AN ADAPTIVE PROJECTED SUBGRADIENT BASED ALGORITHM FOR ROBUST SUBSPACE TRACKING

Symeon Chouvardas[1] *Yiannis Kopsinis*[2] *Sergios Theodoridis*[2]

[1]Computer Technology Institute and Press ,
N. Kazantzaki Str,
University Campus 26504 Rio Greece.
Email: schouv@di.uoa.gr

[2]University of Athens,
Dept. of Informatics and Telecommunications,
Athens 15784, Greece.
Emails: kopsinis@ieee.org, stheodor@di.uoa.gr

ABSTRACT

In this paper, an Adaptive Projected Subgradient Method (APSM) based algorithm for robust subspace tracking is introduced. A properly chosen cost function is constructed at each time instance and the goal is to seek for points, which belong to the zero level set of this function; *i.e.,* the set of points which score a zero loss. In each iteration, an outlier detection mechanism is employed, in order to conclude whether the current data vector contains outlier noise or not. Furthermore, a sparsity–promoting greedy algorithm is employed for the outlier vector estimation allowing the purification of the corrupted data from the outlier noise prior further processing. A theoretical analysis is carried out and experiments within the context of robust subspace estimation exhibit the enhanced performance of the proposed scheme compared to a recently developed state of the art algorithm.

Index Terms— Robust Subspace Tracking, APSM, Greedy Algorithms.

1. INTRODUCTION

A plethora of machine learning and signal processing applications make decisions exploiting information drawn from large data sets. The involved data, which are usually high dimensional vectors, can be generated via the Internet, e–commerce sites, tablets/mobile devices, etc. As the amount of these data increases, the memory storage requirements as well as the computational complexity of the involved algorithms rise. Henceforth, it is a matter of paramount importance to develop *efficient* techniques for processing such large data sets, the so called *big data*, [1, 2].

These data, most often, do not lie anywhere in the high dimensional space but they rather "live" in subspaces of much smaller dimensions. This is a key attribute, which allows the efficient analysis and processing of such data, provided that their low–rank subspace has been accurately estimated. This task can be proved quite challenging, especially when the low–rank subspace is subjected to changes with time, [3]. In such cases, the employed subspace estimation algorithm needs to track the subspace changes online; *i.e,* to update its current estimates based on the data which become available sequentially, one per time instance, as time evolves. Subspace Tracking (ST), *e.g.,* [4, 5, 6, 7], plays a central role in many applications, such as, tracking of moving objects [8], foreground/background separation [9], beamforming [10], just to name a few. Even in cases where the sought subspace is not changing with time, the sequential/online processing offers significant benefits regarding memory requirements and computational complexity compared to the batch mode of operation. In the latter, a) all the available data have to be stored, increasing the memory requirements and the computational complexity and b) the unknown subspace have to be re–computed from scratch whenever a new datum becomes available.

In many ST applications (see for example [9]) the data set includes outliers, which are usually corrupted data that do not adhere to the adopted model. For example, when the data are received from Wireless Sensors, which is the case in localization and tracking applications, outliers may occur due to malfunctioning nodes. If the presence of outliers is not taken into consideration, then the performance of the ST algorithms can be degraded, *e.g.,* [9]. Hence, robustness against outliers is a matter of paramount importance in subspace tracking.

Related Work: Estimating and tracking the subspace, on which a sequence of vectors lie, has been extensively studied over the past decades. A well known algorithm, of relative low complexity, for tracking the signal subspace, is the so–called Projection Approximation Subspace Tracking (PAST) proposed in [4]. The Recursive Least Squares (RLS) technique is employed for the subspace estimation. More recently, the studies in [11, 7, 5] tackle the problem of ST in a scenario, where missing entries are met in the obtained vectors. It is worth pointing out, that the methodology presented in [11] is based on gradient descent iterations on the Grassmannian manifold. Finally, the efforts presented in [6, 9] attack the problem of ST in environments where the measurements are contaminated with outlier noise.

Contributions: In this paper, a novel algorithm for robust subspace tracking is proposed. The presented scheme belongs to the family of the Adaptive Projected Subgradient Method (APSM) algorithms, *e.g.,* [12, 13, 14]. More specifically, at each time instance, based on the most recently received data, a cost function is defined. This cost function scores a zero loss for a non–empty set of points/possible solutions. Our goal consists of finding a point, which belongs to the intersection of the sets corresponding to all received data. Furthermore, the proposed algorithm identifies the time instances at which the data contain outlier noise. In this case, a sparsity–aware greedy technique, namely the Compressed Sampling Orthogonal Matching Pursuit (CoSAMP), [15], estimates the sparse outlier vector, and removes it from the data vector. A theoretical analysis of the proposed scheme is presented. Finally, numerical examples verify the enhanced performance of the proposed algorithm compared to a state of the art algorithm for robust subspace tracking.

Notation: The set of real numbers and the set of non–negative integers are denoted by $\mathbb{R}$ and $\mathbb{N}$ respectively. Matrices are denoted

The project HANDiCAMS acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 323944 by the FET HANDICAM FP7. This work is partly supported by Marie Curie IEF, "SOL", 302898.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

by uppercase boldface letters and vectors by boldface letters. $(\cdot)^T$ stands for vector/matrix transposition. Moreover, $\|\cdot\|$ stands for the Euclidean norm and $\|\cdot\|_F$ for the Frobenius norm. Finally, the $m\times r$ zero matrix and the $m\times 1$ zero vector are denoted by $\boldsymbol{O}_{m\times r}$ and $\mathbf{0}_m$ respectively.

2. PROBLEM STATEMENT

We consider that at each time step, n, we observe an $m\times 1$ vector, $\boldsymbol{x}_n$, generated via the model:

$$\boldsymbol{x}_n = \boldsymbol{U}_*^{(n)}\boldsymbol{w}_n + \boldsymbol{v}_n + \boldsymbol{o}_n,\ \forall n\in\mathbb{N} \tag{1}$$

where $\boldsymbol{U}_*^{(n)}$ is an $m\times r$ orthonormal matrix, $\boldsymbol{w}_n\in\mathbb{R}^r$, $\boldsymbol{v}_n\in\mathbb{R}^m$ is the noise process and $\boldsymbol{o}_n$ is either $\mathbf{0}_m$ or corresponds to an outlier, *i.e.,* $\boldsymbol{o}_n = \boldsymbol{s}_n\in\mathbb{R}^m$. Following a similar rationale as in [9, 6], we assume that the outlier vectors are sparse, that is, $\|\boldsymbol{s}_n\|_0 \ll m$, where with $\|\cdot\|_0$ we denote the ℓ_0 pseudo-norm. The columns of the matrix $\boldsymbol{U}_*^{(n)}$ span an r-dimensional subspace of $\mathbb{R}^m$, say $\mathcal{S}_n$, and our *goal* is to estimate $\mathcal{S}_n$. In other words, the algorithm is allowed to provide as solution any matrix $\hat{\boldsymbol{U}}_*^{(n)}\in\mathbb{R}^{m\times r}$, whose column space span the unknown subspace.

2.1. Projection Approximation Subspace Tracking

The Projection Approximation Subspace Tracking (PAST) algorithm, presented in [4], is fundamental for the ST problem, being the starting point of a number of methods, the one presented here included. PAST assumes that $\boldsymbol{o}_n = \mathbf{0}_m$, $\forall n\in\mathbb{N}$, *i.e.,* no outliers are present and considers minimizing the following scalar cost function, with respect to an $m\times r$ orthonormal matrix $\boldsymbol{U}$:

$$J(\boldsymbol{U}) = \mathbb{E}\|\boldsymbol{x}_n - \boldsymbol{U}\boldsymbol{U}^T\boldsymbol{x}_n\|^2, \tag{2}$$

where $\mathbb{E}$ stands for the expectation operator. The physical reasoning of (2) can be summarized as follows. For an orthonormal matrix, it holds that $\boldsymbol{U}\boldsymbol{U}^T := P_{\widehat{\mathcal{S}}}$, where $P_{\widehat{\mathcal{S}}}$ is the projection operator onto $\widehat{\mathcal{S}}$, that is the subspace spanned by the columns of $\boldsymbol{U}$. The cost in (2) produces a matrix, which minimizes the misfit between the obtained data and their projection onto the subspace generated by this matrix. It is obvious that if the vectors $\boldsymbol{x}_n$ are noiseless and if one computes a matrix, $\widehat{\boldsymbol{U}}$, for which the loss function becomes zero, then the columns of $\widehat{\boldsymbol{U}}$ span the unknown subspace. Indeed, the minimizers of (2) are strongly connected to the subspace to be identified; it has been proved in [4] that if the subspace is fixed, i.e., $\mathcal{S}_n = \mathcal{S}$, $\forall n$, then the global minima of (2) is the only stable stationary point, which is given by $\hat{\boldsymbol{U}} = \boldsymbol{U}_r\boldsymbol{Q}$ where $\boldsymbol{U}_r$ consists of the r dominant eigenvectors of $\boldsymbol{C}_{xx} = \mathbb{E}[\boldsymbol{x}_n\boldsymbol{x}_n^T]$, and $\boldsymbol{Q}\in\mathbb{R}^{r\times r}$ is a unitary matrix. It can be readily obtained that the columns of $\hat{\boldsymbol{U}}$ span the subspace $\mathcal{S}$. The cost $J(\boldsymbol{U})$ is a fourth order function of the elements of $\boldsymbol{U}$ and it is, therefore, a non–convex function. This does not contradict with the fact that (2) assumes a global minimum. It turns out, that all the rest stationary points are not minima, but just saddle points.

Motivated by the exponentially weighted approach, central in the Recursive Least Squares (RLS) algorithm, online processing and tracking abilities are assigned to the PAST algorithm by a slight modification of the loss function

$$J^{(n)}(\boldsymbol{U}) = \sum_{i=1}^{n}\beta^{n-i}\|\boldsymbol{x}_i - \boldsymbol{U}\boldsymbol{U}^T\boldsymbol{x}_i\|^2, \tag{3}$$

$0<\beta\le 1$ is the so called forgetting factor, used in non–stationary environments, where the subspace undergoes changes. Moreover, a further simplification is adopted aiming to convexify (3). In particular:

$$J^{(n)}(\boldsymbol{U}) = \sum_{i=1}^{n}\beta^{n-i}\|\boldsymbol{x}_i - \boldsymbol{U}\boldsymbol{y}_i\|^2, \tag{4}$$

where $\boldsymbol{y}_i = \boldsymbol{U}_{i-1}^T\boldsymbol{x}_i$, *i.e.,* one of the unknown parameter matrices, $\boldsymbol{U}$, is replaced by the respective tentative estimates, $\boldsymbol{U}_{i-1}^T$.

The cost function given in (4) is the exponential Least Squares criterion, which has been extensively studied in adaptive filtering, *e.g.,* [10]. The matrix $\boldsymbol{U}_{LS}^{(n)}$, that minimizes the cost (4) at time instance n, is given by [4]:

$$\boldsymbol{U}_{LS}^{(n)} = \boldsymbol{C}_{xy}(n)\boldsymbol{C}_{yy}^{-1}(n), \tag{5}$$

where

$$\boldsymbol{C}_{xy}(n) = \beta\boldsymbol{C}_{xy}(n-1) + \boldsymbol{x}_n\boldsymbol{y}_n^T, \tag{6}$$

and

$$\boldsymbol{C}_{yy}(n) = \beta\boldsymbol{C}_{yy}(n-1) + \boldsymbol{y}_n\boldsymbol{y}_n^T. \tag{7}$$

Having access to the quantities $(\boldsymbol{C}_{xy}(n), \boldsymbol{C}_{yy}(n))_{n\in\mathbb{N}}$, the classical PAST algorithm employs the Recursive Least Squares (RLS) Method for the estimation of $\boldsymbol{U}_{LS}^{(n)}$, by computing efficiently the matrix $\boldsymbol{C}_{yy}^{-1}(n)$.

2.2. Subspace Tracking via the APSM Algorithm

In this study, a different route is followed. As we have already described before, the PAST algorithm computes, at each step, the matrix $\boldsymbol{U}_{LS}^{(n)}$ by solving (5). The LS solution, which is sought via the time averaged covariance matrices $(\boldsymbol{C}_{xy}(n), \boldsymbol{C}_{yy}(n))$, is likely to deviate from the true solution, *i.e.,* from the true subspace, due to a number of reasons such as: additive noise, measurement and model inaccuracies, as well as calibration errors. In order to accommodate such deviations, following set theoretic arguments, we seek the unknown subspace within an "extended" set of possible solutions which guarantee to include the true one. To be more specific, given a certain tolerance $\epsilon>0$, we define the following cost function

$$\Theta_n:\mathbb{R}^{m\times r}\to[0,+\infty):$$
$$\boldsymbol{U}\mapsto\max\left\{0, \frac{1}{2}\|\boldsymbol{C}_{xy}(n) - \boldsymbol{U}\boldsymbol{C}_{yy}(n)\|_F^2 - \epsilon\right\}, \tag{8}$$

and our goal is to find points, which lie in the level set of this cost function, defined as: $\mathrm{lev}_{\le 0}\Theta_n := \{\boldsymbol{U}\in\mathbb{R}^{m\times r} : \Theta_n(\boldsymbol{U})\le 0\}$. Notice that $\boldsymbol{U}_{LS}^{(n)}\in\mathrm{lev}_{\le 0}\Theta_n$, hence the level–set at each time instance is an "enlarged" set of candidate solutions, since it contains every matrix, which scores a zero loss, instead of containing a single point, *i.e.,* $\boldsymbol{U}_{LS}^{(n)}$.

The algorithm, to be presented here, is based on the set theoretic estimation approach; at each step, a set is constructed, which in our case is the level–set $\mathrm{lev}_{\le 0}\Theta_n$ and the goal is to compute a point in the intersection of these level sets. This can be effectively achieved via the Adaptive Projected Subgradient Method formula, *e.g.,* [13, 16, 12, 14], given next:

$$\boldsymbol{U}_{n+1} = \begin{cases}\boldsymbol{U}_n - \lambda_n\frac{\Theta_n(\boldsymbol{U}_n)}{\|\Theta_n'(\boldsymbol{U}_n)\|_F^2}\Theta_n'(\boldsymbol{U}_n), & \Theta_n'(\boldsymbol{U}_n)\neq\boldsymbol{O}_{m\times r}\\ \boldsymbol{U}_n, & \Theta_n'(\boldsymbol{U}_n)=\boldsymbol{O}_{m\times r},\end{cases} \tag{9}$$

where Θ_n' stands for any subgradient, which belongs to the subdifferential $\partial\Theta_n$, defined as [17]:

$$\partial\Theta_n(\boldsymbol{U}) = \begin{cases}\boldsymbol{O}_{m\times r}, & \frac{1}{2}\|\boldsymbol{C}_{xy}(n) - \boldsymbol{U}\boldsymbol{C}_{yy}(n)\|_F^2<\epsilon,\\ \gamma(\boldsymbol{U}\boldsymbol{C}_{yy}(n) - \boldsymbol{C}_{xy})\boldsymbol{C}_{yy}^T(n),\ \gamma\in[0,1] & \frac{1}{2}\|\boldsymbol{C}_{xy}(n) - \boldsymbol{U}\boldsymbol{C}_{yy}(n)\|_F^2=\epsilon,\\ (\boldsymbol{U}\boldsymbol{C}_{yy}(n) - \boldsymbol{C}_{xy})\boldsymbol{C}_{yy}^T(n), & \frac{1}{2}\|\boldsymbol{C}_{xy}(n) - \boldsymbol{U}\boldsymbol{C}_{yy}(n)\|_F^2>\epsilon,\end{cases} \tag{10}$$

and $\lambda_n \in (0, 2)$.

The algorithm above is hereafter referred to as Subspace Tracking based on Adaptive Projection Subgradient Method (STAPSM).

3. THE ROBUST SUBSPACE TRACKING APSM

In this section, a robust ST algorithm will be developed, using the "tools" described in subsection 2.2. The algorithm tackles the presence of outliers in two phases:

a) *An outlier is detected:* If the error residual, $\|\boldsymbol{x}_n - P_{U_{n-1}}\boldsymbol{x}_n\|$, where $P_{U_{n-1}}$ is the projection matrix, which projects $\boldsymbol{x}_n$ onto the column space of $\boldsymbol{U}_{n-1}$, take a large value, then this is a strong indicator that the measurement vector, $\boldsymbol{x}_n$, is corrupted by outlier noise. Indeed, as the time passes and we obtain an estimate of the subspace, which is close to the true one, then $\|\boldsymbol{x}_n - P_{U_{n-1}}\boldsymbol{x}_n\| \approx \|\boldsymbol{x}_n - P_{\mathcal{S}}(\boldsymbol{x}_n)\| \approx \|\boldsymbol{v}_n + \boldsymbol{o}_n\|$. Hence, if the value of the residual is small (smaller than a specified threshold) then it is likely that $\boldsymbol{o}_n = \boldsymbol{0}_m$, whereas if it takes a relatively large value (larger than the same threshold) then $\boldsymbol{o}_n = \boldsymbol{s}_n$. The threshold, through which we make a decision on whether $\boldsymbol{x}_n$ contains outlier noise or not, is apparently related to the noise variance. Here it is adaptively estimated as follows: It is set equal to $r_n = \delta \bar{r}_{n-1}$, where $\bar{r}_{n-1}$ is the average of the q most recent error residuals, q is a user–defined parameter and δ is a user defined multiplication factor. Apparently, the larger the δ is the less sensitive the algorithm becomes in the detection of outlier vectors. The threshold is computed via an average over the q most recent error residuals, instead of the whole history, so as to "forget" values of the error from the past, which may correspond to previous time instances of the unknown subspace, in a non–stationary environment.

b) *The measurement vector, $\boldsymbol{x}_n$, is cleansed from the outlier noise, if necessary:* If a measurement vector $\boldsymbol{x}_n$ is not corrupted by outlier noise, *i.e.* the residual is lower than the computed threshold of the previous phase, then the statistical quantities $\boldsymbol{C}_{xy}(n)$, $\boldsymbol{C}_{yy}(n)$ are updated as suggested in (6), (7) and the new estimate is produced via (9). On the contrary, if at time instance n, an outlier value is detected, the algorithm estimates its value via the method that will be described in the sequel, and subtracts it from the measurement vector.

Note, that in the general case where $\boldsymbol{U}_{n-1}$ is not orthonormal, the projection matrix $P_{U_{n-1}}$ is given by $\boldsymbol{U}_{n-1}(\boldsymbol{U}_{n-1}^T\boldsymbol{U}_{n-1})^{-1}\boldsymbol{U}_{n-1}^T$. Therefore,

$$\begin{aligned} \boldsymbol{r}_n &:= \boldsymbol{x}_n - P_{U_{n-1}}\boldsymbol{x}_n = \boldsymbol{x}_n - \boldsymbol{U}_{n-1}(\boldsymbol{U}_{n-1}^T\boldsymbol{U}_{n-1})^{-1}\boldsymbol{U}_{n-1}^T\boldsymbol{x}_n \\ &= \boldsymbol{U}_*\boldsymbol{w}_n + \boldsymbol{v}_n + \boldsymbol{s}_n \\ &\quad - \boldsymbol{U}_{n-1}(\boldsymbol{U}_{n-1}^T\boldsymbol{U}_{n-1})^{-1}\boldsymbol{U}_{n-1}^T(\boldsymbol{U}_*\boldsymbol{w}_n + \boldsymbol{v}_n + \boldsymbol{s}_n) \\ &= (\boldsymbol{I}_m - \boldsymbol{U}_{n-1}(\boldsymbol{U}_{n-1}^T\boldsymbol{U}_{n-1})^{-1}\boldsymbol{U}_{n-1}^T)\boldsymbol{s}_n + \boldsymbol{\eta}_n, \end{aligned} \tag{11}$$

where $\boldsymbol{\eta}_n := \boldsymbol{U}_*\boldsymbol{w}_n + \boldsymbol{v}_n - \boldsymbol{U}_{n-1}(\boldsymbol{U}_{n-1}^T\boldsymbol{U}_{n-1})^{-1}\boldsymbol{U}_{n-1}^T(\boldsymbol{U}_*\boldsymbol{w}_n + \boldsymbol{v}_n)$ and $\boldsymbol{I}_m$ is the $m \times m$ identity matrix. It is not difficult to see that, if the estimated subspace is close to the true one, then $\boldsymbol{U}_*\boldsymbol{w}_n \approx \boldsymbol{U}_{n-1}(\boldsymbol{U}_{n-1}^T\boldsymbol{U}_{n-1})^{-1}\boldsymbol{U}_{n-1}^T\boldsymbol{U}_*\boldsymbol{w}_n \Leftrightarrow \boldsymbol{\eta}_n \approx \boldsymbol{v}_n$. Recalling that $\boldsymbol{s}_n$ is a sparse vector, one can resort to the Compressed Sensing armoury, *e.g.,* [18, 19], in order to estimate it via the following linear system:

$$\boldsymbol{r}_n = \boldsymbol{A}_n\boldsymbol{s}_n + \boldsymbol{\eta}_n, \tag{12}$$

where $\boldsymbol{A}_n := \boldsymbol{I}_m - \boldsymbol{U}_{n-1}(\boldsymbol{U}_{n-1}^T\boldsymbol{U}_{n-1})^{-1}\boldsymbol{U}_{n-1}^T$. In the current study, for the estimation of the sparse vector $\boldsymbol{s}_n$, we follow the greedy philosophy. In a nutshell, greedy techniques identify the support set, *i.e.,* the positions of the non–zero coefficients of the unknown vector, and then perform a Least Squares estimate restricted on this subset. Here, for the estimation of the sparse outlier vector, the CoSAMP algorithm is employed, [15].

Table 1.

Robust Subspace Tracking Adaptive Projected Subgradient Method

Initialization: window length q, $\delta > 0$.	An $m \times r$ random matrix $\boldsymbol{U}_0$, $\epsilon > 0$,
FOR $i = 1 : n$ **DO**	
1:	$\boldsymbol{r}_n = \boldsymbol{x}_n - \boldsymbol{U}_{n-1}(\boldsymbol{U}_{n-1}^T\boldsymbol{U}_{n-1})^{-1}\boldsymbol{U}_{n-1}^T\boldsymbol{x}_n$
IF $\|\boldsymbol{r}_n\|^2 =: r_n \leq \delta\bar{r}_{n-1}$	
2:	$\hat{\boldsymbol{s}}_n = \boldsymbol{0}_m$
ELSE	
3:	$\hat{\boldsymbol{s}}_n = \mathrm{CoSAMP}(\boldsymbol{A}_n, \boldsymbol{r}_n)$
ENDIF	
4:	$\boldsymbol{y}_n = (\boldsymbol{U}_{n-1}^T\boldsymbol{U}_{n-1})^{-1}\boldsymbol{U}_{n-1}^T(\boldsymbol{x}_n - \hat{\boldsymbol{s}}_n)$
5:	$\boldsymbol{C}_{xy}(n) = \beta\boldsymbol{C}_{xy}(n-1) + (\boldsymbol{x}_n - \hat{\boldsymbol{s}}_n)\boldsymbol{y}_n^T$
6:	$\boldsymbol{C}_{yy}(n) = \beta\boldsymbol{C}_{yy}(n-1) + \boldsymbol{y}_n\boldsymbol{y}_n^T$
7:	$\boldsymbol{U}_n = \begin{cases} \boldsymbol{U}_{n-1} - \lambda_n \frac{\Theta_n(\boldsymbol{U}_{n-1})}{\|\Theta_n'(\boldsymbol{U}_{n-1})\|_F^2}\Theta_n'(\boldsymbol{U}_{n-1}), & \Theta_n'(\boldsymbol{U}_{n-1}) \neq \boldsymbol{O}_{m\times r} \\ \boldsymbol{U}_{n-1}, & \Theta_n'(\boldsymbol{U}_{n-1}) = \boldsymbol{O}_{m\times r} \end{cases}$
8:	$q' = \max\{0, n-q\}$, $\bar{r}_n = \frac{1}{n-q'}\sum_{j=q'}^{n} r_j$
END	

The steps of the algorithm are summarized in Table 1. In Step 1, the residual vector is computed and depending on its norm the sparse outlier vector is set equal to $\boldsymbol{0}_m$ (Step 2) or it is computed via the CoSAMP algorithm (Step 3). In Step 4, the vector $\boldsymbol{y}_n$ is calculated via the optimization: $\boldsymbol{y}_n = \arg\min_{\boldsymbol{y}\in\mathbb{R}^r} \|\boldsymbol{x}_n - \boldsymbol{U}_{n-1}\boldsymbol{y}\|^2$ and Steps 5–6 update the quantities $\boldsymbol{C}_{xy}$, $\boldsymbol{C}_{yy}$ respectively. In Step 7 the new estimate is updated via the APSM formula and, finally, Step 8 computes the average residual error within the time window of length q.

Remark 1 *The complexity of both STAPSM and Robust STAPSM is of order $O(nr^2)$ springing from the computation of $\boldsymbol{y}_n$. Moreover, the complexity of the CoSAMP is $O(nrT)$, [15], where T is the number of running iterations of the algorithm. However, it should be pointed out that the CoSAMP is not employed at each time step, but whenever the algorithm identifies an outlier.*

3.1. Theoretical Analysis

In this section, we will present the Theoretical Analysis of the proposed scheme, in the conventional ST scenario, *i.e.,* in the case where the data adhere to the model (2) and the outliers equal to $\boldsymbol{0}_m$. The proofs of the theorems and the extension of the theoretical analysis for the robust case will be presented elsewhere due to space limitations.

Assumptions 1

1. *There exists a sufficiently large time–step, n_0, a positive number ρ_1 and any $m \times r$ matrix, say $\bar{\boldsymbol{U}}$ for which: $\|\bar{\boldsymbol{U}} - \boldsymbol{U}_{LS}^{(n)}\|_F \leq \rho_1$, $\forall n \geq n_0$.*
2. *We assume that: $\|\boldsymbol{C}_{yy}(n)\|_F \leq \rho_2$.*

Theorem 1 *Consider that assumptions 1.1 and 1.2 hold true. The proposed algorithm enjoys:*

- ***Monotonicity:*** *The distance of the estimates from $\bar{\boldsymbol{U}}$ is a non–increasing sequence, i.e.,*

$$\|\boldsymbol{U}_{n+1} - \bar{\boldsymbol{U}}\|_F \leq \|\boldsymbol{U}_n - \bar{\boldsymbol{U}}\|_F, \forall n \geq n_0, \tag{13}$$

  provided that $\epsilon \geq \frac{1}{2}\rho_1^2\rho_2^2$ and $\lambda_n \in (0, 2)$.

- ***Asymptotic Optimality:*** *The time varying cost functions are asymptotically minimized. Put in mathematical terms:*

$$\lim_{n\to\infty} \Theta_n(\boldsymbol{U}_n) = 0, \tag{14}$$

  if $\epsilon \geq \frac{1}{2}\rho_1^2\rho_2^2$ and $\lambda_n \in [\epsilon_1, 2 - \epsilon_1] \subset (0, 2)$.

- ***Strong Convergence:*** *The sequence of estimates satisfies:*

$$\lim_{n\to\infty} \boldsymbol{U}_n = \boldsymbol{U}_O, \tag{15}$$

if $\epsilon > 2\rho_1^2\rho_2^2$ *and* $\lambda_n \in [\epsilon_1, 2-\epsilon_1] \subset (0,2)$. *Moreover, the matrix* $\boldsymbol{U}_O$ *satisfies:*

$$\boldsymbol{U}_O \in \overline{\liminf_{n\to\infty} \mathrm{lev}_{\leq 0}\Theta_n}, \tag{16}$$

where $\liminf_{n\to\infty} \mathrm{lev}_{\leq 0}\Theta_n = \bigcup_{n=0}^{\infty}\bigcap_{k\geq n} \mathrm{lev}_{\leq 0}\Theta_k$ *and the overline symbol denotes the closure of a set. In words the sequence of estimates converges to a point that lies arbitrarily close to the intersection of the level sets* $\forall n \geq n_0$.

Remark 2 *It can be shown that assumptions 1. 1, 1. 2 can be guaranteed if the produced estimates are bounded. This assumption is typical in ST problems, see for example [5, Assumption A.3] where boundness of the estimates is assumed. Assumption 1.1 yields that after a number of iterations, the "optimum" matrix* $\boldsymbol{U}_{LS}^{(n)}$ *will not undergo large deviations, or in mathematical terms the matrices* $\boldsymbol{U}_{LS}^{(n)}$, $\forall n \geq n_0$ *belong to a ball with center* $\bar{\boldsymbol{U}}$ *and radius* ρ_1.

4. NUMERICAL EXAMPLES

In this section, we examine the performance of the Robust STAPSM in a stationary and in a non–stationary scenario. The proposed scheme is compared to the Grassmannian Robust Adaptive Subspace Tracking Algorithm (GRASTA) algorithm, [6], which is suitable for robust subspace tracking. This algorithm has the same order of complexity to the proposed one. The adopted performance metric is the angle between the true subspace and the estimated one in logarithmic scale.

The Stationary Scenario: In the first experiment we adopt the model described in (1), with $m = 100$ and $r = 10$. We assume that $\boldsymbol{U}_*^{(n)} = \boldsymbol{U}_*$, $\forall n \in \mathbb{N}$, where the columns of the $m \times r$ matrix $\boldsymbol{U}_*$ are realizations of an i.i.d. $\mathcal{N}(\boldsymbol{0}_m, \boldsymbol{I}_m)$, which are orthonormalized. The coefficients of the vector $\boldsymbol{w}_n$ and the noise $\boldsymbol{v}_n$ are drawn from the Gaussian distribution with zero mean and variance equal to 1 and 10^{-3} respectively. Finally, we assume that 10% of the data contain outlier noise. For the sparse outlier vector we have that $\|\boldsymbol{s}_n\|_0 = 5$ and its coefficients follow the Gaussian distribution with zero mean and variance equal to 4. The positions of the non–zero coefficients of $\boldsymbol{s}_n$ and the time steps, on which the outliers occur, are selected randomly

For the proposed algorithm, we set $\epsilon = 2 \times 10^{-3}$, $\beta = 0.999$, $q = 20, \delta = 3$ and $\lambda_n = 1$. Regarding the parameters ϵ, q, δ we observed that different choices of them do not affect significantly the performance of the proposed scheme. These specific values were chosen, because they led to fast convergence speed and a low error floor after the convergence. Usually, the forgetting factor β in stationary environments is set 1, since there is no need to forget past values, which is the case when $\beta < 1$. Nevertheless, here we chose a slightly smaller β due to the presence of outliers, since possible errors between the true sparse outlier vector and the estimated one, which are carried from past values, may lead to performance degradation. Regarding the step–size λ_n, we obtained, through extensive experimentation, that the larger the λ_n the faster the convergence, at the expense of a higher error floor. Choosing $\lambda_n = 1$ leads to a good trade–off between convergence speed and steady–state error floor. Finally, it is assumed that we have an estimate of the number of non–zero coefficients of the vector $\boldsymbol{s}_n$, employed in the CoSAMP algorithm.

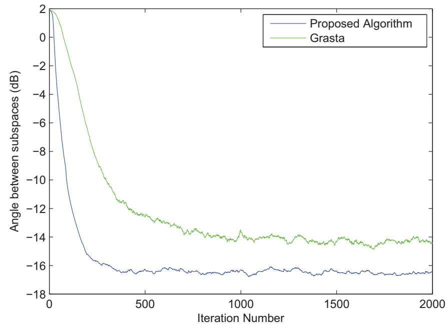

Fig. 1. Angle between the subspaces: the stationary case

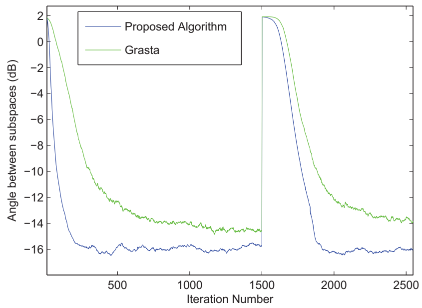

Fig. 2. Angle between the subspaces: the non–stationary case

The code of the GRASTA algorithm has been downloaded from http://nuit-blanche.blogspot.gr/2012/01/grasta-grassmannian-robust-adaptive.html. The number of subgradient iterations in the GRASTA, employed in order to estimate the sparse unknown vector is chosen equal to 60, and the rest of the parameters were chosen as suggested in [6].

Figure 1 illustrates that the proposed algorithm outperforms the GRASTA algorithm, since it converges faster to a lower error floor than the latter algorithm.

The Non–Stationary Scenario: In this experiment, we examine the tracking ability of the proposed scheme. More specifically, the parameters are the same as in the previous experiment, but here at time step 1500 an abrupt change takes place in the matrix $\boldsymbol{U}_*^{(n)}$ and consequently to the involved subspace.

As we have already mentioned, the parameters are the same as with the exception of the forgetting factor β, which is set 0.9. Fig. 2 illustrates that the proposed algorithm enjoys a tracking ability, since it converges fast to a low error floor, after the sudden change.

5. CONCLUSIONS AND FUTURE WORK

A novel algorithm, for robust subspace tracking is introduced. The algorithm is based on the Adaptive Projected Subgradient Method. A properly constructed time–varying cost function is employed at each time step; the goal is to find for a point that lies within the intersection of the level–sets of the previously mentioned cost functions. A theoretical analysis takes place in the conventional subspace tracking scenario, *i.e.,* when the outliers are absent. Finally, the enhanced performance of the proposed scheme against a state of the art robust subspace tracking algorithm is validated through numerical experiments. Future research focuses on generalizing the proposed scheme in scenarios where the data vectors are partially observed and deriving the theoretical analysis in the case where the outlier noise is present.

6. REFERENCES

[1] Trevor Hastie, Robert Tibshirani, and Jerome Friedman, *The Elements of Statistical Learning*, New York Springer, 2009.

[2] Georgios Giannakis, Konstantinos Slavakis, and Gonzalo Mateos, "Dimensionality reduction for big data analytics," *IEEE Signal Processing Magazine, Special issue on Signal processing for Big Data, (Whitepaper submitted)*, 2013.

[3] Tülay Adali and Simon Haykin, *Adaptive signal processing: next generation solutions*, vol. 55, John Wiley & Sons, 2010.

[4] Bin Yang, "Projection approximation subspace tracking," *Signal Processing, IEEE Transactions on*, vol. 43, no. 1, pp. 95–107, 1995.

[5] Morteza Mardani, Gonzalo Mateos, and Georgios B Giannakis, "Rank minimization for subspace tracking from incomplete data," in *In Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 5681–5685.

[6] Jun He, Laura Balzano, and John Lui, "Online robust subspace tracking from partial information," *arXiv preprint arXiv:1109.3827*, 2011.

[7] Yuejie Chi, Yonina C Eldar, and Robert Calderbank, "PETRELS: Subspace estimation and tracking from partial observations," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 3301–3304.

[8] Shahram Shahbazpanahi, Shahrokh Valaee, and Mohammad Hasan Bastani, "Distributed source localization using esprit algorithm," *IEEE Transactions on Signal Processing*, vol. 49, no. 10, pp. 2169–2178, 2001.

[9] Gonzalo Mateos and Georgios B Giannakis, "Robust PCA as bilinear decomposition with outlier-sparsity regularization," *IEEE Transactions on Signal Processing*, vol. 60, pp. 5176–5190, 2012.

[10] A.H. Sayed, *Fundamentals of adaptive filtering*, John Wiley & Sons, New Jersey, 2003.

[11] Jun He, Laura Balzano, and John Lui, "Online robust subspace tracking from partial information," *arXiv preprint arXiv:1109.3827*, 2011.

[12] Konstantinos Slavakis and Isao Yamada, "The adaptive projected subgradient method constrained by families of quasi-nonexpansive mappings and its application to online learning," *SIAM Journal on Optimization*, vol. 23, no. 1, pp. 126–152, 2013.

[13] I. Yamada and N. Ogura, "Adaptive projected subgradient method for asymptotic minimization of sequence of nonnegative convex functions," *Numerical functional analysis and optimization*, vol. 25, no. 7&8, pp. 593–617, 2004.

[14] Sergios Theodoridis, Konstantinos Slavakis, and Isao Yamada, "Adaptive learning in a world of projections," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 97–123, 2011.

[15] Deanna Needell and Joel A Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[16] K. Slavakis, I. Yamada, and N. Ogura, "The adaptive projected subgradient method over the fixed point set of strongly attracting nonexpansive mappings," *Numerical Functional Analysis and Optimization, 27*, vol. 7, no. 8, pp. 905–930, 2006.

[17] Konstantinos Slavakis, Pantelis Bouboulis, and Sergios Theodoridis, "Adaptive multiregression in reproducing kernel hilbert spaces: The multiaccess mimo channel case," *IEEE Transactions on Neural Networks Neural Networks and Learning Systems*, vol. 23, no. 2, pp. 260–276, 2012.

[18] David L Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[19] Sergios Theodoridis, Yannis Kopsinis, and Konstantinos Slavakis, "Sparsity-aware learning and compressed sensing: An overview," *arXiv preprint arXiv:1211.5231*, 2012.