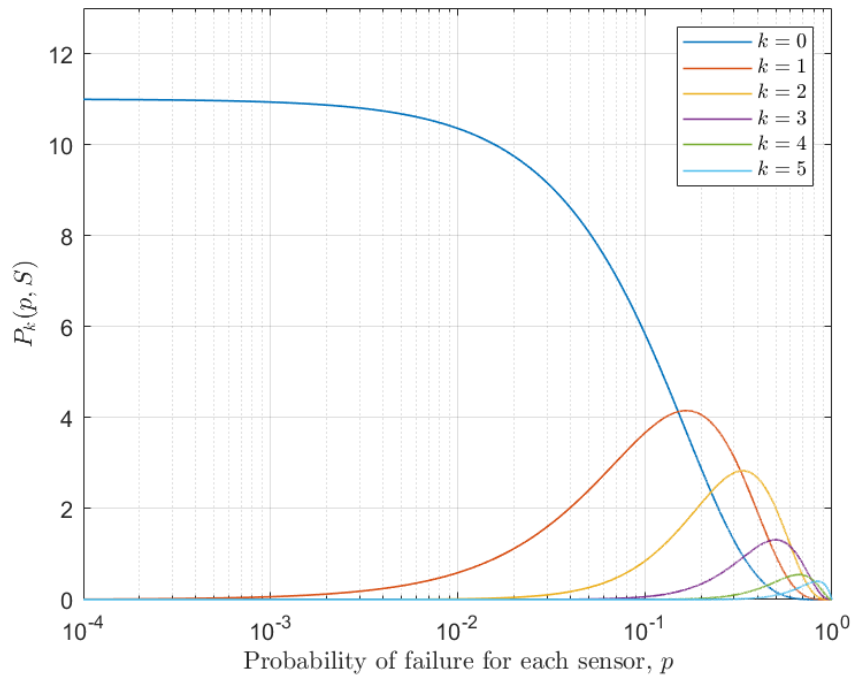
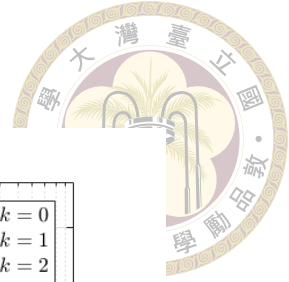


(a)

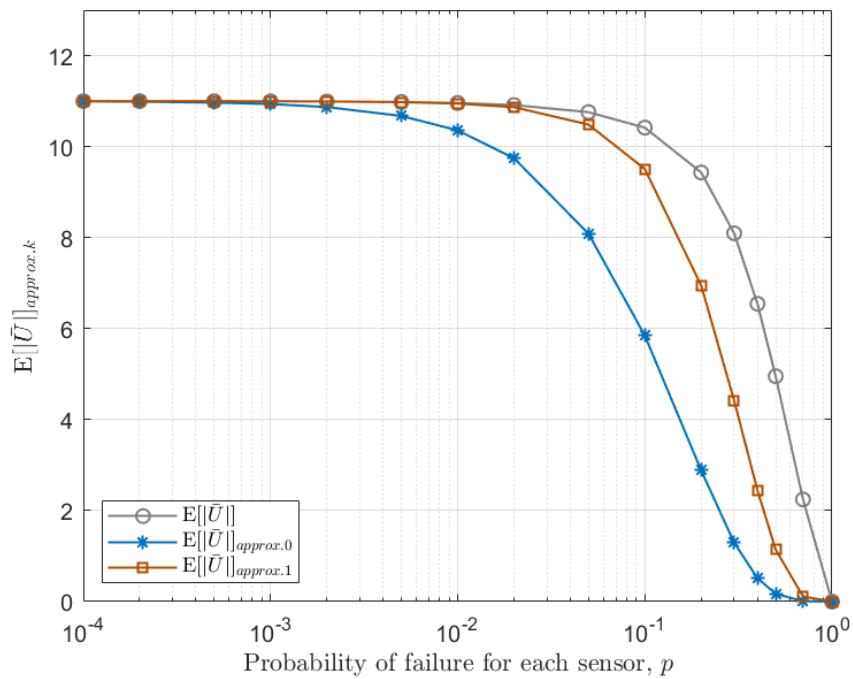


(b)

Figure 4.6: Here we consider the MRA with 6 sensors in Table 4.1. (a) Each term of k in $\mathbb{E} [|\bar{U}|]$. (b) The approximations of $\mathbb{E} [|\bar{U}|]$, $\mathbb{E} [|\bar{U}|]_{approx.0}$ and $\mathbb{E} [|\bar{U}|]_{approx.1}$ defined in (4.21).



(a)



(b)

Figure 4.7: Here we consider the ULA with 6 sensors in Table 4.1. (a) Each term of k in $\mathbb{E} [|\bar{U}|]$. (b) The approximations of $\mathbb{E} [|\bar{U}|]$, $\mathbb{E} [|\bar{U}|]_{approx.0}$ and $\mathbb{E} [|\bar{U}|]_{approx.1}$ defined in (4.21).

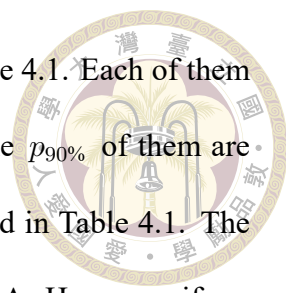
However, the approximations of the ULA are more inaccurate. In Figure 4.7(b), $\mathbb{E} [|\bar{\mathbb{U}}|]_{approx.0}$ separates $\mathbb{E} [|\bar{\mathbb{U}}|]$ from $p = 10^{-3}$, and $\mathbb{E} [|\bar{\mathbb{U}}|]_{approx.1}$ separates $\mathbb{E} [|\bar{\mathbb{U}}|]$ from $p = 10^{-2}$. For the ULA, except for $k = 0$ and $k = 1$, the other $\mathbb{P}_k(p, \mathbb{S})$ also occupy a certain proportion of $\mathbb{E} [|\bar{\mathbb{U}}|]$ (Figure 4.7(a)). If we ignore them, then it will not be able to approximate $\mathbb{E} [|\bar{\mathbb{U}}|]$ well.

Due to these two examples, we know that the approximations are more suitable for the arrays that are not robust, such as MRA and nested arrays. From another point, according to the discussion of (4.17), small value of $\mathcal{F}_k(\mathbb{S}, \mathcal{I}_{\mathbb{U}})$ makes $|\mathbb{U}|$ decrease less. Therefore, the robust arrays like ULA can sustain the original $|\mathbb{U}|$ longer while p is increasing. However, from (4.30), the value of $|\mathbb{U}| \cdot (1 - p)^{|\mathbb{S}|}$ would start dropping rapidly around $p = 10^{-2}$. Thus, $\mathbb{E} [|\bar{\mathbb{U}}|]_{approx.0}$ can fit $\mathbb{E} [|\bar{\mathbb{U}}|]$ better for the MRA. On the other hand, we need to consider more terms of k for the robust array.

4.2.3.2 Comparison of Different Arrays

In order to compare different arrays comprehensively, here we will consider the respective $|\mathbb{U}|$ of the array. Namely, we can evaluate the array by a two-dimensional point, where x -axis is $p_{90\%}$ and y -axis is $0.9 \cdot |\mathbb{U}|$. Moreover, for the convenience of the comparison, we will take the logarithm of $p_{90\%}$ that

$$\mathcal{P} = \log_{10} p_{90\%}. \quad (4.31)$$



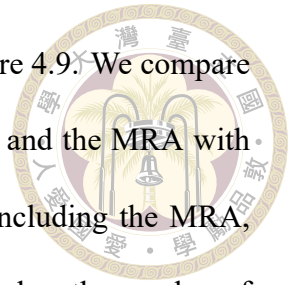
The first comparison is associated with four arrays listed in Table 4.1. Each of them has 6 sensors. The comparison results are shown in Figure 4.8. The $p_{90\%}$ of them are consistent with the results measured by $\mathcal{F}_1(\mathbb{S}, \mathcal{A}_{\mathbb{U}})$ which was listed in Table 4.1. The most robust array is the ULA and the least robust array is the MRA. However, if we compare their y -axis value, $0.9 \cdot |\mathbb{U}|$, the MRA has the highest value. In Figure 4.8, we can observe that there exists a trade-off between the robustness metric $p_{90\%}$ and the performance metric $0.9 \cdot |\mathbb{U}|$. This trade-off also can be observed in Figure 4.2 simulation. For the arrays with the same number of sensors, large size of $|\mathbb{U}|$ indicates that the weight function $w(m)$ on most of the coarray index m are small. According to Property 4 of Corollary 2.2.1, the summation of $w(m)$ for $m \in \mathbb{D}, m \neq 0$, is fixed. It depends on the size of the array. Hence, the elements in \mathbb{U} of the robust array are not easily disappeared based on a fixed p .

The second comparison is associated with four arrays, and each of them has 10 sensors. Their array geometries are shown below.

$$\begin{aligned}
 \mathbb{S}_{\text{MRA}} &= \{0, 1, 3, 6, 13, 20, 27, 31, 35, 36\}, \\
 \mathbb{S}_{\text{nested}} &= \{1, 2, 3, 4, 5, 6, 12, 18, 24, 30\}, \\
 \mathbb{S}_{\text{coprime}} &= \{0, 3, 5, 6, 9, 10, 12, 15, 20, 25\}, \\
 \mathbb{S}_{\text{ULA}} &= \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}.
 \end{aligned} \tag{4.32}$$

Figure 4.9 is the comparison result. With the same reason, if an array is designed for having large size of \mathbb{U} , then the elements in \mathbb{U} are easily disappeared since the corresponding weight function is relatively small. Therefore, in Figure 4.9, the inverse relationship between $p_{90\%}$ and $0.9 \cdot |\mathbb{U}|$ also can be observed. For different arrays with the same number of sensors, it is not possible that an array has the best $p_{90\%}$ and $0.9 \cdot |\mathbb{U}|$ at the same time.

Now we observe these 8 different arrays in Figure 4.8 and Figure 4.9. We compare the same type of the arrays. For example, the MRA with 6 sensors and the MRA with 10 sensors are compared with each other. Then we can find that including the MRA, the nested arrays and the coprime arrays, their $p_{90\%}$ will get smaller when the number of sensors increases. On the other hand, $p_{90\%}$ of the ULA with 10 sensors is larger than $p_{90\%}$ of the ULA with 6 sensors. That is, ULA are more robust with larger number of sensors. Therefore, in Section 4.3, we will discuss the robustness and the performance of ULA with different number of sensors.



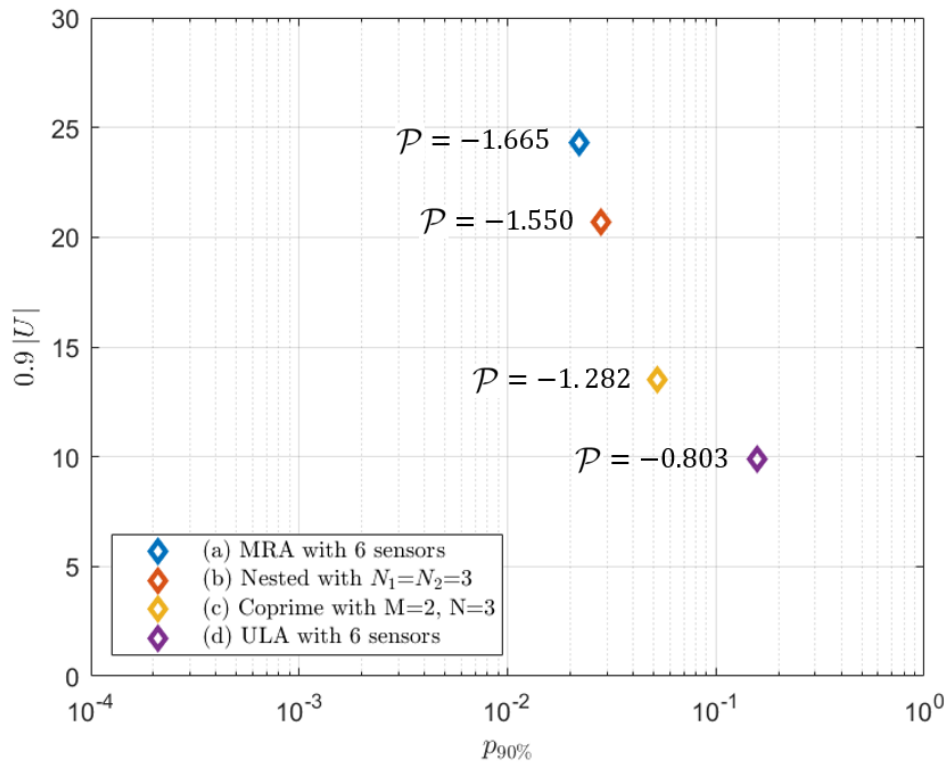
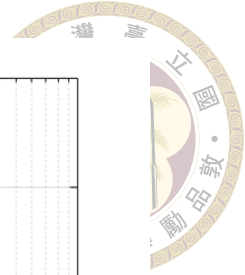


Figure 4.8: The comparison of $p_{90\%}$ and $0.9 \cdot |\mathbb{U}|$ of four arrays. Each array has 6 sensors and the array geometries are listed in Table 4.1.

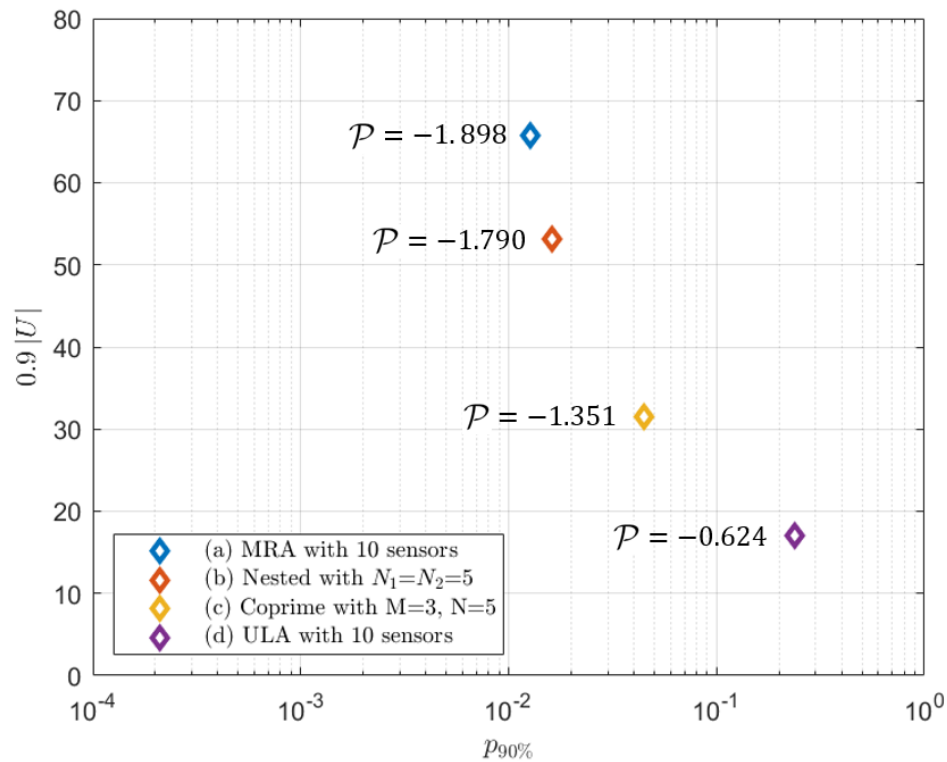
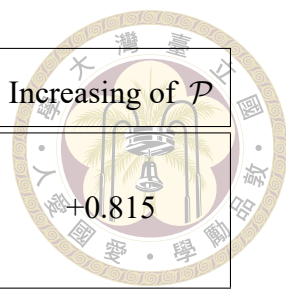


Figure 4.9: The comparison of $p_{90\%}$ and $0.9 \cdot |\mathbb{U}|$ of four arrays. Each array has 10 sensors and (4.32) shows the array geometries.



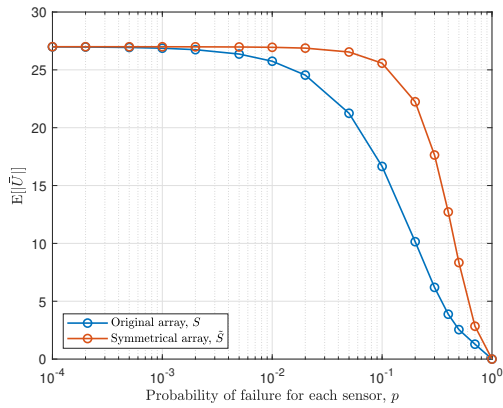
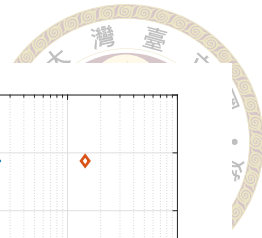
	Arrays	Number of sensors	$p_{90\%}$	\mathcal{P}	Increasing of \mathcal{P}
MRA	\mathbb{S}	6	0.0221	-1.655	+0.815
	$\tilde{\mathbb{S}}$	10	0.1446	-0.840	
Nested Array	\mathbb{S}	6	0.0282	-1.550	+0.826
	$\tilde{\mathbb{S}}$	10	0.1887	-0.724	
Coprime Array	\mathbb{S}	6	0.0523	-1.282	+0.617
	$\tilde{\mathbb{S}}$	8	0.2163	-0.665	

Table 4.2: Symmetrical arrays compare with original arrays (Here \mathbb{S} are the arrays in Table 4.1 and $\tilde{\mathbb{S}}$ are generated from (3.4) and (3.5)).

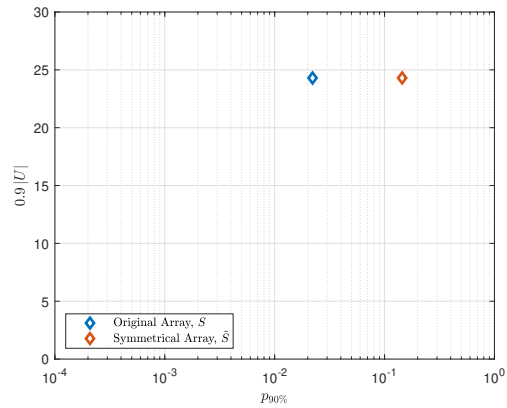
4.2.3.3 The Symmetrical Array Compare with The Original Array

In Chapter 3, we introduced how to transfer an array \mathbb{S} to its symmetrical version $\tilde{\mathbb{S}}$. Here we will compare \mathbb{S} and $\tilde{\mathbb{S}}$. In Table 4.2, we have three examples that all of the \mathbb{S} are the same as Table 4.1. The last column of Table 4.2 is the increasing value of \mathcal{P} after symmetrizing the array. We define $\mathcal{P}_{\mathbb{S}}$ as the \mathcal{P} of the array \mathbb{S} . Thus, the increasing of \mathcal{P} are obtained from $\mathcal{P}_{\tilde{\mathbb{S}}} - \mathcal{P}_{\mathbb{S}}$. It can be observed that the nested array increase the most among these three arrays. However, if we consider the number of sensors, the coprime array is the most economical. Since only two sensors are added to the symmetrical coprime array, adding one sensor can increase 0.3085 of \mathcal{P} on average. On the other hand, there are four new sensors in $\tilde{\mathbb{S}}$ for both the MRA and the nested array. Therefore, adding one sensor can only increase 0.20375 of \mathcal{P} for the MRA and 0.2065 of \mathcal{P} for the nested array.

We use $\mathbb{E} [|\bar{\mathbb{U}}|]_{\mathbb{S}}$ to represent the expected value defined in (4.11) of the array \mathbb{S} . In Figure 4.10 - Figure 4.12, we plot the $\mathbb{E} [|\bar{\mathbb{U}}|]_{\mathbb{S}}$ and the $\mathbb{E} [|\bar{\mathbb{U}}|]_{\tilde{\mathbb{S}}}$ based on the same p .

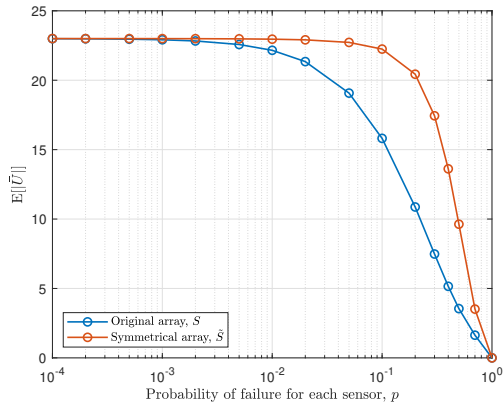


(a) p versus $\mathbb{E} [|\tilde{\mathcal{U}}|]$.

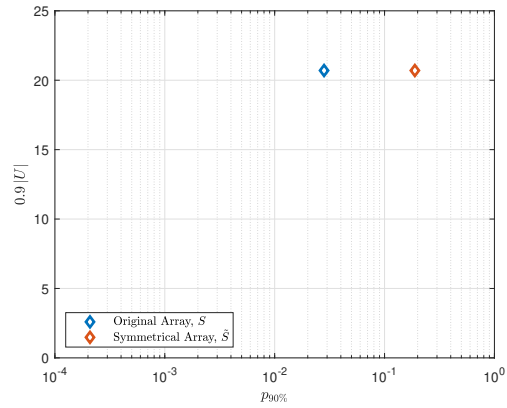


(b) $p_{90\%}$ versus $0.9 \cdot |\mathcal{U}|$.

Figure 4.10: The comparison of \mathcal{S} and $\tilde{\mathcal{S}}$ of the MRA in Table 4.2.

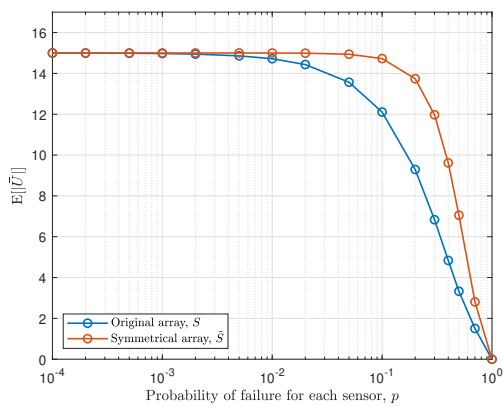


(a) p versus $\mathbb{E} [|\tilde{\mathcal{U}}|]$.

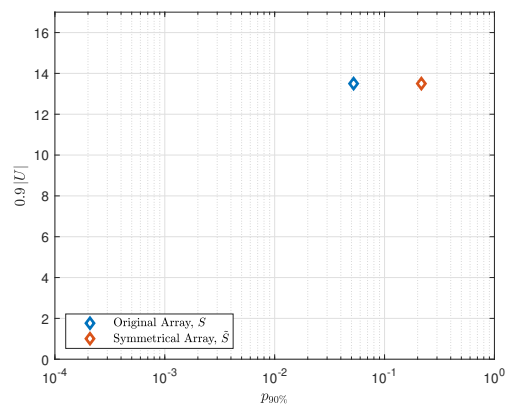


(b) $p_{90\%}$ versus $0.9 \cdot |\mathcal{U}|$.

Figure 4.11: The comparison of \mathcal{S} and $\tilde{\mathcal{S}}$ of the nested array with $N_1 = 3$ and $N_2 = 3$ in Table 4.2.



(a) p versus $\mathbb{E} [|\tilde{\mathcal{U}}|]$.



(b) $p_{90\%}$ versus $0.9 \cdot |\mathcal{U}|$.

Figure 4.12: The comparison of \mathcal{S} and $\tilde{\mathcal{S}}$ of the coprime array with $M = 2$ and $N = 3$ in Table 4.2.

Also, we mark the point of $p_{90\%}$ and $0.9 \cdot |\mathbb{U}|$. All of these three arrays have the property that $\mathbb{D} = \tilde{\mathbb{D}}$, since the MRA and the nested have hole-free difference coarrays and we have proved that $\mathbb{D}_{\text{coprime}} = \tilde{\mathbb{D}}_{\text{coprime}}$ in Proposition 3.4.1. That is why the expected value curves of \mathbb{S} and $\tilde{\mathbb{S}}$ in Figure 4.10 - Figure 4.12 have the same start point when p is quite small. It can be observed that the following inequality always holds true for these arrays.

$$\mathbb{E} [|\bar{\mathbb{U}}|]_{\mathbb{S}} \leq \mathbb{E} [|\bar{\mathbb{U}}|]_{\tilde{\mathbb{S}}}, \quad \text{for } 0 \leq p \leq 1. \quad (4.33)$$

In Proposition 3.3.3, we proved that $\mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{A}_{\mathbb{U}})$ is definitely less than $\mathcal{F}_1(\mathbb{S}, \mathcal{A}_{\mathbb{U}})$ based on $\mathbb{D} = \tilde{\mathbb{D}}$. Here the arrays in Figure 4.10 - Figure 4.12 are consistent with this proof. According to Figure 3.3, the decrease of $\mathcal{F}_1(\mathbb{S}, \mathcal{A}_{\mathbb{U}})$ is usually large empirically. Also, from (4.17), we know that small generalized 1-fragility $\mathcal{F}_1(\mathbb{S}, \mathcal{A}_{\mathbb{U}})$ can get large expected value $\mathbb{E} [|\bar{\mathbb{U}}|]$. Thus, $\tilde{\mathbb{S}}$ is more capable of maintaining the original size of \mathbb{U} than \mathbb{S} . However, we have not found a method that is suitable for proving (4.33) always holds true for arbitrary arrays. Hence, this part will be placed in future work.

Now we use \mathbb{S} and $\tilde{\mathbb{S}}$ to estimate one source, $\theta = \pi/4$, with different sensor failure probability p . SNR is 0 dB and the number of snapshots is 500. Like the experiment in Figure 4.2, random faulty sensors will be removed from the array with a fixed probability p . For every p , we have 5000 $\tilde{\mathbb{S}}$, defined as $\mathbb{S} \setminus \mathbb{A}$. Here \mathbb{A} is composed of the random faulty sensors. Then each $\tilde{\mathbb{S}}$ will be run 100 Monte-Carlo runs. Namely, each data point is averaged from $5 \cdot 10^5$ Monte-Carlo runs. There are three comparisons: the MRA, the nested array and the coprime array. The results are shown in Figure 4.13 - Figure 4.15.

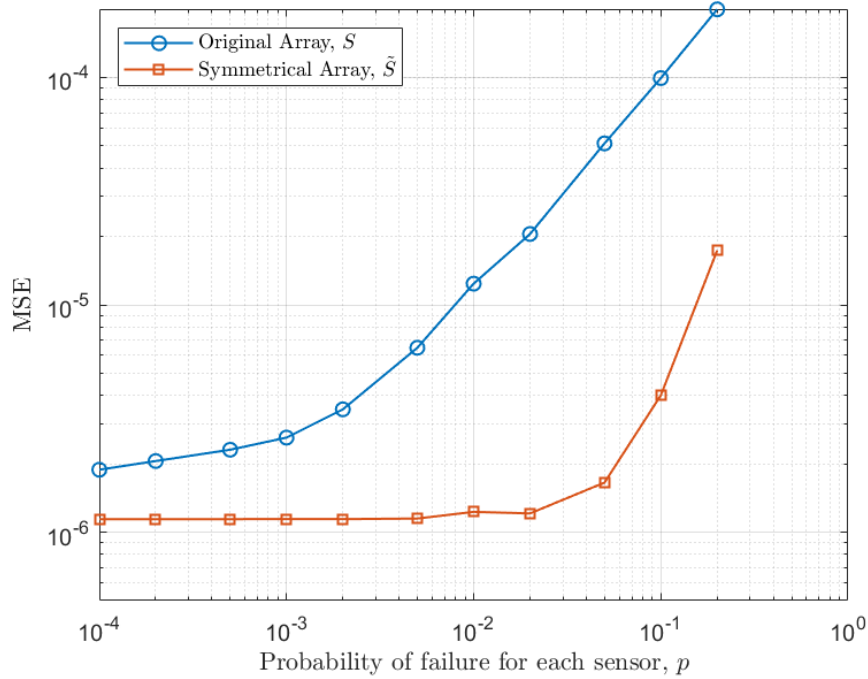
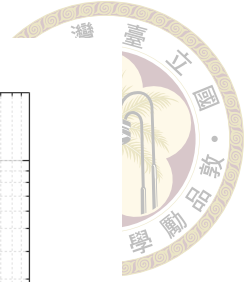


Figure 4.13: The estimation error comparison of the MRA S and its symmetrical version \tilde{S} . We estimate one source with $\theta = \pi/4$. SNR is 0 dB and the number of snapshots is 500. Each point is averaged from $5 \cdot 10^5$ Monte-Carlo runs.

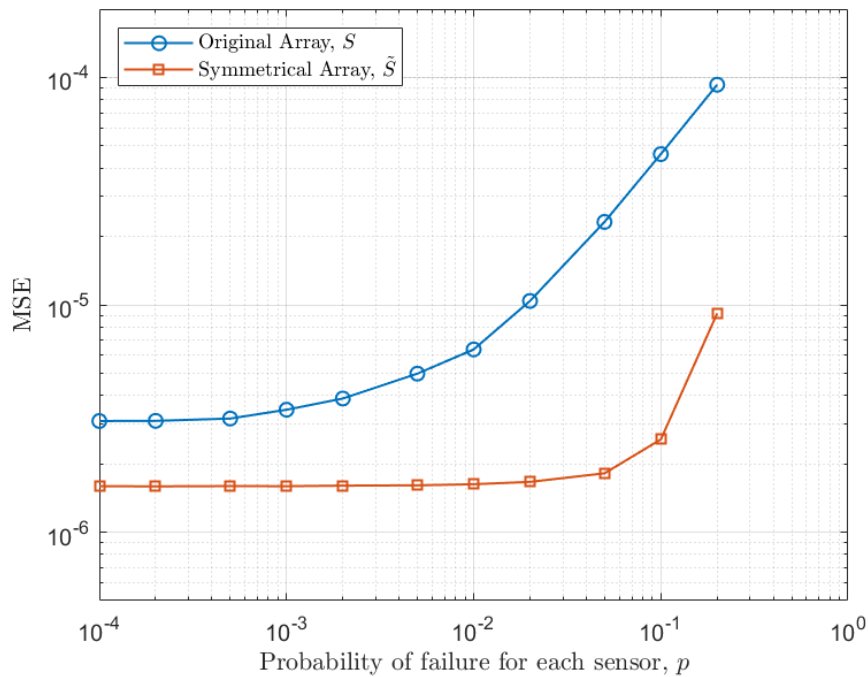


Figure 4.14: The estimation error comparison of the nested array S and its symmetrical version \tilde{S} . We estimate one source with $\theta = \pi/4$. SNR is 0 dB and the number of snapshots is 500. Each point is averaged from $5 \cdot 10^5$ Monte-Carlo runs.

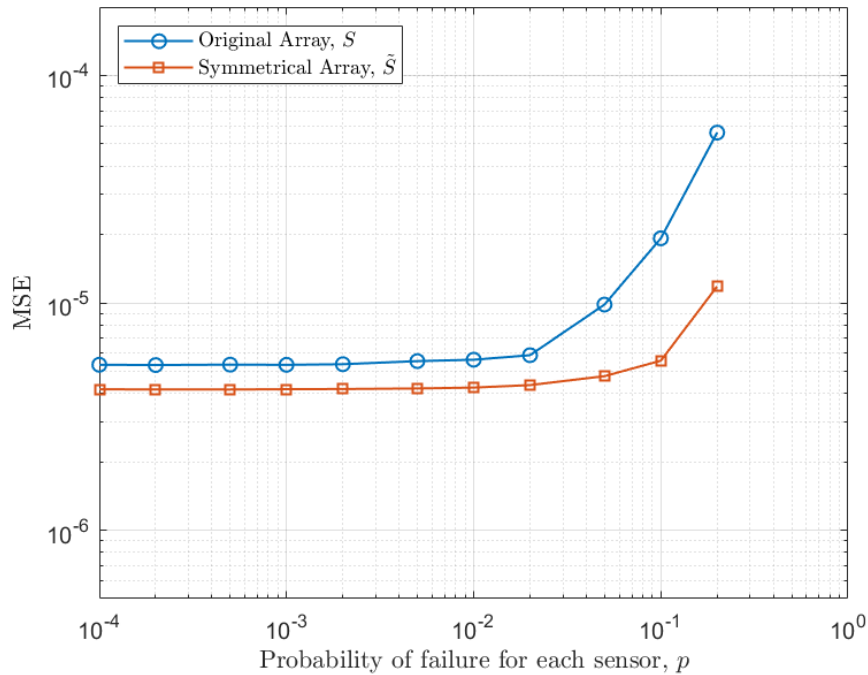
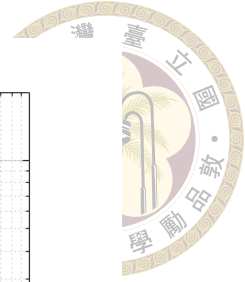


Figure 4.15: The estimation error comparison of the coprime array \mathbb{S} and its symmetrical version $\tilde{\mathbb{S}}$. We estimate one source with $\theta = \pi/4$. SNR is 0 dB and the number of snapshots is 500. Each point is averaged from $5 \cdot 10^5$ Monte-Carlo runs.

In Figure 4.13 - Figure 4.15, the MSE performance simulated by $\tilde{\mathbb{S}}$ is relatively small with all sensor failure probability p . It can be noticed that when p is quite small, $\tilde{\mathbb{S}}$ still has smaller MSE even though the size of \mathbb{U} and $\tilde{\mathbb{U}}$ are the same. We have seen this condition in Section 4.1. It is because of the different weight functions. Here for \mathbb{S} and $\tilde{\mathbb{S}}$, we know that $\mathbb{S} \subseteq \tilde{\mathbb{S}}$. Thus, $\tilde{\mathbb{S}}$ has more physical sensors that can collect the signal data in the environment. Also, with the same location of the virtual sensor, the weight function of $\tilde{\mathbb{S}}$ must not be less than the weight function of \mathbb{S} . That is, for the same virtual sensor, more data are collected in $\tilde{\mathbb{U}}$ than \mathbb{U} . From the figure(b) of Figure 4.10 - Figure 4.12, the robustness of $\tilde{\mathbb{S}}$ of these arrays are better than the robustness of \mathbb{S} based on $p_{90\%}$. This result is reflected in the simulations. It can be found that $\tilde{\mathbb{S}}$ can maintain its least MSE longer than \mathbb{S} . With all these reasons, we can say that symmetrizing the array is very beneficial.

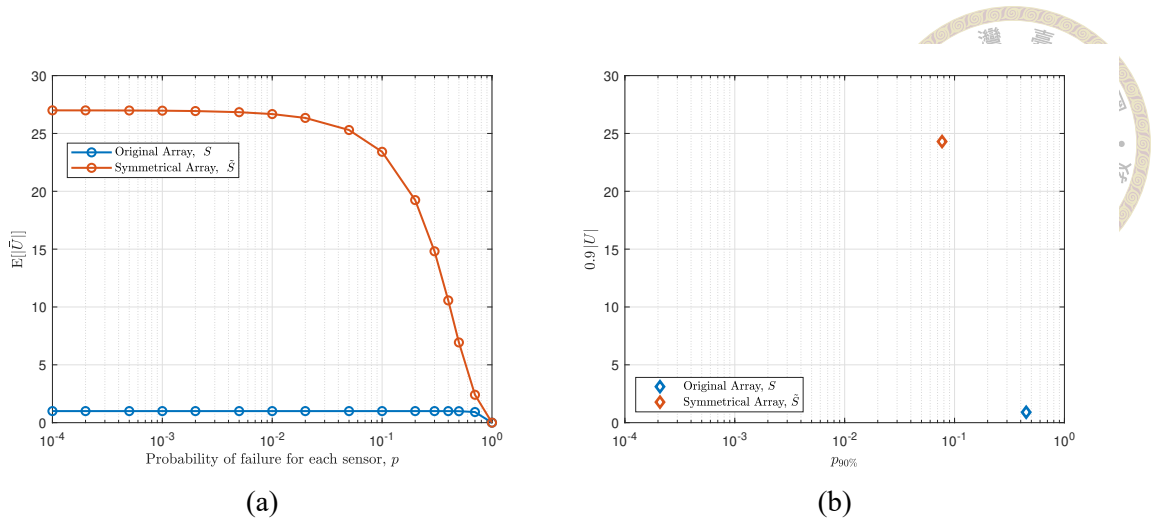


Figure 4.16: The comparison of \mathbb{S} and $\tilde{\mathbb{S}}$ of the array in (4.34). (a) The expected value curve based on (4.11). (b) Robustness metrics $p_{90\%}$ and performance metrics $0.9 \cdot |\tilde{U}|$.

The final comparison is associated with the arrays in Example 3.5.2. In that case,

$$\mathbb{S} = \{0, 3, 5, 8, 10, 13, 15\}, \quad (4.34)$$

$$\tilde{\mathbb{S}} = \{0, 2, 3, 5, 7, 8, 10, 12, 13, 15\},$$

and the robustness increase after symmetrizing the array. The reason is that \mathbb{U} only has one element $\{0\}$, but $\tilde{\mathbb{U}} = \{0, \pm 1, \dots, \pm 13\}$. The difference coarray of \mathbb{S} has the holes that $\mathbb{H} = \{\pm 1, \pm 4, \pm 6, \pm 9, \pm 11, \pm 14\}$, and most of the holes will be filled after symmetrizing the array since $CD(\mathbb{S}_o, \mathbb{S}_o^r) = \{\pm 1, \pm 4, \pm 6, \pm 9, \pm 11\}$. The cross difference was defined in Section 2.2. Here for \mathbb{S} , removing sensors does not have much impact on \mathbb{U} . Hence, in (4.34), \mathbb{S} is more robust than $\tilde{\mathbb{S}}$ based on $\mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{I}_{ess})$ and $\mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{I}_{\mathbb{U}})$. However, if we use them to estimate the source direction, $\tilde{\mathbb{S}}$ will be better than \mathbb{S} . For coarray MUSIC algorithm, \mathbb{S} does not work, but $\tilde{\mathbb{S}}$ can get a MSE performance, $1.1395 \cdot 10^{-6}$, based on (4.1). Here the SNR is 0 dB, the number of snapshots is 500, only one source located at $\theta = \pi/4$, and the MSE is averaged from 300 Monte-Carlo runs.

In Figure 4.16, we evaluate the arrays with $\mathbb{E}[|\tilde{\mathbf{U}}|]$ also mark the $p_{90\%}$ and $0.9 \cdot |\mathbf{U}|$. For $p_{90\%}$, \mathbb{S} is better than $\tilde{\mathbb{S}}$. It meets our description above because $|\mathbf{U}| = 1$. However, if we compare the y -axis, $0.9 \cdot |\mathbf{U}|$, $\tilde{\mathbb{S}}$ is much better than \mathbb{S} , since $|\tilde{\mathbf{U}}| = 27$. For the expected value curve, Eq.(4.33) also holds true. Therefore, considering the robustness and the performance at the same time allows us to evaluate the array more comprehensive, and also has the connection to the simulation.

4.2.3.4 Random Source Direction

In this section, we will use the symmetrical nested array to estimate the source angle. The array configuration we consider is $\tilde{\mathbb{S}} = \{1, 2, 3, 4, 5, 8, 9, 10, 11, 12\}$. The source angle contains $\pi/3$, $\pi/4$, $\pi/6$, and a random angle. The random angle is taken from a uniform distribution over a range from $-\pi/2.5$ to $\pi/2.5$. In this way, we can observe whether strange phenomena occur at a specific angle when we perform the DOA estimation. Here the SNR is 0 dB and the number of snapshots is 500. For the symmetrical nested array, there still exist random sensor failures. The probability we choose is $p \in \{10^{-3}, 10^{-2}, 2 \cdot 10^{-2}, 5 \cdot 10^{-2}, 0.1\}$. The set \mathbb{A} will be generated randomly based on a fixed probability, p . Then we will use the $\bar{\mathbb{S}}$, defined as $\mathbb{S} \setminus \mathbb{A}$, to estimate the source angle. Note that we only estimate one source at a time. Moreover, every $\bar{\mathbb{S}}$ will perform 100 Monte-Carlo runs and there are 1000 $\bar{\mathbb{S}}$ will be generated for each p . Thus, each point is averaged from 10^5 Monte-Carlo runs. For the random source angle, it will be regenerated for every $\bar{\mathbb{S}}$. Therefore, every $\bar{\mathbb{S}}$ will estimate one random source angle and perform 100 Monte-Carlo runs. The curves in Figure 4.17 are the results of estimating the different angles mentioned above.

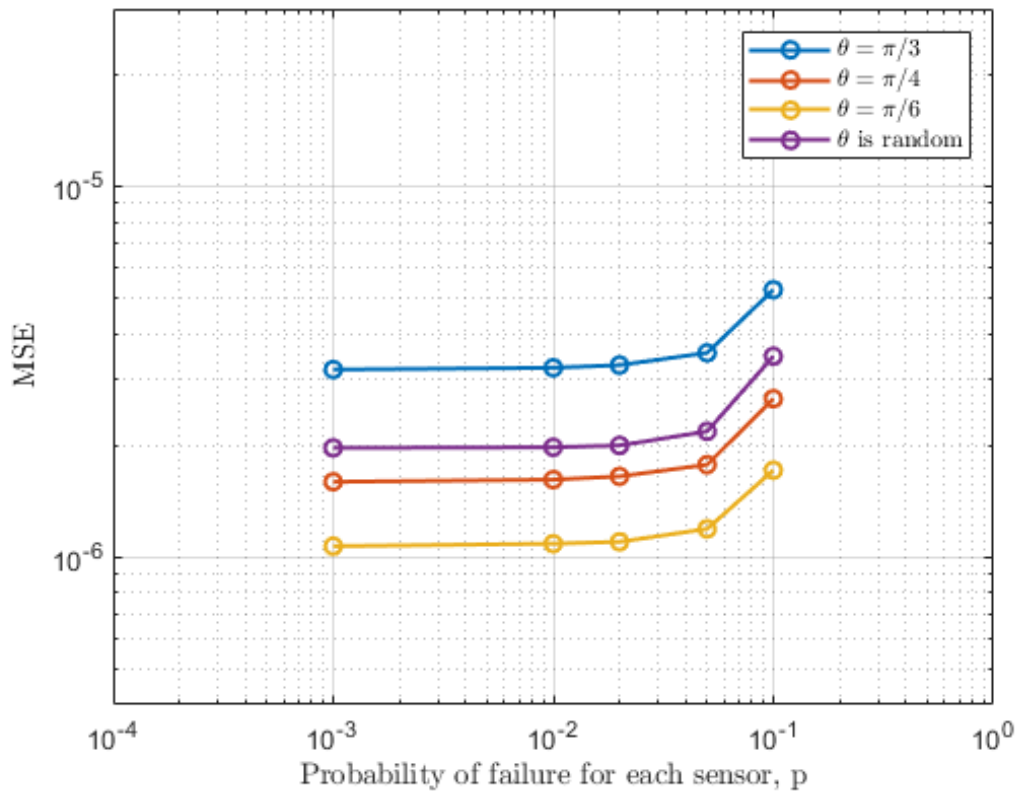
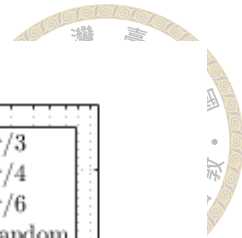


Figure 4.17: The estimation error comparison of the symmetrical nested array estimating different source angles. The fixed angle include $\pi/3$, $\pi/4$, and $\pi/6$. The random angle is $\theta \in [-\pi/2.5, \pi/2.5]$ with uniform distribution. It will be regenerated for every $\bar{\mathbb{S}}$, so every $\bar{\mathbb{S}}$ will estimate one random source angle and perform 100 Monte-Carlo runs. There are 1000 $\bar{\mathbb{S}}$ so that each point is averaged from 10^5 Monte-Carlo runs. We set SNR to 0 dB and the number of snapshots to 500.

It can be observed that the change tendency of the MSE is consistent under the different probability p . These lines do not overlap entirely because the source angle is also one of the factors affecting the MSE [7]. However, strange phenomena such as the MSE fluctuating dramatically do not occur even if we estimate the random angle.



4.3 ULA Analysis

From the experiments in Section 4.2.3.2, we found that ULA is the array that can simultaneously enhance the robustness and performance when increasing the number of sensors. This property is different from the other arrays such as nested arrays and co-prime arrays. Hence, we will specifically analyze the ULA with random faulty sensors in this section. Here we use “ULA N ” to represent the ULA with N sensors defined in (2.24). For ULA, the values of $\mathbb{R}_0(\mathbb{S})$ and $\mathbb{R}_1(\mathbb{S})$ can be easily obtained because of their geometry. If $k = 0$, it means there is no faulty sensor in the array, so

$$\mathbb{R}_0(\mathbb{S}) = \sum_{j=1}^{\binom{|\mathbb{S}|}{0}} |\bar{\mathbb{U}}_0^j| = |\mathbb{U}| = 2N - 1. \quad (4.35)$$

Also, the values of the $\mathcal{F}_1(\mathbb{S}, \mathcal{I}_{ess})$ of the ULA with N sensors is equal to $2/N$ for $N \geq 4$ [18]. Except for the sensors at the end points, “ $\min(\mathbb{S}_{ULA})$ ” and “ $\max(\mathbb{S}_{ULA})$ ”, the difference coarray will not be changed after removing one of the other sensors from the array. The value of $|\bar{\mathbb{U}}|$ still equals to $2N - 1$. On the other hand, removing $\min(\mathbb{S}_{ULA})$ or $\max(\mathbb{S}_{ULA})$ make the value of $|\bar{\mathbb{U}}|$ equal to $2N - 3$. With these results, we can get

$$\begin{aligned} \mathbb{R}_1(\mathbb{S}) &= \sum_{j=1}^{\binom{|\mathbb{S}|}{1}} |\bar{\mathbb{U}}_1^j| = (N - 2) \cdot (2N - 1) + 2 \cdot (2N - 3) \\ &= (2N^2 - N - 4N + 2) + (4N - 6) \\ &= 2N^2 - N - 4. \end{aligned} \quad (4.36)$$

Through (4.35), we know that the size of \mathbb{U} of ULA increases with the number of sensors. However, we want to know whether the size of \mathbb{U} will also increase under the influence of random sensor failures. Thus, we will use the expected value expression in

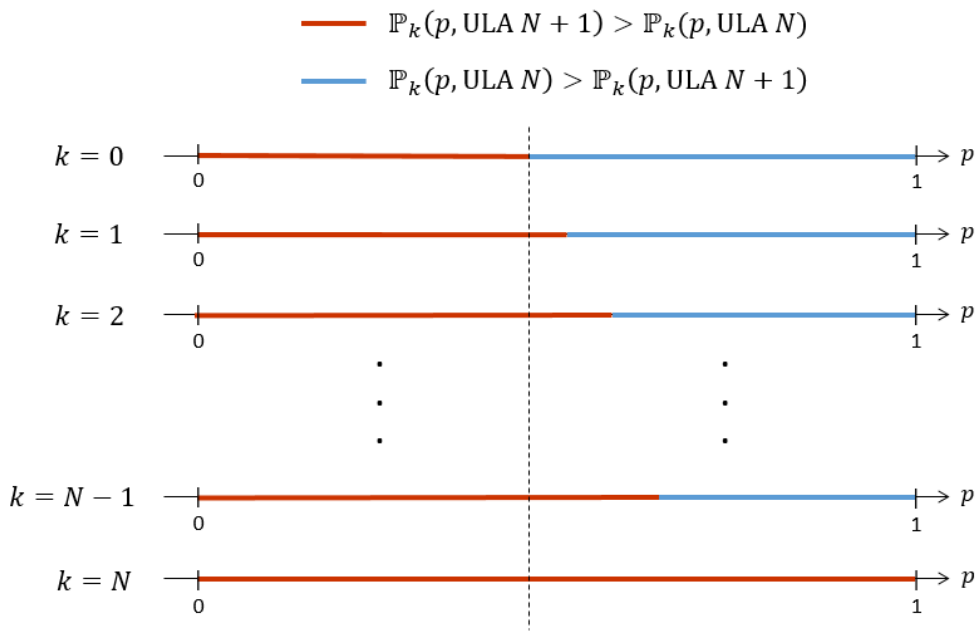
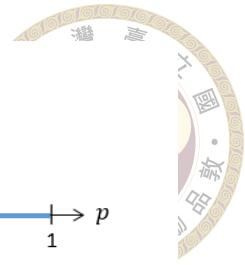
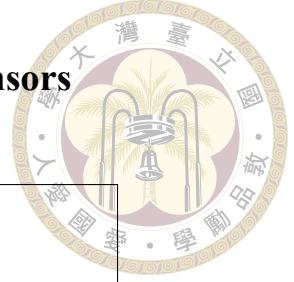


Figure 4.18: A schematic diagram we compare the ULA with different number of sensors.

(4.11) to compare the ULA with different number of sensors. In the following sections, we will discuss two comparisons of the ULA. One is the ULA with N sensors versus the ULA with $N + 1$ sensors, and the other one is the ULA with N sensors versus the ULA with $N + 2$ sensors.

According to (4.11), when computing $\mathbb{E} [|\bar{\mathbf{U}}|]$, we need to calculate the summation from $k = 0$ to $k = |\mathbb{S}|$. However, $k = |\mathbb{S}|$ indicates that all of the sensors are removed. Thus, $|\bar{\mathbf{U}}| = 0$, and $\mathbb{P}_{|\mathbb{S}|}(p, \mathbb{S}) = 0$. Namely, $k = |\mathbb{S}|$ can be ignored, and we only need to consider $k = 0, 1, \dots, |\mathbb{S}| - 1$. Figure 4.18 is a schematic diagram that we compare $\mathbb{P}_k(p, \text{ULA } N)$ and $\mathbb{P}_k(p, \text{ULA } N+1)$ of each k terms. On the p -axis, if $\mathbb{P}_k(p, \text{ULA } N+1) > \mathbb{P}_k(p, \text{ULA } N)$, it will show the red line, otherwise it will show the blue line. Assume that the intersection of p of $\mathbb{P}_k(p, \text{ULA } N+1)$ and $\mathbb{P}_k(p, \text{ULA } N)$ is p_k . Also, p_k satisfy $p_0 \leq p_1 \leq p_2 \leq \dots$. Therefore, we can find the interval of p that it only contains the red lines. Then we can obtain the inequality $\mathbb{E} [|\bar{\mathbf{U}}|]_{\text{ULA } N} < \mathbb{E} [|\bar{\mathbf{U}}|]_{\text{ULA } N+1}$ in that interval of p .



4.3.1 ULA with N sensors and ULA with $N + 1$ sensors

$\mathbb{P}_k(p, \mathbb{S}) \setminus \mathbb{S}$	ULA N	ULA $N+1$
k		
0	$\mathbb{R}_0(\mathbb{S}) \cdot (1 - p)^N$	$\mathbb{R}_0(\mathbb{S}) \cdot (1 - p)^{N+1}$
1	$\mathbb{R}_1(\mathbb{S}) \cdot p(1 - p)^{N-1}$	$\mathbb{R}_1(\mathbb{S}) \cdot p(1 - p)^N$
2	$\mathbb{R}_2(\mathbb{S}) \cdot p^2(1 - p)^{N-2}$	$\mathbb{R}_2(\mathbb{S}) \cdot p^2(1 - p)^{N-1}$
\vdots	\vdots	\vdots
$N-1$	$\mathbb{R}_{N-1}(\mathbb{S}) \cdot p^{N-1}(1 - p)$	$\mathbb{R}_{N-1}(\mathbb{S}) \cdot p^{N-1}(1 - p)^2$
N	0	$\mathbb{R}_N(\mathbb{S}) \cdot p^N(1 - p)$
$N+1$	\times	0

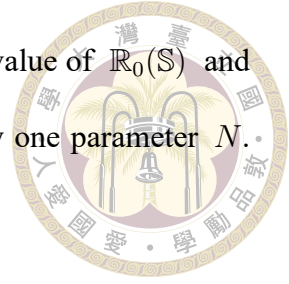
Table 4.3: All the components of $\mathbb{E} [|\bar{\mathbb{U}}|]$ of the ULA with N sensors and $N+1$ sensors.

Table 4.3 lists all the $\mathbb{P}_k(p, \mathbb{S})$ of ULA N and ULA $N+1$ from $k = 0$ to $k = N+1$ based on (4.19). The intersection of p of their $\mathbb{P}_k(p, \mathbb{S})$ for a certain k can be solved by the equation

$$\mathbb{R}_k(\text{ULA } N) \cdot p^k(1 - p)^{N-k} = \mathbb{R}_k(\text{ULA } N+1) \cdot p^k(1 - p)^{N-k+1}, \quad (4.37)$$

where $k = 0, 1, \dots, N$. There are the same components on both sides of (4.37), and they can be deleted. Thus, the answer of p between 0 and 1 is

$$p = 1 - \frac{\mathbb{R}_k(\text{ULA } N)}{\mathbb{R}_k(\text{ULA } N+1)}, \quad (4.38)$$



where $k = 0, 1, \dots, N$. Moreover, through (4.34) and (4.35), the value of $\mathbb{R}_0(\mathbb{S})$ and $\mathbb{R}_1(\mathbb{S})$ of $ULA N$ and $ULA N+1$ can be easily obtained with only one parameter N .

That is, when $k = 0$, the intersection of p is

$$1 - \frac{2N - 1}{2(N + 1) - 1} = \frac{2}{2N + 1} = T_1, \quad (4.39)$$

and when $k = 1$, the intersection of p is

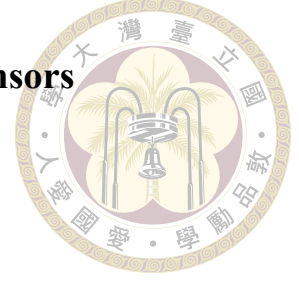
$$1 - \frac{2N^2 - N - 4}{2(N + 1)^2 - (N + 1) - 4} = \frac{4N + 1}{2N^2 + 3N - 3} = T_2. \quad (4.40)$$

If comparing the values of (4.39) and (4.40), then we can find that

$$\begin{aligned} T_1 - T_2 &= \frac{2}{2N + 1} - \frac{4N + 1}{2N^2 + 3N - 3} \\ &= \frac{4N^2 + 6N - 6 - (4N + 1)(2N + 1)}{(2N + 1)(2N^2 + 3N - 3)} \\ &= \frac{-2N^2 - 4}{(2N + 1)(2N^2 + 3N - 3)} \\ &< 0. \end{aligned} \quad (4.41)$$

Thus, it is sure that the intersections of p of $k = 1$ is larger than the intersections of p of $k = 0$. Except for these two intersections that can be obtained from the parameter N , the other intersections must be calculated from the summation of $|\bar{U}_k^j|$, and then get it from (4.38).

4.3.2 ULA with N sensors and ULA with $N + 2$ sensors



$\mathbb{P}_k(p, \mathbb{S})$ \ \mathbb{S}	ULA N	ULA $N+2$
k		
0	$\mathbb{R}_0(\mathbb{S}) \cdot (1 - p)^N$	$\mathbb{R}_0(\mathbb{S}) \cdot (1 - p)^{N+2}$
1	$\mathbb{R}_1(\mathbb{S}) \cdot p(1 - p)^{N-1}$	$\mathbb{R}_1(\mathbb{S}) \cdot p(1 - p)^{N+1}$
2	$\mathbb{R}_2(\mathbb{S}) \cdot p^2(1 - p)^{N-2}$	$\mathbb{R}_2(\mathbb{S}) \cdot p^2(1 - p)^N$
\vdots	\vdots	\vdots
$N-1$	$\mathbb{R}_{N-1}(\mathbb{S}) \cdot p^{N-1}(1 - p)$	$\mathbb{R}_{N-1}(\mathbb{S}) \cdot p^{N-1}(1 - p)^3$
N	0	$\mathbb{R}_N(\mathbb{S}) \cdot p^N(1 - p)^2$
$N+1$	\times	$\mathbb{R}_{N+1}(\mathbb{S}) \cdot p^{N+1}(1 - p)$
$N+2$	\times	0

Table 4.4: All the components of $\mathbb{E} [\bar{U}]$ of the ULA with N sensors and $N+2$ sensors.

Second, we compare ULA N with ULA $N+2$. Table 4.4 lists their respective $\mathbb{P}_k(p, \mathbb{S})$ from $k = 0$ to $k = N + 2$. The process is very similar with the comparison in Section 4.3.1. The intersection of p of each k term can be solved by the equation

$$\mathbb{R}_k(\text{ULA } N) \cdot p^k(1 - p)^{N-k} = \mathbb{R}_k(\text{ULA } N+2) \cdot p^k(1 - p)^{N-k+2}, \quad (4.42)$$

where $k = 0, 1, \dots, N$. Deleting the same components on both sides of (4.42) will let us obtain the answer of p between 0 and 1 that

$$p = 1 - \sqrt{\frac{\mathbb{R}_k(\text{ULA } N)}{\mathbb{R}_k(\text{ULA } N+2)}}. \quad (4.43)$$



When $k = 0$, the intersection of p is

$$1 - \sqrt{\frac{2N - 1}{2(N + 2) - 1}} = W_1, \quad (4.44)$$

and when $k = 1$, the intersection of p is

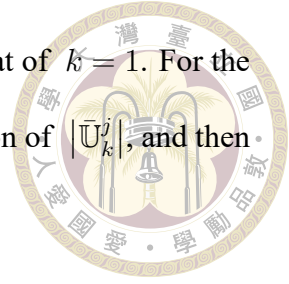
$$1 - \sqrt{\frac{2N^2 - N - 4}{2(N + 2)^2 - (N + 2) - 4}} = W_2. \quad (4.45)$$

Similarly, we compare the values of (4.44) and (4.45).

$$\begin{aligned} & W_1 - W_2 \\ &= \left(1 - \sqrt{\frac{2N - 1}{2(N + 2) - 1}}\right) - \left(1 - \sqrt{\frac{2N^2 - N - 4}{2(N + 2)^2 - (N + 2) - 4}}\right) \\ &= \sqrt{\frac{2N^2 - N - 4}{2(N + 2)^2 - (N + 2) - 4}} - \sqrt{\frac{2N - 1}{2(N + 2) - 1}} \\ &= \sqrt{\frac{(2N^2 - N - 4)(2N + 3)}{(2N^2 + 7N + 2)(2N + 3)}} - \sqrt{\frac{(2N^2 + 7N + 2)(2N - 1)}{(2N^2 + 7N + 2)(2N + 3)}}. \end{aligned} \quad (4.46)$$

Now we just need to compare the numerator, and then we can get

$$\begin{aligned} & (2N^2 - N - 4)(2N + 3) - (2N^2 + 7N + 2)(2N - 1) \\ &= -8N^2 - 6N - 10 \\ &< 0. \end{aligned} \quad (4.47)$$



This result tells us the intersection of p of $k = 0$ is smaller than that of $k = 1$. For the other intersections, we can only obtain the answer from the summation of $|\bar{U}_k^j|$, and then get it from (4.43).

4.3.3 Numerical Results

In Figure 4.19, we compare the $\mathbb{P}_k(p, \mathbb{S})$ of ULA 7 and ULA 8 from $k = 0$ to $k = 7$. Here N is equal to 7. The intersection of p of $k = 0$ can be obtained through (4.39) that

$$\frac{2}{2N + 1} = \frac{2}{15} = 0.1333, \quad (4.48)$$

and the intersection of $k = 1$ can be obtained through (4.40) that

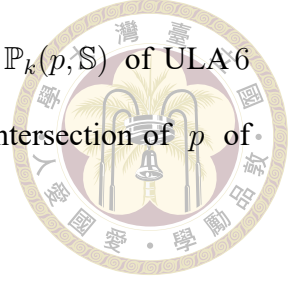
$$\frac{4N + 1}{2N^2 + 3N - 3} = \frac{29}{116} = 0.25. \quad (4.49)$$

We calculate the other intersections of p through (4.38) and $\mathbb{R}_k(\mathbb{S})$. We can find that for $k = 2, 3, \dots, 7$, all the intersections are larger than 0.25. Thus, it is sure that in the range of $0 \leq p \leq 0.1333$, the following inequality holds true.

$$\mathbb{P}_k(p, \text{ULA } N) < \mathbb{P}_k(p, \text{ULA } N+1), \quad \text{for } k = 0, 1, \dots, N. \quad (4.50)$$

Since $\mathbb{E} [|\bar{U}|] = \sum_{k=0}^{|\mathbb{S}|} \mathbb{P}_k(p, \mathbb{S})$, we can get the following result based on (4.50).

$$\mathbb{E} [|\bar{U}|]_{\text{ULA } N} < \mathbb{E} [|\bar{U}|]_{\text{ULA } N+1}, \quad \text{for } 0 \leq p \leq \frac{2}{2N + 1}. \quad (4.51)$$



In Figure 4.20, we have another experiment. The comparison of $\mathbb{P}_k(p, \mathbb{S})$ of ULA 6 and ULA 8 from $k = 0$ to $k = 7$. Here N is equal to 6. The intersection of p of $k = 0$ can be obtained through (4.44) that

$$1 - \sqrt{\frac{11}{15}} = 0.1437, \quad (4.52)$$

and the intersection of $k = 1$ can be obtained through (4.45) that

$$1 - \sqrt{\frac{62}{116}} = 0.2689. \quad (4.53)$$

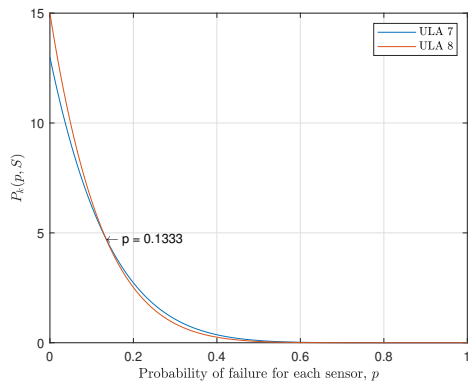
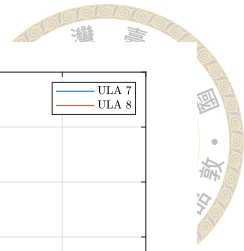
Also, the other intersections of p are calculated from $\mathbb{R}_k(\mathbb{S})$ based on (4.43). Here we can find that for $k = 2, 3, \dots, 6$, all the intersections are larger than 0.2689. Therefore, it is sure that in the range of $0 \leq p \leq 0.1437$, the following inequality holds true.

$$\mathbb{P}_k(p, \text{ULA } N) < \mathbb{P}_k(p, \text{ULA } N+2) \quad \text{for } k = 0, 1, \dots, N. \quad (4.54)$$

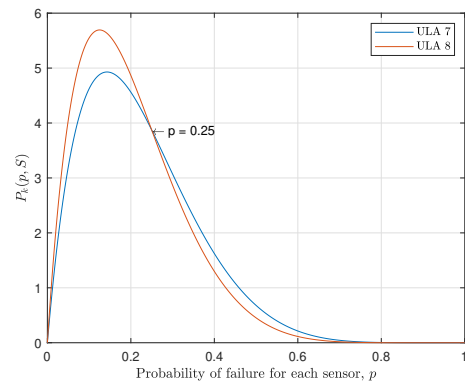
Then we can get the following result based on (4.54).

$$\mathbb{E} [|\bar{U}|]_{\text{ULA } N} < \mathbb{E} [|\bar{U}|]_{\text{ULA } N+2}, \quad \text{for } 0 \leq p \leq 1 - \sqrt{\frac{2N-1}{2(N+2)-1}}. \quad (4.55)$$

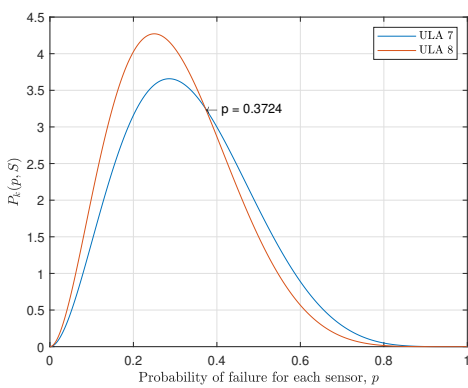
Next, we sum up all the $\mathbb{P}_k(p, \mathbb{S})$ of \mathbb{S} to compare $\mathbb{E} [|\bar{U}|]$ of these ULA with different number of sensors. Figure 4.21 shows the result. It can be observed that the curves confirm the inequality in (4.51) and (4.55).



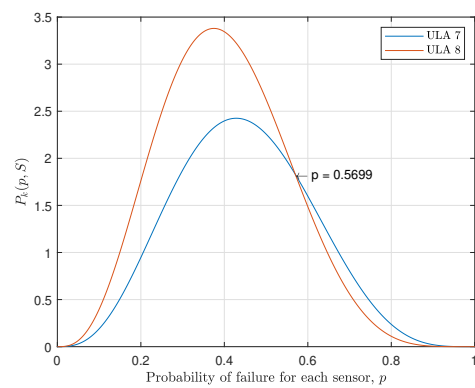
(a) $k = 0$.



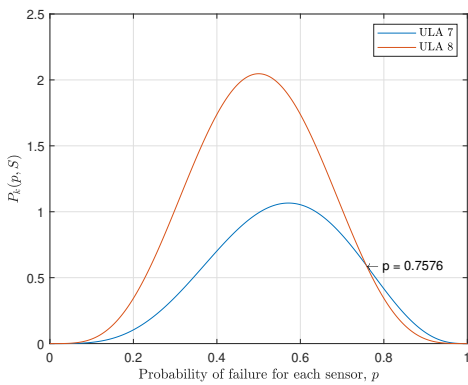
(b) $k = 1$.



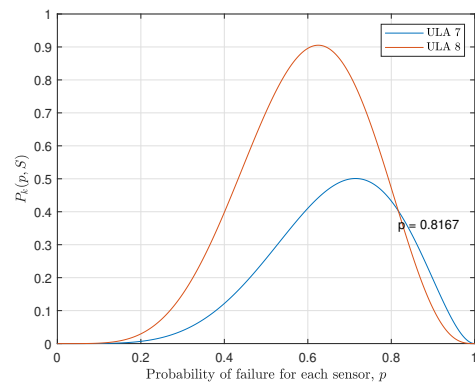
(c) $k = 2$.



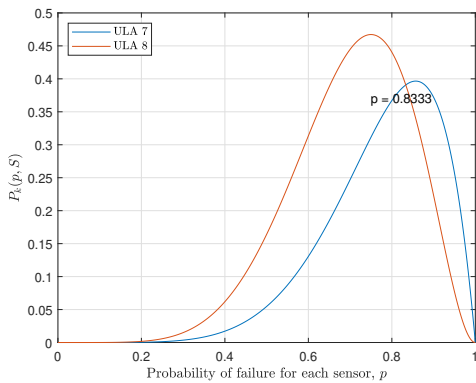
(d) $k = 3$.



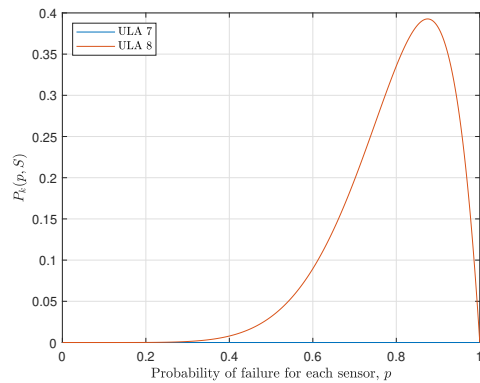
(e) $k = 4$.



(f) $k = 5$.

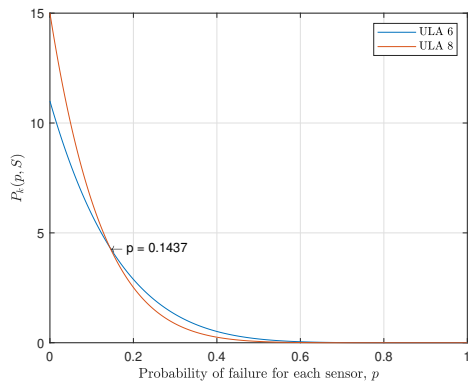
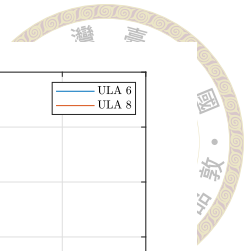


(g) $k = 6$.

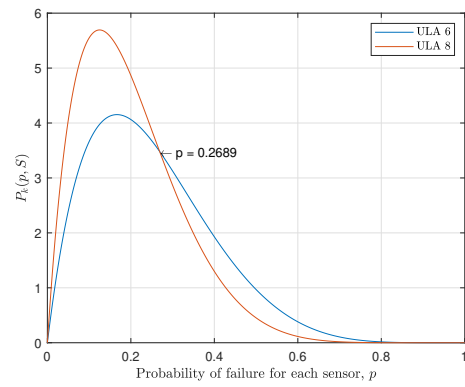


(h) $k = 7$.

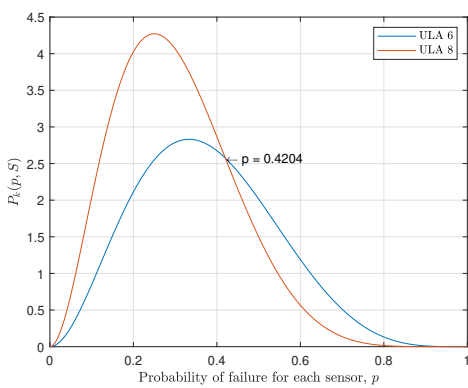
Figure 4.19: The comparison of ULA 7 and ULA 8 of each k item in Table 4.3.



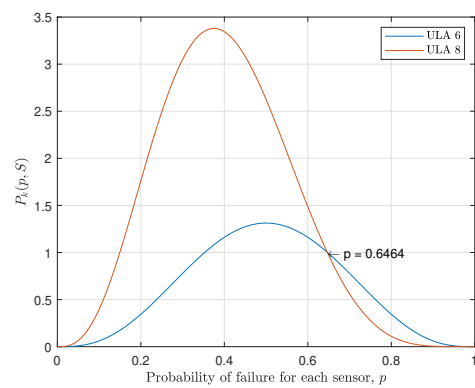
(a) $k = 0$.



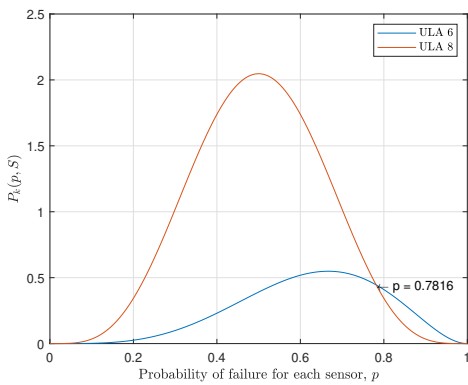
(b) $k = 1$.



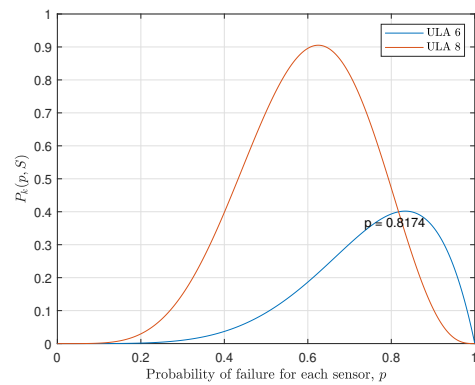
(c) $k = 2$.



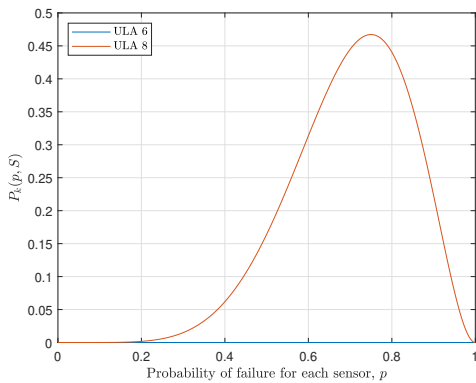
(d) $k = 3$.



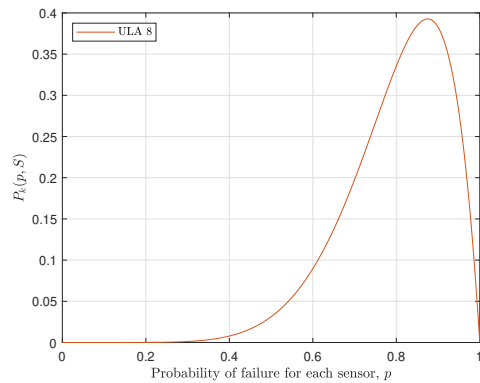
(e) $k = 4$.



(f) $k = 5$.



(g) $k = 6$.



(h) $k = 7$.

Figure 4.20: The comparison of ULA 6 and ULA 8 of each k item in Table 4.4.

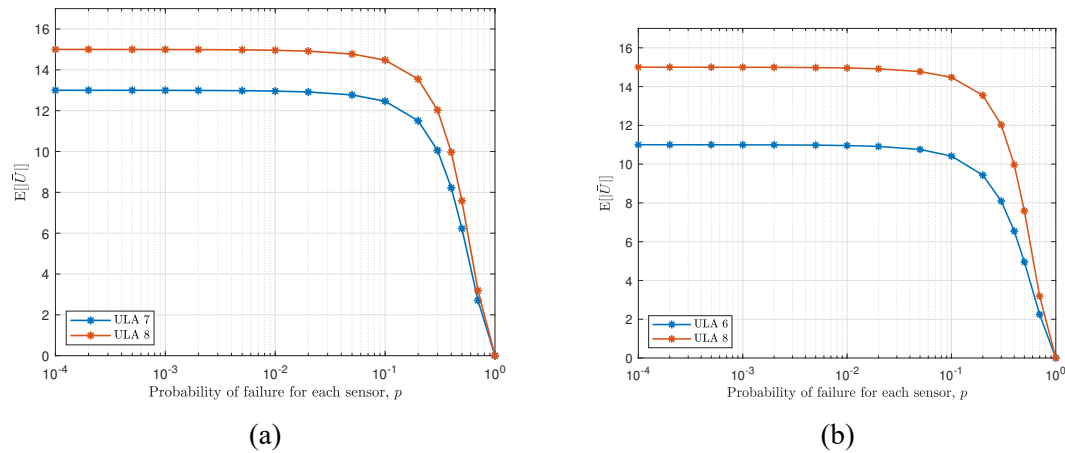
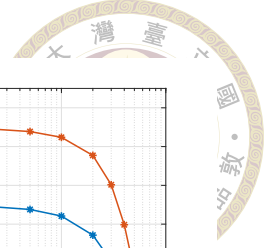


Figure 4.21: The comparisons of $\mathbb{E} [|\bar{\mathbf{U}}|]$ of (a) ULA with 7 sensors and 8 sensors. (b) ULA with 6 sensors and 8 sensors.

Finally, we compare ULA 6, ULA 7 and ULA 8 of their respective $0.9 \cdot |\mathbf{U}|$ and $p_{90\%}$. Also, we will use them to estimate one source, $\theta = \pi/4$, based on random sensor failures. In Figure 4.22, we can find that ULA 8 is the most robust array, and it has the highest $0.9 \cdot |\mathbf{U}|$ as well. If considering the generalized 1-fragility, $\mathcal{F}_1(\text{ULA } 8, \mathcal{S}_{\mathbf{U}})$ is 0.0333, $\mathcal{F}_1(\text{ULA } 7, \mathcal{S}_{\mathbf{U}})$ is 0.0440, and $\mathcal{F}_1(\text{ULA } 6, \mathcal{S}_{\mathbf{U}})$ is 0.0606. Therefore, we know that ULA 8 is more capable of maintaining its original $|\mathbf{U}|$ than ULA 6 and ULA 7 through (4.17).

In the simulation of Figure 4.22, we set SNR to be 0 dB and the number of snapshots to be 500. For each probability p , 5000 $\bar{\mathbf{S}}$ will be generated. $\bar{\mathbf{S}}$ is defined as $\mathbf{S} \setminus \mathbf{A}$. Also, for each $\bar{\mathbf{S}}$, we will do 100 Monte-Carlo runs. Therefore, each data point is averaged from $5 \cdot 10^5$ Monte-Carlo runs. We can find that the estimation MSE of ULA 8 is the least with all the sensor failure probability p . On the other hand, ULA 6 get the worst performance with all probability p . One reason is the number of physical sensors. ULA 8 has more sensors to receive the signal data. From the view of weight function, we mentioned two properties of ULA in Section 2.3.4. We know that for ULA, $w(\pm m) = N - m$, for $0 \leq m \leq N - 1$. Hence, the weight function on the coarray index m of ULA 8 is

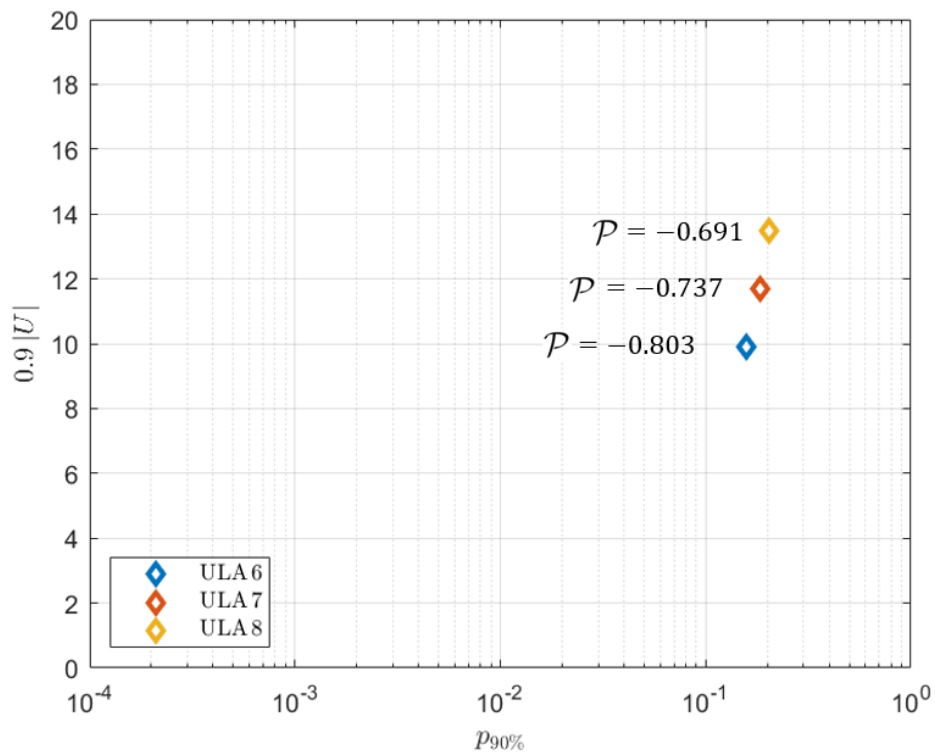
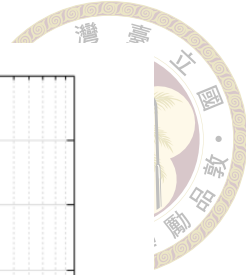


Figure 4.22: The comparisons of $p_{90\%}$ and $0.9 \cdot |U|$ of three ULA. They have 6 sensors, 7 sensors and 8 sensors, respectively.

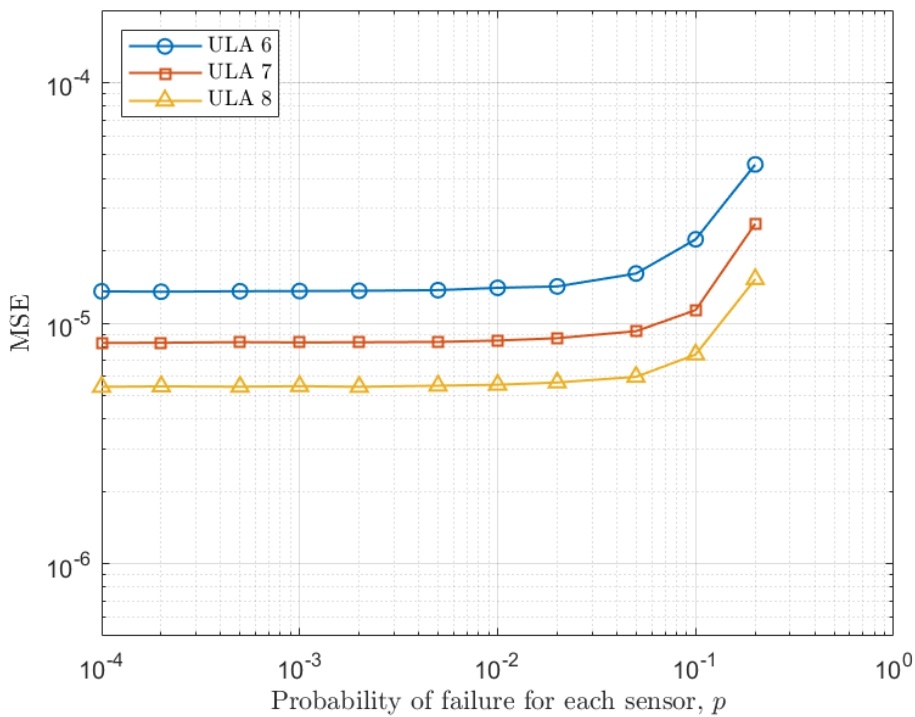


Figure 4.23: The estimation error comparison of the three ULA. They have 6 sensors, 7 sensors and 8 sensors, respectively. We estimate one source with $\theta = \pi/4$. SNR is 0 dB and the number of snapshots is 500. Each point is averaged from $5 \cdot 10^5$ Monte-Carlo runs.

larger than that of ULA 7 and ULA 6. Moreover, we can get some information by observing Figure 4.22 and Figure 4.23 together. In Figure 4.22, the point of ULA 7 is on the top-right side of ULA 6, and ULA 7 have better performance than ULA 6 with all probability p in Figure 4.23. The same result can be observed if we choose any two arrays in Figure 4.22. However, the time to get Figure 4.23 is about 40 hours, and the time to get Figure 4.22 only need about 1 second. Therefore, finding the connection between the two-dimensional point and the simulation result is a research direction.

4.4 Concluding Remarks

In this chapter, we simultaneously considered two factors impacting the estimation performance: the size of \bar{U} and the sensor failure probability p . We derived the expected value of $|\bar{U}|$ based on p . Also, we derived the relationship between $\mathbb{E}[|\bar{U}|]$ and $\mathcal{F}_k(\mathbb{S}, \mathcal{S}_U)$ in (4.17). The approximation of $\mathbb{E}[|\bar{U}|]$ is $\mathbb{E}[|\bar{U}|]_{approx.K}$. For different K , we had the inequality in (4.23). Afterwards, we had some comparisons of different arrays based on $\mathbb{E}[|\bar{U}|]$. The results were shown in Section 4.2.3. For the analysis of ULA, we introduced the closed-form to solve the intersection of p of $\mathbb{P}_k(p, \mathbb{S})$ between two ULA with different number of sensors. Here k is equal to 0 or 1. Then we had the numerical results to obtained the inequality of $\mathbb{E}[|\bar{U}|]$ of ULA in (4.51) and (4.55).

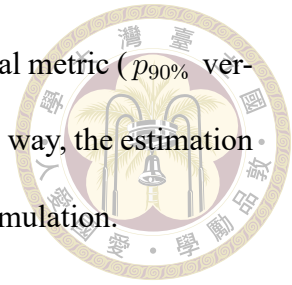


Chapter 5 Conclusion and Future Work

In Chapter 3, we proposed some properties related to the symmetrical arrays. Through these properties, we got the lower bound and the upper bound of the generalized 1-fragility and showed that symmetrizing the array definitely increase the robustness based on $\mathbb{D} = \tilde{\mathbb{D}}$. Also, we proved that symmetrical coprime arrays can reach the lower bound of generalized 1-fragility, namely they are the most robust arrays. In Chapter 4, we considered random sensor failures and proposed the expected value expression of $|\bar{\mathbb{U}}|$. We defined a robustness metric which combine the $|\bar{\mathbb{U}}|$ and the probability p . It has not been proposed before. Through the simulations, we knew that the symmetrical array can obtain better performance than its original version with all the sensor failure probabilities. Additionally, we discussed the ULA with different number of sensors. We found that the ULA with more number of sensors are able to enhance the robustness and the performance at the same time. For the future work, we have three directions:

1. Relaxing the assumptions for proving $\mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{I}_{\mathbb{U}}) < \mathcal{F}_1(\mathbb{S}, \mathcal{I}_{\mathbb{U}})$. For instance, if $|\mathbb{U}| > 1$, than $\mathcal{F}_1(\tilde{\mathbb{S}}, \mathcal{I}_{\mathbb{U}}) < \mathcal{F}_1(\mathbb{S}, \mathcal{I}_{\mathbb{U}})$.
2. Proving $\mathbb{E} [|\bar{\mathbb{U}}|]_{\tilde{\mathbb{S}}} \geq \mathbb{E} [|\bar{\mathbb{U}}|]_{\mathbb{S}}$ for all sensor failure probability p .


3. Finding the theoretical relationship between the two dimensional metric ($p_{90\%}$ versus $0.9 \cdot |\mathbb{U}|$) and the simulation result (p versus MSE). In this way, the estimation performance of the array at each p can be known before the simulation.

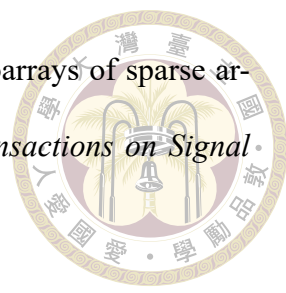


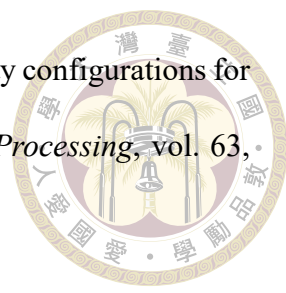


References

- [1] J. Li and P. Stoica, *MIMO radar signal processing*. John Wiley & Sons, 2008.
- [2] J. R. Guerci, *Space-time adaptive processing for radar*. Artech House, 2014.
- [3] H. L. VanTrees, *Optimum array processing: Part IV of detection, estimation, and modulation theory*. John Wiley & Sons, 2004.
- [4] J. Benesty, J. Chen, and Y. Huang, *Microphone array signal processing*, vol. 1. Springer Science & Business Media, 2008.
- [5] P. P. Vaidyanathan and P. Pal, "Sparse sensing with co-prime samplers and arrays," *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 573–586, 2010.
- [6] P. Pal and P. P. Vaidyanathan, "Nested arrays: A novel approach to array processing with enhanced degrees of freedom," *IEEE Transactions on Signal Processing*, vol. 58, no. 8, pp. 4167–4181, 2010.
- [7] M. Wang and A. Nehorai, "Coarrays, MUSIC, and the Cramér–Rao bound," *IEEE Transactions on Signal Processing*, vol. 65, no. 4, pp. 933–946, 2016.
- [8] C. Zhou and J. Zhou, "Direction-of-arrival estimation with coarray ESPRIT for co-prime array," *Sensors*, vol. 17, no. 8, p. 1779, 2017.

- 
- [9] P. Pal and P. P. Vaidyanathan, “Copriime sampling and the MUSIC algorithm,” in *2011 Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)*, pp. 289–294, 2011.
- [10] C. L. Liu and P. P. Vaidyanathan, “Remarks on the spatial smoothing step in coarray MUSIC,” *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1438–1442, 2015.
- [11] P. O’Connor and A. Kleyner, *Practical reliability engineering*. John Wiley & Sons, 2012.
- [12] A. Myers, *Complex system reliability*. Springer, 2010.
- [13] H. Taylor and S. W. Golomb, “Rulers part I,” *Univ. Southern Calif., Los Angeles, Tech. Rep*, pp. 85–05, 1985.
- [14] A. Moffet, “Minimum-redundancy linear arrays,” *IEEE Transactions on antennas and propagation*, vol. 16, no. 2, pp. 172–175, 1968.
- [15] M. Wang, Z. Zhang, and A. Nehorai, “Direction finding using sparse linear arrays with missing data,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3066–3070, 2017.
- [16] S. Vigneshwaran, N. Sundararajan, and P. Saratchandran, “Direction of arrival (DOA) estimation under array sensor failures using a minimal resource allocation neural network,” *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 2, pp. 334–343, 2007.
- [17] C. L. Liu and P. P. Vaidyanathan, “Robustness of coarrays of sparse arrays to sensor failures,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3231–3235, 2018.

- 
- [18] C. L. Liu and P. P. Vaidyanathan, “Robustness of difference coarrays of sparse arrays to sensor failures—part II: Array geometries,” *IEEE Transactions on Signal Processing*, vol. 67, no. 12, pp. 3227–3242, 2019.
- [19] C. L. Liu, “A general framework for the robustness of structured difference coarrays to element failures,” in *2020 IEEE 11th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pp. 1–5, 2020.
- [20] C. L. Liu and P. P. Vaidyanathan, “Optimizing minimum redundancy arrays for robustness,” in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pp. 79–83, 2018.
- [21] X. Xu, Z. Ye, Y. Zhang, and C. Chang, “A deflation approach to direction of arrival estimation for symmetric uniform linear array,” *IEEE Antennas and Wireless Propagation Letters*, vol. 5, pp. 486–489, 2006.
- [22] Z. Ye and X. Xu, “DOA estimation by exploiting the symmetric configuration of uniform linear array,” *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 12, pp. 3716–3720, 2007.
- [23] C. L. Liu and P. P. Vaidyanathan, “Maximally economic sparse arrays and cantor arrays,” in *2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pp. 1–5, 2017.
- [24] C. L. Liu and P. P. Vaidyanathan, “Robustness of difference coarrays of sparse arrays to sensor failures—part I: A theory motivated by coarray music,” *IEEE Transactions on Signal Processing*, vol. 67, no. 12, pp. 3213–3226, 2019.
- [25] D. H. Johnson and D. E. Dudgeon, *Array signal processing: concepts and techniques*. Simon & Schuster, Inc., 1992.

- 
- [26] S. Qin, Y. D. Zhang, and M. G. Amin, “Generalized coprime array configurations for direction-of-arrival estimation,” *IEEE Transactions on Signal Processing*, vol. 63, no. 6, pp. 1377–1390, 2015.
- [27] K. Adhikari, J. R. Buck, and K. E. Wage, “Extending coprime sensor arrays to achieve the peak side lobe height of a full uniform linear array,” *EURASIP Journal on Advances in Signal Processing*, vol. 2014, no. 1, pp. 1–17, 2014.
- [28] A. Raza, W. Liu, and Q. Shen, “Thinned coprime array for second-order difference coarray generation with reduced mutual coupling,” *IEEE Transactions on Signal Processing*, vol. 67, no. 8, pp. 2052–2065, 2019.
- [29] Z. Zheng, W. Q. Wang, Y. Kong, and Y. D. Zhang, “MISC array: A new sparse array design achieving increased degrees of freedom and reduced mutual coupling effect,” *IEEE Transactions on Signal Processing*, vol. 67, no. 7, pp. 1728–1741, 2019.
- [30] M. T. Heideman, “Convolution and polynomial multiplication,” in *Multiplicative Complexity, Convolution, and the DFT*, pp. 27–60, Springer, 1988.