Learning Hidden Markov Sparse Models

Lin Li* and Anna Scaglione*
*Department of Electrical and Computer Engineering
University of California, Davis, CA 95616
Email: {llli, ascaglione}@ucdavis.edu

***Abstract*—This paper considers the problem of separating streams of unknown non-stationary signals from underdetermined mixtures of sources. The source signals are modeled as a *hidden Markov model* (HMM) where each state in the Markov chain is determined by a set of on (i.e., active) or off (i.e., inactive) states of the sources, with some unknown probability density functions (pdfs) in the on-state. Under the assumption that the number of active sources is small compared to the total number of sources (thus the sources are sparse), the goal is to recursively estimate the HMM state and the over-complete mixing matrix (subsequently the source signals) for signal recovery. The proposed approach combines the techniques of HMM-based filtering and manifold-based dictionary learning for estimating both the state and the mixing matrix. Specifically, we model the on/off state of the source signals as a hidden Markov model. In particular, we consider only a sparse set of simultaneously active sources. Thus, this setting generalizes the typical scenario considered in dictionary learning in which there is a sparse number of temporally independent active signals. To extract the activity profile of the sources from the observations, a technique known as change-of-measure is used to decouple the observations from the sources by introducing a new probability measure over the set of observations. Under this new measure, the un-normalized conditional densities of the state and the transition matrix of the Markov chain can be computed recursively. Due to the scaling ambiguity of the mixing matrix, we introduce an equivalence relation, which partitions the set of mixing matrices into a set of equivalence classes. Rather than estimating the mixing matrix by imposing the unit-norm constraint, the proposed algorithm searches directly for an equivalence class that contains the true mixing matrix. In our simulations, the proposed recursive algorithm with manifold-based dictionary learning, compared to algorithms with unit-norm constraint, estimates the mixing matrix more efficiently while maintaining high accuracy.**

I. Introduction

Blind source separation (BSS) studies the problem of recovering the unobserved source signals from the observed linear mixtures, without a priori knowledge of the mixing matrix. Due to its many potential applications in speech processing (for example, cocktail party problem), digital communication, neural networks, biomedical signal processing, telecommunication and array processing, BSS has gained considerable attention especially in the last two decades. A comprehensive review of more recent advances in several research fields can be found in [1], [2].

Today, there exists a number of BSS algorithms. In general, these algorithms can be divided into two categories: those that solve the *complete BSS* problem and those that solve the *underdetermined BSS* (U-BSS) problem. The complete BSS problem refers to the case where the number of sources equals the number of observations. The most relevant approach to solve this type of problem is based on independent component analysis (ICA) [3]. It involves minimizing or maximizing certain objective functions associated to one of the following three statistical properties of the source signals: non-Gaussianity [4]–[6], non-stationary [7]–[11], and time correlation [12]–[14]. In contrast, the U-BSS literature considers the case where the number of sources is greater than the number of observations and focuses on exploiting the sparsity of the source signals. Two main approaches have been widely studied under the umbrella of U-BSS: the so-called time-frequency binary mask [15]–[17] and the dictionary learning (or sparse coding approach) [18]–[29]. The former approach takes advantage of the property that for sufficiently sparse sources, at most one source is dominant in each spectrogram (i.e., time-frequency) segment. Furthermore, segments from the same source are grouped together using various clustering techniques. In contrast, the dictionary learning literature often assumes that the sources are spatially and temporally uncorrelated. The over-complete mixing matrix (also known as *dictionary*), can be learned from the observed mixtures under the sparsity constraint. In the following, we will briefly review the most common approaches to dictionary learning.

In dictionary learning [29], one is interested in expressing a set of observed mixtures $\{\boldsymbol{y}_i \in \mathbb{R}^N\}_{i=1}^m$ as linear combinations of a small number of bases $\boldsymbol{d}_k \in \mathbb{R}^n$ called *atoms*, which are taken from an over-complete matrix $\boldsymbol{D} = [\boldsymbol{d}_1, \cdots, \boldsymbol{d}_K] \in \mathbb{R}^{n \times K}$ with $K > n$. Therefore, any n-dimensional data $\boldsymbol{y}_m$ is represented by

$$\boldsymbol{y}_m = \boldsymbol{D}\,\boldsymbol{x}_m + \boldsymbol{w}_m = \sum_{k=1}^{K} x_m(k)\boldsymbol{d}_k + \boldsymbol{w}_m \qquad (1)$$

where the atoms $\boldsymbol{d}_k$ are typically assumed to be unit norm to resolve the scale ambiguity and $\boldsymbol{w}_m$ represents the additive noise. Since $\boldsymbol{D}$ is over-complete ($K > n$), the atoms d_k are linearly dependent. Various algorithms have been proposed over the years. Despite different derivations of the algorithms, one common feature is that the optimization problem always involves two terms in its cost. The first term measures how well the sparse representation of the sources $\boldsymbol{x}_m$ matches with the observed mixtures $\boldsymbol{y}_m$, i.e., $\sum_m \|\boldsymbol{y}_m - \boldsymbol{D}\,\boldsymbol{x}_m\|^2$. The second term enforces the sparse constraint on $\boldsymbol{x}_m$ by taking either the l_0-norm of $\boldsymbol{x}_m$ or the l_1-norm of $\boldsymbol{x}_m$ (the convex relaxation of the l_0-norm). Hence, the general approach to the conventional dictionary design involves a two-step process: the *inference step* and the *learning step* [29]. The inference step

finds the sparse representation $\boldsymbol{x}_m$ of $\boldsymbol{y}_m$ individually while keeping $\boldsymbol{D}$ constant. The learning step subsequently updates the dictionary $\boldsymbol{D}$ while assuming $\boldsymbol{x}_m$ is fixed for $\forall m$.

It is however, important to note that the above dictionary learning model neglects possible temporal and/or spatial correlations between signal samples across time and/or space. In the applications where statistical information of the source signals is available, Bayesian inference allows us to solve complex problems such as identification, tracking and estimation, in a relatively simple and potentially more accurate way.

The question to ask is how to formulate the source separation problem to capture the temporal and spatial correlation in streams of unknown sources. In this paper, we choose to exploit the temporal and spatial structures of the source sequences using the hidden Markov model (HMM) in which the activities of the sources are the hidden states. There are several reasons why one should be interested in applying HMM to dictionary learning. First of all, HMM can provide a concrete description of the sparsity profile because each hidden state corresponds to a set of active sources. A second reason why HMM is useful is that it allows us to model possible dependencies between sources. For example, if a source is active, then it is likely to stay active with a nonzero probability. Hence, the activity of a source at one time slot is dependent on that at the previous time slot. Moreover, if two sources are communicating with each other, then they are less likely to be active at the same time. Finally, HMM often work extremely well in practical applications and it also allows us to make predictions, i.e. tracking the "sparse" activity of a groups of sources, in an efficient manner.

A. Problem Statement

Consider the problem that K source signals are observed through N sensors, where the number of sensors is less than the number of sources, i.e., $N < K$. Let $x_m(k)$ be a *non-stationary* signal emitted by the k^{th} source and $y_m(n)$ be the observed signal at the n^{th} sensor. Denote by $\boldsymbol{x}_m = [x_m(1), \ldots, x_m(K)]^T$ the source signals at time m and $\boldsymbol{y}_m = [y_m(1), \ldots, y_m(N)]^T$ the observed signals at time m, where $m = 0, 1, 2, \ldots$ The observed output $\boldsymbol{y}_m$ at time m can be expressed as a linear mixture of source signals

$$\boldsymbol{y}_m = \boldsymbol{D}\,\boldsymbol{x}_m + \boldsymbol{w}_m \ . \tag{2}$$

The matrix $\boldsymbol{D} \in \mathbb{C}^{N\times K}$ is an unknown over-complete mixing matrix (also known as the dictionary). The additive noise $\boldsymbol{w}_m \in \mathbb{R}^N$ is assumed to be zero-mean Gaussian with a covariance $\boldsymbol{C} = \sigma_n^2\,\boldsymbol{I}$. The non-stationarity of the sources is tracked by a HMM in the proposed formulation (See Section II). Specifically, each state in the Markov chain is characterized by a set of on (i.e., active) and off (i.e., inactive) states of the sources. Each source is associated to an unknown conditionally independent probability density function (pdf) in the on-state (i.e., active-state) and admits no signal when it is inactive (or equivalently, $p(x \mid \text{inactive}) = \delta(x)$ where $\delta(\cdot)$ is the Dirac delta function). The sources are assumed to be S-sparse, that is, the number of active sources is less or equal to S and $S \ll K$. The objective is to recover the un-observed source signals $\boldsymbol{x}_m$ from the observations. If the mixing system $\boldsymbol{D}$ is known and it satisfies certain identifiability conditions [25], the source signals can be recovered exactly. Namely, the necessary and sufficient condition for having a unique solution $\boldsymbol{x}$ for the noiseless case is that no $2S$ columns of $\boldsymbol{D}$ are linearly dependent. In this case no S-sparse vector $\boldsymbol{x}$ can be confused for another S-sparse vector.

However, without a priori knowledge of $\boldsymbol{D}$, we need to first estimate $\boldsymbol{D}$ from the available observations $\boldsymbol{y}_m$ and, hence, there are other possible ambiguities. It is to be noted that the mixing matrix is not completely identifiable. In the absence of noise there exist two ambiguities: permutation ambiguity and scale ambiguity. The former implies that the re-ordering of the sources is indeterminate and it is not crucial to almost all the applications of interest. The latter suggests that for any nonsingular diagonal matrix Σ, $\boldsymbol{y} = \boldsymbol{D}\boldsymbol{x} = (\boldsymbol{D}\,\Sigma)(\Sigma^{-1}\boldsymbol{x})$. The matrix Σ has to be diagonal, otherwise it will not be possible to have for all m that the sparsity is preserved, i.e., $\|\Sigma^{-1}\boldsymbol{x}_m\|_0 = \|\,\boldsymbol{x}_m\,\|_0$. Usually, scale ambiguity is dealt with by fixing the norm of each column of $\boldsymbol{D}$ to be 1. However, in the proposed setting, imposing the unit-norm constraint is computationally costly because the resulting algorithm has to alternate between updating the mixing matrix $\boldsymbol{D}$ and variances of the source signals. For signals with time-varying variances, it is more difficult to track them and this might lead to poor convergence performance.

In the proposed approach, the key to deal with scale ambiguity is to introduce an *equivalence relation* $\sim$ on the space of the mixing matrices. Precisely, the two matrices $\boldsymbol{D}$ and $\boldsymbol{D}'$ are equivalent i.e., $\boldsymbol{D} \sim \boldsymbol{D}'$, if there exists a nonsingular diagonal matrix Σ such that $\boldsymbol{D}' = \boldsymbol{D}\Sigma$. Given $\boldsymbol{D}$, an *equivalence class* is defined as $[\boldsymbol{D}] = \{\boldsymbol{D}' \mid \boldsymbol{D} \sim \boldsymbol{D}'\}$. Hence, an efficient algorithm for finding the optimal mixing matrix should search for the equivalent class $[\boldsymbol{D}]$ that contains the true $\boldsymbol{D}$ rather than a mixing matrix that subjects to scale ambiguity.

Having defined the equivalence relation, the problem is how to formulate the constraint function to remove the scale ambiguity while allowing us to construct algorithms that exploit the equivalence relation. Section IV-C proposes an approach to identify the equivalence class containing the true $\boldsymbol{D}$ up to an unknown permutation.

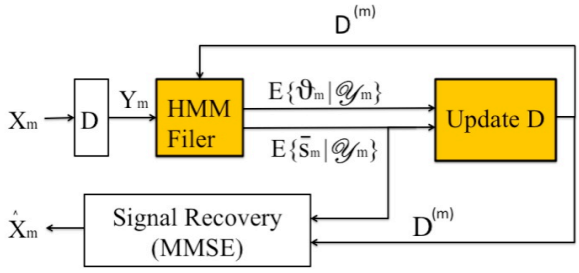

Fig. 1: Proposed Architecture

The proposed U-BSS algorithm combines a powerful HMM-based filtering technique with dictionary learning to

recursively estimate the mixing matrix D and track "sparse" activity of a groups of sources (See Fig. 1). Specifically, we apply a technique known as change-of-measure [30] to decouple the observations from the sources by introducing a new probability measure, so that the observations are independent and uniformly distributed. Under this new measure, estimates of both the state in the Markov chain and the system parameters (i.e., transition matrix of the Markov chain and mixing matrix) can be computed recursively using the expectation-maximization (EM) algorithm.

The contribution of this paper include derivations of the appropriate measure change for recursive HMM state estimation and recursive parameter estimations using expectation-maximization (EM) algorithm, showcasing the application of this HMM filtering technique to blind source separation, and constructing an efficient dictionary learning algorithm with appropriate constraint that enables us to exploit the equivalent properties of the mixing matrices.

B. Paper Organization

The rest of this paper is organized as follows. In Section II, we formulate the temporal dynamics of the source signals as a hidden Markov model (HMM) problem. In Section III, we first apply the change-of-measure technique to the distribution of the observations and then derive an recursive relation for estimating the conditional distribution of the state, i.e., activities of the sources. Section IV derive a recursive expectation-maximization (EM) algorithm for estimating the transition probability of the Markov chain, the over-complete mixing matrix. Finally, numerical results are presented in Section V.

II. HMM Formulation

Recall that $\boldsymbol{x}_m$ at time m is generated by a hidden state $s_m \in \{1, \cdots, N_s\}$, where N_s denotes the total number of states. Given that there are K sources and each source has two processes (i.e., active and inactive), we have $N_s = 2^K$. The hidden state s_m thus specifies the activities of the source signals at each time frame m. Under the sparsity assumption, however, the actual number of states should be much less than 2^K. If we assume that the probability of having more than S number of active sources is zero (i.e., the sparsity of the source signals equals S), then the total number of states is simply $N_s = \sum_{\ell=1}^{S} \binom{K}{\ell}$. In essence, the sparsity constraint, i.e., $\| \boldsymbol{x}_m \|_0 < S + 1$ (or its ℓ_1 convex relaxation) that is encountered in many dictionary learning algorithms for sparse coding, is replaced by the condition that limits the total number of active sources in the construction of the state process. Specifically, in the Markov chain formulation, we consider all possible combinations of no more than S active sources; each combination is represented by a Markov state. This is equivalent to enforcing the ℓ_0 constraint and solving the combinatorial problem.

This is the formulation we pursue here, contrary to the traditional compressive sensing formulations [25], [31], [32]. However we do rely on the sparsity constraint as a condition to have a unique solution, given the dictionary. Since we are looking at all the options combinatorially available, the number of states of the Markov chain (N_s) grows exponentially. However, given the recursive nature of the filtering method, the computational complexity at each iteration is low.

For ease of derivation, in the subsequent discussion, we associate each state $s_m \in \{1, \ldots, N_s\}$ to a canonical basis $\bar{\boldsymbol{s}}_m \in \{\boldsymbol{e}_1, \ldots, \boldsymbol{e}_{N_s}\}$ of $\mathbb{R}^{N_s}$, where e_i has a one in the i^{th} component and zero elsewhere. Thus, we have the relation $\langle \bar{\boldsymbol{s}}_m, \boldsymbol{e}_i \rangle = 1$ if $\bar{\boldsymbol{s}}_m = \boldsymbol{e}_i$ at the m^{th} time frame.

Furthermore, let $\pi_i = P(\bar{\boldsymbol{s}}_1 = \boldsymbol{e}_i)$ be the initial state distribution and $\boldsymbol{A} = [a_{ij}] \in \mathbb{R}^{N_s \times N_s}$ be the transition matrix of the underlying Markov chain, where $a_{ij} = P(\bar{\boldsymbol{s}}_{m+1} = \boldsymbol{e}_j | \bar{\boldsymbol{s}}_m = \boldsymbol{e}_i)$ defines a transition probability. Thus, $\boldsymbol{A}$ is row-stochastic, i.e., $\sum_{j=1}^{N_s} a_{ij} = 1$ for any i, and $\sum_{i=1}^{N_s} \pi_i = 1$.

Let $\sigma\{\bar{\boldsymbol{s}}_\ell,\ 1 \le \ell \le m\}$ be the σ-algebra generated by $\bar{\boldsymbol{s}}_1, \cdots, \bar{\boldsymbol{s}}_m$, then the Markov property implies that $P(\bar{\boldsymbol{s}}_{m+1} = \boldsymbol{e}_i \mid \sigma\{\bar{\boldsymbol{s}}_\ell,\ 1 \le \ell \le m\}) = P(\bar{\boldsymbol{s}}_{m+1} \mid \bar{\boldsymbol{s}}_m)$. Hence, given the transition matrix $\boldsymbol{A}$, we get

$$\mathbb{E}\{\bar{\boldsymbol{s}}_{m+1} \mid \sigma\{\bar{\boldsymbol{s}}_\ell,\ 1 \le \ell \le m\}\} = \mathbb{E}\{\bar{\boldsymbol{s}}_{m+1} \mid \bar{\boldsymbol{s}}_m\} = \boldsymbol{A}^T \bar{\boldsymbol{s}}_m \ . \tag{3}$$

The main advantage of using the canonical basis for the states of the Markov process is that if $\boldsymbol{v}_{m+1} := \bar{\boldsymbol{s}}_{m+1} - \boldsymbol{A}^T \bar{\boldsymbol{s}}_m$, then the state evolution can be written as

$$\bar{\boldsymbol{s}}_{m+1} = \boldsymbol{A}^T \bar{\boldsymbol{s}}_m + \boldsymbol{v}_{m+1}.$$

In particular, $\boldsymbol{v}_m$ is a martingale increment [30] because it follows from (3) that the conditional expectation

$$\mathbb{E}[\boldsymbol{v}_m \mid \sigma\{\bar{\boldsymbol{s}}_\ell,\ 1 \le \ell \le m\}] = \mathbb{E}\{\bar{\boldsymbol{s}}_{m+1} - \boldsymbol{A}^T \bar{\boldsymbol{s}}_m \mid \bar{\boldsymbol{s}}_m\} = 0. \tag{4}$$

The above property plays an important role in the derivations of recursive filters for state and parameter estimations in Section III and IV. (It is to be noted that in HMM literature, the state dynamics are often written as $\bar{\boldsymbol{s}}_{m+1} = \Pi \bar{\boldsymbol{s}}_m + \boldsymbol{v}_{m+1}$ where $\Pi = \boldsymbol{A}^T$ and $\boldsymbol{v}_{m+1} := \bar{\boldsymbol{s}}_{m+1} - \Pi \bar{\boldsymbol{s}}_m$.)

Before proceeding further, we need one additional definition. We denote by

$$\begin{aligned} \mathcal{Y}_m &= \sigma\{\boldsymbol{y}_1, \cdots, \boldsymbol{y}_m\}, \\ \mathcal{G}_m &= \sigma\{\bar{\boldsymbol{s}}_1, \cdots, \bar{\boldsymbol{s}}_m, \boldsymbol{y}_1, \cdots, \boldsymbol{y}_{m-1}\}. \end{aligned}$$

the σ-fields generated by the observations and the combined processes of the states and the observations, respectively.

III. HMM Based Online Filtering

In this section, we first introduce the change of measure technique discussed in [30]. This technique has the important benefit that, under the new probability measure, the observations are independent and identically distributed (i.i.d.). This, in turn, is used to design a recursive filter to estimate the state.

A. Change of Measure

The key motivation for introducing a change in probability measure is that it is often easier to work with data that are i.i.d. It is clear that under the current probability measure P, the observations $\boldsymbol{y}_m$ are correlated. We wish to define a new probability measure $\tilde{P}$ absolutely continuous with respect to P such that $\boldsymbol{y}_m \sim \mathcal{N}(0, \boldsymbol{I})$ at the m^{th} time instant is independent of the state process and parameters and the Radon-Nikodym derivative equals $d\tilde{P}/dP = \lambda$. That is,

$$\int_{\mathcal{A}} d\tilde{P} = \int_{\mathcal{A}} \lambda dP$$

for any measurable set $\mathcal{A}$. The existence of λ follows from Kolmogorov's Extension Theorem and one can also check that (see [30], page 60) for $\boldsymbol{x}_\ell \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda}_\ell)$ i.e., a multivariate Gaussian distribution, the Radon-Nikodym derivative equals

$$\lambda_\ell = \det(\boldsymbol{R}_\ell)^{1/2} f(\boldsymbol{y}_\ell)/f\left(\boldsymbol{R}_\ell^{-1/2}(\boldsymbol{y}_\ell - \boldsymbol{D}\boldsymbol{\mu})\right) \tag{5}$$

where $\boldsymbol{R}_\ell = \boldsymbol{D}\boldsymbol{\Lambda}_\ell\boldsymbol{D}^T + \sigma^2\boldsymbol{I}$ and $f(\cdot)$ is the probability density function of the N-variate Gaussian random variable $\mathcal{N}(0, \boldsymbol{I})$, i.e., $f(\boldsymbol{y}) = (2\pi)^{-N/2}\exp\left(-\frac{1}{2}\boldsymbol{y}^T\boldsymbol{y}\right)$. Thus, the new probability measure $\tilde{P}$ can be constructed by setting the Radon-Nikodym derivative restricted to $\mathcal{G}_m$ to be

$$\left.\frac{d\tilde{P}}{dP}\right|_{\mathcal{G}_m} = \prod_{\ell}^{m} \lambda_\ell \quad \text{and} \quad \lambda_0 = 1.$$

Conversely, suppose that one starts with the probability measure $\tilde{P}$ under which $\boldsymbol{y}_\ell \sim \mathcal{N}(0, \boldsymbol{I})$. To construct the probability measure P from $\tilde{P}$, we shall compute the inverse

$$\left.\frac{dP}{d\tilde{P}}\right|_{\mathcal{G}_m} := \tilde{\Lambda}_m = \prod_{\ell}^{m} \lambda_\ell^{-1} \quad \text{and} \quad \tilde{\Lambda}_0 = 1 \tag{6}$$

such that $w_\ell \sim \mathcal{N}(0, I)$ under the measure P. For simplicity, we denote $\tilde{\lambda}_\ell := \lambda_\ell^{-1}$ for the inverse. The reason for the change of measure is that, under this new probability measure $\tilde{P}$, state and parameter estimations are amenable to be performed recursively.

Remark 1: For most of the results derived in this paper, one big challenge is to determine the appropriate expression for the Radon-Nikodym derivative, i.e., λ_ℓ. It is not a straightforward operation. For the purpose of illustration, we choose to use the multivariate Gaussian distribution for the sources as an example to show the recursive nature of our proposed algorithm.

B. Recursive State Estimation

Let $\tilde{\mathbb{E}}[\cdot]$ denote the expectation under the probability measure $\tilde{P}$ and let $\mathbb{E}$ be the expectation under the probability measure P. Using Conditional Bayes' Theorem [30], the estimate for $\bar{\boldsymbol{s}}_m$ given $\mathcal{Y}_m$ can be written as:

$$\mathbb{E}[\bar{\boldsymbol{s}}_m|\mathcal{Y}_m] = \frac{\tilde{\mathbb{E}}[\tilde{\Lambda}_m\bar{\boldsymbol{s}}_m|\mathcal{Y}_m]}{\tilde{\mathbb{E}}[\tilde{\Lambda}_m|\mathcal{Y}_m]} = \frac{\tilde{\mathbb{E}}[\tilde{\Lambda}_m\bar{\boldsymbol{s}}_m|\mathcal{Y}_m]}{\langle\tilde{\mathbb{E}}[\tilde{\Lambda}_m\bar{\boldsymbol{s}}_m|\mathcal{Y}_m], \mathbf{1}\rangle} \quad . \tag{7}$$

The last equality holds because

$$\tilde{\mathbb{E}}[\tilde{\Lambda}_m|\mathcal{Y}_m] = \tilde{\mathbb{E}}[\tilde{\Lambda}_m\langle\bar{\boldsymbol{s}}_m, \mathbf{1}\rangle|\mathcal{Y}_m] = \langle\tilde{\mathbb{E}}[\tilde{\Lambda}_m\bar{\boldsymbol{s}}_m|\mathcal{Y}_m], \mathbf{1}\rangle. \tag{8}$$

To investigate the recursion for the state estimator $\mathbb{E}[\bar{\boldsymbol{s}}_m|\mathcal{Y}_m]$, it follows from (7) that an alternative approach is to consider a recursive formulation for $\tilde{\mathbb{E}}[\tilde{\Lambda}_m\bar{\boldsymbol{s}}_m|\mathcal{Y}_m]$ under the probability measure $\tilde{P}$.

Clearly, the construction of the Radon-Nikodym derivative restricted to $\mathcal{G}_m$ (i.e., $\tilde{\Lambda}_m$) involves the process $\tilde{\lambda}_\ell$, which is a function of both the observation $\boldsymbol{y}_\ell$ and the state $\bar{\boldsymbol{s}}_\ell$. For ease of notation, we define the following vector

$$\bar{\zeta}(\boldsymbol{y}_\ell) = [\zeta_1(\boldsymbol{y}_\ell), \cdots, \zeta_{N_s}(\boldsymbol{y}_\ell)]^T \tag{9}$$

in which the element $\zeta_i(\boldsymbol{y}_\ell)$ is the value of $\tilde{\lambda}_\ell$ given $\boldsymbol{y}_\ell$ and the state $\bar{\boldsymbol{s}}_\ell = \boldsymbol{e}_i$, i.e., $\zeta_i(\boldsymbol{y}_\ell) := \tilde{\lambda}_\ell|_{\bar{\boldsymbol{s}}_\ell = \boldsymbol{e}_i}$.

Furthermore, let

$$\mathcal{F}_m(\bar{s}_m) := \tilde{\mathbb{E}}[\tilde{\Lambda}_m\bar{s}_m|\mathcal{Y}_m]$$

Then using the relation in (3), the following recursive formulation is derived:

$$\begin{aligned}
\mathcal{F}_{m+1}(\bar{\boldsymbol{s}}_{m+1}) &= \tilde{\mathbb{E}}\{\tilde{\Lambda}_m\tilde{\lambda}_{m+1}\bar{\boldsymbol{s}}_{m+1}|\mathcal{Y}_{m+1}\} \\
&= \sum_{j=1}^{N_s} \tilde{\mathbb{E}}\{\tilde{\Lambda}_m\zeta_j(\boldsymbol{y}_{m+1})\bar{\boldsymbol{s}}_{m+1}\langle\bar{\boldsymbol{s}}_{m+1}, \boldsymbol{e}_j\rangle|\mathcal{Y}_{m+1}\} \\
&= \sum_{i,j=1}^{N_s} \zeta_j(\boldsymbol{y}_{m+1})a_{ij}\tilde{\mathbb{E}}\{\tilde{\Lambda}_m\langle\bar{\boldsymbol{s}}_m, \boldsymbol{e}_i\rangle|\mathcal{Y}_m\}\boldsymbol{e}_j \\
&= \sum_{i,j=1}^{N_s} \zeta_j(\boldsymbol{y}_{m+1})a_{ij}\langle\mathcal{F}_m(\bar{\boldsymbol{s}}_m), \boldsymbol{e}_i\rangle\boldsymbol{e}_j \\
&= \text{diag}\left(\bar{\zeta}(\boldsymbol{y}_{m+1})\right)\boldsymbol{A}^T\mathcal{F}_m(\bar{\boldsymbol{s}}_m).
\end{aligned} \tag{10}$$

Note that $\mathcal{F}_m(\bar{\boldsymbol{s}}_m)$ is an unnormalized conditional expectation of $\bar{\boldsymbol{s}}_m$ conditioned on $\mathcal{Y}_m$. To normalize it, it follows from (7) that the state estimator (i.e., conditional distribution of the state) is given by

$$\mathbb{E}[\bar{\boldsymbol{s}}_m|\mathcal{Y}_m] = \mathcal{F}_m(\bar{\boldsymbol{s}}_m)/\langle\mathcal{F}_m(\bar{\boldsymbol{s}}_m), \mathbf{1}\rangle. \tag{11}$$

In particular, $\mathcal{F}_m(\bar{\boldsymbol{s}}_m)$ is computed recursively using (10).

IV. Fast Online Algorithm Using HMM Filters

In this section, we derive recursive filters to estimate the model parameters. Then we apply the EM algorithm for likelihood maximization. The key is to develop a recursive relation that leads to the derivation of recursive formulae for parameter estimations.

A. E-Step: Recursive Computations of the Q-Function

Consider now the problem of estimating the parameters in the set $\xi = \{\boldsymbol{D}, \boldsymbol{A}\}$. From statistical inference theory, given a family of probability measure $\{P_\xi,\ \xi \in \Xi\}$ on a measurable space absolutely continuous with respect to a reference probability measure P_{ξ_r}, the likelihood for computing ξ based on the information in $\mathcal{Y}$ is

$$\mathcal{L}(\xi) = \mathbb{E}_{\xi_r}\left[\left.\frac{dP_\xi}{dP_{\xi_r}}\right|\mathcal{Y}\right],$$

where $\mathbb{E}_{\xi_r}[\cdot]$ represents the expectation under the probability measure P_{ξ_r} with respect to the parameter ξ_r.

Given the likelihood function $\mathcal{L}(\xi)$, let $\xi^{(m-1)} = \{\boldsymbol{D}^{(m-1)}, \boldsymbol{A}^{(m-1)}\}$ be the estimated parameters at time $m-1$ and $\xi^{(m)} = \{\boldsymbol{D}^{(m)}, \boldsymbol{A}^{(m)}\}$ be the updated parameter after observing the signal $\boldsymbol{y}_m$ at time m. Using Jensen's inequality, it can be shown that

$$\log \mathcal{L}(\xi^{(m)}) - \log \mathcal{L}(\xi^{(m-1)}) \geq Q(\xi^{(m)}, \xi^{(m-1)}) \tag{12}$$

where $Q(\xi^{(m)}, \xi^{(m-1)}) := \mathbb{E}_{\xi^{(m-1)}}\left[\log \frac{dP_{\xi^{(m)}}}{dP_{\xi^{(m-1)}}} \middle| \mathcal{Y}_m\right]$ and the equality in (12) holds if and only if $\xi^{(m)} = \xi^{(m-1)}$. However, maximizing the likelihood function is not straightforward and compute. Using the EM algorithm, the estimated sequence $\{\xi^{(m)}\}$ by maximizing the Q-function is shown to yield non-decreasing values of the likelihood function, which converges to the local maximum of the likelihood function [33].

While detailed formulations of the Q-function with respect to the parameters will be presented shortly, the following processes which will be used to update the set ξ in the M-step, are defined [30]:

$$\mathcal{J}_m^{ij} := \sum_{\ell=1}^{m} \langle \bar{\boldsymbol{s}}_{\ell-1}, \boldsymbol{e}_i \rangle \langle \bar{\boldsymbol{s}}_{\ell}, \boldsymbol{e}_j \rangle \tag{13}$$

$$\mathcal{O}_m^{i} := \sum_{\ell=1}^{m} \langle \bar{\boldsymbol{s}}_{\ell-1}, \boldsymbol{e}_i \rangle \tag{14}$$

$$\mathcal{T}_m^{i}(g) := \sum_{\ell=1}^{m} \langle \bar{\boldsymbol{s}}_{\ell}, \boldsymbol{e}_i \rangle g_\ell \ . \tag{15}$$

where $g_\ell \in \mathbb{R}$ denotes a function of $\boldsymbol{y}_\ell$. In this paper g_ℓ is the $(r,h)^{\text{th}}$ entry of $\boldsymbol{y}_\ell \boldsymbol{y}_\ell^T$ and the expectation of the $\mathcal{T}_m^i(g_\ell)$ for each (r,h) is the corresponding entry of the conditional covariance matrix for the observation. The term $\mathcal{J}_m^{ij}$ defines the the *number of jumps* from state i to state j in time m, while $\mathcal{O}_m^i$ defines the *occupation time* that the Markov chain $\bar{\boldsymbol{s}}_\ell$ occupies the state i up to time m.

1) **Recursive Filters for $\mathcal{J}_m^{ij}$ and $\mathcal{O}_m^i$:** For reasons we will explain shortly, we now provide analogous recursions for $\mathcal{J}_m^{ij}$ and $\mathcal{O}_m^i$. Again, by applying the conditional Bayes' theorem, it is immediate to see that

$$\mathbb{E}[\mathcal{J}_m^{ij}|\mathcal{Y}_m] = \frac{\tilde{\mathbb{E}}[\tilde{\Lambda}_m \mathcal{J}_m^{ij}|\mathcal{Y}_m]}{\tilde{\mathbb{E}}[\tilde{\Lambda}_m|\mathcal{Y}_m]} \ , \ \mathbb{E}[\mathcal{O}_m^{i}|\mathcal{Y}_m] = \frac{\tilde{\mathbb{E}}[\tilde{\Lambda}_m \mathcal{O}_m^{i}|\mathcal{Y}_m]}{\tilde{\mathbb{E}}[\tilde{\Lambda}_m|\mathcal{Y}_m]}. \tag{16}$$

Unlike the expression for the state estimator in (11), there is no recursion for $\tilde{E}[\tilde{\Lambda}_m \mathcal{J}_m^{ij}|\mathcal{Y}_m]$ or $\tilde{E}[\tilde{\Lambda}_m \mathcal{O}_m^{i}|\mathcal{Y}_m]$. However, by exploring the semi-martingale property of the state $\bar{\boldsymbol{s}}_\ell$ given in (3) and let

$$\mathcal{F}_{m+1}(\mathcal{J}_{m+1}^{ij}\bar{\boldsymbol{s}}_{m+1}) := \tilde{E}[\tilde{\Lambda}_{m+1}\mathcal{J}_{m+1}^{ij}\bar{\boldsymbol{s}}_{m+1}|\mathcal{Y}_{m+1}]$$
$$\mathcal{F}_{m+1}(\mathcal{O}_{m+1}^{i}\bar{\boldsymbol{s}}_{m+1}) := \tilde{E}[\tilde{\Lambda}_{m+1}\mathcal{O}_{m+1}^{i}\bar{\boldsymbol{s}}_{m+1}|\mathcal{Y}_{m+1}],$$

the following recursive relations can be derived (Appendix A):

$$\mathcal{F}_{m+1}(\mathcal{J}_{m+1}^{ij}\bar{\boldsymbol{s}}_{m+1}) = \operatorname{diag}(\bar{\zeta}(\boldsymbol{y}_{m+1}))A^T\mathcal{F}_m(\mathcal{J}_m^{ij}\bar{s}_m) + \zeta_j(\boldsymbol{y}_{m+1})a_{ij}\langle \mathcal{F}_m(\bar{s}_m), \boldsymbol{e}_i\rangle \boldsymbol{e}_j \tag{17}$$

$$\mathcal{F}_{m+1}(\mathcal{O}_{m+1}^{i}\bar{\boldsymbol{s}}_{m+1}) = \operatorname{diag}(\bar{\zeta}(\boldsymbol{y}_{m+1}))A^T[\mathcal{F}_m(\mathcal{O}_m^{i}\bar{\boldsymbol{s}}_m) + \langle \mathcal{F}_m(\bar{\boldsymbol{s}}_m), \boldsymbol{e}_i\rangle \boldsymbol{e}_i] \tag{18}$$

where $\bar{\zeta}(\boldsymbol{y}_m)$ is defined in (9). It is to be noted that given any scalar process H_m (i.e., $\mathcal{J}_m^{ij}$ or $\mathcal{O}_m^i$), we get

$$\langle \phi_m(H_m\bar{\boldsymbol{s}}_m), \mathbf{1}\rangle = \phi_m(H_m\langle \bar{\boldsymbol{s}}_m, \mathbf{1}\rangle) = \phi_m(H_m).$$

Replacing H_m with $\mathcal{J}_m^{ij}$ and $\mathcal{O}_m^i$, respectively, we obtain the following relations:

$$\mathcal{F}_m(\mathcal{J}_m^{ij}) := \tilde{\mathbb{E}}[\tilde{\Lambda}_m\mathcal{J}_m^{ij}|\mathcal{Y}_m] = \langle \mathcal{F}_m(\mathcal{J}_m^{ij}\bar{\boldsymbol{s}}_m), \mathbf{1}\rangle$$
$$\mathcal{F}_m(\mathcal{O}_m^{i}) := \tilde{\mathbb{E}}[\tilde{\Lambda}_m\mathcal{O}_m^{i}|\mathcal{Y}_m] = \langle \mathcal{F}_m(\mathcal{O}_m^{i}\bar{\boldsymbol{s}}_m), \mathbf{1}\rangle.$$

Hence, it follows from (16) that the recursive formulations for $\mathcal{J}_m^{ij}$ and $\mathcal{O}_m^i$ can be expressed as

$$\mathbb{E}[\mathcal{J}_m^{ij}|\mathcal{Y}_m] = \langle \mathcal{F}_m(\mathcal{J}_m^{ij}\bar{\boldsymbol{s}}_m), \mathbf{1}\rangle / \langle \mathcal{F}_m(\bar{\boldsymbol{s}}_m), \mathbf{1}\rangle, \tag{19}$$

$$\mathbb{E}[\mathcal{O}_m^{i}|\mathcal{Y}_m] = \langle \mathcal{F}_m(\mathcal{O}_m^{i}\bar{\boldsymbol{s}}_m), \mathbf{1}\rangle / \langle \mathcal{F}_m(\bar{\boldsymbol{s}}_m), \mathbf{1}\rangle. \tag{20}$$

using the filters derived in (17) and (18).

2) **Recursive Filter for $\mathcal{T}_m^i(g)$:** Using the same technique as in the above derivations, define

$$\mathcal{F}_m\left(\mathcal{T}_m^i(g)\bar{\boldsymbol{s}}_m\right) := \tilde{E}\left[\tilde{\Lambda}_m\mathcal{T}_m^i(g)\bar{\boldsymbol{s}}_m|\mathcal{Y}_m\right].$$

The following recursive relation can be derived (Appendix A):

$$\mathcal{F}_{m+1}\left(\mathcal{T}_{m+1}^i(g)\bar{\boldsymbol{s}}_{m+1}\right) = \operatorname{diag}(\bar{\zeta}(\boldsymbol{y}_{m+1}))\boldsymbol{A}^T\mathcal{F}_m\left(\mathcal{T}_m^i(g)\bar{\boldsymbol{s}}_m\right) + \zeta_i(\boldsymbol{y}_{m+1})g_{m+1}\boldsymbol{e}_i^T A^T\mathcal{F}_m(\bar{\boldsymbol{s}}_m)\boldsymbol{e}_i \tag{21}$$

Applying the conditional Bayes' theorem yields

$$\mathbb{E}\left[\mathcal{T}_m^i(g)|\mathcal{Y}_m\right] = \frac{\tilde{\mathbb{E}}\left[\tilde{\Lambda}_m\mathcal{T}_m^i(g)|\mathcal{Y}_m\right]}{\tilde{\mathbb{E}}\left[\tilde{\Lambda}_m|\mathcal{Y}_m\right]} = \frac{\langle \mathcal{F}_m\left(\mathcal{T}_m^i(g)\bar{\boldsymbol{s}}_m\right), \mathbf{1}\rangle}{\langle \mathcal{F}_m(\bar{\boldsymbol{s}}_m), \mathbf{1}\rangle}. \tag{22}$$

Having derived the expressions for the recursive filters in (19), (20) and (22), we are now ready to consider the problem of estimating the parameters $\boldsymbol{A}$ and $\boldsymbol{D}$. In particular, the estimation process is made in the order: $\boldsymbol{A}, \boldsymbol{D}$ at each time instant. These parameters are then updated iteratively in this order upon receipt of new observations, as detailed in the following subsections.

B. M-Step: Recursive Estimation $\boldsymbol{A} = [a_{ij}]$

Recall from (12) that, in order to update the transition matrix $\boldsymbol{A}$ by maximizing the likelihood function, we need to first compute the Radon-Nikodym derivative $\frac{dP_{\xi^{(m)}}}{dP_{\xi^{(m-1)}}}\big|_{\mathcal{G}_m} := \Lambda_m^{\boldsymbol{A}}$ while fixing the values for $\boldsymbol{D}^{(m-1)}$ and $\boldsymbol{\theta}^{(m-1)}$ obtained from the previous update. The superscript $\boldsymbol{A}$ in $\Lambda_m^{\boldsymbol{A}}$ is to indicate that the focus now is on the dependence of the Radon-Nikodym derivative on the transition probability matrix $\boldsymbol{A}$.

Suppose that under the probability measure $P_{\xi^{(m-1)}}$ with respect to the previous estimate $\xi^{(m-1)}$, the process $\bar{\boldsymbol{s}}_\ell$ is a Markov chain with a transition matrix $\boldsymbol{A}^{(m-1)} = [a_{ij}^{(m-1)}]$. We want to construct a new probability measure $P_{\xi^{(m)}}$ induced from $P_{\xi^{(m-1)}}$ such that under $P_{\xi^{(m)}}$ the process $\bar{\boldsymbol{s}}_\ell$ is a Markov chain with a transition matrix $\boldsymbol{A}^{(m)} = [a_{ij}^{(m)}]$. Moreover, let $\mathbb{E}_{\xi^{(m-1)}}[\cdot]$ and $\mathbb{E}_{\xi^{(m)}}[\cdot]$ denote the expectations under the probability measures $P_{\xi^{(m-1)}}$ and $P_{\xi^{(m)}}$, respectively. Then the Radon-Nikodym derivative $\Lambda_m^{\boldsymbol{A}}$ of the induced probability measure $P_{\xi^{(m)}}$ with respect to $P_{\xi^{(m-1)}}$ must satisfy the following relation: given the previous state $\bar{\boldsymbol{s}}_{m-1} = \boldsymbol{e}_i$, for $\forall i, j \in [1, \ldots, N_s]$,

$$a_{ij}^{(m)} := \mathbb{E}_{\xi^{(m)}}[\langle \bar{\boldsymbol{s}}_m, e_j\rangle | \mathcal{Y}_m] = \frac{\mathbb{E}_{\xi^{(m-1)}}[\Lambda_m^A \langle \bar{\boldsymbol{s}}_m, \boldsymbol{e}_j\rangle | \mathcal{Y}_m]}{\mathbb{E}_{\xi^{(m-1)}}[\Lambda_m^{\boldsymbol{A}} | \mathcal{Y}_m]}. \tag{23}$$

Thus, to update the transition probabilities $\boldsymbol{A} = [a_{ij}]$, the Radon-Nikodym derivative equals

$$\Lambda_m^{\boldsymbol{A}} = \prod_{\ell=1}^{m} \prod_{i,j=1}^{N_s} \left[\frac{a_{ij}^{(m)}}{a_{ij}^{(m-1)}} \right]^{\langle \bar{\boldsymbol{s}}_\ell, \boldsymbol{e}_j\rangle \langle \bar{\boldsymbol{s}}_{\ell-1}, \boldsymbol{e}_i\rangle}.$$

This can be justified by plugging the above expression into the relation in (23) (see page 37 in [30]).

Hence, given the previous estimate $\xi^{(m-1)}$, we want to find a transition matrix $\boldsymbol{A}^{(m)}$ maximizing the Q-function, i.e.,

$$\begin{aligned}
\boldsymbol{A}^{(m)} &= \max_{\boldsymbol{A}} \; Q^{\boldsymbol{A}}(\xi, \xi^{(m-1)}) \\
&= \max_{\boldsymbol{A}} \; \mathbb{E}_{\xi^{(m-1)}} \left[\log \Lambda_m^{\boldsymbol{A}} \middle| \mathcal{Y}_m \right] \\
&= \max_{\boldsymbol{A}} \; \mathbb{E}_{\xi^{(m-1)}} \left[\sum_{\ell=1}^{m} \sum_{i,j=1}^{N_s} \langle \bar{\boldsymbol{s}}_\ell, \boldsymbol{e}_j\rangle \langle \bar{\boldsymbol{s}}_{\ell-1}, \boldsymbol{e}_i\rangle \middle| \mathcal{Y}_m \right] \log a_{ij} \\
&= \max_{\boldsymbol{A}} \sum_{i,j=1}^{N_s} \mathbb{E}_{\xi^{(m-1)}} \left[\mathcal{J}_m^{ij} | \mathcal{Y}_m \right] \log a_{ij}.
\end{aligned}$$

Under the constraint that the transition matrix $\boldsymbol{A} = [a_{ij}]$ is row stochastic, using the method of Lagrange multipliers and equating the derivative to 0 yield

$$a_{ij}^{(m)} = \frac{\mathbb{E}_{\xi^{(m-1)}}[\mathcal{J}_m^{ij} | \mathcal{Y}_m]}{\mathbb{E}_{\xi^{(m-1)}}[\mathcal{O}_m^{i} | \mathcal{Y}_m]} = \frac{\langle \mathcal{F}_m(\mathcal{J}_m^{ij} \bar{\boldsymbol{s}}_m), \boldsymbol{1}\rangle}{\langle \mathcal{F}_m(\mathcal{O}_m^{i} \bar{\boldsymbol{s}}_m), \boldsymbol{1}\rangle}.$$

This is our estimate for the transition matrix, given the observations $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_m$ up to time m. Note that the terms $\mathcal{F}_m(\mathcal{J}_m^{ij} \bar{\boldsymbol{s}}_m)$ and $\mathcal{F}_m(\mathcal{O}_m^{i} \bar{\boldsymbol{s}}_m)$ in the preceding expression are computed recursively using the recursive filters derived in (17) and (18).

C. M-Step: Recursive Estimations of $\boldsymbol{D}$

We now provide analogous estimates for the mixing matrix $\boldsymbol{D}$. Again, a change of measure is performed. Let $\tilde{P}$ be the probability measure under which the observation $\{\boldsymbol{y}_m\}$ is an sequence of i.i.d. Gaussian random vectors $\mathcal{N}(0, I)$. One can then construct another measure P_ξ with respect to the current parameter ξ from $\tilde{P}$ such that $w_m \sim \mathcal{N}(0, \boldsymbol{I})$. For example, if $\boldsymbol{x}_\ell \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Lambda}_\ell)$, then it follows from (6) that $dP_\xi / d\tilde{P}$ restricted to $\mathcal{G}_m$ equals

$$\left. \frac{dP_\xi}{d\tilde{P}} \right|_{\mathcal{G}_m} = \prod_{\ell=1}^{m} \frac{f\left(\boldsymbol{R}_\ell^{-1/2}(\boldsymbol{y}_\ell - \boldsymbol{D}\,\boldsymbol{\mu})\right)}{f\left(\boldsymbol{y}_\ell\right) \det(\boldsymbol{R}_\ell)^{1/2}} \tag{24}$$

where $f(\cdot)$ denotes the $\mathcal{N}(0, I)$ density function and $\boldsymbol{R}_\ell = \boldsymbol{D}\boldsymbol{\Lambda}_{\boldsymbol{\ell}}\boldsymbol{D}^T + \sigma^2 \boldsymbol{I}$. Hence, the Radon-Nikodym derivative $\frac{dP_{\xi^{(m)}}}{dP_{\xi^{(m-1)}}}\big|_{\mathcal{G}_m} := \Lambda_m^{\boldsymbol{D}}$ in the Q-function can be expressed as the following product:

$$\Lambda_m^{\boldsymbol{D}} = \left. \frac{dP_{\xi^{(m)}}}{d\tilde{P}} \right|_{\mathcal{G}_m} \left. \frac{d\tilde{P}}{dP_{\xi^{(m-1)}}} \right|_{\mathcal{G}_m}.$$

With a slight abuse of notation, in this section, we denote $\xi^{(m-1)} = \{\boldsymbol{A}^{(m)}, \boldsymbol{D}^{(m-1)}\}$ in which $\boldsymbol{A}^{(m-1)}$ is replaced with its updated value $\boldsymbol{A}^{(m)}$. Then $\boldsymbol{D}^{(m)}$ can be computed by maximizing the Q-function:

$$\begin{aligned}
\boldsymbol{D}^{(m)} &= \max_{\boldsymbol{D}} \overline{Q}(\boldsymbol{D}) := Q^{\boldsymbol{D}}(\xi, \xi^{(m-1)}) \\
&= \max_{\boldsymbol{D}} \mathbb{E}_{\xi^{(m-1)}} \left[\log \Lambda_m^{\boldsymbol{D}} \middle| \mathcal{Y}_m \right] \\
&= \min_{\boldsymbol{D}} \mathbb{E}_{\xi^{(m-1)}} \left[\log \frac{d\tilde{P}}{dP_\xi} \middle| \mathcal{Y}_m \right].
\end{aligned} \tag{25}$$

Hence, maximizing the Q-function is equivalent to minimizing the *cost function* $C(\boldsymbol{D}) := \mathbb{E}_{\xi^{(m-1)}} \left[\log \frac{d\tilde{P}}{dP_\xi} \middle| \mathcal{Y}_m \right]$.

Recall from Section I-A that the key idea for an efficient dictionary learning algorithm is to formulate a constraint function that allows us to exploit the equivalence relation due to the scale ambiguity of the system. Many existing algorithms impose a unit norm constraint on columns of $\boldsymbol{D}$. Under this constraint, the scale ambiguity is reduced to a sign ambiguity, i.e., $[\boldsymbol{d}_k] = \{\boldsymbol{d}_k, -\boldsymbol{d}_k\}$. However, the resulting algorithm does not (fully) exploit the equivalence relation. In particular, the optimization algorithm has to alternate between estimating mixing matrix and signal variances. Thus it is more costly and difficult to achieve fast convergence.

Alternatively, one can also specify the variances of the source signals. For illustrative purpose, consider the case when $x_i(\ell) \sim \mathcal{N}(0, \sigma_i^2(\ell))$, given that the i^{th} source is active at time ℓ. Let $\omega(\bar{\boldsymbol{s}}_\ell) = \{i \mid i \text{ is active at state } \bar{\boldsymbol{s}}_\ell\}$ be the set of active sources at time ℓ and $\Omega(\bar{\boldsymbol{s}}_\ell) \in \mathbb{R}^{K \times K}$ be a diagonal matrix, with ones at the $(\omega_i(\bar{\boldsymbol{s}}_\ell), \omega_i(\bar{\boldsymbol{s}}_\ell))^{\text{th}}$ entries for $i = 1, \ldots, |\omega_\ell|$, and zeros elsewhere. If we impose the following constraint on the variances,

$$\sigma_i^2(\ell) = c \;, \;\; \text{if } \; i \in \omega(\bar{\boldsymbol{s}}_\ell) \tag{26}$$

with the above notation, an equivalent expression of the cost function is

$$C(\boldsymbol{D}) = \mathbb{E}\left\{ \sum_{\ell=1}^{m} \log\det \boldsymbol{R}_\ell + \operatorname{tr}\left[\boldsymbol{y}_\ell \boldsymbol{y}_\ell^T \boldsymbol{R}_\ell^{-1} \right] \mid \mathcal{Y}_m \right\}$$

where $\boldsymbol{R}_\ell = c\boldsymbol{D}\Omega_\ell\boldsymbol{D}^T + \sigma^2\boldsymbol{I}$. Let $g = \boldsymbol{y}_m \boldsymbol{y}_m^T$ and $\tilde{\boldsymbol{\mathcal{T}}}_m^i(g) = \left[\mathcal{T}_m^i(g^{ab})\right]$ denotes the matrix whose $(a,b)^{\text{th}}$

element is $\mathcal{T}^r_m(g^{ab})$. It follows from (24) and (25) that the derivative of the cost function $C(\boldsymbol{D})$ with respect to $\boldsymbol{D}$ is:

$$\dot{C}(\boldsymbol{D}) = \sum_{i=1}^{N_s} \mathbb{E}\{\mathcal{O}^i_m \mid \mathcal{Y}_m\}\boldsymbol{R}_i^{-1}\boldsymbol{D}_i - \sum_{i=1}^{N_s} \boldsymbol{R}_i^{-1}\,\mathbb{E}\{\tilde{\boldsymbol{\mathcal{T}}}^i_m(g) \mid \mathcal{Y}_m\}\boldsymbol{R}_i^{-1}\boldsymbol{D}_i \qquad (27)$$

where $\boldsymbol{D}_i = \boldsymbol{D}\Omega(\boldsymbol{e}_i)$ and $\boldsymbol{R}_i = c\boldsymbol{D}\Omega(\boldsymbol{e}_i)\boldsymbol{D}^T + \sigma^2\boldsymbol{I}$. In particular, $\mathbb{E}\{\mathcal{O}^i_m \mid \mathcal{Y}_m\}$ and elements in $\mathbb{E}[\mathcal{T}^i_m(g)]$ can be computed using the recursive filters derived in (20) and (22). Using the gradient descent method, the estimated mixing matrix $\boldsymbol{D}^{(m)}$ should be attracted toward the set

$$Q_m = \left\{\boldsymbol{D} \in \mathbb{R}^{N\times K} \mid \sigma_i^2(m) = 1 \text{ for } \forall i \in \omega(\bar{\boldsymbol{s}}_m)\right\}.$$

One problem with this approach is that although $\boldsymbol{D}^{(m)}$ can move freely toward Q_m, if the source signals are non-stationary (e.g., signals with time-varying variances), the algorithm needs to explicitly adjust the scale of $\boldsymbol{D}$ to the change in the variance. As a result, the algorithm will likely be inefficient.

In the following, we propose an efficient algorithm that achieves fast convergence and also works well against changes in the variances of the source signals. Specifically, the algorithm imposes no explicit constraint on the mixing matrix $\boldsymbol{D}$ by introducing an equivalence relation on the set of all $\boldsymbol{D}$ and then searches for the equivalent class directly.

Recall the definition of equivalence class in Section I-A. An equivalent class on the set of mixing matrices (i.e., $\mathbb{R}^{N\times K}$) is defined as $[\boldsymbol{D}] = \{\boldsymbol{D}' \mid \boldsymbol{D} \sim \boldsymbol{D}'\}$ with respect to the following equivalence relation:

$$\boldsymbol{D} \sim \boldsymbol{D}' : \boldsymbol{D}' = \boldsymbol{D}\Sigma$$

for some nonsingular diagonal matrix Σ. In particular, the equivalence relation corresponds to a partition of the space $\mathbb{R}^{N\times K} - \{0\}$ into disjoint set of equivalence classes. Furthermore, the equivalence relation defined in the space of $\boldsymbol{D}$ also induces an equivalence relation in $\mathbb{R}^N - \{0\}$, where vectors $\boldsymbol{d}_k, \boldsymbol{d}'_k$ are equivalent if there exists a scalar α_k such that $\boldsymbol{d}'_k = \alpha_k\boldsymbol{d}_k$. Thus an equivalence class in $\mathbb{R}^N - \{0\}$ given $\sim$ is defined as $[\boldsymbol{d}_k] := \{\boldsymbol{d}'_k \mid \boldsymbol{d}_k \sim \boldsymbol{d}'_k\}$. It is well known that a collection of $[\boldsymbol{d}_k]$ in $\mathbb{R}^N - \{0\}$ forms a $(N-1)$-dimensional projective space, denoted P^{N-1}. As a result, the set of all the equivalence classes $[\boldsymbol{D}]$ under $\sim$ can be shown to form a product manifold of K projective spaces, i.e.,

$$\mathcal{M} = P^{N-1} \times \cdots \times P^{N-1}.$$

Moreover, the tangent space $\mathcal{T}_{[\boldsymbol{D}]}\mathcal{M}$ at the equivalence class $[\boldsymbol{D}]$ is the $(KN-K)$-dimensional subspace

$$\mathcal{T}_{[\boldsymbol{D}]}\mathcal{M} = \{\boldsymbol{V} = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_K] \mid \boldsymbol{d}_k^T\boldsymbol{v}_k = 0 \text{ for } \forall k\}.$$

In other words, the tangent space at $[\boldsymbol{D}]$ specifies a set of possible directions at which one can move from $[\boldsymbol{D}]$ to another equivalence class.

Let $\boldsymbol{V}_1, \boldsymbol{V}_2 \in \mathcal{T}_{[\boldsymbol{D}]}\mathcal{M}$ be two tangent vectors at $\boldsymbol{D}$. We define the inner product for $\mathcal{T}_{\boldsymbol{D}}\mathcal{M}$ to be $\langle \boldsymbol{V}_1, \boldsymbol{V}_2\rangle = \mathrm{tr}(\boldsymbol{V}_1^T\boldsymbol{V}_2)$. This leads to the following updating rule:

$$\boldsymbol{D} \leftarrow \boldsymbol{D} - \epsilon\left[\dot{C}(\boldsymbol{D}) - \boldsymbol{D}\Sigma^{-1}(\boldsymbol{D})\mathrm{Diag}\left(\boldsymbol{D}^T\dot{C}(\boldsymbol{D})\right)\right] \qquad (28)$$

where ϵ denotes the step size and $\mathrm{Diag}(X)$ returns a matrix with the given diagonal and zero off-diagonal entries and $\Sigma^{-1}(\boldsymbol{D}) = \mathrm{Diag}(\boldsymbol{D}^T\boldsymbol{D})$. It is straightforward to show that the descent direction $\nabla C(\boldsymbol{D}) = -\dot{C}(\boldsymbol{D}) + \boldsymbol{D}\Sigma(\boldsymbol{D})\mathrm{Diag}(\boldsymbol{D}^T\dot{C}(\boldsymbol{D})) \in \mathcal{T}_{\boldsymbol{D}}\mathcal{M}$ is a tangent vector at the equivalence class $[\boldsymbol{D}]$.

D. Stability Analysis

Although we do not impose constraint on the set of mixing matrices, for practical purpose, it is often necessary to study conditions under which the estimate $\boldsymbol{D}$ is bounded, as investigated in the next lemma.

Lemma 4.1: *The updating rule*

$$\frac{d\boldsymbol{D}}{dt} = -\dot{C}(\boldsymbol{D}) + \boldsymbol{D}\Sigma^{-1}\mathrm{Diag}(\boldsymbol{D}^T\dot{C}(\boldsymbol{D})) \qquad (29)$$

where Σ is a nonsingular diagonal matrix with positive entries, is stable if the diagonal entries of the matrix $\boldsymbol{D}^T\dot{C}(\boldsymbol{D})$ are non-positive.

Proof: Given (29), the derivative of $\mathrm{diag}(\boldsymbol{D}^T\boldsymbol{D})$ is

$$\frac{d\mathrm{Diag}(\boldsymbol{D}^T\boldsymbol{D})}{dt} = -2\mathrm{Diag}(\boldsymbol{D}^T\dot{C}(\boldsymbol{D})) + 2\mathrm{Diag}\left(\boldsymbol{D}^T\boldsymbol{D}\right)\Sigma^{-1}\mathrm{Diag}\left(\boldsymbol{D}^T\dot{C}(\boldsymbol{D})\right).$$

Let $z_k(t) = \boldsymbol{d}_k^T\boldsymbol{d}_k$, then

$$\frac{dz_k}{dt} = 2(z_k(t)\Sigma_{kk}^{-1} - 1)\boldsymbol{d}_k^T\dot{C}(\boldsymbol{d}_k)$$

where $\dot{C}(\boldsymbol{d}_k)$ is the derivative of the cost function with respect to $\boldsymbol{d}_k$ (i.e., the k^{th} column of $\dot{C}(\boldsymbol{D})$) and Σ_{kk} denotes the $(k,k)^{\text{th}}$ element of Σ. Solving the above differential equation yields

$$z_k(t) = \Sigma_{kk} + e^{2\int_0^t \Sigma_{kk}^{-1}\boldsymbol{d}_k^T\dot{C}(\boldsymbol{d}_k)dt},$$

with an initial condition $z_k(0) = \Sigma_{kk}$. Hence, if $\boldsymbol{d}_k^T\dot{C}(\boldsymbol{d}_k) < 0$, then the exponential term decays to zero and the norm $z_k(t) = \boldsymbol{d}_k^T\boldsymbol{d}_k$ is approximately equal to Σ_{kk}. ∎

Suppose that the updating rule is stable and it yields an matrix $\boldsymbol{D}$ whose k^{th} column norm approximately equals Σ_{kk}. Hence, it is straightforward to check that $d\boldsymbol{d}_k/dt = -\dot{C}(\boldsymbol{d}_k) + \boldsymbol{d}_k\boldsymbol{d}_k^T\dot{C}(\boldsymbol{d}_k)$ is orthogonal to $\boldsymbol{d}_k$. Therefore, the updating rule $d\boldsymbol{D}/dt$ is indeed a tangent vector of $\mathcal{M}$ at the equivalence class $[\boldsymbol{D}]$. Moreover, let the cost function $C(\boldsymbol{D}, t)$ be a Lyapunov function candidate (LFC). Then the

time evolution of the Lyapunov function is

$$\begin{aligned}\frac{dC(\boldsymbol{D},t)}{dt} &= \operatorname{tr}\left(\frac{dC(\boldsymbol{D},t)}{d\,\boldsymbol{D}} \times \frac{d\,\boldsymbol{D}}{dt}\right) \\ &= \sum_{k=1}^{N_s} -\|\dot{C}(\boldsymbol{d}_k)\|^2 + \Sigma_{kk}^{-1}\|\boldsymbol{d}_k^T\dot{C}(\boldsymbol{d})_k\|^2 \\ &\leq -\sum_{k=1}^{N_s}\left(1-\|\boldsymbol{d}_k\|^2\Sigma_{kk}^{-1}\right)\|\dot{C}(\boldsymbol{d})_k\|^2 \approx 0.\end{aligned}$$

The last approximation holds because of Lemma 4.1. Hence, the time evolution of the cost function $C(\boldsymbol{D},t)$ is always negative except at the equilibrium point. This corresponds to the case when $\|\boldsymbol{d}_k^T\dot{C}(\boldsymbol{d}_k)\|^2 = \|\boldsymbol{d}_k^T\|^2\|\dot{C}(\boldsymbol{d}_k)\|^2$, i.e., $\boldsymbol{d}_k$ and $\dot{C}(\boldsymbol{d}_k)$ are linearly dependent. It is straightforward to check that the equality condition leads to $\dot{C}(\boldsymbol{d}_k) = 0$ for $\forall k$ at the equilibrium and thus the gradient $\dot{C}(D) = 0$. This shows that under the proposed updating rule when it is stable, the cost function $C(\boldsymbol{D})$ is decreasing monotonically until $C(\boldsymbol{D})$ reaches its local minimum.

It is however, not always the case that the updating rule is such that $\boldsymbol{d}_k^T\dot{C}(\boldsymbol{d}_k) \leq 0$ (thus the system might not be stable). For example, when the active source signals are Gaussian, it follows from (27) that $\boldsymbol{d}_k^T\dot{C}(\boldsymbol{d}_k) = \boldsymbol{d}_k^T\sum_{i=1}^{N_s}\alpha_i\boldsymbol{A}_k^i\boldsymbol{d}_k$, where $\boldsymbol{A}_k^i = [\mathbb{E}\{\mathcal{O}_m^i \mid \mathcal{Y}_m\}\boldsymbol{I} - \boldsymbol{R}_i^{-1}\,\mathbb{E}\{\tilde{\boldsymbol{\mathcal{T}}}^i{}_m(g) \mid \mathcal{Y}_m\}]\boldsymbol{R}_i^{-1}$, $\alpha_i = 1$ if the k^{th} source is active in state i and zero otherwise. Hence, for $\boldsymbol{d}_k^T\dot{C}(\boldsymbol{d}_k) \leq 0$, the matrices $\alpha_i\boldsymbol{A}_k^i$ must be negative semi-definite, or equivalently,

$$\alpha_i\boldsymbol{R}_i \preccurlyeq \mathbb{E}\{\tilde{\boldsymbol{\mathcal{T}}}^i{}_m(g) \mid \mathcal{Y}_m\}/\,\mathbb{E}\{\mathcal{O}_m^i \mid \mathcal{Y}_m\}.$$

However, this relation is not always true because the recursive estimates $\mathbb{E}\{\tilde{\boldsymbol{\mathcal{T}}}^i{}_m(g) \mid \mathcal{Y}_m\}$ and $\mathbb{E}\{\mathcal{O}_m^i \mid \mathcal{Y}_m\}$ change for every new set of observations.

Using Lemma 4.1, one way to ensure that the learned mixing matrix is bounded and stable is to set the value of $\boldsymbol{d}_k^T\dot{C}(\boldsymbol{d}_k)$ to 0 whenever $\boldsymbol{d}_k^T\dot{C}(\boldsymbol{d}_k) > 0$. With this modified approach, it is easy to check that the updating rule at $[\boldsymbol{D}]$ still lies in the tangent space $\mathcal{T}_{[\boldsymbol{D}]}\mathcal{M}$. Simulations in Section V confirm its usefulness in stabilizing the system. However, we cannot guarantee the local convergence because the the time evolution of the Lyapunov function may be zero before reaching its local minimum. The numerical results suggest that convergence occurs nevertheless.

V. Numerical Results

The focus of this section is to numerically validate the performance of our proposed algorithm and compare it with the KSVD [28] and MOD algorithms [22]. The performance of the mixing matrix estimation measured by the Fubini-Study distance. Precisely, let $\boldsymbol{D}, \boldsymbol{D}' \in \mathbb{R}^{N\times K}$ denote two over-complete mixing matrices, with columns $\|\boldsymbol{d}_k\| = 1$ and $\|\boldsymbol{d}_k'\| = 1$, respectively. Then the Fubini-Study distance between the equivalence classes $[\boldsymbol{d}_k]$ and $[\boldsymbol{d}_k']$ in the projective space P^{N-1} is defined as

$$d_{\text{FB}}([\boldsymbol{d}_k],[\boldsymbol{d}_k']) = \cos^{-1}\left(|\boldsymbol{d}_k^T\boldsymbol{d}_k'|\right). \tag{30}$$

Moreover, given the product manifold structure of $\mathcal{M}$, the distance between $[\boldsymbol{D}]$ and $[\boldsymbol{D}']$ is

$$d_{\text{FB}}([\boldsymbol{D}],[\boldsymbol{D}']) = \sum_{k=1}^{K}\cos^{-1}\left(|\boldsymbol{d}_k^T\boldsymbol{d}_k'|\right). \tag{31}$$

In our simulations, 10 source signals are generated according to a predefined activity profile $\bar{\boldsymbol{s}}_m$ with sparsity $S = 2$, where $m = 1,\cdots,10000$. Each source is assumed to have a Gaussian distribution with zero mean and variance σ_i^2 whenever the source i is active. The number of receivers is set to be $N = 5$ and the mixing matrix $\boldsymbol{D}^{\text{true}} \in \mathbb{R}^{N\times K}$ is generated using random i.i.d. Gaussian samples and verifying that it satisfies the condition that every subset of $2S$ columns of $\boldsymbol{D}^{\text{true}}$ are linearly independent. Without loss of generality, we have normalized the columns of $\boldsymbol{D}^{\text{true}}$ to have unit ℓ_2-norm.

Fig. 2 compares the performance of two batch algorithms (namely, KSVD and MOD) with our proposed recursive algorithm. In particular, it shows the evolution of the estimation error $d_{\text{FB}}([\boldsymbol{D}^{\text{true}}],[\boldsymbol{D}^{\text{est}}])$ in the logarithmic scale, as the number of iterations increases. All the algorithms are initialized with the same initial value $\boldsymbol{D}_0$. For the batch algorithms, at each iteration, the execution alternates between sparse coding of the entire dataset (based on the current estimate of the mixing matrix) and dictionary update while keeping the coded dataset constant. For the proposed recursive algorithm, one iteration is considered as one EM update. Rather than streaming one data vector at a time, to speed up the learning processing, the sampled data are processed recursively in blocks; each block contains 10 data vectors. Since the entire set of data consists of 10000 samples, this leads to a total of 1000 iterations. Observe that the proposed algorithm converges after 400 iterations while the batch algorithms still exhibit large errors.

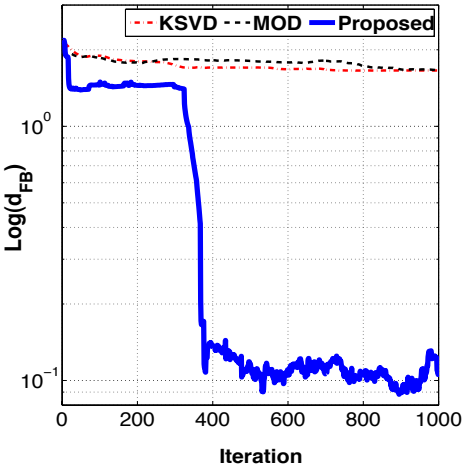

Fig. 2: Performance comparison between KSVD, MOD and the proposed recursive algorithm.

Fig. 3 shows the estimation error for the individual columns of $\boldsymbol{D}^{\text{est}}$, using the Fubini-Study distance defined in (30). We

observe that our proposed algorithm not only converges faster with respect to the distance measure (31), but also with respect to the distance defined on the individual $[\boldsymbol{d}_k]$. For the KSVD and MOD algorithms, the performance is acceptable for certain columns of the mixing matrix (for example, the top 4 plots in Fig. 3) and they both exhibit large errors for the third and fifth plots on the left column of Fig 3.

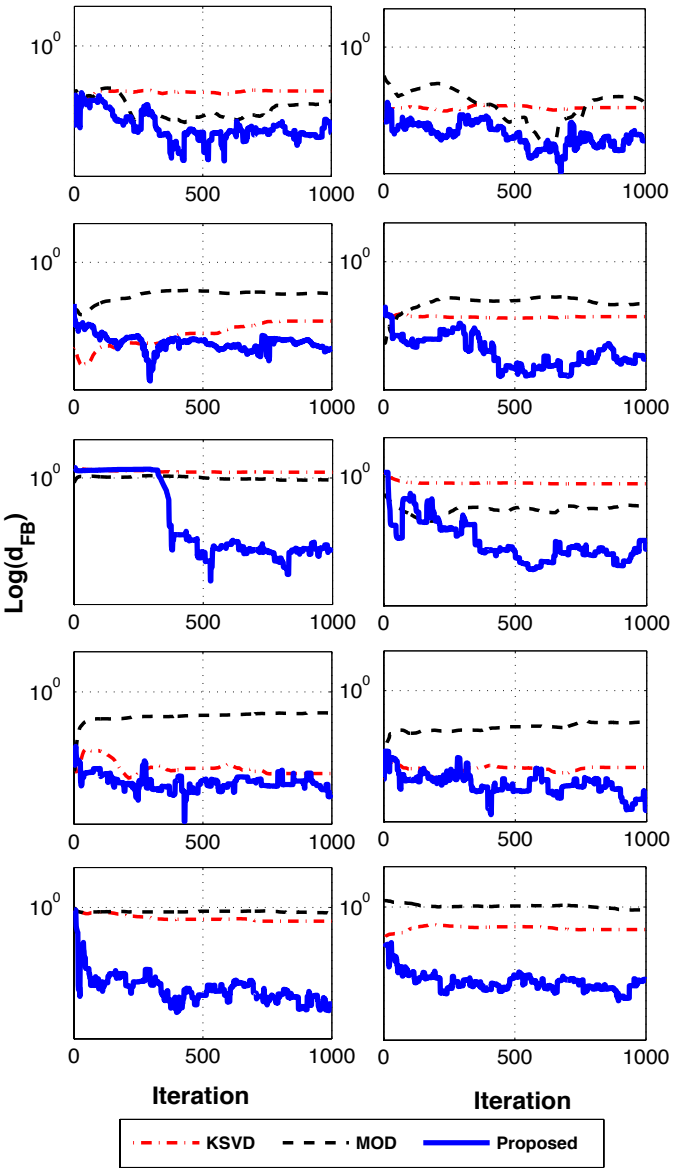

Fig. 3: Performance comparison on columns of $\boldsymbol{D}$ between KSVD, MOD and the proposed recursive algorithm.

Fig. 4 shows the performance of the state estimation using the recursive filter derived in (10) and then choose a state that corresponds to the maximum likelihood of the expected conditional distribution $\max\left(\mathbb{E}\{\bar{\boldsymbol{s}}_m \mid \mathcal{Y}_m\}\right)$. In Fig. 4, the value one indicates an error in the state estimation and zero otherwise. We observe that as the number of iterations increases, the accuracy of the state estimator improves. In particular, for the first 500 iterations, the probability of error in state estimation is 11.2% while the error drops down to 2.6% for the last 500 iterations.

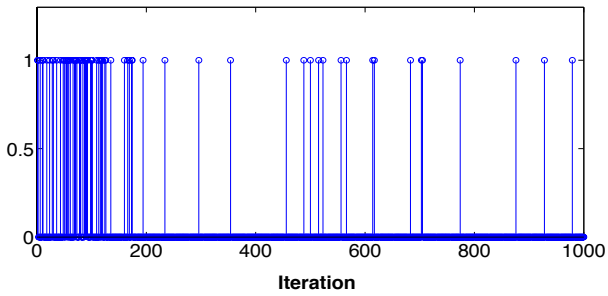

Fig. 4: State Estimation

VI. Conclusion

This paper introduced a hidden Markov model (HMM) based filtering technique for state and parameter estimations. This approach allows us to perform recursive computations of the un-normalized conditional densities of the parameters multiplied by the state $\bar{\boldsymbol{s}}_m$. Hence, the expectation-maximization (EM) algorithm for parameter estimations using this filtering technique is made more efficient than the non-recursive EM approach. Moreover, the algorithm generalizes the literature on under-determined BSS and dictionary learning to include temporal correlation among the sources. To learn the mixing matrix (or dictionary), we proposed a manifold-based method that searches directly for an equivalence class that contains the true mixing matrix. We also analyzed the stability of the system and provided conditions under which the system is stable.

VII. Appendix A

1) **Equation (17):** From the definition (13), we have

$$\mathcal{J}^{ij}_{m+1} = \mathcal{J}^{ij}_{m} + \langle \bar{\boldsymbol{s}}_{m+1}, \boldsymbol{e}_j\rangle\langle \bar{\boldsymbol{s}}_{m}, \boldsymbol{e}_i\rangle.$$

It follows from the same technique as in the derivation of the recursive formulation of the state estimator (10) that

$$\begin{aligned}
&\phi_{m+1}(\mathcal{J}^{ij}_{m+1}\bar{\boldsymbol{s}}_{m+1})\\
&= \tilde{\mathbb{E}}\{\tilde{\Lambda}_{m+1}(\mathcal{J}^{ij}_{m} + \langle \bar{\boldsymbol{s}}_{m+1}, \boldsymbol{e}_j\rangle\langle \bar{\boldsymbol{s}}_{m}, \boldsymbol{e}_i\rangle)\bar{\boldsymbol{s}}_{m+1}|\mathcal{Y}_{m+1}\}\\
&= \sum_{r,s=1}^{N_s} \zeta_s(Y_{m+1})\tilde{\mathbb{E}}\{\tilde{\Lambda}_m\mathcal{J}^{ij}_{m}\langle \boldsymbol{A}^T\bar{\boldsymbol{s}}_{m}, \boldsymbol{e}_s\rangle\langle \bar{\boldsymbol{s}}_{m}, \boldsymbol{e}_r\rangle|\mathcal{Y}_m\}\boldsymbol{e}_s\\
&\quad + \zeta_j(Y_{m+1})\tilde{\mathbb{E}}\{\tilde{\Lambda}_m\langle \boldsymbol{A}^T\bar{\boldsymbol{s}}_{m}, \boldsymbol{e}_j\rangle\langle \bar{\boldsymbol{s}}_{m}, \boldsymbol{e}_i\rangle|\mathcal{Y}_m\}\boldsymbol{e}_j\\
&= \sum_{r,s=1}^{N_s} \zeta_s(Y_{m+1})a_{rs}\tilde{\mathbb{E}}\{\tilde{\Lambda}_m\mathcal{J}^{ij}_{m}\langle \bar{\boldsymbol{s}}_{m}, \boldsymbol{e}_r\rangle|\mathcal{Y}_m\}\boldsymbol{e}_s\\
&\quad + \zeta_j(Y_{m+1})a_{ij}\tilde{\mathbb{E}}\{\tilde{\Lambda}_m\langle \bar{\boldsymbol{s}}_{m}, \boldsymbol{e}_i\rangle|\mathcal{Y}_m\}\boldsymbol{e}_j\\
&= \operatorname{diag}\left(\bar{\zeta}(Y_{m+1})\right)\boldsymbol{A}^T\phi_m(\mathcal{J}^{ij}_{m}\bar{\boldsymbol{s}}_{m})\\
&\quad + \zeta_j(Y_{m+1})a_{ij}\langle \phi_m(\bar{\boldsymbol{s}}_{m}), \boldsymbol{e}_i\rangle\boldsymbol{e}_j.
\end{aligned}$$

2) **Equation (18):** It follows from the definition (14) that $\mathcal{O}^i_{m+1} = \mathcal{O}^i_m + \langle \bar{\boldsymbol{s}}_m, \boldsymbol{e}_i \rangle$. The recursive filter for $\mathcal{O}^i_m \bar{\boldsymbol{s}}_m$ is

$$\begin{aligned}
&\phi_{m+1}(\mathcal{O}^i_{m+1}\bar{\boldsymbol{s}}_{m+1}) \\
&= \tilde{\mathbb{E}}\{\tilde{\Lambda}_{m+1}(\mathcal{O}^i_m + \langle \bar{\boldsymbol{s}}_m, \boldsymbol{e}_i\rangle)\bar{\boldsymbol{s}}_{m+1}|\mathcal{Y}_{m+1}\} \\
&= \sum_{s,r=1}^{N_s} \zeta_s(Y_{m+1})\tilde{\mathbb{E}}\{\tilde{\Lambda}_m \mathcal{O}^i_m \langle \boldsymbol{A}^T\bar{\boldsymbol{s}}_m, \boldsymbol{e}_s\rangle\langle \bar{\boldsymbol{s}}_m, \boldsymbol{e}_r\rangle|\mathcal{Y}_m\}\boldsymbol{e}_s \\
&\quad + \sum_{s=1}^{N_s} \zeta_s(Y_{m+1})\tilde{\mathbb{E}}\{\tilde{\Lambda}_m \langle \boldsymbol{A}^T\bar{\boldsymbol{s}}_m, \boldsymbol{e}_s\rangle\langle \bar{\boldsymbol{s}}_m, \boldsymbol{e}_i\rangle|\mathcal{Y}_m\}\boldsymbol{e}_s \\
&= \sum_{s,r=1}^{N_s} \zeta_s(Y_{m+1})a_{rs}\tilde{\mathbb{E}}\{\tilde{\Lambda}_m \mathcal{O}^i_m \langle \bar{s}_m, e_r\rangle|\mathcal{Y}_m\}e_s \\
&\quad + \sum_{s=1}^{N_s} \zeta_s(Y_{m+1})a_{is}\tilde{\mathbb{E}}\{\tilde{\Lambda}_m \langle \bar{s}_m, e_i\rangle|\mathcal{Y}_m\}e_s \\
&= \mathrm{diag}(\bar{\zeta}(Y_{m+1}))A^T[\phi_m(\mathcal{O}^i_m\bar{s}_m) + \langle \phi_m(\bar{s}_m), e_i\rangle e_i].
\end{aligned}$$

3) **Equation (21):** The derivation of the recursive filter for $\mathcal{T}^i_m(g)\bar{s}_m$ is as follows:

$$\begin{aligned}
&\phi_{m+1}(\mathcal{T}^i_{m+1}(g)\bar{s}_{m+1}) \\
&= \tilde{\mathbb{E}}\{\tilde{\Lambda}_{m+1}(\mathcal{T}^i_m(g) + \langle \bar{s}_{m+1}, e_i\rangle g_{m+1}\bar{s}_{m+1}|\mathcal{Y}_{m+1}\} \\
&= \sum_{s=1}^{N_s} \zeta_s(Y_{m+1})\tilde{\mathbb{E}}\{\tilde{\Lambda}_m \mathcal{T}^i_m(g)e_s\langle \bar{s}_{m+1}, e_s\rangle|\mathcal{Y}_{m+1}\} \\
&\quad + \zeta_i(Y_{m+1})\tilde{\mathbb{E}}\{\tilde{\Lambda}_m\langle \bar{s}_{m+1}, e_i\rangle g_{m+1}e_i|\mathcal{Y}_{m+1}\} \\
&= \sum_{s,r=1}^{N_s} \zeta_s(Y_{m+1})a_{rs}\langle \phi_m(\mathcal{T}^i_m(g)\bar{s}_m), e_r\rangle e_s \\
&\quad + \zeta_i(Y_{m+1})\sum_{r=1}^{N_s} a_{ri}\langle \phi_m(\bar{s}_m), e_r\rangle g_{m+1}e_i \\
&= \mathrm{diag}(\bar{\zeta}(Y_{m+1}))A^T\phi_m(\mathcal{T}^i_m(g)\bar{s}_m) \\
&\quad + \zeta_i(Y_{m+1})g_{m+1}e_i^TA^T\phi_m(\bar{s}_m)e_i.
\end{aligned}$$

ACKNOWLEDGMENT

I would like to express my great appreciation to Prof. Jonathan H. Manton for his valuable suggestions during the development of this research work. His willingness to give his time so generously has been very much appreciated.

REFERENCES

[1] S. Haykin (Ed.), *Unsupervised Adaptive filtering (Volumn I: Blind Source Separation)*, Wiley, New York, 2000.

[2] S. Haykin (Ed.), *Unsupervised Adaptive Filtering (Volume 2 : Blind Deconvolution)*, Wiley, New York, 2000.

[3] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, Wiley, New York, 2001.

[4] S. Amari, S. Douglas, A. Cichocki, and H. Yang, "Multichannel blind deconvolution and equalization using the natural gradient," in *In The First Signal Processing Workshop on Signal Processing Advances in Wireless Communications*, 1997, pp. 101–104.

[5] J.-F. Cardoso and A. Souloumiac, "Blind beamforming for nongaussian signals," in *IEE Proceedings-F*, 1993, vol. 140, pp. 362–370.

[6] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithm and applications," in *Neural Networks*, 2000, vol. 13, pp. 411 – 430.

[7] K. Matsuoka, M. Ohya, and M. Kawamoto, "A neural net for blind separation of nonstationary signals," in *Neural Networks*, 1995, vol. 8, pp. 411–419.

[8] A. Beloucharni and M.G. Amin, "Blind source separation based on time-frequency signal representations," in *IEEE Trans. on Sig. Proc.*, 1998, vol. 46, pp. 2888 – 2897.

[9] L. Parra and C. Spence, "Convolutive blind source separation of nonstationary sources," in *IEEE Trans. on Speech and Audio Proc.*, 2000, pp. 320–327.

[10] D.-T. Pham and J.-F. Cardoso, "Blind separation of instantaneous mixtures of nonstationary sources," in *IEEE Trans. on Sig. Proc.*, 2000, vol. 49, pp. 1837 – 1848.

[11] C. Chang, Z. Ding, S.F. Yau, and F.H.Y. Chan, "A matrix-pencil approach to blind separation of coloered nonstationary signals," in *IEEE Trans. on Sig. Proc.*, 2000, vol. 48, pp. 900–907.

[12] A. Belouchrani, K. Abed-Meraim, J.-F. Cardoso, and E. Moulines, "A blind source separation technique using second-order statistics," in *IEEE Trans. on Sig. Proc.*, 1997, vol. 45, pp. 434–444.

[13] D.-T. Pham and P. Garat, "Blind separation of mixture of independent sources through a quasi-maximum likelihood approach," in *IEEE Trans. on Sig. Proc.*, 1997, vol. 45, pp. 1712–1725.

[14] S. Choi and A. Beloucharni, "Second order nonstationary source separation," in *Journal of VLSI Signal Processing*, 2002, vol. 32, pp. 93–104.

[15] M. Aoki, M. Okamoto, S. Aoki, H. Matsui, T. Sakurai, and Y. Kaneda, "Sound source segregation based on estimating incident angle of each frequency component of input signals acquired by multiple microphones," in *Acoust. Sci. Technol*, 2001, vol. 22, pp. 149–157.

[16] N. Roman, D.-L. Wang, and G.J. Brown, "Speech segregation based on sound localization," in *International Joint Conference on Neural Networks*, 2001, vol. 4, pp. 2861–2866.

[17] O. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," in *IEEE Trans. on Sig. Proc.*, 2004, vol. 52, pp. 1830– 1847.

[18] B.A. Olshausen and D.J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?," in *Vis. Res.*, 1997, vol. 37, pp. 3311–3325.

[19] S. Chen, D.L. Donoho, and Saunders, "Atomic decomposition by basis pursuit," in *SIAM J. Sci. Comput.*, 1998, vol. 20, pp. 33–61.

[20] T.W. Lee, M.S. Lewicki, M. Girolami, and T.J. Sejnowski, "Blind source separation of more sources than mixtures using overcomplete representation," in *IEEE Sig. Proc. Letters*, 1999, vol. 6, pp. 87–90.

[21] M.S. Lewicki and T.J. Sejnowski, "Learning overcomplete representations," in *Neural Comput.*, 2000, vol. 12, pp. 337–365.

[22] K. Engan, S.O. Aase, and J.H. Husøy, "Multi-frame compression: Theory and design," in *Signal Processing*, 2000, vol. 80, pp. 2121–2140.

[23] M. Zibulevsky and B.A. Pearlmutter, "Blind source separation by sparse decomposition," in *Neural Comput.*, 2001, vol. 13, pp. 863–882.

[24] M. Girolami, "A variational method for learning sparse and overcomplete representations," in *Neural Comput.*, 2001, vol. 13, pp. 2517–2532.

[25] David L Donoho and Michael Elad, "Optimally sparse representation in general (non-orthogonal) dictionaries via $l1$ minimization," *Proc Natl Acad Sci U S A*, vol. 100, no. 5, pp. 2197–202, 2003.

[26] Y. Li, A. Cichocki, and S. Amari, "Sparse component analysis for blind source separation with less sensors than sources," in *Proc. 4th Int. Symp. Independent Component Analaysis Blind Signal Separation*, 2003, pp. 89–94.

[27] K. Kreutz-Delgado, J.F. Murray, B.D. Rao, K. Engan, T.W. Lee, and T.J. Sejnowski, "Dictionary learning algorithms for sparse representation," in *Neural Computation*, 2003, vol. 15, pp. 349–396.

[28] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," in *IEEE Trans. on Sig. Proc.*, 2006, vol. 54.

[29] I. Tošić and P. Frossard, "Dictionary learning," in *IEEE Sig. Proc. Mag.*, 2011.

[30] R.J. Elliott, Aggoun L., and J.B. Moore, *Hidden Markov Models: Estimation and Control*, Springer-Verlag, 1994.

[31] E.J. Candes and T. Tao, "Decoding by linear programming," *Information Theory, IEEE Transactions on*, vol. 51, no. 12, pp. 4203 – 4215, dec. 2005.

[32] Emmanuel Candes and Terence Tao, "The dantzig selector: Statistical estimation when p is much larger than n," *The Annals of Statistics*, vol. 35, no. 6, pp. 2313–2351, Dec. 2007.

[33] C.F.J. Wu, "On the convergence properties of the em algorithmg," *Annals of statistics*, vol. 11, no. 1, pp. 95 – 103, Mar. 1985.