Mean and Median of PSD Matrices on a Riemannian Manifold: Application to Detection of Narrowband Sonar Signals

MEAN AND MEDIAN OF PSD MATRICES ON A RIEMANNIAN MANIFOLD: APPLICATION TO DETECTION OF NARROWBAND SONAR SIGNALS

BY

HUIYING JIANG, B.Eng.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF APPLIED SCIENCE

Master of Applied Science (2016)   McMaster University

(Electrical & Computer Engineering)   Hamilton, Ontario, Canada

TITLE: Mean and Median of PSD Matrices on a Riemannian
                      Manifold: Application to Detection of Narrowband Sonar
                      Signals

AUTHOR: Huiying Jiang

                      B.Eng., (Measurement Control Technology and Instru-
                      ment)
                      University of Electronic Science and Technology of
                      China, Chengdu, China

SUPERVISOR: Dr. Kon Max Wong

NUMBER OF PAGES: xi, 97

This thesis is dedicated to my parents,

for their love and encouragement.

Abstract

We investigate the employment of power spectral density (PSD) matrix, which is constructed by the received signals in a multi-sensor system and contains additional cross-correlation information, as a feature in signal processing. Since the PSD matrices are structurally constrained, they form a manifold in signal space. The commonly used Euclidean distance (ED) to measure the distance between two such matrices are not informative or accurate. Riemannian distances (RD), which measure distances along the surface of the manifold, should be employed to give more meaningful measurements. Furthermore, the principle that the geodesics on the manifold can be lifted to an isometric Euclidean space is emphasized since any processing involving the optimization of the geodesics can be lifted to the isometric Euclidean space and be carried out in terms of the equivalent Euclidean metric. Application of this principle is illustrated by having efficient algorithms locating the mean and median of the PSD matrices on the manifold developed. These concepts are then applied to the detection of narrow-band sonar signals from which the decision rule is set up by translating the measure reference. In order to further enhance the detecton performance, an algorithm is developed for obtaining the optimum weighting matrix which can better classify the signal from noise. The experimental results show that the performance by the PSD matrices being the detection feature is very encouraging.

Acknowledgements

I would like to thank all the people who gave me the help to complete this thesis. First and foremost, I would like to express the deep gratitude to my supervisor Dr. Kon Max Wong, who offered me the support and guidance in my research over the past two years. His conscientious academic attitude, insightful thinking and passionate commitment to research have a great impact on me and inspire me to keep working hard. Dr. Wong always made efforts to teach me the way of doing research and develop my ability of thinking and working independently. Without him, I would not make such a progress in research. Besides, this thesis is based on the journal paper that Dr. Wong is going to submit, which helped me a lot as a reference.

In addition, I would like to thank Dr. Jian-Kang Zhang for his patience in helping me resolve the problems. Dr. Zhang, who has a keen insight in both signal processing and mathematics, provided much valuable advice for me.

Furthermore, I would like to appreciate all the colleagues in my lab for their help in academic problems and making my life enjoyable. Especially, I am grateful to my friend Mr. Yanbo Wei for his encouragement and suggestions when there was a tough time in my research process.

Finally, I wish to thank my parents for their support and caring. Their love gives me the courage and confidence to overcome all the challenges in my life.

Notations and abbreviations

Notations

$\boldsymbol{S}$	Matrices
$\boldsymbol{s}$	Column vectors
$(\cdot)^T$	Matrix transpose
$(\cdot)^H$	Matrix hermitian
$(\cdot)^{-1}$	Matrix inverse
$\langle \cdot, \cdot \rangle$	Inner product
$|\cdot|$	Magnitude of a complex quantity
$\|\cdot\|$	Euclidean norm of a vector or a matrix
$\Re(\cdot)$	Real part of a matrix
$\mathbb{E}[\cdot]$	Expectation
$\mathrm{tr}(\cdot)$	Trace of matrices
$\mathrm{vec}\{\cdot\}$	Vectorization of a matrix
$\mathrm{diag}\{\cdot\}$	Diagonal matrix
$\boldsymbol{0}$	Zero matrix
$\boldsymbol{I}_M$	$M \times M$ identity matrix
$\mathcal{C}$	Set

$\mathcal{M}$	Manifold
$\mathcal{H}$	Euclidean space
$\mathcal{T}_{\mathcal{M}}(\cdot)$	Tangent space at a point of $\mathcal{M}$
$\mathcal{U}_{\mathcal{H}}(\cdot)$	Euclidean subspace at a point of $\mathcal{H}$

Abbreviations

PSD	Power Spectral Density
DFT	Discrete Fourier Transform
ED	Euclidean Distance
RD	Riemannian Distance
EMn	Euclidean Mean
EMd	Euclidean Median
RMn	Riemannian Mean
RMd	Riemannian Median
WRMn	Weighted Riemannian Mean
WRMd	Weighted Riemannian Median
SVD	Singular Value Decomposition
SWMA	Split Window Moving Average
ROC	Receiver Operating Characteristics
SNR	Signal to Noise Ratio

Contents

List of Figures

Chapter 1

Introduction

1.1 The PSD Matrix

Multi-sensor signal processing is essential not only to civilian applications but also to military defence. In a multi-sensor observation system, the power spectral density (PSD) matrix is the discrete Fourier transform (DFT) of the covariance matrix of stochastic signals received from different sensors. It is of vital importance in real applications because compared to the conventional power spectrum, the PSD matrix contains more correlation information of received signals between different sensors. From the covariance or PSD matrices, various signal parameters can be extracted. Algorithms applying on these matrices have been widely developed and employed in multi-sensor signal processing for many years, such as beamforming, adaptive signal filtering, the detection of slowly moving targets, localization/separation of signal sources, extraction of signals, classification of targets, etc [1, 2, 3, 4, 5, 6]. Since both covariance and PSD matrices can be used to evaluate various parameters needed for extraction of similar information in a wide-sense stationary process, we will focus on

1

the use of PSD matrices for signal processing in this thesis.

One problem is that it may not be very straightforward to directly use the PSD matrix as a *feature* in signal processing for which we often have to measure the distance between two PSD matrices. After taking the DFT of the covariance matrices, we obtain the *positive semi-definite Hermitian* PSD matrices of the signal [7], which form a *manifold* in the real linear vector space of all Hermitian matrices [8]. Therefore, the commonly used *Euclidean distance* (ED) [9] may not be appropriate to measure the distance between two PSD matrices. A similar concept is the distance between two cities on earth, which is not accurate if measured by the ED. Hence, we realize that it will be more appropriate to measure the distance between two of these matrices along the surface of the manifold, i.e., *Riemannian distance* (RD). Three different closed-form expressions of RD for the PSD matrix (known as d_{R_1}, d_{R_2} and d_{R_3}) have been introduced in [8]. Furthermore, in practice, sometimes we may have the chance to obtain the prior knowledge of matrices in signal processing. In order to emphasize and de-emphasize the matrices based on prior information, the Riemannian distance can be optimally weighted. However, Li and Wong has proved that d_{R_3} is *weight-invariant* [8], thus we pay more attention to d_{R_1} and d_{R_2} in this thesis.

In addition, we may want to evaluate some statistic features of random PSD matrices, such as mean and median, corresponding to different measurement of distances. It is well known [10] that the mean minimizes the sum of squared distances from the points, whereas the median is defined as the value which minimizes the sum of absolute distances from the points. In the case of power spectrum, which is a finite set of real scalars, the mean is the average value of these scalars and the median is the middle value of the sorted set. However, for PSD matrices, since we have different

measurement of distances, according to the definition, we would have different deriva-
tions of mean and median. The objective of this thesis is to address the measurement
of distances on the manifold and find the mean and median of random PSD matri-
ces using these distances. After that, we can apply these concepts to some common
applications of signal processing.

1.2   Contents of Our Thesis

As we mentioned before, the Riemannian distance provides the accurate measure-
ment of distance for PSD matrices. To find the mean of PSD matrices for d_{R_1}, the
algorithm given in [11] may not be able to obtain the optimum weighting matrix,
based on which we develop a new algorithm. For finding the mean for d_{R_2}, the algo-
rithm given in [11] is applied. In many situations, it is desirable to weight the PSD
matrices in order to enhance their similarities/dissimilarities. Following this reason,
we put forward an algorithm to obtain the optimum weighting matrix for weighted
Riemannian distance in terms of mean. A good application of these concepts is the
narrow-band sonar detection. It is verified that using the mean for RD as a detection
feature shows a better detection performance than that for ED, which is consistent
with our expectation. Furthurmore, consider that the mean may have a bias on es-
timating the noise when the number of signals increases in the multi-sensor system.
However, the median, which is a prominent ordered statistic estimator, might be a
more proper choice as a detection feature due to its robustness. Therefore, an algo-
rithm is developed to find the median of PSD matrices for both ED and RD based
on the algorithm in [12] and the optimum weighting matrix in terms of median can
be achieved by a similar algorithm to that in terms of mean. The simulations are

carried out afterwards to show that the median is superior to mean as a detection feature when the number of signals becomes large.

1.3   Structure of the Thesis

There are five chapters in this thesis, which is organized as follows. In Chapter 1, the background knowledge of PSD matrices and our contributions are briefly introduced. In Chapter 2, accurate and informative measurements of distance for PSD matrices, known as Riemannian distance and weighted Riemannian distance, are discussed. In order to investigate the statistic features of random PSD matrices in signal processing, the algorithms of finding mean and median for both ED and RD are demonstrated in Chapter 3. Chapter 4 is devoted to the experimental simulations. In this chapter, the PSD matrix is applied to the detection of narrow-band sonar signals in noise and the performance of different detection features is examined. Further insights of the choice of decision reference is revealed. In order to better distinguish signal from noise, an algorithm for optimum weighting matrix is also established in this chapter. Finally, the conclusions of our thesis and the prospective topics for future research are presented in Chapter 5.

Chapter 2

Distance Between Two PSD Matrices

Assume that we have M channels in the multi-sensor system, the establishment of covariance or PSD matrices is quite easy. The M-channel signal is usually first "cleaned up" by filtering and having artifacts removed. Then the signal is divided into *epochs*, each of T seconds and normalized. The nth epoch can now be represented by an $M \times T$ matrix $\boldsymbol{S}_n$, with the measurements from M channels at the instant t as the elements of its column vector $\boldsymbol{s}_n(t) = [s_{n1}(t), \cdots, s_{nM}(t)]^T$. If T is short enough, $\{\boldsymbol{s}_n(t)\}, t = 1, \cdots, T$ can be considered as wide-sense stationary vectors. Therefore, its arithmetic mean and the covariance matrix can be approximated by taking the corresponding averages over time, i.e., $\boldsymbol{\mu}_n = \mathbb{E}[\boldsymbol{s}_n] \approx \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{s}_n(t) = \hat{\boldsymbol{\mu}}_n$ and $\boldsymbol{R}_n(\tau) = \mathbb{E}\left[\{\boldsymbol{s}_n(t+\tau) - \boldsymbol{\mu}_n\} \{\boldsymbol{s}_n(t) - \boldsymbol{\mu}_n\}^H \right] \approx \frac{1}{T} \sum_{t=1}^{T} \{\boldsymbol{s}_n(t+\tau) - \hat{\boldsymbol{\mu}}_n\} \{\boldsymbol{s}_n(t) - \hat{\boldsymbol{\mu}}_n\}^H$. The resulting $M \times M$ covariance matrix $\boldsymbol{R}_n(\tau)$ is positive semi-definite [7]. Accordingly, we can obtain the *Hermitian, positive semi-definite* PSD matrix after taking the DFT

of $\boldsymbol{R}_n(\tau)$ such that, at frequency ω,

$$\boldsymbol{P}_n(\omega) = \frac{1}{2\pi} \sum_\tau \boldsymbol{R}_n(\tau) e^{-j\omega\tau} \tag{2.1}$$

The PSD matrices of a signal epoch $\boldsymbol{S}_n$ constitute a group of points at different frequencies, forming a curve on $\mathcal{M}$. Hence, the PSD matrices of the mth and nth epochs describe two different curves on $\mathcal{M}$, respectively denoted by $\boldsymbol{P}_m(\omega)$ and $\boldsymbol{P}_n(\omega)$, $\omega \in [\omega_{\min}, \omega_{\max}]$. In order to find the distance between the two curves, a measurement of distance should be established between the two points at the *same* frequency ω_i on the two curves, represented by $\boldsymbol{P}_m(\omega_i)$ and $\boldsymbol{P}_n(\omega_i)$, or simply $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$.

2.1   Euclidean Distance d_{E}

In signal processing, the Euclidean distance (ED) is the most commonly used distance measure because it describes the straight-line distance (inner product) between two points in a 3-dimensional Euclidean space and also represents many important physical quantities. For two N-dimensional vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ in the complex signal space, the ED is defined as [9]

$$d_{\mathrm{E}}(\boldsymbol{a}, \boldsymbol{b}) = \|\boldsymbol{a} - \boldsymbol{b}\| = \sqrt{\sum_{i=1}^{N} |a_i - b_i|^2} \tag{2.2}$$

where a_i and b_i are the ith entries of vector $\boldsymbol{a}$ and $\boldsymbol{b}$, respectively. Expanding to matrices, an $M \times M$ complex matrix can also be regarded as a point in the M^2 complex signal space so that the same idea of distance between two such matrices $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$ can be applied. If we write the two matrices in the vector form, i.e., $\mathrm{vec}\boldsymbol{P}_m$

and $\text{vec}\boldsymbol{P}_n$, using the vec-functions [13], the ED between $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$ can be given by

$$
\begin{aligned}
d_{\mathrm{E}}(\boldsymbol{P}_m, \boldsymbol{P}_n) &= d_{\mathrm{E}}(\text{vec}\boldsymbol{P}_m, \text{vec}\boldsymbol{P}_n) \\
&= \|\text{vec}\boldsymbol{P}_m - \text{vec}\boldsymbol{P}_n\| \\
&= \left(\sum_{i=1}^{M} \sum_{j=1}^{M} |m_{ij} - n_{ij}|^2 \right)^{1/2} \\
&= \sqrt{\text{tr}\left[(\boldsymbol{P}_m - \boldsymbol{P}_n)(\boldsymbol{P}_m - \boldsymbol{P}_n)^H \right]}
\end{aligned}
\tag{2.3}
$$

where m_{ij} and n_{ij} are the entries in the ith row and jth column of $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$, respectively. Eq. (2.3) is also called the *Frobenius distance* [14] which is in fact induced by the inner product norm.

2.2 Riemannian Distance

As we mentioned in Chapter 1, ED is not accurate for measuring the distance on the manifold $\mathcal{M}$. When we consider the distance between two PSD matrices, which are structurally constrained, we should measure along the surface of the manifold. It has been given in [8] that the curve on the manifold, parameterized by θ, linking two PSD matrices $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$ with the minimum length is called a *geodesic*, and the *Riemannian distance* (RD) between two PSD matrices is defined as the length of the geodesic, which is given by

$$
d_{\mathrm{R}}(\boldsymbol{P}_m, \boldsymbol{P}_n) \triangleq \min_{\boldsymbol{P}(\theta):[\theta_m, \theta_n] \to \mathcal{M}} \{\ell\left(\boldsymbol{P}(\theta)\right)\}
\tag{2.4}
$$

where $\ell(\boldsymbol{P}) = \int_{\theta_m}^{\theta_n} g_{\boldsymbol{P}}^{1/2}\left(\dot{\boldsymbol{P}}, \dot{\boldsymbol{P}}\right) \mathrm{d}\theta$ is called a *Riemannian metric* at $\boldsymbol{P}$ on $\mathcal{M}$. Here, $\dot{\boldsymbol{P}} = \frac{\mathrm{d}\boldsymbol{P}}{\mathrm{d}\theta}$ and $g_{\boldsymbol{P}}(\dot{\boldsymbol{P}}, \dot{\boldsymbol{P}})$ is an inner product metric. Different definitions of Riemannian metrics lead to different RD. It is difficult to directly evaluate the RD in Eq. (2.4).

To avoid such difficulties, we can employ a mapping $\pi : \mathcal{M} \to \mathcal{H}$, which associates each point $\boldsymbol{P} \in \mathcal{M}$ with $\pi(\boldsymbol{P}) \triangleq \tilde{\boldsymbol{P}} \in \mathcal{H}$ and thus constitutes the *fibre* [15] above $\boldsymbol{P} \in \mathcal{M}$. Now, $\tilde{\boldsymbol{P}}$ is still an $M \times M$ complex matrix but may no longer be positive semi-definite or Hermitian. In other words, each point $\boldsymbol{P} \in \mathcal{M}$ is linked by the fibre above it with the point in $\mathcal{H}$ through the mapping π. Any point along the fibre satisfies the mapping π since $\pi(\pi^{-1}(\boldsymbol{P})) = \boldsymbol{P}$. Therefore, the PSD manifold can be linked to the Euclidean space. For every PSD matrix $\boldsymbol{P} \in \mathcal{M}$, there exists another matrix $\tilde{\boldsymbol{P}} \in \mathcal{H}$ which, though not unique, can be regarded as a representation of $\boldsymbol{P}$ in the Euclidean space $\mathcal{H}$. We can view $\boldsymbol{P}$ as an "image" on $\mathcal{M}$ of the point $\tilde{\boldsymbol{P}}$ in $\mathcal{H}$. The manifold $\mathcal{M}$ and the Euclidean space $\mathcal{H}$ are often called the *base space* and the *total space* respectively. The mapping makes it possible to bring $\boldsymbol{P} \in \mathcal{M}$ to $\tilde{\boldsymbol{P}} \in \mathcal{H}$ by a lifting process, and return $\tilde{\boldsymbol{P}} \in \mathcal{H}$ back to $\boldsymbol{P} \in \mathcal{M}$ by a projecting process along the fibre.

Furthermore, by choosing a particular mapping π, together with an appropriate Riemannian metric, we can find a Euclidean subspace $\mathcal{U}_{\mathcal{H}}$ at $\tilde{\boldsymbol{P}}$ of $\mathcal{H}$, which is *isometric* with $\mathcal{T}_{\mathcal{M}}(\boldsymbol{P})$, the tangent space at $\boldsymbol{P}$ on $\mathcal{M}$. That is to say, the geodesic between $\boldsymbol{P}_m, \boldsymbol{P}_n \in \mathcal{M}$ can be lifted along the fibres to the Euclidean subspace with $\tilde{\boldsymbol{P}}_m, \tilde{\boldsymbol{P}}_n \in \mathcal{U}_{\mathcal{H}}$. The isometry between $\mathcal{U}_{\mathcal{H}}$ and $\mathcal{T}_{\mathcal{M}}$ means that the RD between $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$ on the manifold $\mathcal{M}$ is equal in length to the ED between $\tilde{\boldsymbol{P}}_m$ and $\tilde{\boldsymbol{P}}_n \in \mathcal{U}_{\mathcal{H}}$. Thus, the problem of measuring RD on the manifold can be transformed to measuring the ED in the Euclidean subspace $\mathcal{U}_{\mathcal{H}}$. Following this method, three closed-form expressions

of RD for the PSD matrices on the manifold have been obtained in [8].

2.2.1   Riemannian Distance d_{R_1}

Consider the mapping π:

$$\boldsymbol{P} = \tilde{\boldsymbol{P}}\tilde{\boldsymbol{P}}^H, \quad \text{i.e.,} \quad \tilde{\boldsymbol{P}} = \boldsymbol{P}^{1/2}\boldsymbol{U} \tag{2.5}$$

where $\tilde{\boldsymbol{P}} \in \mathcal{H}$, $\boldsymbol{P} \in \mathcal{M}$, $\boldsymbol{U}$ is a unitary matrix and choosing the Riemannian metric on $\mathcal{M}$ as $g_{\boldsymbol{P}}(\boldsymbol{A}, \boldsymbol{B}) = \frac{1}{2}\mathrm{tr}\boldsymbol{AK}$ with $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{T}_{\mathcal{M}}(\boldsymbol{P})$ and $\boldsymbol{KP} + \boldsymbol{PK} = \boldsymbol{B}$. Then $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$ can be lifted to $\tilde{\boldsymbol{P}}_m, \tilde{\boldsymbol{P}}_n \in \mathcal{U}_{\mathcal{H}}$ by letting $\tilde{\boldsymbol{P}}_m = \boldsymbol{P}_m^{1/2}\boldsymbol{U}_m$ and $\tilde{\boldsymbol{P}}_n = \boldsymbol{P}_n^{1/2}\boldsymbol{U}_n$. Then the geodesic (path of minimum length) between $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$ can be measured by the equivalent length of the shortest straight line joining $\tilde{\boldsymbol{P}}_m$ and $\tilde{\boldsymbol{P}}_n$ in Euclidean subspace $\mathcal{U}_{\mathcal{H}}$, i.e.,

$$\begin{aligned} d_{\mathrm{R}_1}^2(\boldsymbol{P}_m, \boldsymbol{P}_n) &= \min_{\boldsymbol{U}_m, \boldsymbol{U}_n} \left\| \tilde{\boldsymbol{P}}_m - \tilde{\boldsymbol{P}}_n \right\|^2 \\ &= \min_{\boldsymbol{U}_m, \boldsymbol{U}_n} \mathrm{tr}\boldsymbol{P}_m + \mathrm{tr}\boldsymbol{P}_n - 2\Re\left[\mathrm{tr}\left(\boldsymbol{U}_m \boldsymbol{U}_n^H \boldsymbol{P}_n^{1/2} \boldsymbol{P}_m^{1/2} \right) \right] \end{aligned} \tag{2.6}$$

The minimization of Eq. (2.6) is equivalent to the maximization of its last term with respect to $\boldsymbol{U}_n$ and $\boldsymbol{U}_m$. It is well known [13] that the following solution holds if $\boldsymbol{U}_n$ and $\boldsymbol{U}_m$ are the left and right singular vector matrices of $\boldsymbol{P}_n^{1/2}\boldsymbol{P}_m^{1/2}$, respectively.

$$\max_{\boldsymbol{U}_m, \boldsymbol{U}_n} \Re\left[\mathrm{tr}\left(\boldsymbol{U}_m \boldsymbol{U}_n^H \boldsymbol{P}_n^{1/2} \boldsymbol{P}_m^{1/2} \right) \right] = \mathrm{tr}\left[\left(\boldsymbol{P}_m^{1/2} \boldsymbol{P}_n \boldsymbol{P}_m^{1/2} \right)^{1/2} \right] \tag{2.7}$$

Proof. Since $\boldsymbol{U}_n$ and $\boldsymbol{U}_m$ are the left and right singular vector matrices of $\boldsymbol{P}_n^{1/2}\boldsymbol{P}_m^{1/2}$, we have $\boldsymbol{U}_m \boldsymbol{U}_n^H \boldsymbol{P}_n^{1/2} \boldsymbol{P}_m^{1/2} = \boldsymbol{U}_m \boldsymbol{U}_n^H \boldsymbol{U}_n \boldsymbol{\Sigma}_o \boldsymbol{U}_m^H = \boldsymbol{U}_m \boldsymbol{\Sigma}_o \boldsymbol{U}_m^H$ and $\left(\boldsymbol{U}_m \boldsymbol{U}_n^H \boldsymbol{P}_n^{1/2} \boldsymbol{P}_m^{1/2} \right)^H =$

$\boldsymbol{P}_m^{1/2}\boldsymbol{P}_n^{1/2}\boldsymbol{U}_n\boldsymbol{U}_m^H = \boldsymbol{U}_m\boldsymbol{\Sigma}_o\boldsymbol{U}_n^H\boldsymbol{U}_n\boldsymbol{U}_m^H = \boldsymbol{U}_m\boldsymbol{\Sigma}_o\boldsymbol{U}_m^H$. That means $\boldsymbol{U}_m\boldsymbol{U}_n^H\boldsymbol{P}_n^{1/2}\boldsymbol{P}_m^{1/2}$ is a

Hermitian matrix. Therefore, the last term of Eq. (2.6) can be written as:

$$\Re\left[\mathrm{tr}\left(\boldsymbol{U}_m\boldsymbol{U}_n^H\boldsymbol{P}_n^{1/2}\boldsymbol{P}_m^{1/2}\right)\right] = \mathrm{tr}\left\{\left[\left(\boldsymbol{U}_m\boldsymbol{U}_n^H\boldsymbol{P}_n^{1/2}\boldsymbol{P}_m^{1/2}\right)\left(\boldsymbol{U}_m\boldsymbol{U}_n^H\boldsymbol{P}_n^{1/2}\boldsymbol{P}_m^{1/2}\right)^H\right]^{1/2}\right\}$$

$$= \mathrm{tr}\left[\left(\boldsymbol{U}_m\boldsymbol{U}_n^H\boldsymbol{P}_n^{1/2}\boldsymbol{P}_m\boldsymbol{P}_n^{1/2}\boldsymbol{U}_n\boldsymbol{U}_m^H\right)^{1/2}\right] \tag{2.8}$$

Let $\boldsymbol{P}_n^{1/2}\boldsymbol{P}_m\boldsymbol{P}_n^{1/2} = \boldsymbol{V}\boldsymbol{\Sigma}_r\boldsymbol{V}^H$ with $\boldsymbol{\Sigma}_r$ and $\boldsymbol{V}$ being the eigenvalue and eigenvector

matrices of $\boldsymbol{P}_n^{1/2}\boldsymbol{P}_m\boldsymbol{P}_n^{1/2}$, and thus $\boldsymbol{U} = \boldsymbol{U}_m\boldsymbol{U}_n^H\boldsymbol{V}$ is also a unitary matrix. Then, Eq.

(2.8) can be continuously derived as

$$\Re\left[\mathrm{tr}\left(\boldsymbol{U}_m\boldsymbol{U}_n^H\boldsymbol{P}_n^{1/2}\boldsymbol{P}_m^{1/2}\right)\right] = \mathrm{tr}\left[\left(\boldsymbol{U}_m\boldsymbol{U}_n^H\boldsymbol{V}\boldsymbol{\Sigma}_r\boldsymbol{V}^H\boldsymbol{U}_n\boldsymbol{U}_m^H\right)^{1/2}\right]$$

$$= \mathrm{tr}\left[\left(\boldsymbol{U}\boldsymbol{\Sigma}_r\boldsymbol{U}^H\right)^{1/2}\right]$$

$$= \mathrm{tr}\left(\boldsymbol{U}\boldsymbol{\Sigma}_r^{1/2}\boldsymbol{U}^H\right)$$

$$= \mathrm{tr}\left[\left(\boldsymbol{P}_m^{1/2}\boldsymbol{P}_n\boldsymbol{P}_m^{1/2}\right)^{1/2}\right] \tag{2.9}$$

$\square$

Substituting Eq. (2.9) into Eq. (2.6), the RD d_{R_1} between $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$ can be

obtained by

$$d_{\mathrm{R}_1}(\boldsymbol{P}_m, \boldsymbol{P}_n) = \left\|\boldsymbol{P}_m^{1/2}\boldsymbol{U}_m - \boldsymbol{P}_n^{1/2}\boldsymbol{U}_n\right\|$$

$$= \sqrt{\mathrm{tr}\boldsymbol{P}_m + \mathrm{tr}\boldsymbol{P}_n - 2\mathrm{tr}\left[\left(\boldsymbol{P}_m^{1/2}\boldsymbol{P}_n\boldsymbol{P}_m^{1/2}\right)^{1/2}\right]} \tag{2.10}$$

Here, note that the mapping $\tilde{\boldsymbol{P}}_m = \boldsymbol{P}_m^{1/2}\boldsymbol{U}_m$ and $\tilde{\boldsymbol{P}}_n = \boldsymbol{P}_n^{1/2}\boldsymbol{U}_n$ may not be the only

mapping formula. This lifting process can also be carried out by first lifting $\boldsymbol{P}_m$ to a

fixed point $\tilde{\boldsymbol{P}}_m = \boldsymbol{P}_m^{1/2}$ and then lifting $\boldsymbol{P}_n$ to $\tilde{\boldsymbol{P}}_n = \boldsymbol{P}_n^{1/2}\boldsymbol{U}_n\boldsymbol{U}_m^H$. In this new mapping, by letting $\boldsymbol{U}_n$ and $\boldsymbol{U}_m$ still be the left and right singular vector matrices of $\boldsymbol{P}_n^{1/2}\boldsymbol{P}_m^{1/2}$, the $d_{\mathrm{R}_1}(\boldsymbol{P}_m, \boldsymbol{P}_n)$ between $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$ can be written as

$$
\begin{aligned}
d_{\mathrm{R}_1}^2(\boldsymbol{P}_m, \boldsymbol{P}_n) &= \sqrt{\left\| \boldsymbol{P}_m^{1/2} - \boldsymbol{P}_n^{1/2}\boldsymbol{U}_n\boldsymbol{U}_m^H \right\|^2} \\
&= \sqrt{\mathrm{tr}\boldsymbol{P}_m + \mathrm{tr}\boldsymbol{P}_n - 2\Re\left[\mathrm{tr}\left(\boldsymbol{U}_m\boldsymbol{U}_n^H\boldsymbol{P}_n^{1/2}\boldsymbol{P}_m^{1/2}\right)\right]} \\
&= \sqrt{\mathrm{tr}\boldsymbol{P}_m + \mathrm{tr}\boldsymbol{P}_n - 2\mathrm{tr}\left[\left(\boldsymbol{P}_m^{1/2}\boldsymbol{P}_n\boldsymbol{P}_m^{1/2}\right)^{1/2}\right]}
\end{aligned} \tag{2.11}
$$

which is exactly the same as Eq. (2.10). That means the geodesics between $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$ are equivalent based on these two mapping formulas. However, in the first mapping, $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$ are lifted to the isometric Euclidean space using $\boldsymbol{U}_m$ and $\boldsymbol{U}_n$ as the unitary matrices; whereas in the second mapping, we use $\boldsymbol{I}_M$ and $\boldsymbol{U}_n\boldsymbol{U}_m^H$ as the unitary matrices to lift $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$ to another isometric Euclidean space. These two spaces are different, which reveals that the isometric Euclidean space is not unique.

2.2.2 Riemannian Distance d_{R_2}

Now let us consider another mapping π:

$$
\boldsymbol{P} = \tilde{\boldsymbol{P}}^2, \quad \text{i.e.,} \quad \tilde{\boldsymbol{P}} = \boldsymbol{P}^{1/2} \tag{2.12}
$$

instead of choosing $\boldsymbol{U}_m$ and $\boldsymbol{U}_n$ to be the left and right singular vector matrices of $\boldsymbol{P}_n^{1/2}\boldsymbol{P}_m^{1/2}$, we choose $\boldsymbol{U}_m$ and $\boldsymbol{U}_n$ to be identity matrices. Then together with a suitable Riemannian metric $g_{\boldsymbol{P}}(\boldsymbol{A}, \boldsymbol{B}) = \langle \boldsymbol{A}, \boldsymbol{K} \rangle$ with $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{T}_{\mathcal{M}}(\boldsymbol{P})$ and $\boldsymbol{P}\boldsymbol{K} + $

$\boldsymbol{KP} + 2\tilde{\boldsymbol{P}}\boldsymbol{K}\tilde{\boldsymbol{P}} = \boldsymbol{B}$, the RD d_{R_2} between $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$ on $\mathcal{M}$ can be found to be

$$
\begin{aligned}
d_{\mathrm{R}_2}(\boldsymbol{P}_m, \boldsymbol{P}_n) &= \left\| \boldsymbol{P}_m^{1/2} - \boldsymbol{P}_n^{1/2} \right\| \\
&= \sqrt{\mathrm{tr}\boldsymbol{P}_m + \mathrm{tr}\boldsymbol{P}_n - 2\mathrm{tr}\left(\boldsymbol{P}_m^{1/2}\boldsymbol{P}_n^{1/2} \right)}
\end{aligned} \tag{2.13}
$$

2.2.3   Riemannian Distance d_{R_3}

RD d_{R_3} is given based on the logarithm mapping π such that

$$
\boldsymbol{P} = \exp \tilde{\boldsymbol{P}}, \quad \text{i.e.,} \quad \tilde{\boldsymbol{P}} = \log(\boldsymbol{P}) \tag{2.14}
$$

By choosing the Riemannian metric $g_{\boldsymbol{P}}(\boldsymbol{A}, \boldsymbol{B}) = \mathrm{tr}(\boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{P}^{-1}\boldsymbol{B})$ with $\boldsymbol{A}, \boldsymbol{B} \in \mathcal{T}_{\mathcal{M}}(\boldsymbol{P})$, the RD d_{R_3} between $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$ is shown to be

$$
d_{\mathrm{R}_3}(\boldsymbol{P}_m, \boldsymbol{P}_n) = \sqrt{\mathrm{tr}\left[\left(\log \boldsymbol{P}_m^{-1/2}\boldsymbol{P}_n\boldsymbol{P}_m^{-1/2} \right)^2 \right]} = \sqrt{\sum_{i=1}^{M} \log^2 \lambda_i} \tag{2.15}
$$

where λ_i are the eigenvalues of $\boldsymbol{P}_m^{-1}\boldsymbol{P}_n$ and $\boldsymbol{P}_m$ is invertible.

All three RD satisfy the distance axiom, i.e., (i) positivity, (ii) symmetry, (iii) triangle inequality. For d_{R_3}, which can be established in several different ways, has been in use for a long time in physics and mathematics, especially in General Relativity Theory [16], [17], [18]. In signal processing, d_{R_3} has been studied by various researchers for statistical operations and applied to interpolation, filtering and restoration of PSD matrices [19], [20]. In addition, different classification algorithms have been put forward according to d_{R_3}, and have been employed in the detection of pedestrians, MRI and EEG classifications [21], [22]. On the other hand, d_{R_1} and d_{R_2}

are newly developed [8] and have not been widely used yet. However, since it is much easier to manipulate in mathmatics, d_{R_2} has been employed in robust beamforming and signal detection recently with very encouraging results [23], [24], [25].

The concepts and relation between the PSD manifold and Euclidean space are quite essential because in order to carry out the processing which requires the optimization of the geodesics, the matrices on the manifold can be lifted to the corresponding isometric Euclidean subspace where the optimization can be performed in equivalent measurement of ED. The optimized result can then be projected back to the manifold. The process of lifting, optimization and projection can be operated iteratively until the true optimum solution is reached. For d_{R_2}, since the mapping π does not involve any free unitary matrix, thus there is only one iteration of lifting, optimization and projection; whereas for d_{R_1}, because the unitary matrix is not fixed when the optimization process is carried out, it may need several iterations to obtain the optimum result. The application of these concepts to the optimum processing of PSD matrices on the manifold will be demonstrated in the next chapter.

2.3   Weighting of Riemannian Distances

Generally speaking, applying weighting to features is a simple and effective way to enhance their similarities and dissimilarities in signal processing. Thus, the feature PSD matrices on $\mathcal{M}$ can be emphasized or de-emphasized by the weighting of RD. In order to do that, a positive definite Hermitian weighting matrix $\boldsymbol{W}$ can be applied to the PSD feature matrices such that we write $\boldsymbol{W} = \boldsymbol{\Omega}\boldsymbol{\Omega}^H$, where $\boldsymbol{\Omega}$ is $M \times K$, $K \leq M$. Then the weighted versions of $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$ can be defined as $\boldsymbol{P}_{m\mathrm{W}} = \boldsymbol{\Omega}^H \boldsymbol{P}_m \boldsymbol{\Omega}$ and $\boldsymbol{P}_{n\mathrm{W}} = \boldsymbol{\Omega}^H \boldsymbol{P}_n \boldsymbol{\Omega}$, respectively. It is easy to see that $\boldsymbol{P}_{m\mathrm{W}}$ and $\boldsymbol{P}_{n\mathrm{W}}$ are also positive

semi-definite Hermitian matrices on the manifold. The distance between two weighted PSD matrices then results in a weighted RD. The three weigthed RD between $\boldsymbol{P}_m$ and $\boldsymbol{P}_n \in \mathcal{M}$ corresponding to d_{R_1}, d_{R_2} and d_{R_3} are respectively given by

$$d_{\mathrm{WR}_1}(\boldsymbol{P}_m, \boldsymbol{P}_n) = \sqrt{F_1(\boldsymbol{W}, \boldsymbol{P}_m, \boldsymbol{P}_n)} \tag{2.16}$$

$$d_{\mathrm{WR}_2}(\boldsymbol{P}_m, \boldsymbol{P}_n) = \sqrt{F_2(\boldsymbol{W}, \boldsymbol{P}_m, \boldsymbol{P}_n)} \tag{2.17}$$

$$d_{\mathrm{WR}_3}(\boldsymbol{P}_m, \boldsymbol{P}_n) = d_{R_3}(\boldsymbol{P}_{m\mathrm{W}}, \boldsymbol{P}_{n\mathrm{W}}) \tag{2.18}$$

For F_1, $\boldsymbol{P}_{m\mathrm{W}} = \tilde{\boldsymbol{P}}_{m\mathrm{W}} \tilde{\boldsymbol{P}}_{m\mathrm{W}}^H$ with $\tilde{\boldsymbol{P}}_{m\mathrm{W}_1} = \boldsymbol{\Omega}^H \boldsymbol{P}_m \boldsymbol{U}_m$, $\boldsymbol{U}_m$ being a unitary matrix, so as $\boldsymbol{P}_{n\mathrm{W}}$. Then $\tilde{\boldsymbol{P}}_{m\mathrm{W}_1}$ and $\tilde{\boldsymbol{P}}_{n\mathrm{W}_1}$ are in the Euclidean subspace which is isometric with the tangent space of $\mathcal{M}$. Therefore, we can use the idea similar to Eq. (2.6) to find the minimum distance in Euclidean subspace. Then we have

$$\begin{aligned} F_1 &= \min_{\boldsymbol{U}_m, \boldsymbol{U}_n} \left\| \tilde{\boldsymbol{P}}_{m\mathrm{W}_1} - \tilde{\boldsymbol{P}}_{n\mathrm{W}_1} \right\|^2 \\ &= \min_{\boldsymbol{U}_m, \boldsymbol{U}_n} \mathrm{tr}\boldsymbol{P}_{m\mathrm{W}} + \mathrm{tr}\boldsymbol{P}_{n\mathrm{W}} - 2\Re\left[\mathrm{tr}\left(\boldsymbol{U}_m \boldsymbol{U}_n^H \boldsymbol{P}_n^{1/2} \boldsymbol{\Omega}\boldsymbol{\Omega}^H \boldsymbol{P}_m^{1/2} \right) \right] \end{aligned} \tag{2.19}$$

Again, the last term in Eq. (2.19) is maximized when $\boldsymbol{U}_n$ and $\boldsymbol{U}_m$ are chosen to be the left and right singular vector matrices of $\boldsymbol{P}_n^{1/2} \boldsymbol{\Omega}\boldsymbol{\Omega}^H \boldsymbol{P}_m^{1/2}$. Hence, the expression of F_1 is given by

$$F_1 = \mathrm{tr}(\boldsymbol{W}\boldsymbol{P}_m) + \mathrm{tr}(\boldsymbol{W}\boldsymbol{P}_n) - 2\mathrm{tr}\left[\left(\boldsymbol{P}_m^{1/2} \boldsymbol{W}\boldsymbol{P}_n \boldsymbol{W}\boldsymbol{P}_m^{1/2} \right)^{1/2} \right] \tag{2.20}$$

In the case of F_2, we have $\tilde{\boldsymbol{P}}_{i\mathrm{W}_2} = \boldsymbol{P}_{i\mathrm{W}}^{1/2} = \left(\boldsymbol{\Omega}^H \boldsymbol{P}_i \boldsymbol{\Omega}\right)^{1/2}$ for $i = m, n$. Following the same concept development with Eq. (2.13), we have

$$
\begin{aligned}
F_2 &= \left\| \tilde{\boldsymbol{P}}_{m\mathrm{W}_2} - \tilde{\boldsymbol{P}}_{n\mathrm{W}_2} \right\|^2 = \mathrm{tr}\boldsymbol{P}_{m\mathrm{W}} + \mathrm{tr}\boldsymbol{P}_{n\mathrm{W}} - 2\mathrm{tr}\left(\boldsymbol{P}_{m\mathrm{W}}^{1/2} \boldsymbol{P}_{n\mathrm{W}}^{1/2} \right) \\
&= \mathrm{tr}(\boldsymbol{W}\boldsymbol{P}_m) + \mathrm{tr}(\boldsymbol{W}\boldsymbol{P}_n) - 2\mathrm{tr}\left[\left(\boldsymbol{\Omega}^H \boldsymbol{P}_m \boldsymbol{\Omega}\right)^{1/2} \left(\boldsymbol{\Omega}^H \boldsymbol{P}_n \boldsymbol{\Omega}\right)^{1/2} \right]
\end{aligned}
\tag{2.21}
$$

Unlike d_{R_1} and d_{R_2}, Eq. (2.18) revealed that d_{R_3} is weight-invariant [8]. As we mentioned before, the weighting matrix is usually chosen or designed according to prior information and depending on the application, which will be demonstrated in details in chapter 4. Due to the ineffectivity of the optimum weighting for d_{R_3}, the use of d_{R_3} in signal processing may not be able to take full advantage when prior information is available. For this reason, the focus of our attention in this thesis is on the study and applications of d_{R_1} and d_{R_2}.

Chapter 3

Mean and Median of PSD

Matrices on Manifold

Signal processing often involves the evaluation of signal features for extraction of information. The mean and the median are two fundamental statistics used in signal processing to represent centrality of data points. The arithmetic mean or simply the *mean* of a group of random entities is defined as the centre from which the sum of squared distances to all members is minimum. For a finite set of real scalars, $\{x_1, x_2, \cdots, x_N\}$, the mean can be obtained by taking the derivative of the sum of squared distances from all members and set the result to zero, i.e., $\frac{d}{dx}(x - x_n)^2 = 0$ and $\bar{x} = \frac{1}{N} \sum_n x_n$. The *median* is defined as: the value of the variate which divides the total frequency into two equal halves. An important geometric property of the median is that it minimizes the sum of the absolute distances to all the points [10]. Compared to the mean, the median is a more robust estimate of the "central point" of a group of numbers, being less affected by outliers. This property is of vital significance in signal processing to reduce the interference, which allows median to be applied in

many applications, especially the detection system.

For a group of $M \times M$ PSD matrices, $\{\boldsymbol{P}_n, n = 1, \cdots, N\}$, according to the definition above, the concept of mean and median can be generalized by using their geometric properties. Thus, we have

$$\boldsymbol{C}_x = \arg \min_{\boldsymbol{C}} \sum_{n=1}^{N} d^2(\boldsymbol{P}_n, \boldsymbol{C}) \tag{3.1}$$

$$\boldsymbol{\Gamma}_x = \arg \min_{\boldsymbol{\Gamma}} \sum_{n=1}^{N} d(\boldsymbol{P}_n, \boldsymbol{\Gamma}) \tag{3.2}$$

where d, the distance measured between two matrices, is a general metric. For the particular case in which d is considered to be ED d_{E}, we will call the corresponding central points *Euclidean mean* (EMn) and *Euclidean median* (EMd), denoted by $\boldsymbol{C}_{\mathrm{E}}$ and $\boldsymbol{\Gamma}_{\mathrm{E}}$ respectively. Likewise, if d is taken to be the various RD d_{R}, then the results are respectively called the *Riemannian mean* (RMn), denoted by $\boldsymbol{C}_{\mathrm{R}}$, and the *Riemannian median* (RMd), denoted by $\boldsymbol{\Gamma}_{\mathrm{R}}$ in this paper. Notice that different RD yields different RMn and RMd.

Since the EMn and EMd are obtained based on ED, they are relatively straight-forward to understand and facilitate the evaluation of signal features. Recently, the RMn and RMd of PSD matrices have been studied and important contributions for their evaluations have been made. For example, In [12], based on steepest descent, an algorithm was proposed and applied to the exponential mapping, thus iteratively locating RMd according to d_{R_3}. In [18], it has been proved that the RMn based on the measure d_{R_3} was the solution of $\sum_n \log \left(\boldsymbol{X}^{1/2} \boldsymbol{P}_n^{-1} \boldsymbol{X}^{1/2} \right) = 0$. In [26], each data vector formed its outer product which was assumed to be strictly positive definite and Toeplitz. After that, deterministic and stochastic algorithms for computing the RMn

and RMd based on the exponential mapping (d_{R_3}) of these matrices were developed. Other research such as [27] linearized the log function to find the RMn by applying a descent algorithm. However, it can be seen that invariably, all the studies and algorithms to evaluate RMn and RMd so far are based on the distance measure of d_{R_3}. In this chapter, we will develop algorithms to find the RMn and RMd according to the distance measures of d_{R_1} and d_{R_2} for a group of PSD matrices. Throughout the entire thesis, we assume that all PSD matrices $\boldsymbol{P}_n$ lie in a convex set $\mathcal{C} \subset \mathcal{M}$. That is to say, there is a unique geodesic completely lying in $\mathcal{C}$ between any two matrices in $\mathcal{C}$.

3.1 Euclidean Mean

The conventional mean that is commonly used is the arithmetic mean or Euclidean mean which is defined related to the ED or Frobenius norm. Based on the definition of mean in Eq. (3.1), we can formulate the EMn as follows,

$$\boldsymbol{C}_{\mathrm{E}} = \arg\min_{\boldsymbol{C}} g_{\mathrm{E}} = \arg\min_{\boldsymbol{C}} \sum_{n=1}^{N} d_{\mathrm{E}}^2(\boldsymbol{P}_n, \boldsymbol{C}) \tag{3.3}$$

where

$$\begin{aligned}
d_{\mathrm{E}}^2(\boldsymbol{P}_n, \boldsymbol{C}) &= \|\boldsymbol{P}_n - \boldsymbol{C}\|_2^2 \\
&= \operatorname{tr}\left[(\boldsymbol{P}_n - \boldsymbol{C})(\boldsymbol{P}_n - \boldsymbol{C})^H\right] \\
&= \operatorname{tr}\left[\boldsymbol{P}_n\boldsymbol{P}_n^H + \boldsymbol{C}\boldsymbol{C}^H - \boldsymbol{P}_n\boldsymbol{C}^H - \boldsymbol{C}\boldsymbol{P}_n^H\right] \\
&= \operatorname{tr}(\boldsymbol{P}_n\boldsymbol{P}_n^H) + \operatorname{tr}(\boldsymbol{C}\boldsymbol{C}^H) - 2\operatorname{tr}(\boldsymbol{P}_n\boldsymbol{C}^H) \tag{3.4}
\end{aligned}$$

Noting that g_E is quadratic, the solution to Eq. (3.3) can be obtained by differentiating g_E with respect to C and equating the result to zero. That is,

$$\frac{d}{dC} \left[\sum_{n=1}^{N} d_E^2(P_n, C) \right] = 0 \tag{3.5}$$

Equivalently,

$$\sum_{n=1}^{N} 2 \left(\mathrm{tr} C - 2 P_n \right) = 2NC - 2 \sum_{n=1}^{N} P_n = 0 \tag{3.6}$$

Hence, the minimum value of g_E is achieved by Eq. (3.6) for which

$$C_E = \frac{1}{N} \sum_{n=1}^{N} P_n \tag{3.7}$$

The expression of EMn is quite transparent to understand because it is in accordance with the scalar case $\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n$. However, as we previously indicated, since the ED is not the accurate distance measure for PSD matrices on $\mathcal{M}$, the EMn, though simple to achieve, may not be the proper centre to evaluate the feature matrices. Therefore, it is necessary to develop algorithms for finding the Riemannian means with respect to d_{R_1} and d_{R_2}.

3.2   Riemannian Mean

The problem of locating RMn on $\mathcal{M}$, according to the definition in Eq. (3.1), is to search for a matrix $C \in \mathcal{M}$ which has the minimum sum squared Riemannian distances to all the PSD matrices. In this section, we will analyze in turn, each of the initial formulation of Riemannian means according to d_{R_1}, d_{R_2} and their weighted

versions, and then develop the corresponding algorithms to locate these central points.

3.2.1 Riemannian Mean According to d_{R_1}

The problem can be stated as: Given the Riemannian distance d_{R_1} between two positive semi-definite matrices $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$ as in Eq. (2.10), find RMn$_1$, the Riemannian mean of $\{\boldsymbol{P}_n, n = 1, \cdots, N\}$ according to d_{R_1}, denoted by $\boldsymbol{C}_{R_1}$, such that

$$\boldsymbol{C}_{R_1} = \arg\min_{\boldsymbol{C}} g_{R_1} = \arg\min_{\boldsymbol{C}} \sum_{n=1}^{N} d_{R_1}^2(\boldsymbol{P}_n, \boldsymbol{C}) \tag{3.8}$$

In order to solve this problem, the following facts are the preparations for us to develop an algorithm to locate RMn$_1$.

F1: As we described in chapter 2, to lift the geodesic between $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$ from $\mathcal{M}$ to $\mathcal{U}_{\mathcal{H}}$, we first lift $\boldsymbol{P}_m$ to $\tilde{\boldsymbol{P}}_m = \boldsymbol{P}_m^{1/2}$ and then lift $\tilde{\boldsymbol{P}}_n = \boldsymbol{P}_n^{1/2}\boldsymbol{U}_n\boldsymbol{U}_m^H$ with $\boldsymbol{U}_n$ and $\boldsymbol{U}_m$ being the left and right singular vector matrices of $\boldsymbol{P}_n^{1/2}\boldsymbol{P}_m^{1/2}$.

F2: The isometry between $\mathcal{T}_{\mathcal{M}}$ and $\mathcal{U}_{\mathcal{H}}$ gives the equation $d_{R_1}^2(\boldsymbol{P}_m, \boldsymbol{P}_n) = d_E^2(\tilde{\boldsymbol{P}}_m, \tilde{\boldsymbol{P}}_n)$, therefore, we can equivalently perform the optimization process in $\mathcal{U}_{\mathcal{H}}$ in terms of the Euclidean distance.

F3: For a set of $M \times M$ matrices $\{\tilde{\boldsymbol{P}}_n \in \mathcal{H}, n = 1, \cdots, N\}$, the sample average $\tilde{\boldsymbol{C}} = \frac{1}{N}\sum_{n=1}^{N}\tilde{\boldsymbol{P}}_n$ minimizes the sum of squared ED: $d_E(\tilde{\boldsymbol{P}}_n, \tilde{\boldsymbol{C}}) = \sum_{n=1}^{N}\|\tilde{\boldsymbol{P}}_n - \tilde{\boldsymbol{C}}\|^2$. This has been proved in the previous section.

Based on the three facts above, we can come up with an algorithm to locate RMn$_1$. The main idea is, at first, roughly find a central point $\boldsymbol{C}^{(i)}$, say EMn, on the manifold as the initial point. Then, based on the mapping as described in F1, lift the geodesics

from $\boldsymbol{C}^{(i)}$ to all the PSD matrices $\boldsymbol{P}_n$ to the respective Euclidean subspaces with $\boldsymbol{C}^{(i)1/2}$ as a common point. Now we can carry out the optimization process in the Euclidean subspaces. Find the first optimum centre point in the Euclidean space using the method in F3. Then project this first optimum point back to the manifold and use this point as the next starting central point. Reiterate this lifting, optimization and projection process until the distance between the optimum central point in the final step and last final step is shorter than a preset precision. This process has been summarized in the following algorithm.

Algorithm RMn1:

1. For $i = 0$: Initialize $\boldsymbol{C}^{(i)} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{P}_n$, set a positive precision ϵ.

2. Set: $\tilde{\boldsymbol{C}}^{(i)} = \left(\boldsymbol{C}^{(i)}\right)^{1/2}$.

3. Find $\hat{\boldsymbol{V}}_n^{(i)}$ which minimizes $\left\| \boldsymbol{P}_n^{1/2} \boldsymbol{V}_n^{(i)} - \tilde{\boldsymbol{C}}^{(i)} \right\|_2^2, \forall n$:

   $\hat{\boldsymbol{V}}_n^{(i)} = \boldsymbol{U}_l^{(i)} \boldsymbol{U}_r^{(i)H}$, where $\boldsymbol{U}_l^{(i)} \boldsymbol{\Sigma} \boldsymbol{U}_r^{(i)H} = \boldsymbol{P}_n^{1/2} \tilde{\boldsymbol{C}}^{(i)}$ is the SVD of $\boldsymbol{P}_n^{1/2} \tilde{\boldsymbol{C}}^{(i)}$.

4. Lift to $\mathcal{U}_{\mathcal{H}}$: $\tilde{\boldsymbol{P}}_n^{(i)} = \boldsymbol{P}_n^{1/2} \hat{\boldsymbol{V}}_n^{(i)}$.

5. Get the new arithmetic mean in $\mathcal{H}$: $\tilde{\boldsymbol{C}}'^{(i)} = \frac{1}{N} \sum_{n=1}^{N} \tilde{\boldsymbol{P}}_n^{(i)}$.

6. Update the Riemannian mean on $\mathcal{M}$: $\boldsymbol{C}^{(i+1)} = \tilde{\boldsymbol{C}}'^{(i)} \tilde{\boldsymbol{C}}'^{(i)H}$.

7. Calculate the precision: $h = d_{\mathrm{R}_1} \left(\boldsymbol{C}^{(i+1)}, \boldsymbol{C}^{(i)} \right)$.

8. If $h > \epsilon$, let $i \rightarrow i + 1$ and go back to Step 2. Otherwise, obtain the RMn_1
   $\boldsymbol{C}_{\mathrm{R}_1} = \boldsymbol{C}^{(i+1)}$ and exit. ∎

The algorithm above is established on the basis of its convergence, which is presented in the following theorem:

Theorem 1. *For the PSD matrices $\{\boldsymbol{P}_n\}$, the Riemannian mean according to d_{R_1} can be located by Algorithm RMn1 such that*

$$\boldsymbol{C}_{\mathrm{R}_1} = \lim_{i \to \infty} \mathbf{C}^{(i)} \tag{3.9}$$

where $\boldsymbol{C}_{\mathrm{R}_1} = \arg\min_{C} \sum_{n=1}^{N} d_{\mathrm{R}_1}^2(\boldsymbol{P}_n, \boldsymbol{C})$ is the RMn_1. □

Proof. Proof of Theorem 1 is shown in Appendix A.1. ∎

Basically, the proof of Theorem 1 uses F1 and F2 to find the isometric Euclidean subspace of $\mathcal{M}$ and then, a *contraction* $g_{\mathrm{E}}^{(i+1)} \leq g_{\mathrm{E}}^{(i)}$ is established according to F3, which results in the convergence of the process. Isometry between $\mathcal{U}_{\mathcal{H}}$ and $\mathcal{T}_{\mathcal{M}}$ promises that arriving at the optimum point in $\mathcal{U}_{\mathcal{H}}$ means achieving the RMn_1 on $\mathcal{M}$. Likewise, the same idea is applied in the proofs of convergence for other algorithms.

3.2.2   Riemannian Mean According to d_{WR_1}

In this subsection, we investigate the algorithm for finding the RMn according to the weighted RD d_{WR_1}. Now let us consider the weighting of the PSD matrices by a positive definite Hermitian matrix $\boldsymbol{W} = \boldsymbol{\Omega}\boldsymbol{\Omega}^H$ as described in chapter 2 such that the weighted version of $\boldsymbol{P}_n$ can be written as $\boldsymbol{P}_{n\mathrm{W}} = \boldsymbol{\Omega}^H \boldsymbol{P}_n \boldsymbol{\Omega}$. Then $\boldsymbol{C}_{\mathrm{W}} = \boldsymbol{\Omega}^H \boldsymbol{C} \boldsymbol{\Omega}$ can be regarded as the weighted version of the central point $\boldsymbol{C}$. Since $\boldsymbol{P}_n, \boldsymbol{C} \in \mathcal{M}$, it is easy to see that $\boldsymbol{P}_{n\mathrm{W}}, \boldsymbol{C}_{\mathrm{W}} \in \mathcal{M}$. Again, in order to make it more manipulable, we use the mapping $\boldsymbol{P} = \tilde{\boldsymbol{P}}\tilde{\boldsymbol{P}}^H$ to lift all $\boldsymbol{P}_{n\mathrm{W}}$ and $\boldsymbol{C}_{\mathrm{W}}$ to the isometric Euclidean space such that $\tilde{\boldsymbol{P}}_{n\mathrm{W}_1} = \boldsymbol{\Omega}^H \boldsymbol{P}_{n\mathrm{W}}^{1/2} \boldsymbol{U}_n$ and $\tilde{\boldsymbol{C}}_{\mathrm{W}_1} = \boldsymbol{\Omega}^H \boldsymbol{C}_{\mathrm{W}}^{1/2} \boldsymbol{U}_c$ with $\boldsymbol{U}_n$ and $\boldsymbol{U}_c$ being unitary matrices. Then applying the expression of d_{WR_1} in Eq. (2.16), the problem of finding

the RMn according to d_{WR_1} can be formulated as

$$\boldsymbol{C}_{\mathrm{WR}_1} = \arg\min_{\boldsymbol{C}} g_{\mathrm{WR}_1} = \arg\min_{\boldsymbol{C}} \sum_{n=1}^{N} d^2_{\mathrm{WR}_1}(\boldsymbol{P}_n, \boldsymbol{C}) \qquad (3.10)$$

where

$$g_{\mathrm{WR}_1} = \sum_{n=1}^{N} \mathrm{tr}(\tilde{\boldsymbol{P}}_{n\mathrm{W}_1}\tilde{\boldsymbol{P}}^H_{n\mathrm{W}_1}) + \mathrm{tr}(\tilde{\boldsymbol{C}}_{\mathrm{W}_1}\tilde{\boldsymbol{C}}^H_{\mathrm{W}_1}) - 2\mathrm{tr}\left[\left(\boldsymbol{C}^{1/2}\boldsymbol{W}\boldsymbol{P}_n\boldsymbol{W}\boldsymbol{C}^{1/2}\right)^{1/2}\right] \qquad (3.11)$$

However, *Alogrithm RMn1* cannot be directly applied to the minimization problem in Eq. (3.10) because the expression on the right hand side of Eq. (3.11) cannot be formed into $g_{\mathrm{WE}} = \sum_n \|\tilde{\boldsymbol{P}}_{n\mathrm{W}_1} - \tilde{\boldsymbol{C}}_{\mathrm{W}_1}\|^2$ and consequently, it cannot be optimized in the Euclidean subspace in this case. On the other hand, we can find the upper bound of g_{WE} using the following lemma:

Lemma 1. *The last term of Eq. (3.11) has a lower bound which is given by*

$$\mathrm{tr}\left[\left(\boldsymbol{C}^{1/2}\boldsymbol{W}\boldsymbol{P}_n\boldsymbol{W}\boldsymbol{C}^{1/2}\right)^{1/2}\right] \geq \Re\left[\mathrm{tr}\left(\tilde{\boldsymbol{P}}^H_{n\mathrm{W}_1}\tilde{\boldsymbol{C}}_{\mathrm{W}_1}\right)\right] \qquad (3.12)$$

$\square$

Proof. Referring to the derivation of d_{WR_1} in Eq. (2.19), $\Re\left[\mathrm{tr}\left(\boldsymbol{U}^H_n\boldsymbol{P}^{1/2}_n\boldsymbol{\Omega}\boldsymbol{\Omega}^H\boldsymbol{C}^{1/2}\boldsymbol{U}_c\right)\right]$ is maximized when $\boldsymbol{U}_n$ and $\boldsymbol{U}_c$ are the left and right singular vector matrices of $\boldsymbol{P}^{1/2}_n\boldsymbol{\Omega}\boldsymbol{\Omega}^H\boldsymbol{C}^{1/2}$ respectively [13], which is similar to the maximization of the last term in Eq. (2.19), therefore,

$$\mathrm{tr}\left[\left(\boldsymbol{C}^{1/2}\boldsymbol{W}\boldsymbol{P}_n\boldsymbol{W}\boldsymbol{C}^{1/2}\right)^{1/2}\right] \geq \Re\left[\mathrm{tr}\left(\boldsymbol{U}^H_n\boldsymbol{P}^{1/2}_n\boldsymbol{\Omega}\boldsymbol{\Omega}^H\boldsymbol{C}^{1/2}\boldsymbol{U}_c\right)\right]$$
$$= \Re\left[\mathrm{tr}\left(\tilde{\boldsymbol{P}}^H_{n\mathrm{W}_1}\tilde{\boldsymbol{C}}_{\mathrm{W}_1}\right)\right] \qquad (3.13)$$

$\blacksquare$

Substituting this lower bound into Eq. (3.11), we have

$$
\begin{aligned}
g_{\mathrm{WR}_1} &= \sum_{n=1}^{N} d_{\mathrm{WR}_1}^2(\boldsymbol{P}_n, \boldsymbol{C}) \\
&\leq \sum_{n=1}^{N} \mathrm{tr}(\tilde{\boldsymbol{P}}_{n\mathrm{W}_1} \tilde{\boldsymbol{P}}_{n\mathrm{W}_1}^H) + \mathrm{tr}(\tilde{\boldsymbol{C}}_{\mathrm{W}_1} \tilde{\boldsymbol{C}}_{\mathrm{W}_1}^H) - 2\Re\left[\mathrm{tr}\left(\tilde{\boldsymbol{P}}_{n\mathrm{W}_1}^H \tilde{\boldsymbol{C}}_{\mathrm{W}_1}\right)\right] \\
&= \sum_{n=1}^{N} \left\|\tilde{\boldsymbol{P}}_{n\mathrm{W}_1} - \tilde{\boldsymbol{C}}_{\mathrm{W}_1}\right\|^2 = g_{\mathrm{WE}}
\end{aligned}
\tag{3.14}
$$

It can be noticed that the upper bound in Eq. (3.14) is in the right form to be minimized in the Euclidean subspace. Hence, the *Algorithm RMn1* can be used to minimize this upper bound and obtain the weighted RMn according to d_{WR_1} (WRMn$_1$):

Corollary 1. *For the weighting* $\boldsymbol{W} = \boldsymbol{\Omega}\boldsymbol{\Omega}^H$, *the weighted Riemannian mean according to the weighted RD* d_{WR_1} *can be obtained by Algorithm RMn1 such that*

$$
\boldsymbol{C}_{\mathrm{WR}_1} = \lim_{i \to \infty} \boldsymbol{C}_{\mathrm{W}}^{(i)}
\tag{3.15}
$$

where $\boldsymbol{C}_{\mathrm{WR}_1}$ *is the WRMn$_1$ and* $\boldsymbol{C}_{\mathrm{W}}^{(\infty)}$ *is the final solution of Algorithm RMn1 which is applied on the weighted PSD matrices* $\boldsymbol{P}_{n\mathrm{W}}$. $\qquad \square$

3.2.3   Riemannian Mean According to d_{R_2}

The Riemannian mean of $\{\boldsymbol{P}_n, n = 1, \cdots, N\}$ according to d_{R_2}, abbreviated as RMn_2, is defined as

$$\boldsymbol{C}_{\mathrm{R}_2} = \arg\min_{\boldsymbol{C}} g_{\mathrm{R}_2} = \arg\min_{\boldsymbol{C}} \sum_{n=1}^{N} d_{\mathrm{R}_2}^2(\boldsymbol{P}_n, \boldsymbol{C}) \tag{3.16}$$

In the case of d_{R_2}, the lifting of $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$ from the manifold to the Euclidean space is by fixing the unitary matrices $\boldsymbol{U}_m = \boldsymbol{U}_n = \boldsymbol{I}_M$, i.e., $\tilde{\boldsymbol{P}}_m = \boldsymbol{P}_m^{1/2}$ and $\tilde{\boldsymbol{P}}_n = \boldsymbol{P}_n^{1/2}$. Thus, the algorithm for locating RMn_2 is a one-step procedure, which is given below.

Algorithm RMn2:

1. Lift to $\mathcal{U}_{\mathcal{H}}$: $\tilde{\boldsymbol{P}}_n = \boldsymbol{P}_n^{1/2}$.

2. Calculate the arithmetic mean in $\mathcal{U}_{\mathcal{H}}$: $\tilde{\boldsymbol{C}}_{\mathrm{R}_2} = \frac{1}{N} \sum_{n=1}^{N} \tilde{\boldsymbol{P}}_n$.

3. Back to $\mathcal{M}$: $\boldsymbol{C}_{\mathrm{R}_2} = \tilde{\boldsymbol{C}}_{\mathrm{R}_2}^2$. ∎

The procedure to locate RMn_2 can be summarized in the following theorem:

Theorem 2. *For the PSD matrices $\{\boldsymbol{P}_n\}$, the Riemannian mean according to d_{R_2} is given by*

$$\boldsymbol{C}_{\mathrm{R}_2} = \tilde{\boldsymbol{C}}_{\mathrm{R}_2}^2 \tag{3.17}$$

where $\tilde{\boldsymbol{C}}_{\mathrm{R}_2} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{P}_n^{1/2}$. □

Proof. The objective function in Eq. (3.16) can be written as

$$\sum_{n=1}^{N} d_{\mathrm{R}_2}^2(\boldsymbol{P}_n, \boldsymbol{C}) = \sum_{n=1}^{N} \left[\mathrm{tr}\boldsymbol{P}_n + \mathrm{tr}(\tilde{\boldsymbol{C}}\tilde{\boldsymbol{C}}) - 2\mathrm{tr}(\tilde{\boldsymbol{P}}_n\tilde{\boldsymbol{C}}) \right] \tag{3.18}$$

where $\tilde{\boldsymbol{P}}_n = \boldsymbol{P}_n^{1/2}$ and $\tilde{\boldsymbol{C}} = \boldsymbol{C}^{1/2}$. Differentiating Eq. (3.18) with respect to $\tilde{\boldsymbol{C}}$,

$$\frac{d}{d\tilde{\boldsymbol{C}}}\left[\sum_{n=1}^{N} d_{\mathrm{R}_2}^2(\boldsymbol{P}_n, \boldsymbol{C})\right] = \sum_{n=1}^{N}(2\tilde{\boldsymbol{C}} - 2\tilde{\boldsymbol{P}}_n) = 2N\tilde{\boldsymbol{C}} - 2\sum_{n=1}^{N}\tilde{\boldsymbol{P}}_n \qquad (3.19)$$

Equating Eq. (3.19) to zero, we have

$$\tilde{\boldsymbol{C}}_{\mathrm{R}_2} = \frac{1}{N}\sum_{n=1}^{N}\tilde{\boldsymbol{P}}_n \qquad (3.20)$$

Then the minimum value of the objective function can be achieved. Projecting $\tilde{\boldsymbol{C}}_{\mathrm{R}_2}$ back to the manifold, the Riemannian mean according to d_{R_2} is obtained such that $\boldsymbol{C}_{\mathrm{R}_2} = \tilde{\boldsymbol{C}}_{\mathrm{R}_2}^2$. ∎

Note that here, $\tilde{\boldsymbol{C}}_{\mathrm{R}_2}$ is still Hermitian, and *Algorithm RMn2* yields a closed-form solution for the Riemannian mean of a group of random PSD matrices according to the d_{R_2}. Since the objective function of Eq. (3.16) is proved to be convex in Appendix A.2.1, the RMn$_2$ located by our algorithm is the global optimum point.

3.2.4   Riemannian Mean According to d_{WR_2}

On the basis of locating RMn$_2$, the method of finding the weighted Riemannian mean according to d_{WR_2} (WRMn$_2$) can be established quite readily. Given the definition of d_{WR_2} in Eq. (2.17), for an $M \times M$ positive definite Hermitian matrix $\boldsymbol{W} = \boldsymbol{\Omega}\boldsymbol{\Omega}^H$, the WRMn$_2$ of $\{\boldsymbol{P}_n, n = 1, \cdots, N\}$, denoted by $\boldsymbol{C}_{\mathrm{WR}_2}$, can be found by

$$\boldsymbol{C}_{\mathrm{WR}_2} = \arg\min_{C} g_{\mathrm{WR}_2} = \arg\min_{C} \sum_{n=1}^{N} d_{\mathrm{WR}_2}^2(\boldsymbol{P}_n, \boldsymbol{C}) \qquad (3.21)$$

Again, we can write $\boldsymbol{C}_\mathrm{W} = \boldsymbol{\Omega}^H \boldsymbol{C} \boldsymbol{\Omega}$ and $\boldsymbol{P}_{n\mathrm{W}} = \boldsymbol{\Omega}^H \boldsymbol{P}_n \boldsymbol{\Omega}$. Using the mapping in Eq. (2.12) such that $\tilde{\boldsymbol{C}}_{\mathrm{W}_2} = \boldsymbol{C}_\mathrm{W}^{1/2}$ and $\tilde{\boldsymbol{P}}_{n\mathrm{W}_2} = \boldsymbol{P}_{n\mathrm{W}}^{1/2}$, we have

$$d_{\mathrm{WR}_2}^2(\boldsymbol{P}_n, \boldsymbol{C}) = \mathrm{tr}(\boldsymbol{P}_{n\mathrm{W}}) + \mathrm{tr}(\tilde{\boldsymbol{C}}_{\mathrm{W}_2}\tilde{\boldsymbol{C}}_{\mathrm{W}_2}) - 2\mathrm{tr}(\tilde{\boldsymbol{P}}_{n\mathrm{W}_2}\tilde{\boldsymbol{C}}_{\mathrm{W}_2}) \tag{3.22}$$

Putting Eq. (3.22) into Eq. (3.21) and then taking the derivative with respect to $\tilde{\boldsymbol{C}}_{\mathrm{W}_2}$, we have

$$\frac{d}{d\tilde{\boldsymbol{C}}_{\mathrm{W}_2}} \left[\sum_{n=1}^{N} d_{\mathrm{WR}_2}^2(\boldsymbol{P}_n, \boldsymbol{C}) \right] = \sum_{n=1}^{N} \left(2\tilde{\boldsymbol{C}}_{\mathrm{W}_2} - 2\tilde{\boldsymbol{P}}_{n\mathrm{W}_2} \right) \tag{3.23}$$

By setting Eq. (3.23) to zero, the optimum solution in Euclidean space is given by

$$\tilde{\boldsymbol{C}}_{\mathrm{WR}_2} = \frac{1}{N} \sum_{n=1}^{N} \tilde{\boldsymbol{P}}_{n\mathrm{W}_2} = \frac{1}{N} \sum_{n=1}^{N} \left(\boldsymbol{\Omega}^H \boldsymbol{P}_n \boldsymbol{\Omega} \right)^{1/2} \tag{3.24}$$

Then project the result back to the manifold such that

$$\boldsymbol{C}_{\mathrm{WR}_2} = \tilde{\boldsymbol{C}}_{\mathrm{WR}_2}\tilde{\boldsymbol{C}}_{\mathrm{WR}_2} = \frac{1}{N^2} \left[\sum_{n=1}^{N} \left(\boldsymbol{\Omega}^H \boldsymbol{P}_n \boldsymbol{\Omega} \right)^{1/2} \right]^2 \tag{3.25}$$

Corollary 2. *For the weighting $\boldsymbol{W} = \boldsymbol{\Omega}\boldsymbol{\Omega}^H$, the weighted Riemannian mean according to d_{WR_2} can be obtained by*

$$\boldsymbol{C}_{\mathrm{WR}_2} = \tilde{\boldsymbol{C}}_{\mathrm{WR}_2}^2 \tag{3.26}$$

where $\tilde{\boldsymbol{C}}_{\mathrm{WR}_2} = \frac{1}{N} \sum_{n=1}^{N} \left(\boldsymbol{\Omega}^H \boldsymbol{P}_n \boldsymbol{\Omega} \right)^{1/2}$. □

Here, note that $\boldsymbol{C}_{\mathrm{WR}_2}$ is also a positive definite Hermitian matrix on the manifold and we do not need any iteration to obtain WRMn$_2$.

3.3   Euclidean Median

Now let us examine the median of a group of $M \times M$ PSD matrices $\{\boldsymbol{P}_n, n = 1, 2, \cdots, N\}$. As described in the beginning of this chapter, the median is a generalization of the geometric property of the scalar median, which has the minimum sum of the distances to all the points. The median so defined is sometimes referred to as the *geometric median*. In particular, if the distance is ED d_{E}, the result is called the *Euclidean median*, denoted by $\boldsymbol{\Gamma}_{\mathrm{E}}$.

According to the definition of geometric median in Eq. (3.2), for ED, the Euclidean median (EMd) is given by

$$
\begin{aligned}
\boldsymbol{\Gamma}_{\mathrm{E}} = \arg\min_{\boldsymbol{\Gamma}} f_{\mathrm{E}} &= \arg\min_{\boldsymbol{\Gamma}} \sum_{n=1}^{N} d_{\mathrm{E}}(\boldsymbol{P}_n, \boldsymbol{\Gamma}) \\
&= \arg\min_{\boldsymbol{\Gamma}} \sum_{n=1}^{N} \sqrt{\operatorname{tr}\left[(\boldsymbol{\Gamma} - \boldsymbol{P}_n)(\boldsymbol{\Gamma} - \boldsymbol{P}_n)^H\right]}
\end{aligned}
\tag{3.27}
$$

The objective function of Eq. (3.27) is convex [28], which has been proved in Appendix A.2.2. Hence, a convergent algorithm based on the steepest descent has been proposed to achieved the global optimum point for Eq. (3.27).

Algorithm EMd:

1. For $i = 0$: Initialize $\boldsymbol{\Gamma}^{(i)} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{P}_n$, set a positive precision ϵ.

2. Evaluate $\boldsymbol{\Gamma}$ iteratively:

$$
\boldsymbol{\Gamma}^{(i+1)} = \boldsymbol{\Gamma}^{(i)} - \alpha \boldsymbol{G}^{(i)}
\tag{3.28}
$$

where $0 \leq \alpha \leq 2$ is the step size and

$$\boldsymbol{G}^{(i)} = \sum_{n \in \mathcal{N}^{(i)}} \frac{\boldsymbol{P}_n}{\|\boldsymbol{\Gamma}^{(i)} - \boldsymbol{P}_n\|} \left(\sum_{n \in \mathcal{N}^{(i)}} \frac{1}{\|\boldsymbol{\Gamma}^{(i)} - \boldsymbol{P}_n\|} \right)^{-1} \tag{3.29}$$

with $\mathcal{N}^{(i)} = \{n \in [1, N] : \boldsymbol{\Gamma}^{(i)} \neq \boldsymbol{P}_n\}$.

3. Calculate the precision: $h = \left\|\boldsymbol{\Gamma}^{(i+1)} - \boldsymbol{\Gamma}^{(i)}\right\|$.

4. If $h > \epsilon$, let $i \rightarrow i + 1$ and go back to Step 2. Otherwise, obtain the EMd
   $\boldsymbol{\Gamma}_{\mathrm{E}} = \boldsymbol{\Gamma}^{(i+1)}$ and exit. $\blacksquare$

The convergence of the algorithm above is given in [28]. Essentially speaking, Eq. (3.28) is the same as the result of using the gradient descent method. The gradient of f_{E} exists and is given by

$$\nabla f_{\mathrm{E}}(\boldsymbol{\Gamma}) = \sum_{n=1}^{N} \frac{\boldsymbol{\Gamma} - \boldsymbol{P}_n}{\|\boldsymbol{\Gamma} - \boldsymbol{P}_n\|} \tag{3.30}$$

Based on the gradient descent method, a natural choice for the search direction is the negative gradient [29]. Thus, the iterative process can be written as

$$\begin{aligned}
\boldsymbol{\Gamma}^{(i+1)} &= \boldsymbol{\Gamma}^{(i)} - \alpha \nabla f_{\mathrm{E}}(\boldsymbol{\Gamma}^{(i)}) \\
&= \boldsymbol{\Gamma}^{(i)} - \alpha \sum_{n=1}^{N} \frac{\boldsymbol{\Gamma}^{(i)} - \boldsymbol{P}_n}{\|\boldsymbol{\Gamma}^{(i)} - \boldsymbol{P}_n\|}
\end{aligned} \tag{3.31}$$

For Eq. (3.28), it can be derived as

$$
\begin{aligned}
\boldsymbol{\Gamma}^{(i+1)} &= \boldsymbol{\Gamma}^{(i)} - \alpha \frac{\sum_{n=1}^{N} \frac{\boldsymbol{P}_n}{\|\boldsymbol{\Gamma}^{(i)}-\boldsymbol{P}_n\|}}{\sum_{n=1}^{N} \frac{1}{\|\boldsymbol{\Gamma}^{(i)}-\boldsymbol{P}_n\|}} \\
&= \boldsymbol{\Gamma}^{(i)} - \alpha \frac{\sum_{n=1}^{N} \frac{\boldsymbol{P}_n}{\|\boldsymbol{\Gamma}^{(i)}-\boldsymbol{P}_n\|}}{\sum_{n=1}^{N} \frac{1}{\|\boldsymbol{\Gamma}^{(i)}-\boldsymbol{P}_n\|}} + \alpha \frac{\sum_{n=1}^{N} \frac{\boldsymbol{\Gamma}^{(i)}}{\|\boldsymbol{\Gamma}^{(i)}-\boldsymbol{P}_n\|}}{\sum_{n=1}^{N} \frac{1}{\|\boldsymbol{\Gamma}^{(i)}-\boldsymbol{P}_n\|}} - \alpha \frac{\sum_{n=1}^{N} \frac{\boldsymbol{\Gamma}^{(i)}}{\|\boldsymbol{\Gamma}^{(i)}-\boldsymbol{P}_n\|}}{\sum_{n=1}^{N} \frac{1}{\|\boldsymbol{\Gamma}^{(i)}-\boldsymbol{P}_n\|}} \\
&= \boldsymbol{\Gamma}^{(i)} + \alpha \frac{\sum_{n=1}^{N} \frac{\boldsymbol{\Gamma}^{(i)}-\boldsymbol{P}_n}{\|\boldsymbol{\Gamma}^{(i)}-\boldsymbol{P}_n\|}}{\sum_{n=1}^{N} \frac{1}{\|\boldsymbol{\Gamma}^{(i)}-\boldsymbol{P}_n\|}} - \alpha \boldsymbol{\Gamma}^{(i)}
\end{aligned}
\tag{3.32}
$$

Taking the last term of Eq. (3.32) to the left hand side, we have

$$
\boldsymbol{\Gamma}^{(i+1)} + \alpha \boldsymbol{\Gamma}^{(i)} = \boldsymbol{\Gamma}^{(i)} + \alpha \frac{\sum_{n=1}^{N} \frac{\boldsymbol{\Gamma}^{(i)}-\boldsymbol{P}_n}{\|\boldsymbol{\Gamma}^{(i)}-\boldsymbol{P}_n\|}}{\sum_{n=1}^{N} \frac{1}{\|\boldsymbol{\Gamma}^{(i)}-\boldsymbol{P}_n\|}}
\tag{3.33}
$$

Let

$$
\frac{-\alpha}{\sum_{n=1}^{N} \frac{1}{\|\boldsymbol{\Gamma}^{(i)}-\boldsymbol{P}_n\|}} \triangleq \alpha_i \quad \text{and} \quad \boldsymbol{\Gamma}^{(i+1)} + \alpha \boldsymbol{\Gamma}^{(i)} \triangleq \boldsymbol{\Gamma}^{(i+1)\prime}
\tag{3.34}
$$

where α_i is the variable step size which depends on $\boldsymbol{\Gamma}^{(i)}$ and $\boldsymbol{\Gamma}^{(i+1)\prime}$ is defined as the new $\boldsymbol{\Gamma}^{(i+1)}$. Then Eq. (3.33) becomes

$$
\boldsymbol{\Gamma}^{(i+1)\prime} = \boldsymbol{\Gamma}^{(i)} - \alpha_i \nabla f_{\mathrm{E}}(\boldsymbol{\Gamma}^{(i)})
\tag{3.35}
$$

which is exactly the form of Eq. (3.31).

3.4   Riemannian Median

Following the idea of locating the Riemannian mean, in order to find the Riemannian median, we can first lift all the PSD matrices to the isometric Euclidean space and then perform the optimization process in Euclidean subspace. After obtaining the optimum solution, project back to the manifold. We will investigate the Riemannian median according to different distance measures in this section.

3.4.1   Riemannian Median According to d_{R_1}

The problem of locating the Riemannian median according to d_{R_1} (RMd$_1$) can be stated as: Given the RD $d_{R_1}(\boldsymbol{P}_m, \boldsymbol{P}_n)$ between two PSD matrices $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$, find $\boldsymbol{\Gamma}_{R_1}$, the RMd$_1$ of $\{\boldsymbol{P}_n, n = 1, \cdots, N\}$, such that

$$\boldsymbol{\Gamma}_{R_1} = \arg\min_{\boldsymbol{\Gamma}} f_{R_1} = \arg\min_{\boldsymbol{\Gamma}} \sum_{n=1}^{N} d_{R_1}(\boldsymbol{P}_n, \boldsymbol{\Gamma}) \tag{3.36}$$

where d_{R_1} is presented by Eq. (2.10). The algorithm for locating the RMd$_1$ for $\{\boldsymbol{P}_n\}$ is developed based on the similar facts as we find RMn$_1$ such that

F1: A chosen centre point can be lifted together with all $\boldsymbol{P}_n$ to an isometric Euclidean subspace.

F2: The isometry between $\mathcal{T}_{\mathcal{M}}$ and $\mathcal{U}_{\mathcal{H}}$ enables us to solve the optimization problem in terms of equivalent ED.

F3: The optimization in the Euclidean subspace is convex and a convergent algorithm which is given as follows can be employed to find the optimum point.

Algorithm RMd1:

31

1. For $i = 0$: Initialize $\boldsymbol{\Gamma}^{(i)} = \frac{1}{N} \sum_{n=1}^{N} \boldsymbol{P}_n$, set a positive precision ϵ.

2. Set: $\tilde{\boldsymbol{\Gamma}}^{(i)} = \left(\boldsymbol{\Gamma}^{(i)}\right)^{1/2}$.

3. Find $\hat{\boldsymbol{V}}_n^{(i)}$ which minimizes $\left\| \boldsymbol{P}_n^{1/2} \boldsymbol{V}_n^{(i)} - \tilde{\boldsymbol{\Gamma}}^{(i)} \right\|_2^2, \forall n$:

   $\hat{\boldsymbol{V}}_n^{(i)} = \boldsymbol{U}_l \boldsymbol{U}_r^H$, where $\boldsymbol{U}_l \boldsymbol{\Sigma} \boldsymbol{U}_r^H = \boldsymbol{P}_n^{1/2} \tilde{\boldsymbol{\Gamma}}^{(i)}$ is the SVD of $\boldsymbol{P}_n^{1/2} \tilde{\boldsymbol{\Gamma}}^{(i)}$.

4. Lift to $\mathcal{U}_{\mathcal{H}}$: $\tilde{\boldsymbol{P}}_n^{(i)} = \boldsymbol{P}_n^{1/2} \hat{\boldsymbol{V}}_n^{(i)}$.

5. Get the new median in $\mathcal{H}$: Set the initial value $\tilde{\boldsymbol{\Gamma}}'^{(i,0)}$, then apply *Algorithm EMd* to locate the Euclidean median in $\mathcal{H}$, i.e. $\tilde{\boldsymbol{\Gamma}}'^{(i)}_{\text{E}} = \tilde{\boldsymbol{\Gamma}}'^{(i,J)}$ with J being the last iteration of *Algorithm EMd*.

6. Update the Riemannian median on $\mathcal{M}$: $\boldsymbol{\Gamma}^{(i+1)} = \tilde{\boldsymbol{\Gamma}}'^{(i)}_{\text{E}} \tilde{\boldsymbol{\Gamma}}'^{(i)H}_{\text{E}}$.

7. Calculate the precision: $h = d_{\text{R}_1}\left(\boldsymbol{\Gamma}^{(i+1)}, \boldsymbol{\Gamma}^{(i)}\right)$.

8. If $h > \epsilon$, let $i \to i + 1$ and go back to Step 2. Otherwise, obtain the RMd$_1$ $\boldsymbol{\Gamma}_{\text{R}_1} = \boldsymbol{\Gamma}^{(i+1)}$ and exit. ∎

Theorem 3. *For the PSD matrices $\{\boldsymbol{P}_n\}$, the Riemannian median according to d_{R_1} can be obtained by Algorithm RMd1 and is given by*

$$\boldsymbol{\Gamma}_{\text{R}_1} = \lim_{i \to \infty} \boldsymbol{\Gamma}^{(i)} \tag{3.37}$$

where $\boldsymbol{\Gamma}_{\text{R}_1} = \arg\min_{\boldsymbol{\Gamma}} \sum_{n=1}^{N} d_{\text{R}_1}(\boldsymbol{P}_n, \boldsymbol{\Gamma})$ is the RMd$_1$. □

The convergence of *Algorithm RMd1* can be shown with reference to the same procedure as in proof of Theorem 1. The alternating mapping from $\mathcal{M}$ to $\mathcal{U}_{\mathcal{H}}$ and back with the optimization result obtained by the equivalent sum of d_{E} from all the points in $\mathcal{U}_{\mathcal{H}}$ provide a contraction.

3.4.2   Riemannian Median According to $d_{\mathrm{WR_1}}$

For the weighted Riemannian median according to $d_{\mathrm{WR_1}}$, the problem can be formulated as the following equation.

$$
\begin{aligned}
\boldsymbol{\Gamma}_{\mathrm{WR_1}} &= \arg\min_{\boldsymbol{\Gamma}} f_{\mathrm{WR_1}} = \arg\min_{\boldsymbol{\Gamma}} \sum_{n=1}^{N} d_{\mathrm{WR_1}}(\boldsymbol{P}_n, \boldsymbol{\Gamma}) \\
&= \arg\min_{\boldsymbol{\Gamma}} \sum_{n=1}^{N} \sqrt{\mathrm{tr}\boldsymbol{\Gamma}_{\mathrm{W}} + \mathrm{tr}\boldsymbol{P}_{n\mathrm{W}} - 2\mathrm{tr}\left[\left(\boldsymbol{\Gamma}_{\mathrm{W}}^{1/2}\boldsymbol{W}\boldsymbol{P}_n\boldsymbol{\Gamma}_{\mathrm{W}}^{1/2}\right)^{1/2}\right]}
\end{aligned}
\tag{3.38}
$$

where $\boldsymbol{\Gamma}_{\mathrm{W}} = \boldsymbol{\Omega}^H\boldsymbol{\Gamma}\boldsymbol{\Omega}$ is the weighted version of the central point and $\boldsymbol{P}_{n\mathrm{W}} = \boldsymbol{\Omega}^H\boldsymbol{P}_n\boldsymbol{\Omega}$ is the weighted version of the PSD matrix.

Again, as we discussed in the case of finding the RMn according to $d_{\mathrm{WR_1}}$, the terms under the square root sign cannot be written as $\left\|\tilde{\boldsymbol{P}}_{n\mathrm{W_1}} - \tilde{\boldsymbol{\Gamma}}_{\mathrm{W_1}}\right\|$ directly, where $\tilde{\boldsymbol{P}}_{n\mathrm{W_1}} = \boldsymbol{\Omega}^H\boldsymbol{P}_{n\mathrm{W}}^{1/2}\boldsymbol{U}_n$ and $\tilde{\boldsymbol{\Gamma}}_{\mathrm{W_1}} = \boldsymbol{\Omega}^H\boldsymbol{\Gamma}_{\mathrm{W}}^{1/2}\boldsymbol{U}_{\Gamma}$ with $\boldsymbol{U}_n$ and $\boldsymbol{U}_{\Gamma}$ being unitary matrices. Nevertheless, its upper bound can be written as the form of sum d_{WE} such that

$$
\begin{aligned}
&\sum_{n=1}^{N} \sqrt{\mathrm{tr}(\tilde{\boldsymbol{P}}_{n\mathrm{W_1}}\tilde{\boldsymbol{P}}_{n\mathrm{W_1}}^{H}) + \mathrm{tr}(\tilde{\boldsymbol{\Gamma}}_{\mathrm{W_1}}\tilde{\boldsymbol{\Gamma}}_{\mathrm{W_1}}^{H}) - 2\Re\left[\mathrm{tr}(\tilde{\boldsymbol{P}}_{n\mathrm{W_1}}^{H}\tilde{\boldsymbol{\Gamma}}_{\mathrm{W_1}})\right]} \\
&= \sum_{n=1}^{N} \left\|\tilde{\boldsymbol{P}}_{n\mathrm{W_1}} - \tilde{\boldsymbol{\Gamma}}_{\mathrm{W_1}}\right\|
\end{aligned}
\tag{3.39}
$$

Therefore, this upper bound in Eq. (3.39), i.e., the sum of ED between the lifted points, can be minimized by *Algorithm EMd* in Euclidean space. Hence, we can also apply *Algorithm RMd1* to locate the weighted Riemannian median according to $d_{\mathrm{WR_1}}$ (WRMd$_1$). Then, we have

Corollary 3. *For the weighting $\boldsymbol{W} = \boldsymbol{\Omega}\boldsymbol{\Omega}^H$, the weighted Riemannian median according to the weighted RD $d_{\mathrm{WR_1}}$ is achieved by*

$$\boldsymbol{\Gamma}_{\mathrm{WR_1}} = \lim_{i \to \infty} \boldsymbol{\Gamma}_{\mathrm{W}}^{(i)} \tag{3.40}$$

where $\boldsymbol{\Gamma}_{\mathrm{WR_1}}$ is the WRMd$_1$ and $\boldsymbol{\Gamma}_{\mathrm{W}}^{(\infty)}$ is obtained by iteratively applying Algorithm RMd1 on the weighted PSD matrices $\boldsymbol{P}_{n\mathrm{W}}$. $\qquad\qquad\square$

3.4.3   Riemannian Median According to $d_{\mathrm{R_2}}$

The Riemannian median according to $d_{\mathrm{R_2}}$ (RMd$_2$) can be defined as

$$\boldsymbol{\Gamma}_{\mathrm{R_2}} = \arg\min_{\boldsymbol{\Gamma}} f_{\mathrm{R_2}} = \arg\min_{\boldsymbol{\Gamma}} \sum_{n=1}^{N} d_{\mathrm{R_2}}(\boldsymbol{P}_n, \boldsymbol{\Gamma}) \tag{3.41}$$

Using the idea when we find the RMn$_2$, first, we lift all the PSD matrices to the Euclidean space following the mapping formula $\tilde{\boldsymbol{P}}_n = \boldsymbol{P}_n^{1/2}$. Then we can find the median of $\tilde{\boldsymbol{P}}_n$ in the Euclidean space by the *Algorithm EMd*. After finding the median in Euclidean space, we can project it back to manifold to obtain the RMd$_2$ in manifold. The algorithm is given in details as follows.

Algorithm RMd2:

1. Lift to $\mathcal{U}_{\mathcal{H}}$: $\tilde{\boldsymbol{P}}_n = \boldsymbol{P}_n^{1/2}$.

2. Get the new median in $\mathcal{H}$:   Set the initial value $\tilde{\boldsymbol{\Gamma}}^{(0)}$, then apply *Algorithm EMd* to locate the Euclidean median in $\mathcal{H}$, i.e. $\tilde{\boldsymbol{\Gamma}}_{\mathrm{E}} = \tilde{\boldsymbol{\Gamma}}^{(J)}$ with J being the last iteration of *Algorithm EMd*.

3. Back to $\mathcal{M}$: $\boldsymbol{\Gamma}_{\mathrm{R_2}} = \tilde{\boldsymbol{\Gamma}}_{\mathrm{E}}\tilde{\boldsymbol{\Gamma}}_{\mathrm{E}}$. $\blacksquare$

We note that the mapping from $\mathcal{M}$ onto $\mathcal{U}_\mathcal{H}$ uses the fixed unitary matrix $\boldsymbol{I}_M$ so that we need no re-iteration of the lifting after the first projection. Thus, we have

Theorem 4. *For the PSD matrices $\{\boldsymbol{P}_n\}$, the Riemannian median according to d_{R_2} can be obtained by Algorithm RMd2 such that*

$$\boldsymbol{\Gamma}_{\mathrm{R}_2} = \lim_{j \to \infty} \left(\tilde{\boldsymbol{\Gamma}}^{(j)} \right)^2 \tag{3.42}$$

where $\boldsymbol{\Gamma}_{\mathrm{R}_2} = \arg\min_{\boldsymbol{\Gamma}} \sum_{n=1}^{N} d_{\mathrm{R}_2}(\boldsymbol{P}_n, \boldsymbol{\Gamma})$ is the RMd$_2$ and $\tilde{\boldsymbol{\Gamma}}^{(\infty)}$ is the Euclidean median obtained by employing Algorithm EMd on the sum ED. □

Since the objective function in Eq. (3.41) is not convex (shown in Appendix A.2.3), we need to carry out the optimization process in the isometric Euclidean subspace and then project it back to manifold. Due to the isometry between $\mathcal{T}_\mathcal{M}$ and $\mathcal{U}_\mathcal{H}$, $\boldsymbol{\Gamma}_{\mathrm{R}_2}$ is the global optimum point on the manifold.

3.4.4 Riemannian Median According to d_{WR_2}

Given the knowledge in previous sections, it is fairly straightforward to determine the weighted Riemannian median according to d_{WR_2} (WRMd$_2$). For an $M \times M$ positive definite Hermitian matrix $\boldsymbol{W} = \boldsymbol{\Omega}\boldsymbol{\Omega}^H$, the WRMd$_2$ of $\{\boldsymbol{P}_n, n = 1, \cdots, N\}$, denoted by $\boldsymbol{\Gamma}_{\mathrm{WR}_2}$, can be obtained by

$$\boldsymbol{\Gamma}_{\mathrm{WR}_2} = \arg\min_{\boldsymbol{\Gamma}} f_{\mathrm{WR}_2} = \arg\min_{\boldsymbol{\Gamma}} \sum_{n=1}^{N} d_{\mathrm{WR}_2}(\boldsymbol{P}_n, \boldsymbol{\Gamma}) \tag{3.43}$$

Based on Eq. (2.21), the objective function of Eq. (3.43) can be expressed as

$$
\begin{aligned}
f_{\mathrm{WR}_2} &= \sum_{n=1}^{N} \sqrt{\mathrm{tr}(\tilde{\boldsymbol{P}}_{n\mathrm{W}_2}\tilde{\boldsymbol{P}}_{n\mathrm{W}_2}) + \mathrm{tr}(\tilde{\boldsymbol{\Gamma}}_{\mathrm{W}_2}\tilde{\boldsymbol{\Gamma}}_{\mathrm{W}_2}) - 2\mathrm{tr}(\tilde{\boldsymbol{P}}_{n\mathrm{W}_2}\tilde{\boldsymbol{\Gamma}}_{\mathrm{W}_2})} \\
&= \sum_{n=1}^{N} \left\| \tilde{\boldsymbol{P}}_{n\mathrm{W}_2} - \tilde{\boldsymbol{\Gamma}}_{\mathrm{W}_2} \right\|
\end{aligned}
\tag{3.44}
$$

to which we can apply *Algorithm EMd* directly. Here, $\tilde{\boldsymbol{P}}_{n\mathrm{W}_2} = \left(\boldsymbol{\Omega}^H \boldsymbol{P}_n \boldsymbol{\Omega}\right)^{1/2}$ and $\tilde{\boldsymbol{\Gamma}}_{\mathrm{W}_2} = \left(\boldsymbol{\Omega}^H \boldsymbol{\Gamma} \boldsymbol{\Omega}\right)^{1/2}$. The optimum median achieved in Euclidean space can then be projected directly via the mapping formula $\boldsymbol{P} = \tilde{\boldsymbol{P}}^2$ to provide us with the WRMd$_2$. Therefore, we have the following corollary:

Corollary 4. *For the weighting $\boldsymbol{W} = \boldsymbol{\Omega}\boldsymbol{\Omega}^H$, the weighted Riemannian median according to the weighted RD d_{WR_2} is achieved by*

$$
\boldsymbol{\Gamma}_{\mathrm{WR}_2} = \lim_{j \to \infty} \left(\tilde{\boldsymbol{\Gamma}}_{\mathrm{W}}^{(j)}\right)^2
\tag{3.45}
$$

where $\boldsymbol{\Gamma}_{\mathrm{WR}_2}$ is the WRMd$_2$ and $\tilde{\boldsymbol{\Gamma}}_{\mathrm{W}}^{(\infty)}$ is the weighted Euclidean median obtained by applying Algorithm EMd on the weighted PSD matrices $\boldsymbol{P}_{n\mathrm{W}}$. $\qquad\square$

3.5 Verification of Our Algorithms

We have developed various algorithms to locate means and medians with different distance measures. According to the definition, the mean of N PSD matrices should have the minimum sum of squared distances from all the members. To verify the validity of our algorithms, instead of testing all the matrices which may not be possible, we choose various means and medians to make the comparison.

Example 1:

We first generate $N = 500$ noise PSD matrices, then we evaluate accordingly their various central points. These central points are given below.

$$
\boldsymbol{C}_{\mathrm{E}} = \begin{bmatrix}
0.9821 + 0.0000i & -0.0403 - 0.0392i & -0.0031 + 0.0351i & -0.0216 - 0.0042i \\
-0.0403 + 0.0392i & 0.8574 - 0.0000i & -0.0360 + 0.0295i & 0.0308 + 0.0404i \\
-0.0031 - 0.0351i & -0.0360 - 0.0295i & 1.0315 - 0.0000i & 0.0102 - 0.1079i \\
-0.0216 + 0.0042i & 0.0308 - 0.0404i & 0.0102 + 0.1079i & 0.9181 - 0.0000i
\end{bmatrix}
$$

$$
\boldsymbol{C}_{\mathrm{WE}} = \begin{bmatrix}
0.8192 - 0.0000i & 0.0000 + 0.0000i & 0.0000 - 0.0000i & 0.0000 + 0.0000i \\
0.0000 - 0.0000i & 0.8627 + 0.0000i & 0.0000 - 0.0000i & -0.0000 - 0.0000i \\
0.0000 + 0.0000i & 0.0000 + 0.0000i & 1.0272 + 0.0000i & -0.0000 + 0.0000i \\
0.0000 - 0.0000i & -0.0000 + 0.0000i & -0.0000 - 0.0000i & 1.0647 + 0.0000i
\end{bmatrix}
$$

$$
\boldsymbol{C}_{\mathrm{R}_1} = \begin{bmatrix}
0.2834 + 0.0000i & -0.0663 - 0.0508i & -0.0128 + 0.0368i & -0.0332 + 0.0071i \\
-0.0663 + 0.0508i & 0.0893 + 0.0000i & -0.0761 + 0.0525i & 0.0266 + 0.0790i \\
-0.0128 - 0.0368i & -0.0761 - 0.0525i & 0.3784 + 0.0000i & 0.0327 - 0.2367i \\
-0.0332 - 0.0071i & 0.0266 - 0.0790i & 0.0327 + 0.2367i & 0.1977 + 0.0000i
\end{bmatrix}
$$

$$
\boldsymbol{C}_{\mathrm{WR}_1} = \begin{bmatrix}
0.0038 + 0.0000i & 0.0043 + 0.0050i & 0.0040 - 0.0222i & 0.0290 - 0.0090i \\
0.0043 - 0.0050i & 0.0788 + 0.0000i & 0.0248 - 0.0104i & -0.0043 - 0.0392i \\
0.0040 + 0.0222i & 0.0248 + 0.0104i & 0.3943 - 0.0000i & -0.0138 - 0.0196i \\
0.0290 + 0.0090i & -0.0043 + 0.0392i & -0.0138 + 0.0196i & 0.4547 - 0.0000i
\end{bmatrix}
$$

$$
\boldsymbol{C}_{\mathrm{R_2}} = \begin{bmatrix} 0.2368 + 0.0000i & -0.0145 - 0.0144i & -0.0060 + 0.0126i & -0.0077 + 0.0024i \\ -0.0145 + 0.0144i & 0.1859 + 0.0000i & -0.0143 + 0.0101i & 0.0101 + 0.0213i \\ -0.0060 - 0.0126i & -0.0143 - 0.0101i & 0.2635 + 0.0000i & 0.0047 - 0.0523i \\ -0.0077 - 0.0024i & 0.0101 - 0.0213i & 0.0047 + 0.0523i & 0.2169 + 0.0000i \end{bmatrix}
$$

$$
\boldsymbol{C}_{\mathrm{WR_2}} = \begin{bmatrix} 0.1685 - 0.0000i & -0.0003 - 0.0020i & 0.0008 - 0.0028i & 0.0013 + 0.0009i \\ -0.0003 + 0.0020i & 0.1930 + 0.0000i & 0.0017 - 0.0033i & -0.0020 - 0.0049i \\ 0.0008 + 0.0028i & 0.0017 + 0.0033i & 0.2548 + 0.0000i & -0.0031 - 0.0055i \\ 0.0013 - 0.0009i & -0.0020 + 0.0049i & -0.0031 + 0.0055i & 0.2797 - 0.0000i \end{bmatrix}
$$

$$
\boldsymbol{\Gamma}_{\mathrm{E}} = \begin{bmatrix} 0.7690 + 0.0000i & -0.0211 - 0.0217i & -0.0179 + 0.0165i & -0.0137 + 0.0122i \\ -0.0211 + 0.0217i & 0.6860 - 0.0000i & -0.0176 + 0.0122i & 0.0121 + 0.0426i \\ -0.0179 - 0.0165i & -0.0176 - 0.0122i & 0.8117 + 0.0000i & 0.0059 - 0.0893i \\ -0.0137 - 0.0122i & 0.0121 - 0.0426i & 0.0059 + 0.0893i & 0.7436 - 0.0000i \end{bmatrix}
$$

$$
\boldsymbol{\Gamma}_{\mathrm{WE}} = \begin{bmatrix} 0.6520 - 0.0000i & 0.0006 - 0.0049i & 0.0012 - 0.0078i & 0.0042 - 0.0003i \\ 0.0006 + 0.0049i & 0.7102 + 0.0000i & 0.0073 - 0.0114i & -0.0005 - 0.0160i \\ 0.0012 + 0.0078i & 0.0073 + 0.0114i & 0.7941 + 0.0000i & -0.0143 - 0.0138i \\ 0.0042 + 0.0003i & -0.0005 + 0.0160i & -0.0143 + 0.0138i & 0.8426 + 0.0000i \end{bmatrix}
$$

$$
\boldsymbol{\Gamma}_{\mathrm{R_1}} = \begin{bmatrix} 0.1931 + 0.0000i & -0.0353 - 0.0297i & -0.0513 + 0.0159i & -0.0207 + 0.0297i \\ -0.0353 + 0.0297i & 0.0565 + 0.0000i & -0.0890 + 0.0451i & 0.0235 + 0.0755i \\ -0.0513 - 0.0159i & -0.0890 - 0.0451i & 0.4016 + 0.0000i & 0.0368 - 0.2762i \\ -0.0207 - 0.0297i & 0.0235 - 0.0755i & 0.0368 + 0.2762i & 0.2140 + 0.0000i \end{bmatrix}
$$

$$\mathbf{\Gamma}_{\text{WR}_1} = \begin{bmatrix} 0.0027 - 0.0000i & 0.0010 + 0.0017i & 0.0034 - 0.0211i & 0.0186 + 0.0051i \\ 0.0010 - 0.0017i & 0.0444 - 0.0000i & 0.0426 - 0.0235i & -0.0120 - 0.0711i \\ 0.0034 + 0.0211i & 0.0426 + 0.0235i & 0.3245 + 0.0000i & -0.0401 - 0.0750i \\ 0.0186 - 0.0051i & -0.0120 + 0.0711i & -0.0401 + 0.0750i & 0.4678 + 0.0000i \end{bmatrix}$$

$$\mathbf{\Gamma}_{\text{R}_2} = \begin{bmatrix} 0.2046 + 0.0000i & -0.0106 - 0.0111i & -0.0096 + 0.0068i & -0.0068 + 0.0057i \\ -0.0106 + 0.0111i & 0.1630 + 0.0000i & -0.0109 + 0.0063i & 0.0068 + 0.0213i \\ -0.0096 - 0.0068i & -0.0109 - 0.0063i & 0.2314 + 0.0000i & 0.0030 - 0.0478i \\ -0.0068 - 0.0057i & 0.0068 - 0.0213i & 0.0030 + 0.0478i & 0.1937 + 0.0000i \end{bmatrix}$$

$$\mathbf{\Gamma}_{\text{WR}_2} = \begin{bmatrix} 0.1473 + 0.0000i & -0.0010 - 0.0035i & 0.0007 - 0.0040i & 0.0004 + 0.0015i \\ -0.0010 + 0.0035i & 0.1738 + 0.0000i & 0.0030 - 0.0051i & -0.0026 - 0.0088i \\ 0.0007 + 0.0040i & 0.0030 + 0.0051i & 0.2199 + 0.0000i & -0.0053 - 0.0103i \\ 0.0004 - 0.0015i & -0.0026 + 0.0088i & -0.0053 + 0.0103i & 0.2454 + 0.0000i \end{bmatrix}$$

The corresponding sum of squared distances from the various central points are shown in Table 3.1.

	$\sum\limits_{n=1}^{N} d_{\mathrm{E}}^2$	$\sum\limits_{n=1}^{N} d_{\mathrm{WE}}^2$	$\sum\limits_{n=1}^{N} d_{\mathrm{R_1}}^2$	$\sum\limits_{n=1}^{N} d_{\mathrm{WR_1}}^2$	$\sum\limits_{n=1}^{N} d_{\mathrm{R_2}}^2$	$\sum\limits_{n=1}^{N} d_{\mathrm{WR_2}}^2$
$\boldsymbol{C}_{\mathrm{E}}$	**484.2901**	482.3316	507.8601	507.1634	508.1945	507.3834
$\boldsymbol{C}_{\mathrm{WE}}$	484.2901	**482.3316**	507.8601	507.1634	508.1945	507.3834
$\boldsymbol{C}_{\mathrm{R_1}}$	555.7308	555.1512	**374.6509**	376.0498	390.8902	391.6534
$\boldsymbol{C}_{\mathrm{WR_1}}$	555.9241	553.5835	375.6121	**375.0809**	384.3298	383.7161
$\boldsymbol{C}_{\mathrm{R_2}}$	555.8356	553.5569	376.9549	376.5047	**377.5626**	376.9539
$\boldsymbol{C}_{\mathrm{WR_2}}$	555.9963	553.5435	377.1434	376.5106	377.6096	**376.9076**
$\boldsymbol{\Gamma}_{\mathrm{E}}$	489.1917	487.2034	460.3570	557.0871	460.6884	507.3834
$\boldsymbol{\Gamma}_{\mathrm{WE}}$	489.1917	487.2034	460.3570	557.0871	460.6884	507.3834
$\boldsymbol{\Gamma}_{\mathrm{R_1}}$	561.1248	560.7709	375.5103	377.5273	412.0111	414.0201
$\boldsymbol{\Gamma}_{\mathrm{WR_1}}$	561.5835	559.3636	376.3586	376.0465	395.4412	395.0649
$\boldsymbol{\Gamma}_{\mathrm{R_2}}$	561.0785	558.7656	377.3453	376.9052	378.0116	377.4115
$\boldsymbol{\Gamma}_{\mathrm{WR_2}}$	561.2166	558.7609	377.5129	376.9073	378.0489	377.3589

Table 3.1: Sum of squared distances from $N = 500$ PSD matrices

where d_x^2 is short for $d_x^2\left(\sim, \boldsymbol{P}_n\right)$ with "$\sim$" denoting the corresponding central point of $N = 500$ PSD matrices.

Example 2:

We now generate only $N = 6$ noise PSD matrices and evaluate their corresponding

central points. These are given below:

$$
C_{\mathrm{E}} =
\begin{bmatrix}
0.7646 - 0.0000i & -0.1105 + 0.4338i & -0.0721 + 0.0569i & -0.0528 + 0.2884i \\
-0.1105 - 0.4338i & 0.9486 - 0.0000i & 0.0111 - 0.3087i & 0.0961 - 0.0445i \\
-0.0721 - 0.0569i & 0.0111 + 0.3087i & 0.7616 - 0.0000i & 0.0020 - 0.1285i \\
-0.0528 - 0.2884i & 0.0961 + 0.0445i & 0.0020 + 0.1285i & 0.1843 - 0.0000i
\end{bmatrix}
$$

$$
C_{\mathrm{WE}} =
\begin{bmatrix}
0.8058 - 0.0000i & 0.3018 - 0.1907i & -0.3230 - 0.3818i & 0.1505 - 0.1864i \\
0.3018 + 0.1907i & 0.3710 + 0.0000i & 0.0688 - 0.0990i & 0.1359 - 0.1335i \\
-0.3230 + 0.3818i & 0.0688 + 0.0990i & 0.5697 + 0.0000i & -0.1840 + 0.1235i \\
0.1505 + 0.1864i & 0.1359 + 0.1335i & -0.1840 - 0.1235i & 0.9543 - 0.0000i
\end{bmatrix}
$$

$$
C_{\mathrm{R}_1} =
\begin{bmatrix}
0.4358 + 0.0000i & -0.1846 + 0.4976i & 0.1747 + 0.2348i & -0.0031 + 0.1374i \\
-0.1846 - 0.4976i & 0.6462 + 0.0000i & 0.1941 - 0.2990i & 0.1582 - 0.0547i \\
0.1747 - 0.2348i & 0.1941 + 0.2990i & 0.1966 + 0.0000i & 0.0728 + 0.0568i \\
-0.0031 - 0.1374i & 0.1582 + 0.0547i & 0.0728 - 0.0568i & 0.0434 + 0.0000i
\end{bmatrix}
$$

$$
C_{\mathrm{WR}_1} =
\begin{bmatrix}
0.6084 + 0.0000i & 0.1651 - 0.2604i & -0.2191 - 0.3594i & -0.0780 - 0.2727i \\
0.1651 + 0.2604i & 0.1563 + 0.0000i & 0.0944 - 0.1913i & 0.0956 - 0.1074i \\
-0.2191 + 0.3594i & 0.0944 + 0.1913i & 0.2912 - 0.0000i & 0.1892 + 0.0521i \\
-0.0780 + 0.2727i & 0.0956 + 0.1074i & 0.1892 - 0.0521i & 0.1322 + 0.0000i
\end{bmatrix}
$$

$$
C_{\mathrm{R}_2} =
\begin{bmatrix}
0.2745 + 0.0000i & -0.1143 + 0.2721i & 0.0573 + 0.0474i & -0.0081 + 0.1033i \\
-0.1143 - 0.2721i & 0.4639 + 0.0000i & 0.0202 - 0.2234i & 0.0796 - 0.0377i \\
0.0573 - 0.0474i & 0.0202 + 0.2234i & 0.2881 + 0.0000i & 0.0090 - 0.0154i \\
-0.0081 - 0.1033i & 0.0796 + 0.0377i & 0.0090 + 0.0154i & 0.0468 + 0.0000i
\end{bmatrix}
$$

41

$$
\boldsymbol{C}_{\mathrm{WR}_2} = \begin{bmatrix}
0.3596 + 0.0000i & 0.1059 - 0.0953i & -0.1737 - 0.1861i & 0.0767 - 0.1419i \\
0.1059 + 0.0953i & 0.0991 - 0.0000i & 0.0171 - 0.0678i & 0.1034 - 0.0711i \\
-0.1737 + 0.1861i & 0.0171 + 0.0678i & 0.2227 - 0.0000i & -0.0144 + 0.0580i \\
0.0767 + 0.1419i & 0.1034 + 0.0711i & -0.0144 - 0.0580i & 0.3362 + 0.0000i
\end{bmatrix}
$$

$$
\boldsymbol{\Gamma}_{\mathrm{E}} = \begin{bmatrix}
0.4898 - 0.0000i & -0.2091 + 0.3791i & 0.1445 + 0.0328i & -0.0281 + 0.1909i \\
-0.2091 - 0.3791i & 0.8265 - 0.0000i & -0.0090 - 0.3423i & 0.1129 - 0.0924i \\
0.1445 - 0.0328i & -0.0090 + 0.3423i & 0.6521 - 0.0000i & -0.0180 - 0.0269i \\
-0.0281 - 0.1909i & 0.1129 + 0.0924i & -0.0180 + 0.0269i & 0.1413 - 0.0000i
\end{bmatrix}
$$

$$
\boldsymbol{\Gamma}_{\mathrm{WE}} = \begin{bmatrix}
0.7112 - 0.0000i & 0.2025 - 0.1347i & -0.3647 - 0.2935i & 0.1520 - 0.3325i \\
0.2025 + 0.1347i & 0.2548 - 0.0000i & 0.0187 - 0.0736i & 0.2049 - 0.1161i \\
-0.3647 + 0.2935i & 0.0187 + 0.0736i & 0.4746 + 0.0000i & 0.0012 + 0.1573i \\
0.1520 + 0.3325i & 0.2049 + 0.1161i & 0.0012 - 0.1573i & 0.7209 + 0.0000i
\end{bmatrix}
$$

$$
\boldsymbol{\Gamma}_{\mathrm{R}_1} = \begin{bmatrix}
0.3634 + 0.0000i & -0.1849 + 0.4220i & 0.3112 + 0.0645i & -0.0093 + 0.1260i \\
-0.1849 - 0.4220i & 0.5840 + 0.0000i & -0.0834 - 0.3941i & 0.1510 - 0.0533i \\
0.3112 - 0.0645i & -0.0834 + 0.3941i & 0.2779 + 0.0000i & 0.0144 + 0.1095i \\
-0.0093 - 0.1260i & 0.1510 + 0.0533i & 0.0144 - 0.1095i & 0.0439 + 0.0000i
\end{bmatrix}
$$

$$
\boldsymbol{\Gamma}_{\mathrm{WR}_1} = \begin{bmatrix}
0.5389 + 0.0000i & 0.0235 - 0.1598i & -0.3024 - 0.3033i & 0.0663 - 0.3361i \\
0.0235 + 0.1598i & 0.0484 - 0.0000i & 0.0767 - 0.1029i & 0.1025 + 0.0050i \\
-0.3024 + 0.3033i & 0.0767 + 0.1029i & 0.3404 + 0.0000i & 0.1520 + 0.2259i \\
0.0663 + 0.3361i & 0.1025 - 0.0050i & 0.1520 - 0.2259i & 0.2177 - 0.0000i
\end{bmatrix}
$$

$$
\mathbf{\Gamma}_{\mathrm{R}_2} = \begin{bmatrix} 0.2564 + 0.0000i & -0.1568 + 0.2950i & 0.1832 + 0.0452i & -0.0040 + 0.0912i \\ -0.1568 - 0.2950i & 0.4891 + 0.0000i & -0.0447 - 0.2948i & 0.0946 - 0.0547i \\ 0.1832 - 0.0452i & -0.0447 + 0.2948i & 0.2782 + 0.0000i & 0.0082 + 0.0420i \\ -0.0040 - 0.0912i & 0.0946 + 0.0547i & 0.0082 - 0.0420i & 0.0381 + 0.0000i \end{bmatrix}
$$

$$
\mathbf{\Gamma}_{\mathrm{WR}_2} = \begin{bmatrix} 0.3944 - 0.0000i & 0.0881 - 0.1013i & -0.2036 - 0.1995i & 0.0711 - 0.2368i \\ 0.0881 + 0.1013i & 0.0763 + 0.0000i & 0.0196 - 0.0754i & 0.1134 - 0.0616i \\ -0.2036 + 0.1995i & 0.0196 + 0.0754i & 0.2313 - 0.0000i & 0.0689 + 0.1183i \\ 0.0711 + 0.2368i & 0.1134 + 0.0616i & 0.0689 - 0.1183i & 0.2943 + 0.0000i \end{bmatrix}
$$

The sum of squared distances from various central points are shown in Table 3.2:

	$\sum\limits_{n=1}^{N} d_{\mathrm{E}}^2$	$\sum\limits_{n=1}^{N} d_{\mathrm{WE}}^2$	$\sum\limits_{n=1}^{N} d_{\mathrm{R_1}}^2$	$\sum\limits_{n=1}^{N} d_{\mathrm{WR_1}}^2$	$\sum\limits_{n=1}^{N} d_{\mathrm{R_2}}^2$	$\sum\limits_{n=1}^{N} d_{\mathrm{WR_2}}^2$
$\boldsymbol{C}_{\mathrm{E}}$	**4.7006**	4.6207	5.1430	5.1085	5.3209	5.2823
$\boldsymbol{C}_{\mathrm{WE}}$	4.7006	**4.6207**	5.1430	5.1085	5.3209	5.2823
$\boldsymbol{C}_{\mathrm{R_1}}$	5.4567	5.3972	**3.3767**	3.3942	4.9472	4.9328
$\boldsymbol{C}_{\mathrm{WR_1}}$	5.4571	5.3955	3.3774	**3.3936**	4.9464	4.9305
$\boldsymbol{C}_{\mathrm{R_2}}$	5.5551	5.4605	3.7569	3.7362	**4.0563**	4.0256
$\boldsymbol{C}_{\mathrm{WR_2}}$	5.5562	5.4587	3.7591	3.7361	4.0570	**4.0249**
$\boldsymbol{\Gamma}_{\mathrm{E}}$	4.7997	4.7165	4.8696	5.3834	5.0566	5.2823
$\boldsymbol{\Gamma}_{\mathrm{WE}}$	4.7997	4.7165	4.8696	5.3834	5.0566	5.2823
$\boldsymbol{\Gamma}_{\mathrm{R_1}}$	5.4316	5.3564	3.4771	3.4936	4.9967	4.9708
$\boldsymbol{\Gamma}_{\mathrm{WR_1}}$	5.4296	5.3519	3.4834	3.4987	4.9957	4.9682
$\boldsymbol{\Gamma}_{\mathrm{R_2}}$	5.5280	5.4303	3.7438	3.7248	4.1088	4.0770
$\boldsymbol{\Gamma}_{\mathrm{WR_2}}$	5.5235	5.4235	3.7447	3.7238	4.1119	4.0788

Table 3.2: Sum of squared distances from $N = 6$ PSD matrices

where d_x^2 is short for $d_x^2\left(\sim, \boldsymbol{P}_n\right)$ with "$\sim$" denoting the corresponding central point of $N = 6$ PSD matrices, From Tables 3.1 and 3.2, it can be seen that in each column, the corresponding mean has the minimum sum of squared distances from all the PSD matrices, which is highlighted in bold. Therefore, it can be convinced that the mean located by our algorithm is the true mean.

For median, it is defined as the matrix that has the minimum sum of distances from all the PSD matrices in the group. Similarly, by making the comparison between

the sum of distances corresponding to various means and medians, the validity of our algorithms for locating the median can be shown in the following examples.

Example 3:

Using the same $N = 500$ noise PSD matrices as well as the corresponding central points as in *Example 1*, the sum of distances from various central points are shown in Table 3.3.

	$\sum_{n=1}^{N} d_{\mathrm{E}}$	$\sum_{n=1}^{N} d_{\mathrm{WE}}$	$\sum_{n=1}^{N} d_{\mathrm{R_1}}$	$\sum_{n=1}^{N} d_{\mathrm{WR_1}}$	$\sum_{n=1}^{N} d_{\mathrm{R_2}}$	$\sum_{n=1}^{N} d_{\mathrm{WR_2}}$
$\boldsymbol{C}_{\mathrm{E}}$	446.3844	446.3844	500.4792	500.1360	500.6434	500.2446
$\boldsymbol{C}_{\mathrm{WE}}$	446.3844	446.3844	500.4792	500.1360	500.6434	500.2446
$\boldsymbol{C}_{\mathrm{R_1}}$	469.5990	469.3148	420.2200	420.8369	429.6289	429.8723
$\boldsymbol{C}_{\mathrm{WR_1}}$	469.5123	468.4683	420.9276	420.5898	426.0617	425.6860
$\boldsymbol{C}_{\mathrm{R_2}}$	469.0966	468.0816	422.0814	421.8038	422.4375	422.0678
$\boldsymbol{C}_{\mathrm{WR_2}}$	469.1722	468.0833	422.1890	421.8199	422.4627	422.0542
$\boldsymbol{\Gamma}_{\mathrm{E}}$	**443.4276**	442.4832	474.9875	524.1424	475.1587	500.2446
$\boldsymbol{\Gamma}_{\mathrm{WE}}$	443.4276	**442.4832**	474.9875	524.1424	475.1587	500.2446
$\boldsymbol{\Gamma}_{\mathrm{R_1}}$	472.2872	472.1236	**419.7372**	420.6542	440.4412	441.3058
$\boldsymbol{\Gamma}_{\mathrm{WR_1}}$	472.4080	471.4313	420.3166	**420.0774**	431.4471	431.1869
$\boldsymbol{\Gamma}_{\mathrm{R_2}}$	471.6238	470.5996	421.7746	421.5026	**422.1653**	421.7998
$\boldsymbol{\Gamma}_{\mathrm{WR_2}}$	471.6898	470.6040	421.8706	421.5146	422.1850	**421.7803**

Table 3.3: Sum of distances from $N = 500$ PSD matrices

where d_x is short for $d_x \left(\sim, \boldsymbol{P}_n \right)$ with "$\sim$" denoting the corresponding central points

given in *Example 1*.

Example 4:

Using the same $N = 6$ noise PSD matrices and the same central points as in *Example 2*, the sum of distances from various central points are shown in Table 3.4.

	$\sum_{n=1}^{N} d_{\mathrm{E}}$	$\sum_{n=1}^{N} d_{\mathrm{WE}}$	$\sum_{n=1}^{N} d_{\mathrm{R_1}}$	$\sum_{n=1}^{N} d_{\mathrm{WR_1}}$	$\sum_{n=1}^{N} d_{\mathrm{R_2}}$	$\sum_{n=1}^{N} d_{\mathrm{WR_2}}$
$\boldsymbol{C}_{\mathrm{E}}$	5.1502	5.1010	5.5278	5.5078	5.6242	5.6027
$\boldsymbol{C}_{\mathrm{WE}}$	5.1502	5.1010	5.5278	5.5078	5.6242	5.6027
$\boldsymbol{C}_{\mathrm{R_1}}$	5.4780	5.4396	4.3773	4.3769	5.3245	5.3072
$\boldsymbol{C}_{\mathrm{WR_1}}$	5.4779	5.4386	4.3777	4.3764	5.3251	5.3068
$\boldsymbol{C}_{\mathrm{R_2}}$	5.4945	5.4491	4.6599	4.6436	4.8532	4.8325
$\boldsymbol{C}_{\mathrm{WR_2}}$	5.4943	5.4476	4.6602	4.6427	4.8533	4.8319
$\boldsymbol{\Gamma}_{\mathrm{E}}$	**5.1025**	5.0530	5.3433	5.6287	5.4469	5.6027
$\boldsymbol{\Gamma}_{\mathrm{WE}}$	5.1025	**5.0530**	5.3433	5.6287	5.4469	5.6027
$\boldsymbol{\Gamma}_{\mathrm{R_1}}$	5.3935	5.3475	**4.3257**	4.3220	5.2356	5.2127
$\boldsymbol{\Gamma}_{\mathrm{WR_1}}$	5.3909	5.3435	4.3263	**4.3214**	5.2352	5.2110
$\boldsymbol{\Gamma}_{\mathrm{R_2}}$	5.4459	5.3991	4.5923	4.5759	**4.8277**	4.8059
$\boldsymbol{\Gamma}_{\mathrm{WR_2}}$	5.4430	5.3950	4.5902	4.5727	4.8280	**4.8055**

Table 3.4: Sum of distances from $N = 6$ PSD matrices

where d_x is short for $d_x \left(\sim, \boldsymbol{P}_n \right)$ with "$\sim$" denoting the corresponding central points given in *Example 2*. From Tables 3.3 and 3.4, it can be observed that in each column, the corresponding median has the minimum sum of distances from all the

PSD matrices, which is highlighted in bold. That is to say, our algorithms are effective to find medians corresponding to different distance measures.

After verifying the validity of our algorithms for finding means and medians of random PSD matrices, we can employ them in the practical applications.

Chapter 4

Application to the Detection of Sonar Signals

Having established the concepts of different distance measures in chapter 2 and the algorithms of mean and median on the PSD manifold corresponding to those distances in chapter 3, in this chapter, we will explore how we can apply these results to signal processing. Here, we investigate the particular application about the detection of narrow-band passive sonar signals.

A sonar system is a system that uses sound propagation in underwater environments, usually the ocean, for detection, communication and navigation. The main purpose of sonar systems is to analyse the acoustic signals received by a sensor system and classify the nature of target that has been detected [30]. There are two types of sonar systems: active and passive. The difference between these two systems is that an active sonar system transmits acoustic signals into the water for the purpose of producing echoes, whereas a passive sonar system, instead of emitting any signals, listens to the signals emanated from the underwater targets, using an array of sensors

48

[31]. In this chapter, we perform the signal detection in a passive sonar system. The sensors in a passive sonar system are commonly arranged in the form of a uniformly spaced linear array, each of which receives acoustic signals from different directions. The received signals are then passed through a Fourier analyzer to determine the frequency components. The FFT of the received signals at different sensors constitute the output of a *beamformer* [32]. To perform signal detection, the beam data may need further processing.

In a sonar environment, the received signals consist of two parts: the desired signals from the target vessels and the ambient noise generated by the wind, waves, ocean creatures and even sea traffic. In our consideration, the noise is assumed to be stationary, zero mean Gaussian, and with a flat spectrum over the frequency band of analysis. On the other hand, the signals are usually originated from the mechanical vibrations of the propagation system, fast rotation of the propeller and auxiliary machinery in the moving submarine. These rotations and vibrations generate different sets of harmonics, propagated under multi-path environment. Therefore, the received signals are usually regarded as random narrow-band Gaussian. In the following sections, we will examine the process of determining if a signal is present in a particular frequency bin of the output of a beamformer.

4.1   Signal Model and Classical Detection Method

Consider a uniform linear array with P sensors in total such as Fig. 4.1, where the signal propagates with a known angle of arrival θ at a velocity c and the separation between sensors is Δ.
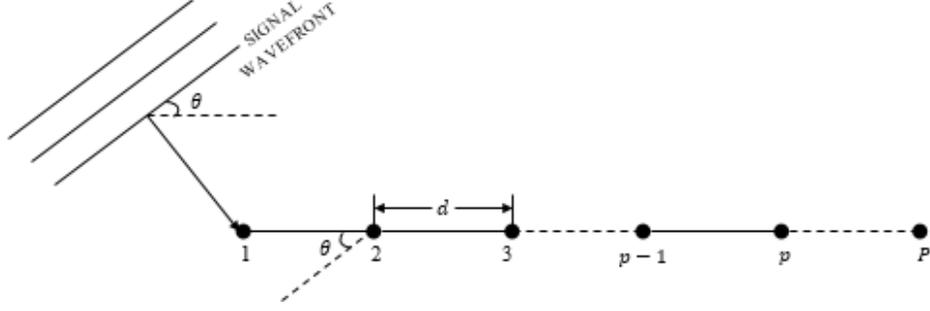
Figure 4.1: Geometry of the linear array

At the instant nT, with T being the sampling period, the discrete-time signal received at pth sensor, $p = 1, 2, \cdots, P$, can be represented as $x_p(t - p\Delta \sin\theta/c)$. Then, the summed output of the sensors, i.e., the *beam*, can be written as

$$b(nT, \theta) = \sum_{p=1}^{P} x_p(t - p\Delta \sin\theta/c) \tag{4.1}$$

In order to achieve more stable power spectrum, the output signal collected by the pth sensor is divided into M segments: $x_{pm}(nT), m = 1, \cdots, M$. The DFT of the beam of the received signal at the frequency bin $k\omega$ in the mth segment is given by

$$B_m(k\omega, \theta) = \sum_{p=1}^{P} X_{pm}(k\omega) e^{-jp(k\omega/c)\Delta \sin\theta}, \qquad m = 1, \cdots, M \tag{4.2}$$

where $X_{pm}(k\omega)$ is the DFT of the received signal $x_{pm}(nT)$ at frequency $k\omega$. An $M \times 1$ vector can then be formed as

$$\boldsymbol{\beta}_k \triangleq \boldsymbol{\beta}(k\omega, \theta) = [B_1(k\omega, \theta) \ B_2(k\omega, \theta) \ \cdots \ B_M(k\omega, \theta)]^T \tag{4.3}$$

50

The power spectrum can be constituted as

$$Z_x(k\omega, \theta) = \frac{1}{M}\boldsymbol{\beta}_k^H \boldsymbol{\beta}_k \qquad (4.4)$$

In classical binary hypothesis testing, we examine the *signal power* at $k\omega$ and compare it with the *mean noise power*. If the true mean noise power $\bar{Z}_\nu$ is known, then we can apply the Neyman-Pearson strategy [33] and the decision rule using the likelihood ratio can be written as [34]

$$\frac{Z_x(k\omega)}{\bar{Z}_\nu} \underset{H_0}{\overset{H_1}{\gtrless}} r_c \qquad (4.5)$$

where r_c is a positive number decided by a constant false alarm rate using a priori probability density function of noise. However, in practice, $\bar{Z}_\nu$ is not available and thus the mean noise power has to be estimated as $\hat{Z}_\nu$ to replace $\bar{Z}_\nu$ in Eq. (4.5) for the detection rule. There are several methods [34] to obtain the estimated noise power $\hat{Z}_\nu$, of which the most common one is the split window moving average (SWMA) method. It shows that the estimated noise power in a particular frequency bin $k\omega$ can be obtained by averaging the samples in the $2L$ neighbouring frequency bins of the same beam, L samples on either side of $k\omega$. This method is unbiased by assuming that the neighbouring bins contain noise samples only. If there are signals dropped in the neighbouring frequency bins, the estimation may be biased. The coefficients of the SWMA filter with window size $2L+1$ are given by

$$a_{-L} = \cdots = a_{-1} = a_1 = \cdots = a_L = \frac{1}{2L}, \qquad a_0 = 0 \qquad (4.6)$$

Now, by varying the threshold r_c from $0 \to \infty$, we can plot the complete receiver operation characteristic (ROC) [33] which shows the probability of detection P_D, with the corresponding probability of false alarm P_F .

4.2　Signal Detection on the PSD Matrix Manifold

In this section, we will set up the detection procedure for the narrow-band sonar signals using the PSD matrix as the detection feature [25]. As we mentioned earlier, since PSD matrices possess additional correlation information between different segments of measured signals, we expect more accurate detection results.

4.2.1　Binary Hypothesis Testing

For each frequency $k\omega$, the $M \times M$ PSD matrix can be formed by the outer product of the beamformer output given in Eq. (4.3) such that

$$\boldsymbol{P}_k = \boldsymbol{\beta}_k \boldsymbol{\beta}_k^H \tag{4.7}$$

To decide if there is noise only or (signal+noise) in the frequency bin $k\omega$ can be formulated as a binary decision problem. Let us separate these matrices into two groups: $\mathcal{N}$ for the noise group and $\mathcal{S}$ for the (signal+noise) group. Here, let us use the term "central point" to refer to either mean or median among the signal or noise PSD matrices, and we denote the central point of signal by $\boldsymbol{M}_s$ and the central point of noise by $\boldsymbol{M}_\nu$. If $\boldsymbol{P}_k \in \mathcal{N}$, our hypothesis claims that there is a certain similarity between $\boldsymbol{P}_k$ and $\boldsymbol{M}_\nu$; otherwise, $\boldsymbol{P}_k$ is dissimilar to $\boldsymbol{M}_\nu$. The similarity can be evaluated by the distance between the two matrices such that any PSD matrix

within the distance of r_α from $\boldsymbol{M}_\nu$ will be decided as a noise matrix, thus our decision rule can be given by

$$d_x(\boldsymbol{P}_k, \boldsymbol{M}_\nu) \underset{H_0}{\overset{H_1}{\gtrless}} r_\alpha \tag{4.8}$$

Eq. (4.8) uses $\boldsymbol{M}_\nu$ as a reference to decide whether $\boldsymbol{P}_k$ is a noise or contains signal. On the other hand, the PSD matrix $\boldsymbol{P}_k$ is expected to satisfy the following model.

$$\boldsymbol{P}_k = \begin{cases} \boldsymbol{M}_s + \boldsymbol{M}_\nu & \text{signal present} \\[2mm] \boldsymbol{0} + \boldsymbol{M}_\nu & \text{no signal} \end{cases} \tag{4.9}$$

This binary hypothesis implies, on average, the PSD matrix in a certain frequency bin is either $\boldsymbol{M}_s$ or $\boldsymbol{0}$, which means we can translate the reference to the origin and the hypothesis decision rule will be

$$d_x(\boldsymbol{P}_k - \boldsymbol{M}_\nu, \boldsymbol{0}) \underset{H_0}{\overset{H_1}{\gtrless}} r_\alpha \tag{4.10}$$

where we compare the distance between the PSD matrix under test less the central noise point and the null matrix, to an assigned radius r_α. Parallel to the Neyman-Pearson strategy, given a prescribed maximum allowable false alarm rate of $\alpha\%$, the radius r_α makes a circle centred at $\boldsymbol{M}_\nu$ such that beyond this boundary, there will be no more than $\alpha\%$ of the noise PSD matrices on the manifold. This can be interpreted as shown in the following Fig. 4.2.
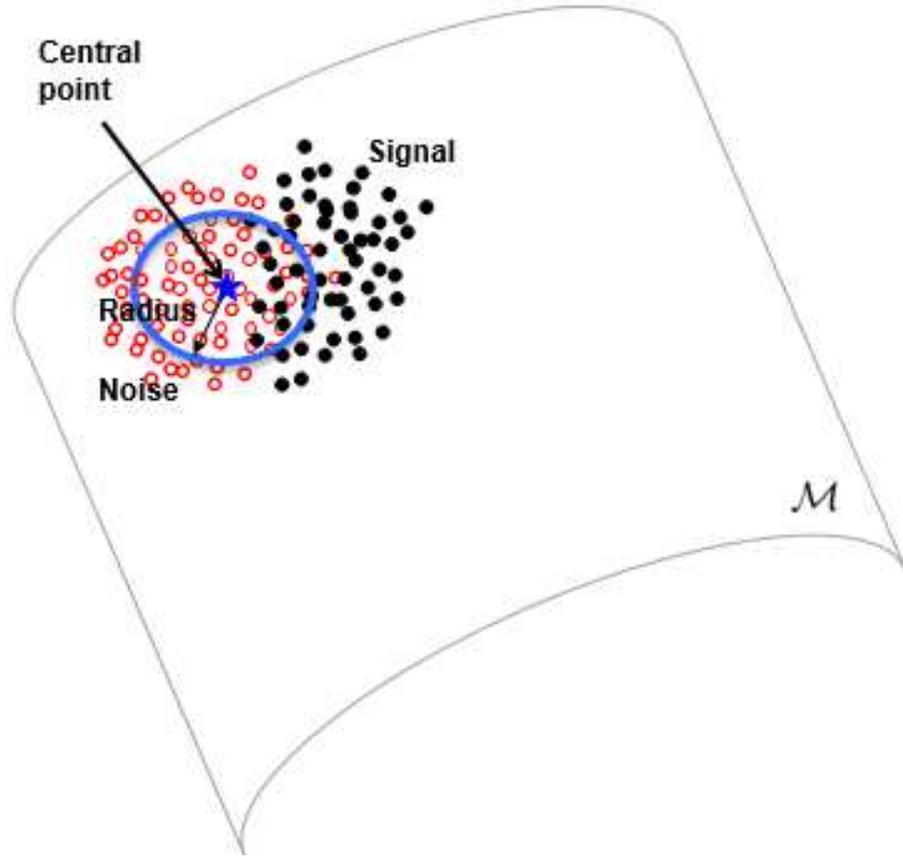
Figure 4.2: Detection regions on manifold

The noise PSD matrices and signal PSD matrices are respectively denoted by the hollow dots and solid dots. The boundary is marked out by a certain radius from the central point denoted by the solid star. The radius r_α can be established from the prior knowledge of the histogram distribution of normalized noise PSD matrices. By choosing different distance measures of d_x (i.e., $d_E, d_{R_1}, d_{R_2}, d_{WR_1}, d_{WR_2}$), together with the corresponding centre points $\boldsymbol{M}_\nu$ (i.e., EMn, EMd, WEMn, WEMd, RMn$_1$, WRMn$_1$, RMd$_1$, WRMd$_1$, RMn$_2$, WRMn$_2$, RMd$_2$, WRMd$_2$), we have different detectors.

Again, as we mentioned in conventional detection, in practice we do not know the

true central point of the noise PSD matrices. Therefore, an estimate $\hat{\boldsymbol{M}}_\nu$ has to be made to replace $\boldsymbol{M}_\nu$ in Eq. (4.10). Employing the SWMA method, we can obtain the estimated central point $\hat{\boldsymbol{M}}_\nu$ by evaluating $2L$ PSD matrices in the neighbourhood of $\boldsymbol{P}_k$ with the assumption that all of those $2L$ matrices are from noise only samples. The estimation of noise centre can follow various algorithms that we developed in chapter 3. Then, the decision rule in practice becomes

$$ d_x(\boldsymbol{P}_k - \hat{\boldsymbol{M}}_\nu, \boldsymbol{0}) \underset{H_0}{\overset{H_1}{\gtrless}} r_\alpha \tag{4.11} $$

By changing the value of r_α, the ROC of different detectors can be obtained and a comparison of the performance can be made.

4.2.2   Choice of Decision Reference

When ED is chosen to be the distance measure, the two decision rules in Eqs. (4.8) and (4.10) are indentical because

$$ d_{\mathrm{E}}(\boldsymbol{P}_k, \boldsymbol{M}_\nu) = \sqrt{\mathrm{tr}[(\boldsymbol{P}_k - \boldsymbol{M}_\nu)(\boldsymbol{P}_k - \boldsymbol{M}_\nu)^H]} = d_{\mathrm{E}}(\boldsymbol{P}_k - \boldsymbol{M}_\nu, \boldsymbol{0}) \tag{4.12} $$

However, if we use RD (either d_{R_1} or d_{R_2}) in the two decision rules, then the results will be different since $d_{\mathrm{R}}(\boldsymbol{P}_k, \boldsymbol{M}_\nu) \neq d_{\mathrm{R}}(\boldsymbol{P}_k - \boldsymbol{M}_\nu, \boldsymbol{0})$. In the following, we will use d_{R_2} to demonstrate this point and explain the reason of choosing Eq. (4.10) as the decision rule. The cases of d_{R_1}, d_{WR_1} and d_{WR_2} can follow the similar illustrations.

Applying the expression of d_{R_2} to Eqs. (4.8) and (4.10), we have

$$d_{\mathrm{R}_2}(\boldsymbol{P}_k, \boldsymbol{M}_\nu) = \sqrt{\mathrm{tr}\boldsymbol{P}_k + \mathrm{tr}\boldsymbol{M}_\nu - 2\mathrm{tr}(\boldsymbol{P}_k^{1/2}\boldsymbol{M}_\nu^{1/2})} \tag{4.13}$$

$$d_{\mathrm{R}_2}(\boldsymbol{P}_k - \boldsymbol{M}_\nu, \boldsymbol{0}) = \sqrt{|\mathrm{tr}\boldsymbol{P}_k - \mathrm{tr}\boldsymbol{M}_\nu|} \tag{4.14}$$

Since we need to maintain the positivity of RD, the absolute value is taken in Eq. (4.14). Taking the Riemannian mean $\boldsymbol{C}$ of a group of noise PSD matrices as the central point, the comparison between $d_{\mathrm{R}_2}(\boldsymbol{P}, \boldsymbol{C})$ and $d_{\mathrm{R}_2}(\boldsymbol{P} - \boldsymbol{C}, \boldsymbol{0})$ is shown in Fig. 4.3 from which a distortion with the solid curve can be seen by shifting the reference point from $\boldsymbol{C}$ to $\boldsymbol{0}$. Similar observations and interpretations persist for the Riemannian median as the detection reference.
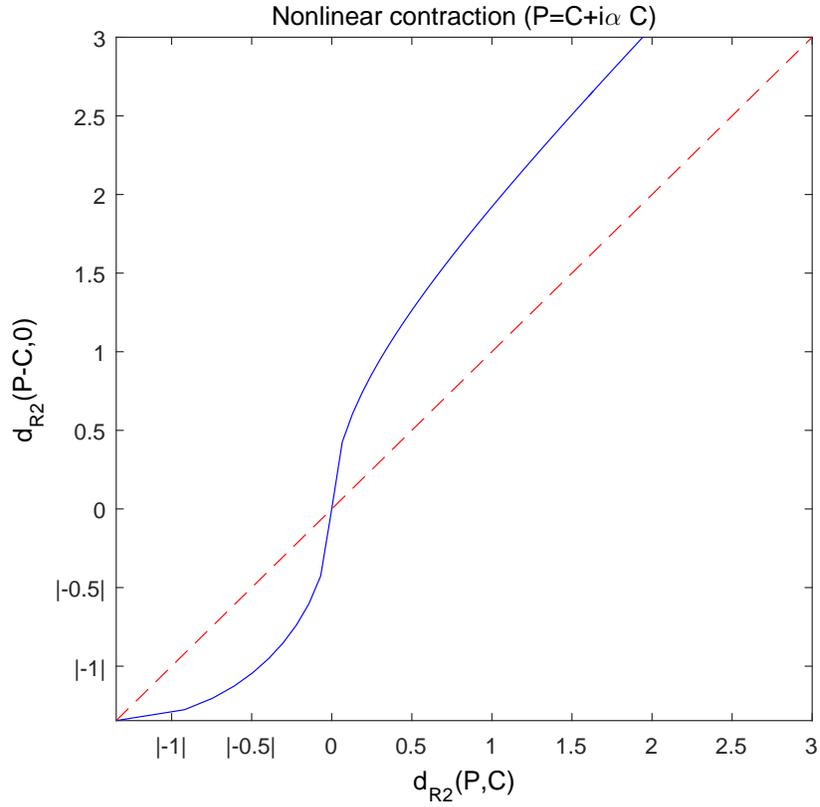
Figure 4.3: Distortion of RD measured from mean by reference shifting

The horizontal axis of Fig. 4.3 represents the increasing RD between different points and C, where the distance in the first quadrant is measured from C to the points beyond given by $P = C + i\alpha C$ with $0 < \alpha \leq 1$ being the step size and $i = 0, 1, \cdots$. The increasingly "negative" distance in the third quadrant is measured from the points extended in the opposite direction with $i = 0, -1, \cdots$. Since the RD is always positive, the negative signs merely represent the direction in which the distance is measured. On the other hand, the vertical axis is the corresponding RD measured from the shifted reference $\mathbf{0}$.

Examination of Fig. 4.3 reveals that the distance measured by shifting the reference to $\mathbf{0}$ has a high expansion rate in the neighbourhood of $\boldsymbol{C}$. As the point goes away from $\boldsymbol{C}$, in both positive and negative directions, the expansion rate decreases and ends up with a contraction. This tells us, by shifting the reference from $\boldsymbol{C}$ to $\mathbf{0}$, the distributions of the noise and signal PSD matrices are distorted with respect to the uneven distance contraction. Consequently, the probability of false alarm and the probability of missing will be affected, which is presented in Fig. 4.4 and Fig. 4.5.
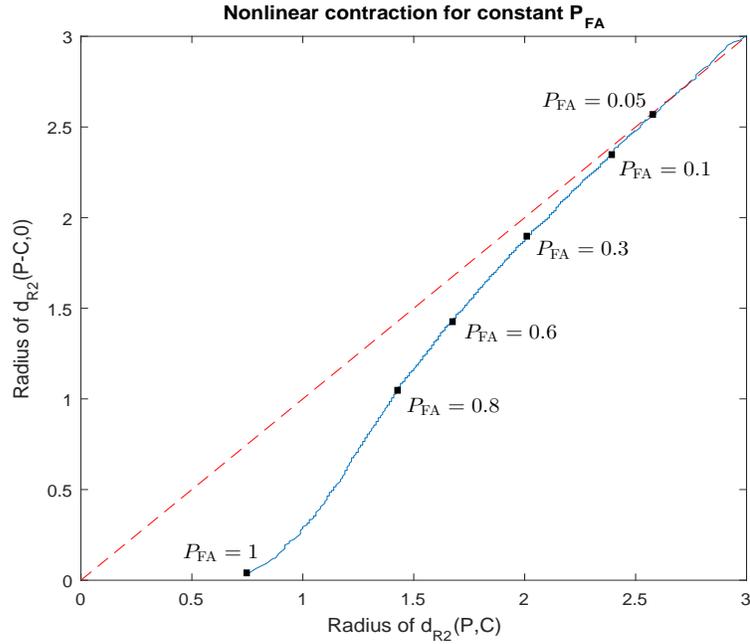
Figure 4.4: False alarm rates at corresponding distances
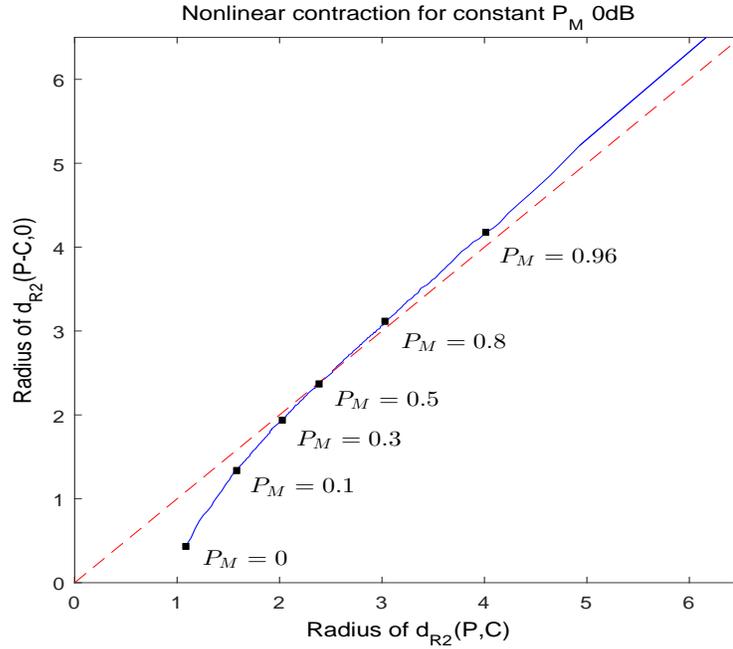
Figure 4.5: Missing rates at corresponding distances, SNR = 0dB

Fig. 4.4 shows the corresponding distance at the same false alarm rate with reference to C and $\mathbf{0}$. It can be seen that for a certain probability of false alarm, the distance measured by shifting reference to $\mathbf{0}$ is shorter. In other words, the same distance yields lower false alarm rate than when referenced to C. On the other hand, Fig. 4.5 shows the corresponding distance at the same missing rate with reference to C and $\mathbf{0}$. Although shifting reference to $\mathbf{0}$ increases the missing rate before the crossing point, the decrease in false alarm rate is much larger. Therefore, a conclusion can be obtained that the detection performance with shifted reference to $\mathbf{0}$ will be more desirable. Similar observations and interpretations persist if other RD measures are used in Eqs. (4.8) and (4.10). Hence, we will choose Eq. (4.10) (Eq. (4.11) in practice) as the detection rule in the following experiments.

4.2.3   Optimum Weighting Matrix for Signal Detection

As we mentioned earlier, the purpose of weighting a distance is to increase the efficiency of signal processing by utilizing the prior information to emphasize certain parts of feature matrices as well as deemphasize others. In chapter 2, we have seen how the different distance measures can be weighted. For signal detection, we aim to distinguish (signal+noise) from noise. The similarity between two feature PSD matrices $\boldsymbol{P}_m$ and $\boldsymbol{P}_n$ can be defined as the amount of correlation such that

$$\sigma(\boldsymbol{P}_m, \boldsymbol{P}_n) = \mathrm{tr}(\boldsymbol{P}_m^H \boldsymbol{P}_n) \qquad (4.15)$$

Suppose that in our prior knowledge, we have a collection of (signal+noise) PSD matrices and noise only PSD matrices. If we divide the collected matrices into two classes: (signal+noise) denoted by $\mathcal{S}$ and noise only denoted by $\mathcal{N}$, the optimum weighting matrix should maximize the correlation between similar classes and minimize the correlation between dissimilar classes. For the case of our detection, we need to determine whether a PSD matrix is (signal+noise) or noise only and the decision rule in Eq. (4.11) is made by judging the distance between the estimated noise matrix and the PSD matrix under test. Therefore, we need to seek for an optimum weighting matrix $\boldsymbol{W} = \Omega\Omega^H$ which minimizes the following objective function

$$F_o(\boldsymbol{\Omega}) = \mathrm{tr}(\boldsymbol{M}_{s\mathrm{W}}^{-1}\boldsymbol{M}_{\nu\mathrm{W}}) \qquad (4.16)$$

where $\boldsymbol{M}_{s\mathrm{W}}$ and $\boldsymbol{M}_{\nu\mathrm{W}}$ are the weighted central points of (signal+noise) and noise only PSD matrices, respectively. Since $\mathrm{tr}(\boldsymbol{A}^{-1}\boldsymbol{B}) \geq (\mathrm{tr}\boldsymbol{A})^{-1}(\mathrm{tr}\boldsymbol{B})$ [35], the upper

bound of Eq. (4.16) can be given by

$$F_o(\boldsymbol{\Omega}) = \text{tr}\left[(\boldsymbol{M}_{s\text{W}}\boldsymbol{M}_{s\text{W}})^{-1}(\boldsymbol{M}_{s\text{W}}\boldsymbol{M}_{\nu\text{W}})\right]$$

$$\geq \text{tr}(\boldsymbol{M}_{s\text{W}}\boldsymbol{M}_{s\text{W}})^{-1}\text{tr}(\boldsymbol{M}_{s\text{W}}\boldsymbol{M}_{\nu\text{W}}) \qquad (4.17)$$

Hence, minimizing $F_o(\boldsymbol{\Omega})$ is in fact minimizing the upper bound of the ratio of the correlation between central point of dissimilar classes to that of similar class.

When the central point refers to WRMn$_1$, WRMd$_1$ or WRMd$_2$, the optimum weighting matrix is difficult to be directly obtained from Eq. (4.16) since these central points can only be evaluated numerically based on the methods that assume the weighting matrix to be fixed. As a result, it requires an iteration process to find the optimum weighting in terms of these central points. On the other hand, when the central point refers to WRMn$_2$ which can be located without iterative procedure, the awkward expression of WRMn$_2$ involves the sum of square roots of matrices and thus makes the manipulation of the right side of Eq. (4.16) difficult as well. In any case, we need the solution of the optimum weighting for fixed central points for which the following theorem is given.

Theorem 5. *Suppose we have the objective function such that*

$$F_o = \text{tr}\left[(\boldsymbol{\Omega}^H\boldsymbol{\Pi}_s\boldsymbol{\Omega})^{-1}(\boldsymbol{\Omega}^H\boldsymbol{\Pi}_\nu\boldsymbol{\Omega})\right] \qquad (4.18)$$

where $\boldsymbol{\Pi}_s$ and $\boldsymbol{\Pi}_\nu$ are certain central points chosen from (signal+noise) and noise PSD matrices, respectively. If $\{\lambda_1, \cdots, \lambda_M\}$ and $\{\boldsymbol{u}_1, \cdots, \boldsymbol{u}_M\}$ are respectively the eigenvalues and eigenvectors of $\boldsymbol{\Pi}_s^{-1}\boldsymbol{\Pi}_\nu$, then the maximum and minimum value of F_o are achieved when $\boldsymbol{\Omega}_{\text{op}}$ is resptectively composed of the first K and the last K

eigenvectors of $\{\boldsymbol{u}_1, \cdots, \boldsymbol{u}_M\}$, *where* $K \leq M$.   □

Proof. The proof of Theorem 5 is shown in Appendix A.3.   ■

In general, the matrix $\boldsymbol{\Pi}_s^{-1}\boldsymbol{\Pi}_\nu$ are not Hermitian, thus the eigenvectors may not be orthonormal, i.e., $\boldsymbol{W} \neq \boldsymbol{I}_M$. However, in our test, $\boldsymbol{\Pi}_s^{-1}\boldsymbol{\Pi}_\nu$ is extremely close to Hermitian, i.e., the eigenvalues are real and the eigenvectors are orthonormal, if we choose $K = M$, then both equalities hold in Eq. (A.19), and the objective function becomes a constant. Therefore, the resulting weighting matrix becomes $\boldsymbol{W} = \boldsymbol{I}_M$, which has no effect. To maximize the possible effect of weighting, we choose $K = M - 1$. In this case, F_o reaches the maximum and minimum value depending on whether we construct $\boldsymbol{\Omega}_{\text{op}}$ using the eigenvectors corresponding to the K largest, or the K smallest eigenvalues. In Theorem 5, it presents the way to find the optimum weighting matrix given the (signal+noise) and noise central points. However, this process should be carried out together with that of locating the different central points according to d_{WR_1} and d_{WR_2}. Thus, combining Theorem 5 with the algorithms attained in chapter 3, we give the following algorithm to get the optimum weighting matrix.

General algorithm for finding optimum weighting matrix:

1. Set accuracy indicators ϵ_ν and ϵ_s.

2. For $i = 0$, initialize the central points:

   $\boldsymbol{\Pi}_s^{(i)} = \frac{1}{N_s} \sum_{\boldsymbol{P}_m \in \mathcal{S}} \boldsymbol{P}_m, \quad \boldsymbol{\Pi}_\nu^{(i)} = \frac{1}{N_\nu} \sum_{\boldsymbol{P}_n \in \mathcal{N}} \boldsymbol{P}_n.$

3. For $K = M$, use Theorem 5 together with $\boldsymbol{\Pi}_s^{(i)}$ and $\boldsymbol{\Pi}_\nu^{(i)}$ to obtain $\boldsymbol{\Omega}^{(i)}$.

4. Form the weighted signal group and the weighted noise group such that

$$\mathcal{S}_{\mathrm{W}}^{(i)} = \left\{ \boldsymbol{\Omega}^{(i)H} \boldsymbol{P}_m \boldsymbol{\Omega}^{(i)} \right\}_{\boldsymbol{P}_m \in \mathcal{S}}, \qquad \mathcal{N}_{\mathrm{W}}^{(i)} = \left\{ \boldsymbol{\Omega}^{(i)H} \boldsymbol{P}_n \boldsymbol{\Omega}^{(i)} \right\}_{\boldsymbol{P}_n \in \mathcal{N}}.$$

5. Lift these weighted PSD matrices to the Euclidean subspace using the respective mapping and apply the respective algorithm to locate the central points $\tilde{\boldsymbol{M}}_{s\mathrm{W}}^{(i)}$ and $\tilde{\boldsymbol{M}}_{\nu\mathrm{W}}^{(i)}$, where $\boldsymbol{M}$ can stand for the mean $\boldsymbol{C}$ or the median $\boldsymbol{\Gamma}$. Then project them back to the manifold to obtain $\boldsymbol{M}_{s\mathrm{W}}^{(i)}$ and $\boldsymbol{M}_{\nu\mathrm{W}}^{(i)}$.

6. Calculate the unweighted central points:

$$\boldsymbol{M}_{s\mathrm{Rw}}^{(i)} = \left(\boldsymbol{\Omega}^{(i)} \right)^{-H} \boldsymbol{M}_{s\mathrm{W}}^{(i)} \left(\boldsymbol{\Omega}^{(i)} \right)^{-1}, \qquad \boldsymbol{M}_{\nu\mathrm{Rw}}^{(i)} = \left(\boldsymbol{\Omega}^{(i)} \right)^{-H} \boldsymbol{M}_{\nu\mathrm{W}}^{(i)} \left(\boldsymbol{\Omega}^{(i)} \right)^{-1}.$$

7. If $d_{\mathrm{R}}(\boldsymbol{M}_{s\mathrm{Rw}}^{(i)}, \boldsymbol{M}_{s\mathrm{Rw}}^{(i-1)}) < \epsilon_s$ and $d_{\mathrm{R}}(\boldsymbol{M}_{\nu\mathrm{Rw}}^{(i)}, M_{\nu\mathrm{Rw}}^{(i-1)}) < \epsilon_\nu$, save $\boldsymbol{\Pi}_s = \boldsymbol{M}_{s\mathrm{Rw}}^{(i)}$ and $\boldsymbol{\Pi}_\nu = \boldsymbol{M}_{\nu\mathrm{Rw}}^{(i)}$, then go to Step 8. Otherwise, let $\boldsymbol{\Pi}_s^{(i+1)} = \boldsymbol{M}_{s\mathrm{Rw}}^{(i)}$ and $\boldsymbol{\Pi}_\nu^{(i+1)} = \boldsymbol{M}_{\nu\mathrm{Rw}}^{(i)}$. Let $i \to i+1$ and go back to Step 3.

8. Set $K = M - 1$ and employ Theorem 5 together with the saved $\boldsymbol{\Pi}_s$ and $\boldsymbol{\Pi}_\nu$ to obtain the optimum weighting matrix $\boldsymbol{\Omega}_{\mathrm{op}}$. ∎

The above algorithm can be employed to find the optimum weighting matrix for WRMn$_1$, WRMn$_2$, WRMd$_1$ and WRMd$_2$ by choosing the mapping for lifting and projecting according to the specific RD, together with applying the corresponding algorithm for the mean or median as the central point.

4.3   Simulation Results

In this section, we will examine and evaluate the performance of the hypothesis decision rule in Eq. (4.11) for different distance measures by computer experiments.

First, we transmit signals comprising of a set of sinusoids of which the ith signal is generated by the following model:

$$s_i = a_i \cos(2\pi f_i nT + \phi_i) + b_i \sin(2\pi f_i nT + \phi_i) \tag{4.19}$$

where $i = 1, \cdots, N_s$ with N_s being the number of signals, a_i and b_i are independent Gaussian random variables with zero mean and variance σ_s^2, f_i is the frequency of the ith signal, $n = 1, \cdots, N_T$ with N_T being the number of sampling points, T is the sampling period and ϕ_i is the random phase evenly distributed between $-\pi$ and π. The noise is generated as a sequence of white Gaussian with zero mean and σ_ν^2 variance. The signal-to-noise ratio (SNR) ρ in the following paragraphs refers to the SNR here measured by $10 \log(\sigma_s^2/\sigma_\nu^2)$. Then the transmitted signal, together with the additive noise, is received by a uniform linear array of sensors with angle of arrival $\theta = 60°$. Then the received signal is passed through the DFT analyser and the frequency-domain beamformer. Consequently, we obtain, for each frequency bin, the output of the beamformer which is the vector $\boldsymbol{\beta}_k$ given in Eq. (4.3). The outer product of $\boldsymbol{\beta}_k$ yields the PSD matrix $\boldsymbol{P}_k$ at the kth frequency bin which is positive semi-definite Hermitian. These PSD matrices are going to be processed for detection on the manifold. Furthermore, we need to collect a library of (signal+noise) and noise only PSD matrices in order to calculate a nominally optimum weighting matrix and prepare it for the weighting of the RD. Just as we mentioned in the previous section, we employ SWMA method to estimate the *average* of the noise matrices. Here, the term "average" is used to indicate either the mean or median. Based on the SWMA filter, we use the different algorithms in chapter 3 to estimate, for various distance measures, the average of the $2L$ neighbouring sample matrices. Notice that

for the first L matrices and the last L matrices of the beam, we also need to apply the SWMA method to make the detection, hence we add L noise matrices each to the beginning and end of the beam. For the PSD matrix in each frequency bin, from the originally first matrix to the originally last matrix, we use the decision rule in Eq. (4.11) to judge whether it is with signal or noise only. Then the probability of detection is calculated as

$$P_{\mathrm{D}} = \frac{N_d}{N_s} \tag{4.20}$$

where N_d is the counted number of successfully detected narrow-band signal matrices which has a larger distance from the estimated noise matrix than r_α, and N_s is the total number of the known signal matrices in the beam. On the other hand, the probability of false alarm is calculated as

$$P_{\mathrm{FA}} = \frac{N_{fa}}{N_n} \tag{4.21}$$

where N_{fa} is the counted number of noise matrices that have a larger distance from the estimated noise matrix than r_α, and N_n is the total number of noise matrices in the beam, i.e. $N_n = N_b - N_s$ with N_b being the number of frequency bins.

Now our experiments are going to substitute one by one, the various distance measures d_{E}, d_{R_1}, d_{R_2}, their weighted versions d_{WE}, d_{WR_1} and d_{WR_2}, together with their corresponding estimated means and medians into Eq. (4.11) so that we can examine the number of successful detections and the number of false alarms for a particular threshold radius r_α. Varying r_α from $0 \to \infty$ will yield the whole range of probabilities of false alarm and detection, giving us the ROC of the different detectors.

We will present our simulation results in the following parts.

4.3.1   ROC results for Reference Translation

We first verify the conclusion predicted in the previous section about the effect of reference translation by making a comparison of the performance between the two binary hypothesis testing rules in Eqs. (4.8) and (4.10). Again, we only show the use of RD d_{R_2} here because the uses of other RD measures achieve similar improvement of performance when shifting the reference from $\boldsymbol{C}$ to $\boldsymbol{0}$. The zoomed ROC performance of two decision rules in Eqs. (4.8) and (4.10) using $d_x = d_{R_2}$ and $\boldsymbol{M}_\nu = \boldsymbol{C}_{R_2}$ with SNR at 0dB and 3dB is shown in Fig. 4.6, respectively denoted by dash-dot, asterisk dash, solid and asterisk solid lines. The abscissas are probabilities of false alarm and the ordinates are the probabilities of detection.

   As we discussed before, the nonlinear contraction resulted from the translation of reference from $\boldsymbol{C}_{R_2}$ to $\boldsymbol{0}$ will lead to the loss in probability of false alarm. It can be observed from the figure that, for the same probability of false alarm, moving reference to $\boldsymbol{0}$ achieves a gain of over 2% in probability of detection for SNR at 0dB. And the improvement of performance for SNR at 3dB can approximately reach 2.5%. Other experiments including the use of d_{R_1} as the distance measure are also carried out and the similar observations are obtained.
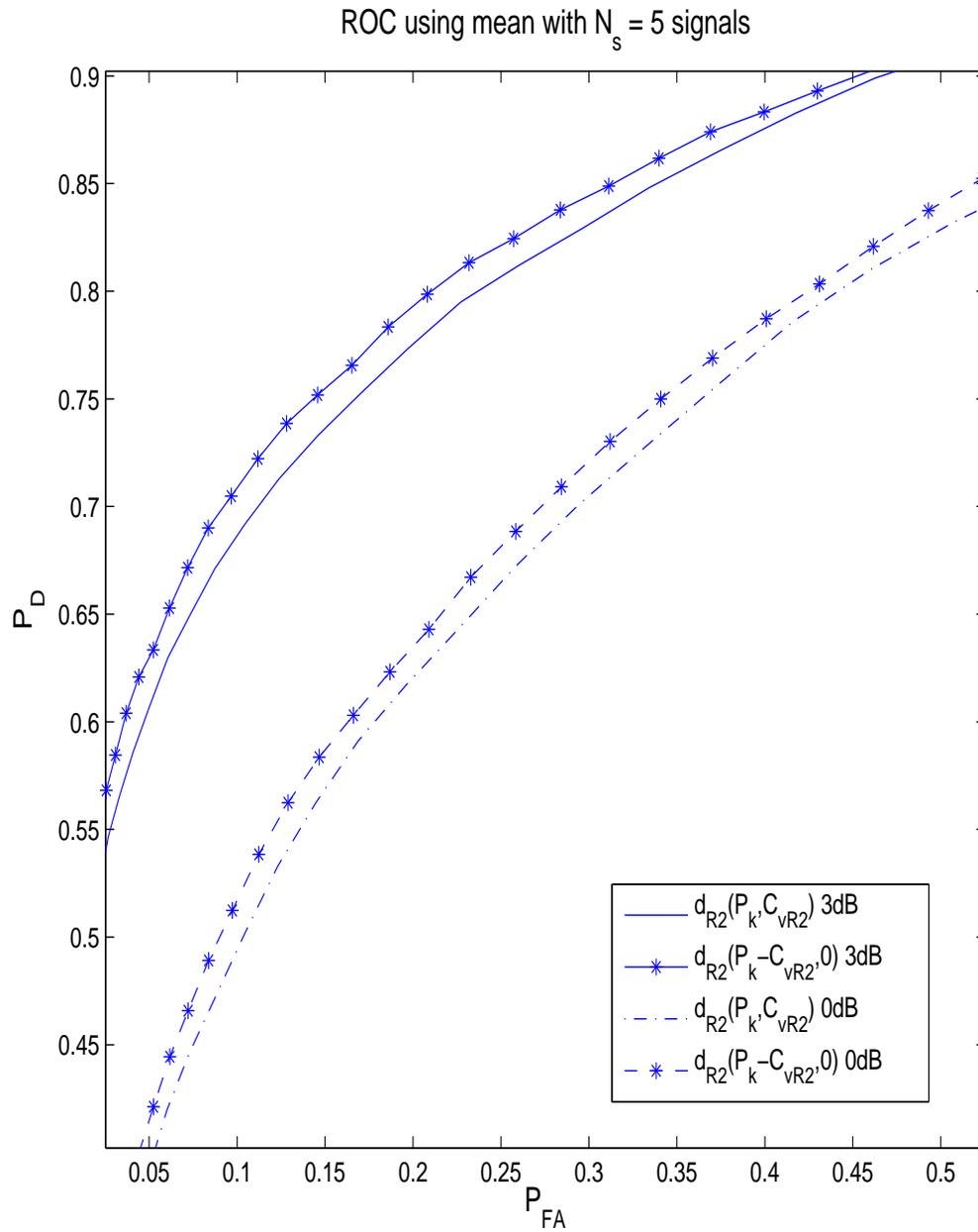
Figure 4.6: ROC of detectors with translated distance references

4.3.2   ROC results for Mean as the Detection Reference

Now, we show the cases in which the respective means are chosen to be the central points as the detection reference for all the different distance measures. Fig. 4.7 and Fig. 4.8 respectively show the ROC of different detectors at SNR of 0dB and 3dB having 5 narrow-band signals in the spectrum of $N_b = 500$ frequency bins. For a more clear view of the comparison, we enlarge the mid-part of the plots to show the different ROC performance according to different distance measures. The abscissas are probabilities of false alarm and the ordinates are the probabilities of detection. The ROC obtained by employing the decision rule in Eq. (4.11) using d_{E}, d_{R_1}, d_{R_2}, d_{WE}, d_{WR_1}, d_{WR_2} and the classical power spectrum are represented by solid, asterisk solid, dash-dot, plus sign solid, circle solid, upward-pointing triangle solid and dotted lines, respectively.

It can be seen in Fig. 4.7 and Fig. 4.8 that, as expected, using PSD matrices as the feature for detection yields obviously better results than using the power spectrum $Z_x(k\omega)$. In the case when $\rho = 0$dB, the improvement in probability of detection at the same probability of false alarm can be over 3%. Moreover, the use of optimum weighting further improves the performance of detection using unweighted PSD matrices by another 3%. The result of $\rho = 3$dB is more significant with the improvement in probability of detection being over 4% and the optimum weighting contributing another 4%. From the figures, it can be noticed that the performance using d_{R_1} and d_{R_2} are very close, so are the weighted RD d_{WR_1} and d_{WR_2}. Among all the detectors, d_{WR_1} and d_{WR_2} have the best performance. In the comparison between using ED and RD, the ED and weighted ED is marginally below the RD and weighted RD respectively.
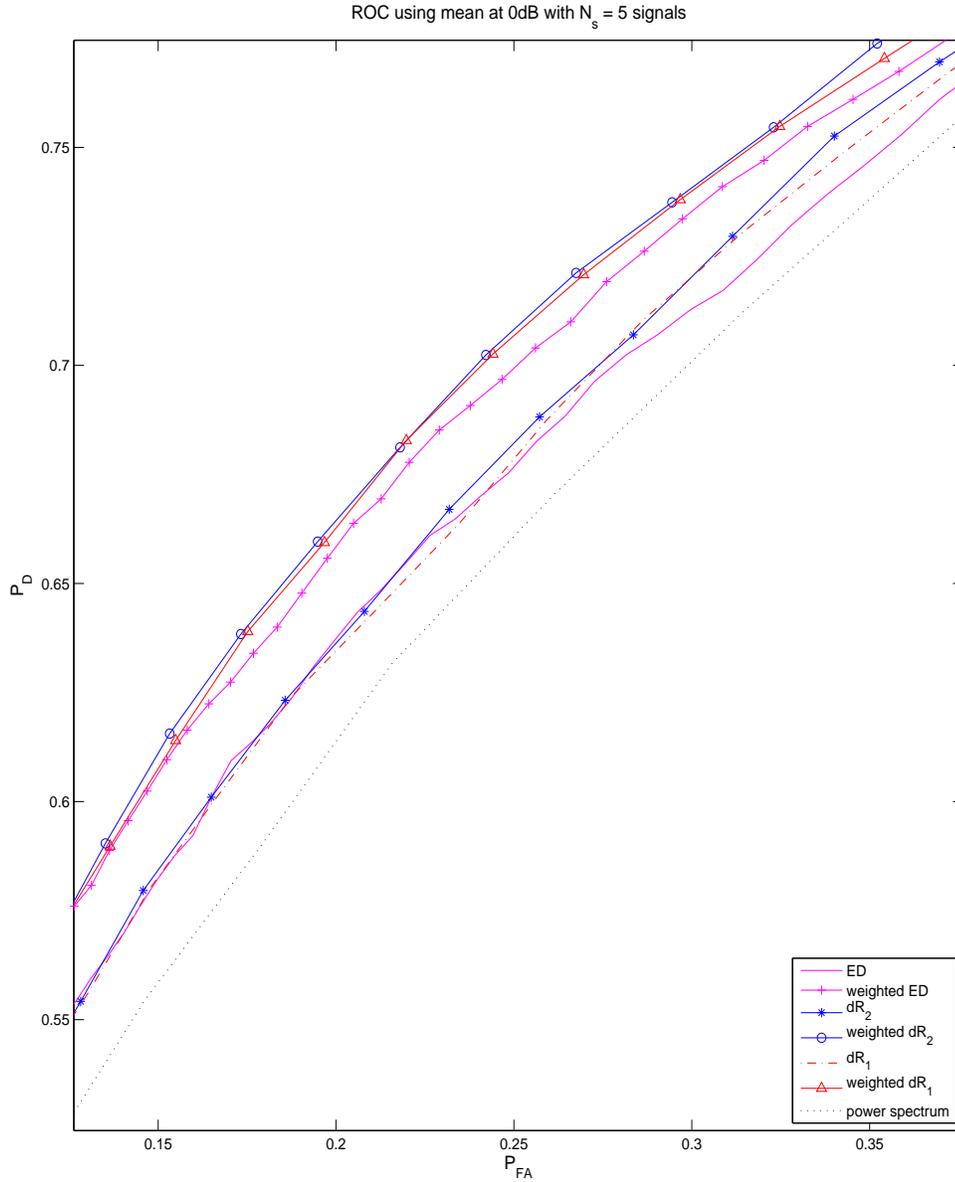
Figure 4.7: ROC of detectors using mean at 0dB with 5 signals
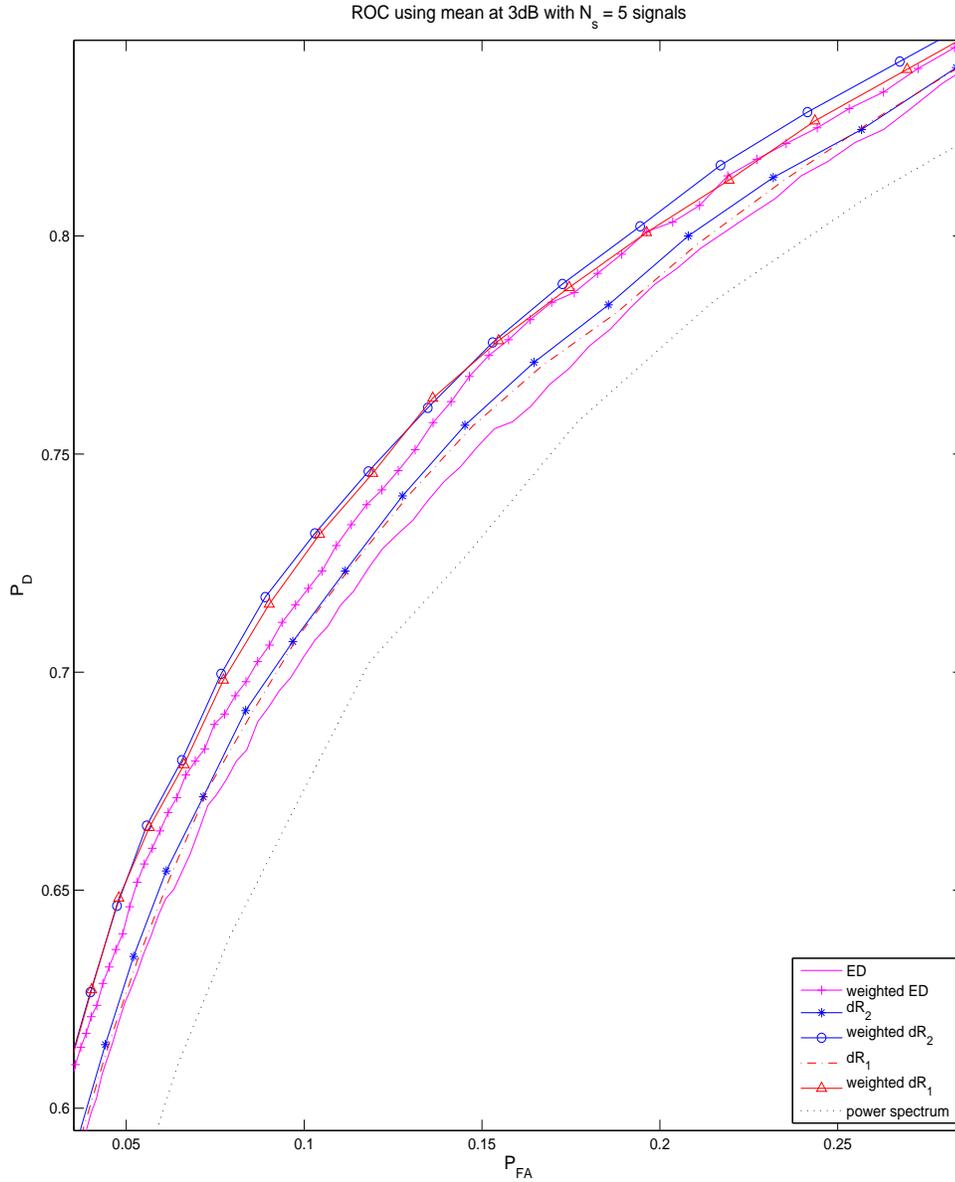
Figure 4.8: ROC of detectors using mean at 3dB with 5 signals

4.3.3   ROC results for Median as the Detection Reference

Then, we consider using the respective medians as the central points for all the different distance measures. Fig. 4.9 and Fig. 4.10 respectively show the ROC of different detectors at SNR of 0dB and 3dB under the same condition with using the mean as the detection feature. It can be seen that the ROC performance ranking is quite akin to that using mean. In order to facilitate the comparison of performance between mean and median, we plot Fig. 4.11 which presents the ROC of two detection features in the same figure at SNR of 3dB. Since the performance of using d_{R_1} and d_{R_2} are very close in all the cases, we only show the performance of d_{R_2} and d_{WR_2} to make the comparison clear. The ROC obtained by employing mean as the central point using d_E, d_{WE}, d_{R_2}, d_{WR_2} and power spectrum are represented by solid, dashed, plus sign solid, circle dashed, and dotted lines, respectively, and the ROC obtained using the corresponding distances but employing median as the central point are plotted with asterisk solid, diamond solid, square solid and upward-pointing triangle solid lines, respectively.

It is observed that in these cases when the number of signals N_s is small in comparison to the number of frequency bins N_b, the difference in performance between using the mean and the median is negligible for RD and weighted RD, whereas in the case of ED and weighted ED, the performance of median is lightly superior to that of mean as the detection feature. Detection of signals at other SNR are also carried out and similar performance ranking is observed.
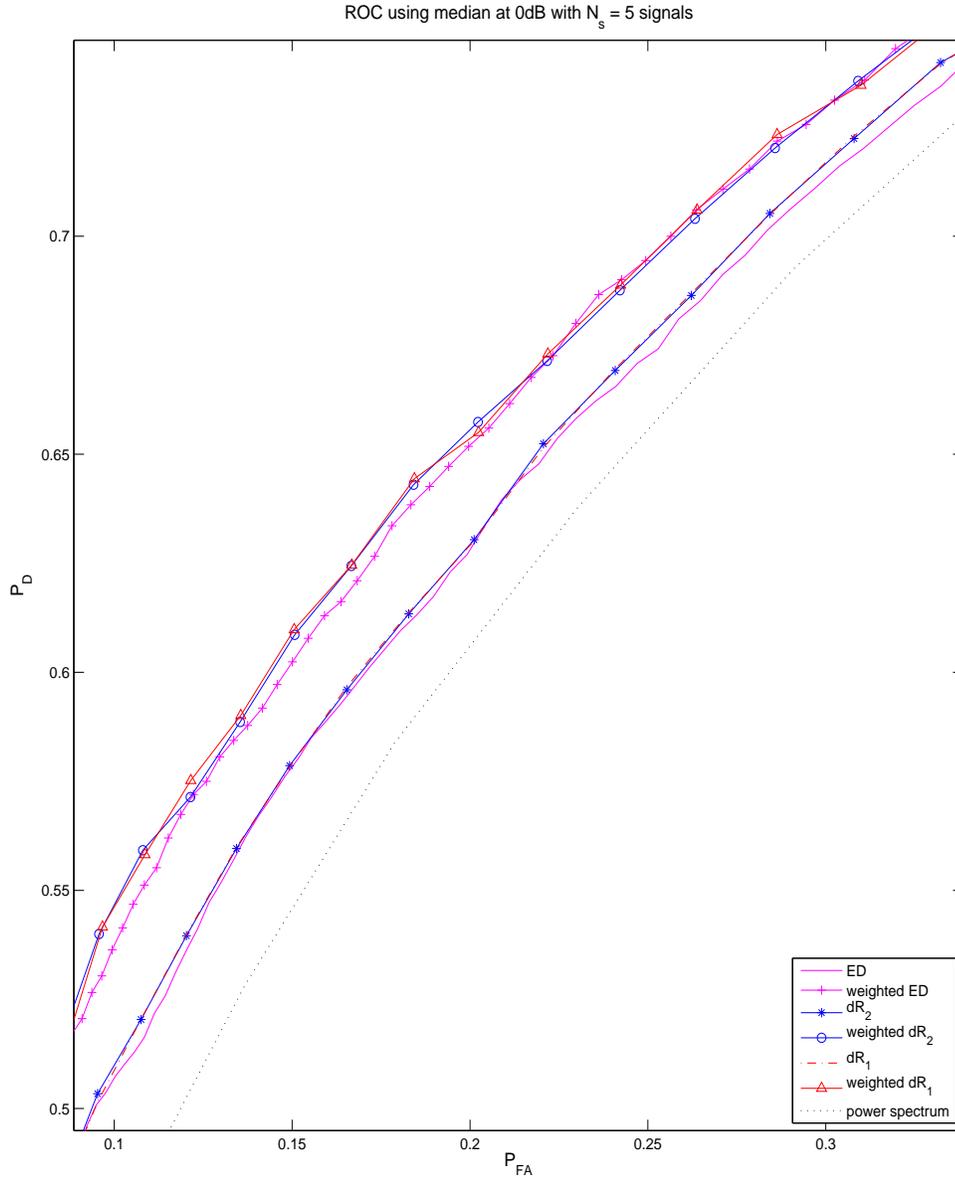
Figure 4.9: ROC of detectors using median at 0dB with 5 signals
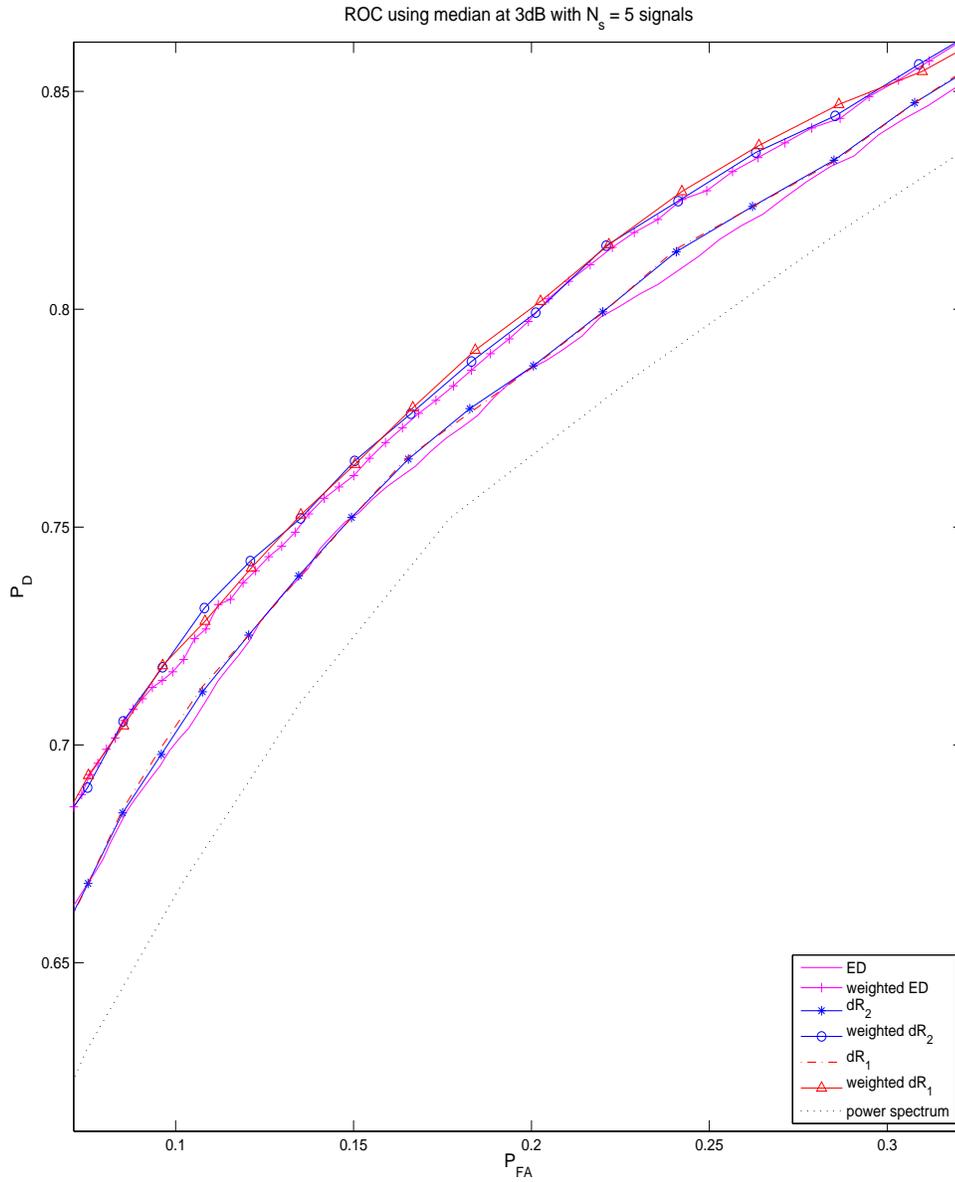
Figure 4.10: ROC of detectors using median at 3dB with 5 signals
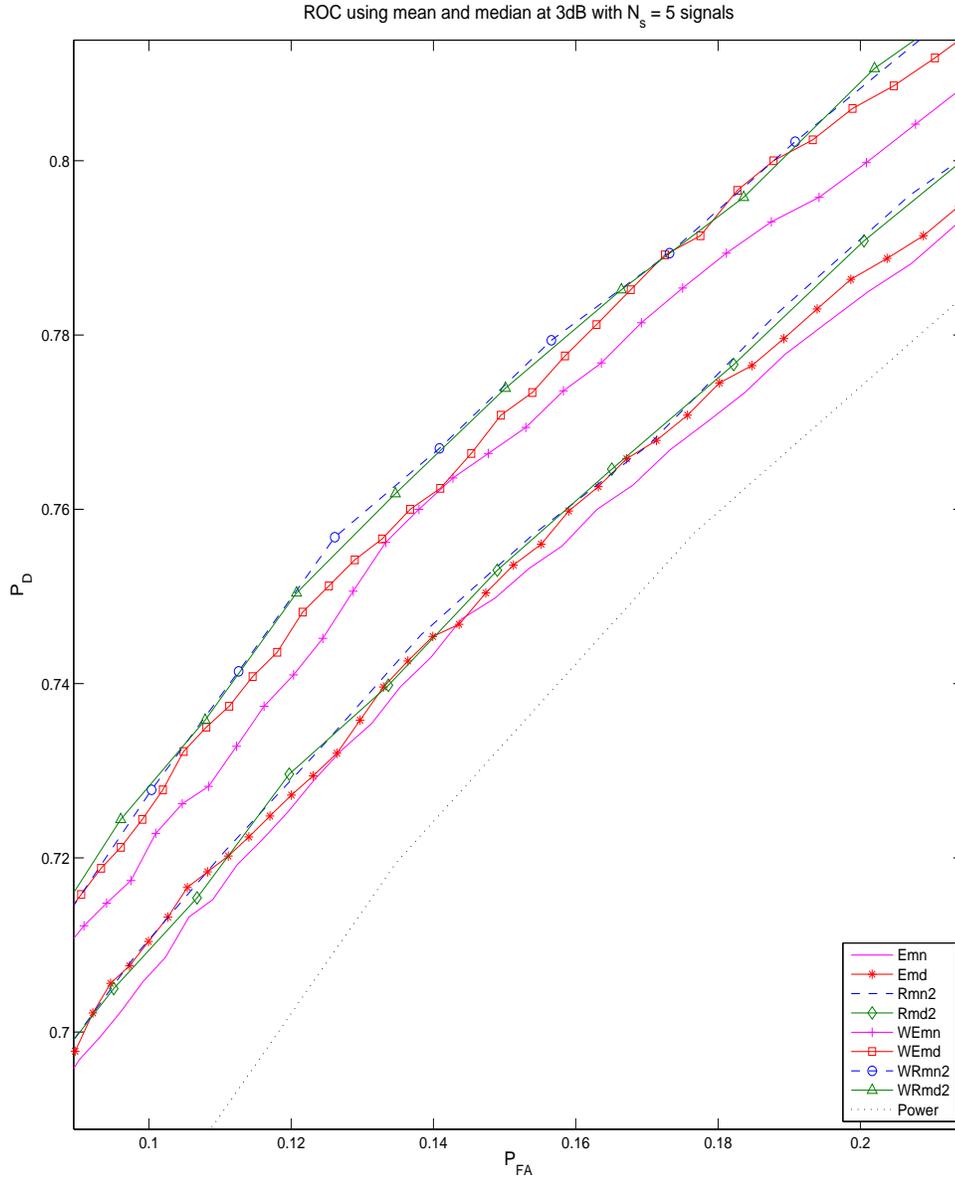
Figure 4.11: ROC performance comparison with 5 signals

4.3.4   Variation of Number of Signals N_s

In this section, we investigate the performance of the various methods under the condition of having large number of signals in the frequency bin. In other words, when the number of signals N_s present in the spectrum increases, some of them may fall into the split window, which introduces bias to the estimation of the noise centres. In these cases when interference of other signals are present, the use of the median instead of mean as the central point provides robustness to the estimation and detection. At $\rho = 3$dB, the performance of the unweighted and weighted distance with $N_s = 50$ are shown in Fig. 4.12 and Fig. 4.13 respectively. Again, due to the closeness in performance with d_{R_2} and d_{WR_2}, we choose not to show the performance of d_{R_1} and d_{WR_1} here. In Fig 4.12, presenting the ROC of $N_s = 5$ as the reference, we can see that increasing $N_s = 50$ makes the performance of every unweighted detector worse. Besides, for the detectors using the mean as the central point, the performance deteriorates more than those using the median. Similar robustness of using the median for the optimally weighted RD and ED detectors is shown in Fig. 4.13. In order to enlarge the effect of the signal interference, another comparison is shown with $N_s = 100$. In Fig.4.14, it can be observed that at a certain P_{FA}, compared to the case of $N_s = 5$, the P_D of RD detectors using median and mean approximately drop by 2% and 2.5% respectively; whereas the P_D of ED detectors using median and mean drop by 3% and 9% respectively. For the optimally weighted distance measures in Fig. 4.15, the performance of weighted RD detectors using median and mean approximately decline by 1% and 2% respectively, while the performance of weighted ED detectors using median and mean decline 2% and 10% respectively.
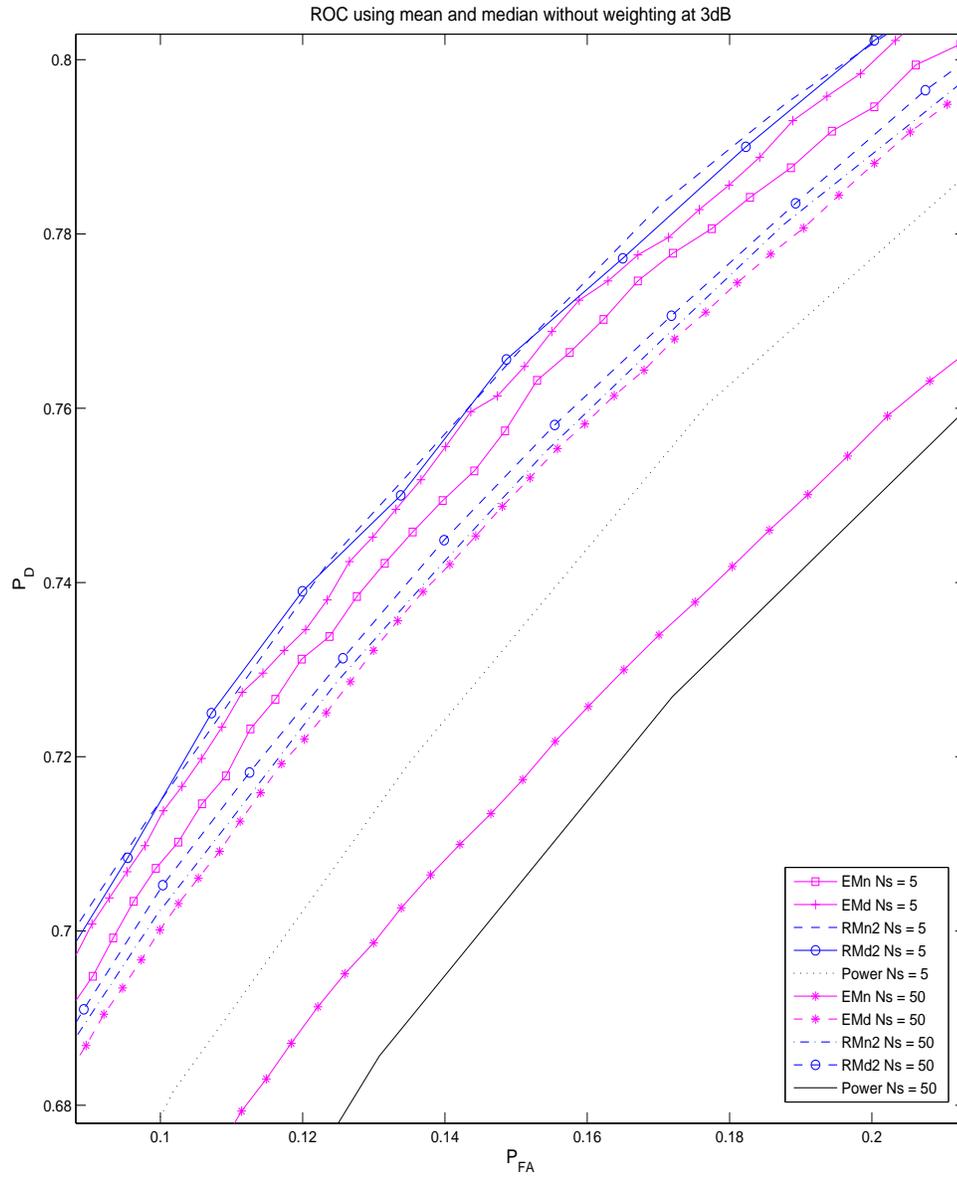
Figure 4.12: ROC comparison for unweighted distance $N_s = 5, N_s = 50$
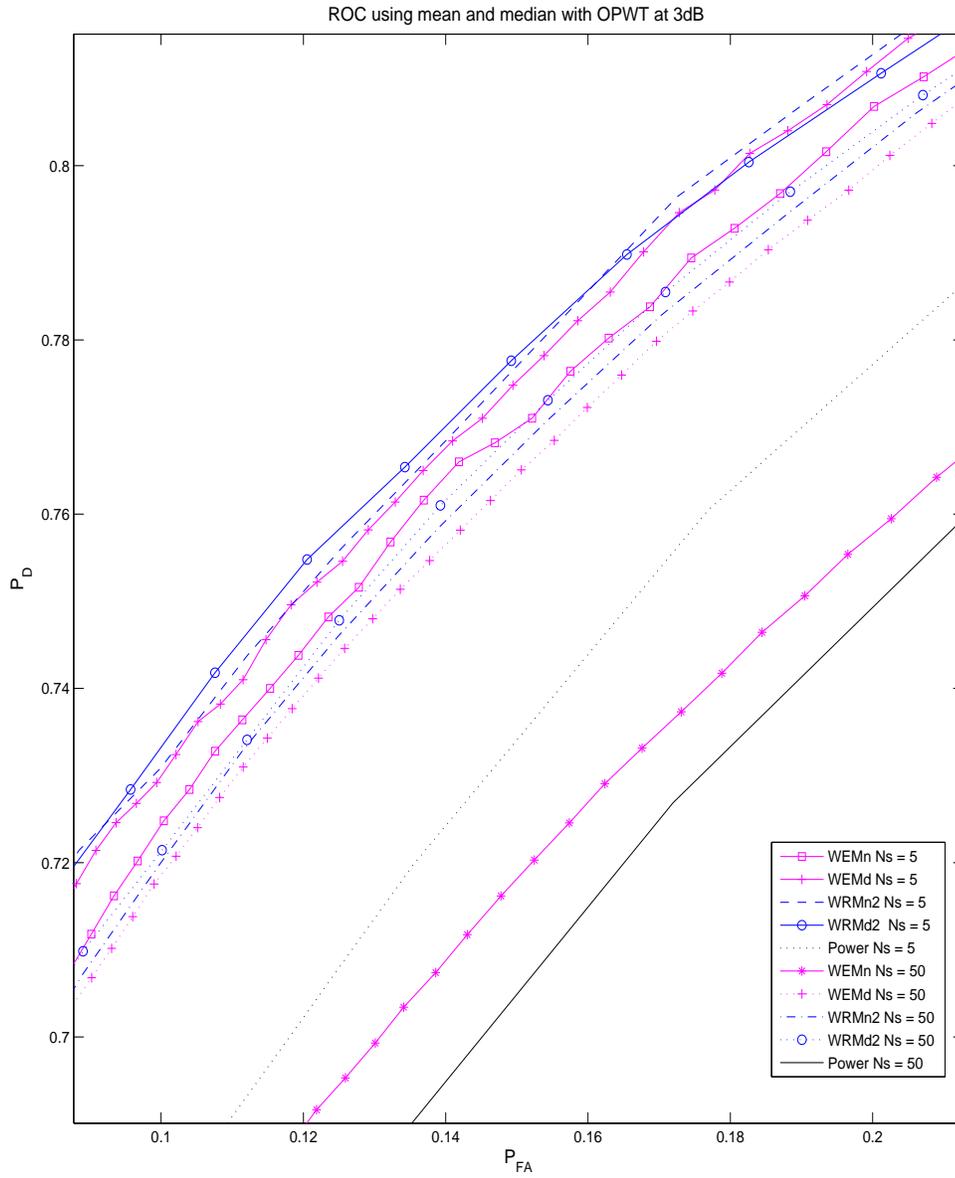
Figure 4.13: ROC comparison for optimally weighted distance $N_s = 5, N_s = 50$
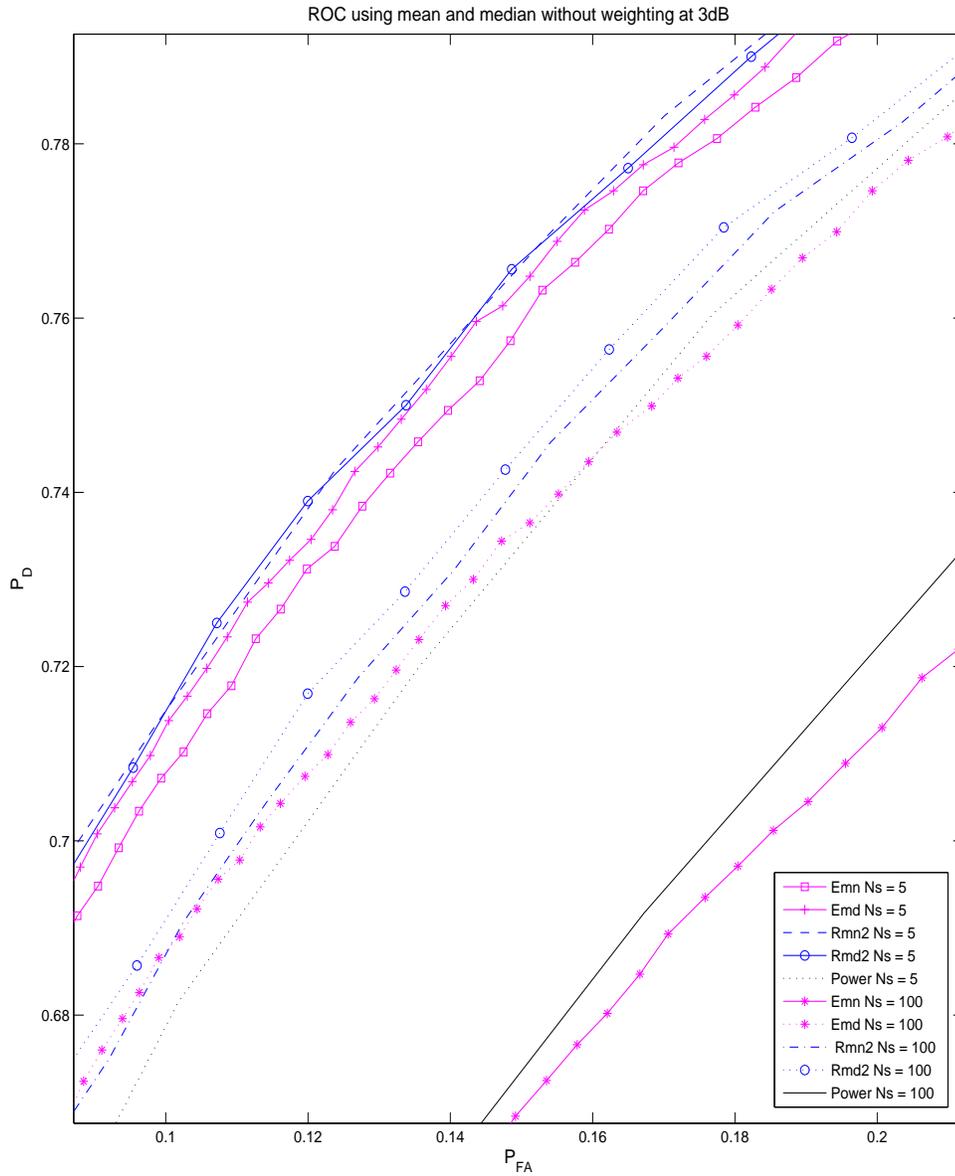
Figure 4.14: ROC comparison for unweighted distance $N_s = 5, N_s = 100$
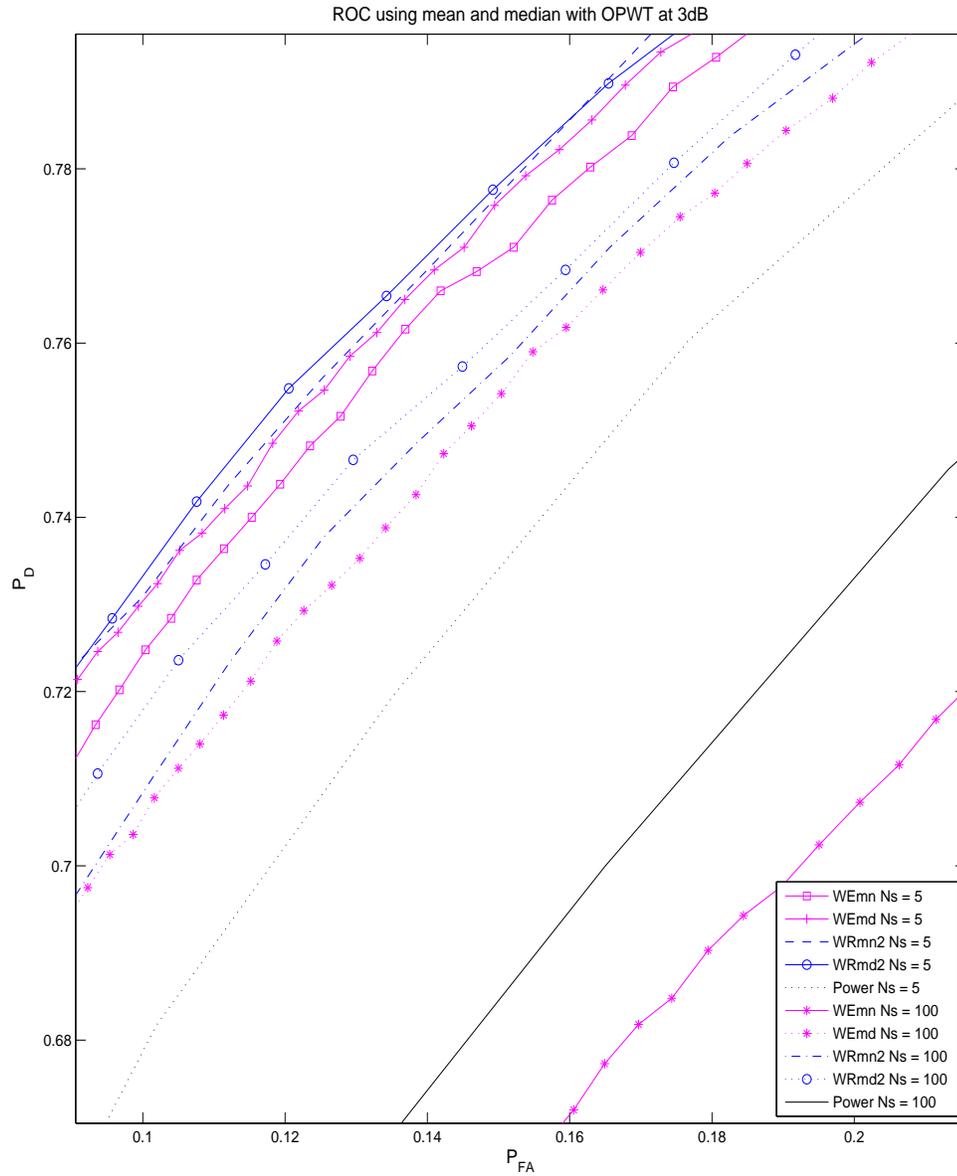
Figure 4.15: ROC comparison for optimally weighted distance $N_s = 5, N_s = 100$

From our simulations of detection performance, we can summarize our observations such that:

1. Due to the existence of nonlinear contraction with the change of reference, translating the reference to origin in the decision rule, comparing to the case when referenced at the central point, exhibits better detection performance.

2. In the narrowband signal detection, the use of PSD matrix, with various distance measures and its corresponding mean or median as the detection feature, yields superior performance over the use of power spectrum.

3. For both mean and median, the optimally weighted distance (d_{WR_1}, d_{WR_2} and d_{WE}) outperforms the unweighted distance (d_{R_1}, d_{R_2} and d_E). Particularly, the performance using d_{R_1} and d_{R_2} are very close and when the number of signals is small, they are both marginally better than that using d_E. Likewise, the performance using d_{WR_1} and d_{WR_2} are also quite close and both more desirable than that using d_{WE}.

4. In the case of small number of signals present in the spectrum range, the performance using mean and median as the central points are fairly comparable. Increasing the number of signals has negative effect on both the performance of mean and median due to the bias on the estimation of noise. However, the detectors using mean disimprove more significantly than those using the median. And the deterioration of ED is even more obvious especially for the Euclidean mean, the performance of which has been worse than the power spectrum.

Chapter 5

Conclusion

5.1 Summary of the Thesis

In this paper, we examined the use of the PSD matrix as the feature for signal processing and demonstrated the advantage of applying PSD matrix in the detection of narrowband sonar signals. We began by introducing the concept of the PSD matrix manifold. Since PSD matrices are structurally constrained, we considered how the distance between these matrices could be measured in terms of RD. Li and Wong have developed the closed-form expressions of RD and weighted RD for the PSD manifold in [8]. The idea of lifting the geodesic from manifold to the isometric Euclidean subspace was proposed, and then the optimization problem involving RD on the manifold can be performed in the isometric Euclidean subspace in terms of ED, which reduces the difficulty in manipulation. According to this concept, various algorithms were developed to locate the mean and the median for different RD. The EMn can be obtained quite straightforward since it is defined as the arithmetic mean; nevertheless, an iterative solution based on the steepest decent method was needed

to find the EMd. For the Riemannian mean, we developed an iterative algorithm to locate RMn_1, while RMn_2 has the closed-form expression. The RMd_1 and RMd_2 can be achieved by the similar algorithms to those of RMn_1 and RMn_2 but replacing the process in the Euclidian subspace with the algorithm of locating EMd. We then put the mean and median of the PSD matrices into the application of narrowband sonar signals detection. In order to maximize the correlation between noise PSD matrices while minimizing the correlation between noise and (signal+noise) PSD matrices, the concept of the optimum weighting matrix for RD was introduced and derived in terms of a collection of sample PSD matrices. Employing a strategy similar to Neyman-Pearson criterion, we set up the hypothesis decision rule such that if the distance between the observed PSD matrix and the estimated noise centre is larger than the threshold radius, we consider there is a signal at that frequency bin; otherwise, that frequency bin only contains noise. The decision rule was chosen by shifting the reference to origin with the consideration of the nonlinear distance distortion. The simulation results show that using the PSD matrices for detection provides superior performance compared to using the classical power spectrum. Furthermore, the use of median as a reference centre exhibits more robustness than that of mean when a large number of signals present in the spectrum. In summary, although the application of PSD matrices for signal processing requires special considerations in distance measures and also necessitates the development of various algorithms to facilitate the processing, the performance of detection has been extremely improved. That means the signal procesing on the PSD manifold appears to be an attractive alternative.

5.2   Future Work

In the application of sonar signals detection, we apply Eq. (4.11) as the decision rule. $\boldsymbol{P}_k$ is a rank-1 matrix formed by the outer product of an M-dimensional sample beam vector $\boldsymbol{\beta}_k$ for the kth frequency bin. On the other hand, the central matrix $\boldsymbol{M}_\nu$ (mean or median) is a full-rank $M \times M$ Hermitain and positive definite matrix representing the "sample average" of the noise PSD matrices. Since $\boldsymbol{P}_k$ is rank-1 and $\boldsymbol{M}_\nu$ is full-rank, we cannot find a $\boldsymbol{P}_k$ which is exactly equal to $\boldsymbol{M}_\nu$, and thus, $d_x(\boldsymbol{P}_k - \hat{\boldsymbol{M}}_\nu, \boldsymbol{0})$ cannot be zero. In the future, we can constitute $\boldsymbol{P}_k$ as a full-rank matrix such that the detection of whether $\boldsymbol{P}_k$ being a noise or (signal+noise) matrix may have a better effect. To construct the full-rank PSD matrices, we need to take more samples in time domain and divide them into I ($I \geq M$) sections. Then, each section is divided into M segments as what we did before. Hence, for each frequency bin, we have I PSD matrices denoted by $\boldsymbol{P}_{ki}$, $i = 1, \cdots, I$. Taking the average over these I matrices, the full-rank PSD matrix for each frequency bin can be obtained such that $\boldsymbol{P}_k = \frac{1}{I} \sum_i \boldsymbol{P}_{ki}$.

Furthermore, for the decision reference translation, we shifted the reference point to the origin such that we compared the distance $d_x(\boldsymbol{P}_k - \boldsymbol{M}_\nu, \boldsymbol{0})$ with a threshold. But is the null matrix the best reference point? In the thesis, we chose the null matrix as the reference point because $\boldsymbol{P}_k$ is either noise or containing signal. Then, we proved that shifting to origin has a better performance than referenced at central point. However, we did not theoretically prove the origin is the best reference point. What if we choose other matrices as the reference points? Further investigation is needed to find the best reference point.

Last but not the least, if we look at Eq. (4.14) which is the power difference

between $\boldsymbol{P}_k$ and $\boldsymbol{M}_\nu$ and then compare to the threshold, it is very similar to the classical detection method. But actually, they are different. For Eq. (4.14), the first term $\text{tr}\boldsymbol{P}_k$ is the signal power under test. However, for the second term $\text{tr}\boldsymbol{M}_\nu$, it is not the simple estimated noise power by taking the average of power over the neighbour windows. $\boldsymbol{M}_\nu$ is obtained by the various algorithms for locating the central points of PSD matrices on the manifold and thus $\text{tr}\boldsymbol{M}_\nu$ is the power of the noise central matrix $\boldsymbol{M}_\nu$. However, for the classical estimated noise power Z_ν, it is obtained by first taking the trace of the PSD matrices in the neighbourhood and then taking the average of these powers. Therefore, Eq. (4.14) is different from the classical method. I think in the future, if we are able to theoretically prove that $\text{tr}\boldsymbol{M}_\nu$ is a better estimation than Z_ν, it will be more convincing to support the advantage of PSD matrix being the detection feature.

Appendix A

A.1 Proof of Theorem 1

Proof. At the ith iteration, we have $\tilde{\boldsymbol{C}}^{(i)} = \left(\boldsymbol{C}^{(i)}\right)^{1/2}$, and $\tilde{\boldsymbol{P}}_n^{(i)} = \boldsymbol{P}_n^{1/2}\boldsymbol{U}_l^{(i)}\boldsymbol{U}_r^{(i)H}$ where $\boldsymbol{U}_l^{(i)}$ and $\boldsymbol{U}_r^{(i)}$ are the left and right singular vector matrices of $\boldsymbol{P}_n^{1/2}\tilde{\boldsymbol{C}}^{(i)}$ respectively (by F1). From F2, the ED between $\tilde{\boldsymbol{P}}_n^{(i)}$ and $\tilde{\boldsymbol{C}}^{(i)}$ is equal to the RD between $\boldsymbol{P}_n$ and $\boldsymbol{C}^{(i)}$ on the manifold, which is given by

$$d_{\mathrm{E}}^2(\tilde{\boldsymbol{P}}_n^{(i)}, \tilde{\boldsymbol{C}}^{(i)}) = d_{\mathrm{R}_1}^2(\boldsymbol{P}_n, \boldsymbol{C}^{(i)}) \tag{A.1}$$

Based on F3, by letting the new sample average $\tilde{\boldsymbol{C}}'^{(i)} = \frac{1}{N}\sum_{n=1}^{N}\tilde{\boldsymbol{P}}_n^{(i)}$ in Euclidean space, it can be ensured that the sum of the squared ED from all $\tilde{\boldsymbol{P}}_n^{(i)}$ to $\tilde{\boldsymbol{C}}'^{(i)}$ is minimized such that

$$\sum_{n=1}^{N} d_{\mathrm{E}}^2(\tilde{\boldsymbol{P}}_n^{(i)}, \tilde{\boldsymbol{C}}'^{(i)}) \leq \sum_{n=1}^{N} d_{\mathrm{E}}^2(\tilde{\boldsymbol{P}}_n^{(i)}, \tilde{\boldsymbol{C}}^{(i)}) \tag{A.2}$$

Now, we can calculate the new Riemannian mean on the manifold by $\boldsymbol{C}^{(i+1)} = \tilde{\boldsymbol{C}}'^{(i)}\tilde{\boldsymbol{C}}'^{(i)H}$ and repeat the process such that $\tilde{\boldsymbol{C}}^{(i+1)} = \left(\boldsymbol{C}^{(i+1)}\right)^{1/2}$ and $\tilde{\boldsymbol{P}}_n^{(i+1)} =$

$\boldsymbol{P}_n^{1/2}\boldsymbol{U}_l^{(i+1)}\boldsymbol{U}_r^{(i+1)H}$. Therefore,

$$d_{\mathrm{E}}^2(\tilde{\boldsymbol{P}}_n^{(i+1)}, \tilde{\boldsymbol{C}}^{(i+1)}) = d_{\mathrm{R}_1}^2(\boldsymbol{P}_n, \boldsymbol{C}^{(i+1)}) \tag{A.3}$$

Since the RD between $\boldsymbol{P}_n$ and $\boldsymbol{C}^{(i+1)}$ is the shortest path length at the moment, the ED between the corresponding lifted points $\tilde{\boldsymbol{P}}_n^{(i+1)}$ and $\tilde{\boldsymbol{C}}^{(i+1)}$ is also the shortest straight line, giving

$$\sum_{n=1}^{N} d_{\mathrm{E}}^2(\tilde{\boldsymbol{P}}_n^{(i+1)}, \tilde{\boldsymbol{C}}^{(i+1)}) \leq \sum_{n=1}^{N} d_{\mathrm{E}}^2(\tilde{\boldsymbol{P}}_n^{(i)}, \tilde{\boldsymbol{C}}'^{(i)}) \tag{A.4}$$

Again, the sample average of all $\{\tilde{\boldsymbol{P}}_n^{(i+1)}\}$ is presented by $\tilde{\boldsymbol{C}}'^{(i+1)} = \frac{1}{N}\sum_{n=1}^{N}\tilde{\boldsymbol{P}}_n^{(i+1)}$. Together with Eqs. (A.2) and (A.4), we have the inequalities as follows.

$$\sum_{n=1}^{N} d_{\mathrm{E}}^2(\tilde{\boldsymbol{P}}_n^{(i+1)}, \tilde{\boldsymbol{C}}'^{(i+1)}) \leq \sum_{n=1}^{N} d_{\mathrm{E}}^2(\tilde{\boldsymbol{P}}_n^{(i+1)}, \tilde{\boldsymbol{C}}^{(i+1)})$$
$$\leq \sum_{n=1}^{N} d_{\mathrm{E}}^2(\tilde{\boldsymbol{P}}_n^{(i)}, \tilde{\boldsymbol{C}}'^{(i)}) \leq \sum_{n=1}^{N} d_{\mathrm{E}}^2(\tilde{\boldsymbol{P}}_n^{(i)}, \tilde{\boldsymbol{C}}^{(i)}) \tag{A.5}$$

where the first inequality is from F3.

When $i \to \infty$, we have

$$\lim_{i \to \infty} \sum_{n=1}^{N} d_{\mathrm{E}}^2(\tilde{\boldsymbol{P}}_n^{(i)}, \tilde{\boldsymbol{C}}'^{(i)}) = \lim_{i \to \infty} \sum_{n=1}^{N} d_{\mathrm{E}}^2(\tilde{\boldsymbol{P}}_n^{(i)}, \tilde{\boldsymbol{C}}^{(i)}) \tag{A.6}$$

That means at this time, $\tilde{\boldsymbol{C}}'^{(\infty)} = \tilde{\boldsymbol{C}}^{(\infty)}$. In other words, further application of the algorithm will repeatedly yield the same minimum value of $\sum_{n=1}^{N} d_{\mathrm{E}}^2(\tilde{\boldsymbol{P}}_n^{(\infty)}, \tilde{\boldsymbol{C}}^{(\infty)})$. Thus, from Eq. (A.1), we can say that $\sum_{n=1}^{N} d_{\mathrm{R}_1}^2(\boldsymbol{P}_n^{(\infty)}, \boldsymbol{C}^{(\infty)})$ is also minimized. Thus, because of the isometry between $\mathcal{M}$ and $\mathcal{U}_{\mathcal{H}}$, $\lim_{i \to \infty} \boldsymbol{C}^{(i)} = \boldsymbol{C}_{\mathrm{R}_1}$. ∎

A.2   Proof of Convexity

A.2.1   Convexity of the Sum of Squared RD d_{R_2}

In order to check the convexity of the objective function $\sum_{n=1}^{N} d_{\mathrm{R}_2}^2 (\boldsymbol{P}_n, \boldsymbol{C})$ in Eq. (3.16), it is equivalent to check the convexity of the following function because summation does not change the convexity.

$$g_{\mathrm{r}}(\boldsymbol{C}) = d_{\mathrm{R}_2}^2(\boldsymbol{P}_n, \boldsymbol{C}) = \mathrm{tr}\boldsymbol{P}_n + \mathrm{tr}\boldsymbol{C} - 2\mathrm{tr}(\boldsymbol{P}_n^{1/2}\boldsymbol{C}^{1/2}) \tag{A.7}$$

The first derivative of $g_{\mathrm{r}}(\boldsymbol{C})$ is given by

$$\frac{d}{d\boldsymbol{C}}g_{\mathrm{r}}(\boldsymbol{C}) = \boldsymbol{I} - \boldsymbol{P}_n^{1/2}\boldsymbol{C}^{-1/2} \tag{A.8}$$

Then, the second derivative of $g_{\mathrm{r}}(\boldsymbol{C})$ is given by

$$\frac{d^2}{d\boldsymbol{C}^2}g_{\mathrm{r}}(\boldsymbol{C}) = 1/2\boldsymbol{P}_n^{1/2}\boldsymbol{C}^{-3/2} \succeq 0 \tag{A.9}$$

where the fact [36] that $\frac{\partial \mathrm{tr}(H(\boldsymbol{X}))}{\partial \boldsymbol{X}} = h(\boldsymbol{X})^H$ with $H(\boldsymbol{X})$ being a differentiable function and $h(\cdot)$ being the scalar derivative of $H(\cdot)$ has been used. Since $\boldsymbol{P}_n$ and $\boldsymbol{C}$ are both positive semi-definite Hermitian matrices, the eigenvalues of $\boldsymbol{P}_n^{1/2}\boldsymbol{C}^{-3/2}$ are also non-negative such that the second derivative of $g_{\mathrm{r}}(\boldsymbol{C})$ is positive semi-definite. Thus, the objective function in Eq. (3.16) is convex. Similar proof of convexity for the objective function in Eq. (3.8) persists and thus the RMn_1 is also the global optimum.

A.2.2   Convexity of the Sum of Euclidean Distance

The convexity of the objective function $\sum_{n=1}^{N} d_{\mathrm{E}}(\boldsymbol{P}_n, \boldsymbol{\Gamma})$ in Eq. (3.27) is in accordance with the following function:

$$f_{\mathrm{e}}(\boldsymbol{\Gamma}) = d_{\mathrm{E}}(\boldsymbol{P}_n, \boldsymbol{\Gamma}) = \sqrt{\operatorname{tr}\left[(\boldsymbol{P}_n - \boldsymbol{\Gamma})(\boldsymbol{P}_n - \boldsymbol{\Gamma})^H\right]} = \|\boldsymbol{P}_n - \boldsymbol{\Gamma}\| = \|\boldsymbol{\Gamma} - \boldsymbol{P}_n\| \quad \text{(A.10)}$$

where $\boldsymbol{P}_n$ and $\boldsymbol{\Gamma}$ are positive semi-definite Hermitian matrices. The first derivative of $f_{\mathrm{e}}(\boldsymbol{\Gamma})$ is given by

$$\frac{d}{d\boldsymbol{\Gamma}} f_{\mathrm{e}}(\boldsymbol{\Gamma}) = \frac{\boldsymbol{\Gamma} - \boldsymbol{P}_n}{\|\boldsymbol{\Gamma} - \boldsymbol{P}_n\|} \quad \text{(A.11)}$$

And the second derivative of $f_{\mathrm{e}}(\boldsymbol{\Gamma})$ is given by

$$\begin{aligned}
\frac{d^2}{d\boldsymbol{\Gamma}^2} f_{\mathrm{e}}(\boldsymbol{\Gamma}) &= \frac{\|\boldsymbol{\Gamma} - \boldsymbol{P}_n\|\boldsymbol{I} - (\boldsymbol{\Gamma} - \boldsymbol{P}_n)\frac{\boldsymbol{\Gamma} - \boldsymbol{P}_n}{\|\boldsymbol{\Gamma} - \boldsymbol{P}_n\|}}{\|\boldsymbol{\Gamma} - \boldsymbol{P}_n\|^2} \\
&= \frac{\|\boldsymbol{\Gamma} - \boldsymbol{P}_n\|^2 \boldsymbol{I} - (\boldsymbol{\Gamma} - \boldsymbol{P}_n)^2}{\|\boldsymbol{\Gamma} - \boldsymbol{P}_n\|^3} \\
&= \frac{\operatorname{tr}\left[(\boldsymbol{\Gamma} - \boldsymbol{P}_n)(\boldsymbol{\Gamma} - \boldsymbol{P}_n)^H\right] \boldsymbol{I} - (\boldsymbol{\Gamma} - \boldsymbol{P}_n)(\boldsymbol{\Gamma} - \boldsymbol{P}_n)^H}{\|\boldsymbol{\Gamma} - \boldsymbol{P}_n\|^3} \quad \text{(A.12)}
\end{aligned}$$

Assume that $\boldsymbol{A} = (\boldsymbol{\Gamma} - \boldsymbol{P}_n)(\boldsymbol{\Gamma} - \boldsymbol{P}_n)^H$, then the numerator of Eq. (A.12) becomes

$$\begin{aligned}
\operatorname{tr}(\boldsymbol{A})\boldsymbol{I} - \boldsymbol{A} &= \left(\sum_{i=1}^{M} \lambda_i\right) \boldsymbol{I} - \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^H \\
&= \left(\sum_{i=1}^{M} \lambda_i\right) \boldsymbol{U}\boldsymbol{U}^H - \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^H \\
&= \boldsymbol{U}\bar{\boldsymbol{\Lambda}}\boldsymbol{U}^H \succeq 0 \quad \text{(A.13)}
\end{aligned}$$

where $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \cdots, \lambda_M)$ and $\boldsymbol{U}$ are respectively the eigenvalue and eigenvector matrices of $\boldsymbol{A}$, $\bar{\mathbf{\Lambda}} = \mathrm{diag}\left(\sum_{i\neq 1}\lambda_i, \cdots, \sum_{i\neq M}\lambda_i\right)$ is the eigenvalue matrix of the numerator in Eq. (A.12). Since $\boldsymbol{A} = (\mathbf{\Gamma} - \boldsymbol{P}_n)(\mathbf{\Gamma} - \boldsymbol{P}_n)^H$ is a positive semi-definite Hermitian matrix with $\lambda_i \geq 0$, $i = 1, \cdots, M$, all the eigenvalues in $\bar{\mathbf{\Lambda}}$ are non-negative and thus the result of Eq. (A.12) can be proved to be positive semi-definite. Therefore, it can be proved that the objective function in Eq. (3.27) is convex.

A.2.3   Convexity of the Sum of RD d_{R_2}

Again, let us check the convexity of the objective function in Eq. (3.41) without summation as below.

$$f_{\mathrm{r}}(\mathbf{\Gamma}) = d_{\mathrm{R}_2}(\boldsymbol{P}_n, \mathbf{\Gamma}) = \sqrt{\mathrm{tr}\boldsymbol{P}_n + \mathrm{tr}\mathbf{\Gamma} - 2\mathrm{tr}(\boldsymbol{P}_n^{1/2}\mathbf{\Gamma}^{1/2})} \qquad (A.14)$$

The first derivative of $f_{\mathrm{r}}(\mathbf{\Gamma})$ is given by

$$\frac{d}{d\mathbf{\Gamma}}f_{\mathrm{r}}(\mathbf{\Gamma}) = \frac{\boldsymbol{I} - \boldsymbol{P}_n^{1/2}\mathbf{\Gamma}^{-1/2}}{2f_{\mathrm{r}}(\mathbf{\Gamma})} \qquad (A.15)$$

And the second derivative of $f_{\mathrm{r}}(\mathbf{\Gamma})$ is given by

$$\begin{aligned}
\frac{d^2}{d\mathbf{\Gamma}^2}f_{\mathrm{r}}(\mathbf{\Gamma}) &= \frac{1}{2}\frac{\left(\frac{1}{2}\boldsymbol{P}_n^{1/2}\mathbf{\Gamma}^{-3/2}\right)f_{\mathrm{r}}(\mathbf{\Gamma}) - \left(\boldsymbol{I} - \boldsymbol{P}_n^{1/2}\mathbf{\Gamma}^{-1/2}\right)\frac{\left(\boldsymbol{I}-\boldsymbol{P}_n^{1/2}\mathbf{\Gamma}^{-1/2}\right)}{2f_{\mathrm{r}}(\mathbf{\Gamma})}}{f_{\mathrm{r}}^2(\mathbf{\Gamma})} \\
&= \frac{\boldsymbol{P}_n^{1/2}\mathbf{\Gamma}^{-3/2}}{4f_{\mathrm{r}}(\mathbf{\Gamma})} - \frac{\left(\boldsymbol{I} - \boldsymbol{P}_n^{1/2}\mathbf{\Gamma}^{-1/2}\right)^2}{4f_{\mathrm{r}}^3(\mathbf{\Gamma})} \qquad (A.16)
\end{aligned}$$

It is hard to tell whether the Hessian matrix in Eq. (A.16) is positive semi-definite or not. Therefore, we first examine its scalar form such that

$$f_{\mathrm{r}}(x) = d_{\mathrm{R}_2}(p, x) = \sqrt{p + x - 2p^{1/2}x^{1/2}} \tag{A.17}$$

Assume $p = 1$, Eq. (A.17) becomes

$$f_{\mathrm{r}}(x) = \sqrt{1 + x - 2x^{1/2}} = \left|\sqrt{x} - 1\right| \tag{A.18}$$

Obviously, when $0 \leq x < 1$, Eq. (A.18) is convex; when $x > 1$, it is concave. Since Eq. (A.14) is not convex in scalar form, it cannot be convex for matrices. Moreover, for the objective function in Eq. (3.36), it can be proved to be non-convex by the similar procedure.

A.3   Proof of Theorem 5

Proof. First, we need the following famous inequality [37]:

For $\boldsymbol{\Xi}$ being an $M \times M$ positive definite matrix with eigenvalues $\mu_1 \geq \cdots \geq \mu_M$ and associated orthonormal eigenvectors $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_M$, let $\boldsymbol{X}$ be an $M \times K, K \leq M$ matrix such that $\boldsymbol{X}^H\boldsymbol{X} = \boldsymbol{I}_K$. Then

$$\sum_{i=M-K+1}^{M} \mu_i \leq \mathrm{tr}(\boldsymbol{X}^H\boldsymbol{\Xi X}) \leq \sum_{i=1}^{K} \mu_i \tag{A.19}$$

The maximum and minimum values of $\mathrm{tr}(\boldsymbol{X}^H\boldsymbol{\Xi X})$ are reached when $\boldsymbol{X}$ is comprised respectively of the first K and the last K eigenvectors of $\{\boldsymbol{v}_1, \cdots, \boldsymbol{v}_M\}$. $\square$

Based on Eq. (A.19), we now let $\boldsymbol{A}$ be an $M \times K, K \leq M$, full column-rank

matrix and $\boldsymbol{\Pi}'_\nu = \boldsymbol{A}^H \boldsymbol{\Pi}_\nu \boldsymbol{A}$ and $\boldsymbol{\Pi}'_s = \boldsymbol{A}^H \boldsymbol{\Pi}_s \boldsymbol{A}$. Let $\boldsymbol{\Delta}$ and $\boldsymbol{\Phi}$ be the eigenvalue and eigenvector matrices of $\boldsymbol{\Pi}'_s$, i.e., $\boldsymbol{\Pi}'_s \boldsymbol{\Phi} = \boldsymbol{\Phi}\boldsymbol{\Delta}$. Then we let $\boldsymbol{\Sigma}$ and $\boldsymbol{\Psi}$ be the eigenvalue and eigenvector matrices of $\boldsymbol{\Delta}^{-1/2}\boldsymbol{\Phi}^H \boldsymbol{\Pi}'_\nu \boldsymbol{\Phi}\boldsymbol{\Delta}^{-1/2}$ and define $\boldsymbol{B} = \boldsymbol{\Phi}\boldsymbol{\Delta}^{-1/2}\boldsymbol{\Psi}$ ($\boldsymbol{B}$ is $K \times K$ and nonsingular). Then, it is easy to verify that

$$\boldsymbol{B}^H \boldsymbol{\Pi}'_\nu \boldsymbol{B} = \boldsymbol{\Sigma} \quad \text{and} \quad \boldsymbol{B}^H \boldsymbol{\Pi}'_s \boldsymbol{B} = \boldsymbol{I}_K \tag{A.20}$$

Also, we have

$$\arg\min_{\boldsymbol{A}} \left[\text{tr}(\boldsymbol{A}^H \boldsymbol{\Pi}_s \boldsymbol{A})^{-1}(\boldsymbol{A}^H \boldsymbol{\Pi}_\nu \boldsymbol{A}) \right]$$
$$= \arg\min_{\boldsymbol{A}} \left[\text{tr}(\boldsymbol{B}^H \boldsymbol{A}^H \boldsymbol{\Pi}_s \boldsymbol{A}\boldsymbol{B})^{-1}(\boldsymbol{B}^H \boldsymbol{A}^H \boldsymbol{\Pi}_\nu \boldsymbol{A}\boldsymbol{B}) \right] \tag{A.21}$$

If we writing $\boldsymbol{\Omega} = \boldsymbol{A}\boldsymbol{B}$, then the problem of Eq. (4.18) can be transformed to

$$\arg\min_{\boldsymbol{\Omega}} \left[\text{tr}(\boldsymbol{\Omega}^H \boldsymbol{\Pi}_\nu \boldsymbol{\Omega}) \right] \quad \text{s.t.} \quad \boldsymbol{\Omega}^H \boldsymbol{\Pi}_s \boldsymbol{\Omega} = \boldsymbol{I}_K \tag{A.22}$$

Further, let $\boldsymbol{\Pi}_s = \boldsymbol{H}\boldsymbol{H}^H$ and $\boldsymbol{\Upsilon} = \boldsymbol{H}^H \boldsymbol{\Omega}$ and we have,

$$\boldsymbol{\Omega}^H \boldsymbol{\Pi}_\nu \boldsymbol{\Omega} = \boldsymbol{\Omega}^H \boldsymbol{H}\boldsymbol{H}^{-1} \boldsymbol{\Pi}_\nu \boldsymbol{H}^{-H} \boldsymbol{H}^H \boldsymbol{\Omega}$$
$$= \boldsymbol{\Upsilon}^H \hat{\boldsymbol{\Pi}}_\nu \boldsymbol{\Upsilon} \tag{A.23}$$

where $\hat{\boldsymbol{\Pi}}_\nu = \boldsymbol{H}^{-1}\boldsymbol{\Pi}_\nu \boldsymbol{H}^{-H}$. Then the problem of Eq. (A.22) becomes

$$\arg\min_{\boldsymbol{\Upsilon}} \left[\text{tr}(\boldsymbol{\Upsilon}^H \hat{\boldsymbol{\Pi}}_\nu \boldsymbol{\Upsilon}) \right] \quad \text{s.t.} \quad \boldsymbol{\Upsilon}^H \boldsymbol{\Upsilon} = \boldsymbol{I}_K \tag{A.24}$$

Now Eq. (A.24) is in the same form as Eq. (A.19). Hence, if $\boldsymbol{\Lambda} = \text{diag}[\lambda_1, \cdots, \lambda_M]$

with $\lambda_1 \geq \cdots \geq \lambda_M$ is the eigenvalue matrix of $\hat{\mathbf{\Pi}}_\nu$ associated with the orthonormal eigenvectors $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_M$, then the optimizing matrix is $\mathbf{\Upsilon}_{\mathrm{op}} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_M]$. In other words, it can be described as

$$\hat{\mathbf{\Pi}}_\nu \mathbf{\Upsilon}_{\mathrm{op}} = (\boldsymbol{H}^{-1}\mathbf{\Pi}_\nu \boldsymbol{H}^{-H})\mathbf{\Upsilon}_{\mathrm{op}} = \mathbf{\Upsilon}_{\mathrm{op}}\mathbf{\Lambda} \tag{A.25}$$

Since $\mathbf{\Upsilon} = \boldsymbol{H}^H \mathbf{\Omega}$, we have $\mathbf{\Upsilon}_{\mathrm{op}} = \boldsymbol{H}^H \mathbf{\Omega}_{\mathrm{op}}$. Substituting into Eq. (A.25) and multiplying $\boldsymbol{H}^{-H}$ at each side, we have

$$\boldsymbol{H}^{-H}\boldsymbol{H}^{-1}\mathbf{\Pi}_\nu \mathbf{\Omega}_{\mathrm{op}} = \mathbf{\Omega}_{\mathrm{op}}\mathbf{\Lambda} \tag{A.26}$$

Following the definition of $\mathbf{\Pi}_s = \boldsymbol{H}\boldsymbol{H}^H$, Eq. (A.26) can be written as

$$\mathbf{\Pi}_s^{-1}\mathbf{\Pi}_\nu \mathbf{\Omega}_{\mathrm{op}} = \mathbf{\Omega}_{\mathrm{op}}\mathbf{\Lambda} \tag{A.27}$$

Eq. (A.27) can be regarded as the eigen-decomposition of the matrix $\mathbf{\Pi}_s^{-1}\mathbf{\Pi}_\nu$. Therefore, if $\{\lambda_1 \geq \cdots \geq \lambda_M\}$ and $\{\boldsymbol{u}_1, \cdots, \boldsymbol{u}_M\}$ are respectively the eigenvalues and eigenvectors of $\mathbf{\Pi}_s^{-1}\mathbf{\Pi}_\nu$, then the result follows. ∎

Bibliography

[1] J. C. Hancock and P. A. Wintz, *Signal detection theory*. McGraw-Hill, 1966.

[2] C. Weber, *Elements of Detection and Signal Design*. Springer-Verlag, 1987.

[3] L. L. Scharf, *Statistical Signal Processing*. Addison-Wesley, 1991.

[4] S. M. Kay, *Fundamentals of statistical signal processing*. Prentice-Hall, 1993.

[5] S. Haykin, "Adaptive filtering theory," *Prentice-Hall*, 1996.

[6] H. L. Van Trees, *Optimum array processing*. Wiley Interscience, 2002.

[7] D. Brillinger, *Time Series*. Society for Industrial and Applied Mathematics, 2001.

[8] Y. Li and K. M. Wong, "Riemannian distances for signal classification by power spectral density," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, pp. 655–669, Aug 2013.

[9] L. E. Franks, *Signal theory*. Englewood Cliffs, N.J. : Prentice-Hall, 1969.

[10] A. Stuart and M. G. Kendall, *The advanced theory of statistics*. Charles Griffin, 1968.

[11] J. Liang, "Detection of narrow-band sonar signals on a riemannian manifold," Master's thesis, MASc Thesis, McMaster University.

[12] P. T. Fletcher, S. Venkatasubramanian, and S. Joshi, "The geometric median on riemannian manifolds with application to robust atlas estimation," *NeuroImage*, vol. 45, no. 1, pp. S143–S152, 2009.

[13] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge university press, 2012.

[14] G. Golub and C. Van Loan, *Matrix Computations*. Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, 1996.

[15] J. Jost, *Riemannian geometry and geometric analysis*. Springer Science & Business Media, 2008.

[16] A. Besse, *Einstein Manifolds:*. Classics in mathematics, Springer, 1987.

[17] R. Bhatia, *Positive Definite Matrices*. Princeton Series in Applied Mathematics, Princeton University Press, 2009.

[18] M. Moakher, "A differential geometric approach to the geometric mean of symmetric positive-definite matrices," *SIAM J. Matrix Anal. Appl.*, vol. 26, pp. 735–747, Mar. 2005.

[19] X. Pennec, P. Fillard, and N. Ayache, "A riemannian framework for tensor computing," *International Journal of Computer Vision*, vol. 66, no. 1, pp. 41–66, 2006.

[20] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, *Fast and Simple Calculus on Tensors in the Log-Euclidean Framework*, pp. 115–122. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.

[21] O. Tuzel, F. Porikli, and P. Meer, "Pedestrian detection via classification on riemannian manifolds," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, no. 10, pp. 1713–1727, 2008.

[22] A. Barachant, S. Bonnet, M. Congedo, and C. Jutten, "Multiclass brain–computer interface classification by riemannian geometry," *Biomedical Engineering, IEEE Transactions on*, vol. 59, no. 4, pp. 920–928, 2012.

[23] D. Ciochina, M. Pesavento, and K. M. Wong, "Worst case robust downlink beamforming on the riemannian manifold," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pp. 3801–3805, IEEE, 2013.

[24] L. Xu, K. M. Wong, J.-K. Zhang, D. Ciochina, and M. Pesavento, "A riemannian distance for robust downlink beamforming," in *Signal Processing Advances in Wireless Communications (SPAWC), 2013 IEEE 14th Workshop on*, pp. 465–469, IEEE, 2013.

[25] J. Liang, K. M. Wong, and J. K. Zhang, "Detection of narrow-band sonar signal on a riemannian manifold," in *Electrical and Computer Engineering (CCECE), 2015 IEEE 28th Canadian Conference on*, pp. 959–964, 2015.

[26] M. Arnaudon, Barbaresco, and L. Yang, "Riemannian medians and means with applications to radar signal processing," *Selected Topics in Signal Processing, IEEE Journal*, vol. 7, no. 4, pp. 595–604, 2013.

[27] D. A. Bini and B. Iannazzo, "Computing the karcher mean of symmetric positive definite matrices," *Linear Algebra and its Applications*, vol. 438, no. 4, pp. 1700–1710, 2013.

[28] L. M. Ostresh Jr, "On the convergence of a class of iterative methods for solving the weber location problem," *Operations Research*, vol. 26, no. 4, pp. 597–609, 1978.

[29] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[30] N. De Moura, J. De Seixas, and R. Ramos, *Passive sonar signal detection and classification based on independent component analysis*. Citeseer, 2011.

[31] B. H. Maranda, *Handbook of Signal Processing in Acoustics*, ch. Passive Sonar, pp. 1757–1781. New York, NY: Springer New York, 2008.

[32] R. Walker, *The Detection Performance of FFT Processors for Narrowband Signals*. DREA Technical Memorium 82/A, Research and Development Branch, Dept. of National Defence Canada, 1982.

[33] H. L. Van Trees, *Detection, estimation, and modulation theory*. John Wiley & Sons, 2004.

[34] K. M. Wong and S. Chen, "Detection of narrow-band sonar signals using order statistical filters," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 5, pp. 597–613, 1987.

[35] L. Zhang, X. Huang, B. Huang, and P. Li, "A pixel shape index coupled with spectral information for classification of high spatial resolution remotely sensed

imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 10, p. 2950, 2006.

[36] K. B. Petersen, M. S. Pedersen, *et al.*, "The matrix cookbook," *Technical University of Denmark*, vol. 7, p. 15, 2008.

[37] H. Lütkepohl, *Handbook of Matrices*. Wiley, 1996.