UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Escola Tècnica Superior d'Enginyeria
de Telecomunicació de Barcelona

telecom
BCN

Contributions to anomaly detection and correction in co-evolving data streams via subspace learning

Carlos Alejandro López Molina

A Master's degree Thesis
Submitted to the Faculty of the
Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona
Universitat Politècnica de Catalunya

In partial fulfillment
of the requirements for the Master's degree

Master's degree in Telecomunications Engineering

Advisor
Prof. Jaume Riba Sagarra

Barcelona, July 2020

Acknowledgements

I would like to express my sincere gratitude to several people, without whom this project would not have been possible. The first one of them is Jaume Riba, my project supervisor, who always encouraged me to keep learning and keep pursuing bigger objectives while still being cautious, so this project could follow the right path, and for having the patience to help me. I would like to make a special mention on Marc Vila, with who I worked together to develop the Affine Projection for Subspace Tracking algorithm in this work. Of course, I am grateful for the support of my friends, girlfriend and family during this 2 years in the Master's Degree.

Abstract

During decades, estimation and detection tasks in many Signal Processing and Communications applications have been significantly improved by using subspace and component-based techniques. More recently, subspace methods have been adopted in many hot topics such as Machine Learning, Data Analytics or smart MIMO communications, in order to have a geometric interpretation of the problem. In that way, the Subspace-based algorithms often arise new approaches for already-explored problems, while offering the valuable advantage of giving interpretability to the procedures and solutions.

On the other hand, in those recent hot topics, one may also find applications where the detection of unwanted or out-of-the-model artifacts and outliers is crucial. To this extend, we were previously working in the domain of GNSS PPP, detecting phase ambiguities, where we found motivation into the development of novel solutions for this application.

After considering the applications and advantages of subspace-based approaches, this work will be focused on the exploration and extension of the ideas of subspace learning in the context of anomaly detection, where we show promising and original results in the areas of anomaly detection and subspace-based anomaly detection, in the form of two new algorithms: the Dual Ascent for Sparse Anomaly Detection and the Subspace-based Dual Ascent for Anomaly Detection and Tracking.

Nomenclature

$\mathbf{x}$	Column vector
$\mathbf{A}$	Matrix
$()^T$	Transpose operator
$\mathbf{1}_N$	$N \times 1$ all ones vector
$\mathbf{0}_N$	$N \times 1$ all zeroes vector
APST	Affine Projection Subspace Tracking
DPM	Data Projection Method
GNSS	Global Navigation Satellite System
i.i.d	Independent and identically distributed
ISTA	Iterative Shrinkage-Thresholding Algorithm
KKT	Karush-Kuhn-Tucker
LS	Least Squares
MAP	Maximum a Posteriori
MCGD	Manifold Conjugate Gradient Descent
MIMO	Multiple Input Multiple Output
ML	Maximum Likelihood
NP	Nondeterministic Polynomial time
s.t.	Subject to
SVD	Singular Value Decomposition
PAST	Projection Aproximation Subspace Tracking
p.d.f	Probability density function
PPP	Precise Point Positioning
WLS	Weighted Least Squares

List of Figures

Contents

1 Introduction

This section will be devoted to the motivation and description of the contents of this thesis. We remark the fact that the realization of this thesis consisted on the exploration of some alternative mathematical concepts in the context of Signal Processing applications. We will further detail the applications and the mathematical concepts.

1.1 Background

Nowadays, there are lots of applications where the detection of unexpected behaviours or anomalies in the current data stream is of a great value. These applications are now related to hot topic areas such as business intelligence, machine learning, computer vision, security applications, among others, where the detection of these out-of-model behaviours can be extremely crucial. Furthermore, assuming that there is some kind of redundancy in the data, being usually the case nowadays, we can improve greatly the performance of the detection of these unexpected behaviours by noting that these anomalies usually lie in a subspace out of the main terms of the data streams. Thus, if those subspaces are precisely estimated, one could decouple the problem of estimating the latent variables and the anomaly detection from the data. This is where the main problem of Subspace Learning makes an appearance, as its main goal is estimating a given subspace from the data, according to certain restrictions. This work is highly motivated by two topics: solving the phase ambiguities in the Global Navigation Satellite System in Precise Point Positioning mode and the geometric approach of non-coherent MIMO communications.

The first inspiring context for this project is the one depicted by GNSS PPP satellites [19]. The general GNSS PPP algorithm is based in both code and phase observations from a dual-frequency receiver, by means of the called ionospheric-free combination, being a data processing whose main objective is to cancel out the effects of the ionosphere, to compute the receivers location and clock from the input signals. The multimodal multisatellite data is processed together in the receiver, so different unkowns can be extracted from it, being the receiver's clock and the receiver location. The inherent nature of phase measurements makes them susceptible to phase discontinuities whenever there are unexpected artifacts in the input signal. In our modeling, these unknowns are the latent variables and the phase ambiguities, the unwanted behaviour or outliers of the signals. These outliers are seen in the input signal as discontinuities in the phase measurements, observed as jumps of integer numbers of the carrier wavelength. What is more, these discontinuities are often the cause why the PPP receivers lose their centimetric resolution.

Focusing on these phase discontinuities, they are caused by the inherent electromagnetic properties of the ionosphere as it is sensible to electromagnetic interactions which may come from solar activity, radio bursts, refraction and difraction of the radio waves, which can be all described by the name of ionospheric scintillation. In our common communications context, this can be modeled as a multipath fading channel, in addition to changes in the mean of the input data stream.

In this modeling, considering a multisatellite approach of the GNSS PPP navigation, the main objective is tackling these phase ambiguities by the cancelation of these mean changes in the signal. However, we do not consider any kind of prior knowledge of the channel, so the proposed solution must also tackle this issue. In this way, it is known that the perceived channel must account for the quick variabilities due to the satellite movement and relative movement between satellites. However, the relative position vectors vary much faster than the subspace spanned by them. This is where the Subspace Tracking fits in, as it is a way to estimate the signal's subspace without needing to estimate the latent variables and in that way, the search for the anomalies can be done in the noise's subspace much more efficiently and decoupled from the estimation of receivers' location and clocks. To this extent, the proposed solution is also independent from whether one would want to use the ionospheric free combination or each frequency's signal, in order to detect phase ambiguities. In Figure 1, we show a general scheme that sums up this application.

On the other hand, the remaining solution is related to smart MIMO communication, where the underlying application is similar to the one in GNSS PPP regarding the Channel State Information. MIMO communications have been a key physical technology which have proven to be necessary to achieve higher rates in wireless communication. The main feature that it contributes is taking advantage of the spatial

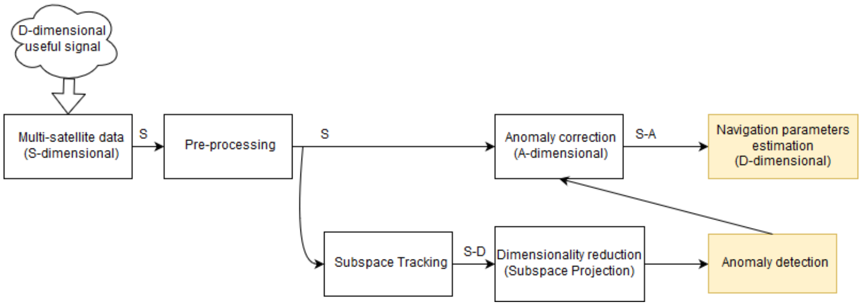

D-dimensional useful signal
Multi-satellite data (S-dimensional)
S
Pre-processing
S
Anomaly correction (A-dimensional)
S-A
Navigation parameters estimation (D-dimensional)
Subspace Tracking
S-D
Dimensionality reduction (Subspace Projection)
Anomaly detection

Figure 1: Subspace-based Anomaly Detection and Correction scheme.

degrees of freedom generated by antenna arrays to thrive data rates by creating additional virtual channels, providing them by spatial multiplexing or increase link reliability by introducing spatial diversity in the channel. However, in millimiter wave communications, it is possible to introduce an extremely large amount of antennas in the devices, due to the reduced carrier wavelenght. This last feature increases inmensely the total dimensionality of the input signal which, consequently, increases the required amount of computational power in order to maintain a reasonable latency figure. This increment in dimensionality brings a change of paradigm, where now this systems are called Large or Extremely Large MIMO, and its classical solutions often bring hybrid analogic and digital solutions. On the other hand, there could be solutions that are computationally able to be fully digital, as the ones based in non-coherent smart MIMO communications.

The millimeter wave large MIMO communications can benefit greatly from the subspace learning from the fact that in this way the total computational cost is reduced. There, the Grassmann manifold has been studied greatly for non-coherent MIMO modulations [16] in smart MIMO communications. Even though coherent MIMO is preferred over non-coherent MIMO for the increased rate, there is still room for non-coherent communications in the applications of low latency-low rate communications in 5G. For example, in Figure 2 we show an example of non-coherent Grassmann-based MIMO communications' scheme proposed in [20].

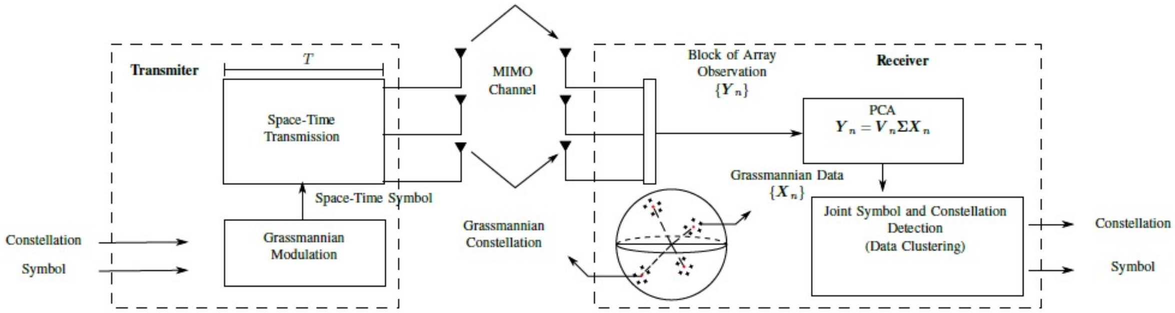


Figure 2: Automatic recognition of space-time constellation by learning on the Grassmann manifold for an intelligent MIMO communication system. Figure extracted from [16].

The solution proposed in Part III of this thesis is highly addressed to tackle both problems in a novel way. In the case of non-coherent MIMO communications, all the proposed framework is focused on anomalies in the signal such as hardware malfunctioning or even interference cancellation.

Finally, we should mention several hot applications of Grassmann-based algorithms which are widely known such as Recommender Systems (NetFlix problem for instance), Deep Learning model's interpretability, Image-set/Video Based Recognition and Classification, among others.

1.2 Thesis outline

This thesis consists on three main parts and each part is focused on independent concepts that we have mentioned. The first two parts are independent of each other, as the related methodology is related but separable, and the final part is the intersection of the firsts two parts. In Figure 3 we show the different parts of this thesis, the different concepts in them and their logical relationship. In the stated figure, we highlight in red the main parts of this thesis and in yellow the original contributions in this thesis, where the ones circled in red are subsceptible to be published.

Here we state a summary of these new ideas, in order to facilitate the appreciation of them.

In the sparse anomaly detection part, the novel ideas are the following:

- Particularization in the Iterative Shrinkage-Thresholding Algorithm, improving its convergence when some conditions are met.
- Dual Ascent for Sparse Anomaly Detection.

In the Subspace Learning part, the novel ideas that are related or new from this work are:

- Affine Projection Subspace Tracking, where this algorithm was developed with the foundations and in support of this work and its results will be presented on EUSIPCO 2021.
- Weighted Least Squares-based Manifold Conjugate Gradient Descent, being based in the Grassmann Rank One Update Subspace Estimation.

And finally in the Subspace-based anomaly detection, all the presented ideas are novel and, in particular, the derivation of the Subspace-based Dual Ascent for Anomaly Detection.

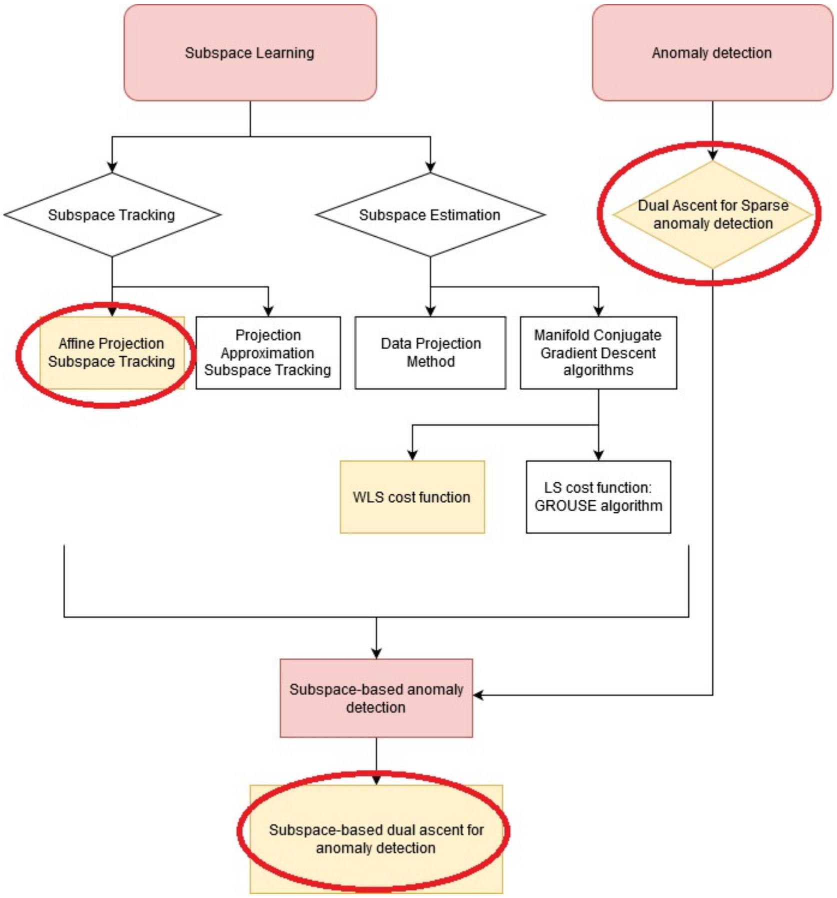

Subspace Learning
Anomaly detection
Subspace Tracking
Subspace Estimation
Dual Ascent for Sparse anomaly detection
Affine Projection Subspace Tracking
Projection Approximation Subspace Tracking
Data Projection Method
Manifold Conjugate Gradient Descent algorithms
WLS cost function
LS cost function: GROUSE algorithm
Subspace-based anomaly detection
Subspace-based dual ascent for anomaly detection

Figure 3: Master's thesis structure.

Part I

Anomaly detection

This part will be mainly focused on the Sparse anomaly detection. It will be seen throughtout this part that we will get rid of the non-stationarity issues that may arise in the motivating applications of this work. The main goal now is to develop an algorithm that is capable of learning all the necessary parameters that it needs, so in this way it minimizes the number of user-defined parameters.

As the reader may note, the presented approach is highly related to the Basis Pursuit denoising algorithm, where the sparsity regularization is usually governed by an arbitrary parameter, λ. In our proposed algorithm, we are giving a more interpretable approach to estimate intuitively this regularizing parameter. The basic Basis Pursuit denoising algorithm approach is based on the following general cost function

$$\min_{\mathbf{x}} \left(||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2 + \lambda ||\mathbf{x}||_1 \right),$$

where it usually considers an abitrary value of λ, which regularizes the amount of sparsity of the solution. This last feature is induced by the l_1-norm term and it has a rationale behind it. It has been greatly studied [4] that the best regularizer term for sparsity inducing solution, understood by the term multiplying the regularizer parameter, λ, is the l_0 norm. However, the set of l_p-norms with $p < 1$ (often called quasinorms) are non-convex, so it makes them unsuitable for an optimization problem because it leads to an NP-hard problem. This kind of problems are non-scalable as it complexity scales exponentially with the dimensionality of the problem. Nevertheless, the proposed solution in the Compress Sensing theory is to relax the l_0-norm with the l_1-norm, as it still keeps the sparse inducing properies. In the following figure, we show a comparison between a set of l_p-norms.

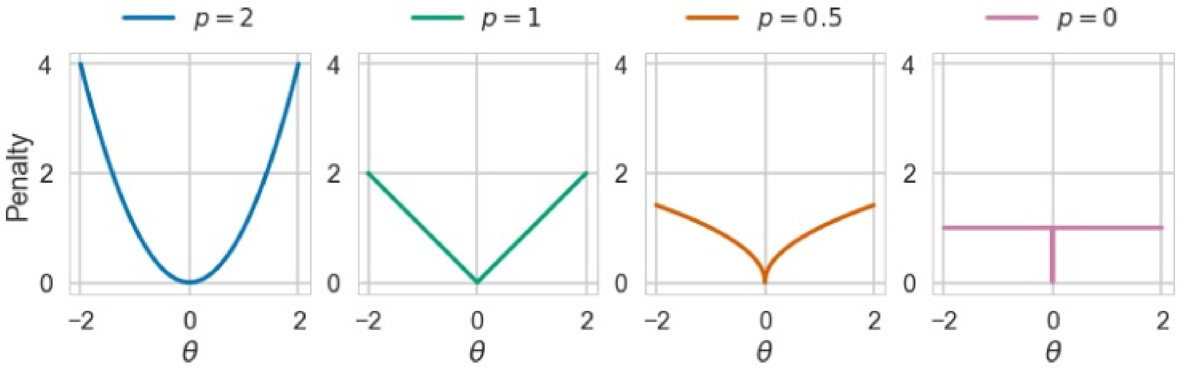


Figure 4: Comparison between some l_p-norms. Figure taken from [30].

The reason behind the sparse inducing properties is related to the spikiness of the norm function. In this way, the spikier the function, the more sparse inducing it is. On the other hand, larger values of λ lead to more sparse solutions, meaning that they have a reduced amount of non-zero entries, as it gives more weighting to the regularizing term. In some sense, the regularization of some arbitrary function can be interpreted as the application of the Occam's razor in the solution that we expect. In other words, what this regularization term means is *among all the possible solutions that we can have for our problem, we are looking for the simpl*est *one.*

The main criticism in Basis Pursuit denoising is that the value of λ is often computed by cross-validation, lossing all sense of interpretability. In this way, some authors, similarly to our extended approach, prefer the following constrained problem instead

$$\min_{\mathbf{x}} ||\mathbf{x}||_1 \quad \text{s.t. } ||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2 \leq \delta.$$

For some arbitrary δ. This version of Basis Pursuit denoising algorithm also benefits from the interpretation point of view, although it hinders the derivation of the solution because of the fulfilling of the Karush-Kuhn-Tucker (KKT) and regularity conditions. As it may be noticed in this part, our approach is focused on the optimization of a generalized version of those cost functions, while having the optimality of the cross-validation and the interpretability of the constrained problem. Also, it offers robustness to a tracking scenario, necessary for the subsequent parts, by the continuous learning of the regularizing parameter, λ.

In order to solve and understand that kind of optimization problems the convex optimization theory is quite helpful. As we will show briefly, all the functions and sets related to the proposed approach for Anomaly Detecion are convex or concave. Therefore, this kind of functions require this theory in order to develop algorithms and tools for this domain.

2 Mathematical tools: Convex optimization theory

In this chapter, we will mainly be using convex optimization concepts in order to derive an algorithm for anomaly estimation and correction in a given data block. In this section we will show some key concepts that we will be helpful to elaborate our proposed algorithm and study its optimality conditions.

Firstly, we will need to recall the convex and concave conditions. Consider two arbitrary points, x and y, belonging to the domain of a convex or concave function, $f(\cdot)$. Then, the next two equations are the convex and concave conditions respectively:

$$f(tx + (1-t)y) \leq tf(x) + (1-t)f(y), \tag{1}$$

$$f(tx + (1-t)y) \geq tf(x) + (1-t)f(y). \tag{2}$$

For all $t \in [0,1]$. In this part of the thesis we will be dealing with a limited set of known functions, so here we summarize the properties some functions that will be appearing continuously:

- The family of norm functions, $f(\mathbf{x}) = ||\mathbf{x}||_p$, is convex as long as $p \geq 1$.
- The matrix quadratic form, $f(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x}$, is convex as long as $\mathbf{A}$ is a semidefinite positive square matrix.
- All affine functions, $f(\mathbf{x}) = \mathbf{a}^T\mathbf{x} + b$, are both convex and concave, as they satisfy the conditions in (1) or (2) with equality.

In the context of mathematical optimization, we can find the concept of subderivative. A subderivative at an arbitrary x_0, easily extended to a subgradient, is the value g such that

$$f(y) \geq f(x_0) + g\cdot(y - x_0) \quad \forall y, \tag{3}$$

where in the following figure we show some insights of these concepts and their usefulness in a non differentiable function.

In this figure, you can see that the subderivative is a generalization of the derivative for convex non-differentiable functions. In this sense, the subderivative can be seen as the value of the slope g such that there exists lines with this slope such that the condition in (3) is satisfied in all points of the function's domain. Note that convexity, or concavity, is a necessary condition of a given function for the existence of a subderivative. We encourage the reader to have this in mind when dealing with the absolute value function or the l_1-norm.

Now, in the following subsections, we will explain the key concepts of convex optimization that are useful and necessary to fully comprehend the current part. What is more, in this thesis we will be dealing with constrained optimization problems, but inside this set of optimization problems, we will not be considering equality contraints. With this idea in mind, we will particularize the convex optimization concepts in those optimization problems without equality constraints.

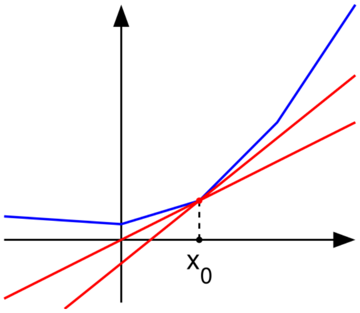


Figure 5: Subgradient in a non differentiable function.

2.1 Duality

Consider the following constrained optimization problem:

$$\arg\min_{\mathbf{x}} f_0(\mathbf{x}) \quad \text{s.t. } f_1(\mathbf{x}) \leq 0, \ldots, f_N(\mathbf{x}) \leq 0, \tag{4}$$

where we get its unconstrained form by deriving its Lagrangian function as:

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = f_0(\mathbf{x}) + \sum_{i=1}^{N} \lambda_i f_i(\mathbf{x}), \tag{5}$$

where now $\boldsymbol{\lambda}$ is the vector containing the Lagrange multipliers, λ_i. Then, one can define the dual problem as the one resulting from getting the value of $\mathbf{x}$ that minimizes the Lagrangian in (4) and evaluating (4) in this optimal point. Mathematically, it can be expressed in the following way:

$$g(\boldsymbol{\lambda}) = \min_{\mathbf{x}} \left(\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda})\right) = \min_{\mathbf{x}} \left(f_0(\mathbf{x}) + \sum_{i=1}^{N} \lambda_i f_i(\mathbf{x}) \right). \tag{6}$$

In the resulting problem, we will now denote $\boldsymbol{\lambda}$ as the dual variables. As we can see from (6), the dual problem is a point-wise infimum of affine functions of the dual variables, $\boldsymbol{\lambda}$, which implies that this function is always concave due to the minimum operation and thus we will look for the maximum value of (6).

This function is interesting in the optimization theory due to the two main properties of this dual function. This function is always concave even when the original problem is not concave or convex, so this enables it to be studied much easier than the original problem. In other words, this problem is much easier to handle than the original, also called the primal problem. The remaining property is the following:

$$g(\boldsymbol{\lambda}) \leq p_{opt}, \tag{7}$$

being p_{opt} the optimal value of the primal problem. If we define the optimal value of the dual problem as d_{opt}, then the duality gap is defined as:

$$g_{dual} = p_{opt} - d_{opt}, \tag{8}$$

and it is zero when the optimization problem has strong duality. The strong duality is a key property in an optimization problem, as it is much easier to optimize the dual problem than the primal. In order to achieve this zero duality gap there are several conditions to fulfill, called the Karush-Kuhn-Tucker conditions, ensuring the optimality of the found solution.

2.2 Karush-Kuhn-Tucker and regularity conditions

In the optimization problem, such as the one in (4) and assuming that the optimal value of the primal problem is $\mathbf{x}_{opt}$ and the optimal for the dual is $\boldsymbol{\lambda}_{opt}$, the KKT conditions over those values ensure the optimality of the solution. These conditions are the following ones:

Stationary condition Considering the Lagrangian formulation in (5), the stationary condition can be formulated as:

$$\nabla f_0(\mathbf{x}_{opt}) + \sum_{i=1}^{N} \lambda_i \nabla f_i(\mathbf{x}_{opt}) = \mathbf{0}, \tag{9}$$

where the ∇ operator can be either the differential or the subdifferential operator.

Primal feasibility The primal problem is feasible if and only if all the constraints are mutually feasible:

$$f_1(\mathbf{x}_{opt}) \leq 0,\ ...,\ f_N(\mathbf{x}_{opt}) \leq 0. \tag{10}$$

Dual feasibility

$$\boldsymbol{\lambda}_{opt} \succeq 0. \tag{11}$$

This condition requires all the dual variables to be greater or equal than zero. In this way, they ensure that the dual problem is always concave.

Complementary slackness

$$\lambda_{i_{opt}} f_i(\mathbf{x}_{opt}) = 0 \quad i = 1,\ ...,\ N. \tag{12}$$

When there are no inequality constraints in the optimization problem, there is no function that must satisfy the complementary slackness condition, so it becomes a nuissance condition.

In the case of convex problems, meaning that all the convex functions and sets are convex, the KKT conditions are sufficient for $\mathbf{x}_{opt}$ and $\boldsymbol{\lambda}_{opt}$ to be primal and dual optimal, and also for ensuring the zero duality gap.

2.3 Shrinkage operator

In the context of this work, we will be dealing with generalizations of the following unconstrained optimization problem:

$$\min_{\mathbf{x}} \frac{1}{2}||\mathbf{y} - \mathbf{x}||_2^2 + \lambda||\mathbf{x}||_1, \tag{13}$$

where in order to derive the optimal solution, it may be easier to firstly consider the one-dimensional version:

$$\min_{x} \frac{1}{2}(y - x)^2 + \lambda|x|. \tag{14}$$

Now we take the derivative (subderivative in the case of the absolute value function) and equate it to zero, so we get to the following equation:

$$y = x + \lambda \text{sign}(x), \tag{15}$$

whose solution is the Shrinkage operator. The solution of the equation in (15) can be derived graphically, as shown in Figure 6.

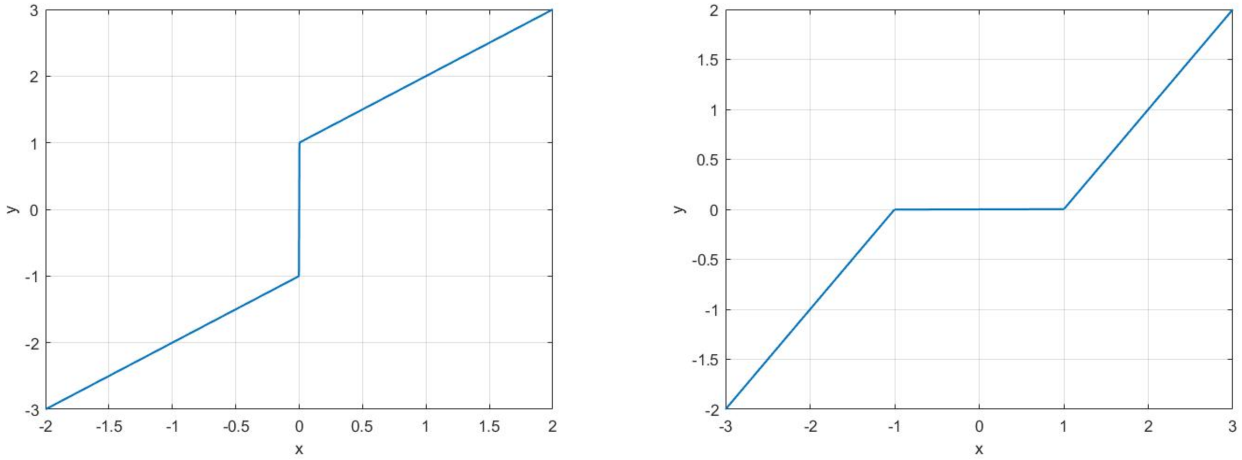
Now, we can see from the inverse function of (15) that the Shrinkage operator is a function defined in the following way:

$$S_\lambda(x) = \begin{cases} x - \lambda & \text{if } x > \lambda \\ 0 & \text{if } |x| \leq \lambda \\ x + \lambda & \text{if } x < -\lambda \end{cases}. \tag{16}$$

This function is also called the soft-thresholding operator. In addition, it can also be rewritten in the following compact form:

$$S_\lambda(x) = \max(|x| - \lambda, 0) \times \text{sign}(x). \tag{17}$$

Finally, we can extend easily this one-dimensional result to the vectorial case by simply extending the function in (17) to an element-wise operator of these characteristics, as this can be done in the vectorial form of optimization problem in (14) because this setting makes all the variables decoupled. In other words, we can find the optimal solution by considering each dimension independently.



(a) Function in (15).

(b) Inverse function of (15).

Figure 6: Shrinkage operator: Graphical derivation with $\lambda = 1$.

3 Problem statement

We will consider the following simplistic model, covering every aspect that we want to explore throughout all this work:

$$\mathbf{s}(n) = \mathbf{H}(n)\mathbf{x}(n) + \mathbf{a}(n) + \mathbf{w}(n). \tag{18}$$

For $n = 1, ..., N$. Vectors $\mathbf{s}(n)$, $\mathbf{w}(n)$ and $\mathbf{a}(n)$ have dimension $M \times 1$ while vector $\mathbf{x}(n)$ has dimensions $D \times 1$, being $D < M$. Then, $\mathbf{w}(n)$ denotes the noise, distributed as $\mathcal{N}(\mathbf{0}, \mathbf{C})$ and $\mathbf{a}(n)$ represents the sparse anomaly, which will be modeling an accumulation of S_0 step functions. As it will be stated later in this section, this term is modeling a change in the mean, often found in the context of Robust Regression. Finally, $\mathbf{H}(n)$ is the $M \times D$ unitary matrix spanning the signal's subspace. Note that we are considering N time instants, so all the stated vectors and matrices could potentially be non-stationary. However, in this part we will consider stationarity, for initial simplicity.

The main purpose of this toy problem is to estimate $\mathbf{a}(n)$ (or subsequent modifications of this vector), where we will be using this estimation of anomalies to correct the input signal, so the estimation of the latent variables,$\mathbf{x}(n)$, is improved. We are assuming that the sparsity is initially known, denoted S_0, and is defined in this case as the average amount of steps that are present in our signal. In Figure 7 is shown an example of the kind of anomaly that we will handle.

As shown in Figure 7, this kind of anomaly has an inherent memory along all the components of $\mathbf{a}(n)$ vector. This memory usually adds more difficulty in the estimation of the anomaly, as the possible classical solutions lead to an NP-hard problem. The classical batch solutions are often based on the combinatorial search of a sequence that best fits the signal, falling into an NP-hard problem which is not scalable and computationally costly. There are also the online-type of algorithms that solve this problem by continuously estimating the optimal sequence of anomalies, being computationally more feasible. This anomaly will be characterized by the following parameters: we will denote as p_a, the probability that an anomaly with any sign occurs in each independent entry of our signal's vector, being $p_{a|pos}$ the probability that a positive signed anomaly conditioned to the event of an anomaly and $p_{a|neg}$ the probability of a negative signed anomaly. As we are expecting anomalies of any sign in these signals, for the sake of simplicity, we will assume the following assumption:

$$p_{a|pos} = p_{a|neg} = 0.5. \tag{19}$$

In this way, we will mainly be focused on p_a. As S_0 is shown to be the average number of discontinuities in the vector of anomalies, $\mathbf{a}(n)$, we could formulate an estimation of this value by using the Weak Law of

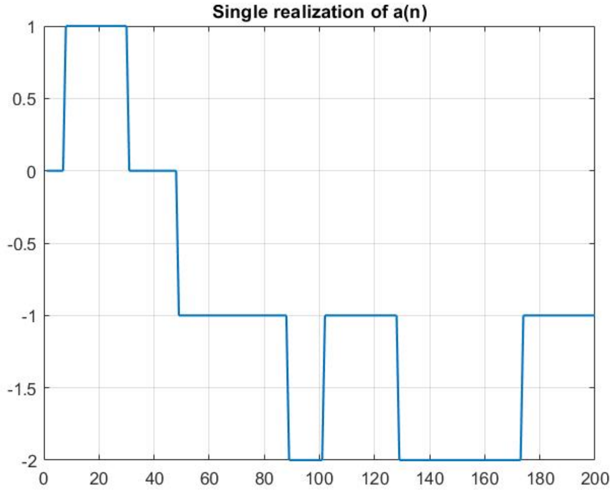


Figure 7: Anomaly example. Sequence with memory.

Large numbers as

$$S_0 \approx \frac{1}{J_0 N} \sum_{i=1}^{N} |\text{diff}(\mathbf{a}(n))|^T \mathbf{1}_{M-1}, \tag{20}$$

where diff$(\cdot)$ is the discrete derivative operator. In a similar way, there is a tight relationship between p_a and the sparsity level S_0, where the explicit relationship is shown to be

$$p_a \approx \frac{1}{J_0 N M} \sum_{i=1}^{N} |\text{diff}(\mathbf{a}(n))|^T \mathbf{1}_{M-1} \approx \frac{S_0}{M}. \tag{21}$$

Note that the above expressions become an equality when N tends to infinity. The last parameter to consider from the anomalies is the step level, J_o, shown in the previous equations, which we consider arbitrary, and not known.

This kind of anomaly makes sense in environments related to frequency or phase estimation ambiguities, such as the ones appearing in GNSS Precise Point Positioning. As it will be stated later in this part, we will particularize this model so it is able to model continuous change in the mean in a single snapshot. Depending on which is the main focus of this problem in a given application, there are two main lines of research: robust regression or non-parametric regression. If one is more interested in the estimation of the latent variables $\mathbf{x}(n)$, then the resulting problem can fit into the robust regression problems, consisting on finding the regression curve that best fit into the data while taking into consideration that the data stream may contain artifacts or anomalies [17]. On the other hand, if one is instead interested in the estimation of the anomaly sequence, $\mathbf{a}(n)$, then these kind of problems are considered an instance of non-parametric regression, due to not being able to parametrically model the distribution of the anomalies [18]. Both problems are quite studied and have huge interest in hot domains such as Machine Learning or Data Analytics.

As a generic processing of our signal to tackle this issue, we will be performing the discrete derivative in the signal, so this step-like anomaly become spikes in the signal, being a more appropriate way to deal with sparse-like terms in the literature [4]. The resulting signal is the following:

$$\mathbf{y}(n) = \text{diff}(\mathbf{s}(n)) = \mathbf{D}\left(\mathbf{H}(n)\mathbf{x}(n) + \mathbf{a}(n) + \mathbf{w}(n)\right) = \mathbf{H}'(n)\mathbf{x}(n) + \boldsymbol{\theta}(n) + \mathbf{w}'(n), \tag{22}$$

where $\mathbf{D}$ is the matrix that performs the discrete derivative, $\mathbf{H}'(n) = \mathbf{D}\mathbf{H}(n)$, $\mathbf{w}'(n)$ is the equivalent noise after the derivative enhanced by this operation, by a factor of 2 in the case of white gaussian noise. This equivalent noise is distributed as $\mathcal{N}(\mathbf{0}, \mathbf{DCD}^H)$. $\boldsymbol{\theta}(n)$ is the transformed memoryless sparse anomaly, which is in essence what a sparse signal is expected to be, this means having S_0 non-zero entries being $S_0 \ll M$. We remark the fact that the discrete derivative makes us lose all the information of the first order statistical moment (the mean) of these terms. However, this loss of information is compensated by the fact that the discrete derivative also removes the memory present in the anomaly $\mathbf{a}(n)$, transformed into $\boldsymbol{\theta}(n)$, while also introducing more correlation onto the noise, now denoted as $\mathbf{w}'(n)$. In the case of white gaussian noise, we are translating the inherent memory of the anomaly to the noise. This reallocation of memory reduces the complexity in the solution due to the fact that it is easier to handle correlation in the Gaussian terms of data streams rather than in other kinds of distributions, especially in cases such as the one that we are trying to solve where the distribution of the anomaly is in principle unknown. In Figure 8, we show the resulting anomaly signal, $\boldsymbol{\theta}(n)$, after taking the discrete derivative from the example in Figure 7.

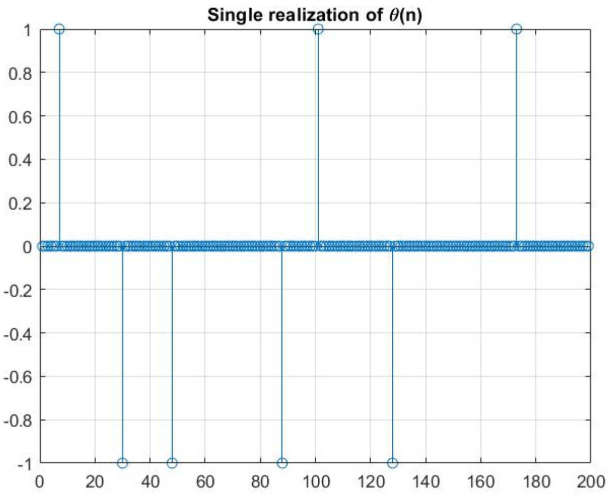


Figure 8: Anomaly example. Sequence without memory.

Note in Figure 8 that we completely removed the memory of this signal and processed this signal into an actual sparse signal, having a reduced amount of non-zero entries.

3.1 Bayesian estimation framework

As we are introducing priors for the sparse anomaly, we can introduce the Bayesian framework. We will make use of the Maximum a Posteriori (MAP) estimation for the sparse anomaly to improve the overall estimation. For a general MAP estimation, the parameters are obtained in the following way:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} f(\boldsymbol{\theta}|\mathbf{y}), \tag{23}$$

where $f(\boldsymbol{\theta}|\mathbf{y})$ is what is called the conditional pdf of the sparse denoting the *belief* in the Bayesian framework. We can obtain an alternative form of this derivation by using the Bayes' rule:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \frac{f(\mathbf{y}|\boldsymbol{\theta})f(\boldsymbol{\theta})}{f(\mathbf{y})} = \arg\max_{\boldsymbol{\theta}} f(\mathbf{y}|\boldsymbol{\theta})f(\boldsymbol{\theta}), \tag{24}$$

where now $f(\mathbf{y}|\boldsymbol{\theta})$ is the *likelihood,* $f(\boldsymbol{\theta})$ is the *prior knowledge* and $f(\mathbf{y})$ is the *evidence.* To estimate the anomalies $\boldsymbol{\theta}$, we will make use of all the available information to obtain the sequence of sparse anomalies that maximizes our *belief.* A more convenient form of the MAP formulation is the following:

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \ln(f(\mathbf{y}|\boldsymbol{\theta})) + \ln(f(\boldsymbol{\theta})), \tag{25}$$

which is equivalent as the one in (25), as it would yield in the same result due to the natural logarithm monotonic behaviour.

Ideally, we would want to make use of the following prior:

$$f(\boldsymbol{\theta}) = \prod_{i=1}^{M-1} p_a^{\mathrm{sign}(|\theta_i|)}(1-p_a)^{1-\mathrm{sign}(|\theta_i|)}, \tag{26}$$

where the reader can note that the memoryless anomalies behave statistically with this distribution, which assumes independence, as the statistical independence is achieved by the memory cancellation. This prior would force us to implement a combinatorial search on the overall sequence to detect the sparse anomaly in any input sequence of M samples. Note that we are considering that θ_i can take any possible value. This combinatorial search would not be scalable with N, as this problem would be equivalent as minimizing the l_0-norm subject to some fitting constraints which is known to be a NP-hard problem. Therefore, one would need to look for alternative solutions and relaxations. In this work, we propose the use of the following prior:

$$f(\boldsymbol{\theta}) = \left(\frac{\lambda}{2}\right)^{M-1} e^{-\lambda||\boldsymbol{\theta}||_1}, \tag{27}$$

where we remark that the scale parameter, λ, must be any positive real constant. Note that this new parameter is the one that is related with the sparsity prior, and hence, with the probability p_a.

The motivation of this new prior is the relaxation of the previous prior in (26) to a distribution from the exponential family of distributions. This can act as a sparse generating prior due to the fact that as this distribution has a higher density around the zero, so it is more prone to produce sparse solutions. By higher density we mean higher values in the probability density function of the distribution. Moreover, the non-zero entries are considered to be outliers, being the entries corresponding to the sparse anomaly, which have also a higher density in the prior in (27) than in the Gaussian distribution case. What is more, the fact that the outliers values have a higher density as compared to other distributions, makes the Laplacian a suitable prior to find out-of-model anomalies.

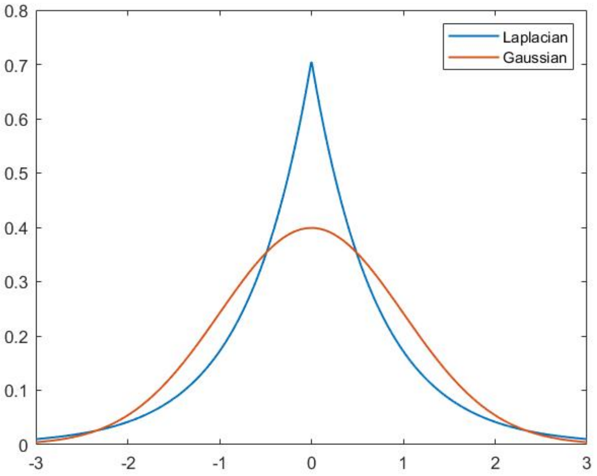


Figure 9: Comparison between zero-mean unit-variance Gaussian and Laplacian distributions.

In Figure 9, we show the comparison between a zero-mean unit-variance Laplacian with a zero-mean unit-variance Gaussian to show the difference in density around zero of both distributions. This difference in their densities can be related as the statistical comparison between the l_1 and l_2 norms [6], in the context of generating sparse solutions with regularization terms.

4 Methodology

As it was stated in the previous sections, we will be mainly focused on the optimization of the MAP function, particularized on the proposed prior in (27) and with a simplification on the signal's model stated in (18), to have an initial simplicity. This simplification has the following form:

$$\mathbf{y}(n) = \mathbf{H}'(n)\mathbf{x}(n) + \boldsymbol{\theta}(n) + \mathbf{w}'(n) = A\mathbf{1} + \boldsymbol{\theta} + \mathbf{w}', \tag{28}$$

where now we note that we have simplified the signal's subspace term to be more manageable, so we can focus the analysis on the anomaly estimation, in addition of removing the temporal variability of the model, where in this case we are just considering a single snapshot, so $N = 1$. Also, for simplicity, we will note the previously shown equivalent noise distribution as $\mathcal{N}(\mathbf{0}, \mathbf{K})$. Note that this particularization is equivalent to a model of a slope immersed in correlated (or not) noise with continuous changes in the mean, considering that the derivative of a slope is a constant term such as the signal's term in (28).

Having particularized the model and shown the framework that we are pursuing, this leads us to simultaneously estimate the signal's parameter, A, and the anomaly's vector, $\boldsymbol{\theta}$, by means of optimizing the MAP function:

$$\hat{\boldsymbol{\theta}}, \hat{A} = \arg\max_{\boldsymbol{\theta},A} f(\mathbf{y}|\boldsymbol{\theta}, A) f(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta},A} \frac{1}{(2\pi)^{(M-1)/2}\det(\mathbf{K})} e^{-(\mathbf{y}-A\mathbf{1}-\boldsymbol{\theta})^T\mathbf{K}^{-1}(\mathbf{y}-A\mathbf{1}-\boldsymbol{\theta})} \left(\frac{\lambda}{2}\right)^{M-1} e^{-\lambda||\boldsymbol{\theta}||_1}. \tag{29}$$

After applying the natural logarithm and removing constants, we get the following objective function:

$$\hat{\boldsymbol{\theta}}, \hat{A} = \arg\min_{\boldsymbol{\theta},A} (\mathbf{y}-A\mathbf{1} - \boldsymbol{\theta})^T\mathbf{K}^{-1}(\mathbf{y}-A\mathbf{1} - \boldsymbol{\theta}) + \lambda||\boldsymbol{\theta}||_1. \tag{30}$$

As both variables are coupled, we have chosen to use an iterative method to solve this optimization, adding the fact that the l_1-norm gradient is not defined in certain points, so we will not have any closed-form solution for this problem. However, we can still optimize this function by means of its subderivative, which we will further explain below. Note that this function is similar to the LASSO [5] or Basis Pursuit objective function.

Another issue that we can find in this approach is estimating the scale parameter, λ. As it may be noted from (30) and comparing to the expressions in the introduction of this part, before the mathematical tools, that this parameter is tightly related to the sparsity level of the solution, as it weights the importance between the amount of desired fitting of the solution, given by the first term in (30), and the sparse solution's inducer, denoted by the second term in (30).

Having these ideas in mind, we want to find an algorithm that estimates simultaneously the latent variables, $\boldsymbol{\theta}$ and A, but also the sparse parameter, λ, according to some predefined constraint while still having robustness in the estimation of the sparse parameter. In this way, we can relate the sparsity of our solution to a more interpretable parameter and thus the algorithm finds a way to find the optimal amount of regularization, as compared to the LASSO or Basis Pursuit denoising algorithms.

4.1 Proposed algorithm definition

In order to accomplish the previously mentioned goals of our proposed algorithm, one needs to note that optimizing the objective function in (30) is equivalent to optimizing the following function:

$$\hat{\boldsymbol{\theta}}, \hat{A} = \arg\min_{\boldsymbol{\theta},A} \frac{1}{\lambda}\left((\mathbf{y}-A\mathbf{1} - \boldsymbol{\theta})^T\mathbf{K}^{-1}(\mathbf{y}-A\mathbf{1} - \boldsymbol{\theta}) - \gamma'\right) + ||\boldsymbol{\theta}||_1. \tag{31}$$

Note that for an arbitrary constant γ, the objective functions in (30) and in (31) have the same optimal solutions for the latent variables. Note that if we impose the following change of variable $\alpha' = \frac{1}{\lambda}$, the objective function in (31) can be interpreted as the Lagrangian function of the following constrained optimization problem:

$$\hat{\boldsymbol{\theta}}, \hat{A} = \arg\min_{\boldsymbol{\theta},A} ||\boldsymbol{\theta}||_1 \quad \text{s.t. } (\mathbf{y} - \boldsymbol{\theta} - A\mathbf{1})^T\mathbf{K}^{-1}(\mathbf{y} - \boldsymbol{\theta} - A\mathbf{1}) < \gamma', \tag{32}$$

where γ' is the desired fitting level that we would like to expect of the solution. Still, this parameter has an issue due to the behaviour of the left hand side of the above inequality. The value of quadratic term in (33) increases with M, even if the anomalies and the signal terms fit correctly in our signal, due to the fact that this term will still add noise components, whose number increases linearly with M. Therefore, a normalization of both the desired fitting level, γ, and the quadratic term, $(\mathbf{y} - \boldsymbol{\theta} - A\mathbf{1})^T\mathbf{K}^{-1}(\mathbf{y} - \boldsymbol{\theta} - A\mathbf{1})$, by a factor that should also be linearly proportional to M is necessary to cope with this property. Consequently, we will redefine γ' and substitute it by $\gamma = \frac{\gamma'}{\mathbf{1}^T\mathbf{K}^{-1}\mathbf{1}}$, so the noise correlation is taken into account. Having this in mind, we propose the following normalization of the contraints:

$$\hat{\boldsymbol{\theta}}, \hat{A} = \arg\min_{\boldsymbol{\theta}, A} ||\boldsymbol{\theta}||_1 \quad \text{s.t. } \frac{(\mathbf{y} - \boldsymbol{\theta} - A\mathbf{1})^T\mathbf{K}^{-1}(\mathbf{y} - \boldsymbol{\theta} - A\mathbf{1})}{\mathbf{1}^T\mathbf{K}^{-1}\mathbf{1}} < \gamma, \tag{33}$$

where, in addition to the issue on the dimensionality, we do not need to know the noise power but the correlation profile of the noise. Note that this normalization can be done and still maintain the equivalence between the problem in (32) and the problem in (33), as this constant can be absorbed by the Lagrange multiplier, α as $\alpha = \mathbf{1}^T\mathbf{K}^{-1}\mathbf{1}\alpha'$. Still, this multiplying constant does not need to be explicitly considered in the derivation of the algorithm as it would be accounted in the optimal numerical solution of the dual variable.

Note that the optimization problem in (33) is similar but not exactly equivalent as the one in (30), in the sense that there exists values of λ such that the solution in (30) is equivalent to the solution in (33), but the solution in (30) is more general (it has more solutions) than the solution in (33). In other words, the problem in (33) is restricting the values of λ, or α, from the optimal solution. In order to tackle this issue, we have chosen the dual ascent iterative method to get the optimal solution of this problem, which we will detail later.

Now, the function that we will be optimizing is the Lagrangian of the problem in (33), being following:

$$p(\boldsymbol{\theta}, A, \alpha) = \alpha \left(\frac{(\mathbf{y} - \boldsymbol{\theta} - A\mathbf{1})^T\mathbf{K}^{-1}(\mathbf{y} - \boldsymbol{\theta} - A\mathbf{1})}{\mathbf{1}^T\mathbf{K}^{-1}\mathbf{1}} - \gamma \right) + ||\boldsymbol{\theta}||_1, \tag{34}$$

where now we will tackle two different problems: the primal problem and the dual problem. In optimization theory, the primal problem would be the one resulting in optimizing the function in (34) with respect to the primal variables, $\boldsymbol{\theta}$ and A, considering the dual variables, α, as constants. Note that the optimal solutions of the latent variables for the primal problem are, in general, a function of the dual variables.

On the other hand, the dual problem is the one resulting from substituting in (34) the optimal solutions of the latent variables. This problem has the following shape:

$$g(\alpha) = \min_{\boldsymbol{\theta}, A} \alpha \left(\frac{(\mathbf{y} - \boldsymbol{\theta} - A\mathbf{1})^T\mathbf{K}^{-1}(\mathbf{y} - \boldsymbol{\theta} - A\mathbf{1})}{\mathbf{1}^T\mathbf{K}^{-1}\mathbf{1}} - \gamma \right) + ||\boldsymbol{\theta}||_1, \tag{35}$$

where we will denote as $\boldsymbol{\theta}_{opt}$ and A_{opt} the optimal values of the primal problem, which in this case is a convex problem, but non-smooth due to the l_1-norm. In order to solve both the primal and the dual problem, we will need to simultaneously compute $\boldsymbol{\theta}_{opt}$ and A_{opt}, while maximizing $g(\alpha)$. We should remark and ensure the fulfilling of the conditions needed to achieve the strong duality of this problem. On the one hand, we need to comply the feasibility and optimality of this problem by means of the KKT conditions. In this way, each one of the KKT optimality conditions must be strictly fulfilled. On the other hand, we must ensure that the problem is completely convex, meaning that the objective function and the feasible set must be convex. There are other more conditions to ensure the strong duality conditions, but the mentioned requirements are sufficient to achieve this property. We can easily show that the proposed approach in (33) is a completely convex problem, as we can easily remark that the objective function, $||\boldsymbol{\theta}||_1$, is convex as it is a norm function and also that the feasibility set is also convex, as the quadratic forms such as the one we are proposing, $\frac{(\mathbf{y} - \boldsymbol{\theta} - A\mathbf{1})^T\mathbf{K}^{-1}(\mathbf{y} - \boldsymbol{\theta} - A\mathbf{1})}{\mathbf{1}^T\mathbf{K}^{-1}\mathbf{1}}$, are also convex.

We are interested in having all the conditions necessary to achieve the zero duality gap due to the fact that we will base our approach in the maximization of the dual problem in (35) by means of the Subgradient Ascent method. The optimization of the dual problem with either the Subgradient or Gradient Ascent method is often called the Dual Ascent method.

4.2 Dual ascent: Derivation of the algorithm

In order to deduce the updating rules for each variable we need to tackle them in an independent way. Firstly, it is straight forward to see that the optimal solution for the signal parameter is

$$A_{opt} = \frac{\mathbf{1}^T\mathbf{K}^{-1}(\mathbf{y} - \boldsymbol{\theta}_{opt})}{\mathbf{1}^T\mathbf{K}^{-1}\mathbf{1}}, \tag{36}$$

where this solution is widely known in the context of classical Signal Processing. This solution is the Minimum Variance Unbiased Estimator of the mean for this problem considering $\boldsymbol{\theta}_{opt}$ the actual value of the anomalies and a Gaussian distribution for the noise. Note that it depends explicitly on the value of $\boldsymbol{\theta}$ and also implicitly on the value of α.

In order to derive the optimal solution, $\boldsymbol{\theta}_{opt}$, we need to notice a property of the model, regarding the maximizer of $\boldsymbol{\theta}$ in the fitting term (the constraint) in the primal problem:

$$\boldsymbol{\theta}_{opt}^{fitting} = \arg\min_{\boldsymbol{\theta}}(\mathbf{y} - \boldsymbol{\theta} - A\mathbf{1})^T\mathbf{K}^{-1}(\mathbf{y} - \boldsymbol{\theta} - A\mathbf{1}) = \arg\min_{\boldsymbol{\theta}} ||\mathbf{y} - A\mathbf{1} - \boldsymbol{\theta}||^2 = \mathbf{y} - A\mathbf{1}, \tag{37}$$

where now we could follow two formulations to get to the optimal solution, and updating rule, for this parameter. The first and easier formulation, is to relax the primal unconstrained cost function in (33) to no longer take into consideration the effects of the precision matrix, $\mathbf{K}^{-1}$, yielding the following function:

$$p_{relaxed}(\boldsymbol{\theta}) = \alpha\left(\frac{||\mathbf{y} - \boldsymbol{\theta} - A\mathbf{1}||^2}{(M-1)} - \gamma\right) + ||\boldsymbol{\theta}||_1, \tag{38}$$

where using the subdifferential of the l_1-norm to find the optimal value of this function, we get to the point-wise shrinkage solution for the residuals of our signal:

$$\boldsymbol{\theta}_{opt} = S_{\frac{1}{\alpha}}\left(\mathbf{y} - A_{opt}\mathbf{1}\right), \tag{39}$$

where $S_{\frac{1}{\alpha}}(\cdot)$ is the shrinkage operator. On the other hand, we could instead use a modification Iterative Shinkage-Thresholing Algorithm (ISTA) approach, derived by us to solve this particular problem. In general, ISTA algorithm [6] looks for the solution of a problem of the form:

$$\arg\min_{\boldsymbol{\theta}} f(\boldsymbol{\theta}) + \lambda||\boldsymbol{\theta}||_1,$$

leading to the adoption of an iterative solution, often using the Gradient Descent method to reach the optimal solution of $f(\boldsymbol{\theta})$, as:

$$\boldsymbol{\theta}_{k+1} = \arg\min_{\boldsymbol{\theta}} ||\boldsymbol{\theta} - (\boldsymbol{\theta}_k - \eta\nabla f(\boldsymbol{\theta}_k))||_2^2 + \lambda||\boldsymbol{\theta}||_1,$$

where if we apply simple calculus and the mathematical tools for the shrinkage operator that we have explained at the beginning of this part, we get to the iterative optimal solution which is:

$$\boldsymbol{\theta}_{k+1} = S_{\eta\lambda}\left(\boldsymbol{\theta}_k - \eta\nabla f(\boldsymbol{\theta}_k)\right),$$

where this solution is both using Gradient Descent and the soft-thresholding operator to find the optimal sparse solution. In our case, we propose a modification of this algorithm, suitable when the optimization of $f(\boldsymbol{\theta})$ has an easy closed form solution. This modification consists on substituting the Gradient Descent term by the optimal solution, which we know in advance for this solution from (37), in the given iteration as

$$\boldsymbol{\theta}_{k+1} = \arg\min_{\boldsymbol{\theta}} ||\boldsymbol{\theta} - \boldsymbol{\theta}_k^{opt}||_2^2 + \frac{1}{\alpha}||\boldsymbol{\theta}||_1, \tag{40}$$

where in this case

$$\boldsymbol{\theta}_k^{opt} = \mathbf{y} - A_k\mathbf{1}. \tag{41}$$

This approach gives the iterative optimal solution version of (39). The modification that we have introduced is taking advantage of the precomputed optimal value $\boldsymbol{\theta}_{opt}^{k}$ in place of the Gradient Descent term in (41).

Finally, we would need to find the optimal equations, or iterative solution, of α from the function in (35). As $\boldsymbol{\theta}_{opt}$ and A_{opt} are both function of α and the objective function has a non-smooth (non-differentiable) term, it makes the evaluation of the exact closed form optimal solution of α, complicated to derive. We cannot even get the exact gradient. However, the partial subderivative of (35) over α is straight forward to compute, as it is the constraint term [1]. Therefore, we could use this subderivative to get the optimal value via the subgradient ascent method. Using the subgradient ascent to solve the optimization of the dual problem is the so-called Dual Ascent algorithm. Having this in mind, the final update equation is:

$$\alpha_{k+1} = \alpha_k + \mu_k \left(\frac{(\mathbf{y} - \boldsymbol{\theta} - A\mathbf{1})^T \mathbf{K}^{-1} (\mathbf{y} - \boldsymbol{\theta} - A\mathbf{1})}{\mathbf{1}^T \mathbf{K}^{-1} \mathbf{1}} - \gamma \right), \tag{42}$$

where μ_k is an appropriate step-size for iteration k.

As a summary for this algorithm, we sum up the iterative method in the following three iterative equations:

$$A_{k+1} = \frac{\mathbf{1}^T \mathbf{K}^{-1} (\mathbf{y} - \boldsymbol{\theta}_k)}{\mathbf{1}^T \mathbf{K}^{-1} \mathbf{1}}, \tag{43}$$

$$\boldsymbol{\theta}_{k+1} = S_{\frac{1}{\alpha_k}} \left(\mathbf{y} - A_{k+1}\mathbf{1} \right), \tag{44}$$

$$u = \alpha_k + \mu_k \left(\frac{(\mathbf{y} - \boldsymbol{\theta}_{k+1} - A_{k+1}\mathbf{1})^T \mathbf{K}^{-1} (\mathbf{y} - \boldsymbol{\theta}_{k+1} - A_{k+1}\mathbf{1})}{\mathbf{1}^T \mathbf{K}^{-1} \mathbf{1}} - \gamma \right),$$

$$\alpha_{k+1} = \max\left(u, \varepsilon\right). \tag{45}$$

Note that the two free parameters of this algorithm are the sequence of step size μ_k and the fitting parameter γ. In addition, the maximum between the updating rule of α_k and some positive small constant ε is necessary to fulfill the dual feasibility condition. Even though, it may seem that this is just an arbitrary mathematical condition, it also yields a statistical interpretation where we remember that this α is a value derived from the scale parameter of the Laplacian distribution, λ, which should be always positive so this function is indeed obtained from a probability density function. Consequently, if α is positive then λ is also positive, as α is inversely proportional to λ. Also, in the convex optimization domain, if α at some point reaches a negative value, then the problem would not fulfill the convexity condition and hence this algorithm would suffer from numerical issues. For instance, for some realizations it would diverge.

As an optional remark, in the Machine Learning interpretation of the algorithms, the iterative version of the Dual Ascent for Anomaly Detection wants to overfit the input signal as much as possible, because we are not asking it to generalize to other realizations of the input signal. Nevertheless, in the part III of this thesis, we will see that this approach will not be focused on overfitting the data, but to generalize it to possibly new realizations of the data stream.

4.2.1 Step sizes

In order to guarantee that the algorithm converges to a stationary point, we must ensure that the sequence of step-sizes, μ_k, must fulfill the following conditions [31]:

$$\sum_{k=1}^{\infty} \mu_k \to \infty, \tag{46}$$

$$\mu_k \to 0 \quad \text{when} \quad k \to \infty. \tag{47}$$

However, in practice, we may not exactly fulfill both conditions, for instance by making the sum of this sequence converge to a given value, and still get to a stationary point in the objective function. The simplest sequence that fulfills both conditions is $\mu_k = \frac{C}{k}$, being C a positive constant, but still, we may also consider the sequence $\mu_k = C\eta^k$, being η a positive scalar between 0 and 1, ideally closer to 1 in order to make this sequence more compliant with the first condition.

What is more, we could use the optimal step size choice, being the line search family of algorithms a set of methods that estimate the best sequence of μ_k, by means of maximizing the step size such that the objective function has a maximum decrement at each iteration. In particular, for this algorithm we found that the Backtracing Line Search algorithm is the one suitable for our problem, having an initial guess of the step size being in the order of 10^2. The line search family of algorithms consists on solving the optimization problem as a function of the step size, where mathematically it would be a problem of the following form [29]:

$$\arg\min_{\mu_k} h(\mu_k) = \arg\min_{\mu_k} f(\mathbf{x} - \mu_k \mathbf{d}), \tag{48}$$

where $\mathbf{x}$ is the current estimate and $\mathbf{d}$ is the direction of maximum increment.

4.2.2 Initializations

In this algorithm, we will need to initialize essentially two variables: $\boldsymbol{\theta}_0$ and α_0. As the value of α is tightly related to the amount of sparsity in the solution of $\boldsymbol{\theta}$, we would need to initialize both variables such that they are mutually consistent. In this way, the easiest and more reasonable way to initialize these two variables is to assume the maximum amount of sparsity, being all entries of $\boldsymbol{\theta}$ equal to 0, and initialize them acordingly. These proposals of initialization are the following:

$$\boldsymbol{\theta}_0 = \mathbf{0}, \tag{49}$$

$$\alpha_0 \to 0, \tag{50}$$

where α_0 should be a value arbitrary small, but not equal to zero because it would produce numerical issues in equation (45). In this way the algorithm is forced to increase the total amount of sparisty until it reaches the optimal value such that the fitting constraint is fulfilled.

4.2.3 Stopping rule

In order to have an efficient use of computational resources, it is a common practice to define a stopping rule where the algorithm stops to offer its optimal solution. In our proposed algorithm, there are several possible criteria to use, and each one of the have its own meaning, which depending on the application, it could be useful to choose one or another. We will denote by s_k the stopping parameter at the k-th iteration.

Fitting level We may use the constraint term to do an early stop of our algorithm. This has much sense in the application of estimating the latent variable, as it stops when the total amount of fitting converges up to a certain point. The stopping term is the following:

$$s_k = \frac{(\mathbf{y} - \boldsymbol{\theta}_k - A_k\mathbf{1})^T\mathbf{K}^{-1}(\mathbf{y} - \boldsymbol{\theta}_k - A_k\mathbf{1})}{\mathbf{1}^T\mathbf{K}^{-1}\mathbf{1}} - \gamma. \tag{51}$$

Note that the fitting constraint, γ, may not be necessary in the sense of a stopping criteria, but it gives more interpretation to the solution.

Sparsity level It may also be useful to stop when the total amount of sparsity gets to a stationary solution. This has sense in the applications where we are much more interested in the anomaly detection, for instance if we have the step's level, J_o, as a prior:

$$s_k = ||\boldsymbol{\theta}_k||_1. \tag{52}$$

Dual optimality This criteria may be the least informative, from the application perspective, but still it is a balanced choice between the previous two examples:

$$s_k = \alpha_k. \tag{53}$$

Finally, after defining the stopping rules criteria, we have to define the condition where the algorithm must stop. It is an appropriate way to handle this issue by using the following quotient to determine the stopping point:

$$\delta_{k+1} = |\frac{s_{k+1} - s_k}{s_k}|. \tag{54}$$

Having the recursive expression in (54) as an indicator, one may note that this is a measure of the relative change in the $k+1$ iteration. The algorithm may be stopped if the following condition is fulfilled:

$$\delta_{k+1} < \epsilon, \tag{55}$$

where this ϵ must be any small positive scalar. In general, this using any kind of stopping criteria leads to an efficient use of the computing resources, so the algorithm does not spent additional futile iterations. One usually defines the maximum amount of iterations to be a high number, to ensure convergence in the case when needed. However, with these stopping criteria the algorithm shows to be quite fast in terms of convergence, as we will evidence in the simulations section.

5 Simulation results

We will divide this section into two sets of simulations, being the first block of the simulations that are necessary to understand the algorithm and the second block are the tests that show the overall performance of the latent variales estimators derived from this algorithm in comparison with other simpler latent variables' estimators.

5.1 Understanding the dual ascent algorithm

This subsection is devoted to understand each of the parameters related to our proposed algorithm. In this subsection, we will be showing a quite favorable scenario, understood by the combination of J_o and p_a such that the features of this algorithm are highlighted, in order to show the evolution of some parameters of this algorithm along the iterations and discuss the meaning that they contribute. We will need to define γ numerically in a way that it is more interpretable. This new definition is the following:

$$\gamma = \rho \frac{(\mathbf{y} - \boldsymbol{\theta}_0 - A_0\mathbf{1})^T\mathbf{K}^{-1}(\mathbf{y} - \boldsymbol{\theta}_0 - A_0\mathbf{1})}{\mathbf{1}^T\mathbf{K}^{-1}\mathbf{1}}, \tag{56}$$

where ρ is a positive scalar between 0 and 1, and will denote the amount of improvement that we expect from the algorithm. This improvement is measured in how the algorithm must adapt the solution, in order to reduce the fitting level that we get without taking into consideration the anomalies. Taking this definition into consideration, we must define the initialization of the latent variables as:

$$A_0 = \frac{\mathbf{1}^T\mathbf{K}^{-1}(\mathbf{y} - \boldsymbol{\theta}_0)}{\mathbf{1}^T\mathbf{K}^{-1}\mathbf{1}}, \tag{57}$$

and $\boldsymbol{\theta}_0 = \mathbf{0}$ as initially it must not consider any kind of anomaly, which motivates the initialization that we have proposed in the previous section.

Now, we must differentiate two scenarios of operation in terms of the only free parameter that we are using, being γ or, equivalently, ρ. This two scenarios are related to the capability of our algorithm to fulfill the fitting constraint. In this way, we are conscious that there are values of ρ that make this problem strictly unfeasible and values of ρ making the initialization unfeasible until convergence. Even though, there are some unfeasible values of ρ, the algorithm still offers a reasonable solution to the anomaly estimation. Last but not least, we will not be using the stopping rules that we have defined previously in these simulations, because they possibly hide some properties of this parameters if stopped earlier, which is the main objective of this simulations. The global simulation parameters, used in all the simulations in this subsections, are the ones stated in the following table:

Parameter	Value
J_o	0.75
$\mathbf{K}$	$0.02 \begin{bmatrix} 1 & -0.5 & 0 & ... \\ -0.5 & 1 & -0.5 & 0 \\ 0 & -0.5 & 1 & ... \\ ... & 0 & ... & ... \end{bmatrix}$
M	50
Max. amount of iterations	5000
p_a	0.1
Total number of tests	8

5.1.1 Strict unfeasibility

In this setting, we need to take a values of ρ small enough to force that the primal constraint is strictly unfeasible. They should be low because we are asking the algorithm to fit the data too tightly, so it is impossible to achieve with the current data. In this case, we have set $\rho = 0.0005$ to ensure that the problem is primal unfeasible. Consequently, in Figure 10 we show the evolution of the constraint value, which consists on the following function:

$$c_k = \frac{(\mathbf{y} - \boldsymbol{\theta}_k - A_k\mathbf{1})^T\mathbf{K}^{-1}(\mathbf{y} - \boldsymbol{\theta}_k - A_k\mathbf{1})}{\mathbf{1}^T\mathbf{K}^{-1}\mathbf{1}} - \gamma,$$

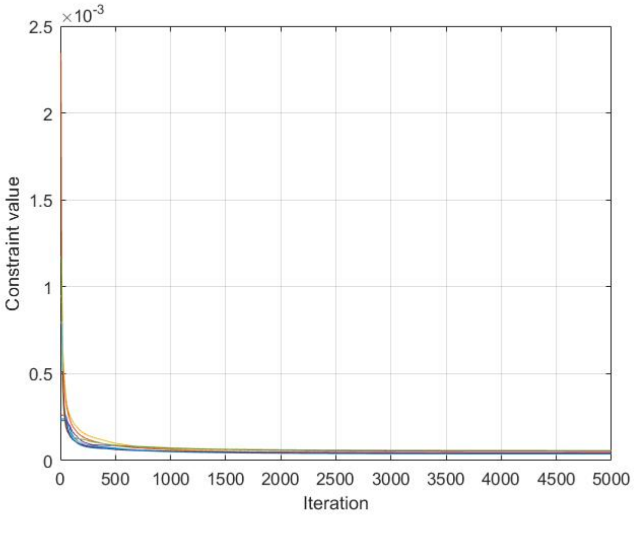
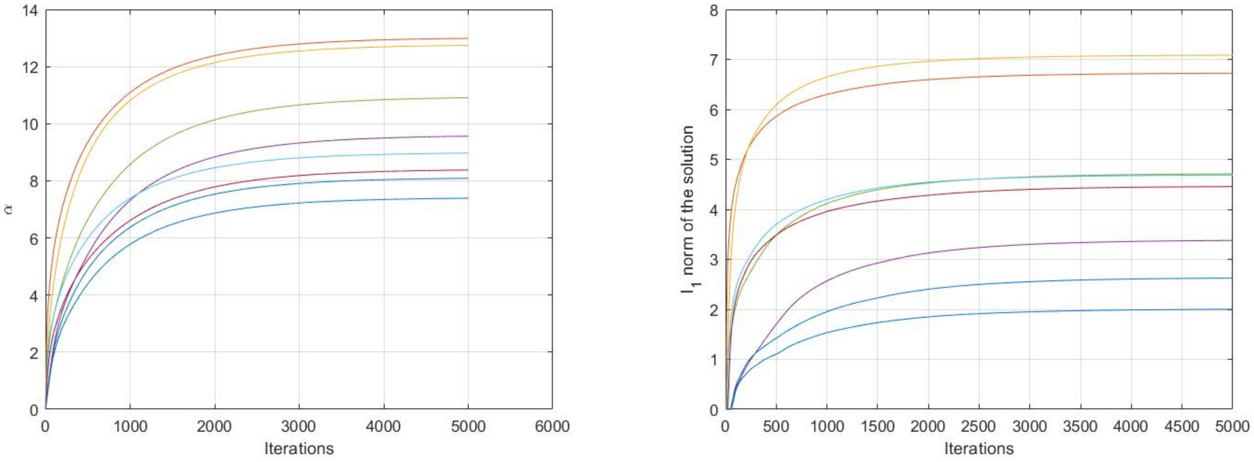


Figure 10: Strict unfeasibility. 8 realizations of the Constraint value c_k, along the iterations.

It is known that this constraint value will be a relatively small value due to the normalization that we have proposed in the constraint, but the important remark in Figure 8 is that it is always positive and do not have values closer to zero, meaning that $\frac{(\mathbf{y}-\boldsymbol{\theta}_k-A_k\mathbf{1})^T\mathbf{K}^{-1}(\mathbf{y}-\boldsymbol{\theta}_k-A_k\mathbf{1})}{\mathbf{1}^T\mathbf{K}^{-1}\mathbf{1}} > \gamma$ in all the iterations. In this figure, it is also shown that the algorithm has converged to a stationary solution and still it was unable to fulfill the required fitting constraint because this function does not reach zero. In Figure 11 we show the evolution of the amount of sparsity of the solution by showing the two parameters related to the sparsity of the solution, namely λ and $||\boldsymbol{\theta}_k||_1$.



(a) Dual variable evolution along the iterations.

(b) l_1 norm evolution along the iterations.

Figure 11: Strict unfeasibility. 8 realizations of Sparsity indicators.

Note in Figure 11 that there is a tight relationship between λ and $||\boldsymbol{\theta}_k||_1$, as the sparsity is related to the soft threshold operator, governed by λ. We also remark that in this case, in each iteration, the solution loses sparsity (increases the l_1 norm) in order to increase the amount of fitting of our solution to the signal, which never reaches a maximum sparsity level because the constraint does not get to zero. In addition, the final value of convergence also depends on the amount of jumps, governed by the probability p_a because more jumps means less sparsity in the optimal solution.

Finally, in Figure 12 we show 8 realizations of the real anomaly with the convex solution and detected jumps, by applying an intuitive threshold over the convex solution. In these figures, we show that even if the problem is not feasible, the algorithm offers a reasonable solution to the anomaly estimation and detection.

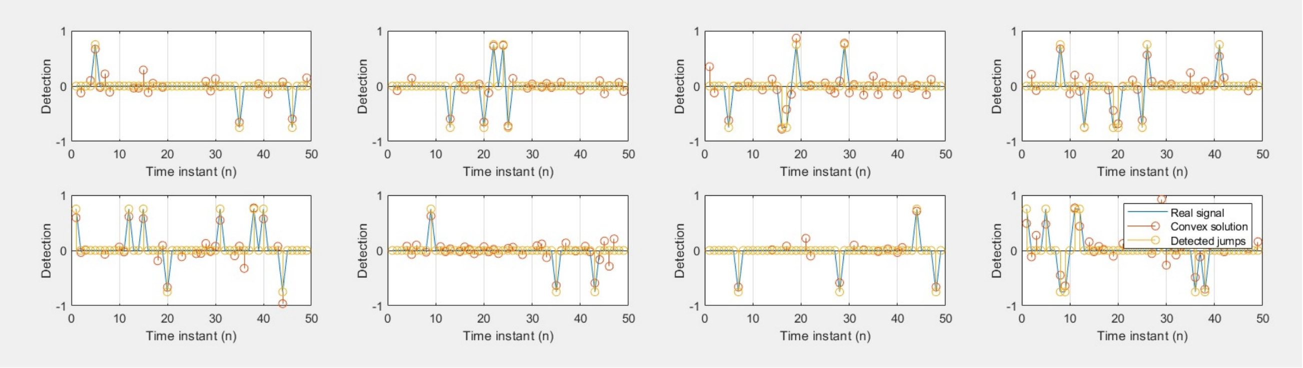


Figure 12: Strict unfeasibility: 8 realizations of anomaly detection.

5.1.2 Initially unfeasible to feasible convergence

In contrast with the previous setting, we need to set ρ small enough to force that in the initialization the constraint takes positive values, being the problem initially unfeasible, but large enough so the optimal solution of the problem is feasible. We have chosen $\rho = 0.3$ to ensure these conditions. In Figure 13, we show again the evolution of c_k

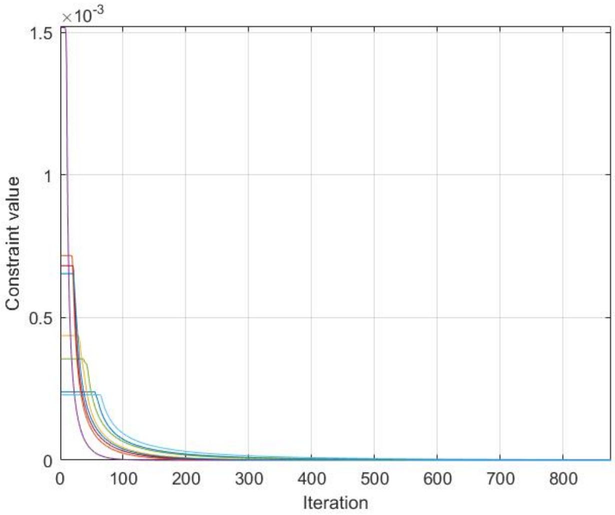
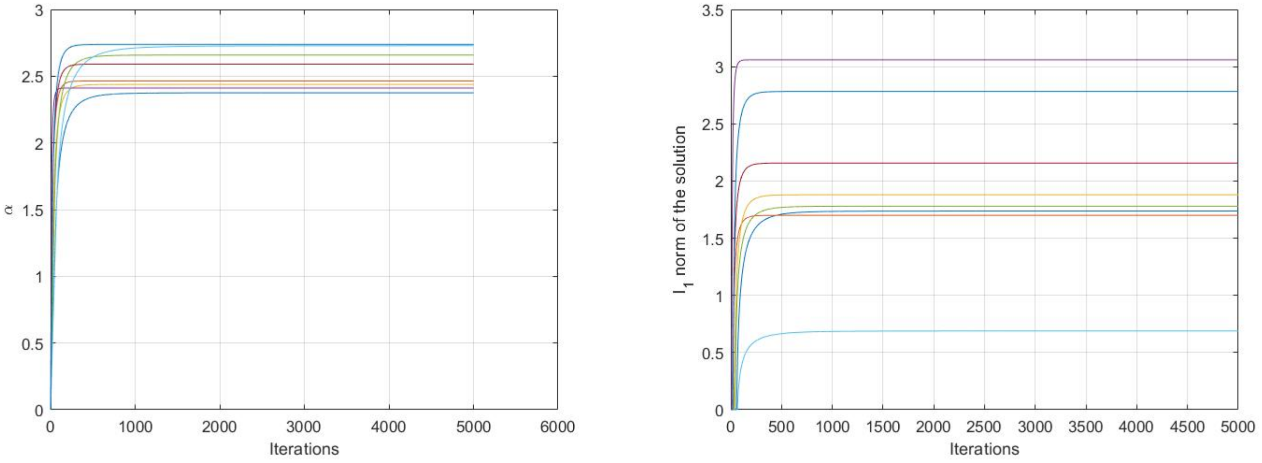


Figure 13: Feasible convergence. Constraint value, c_k, along the iterations.

Again, the constraint takes small values, which *per se* do not give much more insights. However, the sign that the constraint value takes at each iteration is of great interest to see the subgradient effect. In Figure 11, we can evidence that there is a contrast with the previous case. The behaviour is quite similar, but in this case the algorithm is capable to fulfil the primal feasibility constraint, as it can be seen from the fact that it is capable to reache zero. As an important remark, this figure shows that the algorithm is looking for the best sequence of anomalies such that it fits the best into the signal and it stops when it has achieved it. Also, as we will see in a subsequent figure, Figure 11 will be helpful to get insights about the relationship between the constraint value and the subgradient of α_k, and equivalently the subgradient of $||\boldsymbol{\theta}_k||$. In Figure 14, as in the previous case, it is shown the evolution of the amount of sparsity of the solution, where we will find a direct relation with the behaviour of the constraint value.



(a) Dual variable evolution along the iterations.

(b) l_1 norm evolution along the iterations.

Figure 14: Feasible convergence. 8 realizations of Sparsity indicators.

Now, we see clearly in Figure 14 that the constraint value corresponds to the amount of increment in the sparsity indicators by looking also at Figure 13. This scenario makes more evident the fact that the constraint value is indeed a subderivative of the dual problem.

Finally, in Figure 15 we again show 8 realizations of the detected signals the detected signals. It can be seen in this case that the solution is still capable of detecting the anomalies, by applying a lower threshold for instance, but an anomaly is not that evident in the final solution, as compared to the previous scenario.

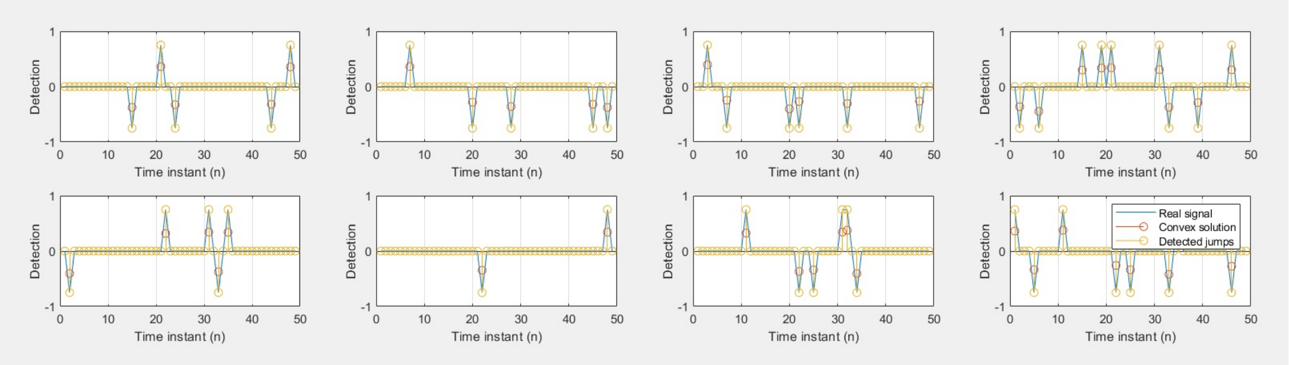


Figure 15: Strict unfeasibility: 8 realizations of anomaly detection.

5.2 Performance measures

This subsection will be focused on the numerical characterization of the Dual Ascent for Anomaly Detection's algorithm. First, we will need to define some estimators that we will be making use to compare possible uses of this algorithm with some other known estimators for this problem:

- The Cramer-Rao Bound (CRB) solution, which in this case equivalent to Maximum Likelihood of the signal without any consideration of the anomalies applied upon the clean signal without anomalies. It is also the Minimum Variance Unbiased Estimator of this problem, so it achieves the Cramer-Rao

Lower Bound. Its expression is the following:

$$\hat{A} = \frac{\mathbf{1}^T \mathbf{C}^{-1} \mathbf{y}_{clean}}{\mathbf{1}^T \mathbf{C}^{-1} \mathbf{1}}. \tag{58}$$

where in this case, its variance closed form and can be computed as:

$$\sigma^2_{CRB} = \frac{1}{\mathbf{1}^T \mathbf{C}^{-1} \mathbf{1}}. \tag{59}$$

- The Convex Solution estimator is what we obtain directly from the algorithm.
- The Maximum Likelihood over the signal, applied over the contaminated signal:

$$\hat{A} = \frac{\mathbf{1}^T \mathbf{C}^{-1} \mathbf{y}}{\mathbf{1}^T \mathbf{C}^{-1} \mathbf{1}}. \tag{60}$$

- The sparse aware is the one we obtain by correcting the signal with a hard decision over the anomaly solution, and having as a prior knowledge the value of J_0. The expression is the following:

$$\hat{A} = \frac{\mathbf{1}^T \mathbf{C}^{-1} \left(\mathbf{y} - J_o \mathbf{1}_{\left(\hat{\boldsymbol{\theta}} > \varrho * J_o\right)} \right)}{\mathbf{1}^T \mathbf{C}^{-1} \mathbf{1}}, \tag{61}$$

  where $\mathbf{1}_{(a)}$ is the entry-wise indicator function that returns in the i-th entry 1 when the argument in this entry is true and 0 otherwise, and ϱ is the minimum fraction of the anomaly's level in order to detect a jump, which mainly depends on the constraint value ρ in the following way:

$$\varrho = \frac{10^{-5}}{\rho}. \tag{62}$$

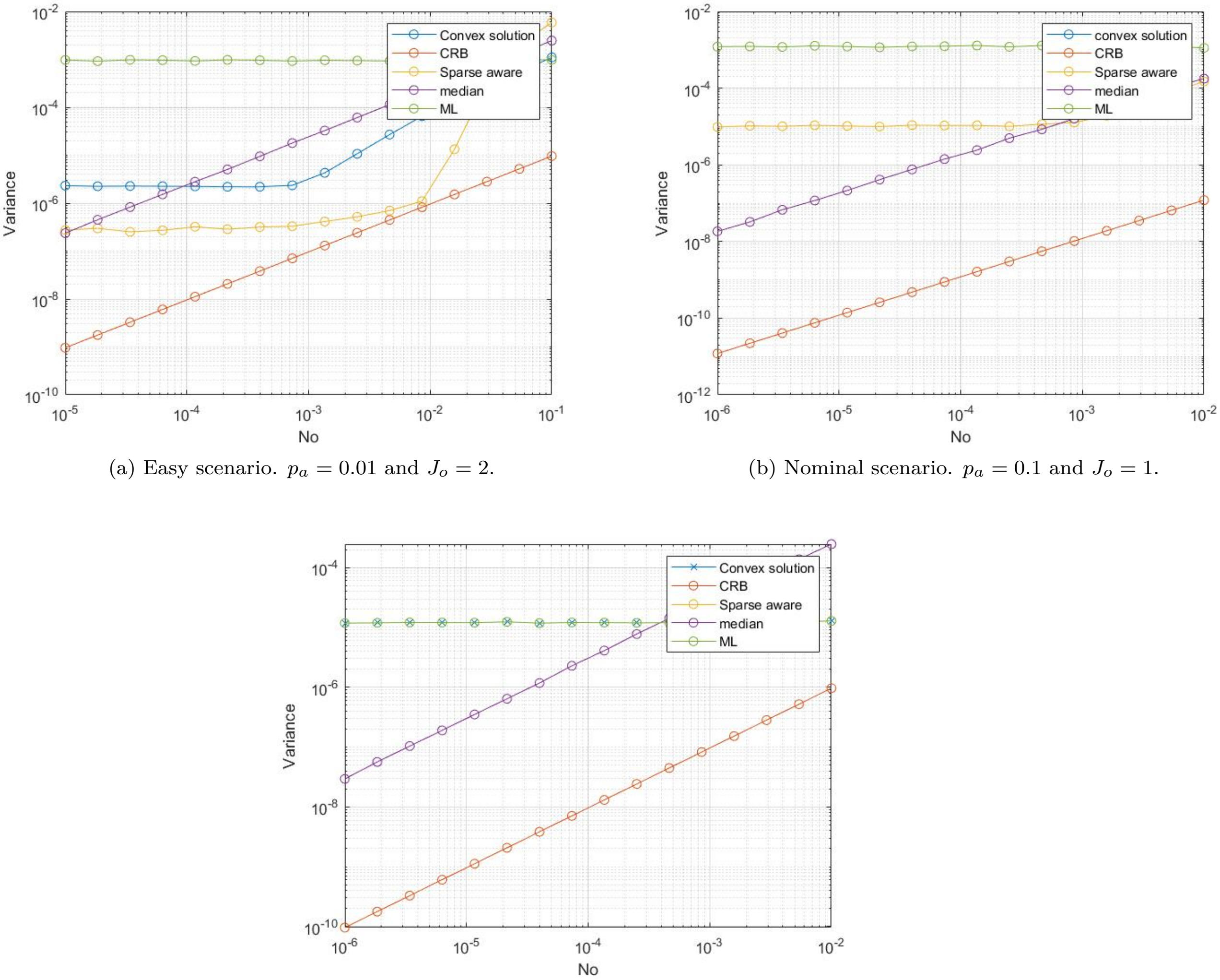
  Therefore ρ must never be smaller than 10^{-5}. The rationale for this value is that whenever it is more costly for the algorithm to achieve the constraint requirement, i.e ρ is smaller, then the output solution of the algorithm is less sparse, so it must account for it as it may be more entries that could be mistaken by noise. The same reasoning can be done for the other way round. Furthermore, one could also go to Figures 15 and 12 to get more insights for this reasoning.

- The median of the input signal, $\hat{A} = \text{median}(\mathbf{y})$.

Then, we would also need to define three different scenarios that consists on combinations of p_a and J_o such that we can see the performance of the Dual Ascent for Anomaly Detection, in terms of the variance of the above estimators. These tests will try to emulate difficult, easy and nominal situations where this algorithm can be applied. The general simulations' parameters that have been used in this test are the ones summed up in the following table

Parameter	Value
$\mathbf{K}$	$0.02 \begin{bmatrix} 1 & -0.5 & 0 & ... \\ -0.5 & 1 & -0.5 & 0 \\ 0 & -0.5 & 1 & ... \\ ... & 0 & ... & ... \end{bmatrix}$
M	50
Max. amount of iterations	5000
MonteCarlo iterations	10000

In Figure 16, we can see all three scenarios with their corresponding set of variables, p_a and J_o, where we are looking how does the variance change as a function of the pre-differentiated noise power, N_o.



(a) Easy scenario. $p_a = 0.01$ and $J_o = 2$.

(b) Nominal scenario. $p_a = 0.1$ and $J_o = 1$.

(c) Hard scenario. $p_a = 0.05$ and $J_o = 0.1$.

Figure 16: Measures of Variance as a function of N_o, for the three scenarios.

Firstly, what we can see from the above figure is that our definitions of hard, nominal and easy are targeting the problem of the anomaly detection. In this sense, we are evaluating each set of the two anomaly parameters as how easy they would be for an algorithm so that they are detected. In this way, we can see that huge anomalies are quite easy to be detected and less frequent, but smaller anomalies are much harder. This figure also shows the fact that the median estimator is quite robust to the anomalies, as one may expect from the definition of median that it would be quite robust to outliers. In contrast, the variance of the ML estimator is totally controlled by the anomalies' statistics for these levels of noise.

As for the Dual Ascent for Anomaly detection estimators, being the convex solution and the sparse-aware, we can see that they result in an improvement from the ML estimator in nominal and easy scenarios, even being better than the median estimator for the easier simulation, while giving the same value in a hard combination of anomaly statistics. The main difference in performance between the convex solution and the sparse-aware solution is given by the fact that whenever the anomaly is correctly detected, having the knowledge of the anomaly's level, J_o, must result in an improvement in the latent variable's estimator's variance, as the convex solution does not offer the exact solution of the anomaly's level, but an approximation. Also in the hard scenario, the performance of the novel approach is equivalent to the ML estimator as in this scenario the anomaly's level, J_o, is so low that it is often confused with the noise.

Now, we will use the nominal set of anomaly's statistics to test the impact of the only free parameter of our algorithm, the constraint restriction γ or equivalently ρ. In Figure 17, we show how the variance is affected by this value.

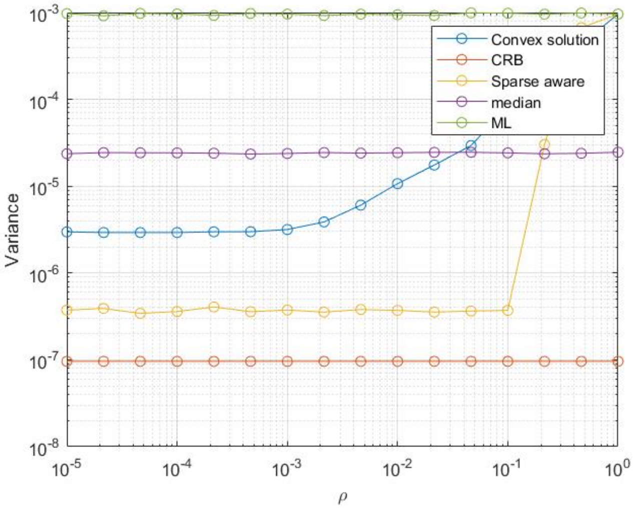


Figure 17: Performance measures. Impact of ρ in the variance.

We can see in this figure that for low values of ρ, the performance of the estimators are quite noticeable as the Dual Ascent-based estimators are capable to outperform the median estimator. This is due to the fact that at lower values, the algorithm is looking for the best anomaly sequence that best fits the signal, but it never reaches the desired value of fitting, so the algorithm gives a good estimation of the anomaly. However, as ρ increases, it is much easier for the algorithm to achieve the desired fitting level and therefore it tends to increase the amount of sparsity (the zero entries) of the solution, until it reaches the same performance of the ML solution where it stops.

To conclude, we also tested which is the minimum value of the anomaly's level, J_o, that our algorithm is capable to handle. This can be seen in Figure 18, where we used the nominal scenario with the noise power being, $N_o = 0.01$.

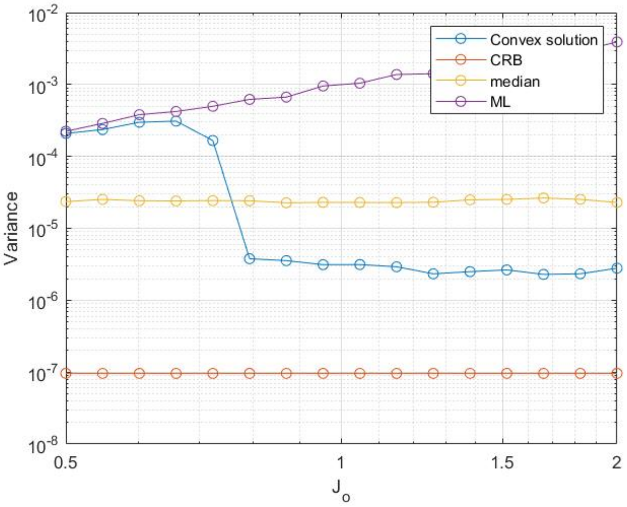


Figure 18: Performance measures. Impact of J_o in the variance.

In this last figure, it is evidenced the fact that there is a minimum value of J_o such that it is capable of consistently detect the anomalies. The Dual Ascent approach always gives the ML solution, unless there is an anomaly. In this sense, we can say that this approach is robust and also maintains all the asymptotic properties of the ML solution. Also, and more subtly, in this last figure we can see that J_o has a lower impact in the performance of the aforementioned estimators than the probability of having an anomaly p_a.

Last but not least, the remaining figure shows the impact of the anomaly probability, p_a, in the previous estimators.

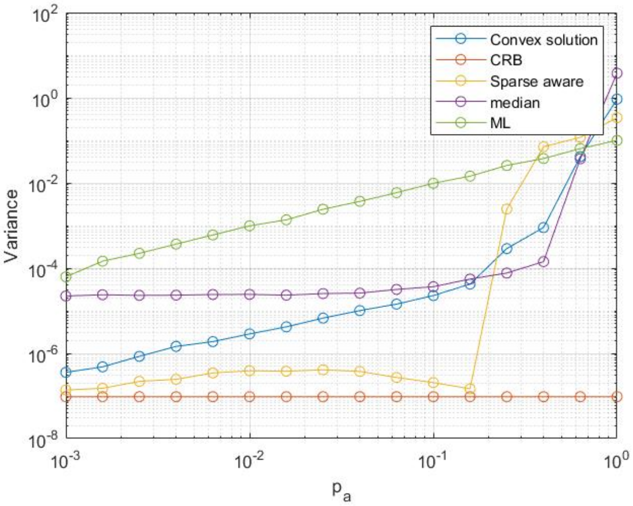


Figure 19: Performance measures. Impact of p_a in the variance.

Here, the plot shows that p_a, or equivalently the amount of sparsity S_0, has a huge impact in the estimators performance, even greater than the anomaly's level J_o. The ML estimator still suffers at lower values of p_a from the consequences of losing the autocorrelation profile, but at lower values of p_a the Dual Ascent for Anomaly detection manages to get extremely close to the Cramer Rao Bound. Evidently, if p_a has a high value, the problem may become intractable as seen in this figure, where all the estimators suffer from having too many anomalies to handle.

Part II
Subspace Learning

Until now, we have solved the Anomaly Detection problem, being one of the central point of this thesis. However, in order to achieve and develop a Subspace-based Anomaly Detection algorithm, we must now solve the remaining problem of subspace identification. This part will be focused only on the Subspace Learning problem. It can be defined in a general way as the optimization of an arbitrary cost function with orthonormality restrictions:

$$\arg\min_{\mathbf{H}(n)} f\left(\mathbf{H}(n)\right) \quad \text{s.t. } \mathbf{H}(n)^T\mathbf{H}(n) = \mathbf{I},$$

where the index n accounts for the possibility of having a non-stationary environment. We will also assume a sparsity prior in terms of the dimensionality of $\mathbf{H}(n)$. Therefore we will assume that it is a $M \times D$ matrix with $D < M$. On the other hand, the orthonormality contraint on $\mathbf{H}(n)$ forces this matrix to belong to the set of all the orthonormal matrices embedded in the M-dimensional space. However, the general Subspace Learning framework adds an homogeneity assumption over the function $f(\cdot)$, being such assumption:

$$f\left(\mathbf{H}(n)\right) = f\left(\mathbf{H}(n)\mathbf{R}\right),$$

where $\mathbf{R}$ is any orthonormal $D \times D$ matrix. In this sense, by this assumption, the optimal solution of this function is not unique in general, as from those two restrictions we can see that the solution for this problem is the subspace that minimizes the cost function $f(\mathbf{H}(n))$. To this extend, in this part we will show three state-of-the-art solutions for the subspace identification, being:

- Data Projection Method (DPM), the LMS-like algorithm for subspace identification
- Projection Aproximation Subspace Tracking (PAST) algorithm, the RLS-like algorithm for subspace tracking
- Grassmann Rank One Update for Subspace Estimation (GROUSE) alrogithm, the Grassmann-based subspace tracking algorithm.

In addition, we will also show two novel approaches that have been developed originally in this thesis, which are the following ones:

- Affine Projection for Subspace Tracking (APST) algorithm, the Affine Projection version of subspace identification which at the same time is a future publication related to this work.
- Weighted Least Squares-based Manifold Conjugate Gradient Descent (WLS-MCGD), which is an improvement of the GROUSE algorithm for correlated noise.

6 Mathematical tools: Grassmann Manifold

To develop and understand the algorithms and procedures in this chapter, we will need to introduce some concepts related to the Grassmann Manifold.

6.1 Grassmann Manifold

The Grassmann Manifold, often represented by the mathematical notation $Gr(M, D)$ with $M \geq D > 0$, can be interpreted as the space of D-dimensional linear subspaces embedded in an M-dimensional space. This will be used in the development of this chapter, as we will be interested in identifying a given optimal point that fulfills both orthonormality constraints and the homogeneity assumption, inherent to the $Gr(M, D)$ Grassmann Manifold. As the reader may have noticed, there is not a unique matricial representation for an arbitrary point in the Grassmanian $Gr(M, D)$. In order to illustrate this, we will write mathematically the Grassmanian $Gr(M, D)$ in the following way:

$$Gr(M, D) = \left\{ \text{span}(\mathbf{X}) : \mathbf{X} \in \mathbb{R}^{M \times D}, \mathbf{X}^T \mathbf{X} = \mathbf{I}_D \right\}, \tag{63}$$

where we can note that there is not a unique orthonormal matrix $\mathbf{X}$ such that it represents a given subspace, and we will show this fact in a simple way. Let $\mathbf{X}$ be an arbitrary matrix that represents a point in the Grassmannian $Gr(M, D)$, then any matrix that can be obtained by:

$$\mathbf{X}' = \mathbf{X}\mathbf{R}, \tag{64}$$

is a matrix that still represents this arbitrary point in the Grassmanian if the matrix $\mathbf{R}$ is an orthogonal matrix, because the columns of matrix $\mathbf{X}'$ are orthonormal linear combinations of the columns $\mathbf{X}$ which still fulfills the orthonormality constraint, $\mathbf{X}'^T \mathbf{X}' = \mathbf{I}_D$. In other words, the Grassmann manifold, $Gr(M, D)$, is the space that contain all the D-dimensional subspaces embedded in a M-dimensional space. As matrix $\mathbf{R}$ is an orthonormal, it implies that it must be a $D \times D$ matrix such that $\mathbf{R}^T \mathbf{R} = \mathbf{I}_D$ and $\mathbf{R}\mathbf{R}^T = \mathbf{I}_D$.

However, from this subspace spanning matrices we can define another matrix derived from these Grassmann manifold point representers, eliminating this rotation ambiguity in the sense that it is a unique representative of a single point in the Grassmanian. This derived matrix will be called the projector matrix and it will be defined, on the Grassmanian, given an arbitrary matrix $\mathbf{X}$ as:

$$\mathbf{P}_X = \mathbf{X}\mathbf{X}^T, \tag{65}$$

where $\mathbf{P}_X$ is a $M \times M$ matrix. Let's remark that, in general, a projection matrix is defined as $\mathbf{P}_A \triangleq \mathbf{A}(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T$ for an arbitrary matrix $\mathbf{A}$, but considering the orthonormality constraint $\mathbf{X}^T \mathbf{X} = \mathbf{I}_D$, we get to the equality in (65). In the Figure 20, we will show a particular case of a Grassmann manifold, where we can intuitively show the concepts that are about to come.

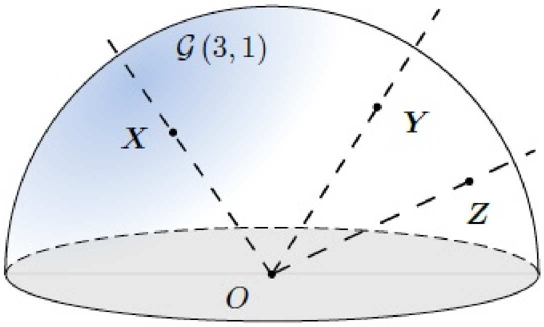


Figure 20: Example of the Grassmann manifold Gr(3,1). $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}$ are arbitrary elements of this manifold. Figure extracted from [9].

In the above figure, we can see that in $Gr(3, 1)$ an arbitrary element would be a unit normed vector. In this set the tangent space in each point would be the tangent plane to each of the three points and the geodesic are defined as the shortest curve that links two different elements. Note that the geodesic will not the straight line that links two elements, but the path along the sphere that connect those mentioned elements. The concept of tangent space is necessary to derive the definition of the Grassmann gradient and the geodesics are necessary to follow a path inside the Grassmaniann, also a necessary concept for the Manifold Conjugate Gradient Descent which follows a curve along the Grassmann manifold in the Grassmann gradient direction.

6.1.1 Distances in the Grassmann Manifold

In order to find the distances between two subspaces in the Grassmanian, spanned by two elements of the Grassman manifold $\mathbf{A}, \mathbf{B} \in Gr(M, D)$, there is a useful measure between two subspaces which is the principal angles between those subspaces. These angles are defined recursively as [9]:

$$\begin{cases} \cos(\theta_i) = \max\limits_{\substack{\mathbf{a} \in \mathbf{A} \\ \mathbf{b} \in \mathbf{B}}} \quad \mathbf{a}^T\mathbf{b} \\ \mathbf{a}_i^T\mathbf{a}_i = 1, \quad \mathbf{b}_i^T\mathbf{b}_i = 1 \\ \mathbf{a}_i^T\mathbf{a}_j = 0, \quad \mathbf{b}_i^T\mathbf{b}_j = 0 \qquad \forall j < i \end{cases},$$

where as seen in the above expression there is implicitly the orthonormality constraints of the subspace spanning matrices $\mathbf{A}$ and $\mathbf{B}$. In a numerical way, they can be computed by deriving the Singular Value Decomposition (SVD) of the matrix $\mathbf{A}^T\mathbf{B}$, where the singular values of this product matrix are the cosine of these principal angles denoted by θ_i for $i = 1, ..., D$ or by the vector $\boldsymbol{\theta}$. Intuitively, they can be seen as the minimal angles between two linear subspaces spanned by their given orthonormal bases. For instance, in the case of $D = 1$ that for two unit norm vectors, $\mathbf{x}$ and $\mathbf{y}$, we have:

$$\cos(\theta_1) = \mathbf{x}^T\mathbf{y}, \tag{66}$$

so this is a generalization of this simpler concept. With this concept in mind we can use some of the many metrics in the Grassmann manifold to define distances between two subspaces, which will be a function of these principal angles as it will be shown in the following paragraphs [9]. In the development of this work, we have considered only two subspace distances on the Grassmann manifold to test the algorithms and methods: the chordal distance and the subspace angle between two subspaces.

Chordal distance The Chordal distance, or also Projection distance, is a useful metric to compare how different two subspaces are. Given the previous two matrices, $\mathbf{A}$ and $\mathbf{B}$, whose columns span two distinct subspaces, being their respective projection matrices denoted by $\mathbf{P}_A$ and $\mathbf{P}_B$ respectively. Then, the Chordal distance between those subspaces is defined as:

$$d_C(\mathbf{A}, \mathbf{B}) \triangleq ||\mathbf{P}_A - \mathbf{P}_B||_F. \tag{67}$$

As the projection matrices are invariant to rotations in the subspace, this metric show perfectly the dissimilarity between two subspaces. This metric can also be expressed as a function of the principal angles between $\mathbf{A}$ and $\mathbf{B}$, and it has the following form:

$$d_C(\mathbf{A}, \mathbf{B}) = \left(\sum_{i=1}^{D} \sin^2(\theta_i)\right)^{-1/2}. \tag{68}$$

However, we will use a modification this norm as a performance measure, for simplicity. The normalized chordal error proposed is:

$$e = \frac{1}{2D} \left\|\hat{\mathbf{P}}_X - \mathbf{P}_X\right\|_F^2, \tag{69}$$

where $\hat{\mathbf{P}}_X$ is the estimated projector matrix under the prior knowledge of the dimension D, and $\|.\|_F$ is the Frobenius norm. Note that the performance measure has a rationale for this specific normalization. This is because we have:

$$0 \leq e \leq 1,$$

where the left-hand equality is given when the estimated subspace is equal to the true subspace, that is $\hat{\mathbf{P}}_X = \mathbf{P}_X$, and the right-hand equality is given when the estimated subspace is orthogonal to the true subspace. Effectively when this is the case we have $\hat{\mathbf{P}}_X\mathbf{P}_X = \mathbf{0}$, implying that $\left\|\hat{\mathbf{P}}_X - \mathbf{P}_X\right\|_F^2 = \left\|\hat{\mathbf{P}}_X\right\|_F^2 + \|\mathbf{P}_X\|_F^2$. As the Frobenius norm of a projector is equal to the subspace dimension, we have $\left\|\hat{\mathbf{P}}_X - \mathbf{P}_X\right\|_F^2 = 2D$, confirming that $e = 1$ in that case. Finally, we remark that the performance measure in (69) is not a norm.

Subspace angle The subspace angle is a simpler way to compute the dissimilarity between two subspaces. It is derived by computing all the principal angles between two subspaces and then keep the greatest angle among them. Note that the maximum value of this metris is $\frac{\pi}{2}$, showing total linear independence between the pair of subspaces. In terms of the SVD of the matrix $\mathbf{A}^T\mathbf{B}$, it corresponds to the arc cosine of the smallest singular value. Mathematically, we can define it as:

$$d_A(\mathbf{A}, \mathbf{B}) \triangleq \min(\theta_i) \quad \forall i. \tag{70}$$

In Figure 21, we will show a geodesic in an arbitrary manifold to get the intuition behind this distances.

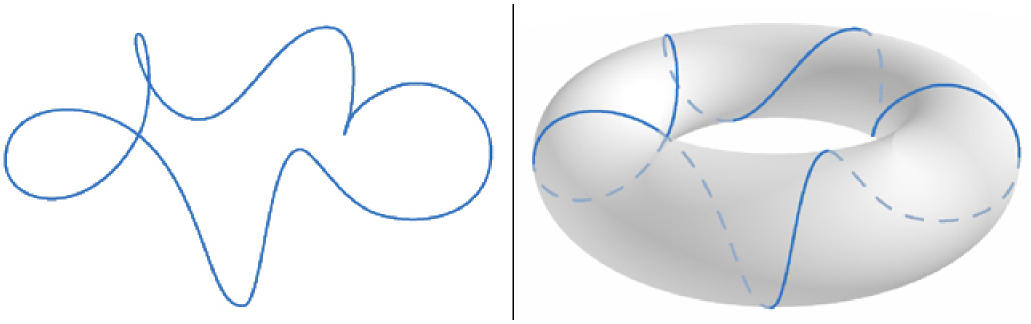


Figure 21: Geodesic and distances in an arbitrary manifold. Based on an image created by Mark Irons.

In the above figure we show an arbitrary manifold, being the surface of a thorus and the corresponding geodesic in this manifold. In this way, the geodesic is described by following a straight line in the manifold and the distances defined in that manifold are different definitions to compute the lenght of this straight line between two arbitrary points.

6.1.2 Gradients in the Grassmann Manifold

To understand the Grassmann-based algorithms in this chapter, it is necessary the introduction of the definition of the Gradient in the Grassmann Manifold and how do we follow a straight line, also called geodesic curve, in the Grassmanian. We will start with the concept of Tangent space, which is a key concept in the definition of a gradient in the Grassmann Manifold. A Tangent Space in the Grasmannian $Gr(M, D)$, $\mathbf{\Delta}$, at a given point in the Grassmanian, $\mathbf{X}$, is the set of all vectors, spanning a given subspace in the $\mathbb{R}^M$ space, such that they satisfy the following condition:

$$\mathbf{X}^T\mathbf{\Delta} = \mathbf{0}, \tag{71}$$

where $\mathbf{0}$ is a matrix filled with 0 along all its entries. An intuitive representation of this concept of Tangent space can be seen in Figure 22.

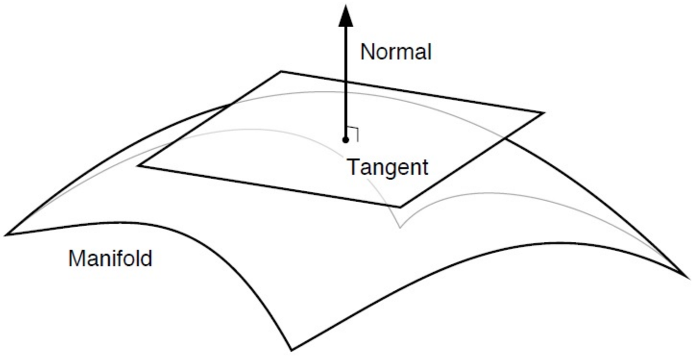


Figure 22: The tangent and normal spaces of an embedded or constraint manifold. Figure extracted from [8].

Note that in Figure 22 there is also depicted the normal subspace, defined in the Grassmann manifold [8]. Nevertheless, we will not make use of this concept in this work. Now, as we have already defined the concept of tangent space in the Grassmannian, we will now need to define the gradient of a function in the Grassmann manifold. If we denote by $\nabla_{\mathbf{X}} f(\mathbf{X})$ the "Euclidean Gradient" of a function $f(\mathbf{X})$, being the "Euclidean Gradient" the matrix that we get from the general gradient rules for matrices. Then, the gradient in the Grassmanian, $Gr(M, D)$, is defined as:

$$\nabla_{\mathbf{X}} F_{Gr(M,D)}(\mathbf{X}) = \mathbf{P}_X^{\perp} \nabla_{\mathbf{X}} f(\mathbf{X}) = (\mathbf{I}_M - \mathbf{X}\mathbf{X}^T) \nabla_{\mathbf{X}} f(\mathbf{X}), \tag{72}$$

which is equivalent as projecting the "Euclidean gradient" onto the orthogonal complement of $\mathbf{X}$, being the tangent direction in this point of the manifold. Using this kind of gradient in a Gradient Descent-like approach could also be interpreted as a particular case of a Projected Gradient Descent approach, where in this case the gradients are continuously projected onto the tangent direction of the set containing all the elements that satisfy the orthonormality constraints and have met the homogeinity assumption. Here in Figure 23, we show some insights of the Projected Gradient Descent approach.

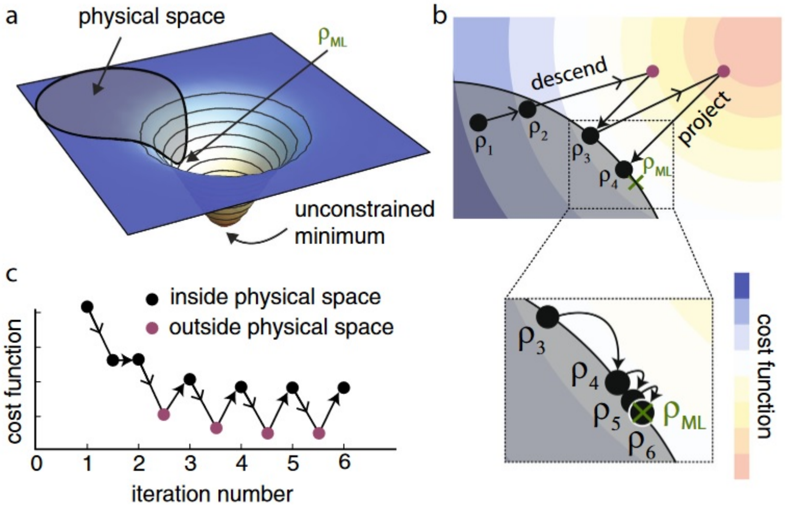


Figure 23: Insights about Projected Gradient Descent, related to the Manifold Conjugate Gradient Descent algorithm. In our case, the physical space would be $Gr(M, D)$ and outside the physical space would be the whole space, $\mathbb{R}^M$. Figure taken from [21].

Now, we can clearly see that the Grassmann gradient defined in (72) belongs to the tangent space, because since it is a projection onto the orthogonal complement, it is easily shown that:

$$\nabla_{\mathbf{X}} F_{Gr(M,D)}(\mathbf{X})^T \mathbf{X} = \mathbf{0}. \tag{73}$$

6.1.3 Grassmann geodesics

Finally, the last concept that we will need to show to understand the Grassmann-based approaches that we will be showing in this chapter is the concept of Grassmann geodesics. It is out of the scope of this work to show the proof of this concept, but it is needed to develop the Manifold Conjugate Gradient Descent methods, and thus we will show the intuition and the basic expressions. Given an arbitrary point in the Grassmanian, $\mathbf{X}$, and a gradient in this point with value $\nabla_{\mathbf{X}} F_{Gr(M,D)}(\mathbf{X}) = \mathbf{Y}$ with a compact SVD being equal to $\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, which is a redefinition of the SVD such that $\boldsymbol{\Sigma}$ is a square matrix ($D \times D$ in this case) not considering the null singular values. Then the formula to follow a path along the Grassmann geodesic is:

$$\mathbf{X}(n) = \left[\begin{array}{cc} \mathbf{XV} & \mathbf{U} \end{array} \right] \left[\begin{array}{c} \cos(n\eta\boldsymbol{\Sigma}) \\ \sin(n\eta\boldsymbol{\Sigma}) \end{array} \right] \mathbf{V}^T. \tag{74}$$

For some arbitrary step size η, and being the functions $\cos(\cdot)$ and $\sin(\cdot)$ the element-wise cosine and sine functions on the diagonal entries of the input matrix. To have more insights about this concept, Figure 21 may be helpful.

7 Problem statement

In the subspace learning domain, we will consider the following general model:

$$\mathbf{y}(n) = \mathbf{H}(n)\mathbf{x}(n) + \mathbf{w}(n). \tag{75}$$

For $n = 1 \ldots N$. Vectors $\mathbf{y}(n)$ and $\mathbf{w}(n)$ have dimension M while vector $\mathbf{x}(n)$ has dimension $D < M$ so then $\mathbf{H}(n)$ has dimensions $M \times D$. In this case, we can say that we have a sparse prior on the dimensionality of the interested signal. In our approach, we assume the dimension D as prior knowledge for our problem. The problem that we want to solve is estimating the subspace that embeds our signal $\mathbf{x}(n)$ from the data $\{\mathbf{y}(n)\}_{n=1,\ldots,N}$ and study the performance quality of this subspace learning problem in different scenarios, for some prior knowledge of the dimension D. We want to understant how different approaches behave in mainly two different scenarios that can be considered part of the subspace learning problem, being the first one the Subspace Tracking, consisting on how well the algorithm is capable to track changes in the signal's subspace, and the second one being the Subspace Estimation problem, consisting on the pure estimation of the signal's subspace. In the latter scenario, we can find derived versions of this problem depending on the restrictions of our input data. If we consider that we have a limited amount of data, for instance having a limited N, then we can transform the model in (75) into:

$$\mathbf{Y} = \mathbf{HX} + \mathbf{W}, \tag{76}$$

where now the matrix $\mathbf{Y}$ is $M \times N$ being the input data, $\mathbf{H}$ is $M \times D$ still being the matrix spanning the signal's subspace, $\mathbf{X}$ is $D \times N$ being the extended latent variables and finally $\mathbf{W}$ has dimensions $M \times N$ being the noise term. In this remodelation of the initial problem in (75), we can see that it can be considered a matrix completion problem, which is of a great interest in the Compress Sensing domain. In the latter hot topic, the problem of matrix completion consists on finding representations of an arbitrary matrix according to some restrictions in the representation. One of the more usual restrictions is that both matrix are semidefinite positive and thus it becomes a convex optimization problem. Still, there are some approaches that try to generalize this results to non semidefinite positive matrices by means of relaxations or non-convex optimization algorithms.

In this way, we can estimate the subspace where the information is confined, so with further processing one could compress or recover information in a more efficient way. Moreover, another interesting application of this kind of solutions is the domain of non-coherent communications, so in this way communications protocols could be depicted to transmit information on a certain subspace, reducing greatly the amount of resources that must be needed while still having reasonable performance, as we will show later in this chapter when we study the performance of these kind of algorithms. In these non-coherent communications, we may intuitively see that the algorithm is able to estimate where the information is located in the space, so it enables the receiver and transceiver to improve the decodification process to achieve better performance than previous non-coherent algorithms. The subspace learning and tracking eliminates the need of a continuous channel estimation, which changes rapidly, and the up-link channel overload due to the notification to the transmitter of the channel state information (CSIT). Instead, the channel subspace changes much slower than the channel matrix, $\mathbf{H}(n)$. What is more, these algorithms are necessary for the Subspace-based anomaly detection.

Note that in this general context of subspace learning, although M increases the number of unknowns of the problem. We will not be interested in estimating directly the matrix $\mathbf{H}$, but only the subspace that it spans which requires less degrees of freedom than estimating the actual matrix. Therefore, the intrinsic parameters for identifying that subspace do not increase with M. Estimating the subspace is a more efficient approach than estimating directly the matrix $\mathbf{H}$, because it is needed fewer data than estimating directly the matrix $\mathbf{H}$ (fewer degrees of freedom) and it contains the same amount of information, geometrically speaking. Note that this is a clear example of an estimation problem on the Grassmann manifold. As we are interested in where the information is contained, we will consider that $\mathbf{H}$ has an orthonormal structure as follows:

$$\mathbf{H}^H\mathbf{H} = \mathbf{I}_D, \tag{77}$$

where the projection matrix has the following shape:

$$\mathbf{P}_H = \mathbf{H}\mathbf{H}^H. \tag{78}$$

Consider that $\mathbf{x}(n)$ and $\mathbf{w}(n)$ are normal vector processes of arbitrary variance per component, uncorrelated in time and space. They can be statistically modeled as:

$$\mathbf{x}(n) \sim \mathcal{N}\left(\mathbf{0}, \sigma_x^2\mathbf{I}_D\right),$$

$$\mathbf{w}(n) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{C}\right).$$

Then the received data stream will be distributed as:

$$\mathbf{y}(n) \sim N\left(\mathbf{0}, \sigma_x^2\mathbf{H}\mathbf{H}^H + \mathbf{C}\right) = N\left(\mathbf{0}, \sigma_x^2\mathbf{P}_H + \mathbf{C}\right).$$

Let's recall a key feature of the Grassmann manifold in order to understand the expected solution. As we are interested in estimating the signal subspace, the main signal feature of our interest is the projector $\mathbf{P}_H$ and thus, the final solution fulfills the following property:

$$\mathbf{H}^{opt} \to \mathbf{H}' = \mathbf{H}^{opt}\mathbf{B},$$

with

$$\mathbf{B}\mathbf{B}^T = \mathbf{B}^T\mathbf{B} = \mathbf{I}_D, \tag{79}$$

where if we find $\mathbf{H}^{opt}$ then $\mathbf{H}'$ is also an optimal solution, so this means that we are not interested the specific rotation from which the signal was generated, only in the spanned subspace. Note that if (79) is fulfilled, the projector is still the same (unique for every subspace). As an additional remark, the optimization on the Grassmanians is, in general, non-convex.

8 Methodology

In this section we will be showing two different families of approaches: the ones that are Grassmann-based, where we show an improvement of the classical GROUSE algorithm among them, and the ones that are based on the direct or indirect estimation of the covariance matrix of the data stream, from which we present the Affine Projection Subspace Tracking algorithm that is a novel algorithm developed with the aid of this work.

8.1 Manifold Conjugate Gradient Descent methods

The Manifold Conjugate Gradient Descent methods for this case will be derived from the single snapshot model. In this way, we will start from the following model:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w}, \tag{80}$$

where the dimensions of each vector are the ones stated in the model in (75). Then, due to the Gaussian distribution behaviour of the data, we can propose the following cost function to optimize in the $Gr(M, D)$ domain:

$$f(\mathbf{H}) = \arg\min_{\mathbf{x}} ||\mathbf{y} - \mathbf{H}\mathbf{x}||_2^2, \tag{81}$$

whose "Euclidean gradient" with respect to $\mathbf{H}$ is:

$$\nabla f(\mathbf{H}) = -2(\mathbf{y} - \mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y})\mathbf{y}^T\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1} = -2\mathbf{r}\mathbf{w}^T, \tag{82}$$

being $\mathbf{r} = (\mathbf{y} - \mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y}) = \mathbf{P}_H^{\perp}\mathbf{y}$, the residual from the Least Squares solution and $\mathbf{w} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y}$, the least squares solution for the initial model. Now, the Grassmaniann gradient can be derived in the following way:

$$\nabla_{Gr(M,D)} f(\mathbf{H}) = (\mathbf{I} - \mathbf{H}\mathbf{H}^T)\nabla f(\mathbf{H}) = (\mathbf{I} - \mathbf{H}\mathbf{H}^T)(-2\mathbf{r}\mathbf{w}^T) = -2\mathbf{r}\mathbf{w}^T, \tag{83}$$

where we note that in this case the "Euclidean gradient" is equivalent as the Grassmaniann gradient, due to the orthogonality of $\mathbf{r}$ with respect to $\mathbf{H}$. Then, we should follow a geodesic path inside the Grassmanian. In order to do so, we must first compute the SVD of $\nabla_{Gr(M,D)} f(\mathbf{H})$, which in this case is trivial as it is rank-one:

$$-2\mathbf{r}\mathbf{w}^T = \left[\begin{array}{cccc} \frac{-\mathbf{r}}{||\mathbf{r}||} & \mathbf{x}_2 & \dots & \mathbf{x}_D \end{array}\right] \times \mathrm{diag}\left(\begin{array}{cccc} 2||\mathbf{r}||||\mathbf{w}|| & 0 & \dots & 0 \end{array}\right) \times \left[\begin{array}{cccc} \frac{\mathbf{w}}{||\mathbf{w}||} & \mathbf{y}_2 & \dots & \mathbf{y}_D \end{array}\right]^T = \mathbf{U}\mathbf{D}\mathbf{V}^T. \tag{84}$$

Even though it may seem as a feature not that important, the fact that it is rank-one makes this gradient suitable for a Manifold Conjugate algorithm because the SVD may be computationally costly in general, making rank-one gradients a really powerful feature to look for when solving this kind of problems. Now, continuing and by using the expression of the Grassmanian geodesic step in the $-\nabla_{Gr(D,N)} f(\mathbf{H})$ direction, we get the updating rule:

$$\mathbf{H}_k = \left[\begin{array}{cc} \mathbf{H}_{k-1}\mathbf{V} & \mathbf{U} \end{array}\right] \left[\begin{array}{c} \cos(\mathbf{D}\eta) \\ \sin(\mathbf{D}\eta) \end{array}\right] \mathbf{V}^T,$$

$$\mathbf{H}_k = \mathbf{H}_{k-1} - \mathbf{H}_{k-1}\frac{\mathbf{w}_k\mathbf{w}_k^T}{||\mathbf{w}_k||^2} + \mathbf{H}_{k-1}\frac{\mathbf{w}_k\mathbf{w}_k^T}{||\mathbf{w}_k||^2}\cos(2||\mathbf{r}_k||||\mathbf{w}_k||\eta) + \sin(2||\mathbf{r}_k||||\mathbf{w}_k||\eta)\frac{\mathbf{r}_k\mathbf{w}_k^T}{||\mathbf{r}_k||||\mathbf{w}_k||},$$

$$\mathbf{H}_k = \mathbf{H}_{k-1} + \left((\cos(2||\mathbf{r}_k||||\mathbf{w}_k||\eta) - 1)\frac{\mathbf{H}_{k-1}\mathbf{w}_k}{||\mathbf{w}_k||} + \sin(2||\mathbf{r}_k||||\mathbf{w}_k||\eta)\frac{\mathbf{r}_k}{||\mathbf{r}_k||}\right)\frac{\mathbf{w}_k^T}{||\mathbf{w}_k||}. \tag{85}$$

We have finally achieved the updating rule for the Least Squares cost function for an arbitrary M and D. This solution is often called the Grassmannian Rank-One Update Subspace Estimation (GROUSE) [7]. In order to highlight the intuition of this result, one could particularize it in $Gr(M, 1)$ by considering the following model:

$$\mathbf{y} = a\mathbf{h} + \mathbf{w}, \tag{86}$$

where a is any positive scalar. In this particularization, the MCGD for the Least Squares function has the following shape:

$$\mathbf{h}_{k+1} = \mathbf{h}_k \cos(||\mathbf{r}_k||\eta) + \frac{\mathbf{r}_k}{||\mathbf{r}_k||} \sin(||\mathbf{r}_k||\eta), \tag{87}$$

where we can see that it gets to an intuitive solution. The previous solution can be seen as a rotation in the unit sphere to look for the vector that minimizes $\mathbf{r}$. We may also remark that this algorithm is minimizing the following function with the orthonormality constraints from the Grassmanian, essentially:

$$f(\mathbf{H}) = ||(\mathbf{I}_M - \mathbf{H}\mathbf{H}^T)\mathbf{y}||_2^2 = ||\mathbf{P}_H^{\perp}\mathbf{y}||_2^2. \tag{88}$$

8.1.1 Weighted Least Squares MCGD

In order to improve the previous algorithm, and as a novel proposal of this work, a better minimizer of the objective function would yield a better performance, especially if the noise is correlated and M is high. As M increases, known results for channel identification start to fail due to non-scalable operations such as matrix inverses or matrix decompositions. In this way, the mentioned algorithms become more computationally efficient than the classical ones. Then, in this particular case of the weighted least squares, we are always required to do an inverse of a $D \times D$ matrix, being computationally efficient, and its solution improves considerably the performance of the previous algorithm, in terms of convergence. The only thing that is modified from the previous algorithm is the definitions of $\mathbf{r}$ and $\mathbf{w}$, where they become:

$$\mathbf{r} = \mathbf{y} - \mathbf{H}(\mathbf{H}^T\mathbf{C}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{C}^{-1}\mathbf{y},$$

$$\mathbf{w} = (\mathbf{H}^T\mathbf{C}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{C}^{-1}\mathbf{y}.$$

Note that these redefinitions fit into the consideration of the following cost function:

$$f(\mathbf{H}) = \arg\min_{\mathbf{x}} (\mathbf{y} - \mathbf{H}\mathbf{x})^T\mathbf{C}^{-1}(\mathbf{y} - \mathbf{H}\mathbf{x}),$$

where $\mathbf{C}^{-1}$ is the inverse of the covariance matrix of $\mathbf{y}$, also called the precision matrix. However, this modification loses the orthonormality between matrix $\mathbf{H}$ and $\mathbf{r}$, because we are now not performing an orthogonal projection but an oblique projection. In this way, now the actual Grassmann gradient of this function would be:

$$\nabla_{Gr(M,D)} f(\mathbf{H}) = (\mathbf{I} - \mathbf{H}\mathbf{H}^T)\nabla f(\mathbf{H}) = (\mathbf{I} - \mathbf{H}\mathbf{H}^T)(-2\mathbf{r}\mathbf{w}^T),$$

without the orthonormal simplification that we see in equation (80), adding a limitation to this modification of the algorithm since we do not know the orthogonal projector $\mathbf{P}_H^{\perp} = \mathbf{I} - \mathbf{H}\mathbf{H}^T$. This forces the algorithm to re-project each iteration to the Grassmann manifold by applying an orthonormalization function such as the Gramm-Schmidt procedure, much similar to the Projected Gradient Descent algorithm. The new updating equations would be:

$$\mathbf{A}_k = \mathbf{H}_{k-1} + \left((\cos(2||\mathbf{r}_k||||\mathbf{w}_k||\eta) - 1)\frac{\mathbf{H}_{k-1}\mathbf{w}_k}{||\mathbf{w}_k||} + \sin(2||\mathbf{r}_k||||\mathbf{w}_k||\eta)\frac{\mathbf{r}_k}{||\mathbf{r}_k||} \right) \frac{\mathbf{w}_k^T}{||\mathbf{w}_k||}, \tag{89}$$

$$\mathbf{H}_k = \mathrm{orth}\{\mathbf{A}_k\}, \tag{90}$$

where the orth$\{\cdot\}$ function performs the Gramm-Schmidt orthogonalization procedure. Another alternative to this idea of re-projecting the solution onto the Grassmanian by Gramm-Schmidt orthogonalization, is projecting the Gradient onto the estimated orthogonal complement at each iteration, being defined as:

$$\mathbf{P}_{H_k}^{\perp} = \left(\mathbf{I} - \mathbf{H}_k\mathbf{H}_k^T\right).$$

The final equations for this approach would be:

$$\mathbf{r}_k = \mathbf{P}_{H_k}^{\perp}\left(\mathbf{y} - \mathbf{H}_k(\mathbf{H}_k^T\mathbf{C}\mathbf{H}_k)^{-1}\mathbf{H}_k^T\mathbf{C}\mathbf{y}\right),$$

$$\mathbf{w}_k = (\mathbf{H}_k^T \mathbf{C} \mathbf{H}_k)^{-1} \mathbf{H}_k^T \mathbf{C} \mathbf{y},$$

$$\mathbf{H}_k = \mathbf{H}_{k-1} + \left((\cos(2||\mathbf{r}_k||||\mathbf{w}_k||\eta) - 1) \frac{\mathbf{H}_{k-1}\mathbf{w}_k}{||\mathbf{w}_k||} + \sin(2||\mathbf{r}_k||||\mathbf{w}_k||\eta) \frac{\mathbf{r}_k}{||\mathbf{r}_k||} \right) \frac{\mathbf{w}_k^T}{||\mathbf{w}_k||},$$

where the only modification would be in the residuals vector, $\mathbf{r}$, which is where we put the orthogonal projection onto the estimated orthogonal complement of each iteration. We would expect that this last approach is slower than the one with the Gramm-Schmidth orthogonalization, due to the uncertainty in the orthogonal projector, but still this approach yields similar results.

Here in Figure 24, there is a plot that shows the amount of gain between the Weighted Least Squares and Least Squares cost functions for the Gramm-Schmidt orthogonalization approach, evidencing it in terms of the speed of convergence. Figure 24 shows 20 Monte-Carlo averages on the realizations of the chordal norm modification in (69) for both objective functions.

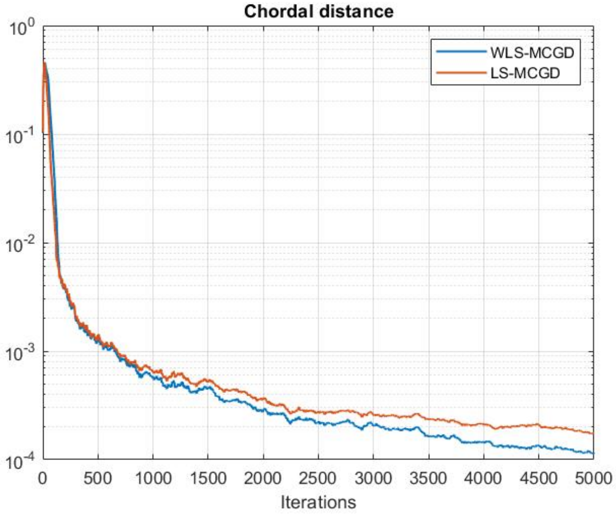


Figure 24: WLS and LS convergence. WLS is Gramm-Schmidt based.

And now, in Figure 25 we used the projection onto each iteration estimated orthogonal complement, $\mathbf{P}_{H_k}^{\perp}$, in order to compare with the previous case. In this simulation we have also used 20 Monte-Carlo tests.

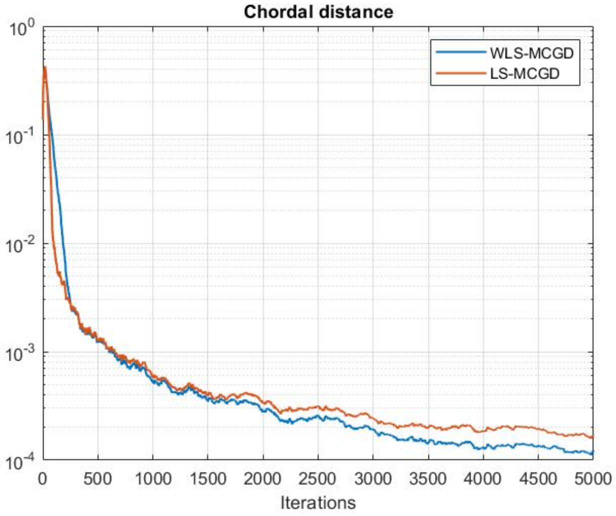


Figure 25: WLS and LS convergence. WLS gradient is projected onto each step orthogonal complement estimation.

Comparing the previous two figures, we can see that there is almost no loss in the orthogonal projector estimation, so both approaches yield the same performance. However, the orthogonal complement estimation approach is prefered over the one based on Gramm-Schmidt orthogonalization because it requires less computational complexity.

8.2 Covariance-based algorithms

This family of algorithms is based on the direct or indirect estimation of the covariance matrix $\mathbf{C}$. In this subsection, we will show the novel algorithm that we have been working with while researching for this project, which is the Affine Projection Subspace Tracking algorithm, in contrast with other state-of-the-art algorithms. Those algorithms are the following ones:

- Data Projection Method (DPM) that can be interpreted as the adaptive Least Mean Square version for subspace tracking and estimation.
- tProjection Aproximation Subspace Tracking (PAST), being the RLS version of the Subspace Tracking algorithm.

8.2.1 Affine Projection Subspace Tracking algorithm

This algorithm is a novel solution that we propose to be half-way between the simplicity and adaptability of the LMS-like algorithms and all the tracking capabilities of the RLS-like algorithms. In order to do so, we opted to use an Affine Projection-like algorithm by using the following restriction:

$$\mathrm{trace}(\mathbf{H}^T\mathbf{C}\mathbf{H}) = D, \tag{91}$$

where $\mathbf{C}$ is the data covariance matrix. In this way, and adding into consideration the orthonormality constraint, this constraint is actually maximizing the left-hand term of this equality, as it is the maximum value that this function can achieve, if some normalizations are being done. Then, the optimization problem to solve to derive this algorithm is the following:

$$\mathbf{X}[n] \triangleq \mathbf{y}[n]\mathbf{y}^T[n]\mathbf{W}^T[k],$$

$$\mathbf{W}[k+1] \triangleq \arg \min_{\mathbf{W}[k+1]} \|\mathbf{W}[k+1] - \mathbf{W}[k]\|_F^2 \quad \text{s. t. } \mathbf{W}^T[k+1\mathbf{X}[p] \equiv \mathbf{I}_D \quad , \quad p = 0, ..., P-1, \tag{92}$$

where P is what we call the projection order, denoting the amount of memory of the algorithm. The efficient solution for this problem, following the usual APA procedures, considering $D > P$ is:

$$\bar{\mathbf{Y}}[k] \triangleq \mathbf{Y}^T[k]\mathbf{Y}[k]\mathbf{W}[k],$$

where $\mathbf{Y}[k]$ is the matrix containing k data vectors:

$$\mathbf{Z}[k] \triangleq \mathbf{Y}[k]\bar{\mathbf{Y}}[k],$$

$$\mathbf{W}[k+1] = \mathbf{W}[k] + \mu\mathbf{Y}^T[k]\left(\mathbf{Z}[k]\mathbf{Z}^T[k] + \delta\mathbf{I}_P\right)^{-1}\mathbf{Z}[k]\bar{\mathbf{E}}[k], \tag{93}$$

where δ is a regularization factor that fixes the numerical problem that may happen when inverting a matrix whose rank is unknown. Note that this is an inverse of a $P \times P$ matrix. Applying the Woodbury inversion lemma, we can lead to a solution being more efficient when $P > D$, as we will be inverting a $D \times D$ matrix instead. This solution is the following one:

$$\mathbf{W}[k+1] = \mathbf{W}[k] + \mu\mathbf{X}^T[k]\mathbf{Z}[k]\left(\mathbf{Z}^T[k]\mathbf{Z}[k] + \delta\mathbf{I}_D\right)^{-1}\bar{\mathbf{E}}[k]. \tag{94}$$

For more details, you may want to read the Annex of this work, where we show all the details of this algorithm.

8.2.2 Data Projection Method

Several adaptive solutions of Subspace Tracking are based on the following equation [5]:

$$\mathbf{H}_n = \text{orth}\{\mathbf{R}\mathbf{H}_{n-1}\}, \tag{95}$$

where $\mathbf{R}$ is any semidefinite positive matrix and the sequence $\{\mathbf{H}_n\}$ is given by $N \times D$ matrices which are proven to converge to an orthonormal basis spanned by the firsts D singular vectors of $\mathbf{R}$. However, we will make use of an alternative form of (95), leading to a more flexible algorithm, for instance to estimate the noise subspace instead of the signal subspace directly. This alternative form is the following:

$$\mathbf{H}_n = \text{orth}\{(\mathbf{I}_M \pm \mu\mathbf{R})\,\mathbf{H}_{n-1}\}, \tag{96}$$

where the $+$ will be used to estimate the signal subspace and the $-$ is used for the noise subspace. For simplicity, we have chosen the function orth$\{\cdot\}$ to be the Gram-Schmidt orthonomalization method, but it could potentially be any arbitrary orthonormalization algorithm. Note that we are either maximizing or minimizing a function by changing this sign. In our context, $\mathbf{R}$ will be usually the autocorrelation matrix of our input data, so the different algorithms differ in how do we estimate the autocorrelation matrix. In the case of the DPM, it uses the simplest estimator of $\mathbf{R}$ which is:

$$\mathbf{R}_n = \mathbf{y}(n)\mathbf{y}(n)^T, \tag{97}$$

where now we can derive the adaptive subspace tracking algorithm from (96)

$$\mathbf{H}_n = \text{orth}\{\left(\mathbf{I}_M \pm \mu\mathbf{y}(n)\mathbf{y}(n)^T\right)\mathbf{H}_{n-1}\}. \tag{98}$$

The rationale about the ± and the above equation is that this solution comes from the optimization of the following cost function:

$$J(\mathbf{H}) = \text{trace}(\mathbf{H}^T\mathbf{C}\mathbf{H}), \tag{99}$$

where $\mathbf{C}$ is the autocorrelation matrix of the input data. In this case, given the fact that the matrix $\mathbf{H}$ is $M \times D$, the optimization of (95) can get to two different solutions: if we maximize this function, the optimal solution would be the D eigenvectors corresponding to the largest eigenvalues of $\mathbf{C}$, corresponding to the signal subspace, and if we minimize this function, the optimal solution would be the D eigenvectors corresponding to the smallest eigenvalues of $\mathbf{C}$, corresponding to the noise subspace. Also, it was shown in [12] that the optimal solution of this function can be derived recursively by the matrix sequence in (95). However, as this is an stochastic approach, it may have numerical issues from the fact that the product $\mathbf{RH}_n$ may result in a non-semidefinite positive matrix. The solution to this issue is the modification in (96) as the identity matrix ensures this condition for a sufficiently small value of μ. The details on this work can be found in [11], or more explicitly and detailed, in [12].

8.2.3 Projection Approximation Subspace Tracking

Another approach to reach a solution to this problem is to adopt an RLS-based algorithm which also requires the implicit estimation of $\mathbf{C}^{-1}$. To do so and based on [10], we start from the following cost function:

$$J(\mathbf{H}_N) = \sum_{j=1}^{N} \beta^{n-j} ||\mathbf{y}(j) - \mathbf{H}_n\mathbf{H}_n^T\mathbf{y}(j)||^2, \tag{100}$$

where it is proven in [10] that this cost function is equivalent to the one in the DPM. Note that if we force in each iteration that $\mathbf{H}_n$ is orthonormal, then this cost function becomes:

$$J(\mathbf{H}_N) = \sum_{j=1}^{N} \beta^{n-j} || \left(\mathbf{I}_M - \mathbf{H}_n\mathbf{H}_n^T\right)\mathbf{y}(j)||^2 = \sum_{j=1}^{N} \beta^{n-j} ||\mathbf{P}_{H_n}^{\perp}\mathbf{y}(j)||^2, \tag{101}$$

where this can be seen as the data centric measure of subspace tracking algorithms, with a forgetting factor. Now, as it is done in their work, if we now apply the typical RLS procedures which are the recursive estimation of the latent variables and covariance matrix (in this case, the inverse of this matrix), we get to the RLS-form of this problem:

$$\mathbf{z}_n = \mathbf{H}_{n-1}^T\mathbf{y}(n), \tag{102}$$

$$\mathbf{h}_n = \mathbf{P}_{n-1}\mathbf{z}_n, \tag{103}$$

$$\mathbf{g}_n = \frac{1}{\beta + \mathbf{z}_n^T\mathbf{h}_n}\mathbf{h}_n, \tag{104}$$

$$\mathbf{P}_n = \text{Tri}\left(\frac{1}{\beta}\mathbf{P}_{n-1} - \mathbf{g}_n\mathbf{h}_n^T\right), \tag{105}$$

where Tri(·) is a function that computes just the upper triangular entries of the argument:\mathbf{R}

$$\mathbf{Q}_n = \frac{1}{\mu}\left(\mathbf{I} - \frac{\mathbf{Q}_{n-1}\mathbf{y}(n)\mathbf{y}(n)^T}{\mu + \mathbf{y}^T(n)\mathbf{Q}_{n-1}\mathbf{y}(n)}\right)\mathbf{Q}_{n-1}, \tag{106}$$

$$\mathbf{e}_n = \mathbf{Q}_n\mathbf{y}(n) - \mathbf{H}_{n-1}\mathbf{z}_n, \tag{107}$$

$$\mathbf{H}_n = \mathbf{H}_{n-1} - \mathbf{e}_n\mathbf{g}_n^T. \tag{108}$$

The initialization of all the required matrices is similar as in the usual RLS algorithm. Note that matrix $\mathbf{Q}_n$ is the n-th iteration in the estimated inverse of the correlation matrix, therefore the usual initialization is:

$$\mathbf{Q}_0 = \varepsilon\mathbf{I}, \tag{109}$$

where ε has a large value. In the case of matrix $\mathbf{P}_n$, the authors recommend the following initialization:

$$\mathbf{P}_0 = \alpha\mathbf{I}, \tag{110}$$

where α is any appropriate positive scalar. This solution offers a quick response to the subspace estimation, but it lacks in its adaptability to change capabilities, as it will be shown in the results section.

9 Simulation results

In this section we will compare the performances of all the aforementioned algorithms. Although this task may seem hard due to differences in their approaches and specializations, we will try to be as fair as possible, for instance by tuning each algorithm to achieve their best performance and comparing those top performances. In the Subspace Learning domain we can differentiate mainly two scenarios: the Subspace Estimation problem and the Subspace Tracking problem.

In the Subspace Estimation domain, we are interested in algorithms that can achieve the lowest value in the least amount of iterations of each cost function. This can be expected and make sense because in this scenario the subspace is assumed to be stationary, so we can study the lower bounds of this metrics, tune the algorithms, so they reach a stationary solution. In this scenario, we will test the performance of both the MCGD algorithms, the Least Squares and the Weighted Least Squares versions, and the DPM algorithm, because the other two algorithms, PAST and APST, are more Subspace Tracking oriented and often arise numerical issues making the comparison with the rest of the algorithms unfair and much more complicated.

On the other hand, we have the Subspace Tracking problem. There, we do not expect that the subspace is stationary along the iterations, it may change abruptly or smoothly, so we do not expect that they are capable of giving a stationary solution. However, we expect that they react quite fast when these sudden changes occur or that they are capable to follow the smooth change along the iterations.

To study how well these algorithms perform in one case, we will use performance measures that we have already been showing throughout this part:

- Least Squares cost function: This cost function is more Subspace Tracking oriented, as it usually reaches a minimum at the noise level, even if we are using the actual subspace to test this cost function. It may also be used in Subspace Tracking as an indicator. It has the following shape, assuming that the matrix $\mathbf{H}$ is orthonormal:
$$f(\mathbf{H}) = || \left(\mathbf{I}_M - \mathbf{H}\mathbf{H}^T\right) \mathbf{y}||_2^2 = ||\mathbf{P}_H^\perp \mathbf{y}||_2^2. \tag{111}$$
- Normalized modification of the chordal distance: This cost function is both useful in the Subspace Tracking and Subspace estimation. Remember that it was defined in (69). It has the inconvenience of being essentially an order 4 moment and not having all the norm properties, but still it is extremely useful for its invariance against rotations:
$$e(\mathbf{H}) = \frac{1}{2D} \left\| \hat{\mathbf{P}}_X - \mathbf{P}_X \right\|_F^2. \tag{112}$$
- Subspace angle: This function has a similar shape as the normalized chordal distance due to the fact that both metrics are function of the principal angles of the subspace. However, this cost function gives insights about the amount of change in a subspace because its values are much more intuitive and familiar:
$$d_A(\mathbf{H}, \hat{\mathbf{H}}) = \min(\theta_i) \quad \forall i. \tag{113}$$

9.1 Subspace Estimation

In this scenario, we will consider the following simple model:

$$\mathbf{y}(n) = \mathbf{H}\mathbf{x}(n) + \mathbf{w}(n). \tag{114}$$

Note that in this case the subspace spanning matrix, $\mathbf{H}$, is stationary and the rest of the vectors, $\mathbf{x}(n)$ and $\mathbf{w}(n)$, are i.i.d. Gaussian vectors. The latter vectors should never be non-stationary because if not, the algorithms would not have enough information to estimate the subspace. This would obviously apply also in subspace tracking.

This kind of environment may be much more theoretical, because in this way we are able to study the lower bounds of the metrics and behaviour of the algorithms. In this scenario, we may want to fulfill the conditions for a stationary solution for the step sizes, which are:

$$\sum_{k=1}^{\infty} \mu_k \to \infty,$$

$$\mu_k \to 0 \quad \text{when} \quad k \to \infty.$$

For simplicity, we will chose $\mu_k = \frac{C_1}{k}$ for DPM and $\eta_k = \frac{C_2}{k}$ for the MCGD algorithms. Note that these constants must be different because essentially both algorithms behave differently. In this simulation we have used the following parameters to compare both kinds of algorithms:

Parameter	Value
N_o, Noise power	0.001
M, total dimensionality	10
D, Signal's subspace	5
Monte-Carlo iterations	10
σ_x^2	1
C_1	1
C_2	10

The key parameters in terms of the subspace estimations is the relationship between M and D. In the subspace estimation theory, it is known that the Cramer-Rao bound-like for this kind of problem is proportional to $(D(M-D))^{\beta}$, seen in [13] and [14], where β depends on the metric that one is using, and it reaches a maximum value when $D = \lfloor \frac{M}{2} \rfloor$. In the case of the chordal norm, as we are approaching it, this constant is $\beta = 1$, as we will show in a following figure. The rationale of the values of C_1 and C_2 are based on their related works [12] and [7] respectively, as in the case of the DPM it usually expects a value between 0 and 1 and in the case of MCGD, the GROUSE authors recommends higher values than one may expect. We should remark that this comparison is not that evident, as the Grassmann algorithms are a complete change of paradigm. In Figure 26, we present the comparison between these two families of algorithms.

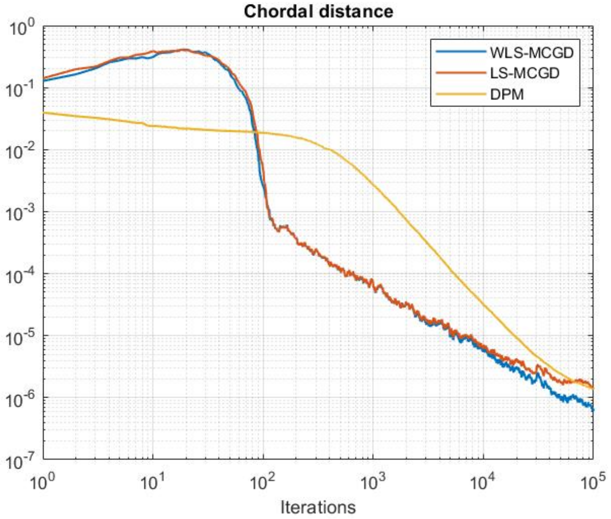
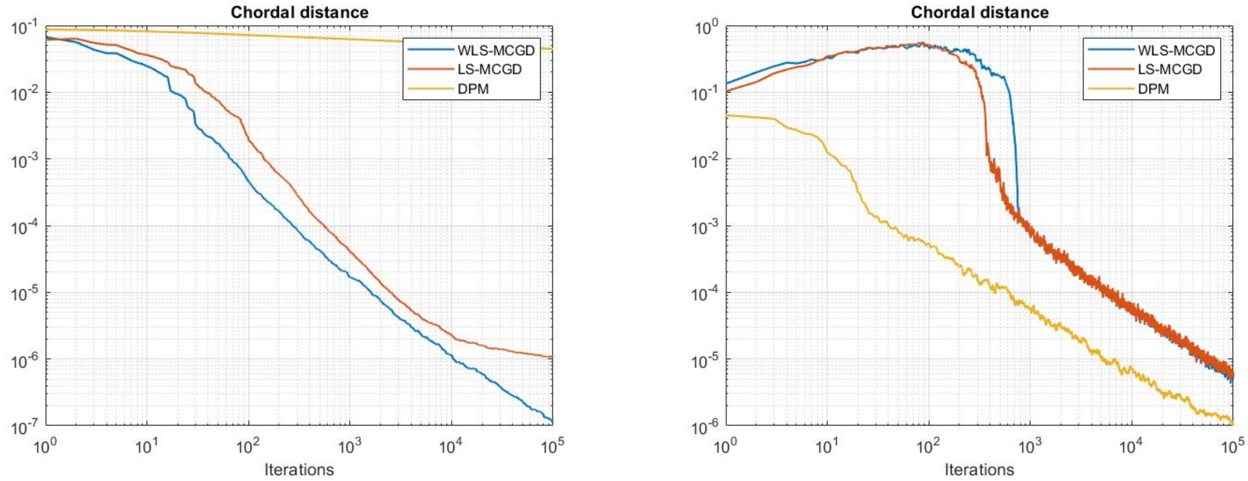


Figure 26: Subspace Estimation. Chordal norm along the iterations. $C_1 = 1$ and $C_2 = 10$.

In this figure, we can observe that there are mainly two regions for each curve. The first region is a transitory region where the algorithms still have not identified the subspace and are looking for the proper direction, as it is more clear in the case of MCGD algorithms, and the second region when the algorithms are converging to the actual subspace. In the case of the Least Squares-MCGD and the DPM algorithms, we can see that there is also a third region where they are getting to a stationary solution, although it is slightly truncated.

It is also evident that the Grassmann-based algorithms find the proper direction much earlier than the DPM, but suffer from a worse transient state. This transient state is known in the subspace estimation domain and the behaviour is tightly related to the kind of algorithms that one is studying, in terms of the duration and the kind of evolution in this transient state. In the following figure, we changed the values of C_1 and C_2 to 0.1 and 1, respectively, in one plot and to 10 and 100 in the other, to evidence the impact of these parameters.



(a) $C_1 = 0.1$ and $C_2 = 1$.

(b) $C_1 = 10$ and $C_2 = 100$.

Figure 27: Subspace Estimation. Impact of C_1 and C_2.

In this latter figure, it is quite evidenced that the comparison between these two families of algorithms is not obvious, but still one can observe that the trends are the ones that have already explained in the previous figure. Lastly, as a remark, note that we are not taking into account the amount of noisiness of the solution, as we are making MonteCarlo averages along the realizations of each experiment.

9.2 Subspace Tracking

On the other hand, to model the subspace tracking scenario we are considering the following model:

$$\mathbf{y}(n) = \mathbf{H}(n)\mathbf{x}(n) + \mathbf{w}(n). \tag{115}$$

In this case $\mathbf{H}(n)$ spans a subspace which is non-stationary, and the rest of vectors, $\mathbf{x}(n)$ and $\mathbf{w}(n)$ are i.i.d. Gaussian vectors. We are testing the adaptability of the algorithms when the environment changes, because there are possible applications that may suffer from this kind of changes. In this sense, there are two possible kinds of variations in the spanned subspace: A smooth evolution of the subspace and abrupt changes in the subspace.

An example of sudden and abrupt changes in the subspace may be found in non-coherent MIMO channels as one could experience a channel outage so then, when the communication is recovered, the channel may have changed completely. Even if the channel suffers from partial outages, i.e. an antenna stops working, the subspace still may change abruptly. Even in the case of nominal conditions, similarly as in the GNSS PPP context, it is observed that the fast fading of these millimiter wave channels acts quite fast as compared to the subspace spanned by the channel, which is much more slower.

On the other hand, smooth changes in the subspace is a suitable model in the context of GNSS PPP kind of environments, as the relative continuous movement of the satellites leads to an approximately smooth changes, which we will model in the smooth evolution subsubsection.

In this section, we will show and compare the performance of the shown novel Subspace tracking approaches (MCGD and APST) with the state-of-the-art approaches (PAST and DPM). As the comparison of these algorithms is hard to perform, as shown in the previous subsection, we will again use the step size that yields the best performance for all the algorithms, as this is the main free parameter that we have. In the case of subspace tracking scenarios, we must set all the possible step sizes-related parameters to be a constant. It must be in this way and not satisfying the stationary-solution conditions for step sizes, recalling that they are the following ones:

$$\sum_{k=1}^{\infty} \mu_k \to \infty,$$

$$\mu_k \to 0 \ \text{ when } \ k \to \infty.$$

This is due to the fact that in subspace tracking the algorithms are not expected to have the smallest possible error that they can achieve, being achieved by reaching a stationary point in their solution, but instead they are required to be able to track the possible changes that the environment may have and thus, this is not compatible with having a stationary solution. In the following table we sum up the parameters that we use in the following experiments:

Parameter	Value
N_o, Noise power	0.01
M, total dimensionality	10
D, Signal's subspace	5
Monte-Carlo iterations	30
σ_x^2	1

The key point of these parameters is the fact that $D = \lfloor \frac{M}{2} \rfloor$ because it is the worst case scenario in the subspace learning, as we have already mentioned previously.

9.2.1 Abrupt changes: Subspace Tracking

In this scenario, we will test the sudden abrupt changes in the subspace. We will make two tests in a single simulation consisting on the first one being a total change in the subspace, for instance, we will change completely the spanned subspace by changing the orthonormal matrix from the MIMO channel entirely. The second test will be done by just changing one single vector of the previous orthonormal matrix. In Figure 28, we show the subspace metrics that we have already presented along the iterations for these two scenarios.

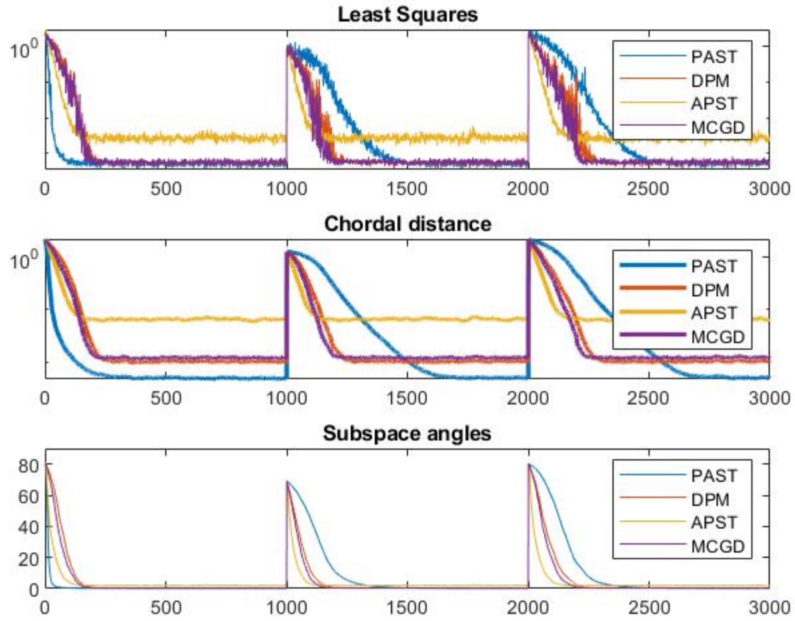


Figure 28: Subspace Tracking abrupt changes. Subspace metrics along the iterations.

There are several things to remark in Figure 28. Firstly, there are three sets of 1000 iterations each, in terms of the change in the subspace, and in each set of 1000 iterations the subspace spanned by the signal is constant so it is clear that the algorithms converged to a steady state. Note that inside each independent set, the conditions are equivalent to those in the Subspace Estimation. Secondly, we can see that there is a difference in the disruption between the first and the second sudden change and this is due to the fact that the first sudden change is a partial matrix change, where we just changed one single vector of the previous matrix which as we can see in the figure, it changes almost completely the spanned subspace; and in the second sudden change, we simply changed the matrix entirely by another that spanned another subspace.

We can also see in this figure that the different algorithms converge to a different value in the subspace based metrics, being much more clear in the chordal distance plot. In this sense, the PAST algorithm is capable of reaching the lowest value, followed by the DPM and the MCGD, and finally the APST always converges to a higher value than the rest of algorithms. However, the adaptability to sudden changes goes all the way around, the APST is the algorithm that adapts better to sudden changes, but still the DPM and MCGD algorithms can adapt to these changes quite fast. Note that the free parameter of each algorithm, step sizes mainly, have been tuned in so their perform at their best in terms of adaptability and convergence. For example, if the step size in the PAST algorithm is increased, and inversely does the forgetting factor, it suffers from numerical issues (it diverges), while the other algorithms may become faster if we increase their step size, but so does the convergence point. What is more, we can see that in the Least Squares cost function all the algorithms, but the APST, converge to the same value due to the fact that the subspace error is much lower than the error coming from the noise.

To conclude, it is clear that these scenarios show a clear trade-off between the adaptability and the convergence point of this kind of algorithms. Furthermore, it also shows that the DPM and MCGD algorithms have a similar performance, although they are essentially different algorithms.

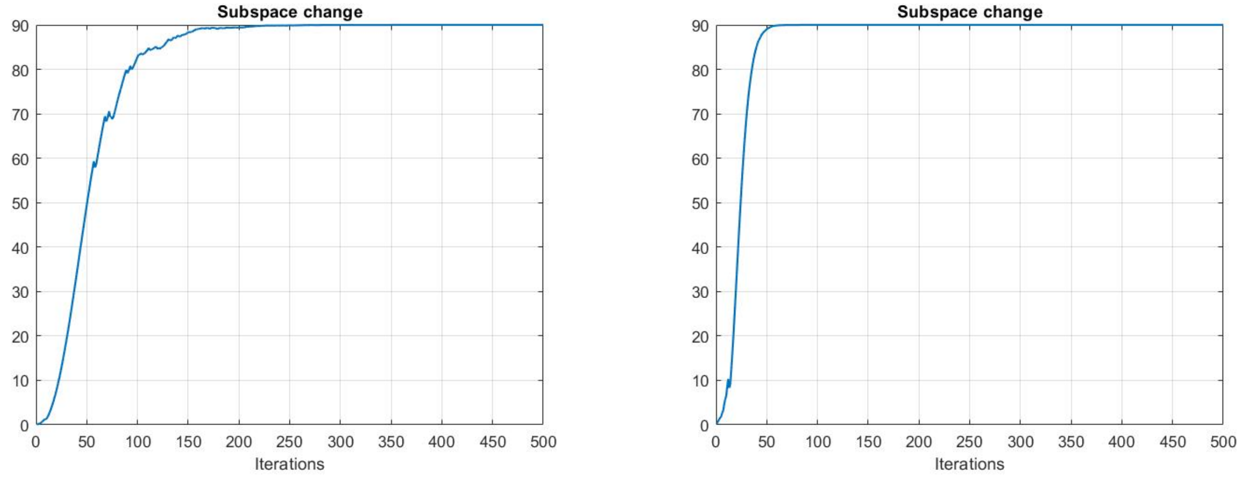
9.2.2 Smooth changes: Subspace Tracking

Now, we are interested in a continuous change of subspace and how the algorithms are capable of tracking this continuous change. In order to perform this smooth change, we have chosen to move along the Grassmanian by moving along a geodesic curve in this set. In order to do so, we may just simply follow a geodesic of a random gradient, so considering that $\mathbf{H}(0) = \mathbf{H}_0$ and the random gradient $\nabla_{\mathbf{H}} f(\mathbf{H}) = \mathbf{A}$, the path along the geodesic is computed with the following equations:

$$\nabla_{\mathbf{H}} F_{Gr(M,D)}(\mathbf{H}(n)) = \mathbf{P}_H^{\perp} \nabla_{\mathbf{H}} f(\mathbf{H}) = (\mathbf{I}_M - \mathbf{H}_0 \mathbf{H}_0^T)\mathbf{A}, \tag{116}$$

$$\mathbf{H}(n) = \left[\begin{array}{cc} \mathbf{H}_0\mathbf{V} & \mathbf{U} \end{array} \right] \left[\begin{array}{c} \cos(n\eta\mathbf{\Sigma}) \\ \sin(n\eta\mathbf{\Sigma}) \end{array} \right] \mathbf{V}^T, \tag{117}$$

being $\mathbf{U\Sigma V}^T$ the compact SVD of $\mathbf{A}$. In this equations, the key parameter that regulates the amount of subspace change is η, so we will just focus on this parameter to test this feature of the algorithms. In order to illustrate, intuitively, the impact of η, Figure 29 shows the amount of change of $\mathbf{H}(n)$ with respect to $\mathbf{H}_0$, as measured in subspace angles, for a couple of values of η which will be the ones that we will be using in these tests.

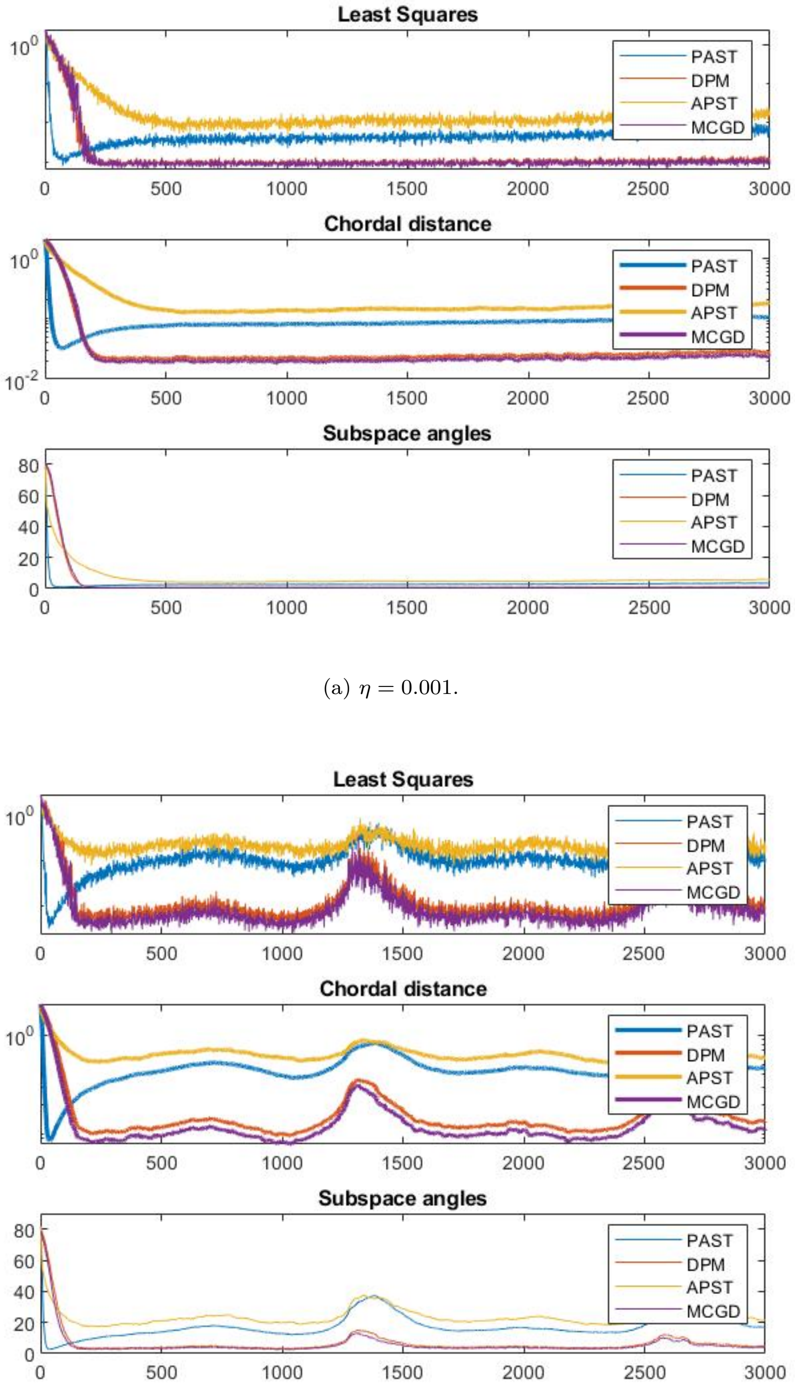


(a) $\eta = 0.001$.

(b) $\eta = 0.005$.

Figure 29: Subspace Tracking. Subspace change measured in subspace angles.

Note in Figure 29 that whenever those curves reach the value of 90 degrees, the subspace may still be changing but the test is not capable to see the changes above this angle as the subspace is already perpendicular to our initial subspace. Now, in Figure 30 we will see how the algorithms are capable to cope with this amount of change.



(a) $\eta = 0.001$.

(b) $\eta = 0.005$.

Figure 30: Subspace Tracking. Smooth tracking.

Finally, we can see in Figure 30 that the performance is almost kept in the smooth subspace change, except for the PAST algorithm. In this test, the PAST algorithm falls behind to the convergence point of the

APST algorithm, being also increased quite heavily in this case. In this way, we see that even if the PAST is capable of having a much lower convergence point, it lacks the tracking capabilities of other algorithms. This makes more obvious the trade-off between convergence point and tracking capabilities.

Part III
Subspace-based anomaly detection

We have finally solved the two independent problems of Anomaly Detection and Subspace Learning separately. Even though they were related, the approaches and philosophy of the two paradigms are completely independent and so where the concepts involved. Now that we have developed robust solutions for both independent problem, we are now capable of joining the Sparse Anomaly Detection and Subspace Learning into a single algorithm. The robustness in the Dual Ascent for Anomaly Detection algorithm to new data (avoiding overfitting) makes it a suitable choice for continuous anomaly detection and correction. We will generalize it to the Subspace Learning environment, where the algorithm must simultaneously estimate the subspace and the anomalies from the data.

As one may expect, doing both tasks continuously is more costly, so the performance of both subproblems will worsen slightly. However, we will show that the separability of both problems can be helpful to develop an algorithm that continously detects anomalies while learning the inherent subspace from the data. In contrast with part I, in this section the Dual Ascent for Anomaly detection will be working as an online recursive algorithm which looks for generalization of the solution for new incoming data, instead of a batch iterative algorithm whose main objective is to overfit as much as possible the solution to the input signal.

10 Problem statement

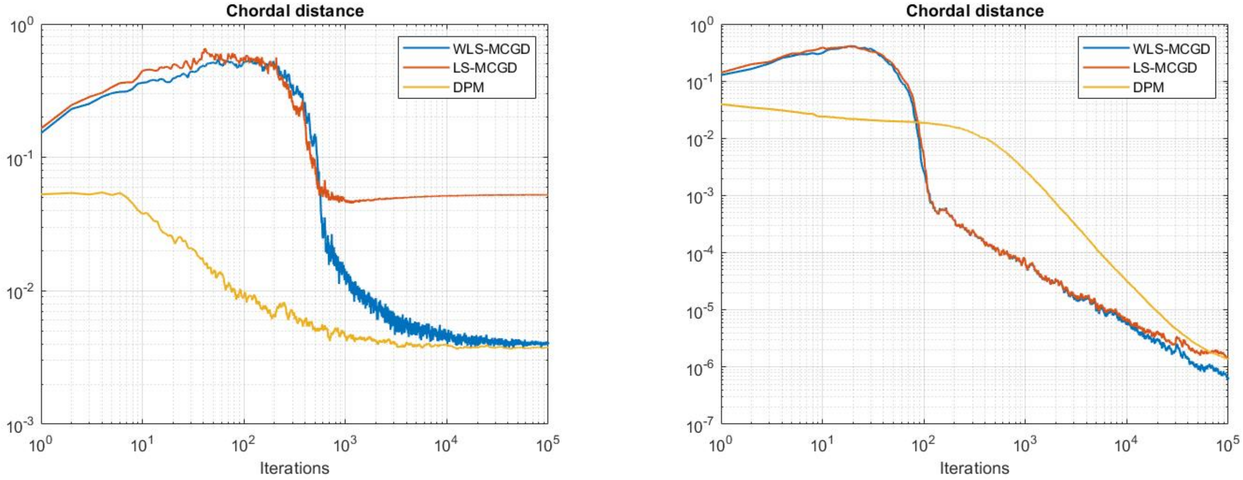
Having arrived to this point of the thesis, we are able to solve with novel ideas and algorithms the two independent problems that we have presented: the Sparse Anomaly Detection and the Subspace Tracking algorithm. Thus, we are now interested in solving both problems at once, for the previously shown signal model. In this sense, we will use the general model shown in the Sparse Anomaly Detection part in this work, which we recall is the following one:

$$\mathbf{s}(n) = \mathbf{H}(n)\mathbf{x}(n) + \boldsymbol{\theta}(n) + \mathbf{w}(n), \tag{118}$$

where now we will consider that the anomalies, $\boldsymbol{\theta}(n)$, are S_0-sparse initially and recalling the dimensions of these vectors: $\boldsymbol{\theta}(n)$, $\mathbf{s}(n)$ and $\mathbf{w}(n)$ (Gaussian vector) are $M \times 1$ vectors, $\mathbf{x}(n)$ (Gaussian vector) is a $D \times 1$ vector and $\mathbf{H}(n)$ is a $M \times D$ matrix, with $M > D$. Also, now all the components of this model will be non-stationary, meaning that they may change as n evolves, except from one restriction: the anomalies must be stationary in a time period N_A. In order to exemplify this feature, we will denote $\{\boldsymbol{\theta}_0, \boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_{N-1}\}$ the sequence of all possible states of $\boldsymbol{\theta}(n)$, which must be S_0-sparse, and $\left\{N_{A_0}, N_{A_1}, ..., N_{A_{N-1}}\right\}$ the temporal period that each state will be held. Subsequently, we are aiming this part to find a Subspace-based anomaly detection algorithm using the developed tools in this work. What is more, in this model we can also consider that there is redundancy in the data, as $M > D$, so if we consider n to be the temporal dimension, we could consider the solution of this problem to be a multimodal data fusion in presence of an anomaly. The problem of multimodal data fusion consists on fusing all the available information, consisting in the M entries of the input vector, into estimations from the data which have reduced dimensionality, D, which in this case we may consider either the estimation of the latent variables or the subspace estimation fitting into this definition.

Note that in this case, we are simplifying the anomaly term, $\boldsymbol{\theta}(n)$, in contrast with the Anomaly Detection part, so we can focus this part to handle the Subspace-based anomaly detection. If we considered the previous kind of anomalies, we would require additional processing which is not the scope of this part. In addition, this simplification also makes sense in terms of the application, as it can model artifacts in MIMO communications, such as interferences or hardware malfunctioning, or also in the GNSS PPP domain this kind of anomalies could be linked with the phase ambiguities in a multisatellite approach. In this way, the model in (118) could be seen as a continuous change in the mean in the temporal dimension n, so it fits the phase discontinuities in the GNSS PPP application and in the case of MIMO communications it could also model antenna failures. In addition, for large values of M, the computational efficiency of the subspace-based approach increases greatly, as in this scenario the computationally costly operations are being avoided or done efficiently, for example by computing $D \times D$ matrix inverses instead of $M \times M$ (necessary in the case of covariance matrix-based signal procesing) among other examples.

If one tries to estimate the subspace in the model in (118) regardless of the anomaly, one would find out that the performance of the subspace estimation is completely harmed by the anomaly. In Figure 31, we show the impact of this anomaly in the subspace estimation for the Data Projection Method and the Manifold Conjugate Gradient Descent algorithms, in an equivalent test to the ones shown in the previous part.



(a) Anomaly.

(b) No anomaly (Figure 23).

Figure 31: Subspace Estimation in presence of an arbitrary anomaly. Chordal norm along the iterations, with $C_1 = 1$ and $C_2 = 10$.

One could compare Figure 31 with the equivalent figure in the previous part, which recalling is Figure 23 and also stated in this figure, and notice inmediately the loss of performance. The two families of algorithms have a related, but different reasoning in why they are failing. Note that both families of algorithms still mantain the transitory phase of the subspace learning, but their performance is worsened by approximately a factor of 10^4, in addition to the fact that they reach a minimum value regardless of the iterations.

On the one hand, we have the MCGD algorithms being based on the Least Squares and Weighted Least Squares estimation of the latent variables. It is widely known that the LS or the WLS solutions are not robust to anomalies in the mean, given the fact that they are the Maximum Likelihood estimator of the mean in a Gaussian distribution, being known for the fact that this statistical distribution does not account for outliers.

On the other hand, the covariance-based algorithms will fail due to the fact that the anomalies change the spectral distribution of the signal. In other words, the subspace spanned by the eigenvectors related to the largest eigenvalues is changed if we do not take into account these outliers.

11 Methodology: Mixing Subspace Tracking and Sparse Anomaly Detection

Our proposed approach to this problem is processing the data in a way that the Subspace Estimation and the Anomaly Detection become decoupled and independent. Our proposed methodology is based on the following equation:

$$\mathbf{P}_H^{\perp}\mathbf{s}(n) = \mathbf{P}_H^{\perp}(n)(\mathbf{H}(n)\mathbf{x}(n) + \boldsymbol{\theta}(n) + \mathbf{w}(n)), \tag{119}$$

where $\mathbf{P}_H^{\perp}(n)$ is the projector of the orthogonal complement of $\mathbf{H}(n)$, we have:

$$\mathbf{s}_{\perp}(n) = \mathbf{P}_H^{\perp}(n)\boldsymbol{\theta}(n) + \mathbf{w}^{\perp}(n), \tag{120}$$

where $\mathbf{w}^{\perp}(n)$ is the equivalent noise. Note that in this way, we can totally cancel out the signal's term and keep the projections onto the orthogonal complement of the signal subspace of the anomalies and the noise, from where the anomalies have most of its power lying on this subspace. This last idea is ensured by the sparsity of the anomalies where statistically their power will cover all the available space, which in general will be much larger than the signal's subspace, being $M \gg D$. In equation (120) we have a similar signal model, as compared to the one in the anomaly detection part, except for the constant term and the projection matrix multiplying the anomaly term. The aim of this operation is to decouple completely the anomaly detection problem from the estimation of the latent varibles $\mathbf{x}(n)$, so in this way we can focus independently and using the tools that we have already explored in the previous part, the problem of subspace estimation and tracking of our signal. As an additional remark, we will be formulating the algorithm for the stationary version of the vectors.

11.1 Subspace-based dual ascent for anomaly detection

Having this modification of the model in (116) in mind, and recalling the procedures from the anomaly detection's part, the objective function that we will solve by applying the dual ascent solution is the following:

$$\hat{\boldsymbol{\theta}} = \arg \min_{\boldsymbol{\theta}, \mathbf{P}_H^{\perp}} ||\boldsymbol{\theta}||_1 \quad \text{s.t } \frac{(\mathbf{s}_{\perp} - \mathbf{P}_H^{\perp}\boldsymbol{\theta})^T \mathbf{K}^{-1} (\mathbf{s}_{\perp} - \mathbf{P}_H^{\perp}\boldsymbol{\theta})}{\mathbf{1}^T \mathbf{K}^{-1} \mathbf{1}} < \gamma, \tag{121}$$

where $\mathbf{K}$ is the equivalent covariance matrix of $\mathbf{w}^{\perp}$. Its unconstrained version, or the Lagrangian of this problem, can be derived as:

$$p(\boldsymbol{\theta}, A, \alpha) = \alpha \left(\frac{(\mathbf{s}_{\perp} - \mathbf{P}_H^{\perp}\boldsymbol{\theta})^T \mathbf{K}^{-1} (\mathbf{s}_{\perp} - \mathbf{P}_H^{\perp}\boldsymbol{\theta})}{\mathbf{1}^T \mathbf{K}^{-1} \mathbf{1}} - \gamma \right) + ||\boldsymbol{\theta}||_1, \tag{122}$$

whose dual problem is easily obtained as:

$$g(\alpha) = \min_{\boldsymbol{\theta}, \mathbf{P}_H^{\perp}} \alpha \left(\frac{(\mathbf{s}_{\perp} - \mathbf{P}_H^{\perp}\boldsymbol{\theta})^T \mathbf{K}^{-1} (\mathbf{s}_{\perp} - \mathbf{P}_H^{\perp}\boldsymbol{\theta})}{\mathbf{1}^T \mathbf{K}^{-1} \mathbf{1}} - \gamma \right) + ||\boldsymbol{\theta}||_1. \tag{123}$$

Now, we can derive the dual ascent updating equations for this problem. In this case, we will need to estimate both the anomaly and the noise's subspace projector, $\mathbf{P}_H^{\perp}$. However, this joint optimization can be done separately. As for the anomaly estimation, we will need to note that in order to get the maximizer $\boldsymbol{\theta}_{opt}^{fitting}$ for the fitting function, we will have to solve an equation of the following kind:

$$\left(\mathbf{P}_H^{\perp} \mathbf{K}^{-1} \mathbf{P}_H^{\perp}\right) \boldsymbol{\theta}_{opt}^{fitting} = \left(\mathbf{P}_H^{\perp} \mathbf{K}^{-1}\right) \mathbf{s}_{\perp}, \tag{124}$$

where we note that the matrix $(\mathbf{P}_H^{\perp} \mathbf{K}^{-1} \mathbf{P}_H^{\perp})$ is not full rank (it has a maximum rank $M - D$, in general) and hence, not invertible. The straight forward solution would be to use the Moore-Penrose pseudoinverse of this matrix, denoted by $(\mathbf{P}_H^{\perp} \mathbf{K}^{-1} \mathbf{P}_H^{\perp})^{\dagger}$, yielding the minimum l_2-norm solution of the equation in (124). Then, applying the proposed modification of the ISTA algorithm previously mentioned, which recalling its expression:

$$\boldsymbol{\theta}_{k+1} = \arg \min_{\boldsymbol{\theta}} ||\boldsymbol{\theta} - \boldsymbol{\theta}_k^{opt}|| + \frac{1}{\alpha} ||\boldsymbol{\theta}||_1, \tag{125}$$

being $\boldsymbol{\theta}_k^{opt}$ the solution of the equation in (124) in the k-th iteration, being:

$$\boldsymbol{\theta}_k^{opt} = (\mathbf{P}_H^{\perp} \mathbf{K}^{-1} \mathbf{P}_H^{\perp})^{\dagger} (\mathbf{P}_H^{\perp} \mathbf{K}^{-1}) \mathbf{s}_{k\perp}.$$

From here, we can directly derive the updating equation for $\boldsymbol{\theta}_k$:

$$\boldsymbol{\theta}_k = S_{\frac{1}{\alpha}}\left((\mathbf{P}_H^{\perp}\mathbf{K}^{-1}\mathbf{P}_H^{\perp})^{\dagger}(\mathbf{P}_H^{\perp}\mathbf{K}^{-1})\mathbf{s}_{k\perp}\right). \tag{126}$$

Finally, recalling the dual ascent algorithm, we can now easily derive all the equations for $\boldsymbol{\theta}_k$, α_k and $\mathbf{P}_{H_k}^{\perp}$, where we can choose any subspace tracking method:

$$\mathbf{P}_{H_k}^{\perp} = \mathrm{STM}\left(\mathbf{s}(k) - \boldsymbol{\theta}_{k-1}\right), \tag{127}$$

$$\mathbf{s}_{\perp}(k) = \mathbf{P}_{H_k}^{\perp}\mathbf{s}(k), \tag{128}$$

$$\boldsymbol{\theta}_k = S_{\frac{1}{\alpha}}\left((\mathbf{P}_{H_k}^{\perp}\mathbf{K}^{-1}\mathbf{P}_{H_k}^{\perp})^{\dagger}(\mathbf{P}_{H_k}^{\perp}\mathbf{K}^{-1})\mathbf{s}_{k\perp}\right), \tag{129}$$

$$\alpha_{k+1} = \max\left(\alpha_k + \mu_k\left(\frac{(\mathbf{s}_{\perp} - \mathbf{P}_{H_k}^{\perp}\boldsymbol{\theta}_k)^T\mathbf{K}^{-1}(\mathbf{s}_{\perp} - \mathbf{P}_{H_k}^{\perp}\boldsymbol{\theta}_k)}{\mathbf{1}^T\mathbf{K}^{-1}\mathbf{1}} - \gamma\right), \varepsilon\right), \tag{130}$$

where now STM$(\cdot)$ is the function that performs one single iteration of a Subspace Tracking method. This problem approach, requires thatμ_k adopts a constant value, even if it do never reaches an stationary solution. This scenario, as we are setting a subspace tracking-like environment assuming non-stationarity in $\mathbf{s}_{\perp}$ and $\mathbf{H}(n)$, this step size must also take a high enough value so the subgradient is continuously having an impact. It must be in the order of $\mu_k = C \sim 10^2$, depending on the coherent time of the anomalies, defined as the amount of samples until there is a change in the anomaly's vector state. In addition, the Subspace Tracking method step size must be set an appropriate constant value for the same reasons. In this way, this algorithm is capable of tracking the signal's subspace, or equivalently the noise's subspace, and the anomalies as long as the Subspace Tracking method and Duality Ascent for Sparse Anomaly Detection limitations are taken into consideration.

11.2 Additional source of error

In this case, we do not have any latent variable to estimate, but intead we have to continuously estimate the signal's subspace orthonormal complement projector, $\mathbf{P}_H^{\perp}(n)$. Note that due to the fact that we will not estimate perfectly this orthonormal projector, we will have an additional source of error in our model:

$$\mathbf{s}_{\perp}(n) = \mathbf{P}_H^{\perp}(n)\boldsymbol{\theta}(n) + \mathbf{w}^{\perp}(n) + \boldsymbol{\epsilon}(n), \tag{131}$$

where $\boldsymbol{\epsilon}(n)$ is a Gaussian distributed $M \times 1$ vector, coming from the residual of $\mathbf{P}_{\hat{H}}^{\perp}(n)\mathbf{H}(n)\mathbf{x}(n)$, being $\mathbf{P}_{\hat{H}}^{\perp}(n)$ the estimation of the orthogonal complement projector of $\mathbf{H}(n)$ projection matrix, $\mathbf{P}_H^{\perp}$. Remember that $\mathbf{x}(n)$ was also Gaussian distributed in the context of subspace tracking, as it is needed that the latent variables change along the iterations in order to be able to identify the subspace. If we study the distributions of the above gaussian vectors, we have that if we consider the following statistical distributions for $\mathbf{x}(n)$ and $\mathbf{w}(n)$:

$$\mathbf{x}(n) \sim \mathcal{N}(\mathbf{0}, \sigma_x^2\mathbf{I}_D),$$

$$\mathbf{w}(n) \sim \mathcal{N}(\mathbf{0}, \mathbf{C}).$$

Then, the equivalent noise and projection error's distribution would be:

$$\boldsymbol{\epsilon}(n) \sim \mathcal{N}\left(\mathbf{0}, \sigma_x^2\mathbf{P}_{\hat{H}}^{\perp}(n)\mathbf{P}_H(n)\mathbf{P}_{\hat{H}}^{\perp}(n)\right),$$

$$\mathbf{w}^{\perp}(n) \sim \mathcal{N}\left(\mathbf{0}, \mathbf{P}_{\hat{H}}^{\perp}(n)\mathbf{C}\mathbf{P}_{\hat{H}}^{\perp}(n)\right),$$

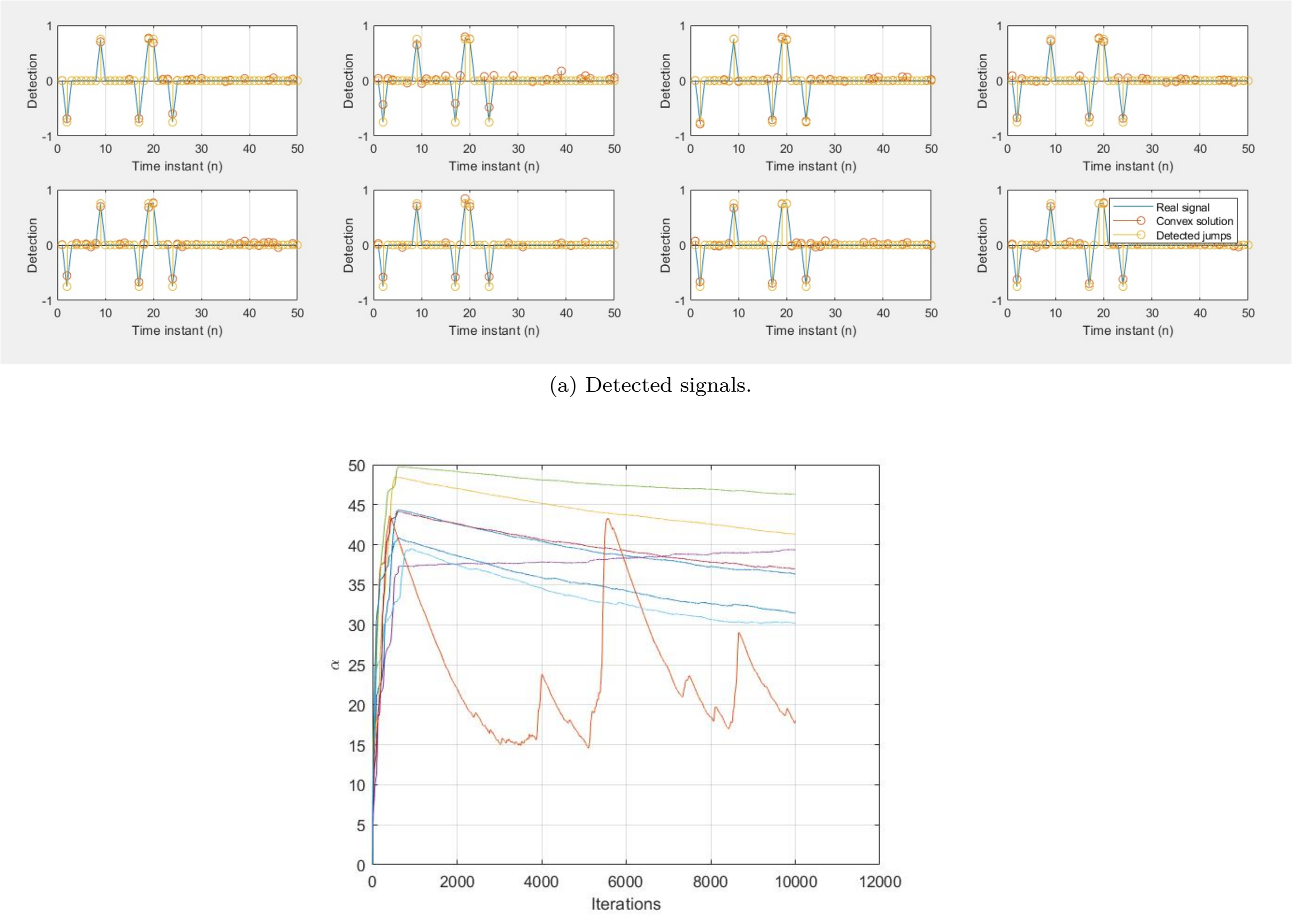
from where we can deduce that the errors corresponding to the subspace processing are an additive source of uncertainty, as being both of the previous vectors independent between them, the resulting covariance is the sum of both covariances.

12 Numerical results

In this section, we will show the performance of the Subspace-based anomaly detection as shown in this work. In order to do so, we will consider a period of N_A samples where the anomaly has a stationary state, $\boldsymbol{\theta}_0$, as depicted in the problem statement of this part. The parameters used in the simulations that we present in this section are the ones showed in the following table:

Parameter	Value
J_o	0.75
$\mathbf{K}$	$0.02 \begin{bmatrix} 1 & -0.5 & 0 & \dots \\ -0.5 & 1 & -0.5 & 0 \\ 0 & -0.5 & 1 & \dots \\ \dots & 0 & \dots & \dots \end{bmatrix}$
M	50
Max. amount of iterations	5000
σ_x^2	1

We plot in Figure 32 eight different experiments on the same anomaly stationary state in order to show that the above algorithm works.



(a) Detected signals.

(b) Dual variable evolution along the iterations.

Figure 32: Subspace-based anomaly detection. Anomaly detection's realizations.

In this last figure, it is evidenced that the performance of the anomaly detection is still kept and is capable to handle a subspace tracking environment. Even though the dual variable evolution along the iterations may be noisy, as seen in some realizations, this due to the fact that the latent variables are also changing throughout the iterations. But still, it is capable of detecting the anomalies quite consistently. In this way, we can say that the algorithm is able to learn and generalize this parameter for new realizations of the data stream.

In contrast, in the next figure we depict the evolution of the chordal norm, averaged in those eight tests. Note that this is a stationary scenario, in terms of the subspace change.

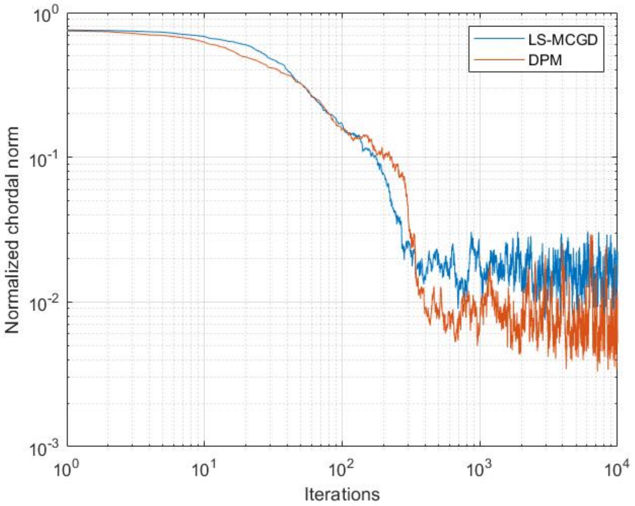


Figure 33: Subspace-based anomaly detection. Frobenius norm of the detected subspace.

In contrast with the anomaly detection features of this detector, if we consider the subspace tracking performance of the Subspace-based Anomaly Detection, we can see that the algorithm has lost subspace estimation capabilities, in the sense that now the Subspace Tracking methods struggle to minimize with the same performance as in the non-anomaly scenario. This is mainly due to the simultaneous optimization of two independent problems, which require more iterations to succeed. Still, this minimum value is enough to be able to detect the anomalies, which quite often are the main focus of these algorithms and applications, such as the GNSS PPP where if one anomaly is lost, the total resolution is degraded by a factor of 2 approximately. This non-perfect subspace estimation also means that the anomaly detection step is less robust to the noise, as there is part of the expected uncertainty that must handle the non-zero projection error, in addition to the degradation of the algorithm due to the persecution error of the tracking environment (i.e. by having a constant step size, possible changes in the anomaly, and so on). Moreover, we cannot define any stopping criteria for this kind of algorithm due to the fact that the algorithm does not know whether the data is stationary or not.

13 Conclusions and future development

As it was mentioned earlier, the main goal of this project is to explore several tools and methodologies in the areas of sparse-aware Signal Processing and the Grassmann Manifold. Through the course of this project, the developed approaches and algorithms have been changing from the start of this project, as we learnt about these topics.

The first set of novel ideas is the Dual Ascent for Anomaly Detection, which could be generalized for more regularizing functions in place of the l_1-norm such as the entropy of the anomaly, or many other information theoretic measures. This is the research problems that we aim to pursue after the finalization of this thesis. The great advantage of all the information theoretic measures is that they have a physical intuition behind them, in contrast with the l_1-norm being greatly justified in the Compress Sensing theory, but it lacks some interpretability.

The second set of novel ideas are the ones related to subspace learning. In this domain we have proposed two new approaches which are the Affine Projection Subspace tracking and the Weighted Least Squares-based Manifold Conjugate Gradient Descent for subspace learning. In this way, the first algorithm was developed with the aid and support of this work and the second one is a proposal to improve the GROUSE algorithm [7], often used in this domain. The subspace learning helped us to get more insights in the Grassmann Manifold theory and made us realize that this theory is necessary to develop more geometry-based signal processing or communication solutions. In this domain, we have assumed a prior knowledge on the signal subspace dimension, also referred in equivalent problems as the intrinsic dimension of the data stream. In a more ambitious approach, we are interested in the estimation of this parameter. To this extent, we have already explored this idea by using Model Order Selection criteria in parallel independent Gaussian channels in a previous work [15], but yet a generalization to MIMO channels is still needed and would fit perfectly in the Subspace Learning part of this work. In addition to this, Grassmann-based methods are yet to be explored in the context of MIMO communications to solve current issues in Large MIMO, from which we are hugely interested, by generalizing those concepts in other applications such as interference cancellation or robust constellations.

Finally, the last novel idea in this project is the proposed approach in Subspace-based anomaly detection. We have shown that it is possible to continuously track a subspace while still detecting anomalies in the signal, up to some restrictions. However, we still have to prove that this algorithm is capable of tracking also the anomaly without any stationary restrictions, while still maintaining the sparsity prior. To conclude, we have shown that we have accomplished the objectives of this project, which can be summed up in part III, where we proposed a novel solution to the Subspace-based anomaly detection problems.

Impact of this project

This project has served as an auxiliary work for other projects. For example, in the domain of anomaly detection, there is a related work whose main objective is to find and online solution for this problem in the form of a Master's thesis which is yet still to be presented. Therefore, this work may help as a comparison because the presented approach in this work may be considered the offline batch processing version of this solution. This solution consists on the utilization of the Viterbi algorithm in the online estimation of the anomalies and the Kalman Filter to estimate the latent variables. This new work is an extension of a solution offered in the ESA's project titled SCIONAV.

On the other hand, in the subspace learning domain, this work served as the foundation and motivation for the Affine Projection Subspace Tracking algorithm, which also served as an auxiliary solution for this work.

Finally, we are planning to write the article versions the Dual Ascent for Anomaly Detection and the Affine Projection Subspace Tracking algorithm, as we think that these are novel and interesting ideas for each domain.

14 Annex

In the following pages, we show two papers that conform the annex of this project. They consist on the paper that we will submit to EUSIPCO, regarding the APST algorithm, and the other one, being the paper that we submitted to ICASSP, related to this work.

14.1 Affine Projection Subspace Tracking (Draft to be submitted to EUSIPCO 2021)

In this related work, we have worked with Marc Vila Insa, in order to derive the Affine Projection for Subspace Tracking algorithm. In this paper, we detailed all the derivation of this algorithm. Still, this draft must be extended for EUSIPCO 2021, which will take place in August 2021.

14.2 Estimation of Information in Parallel Gaussian Channels via Model Order Selection

This work has already been submitted to ICASSP 2020 in May, where we explored the idea of Model Order Selection, being the main objective to estimate the intrinsic dimension of the data. With this idea, it would be possible to relax some of the assumptions that we have in the Subspace Learning domain, as we would not need the prior of the signal subspace' dimension.

Affine Projection Subspace Tracking (Draft to be submitted to EUSIPCO 2021)

Marc Vilà Insa[1], Carlos A. Lopez Molina[1] and Jaume Riba Sagarra[1]

Abstract— **The purpose of this article is to present a novel approach for tracking a signal subspace adaptively. Its development is based on the ideas behind the classical PAST algorithm, as well as the promising yet still underexploited Affine Projection Algorithm. Its properties are then tested in numerical simulations and compared with the ones of the PAST and the stochastic gradient descent applied to subspace learning.**

I. Introduction

Data taken from real world processes present a lot of structure in very high-dimensional spaces. Many times, however, observed phenomena manifest as signals in varying low-dimensional subspaces, while the remaining dimensions are filled with noise that does not provide useful information about them. Dealing with the complete space may become unfeasible, due to computational limitations. Instead, being able to follow the evolution of the subspace of interest would considerably reduce such complexity.

This classical problem has been studied for a long time, and a vast range of solutions has been developed over the years, based on various different approaches. Many of these techniques are centered around the eigendecomposition of a sample correlation matrix or on the singular value decomposition of a data matrix. They provide exceptionally precise estimations at the cost of requiring computationally demanding operations, which makes them unsuitable for adaptive processing scenarios, such as real-time estimation of the direction of arrival of waves reaching an antenna array. To overcome this limitation, various adaptive algorithms have been developed over the years, being the *Projection Approximation Subspace Tracking (PAST)* [1] the most notable case. Following this trend of adaptive subspace learning, this paper presents and derives the *Affine Projection Subspace Tracking (APST)*, a novel technique based on exploiting the ideas behind the *Affine Projection Family* of algorithms [2].

The sources of inspiration for the elaboration of this paper have been numerous, but two of them are of remarkable interest. On the one hand, the book *Theory of Affine Projection Algorithms for Adaptive Filtering* [2] by **Kazuhiko Ozeki** has provided extensive and in-depth information regarding the Affine Projection Family of algorithms, with special attention on their theoretical foundations. On the other, the renowned paper *Projection Approximation Subspace Tracking* [1] by **Bin Yang** has presented an alternative approach to a set of subspace learning problems which has proven to be highly useful for the development of this work.

The organization of this work is as follows. Section II presents the signal and noise model that will be used to explain the subspace tracking problem. Section III introduces some technical tools that will be utilized later on in Section IV, where an adaptive algorithm is described and developed. Finally, Section V will present some numerical simulation results to show the operation properties and performance of this algorithm compared to already existing ones.

The next notations are used in this paper. Matrices and vectors are represented by boldface characters, uppercase for the former and lowercase for the latter. $\mathbf{I}_V$ is the identity matrix of size $V \times V$. The superscripts $\cdot^T$ and $\cdot^+$ denote transposition and the Moore-Penrose pseudoinverse, respectively. The Frobenius norm is expressed with $\|\cdot\|_F$. $\mathrm{E}[\cdot]$ and $\mathrm{Tr}\{\cdot\}$ denote the expectation and trace operators. Finally, $\mathrm{diag}(d_1, \dots, d_N)$ is a diagonal matrix consisting of the diagonal elements d_i.

II. Problem statement

The signal model and problem statement presented next are similar to the ones found in [1]. Let $\mathbf{x}[k] \in \mathbb{R}^N$ be a data vector observed at the kth snapshot. It is considered to be obtained from the following model:

$$\mathbf{x}[k] \triangleq \sum_{r=1}^{R} s_r[k]\mathbf{h}_r[k] + \mathbf{n}[k] = \mathbf{H}[k]\mathbf{s}[k] + \mathbf{n}[k] \tag{1}$$

where $\mathbf{H}[k] \triangleq [\mathbf{h}_1[k] \dots \mathbf{h}_R]$ and $\mathbf{s}[k] \triangleq [s_1[k] \dots s_R[k]]^T$. $\mathbf{H}[k] \in \mathbb{R}^{N\times R}$ is a deterministic matrix whose value might change over time, while $\mathbf{s}[k]$ is a random source vector with correlation matrix $\mathbf{C_s} \triangleq \mathrm{E}\left[\mathbf{s}[k]\mathbf{s}^T[k]\right]$. R is assumed to be known in this paper. Finally, $\mathbf{n}[k]$ is a noise component added to the observation vector and uncorrelated to $\mathbf{s}[k]$. For simplicity, it will be considered AWGN, zero-mean and with a variance equal to σ_n^2 *i.e.* $\mathbf{n}[k] \sim \mathcal{N}(\mathbf{0}, \sigma_n^2\mathbf{I})$.

Assume $\mathbf{H}[k]$ remains constant, *i.e.* $\mathbf{H}[k] \equiv \mathbf{H}$. Let $\mathbf{C_x}$ be the observation correlation matrix, defined as:

$$\mathbf{C_x} \triangleq \mathrm{E}\left[\mathbf{x}[k]\mathbf{x}^T[k]\right] = \mathbf{H}\mathbf{C_s}\mathbf{H}^T + \sigma_n^2\mathbf{I} \tag{2}$$

which can be decomposed into eigenvalues λ_r and their corresponding orthonormal eigenvectors $\mathbf{u}_r$ as $\mathbf{C_x} \triangleq \mathbf{U\Sigma U}^T$, with $\Sigma \triangleq \mathrm{diag}\left(\lambda_1, \dots, \lambda_N\right)$ and $\mathbf{U} \triangleq [\mathbf{u}_1, \dots, \mathbf{u}_N]$. Given that $R < N$, the eigenvalues of $\mathbf{\Sigma}$ can be ordered as follows:

$$\underbrace{\lambda_1 \geq \dots \geq \lambda_R}_{\text{Signal eigenvalues}} > \underbrace{\lambda_{R+1} = \dots = \lambda_N}_{\text{Noise eigenvalues}} = \sigma_n^2 \tag{3}$$

The first R dominant eigenvalues and their associated eigenvectors correspond to the signal, while the $N - R$ last

[1] Universitat Politècnica de Catalunya, Signal Processing and Communications Group

eigenpairs are related to the noise. Having this order in mind, $\mathbf{U}$ can be divided into two column spans:

$$\mathbf{U} \triangleq [\mathbf{U_s}|\mathbf{U_n}] = [\mathbf{u}_1 \dots \mathbf{u}_R|\mathbf{u}_{R+1} \dots \mathbf{u}_N] \tag{4}$$

which define the signal and noise subspaces, orthogonal with each other. $\mathbf{U_s}$ spans the same subspace as $\mathbf{H}$.

Quite often, the observation vector dimension, N, is much larger than the signal dimension, R, which makes it considerably more efficient to work with the lower dimensional signal subspace rather than with the complete space. Many times, the eigenvectors are not necessarily required in order to define this subspace: just finding an equivalent base $\mathbf{W}$ suffices. This is the approach that will be taken in this paper. In the following sections, an adaptive algorithm will be developed, which will be able to estimate the signal subspace and track its movements whenever it changes.

A. *Proposed formulation*

Taking the signal model presented in the previous section, consider the following equation $\mathbf{G}$ dependent on an $N \times R$ matrix $\mathbf{W}$:

$$\begin{aligned}\mathbf{G}(\mathbf{W}) &\triangleq \mathrm{E}\left[(\mathbf{W}^T\mathbf{x})(\mathbf{W}^T\mathbf{x})^T\right] = \mathbf{W}^T\,\mathrm{E}\left[\mathbf{x}\mathbf{x}^\mathbf{T}\right]\mathbf{W} \\ &= \mathbf{W}^T\mathbf{C_x}\mathbf{W} = \mathbf{W}^T\mathbf{U}\boldsymbol{\Sigma}\mathbf{U}^T\mathbf{W}\end{aligned} \tag{5}$$

As shown before, the orthonormal base $\mathbf{U}$ can be split into two submatrices, $\mathbf{U_s}$ and $\mathbf{U_n}$, each one corresponding to the signal and noise subspace bases, respectively.

Let $\tilde{\mathbf{U}}_\mathbf{s}$ be the orthogonal signal subspace base with each component scaled by a factor of 1 over the square root of their respective eigenvalue λ_r, *i.e.* $\tilde{\mathbf{U}}_\mathbf{s} \triangleq \left[\frac{1}{\sqrt{\lambda_1}}\mathbf{u}_1 \dots \frac{1}{\sqrt{\lambda_R}}\mathbf{u}_R\right]$. Alternatively, it can be written as $\tilde{\mathbf{U}}_\mathbf{s} \triangleq \mathbf{U_s}\boldsymbol{\Gamma}$, where $\boldsymbol{\Gamma} \triangleq \mathrm{diag}(\frac{1}{\sqrt{\lambda_1}}, \dots, \frac{1}{\sqrt{\lambda_R}})$. By forcing $\mathbf{W} \equiv \tilde{\mathbf{U}}_\mathbf{s}$, the expression then becomes:

$$\begin{aligned}\mathbf{G}(\tilde{\mathbf{U}}_\mathbf{s}) &= \tilde{\mathbf{U}}_\mathbf{s}^T\left[\mathbf{U_s}\,\mathbf{U_n}\right]\boldsymbol{\Sigma}\begin{bmatrix}\mathbf{U_s}^T \\ \mathbf{U_n}^T\end{bmatrix}\tilde{\mathbf{U}}_\mathbf{s} \\ &= [\boldsymbol{\Gamma}\,\mathbf{0}]\begin{bmatrix}\lambda_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \lambda_N\end{bmatrix}\begin{bmatrix}\boldsymbol{\Gamma} \\ \mathbf{0}\end{bmatrix} = \mathbf{I_R}\end{aligned} \tag{6}$$

In following sections, this development will be useful to set a constraint on an optimization problem, such as:

$$\mathrm{Tr}\left\{\mathbf{W}^H\mathbf{C_x}\mathbf{W}\right\} \equiv R \tag{7}$$

III. Tools used in this work

A. *Affine Projection Algorithms*

The Affine Projection Algorithm (APA) family is a set of adaptive algorithms with many desirable properties and potential, yet it still remains underutilized. These solutions present a configurable compromise between the detection speed of the recursion-based algorithms (e.g. Recursive Least Squares) and the computational simplicity of the stochastic gradient-based ones (e.g. Least Mean Squares).

The core idea behind their design is to exploit the P most recent data samples to improve their tracking and estimation capabilities. From a geometrical point of view, at each time step the algorithm forces the next prediction, $\mathbf{w}[k+1]$, to be contained in the intersection of the hyperplanes defined by the P most recent data samples. This configurable parameter P defines how many hyperplanes intersect, thus it is called *projection order*. This is both a generalization and an improvement over the $NLMS$, which projects the estimation on just a single hyperplane (the one corresponding to the last data sample).

The derivation of the Affine Projection Algorithm can be drawn from many approaches. The most straightforward is to interpret it as a constrained optimization problem. Consider the estimation of an ideal weights vector $\mathbf{w_0}$ such that when multiplied by the samples of a known signal, $\mathbf{x}[k]$, it provides the desired output response $y[k]$ (*i.e.* reference signal) for the P most recent time instants:

$$\mathbf{w}_0^T\mathbf{x}[k-p] = y[k-p]\,,\; p = 0, 1, \dots, P-1 \tag{8}$$

Fulfilling these constraints equates to projecting the weights vector onto the hyperplanes mentioned. What the algorithm is expected to achieve is to minimize the Euclidean distance between the current and next estimate while constrained by the previous requirements. The optimization task may then be expressed as:

$$\boxed{\begin{array}{c}\mathbf{w}[k+1] \triangleq \underset{\mathbf{w}[k+1]}{\arg\min}\,\|\mathbf{w}[k+1]-\mathbf{w}[k]\|^2 \\ \text{subject to} \\ \mathbf{w}^T[k+1]\mathbf{x}[k-p] \equiv y[k-p] \quad , \quad p = 0, ..., P-1\end{array}} \tag{9}$$

Solving this problem produces the *Basic Affine Projection Algorithm (B-APA)*. Many variations of this initial approach exist in the literature, which try to improve its capabilities and reduce the required computational cost. However, they all share the same core principle of reusing past data samples to obtain a better estimation. This idea is the one which, among many others, will be utilized to derive an adaptive subspace learning algorithm in the following sections of the paper.

B. *Chordal Distance*

The *Chordal Distance* or *Procrustes Distance* is a useful metric to compare how different two subspaces are. Given two matrices, $\mathbf{A}$ and $\mathbf{B}$, whose columns span two distinct subspaces, their orthogonal projectors are defined as $\boldsymbol{\Pi}_\mathbf{A} \triangleq \mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ and $\boldsymbol{\Pi}_\mathbf{B} \triangleq \mathbf{B}(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T$, respectively. Then, the chordal distance between those subspaces is defined as:

$$d_C(\mathbf{A}, \mathbf{B}) \triangleq \|\boldsymbol{\Pi}_\mathbf{A} - \boldsymbol{\Pi}_\mathbf{B}\|_F \tag{10}$$

This metric is invariant to rotations, i.e. its value is the same for equivalent subspace bases. It will be one of the major parameters that will be taken into account in Section V when comparing the performance of various subspace tracking algorithms, among others.

IV. Methodology

In this section, the subspace learning problem presented in Section II will be expressed as a constrained optimization problem, in a similar manner as how the author develops the *Regularized Affine Projection Algorithm (R-APA)* in [2]. The resulting adaptive solution is the *Affine Projection Subspace Tracking (APST)* algorithm.

A. Derivation

At every iteration, the goal of the algorithm is to generate a matrix $\mathbf{W}[k+1] \triangleq [\mathbf{w}_1[k]\dots\mathbf{w}_R[k]] \in \mathbb{R}^{N\times R}$ whose columns span an estimation of the signal subspace of a data vector $\mathbf{x}[k]$ (constructed using the model of Section II). To do so, such matrix will be constrained to fulfill the restriction presented in Section II-A:

$$\mathbf{W}[k+1]^T\mathbf{C}_{\mathbf{x}}\mathbf{W}[k+1] \equiv \mathbf{I}_R \tag{11}$$

Since the correlation matrix of $\mathbf{x}[k]$, $\mathbf{C}_{\mathbf{x}}$, is not known, an alternative estimation has to be defined. Thus, for $p = 0, \dots, P-1$:

$$\mathbf{W}^T[k+1]\mathbf{x}[k-p]\mathbf{x}^T[k-p]\mathbf{W}[k+1] \equiv \mathbf{I}_R \tag{12}$$

This solution is reminiscent to the projection onto the intersection of affine hyperplanes used in [2]. In this context, however, instead of minimizing the Euclidean distance between the current and the next estimates, the algorithm will aim at minimizing the Frobenius distance between $\mathbf{W}[k]$ and $\mathbf{W}[k+1]$. Nonetheless, P will be called projection order in the same way.

As a final note before establishing the definitive optimization problem, yet another approximation will be considered, taken from [1], which will heavily simplify its solution in an iterative way:

$$\mathbf{x}[k-p]\mathbf{x}^T[k-p]\mathbf{W}^T[k+1] \approx \mathbf{x}[k-p]\mathbf{x}^T[k-p]\mathbf{W}^T[k]$$

$$\mathbf{Y}[k-p] \triangleq \mathbf{x}[k-p]\mathbf{x}^T[k-p]\mathbf{W}^T[k] \tag{13}$$

With all the presented approximations, an optimization problem can finally be constructed. It is as follows:

$$\boxed{\begin{array}{c} \mathbf{W}[k+1] \triangleq \underset{\mathbf{W}[k+1]}{\arg\min} \|\mathbf{W}[k+1]-\mathbf{W}[k]\|_F^2 \\ \text{subject to} \\ \mathbf{W}^T[k+1]\mathbf{Y}[k-p] \equiv \mathbf{I}_R \quad , \quad p = 0, ..., P-1 \end{array}} \tag{14}$$

In order to solve it, the strategy taken next is to use the *Method of Lagrange Multipliers*. Firstly, it is proceeded to define the *Lagrangian Function*:

$$\begin{aligned} \mathcal{L}\left(\mathbf{W}[k+1]\right) \triangleq{} & \|\mathbf{W}[k+1]-\mathbf{W}[k]\|_F^2 \\ & + \sum_{p=0}^{P-1} \lambda_p \operatorname{Tr}\left\{\mathbf{I}_R - \mathbf{W}^T[k+1]\mathbf{Y}[k-p]\right\} \end{aligned} \tag{15}$$

where λ_p are the *Lagrange multipliers* corresponding to each of the P restrictions.

To find the minimum of $\mathcal{L}$, its gradient in terms of $\mathbf{W}[k+1]$ has to be computed and set to zero, *i.e.* $\nabla_{\mathbf{W}[k+1]}\mathcal{L}\left(\mathbf{W}[k+1]\right) \equiv \mathbf{0}$. Therefore:

$$\begin{aligned} & \nabla_{\mathbf{W}[k+1]}\mathcal{L}\left(\mathbf{W}[k+1]\right) = \\ & \qquad 2\left(\mathbf{W}[k+1]-\mathbf{W}[k]\right) - \sum_{p=0}^{P-1} \lambda_p \mathbf{Y}[k-p] \equiv \mathbf{0} \end{aligned} \tag{16}$$

$$\mathbf{W}[k+1]-\mathbf{W}[k] = \frac{1}{2}\sum_{p=0}^{P-1} \lambda_p \mathbf{Y}[k-p] \tag{17}$$

To derive the final expression of the algorithm, the value of the Lagrange multipliers has to be obtained. Using the same notation as [2], the *regressor block* $\mathbf{X}[k]$ is defined as:

$$\mathbf{X}[k] \triangleq \begin{bmatrix} \mathbf{x}^T[k] \\ \mathbf{x}^T[k-1] \\ \vdots \\ \mathbf{x}^T[k-P+1] \end{bmatrix}$$

Using it, equation 17 can be compactly written as:

$$\mathbf{W}[k+1]-\mathbf{W}[k] = \mathbf{X}^T[k]\mathbf{\Lambda}\mathbf{X}[k]\mathbf{W}[k] \tag{18}$$

where

$$\mathbf{\Lambda} \triangleq \frac{1}{2}\operatorname{diag}\left(\lambda_0, \dots, \lambda_{P-1}\right)$$

Defining the matrix $\bar{\mathbf{Y}}[k] \triangleq \mathbf{X}^T[k]\mathbf{X}[k]\mathbf{W}[k]$, it is easily verifiable that $\mathbf{W}^T[k+1]\bar{\mathbf{Y}}[k] = P \times \mathbf{I}_R$. It will be also useful to define an error matrix $\bar{\mathbf{E}}[k]$ between the current and the next estimates:

$$\bar{\mathbf{E}}[k] \triangleq P \times \mathbf{I}_R - \mathbf{W}^T[k]\bar{\mathbf{Y}}[k] \tag{19}$$

With all these expressions, problem 18 can be finally solved, following these steps:

$$\mathbf{W}[k+1]-\mathbf{W}[k] = \mathbf{X}^T[k]\mathbf{\Lambda}\mathbf{X}[k]\mathbf{W}[k]$$

$$\bar{\mathbf{Y}}^T[k]\left(\mathbf{W}[k+1]-\mathbf{W}[k]\right) = \bar{\mathbf{Y}}^T[k]\mathbf{X}^T[k]\mathbf{\Lambda}\mathbf{X}[k]\mathbf{W}[k]$$

$$\bar{\mathbf{E}}^T[k] = \bar{\mathbf{E}}[k] = \left(\mathbf{X}[k]\bar{\mathbf{Y}}[k]\right)^T\mathbf{\Lambda}\left(\mathbf{X}[k]\mathbf{W}[k]\right)$$

$$\boxed{\mathbf{\Lambda} = \left(\bar{\mathbf{Y}}^T[k]\mathbf{X}^T[k]\right)^+\bar{\mathbf{E}}[k]\left(\mathbf{X}[k]\mathbf{W}[k]\right)^+}$$

By setting the correction matrix $\Delta\mathbf{W}[k]$ as the difference between the current and next estimates, then:

$$\begin{aligned} \Delta\mathbf{W}[k] &\triangleq \mathbf{W}[k+1]-\mathbf{W}[k] \\ &= \mathbf{X}^T[k]\left(\bar{\mathbf{Y}}^T[k]\mathbf{X}^T[k]\right)^+\bar{\mathbf{E}}[k] \end{aligned}$$

For clarity of notation, the product $\mathbf{X}[k]\bar{\mathbf{Y}}[k]$ will be written as $\mathbf{Z}[k]$. Plugging $\Delta\mathbf{W}[k]$ in the general adaptive filter update equation produces the following expression:

$$\begin{aligned} \mathbf{W}[k+1] &= \mathbf{W}[k] + \mu\Delta\mathbf{W}[k] \\ &= \mathbf{W}[k] + \mu\mathbf{X}^T[k]\left(\mathbf{Z}^T[k]\right)^+\bar{\mathbf{E}}[k] \end{aligned} \tag{20}$$

$$\boxed{\mathbf{W}[k+1] = \mathbf{W}[k] + \mu \mathbf{X}^T[k]\left(\mathbf{Z}[k]\mathbf{Z}^T[k]\right)^{-1}\mathbf{Z}[k]\bar{\mathbf{E}}[k]} \tag{21}$$

Expressions of this form entail some numerical issues which have to be addressed, as stated in [2]. More precisely, matrix $\mathbf{Z}[k]\mathbf{Z}^T[k]$ may become *ill-conditioned*, which complicates its inversion. In order to prevent this limitation, a regularization factor has to be added, which will noticeably increase the robustness of the algorithm. Letting $\delta > 0$:

$$\boxed{\mathbf{W}[k+1] = \mathbf{W}[k] + \mu \mathbf{X}^T[k]\left(\mathbf{Z}[k]\mathbf{Z}^T[k] + \delta \mathbf{I}_P\right)^{-1}\mathbf{Z}[k]\bar{\mathbf{E}}[k]} \tag{22}$$

For some applications, it would be of interest to increase the projection order ($P >> R$), which may noticeably increase the computational cost at each time step, since the matrix that has to be inverted is $P \times P$. By applying *Woodbury's Inversion Lemma*, an expression alternative to 22 can be derived, whose inverted matrix will then be $R \times R$:

$$\boxed{\mathbf{W}[k+1] = \mathbf{W}[k] + \mu \mathbf{X}^T[k]\mathbf{Z}[k]\left(\mathbf{Z}^T[k]\mathbf{Z}[k] + \delta \mathbf{I}_R\right)^{-1}\bar{\mathbf{E}}[k]} \tag{23}$$

Summarizing the complete development of the APST, Algorithm 1 contains each step involved in the computation of the estimated subspace base.

```
Data: $\mathbf{x}[k]$
Result: $\mathbf{W}[k]$
Initialization: $\mu > 0$
for $k = 1, \dots, K$ do
    $\mathbf{X}[k] = [\mathbf{x}[k] \ \dots \ \mathbf{x}[k-P+1]]^T$
    $\bar{\mathbf{Y}}[k] = \mathbf{X}^T[k]\mathbf{X}[k]\mathbf{W}[k]$
    $\mathbf{Z}[k] = \mathbf{X}[k]\bar{\mathbf{Y}}[k]$
    $\bar{\mathbf{E}}[k] = P \times \mathbf{I}_R - \mathbf{W}^T[k]\bar{\mathbf{Y}}[k]$
    if $P \leq R$ then
        $\Delta\mathbf{W}[k] = \mathbf{X}^T[k]\left(\mathbf{Z}[k]\mathbf{Z}^T[k] + \delta\mathbf{I}_P\right)^{-1}\mathbf{Z}[k]\bar{\mathbf{E}}[k]$
    else
        $\Delta\mathbf{W}[k] = \mathbf{X}^T[k]\mathbf{Z}[k]\left(\mathbf{Z}^T[k]\mathbf{Z}[k] + \delta\mathbf{I}_R\right)^{-1}\bar{\mathbf{E}}[k]$
    end
    $\mathbf{W}[k+1] = \mathbf{W}[k] + \mu\Delta\mathbf{W}[k]$
end
```

Algorithm 1: Affine Projection Subspace Tracking Algorithm

V. Numerical results

Once the APST algorithm has been developed theoretically, some numerical simulations will be shown in this section. They will demonstrate its applicability and performance in a practical scenario. The signal model used is the same as in Section II:

$$\mathbf{x}[k] \triangleq \mathbf{H}[k]\mathbf{s}[k] + \mathbf{n}[k] \tag{24}$$

The algorithm will attempt to estimate a matrix $\mathbf{W}[k]$ which spans the same subspace as $\mathbf{H}[k]$.

A. Performance metrics

Before displaying the results of the simulations, it would be appropriate to explain and justify which metrics will be used to evaluate the performance of the APST. They can be divided into two distinct categories:

- **Data-centric**: In [1], the author presents a cost function $J(\mathbf{W}[\mathbf{k}])$ which depends on an observation vector $\mathbf{x}[k]$ and proves that it reaches a global minimum when $\mathbf{W}[\mathbf{k}] \equiv \mathbf{U_s}\mathbf{Q}$. $\mathbf{Q}$ is an arbitrary unitary matrix while $\mathbf{U_s}$ contains the R dominant eigenvalues of the data correlation matrix. For this reason, J is a good indicator of how well an algorithm is at estimating the signal subspace of $\mathbf{x}[k]$. Instead of using the same function, a similar *Instantaneous Least Squares* J' alternative is proposed:

$$J'\left(\mathbf{W}[k]\right) \triangleq \left\|\mathbf{x}[k] - \mathbf{W}[k]\mathbf{W}^T[k]\mathbf{x}[k]\right\|^2 \tag{25}$$

- **Subspace-centric**: Since ultimately, the goal of the algorithm is estimating a base that spans the signal subspace, it makes sense to consider a metric that is independent on the observations. The best candidate is the *Chordal distance* between the estimated subspace and the real one on which the signal is projected; in other words: $d_C(\mathbf{H}[k], \mathbf{W}[k])$.

As it will be shown in the simulations, there is no direct correspondence between the two metrics considered.

In order to have some illustrative guidelines to understand how well the APST performs, it will be compared against the the *Exponentially Weighted PAST* and the stochastic gradient descent (*GRAD*) algorithms described in [1] (see APPENDIX). These two solutions have not been chosen arbitrarily: analogous to how the APA was envisioned as a compromise between the simplicity of the LMS and the convergence speed of the RLS, the APST shares the same goal but between the stochastic gradient descent and the PAST, respectively. It is then reasonable to make these comparisons.

B. Global configuration

In the following simulations, a 5-dimensional scenario is considered ($N = 5$) with a 3-dimensional ($R = 3$) gaussian signal ($\mathbf{s}[k] \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_R)$). 30000 time samples will be taken ($K = 30000$). The additive noise will be gaussian as well ($\mathbf{n}[k] \sim \mathcal{N}(\mathbf{0}, 10^{-4}\mathbf{I}_N)$. The results displayed will be the average of 5 simulations.

Regarding the configuration of the three algorithms, the *forgetting factor* of the PAST will be $\beta = 0.999$, while the step size of the gradient descent algorithm will be $\mu_{GRAD} = 1 - \beta = 10^{-3}$. Both the projection order P and step size μ_{APST} of the APST are configurable parameters to tune its performance, and as such, will be indicated on every test.

C. Stationary setting

In the first scenario, a stationary setting is considered; that is $\mathbf{H}[k] \equiv \mathbf{H}$. Two variations of the test are presented: one setting $P = 3$ and trying different values of μ_{APST} (figures 1 and 2) and the other setting $\mu_{APST} = 10^{-3}$ and varying the projection order (figures 3 and 4).

Looking at Figure 1, it is clear that, while increasing the APST step size results in a slightly higher error floor, the trade off is positive, since a considerable amount of convergence speed is obtained by sacrificing very few precision.

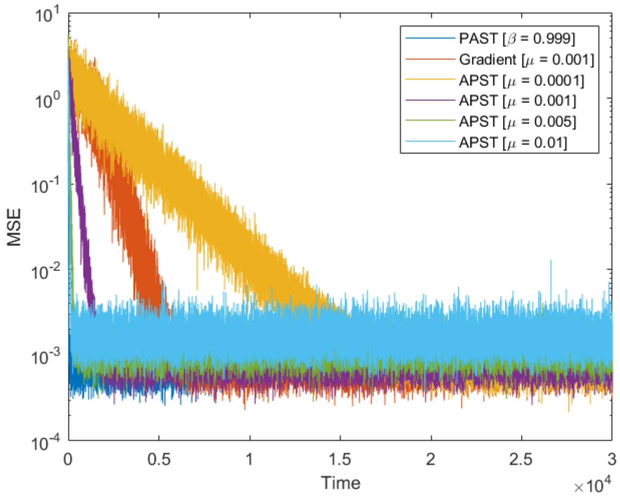


Fig. 1. PAST instantaneous cost function ($P = 3$)

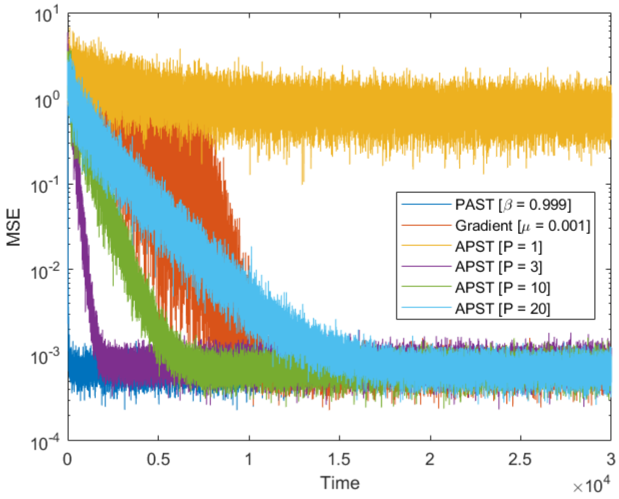


Fig. 3. PAST instantaneous cost function ($\mu_{APST} = 10^{-3}$)

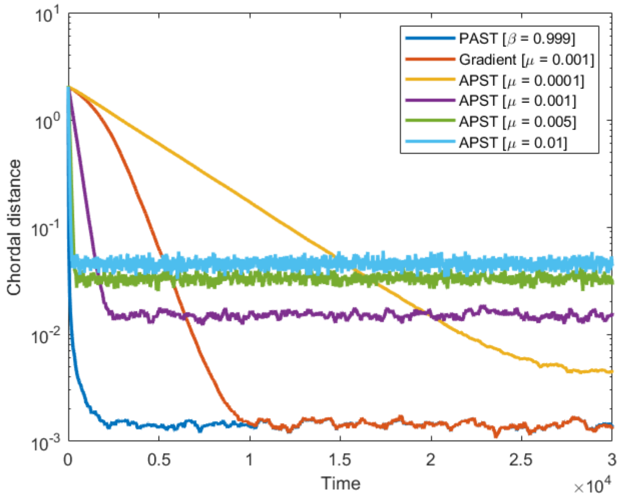


Fig. 2. Chordal distance ($P = 3$)

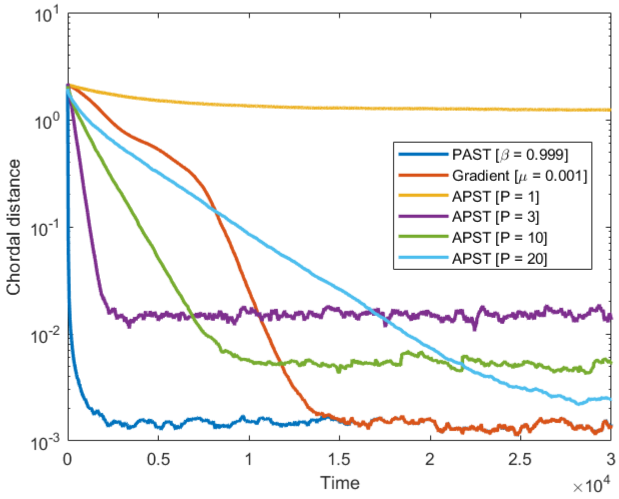


Fig. 4. Chordal distance ($\mu_{APST} = 10^{-3}$)

This, however, is not translated to the chordal distance, on Figure 2, where it is seen that a faster algorithm estimates the subspace more poorly. Furthermore, the step size has to be reduced significantly to reach the same chordal distance that the PAST and GRAD provide.

Changing the projection order of the APST results in a similar effect. Regarding the cost function (Figure 3), moving from no hyperplane intersection ($P = 1$) to a projection order of 3 produces an enormous improvement in speed and error reduction. Further augmenting this value results in almost the same estimates but at the cost of slower convergence. On the contrary, this trade-off is more reasonable from the chordal distance point of view: the error floor descends notably for higher projection orders.

D. Sudden subspace change

In the second scenario, a sudden change of the signal subspace is applied at the sample 15000. The APST is configured with $\mu_{APST} = 0.001$ and $P = 3$.

This test emphasizes the strengths of the APST. Looking at Figure 5, when the subspace change occurs, the PAST algorithm takes almost the same amount of steps as the GRAD does to track it back. In other words, its performance is severely affected by sudden changes on the observations. In that sense, the APST is more robust, as its speed is not affected at all when the disturbance occurs, outperforming the other two algorithms. On the chordal distance plot (Figure 6), the same phenomenon is observed, although the error floor is higher for the APST, as explained in the stationary scenario.

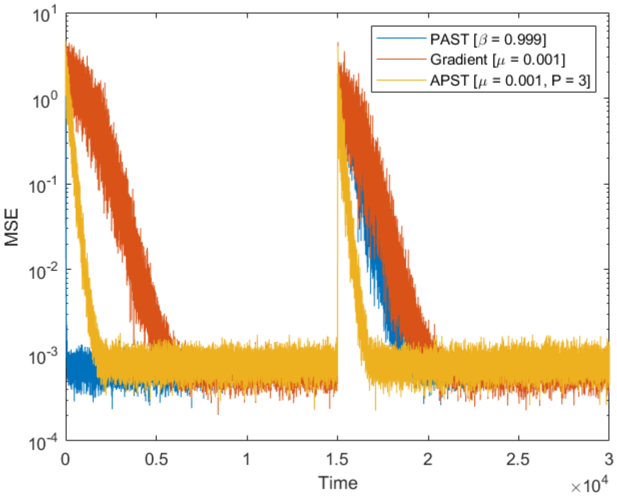


Fig. 5. PAST instantaneous cost function

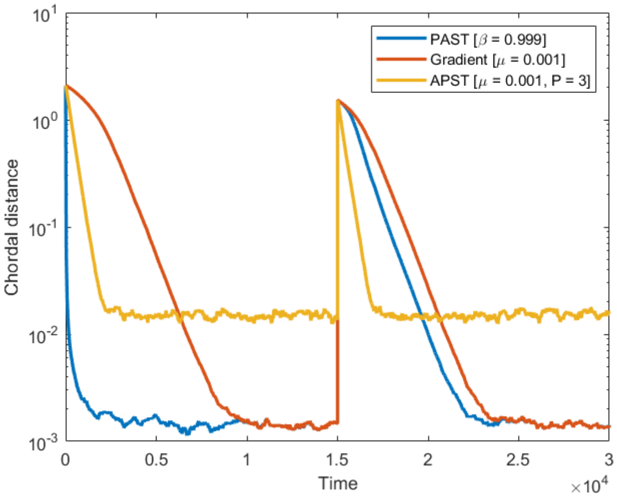


Fig. 6. Chordal distance

VI. Conclusions

In this work, a novel subspace learning algorithm has been introduced and developed, based on the Affine Projection Algorithm and the signal subspace interpretation that motivated the derivation of the PAST in its original paper [1] from 1995. The resulting algorithm, the Affine Projection Subspace Tracking, provides an adaptive solution to estimating the signal subspace from some observation data. As proven in numerical simulations, the APST is able to track the changes produced on the subspace over time, while offering a higher degree of robustness than classical and better known alternatives.

With the developments presented, many future lines of research have arisen. An extension for complex valued data and in-depth analysis of its convergence properties are the most straightforward ones. Testing the capabilities of the APST in a real-world scenario is also of major interest. Another proposal might be implementing the APST based on different variations of the basic APA that provide desirable features, such as the *Fast APA (F-APA)*, which reduces its computational complexity, or the *Variable Step-Size APA (VSS-APA)*, which aims at improving both convergence speed and accuracy. The range of possibilities is vast.

Appendix

A. Additional algorithms used in this work

```
Data: $\mathbf{x}[k]$
Result: $\mathbf{W}[k]$
Initialization: $\mathbf{W}[1] = \begin{bmatrix} \mathbf{I}_R \\ \mathbf{0} \end{bmatrix}$, $0 < \mu << 1$
for $k = 1, \ldots, K$ do
    $\mathbf{y}[k] = \mathbf{W}^T[k]\mathbf{x}[k]$
    $\boxed{\mathbf{W}[k+1] = \mathbf{W}[k] + \mu\left(\mathbf{x}[k] - \mathbf{W}[k]\mathbf{y}[k]\right)\mathbf{y}^T[k]}$
end
```

Algorithm 2: Stochastic gradient descent subspace tracking algorithm

```
Data: $\mathbf{x}[k]$
Result: $\mathbf{W}[k]$
Initialization:
  $\mathbf{P}[1] = \mathbf{I}_R$, $\mathbf{W}[1] = \begin{bmatrix} \mathbf{I}_R \\ \mathbf{0} \end{bmatrix}$, $0 < \beta < 1$
for $k = 1, \ldots, K$ do
    $\mathbf{y}[k] = \mathbf{W}^T[k]\mathbf{x}[k]$
    $\mathbf{h}[k] = \mathbf{P}[k]\mathbf{y}[k]$
    $\mathbf{g}[k] = \frac{\mathbf{h}[k]}{\beta + \mathbf{y}^T[k]\mathbf{h}[k]}$
    $\mathbf{P}[k+1] = \frac{1}{\beta}\left(\mathbf{P}[k] - \mathbf{g}[k]\mathbf{h}^T[k]\right)$
    $\mathbf{e}[k] = \mathbf{x}[k] - \mathbf{W}[k]\mathbf{y}[k]$
    $\boxed{\mathbf{W}[k+1] = \mathbf{W}[k] + \mathbf{e}[k]\mathbf{g}^T[k]}$
end
```

Algorithm 3: Exponentially weighted PAST algorithm

References

[1] B. Yang, "Projection Approximation Subspace Tracking," *IEEE Transactions on Signal Processing*, vol. 43, no. 1, pp. 95–107, 1995.

[2] K. Ozeki, *Theory of Affine Projection Algorithms for Adaptive Filtering*. Springer, 2016.

[3] L. Balzano, R. Nowak, and B. Recht, "Online identification and tracking of subspaces from highly incomplete information," in *2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2010, pp. 704–711.

[4] D. R. Fuhrmann, "A geometric approach to subspace tracking," in *Conference Record of the Thirty-First Asilomar Conference on Signals, Systems and Computers (Cat. No.97CB36136)*, vol. 1, 1997, pp. 783–787 vol.1.

[5] B. Yang, "Subspace tracking based on the projection approach and the recursive least squares method," in *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, 1993, pp. 145–148 vol.4.

[6] Y. C. Eldar, A. O. Hero III, L. Deng, J. Fessler, J. Kovacevic, H. V. Poor, and S. Young, "Challenges and open problems in signal processing: Panel discussion summary from icassp 2017 [panel and forum]," *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 8–23, 2017.

[7] A. Valizadeh, M. Karimi, and E. Member, “Fast Subspace Tracking Algorithm Based on the Constrained Projection Approximation,” *EURASIP Journal on Advances in Signal Processing*, vol. 2009, 2009.

[8] A. Valizadeh, R. Mohammadian, A. Rafiei, and A. Rafati, “A novel algorithm for signal subspace tracking based on a new subspace information criterion,” in *2007 6th International Conference on Information, Communications Signal Processing*, 2007, pp. 1–4.

[9] A. Gonzalez, M. Ferrer, F. Albu, and M. De Diego, “Affine projection algorithms: Evolution to smart and fast algorithms and applications,” *European Signal Processing Conference*, no. 9, pp. 1965–1969, 2012.

[10] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. The Johns Hopkins University Press, 1996.

[11] S. J. Axler, *Linear Algebra Done Right*, ser. Undergraduate Texts in Mathematics. New York: Springer, 1997. [Online]. Available: http://linear.axler.net/

[12] T. Adali and S. Haykin, *Subspace Tracking for Signal Processing*. Wiley-IEEE Press, 2010, pp. 211–270.

[13] A. Edelman, T. A. Arias, and S. T. Smith, “The geometry of algorithms with orthogonality constraints,” *SIAM Journal on Matrix Analysis and Applications*, vol. 20, no. 2, pp. 303–353, 1998.

[14] J. Zhang, G. Zhu, R. W. Heath, and K. Huang, “Grassmannian Learning: Embedding Geometry Awareness in Shallow and Deep Learning,” *IEEE Signal Processing Magazine*, pp. 1–18, 2018.

[15] J.-P. Delmas, *Subspace tracking for signal processing*. Wiley-IEEE Press, 2010.

Estimation of Information in Parallel Gaussian Channels via Model Order Selection

Carlos Alejandro López, Ferran de Cabrera, *Student Member, IEEE*, and Jaume Riba, *Senior Member, IEEE*
Signal Theory and Communications Department, Technical University of Catalonia (SPCOM/UPC)
{carlos.alejandro.lopez, ferran.de.cabrera, jaume.riba}@upc.edu

***Abstract*—We study the problem of estimating the overall mutual information in M independent parallel discrete-time memory-less Gaussian channels from N independent data sample pairs per channel (inputs and outputs). We focus on the case where the number of active channels L is sparse in comparison with the total number of channels ($L \ll M$), for which the direct application of the *maximum likelihood* principle is problematic due to *overfitting*, especially for moderate to small N. For this regime, we show that the bias of the mutual information estimate is reduced by resorting to the *minimum description length* (MDL) principle. As a result, simple pre-processing based on a per-channel threshold on the empirical *squared correlation coefficient* is required with a fixed threshold that monotonically decreases with N as $1 - N^{-1/N}$, for $N \geq 4$. The resulting improvement is shown in terms of the estimated information bias.**

***Index Terms*—Min. Description Length (MDL), Bayesian Info. Criterion (BIC), Locally Most Powerful Invariant Test (LMPIT), Maximum Likelihood (ML), Squared Pearson Coefficient, Mutual Inf. (MI), Generalized Likelihood Ratio Test (GLRT).**

I. Introduction

A general and important problem in the field of multivariate statistical analysis [5] is testing whether two M-dimensional Gaussian vectors are uncorrelated or not. It has been shown in [7] that the Locally Most Powerful Invariant Test (LMPIT) for this problem is given by the Frobenius norm of the *sample coherence matrix*. This fundamental test is given by the sum of squared canonical correlations, which are the squared Pearson coefficients of virtual independent parallel channels given by the canonical coordinates.

In the case of data with unknown statistics, a more challenging problem is estimating the mutual information between two sources and, more generally, the so-called *universal* information measures (see [16] for methods and applications concerned with this field). This line of research finds numerous applications in data science and machine learning. Recently, the problem of estimating information has been linked in [2] with the aforementioned problem of coherence estimation by mapping the bivariate data onto a high-dimensional feature space based on the *empirical characteristic function*. In particular, the Frobenius norm of the coherence matrix computed after this high-dimensional mapping converges with M to the so-called *squared-loss* mutual information [14].

In all the aforementioned applications, the model consisting of parallel-channels with independent information per channel plays an important role in the process of simplification and deep understanding of the original problems. Note that the independence assumption is very useful in modeling many wireless communications scenarios, and it appears in a variety of areas such as radar, multitone transmissions and multi-antenna schemes [15]. As a more direct example, the independence arises as well in the frequency-domain representation of stationary time-series, where the canonical correlations coincide asymptotically (w.r.t. data size) with the squared roots of the magnitude squared coherence spectrum [11].

One of the problems to be faced when working with high-dimensional data is the fact that, in most practical situations, only few L components are correlated among the large amount of M parallel virtual channels. The necessity of detecting the presence of a sparse correlated subset of components emerge naturally in numerous scenarios (see [1], [6] and references therein for a motivation). This means that the high-dimensional data tend to exhibit a low-rank structure irrespective of the application. This paper focuses on that scenario while holding the parallel channel model for simplicity. The purpose is to show that simple sparse-aware detectors and estimators can be obtained via the well-known Minimum Description Length (MDL) principle proposed by Rissanen for model order selection [8] [9], which coincides with the Bayesian Information Criterion (BIC) by Schwarz (see [13] for an excellent overview). We show that the MDL principle applied to the problem of estimating information improves both the LMPIT and the ML estimator.

Prior work on the application of the MDL principle can also be found in [12], [10] in the context of Principal Component Analysis (PCA) and Canonical Correlation Analysis (CCA) as a means to handle the small sample support problem. While [10] and [12] focus on rank-reduction, the present paper moves the goal to the estimation and detection of information, although focusing on parallel channels for simplicity.

II. Maximum likelihood estimation of mutual information

Let's consider a set of M mutually independent pairs of sequences given by $x_m(n)$ and $y_m(n)$, with $m = 1, 2, \ldots, M$, from which we have N i.i.d. samples, with $n = 1, 2, \ldots, N$. The m-th pair is represented by the N-length 2×1 vector sequence $\mathbf{z}_m(n)$ defined as

$$\mathbf{z}_m(n) = \left[\begin{array}{c} x_m(n) \\ y_m(n) \end{array} \right]. \tag{1}$$

This work is supported by projects TEC2016-76409-C2-1-R (WINTER), Ministerio de Economia y Competividad, Spanish National Research Plan, and 2017 SGR 578 - AGAUR, Catalan Government.

In the case of zero-mean Gaussian signals, the mutual information of the m-th pair is given by

$$I(x_m; y_m) = -\frac{1}{2} \ln \det \mathbf{C}_m, \tag{2}$$

where $\mathbf{C}_m$ is the coherence matrix associated to the m-th channel defined as

$$\mathbf{C}_m = \mathbf{D}_m^{-1/2} \mathbf{R}_m \mathbf{D}_m^{-1/2}, \tag{3}$$

and matrices

$$\mathbf{R}_m = E\left[\mathbf{z}_m(n) \mathbf{z}_m^T(n)\right] \tag{4}$$

$$\mathbf{D}_m = \text{diag}(\mathbf{R}_m) = \begin{bmatrix} v_{x,m} & 0 \\ 0 & v_{y,m} \end{bmatrix} \tag{5}$$

are, respectively, the 2×2 autocorrelation matrix and a diagonal matrix containing the two, non-zero variances. Note that

$$\mathbf{C}_m = \begin{bmatrix} 1 & \rho_m \\ \rho_m & 1 \end{bmatrix}, \tag{6}$$

where ρ_m (with $-1 < \rho_m < 1$) is the Pearson coefficient associated to signals $x_m(n)$ and $y_m(n)$. It is clear that the mutual information $I(x_m; y_m)$ depends solely on three free parameters, namely $v_{x,m}$, $v_{y,m}$ and $r_{xy,m}$, being $r_{xy,m}$ the cross correlation between x_m and y_m. We can relate these three parameters as

$$\rho_m = \frac{r_{xy,m}}{\sqrt{v_{x,m} v_{y,m}}}. \tag{7}$$

As the pairs are independent, the overall mutual information is given by the sum of pairwise mutual information values:

$$I(\mathbf{x}; \mathbf{y}) = \sum_{m \in \mathcal{S}_M} I(x_m; y_m) = -\frac{1}{2} \sum_{m \in \mathcal{S}_M} \ln(1 - \rho_m^2) \tag{8}$$

with $\mathcal{S}_M = \{1 : M\}$.

The objective is estimating $I(\mathbf{x}; \mathbf{y})$ from the available data under the prior knowledge that some of them provide no information, that is, it exists some finite $L \leq M$ such that $\rho_m = 0$ for $m \notin \mathcal{S}_L$, where $\mathcal{S}_L$ is a set of integers indexing the active channels, with cardinality $|\mathcal{S}_L| = L$. From the above exposition, it is clear that we are in front of a parametric formulation of the problem of mutual information estimation. Effectively, the finite number of (continuous) parameters of the problem is equal to $3L$, and L is also a (discrete) parameter to be estimated from the data (the model order).

We want to obtain consistency of the resulting estimate as both $N \to \infty$ and $M \to \infty$. To this end, the model order selection of $L \leq M$ is mandatory in order to avoid an uncontrolled number of parameters to be estimated, which would prevent from achieving the desired consistency due to overfitting.

The ML estimation of $\mathbf{R}_m$ can be formulated as follows. The log-likelihood function associated to the overall multichannel data staked at the columns of $\mathbf{X}$, $\mathbf{Y}$ and $\mathbf{Z}_m$, and conditioned to the complete set $\{\mathbf{R}_m\}_{m \in \mathcal{S}_M}$, is given by:

$$\ln p(\mathbf{X}; \mathbf{Y} | \{\mathbf{R}_m\}_{m \in \mathcal{S}_M}) = \sum_{m \in \mathcal{S}_M} \ln p(\mathbf{Z}_m | \mathbf{R}_m), \tag{9}$$

where the log-likelihood function associated to the m-th channel is:

$$\ln p(\mathbf{Z}_m | \mathbf{R}_m) = -\frac{1}{2} \sum_{n=1}^{N} \left(\ln\left(2\pi \det \mathbf{R}_m\right) + \mathbf{z}_m^T(n) \mathbf{R}_m^{-1} \mathbf{z}_m(n) \right). \tag{10}$$

Equivalently:

$$\ln p(\mathbf{Z}_m | \mathbf{R}_m) = -\frac{N}{2} \left(\ln\left(2\pi \det \mathbf{R}_m\right) + \text{tr}(\mathbf{R}_m^{-1} \hat{\mathbf{R}}_m) \right), \tag{11}$$

where $\hat{\mathbf{R}}_m$, for $m \in \mathcal{S}_L$, is the per-channel sample covariance matrix:

$$\hat{\mathbf{R}}_m = \frac{1}{N} \mathbf{Z}_m \mathbf{Z}_m^T, \tag{12}$$

which is known [4] to be the maximizer of $\ln p(\mathbf{Z}_m | \mathbf{R}_m)$ with respect to $\mathbf{R}_m$ (that is, the sample covariance matrix is the ML estimate of the covariance matrix for Gaussian signals). For $m \notin \mathcal{S}_L$, however, the ML estimate of $\hat{\mathbf{R}}_m$ is

$$\hat{\mathbf{R}}_m = \frac{1}{N} \text{diag}(\mathbf{Z}_m \mathbf{Z}_m^T) = \hat{\mathbf{D}}_m \tag{13}$$

as we have the prior-knowledge that $\rho_m = 0$, although nothing is known about the variances. Substituting the ML estimates on the log-likelihood function yields the log-likelihood function conditioned only to the active set knowledge:

$$\begin{aligned} &\ln p(\mathbf{X}; \mathbf{Y} | \mathcal{S}_L) = \\ &-\frac{N}{2} \left(\sum_{m \in \mathcal{S}_L} \ln\left(2\pi \det \hat{\mathbf{R}}_m\right) + 2 \sum_{m \in \mathcal{S}_L} \text{tr}(\hat{\mathbf{R}}_m^{\text{-}1} \hat{\mathbf{R}}_m) \right. \\ &\left. + \sum_{m \notin \mathcal{S}_L} \ln\left(2\pi \det \hat{\mathbf{D}}_m\right) + \sum_{m \notin \mathcal{S}_L} \text{tr}(\hat{\mathbf{D}}_m^{\text{-}1} \hat{\mathbf{R}}_m) \right). \end{aligned} \tag{14}$$

Note that

$$\text{tr}(\hat{\mathbf{R}}_m^{\text{-}1} \hat{\mathbf{R}}_m) = \text{tr}(\mathbf{I}_2) = 2, \tag{15}$$

and

$$\text{tr}(\hat{\mathbf{D}}_m^{\text{-}1} \hat{\mathbf{R}}_m) = \text{tr}(\hat{\mathbf{D}}_m^{\text{-}1/2} \hat{\mathbf{R}}_m \hat{\mathbf{D}}_m^{\text{-}1/2}) = \text{tr}(\mathbf{C}_m) = 2. \tag{16}$$

Therefore, we have

$$\begin{aligned} \ln p(\mathbf{X}; \mathbf{Y} | \mathcal{S}_L) = -\frac{N}{2} &\left(\sum_{m \in \mathcal{S}_L} \ln\left(2\pi \det \hat{\mathbf{R}}_m\right) + 2L \right. \\ &\left. + \sum_{m \notin \mathcal{S}_L} \ln\left(2\pi \det \hat{\mathbf{D}}_m\right) + 2(M - L) \right) \end{aligned} \tag{17}$$

$$\begin{aligned} = -\frac{N}{2} &\left(\sum_{m \in \mathcal{S}_L} \ln\left(2\pi \det \hat{\mathbf{R}}_m\right) \right. \\ &\left. + \sum_{m \notin \mathcal{S}_L} \ln\left(2\pi \det \hat{\mathbf{D}}_m\right) + 2M \right) \end{aligned} \tag{18}$$

$$= -\frac{N}{2} \left(\sum_{m \in \mathcal{S}_L} \ln \det \hat{\mathbf{R}}_m + \sum_{m \notin \mathcal{S}_L} \ln \det \hat{\mathbf{D}}_m + c \right), \tag{19}$$

where $c = (2 + \ln(2\pi)) M$. Since $\det \hat{\mathbf{R}}_m = \det(\hat{\mathbf{D}}_m \hat{\mathbf{C}}_m)$, then

$$\ln p(\mathbf{X}; \mathbf{Y} | \mathcal{S}_L) = -\frac{N}{2} \left(\sum_{m \in \mathcal{S}_L} \ln \det(\hat{\mathbf{D}}_m \hat{\mathbf{C}}_m) + \right.$$

$$+ \sum_{m \notin \mathcal{S}_L} \ln \det \hat{\mathbf{D}}_m + c \Bigg) \tag{20}$$

$$= -\frac{N}{2} \Bigg(\sum_{m \in \mathcal{S}_L} \ln \det \hat{\mathbf{D}}_m + \sum_{m \in \mathcal{S}_L} \ln \det \hat{\mathbf{C}}_m + \sum_{m \notin \mathcal{S}_L} \ln \det \hat{\mathbf{D}}_m + c \Bigg) \tag{21}$$

$$= -\frac{N}{2} \left(\sum_{m \in \mathcal{S}_M} \ln \det \hat{\mathbf{D}}_m + \sum_{m \in \mathcal{S}_L} \ln \det \hat{\mathbf{C}}_m + c \right). \tag{22}$$

Note that the positivity of $\det \hat{\mathbf{D}}_m$ in (22) is ensured with probability 1 because we consider non-null variances, and the positivity of $\det \hat{\mathbf{C}}_m$ is ensured with probability 1 as a result of the Schwarz inequality and the fact that $0 \leq \hat{\rho}_m^2 < 1$. Ignoring additive constants that do not depend on the unknown order L, we have:

$$-\ln p(\mathbf{X}; \mathbf{Y} | \mathcal{S}_L) = \frac{N}{2} \sum_{m \in \mathcal{S}_L} \ln \det \hat{\mathbf{C}}_m + \text{const.} \tag{23}$$

Finally, and more clearly,

$$-\ln p(\mathbf{X}; \mathbf{Y} | \mathcal{S}_L) = -N \hat{I}_{ML}(\mathbf{x}; \mathbf{y} | \mathcal{S}_L) + \text{const,} \tag{24}$$

where, in view of (8) and from the invariance property of ML,

$$\hat{I}_{ML}(\mathbf{x}; \mathbf{y} | \mathcal{S}_L) = -\frac{1}{2} \sum_{m \in \mathcal{S}_L} \ln(1 - \hat{\rho}_m^2) \tag{25}$$

is the ML estimate of mutual information assuming that only L channels within the set $\mathcal{S}_L$ provide non-null information.

III. Incorporation of the BIC rule

Assume that L is unknown and should be estimated. Assume for clarity that $\hat{\rho}_m^2 > \hat{\rho}_{m'}^2$ for $m' > m$ (this will become irrelevant later on). Under this assumption, the negative log-likelihood function for selecting L in (24) is a $\cup$ convex, non-increasing function of L, which would then yield $\hat{L} = M$ as the optimal value. This is the well-known problem of model order selection: the ML rule yields to assume maximum complexity of the data. In general, to avoid overfitting, the BIC rule for model order selection incorporates a penalty term of the form $(L/2) \ln N$ to the joint likelihood function conditioned to a given model complexity [13]. Applying the idea to the result obtained in (24), the final function to be minimized against L becomes:

$$\text{BIC}(L) = -\hat{I}_{ML}(\mathbf{x}; \mathbf{y} | \mathcal{S}_L) + \frac{L \ln N}{2N}. \tag{26}$$

Clearly, the right side term in (26) increases linearly with the model complexity L, with a rate that goes to zero as N goes to infinity, such that the selection of active channels becomes more restrictive for small N and more permissive for large N. The minimizer is now

$$\hat{L} = \arg \min_{L=1,2,\ldots,M} \text{BIC}(L). \tag{27}$$

In the particular application of the BIC rule in this paper, it is possible to further simplifying the computation of $\hat{L}$ by means of the following argument, which is not possible in other problems. Note that the difference between two consecutive trial values of the BIC indicator is:

$$\triangle(L) = \text{BIC}(L) - \text{BIC}(L-1) = \frac{1}{2} \ln \left(1 - \hat{\rho}_L^2\right) + \frac{\ln N}{2N}. \tag{28}$$

If $\triangle(L) < 0$, we need to keep increasing L to minimize $\text{BIC}(L)$. Otherwise, we stop searching. From (28), this observation implies assigning channel m as active if and only if

$$-\frac{1}{2} \ln(1 - \hat{\rho}_m^2) > \frac{\ln N}{2N}, \tag{29}$$

which yields to M per-channel independent decisions such that m is declared active if and only if

$$\hat{\rho}_m^2 > 1 - N^{-1/N}. \tag{30}$$

Note that since the above rule implies making independent decisions per channel, the ordering of the channels is in fact not required.

Summarizing, the BIC rule applied to the problem of estimating information yields:

$$\hat{I}_{BIC}(\mathbf{x}; \mathbf{y}) = -\frac{1}{2} \sum_{m \in \mathcal{S}_M} \ln \left(1 - \hat{\rho}_m^2 1_{(\hat{\rho}_m^2 > 1 - N^{-1/N})}\right), \tag{31}$$

as the final regularized estimate of information, where 1_a is the indicator function such that $1_a = 1$ if the event a is true and $1_a = 0$ otherwise.

Finally, from the $\hat{I}_{BIC}$ estimator, a GLRT detector [3] of the presence of information can be formulated as declaring the presence of information if

$$\hat{I}_{BIC}(\mathbf{x}; \mathbf{y}) > \gamma_{BIC}, \tag{32}$$

where $\gamma_{BIC} > 0$ is some threshold designed to achieve the specified false alarm probability. This detector is expected to perform better than the GLRT $\left(\hat{I}_{ML}(\mathbf{x}; \mathbf{y} | \mathcal{S}_M) > \gamma_{GLRT}\right)$ and the LMPIT $\left(\sum_{m \in \mathcal{S}_M} \hat{\rho}_m^2 > \gamma_{LMPIT}\right)$.

IV. Simulation results

In this section we show the performance of the BIC estimator in (31) in contrast to the ML estimator in (25) for $L = M$. The main phenomena that we are solving is overfitting and thus we will focus on studying the bias of both kinds of estimators. In order to do so, we focus our study on two scenarios: observing the evolution of the bias with M and studying it with N. We consider two modifications of the proposed threshold in (30) that we formulate as

$$\lambda_1 = 1 - N^{-1/N}, \;\; \lambda_2 = 4\lambda_1, \;\; \lambda_3 = \frac{1}{4}\lambda_1. \tag{33}$$

The main goal of introducing these new thresholds is to study the optimality of the found threshold. For instance, we expect λ_2 to be more robust to an increasing number of channels M since it tends to discard more channels than the other thresholds, but to have little robustness to lower values of N since worse estimates of $\hat{\rho}_m^2$ increase the probability of labeling as inactive an actual active channel. On the other hand, λ_3 acts in the opposite sense by having little robustness to M but a better performance at low N.

We also defined the overall Pearson coefficients so that the mutual information is linear in terms of $m = 0, ..., L-1$, the active channel identifier. In particular, the mutual information of each channel, I_m, is such that it satisfies

$$I_m = C\left(1 - \frac{m}{L}\right), \tag{34}$$

where C is an arbitrary constant which can be computed by fixing the overall mutual information as

$$I = \sum_{m \in \mathcal{S}_L} I_m. \tag{35}$$

For the first scenario we have performed two experiments with $L = 20$ active channels and two different values of N and actual mutual information I. These experiments can be seen in Fig. 1, where we show that the ML estimator presents a bias that is directly proportional to the total space dimension, due to the extra bias that comes from the non-active channels, and that by estimating $\hat{I}_{BIC}$ it presents robustness to this effect. However, if we focus on the different variations of λ, we can see that the restrictive threshold λ_2 is compensating the extra bias due to the noise by adding the contribution of fewer channels, but λ_3 presents higher bias as the number of channels increases. Also, note that for the case of λ_3 the estimator starts to approach the ML estimator, which means that the estimator bias tends to be linear with respect to the dimensionality for very small values of the threshold.

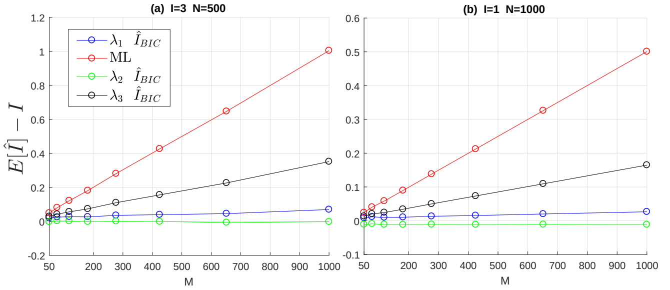


Fig. 1. Bias of the proposed estimator with three different thresholds compared with the ML estimate as a function of M.

On the other hand, the scenario depicted in Fig. 2 is handled in a similar way as the first one, where we fix $L = 20$ and $I = 3$, but we consider an experiment with $M = 1000$ and another with $M = 100$. In this figure we show that by using λ_1 to estimate mutual information, the estimate converges faster to zero than the ML estimator. However, for $M = 100$ we show that being restrictive with λ_3 can be harmful if the environment has a moderated dimensionality, that is when the actual number of active channels is reasonably close to the total number of channels, so the optimal threshold λ_1 or even more permissive ones as λ_2 are encouraged.

Finally, we study the impact of the different thresholds in the overall estimation of channels for $L = 20$. This experiment can be seen in Fig. 3, where we can see that λ_2 performs poorly when the mutual information is too low, and its convergence to the correct number of active channels is much slower than λ_1. Regarding λ_3, it is clear that it always detects more channels than necessary, explaining the increased bias in the previous experiments. To conclude, λ_1 stabilizes the total number of active channels detected to the actual value L, but a limited number of samples N may induce an error in the detection, thus converging to a higher value than the actual one.

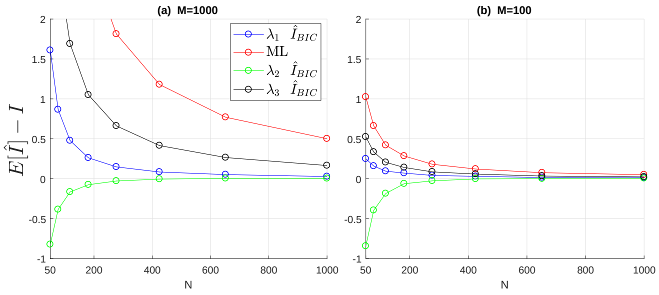


Fig. 2. Bias of the proposed estimator with three different thresholds compared with the ML estimate as a function of N.

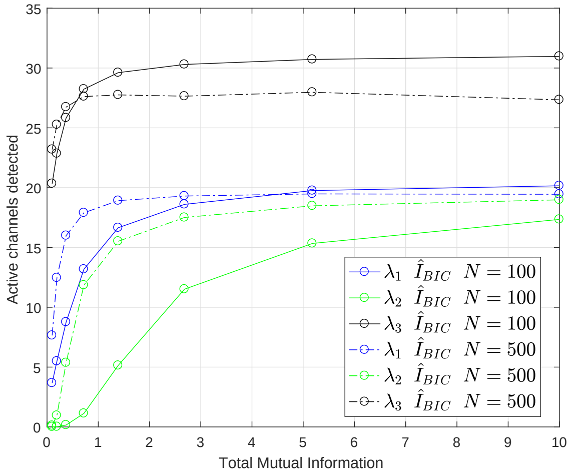


Fig. 3. Active channels detected with three different thresholds as a function of the mutual information value I, with $L = 20$ and $M = 100$.

V. Conclusions

In this paper we have shown that the problem of mutual information estimation for parallel independent channels can be improved by discarding the inactive channels following a given rule. This rule is based on the MDL principle, and it naturally determines an optimal threshold for determining which channels are active and which ones are not. Moreover, if this threshold is not followed, any variation may induce less robustness in terms of bias with respect to the total number of channels M or number of observations N. We have observed that in this particular problem it is possible to achieve a single channel criterion from a multi-channel approach. This feature is of great interest as single channel criteria are computationally less expensive than multi-channel processing.

In view of the proposed ideas, the natural extension of this work would be the case of general $M_x \times M_y$ MIMO channels with low-rank $L < \min(M_x, M_y)$, with Gaussian inputs.

References

[1] E. Arias-Castro, S. Bubeck, and G. Lugosi. Detection of correlations. *The Annals of Statistics*, 40(1), 2012.

[2] F. de Cabrera and J. Riba. Squared-loss mutual information via high-dimension coherence matrix estimation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5142–5146, 2019.

[3] S. M. Kay. *Fundamentals of Statistical Signal Processing: Detection Theory*, volume II. Prentice-Hall, New York, 1993.

[4] S. M. Kay. *Fundamentals of Statistical Signal Processing: Estimation Theory*. Prentice-Hall, New York, 1998.

[5] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate Analysis*. London, U.K.: Academic, 1995.

[6] D. Ramírez, G. Vázquez-Vilar, R. López-Valcarce, J. Vía, and I. Santamaría. Detection of rank-P signals in cognitive radio networks with uncalibrated multiple antennas. *IEEE Transactions on Signal Processing*, 59(8):3764–3774, Aug 2011.

[7] D. Ramírez, J. Vía, I. Santamaría, and L. L. Scharf. Locally most powerful invariant tests for correlation and sphericity of Gaussian vectors. *IEEE Trans. Inf. Theory*, 59(4):2128–2141, Apr. 2013.

[8] J. Rissanen. Modeling by shortest data description. *Automatica*, 14(5):465 – 471, 1978.

[9] T. Roos, P. Myllymäki, and J. Rissanen. MDL denoising revisited. *IEEE Transactions on Signal Processing*, 57(9):3347–3360, Sep. 2009.

[10] N. J. Roseveare and P. J. Schreier. Model-order selection for analyzing correlation between two data sets using CCA with PCA preprocessing. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5684–5687, 2015.

[11] I. Santamaría and J. Vía. Estimation of the magnitude squared coherence spectrum based on reduced-rank canonical coordinates. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 985–988, 2007.

[12] Y. Song, P. Schreier, D. Ramírez, and T. Hasija. Canonical correlation analysis of high-dimensional data with very small sample support. *Signal Processing*, pages 449–458, Nov. 2016.

[13] P. Stoica and Y. Selén. Model-order selection: a review of information criterion rules. *IEEE Signal Process. Magazine*, July 2004.

[14] T. Suzuki, M. Sugiyama, T. Kanamori, and J. Sese. Mutual information estimation reveals global associations between stimuli and biological processes. *BMC Bioinformatics*, 10(1):S52 (12 pages), 2009.

[15] David Tse. *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[16] Q. Wang, S. R. Kulkarni, and S. Verdú. *Universal estimation of Information measures for analog sources*. Number 5:3. Foundations and trends in Communications and Information Theory, 2009.

References

[1] S. Boyd, L. Vandenberghe. *Convex Optimization.* Cambridge University. Cambridge University Press, 2009.

[2] I. Selesnick. *Penalty and Shrinkage Functions for Sparse Signal Processing.* Polytechnic Institute of New York University. 2013.

[3] A. Adler, M. Elad, Y. Hel-Or, E. Rivlin. *Sparse Coding with Anomaly Detection.* IEEE International Workshop on Machine Learning for Signal Processing, Sept. 22–25, 2013, Southampton, UK, 2013.

[4] Marco F. Duarte Member, IEEE, and Yonina C. Eldar, Senior Member, IEEE. *Structured Compressed Sensing: From Theory to Applications.* IEEE Transactions on Signal Processing, Vol. 59, No. 9, September 2011.

[5] R. Tibshirani. *Regression Shrinkage and Selection via the Lasso.* University of Toronto, Canada 1995.

[6] A. Beck, M. Tebouelle. (2009). *A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems.* Society for Industrial and Applied Mathematics: SIAM Journal on Imaging Ssiences, Vol. 2, No. 1, pp. 183–20.

[7] Balzano, R. Nowak, and B. Recht. *Online identification and tracking of subspaces from highly incomplete information.* 2010 48th Annual Allerton Conference on Communication, Control, and Computing (Allerton), 2010, pp. 704–711.

[8] A. Edelman, T. A. Arias, and S. T. Smith. *The geometry of algorithms with orthogonality constraints.* SIAM Journal on Matrix Analysis and Applications, vol. 20, no. 2, pp. 303–353, 1998.

[9] J. Zhang, G. Zhu, R. W. Heath, and K. Huang. *Grassmannian Learning: Embedding Geometry Awareness in Shallow and Deep Learning.* IEEE Signal Processing Magazine, pp. 1–18, 2018.

[10] B. Yang. *Projection Approximation Subspace Tracking.* IEEE Transactions on Signal Processing, vol. 43, no. 1, pp. 95–107, 1995.

[11] X. G. Doukopoulos and G. V. Moustakides. *Fast and Stable Subspace Tracking.* IEEE Transactions on Signal Processing, Vol. 56, No. 4, April 2008.

[12] J. Yang and M. Kaveh. *Adaptive Eigensubspace Algorithms for Direction or Frequency Estimation and Tracking.* IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 36, No. 2, February 1988.

[13] S. T. Smith. *Intrinsic Cramér-Rao bounds and subspace estimation accuracy.* Lincoln Laboratory, Massachusetts Institute of Technology, 2016.

[14] S. T. Smith. *Covariance, Subspace, and Intrinsic Cramér–Rao Bounds.* IEEE Transactions on Signal Processing, Vol. 53, No. 5, May 2005.

[15] C. A. Lopez, F. de Cabrera and J. Riba. *Estimation of Information in Parallel Gaussian Channels via Model Order Selection.* 2020 IEEE International Conference on Acoustics, Speech and Signal Processing, May 2020.

[16] B. M. Hochwald and T. L. Marzetta. *Unitary space-time modulation for multiple-antenna communications in Rayleigh flat fading.* IEEE Trans. on Information Theory, vol. 46, no. 2, pp. 543–564, 2000.

[17] O. G. Alma. *Comparison of Robust Regression Methods in Linear Regression.* Int. J. Contemp. Math. Sciences, Vol. 6, 2011, no. 9, 409 - 421.

[18] R. Tibshirani an L. Wassermann. *Nonparametric Regression.* Statistical Machine Learning, Spring 2015.

[19] J. Sanz, J. M. Juan, and M. Hernández, *GNSS Data Processing.* ESA Communications, (ESA TM-23/1, May 2013) 2013, vol. 1: Fundamentals and Algorithms.

[20] Y. Du, G. Zhu, J. Zhang, and K. Huang. *Automatic recognition of space-time constellations by learning on the Grassmann manifold.* Submitted to IEEE Trans. on Signal Processing, 2018. [Online]. Available in: arxiv.org/abs/1804.03593

[21] Bolduc, E., Knee, G.C., Gauger, E.M. et al. *Projected gradient descent algorithms for quantum state tomography.* npj Quantum Inf 3, 44 (2017). https://doi.org/10.1038/s41534-017-0043-1

[22] J. Lee, Y. J. Morton, J. Lee, H.-S. Moon, J. Seo. *Monitoringand mitigation of ionospheric anomalies for GNSS-based safety critical systems.* IEEE Signal Process. Magazine, pp. 96–110, Sept. 2017.

[23] J. Riba, F. de Cabrera. *Multi-Satellite Cycle-Slip Detection and Exclusion using the Noise Subspace of Residual Dynamics.* 2018 26th European Signal Processing Conference (EUSIPCO)

[24] D. Lahat, T. Adali, C. Jutten, *Multimodal Data Fusion: An Overview of Methods, Challenges, and Prospects.* Vol. 103, No. 9, September 2015

[25] J. Yang, H. Xi, F. Yang, Y. Zhao. *RLS-Based Adaptive Algorithms for Generalized Eigen-Decomposition.* IEEE Transactions on Signal Processing, Vol. 54, No. 4, April 2006

[26] W. Dai and O. Milenkovic. *SET: An algorithm for consistent matrix completion.* 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, Dallas, TX, 2010, pp. 3646-3649, doi: 10.1109/ICASSP.2010.5495899.

[27] R. H. Keshavan, A. Montanari and S. Oh. *Matrix Completion From a Few Entries.* in IEEE Transactions on Information Theory, vol. 56, no. 6, pp. 2980-2998, June 2010, doi: 10.1109/TIT.2010.2046205.

[28] S. Kim, K. Koh, M. Lustig, S. Boyd and D. Gorinevsky. *An Interior-Point Method for Large-Scale ℓ_1-Regularized Least Squares.* in IEEE Journal of Selected Topics in Signal Processing, vol. 1, no. 4, pp. 606-617, Dec. 2007, doi: 10.1109/JSTSP.2007.910971.

[29] J. Nocedal, S. J.Wright (2000). *Numerical Optimization.* Springer-Verlag, ISBN 0-387-98793-2

[30] C. Louizos, M. Welling, D. P. Kingma. *Learning Sparse Neural Networks with l_0-norm.* Sixth International Conference on Learning Representations. Vancouver, Canada. 2018

[31] H. J. Kushner and G. G. Yin. Stochastic Approximation and Recursive Algorithms and Applications. Springer-Verlag, New York, second edition, 2003.