

ROBUST SUBSPACE TRACKING WITH CONTAMINATION MITIGATION VIA α -DIVERGENCE

Le Trung Thanh^{*,‡}, Aref Miri Rekavandi[†], Abd-Krim Seghouane[†], Karim Abed-Meraim^{*}

^{*} University of Orléans, Orléans, France

[†] University of Melbourne, Melbourne, Australia

[‡] VNU University of Engineering and Technology, Hanoi, Vietnam

ABSTRACT

We studied the problem of robust subspace tracking (RST) in contaminated environments. Leveraging the fast approximated power iteration and α -divergence, a novel robust algorithm called α FAP was developed for tracking the underlying principal subspace of streaming data over time. α FAP is fast and it outperforms many RST methods while only having a low complexity linear to the data dimension. Some experiments were conducted to illustrate the performance of α FAP.

Index Terms— Subspace tracking, robust algorithms, non-Gaussian noises, contamination, α -divergence.

1. INTRODUCTION

Subspace tracking (ST), which corresponds to the problem of estimating the underlying subspace of streaming data over time, plays an important role in adaptive signal processing [1]. In practice, ST has already been applied to many applications, from direction of arrivals (DOA) [2] and channel estimation [3] to video background/foreground separation [4].

The emergence of big data streams has recently led to several challenges for streaming data analysis and subspace tracking in particular [5, 6]. Among them is the problem of dealing with uncertainty, imperfection, and contamination which may occur at any level in modern online applications [6]. ST handling such issues is referred to as robust ST (RST). Particularly, RST requires not only robustness against data corruption together with high subspace estimation accuracy but also fast implementation. Over the years, many RST methods have been proposed for specific scenarios (e.g., sparse outliers and missing data) and we refer the readers to [7–10] for good surveys. Most of the existing (robust) subspace trackers are, however, not designed for dealing with contaminated noises (i.e., non-standard Gaussian noises) which have been observed in several signal processing applications [11].

In the literature, there exist some RST methods capable of tracking the underlying subspace in contaminated environments [10]. To deal with impulsive noises for example, a few of them were developed by incorporating robust statistics into the well-known projection approximation subspace tracking (PAST) algorithm [12], such as RPAST [13], MCC-PAST [14], and BNC-PAST [15]. Some others were based on the adaptive Kalman filtering technique, e.g., KFVM [16] and its variant KFVNM [17]. Another good approach is the weighted recursive least-squares (WRLS) method. Some notable WRLS-based RST trackers are ROBUSTA [18], ROBUSTQR [19],

RYAST [19], and TRPAST [20]. To deal with colored noises, the authors in [21] proposed two variants of PAST, namely IV-PAST and extended IV-PAST, using instrumental variable (IV) technique. Another IV-based PAST's variant with a variable forgetting factor and a variable regularization was proposed in [22]. Oblique projection is also a potential direction to deal with colored noises, e.g., obPAST [23] and obYAST [24]. In parallel, there are several RST algorithms capable of dealing with sparse outliers, such as GRASTA [25], PETRELS-ADMM [4], ReProCS [26], FedOA-PM [27]. They are mainly based on using ℓ_p -norm minimization techniques. Despite having some advantages, most of the subspace trackers above are either designed for a specific kind of noise or sensitive to a particular choice of algorithmic parameters. This motivates us to develop a new generalized robust subspace tracker to avoid the drawbacks.

Contribution: The main contribution of this paper is a robust subspace tracker called α FAP which is capable of tracking successfully the underlying subspace over time in the presence of data corruption. α FAP provides several appealing advantages over the state-of-the-art subspace trackers. Among them is that it offers robustness against outliers and contaminated mixture noises. Specifically, a robust version of the sample covariance matrix with a novel weight is introduced to mitigate the detrimental effect of such contamination, thanks to α -divergence. As a result, α FAP outperforms most of the existing (robust) ST algorithms. In addition, α FAP is a fast and efficient subspace tracker which only requires a low complexity linear to the dimension of data. This feature is highly beneficial for dealing with high-dimensional and high-velocity data streams in fast time-varying environments. Moreover, α FAP guarantees the orthonormality of the estimated subspace basis which is useful to improve the numerical stability and also for some tracking applications, such as DOA estimation.

2. PRELIMINARIES

In this section, we first formulate the problem of robust subspace tracking and then recall the well-known FAP algorithm on which we highly leverage for developing our method. After that, we briefly introduce the α -divergence which is exploited to design a novel weight against data contamination.

2.1. Subspace Tracking Problem

In this work, we consider the following signal model

$$\mathbf{x}(t) = \ell(t) + \mathbf{n}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t), \quad (1)$$

Here, $\mathbf{x}(t) \in \mathbb{C}^{n \times 1}$ is an observation vector, $\ell(t) \in \mathbb{C}^{n \times 1}$ is the low-rank component of $\mathbf{x}(t)$ which lives in a subspace spanned by a deterministic rank- r matrix¹ $\mathbf{A} \in \mathbb{C}^{n \times r}$ (i.e., $\ell(t) = \mathbf{A}\mathbf{s}(t)$ where $\mathbf{s}(t) \in \mathbb{C}^{r \times 1}$ is a coefficient vector), and $\mathbf{n}(t) \in \mathbb{C}^{n \times 1}$ is a noise vector independent of the signal $\ell(t)$. On the arrival of $\mathbf{x}(t)$ at time t , we want to estimate a principal subspace such that it covers the span of \mathbf{A} .

Ideally, the underlying subspace can be derived from the spectral analysis of $\mathbf{C}_{xx}(t) = \mathbb{E}\{\mathbf{x}(t)\mathbf{x}(t)^H\}$. Denote by $\{\lambda_i, \mathbf{u}_i\}$ the i -th pair of eigenvalue and eigenvector of $\mathbf{C}_{xx}(t)$ and assume that the eigenvalues are sorted in decreasing order. Accordingly, the matrix $\mathbf{U}_s = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_r] \in \mathbb{C}^{n \times r}$ represents the principal (signal) subspace of $\mathbf{C}_{xx}(t)$. As $\mathbf{C}_{xx}(t)$ is not always available in practice, we often use the following sample covariance matrix as a surrogate

$$\mathbf{C}_{xx}(t) = \sum_{k=1}^t \beta^{t-k} \mathbf{x}(k)\mathbf{x}(k)^H = \beta \mathbf{C}_{xx}(t-1) + \mathbf{x}(t)\mathbf{x}(t)^H, \quad (2)$$

where $1 \geq \beta > 0$ is a forgetting factor aimed to discount the effect of past observations and to facilitate the tracking ability of subspace estimators in nonstationary environments.

2.2. FAPI Algorithm

We begin with the conventional power iteration method for computing the dominant eigenvectors of $\mathbf{C}_{xx}(t)$:

$$\mathbf{C}_{xy}(t) = \mathbf{C}_{xx}(t)\mathbf{U}(t-1), \quad (3a)$$

$$\mathbf{U}(t)\mathbf{R}(t) = \mathbf{C}_{xy}(t), \quad (3b)$$

where $\mathbf{U}(t-1)$ is the old estimation of the subspace at $t-1$. Badeau *et al.* in [28] proposed the well-known FAPI algorithm which offers a fast approximated implementation of (3). In particular, the following orthogonal projection was exploited: $\mathbf{U}(t)\mathbf{U}(t)^H \simeq \mathbf{U}(t-1)\mathbf{U}(t-1)^H$ or $\mathbf{U}(t) \simeq \mathbf{U}(t-1)\mathbf{\Theta}(t)$, where $\mathbf{\Theta}(t)$ plays the role of a state transition matrix. As a result, we can rewrite the matrix $\mathbf{C}_{xy}(t)$ in (3) as follows

$$\mathbf{C}_{xy}(t) = \beta \mathbf{C}_{xy}(t-1)\mathbf{\Theta}(t-1) + \mathbf{x}(t)\mathbf{y}(t)^H, \quad (4)$$

with $\mathbf{y}(t) = \mathbf{U}(t-1)^H \mathbf{x}(t)$. Based on (4), the underlying subspace $\mathbf{U}(t)$ can be recursively updated by

$$\mathbf{U}(t) = [\mathbf{U}(t-1) + \mathbf{e}(t)\mathbf{g}(t)^H] \mathbf{\Theta}(t), \quad (5)$$

where

$$\mathbf{e}(t) = \mathbf{x}(t) - \mathbf{U}(t-1)\mathbf{y}(t), \quad (6)$$

$$\mathbf{g}(t) = \mathbf{h}(t)/(\beta + \mathbf{y}(t)^H \mathbf{h}(t)), \quad (7)$$

$$\mathbf{h}(t) = \mathbf{Z}(t-1)\mathbf{y}(t), \quad (8)$$

$$\mathbf{Z}(t) = \beta^{-1} \mathbf{\Theta}(t)^H [\mathbf{I} - \mathbf{g}(t)\mathbf{y}(t)^H] \mathbf{Z}(t-1) \mathbf{\Theta}(t)^{-H}, \quad (9)$$

and $\mathbf{\Theta}(t)$ can be an inverse square root of $\mathbf{I} + \|\mathbf{e}(t)\|_2^2 \mathbf{g}(t)\mathbf{g}(t)^H$. We refer the readers to [28] for its derivations and variants.

2.3. α -Divergence

α -divergence is a family of divergence measures generalizing the Kullback–Leibler divergence (KLD) [29, 30]. There are several forms, modifications, and extensions of this divergence

¹This matrix is represented as a constant one but allowed later to be slowly time varying in our adaptive scheme.

Algorithm 1: α FAPI

INPUT:

- $\{\mathbf{x}(t)\}_{t=1}^N, \mathbf{x}(t) \in \mathbb{C}^{n \times 1}$ and target rank r
- Parameters: $0 < \beta \leq 1, 0 < \alpha \leq 1, 0 < p \leq 2$

INITIALIZATION: $\mathbf{U}_0 = [\mathbf{I}_r]$ and $\mathbf{Z} = \mathbf{I}_r$.

MAIN PROGRAM:

for $t = 1, 2, \dots, N$ do

 // Compute weight

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

$$\mathbf{e}(t) = \mathbf{x}(t) - \mathbf{U}(t-1)\mathbf{y}(t)$$

$$\omega(t) = \exp(-\frac{1-\alpha}{2} \|\mathbf{e}(t)\|^p)$$

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

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

 // Update subspace

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

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

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

$$\mathbf{w}(t) = \mathbf{Z}(t-1)^H (\xi(t)\mathbf{y}(t) + \tau(t)\bar{\mathbf{g}}(t))$$

$$\mathbf{r}(t) = \tau(t)\xi(t)^{-1}(\mathbf{Z}(t-1)\bar{\mathbf{g}}(t) - \mathbf{w}(t)^H \bar{\mathbf{g}}(t))\bar{\mathbf{g}}(t)$$

$$\mathbf{Z}(t) = \beta^{-1}(\mathbf{Z}(t-1) - \bar{\mathbf{g}}(t)\mathbf{w}(t)^H + \mathbf{r}(t)\bar{\mathbf{g}}(t)^H)$$

$$\mathbf{U}(t) = \mathbf{U}(t-1) + (\xi(t)\mathbf{x}(t) - \mathbf{U}(t-1)\mathbf{w}(t))\bar{\mathbf{g}}(t)^H$$

end

OUTPUT: $\mathbf{U}(t) \in \mathbb{C}^{n \times r}$

(see [30] for a good survey). Here, we adopt the form considered in [20]

$$D_\alpha\{g \parallel f\} = \frac{1}{\alpha(1-\alpha)} \left[\int g(\theta)^\alpha f(\theta)^{1-\alpha} d\theta - 1 \right], \quad (10)$$

where g and f are two given distributions and $\alpha \in \mathbb{R} \setminus \{0, 1\}$. At $\alpha = 0$ and $\alpha = 1$, $D_\alpha\{p \parallel q\}$ is defined as limiting cases for $\alpha \rightarrow 0$ and $\alpha \rightarrow 1$, respectively. Specifically when $\alpha \rightarrow 1$, $D_\alpha\{p \parallel q\}$ boils down to the KLD between p and q . One of the most appealing features of α -divergence is that it can offer robustness to outliers and non-Gaussian noises which is then exploited in the next section.

3. PROPOSED METHOD

In this section, we introduce a novel robust variant of FAPI, called α FAPI, to track the underlying principal subspace using α -divergence. The main steps of α FAPI are summarized in Algorithm 1. In the following, we show how to incorporate this divergence into FAPI in order to bolster its robustness against data contamination.

Robust Sample Covariance Matrix: Instead of $\mathbf{C}_{xx}(t)$, we propose to use the following robust sample covariance matrix

$$\bar{\mathbf{C}}_{xx}(t) = \beta \bar{\mathbf{C}}_{xx}(t-1) + \omega(t)\mathbf{x}(t)\mathbf{x}(t)^H. \quad (11)$$

Here, $0 \leq \omega(t) \leq 1$ plays as a weight to mitigate the effect of corruption caused by, for examples, outliers and impulsive noises. In the presence of strong corruption, a small weight $\omega(t)$ is set to reduce its impact on the subspace tracking. When $\omega(t) = 1$, $\bar{\mathbf{C}}_{xx}(t)$ becomes the sample covariance matrix $\mathbf{C}_{xx}(t)$ in (2). The form of $\omega(t)$ will be provided later.

Subspace Update: With the new robust variant $\bar{\mathbf{C}}_{xx}(t)$, we can recast (4) into the following form

$$\bar{\mathbf{C}}_{xy}(t) = \beta \bar{\mathbf{C}}_{xy}(t-1)\mathbf{\Theta}(t-1) + \omega(t)\mathbf{x}(t)\mathbf{y}(t)^H. \quad (12)$$

Let $\bar{\mathbf{C}}_{yy}(t) = \beta \bar{\mathbf{C}}_{yy}(t-1) + \omega(t) \mathbf{y}(t) \mathbf{y}(t)^H$. It is easy to indicate that $\bar{\mathbf{C}}_{yy}(t) = \boldsymbol{\Theta}(t) \mathbf{R}(t)$ which then plays the same role as $\mathbf{Z}(t)^{-1}$ of FAPI. As $\bar{\mathbf{C}}_{yy}(t)$ is a rank-one-update matrix, applying the Sherman–Morrison formula yields

$$\mathbf{Z}(t) = \beta^{-1} \boldsymbol{\Theta}(t)^H [\mathbf{I} - \bar{\mathbf{g}}(t) \mathbf{y}(t)^H] \mathbf{Z}(t-1) \boldsymbol{\Theta}(t)^{-H}, \quad (13)$$

where the vector $\mathbf{g}(t)$ in (7) turns out to be

$$\bar{\mathbf{g}}(t) = \frac{\omega(t) \mathbf{h}(t)}{\beta + \omega(t) \mathbf{y}(t)^H \mathbf{h}(t)}. \quad (14)$$

Now, we can utilize the fast implementation of FAPI to update $\mathbf{Z}(t)$ in (13) and the underlying subspace $\mathbf{U}(t)$ with a linear complexity as follows. Let $\epsilon^2(t) = \|\mathbf{x}(t)\|^2 - \|\mathbf{y}(t)\|^2$,

$$\tau(t) = \frac{\epsilon^2(t)}{1 + \epsilon^2(t) \|\bar{\mathbf{g}}(t)\|^2 + \sqrt{1 + \epsilon^2(t) \|\bar{\mathbf{g}}(t)\|^2}}, \quad (15)$$

and $\xi(t) = 1 - \tau(t) \|\bar{\mathbf{g}}(t)\|^2$. The state transition matrix $\boldsymbol{\Theta}(t)$ can be given by

$$\boldsymbol{\Theta}(t) = \mathbf{I} - \tau(t) \bar{\mathbf{g}}(t) \bar{\mathbf{g}}(t)^H, \quad (16)$$

which is an inverse square root of $\mathbf{I} + \epsilon^2(t) \bar{\mathbf{g}}(t) \bar{\mathbf{g}}(t)^H$. Finally, substituting (16) into (13) and (5) results in

$$\mathbf{Z}(t) = \beta^{-1} (\mathbf{Z}(t-1) - \bar{\mathbf{g}}(t) \mathbf{w}(t)^H + \mathbf{r}(t) \bar{\mathbf{g}}(t)^H), \quad (17)$$

$$\mathbf{U}(t) = \mathbf{U}(t-1) + (\xi(t) \mathbf{x}(t) - \mathbf{U}(t-1) \mathbf{w}(t)) \bar{\mathbf{g}}(t)^H, \quad (18)$$

where $\mathbf{r}(t) = \tau(t) \xi(t)^{-1} (\mathbf{Z}(t-1) \bar{\mathbf{g}}(t) - \mathbf{w}(t)^H \bar{\mathbf{g}}(t)) \bar{\mathbf{g}}(t)$ with $\mathbf{w}(t) = \mathbf{Z}(t-1)^H (\xi(t) \mathbf{y}(t) + \tau(t) \bar{\mathbf{g}}(t))$.

How To Choose The Weight? Here, we propose to use the following weight:

$$\omega(t) = \exp \left(-\frac{1-\alpha}{2} \|\mathbf{x}(t) - \mathbf{U}(t-1) \mathbf{y}(t)\|^p \right), \quad (19)$$

where $0 < \alpha \leq 1$ and $0 < p \leq 2$. Particularly, the choice of $\omega(t)$ stems from the following observations.

Assume that the assumed density and true density of (1) are, respectively, $f(\mathbf{x}, \mathbf{U})$ and $g(\mathbf{x}, \mathbf{U}^*) = (1-\delta)f(\mathbf{x}, \mathbf{U}^*) + \delta h(\mathbf{x})$ where $f(\mathbf{x}, \mathbf{U}^*)$ is the “nominal” one, $h(\mathbf{x})$ is to represent contamination in the data, and δ is to control the contamination proportion. The authors in [20] indicated that when data samples are corrupted, the underlying principal subspace can be obtained by minimizing α -divergence $D_\alpha(g(\mathbf{x}, \mathbf{U}^*) \| f(\mathbf{x}, \mathbf{U}))$ which is defined as in (10). In practice $g(\mathbf{x}, \mathbf{U}^*)$ is not known generally, we can solve the following optimization instead

$$\mathbf{U}_\alpha(t) = \underset{\mathbf{U}}{\operatorname{argmax}} \frac{1}{1-\alpha} \sum_{k=1}^t \beta^{t-k} f(\mathbf{x}(k), \mathbf{U})^{1-\alpha}, \quad (20)$$

see [20, appendix] for details. In parallel, the multivariate Gaussian density is often adopted for modeling $f(\mathbf{x}, \mathbf{U})$ (i.e., standard Gaussian noise). In such a case, (20) is equivalent to

$$\mathbf{U}_\alpha(t) = \underset{\mathbf{U}}{\operatorname{argmin}} \sum_{k=1}^t \beta^{t-k} \ell_\alpha(\mathbf{x}(k), \mathbf{U}) \quad \text{where} \quad (21)$$

$$\ell_\alpha(\mathbf{x}, \mathbf{U}) = \frac{1}{1-\alpha} \left(1 - \exp(-c \|\mathbf{x} - \mathbf{U} \mathbf{U}^H \mathbf{x}\|^2) \right), \quad (22)$$

with $c = (1-\alpha)/2$. Under the assumption that $\mathbf{U}(t) \mathbf{U}(t)^H \simeq \mathbf{U}(t-1) \mathbf{U}(t-1)^H$, taking the gradient of $\ell_\alpha(\cdot)$ with respect

to \mathbf{U} results in the following score function

$$\ell'_\alpha(\mathbf{x}(t), \mathbf{U}) \simeq \exp(-c \|\mathbf{e}(t)\|^2) (\mathbf{x}(t) - \mathbf{U} \mathbf{y}(t)) \mathbf{y}(t)^H. \quad (23)$$

Interestingly, (23) can lead to a weighted recursive least-squares (LS) estimation with a weight relatively proportional to $\exp(-c \|\mathbf{e}(t)\|^2)$. On the other hand, the principal subspace of $\bar{\mathbf{C}}_{xx}(t)$ in (11) can also be derived from minimizing the following exponentially weighted LS function [18]

$$\underset{\mathbf{U}}{\operatorname{argmin}} \sum_{k=1}^t \beta^{t-k} \omega(k) \|\mathbf{x}(k) - \mathbf{U} \mathbf{U}^H \mathbf{x}(k)\|^2. \quad (24)$$

As a result, we can set $\omega(t) = \exp(-c \|\mathbf{e}(t)\|^p)$ with $0 < p \leq 2$. The introduction of p is to remedy the sensitivity of α FAPI to the choice of α , see Fig. 1 for an illustration. Some features of $\omega(t)$ are as follows

$$\text{“clean” data: } \lim_{\|\mathbf{e}(t)\| \rightarrow 0} \omega(t) = 1, \quad (25)$$

$$\text{outlier data: } \lim_{\|\mathbf{e}(t)\| \rightarrow \infty} \omega(t) = 0, \quad (26)$$

for every $0 < \alpha < 1$ and $0 < p \leq 2$. When $\alpha = 1$, $\omega(t) = 1$ which corresponds to the case where observations are outlier-free.

Complexity: With respect to the space complexity, α FAPI only requires a $nr + r^2$ words of memory to save $\mathbf{U}(t)$ and $\mathbf{Z}(t)$ at time t . With respect to the computational complexity, the overall cost of α FAPI is $\mathcal{O}(nr)$ flops, similar to the conventional FAPI algorithm.

4. EXPERIMENTS

In this section, we provide some experiments to demonstrate the tracking ability of α FAPI. Its performance is evaluated in comparison with the state-of-the-art subspace trackers, including FAPI [28], LORAF [31], YAST [32], RYAST [19], and TRPAST [20]. Source codes are available online at <https://github.com/thanhtbt/aFAPI/>.

4.1. Robust Subspace Tracking

Signals $\{\mathbf{x}(t)\}_{t \geq 1}$ were generated under the following model:

$$\mathbf{x}(t) = \mathbf{A}(t) \mathbf{s}(t) + \mathbf{n}(t). \quad (27)$$

Here, the underlying subspace $\mathbf{A}(t) \in \mathbb{R}^{n \times r}$ was simulated as $\mathbf{A}(t) = \mathbf{A}(t-1) + \varepsilon \mathbf{V}(t)$ where $\mathbf{V}(t)$ was a normalized Gaussian noise matrix and $\varepsilon > 0$ was the time-varying factor aimed to control the subspace variation over time. At $t = 0$, each element in $\mathbf{A}_0(i, j)$ was independent and identically distributed (i.i.d.) from $\mathcal{N}(0, 1)$. The subspace coefficient $\mathbf{s}(t) \in \mathbb{R}^{r \times 1}$ was an i.i.d. Gaussian random vector of zero mean and unit variance. The noise vector $\mathbf{n}(t) \in \mathbb{R}^{n \times 1}$ was derived from the following contaminated mixture model:

$$[\mathbf{n}(t)]_i \sim (1-\delta) \mathcal{N}(0, \sigma_n^2) + \delta \mathcal{N}(\mu, \eta \sigma_n^2), \quad 1 \leq i \leq n, \quad (28)$$

where $(1-\delta)$ and δ were the mixing probabilities (i.e., δ represents the contamination rate²), $\sigma_n > 0$ was a standard deviation of the “main” noise distribution, and the pair $\{\mu \geq 0, \eta \geq 1\}$ was to define the “contaminated” distribution.

²When $\delta = 0$, (28) becomes a Gaussian model, i.e., $\mathbf{n}(t)$ is a Gaussian noise. The value $\delta = 0.2$, for example, represents 20% data contamination.

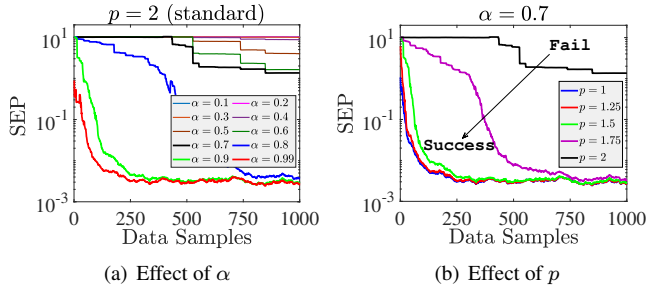


Fig. 1: Performance of α FAPI with different choices $\{\alpha, p\}$.

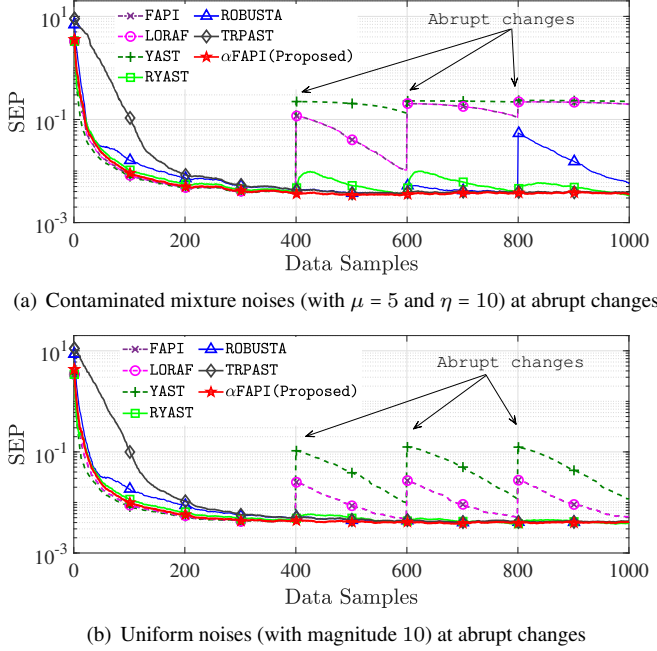


Fig. 2: Subspace tracking with non-Gaussian noises and abrupt changes at $t = 400, 600$, and 800 .

To measure the tracking ability of ST algorithms, we used the following Subspace Estimation Performance (SEP) metric

$$\text{SEP} = \frac{\text{tr} \{ \mathbf{U}(t)^\# (\mathbf{I}_n - \mathbf{A}(t) \mathbf{A}(t)^\#) \mathbf{U}(t) \}}{\text{tr} \{ \mathbf{U}(t)^\# (\mathbf{A}(t) \mathbf{A}(t)^\#) \mathbf{U}(t) \}},$$

where $\mathbf{U}(t)$ is the tracked subspace at each time t . The smaller the value of SEP is, the better accuracy the tracker has.

We set the experimental parameters as follows: data dimension $n = 50$, rank $r = 5$, the time varying factor $\epsilon = 10^{-3}$, the noise level $\sigma_n = 1$, the contaminate rate $\delta = 0.2$, and $\mu = \eta = 1$. The forgetting factor β was fixed at 0.99 for all subspace trackers. Both TRPAST and α FAPI are dependent on the α -divergence parameter which was set to 0.9. Fig. 1 illustrates the performance of α FAPI with different choices of α and p . It can be seen that when $p < 2$, it can remedy the sensitivity of α FAPI to the particular choice of α . Thus, we fixed $p = 1.5$ for α FAPI in all performance comparisons. Fig. 2(a) and (b) show the performance of subspace tracking algorithms when abrupt changes (at $t = 400, 600$, and 800) were created by strong contaminated mixture noises with $\{\mu = 10, \eta = 5\}$

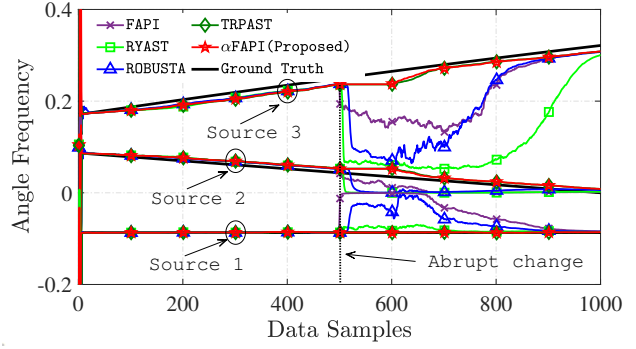


Fig. 3: DOA tracking: 20 sensors, 3 target sources, the background noise level $\text{SNR} = 0\text{dB}$, and contaminated mixture noises ($\delta = 0.2, \mu = 10, \eta = 5$) appear from $t = 500$ to $t = 599$.

and strong uniform noises with magnitude in interval $[0, 10]$, respectively. FAPI, LORAF, and YAST failed to track the underlying subspace. RYAST and ROBUSTA provided better estimation accuracy than the three former trackers, but they were still affected by abrupt changes. TRPAST and α FAPI were capable of tracking successfully the underlying subspace over time. Specifically, the convergence rate of α FAPI was faster than that of TRPAST.

4.2. Direction of Arrival (DOA) Tracking

In this task, we considered a uniform linear array with 20 sensors (two adjacent sensors were separated by half of a wavelength) and 3 target signals impinging on it. The steering matrix had the form $\mathbf{A}(t) = [\mathbf{a}(\omega_1(t)), \dots, \mathbf{a}(\omega_3(t))]$ with $\mathbf{a}(\omega_k(t)) = [1, e^{j\omega_k(t)}, \dots, e^{j(n-1)\omega_k(t)}]^\top$ where $\omega_k(t)$ was the angular frequency associated to the k -th source at time t , i.e., $\omega_k(t) = \pi \sin \theta_k(t)$. The 3×1 vector $\mathbf{n}(t)$ (source signal) is a complex normal random vector following a covariance matrix $\mathbf{C}_s = \mathbf{I}_3$. The background Gaussian noise level was fixed at $\text{SNR} = 0\text{dB}$. To create abrupt changes, data observations were corrupted by the contaminated mixture noises (with $\delta = 0.2, \mu = 10, \eta = 5$) from $t = 500$ to $t = 599$.

To track the DOAs of the moving sources (by linearly varying the value of $\theta_k(t)$), we first applied ST algorithms to estimate the subspace $\mathbf{A}(t)$ and then used the well-known ESPRIT method [33] to extract $\{\omega_k(t)\}_{k=1}^3$. The results are illustrated in Fig. 3. We can see that FAPI, RYAST, and ROBUSTA were capable of tracking the DOAs over time when the background noises were Gaussian. However, they failed to deal with the contaminated mixture noises. Only TRPAST and α FAPI were robust to abrupt changes due to such noises.

5. CONCLUSIONS

In this paper, we have addressed the problem of robust subspace tracking in the presence of nonstandard Gaussian noises. A novel robust adaptive algorithm called α FAPI has been proposed to track the principal subspace of data streams over time. The experimental results indicated that α FAPI was capable of dealing with different data contamination and outperformed other (robust) subspace tracking algorithms.

6. REFERENCES

- [1] J. P. Delmas, "Subspace tracking for signal process." *Adaptive Signal Process.: Next Generation Solutions*, pp. 211–270, 2010.
- [2] B. Liao, Z.-G. Zhang, and S.-C. Chan, "DOA estimation and tracking of ULAs with mutual coupling," *IEEE Trans. Aerospace Electr. Syst.*, vol. 48, no. 1, pp. 891–905, 2012.
- [3] O. Simeone, Y. Bar-Ness, and U. Spagnolini, "Pilot-based channel estimation for OFDM systems by tracking the delay-subspace," *IEEE Trans. Wirel. Commun.*, vol. 3, no. 1, pp. 315–325, 2004.
- [4] L. T. Thanh, N. V. Dung, N. L. Trung, and K. Abed Meraim, "Robust subspace tracking with missing data and outliers: Novel algorithm with convergence guarantee," *IEEE Trans. Signal Process.*, vol. 69, pp. 2070–2085, 2021.
- [5] T. Kolajo, O. Daramola, and A. Adebisi, "Big data stream analysis: A systematic literature review," *J. Big Data*, vol. 6, no. 1, pp. 1–30, 2019.
- [6] N. Vaswani, Y. Chi, and T. Bouwmans, "Rethinking PCA for modern data sets: Theory, algorithms, and applications," *Proc. IEEE*, vol. 106, no. 8, pp. 1274–1276, 2018.
- [7] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, "Robust subspace learning: Robust PCA, robust subspace tracking, and robust subspace recovery," *IEEE Signal Process. Mag.*, vol. 35, no. 4, pp. 32–55, 2018.
- [8] L. Balzano, Y. Chi, and Y. M. Lu, "Streaming PCA and subspace tracking: The missing data case," *Proc. IEEE*, vol. 106, no. 8, pp. 1293–1310, 2018.
- [9] N. Vaswani and P. Narayanamurthy, "Static and dynamic robust PCA and matrix completion: A review," *Proc. IEEE*, vol. 106, no. 8, pp. 1359–1379, 2018.
- [10] N. V. Dung, N. L. Trung, and K. Abed-Meraim, "Robust subspace tracking algorithms in signal process.: A brief survey," *REV J. Electr. Commun.*, vol. 11, no. 1–2, pp. 16–25, 2021.
- [11] A. M. Zoubir, V. Koivunen, Y. Chakhchoukh, and M. Muma, "Robust estimation in signal process.: A tutorial-style treatment of fundamental concepts," *IEEE Signal Process. Mag.*, vol. 29, no. 4, pp. 61–80, 2012.
- [12] B. Yang, "Projection approximation subspace tracking," *IEEE Trans. Signal Process.*, vol. 43, no. 1, pp. 95–107, 1995.
- [13] S.-C. Chan, Y. Wen, and K.-L. Ho, "A robust PAST algorithm for subspace tracking in impulsive noise," *IEEE Trans. Signal Process.*, vol. 54, no. 1, pp. 105–116, 2006.
- [14] J. Zhang and T.-S. Qiu, "A robust correntropy based subspace tracking algorithm in impulsive noise environments," *Digital Signal Process.*, vol. 62, pp. 168–175, 2017.
- [15] S. Luan, T. Qiu, L. Yu, J. Zhang, A. Song, and Y. Zhu, "BNC-based projection approximation subspace tracking under impulsive noise," *IET Radar Sonar Navig.*, vol. 11, no. 7, pp. 1055–1061, 2017.
- [16] S. Chan, Z. Zhang, and Y. Zhou, "A new adaptive Kalman filter-based subspace tracking algorithm and its application to DOA estimation," *IEEE Symp. Circuits Syst.*, pp. 129–132, 2006.
- [17] B. Liao, Z. Zhang, and S.-C. Chan, "A new robust Kalman filter-based subspace tracking algorithm in an impulsive noise environment," *IEEE Trans. Circuits Syst. II Express Briefs*, vol. 57, no. 9, pp. 740–744, 2010.
- [18] N. L. Trung, V. D. Nguyen, M. Thameri, T. Minh-Chinh, and K. Abed-Meraim, "Low-complexity adaptive algorithms for robust subspace tracking," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1197–1212, 2018.
- [19] V.-D. Nguyen, N. L. Trung, and K. Abed-Meraim, "Robust subspace tracking algorithms using fast adaptive Mahalanobis distance," *Signal Process.*, vol. 195, p. 108402, 2022.
- [20] A. M. Rekavandi, A.-K. Seghouane, and K. Abed-Meraim, "TRPAST: A tunable and robust projection approximation subspace tracking method," *IEEE Trans. Signal Process.*, 2022 (submitted).
- [21] T. Gustafsson, "Instrumental variable subspace tracking using projection approximation," *IEEE Trans. Signal Process.*, vol. 46, no. 3, pp. 669–681, 1998.
- [22] S. Chan, H. Tan, and J. Lin, "A new variable forgetting factor and variable regularized square root extended instrumental variable PAST algorithm with applications," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 3, pp. 1886–1902, 2020.
- [23] M. Chen and Z. Wang, "Subspace tracking in colored noise based on oblique projection," *IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 556–559, 2006.
- [24] F. Yger, M. Berar, G. Gasso, and A. Rakotomamonjy, "Oblique principal subspace tracking on manifold," *IEEE Int. Conf. Acoust. Speech Signal Process.*, pp. 2429–2432, 2012.
- [25] J. He, L. Balzano, and A. Szlam, "Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video," *IEEE Conf. Comput. Vis. Pattern Recogn.*, pp. 1568–1575, 2012.
- [26] P. Narayanamurthy and N. Vaswani, "Provable dynamic robust PCA or robust subspace tracking," *IEEE Trans. Inf. Theory*, vol. 65, no. 3, pp. 1547–1577, 2019.
- [27] P. Narayanamurthy, N. Vaswani *et al.*, "Federated over-air subspace tracking from incomplete and corrupted data," *IEEE Trans. Signal Process.*, vol. 70, pp. 3906–3920, 2022.
- [28] R. Badeau, B. David, and G. Richard, "Fast approximated power iteration subspace tracking," *IEEE Trans. Signal Process.*, vol. 53, no. 8, pp. 2931–2941, 2005.
- [29] S.-I. Amari, " α -divergence is unique, belonging to both f -divergence and Bregman divergence classes," *IEEE Trans. Inf. Theory*, vol. 55, no. 11, pp. 4925–4931, 2009.
- [30] A. Cichocki and S.-I. Amari, "Families of alpha-beta-and gamma-divergences: Flexible and robust measures of similarities," *Entropy*, vol. 12, no. 6, pp. 1532–1568, 2010.
- [31] P. Strobach, "The fast recursive row-householder subspace tracking algorithm," *Signal Process.*, vol. 89, no. 12, pp. 2514–2528, 2009.
- [32] M. Arjomandi-Lari and M. Karimi, "Generalized YAST algorithm for signal subspace tracking," *Signal Process.*, vol. 117, pp. 82–95, 2015.
- [33] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Trans. Acoust. Speech Signal Process.*, vol. 37, no. 7, pp. 984–995, 1989.