



**HAL**  
open science

# Robust Subspace Tracking With Missing Data and Outliers: Novel Algorithm With Convergence Guarantee

Le Trung Thanh, Viet Dung Nguyen, Nguyen Linh Trung, Karim  
Abed-Meraim

## ► To cite this version:

Le Trung Thanh, Viet Dung Nguyen, Nguyen Linh Trung, Karim Abed-Meraim. Robust Subspace Tracking With Missing Data and Outliers: Novel Algorithm With Convergence Guarantee. IEEE Transactions on Signal Processing, 2021, 69, pp.2070-2085. 10.1109/TSP.2021.3066795. hal-03212505

**HAL Id: hal-03212505**

<https://ensta-bretagne.hal.science/hal-03212505v1>

Submitted on 3 May 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Robust Subspace Tracking with Missing Data and Outliers: Novel Algorithm with Convergence Guarantee

Le Trung Thanh, Nguyen Viet Dung, *Member, IEEE*, Nguyen Linh Trung, *Senior Member, IEEE*, and Karim Abed-Meraim, *Fellow, IEEE*

**Abstract**—In this paper, we propose a novel algorithm, namely PETRELS-ADMM, to deal with subspace tracking in the presence of outliers and missing data. The proposed approach consists of two main stages: outlier rejection and subspace estimation. In the first stage, alternating direction method of multipliers (ADMM) is effectively exploited to detect outliers affecting the observed data. In the second stage, we propose an improved version of the parallel estimation and tracking by recursive least squares (PETRELS) algorithm to update the underlying subspace in the missing data context. We then present a theoretical convergence analysis of PETRELS-ADMM which shows that it generates a sequence of subspace solutions converging to the optimum of its batch counterpart. The effectiveness of the proposed algorithm, as compared to state-of-the-art algorithms, is illustrated on both simulated and real data.

**Index Terms**—Robust subspace tracking, online robust PCA, robust matrix completion, missing data, outliers, alternating direction method of multipliers (ADMM).

## I. INTRODUCTION

Subspace estimation plays an important role in signal processing with numerous applications in wireless communications, radar, navigation, image/video processing, biomedical imaging, etc. [1], especially processing modern datasets in today’s big and messy data [2]. It corresponds to estimating an appropriate  $r$ -dimensional subspace  $\mathbf{U}$  of  $\mathbb{R}^n$  where  $r \ll n$ , from a set of  $m$  observed data vectors  $\{\mathbf{x}_i\}_{i=1}^m$ , or equivalently, a measurement data matrix  $\mathbf{X}$  of size  $n \times m$ . To this end, the standard approach is to solve an eigen-problem

Le Trung Thanh is with the Advanced Institute of Engineering and Technology, University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam, and the PRISME Laboratory, University of Orleans, France. Emails: thanhletrung@vnu.edu.vn, trung-thanh.le@univ-orleans.fr.

Nguyen Viet Dung is with the Advanced Institute of Engineering and Technology, University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam and the National Institute of Advanced Technologies of Brittany (ENSTA Bretagne), Brest, France. Email: dungnv@vnu.edu.vn, viet.nguyen@ensta-bretagne.fr.

Nguyen Linh Trung (corresponding author) is with the the Advanced Institute of Engineering and Technology, University of Engineering and Technology, Vietnam National University, Hanoi, Vietnam. Email: linhtrung@vnu.edu.vn.

Karim Abed-Meraim is with the PRISME Laboratory, University of Orleans, Orleans, France. Email: karim.abed-meraim@univ-orleans.fr.

This work was funded by the National Foundation for Science and Technology Development (NAFOSTED) of Vietnam under grant number 102.04-2019.14.

This paper has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the authors. The material includes detailed proofs of several mathematical results presented in the main article and additional experimental results. This material is 2.3 Mb in size.

in a batch manner where the underlying subspace can be obtained from either singular value decomposition of the data matrix or eigenvalue decomposition of its covariance matrix. In certain online or large-scale applications, batch algorithms become inefficient due to their high computational complexity,  $\mathcal{O}(nm \min(m, n))$ , and memory cost,  $\mathcal{O}(nm)$  [3]. Subspace tracking or adaptive (dynamic) principal component analysis (PCA) has been an excellent alternative with a much lower computational complexity as well as memory cost (i.e., linear to the data vector size  $n$  and the subspace dimension  $r$ ).

In the signal processing literature, several good surveys of the standard algorithms for subspace tracking can be found, e.g. [1], [4]. The algorithms can be categorized into three classes in terms of their computational complexity: high complexity  $\mathcal{O}(n^2r)$ , moderate complexity  $\mathcal{O}(nr^2)$  and low complexity  $\mathcal{O}(nr)$ . Note that, there usually exists a trade-off among estimation accuracy, convergence rate and computational complexity. However, the performance of standard algorithms may be degraded significantly if the measurement data are corrupted by even a small number of outliers or missing observations [5]. Recent surveys [6]–[8] show that missing data and outliers are ubiquitous and more and more common in the big data regime. This has led to attempts to define robust variants of subspace learning, namely robust subspace tracking (RST), or online robust PCA. In this work, we aim to investigate the RST problem in the presence of both outliers and missing data.

Our study is also motivated by several emerging applications in diverse fields. In big data analysis, subspace tracking is used to monitor dynamic cardiac magnetic resonance imaging (MRI), track network-traffic anomalies [9] or mitigate radio frequency interference (RFI) in radio astronomy [10]. Moreover, in 5G wireless communication, subspace tracking have recently been exploited for channel estimation in massive MIMO [11] and millimeter wave multiuser MIMO [12].

## A. Related Works

In the literature, there have been several studies on subspace tracking in the missing data context. Among them, Grassmannian rank-one update subspace estimation (GROUSE) [13] is an incremental gradient subspace algorithm that performs the stochastic gradient descent on the Grassmannian manifold of the  $r$ -dimensional subspace. It belongs to the class of low complexity and its convergence has recently been

proved in [14]. A robust version of GROUSE for handling outliers is Grassmannian robust adaptive subspace tracking (GRASTA) [15]. GRASTA first uses an  $\ell_1$ -norm cost function to reduce the effect of sparse outliers and then performs the incremental gradient on the Grassmannian manifold of subspace  $\mathbf{U}$  in a similar way as in GROUSE. Although GRASTA is one of the fastest RST algorithms for handling missing data corrupted by outliers, convergence analysis of this algorithm is not available.

Parallel estimation and tracking by recursive least squares (PETRELS) [16] can be considered as an extension of the well-known projection approximation subspace tracking (PAST) algorithm [17] in order to handle missing data. Specifically, PETRELS is a recursive least squares-type algorithm applying the second order stochastic gradient descent to the cost function. Inspired by PETRELS, several variants have been proposed to deal with missing data in the same line such as [9], [18], [19]. The subspace tracking algorithm in [9] is derived from minimizing the sum of squared residuals, but adding a regularization of the nuclear norm of subspace  $\mathbf{U}$ . Robust online subspace estimation and tracking (ROSETA) in [18] applies an adaptive step size at the stage of subspace estimation to enhance the convergence rate. Meanwhile the main idea of PETRELS-CFAR algorithm [19] is to handle “outliers-removed” data (i.e., outliers are first removed before performing subspace tracking) using a constant false alarm rate (CFAR) detector. However, the convergence of these PETRELS-based algorithms has not been mathematically proved yet.

Recursive projected compressive sensing (ReProCS)-based algorithms [20], [21] are also able to adaptively reconstruct a subspace from missing observations. They provide not only a memory-efficient solution, but also a precise subspace estimation as compared to the state-of-the-arts. However, they require strong assumptions on subspace changes, outlier magnitudes and accurate initialization.

Other subspace tracking algorithms, able to deal with missing data, include pROST [22], APSM [23], POPCA [24] and OVBSL [25]. They either require memorizing previous observations and good initialization or do not provide a convergence guarantee.

Among the subspace tracking algorithms reviewed above, only a few of them are robust in the presence of both outliers and missing observations, including GRASTA [15], pROST [22], ROSETA [18], ReProCS-based algorithms [20], [21] and PETRELS-CFAR [19].

## B. Contributions

Adopting the approach of PETRELS-CFAR [19] but aiming to improve RST performance, we are interested in looking for a method that can remove outliers more effectively. Following our preliminary study presented in [26], the main contributions of the paper are as follows.

First, we propose a novel algorithm, namely PETRELS-ADMM, for the RST problem to deal with both missing data and outliers. It includes two main stages: outlier rejection and subspace estimation and tracking. Outliers residing in the

measurement data are detected and removed by our ADMM solver in an effective way. Particularly, we design an efficient augmented Lagrangian alternating direction method for the  $\ell_1$ -regularized loss minimization. Furthermore, we propose an improved version of PETRELS, namely iPETRELS. It is observed that PETRELS is ineffective when the fraction of missing data is too large. We thus add a regularization of the  $\ell_{2,\infty}$ -norm, which aims to control the maximum  $\ell_2$ -norm of rows in  $\mathbf{U}$ , in the objective function to avoid such performance loss. In addition, we introduce an adaptive step size to speed up the convergence rate as well as enhance the subspace estimation accuracy.

Second, we provide a convergence analysis of the proposed algorithm where we show that the solutions  $\{\mathbf{U}_t\}_{t=1}^{\infty}$  generated by PETRELS-ADMM converge to a stationary point of the expected loss function  $f(\mathbf{U})$  asymptotically. To the best of our knowledge, this is a pioneer analysis for RST algorithm’s convergence in the presence of both outliers and missing data, *under mild conditions*.

Finally, we provide extensive experiments on both simulated and real data to illustrate the effectiveness of PETRELS-ADMM in three application contexts: robust subspace tracking, robust matrix completion and video background-foreground separation.

There are several differences between PETRELS-ADMM and the state-of-the-art RST algorithms. In particular, our mechanism for outlier rejection can facilitate the subspace estimation ability of RST algorithms where “clean” data involve the process only, thus improving overall performance. Excepting PETRELS-CFAR, the common principle of the state-of-the-art algorithms is “outlier-resistant” (i.e., to have a “right” direction toward the true subspace). The algorithms thus require robust cost functions as well as additional adaptive parameter selection. For examples, GRASTA and ROSETA use the  $\ell_1$ -norm robust estimator to reduce the effect of outliers while pROST applies the  $\ell_0$ -norm one instead. However, there is no guarantee that the  $\ell_p$ -norm robust estimator (i.e.,  $p \in [0, 1]$ ) can provide an optimal solution because of non-convexity. Accordingly, the effect of outliers can not be completely removed in tracking. This is why the algorithms can fail in the appearance of a large fractions of outliers or significant subspace changes in practice. By contrast, ‘detect and skip’ approach like PETRELS-CFAR can utilize advantage (i.e., competitive performance) of the original PETRELS in missing observations and then treat outliers as missing data to facilitate the subspace tracking.

Compared to PETRELS-CFAR, our ADMM solver may be efficient than CFAR in terms of memory cost and flexibility. The constant false alarm rate method (CFAR) [27] uses a moving window to detect outliers (i.e., using both old and new observations at each time instant). By contrast, our ADMM solver exploits only a new incoming data vector, hence requiring a lower storage complexity. Moreover, the performance of CFAR depends on predefined parameters such as the probability of false alarm and the size of the reference window [19]. Our ADMM solver does not involve such parameters and hence it is more efficient. Third, PETRELS-CFAR may provide an unstable solution in the presence of a

high corruption fraction due to lack of regularization (i.e., in the similar way as PETRELS).

Moreover, PETRELS-ADMM can be classified to a class of provable ST algorithms [20], [21] where a performance guarantee is provided. Our proposed algorithm takes both advantages of streaming solution (need only single-pass of data) and preserved convergence.

The structure of the paper is organized as follows. Section II formulate the RST problem. Section III establishes our PETRELS-ADMM algorithm for RST and Section IV gives its theoretical convergence analysis. Section V presents extensive experiments to illustrate the effectiveness of PETRELS-ADMM as compared to the state-of-the-art algorithms. Section VI concludes the paper.

### C. Notations

We use lowercase (e.g.  $a$ ), boldface lowercase (e.g.  $\mathbf{a}$ ), capital boldface (e.g.  $\mathbf{A}$ ) and calligraphic letters (e.g.  $\mathcal{A}$ ) letters to denote scalars, vectors, matrices and sets respectively. The  $i$ -th entry of a vector  $\mathbf{a}$  is denoted by  $\mathbf{a}(i)$ . For a matrix  $\mathbf{A}$ ,  $(i, j)$ -th entry is denoted by  $\mathbf{A}(i, j)$ ;  $\mathbf{A}_{:,k}$  and  $\mathbf{A}_{l,:}$  are  $k$ -th column and  $l$ -th row of  $\mathbf{A}$  respectively. Operators  $(\cdot)^\top$ ,  $(\cdot)^\#$ ,  $\mathbb{E}[\cdot]$ ,  $\text{tr}(\cdot)$  denote the transportation, pseudo-inverse, expectation and trace operator respectively. For  $1 \leq p < \infty$ ,  $\ell_p$ -norm of a vector  $\mathbf{a} \in \mathbb{R}^{n \times 1}$  is  $\|\mathbf{a}\|_p \triangleq (\sum_{i=1}^n |\mathbf{a}(i)|^p)^{1/p}$ ;  $\ell_0$ -norm is  $\|\mathbf{a}\|_0 \triangleq \lim_{p \rightarrow 0} (\sum_{i=1}^n |\mathbf{a}(i)|^p)$ ;  $\ell_\infty$ -norm is  $\|\mathbf{a}\|_\infty \triangleq \max_i |a(i)|$ . The  $\ell_{2,\infty}$ -norm of  $\mathbf{A}$  is defined as the maximum  $\ell_2$ -norm of all rows in  $\mathbf{A}$ , i.e.,  $\|\mathbf{A}\|_{2,\infty} = \max_l \|\mathbf{A}_{l,:}\|_2$ . The Frobenius norm of a matrix  $\mathbf{A} \in \mathbb{R}^{n \times m}$  is  $\|\mathbf{A}\|_F \triangleq (\sum_{i=1}^n \sum_{j=1}^m \mathbf{A}(i, j)^2)^{1/2} = \sqrt{\text{tr}(\mathbf{A}^\top \mathbf{A})}$ . The condition number of matrix  $\mathbf{A}$  is  $\kappa(\mathbf{A}) = \frac{\sigma_{\max}(\mathbf{A})}{\sigma_{\min}(\mathbf{A})}$ , where  $\sigma_{\max}(\mathbf{A})$  and  $\sigma_{\min}(\mathbf{A})$  are maximal and minimal singular values of  $\mathbf{A}$  respectively.

## II. PROBLEM FORMULATION

### A. Robust Subspace Tracking

Assume that at each time instant  $t$ , we observe a signal  $\mathbf{x}_t \in \mathbb{R}^n$  satisfying the following model:

$$\mathbf{x}_t = \mathbf{P}_t(\boldsymbol{\ell}_t + \mathbf{n}_t + \mathbf{s}_t), \quad (1)$$

where  $\boldsymbol{\ell}_t \in \mathbb{R}^n$  is the true signal that lies in a low dimensional subspace<sup>1</sup> of  $\mathbf{U} \in \mathbb{R}^{n \times r}$  (i.e.,  $\boldsymbol{\ell}_t = \mathbf{U}\mathbf{w}_t$ , where  $\mathbf{w}_t$  is a weight vector and  $r \ll n$ ),  $\mathbf{n}_t \in \mathbb{R}^n$  is the noise vector,  $\mathbf{s}_t \in \mathbb{R}^n$  is the sparse outlier vector, while the diagonal matrix  $\mathbf{P}_t \in \mathbb{R}^{n \times n}$  is the observation mask indicating whether the  $k$ -th entry of  $\mathbf{x}_t$  is observed (i.e.,  $\mathbf{P}_t(k, k) = 1$ ) or not (i.e.,  $\mathbf{P}_t(k, k) = 0$ ). For the sake of convenience, let  $\Omega_t$  be the set of observed entries at time  $t$ .

Before introducing the RST formulation, we first define a loss function  $\ell(\cdot)$  that remains convex while still promoting sparsity: For a fixed subspace  $\mathbf{U} \in \mathbb{R}^{n \times r}$  and a signal  $\mathbf{x} \in \mathbb{R}^n$  under an observation mask  $\mathbf{P}$ , the loss function  $\ell(\mathbf{U}, \mathbf{P}, \mathbf{x})$

with respect to  $\mathbf{U}$  and  $\{\mathbf{P}, \mathbf{x}\}$  is derived from minimizing the projection residual on the observed entries and accounting for outliers as

$$\ell(\mathbf{U}, \mathbf{P}, \mathbf{x}) \triangleq \min_{\mathbf{s}, \mathbf{w}} \tilde{\ell}(\mathbf{U}, \mathbf{P}, \mathbf{x}, \mathbf{w}, \mathbf{s}) \quad (2)$$

$$\text{with } \tilde{\ell}(\mathbf{U}, \mathbf{P}, \mathbf{x}, \mathbf{w}, \mathbf{s}) = \|\mathbf{P}(\mathbf{U}\mathbf{w} + \mathbf{s} - \mathbf{x})\|_2^2 + \rho \|\mathbf{s}\|_1, \quad (3)$$

where we here use the  $\ell_1$  regularization to promote entry-wise sparsity on  $\mathbf{s}$  and  $\rho > 0$  is a regularization parameter to control the degree of the sparsity.<sup>2</sup>

Now, given a streaming set of observed signals,  $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^t$  in (1), we wish to estimate a rank- $r$  matrix  $\mathbf{U}_t \in \mathbb{R}^{n \times r}$  such that it can cover the span of the complete-data noiseless signal  $\boldsymbol{\ell}_t$ .

RST can be achieved via the following minimization problem:

$$\mathbf{U}_t = \underset{\mathbf{U} \in \mathbb{R}^{n \times r}}{\text{argmin}} \left[ f_t(\mathbf{U}) \triangleq \frac{1}{t} \sum_{i=1}^t \lambda_i^{t-i} \ell(\mathbf{U}, \mathbf{P}_i, \mathbf{x}_i) \right], \quad (4)$$

where the forgetting factor  $\lambda_i \in (0, 1]$  is to discount the effect of past observations. For the convergence analysis, we will consider the expected cost  $f(\mathbf{U})$  on signals distributed by the true data-generating distribution  $\mathbb{P}_{\text{data}}$ , instead of the empirical cost  $f_t(\mathbf{U})$ . Thanks to the law of large numbers, expectation of the observations without discounting (i.e.,  $\lambda = 1$ ) converges to the true value when  $t$  tends to infinity,

$$\hat{\mathbf{U}} = \underset{\mathbf{U} \in \mathbb{R}^{n \times r}}{\text{argmin}} f(\mathbf{U}) \quad (5)$$

$$\text{with } f(\mathbf{U}) \triangleq \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_{\text{data}}} [\ell(\mathbf{U}, \mathbf{P}, \mathbf{x})] = \lim_{t \rightarrow \infty} f_t(\mathbf{U}). \quad (6)$$

From the past estimations  $\{\mathbf{s}_i, \mathbf{w}_i\}_{i=1}^t$ , instead of minimizing the empirical cost function  $f_t(\mathbf{U})$  in (4), we propose to optimize the surrogate  $g_t(\mathbf{U})$  of  $f_t(\mathbf{U})$ , which is defined as

$$g_t(\mathbf{U}) = \frac{1}{t} \sum_{i=1}^t \lambda_i^{t-i} \left( \|\mathbf{P}_i(\mathbf{U}\mathbf{w}_i + \mathbf{s}_i - \mathbf{x}_i)\|_2^2 + \rho \|\mathbf{s}_i\|_1 \right), \quad (7)$$

where  $\{\mathbf{s}_i, \mathbf{w}_i\}_{i=1}^t$  are considered as constants. Note that, the surrogate function provides an upper bound on  $f_t(\mathbf{U})$ . In our convergence analysis, we will prove that  $f_t(\mathbf{U}_t)$  and  $g_t(\mathbf{U}_t)$  converge almost surely to the same limit. As a result, the solution  $\mathbf{U}_t$  obtained by minimizing  $g_t(\mathbf{U})$  is exactly the solution of  $f_t(\mathbf{U})$  when  $t$  tends to infinity.

### B. Assumptions

We make the following assumptions for convenience of convergence analysis as well as helping deploy our optimization algorithm:

(A-1): The data-generation distribution  $\mathbb{P}_{\text{data}}$  has a compact support,  $\mathbf{x} \stackrel{\text{i.i.d.}}{\sim} \mathbb{P}_{\text{data}}$ . Indeed, real data are often bounded such as audio, image and video, hence this assumption naturally holds in many situations.

<sup>2</sup> The most direct way of enforcing sparsity constraints is to control the  $\ell_0$ -norm of the solution which counts the number of non-zero entries. Following this way, the problem of (2) is well specified but computationally intractable. Interestingly, the  $\ell_1$  relaxation can recover the original sparse solution of the  $\ell_0$  problem while still preserving convexity [28].

<sup>1</sup>In an adaptive scheme, this subspace might be slowly time-varying, i.e.,  $\mathbf{U} = \mathbf{U}_t$ , and hence the adaptive RST algorithm introduced next would not only estimate  $\mathbf{U}$  but also track its variations along the iterations.

(A-2):  $\mathbf{U}$  is constrained to the set  $\mathcal{U} \triangleq \{\mathbf{U} \in \mathbb{R}^{n \times r}, \|\mathbf{U}_{:,k}\|_2 \leq 1, 1 \leq \kappa(\mathbf{U}) \leq \alpha\}$  with a constant  $\alpha$ . The first constraint  $\|\mathbf{U}_{:,k}\|_2 \leq 1$  is not restrictive as it is considered to bound the scale of basis vectors in  $\mathbf{U}$  and hence prevents the arbitrarily very large values of  $\mathbf{U}$ . While the low condition number of the subspace  $\kappa(\mathbf{U})$  is to prevent the ill-conditioned computation.

(A-3): Coefficients  $\mathbf{w}$  are constrained to the set  $\mathcal{W} = \{\mathbf{w} \in \mathbb{R}^r, \omega_1 \leq |w(i)| \leq \omega_2, i = 1, 2, \dots, r\}$  with two constants  $\omega_1$  and  $\omega_2$ ,  $0 \leq \omega_1 < \omega_2$ . Since the data  $\mathbf{x}$  and subspace  $\mathbf{U}$  are assumed to be bounded, it is natural that the subspace weight vector  $\mathbf{w}$  is bounded too.

(A-4): The subspace changes at two successive time instances is small, i.e., the largest principal angle between  $\mathbf{U}_t$  and  $\mathbf{U}_{t-1}$  is  $0 \leq \theta_{\max} \ll \pi/2$ , or the distance between the two subspaces,  $d(\mathbf{U}_t, \mathbf{U}_{t-1}) = \sin(\theta_{\max})$ , satisfies  $0 \leq d(\mathbf{U}_t, \mathbf{U}_{t-1}) \ll 1$ .

### III. PROPOSED PETRELS-ADMM ALGORITHM

In this section, we present a novel algorithm, namely PETRELS-ADMM, for RST to handle missing data in the presence of outliers. The main idea is to minimize the empirical cost function  $g_t$  in (7) by updating outliers  $\mathbf{s}_t$ , weight vector  $\mathbf{w}_t$  and subspace  $\mathbf{U}_t$  alternatively.

Under the assumption (A-2) that the underlying subspace  $\mathbf{U}$  changes slowly, we can detect outliers in  $\mathbf{s}_t$  by projecting the new observation  $\mathbf{x}_t$  into the space spanned by the formerly estimated subspace  $\mathbf{U}_{t-1}$  in the previous phase. Specifically, we solve the following minimization problem:

$$\{\mathbf{s}_t, \mathbf{w}_t\} \triangleq \underset{\mathbf{s}, \mathbf{w}}{\operatorname{argmin}} \tilde{\ell}(\mathbf{U}_{t-1}, \mathbf{P}_t, \mathbf{x}_t, \mathbf{w}, \mathbf{s}) \quad \text{with} \quad (8)$$

$$\tilde{\ell}(\mathbf{U}_{t-1}, \mathbf{P}_t, \mathbf{x}_t, \mathbf{w}, \mathbf{s}) = \|\mathbf{P}_t(\mathbf{U}_{t-1}\mathbf{w} + \mathbf{s} - \mathbf{x}_t)\|_2^2 + \rho \|\mathbf{s}\|_1.$$

In the second phase, the subspace  $\mathbf{U}_t$  can be estimated by minimizing the sum of squared residuals:

$$\underset{\mathbf{U}}{\operatorname{argmin}} \frac{1}{t} \sum_{i=1}^t \lambda^{t-i} \frac{\operatorname{tr}(\tilde{\mathbf{P}}_i)}{n} \|\tilde{\mathbf{P}}_i(\mathbf{U}\mathbf{w}_i - \mathbf{x}_i)\|_2^2 + \frac{\alpha}{2t} \|\mathbf{U}\|_{2,\infty}^2, \quad (9)$$

where the regularization  $\frac{\alpha}{2t} \|\mathbf{U}\|_{2,\infty}^2$  is to bound the scale of vectors in  $\mathbf{U}$  while the outliers  $\mathbf{s}_t$  has been disregarded and the new observation  $\tilde{\mathbf{P}}_i$  are determined by the following rule:

$$\begin{cases} \tilde{\mathbf{P}}_i(k, k) = \mathbf{P}_i(k, k), & \text{if } \mathbf{s}_i(k) = 0, \\ \tilde{\mathbf{P}}_i(k, k) = 0, & \text{otherwise,} \end{cases} \quad (10)$$

which we aim to skip the corrupted entries of  $\mathbf{x}_i$ .

Our algorithm first applies the ADMM framework in [29], which has been widely used in previous works for solving (8), and then propose a modification of PETRELS [16] to handle (9). In the outlier rejection stage, we emphasize here that we propose to focus on augmenting  $\mathbf{s}$  (as shown in (12)) to further annihilate outlier effect, unlike GRASTA and ROSETA which focus on augmenting the residual error only<sup>3</sup>. Meanwhile, we modify the subspace update step in

<sup>3</sup>In GRASTA [15] and ROSETA [18], both the authors aimed to detect outliers  $\mathbf{s}$  by solving the augmented Lagrangian of (8) as follows

$$\mathcal{L}(\mathbf{s}, \mathbf{y}, \mathbf{w}) = \|\mathbf{s}\|_1 + \mathbf{y}^\top (\mathbf{P}_t(\mathbf{U}_{t-1}\mathbf{w} + \mathbf{s} - \mathbf{x}_t)) + \frac{\rho}{2} \|\mathbf{P}_t(\mathbf{U}_{t-1}\mathbf{w} + \mathbf{s} - \mathbf{x}_t)\|_2^2.$$

---

#### Algorithm 1: Proposed PETRELS-ADMM

---

**Input:** A set of observed signals  $\{\mathbf{x}_i\}_{i=1}^t, \mathbf{x}_i \in \mathbb{R}^{n \times 1}$ , observation masks  $\{\mathbf{P}_i\}_{i=1}^t, \mathbf{P}_i \in \mathbb{R}^{n \times n}$ , true rank  $r$ .

**Output:**  $\mathbf{U}_t$

**Procedure:**

-----  
**for**  $i = 1$  to  $t$  **do**

    // Estimate outliers  $\mathbf{s}_i$  and coefficient  $\mathbf{w}_i$  using Algorithm 2:

$$\{\mathbf{s}_i, \mathbf{w}_i\} = \underset{\mathbf{s}, \mathbf{w}}{\operatorname{argmin}} \|\mathbf{P}_i(\mathbf{U}_{i-1}\mathbf{w} + \mathbf{s} - \mathbf{x}_i)\|_2^2 + \rho \|\mathbf{s}\|_1.$$

    // New observation  $\tilde{\mathbf{P}}_i$ :

$$\begin{cases} \tilde{\mathbf{P}}_i(k, k) = \mathbf{P}_i(k, k), & \text{if } \mathbf{s}_i(k) = 0, \\ \tilde{\mathbf{P}}_i(k, k) = 0, & \text{otherwise.} \end{cases}$$

    // Estimate subspace  $\mathbf{U}_i$  using Algorithm 3:

$$\mathbf{U}_i = \underset{\mathbf{U}}{\operatorname{argmin}} \frac{1}{i} \sum_{j=1}^i \lambda^{i-j} \frac{\operatorname{tr}(\tilde{\mathbf{P}}_j)}{n} \|\tilde{\mathbf{P}}_j(\mathbf{x}_j - \mathbf{U}\mathbf{w})\|_2^2 + \frac{\alpha}{2i} \|\mathbf{U}\|_{2,\infty}^2.$$

**end for**

---



---

#### Algorithm 2: Outlier Detection

---

**Input:** Observed signal  $\mathbf{x}_t \in \mathbb{R}^{n \times 1}$ , observation mask

$\mathbf{P}_t \in \mathbb{R}^{n \times n}$ , the previous estimate  $\mathbf{U}_{t-1} \in \mathbb{R}^{n \times r}$ , maximum iteration  $K$ , penalty parameters  $\rho_1, \rho_2$ , absolute and relative tolerances  $\epsilon_{\text{abs}}$  and  $\epsilon_{\text{rel}}$ .

**Output:**  $\mathbf{s}, \mathbf{w}$

**Initialization:**

- Choose  $\{\mathbf{u}^0, \mathbf{s}^0, \mathbf{w}^0, \mathbf{z}^0, \mathbf{e}^0\}$  randomly.
- $\{\mathbf{r}^0, \mathbf{e}^0\} \leftarrow \mathbf{0}^n$

**Procedure:**

-----  
**for**  $k = 0$  to  $K$  **do**

    // Update  $\mathbf{w}$

$$\begin{aligned} \mathbf{w}^{k+1} &= (\mathbf{P}_t \mathbf{U}_{i-1})^\# \mathbf{P}_t (\mathbf{x}_t - \mathbf{s}^k + \mathbf{e}^k) \\ \mathbf{z}^{k+1} &= \mathbf{P}_t (\mathbf{U}_{t-1} \mathbf{w}^{k+1} + \mathbf{s}^k - \mathbf{x}_t) \\ \mathbf{e}^{k+1} &= \frac{\rho_2}{1+\rho_2} \mathbf{z}^{k+1} + \frac{1}{1+\rho_2} \mathbf{S}_{1+\frac{1}{\rho_2}} (\mathbf{z}^{k+1}) \end{aligned}$$

    // Update  $\mathbf{s}$

$$\begin{aligned} \mathbf{u}^{k+1} &= \frac{1}{1+\rho_1} (\mathbf{P}_t (\mathbf{x}_t - \mathbf{U}_{t-1} \mathbf{w}^{k+1}) - \rho_1 (\mathbf{s}^k - \mathbf{r}^k)) \\ \mathbf{s}^{k+1} &= \mathbf{S}_{\rho/\rho_1} (\mathbf{u}^{k+1} + \mathbf{r}^k) \\ \mathbf{r}^{k+1} &= \mathbf{r}^k + \mathbf{u}^{k+1} - \mathbf{s}^{k+1} \end{aligned}$$

    // Stopping criteria

$$\text{if } \|\mathbf{s}^{k+1} - \mathbf{s}^k\|_2 < \sqrt{n} \epsilon_{\text{abs}} + \epsilon_{\text{rel}} \|\rho_1 \mathbf{r}^{k+1}\|_2 \text{ break;}$$

**end if**

**end for**

---

**Cost**

$$2\Omega_t r^2 + \Omega_t r$$

$$\Omega_t r$$

$$\Omega_t$$

$$\Omega_t r$$

$$\Omega_t$$

$$\Omega_t$$

$$\Omega_t$$

PETRELS by adding an adaptive step size  $\eta_t \in (0, 1]$  at each time instant  $t$ , instead of a constant one as in the original version. The modification can be interpreted as an approximation of Newton method. The proposed method is summarized in Algorithm 1.

#### A. Online ADMM for Outlier Detection

We show in the following how to solve (8) step-by-step:

**Update  $\mathbf{s}_t$ :** To estimate outlier  $\mathbf{s}_t$  given  $\mathbf{w}$ , we exploit the fact that (8) can be cast into the ADMM form as follows:

$$\min_{\mathbf{u}, \mathbf{s}} h(\mathbf{u}) + q(\mathbf{s}), \quad \text{subject to } \mathbf{u} - \mathbf{s} = \mathbf{0}, \quad (11)$$

where  $\mathbf{u}$  is the additional decision variable,  $h(\mathbf{u}) = \frac{1}{2} \|\mathbf{P}_t(\mathbf{U}_{t-1}\mathbf{w} + \mathbf{u} - \mathbf{x}_t)\|_2^2$  and  $q(\mathbf{s}) = \rho \|\mathbf{s}\|_1$ . The corresponding augmented Lagrangian with the dual variable vector  $\boldsymbol{\beta}$  is thus given by

$$\mathcal{L}(\mathbf{s}, \mathbf{u}, \boldsymbol{\beta}) = q(\mathbf{s}) + h(\mathbf{u}) + \boldsymbol{\beta}^\top (\mathbf{u} - \mathbf{s}) + \frac{\rho_1}{2} \|\mathbf{u} - \mathbf{s}\|_2^2, \quad (12)$$

where  $\rho_1 > 0$  is the regularization parameter<sup>4</sup>. Let  $\mathbf{r} = \boldsymbol{\beta}/\rho_1$  be the scaled dual variable, we can rewrite the Lagrangian (12) as follows:

$$\mathcal{L}(\mathbf{s}, \mathbf{u}, \mathbf{r}) = q(\mathbf{s}) + h(\mathbf{u}) + \rho_1 \mathbf{r}^\top (\mathbf{u} - \mathbf{s}) + \frac{\rho_1}{2} \|\mathbf{u} - \mathbf{s}\|_2^2. \quad (13)$$

The optimization of (13) is achieved iteratively where we have the following update rule using the scaled dual variable at the  $k$ -th iteration,

$$\mathbf{u}^{k+1} = \underset{\mathbf{u}}{\operatorname{argmin}} (h(\mathbf{u}) + \rho_1 (\mathbf{r}^k)^\top (\mathbf{u} - \mathbf{s}^k) + \frac{\rho_1}{2} \|\mathbf{u} - \mathbf{s}^k\|_2^2), \quad (14)$$

$$\mathbf{s}^{k+1} = \underset{\mathbf{s}}{\operatorname{argmin}} (q(\mathbf{s}) - \rho_1 (\mathbf{r}^k)^\top \mathbf{s} + \frac{\rho_1}{2} \|\mathbf{u}^{k+1} - \mathbf{s}\|_2^2), \quad (15)$$

$$\mathbf{r}^{k+1} = \mathbf{r}^k + \mathbf{u}^{k+1} - \mathbf{s}^{k+1}. \quad (16)$$

In particular, we first exploit that the minimization (14) can be formulated as a convex quadratic form:

$$\begin{aligned} \mathbf{u}^{k+1} &= \underset{\mathbf{u}}{\operatorname{argmin}} \left( \frac{1 + \rho_1}{2} \|\mathbf{u}\|_2^2 - \right. \\ &\quad \left. [\mathbf{P}_t(\mathbf{x}_t - \mathbf{U}_{t-1}\mathbf{w}) - \rho_1(\mathbf{s}^k - \mathbf{r}^k)]^\top \mathbf{u} \right) \\ &= \frac{1}{1 + \rho_1} (\mathbf{P}_t(\mathbf{x}_t - \mathbf{U}_{t-1}\mathbf{w}) - \rho_1(\mathbf{s}^k - \mathbf{r}^k)). \end{aligned} \quad (17)$$

Meanwhile, the problem (15) is a standard proximal minimization with the  $\ell_1$ -norm [33] as

$$\begin{aligned} \mathbf{s}^{k+1} &= \underset{\mathbf{s}}{\operatorname{argmin}} (\rho \|\mathbf{s}\|_1 + \frac{\rho_1}{2} \|\mathbf{s} - (\mathbf{u}^{k+1} + \mathbf{r}^k)\|_2^2) \\ &= S_{\rho/\rho_1}(\mathbf{u}^{k+1} + \mathbf{r}^k), \end{aligned} \quad (18)$$

where  $S_a(x)$  is a thresholding operator applied element-wise and defined as

$$S_a(x) = \begin{cases} 0, & \text{if } |x| \leq a, \\ x - a, & \text{if } x > a, \\ x + a, & \text{if } x < -a, \end{cases}$$

which is a proximity operator of the  $\ell_1$ -norm.

Finally, a simple update rule for the scaled dual variable  $\mathbf{r}$  can be given by the dual ascent, as

$$\mathbf{r}^{k+1} = \mathbf{r}^k + \beta^k \nabla_{\mathcal{L}}(\mathbf{r}^k),$$

where the gradient  $\nabla_{\mathcal{L}}(\mathbf{r}^k)$  is computed by  $\nabla_{\mathcal{L}}(\mathbf{r}^k) = \rho_1(\mathbf{u}^{k+1} - \mathbf{s}^{k+1})$  and  $\beta^k > 0$  is the step size controlling the convergence rate. For ADMM methods, the regularization parameter is often used as the step size for updating dual variables [29]. Due to the scaled version  $\mathbf{r}$  of the dual variable  $\boldsymbol{\beta}$ , the step size  $\beta^k$  is here set to be  $\beta^k = 1/\rho_1$  at the  $k$ -th iteration.

**Update  $\mathbf{w}_t$ :** To estimate  $\mathbf{w}_t$  given  $\mathbf{s}$ , (8) can be recast into the following ADMM form:

$$\begin{aligned} \min_{\mathbf{w} \in \mathcal{W}, \mathbf{e} \in \mathbb{R}^{n \times 1}} & \frac{1}{2} \|\mathbf{P}_t(\mathbf{U}_{t-1}\mathbf{w} + \mathbf{s} - \mathbf{x}_t)\|_2^2 + y(\mathbf{e}) \\ \text{subject to} & \quad \mathbf{P}_t(\mathbf{U}_{t-1}\mathbf{w} + \mathbf{s} - \mathbf{x}_t) = \mathbf{e} \end{aligned} \quad (19)$$

where  $y(\mathbf{e})$  is a convex regularizer function for the noise  $\mathbf{e}$ , (e.g.  $y(\mathbf{e}) = \frac{\sigma}{2} \|\mathbf{e}\|_2^2$ , with  $\sigma^{-1}$  can be chosen as the signal

<sup>4</sup>It is referred to as the penalty parameter. Although the convergence rate of the proposed algorithm depends on a specific chosen value, our convergence analysis indicates that the ADMM solver can converge for any positive fixed penalty parameters. However, varying penalty parameters can give superior convergence in practice [29]–[32].

to noise ratio, SNR). The minimization (19) is equal to the following optimization:

$$\begin{aligned} \min_{\mathbf{w} \in \mathcal{W}, \mathbf{e} \in \mathbb{R}^{n \times 1}} & \|\mathbf{e}\|_2^2 \\ \text{subject to} & \quad \mathbf{P}_t(\mathbf{U}_{t-1}\mathbf{w} + \mathbf{s} - \mathbf{x}_t) = \mathbf{e}. \end{aligned} \quad (20)$$

However, the noise  $\mathbf{e}$  is still affected by outliers because  $\mathbf{s}$  may not be completely rejected in each iteration. Therefore, (20) can be cast further into the ADMM form such that it can lie between least squares (LS) and least absolute deviations to reduce the impact of outliers. The Huber fitting can bring transition between the quadratic and absolute terms of  $\mathcal{L}_{\mathbf{w}, \mathbf{e}}(\mathbf{w}, \mathbf{e})$ <sup>5</sup>, as

$$\mathcal{L}_{\mathbf{w}, \mathbf{e}}(\mathbf{w}, \mathbf{e}) = f^{\text{Hub}}(\mathbf{e}) + \frac{\rho_2}{2} \|\mathbf{P}_t(\mathbf{U}_{t-1}\mathbf{w} + \mathbf{s} - \mathbf{x}_t) - \mathbf{e}\|_2^2, \quad (21)$$

where  $\rho_2 > 0$  is the penalty parameter whose characteristics are similar to that of  $\rho_1$  and the Huber function is given by [29]

$$f^{\text{Hub}}(x) = \begin{cases} x^2/2, & |x| \leq 1, \\ |x| - 1/2, & |x| > 1. \end{cases}$$

As a result,  $\mathbf{e}$ -updates for estimating  $\mathbf{w}$  involves the proximity operator of the Huber function, that is,

$$\begin{aligned} \mathbf{e}^{k+1} &= \frac{\rho_2}{1 + \rho_2} \mathbf{P}_t(\mathbf{U}_{t-1}\mathbf{w}^{k+1} + \mathbf{s} - \mathbf{x}_t) \\ &\quad + \frac{1}{1 + \rho_2} S_{1 + \frac{1}{\rho_2}}(\mathbf{P}_t(\mathbf{U}_{t-1}\mathbf{w}^{k+1} + \mathbf{s} - \mathbf{x}_t)). \end{aligned}$$

Hence, at the  $(k+1)$ -th iteration,  $\mathbf{w}^{k+1}$  can be updated using the following closed-form solution of the convex quadratic function:

$$\mathbf{w}^{k+1} = (\mathbf{P}_t \mathbf{U}_{t-1})^\# \mathbf{P}_t(\mathbf{x}_t - \mathbf{s} + \mathbf{e}^k).$$

To sum up, the rule for updating  $\mathbf{w}_t$  can be given by

$$\mathbf{w}^{k+1} = (\mathbf{P}_t \mathbf{U}_{t-1})^\# \mathbf{P}_t(\mathbf{x}_t - \mathbf{s} + \mathbf{e}^k), \quad (22)$$

$$\mathbf{z}^{k+1} = \mathbf{P}_t(\mathbf{U}_{t-1}\mathbf{w}^{k+1} + \mathbf{s} - \mathbf{x}_t), \quad (23)$$

$$\mathbf{e}^{k+1} = \frac{\rho_2}{1 + \rho_2} \mathbf{z}^{k+1} + \frac{1}{1 + \rho_2} S_{1 + \frac{1}{\rho_2}}(\mathbf{z}^{k+1}). \quad (24)$$

We note that, by using the Huber fitting operator, our algorithm is better in reducing the impact of outliers than GRASTA and ROSETA which use  $\ell_2$ -norm regularization.

The procedure is stopped when the number of iterations reaches the maximum or the accuracy tolerance for the primal residual and dual norm has been met, i.e.,

$$\|\mathbf{s}^{k+1} - \mathbf{s}^k\|_2 < \sqrt{n} \epsilon_{\text{abs}} + \epsilon_{\text{rel}} \|\rho_1 \mathbf{r}^{k+1}\|_2,$$

where  $\epsilon_{\text{abs}} > 0$  and  $\epsilon_{\text{rel}} > 0$  are predefined tolerances for absolute and relative part respectively. A reasonable range for the absolute tolerance may be  $[10^{-6}, 10^{-3}]$ , while  $[10^{-4}, 10^{-2}]$  is good for the relative tolerance, see [29] for further details of the stopping criterion. The main steps of the outlier detection are summarized as Algorithm 2.

---

**Algorithm 3: Improved PETRELS for updating  $\mathbf{U}_t$** 


---

**Input:** Observed signals  $\{\mathbf{x}_i\}_{i=1}^t$ , observation mask  $\tilde{\mathbf{P}}_t$ , the previous estimate  $\mathbf{U}_{t-1}$ , forgetting factor  $\lambda$ , regularization parameter  $\alpha$ , the step size  $\eta$ , the previous matrix  $\mathbf{H}_{t-1}$ .

**Output:**  $\mathbf{U}_t$

**Procedure:**

$$x_t = \frac{\|\tilde{\mathbf{P}}_t \mathbf{x}_t - \tilde{\mathbf{P}}_t \mathbf{U}_{t-1} \mathbf{w}_t\|_2}{\|\mathbf{w}_t\|_2}$$

$$\eta_t = \frac{x_t}{\sqrt{x_t^2 + 1}}$$

**if**  $\eta_t > \eta$  **then**  $\eta_t = 1$  **end if**

**for**  $m = 1$  to  $n$  **do**

$$\mathbf{R}_t^m = \lambda_t \mathbf{R}_{t-1}^m + \tilde{\mathbf{P}}_t(m, m) \mathbf{w}_t \mathbf{w}_t^\top$$

$$\mathbf{H}_t^m = \mathbf{R}_t^m + \frac{\alpha}{2} \mathbf{I}$$

$$\mathbf{a}_t = (\mathbf{H}_t^m)^{-1} \mathbf{w}_t$$

$$\mathbf{u}_t^m = \mathbf{u}_{t-1}^m + \eta_t \beta_t \tilde{\mathbf{P}}_t(m, m) (\mathbf{x}_t^{\text{se}}(m) - \mathbf{w}_t^\top \mathbf{u}_{t-1}^m) \mathbf{a}_t$$

**end for**

**Cost**

$$\Omega_t r$$

$$\mathcal{O}(1)$$

$$\mathcal{O}(1)$$

$$r^2$$

$$r$$

$$\mathcal{O}(r^2)$$

$$r$$

### B. Improved PETRELS for Subspace Estimation

Having estimated  $\mathbf{s}_t$ , we optimize the following minimization

$$\mathbf{U}_t := \underset{\mathbf{U}}{\operatorname{argmin}} \tilde{g}_t(\mathbf{U}) \quad \text{with} \quad (25)$$

$$\tilde{g}_t(\mathbf{U}) = \frac{1}{t} \sum_{i=1}^t \lambda^{t-i} \frac{\operatorname{tr}(\tilde{\mathbf{P}}_i)}{n} \|\tilde{\mathbf{P}}_i (\mathbf{x}_i - \mathbf{U} \mathbf{w}_i)\|_2^2 + \frac{\alpha}{2t} \|\mathbf{U}\|_{2,\infty}^2,$$

where the observation mask  $\tilde{\mathbf{P}}_i$  is computed by (10).

Thanks to the parallel scheme of PETRELS [16], the optimal solution of the problem (25) can be obtained by solving its subproblems at each row  $\mathbf{u}^m$  of  $\mathbf{U}$ ,  $m = 1, 2, \dots, n$ , that is,

$$\min_{\mathbf{u}^m} \frac{1}{t} \sum_{i=1}^t \lambda^{t-i} \beta_i \tilde{\mathbf{P}}_i(m, m) (\mathbf{x}_i(m) - \mathbf{w}_i^\top \mathbf{u}^m)^2 + \frac{\alpha}{2t} \|\mathbf{u}^m\|_2^2,$$

where  $\beta_i = \frac{\operatorname{tr}(\tilde{\mathbf{P}}_i)}{n}$ . In this way, we can speed up the subspace update by ignoring the  $\mathbf{u}^m$  if the  $m$ -th entry of  $\mathbf{x}_t$  is labeled as missing observation or outlier.

Thanks to Newton's method, we can update each row of the subspace by the following rule:

$$\mathbf{u}_t^m = \mathbf{u}_{t-1}^m - (\mathbf{H}_t(\mathbf{u}^m))^{-1} \frac{\partial \tilde{g}_t(\mathbf{U})}{\partial \mathbf{u}^m} \Big|_{\mathbf{u}^m = \mathbf{u}_{t-1}^m}, \quad (26)$$

where the first derivative of  $\tilde{g}_t$  is given by

$$\frac{\partial \tilde{g}_t(\mathbf{U})}{\partial \mathbf{u}^m} = \frac{-2}{t} \sum_{i=1}^t \lambda^{t-i} \beta_i \tilde{\mathbf{P}}_i(m, m) (\mathbf{x}_i(m) - \mathbf{w}_i^\top \mathbf{u}^m) \mathbf{w}_i^\top + \frac{\alpha}{t} \mathbf{u}^m,$$

and the second derivative of  $\tilde{g}_t$ , Hessian matrix, is given by

$$\mathbf{H}_t(\mathbf{u}^m) = \frac{2}{t} \sum_{i=1}^t \lambda^{t-i} \beta_i \tilde{\mathbf{P}}_i(m, m) \mathbf{w}_i \mathbf{w}_i^\top + \frac{\alpha}{t} \mathbf{I}.$$

<sup>5</sup>Due to the natural  $\ell_2$ -ball behavior of the noise (i.e., normal distributed vector) and the sparsity of some unremoved parts of outliers, Huber fitting can be a reasonable choice. The Huber function consists of square and linear terms, so it is less sensitive to variables which have a strong effect on the function  $\ell_2$ -norm, but also does not encourage the sparsity like  $\ell_1$ -norm.

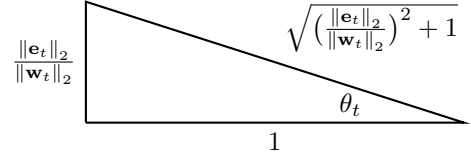


Fig. 1: Adaptive step size  $\eta_t$ .

Specifically, the partial derivative  $\frac{\partial \tilde{g}_t(\mathbf{U})}{\partial \mathbf{u}^m}$  at  $\mathbf{u}_{t-1}^m$  can be expressed by

$$\begin{aligned} \frac{\partial \tilde{g}_t(\mathbf{U})}{\partial \mathbf{u}^m} \Big|_{\mathbf{u}^m = \mathbf{u}_{t-1}^m} &= \frac{\partial \tilde{g}_{t-1}(\mathbf{U})}{\partial \mathbf{u}^m} \Big|_{\mathbf{u}^m = \mathbf{u}_{t-1}^m} + \frac{\alpha}{t} (\mathbf{u}_{t-1}^m - \mathbf{u}_{t-2}^m) \\ &\quad - \frac{2}{t} \beta_t \tilde{\mathbf{P}}_t(m, m) (\mathbf{x}_t(m) - \mathbf{w}_t^\top \mathbf{u}_{t-1}^m) \mathbf{w}_t^\top. \end{aligned}$$

Since  $\mathbf{u}_{t-1}^m = \underset{\mathbf{u}^m}{\operatorname{argmin}} \frac{\partial \tilde{g}_{t-1}(\mathbf{U})}{\partial \mathbf{u}^m}$  and the regularization parameter  $\alpha/t$  is small, so  $\frac{\partial \tilde{g}_{t-1}(\mathbf{U})}{\partial \mathbf{u}^m} \Big|_{\mathbf{u}^m = \mathbf{u}_{t-1}^m} = \mathbf{0}$  and then

$$\frac{\partial \tilde{g}_t(\mathbf{U})}{\partial \mathbf{u}^m} \Big|_{\mathbf{u}^m = \mathbf{u}_{t-1}^m} \approx \frac{-2}{t} \beta_t \tilde{\mathbf{P}}_t(m, m) (\mathbf{x}_t(m) - \mathbf{w}_t^\top \mathbf{u}_{t-1}^m) \mathbf{w}_t^\top.$$

Let us denote  $\mathbf{R}_t^m = \sum_{i=1}^t \lambda_i^{t-i} \beta_i \tilde{\mathbf{P}}_t(m, m) \mathbf{w}_i \mathbf{w}_i^\top$ , the Hessian matrix can be rewritten by

$$\mathbf{H}_t(\mathbf{u}_{t-1}^m) = \frac{2}{t} \left( \mathbf{R}_t^m + \frac{\alpha}{2} \mathbf{I} \right).$$

Therefore, a relaxed approximation of the recursive update (26) can be given by

$$\mathbf{u}_t^m \approx \mathbf{u}_{t-1}^m + \eta_t \beta_t \tilde{\mathbf{P}}_t(m, m) (\mathbf{x}_t(m) - \mathbf{w}_t^\top \mathbf{u}_{t-1}^m) \mathbf{a}_t^\top, \quad (27)$$

where  $\mathbf{H}_t^m = \mathbf{R}_t^m + \frac{\alpha}{2} \mathbf{I}$ ,  $\mathbf{a}_t = (\mathbf{H}_t^m)^{-1} \mathbf{w}_t$  and  $\eta_t$  denotes the adaptive step size  $\eta_t \in [0, 1]$  at each time instant  $t$ , instead of a constant as in the original PETRELS [16]. We here determine the adaptive step size  $\eta_t$  as follows

$$\eta_t = \frac{x_t}{\sqrt{x_t^2 + 1}} \quad \text{with} \quad x_t = \frac{\|\mathbf{e}_t\|_2}{\|\mathbf{w}_t\|_2}, \quad (28)$$

where the residual error  $\mathbf{e}_t$  is computed by  $\mathbf{e}_t = \tilde{\mathbf{P}}_t \mathbf{x}_t - \tilde{\mathbf{P}}_t \mathbf{U}_{t-1} \mathbf{w}_t$ . Note that, the adaptive step size  $\eta_t$  can be expressed by  $\eta_t = \sin(\theta_t)$ , see Fig. 1. The smaller angle  $\theta_t$  is, the closer to the true subspace we are, the smaller step size is needed. The update is summarized in Algorithm 3.

### C. Computational Complexity Analysis

The number of floating-point operations (flop) is used to measure the computational complexity of the proposed PETRELS-ADMM. At the  $k$ -th iteration in the outlier detection phase, our method requires  $\mathcal{O}(\Omega r^2)$  flops where  $\Omega$  is average number of observed entries at each time instant ( $\Omega \leq n$ ). It is practically stated that the ADMM solver can converge within a few tens of iterations [29] (also see Fig. 3). Therefore, the removal of outliers costs the averaged  $\mathcal{O}(\Omega r^2)$ . The complexity of the subspace estimation phase

<sup>6</sup> $\mathbf{H}_t^m \in \mathbb{R}^{r \times r}$  is a matrix of rank-one updates, so its inverse matrix can be efficiently computed recursively, thanks to Sherman–Morrison formula [34]. Also, the small regularization parameter  $\alpha > 0$  can help the recursive update having a better numerical stability. The computational complexity is of order  $\mathcal{O}(r^2)$ .

is also  $\mathcal{O}(\Omega r^2)$  as the original PETRELS [16]. The overall computational complexity of PETRELS-ADMM is of order  $\mathcal{O}(\Omega r^2)$  flops.

#### IV. PERFORMANCE ANALYSIS

In this section, we provide a convergence analysis for the proposed PETRELS-ADMM algorithm. Inspired by the results of convergence of empirical processes for online sparse coding in [35] and online robust PCA in [36], [37], we derive a theoretical approach to analyze the convergence of values of the objective function  $\{f_t(\mathbf{U}_t)\}_{t=1}^{\infty}$  as well as the solutions  $\{\mathbf{U}_t\}_{t=1}^{\infty}$  generated by PETRELS-ADMM.

Given assumptions defined in Section II-B, our main theoretical result can be stated by the following theorem:

**Theorem 1.** (Convergence of PETRELS-ADMM): *In the stationary context, let  $\{\mathbf{U}_t\}_{t=1}^{\infty}$  be the sequence of solutions generated by PETRELS-ADMM, then the sequence converges to a stationary point of the expected loss function  $f(\mathbf{U})$  when  $t \rightarrow \infty$ .*

*Proof Sketch.* Our proof can be divided into three main stages as follows: We first prove that the solutions  $\{\mathbf{U}_t, \mathbf{s}_t\}_{t \geq 1}$  generated by the PETRELS-ADMM algorithm are optimal w.r.t. the cost function in (7). We then prove that a nonnegative sequence  $\{g_t(\mathbf{U}_t)\}_{t=1}^{\infty}$  converges almost surely where  $\{\mathbf{U}_t\}_{t=1}^{\infty}$  is the sequence of optimal solutions generated by the PETRELS-ADMM algorithm. After that, we prove that the surrogate  $\{g_t(\mathbf{U}_t)\}_{t=1}^{\infty}$  converges almost surely to the empirical loss function  $\{f_t(\mathbf{U}_t)\}_{t=1}^{\infty}$  as well as the true loss function, i.e.,  $g_t(\mathbf{U}_t) \xrightarrow{a.s.} f_t(\mathbf{U}_t) \xrightarrow{a.s.} f(\mathbf{U}_t)$ , thanks to the central limit theorem.

Due to space limitation, we here present key results and report their proof sketch. The details of their proofs are provided in our supplemental material.  $\square$

**Lemma 1.** (Convergence of Algorithm 2): *At each time instant  $t$ , let  $\{\mathbf{s}^k, \mathbf{u}^k, \mathbf{r}^k, \mathbf{w}^k, \mathbf{e}^k\}_{k=1}^{\infty}$  be a sequence generated by Algorithm 2 for outlier detection, there always exists a set of positive numbers  $\{c_u, c_s, c_r, c_w, c_e\}$  such that, at each iteration, the minimizers satisfy*

$$\begin{aligned} & \mathcal{L}(\mathbf{s}^{k+1}, \mathbf{u}^{k+1}, \mathbf{r}^{k+1}, \mathbf{w}^{k+1}, \mathbf{e}^{k+1}) \\ & \leq \mathcal{L}(\mathbf{s}^k, \mathbf{u}^k, \mathbf{r}^k, \mathbf{w}^k, \mathbf{e}^k) - c_u \|\mathbf{u}^k - \mathbf{u}^{k+1}\|_2^2 - c_s \|\mathbf{s}^k - \mathbf{s}^{k+1}\|_2^2 \\ & \quad - c_r \|\mathbf{r}^k - \mathbf{r}^{k+1}\|_2^2 - c_w \|\mathbf{w}^k - \mathbf{w}^{k+1}\|_2^2 - c_e \|\mathbf{e}^k - \mathbf{e}^{k+1}\|_2^2, \end{aligned}$$

where the Lagrangian  $\mathcal{L}(\mathbf{s}, \mathbf{u}, \mathbf{r}, \mathbf{w}, \mathbf{e})$  for updating these variables is a combination of two functions (13) and (21), as

$$\begin{aligned} \mathcal{L}(\mathbf{s}, \mathbf{u}, \mathbf{r}, \mathbf{w}, \mathbf{e}) &= q(\mathbf{s}) + h(\mathbf{u}) + \rho_1 \mathbf{r}^\top (\mathbf{u} - \mathbf{s}) + \frac{\rho_1}{2} \|\mathbf{u} - \mathbf{s}\|_2^2 \\ & \quad + f^{Hub}(\mathbf{e}) + \frac{\rho_2}{2} \|\mathbf{P}_t(\mathbf{U}_{t-1} \mathbf{w} + \mathbf{s} - \mathbf{x}_t) - \mathbf{e}\|_2^2. \end{aligned}$$

The asymptotic variation of  $\mathbf{s}^k$  (i.e., outliers) is then given by

$$\lim_{k \rightarrow \infty} \|\mathbf{s}^{k+1} - \mathbf{s}^k\|_2^2 = 0.$$

*Proof Sketch.* We state the following proposition, which is in the same line as in previous convergence analysis of ADMM algorithms [38], [39], used to prove the first part of lemma 1.

**Proposition 1.** *Let  $\{\mathbf{s}^k, \mathbf{u}^k, \mathbf{r}^k, \mathbf{w}^k, \mathbf{e}^k\}_{k=1}^{\infty}$  be a sequence generated by Algorithm 2 and denote  $\mathbf{q}^k$  be one of these variables, the minimizer  $\mathbf{q}^{k+1}$  of (13) satisfies*

$$\mathcal{L}(\mathbf{q}^{k+1}, \cdot) \leq \mathcal{L}(\mathbf{q}^k, \cdot) - c_q \|\mathbf{q}^k - \mathbf{q}^{k+1}\|_2^2,$$

where  $c_q$  is a positive number.

As a result, the cluster  $\{\mathbf{s}^k, \mathbf{u}^k, \mathbf{r}^k, \mathbf{w}^k, \mathbf{e}^k\}$  converges to stationary point of  $\mathcal{L}(\mathbf{s}, \mathbf{u}, \mathbf{r}, \mathbf{w}, \mathbf{e})$  when  $k \rightarrow \infty$  and it also implies that the sequence  $\{\mathbf{s}_k\}_{k=0}^{\infty}$  is convergent, i.e.,

$$\lim_{k \rightarrow \infty} \|\mathbf{s}^{k+1} - \mathbf{s}^k\|_2^2 = 0.$$

$\square$

**Proposition 2.** (Convexity of the surrogate functions  $g_t(\mathbf{U})$ ): *Given assumptions in Section II-B, the surrogate function  $g_t(\mathbf{U})$  defined in Eq. (7) is not only strongly convex, but also Lipschitz function, i.e., there always exists two positive numbers  $m_1$  and  $m_2$  such that*

$$\begin{aligned} m_1 \|\mathbf{U}_{t+1} - \mathbf{U}_t\|_F^2 &\leq |g_t(\mathbf{U}_{t+1}) - g_t(\mathbf{U}_t)|, \\ m_2 \|\mathbf{U}_{t+1} - \mathbf{U}_t\|_F &\geq |g_t(\mathbf{U}_{t+1}) - g_t(\mathbf{U}_t)|. \end{aligned}$$

*Proof Sketch.* To prove that  $g_t(\mathbf{U})$  is strongly convex, we state the following facts:  $g_t(\mathbf{U})$  is continuous and differentiable; its second derivative is a positive semi-definite matrix (i.e.,  $\nabla_{\mathbf{U}}^2 g_t(\mathbf{U}) \geq m\mathbf{I}$ ); and the domain of  $g_t(\mathbf{U})$  is convex. In order to satisfy the Lipschitz condition, we show that the first derivative of  $g_t(\mathbf{U})$  is bounded.  $\square$

**Lemma 2.** (Convergence of Algorithm 3): *Given an outlier vector  $\mathbf{s}_t$  generated by Algorithm 2 at each time instant  $t$ , Algorithm 3 can provide a local optimal solution  $\mathbf{U}_t$  for minimizing  $g_t(\mathbf{U})$ . Moreover, the asymptotic variation of estimated subspaces  $\{\mathbf{U}_t\}_{t \geq 1}$  is given by*

$$\|\mathbf{U}_t - \mathbf{U}_{t+1}\|_F \xrightarrow{a.s.} \mathcal{O}\left(\frac{1}{t}\right).$$

*Proof.* To establish the convergence, we exploit the fact that our modification can be seen as an approximate of the Newton method,

$$\mathbf{U}_t \cong \mathbf{U}_{t-1} - \eta_t [\mathbf{H}_t(\mathbf{U}_{t-1})]^{-1} \nabla \tilde{g}_t(\mathbf{U}_{t-1}),$$

where  $\mathbf{H}_t(\mathbf{U}_{t-1})$  and  $\nabla \tilde{g}_t(\mathbf{U}_{t-1})$  are the Hessian matrix and gradient of the function  $\tilde{g}_t(\mathbf{U})$  at  $\mathbf{U}_{t-1}$ , as shown in Section III-B. It implies that the estimated  $\mathbf{U}_t$  converges to the stationary point of  $g_t(\mathbf{U})$ .

Furthermore, since  $g_t(\mathbf{U})$  is strongly convex and Lipschitz function w.r.t the variable  $\mathbf{U}$  as shown in Proposition 2, we have the following inequality

$$\begin{aligned} m_1 \|\mathbf{U}_{t+1} - \mathbf{U}_t\|_F^2 &\leq |g_t(\mathbf{U}_{t+1}) - g_t(\mathbf{U}_t)| \leq m_2 \|\mathbf{U}_{t+1} - \mathbf{U}_t\|_F \\ \Leftrightarrow \|\mathbf{U}_t - \mathbf{U}_{t+1}\|_F &\leq \frac{m_2}{m_1} = \mathcal{O}\left(\frac{1}{t}\right). \end{aligned}$$

Note that the positive number  $m_2 = \mathcal{O}(1/t)$  is already given in the proof of Proposition 2 in the supplemental material, while  $m_1$  is a constant.  $\square$

**Lemma 3.** (Convergence of the surrogate function  $g_t(\mathbf{U})$ ): *Without discounting past observations, let  $\{\mathbf{U}_t\}_{t=1}^{\infty}$  be a*



sequence of solutions generated by Algorithm 1 at each time instant  $t$ , the sequence  $\{g_t(\mathbf{U}_t)\}_{t=1}^{\infty}$  converges almost surely, i.e.,

$$\sum_{t=1}^{\infty} |\mathbb{E}[g_{t+1}(\mathbf{U}_{t+1}) - g_t(\mathbf{U}_t) | \mathcal{F}_t]| < +\infty \quad a.s.,$$

where  $\{\mathcal{F}_t\}_{t>0}$  is the filtration of the past estimations at time instant  $t$ .

*Proof Sketch.* Let us define the indicator function  $\delta_t$  as follows

$$\delta_t \triangleq \begin{cases} 1 & \text{if } \mathbb{E}[g_{t+1}(\mathbf{U}_{t+1}) - g_t(\mathbf{U}_t) | \mathcal{F}_t] > 0, \\ 0 & \text{otherwise.} \end{cases}$$

According to the quasi-martingale convergence theorem [40, Section 4.4], in order to show the convergence of the nonnegative stochastic process  $\{g_t(\mathbf{U}_t)\}_{t=1}^{\infty}$ , we will prove

$$\sum_{t=0}^{\infty} \mathbb{E}[\delta \mathbb{E}[g_{t+1}(\mathbf{U}_{t+1}) - g_t(\mathbf{U}_t) | \mathcal{F}_t]] < \infty.$$

In particular, we first indicate the following inequality:

$$g_{t+1}(\mathbf{U}_{t+1}) - g_t(\mathbf{U}_t) \leq \frac{\ell(\mathbf{U}_t, \mathbf{P}_{t+1}, \mathbf{x}_{t+1}) - f_t(\mathbf{U}_t)}{t+1}.$$

Since  $\mathbb{E}[\ell(\mathbf{U}_t, \mathbf{P}_{t+1}, \mathbf{x}_t)] = f(\mathbf{U}_t)$ , we have a nice property:

$$\begin{aligned} \mathbb{E}[g_{t+1}(\mathbf{U}_{t+1}) - g_t(\mathbf{U}_t) | \mathcal{F}_t] &\leq \frac{\mathbb{E}[\ell(\mathbf{U}_t, \mathbf{P}_{t+1}, \mathbf{x}_{t+1}) - f_t(\mathbf{U}_t) | \mathcal{F}_t]}{t+1} \\ &= \frac{f(\mathbf{U}_t) - f_t(\mathbf{U}_t)}{t+1}. \end{aligned}$$

We then have

$$\begin{aligned} \mathbb{E}[\delta \mathbb{E}[g_{t+1}(\mathbf{U}_{t+1}) - g_t(\mathbf{U}_t) | \mathcal{F}_t]] \\ \leq \mathbb{E}[\sqrt{t}(f(\mathbf{U}_t) - f_t(\mathbf{U}_t))] \frac{1}{\sqrt{t}(t+1)}. \end{aligned}$$

Under the given assumptions, we exploit the fact that the set of measurable functions  $\{\ell(\mathbf{U}_i, \mathbf{P}, \mathbf{x})\}_{i \geq 1}$  defined in (2) is  $\mathbb{P}$ -Donsker. Therefore, the centered and scaled version of the empirical function  $f_t(\mathbf{U}_t)$  satisfies the following proposition:

$$\mathbb{E}[\sqrt{t}(f(\mathbf{U}_t) - f_t(\mathbf{U}_t))] = \mathcal{O}(1),$$

thanks to Donsker theorem [41, Sec 19.2]. Furthermore, we also indicate that the sum  $\sum_{t=1}^{\infty} 1/(\sqrt{t}(t+1))$  converges. The two facts result in

$$\sum_{t=0}^{\infty} \mathbb{E}[\delta \mathbb{E}[g_{t+1}(\mathbf{U}_{t+1}) - g_t(\mathbf{U}_t) | \mathcal{F}_t]] < \infty.$$

Since  $g_t(\mathbf{U}_t) > 0$ , we can conclude that  $\{g_t(\mathbf{U}_t)\}_{t>0}$  is quasi-martingale and converges almost surely.  $\square$

**Lemma 4.** (Convergence of the empirical loss function  $f_t(\mathbf{U})$ ): The empirical loss functions  $f_t(\mathbf{U}_t)$  and its surrogate  $g_t(\mathbf{U}_t)$  converge to the same limit, i.e.,

$$g_t(\mathbf{U}_t) \xrightarrow{a.s.} f_t(\mathbf{U}_t).$$

*Proof Sketch.* We begin the proof with providing the following inequality:

$$\frac{g_t(\mathbf{U}_t) - f_t(\mathbf{U}_t)}{t+1} \leq \underbrace{u_t - u_{t+1}}_{(S-1)} + \underbrace{\frac{\ell(\mathbf{U}_t, \mathbf{P}_{t+1}, \mathbf{x}_{t+1}) - f_t(\mathbf{U}_t)}{t+1}}_{(S-2)},$$

where  $u_t \triangleq g_t(\mathbf{U}_t)$ . We then prove that the two sequences (S-1)-(S-2) converge almost surely. As a result, the sequence  $\{(g_t(\mathbf{U}_t) - f_t(\mathbf{U}_t)) \frac{1}{t+1}\}$  also convergence almost surely, i.e.,

$$\sum_{t=0}^{\infty} (g_t(\mathbf{U}_t) - f_t(\mathbf{U}_t)) \frac{1}{t+1} < \infty.$$

In parallel, we exploit that the real sequence  $\{\frac{1}{t+1}\}_{t \geq 0}$  diverges, i.e.,  $\sum_{t=1}^{\infty} \frac{1}{t+1} = \infty$ . It implies that  $g_t(\mathbf{U}_t) - f_t(\mathbf{U}_t)$  converges.  $\square$

**Corollary 1.** The expected loss function  $\{f(\mathbf{U}_t)\}_{t=1}^{\infty}$  converges almost surely when  $t \rightarrow \infty$ .

*Proof.* Since  $f_t(\mathbf{U}_t) \xrightarrow{a.s.} f(\mathbf{U}_t)$  and  $g_t(\mathbf{U}_t) \xrightarrow{a.s.} f_t(\mathbf{U}_t)$ , then  $g_t(\mathbf{U}_t) \xrightarrow{a.s.} f(\mathbf{U}_t)$ . Since  $g_t(\mathbf{U}_t)$  converges almost surely,  $f(\mathbf{U}_t)$  also converges almost surely when  $t \rightarrow \infty$ .  $\square$

**Corollary 2.** When  $t \rightarrow \infty$ , let  $\mathbf{U}_t = \underset{\mathbf{U} \in \mathbb{R}^{n \times r}}{\operatorname{argmin}} g_t(\mathbf{U})$ , we have

$$f_t(\mathbf{U}_t) \leq f_t(\mathbf{U}) + \frac{L}{2} \|\mathbf{U} - \mathbf{U}_t\|_F^2, \forall \mathbf{U} \in \mathbb{R}^{n \times r},$$

where  $L$  is a positive constant. In other words,  $\mathbf{U}_t$  is the minimum point of  $f(\mathbf{U})$ .

*Proof Sketch.* Let us denote the error function  $e_t(\mathbf{U}) = g_t(\mathbf{U}) - f_t(\mathbf{U})$ .

Due to  $g_t(\mathbf{U}_t) \xrightarrow{a.s.} f_t(\mathbf{U}_t)$  when  $t \rightarrow \infty$ , we have  $\nabla e_t(\mathbf{U}_t) = \mathbf{0}$  and hence the following inequality

$$\|\nabla e_t(\mathbf{U})\| \leq \frac{L}{2} \|\mathbf{U} - \mathbf{U}_t\|_F.$$

It is therefore that

$$\frac{|e_t(\mathbf{U}) - e_t(\mathbf{U}_t)|}{\|\mathbf{U} - \mathbf{U}_t\|_F} \leq \frac{L}{2} \|\mathbf{U} - \mathbf{U}_t\|_F,$$

thanks to the mean value theorem. In other word, we have  $|e_t(\mathbf{U})| \leq \frac{L}{2} \|\mathbf{U} - \mathbf{U}_t\|_F^2$  because of  $e_t(\mathbf{U}_t) \xrightarrow{a.s.} 0$ .

In addition, for all  $\mathbf{U} \in \mathbb{R}^{n \times r}$ , we always have  $f_t(\mathbf{U}_t) \leq g_t(\mathbf{U})$ . Therefore, we can conclude the corollary as follows

$$\begin{aligned} f_t(\mathbf{U}_t) \leq g_t(\mathbf{U}_t) &= f_t(\mathbf{U}) + e_t(\mathbf{U}) \\ &\leq f_t(\mathbf{U}) + \frac{L}{2} \|\mathbf{U} - \mathbf{U}_t\|_F^2. \end{aligned}$$

It ends the proof.  $\square$

## V. EXPERIMENTS

In this section, we evaluate the performance of the proposed algorithm by comparing it to the state-of-the-art in three scenarios relative to: robust subspace tracking, robust matrix completion and video background-foreground separation respectively. In particular, extensive experiments on simulated data are conducted to demonstrate the convergence and robustness of our PETRELS-ADMM algorithm for subspace tracking

and matrix completion. While four real video sequences are used to illustrate the effectiveness of PETRELS-ADMM for background-foreground separation.

### A. Robust Subspace Tracking

In the following experiments, data  $\mathbf{x}_t$  at each time  $t$  is generated randomly using the standard signal model as in (1)

$$\mathbf{x}_t = \mathbf{P}_t(\mathbf{U}\boldsymbol{\omega}_t + \mathbf{n}_t + \mathbf{s}_t),$$

where  $\mathbf{U} \in \mathbb{R}^{n \times r}$  denotes a mixing matrix,  $\boldsymbol{\omega}_t$  is a random vector living on  $\mathbb{R}^r$  space (i.e.,  $\boldsymbol{\ell}_t = \mathbf{U}\boldsymbol{\omega}_t$ ) and they are Gaussian i.i.d. of pdf  $\mathcal{N}(0,1)$ ;  $\mathbf{n}_t$  represents the white Gaussian noise  $\mathcal{N}(0, \sigma^2)$ , with  $\text{SNR} = -10 \log_{10}(\sigma^2)$  is the signal-to-noise ratio to control the impact of noise on algorithm performance; and  $\mathbf{s}_t$  is uniform i.i.d. over  $[0, \text{fac-outlier}]$  given the magnitude fac-outlier of outliers that aim to create a space for outliers. Indices of missing entries and outliers are generated randomly using the Bernoulli model with the probability  $\omega_{\text{missing}}$  and  $\omega_{\text{outlier}}$  respectively. The two probabilities represent the density of missing entries and outliers in the data.

In order to evaluate the subspace estimation accuracy, we use the subspace estimation performance (SEP) [19] metric

$$\text{SEP} = \frac{1}{L} \sum_{i=1}^L \frac{\text{tr} \{ \mathbf{U}_{\text{es-}i}^\# (\mathbf{I} - \mathbf{U}_{\text{ex}} \mathbf{U}_{\text{ex}}^\#) \mathbf{U}_{\text{es-}i} \}}{\text{tr} \{ \mathbf{U}_{\text{es-}i}^\# (\mathbf{U}_{\text{ex}} \mathbf{U}_{\text{ex}}^\#) \mathbf{U}_{\text{es-}i} \}},$$

where  $L$  is the number of independent runs,  $\mathbf{U}_{\text{ex}}$  and  $\mathbf{U}_{\text{es-}i}$  are the true and the estimated subspaces at the  $i$ -th run respectively. Particularly, the denominator measures the sum of the squares of the cosines of the principal angles between  $\mathbf{U}_{\text{es-}i}$  and  $\mathbf{U}_{\text{ex}}$ , while the numerator evaluates the similar sum but for the two subspaces  $\mathbf{U}_{\text{es-}i}$  and the orthogonal complement  $\mathbf{U}_{\text{ex}}^\perp$ . Accordingly, the lower SEP is, the better the algorithm performance is.

State-of-the-art algorithms for comparison are: GRASTA [15], ROSETA [18] and PETRELS-CFAR [19], ReProCS [20] and NORST [21]. Throughout our experiments, their algorithm parameters are set by default as mentioned in the algorithms. In particular, we set a penalty parameter  $\rho = 1.8$  and a constant step-size scale  $C = 2$  in GRASTA. An adaptive step size of ROSETA is initialized at  $\mu_0 = \frac{C}{1+\eta_0}$  with  $C = 8$  and  $\eta_0 = 99$ , while two thresholds for controlling the step size are set at  $\eta_{\text{low}} = 50$  and  $\eta_{\text{high}} = 100$ . PETRELS-CFAR includes a forgetting factor set at  $\lambda = 0.999$ , a window size  $N_w = 150$  and a false alarm probability  $P_{\text{fa}}$  varied from  $[0.1, 0.7]$  depended on the outlier intensity. Both ReProCS and NORST require several predefined parameters, including  $t_{\text{train}} = 200$  data samples,  $\alpha = 60$ ,  $K = 33$  and  $\omega_{\text{eval}} = 7.8 \times 10^{-4}$ . For our algorithm, we set the penalty parameters at 1.5, the regularization parameter  $\alpha = 0.1$  and the step-size threshold  $\eta = \sin(\pi/3)$ , while the maximum number of iterations for outlier detection phase is fixed at  $K = 50$ . Matlab codes are available online<sup>7</sup>. The experimental results are averaged over 100 independent runs.

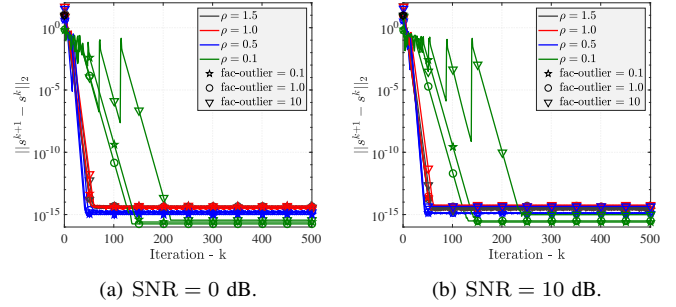


Fig. 2: Convergence of PETRELS-ADMM in terms of the variation  $\|\mathbf{s}^{k+1} - \mathbf{s}^k\|_2$ :  $n = 50, r = 2$ , 90% entries observed and outlier density  $\omega_{\text{outlier}} = 0.1$ .

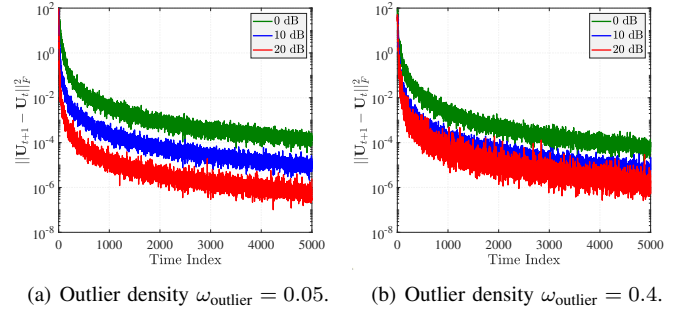


Fig. 3: Convergence of PETRELS-ADMM in terms of the variation  $\|\mathbf{U}_{t+1} - \mathbf{U}_t\|_F$ :  $n = 50, r = 2$ , 90% entries observed and outlier intensity fac-outlier = 10.

1) *Convergence of PETRELS-ADMM*: To demonstrate the convergence of our algorithm, we use a synthetic data whose number of row  $n = 50$ , rank  $r = 2$ , and 5000 vector samples with 90% entries observed on average. Specifically, the outlier density  $\omega_{\text{outlier}}$  is varied from 0.05 to 0.4, while the outlier intensity is set at three values representing a low, medium and high level (i.e., fac-outlier = 0.1, 1 and 10 respectively). The penalty parameter  $\rho$  varies in the range  $[0.1, 1.5]$ . Also, two noise levels are considered, with  $\text{SNR} \in \{0, 10\}$  dB. The results are shown as in Fig. 2 and Fig. 3.

Fig. 2 shows the convergence behavior of PETRELS-ADMM w.r.t the two variables: fac-outlier and the weight  $\rho$ . We can see that, the variation of  $\{\mathbf{s}^k\}_{k \geq 1}$  always converges in all testing cases. When the penalty parameter  $\rho \geq 0.5$ , the convergence rate is fast, i.e. the variation  $\|\mathbf{s}^{k+1} - \mathbf{s}^k\|_2$  can converge in 50 iterations in both low- and high-noise cases. The results are practical evidences of Lemma 1. Similarly, Fig. 3 shows that the convergence of the variations of the sequence  $\{\mathbf{U}_t\}_{t \geq 0}$ , generated by PETRELS-ADMM follows the theoretical behavior proved in Lemma 2, that is,  $\|\mathbf{U}_t - \mathbf{U}_{t+1}\|_F \xrightarrow{a.s.} \mathcal{O}(\frac{1}{t})$  almost surely.

2) *Outlier Detection*: Following the above experiment, we next assess the ability of PETRELS-ADMM for outlier detection against the noise level. The three statistical metrics including Sensitivity (SEN) and Specificity (SEP) and Accuracy (ACC) are used to evaluate its outlier detection performance [42]. Particularly, SEN measures the percentage of outliers detected correctly over the total outliers in the measurement data. SEP is similar to SEN, but for normal entries and ACC indicates how the estimator makes the

<sup>7</sup>GRASTA: <https://sites.google.com/site/hejunzz/grasta>  
ROSETA: <http://www.merl.com/research/license#ROSETA>  
ReProCS: <https://github.com/praneethmurthy/ReProCS>  
Our code: <https://avitech.uet.vnu.edu.vn/en/petrels-admm>

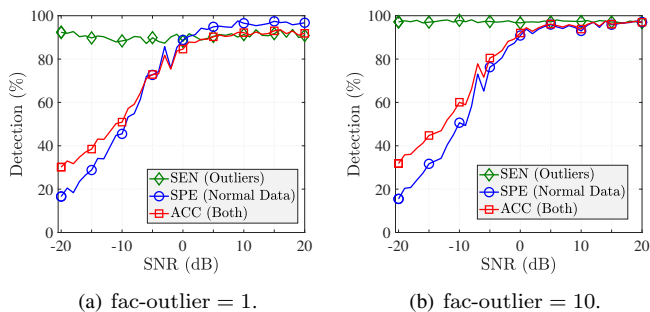


Fig. 4: Outlier detection accuracy versus the noise level:  $n = 50$ ,  $r = 2$ , 80% entries observed and 20% outliers.

correct detection. We use the same data above, but 20% of the observations are missing. The outlier density  $\omega_{\text{outlier}}$  is set at 0.2, while two intensity levels are considered, with  $\text{fac-outlier} \in \{1, 10\}$ .

Fig. 4 illustrates the outlier detection performance of PETRELS-ADMM versus the noise level SNR. As can be seen that when we increase the value of SNR from  $-20$  dB to  $20$  dB, the detection accuracy goes up first and then converges towards a constant level. At very low SNRs (i.e.,  $< 0$  dB), the proposed algorithm does not work well in which many normal entries are labeled as outliers, although the number of correctly detected outliers are high. When  $\text{SNR} > 0$  dB, PETRELS-ADMM achieves a competitive prediction accuracy with respect to all three evaluation metrics.

Fig. 5 provides more practical evidences to demonstrate the effectiveness of PETRELS-ADMM for the outlier detection. Particularly, the locations of outliers  $s_t$  are well detected even when the measurement data is corrupted by noise with a moderate SNR value (e.g. 10 dB). Also, amplitude of the outliers is recovered nearly correctly with a small relative error ( $\text{RE} = \frac{\|s_t - \hat{s}_t\|_2}{\|s_t\|_2}$ ) in both cases (e.g.  $\text{RE} = 0.0616$  at the 20 dB noise level). As a result, the corrupted signals are also well reconstructed, as illustrated in Fig. 5(b) and (d).

3) *Robustness of PETRELS-ADMM*: To investigate the robustness of PETRELS-ADMM, we vary the outlier intensity, density and missing density and then measure the SEP metric. Moreover, we also demonstrate the effectiveness of PETRELS-ADMM against noisy and time-varying environments.

a) *Impact of outlier intensity on algorithm performance*:

We fix  $n = 50$ ,  $r = 2$ , 90% entries observed, outlier density  $\omega_{\text{outlier}} = 0.1$ ,  $\text{SNR} = 20$  dB while varying  $\text{fac-outlier}$  in the range  $[0.1, 10]$ . We can see from Fig. 6 that PETRELS-ADMM always outperforms other state-of-the-art algorithms in all testing cases with different  $\text{fac-outlier}$  values. At low outlier intensity (i.e.,  $\text{fac-outlier} \leq 1$ ), all algorithms yield good accuracy with fast convergences, though ROSETA and ReProCS obtain the higher SEP (i.e.,  $\approx 10^{-3}$ ) as compared to that of the four remaining algorithms. In particular, PETRELS-ADMM provides the best subspace estimation accuracy, i.e.,  $\text{SEP} \approx 10^{-5}$  in both cases (see Fig. 6(a)-(b)). At a high intensity level (e.g.  $\text{fac-outlier} = 5$  or  $10$ ), PETRELS-ADMM again provides the best performance in terms of both convergence rate and accuracy. GRASTA performs similarly to ReProCS and slightly worse than PETRELS-CFAR (i.e., their

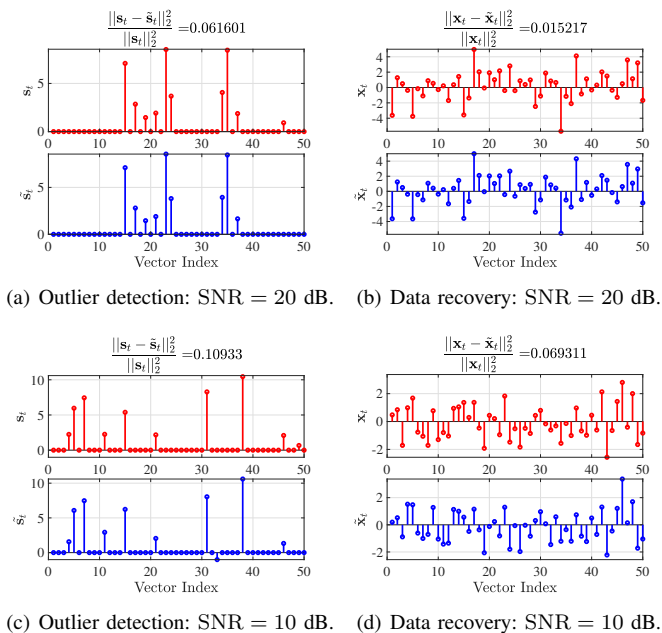


Fig. 5: Outlier detection and data reconstruction:  $n = 50$ ,  $r = 2$ , 90% entries observed, outlier intensity  $\text{fac-outlier} = 1$ , and outlier density  $\omega_{\text{outlier}} = 0.1$ .

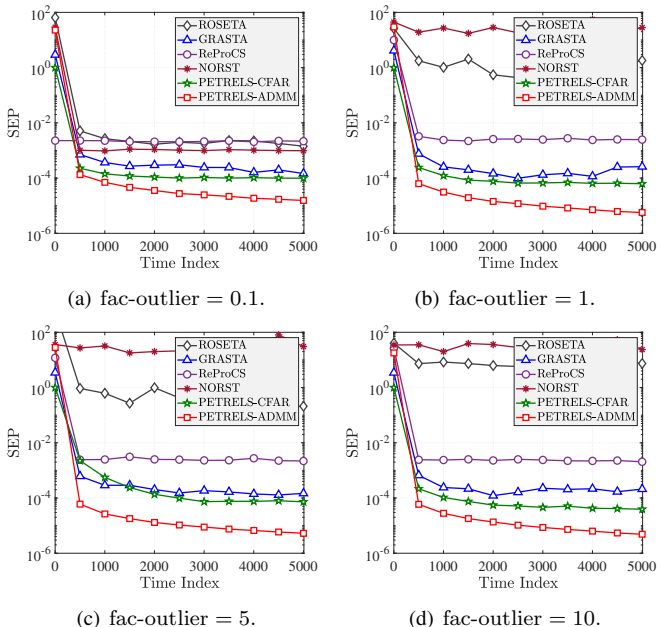


Fig. 6: Impact of outlier intensity on algorithm performance:  $n = 50$ ,  $r = 2$ , 90% entries observed, outlier density  $\omega_{\text{outlier}} = 0.1$  and  $\text{SNR} = 20$  dB.

SEP values are around  $10^{-4}$ ). While ROSETA and NORST fail to recover the underlying subspace in the presence of strong outliers. Note that, in all four experiments above, PETRELS-ADMM always obtains the best SEP value of around  $10^{-5}$  and hence is robust to outlier intensity.

b) *Impact of outlier density on algorithm performance*:

We fix  $n = 50$ ,  $r = 2$ , 90% entries observed, outlier intensity  $\text{fac-outlier} = 5$ ,  $\text{SNR} = 20$  dB while varying the outlier density  $\omega_{\text{outlier}}$  in the range  $[0.05, 0.4]$ . The results are shown as in Fig. 7. PETRELS-ADMM outperforms the

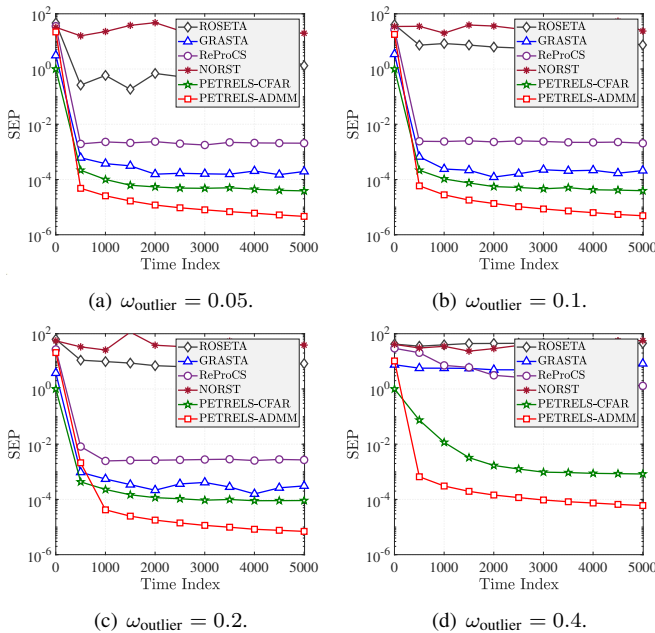


Fig. 7: Impact of outlier density on algorithm performance:  $n = 50$ ,  $r = 2$ , 90% entries observed, outlier intensity fac-outlier = 10 and SNR = 20 dB.

four remaining algorithms in this context. In particular, our algorithm performs very well even when the fraction of outliers is high (e.g.  $\omega_{\text{outlier}} = 0.4$ ). By contrast, four algorithms including GRASTA, ROSETA, ReProCS and NORST may fail to track subspace in the case of a high outlier density (see Fig. 7(d)). The PETRELS-CFAR works well but has a lower convergence rate and accuracy in terms of SEP metric as compared to PETRELS-ADMM. When the measurement data is corrupted by a smaller number of outliers, PETRELS-ADMM still provides better performance than the others, as shown in Fig. 7 (a)-(c).

*c) Impact of the density of missing entries on algorithm performance:* Following the above experiments, we change the number of missing entries in the measurement data by varying the probability  $\omega_{\text{missing}}$  while fixing the other attributes. The results are reported in Fig. 8 and Fig. 9. In particular, the effect of  $\omega_{\text{missing}}$  on algorithm performance is presented in Fig. 8. Similarly, PETRELS-ADMM yields the best performance in four cases of missing observations. Three algorithms including PETRELS-CFAR, GRASTA and ReProCS provide good performance but with slower convergence rate and accuracy, while ROSETA and NORST have failed again in this task due to the high outlier intensity (i.e., fac-outlier = 10). As can be seen from Fig. 9(a)-(c) that the state-of-the-art algorithms only perform well when the number of corruptions is smaller than half the number of entries in the data measurement. While PETRELS-ADMM still obtains the reasonable subspace estimation performance in terms of SEP (i.e.,  $\approx 10^{-3}$ ) in the case of very high corruptions, see Fig. 9(d).

*d) Noisy and Time-Varying Environments:* We first investigate the effect of the noise on the performance of PETRELS-ADMM in comparison with the state-of-the-art algorithms. We vary the value of SNR in the range from 0 dB to

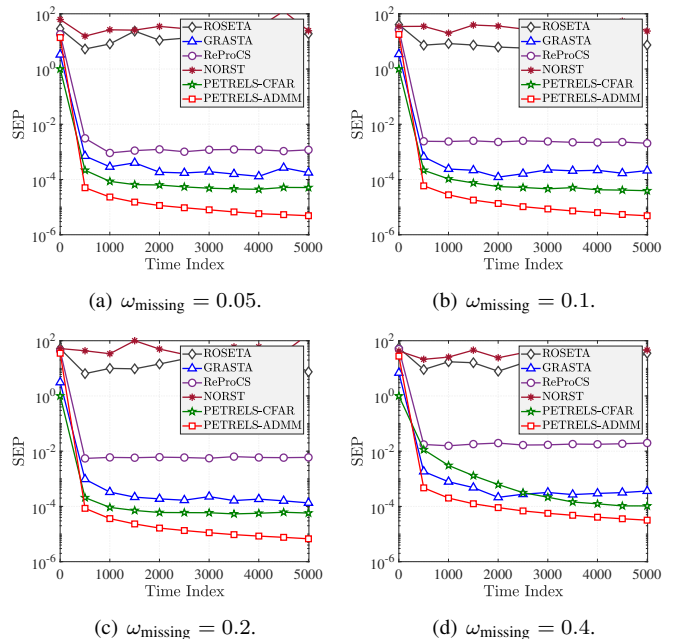


Fig. 8: Impact of the density of missing entries on algorithm performance:  $n = 50$ ,  $r = 2$ , outlier density  $\omega_{\text{outlier}} = 0.1$ , outlier intensity fac-outlier = 10 and SNR = 20 dB.

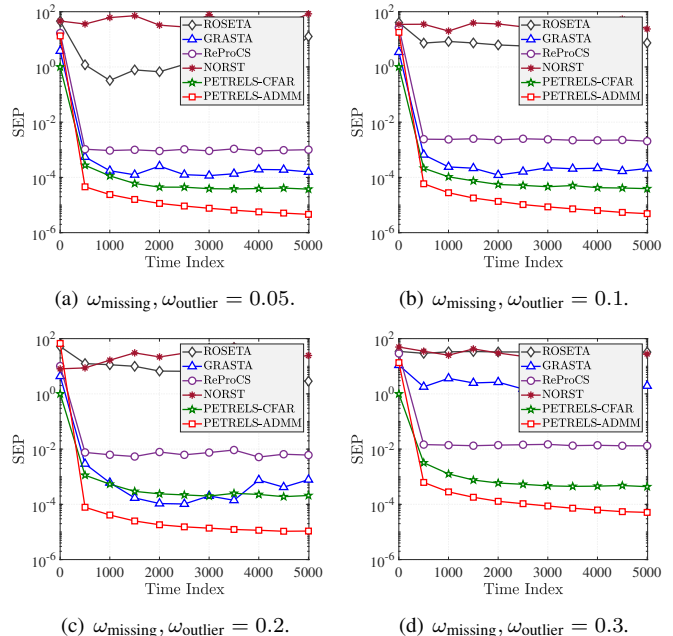


Fig. 9: Impact of the corruption fraction by missing data and outliers on algorithm performance:  $n = 50$ ,  $r = 2$  and fac-outlier = 10 and SNR = 20 dB.

20 dB and assess their performance on the same data above. Experimental results are illustrated in Fig. 10. As can be seen that the convergence rate of PETRELS-ADMM is not affected by SNR, but only its estimation accuracy, as shown in Fig. 10(a). Specifically, when we decrease the value of SNR, the estimation error between the true subspace and the estimation increases gradually. At a high SNR level (e.g. 20 dB), previous experiments indicate that PETRELS-ADMM outperforms state-of-the-art algorithms, see Fig. 6-9. At a low



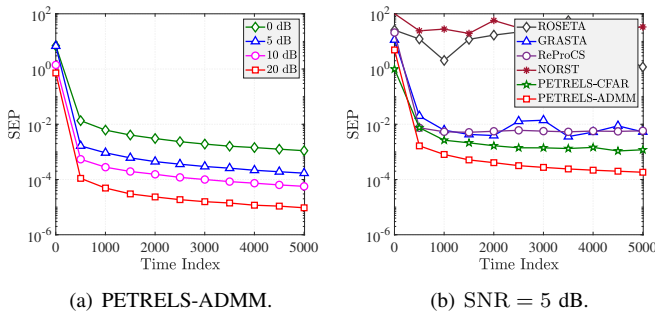


Fig. 10: Impact of the additive noise on algorithm performance:  $n = 50, r = 2$ , 90% entries observed and 10% outliers with intensity fac-outlier = 10.

SNR level (e.g. 5 dB), PETRELS-ADMM yields the best estimation accuracy as well as convergence rate again, as illustrated in Fig. 10(b). Similar outstanding performance of PETRELS-ADMM were also observed at lower SNR levels of 10, 5 or 0 dB (please see Figs. 8-10 of the supplementary material).

The robustness of PETRELS-ADMM is next investigated against nonstationary and time-varying environments. Particularly, the true subspace  $\mathbf{U}$  is supposed to be varying with time under the model  $\mathbf{U}_t = (1 - \varepsilon)\mathbf{U}_{t-1} + \varepsilon\mathbf{N}_t$ , where  $\mathbf{N}_t \in \mathbb{R}^{n \times r}$  is a Gaussian noise matrix (zero-mean and unit-variance) and  $\varepsilon$  is to control the subspace change which is chosen among  $\{10^{-1}, 10^{-2}, 10^{-3}\}$ . We use the same signal model as in the previous tasks and 1000 vector samples. Also, we create an abrupt change at  $t = 500$  to see how fast the proposed algorithm can converge. We measure the performance of PETRELS-ADMM at two noise levels (SNR = 5 and 10 dB) with different corruption fractions. Experimental results are illustrated in Fig. 11(a)-(d). In the same manner to the effect of the noise, the time-varying factor  $\varepsilon$  does not affect the convergence rate of PETRELS-ADMM, but only its subspace estimation. Fig. 11 shows that the estimation accuracy of the proposed algorithm will decrease if the time-varying factor  $\varepsilon$  increases. When the underlying subspace varies slowly (e.g.  $\varepsilon \leq 10^{-2}$ ), the resulting values of SEP, which always converge towards an error floor, indicate that PETRELS-ADMM can be robust to slowly time-varying scenarios.

### B. Robust Matrix Completion

We compare here the robust matrix completion (RMC) performance using PETRELS-ADMM with GRASTA [15], LRGeomGC [43] and RPCA-GD [44].

The measurement data  $\mathbf{X} = \mathbf{P} \circledast (\mathbf{U}\mathbf{W} + \mathbf{S} + \mathbf{N})$  used for this task corresponds to the rank-2 matrices of size of  $400 \times 400$ , where the operator  $\circledast$  denotes the Hadamard product. Particularly, we generated the mixing matrix  $\mathbf{U} \in \mathbb{R}^{400 \times 2}$  and the coefficient matrix  $\mathbf{W} \in \mathbb{R}^{2 \times 400}$  at random. Their entries were random variables that follow Gaussian distribution with zero mean and unit variance. The measurement data  $\mathbf{X}$  was corrupted by a white Gaussian noise  $\mathbf{N} \in \mathbb{R}^{400 \times 400}$  whose SNR is fixed at 40 dB. In the literature, the SNR value of around 40 dB is used for performance evaluation of completion algorithms due to missing observations and/or outliers at

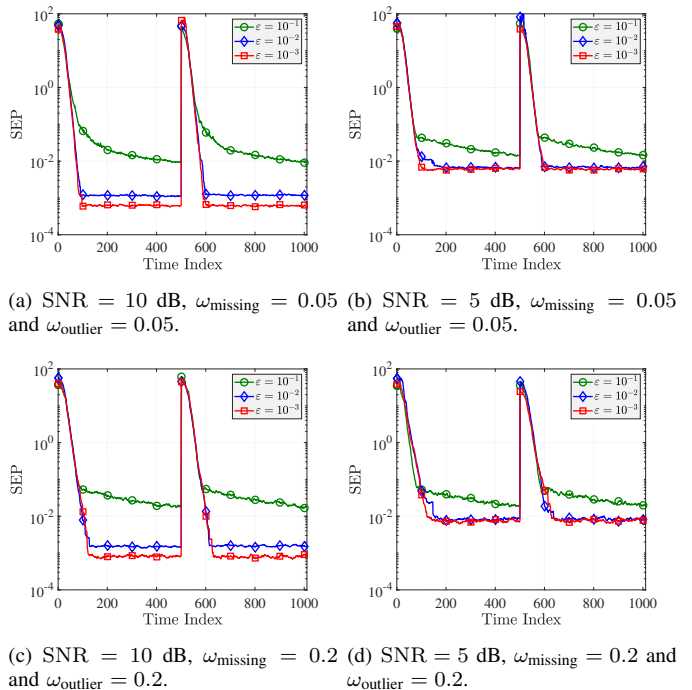


Fig. 11: PETRELS-ADMM in time-varying scenarios.

low-noise conditions [45]. The data matrix was affected by different percentages of missing ( $\mathbf{P}$ ) and outliers ( $\mathbf{S}$ ) from 0% – 90%. The location and value of corrupted entries (including missing and outliers) were uniformly distributed.

Fig. 12 shows that the proposed algorithm of PETRELS-ADMM based RMC outperforms GRASTA, LRGeomGC and RPCA-GD. At low outlier intensity (i.e., fac-outlier = 0.1), PETRELS-ADMM based RMC, LRGeomGC and RPCA-GD provide excellent performance even when the data is corrupted by a very high corruption fraction. At high outlier intensity (i.e., fac-outlier  $\geq 1$ ), PETRELS-ADMM based RMC provides the best matrix reconstruction error performance, GRASTA still retain good performance, while RPCA-GD and LRGeomGC fail to recover corrupted entries.

### C. Video Background/Foreground Separation

We further illustrate the effectiveness of the proposed PETRELS-ADMM algorithm in the application of RST for video background/foreground separation, and compare with GRASTA and PETRELS-CFAR. We use four real video sequences for this task, including Hall, Lobby, Sidewalk and Highway datasets. In particular, the two former datasets are from GRASTA's homepage<sup>8</sup>, while the two latter datasets are from CD.net2012<sup>9</sup> [46]. The Hall dataset consists of 3584 frames of size  $174 \times 144$  pixels, while the Lobby dataset has 1546 frames of size  $144 \times 176$  pixels. The Sidewalk dataset includes 1200 frames of size  $240 \times 352$  pixels. Highway dataset has 1700 frames of size  $240 \times 320$  pixels. We can see from Fig. 13, PETRELS-ADMM is capable of detecting objects in video and provides competitive performance as compared to GRASTA and PETRELS-CFAR.

<sup>8</sup><https://sites.google.com/site/hejunzz/grasta>

<sup>9</sup><http://jacarini.dinf.usherbrooke.ca/dataset2012>

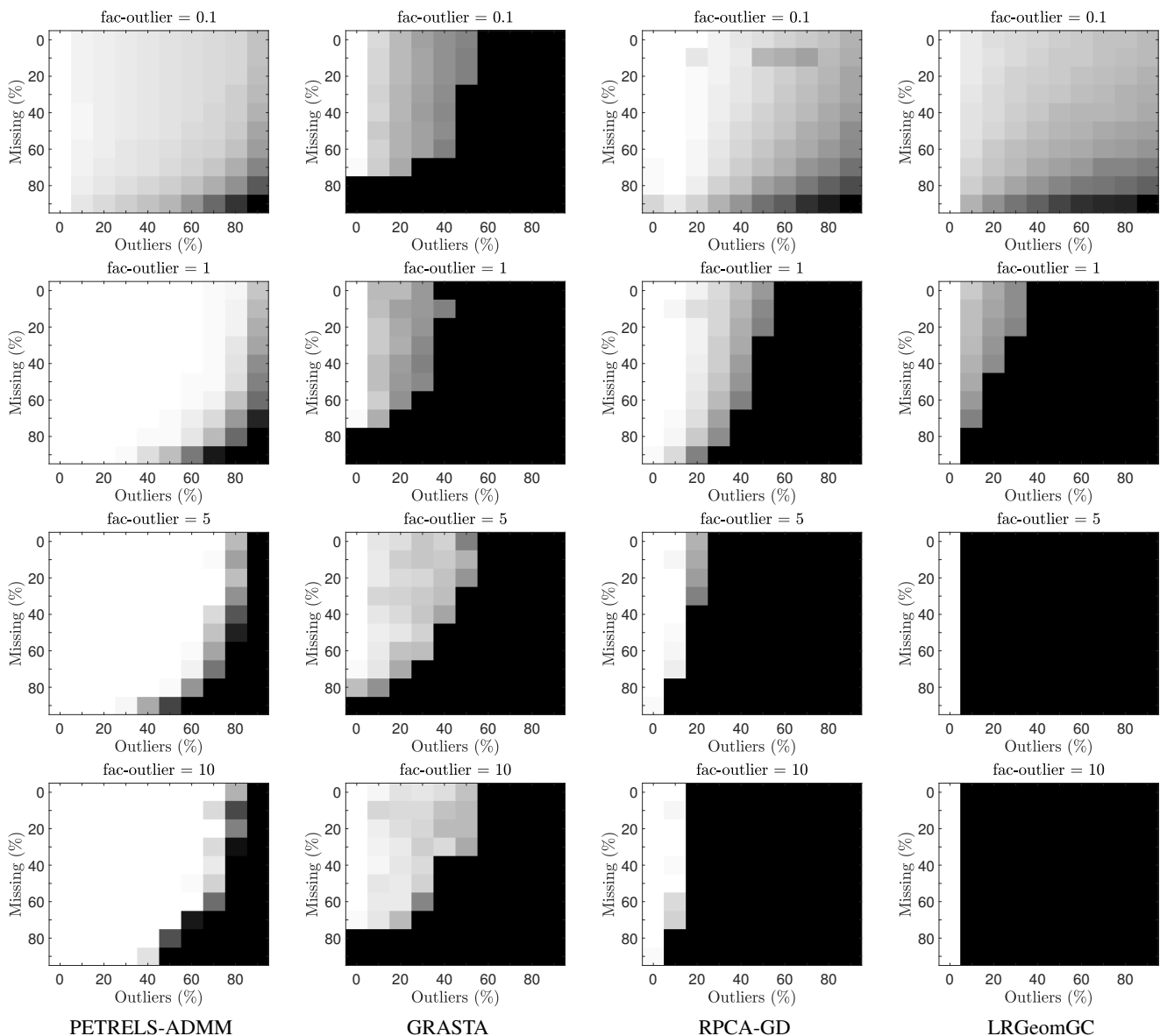


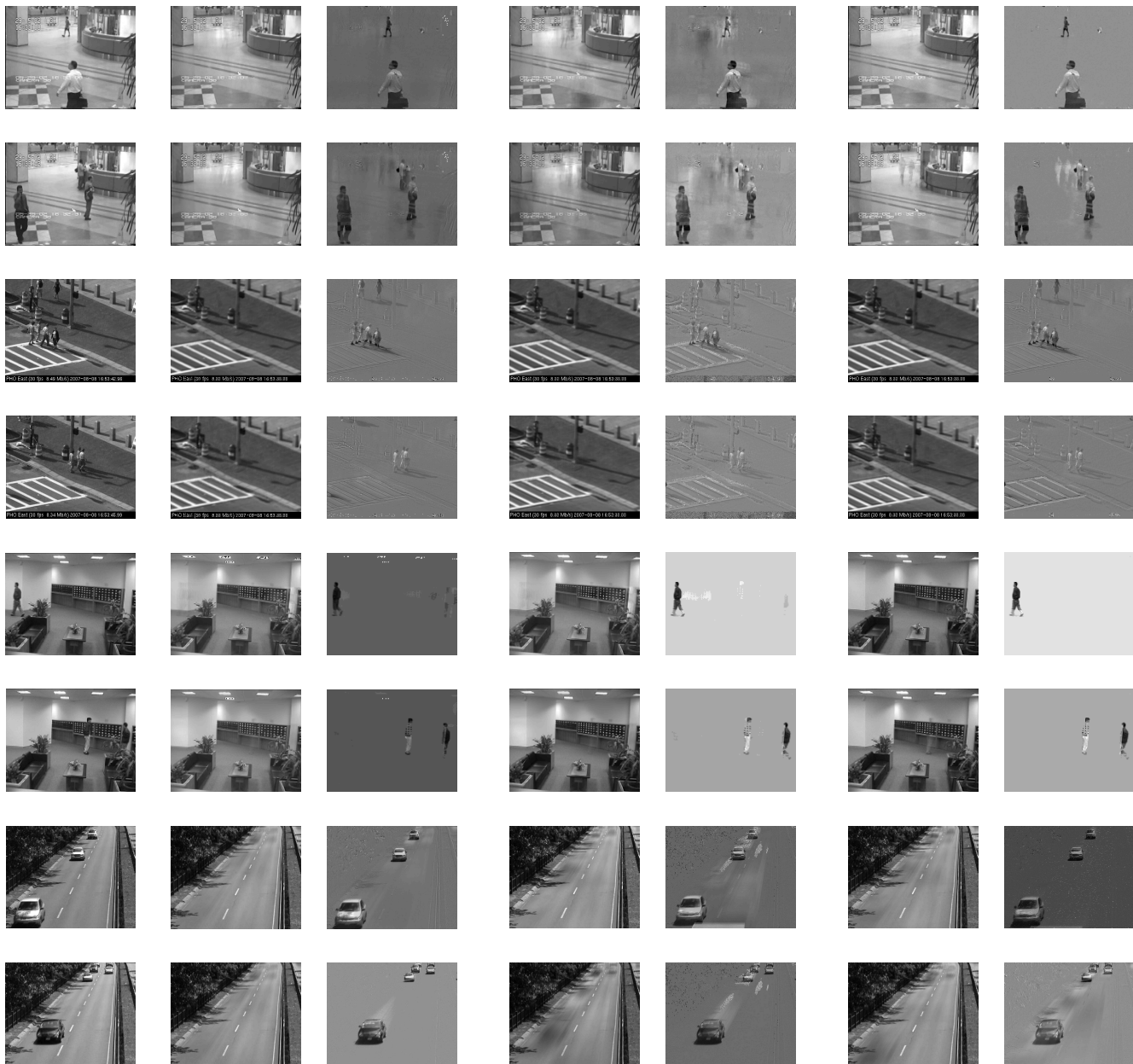
Fig. 12: Effect of outlier intensity on robust matrix completion performance. White color denotes perfect recovery, black color denotes failure and gray colour is in between.

## VI. CONCLUSIONS

In this paper, we have proposed an efficient algorithm, namely PETRELS-ADMM, for the robust subspace tracking problem to handle missing data in the presence of outliers. By converting the original RST problem to a surrogate one, which facilitates the tracking ability, we have derived an online implementation for outlier rejection with a low computational complexity and a fast convergence rate while still retaining a high subspace estimation performance. We have established a theoretical convergence which guarantees that the solutions generated by PETRELS-ADMM will converge to a stationary point asymptotically. The simulation results have suggested that our algorithm is more effective than the state-of-the-art algorithms for robust subspace tracking and robust matrix completion. The effectiveness of PETRELS-ADMM was also verified for the problem of video background-foreground separation.

## REFERENCES

- [1] A. Tulay and H. Simon, *Adaptive signal processing: Next generation solutions*. John Wiley & Sons, 2010.
- [2] N. Vaswani, Y. Chi, and T. Bouwmans, “Rethinking PCA for modern data sets: Theory, algorithms, and applications,” *Procc. IEEE*, vol. 106, no. 8, pp. 1274–1276, 2018.
- [3] G. H. Golub and C. F. Van Loan, *Matrix computations*. JHU press, 2012.
- [4] P. Comon and G. H. Golub, “Tracking a few extreme singular values and vectors in signal processing,” *Procc. IEEE*, vol. 78, no. 8, pp. 1327–1343, 1990.
- [5] I. T. Jolliffe and J. Cadima, “Principal component analysis: A review and recent developments,” *Philos. Trans. Royal Soc. A*, vol. 374, no. 2065, p. 20150202, 2016.
- [6] C. Wang, Y. C. Eldar, and Y. M. Lu, “Subspace estimation from incomplete observations: A high-dimensional analysis,” *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 6, pp. 1240–1252, 2018.
- [7] L. Balzano, Y. Chi, and Y. M. Lu, “Streaming PCA and subspace tracking: The missing data case,” *Procc. IEEE*, vol. 106, no. 8, pp. 1293–1310, Aug 2018.
- [8] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, “Robust subspace learning: Robust PCA, robust subspace tracking, and robust



Video Frame

PETRELS-ADMM

PETRELS-CFAR

GRASTA

Fig. 13: Qualitative illustration of video background-foreground separation application.

- subspace recovery,” *IEEE Signal Process. Mag.*, vol. 35, no. 4, pp. 32–55, 2018.
- [9] M. Mardani, G. Mateos, and G. B. Giannakis, “Subspace learning and imputation for streaming big data matrices and tensors,” *IEEE Trans. Signal Process.*, vol. 63, no. 10, pp. 2663–2677, 2015.
- [10] V. Nguyen, K. Abed-Meraim, N. Linh-Trung, and R. Weber, “Generalized minimum noise subspace for array processing,” *IEEE Trans. Signal Process.*, vol. 65, no. 14, pp. 3789–3802, 2017.
- [11] S. Haghghatshoar and G. Caire, “Low-complexity massive MIMO subspace estimation and tracking from low-dimensional projections,” *IEEE Trans. Signal Process.*, vol. 66, no. 7, pp. 1832–1844, 2018.
- [12] S. Buzzi and C. D’Andrea, “Subspace tracking and least squares approaches to channel estimation in millimeter wave multiuser MIMO,” *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 6766–6780, 2019.
- [13] L. Balzano, R. Nowak, and B. Recht, “Online identification and tracking of subspaces from highly incomplete information,” in *Ann. Allerton Conf. Commun., Cont. Comput.* IEEE, 2010, pp. 704–711.
- [14] D. Zhang and L. Balzano, “Global convergence of a Grassmannian gradient descent algorithm for subspace estimation,” in *Int. Conf. Art. Intel. Stat.*, Cadiz, Spain, 2016, pp. 1460–1468.
- [15] J. He, L. Balzano, and A. Szlam, “Incremental gradient on the Grassmannian for online foreground and background separation in subsampled video,” in *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.* IEEE, 2012, pp. 1568–1575.
- [16] Y. Chi, Y. C. Eldar, and R. Calderbank, “PETRELS: Parallel subspace estimation and tracking by recursive least squares from partial observations,” *IEEE Trans. Signal Process.*, vol. 61, no. 23, pp. 5947–5959, 2013.
- [17] B. Yang, “Projection approximation subspace tracking,” *IEEE Trans. Signal Process.*, vol. 43, no. 1, pp. 95–107, 1995.
- [18] H. Mansour and X. Jiang, “A robust online subspace estimation and tracking algorithm,” in *IEEE Int. Conf. Acoust. Speech Signal Process.* IEEE, 2015, pp. 4065–4069.
- [19] N. Linh-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.
- [20] 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.

- [21] P. Narayanamurthy, V. Daneshpajoo, and N. Vaswani, “Provable subspace tracking from missing data and matrix completion,” *IEEE Trans. Signal Process.*, vol. 67, no. 16, pp. 4245–4260, 2019.
- [22] C. Hage and M. Kleinstueber, “Robust PCA and subspace tracking from incomplete observations using  $\ell_0$ -surrogates,” *Comput. Stat.*, vol. 29, no. 3–4, pp. 467–487, 2014.
- [23] S. Chouvardas, Y. Kopsinis, and S. Theodoridis, “Robust subspace tracking with missing entries: The set-theoretic approach,” *IEEE Trans. Signal Process.*, vol. 63, no. 19, pp. 5060–5070, 2015.
- [24] A. Gonen, D. Rosenbaum, Y. C. Eldar, and S. Shalev-Shwartz, “Subspace learning with partial information,” *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1821–1841, 2016.
- [25] P. V. Giampouras, A. A. Rontogiannis, K. E. Themelis, and K. D. Koutroumbas, “Online sparse and low-rank subspace learning from incomplete data: A Bayesian view,” *Signal Process.*, vol. 137, pp. 199–212, 2017.
- [26] L. T. Thanh, V.-D. Nguyen, N. Linh-Trung, and K. Abed-Meraim, “Robust subspace tracking with missing data and outliers via ADMM,” in *European Signal Process. Conf.*, 2019, pp. 1–5.
- [27] M. Shor and N. Levanon, “Performances of order statistics CFAR,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 27, no. 2, pp. 214–224, 1991.
- [28] J. A. Tropp, “Just relax: convex programming methods for identifying sparse signals in noise,” *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 1030–1051, 2006.
- [29] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.
- [30] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, “Optimal parameter selection for the alternating direction method of multipliers (ADMM): Quadratic problems,” *IEEE Trans. Auto. Cont.*, vol. 60, no. 3, pp. 644–658, 2015.
- [31] Y. Xu, M. Liu, Q. Lin, and T. Yang, “ADMM without a fixed penalty parameter: Faster convergence with new adaptive penalization,” in *Adv. Neural Inf. Process. Syst.*, 2017, pp. 1267–1277.
- [32] W. Tian and X. Yuan, “An alternating direction method of multipliers with a worst-case  $\mathcal{O}(1/n^2)$  convergence rate,” *Math. Comput.*, vol. 88, no. 318, pp. 1685–1713, 2019.
- [33] N. Parikh, S. Boyd *et al.*, “Proximal algorithms,” *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, 2014.
- [34] W. W. Hager, “Updating the inverse of a matrix,” *SIAM Review*, vol. 31, no. 2, pp. 221–239, 1989.
- [35] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online learning for matrix factorization and sparse coding,” *J. Mach. Learn. Res.*, vol. 11, no. Jan, pp. 19–60, 2010.
- [36] J. Feng, H. Xu, and S. Yan, “Online robust PCA via stochastic optimization,” in *Adv. Neural Inf. Process. Syst.*, 2013, pp. 404–412.
- [37] J. Shen, H. Xu, and P. Li, “Online optimization for max-norm regularization,” *Machi. Learn.*, vol. 106, no. 3, pp. 419–457, 2017.
- [38] G. Li and T. K. Pong, “Global convergence of splitting methods for nonconvex composite optimization,” *SIAM J. Optim.*, vol. 25, no. 4, pp. 2434–2460, 2015.
- [39] Y. Wang, W. Yin, and J. Zeng, “Global convergence of ADMM in nonconvex nonsmooth optimization,” *J. Sci. Comput.*, vol. 78, no. 1, pp. 29–63, 2019.
- [40] L. Bottou, *On-Line Learning and Stochastic Approximations*. USA: Cambridge University Press, 1999.
- [41] A. W. Van der Vaart, *Asymptotic statistics*. Cambridge University Press, 2000.
- [42] D. M. Powers, “Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation,” *J. Mach. Learn. Tech.*, vol. 2, no. 1, pp. 37–63, 2011.
- [43] B. Vandereycken, “Low-rank matrix completion by Riemannian optimization,” *SIAM J. Optim.*, vol. 23, no. 2, pp. 1214–1236, 2013.
- [44] X. Yi, D. Park, Y. Chen, and C. Caramanis, “Fast algorithms for robust PCA via gradient descent,” in *Adv. Neural Inf. Process. Syst.*, 2016, pp. 4152–4160.
- [45] L. T. Nguyen, J. Kim, and B. Shim, “Low-rank matrix completion: A contemporary survey,” *IEEE Access*, vol. 7, pp. 94 215–94 237, 2019.
- [46] N. Goyette, P. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, “Changetection.net: A new change detection benchmark dataset,” in *IEEE Conf. Comput. Vis. Pattern Recognit.* IEEE, 2012, pp. 1–8.



**Le Trung Thanh** received B.Sc. and M.Sc. degrees in Electronics and Telecommunications from VNU University of Engineering and Technology, Vietnam National University, Hanoi (VNU) in 2016 and 2018 respectively. He is now pursuing his Ph.D. study at the University of Orleans, France. His research interests include signal processing, subspace tracking, tensor analysis, and system identification.



**Nguyen Viet Dung** received the B.Sc. in Electronics and Telecommunication Engineering from VNU University of Engineering and Technology, Vietnam National University, Hanoi (VNU) in 2009, M.Sc. in Networks and Telecommunications from École Normale Supérieure (ENS) de Cachan, Université Paris XI (now, University of Paris-Saclay, France), in 2012, and Ph.D. in Signal Processing from the University of Orleans (France) in 2016. He did a postdoc at CentraleSupélec, University of Paris-Saclay from 2017 to 2018. From 2019 to now, he is a research engineer at Lab-STICC, UMR 6285 CNRS ENSTA Bretagne, Brest, France. His research interests include channel modelling in array signal processing, adaptive matrix and tensor analysis, blind source separation, and statistical performance analysis.



**Nguyen Linh Trung** obtained his B.Eng. and Ph.D. degrees, both in Electrical Engineering, from Queensland University of Technology, Brisbane, Australia, in 1998 and 2005. Since 2006, he has been on the faculty of VNU University of Engineering and Technology, Vietnam National University, Hanoi (VNU), where he is currently an associate professor of electronic engineering in the Faculty of Electronics and Telecommunications and director of the Advanced Institute of Engineering and Technology. He is interested in signal processing methods, including time-frequency signal analysis, blind processing, adaptive filtering, compressive sampling, tensor-based signal analysis, graph signal processing, and apply them to wireless communication, networking, and biomedical engineering.



**Karim Abed-Meraim** was born in 1967. He received the State Engineering Degree from Ecole Polytechnique, Palaiseau, France, in 1990, the State Engineering Degree from Ecole Nationale Supérieure des Télécommunications (ENST), Paris, France, in 1992, the M.Sc. degree from Paris XI University, Orsay, France, in 1992 and the Ph.D. degree from the ENST in 1995 (in the field of Signal Processing and communications). From 1995 to 1998, he took a position as a research staff at the Electrical Engineering Department of the University of Melbourne where he worked on several research project related to “Blind System Identification for Wireless Communications”, “Blind Source Separation”, and “Array Processing for Communications”, respectively. From 1998 to 2012 he has been Assistant then Associate Professor at the Signal and Image Processing Department of Telecom-ParisTech. His research interests are in signal processing for communications, adaptive filtering and tracking, array processing and statistical performance analysis. In September 2012 he joined the University of Orleans (PRISME Lab.) as a full Professor. He is the author of over 450 scientific publications including book chapters, international journal and conference papers and patents. Dr. Abed-Meraim is an IEEE Fellow, an IEEE SAM-TC member and a past associate editor for the IEEE Transactions on Signal Processing.