الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي و البحث العلمي

جامعة وهران للعلوم و التكنولوجيا محمد بوضياف

THÈSE

En vue de l'obtention du Diplôme de Doctorat

Présentée par :

BEKHTAOUI Zineb

Intitulé

State Representation and Robust Subspace Tracking

Faculté	***: Génie Électrique***
Département	***: Electronique***
Domaine	***: Sciences et Technologies***
Filière	***: Génie Electrique***
Intitulé de la Formation	***: Vision et Technologies de l'Information et de la Communication***

Devant le Jury Composé de :

Membres de Jury	Grade	Qualité	Domiciliation
KECHE Mokhtar	Professeur	Président	USTO - MB
MECHE Abdelkrim	Professeur	Encadreur	USTO - MB
ABED MERAIM Karim	Professeur	Co-Encadreur	Université d'Orléans
OUAMRI Abdelaziz	Professeur	Examinateur	USTO - MB
ZEMALACHE MEGUNNI Kadda	Professeur	Examinateur	USTO - MB
BELOUCHRANI Adel	Professeur	Examinateur	ENP
DAHMANI Mohammed	Professeur	Invité	USTO - MB

Année Universitaire : 2022 / 2023

To my very supportive Parents: Menouar and Badia
To my beloved siblings: Wafaa, Younes, and Moufida
To my cute little nephews: Rayane and Lilya
To my grandparents: Si Mohamed and Zohra, may Allah bless them
To my never to be forgotten Hanna Zineb, may Allah have mercy on her
To my aunties, uncles, and cousins, Meriem and Rym in particular
To whom this work was as important to, as it was to me

Abstract

In this thesis, we have considered signal processing using the subspaces technique. Indeed, due to its various applications such as data compression, parametric estimation, source separation, tensor decomposition, etc., the estimation of the main or minor subspaces of a multi-variant signal has been the subject of a large number of research works, in particular in the case where the considered system is time-varying. In this case, the estimation of the subspace must be done in an adaptive way, and if possible at a low cost, we, therefore, aim to always reduce the computational complexity. In our work cited below, we have addressed the problem of non-Gaussian noise, i.e. impulsive noise such as burst noise or α-stable noise, outliers, or missing data. In this context, we have presented new robust subspace tracking algorithms. Later, we applied these algorithms to the state representation in the case of the estimation and tracking of directions of arrival. We then extended our algorithms to the case of principal eigenvectors estimation. Finally, we have applied them to blind source separation. The proposed algorithms' performances have been evaluated and validated by simulation and have all been submitted and accepted for scientific production

Keywords

Robust subspace; Eigenvectors; State representation; Blind source separation; Source localization; Signal processing; Robust estimation

Résumé

Dans ce mémoire, nous avons considéré le traitement de signal par la technique des sousespaces. En effet, de par ses applications diverses telles que la compression de données, l'estimation paramétrique, la séparation de sources, la décomposition tensorielle, etc., l'estimation des sous-espaces principaux ou mineurs d'un signal multi-variant a fait l'objet d'un grand nombre de travaux de recherche, en particulier dans le cas où le système considéré est variant dans le temps. Dans ce cas, l'estimation du sous-espace doit être réalisée de manière adaptative et si possible à faible coût, nous visons par conséquent à toujours réduire la complexité calculatoire. Dans nos travaux cités ci-après, nous nous sommes penchés sur la problématique des bruits non-gaussien, soit le bruit impulsif tel que le bruit en créneaux ou le bruit α-stable, ou les données erronées ou manquantes. Nous avons donc présenté dans ce contexte, de nouveaux algorithmes robustes de poursuite de sous-espaces. Plus tard, nous avons appliqué ces algorithmes à la représentation d'état avec le cas de l'estimation et la poursuite des directions d'arrivées de cibles mouvantes en champ lointain. Nous avons par la suite étendu nos algorithmes au cas de l'estimation des vecteurs propres principaux. Enfin, nous avons appliqué ces derniers pour le blanchiment adaptatif en séparation aveugle de sources. Les performances de tous les algorithmes proposés ont été évalués et validés par simulation, et les travaux corrspendants ont tous été soumis et acceptés en production scientifique.

Mots-clé

Sous-espace robuste ; Vecteurs propres ; Représentation d'état ; Séparation aveugle de sources ; Localisation angulaire ; Traitement du signal ; Estimation robuste

ملخص

في هذه الأطروحة ، درسنا معالجة الإشارات باستخدام تقنية الفضاءات الجزئية. بالفعل ، نظرًا لتطبيقاته المختلفة مثل ضغط البيانات ، تقدير البرمترات ، فصل المصادر ، تحليل الموترات ، وما إلى ذلك ، كان تقديرالفضاءات الجزئية الرئيسية أو الثانوية للإشارة المتعددة المتغيرات موضوعًا لعدد كبير من الأعمال البحثية ، على وجه الخصوص في الحالة التي يكون فيها النظام قيد النظر متغيرًا بمرور الزمن. في هذه الحالة ، يجب أن يتم تقدير الفضاء الجزئي بطريقة تكيفية ، وإذا أمكن بتكلفة منخفضة ، فإننا نهدف دائمًا إلى تقليل التعقيد الحسابي. في عملنا المذكور أدناه ، تناولنا مشكلة الضجيج غير الغاوسية، مثل الضجيج الاندفاعي كضجيج كبير لمدة زمنية قصيرة أو ضجيج حسب توزيعات الفا-المستقرة أو القيم المتطرفة أو البيانات المفقودة. في هذا السياق ، قدمنا خوارزميات مقاومة جديدة لتتبع الفضاء الجزئي. في وقت لاحق ، قمنا بتطبيق هذه الخوارزميات على تمثيل الحالة في حالة تقدير وتتبع اتجاهات الوصول. ثم قمنا بتوسيع الخوارزميات الخاصة بنا لتشمل حالة تقدير الأشعة الذاتية الرئيسية. أخيرًا ، قمنا بتطبيقها على فصل المصادر الأعمى. تم تقييم أداء الخوارزميات المقترحة والتحقق من صحتها عن طريق المحاكاة وتم تقديم جميعها وقبولها لإنتاجات العلمية.

كلمات مفتاحية

الفضاءات الجزئية المقاومة؛ الأشعة الذاتية؛ تمثيل الحالة؛ فصل المصادر الأعمى؛ تحديد موقع المصدر؛ معالجة الإشارات؛ التقدير المقاوم

Acknowledgment

First of all, I would like to show my gratitude and my thankfulness towards Allah the almighty for giving me the ability, strength, and motivation the achieve my research work.

To my supervisors, Prof. ABED MERAIM Karim and Prof. MECHE Abdelkrim, thank you for your precious support, guidance, ideas, and discussions. Thanks to you, this experience was full of enrichment both in research and in daily life. My appreciations go to Prof. MECHE Abdelkrim for believing in me and always pushing me beyond what I thought was my limits. To Prof. ABED MERAIM Karim, your kindness and hard work have set an example for me to follow, thank you for being a model to be inspired by.

I would like to thank Prof. OUAMRI Abdelaziz for welcoming me in his laboratory *Laboratoire Signaux et Image*, for his encouraging words during my years in the lab, and for his acceptance to be a member of my thesis jury.

Special recognition goes to Prof. KECHE Mokhtar, for his support, help, and encouragement during the preparation period. But also for having done me the honor of accepting the presidency of my thesis jury.

I would like to express my gratitude to the jury members, Prof. BELOUCHRANI Adel, Prof. ZEMALACHE MEGUENNI Kadda, and Prof. DAHMANI Mohammed for having accepted to evaluate my work.

Lastly, I would like to thank my parents, without whom I couldn't have achieved any of this work. I am forever indebted to them. Special thanks to my beloved siblings and all my family members and friends for their unconditional support.

Contents

List of Figures

Abbreviations

ADMM	*Alternating Direction Method of Multipliers*
API	*Approximated Power Iteration*
ASOBI	*Adaptive Second Order Blind Identification*
BNC-PAST	*Bounded Non-linear Covariance - PAST*
BSS	*Blind Source Separation*
CA	*discrete-time Constant-Acceleration model*
CV	*discrete-time Constant-Velocity model*
DoA	*Directions of Arrival*
DPM	*Data Projection Method*
DWNA	*Discrete White Noise Acceleration model*
DWPA	*Discrete Wiener Process Acceleration model*
ESPRIT	*Estimation of Signal Parameters via Rotational Invariance Techniques*
EVD	*Eigenvalue Decomposition*
FAPI	*Fast Approximated Power Iteration*
FDPM	*Fast Data Projection Method*
FOOja	*Fast OOja*
FRANS	*Fast Rayleigh quotient-based Adaptive Noise Subspace algorithm*
GOPAST	*Givens based Orthogonal Projection Approximation for Subspace Tracking*
GxFAPI	*Givens (M-HT) FAPI*
HTFAPI	*Hard Thresholding based Fast Approximated Power Iteration*
IoT	*Internet of Things*
IQR	*Inter Quartile Range*
KF	*Kalman Filter*
KFVM	*Kalman Filter with Variable number of Measurments*
KFVNM	*Kalman Filter with Variable Number of Measurments*
MCC-PAST	*Maximum Correntropy Criterion PAST*
MFAPI	*Mahalanobis based Fast Approximated Power Iteration*
MUSIC	*MUltiple SIgnal Classification*
NCV	*discrete-time Nearly-Constant-Velocity model*
NP3	*Natural Power method 3*
OOja	*Othogonal Oja*

OPAST	*Orthogonal Projection Approximation for Subspace Tracking*
OROBUSTA	*Orthogonal ROBUSTA*
OS	*Order Statistics*
PAST	*Projection Approximation for Subspace Tracking*
PCA	*Principal Components Analysis*
PCAPA	*PCA using PETRELS-ADMM*
PETRELS	*Parallel Estimation and Tracking by REcursive Least Squares*
PSA	*Principal Subspace Analysis*
RA-SOBI	*Robust Adaptive SOBI*
RLS	*Recursive Least Squares*
ROBUSTA	*ROBUst Subspace Tracking Algorithm*
RPAST	*Robust Projection Approximation for Subspace Tracking*
RST	*Robust Subspace Tracking*
SE	*Subspace Estimation*
SEP	*Subspace Estimation Performance*
SIRV	*Spherically Invariant Random Variable*
SOBI	*Second Order Blind Identification*
ST	*Subspace Tracking*
SVD	*Singular Value Decomposition*
TLS-ESPRIT	*Total Least Squares-ESPRIT*
TWS	*Track While-Scan radar system*
WLS	*Weighted Least Squares*

Notations

Notation	Meaning
a	scalar
$\mathbf{a}$	vector
$\mathbf{A}$	Matrix
$\mathcal{O}(.)$	Order of complexity
$\mathbf{I}_j$	a $(j \times j)$ eye matrix
$\mathbf{0}_{j \times k}$	a $(j \times k)$ matrix which entries are all zeros
$\stackrel{\Delta}{=}$	equal by definition
$tr(\cdot)$	trace matrix operator
$Q1(\cdot), Q3(\cdot)$	First and third quartile respectively
$\|\|.\|\|_F$	the Frobenius norm
$E[\cdot]$	Mathematical expectation
$a_{j\|i}$	the j^{th} expected value of a at time i

Chapter 1

General Introduction

Stream processing is attracting more and more attention from both academia and industry due to the continuous increase of massive data stream collection over the years and their high utilization in the discovery of new insights and valuable information. Several examples can be cited such as online applications, Servers and security logs, or moreover, the Internet of Things (IoT) where a huge number of sensing devices Are installed and in use. In the latter for example, The devices are capable of real-time data collection, management, and transmission via IoT networks. Therefore, stream processing is necessary to quickly and efficiently extract crucial insights from such data in order to support real-time decision-making.

It is very well known that Singular Value Decomposition (SVD) is one of the most powerful and widely-used linear algebra techniques with a number of applications in various domains. In an online setting, data samples are continuously collected with time. Accordingly, recomputing the batch Principal Components Analysis (PCA) methods like SVD and Eigen value Decomposition (EVD) at each time step becomes inefficient due to their high complexity and time variation. This has led to defining a variant of the PCA called online (adaptive) PCA in which we may want to track the underlying process that generates streaming data with time.

Later, it has been shown that in a lot of applications, the exact principal components are not needed. Instead, only an orthogonal basis that spans the same subspace was. This was the beginning of the Principal Subspace Analysis (PSA) journey. Moreover, the need for real-time applications has been raised. Thus, researchers began to aim for fast (low complexity) algorithms that can estimate subspaces in the most accurate way.

Although this was achieved in a satisfactory way in the last decade, the limitations of harsh environments are still up to date. Indeed, in practice, the Gaussian noise environment is rarely encountered, instead, one can find several adverse scenarios such as impulsive noise, sparse outliers, or missing data, but the literature about fast methods is still very thin. The latter have motivated as to dive into the subject and propose new algorithms that could thrive in the field.

Consequently, in this thesis, we will introduce new approaches to deal with the subject. But also, apply the latter to real-world applications such as target tracking, source localization, and blind source separation. The rest of this thesis is organized as the following outline.

Chapter 2

In the second chapter of this thesis, we will present a general overview of the problem handled. Indeed, we will detail the system equations as well as the various noises that can affect it. Then, we will state the problematic addressed along with the main objectives of our work. Just after that, we will cite the state of the art with all the literature works related to our study, and the contributions we have made in relation to them including the motivation of the latter.

Chapter 3

In the third chapter, we will address low-cost robust subspace tracking in an impulsive noise environment. Indeed, we will present a new cost function

based on the weighted least square criterion and a projection approximation, where two new methods of calculating the weight will be proposed: a calculated method based on the robust estimation of the covariance matrix and a hard-thresholding one leading to a reduction of the computational cost. The proposed algorithms will be both tested via simulation in both burst noise and α-stable noise.

Chapter 4

In this fourth chapter, we will first extend the subspace tracking issue to the PCA one. Indeed we will present new fast algorithms that estimate and track the eigenvectors of a system evolving in an impulsive noise environment as well as s system conflicted with outliers and missing data. We will then follow our works with an application to the blind source separation context using the second-order statistics technique. All of our proposed work will be tested and validated with simulation.

Chapter 5

Chapter five will be about state representation, where the domain of tracking a mobile point is introduced. We will, thus, present one of the various applications of subspace tracking; the estimation and tracking of directions of arrival. In this context, we will present a novel robust algorithm based on our robust subspace tracking followed by a results smoothing using adaptive filtering. Indeed, two tracking filters will be detailed, e.i. The Kalman filter and its low-cost, steady-state variant the $\alpha\beta$ filter. Again, the algorithm performance will be evaluated with monte-Carlo simulation and compared to similar algorithms' performances.

Chapter 6

Finally, in the last chapter of the thesis, we will present a general conclusion where a discussion of our proposed algorithms will be conducted. Some limitations will be also discussed and an overall perspective will be proposed.

List of publications

Almost all of our contributions and proposed algorithms have been published/accepted for publication in the following papers:

Journal papers

[1] Zineb Bekhtaoui, Karim Abed-Meraim, and Abdelkrim Meche. Robust adaptive algorithms for fast principal component analysis. Digital Signal Processing, 127 :103561, 2022.
[2] Zineb Bekhtaoui, Karim Abed-Meraim, Abdelkrim Meche, and Messaoud Thameri. A new robust adaptive algorithm for second order blind source separation. ENP Engineering Science Journal, 2(1) :21–28, 2022.

Conference papers

[3] Z Bekhtaoui, A Meche, M Dahmani, and K Abed-Meraim. Maneuvering target tracking using q-learning based kalman filter. In 2017 5^{th} International Conference on Electrical Engineering-Boumerdes, Algeria(ICEE-B), pages 1–5. IEEE, 2017.
[4] Z Bekhtaoui, A Meche, K Abed-Meraim, and M Dahmani. A new robust subspace tracking algorithm and its application to direction of arrival estimation and tracking. In 2020 In the International Conference on Defense systems: Architectures and Technologies - Constantine, Algeria (IC-DAT). 2020.
[5] Zineb Bekhtaoui, Abdelkrim Meche, Karim Abed-Meraim, and Mohammed Dahmani. Direction of arrival tracking using adaptive robust subspace decomposition and kalman filter. In 2023 20^{th} International Multi-Conference on Systems, Signals and Devices - Mahdia, Tunisia (SSD). IEEE, 2023.

Chapter 2

Problem Statement and State of the Art

2.1 Introduction

Subspace tracking is a powerful technique used in signal processing and control theory to estimate the dynamics of a system from a set of observations. It involves tracking the evolution of a low-dimensional subspace that captures the important dynamics of the system. Subspace tracking has applications in a wide range of fields, including radar signal processing, computer vision, and structural health monitoring. In this technique, the subspace is estimated recursively from a sequence of data, allowing for real-time tracking of system dynamics. The accuracy of the subspace estimate depends on several factors, including the quality of the data and the complexity of the system being tracked. Subspace tracking has proven to be a valuable tool for a wide range of applications and continues to be an active area of research in the field of signal processing and control theory. However, and most likely in the practical field, one can come across harsh environments such as systems with missing data, outliers, or impulsive noises. Here we are interested in the latter, and we will present new solutions for both subspace tracking and principal components tracking in adverse scenarios and at a low computational cost.

In this chapter, we will define the system studied along with its possible adverse complications. Then we will overview the most popular methods on the subject in the recent literature.

2.2 Problem statement

2.2.1 The structure of the studied system

Since we are interested in subspace tracking in the adaptive context, we consider in the rest of this thesis a continuous observed data stream in time. This observations are modeled by a multivariate vector $\mathbf{x}(t)$ of dimension n and covariance $\mathbf{C}_x = E[\mathbf{x}\mathbf{x}^H]$, composed of source signals $\mathbf{s}(t)$ affected by a noise $\mathbf{n}(t)$ of variance σ_n^2 according to:

$$\mathbf{x}(t) = \mathbf{s}(t) + \mathbf{n}(t) \tag{2.1}$$

2.2.1.1 Signal structure

For the works considered in this thesis, we take the particular case where the received signals are a combination of any p signal sources $\tilde{\mathbf{s}}(t)$ mixed by a $(n \times p)$ mixing matrix $\mathbf{A}$:

$$\mathbf{s}(t) = \mathbf{A}\tilde{\mathbf{s}}(t) \tag{2.2}$$

We consider two cases of mixing: either, the general case where $\mathbf{A}$ is a random orthogonal matrix, or, the case of a linear antenna array regularly spaced by a distance of half the received plane wavelength:

$$\mathbf{A} = [\mathbf{a}(\omega_1); \mathbf{a}(\omega_2); \cdots; \mathbf{a}(\omega_p)] \tag{2.3}$$

with: $\mathbf{a}(\omega_i) = [1, e^{j\omega_i}, \cdots, e^{j\omega_i(n-1)}]^T$ and $\omega_i = \pi \sin(\theta_i)$ where θ_i represent the angles of arrival of the different sources.

2.2.1.2 Noise structure

In theory, as in practice, there are several types of noise. In this part, we present some of these noises that we will treat later in our work.

a **Gaussian noise:** This is the most common modeling. In this case, $\mathbf{n}(t) = \mathbf{n}_G(t)$ in (2.1) is an additive white noise with Gaussian distribution and variance σ_G^2.

b **Burst noise:** In this case, $\mathbf{n}(t)$ in (2.1) is the sum of $\mathbf{n}_G(t)$ and an impulsive term. The latter is modeled as in [6] by a burst noise such as:

$$\mathbf{n}(t) = \mathbf{n}_G(t) + \sum_{i=1}^{N_I} u\Big(\frac{t - t_i}{b_i}\Big)\mathbf{n}_I^i(t) \tag{2.4}$$

where $\mathbf{n}_I^i(t)$ is also a white and centered Gaussian of variance $\sigma_I^2 >> \sigma_G^2$. $\mathbf{n}_I^i(t)$ weighted by a rectangular function $u(.)$. This one is used to describe the appearance of impulsive noise for a short period of time. N_I refers to the number of impulsive events while t_i and b_i denote respectively the center of the i^{th} event and its duration.

c α**-stable noise:** Here, the noise $\mathbf{n}(t)$ has an α-stable distribution named here after $\mathbf{n}_\alpha(t)$ [7]. This one represents events with a heavy tail characterized by a factor $0 < \alpha \leq 2$. This factor controls the heaviness of the tail. These trails are then heavier, and therefore the events are more impulsive, for small values of α, while for large values, distribution is less impulsive. Finally, $\alpha = 2$ corresponds to a Gaussian distribution.

$$\mathbf{n}(t) = \mathbf{n}_\alpha(t) \tag{2.5}$$

d **Outliers:** For this case, the data are modeled by the desired signal $\mathbf{s}(t)$, an additive Gaussian noise $\mathbf{n}_G(t)$ plus a vector of outliers $\mathbf{i}(t)$ assumed large and sparse:

$$\mathbf{n}(t) = \mathbf{n}_G(t) + \mathbf{i}(t) \tag{2.6}$$

2.2.1.3 Other adversities

Besides noises, one can come across other adversities such as:

a **Missing data:** In this case, we consider that some data are missing. The observations are therefore written as follows:

$$\mathbf{x}(t) = \mathbf{P}(t)\left(\mathbf{s}(t) + \mathbf{n}_G(t)\right) \tag{2.7}$$

where $\mathbf{P}(t)$ is a random diagonal matrix with 1s or 0s entries depending on whether the input is observed or not.

b **Simultaneous outliers and missing data:** In the latter case, the worst one, we model a system that contains missing values and is affected by erroneous values at the same time:

$$\mathbf{x}(t) = \mathbf{P}(t)\left(\mathbf{s}(t) + \mathbf{n}(t) + \mathbf{i}(t)\right) \tag{2.8}$$

2.2.2 Objectives

One of the most common techniques in signal processing is the decomposition of the signal into its edge components (PCA for "Principal Components Analysis"). The main objective is therefore to estimate in time the edge components of the covariance matrix $\mathbf{C}_x$. That is its p eigenvectors $\mathbf{U}(t) = [\mathbf{u}_1(t), \cdots, \mathbf{u}_p(t)]$ corresponding to its first p eigenvalues, where the eigenvalues λ_i are supposed to obey to $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p > \lambda_{p+1}$.

However, in many applications, one would only need a $\mathbf{D}(t)$ basis that generates the same subspace as the one generated by the principal eigenvectors, called the principal subspace or the signal subspace. The latter is called Principal Subspace Analysis (PSA).

This has been widely discussed in the literature, however, there are still limitations to overcome such as the robustness in the presence of non-Gaussian noises as presented previously. In the following section, we will discuss the different existing methods dealing with the tracking of subspaces and eigenvectors, as well as the difficulties encountered in the field.

2.3 State of the art

2.3.1 Subspace tracking

The PSA methods have known a very wide interest and continue to evolve on several axes. One can find that they have been classified according to two main criteria:

Classification according to the estimation method in time: we will find two types of method classes;

- Batch methods: They are applied on blocks of data supposed to be stationary or invariant in time. The most used algorithms for that are the single value decomposition (SVD) applied to the signal's sequence, and the eigenvalue decomposition (EVD) applied to the covariance matrix of the signal. These types of algorithms are very expensive and are therefore used offline.

- adaptive algorithms: they are applied in non-stationary systems, where an estimation of the eigenstructures is made at each instant using the observations available at that instant. In this type of algorithm, the EVD and the SVD are very computationally expensive. In addition, and in several applications, it has been shown that instead of computing the whole structure it would be sufficient to compute only the basis of the desired signals or only an approximation of it. Several algorithms have been proposed including modified Batch algorithms and optimization methods. Finally, it is worth mentioning, due to the high demand, that these methods can be used in online mode.

Classification by complexity rank:
In general, in the estimation of subspaces, we find methods such as EVD and SVD in batch computation, although these algorithms have a complexity of order three, i.e. $\mathcal{O}(n^3)$ and are therefore not adequate for adaptive algorithms. The latter has been developed according to three (03) classes:

- high complexity algorithms. They require a computation of $\mathcal{O}(n^2p)$ or $\mathcal{O}(n^2)$ where p is the rank of the principal subspace. The algorithms of this class offer a very good convergence but are very expensive in computation.

- algorithms with medium or moderate complexity. With a computation of $\mathcal{O}(np^2)$, this class offers a very good compromise between convergence and cost.

- Finally, the low complexity algorithms or linear complexity have a computation of $\mathcal{O}(np)$, that we would always want to reach while working on the performance of its algorithms.

In the rest of this thesis, we are interested in linear complexity methods, since this one is the best adapted for online processing, and consequently, the best adapted for tracking.

As already stated, Subspace Tracking (ST) is a crucial topic that has been extensively researched and surveyed in the literature. Comon and Golub were among the first researchers to conduct a survey on principal subspace tracking algorithms, as described in [8]. Their survey primarily focused on methods that could handle the low-rank approximation of covariance matrices that vary slowly over time, and which had high to moderate computational complexity. Another significant survey, provided by Delmas in [9], offers a comprehensive overview of the advancements made in classical ST algorithms with low (linear) complexity. These surveys provide valuable insights into the various techniques and approaches used in ST and serve as an essential resource for researchers and practitioners alike.

In [9], it has been stated that the adaptive eigenvector estimation with linear complexity was introduced by Oja in [10], initially for a single principal eigenvalue $p = 1$. This marked the beginning of the era of fast algorithms. We note that the methods based on power iteration as well as those based on least square errors represent derivatives of Oja's method, if we perform a gradient descent estimation on their respective cost functions [9]. This method has been extended for the case of $p > 1$ in [11].

Indeed, Oja's algorithm converges to an orthonormal basis but does not ensure orthogonality at each time instant, a property that is highly favored in the field. For this reason, the Orthogonal Oja (OOja) algorithm was proposed in [12] where orthogonalization is ensured at each time instant where the new basis $\mathbf{D}(t)$ is replaced by $\mathbf{D}(t) = \mathbf{D}(t)\mathbf{D}^T(t)\mathbf{D}^{-1/2}(t)$. The linear completeness is then ensured by the fast implementation of the following property:

$$\left(\mathbf{I} + \mathbf{x}\mathbf{x}^T\right)^{-1/2} = \mathbf{I} + \left(\frac{1}{\sqrt{1 + ||\mathbf{x}||^2}} - 1\right)\frac{\mathbf{x}\mathbf{x}^T}{||\mathbf{x}||^2} \tag{2.9}$$

In [13], they proposed to normalize the forgetting factor by estimating it adaptively in order to improve the convergence speed of Oja's algorithm as well as its orthogonal version OOja. However, in all the cases proposed so far, the instant covariance matrix used is estimated as $\mathbf{C}_x = \mathbf{x}\mathbf{x}^T$, while a better accuracy can be achieved with the estimation $\mathbf{C}_x = \beta\mathbf{C}_x(t-1) + \mathbf{x}\mathbf{x}^T$ [9].

Later on, another version of OOja, named Fast OOja (FOOja) has been proposed in [14]. it has been proven theoretically and by simulation that it is more stable than the others but was outperformed by the Fast Data Projection Method (FDPM) which will be discussed later in this section in terms of computation cost despite giving the same accuracy.

On the other hand, In [15], two new theorems were presented by Yang, stating that an orthonormal basis of the principal subspace can be achieved by minimizing the following least square criterion:

$$\begin{aligned} J(\mathbf{D}) &= E\|\mathbf{x} - \mathbf{D}\mathbf{D}^H\mathbf{x}\| \\ &= tr(\mathbf{C}_x) - 2tr(\mathbf{D}^H\mathbf{C}_x\mathbf{D}^H) + tr(\mathbf{D}^H\mathbf{C}_x\mathbf{D}\mathbf{D}^H\mathbf{D}) \\ \Delta(J(\mathbf{D})) &= 2[-2\mathbf{C}_x + \mathbf{C}_x\mathbf{D}\mathbf{D}^H + \mathbf{D}\mathbf{D}^H\mathbf{C}_x]\mathbf{D} \end{aligned} \tag{2.10}$$

- $\mathbf{D}$ is a stationary point of $J(\mathbf{D})$, if and only if $\mathbf{D} = \mathbf{U}\mathbf{Q}$ where the columns of $\mathbf{U}$ are p distinct eigenvectors of $\mathbf{C}_x$ and $\mathbf{Q}$ is an arbitrary unitary matrix

- All stationary points are saddle points except when $\mathbf{U}$ contains the p eigenvectors corresponding to the p greatest eigenvalues.

As stated before, the matrix $\mathbf{D}$ that achieves the minimization of the cost function doesn't contain the eigenvectors of $\mathbf{C}_x$ but only an orthonormal basis that spans the signal subspace. Consequently, the matrix $\mathbf{D}$ is not unique, only the projection matrix $\mathbf{D}\mathbf{D}^H$ that projects the observation $\mathbf{x}$ into the signal subspace is.

From here, Yang proposed to approximate the previous cost function with its exponential window mean with a forgetting factor $0 < \beta < 1$ in order to achieve adaptability:

$$J(\mathbf{D}(t)) = \sum_{i=1}^{t} \beta^{t-i} \|\mathbf{x}(i) - \mathbf{D}(t)\mathbf{D}^H(t)\mathbf{x}(i)\|^2 \tag{2.11}$$

$$= tr(\mathbf{C}_x(t)) - 2tr(\mathbf{D}^H\mathbf{C}_x(t)\mathbf{D}^H) + tr(\mathbf{D}^H\mathbf{C}_x(t)\mathbf{D}\mathbf{D}^H\mathbf{D}) \tag{2.12}$$

Where the covariance matrix $\mathbf{C}_x$ is also estimated as an exponential mean of $\mathbf{x}\mathbf{x}^H$:

$$\mathbf{C}_x(t) = \sum_{i=1}^{t} \beta^{t-i}\mathbf{x}(i)\mathbf{x}^H(i) = \beta\mathbf{C}_x(t-1) + \mathbf{x}(t)\mathbf{x}^H(t) \tag{2.13}$$

It is indeed stated that the two precedent theorems stay valid for the above approximation.

Furthermore, a projection approximation $\mathbf{D}^H\mathbf{x}(i) \simeq \mathbf{D}^H(i-1)\mathbf{x}(i)$ where $1 \leq i \leq t$ was introduced to achieve the minimization of the later cost function. This led to the new cost function:

$$J'(\mathbf{D}(t)) = \sum_{i=1}^{t} \beta^{t-i} \|\mathbf{x}(i) - \mathbf{D}(t)\mathbf{y}(i)\|^2 \tag{2.14}$$

with $\mathbf{y}(i) = \mathbf{D}^H(i-1)\mathbf{x}(i)$

One can note that the error for this approximation is negligible when we take into account that the signals vary slowly in time, in particular, when

i is close to t. Moreover, even when i is further away, the deviation in the approximation does not affect the estimation of the subspace thanks to the forgetting factor.

Finally, the minimization of the latter can be achieved using a Recursive Least Squares (RLS) algorithm, and by making good use of the inversion lemma, a fast algorithm can be deducted straightforwardly giving the Projection Approximation Subspace Tracking (PAST) algorithm.

However, due to the approximation, the estimated basis of the principal subspace is not orthogonal. Later on, it has been proposed in [16], an orthogonalization step that ensured the orthonormality of the subspace at each time instant. the latter was reached at a low-cost by using the propriety stated in (2.9) which led to the Orthogonal PAST algorithm (OPAST).

Meanwhile, other methods based on an instant iterative estimation of the covariance matrix have been discussed. Those methods rely on the power method and are discussed hereafter.

The primary objective of the first subspace problem is to determine the eigenvector that corresponds to the largest eigenvalue of a given matrix. One of the simplest iterative techniques for solving this problem is the power method, which is described in the following section. If the unique dominant eigenvalue of a real symmetric matrix $\mathbf{C}$ is denoted as λ_1, and its corresponding eigenvector with a unit 2-norm is $\mathbf{u}_1$, then the power method, starting from an arbitrary unit 2-norm $\mathbf{d}_0$ that is not orthogonal to $\mathbf{u}_1$, produces a sequence $(\alpha_i; \mathbf{d}_i)$ that converges to the largest eigenvalue λ_1 and its corresponding eigenvector with a unit 2-norm $\mathbf{u}_1$. The proof of the sequel can be found in [17].

$$
\begin{aligned}
&\mathbf{d}_0 \text{ arbitrary such that } \mathbf{d}_0^T \mathbf{u}_1 \neq 0 \\
&\text{for } t = 0, 1, \ldots \\
&\qquad \mathbf{d}'_{t+1} = \mathbf{C}\mathbf{d}_t \\
&\qquad \mathbf{d}'_{t+1} = \mathbf{d}'_{t+1} / ||\mathbf{d}'_{t+1}||_2 \\
&\qquad \alpha_{t+1} = \mathbf{d}_{t+1}^T \mathbf{C}\mathbf{d}_t
\end{aligned}
\tag{2.15}
$$

Assuming that the matrix $\mathbf{C}$ is non-negative, a simple extension of the power method enables the calculation of the p eigenvectors that correspond to the p largest eigenvalues of $\mathbf{C}$, given that the first $p+1$ eigenvalues are distinct. Alternatively, the same method can be used to compute the subspace that corresponds to the p largest eigenvalues of $\mathbf{C}$ if the p-th eigenvalue is strictly greater than the $(p+1)$-th eigenvalue, denoted as λ_p and λ_{p+1}, respectively. This method can be found in the literature under the name of orthogonal iteration, e.g., in [17], subspace iteration, e.g., in [18] or simultaneous iteration method, e.g., in [19].

$$\begin{aligned}
&\mathbf{D}_0 \text{ arbitrary } (n \times p) \text{ matrix such that } \mathbf{D}_0^T\mathbf{U}_1 \text{is not singular} \\
&for\ t = 0, 1, \ldots \\
&\quad \mathbf{D}'_{t+1} = \mathbf{C}\mathbf{D}_t \\
&\quad \mathbf{D}'_{t+1} = \mathbf{D}'_{t+1}\mathbf{R}_{t+1} \text{ skinny } \mathbf{QR} \text{ factorization} \\
&\quad \Lambda_{t+1} = diag(\mathbf{D}_{t+1}^T\mathbf{C}\mathbf{D}_t)
\end{aligned} \tag{2.16}$$

or

$$\begin{aligned}
&\mathbf{D}_0 \text{ arbitrary } (n \times p) \text{ matrix such that } \mathbf{D}_0^T\mathbf{U}_1 \text{ not singular} \\
&for\ t = 0, 1, \ldots \mathbf{D}_{t+1} = Orthonorm\{\mathbf{C}\mathbf{D}_t\}
\end{aligned} \tag{2.17}$$

where the orthonormalization (Orthonorm) procedure is not necessarily given by the QR factorization.

As stated in [9], we can write the following variant of the adaptive implementation:

$$\mathbf{D}(t+1) = Orthonorm\{(\mathbf{I} + \mu\mathbf{C})\mathbf{D}(t)\} \tag{2.18}$$

Where $\mu > 1$ is a small parameter known as a step size. Now, by replacing $\mathbf{C}$ bu its instant estimate $\mathbf{C}(t)$, one can have the adaptive orthogonal iterative algorithm:

$$\mathbf{D}(t+1) = Orthonorm\{(\mathbf{I} + \mu\mathbf{C}(t))\mathbf{D}(t)\} \tag{2.19}$$

At this point, depending on the choice of the instant estimate $\mathbf{C}(t)$ and the chosen orthonormalization process, we can obtain various alternative ST algorithms.

One straightforward option for selecting the matrix $\mathbf{C}(t)$ is to use the immediate approximation $\mathbf{x}(t)\mathbf{x}^T(t)$, which leads to the Data Projection Method (DPM) that was initially presented in [20]. In this approach, the orthonormalization process is carried out utilizing the Gram-Schmidt procedure.

From here, one can write:

$$\mathbf{D}(t+1) = \big(\mathbf{D}(t) + \mu\mathbf{x}(t)\mathbf{x}^T(t)\mathbf{D}(t)\big)\,\mathbf{G}(t+1) \tag{2.20}$$

To reduce the complexity of the $\mathbf{G}(t+1)$ computation, two families of algorithms have been proposed in the literature. The approximate symmetric orthonormalization family which leads back to the Oja derivatives algorithms stated previously, and The exact orthonormalization family.

The latter can be performed exactly at each iteration by two ways; First, by the symmetric square root inverse of $\mathbf{D'}^T(t+1)\mathbf{D'}(t+1)$ using the property 2.9. This leads to the Fast Rayleigh quotient-based Adaptive Noise Subspace algorithm (FRANS) introduced by Attallah et al. in [21]. Or, using the two steps method introduced in [22] and [23], such as:

$$\mathbf{D}(t+1) = Normalize\{\big(\mathbf{D}(t) + \mu\mathbf{x}(t)\mathbf{x}^T(t)\mathbf{D}(t)\big)\,\mathbf{G}(t+1)\} \tag{2.21}$$

Where, $Normalize\{\cdot\}$ stands for the normalization of the columns of its entry. Note that here $\mathbf{G}(t+1)$ is not unique. For the fast computation of the latter, Doukopoulos et al. proposed to use the Householder transform in [24] which gave the very stable algorithm: Fast Data Projection Method (FDPM).

Furthermore, The algorithms mentioned above, which don't rely on the rank one characteristic (e.i. $p = 1$) of the instantaneous estimate $\mathbf{x}(t)\mathbf{x}^T(t)$

of $\mathbf{Cx}(t)$, can be applied to the exponential or sliding window estimates of $\mathbf{C}(t)$. However, this would result in a significant increase in complexity. To overcome that, the Natural Power method 3 (NP3), that has been proposed in [25], uses the exponential windowed estimate:

$$\mathbf{C}(t) = \beta\mathbf{C}(t-1) + \mathbf{x}(t)\mathbf{x}^T(t). \tag{2.22}$$

based on the approximation:

$$\mathbf{C}(t-1)\mathbf{D}(t) = \mathbf{C}(t-1)\mathbf{D}(T-1) \tag{2.23}$$

which is clearly valid if $\mathbf{D}(t)$ is slowly varying in time.

Later on, an algorithm called the Approximated Power Iteration (API) has been introduced in [26], which is based on the assumption that $\mathbf{D}(t)$ and $\mathbf{D}(t+1)$ span the same r-dimensional subspace. This algorithm, along with its faster implementation (FAPI), uses the same power-based approach but unlike the NP3 algorithm, the API algorithm can accommodate the exponential or sliding windowed estimates of $\mathbf{C}_x(t)$ within the same framework, with a complexity of $O(nr)$ operations.

In addition, many practitioners (e.g., [27, 9]) have acknowledged the FAPI algorithm as the top-performing power-based subspace tracking method among those based on the exponential or sliding window and having the same computational complexity.

2.3.2 Robust subspace traking

Although the previous algorithms present good performances in accuracy, complexity, and stability, they only handle environments with the presence of Gaussian noise only. Indeed, when impulsive noise occurs, those algorithms diverge very quickly and all accuracy is lost.

When we refer to "impulsive" noise, we are primarily talking about three types of noise: burst noise [28, 29], Spherically Invariant Random Variable (SIRV) noise [30, 31], or alpha-stable noise [32, 7]. Although the algorithms in the literature were developed to mitigate the impact of impulsive noise

in general, the majority of the simulation results only demonstrate their performance in the context of burst noise.

2.3.2.1 Robust variants of PAST

Various approaches have been proposed in the literature to handle impulsive noise, with many relying on robust statistics. Among them, some studies have introduced robust versions of the PAST algorithm to mitigate impulsive noise. For example, in [33], a Robust PAST (RPAST) was proposed, which consists of two main steps. Firstly, the algorithm identifies the presence of impulsive noise by applying a threshold, and then removes any undesirable effects by discarding contaminated observations. The threshold is determined empirically using a function of the noise variance, assuming that the error vectors are subject to a Gaussian distribution that has been corrupted by additive impulsive noise.

In their work, Zhang et al. proposed a variant of PAST, known as MCC-PAST, that utilizes the Maximum Correntropy Criterion (MCC) to address impulsive noise, as described in [34, 35, 36]. The MCC-PAST algorithm incorporates a correntropy, a new statistic that can capture both the temporal structures and the statistics of two random processes, to mitigate the effects of impulsive noise. The objective function of PAST is modified by replacing the mean square error criterion with the MCC. The MCC-PAST algorithm is implemented using the RLS technique, and a variable forgetting factor technique is introduced to improve its tracking performance.

In parallel, Shengyang et al. developed a different variant of PAST, called BNC-PAST, which aims to track the underlying subspace using a novel criterion [37]. To deal with non-Gaussian noise with a heavy-tailed distribution, the authors introduced the concept of Bounded Non-linear Covariance (BNC) and utilized bounded nonlinear maps to eliminate the effect of impulsive noise. As a result, a new robust PAST algorithm based on BNC was derived.

2.3.2.2 Adaptive Kalman filtering

Adaptive Kalman filtering is another viable option for coping with impulsive noise. Liao et al. presented a robust subspace tracking (RST) algorithm that employs an adaptive Kalman Filter with a Variable number of Measurements (KFVM) in [38] and its variant the adaptive Kalman Filter with a Variable Number of Measurements in [39]. KFVNM enables the tracking of fast-varying subspaces by dynamically adjusting the number of past observations used in the recursion process [39]. To handle impulsive noise, the algorithm employs the M-estimate technique. However, the KFVNM-based algorithm is computationally more demanding than PAST-based algorithms, especially when a large number of observations are used for the subspace update.

2.3.2.3 Weighted recursive Least-Squares method

In a recent work, Linh-Trung *et al.* proposed a robust subspace tracking algorithm with linear computational complexity, called ROBUst Subspace Tracking Algorithm (ROBUSTA), based on a weighted RLS approach using robust statistics [6]. The convergence analysis of ROBUSTA was provided in the presence of SIRV noise, which showed that the algorithm corresponds to adaptive robust covariance estimation. The results showed that ROBUSTA outperformed many state-of-the-art algorithms for various types of impulsive noise including burst noise, SIRV noise, and α-stable noise. Moreover, ROBUSTA can be easily adapted by incorporating pre-processing steps to handle α-stable noise.

As for the robustness in terms of missing data and outliers, the literature is very rich in methods that deal with the subject. Indeed, one can refer to the recently published survey [40] to have a full idea of what has been done so far.

2.4 Conclusion

In this chapter, after stating the problem and the main objectives of our work, we have tackled the state of the art regarding both the subspace tracking field

and the robust subspace tracking one.

From what we could have gathered, one can notice that despite the missing data and outliers cases being often handled, the works about impulsive noise remain very thin. Reason for which, we will focus our efforts on this area for the upcoming chapters.

Chapter 3

Robust Subspace Tracking

3.1 Introduction

As mentioned in the previous chapter, the Subspace tracking field has long attracted researchers due to its various applications. For that matter, let's recall that the main aim is not only to estimate an orthonormal basis of the principal subspace but to do it in an adaptive way as well. Such an approach is very convenient, especially for online processing, however, it is often subject to computational cost and complexity limitations. This gave the motivation for the appearance of fast subspace tracking algorithms such as [10, 12, 13, 15, 16, 24, 26].

Despite their efficiencies, the latter algorithms assume the noise in (2.1) to be of Gaussian-centered white distribution and thus lack robustness. While our interest here is focused on the impulsive noise environment represented in the sequel by either burst noise or α-stable distribution.

For that concern, one can find in the literature a few robust algorithms including the RPAST algorithm [33] and ROBUSTA [6]. It has been shown in [6] that the ROBUSTA outperforms the other algorithms in this case. Indeed, the algorithm proposes to replace the least squares criterion of the subspace estimation by a more appropriate weighted criterion WLS "Weighted LS". This helps significantly to reduce the effect of corrupted data through the choice of an appropriate attenuation coefficient. Yet, this algorithm uses the

projection approximation introduced in [15], while it has been pointed in [26] that a more accurate approximation exists.

In this chapter, we present our proposed robust algorithm Mahalanobis FAPI (MFAPI) to impulsive noise that is derived from the FAPI approximation in [26], and exploit the weighted criterion to deal with impulsiveness, as well as its low-cost version, the Hard Thresholding FAPI (HTFAPI).

the remaining of this chapter is organized as follows: Section 3.2 gives some details about the related works. The proposed algorithms are presented and detailed in section 3.3. Simulation results are given in section 3.4. While section 3.5 concludes the chapter.

3.2 Related works

In order to estimate and track the signal subspace (represented by the $n \times p$ matrix $\mathbf{D}(t)$) of the covariance matrix $\mathbf{C}_x(t) = E[\mathbf{x}(t)\mathbf{x}^H(t)]$, it has been shown in [15] that one can minimize, under unitary constraint on $\mathbf{D}(t)$, the following cost function:

$$J\left(\mathbf{D}(t)\right) = \sum_{i=1}^{t} \beta^{t-i} \left\|\mathbf{x}(i) - \mathbf{D}(t)\mathbf{D}^H(t)\mathbf{x}(i)\right\|^2 \tag{3.1}$$

where $0 < \beta \leq 1$ is a forgetting factor. To simplify this non-linear optimization problem, a projection approximation was proposed in [15], for slow time-varying systems. The latter consists of approximating $\mathbf{y}(i) = \mathbf{D}^H(t)\mathbf{x}(i) \approx \mathbf{D}^H(i-1)\mathbf{x}(i)$. This leads to a simplified cost function:

$$J'\left(\mathbf{D}(t)\right) = \sum_{i=1}^{t} \beta^{t-i} \left\|\mathbf{x}(i) - \mathbf{D}(t)\mathbf{y}(i)\right\|^2 \tag{3.2}$$

The resolution of this latter is done usually by a recursive least squares algorithm, conducting the method to converge promptly and at a low computational cost. However, the obtained basis is not orthonormal. Reason for which, the algorithm OPAST was proposed in [16] where a low complexity orthonormalization was introduced.

Algorithm 1 (table 1): FAPI algorithm

Initialization

$\mathbf{D}(0) = \begin{bmatrix} \mathbf{I}_p \\ 0_{(n-p)\times p} \end{bmatrix}$; $\mathbf{Z}(0) = \mathbf{I}_p$

for each time step do :

input vector $\mathbf{x}(t)$

$\mathbf{y}(t) = \mathbf{D}^H(t-1)\mathbf{x}(t)$

$\mathbf{h}(t) = \mathbf{Z}(t-1)\mathbf{y}(t)$

$\epsilon^2(t) = ||\mathbf{x}(t)||^2 - ||\mathbf{y}(t)||^2$

$\mathbf{g}(t) = \frac{\mathbf{h}(t)}{\beta + \mathbf{y}^H(t)\mathbf{h}(t)}$

FAPI common section

$\tau(t) = \frac{\epsilon^2(t)}{1+\epsilon^2(t)||\mathbf{g}(t)||^2+\sqrt{1+\epsilon^2(t)||\mathbf{g}(t)||^2}}$

$\eta(t) = 1 - \tau(t)||\mathbf{g}(t)||^2$

$\mathbf{y}'(t) = \eta(t)\mathbf{y}(t) + \tau(t)\mathbf{g}(t)$

$\mathbf{h}'(t) = \mathbf{Z}^H(t-1)\mathbf{y}'(t)$

$\boldsymbol{\epsilon}(t) = \frac{\tau(t)}{\eta(t)}(\mathbf{Z}(t-1)\mathbf{g}(t) - (\mathbf{h}'(t)^H\mathbf{g}(t))\mathbf{g}(t))$

$\mathbf{Z}(t) = \frac{1}{\beta}(\mathbf{Z}(t-1) - \mathbf{g}(t)\mathbf{h}'(t)^H + \boldsymbol{\epsilon}(t)\mathbf{g}^H(t))$

$\mathbf{e}'(t) = \eta(t)\mathbf{x}(t) - \mathbf{D}(t-1)\mathbf{y}'(t)$

$\mathbf{D}(t) = \mathbf{D}(t-1) + \mathbf{e}'(t)\mathbf{g}^H(t)$

Later on, a novel projection approximation was proposed in [41]: $\mathbf{D}(t) \approx \mathbf{D}(t-1)\Theta(t)$ where $\Theta(t)$ is an orthonormal matrix of size $p \times p$. This results in the cost function given by:

$$J''(\mathbf{D}(t)) = \sum_{i=1}^{t} \beta^{t-i} \left\|\mathbf{x}(i) - \mathbf{D}(t)\Theta(t)\mathbf{y}(i)\right\|^2 \tag{3.3}$$

In [41], it was shown that this method named API, leads to a more accurate projection approximation than the one proposed in [15]. Moreover, the estimated basis of the signal subspace is perfectly orthonormal. Finally, a fast implementation of this method, i.e. Fast API (FAPI), has been proposed in [26], where both orthonormality and linear complexity are satisfied.

This FAPI algorithm, which provides a basis for the next robust methods, is summarized in table 1.

3.3 Robust PSA algorithms for impulsive noise case

In [6], where PSA is considered in an impulsive noise environment, the least squares criterion in (3.1) is replaced by the weighted least squares criterion (denoted J_i where i stands for impulsive) given by equation (3.5). The weighting factor $\omega(t)$, usually considered for the fixed-point robust estimation of the covariance matrix, e.g. [42], is given by

$$\omega(t) = \frac{1}{\mathbf{x}^H(t)\mathbf{C}_x^{-1}(t-1)\mathbf{x}(t)} \tag{3.4}$$

This factor represents a 'soft' threshold which takes small values when large impulsive noise values occur so that the impact of erroneous data is mitigated.

$$J_i(\mathbf{D}(t)) = \sum_{i=1}^{t} \beta^{t-i}\omega(i) \left\|\mathbf{x}(i) - \mathbf{D}(t)\mathbf{D}^H(t)\mathbf{x}(i)\right\|^2 \tag{3.5}$$

By applying the FAPI's projection approximation, the previous cost function is modified as:

$$J_i'(\mathbf{D}(t)) = \sum_{i=1}^{t} \beta^{t-i}\omega(i) \left\|\mathbf{x}(i) - \mathbf{D}(t)\Theta(t)\mathbf{y}(i)\right\|^2 \tag{3.6}$$

To solve this new cost function, one can follow the pattern of the FAPI's algorithm as follows: From the power iteration method in [43], the computation of $\mathbf{D}(t)$ reduces to a data compression (projection) step (3.7) and an orthonormalization step at each iteration (3.8):

$$\mathbf{C}_{xy}(t) = \mathbf{C}_x(t)\mathbf{D}(t-1) \tag{3.7}$$

$$\mathbf{D}(t)\mathbf{R}(t) = \mathbf{C}_{xy}(t) \tag{3.8}$$

Where $\mathbf{C}_{xy}$ is an $n \times p$ correlation matrix between the data vectors $\mathbf{x}(t)$ and $\mathbf{y}(t)$, $\mathbf{R}(t)$ is a $p \times p$ matrix such that $\mathbf{R}^H(t)\mathbf{R}(t) = \Phi(t)$ and $\Phi(t) = \mathbf{C}_{xy}^H\mathbf{C}_{xy}$.

Thus, $\mathbf{R}^H(t)$ is a square root of $\Phi(t)$. In order to reduce the complexity, the projection approximation proposed in [26] is used, according to:

$$\mathbf{D}(t) \simeq \mathbf{D}(t-1)\Theta(t) \tag{3.9}$$

Where the $p \times p$ matrix $\Theta(t)$ is orthonormal. Considering $\mathbf{D}(t-1)$ to be orthonormal, $\Theta(t)$ is then obtained as:

$$\Theta(t) = \mathbf{D}(t-1)^H \mathbf{D}(t) \tag{3.10}$$

For a fast implementation of the latter, one must proceed through the following steps:

3.3.1 Recursive data compression

The projection approximation can be used in (3.7) to compute $\mathbf{C}_{xy}(t)$ recursively, then, in (3.8) to update $\mathbf{D}(t)$ as described below.

First, let us set a robust estimate of the covariance matrix $\mathbf{C}_x$ according to:

$$\mathbf{C}_x(t) = \beta \mathbf{C}_x(t-1) + \omega(t)\mathbf{x}(t)\mathbf{x}(t)^H \tag{3.11}$$

Substituting (3.11) into (3.7) yields to

$$\mathbf{C}_{xy}(t) = \beta \mathbf{C}_x(t-1)\mathbf{D}(t-1) + \omega(t)\mathbf{x}(t)\mathbf{y}(t)^H \tag{3.12}$$

Then, applying the projection approximation at time $t-1$ leads to

$$\mathbf{C}_{xy}(t) = \beta \mathbf{C}_{xy}(t-1)\Theta(t-1) + \omega(t)\mathbf{x}(t)\mathbf{y}(t)^H \tag{3.13}$$

3.3.2 Recursive orthonormalization

The recursive computation of $\mathbf{D}(t)$ requires the introduction of an intermediate $p \times p$ matrix $\mathbf{Z}(t)$ [26]. Let's define $\mathbf{S}(t) \triangleq \big(\mathbf{R}(t)\Theta(t)\big)^H$ a $p \times p$ non-singular matrix and

$$\mathbf{Z}(t) \triangleq \mathbf{S}(t)^{-1} \tag{3.14}$$

Where the symbol $\stackrel{\Delta}{=}$ refers to equal by definition.

To find a recursion to the latter, we substitute equation (3.8) into equation (3.13) and left multiply it by $\mathbf{D}(t-1)^H$. Then considering the projection approximation, we have:

$$\Theta(t)\mathbf{R}(t) = \beta\mathbf{S}^H(t-1) + \omega(t)\mathbf{y}(t)\mathbf{y}(t)^H \tag{3.15}$$

After applying the Schur matrix inversion lemma to the latter, we obtain

$$\Big(\Theta(t)\mathbf{R}(t)\Big)^{-1} = \frac{1}{\beta}\mathbf{Z}(t-1)^H\Big(\mathbf{I}_p - \mathbf{y}(t)\mathbf{g}(t)^H\Big) \tag{3.16}$$

where:

$$\mathbf{g}(t) = \frac{\omega(t)\mathbf{h}(t)}{\beta + \omega(t)\mathbf{h}(t)^H\mathbf{y}(t)} \tag{3.17}$$

and

$$\mathbf{h}(t) = \mathbf{Z}(t-1)\mathbf{y}(t) \tag{3.18}$$

Now, by left multiplying the complex conjugate transpose of (3.16) by $\Theta(t)^H$ and right multiplying it by $\Theta(t)^{-H}$, one obtain

$$\mathbf{Z}(t) = \frac{1}{\beta}\Theta(t)^H(\mathbf{I}_p - \mathbf{g}(t)\mathbf{y}(t)^H)\mathbf{Z}(t-1)\Theta(t)^{-H} \tag{3.19}$$

Next, in the aim of finding a recursion of $\mathbf{D}(t)$, we substitute equation (3.8) into equation (3.13) and right multiply it by $\Theta(t)$ to get

$$\mathbf{D}(t)\mathbf{S}(t)^H = \Big(\beta\mathbf{D}(t-1)\mathbf{S}(t-1)^H + \omega(t)\mathbf{x}(t)\mathbf{y}(t)^H\Big)\Theta(t) \tag{3.20}$$

Then substituting equation (3.15) into the above yields to

$$\mathbf{D}(t)\mathbf{S}(t)^H = \mathbf{D}(t-1)\Theta(t)\mathbf{S}(t)^H + \omega(t)\mathbf{e}(t)\mathbf{y}(t)^H\Theta(t) \tag{3.21}$$

where

$$\mathbf{e}(t) = \mathbf{x}(t) - \mathbf{D}(t-1)\mathbf{y}(t) \tag{3.22}$$

On the other hand, left multiplying (3.15) by $\mathbf{g}(t)^H$ and replacing $\mathbf{g}(t)$ by its

definition in (3.17) gives

$$\mathbf{g}(t)^H\Theta(t)\mathbf{R}(t) = \frac{\omega(t)}{\beta + \omega(t)\mathbf{h}(t)^H\mathbf{y}(t)}\Big(\beta\mathbf{h}(t)^H\mathbf{S}(t-1)^H + \omega(t)\mathbf{h}(t)^H\mathbf{y}(t)\mathbf{y}(t)^H\Big) \tag{3.23}$$

Using equations (3.14) and (3.18), we can conclude that

$$\beta\mathbf{h}(t)^H\mathbf{S}(t-1)^H = \beta\mathbf{y}(t)^H \tag{3.24}$$

Hence, substituting the latter into (3.23) leads to

$$\mathbf{g}(t)^H\Theta(t)\mathbf{R}(t) = \omega(t)\mathbf{y}(t)^H \tag{3.25}$$

Then substituting equation (3.25) into equation (3.21) and right multiplying it by $\mathbf{S}(t)^{-H}$ yields finally to the recursion

$$\mathbf{D}(t) = \Big(\mathbf{D}(t-1) + \mathbf{e}(t)\mathbf{g}(t)^H\Big)\Theta(t) \tag{3.26}$$

3.3.3 A fast solution for $\Theta(t)$

Since $\mathbf{D}(t-1)$ is orthonormal, then $\mathbf{e}(t)$ is orthogonal to $\mathbf{D}(t-1)$. Besides, considering the orthonormality of $\mathbf{D}(t)$ associated with the equation (3.26), we come to

$$\Theta(t)\Theta(t)^H = \Big(\mathbf{I}_p + \mathbf{g}(t)(\mathbf{e}(t)^H\mathbf{e}(t))\mathbf{g}(t)^H\Big)^{-1} \tag{3.27}$$

To fast compute a solution for (3.27), let's set $\epsilon(t)$ to be the square root of $\mathbf{e}(t)^H\mathbf{e}(t)$:

$$\begin{aligned}
\epsilon^2(t) &= \mathbf{e}^H(t)\mathbf{e}(t) \\
&= \mathbf{x}^H(t)\mathbf{x}(t) - \mathbf{x}^H(t)\mathbf{D}(t)\mathbf{y}(t) - \mathbf{y}^H(t)\mathbf{D}^H(t)\mathbf{x}(t) + \mathbf{y}^H(t)\mathbf{D}^H(t)\mathbf{D}(t)\mathbf{y}(t) \\
&= \mathbf{x}^H(t)\mathbf{x}(t) - \mathbf{y}^H(t)\mathbf{y}(t) = ||\mathbf{x}(t)||^2 - ||\mathbf{y}(t)||^2
\end{aligned} \tag{3.28}$$

Substituting (3.28) into (3.27) then applying the Schur matrix inversion

lemma, we get

$$\Theta(t)\Theta(t)^H = \mathbf{I}_p - \epsilon(t)^2\mathbf{g}(t)\mathbf{g}(t)^H/\rho(t) \tag{3.29}$$

Where $\rho(t) = 1 + \epsilon(t)^2||\mathbf{g}(t)||^2$. To further reduce the computation cost, we look for a solution of the special form

$$\Theta(t) = \mathbf{I}_p - \tau(t)\mathbf{g}(t)\mathbf{g}(t)^H \tag{3.30}$$

Where $\tau(t) = \epsilon(t)^2/\rho'(t)$. Manipulating equations (3.29) and (3.30), we find $\rho'(t) = \rho(t) + \sqrt{\rho(t)}$. For the sequel, let's set

$$\eta(t) = 1 - \tau(t)\mathbf{g}(t)^H\mathbf{g}(t). \tag{3.31}$$

3.3.4 Principal subspace update

Substituting (3.30) into (3.19) leads to

$$\mathbf{Z}(t) = \frac{1}{\beta}\Big(\mathbf{Z}(t-1) - \mathbf{g}(t)\mathbf{h}'(t)^H + \boldsymbol{\epsilon}(t)\mathbf{g}(t)^H\Big) \tag{3.32}$$

where

$$\mathbf{y}'(t) = \mathbf{y}(t)\eta(t) + \mathbf{g}(t)\tau(t) \tag{3.33}$$

$$\mathbf{h}'(t) = \mathbf{Z}(t-1)^H\mathbf{y}'(t) \tag{3.34}$$

$$\boldsymbol{\epsilon}(t) = \frac{\tau(t)}{\eta(t)}\Big(\mathbf{Z}(t-1)\mathbf{g}(t) - (\mathbf{h}'(t)^H\mathbf{g}(t))\mathbf{g}(t)\Big) \tag{3.35}$$

Finally, substituting (3.31) into (3.26) yields

$$\mathbf{D}(t) = \mathbf{D}(t-1) + \mathbf{e}'(t)\mathbf{g}(t)^H \tag{3.36}$$

Where $\mathbf{e}'(t) = \eta(t)\mathbf{x}(t) - \mathbf{D}(t-1)\mathbf{y}'(t)$.

3.3.5 Fast computation of $\omega(t)$

Now, the direct computation of the weighting coefficient $\omega(t)$ costs $O(n^2)$ flops which is relatively expensive. Next, we propose to exploit the covariance structure of the data model in (2.1) for its fast computation.

As in [6], the weight $\omega(t)$ is taken to be the inverse of the square of Mahalanobis distance, i.e.

$$\omega(t) = \frac{1}{d_M^2\Big(\mathbf{x}(t), \mathbf{C}_x^{-1}(t-1)\Big)} = \frac{1}{\mathbf{x}^H(t)\mathbf{C}_x^{-1}(t-1)\mathbf{x}(t)} \tag{3.37}$$

Indeed, one can see that when the observation is corrupted, the distance takes large values, and thus the weighting factor is small which mitigates the impact of the corrupted data.

A direct computation of this factor involves the inversion of the covariance matrix $\mathbf{C}_x^{-1}(t-1)$ which is computed recursively. By applying the matrix inversion lemma to (3.11), we obtain

$$\mathbf{C}_x^{-1}(t) = \frac{1}{\beta}\left[\mathbf{C}_x^{-1}(t-1) - \frac{\omega(t)\mathbf{C}_x^{-1}(t-1)\mathbf{x}(t)\mathbf{x}^H(t)\mathbf{C}_x^{-1}(t-1)}{\beta + \omega(t)\mathbf{x}^H(t)\mathbf{C}_x^{-1}(t-1)\mathbf{x}(t)}\right] \tag{3.38}$$

Let $\mathbf{K}(t) = \mathbf{C}_x^{-1}(t)$. Then, the previous equation can be rewritten as:

$$\mathbf{K}(t) = \frac{1}{\beta}\Big(\mathbf{K}(t-1) + \omega(t)\mathbf{u}(t)\mathbf{v}^H(t)\Big) \tag{3.39}$$

Where

$$\mathbf{u}(t) = \mathbf{K}(t-1)\mathbf{x}(t) \quad \text{and} \quad \mathbf{v}(t) = \frac{\mathbf{u}(t)}{\beta + \omega(t)\mathbf{x}^H(t)\mathbf{u}(t)}$$

We can then deduce:

$$\omega(t) = \frac{1}{\delta(t)}, \quad \delta(t) = \mathbf{x}^H(t)\mathbf{u}(t) \tag{3.40}$$

The overall cost of this direct computation of the weighting factor $\omega(t)$ is of order $O(n^2)$. For a faster computation of the latter, let us set $\mathbf{D}_s$ to be an orthonormal basis of the signal subspace, thus $\Pi_s = \mathbf{D}_s\mathbf{D}_s^H$ is the orthogonal

projection matrix on this subspace. Given the structure of the covariance matrix $\mathbf{C}_x$, we can write

$$\begin{aligned}\mathbf{x}(t)^H\mathbf{C}_x^{-1}\mathbf{x}(t) &= \mathbf{x}(t)^H\Pi_s\mathbf{C}_x^{-1}\Pi_s\mathbf{x}(t) + (\mathbf{x}(t)^H(\mathbf{I}-\Pi_s)\mathbf{x}(t))/\sigma_n^2(t) \\ &= \mathbf{x}_s(t)^H\mathbf{C}_{x_sx_s}^{-1}\mathbf{x}_s(t) + \frac{||\mathbf{x}(t)||^2 - ||\mathbf{y}(t)||^2}{\sigma_n^2(t)}\end{aligned} \tag{3.41}$$

Where $\mathbf{C}_{x_s} = E[\mathbf{x}_s(t)\mathbf{x}_s(t)^H]$ and $\mathbf{x}_s(t) = \mathbf{D}_s^H\mathbf{x}(t)$. Now, let's approximate $\mathbf{D}_s$ with the estimated subspace basis $\mathbf{D}(t)$ and $\mathbf{x}_s(t)$ by $\mathbf{y}(t)$. Since $\mathbf{Z}(t)$ can be seen as an estimate of $\mathbf{C}_y^{-1}(t)$, we can then put

$$\mathbf{x}(t)^H\mathbf{C}_x^{-1}(t-1)\mathbf{x}(t) \approx \mathbf{y}(t)^H\mathbf{h}(t) + \frac{||\mathbf{x}(t)||^2 - ||\mathbf{y}(t)||^2}{\sigma_n^2(t-1)} \tag{3.42}$$

On the other hand, one can estimate the noise power $\sigma_n^2(t)$ as

$$\sigma_n^2(t) = \frac{tr\big(\mathbf{C}_x(t)\big) - tr\big(\mathbf{C}_y(t)\big)}{n-p} \tag{3.43}$$

where $tr(.)$ refers to the matrix trace operator. Considering that $T_x(t) = tr\big(\mathbf{C}_x(t)\big)$ and $T_y(t) = tr\big(\mathbf{C}_y(t)\big)$, then (3.43) can be effectively calculated using

$$\begin{aligned}T_x(t) &= \beta T_x(t-1) + \omega(t)||\mathbf{x}(t)||^2 \\ T_y(t) &= \beta T_y(t-1) + \omega(t)||\mathbf{y}(t)||^2\end{aligned}$$

We can now write

$$\omega(t) = 1/(\mathbf{y}(t)^H\mathbf{h}(t) + \frac{(||\mathbf{x}(t)||^2 - ||\mathbf{y}(t)||^2)(n-p)}{T_x(t-1) - T_y(t-1)}) \tag{3.44}$$

This algorithm, referred to as MFAPI, achieves a robust PSA and is summarized in table 2.

Algorithm 2 (table 2): MFAPI algorithm

1: **Initialization**

2: $\mathbf{D}(0) = \begin{bmatrix} \mathbf{I}_p \\ \mathbf{0}_{(n-p)\times p} \end{bmatrix}$; $\mathbf{Z}(0) = \mathbf{I}_p$; $T_x = 0; T_y = 0;$

for each time step do :

3: *input vector* $\mathbf{x}(t)$

4: $\mathbf{y}(t) = \mathbf{D}^H(t-1)\mathbf{x}(t)$

5: $\mathbf{h}(t) = \mathbf{Z}(t-1)\mathbf{y}(t)$

6: $\epsilon^2(t) = ||\mathbf{x}(t)||^2 - ||\mathbf{y}(t)||^2$

fast computation of the weight

7: $\delta'(t) = \mathbf{y}(t)^H\mathbf{h}(t) + \frac{(||\mathbf{x}(t)||^2 - ||\mathbf{y}(t)||^2)(n-p)}{T_x(t-1) - T_y(t-1)}$

8: $\omega(t) = 1/\delta'(t)$

9: $T_x(t) = \beta T_x(t-1) + \omega(t)||\mathbf{x}(t)||^2$

10: $T_y(t) = \beta T_y(t-1) + \omega(t)||\mathbf{y}(t)||^2$

11: $\mathbf{g}(t) = \frac{\mathbf{h}(t)\omega(t)}{\beta + \mathbf{y}^H(t)\mathbf{h}(t)\omega(t)}$

12: **FAPI common section** *tb. 1*, lines: (8-15)

3.3.6 HTFAPI for robust PSA

To further reduce the computation cost, we propose now a new method [1] based on order statistics (OS) [44] to determine the value of the weighting factor in such a way the corrupted data are disregarded.

Given the fact that the quantity $\epsilon^2(t)$ in (3.28) is positive valued, this new method consists of thresholding it to determine whether the new weighting factor $\omega(t)$ takes 0 or a 1 value at time instant t.

Indeed, since $\epsilon^2(t)$ represents approximately (up to a constant factor) the instantaneous noise power, and given the fact that impulsive noise realizations would have instantaneous power much higher and less frequent than the median power value, we propose here to use the well known Inter Quartile Range (IQR) method[1] for outliers detection [45, 46]. Using a sliding window of size L, we define:

$$\tilde{\epsilon}(t) = \{\epsilon^2(t-L+1) \cdots \epsilon^2(t)\}$$

[1]The threshold considered by this method has been chosen in such a way that, for Gaussian distribution, about 99% of the data falls under its value.

Algorithm 3 (table 3): HTFAPI algorithm

1: **Initialization**
2: $\mathbf{D}(0) = \begin{bmatrix} \mathbf{I}_p \\ \mathbf{0}_{(n-p)\times p} \end{bmatrix}$; $\mathbf{Z}(0) = \mathbf{I}_p$;

for each time step do :
3: *input vector* $\mathbf{x}(t)$
4: $\mathbf{y}(t) = \mathbf{D}^H(t-1)\mathbf{x}(t)$
5: $\mathbf{h}(t) = \mathbf{Z}(t-1)\mathbf{y}(t)$
6: $\epsilon^2(t) = ||\mathbf{x}(t)||^2 - ||\mathbf{y}(t)||^2$
compute weight:
7: $\tilde{\epsilon}(t) = \left[\epsilon^2(t-L+1)\cdots\epsilon^2(t)\right]$
8: $IQR(t) = Q3\big(\tilde{\epsilon}(t)\big) - Q1\big(\tilde{\epsilon}(t)\big)$
9: $threshold(t) = Q3\big(\tilde{\epsilon}(t)\big) + 1.5IQR(t)$
10: **if** $\epsilon^2(t) < threshold(t)$ **then**
11: $\omega(t) = 1$
12: **else**
13: $\omega(t) = 0$
14: **end if**
15: $\mathbf{g}(t) = \frac{\mathbf{h}(t)\omega(t)}{\beta + \mathbf{y}^H(t)\mathbf{h}(t)\omega(t)}$
16: **FAPI common section** *tb. 1*, lines: (8-15)

$$IQR(t) = Q3\big(\tilde{\epsilon}(t)\big) - Q1\big(\tilde{\epsilon}(t)\big)$$

$$threshold(t) = Q3\big(\tilde{\epsilon}(t)\big) + 1.5IQR(t) \tag{3.45}$$

Where $Q1(.)$ and $Q3(.)$ represent respectively the lower and the upper quartiles. Note that the adaptive computation of these quartiles is of negligible cost and many algorithms exist for its fast, adaptive evaluation, e.g. [47]. The weighting factor is then determined as follows:

$$\begin{cases} \omega(t) = 0 & \text{if } \epsilon^2(t) > threshold(t) \\ \omega(t) = 1 & \text{if } \epsilon^2(t) \leq threshold(t) \end{cases} \tag{3.46}$$

This algorithm, named HTFAPI (Hard Threshold FAPI), is summarized in table 3.

3.4 Simulations and results

To assess the performances of the proposed algorithms, we run several simulation experiments relative to the adverse scenarios considered in this chapter.

The performance quality is measured via the following criteria evaluated for a sample size of $N = 5000$ (in all simulations, we consider the sampling period as unit time reference, then we will not use the unit time). The estimation process are averaged over 100 independent runs:

- Subspace Tracking Error [6] (i refers to the i-th run):

$$STE_i(t) = \frac{tr\Big(\mathbf{D}_i^H(t)\big(\mathbf{I} - \mathbf{D}_{ex}(t)\mathbf{D}_{ex}^H(t)\big)\mathbf{D}_i(t)\Big)}{tr\Big(\mathbf{D}_i^H(t)\big(\mathbf{D}_{ex}(t)\mathbf{D}_{ex}^H(t)\big)\mathbf{D}_i(t)\Big)} \tag{3.47}$$

  where $\mathbf{D}_i(t)$ is the subspace estimate at time step t and $\mathbf{D}_{ex}(t)$ is the exact (orthonormal) principal subspace weight matrix.

- Orthonormality Error [15]:

$$OE_i(t) = ||\mathbf{I} - \mathbf{D}_i^H(t)\mathbf{D}_i(t)||_F^2 \tag{3.48}$$

  where $||.||_F$ denote the Frobenius norm.

Based on the previous metrics, the algorithms accuracy is compared to the SVD applied adaptively to the robust estimate of the covariance matrix $\mathbf{C}_x(t)$ given in (3.11), as well as the Orthogonal ROBUSTA (OROBUSTA) algorithm from [6] and the RPAST algorithm from [33]. They are also compared to the OPAST algorithm [16] to showcase their performance in contrast to the one of an adaptive standard but 'non-robust' algorithm. The forgetting factor in all algorithms is set to $\beta = 0.999$, hence, The newest data is more taken into consideration over the old one.

In all scenarios, we generate signals of the form 2.2. In the sequel, we consider observations $\mathbf{x}(t)$ of length $n = 80$ and signals $\tilde{\mathbf{s}}(t)$ of size $p = 4$.

Note also that our comparisons are done with existing algorithms of similar computational complexity (except for the SVD which is used here just for

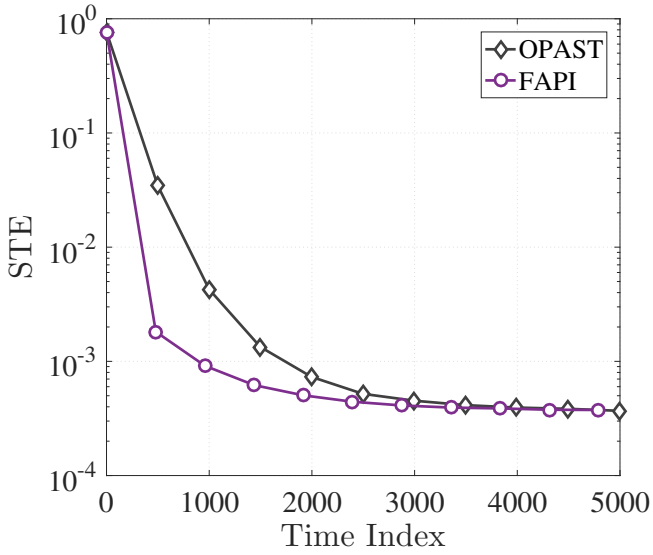

Figure 3.1: Principal subspace tracking error in a Gaussian noise environment

benchmarking). More precisely, all considered algorithms robust to impulsive noise (MFAPI, HTFAP, RPAST, OROBUSTA) are of linear complexity[2], i.e. $O(np)$ flops per iteration.

3.4.1 Gaussian noise environment

Here, we compare the FAPI algorithm with OPAST. The mixing matrix $\mathbf{A}$ is defined as $\mathbf{A} = \left[\mathbf{a}(\omega_1), \mathbf{a}(\omega_2), \cdots, \mathbf{a}(\omega_p)\right]$ with $\mathbf{a}(\omega_k) = \left[1; e^{j\omega_k}; \cdots; e^{j\omega_k(n-1)}\right]$. ω_k are chosen to be as in [26], hence $\omega_k \in [0.11; 0.08; 0.05; 0.025]$. The additive noise $\mathbf{n}(t)$ is a zero mean Gaussian white noise.

We can see from figure 3.1 that, besides the fact that the projection approximation in FAPI is more accurate to that of the OPAST, FAPI outperforms the OPAST in terms of convergence speed (at least for the first iterations). Reasons why, we have chosen the FAPI as a base for our proposed algorithms.

[2]There are slight differences in their respective numerical complexities, but the dominant cost remains of order $O(np)$.

3.4.2 Non-Gaussian noise environment

To assess the robustness performance of the proposed algorithms, we investigate two types of impulsive noise: Burst noise and α-stable noise. The mixing matrix $\mathbf{A}$ is generated randomly (with Gaussian entries).

3.4.2.1 Burst noise case

Here, we consider the noise $\mathbf{n}(t)$ in (2.4)

In the absence of an impulsive noise, the signals will each have an $SNR = 10dB$. However, in four distinct period of time: $P1 = [1000, 1050]$, $P2 = [2500, 2600]$, $P3 = [3500, 3550]$ and $P4 = [4000, 4050]$, an additive high amplitude Gaussian noise forces the SNR to drop to $-40dB$.

the threshold of the HTFAPI is defined over a time window equal to $L = 1000$.

Our simulation results, given in figures 3.2 and 3.3, show that in a burst noise environment, our proposed algorithms (ie. MFAPI and HTFAPI) perform almost as well as the SVD while maintaining a linear complexity and the orthonormality of the estimated weight matrix.

3.4.2.2 α-stable noise case

Now, we consider the impulsive noise $\mathbf{n}(t)$ in 2.1 to be α-stable distributed. Indeed, for small values of α, the mass of the tail becomes significant, representing a challenging problem.

One can note from figures 3.4, 3.5 and 3.6, that our algorithms still outperform the others up to $\alpha = 1.3$, with the MFAPI performing slightly better than the HTFAPI for the smaller values of α. Indeed, when the noise impulsiveness is high, the chosen IQR based threshold becomes less efficient due to the heaviness of the distribution tail.

Note that all listed algorithms handling non-adverse scenarios or designed for impulsive noise environments are of linear complexity of order $\mathcal{O}(np)$ flops per iteration.

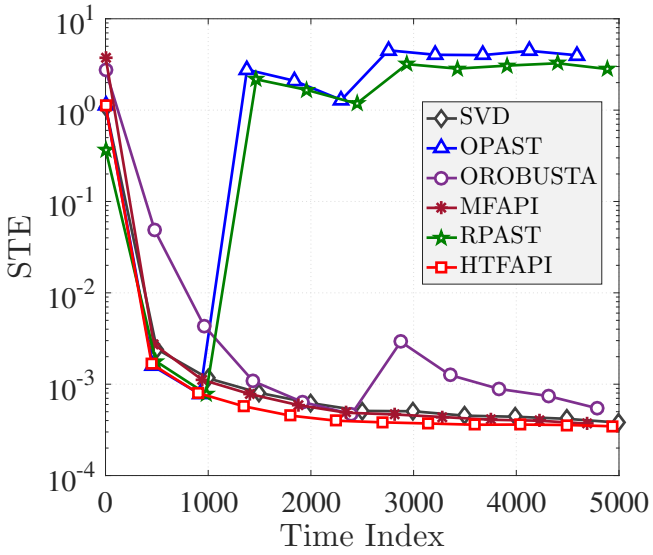

Figure 3.2: Subspace tracking error in an impulsive noise environment: Burst noise.

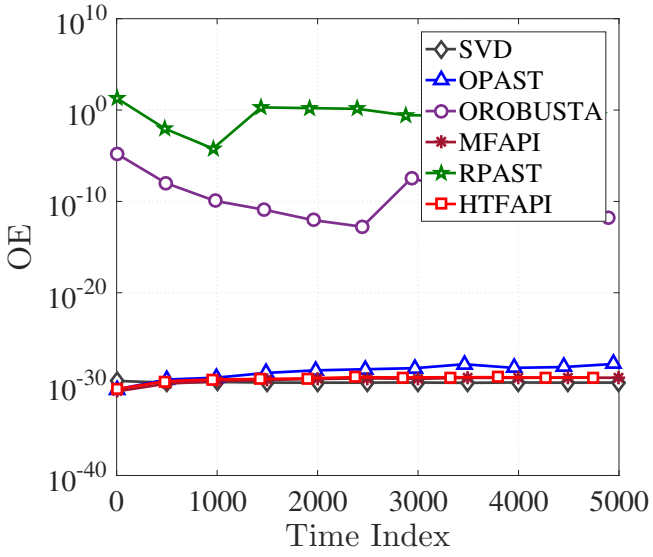

Figure 3.3: Orthonormality error in an impulsive noise environment: Burst noise.

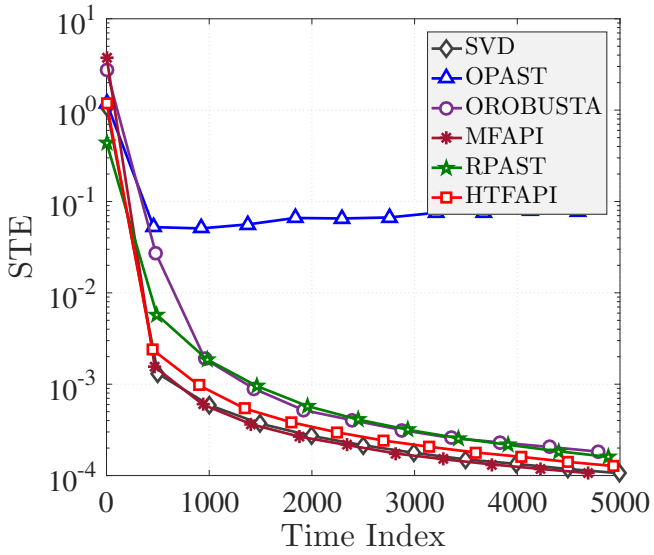

Figure 3.4: Principal Subspace tracking error in an impulsive noise environment: α-stable noise. $\alpha = 1.5$

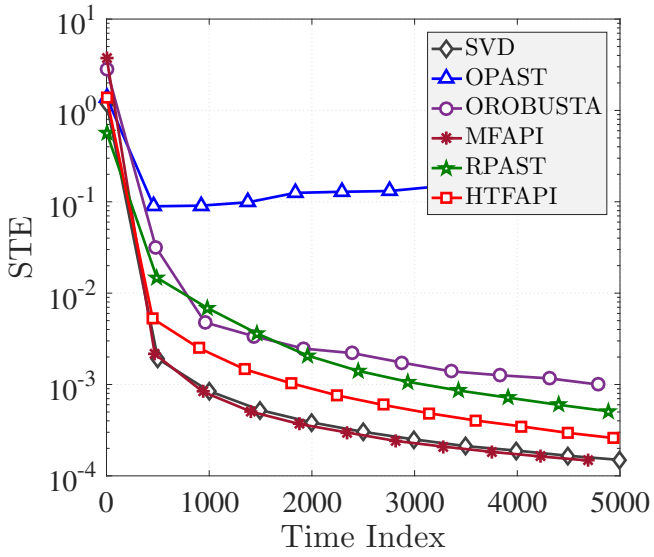

Figure 3.5: Principal subspace tracking error in an impulsive noise environment: α-stable noise. $\alpha = 1.4$

3.5 Conclusion

In this chapter, we provided new robust and fast adaptive algorithms for the estimation and tracking of the principal subspace. Indeed first, we en-

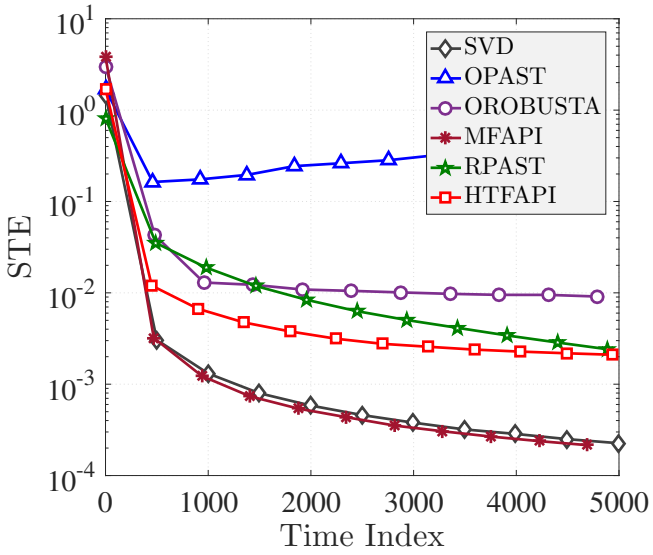

Figure 3.6: Principal subspace tracking error in an impulsive noise environment: α-stable noise. $\alpha = 1.3$

hanced the robustness of the Mahalanobis distance for the computation of the weighting factor by using the robust estimate of the covariance matrix, which yields to the MFAPI algorithm. Then we proposed new hard thresholding to determine the same weighting factor in order to reduce further the computational cost, this case corresponds to the HTFAPI algorithm. Finally, the effectiveness and accuracy of the proposed algorithms were assessed through simulated experiments where it has been shown that they outperform the existing methods in both accuracy and orthonormality.

In the next chapters, we will investigate the efficiency of the proposed algorithms for two applications. First, we utilize them in the localization field, by estimating and tracking the directions of arrival from the principal subspace. Then, we employ them in blind source separation after extracting the principal eigenvectors.

Chapter 4

PCA and its Application to Blind Source Separation

4.1 Introduction

As mentioned before, PSA and PCA are important problems involved in many applications such as array processing and source localization [48, 49], equalization and multi-user detection in communications [50], space-time adaptive processing for radar systems [51], blind system identification [52, 53, 54], background subtraction in computer vision [55], etc.

While for the PSA, only a basis of the principal subspace is considered, the PCA, our target in this chapter, consists of estimating the principal eigenvectors of the data covariance matrix. The conventional matrix algebraic approaches such as eigenvalue or singular value decompositions provide a good solution for batch processing, but they become inappropriate (too expensive) for real-time applications especially when the dimensions are large.

Hence, our objective was to develop adaptive solutions for the PCA, appropriate for streaming data or time-varying contexts. In the non-adverse scenario (e.g. in the Gaussian noise case), several effective solutions for PCA exist in the literature ranging from low complexity algorithms, e.g. [56] which cost $O(np)$ flops per iteration (n being the size of the observation vector and $p << n$ is the number of principal eigenvectors to estimate), to high com-

plexity algorithms with a cost of $O(n^2p)$ flops or more per time instant [57].

In our study here below, we are first, interested in adaptive PCA in adverse scenarios where impulsive noise or strong and sparse outliers affect the observations, along with the case of missing data.

Indeed, we extend the FAPI method [26] from PSA to the PCA using Givens rotations. The resulting algorithm referred to as GFAPI (G stands for Givens), extracts and tracks the PCA in non-adverse contexts (e.g. Gaussian noise case). This allows us to proceed in the same manner for the robust versions algorithms of the FAPI, hence MFAPI and HTFAPI, when dealing with impulsive noise such as burst noise and α-stable noise, which leads to the robust PCA algorithms Givens MFAPI (GMFAPI) and Givens HTFAPI (GHTFAPI).

Moreover, in the presence of sparse outliers and/or missing data, we propose a PCA method adapted from the robust PSA algorithm in [58] which relies on PETRELS method (Parallel Estimation and Tracking by REcursive Least Squares) [59] and the ADMM (Alternating Direction Method of Multipliers) technique for the outliers detection. We refer to this algorithm as PCAPA for Principal Component Analysis using PETRELS-ADMM.

In a second place, We are interested in the application of the PCA in the Blind Source Separation (BSS) field. Indeed, the BSS consists of the extraction of source signals from their observed mixtures without prior knowledge of the mixing matrix or its inputs. BSS is widely used in many signal processing applications, and a plethora of works have been devoted to develop solutions in different contexts and under different mixing models, e.g., [60, 61, 62, 63, 64, 65, 66]. In particular, second order statistics based methods are highly regarded due to their low computation load and efficiency to separate temporally coherent (colored) sources. These features make them suitable for adaptive scheme when dealing with streaming data. Several algorithms have been already dedicated to such an adaptive scheme including [63, 64, 67, 68, 69, 70, 71, 72]. However, most of the existing algorithms consider noise as being Gaussian or negligible. Whereas, in many applications, the measurements are affected by impulsive noise or outliers, e.g., [73, 74, 75], in which context standard methods fail to achieve the BSS. To deal with im-

pulsive noise, some authors have proposed robust batch BSS algorithms, e.g., [76, 77, 78, 79, 80, 66].

here, we propose to deal with both streaming data (i.e. adaptive scheme) and impulsive noise. Hence, a new approach that ensures both robustness and adaptivity is introduced based on second-order decorrelation approach.

To achieve the latter, we rely on the streaming data case and introduce the adaptive Second Order Blind Identification (SOBI) algorithm for the Gaussian noise case. In this algorithm, referred to as Adaptive SOBI (A-SOBI), the source separation is performed in two steps: whitening and joint diagonalization. The whitening is achieved using an adaptive PCA algorithm Givens-OPAST (GOPAST) followed by the joint diagonalization conducted on several non-zero lag correlation matrices. Similarly, our proposed Robust Adaptive SOBI (RA-SOBI) algorithm is derived via the use of our robust PCA tracking algorithms [1] for the whitening step. Moreover, to improve furthermore the robustness of the algorithm, we propose here to estimate the non-zero lag correlation matrices, considered in the joint diagonalization step, via robust estimation techniques [81].

The rest of this chapter is organized as follows. Our robust PCA algorithms are detailed in section 4.2. Then sections 4.3 and 4.4 present both the whitening and the joint diagonalization steps respectively. Algorithms performances and assessment are given in section 4.5. Finally, section 4.6 concludes this chapter.

4.2 Eigenvectors tracking: PCA

4.2.1 Impulsive noise

In order to extract the PCA, Thameri et al. proposed in [56] and [82] to apply Givens rotations to the principal subspace estimated from OPAST leading to the GOPAST algorithm. In the same manner, we propose here to exploit the FAPI algorithm's subspace estimate as well as Givens rotations to transform the orthonormal basis of the principal subspace $\mathbf{D}(t)$ into the desired principal eigenvectors basis (denoted $\mathbf{U}(t)$). Indeed, since the two

basis are orthonormal and span the same subspace, there exists an orthogonal matrix $\mathbf{Q}(t)$ that links the two previous basis, i.e. $\mathbf{U}(t) = \mathbf{D}(t)\mathbf{Q}(t)$. We compute matrix $\mathbf{Q}(t)$ iteratively as product of elementary Givens rotations. This algorithm will be referred to as GxFAPI (GFAPI, GMFAPI or GHTFAPI depending on which algorithm we use to estimate the subspace $\mathbf{D}(t)$).

A $p \times p$ Givens rotation denoted $\mathbf{G}_{l,m}(\theta, \phi)$ is a matrix equal to the identity except for its (l,l), (l,m), (m,l) and (m,m) entries. It is given by:

$$\mathbf{G}_{l,m}(\theta,\phi) = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & -s^* & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & s & \cdots & c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \tag{4.1}$$

where $c = \cos(\theta) \quad$ and $\quad s = \sin(\theta)e^{-j\phi}$.

To compute matrix $\mathbf{G}$ in equation (4.1), we consider the fact that the exact matrix $\mathbf{Z}' = (\mathbf{U}^H \mathbf{C}_x \mathbf{U})^{-1}$ is diagonal. Hence, every matrix $\mathbf{G}$ is computed in such a way one minimizes the departure from diagonal structure of $\mathbf{Z}' = \mathbf{G}\mathbf{Z}\mathbf{G}^H$ where $\mathbf{Z}$ is the matrix involved in FAPI algorithm (see table 1). This corresponds to writing $\mathbf{U} \approx \mathbf{D}\mathbf{G}^H$. The Givens parameters are obtained by minimizing the sum of the square modulus of the off diagonal elements of $\mathbf{Z}'$ according to:

$$(\theta, \phi) = \arg\min_{\theta,\phi} \sum_{a \neq b} |Z'_{ab}|^2 \tag{4.2}$$

After some straightforward derivations, the latter cost function can be shown to correspond to:

$$(\theta, \phi) = \arg\max_{\theta,\phi} |\mathbf{v}^T \mathbf{f}(t)|^2 \tag{4.3}$$

where

$$\mathbf{v} = \begin{bmatrix} \cos(2\theta) \\ \sin(2\theta)\cos(\phi) \\ \sin(2\theta)\sin(\phi) \end{bmatrix} \quad \text{and} \quad \mathbf{f}(t) = \begin{bmatrix} Z_{ll}(t) - Z_{mm}(t) \\ 2\Re\left(Z_{lm}(t)\right) \\ 2\Im\left(Z_{lm}(t)\right) \end{bmatrix}$$

$\Re(.)$ and $\Im(.)$ denote the real and imaginary parts of their argument and Z_{lm} is the (l,m)-th entry of matrix $\mathbf{Z}$. An optimal solution of (4.3) is finally given by $\mathbf{v} = [v_1, v_2, v_3]^T = sign(f_1(t))\mathbf{f}(t)/||\mathbf{f}(t)||$, (where $sign(f_1(t))$ refers to the sign of the first entry of $\mathbf{f}(t)$) and the angle parameters are obtained as:

$$c = \sqrt{\frac{v_1 + 1}{2}} \quad \text{and} \quad s = \frac{v_2 - jv_3}{2c}$$

Now, to select the rotation indices (l,m), different strategies can be considered. Also, several rotations can be applied at each time instant t. In our work, we have opted to use 2 rotations per time instant, as a good trade-off between complexity and the algorithm's accuracy. For the first rotation, the indices correspond to those of the off-diagonal element of $\mathbf{Z}(t)$ with the largest amplitude, i.e.

$$(l,m) = \arg \max_{\{(a,b)|a<b\}} |Z_{ab}(t)| \tag{4.4}$$

The second rotation's indices, denoted by (l',m'), are chosen in such a way that all indices are visited periodically along the iterations according to:

$$(l',m') = \begin{cases} (l, m+1) & \text{if } m < p \\ (l+1, l+2) & \text{if } m = p \text{ and } l < p-1 \\ (1,2) & \text{if } m = p \text{ and } l = p-1 \end{cases} \tag{4.5}$$

(l,m) being the chosen indices at time $t-1$. Table 4 summarizes this algorithm.

Algorithm 4 (table 4): GxFAPI algorithm

1: **Initialization**
$(l, m) = (1, 2)$

---

for each time step do :
2: *input:* $\mathbf{D}(t)$
1^{st} *rotation:*
3: Select rotation indices as in (4.4)
4: $\mathbf{f}(t) = \left[Z_{ll}(t) - Z_{mm}(t), 2\Re(Z_{lm}(t)), 2\Im(Z_{lm}(t))\right]^T$
5: $\mathbf{v} = sign(f_1(t))\mathbf{f}(t)/||f(t)||$
6: $c = \sqrt{\frac{v_1+1}{2}}$ and $s = \frac{v_2-v_3}{2c}$
7: Determine $\mathbf{G}$ as in equation (4.1)
8: $\mathbf{Z}(t) = \mathbf{G}\mathbf{Z}(t)\mathbf{G}^H$
9: $\mathbf{D}(t) = \mathbf{D}(t)\mathbf{G}^H$
10: $\mathbf{U}(t) = \mathbf{D}(t)$
2^{nd} *rotation*
11: Select rotation indices as in (4.5)
12: Apply the same steps as for the first rotation, lines: (4-10)

4.2.2 PCAPA algorithm for missing data and sparse outliers case

In this section, we focus on the adverse scenarios d, e and f in section 2.2.1.2. Indeed, the standard PCA algorithms performance may be degraded significantly if the measurement data are corrupted by outliers or missing observations [83]. Many recent papers highlighted the fact that latter scenario is ubiquitous and more and more frequent in large dimensional systems, e.g. [84].

Recently, an efficient algorithm, namely PETRELS-ADMM [58] has been proposed for robust subspace tracking (i.e. PSA). Our purpose here is to modify it in such a way, one achieves the PCA objective in such an adverse context. PETRELS-ADMM relies on two techniques to deal with both outliers and missing data. It uses the approach considered in PETRELS algorithm [59] to solve the PSA problem in the case of missing data. On the other side, it exploits the ADMM technique for the outliers detection. Once detected, the corrupted measurements are simply disregarded and treated as

Algorithm 5 (table 5): PCAPA algorithm

For each time step do:

Subspace estimation:

- Detect outliers unsing ADMM. Algorithm 2 in [58]
- Update $\mathbf{P}(t)$ to remove the outliers. Algorithm 1 in [58]
- Estimate $\mathbf{D}(t)$ using PETRELS. Algorithm 3 in [58]

PCA extraction:

- $\tilde{\mathbf{D}}(t) = \mathbf{D}(t)\big(\mathbf{D}^H(t)\mathbf{D}(t)\big)^{-1/2}$ % Orthonormalizing $\mathbf{D}(t)$
- $\tilde{\mathbf{y}}(t) = \tilde{\mathbf{D}}^H(t)\mathbf{x}(t)$
- $\mathbf{R}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}}(t) = \beta\mathbf{R}_{\tilde{\mathbf{y}}\tilde{\mathbf{y}}}(t-1) + \tilde{\mathbf{y}}(t)\tilde{\mathbf{y}}^H(t)$
- $\mathbf{U}(t) = diagonalization\ of\ \mathbf{R}_{\tilde{y}\tilde{y}}$ % $\mathbf{R}_{\tilde{y}\tilde{y}} = \mathbf{U}(t)\mathbf{\Lambda}(t)\mathbf{U}^H(t)$
- $\mathbf{U}(t) = \tilde{\mathbf{D}}(t)\mathbf{U}(t)$

missing data. The data model encompassing these two situations is described in (2.8):

In this work, we use Algorithm 2 in [58] for outliers detection which are then removed before applying PETRELS algorithm (i.e. Algorithm 1 in [58]) for subspace tracking in presence of missing data (see [58] for more details). PETRELS-ADMM output is a $n \times p$ matrix $\mathbf{D}$ whose columns span the principal subspace.

Now, in order to extract the p principal components of the data covariance matrix, we propose to proceed according to the three following steps:

- Orthonormalize matrix $\mathbf{D}$, using fast QR decomposition or simply by computing: $\tilde{\mathbf{D}} = \mathbf{D}(\mathbf{D}^H\mathbf{D})^{-1/2}$ which costs $O(np^2)$ flops per iteration.
- Compute $\tilde{\mathbf{y}} = \tilde{\mathbf{D}}^H\mathbf{x}$ as well as its (adaptively estimated) covariance matrix $\mathbf{R}_{\tilde{y}\tilde{y}}$. This costs $O(np)$ flops per iteration.
- Compute the eigendecomposition of $\mathbf{R}_{\tilde{y}\tilde{y}}$ according to $\mathbf{R}_{\tilde{y}\tilde{y}} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^H$, ($\mathbf{U}$ being the unitary eigenvectors matrix while $\mathbf{\Lambda}$ is the diagonal eigenvalues matrix). The desired principal components are obtained as $\tilde{\mathbf{D}}\mathbf{U}$. This final step costs again $O(np^2)$ flops per iteration.

This new PCAPA algorithm is summarized in table 5.

The overall cost of the resulting algorithm is of order $O(np^2)$. The latter can be reduced to $O(np)$ if one replaces PETRELS by OPAST and instead of

a full eigendecomposition, one applies only few Givens rotations. However, this cost reduction leads to a significant loss in terms of estimation accuracy, especially for large dimensional systems, and hence is not considered here.

4.3 Data whitening

This step consists of projecting the observed vector $\mathbf{x}(t)$ onto the principal subspace spanned by the column vectors of the mixing matrix $\mathbf{A}$, with the purpose of transforming it into a unitary matrix. The used matrix in this transformation is called the whitening matrix $\mathbf{W}$.

In [62], it has been shown that the whitening matrix $\mathbf{W}$ can be computed from the eigendecomposition of the covariance matrix of $\mathbf{x}(t)$, denoted $\mathbf{C}_x$, as follows:

$$\mathbf{C}_x = \mathbf{U}_s \mathbf{\Lambda} \mathbf{U}_s^H + \sigma^2 \mathbf{I} \tag{4.6}$$

$$\mathbf{W} = \mathbf{\Lambda}^{-1/2} \mathbf{U}_s^H \tag{4.7}$$

where $\mathbf{U}_s$ and $\mathbf{\Lambda}$ are the matrices of the p principal eigenvectors and eigenvalues of the noise-free covariance matrix. For a streaming data scheme, Thameri et al. proposed in [56, 68] to replace the exact eigenvectors and eigenvalues by their adaptive estimates using GOPAST algorithm according to:

$$\mathbf{W}(t) = \mathbf{\Lambda}(t)^{-1/2} \mathbf{U}_s^H(t) \tag{4.8}$$

In the same manner, in order to compute the whitening matrix $\mathbf{W}(t)$, we use here the robust estimates of the principal components $\mathbf{U}(t)$ obtained in the previous section, as well as the matrix $\mathbf{Z}(t)$, which represents the inverse of the eigenvalues up to a scaling factor, which results in preserving a linear cost i.e. $O(np)$ flops per iteration. This will lead us to the whitening equation:

$$\mathbf{W}(t) = \mathbf{Z}(t)^{1/2} \mathbf{U}^H(t) \tag{4.9}$$

Remark: In the above, $\mathbf{\Lambda}$ corresponds to the diagonal matrix of the principal

eigenvalues of the covariance matrix $\mathbf{C}_x$. Implicitly, this means that the noise term (i.e. noise power σ^2) has been neglected in (4.6). In case, the latter cannot be considered negligible, one can estimate it as shown in (3.43) and replace in (4.9) $\mathbf{\Lambda}(t)$ by $\mathbf{\Lambda}(t) - \sigma^2(t)\mathbf{I}$.

4.4 Joint diagonalization

After whitening, the mixing matrix is approximately reduced to a $p \times p$ unitary matrix denoted $\mathbf{W}^H(t)$ and hence the noiseless whitened signal can be written as $\tilde{\mathbf{x}}(t) = \mathbf{W}(t)\mathbf{x}(t) \approx \mathbf{B}^H(t)\mathbf{s}(t)$. To estimate the separation matrix $\mathbf{B}(t)$, A-SOBI uses a joint diagonalization of K correlation matrices $\mathbf{R}_t(\tau_k)$ corresponding to K chosen non-zero lags $\tau_1, \cdots, \tau_K$. The latter are adaptively estimated as:

$$\mathbf{R}_t(\tau_k) = \beta \mathbf{R}_{t-1}(\tau_k) + \hat{\mathbf{s}}(t)\hat{\mathbf{s}}^H(t - \tau_k) \tag{4.10}$$

Where $0 < \beta < 1$ is a forgetting factor and $\hat{\mathbf{s}}(t) = \mathbf{B}(t)\tilde{\mathbf{x}}(t)$.

To further enhance the robustness of the proposed algorithm, we propose to estimate the correlation matrices via robust estimation techniques.

Hence, since we consider several correlation matrices with different time lags for the diagonalization step, it is important to take into account whether the observations at these time lags are corrupted or valid. Thus, we propose here to weight the estimates of the correlations with a combination of $\omega(t)$ and $\omega(t - \tau_k)$ to ensure the mitigation of erroneous data.

Thus, the correlation matrices are estimated as

$$\mathbf{R}_t(\tau_k) = \beta \mathbf{R}_{t-1}(\tau_k) + \sqrt{\omega(t)\omega(t - \tau_k)}\hat{\mathbf{s}}(t)\hat{\mathbf{s}}^H(t - \tau_k) \tag{4.11}$$

Now, to achieve the joint diagonalization, it is first stated that the unitary separation matrix can be computed as a product of elementary Givens rotations:

$$\mathbf{B} = \prod_{l=1}^{p-1} \prod_{m=l+1}^{p} \mathbf{G}_{l,m}(\theta, \phi) \tag{4.12}$$

where $\mathbf{G}_{l,m}(\theta,\phi)$ is a $p \times p$ matrix equal to the identity except for its (l,l), (l,m), (m,l) and (m,m) entries. It is given by:

$$\mathbf{G}_{l,m}(\theta,\phi) = \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots & & \vdots \\ 0 & \cdots & c & \cdots & -s^* & \cdots & 0 \\ \vdots & & \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & s & \cdots & c & \cdots & 0 \\ \vdots & & \vdots & & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix} \tag{4.13}$$

where $c = \cos(\theta) \quad \text{and} \quad s = \sin(\theta)e^{-j\phi}$. Thanks to the mutual decorrelation of the source signals, the separation is achieved when the correlation matrices $\mathbf{R}(\tau_k)$ are diagonalized.

To perform the latter diagonalization in an adaptive way, one multiplies at each time step $\mathbf{R}(\tau_k)$ at its left and right sides by the elementary Givens rotation $\mathbf{G}(t)$ according to:

$$\mathbf{R}'(\tau_k) = \mathbf{G}(t)\mathbf{R}(\tau_k)\mathbf{G}^H(t) \tag{4.14}$$

The rotation indices l and m are selected to be the corresponding indices of the entries of largest amplitude according to:

$$(l,m) = \arg \max_{\{(a,b)|a<b\}} \sum_{i=1}^{K} |\mathbf{R}_{ab}(\tau_i)| \tag{4.15}$$

Remark: Another way to select rotation indices (l,m) at time t would be to visit periodically along the iterations all entries of the correlation matrices, according to:

$$(l,m) = \begin{cases} (l', m'+1) & \text{if } m < p \\ (l'+1, l'+2) & \text{if } m = p \text{ and } l' < p-1 \\ (1,2) & \text{if } m = p \text{ and } l' = p-1 \end{cases} \tag{4.16}$$

(l', m') being the chosen indices at time $t-1$. Also, instead of one single rotation per time instant, one can use two rotations with indices chosen according to the previous two methods in order to increase the algorithm's convergence rate, at the cost of increased computational complexity.

Finally, the rotation angles are obtained by minimizing the sum of off-diagonal elements of the K considered correlation matrices:

$$(\theta, \phi) = \arg\min_{\theta,\phi} \sum_{a \neq b} \sum_{i=1}^{K} |\mathbf{R}'_{ab}(\tau_i)|^2 \tag{4.17}$$

This is proven to be equivalent to solving

$$(\theta, \phi) = \arg\max_{\theta,\phi} \mathbf{v}^H \mathbf{F}\mathbf{F}^H \mathbf{v} \tag{4.18}$$

where

$$\mathbf{v} = \begin{bmatrix} \cos(2\theta) \\ \sin(2\theta)\cos(\phi) \\ \sin(2\theta)\sin(\phi) \end{bmatrix}$$

and

$$\mathbf{F} = \begin{bmatrix} \mathbf{R}_{ll}(\tau_1) - \mathbf{R}_{mm}(\tau_1) & \cdots & \mathbf{R}_{ll}(\tau_K) - \mathbf{R}_{mm}(\tau_K) \\ 2\Re\left(\mathbf{R}_{lm}(\tau_1)\right) & \cdots & 2\Re\left(\mathbf{R}_{lm}(\tau_K)\right) \\ 2\Im\left(\mathbf{R}_{lm}(\tau_1)\right) & \cdots & 2\Im\left(\mathbf{R}_{lm}(\tau_K)\right) \end{bmatrix}$$

$\Re(\cdot)$ and $\Im(\cdot)$ denote the real and imaginary parts of their argument and $\mathbf{R}_{lm}(\tau_k)$ is the (l,m)-th entry of matrix $\mathbf{R}(\tau_k)$. An optimal solution is finally given by $\mathbf{v} = [v_1, v_2, v_3]^T = sign(u_1)\mathbf{u}$ (u_1 being the 1^{st} entry of $\mathbf{u}$) where $\mathbf{u}$ is the unit-norm eigenvector corresponding to the principal eigenvalue of matrix $\mathbf{F}\mathbf{F}^H$. The angle parameters are finally obtained as:

$$c = \sqrt{\frac{v_1 + 1}{2}} \quad \text{and} \quad s = \frac{v_2 - jv_3}{2c} \tag{4.19}$$

The above proposed RA-SOBI algorithm is summarized in the table 6

Algorithm 6 (table 6): RA-SOBI

Initialization

$\mathbf{B}(0) = \mathbf{I}_p$

---

for each time step do :

input: $\mathbf{Z}(t)$ *and* $\mathbf{U}(t)$ *from tb. 4*

$\mathbf{W}(t) = \mathbf{Z}(t)^{1/2}\mathbf{U}^H(t)$

for $k = 1, \cdots, K$

$\hat{\mathbf{s}}(t - \tau_k) = \mathbf{B}(t)\mathbf{W}(t)\mathbf{x}(t - \tau_k); k = 1, \cdots, K$

$\mathbf{R}_t(\tau_k) = \beta\mathbf{R}_{t-1}(\tau_k) + \sqrt{\omega(t)\omega(t - \tau_k)}\hat{\mathbf{s}}(t)\hat{\mathbf{s}}^H(t - \tau_k)$

end for

determine (l, m) as in (4.15) or (4.16)

$$\mathbf{F} = \begin{bmatrix} \mathbf{R}_{ll}(\tau_1) - \mathbf{R}_{mm}(\tau_1) & \cdots & \mathbf{R}_{ll}(\tau_K) - \mathbf{R}_{mm}(\tau_K) \\ 2\Re\left(\mathbf{R}_{lm}(\tau_1)\right) & \cdots & 2\Re\left(\mathbf{R}_{lm}(\tau_K)\right) \\ 2\Im\left(\mathbf{R}_{lm}(\tau_1)\right) & \cdots & 2\Im\left(\mathbf{R}_{lm}(\tau_K)\right) \end{bmatrix}$$

$\mathbf{u} = eigs(\mathbf{F}\mathbf{F}^H, 1)$

$\mathbf{v} = sign(u_1)\mathbf{u}$

$c = \sqrt{\frac{\tilde{v}_1+1}{2}}$ and $s = \frac{\tilde{v}_2-\tilde{v}_3}{2c}$

define $\mathbf{G}(t)$ as in (4.1)

for $k = 1, \cdots, K$

$\mathbf{R}(\tau_k) = \mathbf{G}(t)\mathbf{R}(\tau_k)\mathbf{G}^H(t)$

end for

$\mathbf{B}(t) = \mathbf{B}(t)\mathbf{G}^H(t)$

4.5 Simulations and results

4.5.1 PCA

To assess the proposed algorithms' performance related to the PCA problem, we run several simulation experiments relative to the adverse scenarios considered above.

Here the performance quality is measured via the PCA Tracking Error (PTE) criterion [85, 56] evaluated for a sample size of $N = 5000$ and averaged over 100 independent runs:

$$PTE_i(t) = ||\mathbf{U}_{ex}(t) - \mathbf{U}_i(t)||_F^2 \tag{4.20}$$

where $\mathbf{U}_{ex}(t)$ is the matrix of the exact principal eigenvectors of $\mathbf{C}_x(t)$. $||.||_F$

denote the Frobenius norm.

Based on the previous metric, the algorithms' accuracy is compared to the SVD applied adaptively to the robust estimate of the covariance matrix $\mathbf{C}_x(t)$ given in (3.11). They are also compared to the GOPAST algorithm [56, 82] to showcase their performance in contrast to the one of an adaptive standard but 'non-robust' algorithm. The forgetting factor in all algorithms is set to $\beta = 0.999$.

In the sequel, we consider observations $\mathbf{x}(t)$ of length $n = 80$ and signals $\tilde{\mathbf{s}}(t)$ of size $p = 4$.

Note also that our comparisons are done with existing algorithms of similar computational complexity (except for the SVD which is used here just for benchmarking). More precisely, all considered algorithms that are robust to impulsive noise (GMFAPI, GHTFAP, RPAST, OROBUSTA) are of linear complexity[1], i.e. $O(np)$ flops per iteration, while all considered algorithms dealing with missing data and outliers (PCAPA, PETRELS-CFAR, GRASTA) are of complexity order $O(np^2)$.

4.5.1.1 Gaussian noise

In this first experiment, we consider the scenario corresponding to the model described by equation 2.1 where the observation is affected by an additive Gaussian noise (a in 2.2.1.2).

Indeed, here, we compare GFAPI algorithm with GOPAST. The mixing matrix $\mathbf{A}$ is defined as $\mathbf{A} = \left[\mathbf{a}(\omega_1), \mathbf{a}(\omega_2), \cdots, \mathbf{a}(\omega_p)\right]$ with $\mathbf{a}(\omega_k) = \left[1; e^{j\omega_k}; \cdots; e^{j\omega_k(n-1)}\right]$. ω_k are chosen to be as in [26], hence $\omega_k \in [0.11; 0.08; 0.05; 0.025]$. The additive noise $\mathbf{n}(t)$ is a zero mean Gaussian white noise.

We can see from figures 4.1 that our proposed algorithm GFAPI, outperforms GOPAST in terms of convergence speed. This result is due, in one way, to the fact that FAPI outperforms OPAST, and hence GFAPI is slightly more accurate (at least for the first iterations) than GOPAST. On the other hand, the Givens rotations, when applied to the subspace estimate

[1]There are slight differences in their respective numerical complexities, but the dominant cost remains of order $O(np)$.

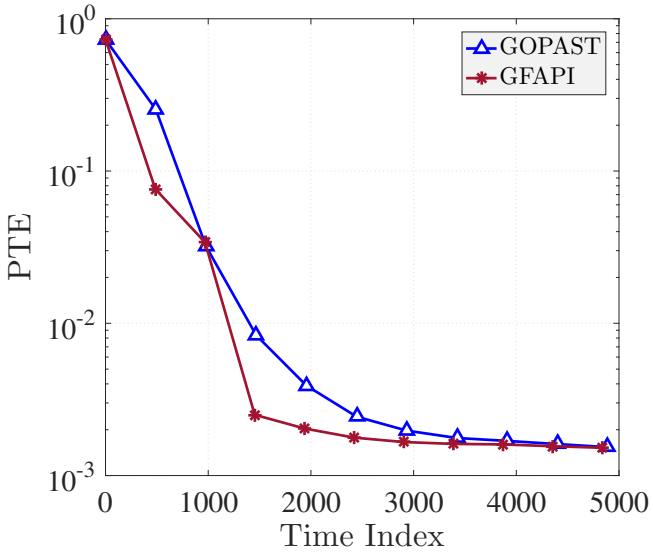

Figure 4.1: Principal components tracking error in a Gaussian noise environment

of OPAST, affect the correctness of the projection approximation used in this latter and impact its convergence rate as compared to GFAPI.

4.5.1.2 Impulsive noise

In this case, To assess the performance of the proposed algorithms, we investigate two types of impulsive noise: burst noise and α-stable noise (b and c in 2.2.1.2). The mixing matrix $\mathbf{A}$ is generated randomly (with Gaussian entries).

Burst noise case: In the absence of impulsive noise, the signals will each have an $SNR = 10dB$. However, in four distinct period of time: $P1 = [1000, 1050]$, $P2 = [2500, 2600]$, $P3 = [3500, 3550]$ and $P4 = [4000, 4050]$, an additive high amplitude Gaussian noise forces the SNR to drop to $-40dB$.

In this scenario, we aim to evaluate our proposed algorithms which handle the impulsive noise, namely the GMFAPI and the GHTFAPI (the threshold of the GHTFAPI is defined over a time window equal to $L = 1000$), by comparing them to GOPAST [56] and the SVD when applied to the robust covariance matrix estimate $\mathbf{C}_x(t)$ in (3.11). In addition, we have compared

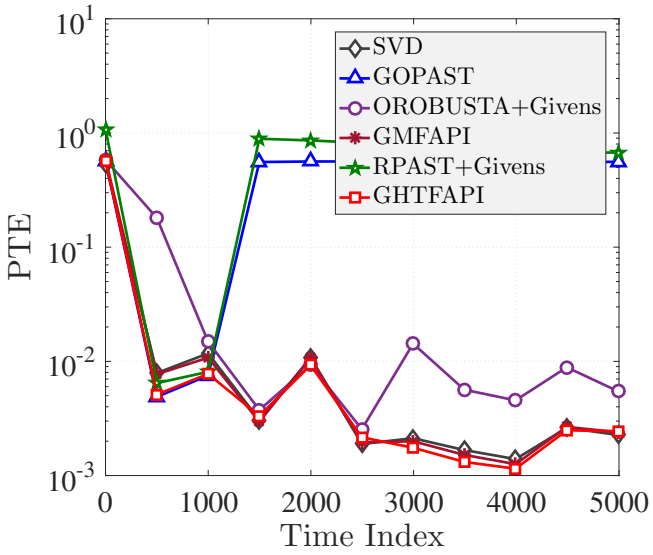

Figure 4.2: Principal components tracking error in an impulsive noise environment: Burst noise.

our proposed algorithms to existing adaptive PSA algorithms [33, 6] to which we have added the PCA extraction step via the use of two Givens rotations as for GFAPI.

Our simulation results given in figure 4.2, show that in a burst noise environment, our proposed algorithms perform almost as well as the SVD while maintaining a linear complexity.

α-stable noise case Now, we consider the impulsive noise $\mathbf{n}(t)$ in 2.1 to be α-stable distributed. For small values of α, the mass of the tail becomes significant, representing a challenging problem.

One can note from figures 4.3, 4.4 and 4.5, that our algorithms still outperform the others up to $\alpha = 1.3$, with the GMFAPI performing slightly better than the GHTFAPI for the smaller values of α. Indeed, when the noise impulsiveness is high, the chosen IQR based threshold becomes less efficient due to the heaviness of the distribution tail.

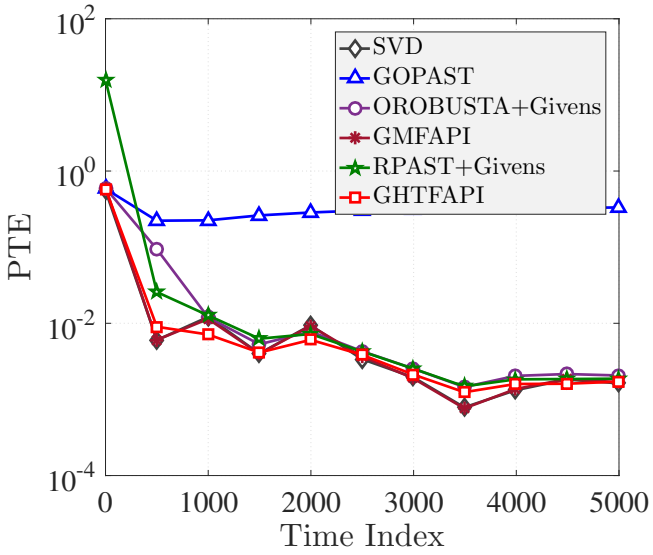

Figure 4.3: Principal components tracking error in an impulsive noise environment: α-stable noise. $\alpha = 1.5$

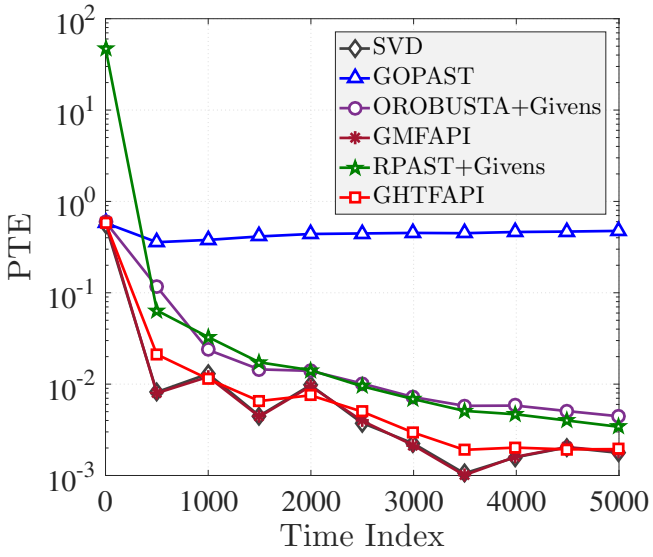

Figure 4.4: Principal components tracking error in an impulsive noise environment: α-stable noise. $\alpha = 1.4$

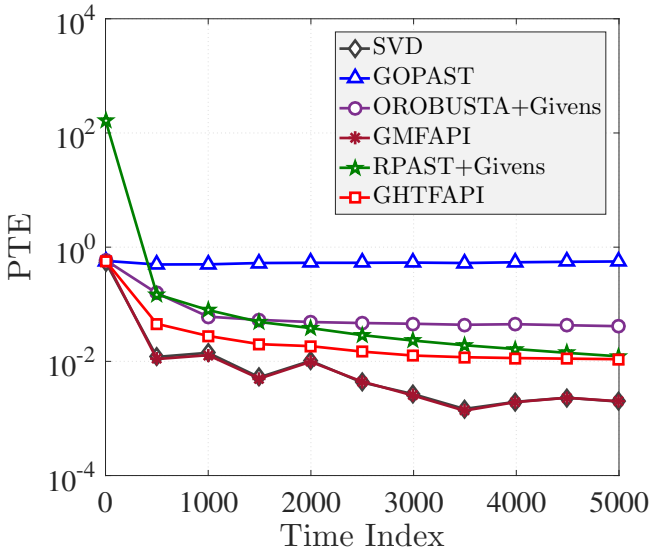

Figure 4.5: Principal components tracking error in an impulsive noise environment: α-stable noise. $\alpha = 1.3$

4.5.1.3 PCAPA

In this scenario, we aim to assess the performance of our last proposed algorithm dealing with missing data and outliers (d, e and f in 2.2.1.2). To do so, we run a simulation with the same observations as earlier for $SNR = 20dB$. The system might then be corrupted as in (2.8) with sparse outliers $\mathbf{i}(t)$ of density $\psi_{outliers}$ and i.i.d. magnitude uniformly distributed over $[0, 10]$. In addition, some entries might be non-observed (missed), with probability of missing data denoted $\psi_{missing}$.

We compare the algorithm's performance with the SVD when applied to the raw data. Also, for comparison purpose, we consider also the GRASTA [86] and the PETRELS-CFAR [6] (both are PSA adaptive algorithms) to which we added a PCA extraction step based of the proposed diagonalization approach.

First, we assess our algorithm's performance in the case where none of the entries are missing and no outliers occur. As can be seen from figure 4.6, its accuracy in this case is close to that of the SVD.

In figure 4.7, we considered the case where 20% of the data is missing and

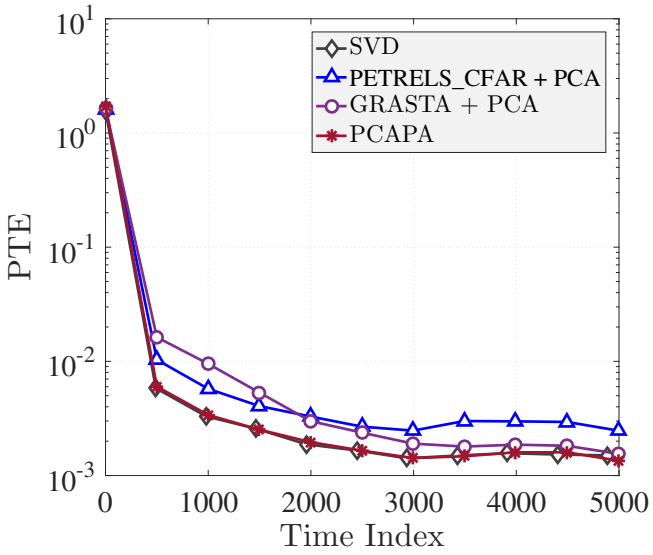

Figure 4.6: PCAPA performance comparison: absence of outliers and missing data

in figure 4.8 the case where 20% of the data is missing and 20% of it is affected by outliers. As can be seen, our dedicated PCAPA algorithm preserves its relatively good performance in these adverse contexts and outperforms both GRASTA and PETRELS-CFAR algorithms.

Next, we investigate in more details the impact of the outliers and missing data densities on the algorithm's performance.

Outliers impact In this experiment, we fix the missing data density low at $\psi_{missing} = 0.1$, and we vary the outliers density denoted $\psi_{outliers}$ in the range $[0.1, 0.6]$. We can see in figure 4.9 that the algorithm preserves a relatively good estimation performance up to $\psi_{outliers} = 0.4$, a situation where approximately half of the data is corrupted or missing (i.e. 10% of data missing + 40% of the observed data corrupted by outliers).

Missing data impact Now, we fix the outliers density low at $\psi_{outliers} = 0.1$, and we vary the $\psi_{missing}$ in the interval $[0.1, 0.6]$. From figure 4.10, we can see that, again, the algorithm still provides accurate eigenvectors estimation even when half of the entries are not observed.

Joint missing data and outliers impact: Finally, we assess the perfor-

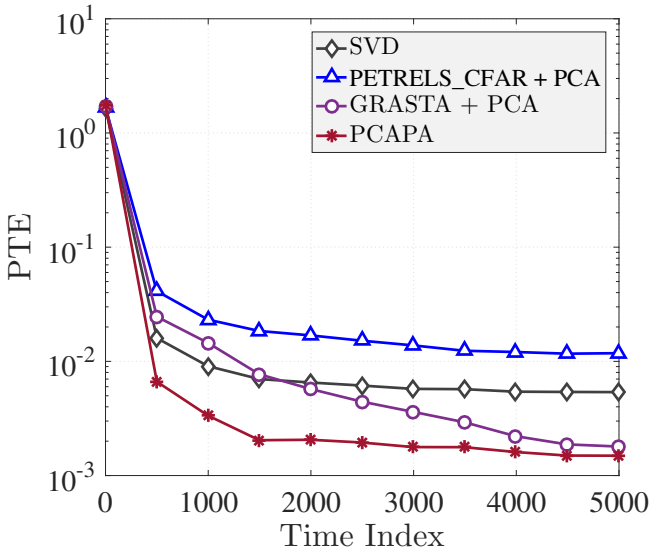

Figure 4.7: PCAPA performance comparison: 20% missing data case

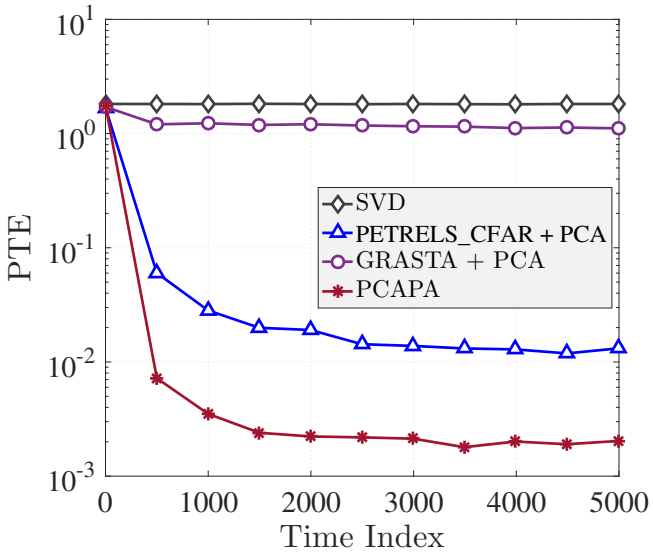

Figure 4.8: PCAPA performance comparison: 20% missing data and 20% outliers

mance of the algorithm in presence of both missing data and outliers. In this case, we increase the densities simultaneously so that both $\psi_{outliers}$ and $\psi_{missing}$ vary in the range $[0.1, 0.5]$.

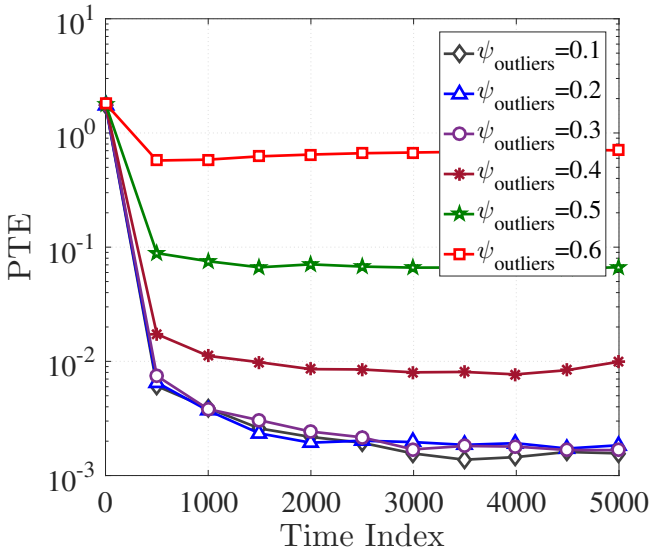

Figure 4.9: Impact of outliers density on PCAPA performance

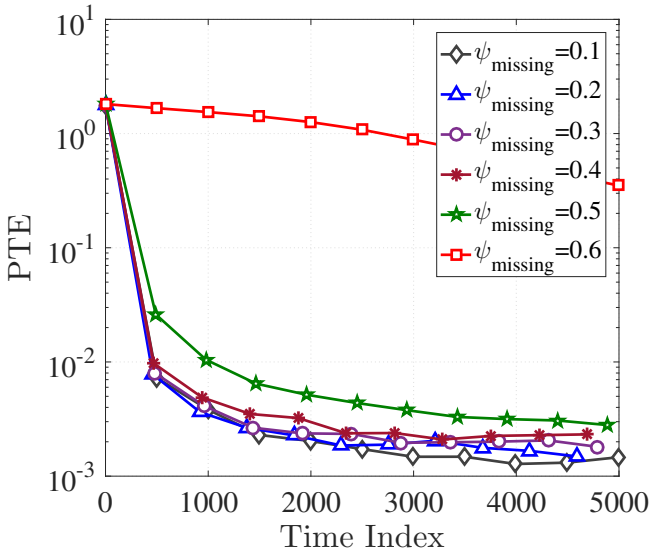

Figure 4.10: Impact of missing data density on PCAPA performance

One can notice in figure 4.11 that even in bad scenarios with high corruption rates, the algorithm still achieves a relatively good estimation accuracy (i.e., steady state PTE of order 10^{-2}).

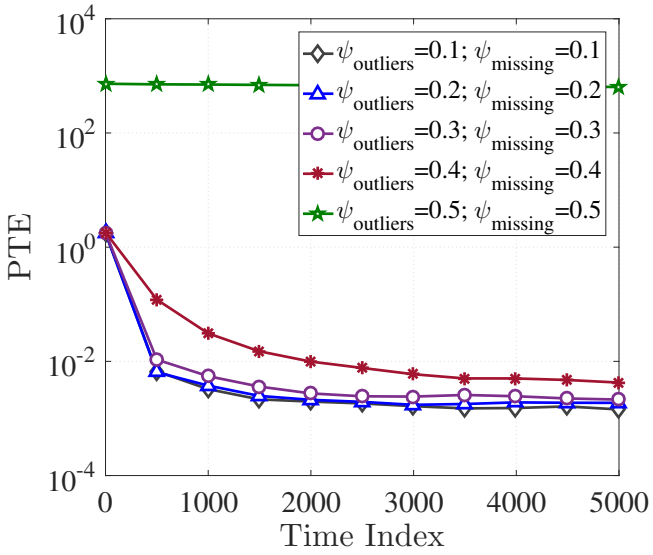

Figure 4.11: Impact of both outliers and missing data densities on PCAPA performance

4.5.2 BSS

In order to investigate our algorithms performance regarding blind source separation, we simulate the streaming data vectors $\mathbf{x}(t)$ of dimension $n = 8$ during $N = 6000$ time steps. Those observations are generated using $p = 3$ source signals $\tilde{\mathbf{s}}(t)$ corresponding to filtered complex circular white Gaussian processes by three AR filters of order 1 with respective coefficients $a_1 = 0.95\exp(j0.5)$, $a_2 = 0.75\exp(j0.7)$ and $a_3 = 0.55\exp(j0.3)$.

These signals are then mixed and corrupted with additive white centered Gaussian noise $\mathbf{n}_G(t)$ imposing an SNR of $5dB$.

We run $M = 100$ Monte Carlo simulations for all scenarios and we evaluate the algorithms performance using the mean rejection level defined in [62] as:

$$I(t) = \frac{1}{M}\sum_{m=1}^{M}\Big(\sum_{i=1}^{p}\frac{\sum_{j\neq i}|\mathbf{L}_{ij}(t)|^2}{|\mathbf{L}_{ii}(t)|^2}\Big) \tag{4.21}$$

Where $\mathbf{L}(t) = \mathbf{B}(t)\mathbf{A}(t)$ is close to a diagonal matrix (after removing the permutation indeterminacy).

Note that for all algorithms the forgetting factor is set to be $\beta = 0.999$

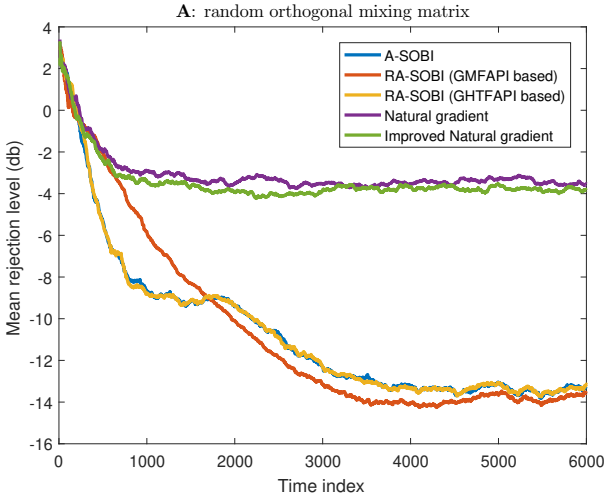

Figure 4.12: Algorithms performance in case of random mixing matrix and Gaussian noise only environment

and the number of considered correlation matrices is $K = 10$. Also, the number of samples used in the threshold determination for the GHTFAPI algorithm is $L = 1000$.

To make sure our algorithm performs well in an adaptive manner, we first run the simulation without adding the burst noise. From Fig. 4.12, we can clearly notice that our proposed algorithm RA-SOBI in its two versions as well as the A-SOBI outperform the other state of the art algorithms when dealing with Gaussian noise environment.

In addition, to evaluate the robustness of the algorithm, we simulate a burst noise as defined in (2.4) that occurs during four periods of time, namely: $P_1 = [1500, 1550]$, $P_3 = [2500, 2600]$, $P_3 = [3500, 3600]$ and $P_4 = [4500, 4600]$ causing the SNR to drop to $-40dB$.

Furthermore, we investigate three scenarios with different mixing matrices:

- First, we consider a general case with $\mathbf{A}$ being an $(n \times p)$ random matrix.
- Then, we investigate a scenario where the mixing matrix $\mathbf{A}$ is structured as in (2.3):

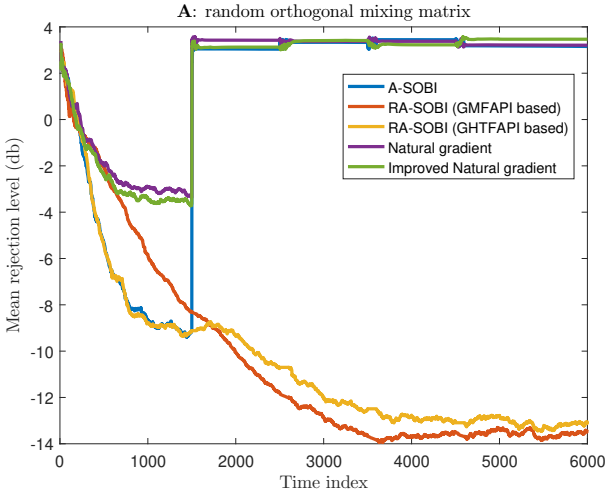

Figure 4.13: Algorithms performance in case of random mixing matrix

Where θ_k is a direction of arrival chosen in this experiment as $\theta_1 = 10°$, $\theta_2 = 30°$ and $\theta_3 = 50°$.

- Finally, we study the case where the system is slowly time varying. For that, we use the latter structure of $\mathbf{A}$ while linearly varying the directions of arrival such that it begins with $\theta_1(0) = 20°$, $\theta_2(0) = 10°$, $\theta_3(0) = -10°$ and it ends at $\theta_1(N-1) = 30°$, $\theta_2(N-1) = 0°$, $\theta_3(N-1) = -10°$ (i.e. θ_3 is kept time invariant).

As we can see from Fig. 4.13, Fig. 4.14, and 4.15, it is clear that our robust algorithm RA-SOBI (with its two versions) maintains a good source separation throughout the entire testing period, while the non-robust algorithms (A-SOBI, natural gradient and the improved natural gradient [68]) collapse at the occurrence of the first noise impulse.

4.6 Conclusion

In this chapter, we have introduced robust adaptive algorithms for blind source separation based on second-order decorrelation. The latter was achieved thanks to robust fast whitening, as well as Givens rotations-based joint diagonalization performed on robust estimates of correlation matrices. Our

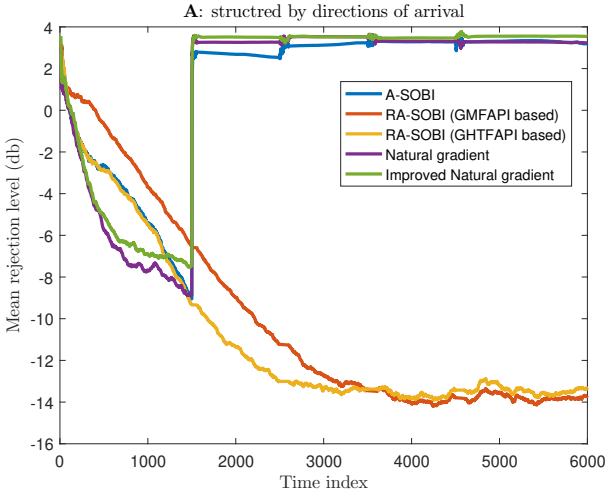

Figure 4.14: Algorithms performance in case of directions of arrival dependent mixing matrix

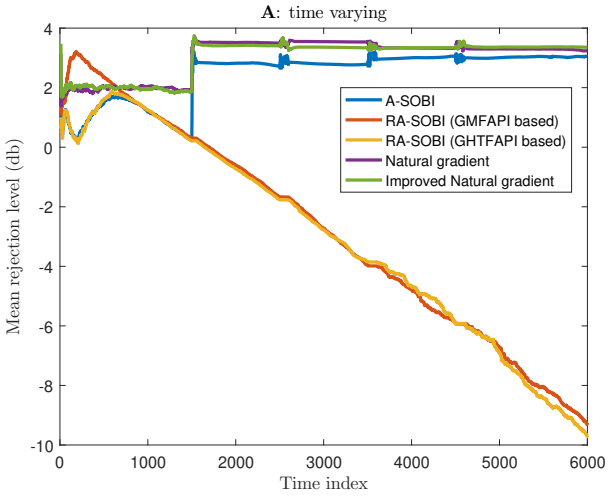

Figure 4.15: Algorithms performance in case of time-varying mixing matrix

algorithms are shown, via simulation experiments, to be effective in an impulsive noise environment, while having low computational complexity of order $O(np + pK)$ flops per iteration.1

Chapter 5

State Representation based on Subspace Tracking: Direction of Arrivals Estimation and Tracking

5.1 Introduction

Directions of Arrival (DoA) estimation and tracking have long presented a great interest in signal processing. In the array signal processing literature, we can come across several research axes on this subject. The high-resolution methods that appeared in that context were the MUltiple SIgnal Classification (MUSIC) method [87] and the Estimation of Signal Parameters via Rotational Invariance Techniques (ESPRIT) algorithm [88]. However, these methods are very expensive in terms of complexity due to the eigendecomposition of the data covariance. Thankfully, adaptive algorithms with linear complexity have emerged to lower the cost of this eigendecomposition step, such as the PAST [15], OPAST [16], PAST with the deflation technique (PASTd) [15], FAPI [26] and FDPM [24]. Furthermore, to improve the results of these algorithms, several works have proposed to add a Kalman filter to smooth the tracking of the DoA [89, 90, 91]. Despite their efficiency, these algorithms lack of robustness. Indeed, in the presence of impulsive noise, these algorithms diverge and the estimation error of the DoA becomes too

large. In this context, one can find several subspace tracking algorithms that deal with this problem [6, 33, 92, 58]. It has been shown in [1] that our algorithm called MFAPI and its lower cost version HTFAPI out-perform the other algorithms in the presence of data corrupted by impulsive noise or outliers.

As seen in the previous chapter, the latter algorithms propose to replace the least squares criterion of the subspace estimation with a more appropriate weighted criterion WLS in order to mitigate the impact of corrupted data through the choice of a proper attenuation coefficient as in [6]. In addition, they use the projection approximation proposed in [26] which helps to reduce significantly the computational cost.

In the sequel, we propose to smooth the results obtained by the HTFAPI algorithm by using a Kalman filter. More precisely, the directions of arrival given by HTFAPI followed by a Total Least Squares ESPRIT (TLS-ESPRIT), are considered to be measurements for Kalman's filter. Besides, the parameters of the latter are chosen dynamically according to the weighting coefficient of the algorithm HTFAPI.

The rest of this chapter is organized as follows. Section 5.2 details the problem statement of a moving point modelization. Then, the Kalman filter principal is depicted in section 5.3. Moreover, the filter $\alpha\beta$ is introduced in section 5.4. The proposed smoothing algorithm is presented in section 5.5, while the results of simulations are given in section 5.6. Finally, section 5.7 concludes this chapter.

Note: The Kalman filter related variables in this chapter are preceded with a subscript K to not create confusion.

5.2 Moving point Modelization

5.2.1 the model

Let's consider a moving point for which we aim to represent its state in space. The model of this point sits on two differential equations (scalar or vectorial). the first one describes the system dynamic whilst the second one

represents the measurement. The continuous model is given by the following two equations

$$ {}_{\mathbf{K}}\dot{\mathbf{x}}(t) = f({}_{\mathbf{K}}\mathbf{x}(t); {}_{\mathbf{K}}\mathbf{u}(t); {}_{\mathbf{K}}\mathbf{w}(t); t) \tag{5.1} $$

$$ {}_{\mathbf{K}}\mathbf{z}(t) = h({}_{\mathbf{K}}\mathbf{x}(t); {}_{\mathbf{K}}\mathbf{v}(t); t) \tag{5.2} $$

The n_x dimensional state vector ${}_{\mathbf{K}}\mathbf{x}(t)$ holds the mobile dynamics' evolution parameters. ${}_{\mathbf{K}}\mathbf{u}(t)$ is the user entry vector of dimension n_u (it represents known parameters such as a command or the observer position). ${}_{\mathbf{K}}\mathbf{z}(t)$ is the n_z dimensional measurement vector. For simplicity, the noises ${}_{\mathbf{K}}\mathbf{w}(t)$ and ${}_{\mathbf{K}}\mathbf{v}(t)$ are considered mutually independent zero mean white sequences and are usually additive and Gaussian.

${}_{\mathbf{K}}\mathbf{w}(t)$ is called the noise process and represents the modelization errors. it can also represent the exact model lack of knowledge [93], e.g. when a target performs a maneuver while following a second-order motion (see further in this section) we have to use a noise with a large variance. And while the noise process is introduced by the user, the measurement noise ${}_{\mathbf{K}}\mathbf{v}(t)$ is related to the measurement mechanism.

$f(\cdot)$ and $h(\cdot)$ are mostly non-linear functions, but for simplicity and/or implementation concerns, we will consider them linear in the above. Now, if the system is discrete or is processed through a discretization, the previous equations become [94]

$$ {}_{\mathbf{K}}\mathbf{x}_{t+1} = f_t({}_{\mathbf{K}}\mathbf{x}_t; {}_{\mathbf{K}}\mathbf{u}_t; {}_{\mathbf{K}}\mathbf{w}_t) \tag{5.3} $$

$$ {}_{\mathbf{K}}\mathbf{z}_t = h_t({}_{\mathbf{K}}\mathbf{x}_t; {}_{\mathbf{K}}\mathbf{v}_t) \tag{5.4} $$

With the index t being discrete. To simplify the writing, we omitted the sampling time T by replacing tT with t.

In this manuscript, we will consider the above discrete model. In particular, the linear discrete model, as is applied in practice to a wild selection of applications. It is given by the following differential equations:

$$ {}_{\mathbf{K}}\mathbf{x}_{t+1} = {}_{\mathbf{K}}\mathbf{F}_{\mathbf{K}}\mathbf{x}_t + {}_{\mathbf{K}}\mathbf{G}_{\mathbf{K}}\mathbf{u}_t + {}_{\mathbf{K}}\Gamma_{\mathbf{K}}\mathbf{w}_t \tag{5.5} $$

$$ {}_{\mathbf{K}}\mathbf{z}_t = {}_{\mathbf{K}}\mathbf{H}_{\mathbf{K}}\mathbf{x}_t + {}_{\mathbf{K}}\mathbf{v}_t \tag{5.6} $$

Where:

- ${}_{\mathbf{K}}\mathbf{F}$ is the transition matrix from the instant t to the instant $t+1$
- ${}_{\mathbf{K}}\mathbf{G}$ is the weighting matrix of entry ${}_{\mathbf{K}}\mathbf{u}_t$. In the absence of the latter, the matrix ${}_{\mathbf{K}}\mathbf{G}$ is null.
- The matrix ${}_{\mathbf{K}}\mathbf{H}$ is called the measurement matrix.
- the sequence ${}_{\mathbf{K}}\mathbf{w}_t$ is often assumed Gaussian and centered with a covariance matrix equal to an identity matrix multiplied by $\sigma^2_{\mathbf{w}}$. Thus, the noise process is centered and of the covariance matrix

  $$ {}_{\mathbf{K}}\mathbf{Q} = E[({}_{\mathbf{K}}\Gamma_{\mathbf{K}}\mathbf{w}_t)({}_{\mathbf{K}}\Gamma_{\mathbf{K}}\mathbf{w}_t)'] = \sigma^2_{\mathbf{w}\mathbf{K}}\Gamma_{\mathbf{K}}\Gamma' $$

  , where $E[\cdot]$ is the statistical mean.
- Likewise, the measurement noise ${}_{\mathbf{K}}\mathbf{v}_t$ is centered and is associated to the covariance matrix ${}_{\mathbf{K}}\mathbf{R} = E[{}_{\mathbf{K}}\mathbf{v}_t\ {}_{\mathbf{K}}\mathbf{v}'_t]$

5.2.2 the state vector

A state vector in the tracking field can contain various information about the mobile according to its dynamics. Indeed, a mobile can move on a single axis (1D), in a plane (2D), or in space (3D). Here we will detail the (2D) case as the other cases can easily be deduced from it. In addition, the nature of the mobile's movement can be modeled by a constant velocity 5.7, a constant acceleration 5.8, or it can be described by a hybrid model containing both acceleration and rotary movements 5.9.

$$ {}_{\mathbf{K}}\mathbf{x}_{CV} = \begin{bmatrix} x & y & V_x & V_y \end{bmatrix}' \tag{5.7} $$

$$ {}_{\kappa}\mathbf{x}_{CA} = \begin{bmatrix} x & y & V_x & V_y & a_x & a_y \end{bmatrix}' \tag{5.8} $$

$$ {}_{\kappa}\mathbf{x}_{R} = \begin{bmatrix} x & y & V_x & V_y & {}_{\kappa}\omega \end{bmatrix}' \tag{5.9} $$

When choosing the state vector, one has to take into account the system dynamics. Thus, for the sequel, we will consider the constant velocity and the constant acceleration cases.

5.2.3 System's dynamics

In the tracking field, we often divide mobile points into two categories: maneuvering and non-maneuvering mobiles [95]. Indeed, a mobile moving according to a constant velocity is considered to be in a non-maneuvering mode, otherwise, it is in a maneuvering one. By applying the mechanics' rules of a mobile point moving at a constant velocity, and assuming the sum of forces applied to the said mobile of mass m at the instant t is $\overrightarrow{F}$, then we have:[96]:

$$ \begin{bmatrix} \overrightarrow{\ddot{x}}(t) \\ \overrightarrow{\ddot{y}}(t) \end{bmatrix} = \frac{1}{m} \begin{bmatrix} \overrightarrow{F_x}(t) \\ \overrightarrow{F_y}(t) \end{bmatrix} \tag{5.10} $$

When the acceleration is null, or more precisely nearly null, we consider the mobile in a non-maneuvering mode. In this case, the forces are mostly reduced to friction. Let consider only the x component and define the acceleration by $a_x(t) = F_x/m$, we thus obtain:

$$ \ddot{x}(t) = a_x(t) \tag{5.11} $$

5.2.3.1 Movement at a constant velocity: CV

The state vector of a point moving in the plan at a nearly constant velocity is composed of its position in the two directions, plus their respective velocities as shown in (5.7). In the absence of commands, assuming the two coordinates aren't correlated and considering a discrete model with an acceleration modeled by a white noise: the Discrete White Noise Acceleration

model (DWNA) leads to the following differential equation [93]:

$$_{\kappa}\mathbf{x}_{t+1} = {}_{\kappa}\mathbf{F}_{CV}\ {}_{\kappa}\mathbf{x}_t + {}_{\kappa}\Gamma_{CV}\ {}_{\kappa}\mathbf{w}_t \tag{5.12}$$

with:

$$_{\kappa}\mathbf{F}_{CV} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.13}$$

and

$$_{\kappa}\Gamma_{CV} = \begin{bmatrix} \frac{T^2}{2} & 0 & T & 0 \\ 0 & \frac{T^2}{2} & 0 & T \end{bmatrix}' \tag{5.14}$$

The covariance matrix of the noise process corresponding to this model is

$$_{\kappa}\mathbf{Q}_{CV} = \sigma^2_{\mathbf{w}} \begin{bmatrix} \frac{T^4}{4} & 0 & \frac{T^3}{2} & 0 \\ 0 & \frac{T^4}{4} & 0 & \frac{T^3}{2} \\ \frac{T^3}{2} & 0 & T^2 & 0 \\ 0 & \frac{T^3}{2} & 0 & T^2 \end{bmatrix} \tag{5.15}$$

In practice, $\sigma_{\mathbf{w}}$ must be within the interval $[0.5a_{max}; a_{max}]$ where a_{max} represents the maximal possible acceleration within a time step [93]. In order to obtain (5.12) [96]:

- We assume that the acceleration (5.11) is constant between to sampling instants Tt and $T(t+1)$.
- We suppose the two components to be independent between to sampling instants.
- Finally, we incorporate two times (5.11) between those two instants.

This model is very popular and it is known under the appellation 'discrete-time Constant-Velocity model' (CV), or more precisely, 'discrete-time Nearly-Constant-Velocity model' (NCV) [93]. This is due to the fact that a low acceleration in both directions is systematically modelized by the noise process.

5.2.3.2 Movement at a Constant Acceleration: CA

The state vector of a point moving in the plan at a constant acceleration is composed of its position, velocity, and acceleration in both directions as shown in (5.8). In the absence of commands, assuming the two coordinates aren't correlated and considering a discrete model with an acceleration modeled by a Wiener process: the 'Discrete Wiener Process Acceleration model' (DWPA) leads to the following differential equation :

$$ {}_{\kappa}\mathbf{x}_{t+1} = {}_{\kappa}\mathbf{F}_{CA}\ {}_{\kappa}\mathbf{x}_t + {}_{\kappa}\Gamma_{CA}\ {}_{\kappa}\mathbf{w}_t \tag{5.16} $$

with:

$$ {}_{\kappa}\mathbf{F}_{CA} = \begin{bmatrix} 1 & 0 & T & 0 & \frac{T^2}{2} & 0 \\ 0 & 1 & 0 & T & 0 & \frac{T^2}{2} \\ 0 & 0 & 1 & 0 & T & 0 \\ 0 & 0 & 0 & 1 & 0 & T \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.17} $$

and

$$ {}_{\kappa}\Gamma_{CA} = \begin{bmatrix} \frac{T^2}{2} & 0 & T & 0 & 1 & 0 \\ 0 & \frac{T^2}{2} & 0 & T & 0 & 1 \end{bmatrix}' \tag{5.18} $$

The covariance matrix of the noise process corresponding to this model is

$$
{}_{\mathbf{K}}\mathbf{Q}_{CA} = \sigma_{\mathbf{w}}^2 \begin{bmatrix} \frac{T^4}{4} & 0 & \frac{T^3}{2} & 0 & \frac{T^2}{2} & 0 \\ 0 & \frac{T^4}{4} & 0 & \frac{T^3}{2} & 0 & \frac{T^2}{2} \\ \frac{T^3}{2} & 0 & T^2 & 0 & T & 0 \\ 0 & \frac{T^3}{2} & 0 & T^2 & 0 & T \\ \frac{T^2}{2} & 0 & T & 0 & 1 & 0 \\ 0 & \frac{T^2}{2} & 0 & T & 0 & 1 \end{bmatrix} \tag{5.19}
$$

In practice, $\sigma_{\mathbf{w}}$ must be within the interval $[0.5\Delta a_{max}; \Delta a_{max}]$ where Δa_{max} represents the maximal possible increase in acceleration within a time step [93].

5.2.4 The measurement

In the tracking field, the measurement represents the image of the position given by the measurement device. In the absence of noise, the relation between the state vector and the measurement vector is given by:

$$
{}_{\mathbf{K}}\bar{\mathbf{z}} = \begin{bmatrix} x \\ y \end{bmatrix} = {}_{\mathbf{K}}\mathbf{H}_{\mathbf{K}}\mathbf{x} \tag{5.20}
$$

with ${}_{\mathbf{K}}\mathbf{H} = {}_{\mathbf{K}}\mathbf{H}_{CV}$ in case of the CV model and ${}_{\mathbf{K}}\mathbf{H} = {}_{\mathbf{K}}\mathbf{H}_{CA}$ for the CA model:

$$
{}_{\mathbf{K}}\mathbf{H}_{CV} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}' \tag{5.21}
$$

$$
{}_{\mathbf{K}}\mathbf{H}_{CA} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}' \tag{5.22}
$$

In this context, for the movement estimation, we rely on the hypothesis of the two coordinates de-correlation x and y.

5.3 Kalman filter principle

The system of equations of the Kalman filter (KF) relies on the definition of two models describing the process and the measurement [96]. The process model which describes the system evolution is given by (5.5) while the one that describes the measurement is given in (5.6) and provides the link between the measurement itself, the state and measurement noise.

- the noises ${}_{\mathbf{K}}\mathbf{w}_t$ and ${}_{\mathbf{K}}\mathbf{v}_t$ have typically different meanings: ${}_{\mathbf{K}}\mathbf{w}_t$is considered the noise impacting the process, while ${}_{\mathbf{K}}\mathbf{v}_t$ represents the error on the measurement.

- for simplicity reasons, we assume that the noises ${}_{\mathbf{K}}\mathbf{w}_t$ and ${}_{\mathbf{K}}\mathbf{v}_t$ are mutually independent. The extension to the case where these noises are correlated does not present any major difficulties.

- It is not necessary to make the assumption that the noises and the initial condition are Gaussian [97, 98]. Indeed without this assumption, the Kalman estimates are minimum variance estimates; with this assumption, they are also optimal estimates in the sense of the conditional mean.

- When using this filter, the assumptions are made such that the state and the measurement noises have a priori known Gaussian distributions, independent, and independent of the initial state of the system. The independence of the noises allows simplifying the derivation of the Kalman filter equations.

- The initial state, along with its covariance is assumed to be known.

After an initialization step, the Kalman filter is used by recursively repeating a prediction step, then an update step. This allows us to follow and estimate the evolution of the state of a system.

5.3.1 Initialization

The initial state is assumed to be a Gaussian random variable, independent of the noises, with a known mean and covariance matrix given by [99]:

$$ {}_{\kappa}\hat{\mathbf{x}}_{0|0} = E[{}_{\kappa}\mathbf{x}_0] \tag{5.23} $$

$$ {}_{\kappa}\mathbf{P}_{0|0} = E[({}_{\kappa}\mathbf{x}_0 - {}_{\kappa}\hat{\mathbf{x}}_{0|0})({}_{\kappa}\mathbf{x}_0 - {}_{\kappa}\hat{\mathbf{x}}_{0|0})'] \tag{5.24} $$

5.3.2 Prediction step

Given the estimated state and its covariance matrix at the time instant t, we can make the prediction of the state and its precision at time $t+1$, as well as the prediction of the measurement. We, thus, obtain the predicted state ${}_{\kappa}\hat{\mathbf{x}}_{t+1|t}$ with its associated covariance matrix [99]:

$$ {}_{\kappa}\hat{\mathbf{x}}_{t+1|t} = {}_{\kappa}\mathbf{F}\,{}_{\kappa}\hat{\mathbf{x}}_{t|t} \tag{5.25} $$

$$ {}_{\kappa}\mathbf{P}_{t+1|t} = {}_{\kappa}\mathbf{F}\ {}_{\kappa}\mathbf{P}_{t|t}\ {}_{\kappa}\mathbf{F}' + {}_{\kappa}\mathbf{Q} \tag{5.26} $$

The measurement prediction equation is given by:

$$ {}_{\kappa}\hat{\mathbf{z}}_{t+1|t} = {}_{\kappa}\mathbf{H}\ {}_{\kappa}\hat{\mathbf{x}}_{t+1|t} \tag{5.27} $$

5.3.3 Update step

Once the measure ${}_{\kappa}\mathbf{z}_{t+1}$ is available, we compute the innovation ${}_{\kappa}\hat{\mathbf{s}}_{t+1}$, which represents the prediction error of the observation, whose associated covariance matrix is ${}_{\kappa}\mathbf{S}_{t+1}$. The predicted state can then be corrected by this innovation weighted by the gain of the filter ${}_{\kappa}\mathbf{K}_{t+1}$. We then deduce the estimated state, ${}_{\kappa}\hat{\mathbf{x}}_{t+1|t+1}$, with its associated covariance matrix ${}_{\kappa}\mathbf{P}_{t+1|t+1}$ [93, 94]:

$$ {}_{\kappa}\hat{\mathbf{s}}_{t+1} = {}_{\kappa}\mathbf{z}_{t+1} - {}_{\kappa}\hat{\mathbf{z}}_{t+1|t} \tag{5.28} $$

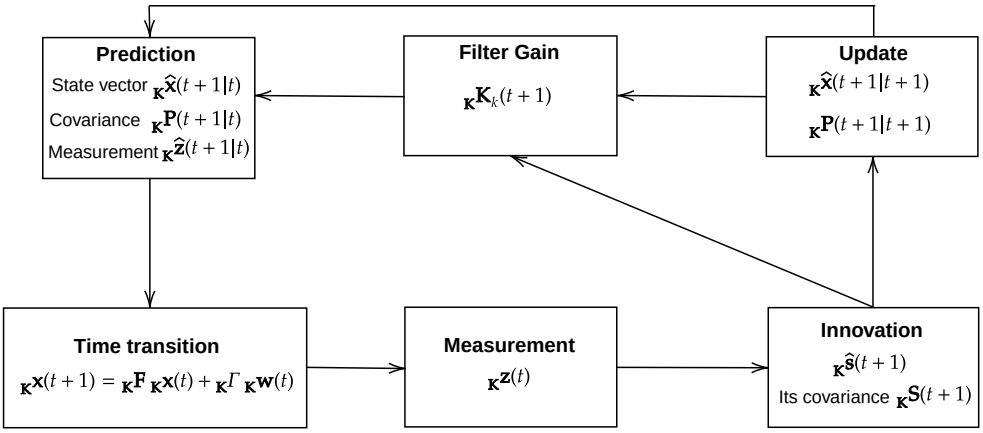

Figure 5.1: A Kalman Filter cycle diagram

$$ {}_{\mathbf{K}}\mathbf{S}_{t+1} = {}_{\mathbf{K}}\mathbf{H}\ {}_{\mathbf{K}}\mathbf{P}_{t+1|t}\ {}_{\mathbf{K}}\mathbf{H}' + {}_{\mathbf{K}}R \tag{5.29} $$

$$ {}_{\mathbf{K}}\mathbf{K}_{t+1} = {}_{\mathbf{K}}\mathbf{P}_{t+1|t}\ {}_{\mathbf{K}}\mathbf{H}'\ {}_{\mathbf{K}}\mathbf{S}_{t+1}^{-1} \tag{5.30} $$

$$ {}_{\mathbf{K}}\hat{\mathbf{x}}_{t+1|t+1} = {}_{\mathbf{K}}\hat{\mathbf{x}}_{t+1|t} + {}_{\mathbf{K}}\mathbf{K}_{t+1\,\mathbf{K}}\hat{\mathbf{s}}_{t+1} \tag{5.31} $$

$$ {}_{\mathbf{K}}\mathbf{P}_{t+1|t+1} = (\mathbf{I} - {}_{\mathbf{K}}\mathbf{K}_{t+1}\ {}_{\mathbf{K}}\mathbf{H})\ {}_{\mathbf{K}}\mathbf{P}_{t+1|t} \tag{5.32} $$

The Kalman gain, ${}_{\mathbf{K}}\mathbf{K}_{t+1}$, takes into account the relative uncertainties of the current estimate and the data. If the uncertainty ${}_{\mathbf{K}}\mathbf{R}$ of the data is negligible compared to the uncertainty of the model ${}_{\mathbf{K}}\mathbf{P}_{t+1|t}$, we should have a strong gain, i.e. the data is reliable. Conversely, if the uncertainty of the data is large compared to that of the estimate, the gain should be very small, i.e. the data is unreliable; it is normal then, that it does not, or only slightly, modulate the uncertainty of the data.

A KF cycle is illustrated in Figure 5.1.

5.3.4 The Kalman Filter advantages

The Kalman filter uses only the first two moments of the state, namely the mean and the covariance, and ignores the moments of higher orders. this approximation offers many practical advantages:

- The mean and covariance of a distribution requires only a small amount of information, while this information is sufficient to cover a wide range of applications. Thus we can say that the KF is an ideal compromise between computational complexity and flexibility of representation.
- The mean and covariance or its square root are linearly translatable quantities. This means that the mean and the covariance can be effectively evaluated when they undergo a linear transformation. We note here that this characteristic does not remain valid for the other moments of a distribution.

5.4 The $\alpha\beta$ algorithm

In the mid-'50s, Radar systems qualified for tracking while scanning, Track While-Scan radar system (TWS), have known the advancement of the structure of target tracking based on the $\alpha\beta$ filter [100]. We note that the Kalman filter appeared only in 1960 [98]. If the system is well-designed to provide the state estimate, then we can get a good prediction of the next scan to guarantee the track correlation [101]. In order to design such a system, the best compromise must be found between intense filtering (steady state) and the ability to follow a possible maneuver (transitional phase). Benedict and Bordner proposed a filter topology based on the determination of a single parameter ensuring this trade-off and established the first relationship between the parameters α and β which was $\beta = \alpha^2/(2-\alpha)$[101]. The best choice of the value α remained unknown until 1984 when Kalata introduced a factor he called the tracking index. This factor is a function of the scan period and the variances of the system and measurement noises [102].

5.4.1 Derivation principle of the $\alpha\beta$ filter

The derivation of the $\alpha\beta$ filter is possible in two ways, the first one proposed by Benedict-Bordner [101] is based on the theory of filtering and system stability applied to time-invariant linear systems and has as objective a good noise reduction and tracking during the maneuver. The equations obtained are based on the calculation of variations. The second technique, which is the most known, takes advantage of the Kalman filter equations. In fact, It has been shown that this filter converges to an equilibrium or static state [93, 94]. The $\alpha\beta$ filter is used in the case of the CV model while in the case of the CA model, it is the $\alpha\beta\gamma$ filter that is used. In this thesis, we will consider only the approach followed by Kalata [93, 102] for the derivation of the $\alpha\beta$ filter, as we only consider a CV model.

For the sequel, we only consider a one-dimensional kinematic model. The state equation and the measurement for the x component are given by:

$$\begin{bmatrix} x_{t+1} \\ V_{x_{t+1}} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_t \\ V_{x_t} \end{bmatrix} + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} {}_{\mathbf{K}}\mathbf{w}_t \tag{5.33}$$

$${}_{\mathbf{K}}\mathbf{z}_t = {}_{\mathbf{K}}\mathbf{H} \begin{bmatrix} x_t \\ V_{x_t} \end{bmatrix} + {}_{\mathbf{K}}\mathbf{v}_t, \quad {}_{\mathbf{K}}\mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix} \tag{5.34}$$

We note the covariance of the system noise ${}_{\mathbf{K}}\mathbf{Q}$ and the variance of the measurement noise ${}_{\mathbf{K}}\mathbf{R}$. The covariance matrices of the estimated state and the predicted state in steady state are respectively noted:

$$\lim_{t\to\infty} {}_{\mathbf{K}}\mathbf{P}_{t|t} = \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} \tag{5.35}$$

$$\lim_{t\to\infty} {}_{\mathbf{K}}\mathbf{P}_{t+1|t} = \begin{bmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{bmatrix} \tag{5.36}$$

The existence, uniqueness, and positivity of the matrix (5.35) is guaranteed as long as the observability and controllability conditions are satisfied [93]. In addition, the gain of the $\alpha\beta$ filter is:

$$ {}_{\mathbf{K}}\mathbf{W} = \lim_{t\to\infty} {}_{\mathbf{K}}\mathbf{W}_t \triangleq \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \triangleq \begin{bmatrix} \alpha \\ \beta/T \end{bmatrix} \tag{5.37} $$

The parameters α and β as defined in this equation have no dimension.

The expression of the innovation covariance 5.29 is:

$$ {}_{\mathbf{K}}\mathbf{S} = {}_{\mathbf{K}}\mathbf{H} \begin{bmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{bmatrix} {}_{\mathbf{K}}\mathbf{H}' + {}_{\mathbf{K}}R = m_{11} + {}_{\mathbf{K}}R \tag{5.38} $$

The gain in 5.30 becomes:

$$ {}_{\mathbf{K}}\mathbf{W} = \begin{bmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{bmatrix} {}_{\mathbf{K}}\mathbf{H}' {}_{\mathbf{K}}\mathbf{S}^{-1} = \begin{bmatrix} \frac{m_{11}}{m_{11}+{}_{\mathbf{K}}R} & \frac{m_{12}}{m_{11}+{}_{\mathbf{K}}R} \end{bmatrix}' \tag{5.39} $$

From 5.37 and 5.39 we pull:

$$ g_1 = \frac{m_{11}}{m_{11} + {}_{\mathbf{K}}R} \tag{5.40} $$

$$ g_2 = \frac{m_{12}}{m_{11} + {}_{\mathbf{K}}R} = g_1 \frac{m_{12}}{m_{11}} \tag{5.41} $$

The covariance matrix becomes:

$$ \begin{aligned} \begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} &= (\mathbf{I}_2 - {}_{\mathbf{K}}\mathbf{W}\ {}_{\mathbf{K}}\mathbf{H}) \begin{bmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{bmatrix} \\ &= \begin{bmatrix} (1-g_1)\, m_{11} & (1-g_1)\, m_{12} \\ (1-g_1)\, m_{12} & m_{22} - g_2 m_{12} \end{bmatrix}, \end{aligned} \tag{5.42} $$

where $\mathbf{I}_n$ is an n dimensional identity matrix. The equation 5.26 can be then written as follows:

$$ {}_{\mathbf{K}}\mathbf{P}_{t|t} = {}_{\mathbf{K}}\mathbf{F}^{-1} \left[{}_{\mathbf{K}}\mathbf{P}_{t+1|t} - {}_{\mathbf{K}}\mathbf{Q} \right] \left({}_{\mathbf{K}}\mathbf{F}^{-1} \right)', \quad {}_{\mathbf{K}}\mathbf{F}^{-1} = \begin{bmatrix} 1 & -T \\ 0 & 1 \end{bmatrix} \tag{5.43} $$

The static solution for the gain and covariance is given by the nonlinear equations from 5.40 to 5.43, using the appropriate expression for the system

noise covariance matrix. This development can also be applied to a model obtained by discretization of the continuous state equation, but in our case, we have limited ourselves to the DWNA model, see section 5.2.3.1. Using 5.15, corresponding to the covariance matrix of the DWNA model, in 5.43, we obtain:

$$\begin{bmatrix} p_{11} & p_{12} \\ p_{12} & p_{22} \end{bmatrix} = \begin{bmatrix} m_{11} - 2Tm_{12} + T^2 m_{22} - \frac{1}{4}T^4\sigma_{\mathbf{w}}^2 & m_{12} - Tm_{22} + \frac{1}{2}T^3\sigma_{\mathbf{w}}^2 \\ m_{12} - Tm_{22} + \frac{1}{2}T^3\sigma_{\mathbf{w}}^2 & m_{22} - T^2\sigma_{\mathbf{w}}^2 \end{bmatrix} \tag{5.44}$$

By identification between 5.42 and 5.44 we obtain

$$g_1 m_{11} = 2Tm_{12} - T^2 m_{22} + \frac{1}{4}T^4\sigma_{\mathbf{w}}^2 \tag{5.45}$$

$$g_1 m_{12} = Tm_{22} - \frac{1}{2}T^3\sigma_{\mathbf{w}}^2 \tag{5.46}$$

$$g_2 m_{12} = T^2\sigma_{\mathbf{w}}^2 \tag{5.47}$$

At this stage, we have the system of equations 5.40, 5.41, 5.4.1, 5.46, and 5.47 with the five unknowns g_1, g_2, m_{11}, m_{12}, and m_{22}. We can thus resolve the system as follows:

From 5.40 and 5.41 we have:

$$m_{11} = \frac{g_1}{1 - g_1}\sigma_{\mathbf{v}}^2 \tag{5.48}$$

$$m_{12} = \frac{g_2}{1 - g_1}\sigma_{\mathbf{v}}^2 \tag{5.49}$$

From 5.46 and 5.47, we obtain:

$$m_{22} = \frac{g_1 m_{12}}{T} + \frac{1}{2}T^2\sigma_{\mathbf{w}}^2 = \left(\frac{g_1}{T} + \frac{g_2}{2}\right) m_{12} \tag{5.50}$$

By using equations from 5.47 to 5.50 in 5.4.1, we have:

$$\frac{g_1^2}{1-g_1}\sigma_{\mathbf{w}}^2 = 2T\frac{g_2}{1-g_1}\sigma_{\mathbf{w}}^2 - T^2\left(\frac{g_1}{T}-\frac{g_2}{2}\right)\frac{g_2}{1-g_1}\sigma_{\mathbf{w}}^2 + \frac{1}{4}T^4\frac{g_2^2}{1-g_1}\sigma_{\mathbf{w}}^2 \quad (5.51)$$

After rearrangement, the above gives:

$$g_1^2 - 2Tg_2 + Tg_1g_2 + \frac{T^2}{4}g_2^2 = 0 \quad (5.52)$$

Using 5.37, 5.52 becomes:

$$\alpha^2 - 2\beta + \alpha\beta + \frac{\beta^2}{4} = 0 \quad (5.53)$$

The latter equation gives the first relation between α and β, namely:

$$\alpha = \sqrt{2\beta} - \frac{\beta}{2} \quad (5.54)$$

The second relation between the two parameters can be deduced directly from 5.47 and 5.49:

$$m_{12} = \frac{T^2\sigma_{\mathbf{w}}^2}{\beta/T} = \frac{\beta/T}{1-\alpha}\sigma_{\mathbf{v}}^2, \quad (5.55)$$

Thus:

$$\frac{\beta^2}{1-\alpha} = \frac{T^4\sigma_{\mathbf{w}}^2}{\sigma_{\mathbf{v}}^2} \quad (5.56)$$

By definition, we note:

$$\lambda \triangleq \frac{T^2\sigma_{\mathbf{w}}}{\sigma_{\mathbf{v}}} \quad (5.57)$$

The quantity λ was introduced by Kalata in [102], it is called the "target maneuvering index" or the "target tracking index". Those appellations are due to the fact that this quantity is proportional to:

- the uncertainty on the motion, denoted by the standard deviation of the system noise,
- the uncertainty on the measurement, denoted by the standard deviation of measurement noise.

The elimination of α from 5.54 and 5.56 gives:

$$\beta = \frac{1}{4}\left(\lambda^2 + 4\lambda - \lambda\sqrt{\lambda^2 + 8\lambda}\right), \tag{5.58}$$

Finally, from 5.54 and 5.58, we obtain:

$$\alpha = -\frac{1}{8}\left(\lambda^2 + 8\lambda - (\lambda + 4)\sqrt{\lambda^2 + 8\lambda}\right) \tag{5.59}$$

Remarques

- When the system noise is relatively important to the measurement noise, we get a high target maneuvering index λ. The latter generates a higher gain in position α and thus, the filter gives a large weight to the most recent measurement and, a small weight to old data, resulting in low noise reduction.
- In the same way, a low λ gives a small α and an important noise reduction. However taking arbitrarily a small value for α does not reduce the noise further, unless this coefficient is determined in an optimal way.
- The coefficients α and β can not be chosen undependably for one another. Figure 5.2 represents the relation that exists between those two coefficients.

5.5 DoA smoothing with Kalman filter

In the following, we consider the case where the targets have time-varying directions. The state vector will therefore be ${}_{\mathbf{K}}\mathbf{x}_t(i) = \left[\theta_k(i); \quad \dot{\theta}_k(i)\right]$ where $\theta_k(i)$ is k^{th} direction of arrival and $\dot{\theta}_k(i)$ is its rate of variation.

The dynamics of this motion, as well as the measurements, are modeled by equations (5.60) and (5.61):

$${}_{\mathbf{K}}\mathbf{x}_k(t+1) = {}_{\mathbf{K}}\mathbf{F}\ {}_{\mathbf{K}}\mathbf{x}_k(t) + {}_{\mathbf{K}}\mathbf{w}_k(t) \tag{5.60}$$

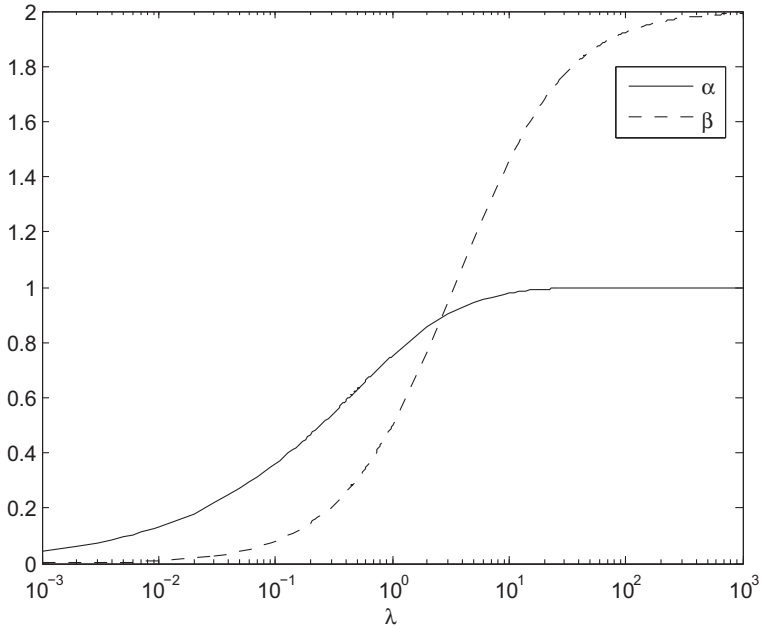

Figure 5.2: $\alpha\beta$ coefficients of the static gain for the DWNA model.

$$\theta_k(t) = {}_{\mathbf{K}}\mathbf{H}\ {}_{\mathbf{K}}\mathbf{x}_k(t) + {}_{\mathbf{K}}\mathbf{v}_k(t) \tag{5.61}$$

Where:

- ${}_{\mathbf{K}}\mathbf{F} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$ is the transition matrix with a time step T, and ${}_{\mathbf{K}}\mathbf{H} = \begin{bmatrix} 1 & 0 \end{bmatrix}$ is the measurement matrix.
- ${}_{\mathbf{K}}\mathbf{w}_k(t)$ is the noise process supposed to be Gaussian with covariance

$${}_{\mathbf{K}}\mathbf{Q}_k(t) = \begin{bmatrix} \frac{T^4}{4} & \frac{T^3}{2} \\ \frac{T^3}{2} & T^2 \end{bmatrix} \sigma^2_{\mathbf{w}_k}$$

- ${}_{\mathbf{K}}\mathbf{v}_k(t)$ is the measurement noise with variance

$${}_{\mathbf{K}}R_k(t) = \sigma^2_{\mathbf{v}_k}$$

The Kalman filter proceeds in two steps: a prediction before the mea-

surement and an update after it:

5.5.1 Prediction step:

Following the algorithm in section 5.3 and from the data available at time t, we can make a prediction of the state vectors of the next time step, of their covariance matrices ${}_{\kappa}\mathbf{P}_k$, as well as of the following measurements:

$$ {}_{\kappa}\hat{\mathbf{x}}_k(t+1|t) = {}_{\kappa}\mathbf{F}\ {}_{\kappa}\mathbf{x}_k(t|t) \tag{5.62} $$

$$ {}_{\kappa}\mathbf{P}_k(t+1|t) = {}_{\kappa}\mathbf{F}\ {}_{\kappa}\mathbf{P}_k(t|t)\ {}_{\kappa}\mathbf{F}^T + {}_{\kappa}\mathbf{Q}_k(t) \tag{5.63} $$

$$ \hat{\theta}_k(t+1|t) = {}_{\kappa}\mathbf{H}\ {}_{\kappa}\hat{\mathbf{x}}_k(t+1|t) \tag{5.64} $$

- At this point, the measurements $\theta_k(t)$ are taken to be the directions of arrival estimated by the HTFAPI algorithm followed by a TLS-ESPRIT.

5.5.2 Update step:

After obtaining the new measurements, the innovation, and its covariance are computed as follows:

$$ {}_{\kappa}s(t) = \theta_k(t) - \hat{\theta}_k(t+1|t) \tag{5.65} $$

$$ {}_{\kappa}S_k = {}_{\kappa}\mathbf{H}\ {}_{\kappa}\mathbf{P}_k(t+1|t)\ {}_{\kappa}\mathbf{H}^H + {}_{\kappa}R_k(t) \tag{5.66} $$

This innovation is then used to improve the DoA estimation and their corresponding covariance updates:

$$ {}_{\kappa}\hat{\mathbf{x}}_k(t+1|t+1) = {}_{\kappa}\hat{\mathbf{x}}_k(t+1|t) + {}_{\kappa}\mathbf{K}_k(t+1)\ {}_{\kappa}s(t) \tag{5.67} $$

$$ {}_{\kappa}\mathbf{P}_k(t+1|t+1) = [\mathbf{I}_2 - {}_{\kappa}\mathbf{K}_k(t+1)\ {}_{\kappa}\mathbf{H}]\ {}_{\kappa}\mathbf{P}_k(t+1|t) \tag{5.68} $$

Where ${}_{\kappa}\mathbf{K}_k(t+1)$ is the Kalman filter gain given by:

$$ {}_{\kappa}\mathbf{K}_k(t+1) = {}_{\kappa}\mathbf{P}_k(t+1|t)\ {}_{\kappa}\mathbf{H}^H {}_{\kappa}S_k^{-1} \tag{5.69} $$

To initialize the Kalman filter we use the two points technique so that:

$$ {}_{\kappa}\mathbf{x}_k(2) = \begin{bmatrix} \theta_k(2); & \dfrac{\theta_k(2) - \theta_k(1)}{T} \end{bmatrix} \tag{5.70} $$

Therefore, the initial covariance depends on the measurement variance as follows

$$ {}_{\kappa}\mathbf{P}_k(2) = \begin{bmatrix} 1 & \frac{1}{T} \\ \frac{1}{T} & \frac{1}{T^2} \end{bmatrix} {}_{\kappa}R_k(2) \tag{5.71} $$

Now, the choice of the Kalman filter parameters depends on the chosen model and the measurement accuracy.

To detect the presence of impulsive noise and thus to better tune the filter's parameters, we propose to use the weight $\omega(t)$ calculated in the HT-FAPI algorithm. Indeed, since it takes two distinct values (0 or 1) according to whether there is a noise impulsion, we can determine the Kalman filter parameters according to the context as explained in the next section.

5.6 Simulations and results

In this section, we investigate the performance of our proposed algorithm according to the scenario below.

For the impulsive noise model, we consider here a burst noise generated according to equation (2.4):

$$ \mathbf{n}(t) = \mathbf{n}_G(t) + \sum_{j=1}^{N_I} u(\frac{i - i_j}{b_j})\mathbf{n}_I^j(t) \tag{5.72} $$

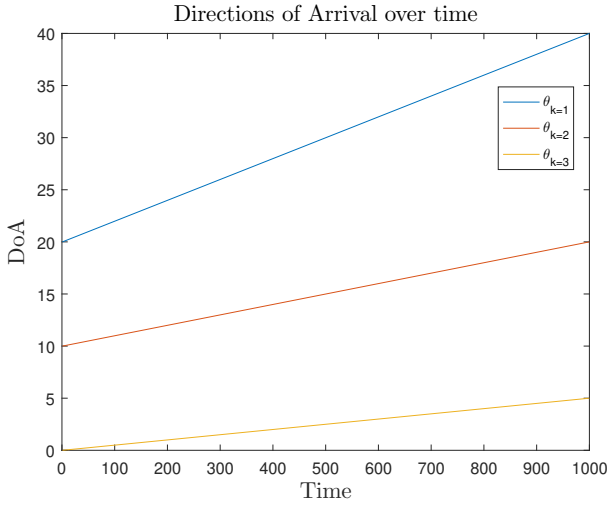

Figure 5.3: Directions of arrival changes over the studied time sequence

where $\mathbf{n}_G(t)$ is a white Gaussian and centered noise vector of variance σ_G^2 and $\mathbf{n}_I$ is white centered Gaussian of variance $\sigma_I^2 >> \sigma_G^2$ weighted by $u(\cdot)$ a rectangular function, which is used to describe the short duration appearance of the burst noise. N_I refers to the number of impulsive events, i_j is the center of the j-th impulsive event, and b_j is its duration.

We aim to estimate and track the directions of arrival of $p = 3$ sources impinging on an antenna array of $n = 9$ sensors. We run ten Monte Carlo simulations during $N = 1000$ instant time each.

The first source changes direction uniformly from 20° to 40° during the observation sequence, the second begins at 10° and ends the sequence at 20° while the third begins at 0° and ends at 5°, as displayed in Fig.5.3:

The noise model followed is the one described in (2.4), where the signal is disturbed by an additive noise letting its signal-to-noise ratio (SNR) to be at $SNR = 0dB$ during the entire observation sequence except for two short periods of time: $P1 = [201, 250]$ and $P2 = [601, 700]$, where an impulsive noise appears inducing the signal SNR to be $SNR = -20dB$. A second simulation is run to show the behavior of the filters in the presence of a more important impulsive noise. In this latter, the signal SNR during $P1$ and $P2$ is set to be $SNR = -40dB$.

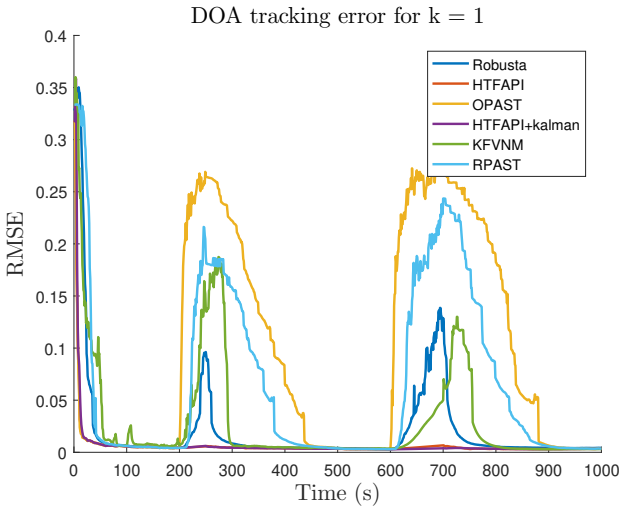

Figure 5.4: RMSE of the $k = 1$ DoA tracking in signal with SNR=-20dB in the presence of impulsive noise

The Kalman filter parameters are adjusted as follows: the noise process is set to be $\sigma^2_{\psi_k} = 0.00001^2$ during the entire experiment while the measurement noise is chosen dynamically as:

$$\begin{cases} \sigma^2_{\upsilon_k} = 0.002 & \text{if } \omega(t) = 1 \\ \sigma^2_{\upsilon_k} = 120 * 0.002 & \text{if } \omega(t) = 0 \end{cases}$$

The performance of this algorithm is evaluated according to the RMSE of the angular velocities $\omega_k = \pi \sin \theta_k$. The results shown in Fig.5.4, 5.5 and 5.6 are compared to those of the OPAST algorithm as a benchmark to showcase the need for robust algorithms. Also, they are compared to those of the ROBUSTA proposed in [6] and the RPAST [33] to illustrate the motivation behind the choice of the subspace tracking algorithm. Lastly, our algorithm is compared to the KFVNM [38] to showcase the performance difference as compared to a similar algorithm. It is clearly noticeable that our algorithm outperforms the other ones in the presence of impulsive noise. Moreover, this performance gain is more noteworthy when the impulsive noise is more important as shown in Fig.5.7, 5.8 and 5.9, when the signal SNR during the impulsion is about $SNR = -40dB$.

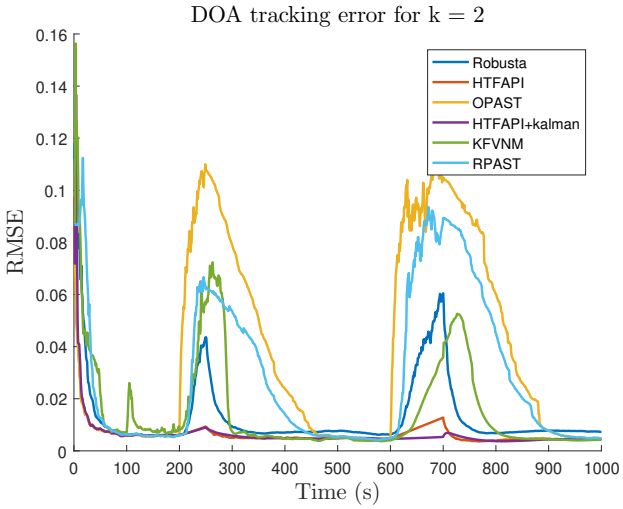

Figure 5.5: RMSE of the $k = 2$ DoA tracking in signal with SNR=-20dB in the presence of impulsive noise

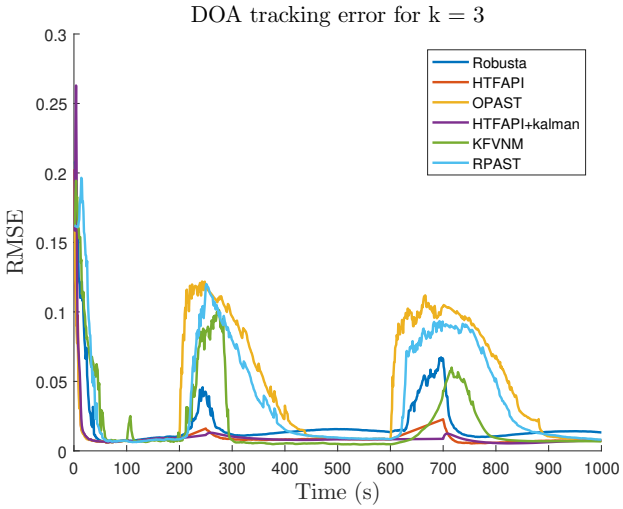

Figure 5.6: RMSE of the $k = 3$ DoA tracking in signal with SNR=-20dB in the presence of impulsive noise

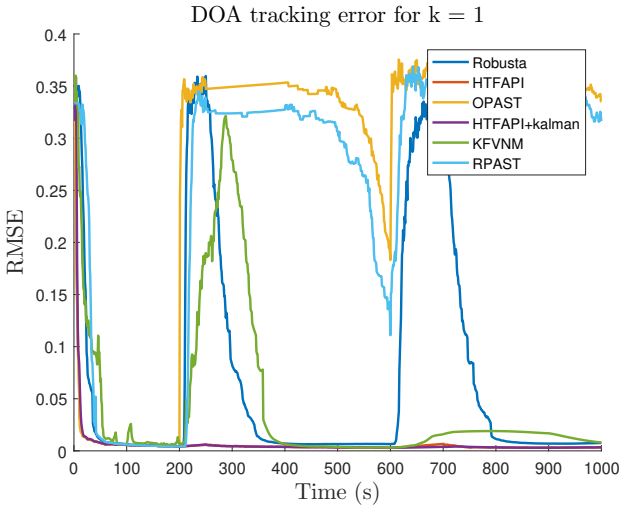

Figure 5.7: RMSE of the $k = 1$ DoA tracking in signal with SNR=-40dB in the presence of impulsive noise

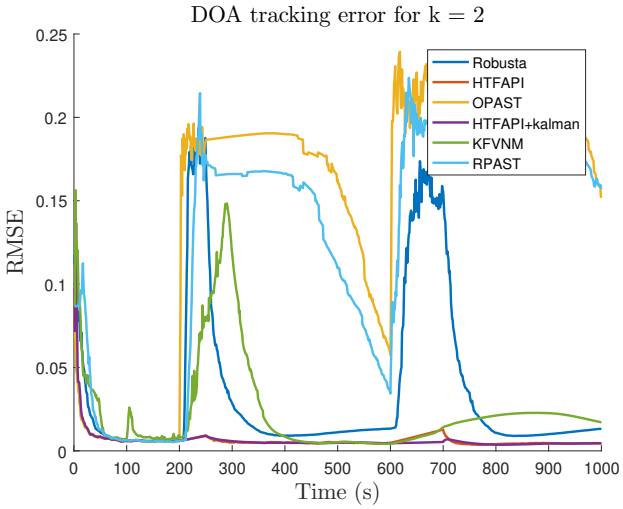

Figure 5.8: RMSE of the $k = 2$ DoA tracking in signal with SNR=-40dB in the presence of impulsive noise

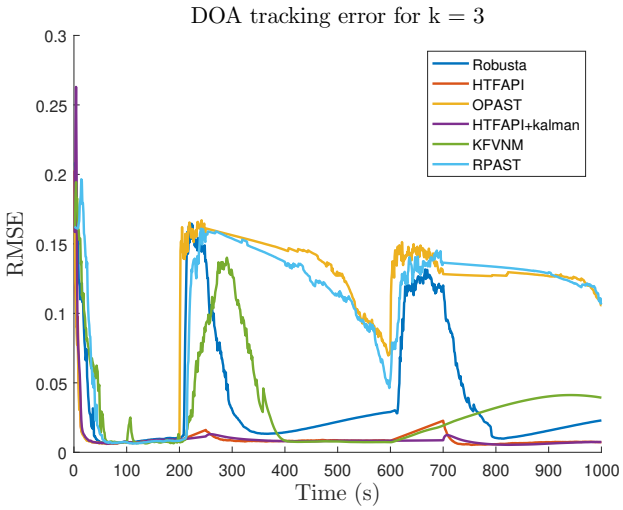

Figure 5.9: RMSE of the $k = 3$ DoA tracking in signal with SNR=-40dB in the presence of impulsive noise

5.7 Conclusion

We have proposed in this chapter a new approach to enhance the directions of arrival estimation and tracking in an impulsive noise environment using a robust subspace tracking method followed by a conventional Kalman filter. Indeed, the simulated experiments have shown that this latter combination improves considerably the performances of the tracking in the presence of strong impulsive noise. For future works, to further reduce the cost, we also consider replacing the Kalman filter with its low-cost alternative, the $\alpha\beta$ filter.

Chapter 6

General Conclusion

In this thesis, we have considered the subspace tracking technique to deal with signal processing in the context of data stream in a non-Gaussian environment while always keeping the lowest computation cost possible.

Indeed, After the general introduction, we have seen in chapter two, different studied systems along with the objectives we were trying to achieve. Then we overlooked the related works that dealt with similar contexts. As we have stated there, although a vast plethora of works exists, the ones that deal with impulsive noise stay very slim. Reason for which, we have suggested new algorithms presented in chapter three.

In chapter three, we proposed 2 novel, robust, and fast algorithms for subspace tracking MFAPI and HTFAPI. the idea was to use a weighted least square criterion combined with an accurate projection approximation for low-cost estimation. The weight itself was then calculated by two means; the first one used the Mahalanobis distance to decide whether the coming data was valid or corrupted, and the second used robust statistical estimation to do the same. the simulation results have shown that the proposed algorithms outperform all those of the literature regarding slow time-variant systems. This led us to think about implementing them with sliding window criterion in order to include systems with sudden changes in behavior.

In Chapter 4, we have considered an application of subspace tracking; the blind source separation. Indeed, we have extended the results of robust subspace tracking to a principal components analysis point of view, all while always keeping a linear complexity. The outcome was then used in data whitening and later in actual source estimation. Simulation results have proven the accuracy of the proposed methods as compared with similar second-order-based ones.

In the last chapter, we handled another application of subspace tracking. Indeed, the latter is very well-known regarding parameters' estimations. Thus, we used it in the field of source localization and state representation. We have then proposed a new algorithm that relies on robust subspace tracking to suppress impulsiveness in the received data, retrieve directions of arrival, then use adaptive filtering to smooth and enhance the results. The results obtained here were promising, and again, it opened for us the possibility to consider maneuvering targets, as it has been wildly discussed in the tracking area.

Bibliography

[1] Zineb Bekhtaoui, Karim Abed-Meraim, and Abdelkrim Meche. Robust adaptive algorithms for fast principal component analysis. *Digital Signal Processing*, 127:103561, 2022.

[2] Zineb Bekhtaoui, Karim Abed-Meraim, Abdelkrim Meche, and Messaoud Thameri. A new robust adaptive algorithm for second order blind source separation. *ENP Engineering Science Journal*, 2(1):21–28, 2022.

[3] Z Bekhtaoui, A Meche, M Dahmani, and K ABED Meraim. Maneuvering target tracking using q-learning based kalman filter. In *2017 5th International Conference on Electrical Engineering-Boumerdes (ICEE-B)*, pages 1–5. IEEE, 2017.

[4] Z Bekhtaoui, A Meche, Karim Abed-Meraim, and M Dahmani. A new robust subspace tracking algorithm and its application to direction of arrival estimation and tracking. In *DAT*, 2020.

[5] Zineb Bekhtaoui, Abdelkrim Meche, Karim Abed-Meraim, and Mohammed Dahmani. Direction of arrival tracking using adaptive robust subspace decomposition and kalman filter. In *2023 20th International Multi-Conference on Systems, Signals and Devices (SSD)*. IEEE, 2023.

[6] Nguyen Linh-Trung, Viet-Dung Nguyen, Messaoud Thameri, Truong Minh-Chinh, and Karim Abed-Meraim. Low-complexity adaptive algorithms for robust subspace tracking. *IEEE Journal of Selected Topics in Signal Processing*, 12(6):1197–1212, 2018.

[7] Panayiotis G. Georgiou, Panagiotis Tsakalides, and Chris Kyriakakis. Alpha-stable modeling of noise and robust time-delay estimation in the presence of impulsive noise. *IEEE transactions on multimedia*, 1(3):291–301, 1999.

[8] Pierre Comon and Gene H Golub. Tracking a few extreme singular values and vectors in signal processing. *Proceedings of the IEEE*, 78(8):1327–1343, 1990.

[9] Jean-Pierre Delmas. Subspace tracking for signal processing, 2010.

[10] Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of mathematical biology*, 15(3):267–273, 1982.

[11] Erkki Oja. Principal components, minor components, and linear neural networks. *Neural networks*, 5(6):927–935, 1992.

[12] Karim Abed-Meraim, Samir Attallah, Ammar Chkeif, and Yingbo Hua. Orthogonal oja algorithm. *IEEE Signal processing letters*, 7(5):116–119, 2000.

[13] Samir Attallah and Karim Abed-Meraim. Fast algorithms for subspace tracking. *IEEE Signal Processing Letters*, 8(7):203–206, 2001.

[14] Steve Bartelmaos and Karim Abed-Meraim. Principal and minor subspace tracking: Algorithms & stability analysis. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, volume 3, pages III–III. IEEE, 2006.

[15] Bin Yang. Projection approximation subspace tracking. *IEEE Transactions on Signal processing*, 43(1):95–107, 1995.

[16] Karim Abed-Meraim, Ammar Chkeif, and Yingbo Hua. Fast orthonormal past algorithm. *IEEE Signal processing letters*, 7(3):60–62, 2000.

[17] Charles F Van Loan and G Golub. Matrix computations (johns hopkins studies in mathematical sciences). *Matrix Computations*, 1996.

[18] BN Parlett. The symmetric eigenvalue problem/prentice-hall series in computational mathematics, 1980.

[19] Hideaki Sakai and Kouhei Shimizu. A new adaptive algorithm for minor component analysis. *Signal Processing*, 71(3):301–308, 1998.

[20] J-F Yang and Mostafa Kaveh. Adaptive eigensubspace algorithms for direction or frequency estimation and tracking. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 36(2):241–251, 1988.

[21] S Attallah and K Abed-Meraim. Low-cost adaptive algorithm for noise subspace estimation. *Electronics Letters*, 38(12):1, 2002.

[22] XG Doukopoulos. Power techniques for blind channel estimation in wireless communication systems. *Unpublished doctoral dissertation, IRISA-INRIA, University of Rennes, France*, 2004.

[23] Xenofon G Doukopoulos and George V Moustakides. The fast data projection method for stable subspace tracking. In *2005 13th European Signal Processing Conference*, pages 1–4. IEEE, 2005.

[24] Xenofon G Doukopoulos and George V Moustakides. Fast and stable subspace tracking. *IEEE Transactions on Signal Processing*, 56(4):1452–1465, 2008.

[25] Yingbo Hua, Yong Xiang, Tianping Chen, Karim Abed-Meraim, and Yongfeng Miao. A new look at the power method for fast subspace tracking. *Digital Signal Processing*, 9(4):297–314, 1999.

[26] R. Badeau, B. David, and G. Richard. Fast approximated power iteration subspace tracking. *IEEE Transactions on Signal Processing*, 53(8):2931–2941, August 2005.

[27] Sophie Beau and Sylvie Marcos. Range dependent clutter rejection using range-recursive space-time adaptive processing (stap) algorithms. *Signal Processing*, 90(1):57–68, 2010.

[28] Kenneth L. Blackard, Theodore S. Rappaport, and Charles W. Bostian. Measurements and models of radio frequency impulsive noise for indoor wireless communications. *IEEE Journal on selected areas in communications*, 11(7):991–1001, 1993.

[29] William J Ebel and William H Tranter. The performance of reed-solomon codes on a bursty-noise channel. *IEEE Transactions on Communications*, 43(2/3/4):298–306, 1995.

[30] Kung Yao. A representation theorem and its applications to spherically-invariant random processes. *IEEE Transactions on Information Theory*, 19(5):600–608, 1973.

[31] Esa Ollila, David E Tyler, Visa Koivunen, and H Vincent Poor. Complex elliptically symmetric distributions: Survey, new results and applications. *IEEE Transactions on signal processing*, 60(11):5597–5625, 2012.

[32] Chrysostomos L Nikias and Min Shao. *Signal processing with alpha-stable distributions and applications*. Wiley-Interscience, 1995.

[33] Shing-Chow Chan, Yu Wen, and Ka-Leung Ho. A robust PAST algorithm for subspace tracking in impulsive noise. *IEEE transactions on signal processing*, 54(1):105–116, 2005.

[34] Jin-feng Zhang and Tian-shuang Qiu. A robust correntropy based subspace tracking algorithm in impulsive noise environments. *Digital Signal Processing*, 62:168–175, 2017.

[35] Jin-feng ZHANG, Tian-shuang QIU, and Sen LI. A robust past algorithm based on maximum correntropy criterion for impulsive noise environments. *Acta Electonica Sinica*, 43(3):483, 2015.

[36] Jinfeng Zhang and Tianshuang Qiu. A novel tracking method for fast varying subspaces in impulsive noise environments. In *2016 10th International Conference on Signal Processing and Communication Systems (ICSPCS)*, pages 1–7. IEEE, 2016.

[37] Shengyang Luan, Tianshuang Qiu, Ling Yu, Jinfeng Zhang, Aimin Song, and Yonhjie Zhu. Bnc-based projection approximation subspace tracking under impulsive noise. *IET Radar, Sonar & Navigation*, 11(7):1055–1061, 2017.

[38] SC Chan, ZG Zhang, and Yi Zhou. A new adaptive kalman filter-based subspace tracking algorithm and its application to doa estimation. In *2006 IEEE International Symposium on Circuits and Systems*, pages 4–pp. IEEE, 2006.

[39] Bin Liao, ZG Zhang, and Shing-Chow Chan. A new robust kalman filter-based subspace tracking algorithm in an impulsive noise environment. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 57(9):740–744, 2010.

[40] Trung Thanh Le, Viet Dung Nguyen, Linh Trung Nguyen, and Karim Abed-Meraim. Robust subspace tracking algorithms in signal processing: A brief survey. *REV Journal on Electronics and Communications*, 11(1-2):16–25, 2021.

[41] R. Badeau, G. Richard, B. David, and K. Abed-Meraim. Approximated power iterations for fast subspace tracking. In *Seventh International Symposium on Signal Processing and Its Applications, 2003. Proceedings.*, pages 583–586 vol.2, Paris, France, 2003. IEEE.

[42] David E Tyler. A distribution-free M-estimator of multivariate scatter. *The Annals of Statistics*, pages 234–251, 1987.

[43] Yingbo Hua, Yong Xiang, Tianping Chen, Karim Abed-Meraim, and Yongfeng Miao. A new look at the power method for fast subspace tracking. *Digital Signal Processing*, 9(4):297–314, 1999.

[44] N. Balakrishnan, C. R. Rao. Order Statistics: Theory and Methods. *Handbook of Statistics, Elsevier.*, 1998.

[45] Giulio Barbato, EM Barini, Gianfranco Genta, and Raffaello Levi. Features and performance of some outlier detection methods. *Journal of Applied Statistics*, 38(10):2133–2149, 2011.

[46] Paweł D Domański. Study on statistical outlier detection and labelling. *International Journal of Automation and Computing*, 17(6):788–811, 2020.

[47] A. Arasu and G. S. Manku. Approximate counts and quantiles over sliding windows. *ACM Symposium on Principles of Database Systems (PODS)*, pages 286–296, 2004.

[48] W. Jiao, S. Yang and Y. Chang. DOA Estimation of Multiple Convolutively Mixed Sources Based on Principle Component Analysis. *Lecture Notes In Computer Science*, pages 340–348, 2009.

[49] S Bartelmaos, K Abed-Meraim, and S Attallah. Mobile localization using subspace tracking. In *2005 Asia-Pacific Conference on Communications*, pages 1009–1013. IEEE, 2005.

[50] H. Zhou and P. Xiao. Robust multiuser detection using Kalman filter and windowed projection approximation subspace tracking algorithm. *IET*, pages 495–504, 2011.

[51] S. Dib, M. Barkat and M. Grimes. PAST and OPAST algorithms for STAP in monostatic airborne radar. *INISTA*, pages 177 – 181, 2011.

[52] V. Solo. High Dimensional Point Process System Identification: PCA and Dynamic Index Models. *45th IEEE Conference on Decision and Control*, pages 829 – 833, 2006.

[53] Tu Chao-Cheng and B. Champagne. Subspace tracking of fast time-varying channels in precoded MIMO-OFDM systems. *ICASSP*, pages 2565 – 2568, 2009.

[54] N. Lassami, A. Aïssa-El-Bey, and K. Abed-Meraim. Blind joint MIMO channel and data estimation based on regularized ML. *Digital Signal Processing, Elsevier*, vol: 117(October, 2021), 2021.

[55] Jong-Hoon Ahn. Fast adaptive robust subspace tracking for online background subtraction. In *2014 22nd International Conference on Pattern Recognition*, pages 2555–2559. IEEE, 2014.

[56] Messaoud Thameri, Karim Abed-Meraim, and Adel Belouchrani. Low complexity adaptive algorithms for principal and minor component analysis. *Digital Signal Processing*, 23(1):19–29, 2013.

[57] Marc Moonen, Paul Van Dooren, and Joos Vandewalle. Updating singular value decompositions. a parallel implementation. In *Advanced Algorithms and Architectures for Signal Processing IV*, volume 1152, pages 80–91. International Society for Optics and Photonics, 1989.

[58] Nguyen Viet Dung, Nguyen Linh Trung, Karim Abed-Meraim, et al. Robust subspace tracking with missing data and outliers: Novel algorithm with convergence guarantee. *IEEE Transactions on Signal Processing*, 69:2070–2085, 2021.

[59] Yuejie Chi, Yonina C Eldar, and Robert Calderbank. Petrels: Parallel subspace estimation and tracking by recursive least squares from partial observations. *IEEE Transactions on Signal Processing*, 61(23):5947–5959, 2013.

[60] Simon Haykin. Unsupervised adaptive filtering, volume 1. *Blind source separation*, 1, 2000.

[61] Pierre Comon and Christian Jutten. *Handbook of Blind Source Separation: Independent component analysis and applications*. Academic press, 2010.

[62] Adel Belouchrani, Karim Abed-Meraim, J-F Cardoso, and Eric Moulines. A blind source separation technique using second-order statistics. *IEEE Transactions on signal processing*, 45(2):434–444, 1997.

[63] J-F Cardoso and Beate H Laheld. Equivariant adaptive source separation. *IEEE Transactions on signal processing*, 44(12):3017–3030, 1996.

[64] Li Jiawen and Li Congxin. A two-step adaptive blind source separation for machine sound. In *2006 6th World Congress on Intelligent Control and Automation*, volume 2, pages 5424–5427. IEEE, 2006.

[65] Christoph Mühlmann. *Advances in blind source separation for spatial data*. PhD thesis, Wien, 2021.

[66] Yan Pan, Markus Matilainen, Sara Taskinen, and Klaus Nordhausen. A review of second-order blind identification methods. *Wiley interdisciplinary reviews: computational statistics*, page e1550, 2021.

[67] Hussein A Al-Barhan, Sinan M Elyass, Thamir R Saeed, Ghufran M Hatem, and Hadi T Ziboon. Blind source separation based on random mixing coefficient. In *2020 3rd International Conference on Engineering Technology and its Applications (IICETA)*, pages 164–168. IEEE, 2020.

[68] Messaoud Thameri, Karim Abed-Meraim, and Adel Belouchrani. New algorithms for adaptive bss. In *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA)*, pages 590–594. IEEE, 2012.

[69] Xiaofu Xie, Qingyan Shi, and Renbiao Wu. A new variable step-size equivariant adaptive source separation algorithm. In *2007 Asia-Pacific Conference on Communications*, pages 479–482. IEEE, 2007.

[70] Xiao-Long Zhu and Xian-Da Zhang. Adaptive rls algorithm for blind source separation using a natural gradient. *IEEE Signal Processing Letters*, 9(12):432–435, 2002.

[71] Yong Xiang, Karim Abed-Meraim, and Yingbo Hua. Adaptive blind source separation by second order statistics and natural gradient. In *1999 IEEE International Conference on Acoustics, Speech, and Signal*

Processing. Proceedings. ICASSP99 (Cat. No. 99CH36258), volume 5, pages 2917–2920. IEEE, 1999.

[72] Andreas Ziehe, Pavel Laskov, Guido Nolte, and Klaus-Robert MÃžller. A fast algorithm for joint diagonalization with non-orthogonal transformations and its application to blind source separation. *Journal of Machine Learning Research*, 5(Jul):777–800, 2004.

[73] Laurent Clavier, Gareth W Peters, François Septier, and Ido Nevat. Impulsive noise modeling and robust receiver design. *EURASIP Journal on Wireless Communications and Networking*, 2021(1):1–30, 2021.

[74] Ali Awad. Denoising images corrupted with impulse, gaussian, or a mixture of impulse and gaussian noise. *Engineering Science and Technology, an International Journal*, 22(3):746–753, 2019.

[75] Hongyuan Gao, Menghan Chen, Yanan Du, and Andreas Jakobsson. Monostatic mimo radar direction finding in impulse noise. *Digital Signal Processing*, 117:103198, 2021.

[76] Niko Lietzén, Klaus Nordhausen, and Pauliina Ilmonen. Complex valued robust multidimensional sobi. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 131–140. Springer, 2017.

[77] Pauliina Ilmonen, Klaus Nordhausen, Hannu Oja, and Fabian Theis. An affine equivariant robust second-order bss method. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 328–335. Springer, 2015.

[78] Jari Miettinen, Markus Matilainen, Klaus Nordhausen, and Sara Taskinen. Extracting conditionally heteroskedastic components using independent component analysis. *Journal of time series analysis*, 41(2):293–311, 2020.

[79] Fabian J Theis, Nikola S Müller, Claudia Plant, and Christian Böhm. Robust second-order source separation identifies experimen-

tal responses in biomedical imaging. In *International conference on latent variable analysis and signal separation*, pages 466–473. Springer, 2010.

[80] Adel Belouchrani and Andrzej Cichocki. Robust whitening procedure in blind source separation context. *Electronics letters*, 36(24):2050–2051, 2000.

[81] Rand R Wilcox. *Introduction to robust estimation and hypothesis testing*. Academic press, 2011.

[82] Messaoud Thameri, Abla Kammoun, Karim Abed-Meraim, and Adel Belouchrani. Fast principal component analysis and data whitening algorithms. In *Int. Workshop on Systems, Sig. Proc. and their Applications, WOSSPA*, pages 139–142, Tipaza, Algeria, May 2011.

[83] Ian T Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

[84] Chuang Wang, Yonina C Eldar, and Yue M Lu. Subspace estimation from incomplete observations: A high-dimensional analysis. *IEEE Journal of Selected Topics in Signal Processing*, 12(6):1240–1252, 2018.

[85] Steve Bartelmaos and Karim Abed-Meraim. Fast principal component extraction using givens rotations. *IEEE Signal Processing Letters*, 15:369–372, 2008.

[86] Jun He, Laura Balzano, and Arthur Szlam. Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1568–1575. IEEE, 2012.

[87] Andy Vesa. Direction of arrival estimation using MUSIC and root-MUSIC algorithm. In *18th Telecommunications Forum*, pages 582–585, 2010.

[88] Cheng Qian, Lei Huang, and H.C. So. Computationally efficient ESPRIT algorithm for direction-of-arrival estimation based on Nyström method. *Signal Processing*, 94:74–80, January 2014.

[89] Gangqiang Guan, Deyong Xian, Zhou Yi, and Weihua Xie. A Stable DOA Tracking Method Using FDPM and Kalman Filter. In Jiadong Sun, Changfeng Yang, and Shuren Guo, editors, *China Satellite Navigation Conference (CSNC) 2018 Proceedings*, volume 499, pages 445–453. Springer Singapore, Singapore, 2018.

[90] J. Sanchez-Araujo and S. Marcos. An efficient PASTd-algorithm implementation for multiple direction of arrival tracking. *IEEE Transactions on Signal Processing*, 47(8):2321–2324, August 1999.

[91] Amir Valizadeh, Mahmood Karimi, and Omid Oliaeid. An Efficient Algorithm for Mutiple Direction of Arrival Tracking Based on the Constrained Projection Approximation Approach and Kalman Filter. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, pages II–1089–II–1092, Honolulu, HI, USA, 2007. IEEE.

[92] Messaoud Thameri, Karim Abed-Meraim, and Adel Belouchrani. Robust subspace tracking in impulsive noise environment. In *International Conference on Automatic control, Telecommunications and Signals (ICATS15)*, 2015.

[93] Yaakov Bar-Shalom, X Rong Li, and Thiagalingam Kirubarajan. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.

[94] Dan Simon. *Optimal state estimation: Kalman, H infinity, and nonlinear approaches*. John Wiley & Sons, 2006.

[95] X Rong Li and Vesselin P Jilkov. Survey of maneuvering target tracking. part i. dynamic models. *IEEE Transactions on aerospace and electronic systems*, 39(4):1333–1364, 2003.

[96] Meche, Abdelkrim. *Le filtre de Kalman et ses applications à la poursuite et à la détection de voies.* PhD thesis, Laboratoire Signaux et Images, département d'électronique, USTO-MB, Oran, Algérie, July 2013.

[97] Simon J Julier and Jeffrey K Uhlmann. Unscented filtering and nonlinear estimation. *Proceedings of the IEEE*, 92(3):401–422, 2004.

[98] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82-D:35–45, 1960.

[99] Yaakov Bar-Shalom and Xiao-Rong Li. Estimation and tracking- principles, techniques, and software. *Norwood, MA: Artech House, Inc, 1993.*, 1993.

[100] Paul R Kalata. α-β target tracking systems: A survey. In *1992 American Control Conference*, pages 832–836. IEEE, 1992.

[101] T Benedict and G Bordner. Synthesis of an optimal set of radar track-while-scan smoothing equations. *IRE Transactions on Automatic Control*, 7(4):27–32, 1962.

[102] Paul R Kalata. The tracking index: A generalized parameter for α-β and α-β-γ target trackers. *IEEE Transactions on Aerospace and Electronic Systems*, AES-20(2):174–182, 1984.