Algorithm for Independent Vector Extraction Based on Semi-Time-Variant Mixing Model

Zbyněk Koldovský*, Václav Kautský†, Tomáš Kounovský* and Jaroslav Čmejla*
*Acoustic Signal Analysis and Processing Group, Faculty of Mechatronics, Informatics, and Interdisciplinary Studies,
Technical University of Liberec, Czech Republic.
†Faculty of Nuclear Sciences and Physical Engineering, Czech Technical University in Prague, Czech Republic

***Abstract*— A new algorithm for dynamic independent vector extraction is proposed. It is based on the mixing model where mixing parameters related to the source-of-interest (SOI) are time-variant while the separating parameters are time-invariant. A contrast function based on the quasi-likelihood approach is optimized using the Newton-Raphson approach. The update is computed without imposing the orthogonal constraint, and the orthogonality is enforced afterward. This yields an algorithm that is significantly faster than gradient-based algorithms while different from fixed-point methods, which are even faster. We show advantageous properties of the proposed algorithm compared to the fixed-point methods in an on-line processing regime where stable convergence to the SOI is the important issue. The effectiveness of the method is demonstrated in a speech extraction experiment with a dense microphone array.**

***Index Terms*— Blind Source Separation, Blind Source Extraction, Independent Component Analysis, Independent Vector Analysis, Newton-Raphson Algorithm**

I. Introduction

Independent Component and Vector Analysis (ICA/IVA) are popular methods for Blind Source Separation (BSS). The goal is to separate K linear instantaneous mixtures of original signals into *independent* components that correspond to these signals up to indeterminable scales and order; $K = 1$ in case of ICA. In IVA, $K > 1$ mixtures are separated jointly based on a joint probabilistic source model where dependencies between the components of the same source are taken into account; these components form so-called vector component [1], [2]. In Independent Component/Vector Extraction (ICE/IVE), the goal is to extract only a particular source-of-interest (SOI), i.e., to separate it from the other signals [3]–[6].

Conventional methods assume the static linear mixing model given by

$$\mathbf{x}_k = \mathbf{A}_k \mathbf{s}_k, \qquad k = 1, \dots, K, \tag{1}$$

where $\mathbf{x}_k$ is a $d\times 1$ vector of random variables (RVs) representing the mixed signals; $\mathbf{A}_k$ is a $d\times d$ non-singular time-invariant mixing matrix; and $\mathbf{s}_k$ is a $d \times 1$ vector of independent RVs representing the original signals; k is the mixture index. The determined case, i.e. when $\mathbf{A}_k$ are square, is often preferred because then the goal can be formulated as to estimate square de-mixing matrices $\mathbf{W}_k$ such that $\mathbf{y}_k = \mathbf{W}_k\mathbf{x}_k$ correspond to $\mathbf{s}_k$ up to the indeterminable scales and order. $\mathbf{W}_k$ are sought such that the signals $\mathbf{y}_k$ are as independent as possible [7].

In many real-world situations, the sources are moving, which can be modeled by assuming that $\mathbf{A}_k$, hence also $\mathbf{W}_k$, are time-varying. Typically, adaptive methods are derived from off-line algorithms in a sequential manner so that $\mathbf{A}_k$ or $\mathbf{W}_k$ are progressively updated; see, e.g., [8]–[10]. Inherently, these methods apply the static mixing model (1) on short intervals of data. Recently, a novel dynamic mixing model for ICE/IVE has been introduced referred to as CSV (Constant Separating Vector) [11]–[13]. In CSV, the mixing parameters related to the SOI are time-variant while the separating parameters are time-invariant. The idea behind is that an invariant separating vector extracts the SOI from the entire area of its motion (during the interval of data), which is delimited by the time-varying mixing parameters[1]. Advantageous theoretical properties and practical usefulness of CSV for off-line as well as on-line blind extraction of moving sources have been recently reported in [11], [12], [17].

In this paper, a new algorithm for IVE based on the CSV model, hereafter referred to as QuickIVE[2], is proposed. It seeks optimum points of a quasi-likelihood function using the Newton-Raphson optimization. The update is computed without imposing the orthogonal constraint as compared to the FastDIVA algorithm in [13], while the orthogonality is enforced afterwards. In experiments, the algorithm is considerably faster than gradient-based algorithms, comparable or faster than auxiliary function-based methods [5], [12], [18] while slower than fixed-point methods such as FastICA [19], FastIVA [20] and FastDIVA [13]. Nevertheless, QuickIVE appears to be more useful in on-line processing deployments where fixed-point methods show high sensitivity to a varying non-Gaussianity of the SOI, which can cause their sudden divergence (in particular, convergence to a different source than the SOI).

The paper is organized as follows. The new algorithm is derived in Section II and is compared with the fixed-point algorithms in Section III. Section IV is focused on numerical

This work was supported by The Czech Science Foundation through Project No. 20-17720S, and by the department of the Navy, Office of Naval Research Global, through Project No. N62909-19-1-2105

[1] Similar idea based on beam-widening appears in the literature on robust beamforming; see, e.g., [14]–[16].

[2] Available at `https://asap.ite.tul.cz/downloads/ice/`

and experimental evaluations and comparisons with the other algorithms. Section V concludes this paper.

Notations: Plain, bold, and bold capital letters denote, respectively, scalars, vectors, and matrices. Upper index $\cdot^T$, $\cdot^H$, or $\cdot^*$ denotes, respectively, transposition, conjugate transpose, or complex conjugate. The Matlab convention for matrix/vector concatenation will be used, e.g., $[1;\, \mathbf{g}] = [1,\, \mathbf{g}^T]^T$. $\mathrm{E}[\cdot]$ stands for the expectation operator, and $\hat{\mathrm{E}}[\cdot]$ is the sample-based average taken over all available samples of the argument; $\hat{a}$ denotes an estimate of a; $\{\mathbf{w}_k\}_k$ is a short notation of $\mathbf{w}_1, \ldots, \mathbf{w}_K$; $\langle a_t \rangle_t$ is a short notation of $\frac{1}{T}\sum_{t=1}^{T} a_t$.

II. Proposed Algorithm

A. Problem formulation

We consider complex-valued signals and parameters, however, the conclusions are valid also for the real-valued case. Let N samples of signals be available in K mixtures, and let the samples be divided into $T \geq 1$ time-intervals called blocks, for simplicity, of the same length N_b; hence $N = T \cdot N_b$. CSV is a semi-time-varying model described by

$$\mathbf{x}_{k,t} = \mathbf{A}_{k,t}\mathbf{s}_{k,t}, \tag{2}$$

where $k = 1, \ldots, K$ is the mixture index; $t = 1, \ldots, T$ is the block index; $\mathbf{A}_{k,t}$ is the square non-singular mixing matrix; $\mathbf{s}_{k,t}$ and $\mathbf{x}_{k,t}$ are RVs representing samples of the original and observed signals, respectively; $\mathbf{s}_{k,t}$ is divided as $\mathbf{s}_{k,t} = [s_{k,t}; \mathbf{z}_{k,t}]$ where $s_{k,t}$ stands for the SOI while $\mathbf{z}_{k,t}$ represents *background*, which is an unspecified mixture of the other signals. The mixing matrices $\mathbf{A}_{k,t}$ are parameterized as

$$\mathbf{A}_{k,t} = \begin{pmatrix} \mathbf{a}_{k,t} & \mathbf{Q}_{k,t} \end{pmatrix} = \begin{pmatrix} \gamma_{k,t} & \mathbf{h}_k^H \\ \mathbf{g}_{k,t} & \frac{1}{\gamma_{k,t}}(\mathbf{g}_{k,t}\mathbf{h}_k^H - \mathbf{I}_{d-1}) \end{pmatrix}, \tag{3}$$

where $\mathbf{a}_{k,t} = [\gamma_{k,t}; \mathbf{g}_{k,t}]$ is the *mixing vector* and $\mathbf{w}_k = [\beta_k; \mathbf{h}_k]$ is the *separating* vector corresponding to the SOI in the kth mixture and tth block. Note that $\mathbf{a}_{k,t}$ depends on t while $\mathbf{w}_k$ is constant over t. For $T = 1$, CSV coincides with the time-invariant (static) IVE model [4].

The vectors $\mathbf{a}_{k,t}$ and $\mathbf{w}_k$ are free vector parameters linked through $\mathbf{a}_{k,t}^H\mathbf{w}_k = 1$ for all t and k, which is referred to as the *distortionless constraint*. This constraint, together with the structure of $\mathbf{A}_{k,t}$ given by (3), ensures that $\mathbf{A}_{k,t}^{-1}$ is equal to

$$\mathbf{W}_{k,t} = \begin{pmatrix} \mathbf{w}_k^H \\ \mathbf{B}_{k,t} \end{pmatrix} = \begin{pmatrix} \beta_k^* & \mathbf{h}_k^H \\ \mathbf{g}_{k,t} & -\gamma_{k,t}\mathbf{I}_{d-1} \end{pmatrix}, \tag{4}$$

where $\mathbf{I}_d$ is the $d \times d$ identity matrix. It means that $\mathbf{W}_{k,t}$ separates $\mathbf{x}_{k,t}$ into $s_{k,t} = \mathbf{w}_k^H\mathbf{x}_{k,t}$ and $\mathbf{z}_{k,t} = \mathbf{B}_{k,t}\mathbf{x}_{k,t}$; see also Section II in [13].

B. Contrast function

Let $p(\mathbf{s}_t)$ and $p_{\mathbf{z}_{k,t}}(\mathbf{z}_{k,t})$ denote the joint pdf of the SOI vector component $\mathbf{s}_t = [s_{1,t}, \ldots, s_{K,t}]^T$ and of $\mathbf{z}_{k,t}$, respectively. For simplicity, $p(\cdot)$ will be assumed constant over t, and $\mathbf{z}_{1,t}, \ldots, \mathbf{z}_{K,t}$ are assumed to be independent.

Since $p(\cdot)$ and $p_{\mathbf{z}_{k,t}}(\cdot)$ are not known in the blind setting, they have to be replaced by suitable model densities. In [13] (Section III.B), it justified that $p(\cdot)$ can be replaced by

$$p(\mathbf{s}_t) \approx f\left(\left\{\frac{s_{k,t}}{\hat{\sigma}_{k,t}}\right\}_k\right)\left(\prod_{k=1}^{K}\hat{\sigma}_{k,t}\right)^{-2}, \tag{5}$$

where $f(\cdot)$ should be a suitable normalized non-Gaussian pdf, and $\hat{\sigma}_{k,t}^2$ is the sample-based variance of the estimate of $s_{k,t}$; it holds that $\hat{\sigma}_{k,t} = \sqrt{\mathbf{w}_k^H\widehat{\mathbf{C}}^{k,t}\mathbf{w}_k}$ where $\widehat{\mathbf{C}}^{k,t} = \hat{\mathrm{E}}[\mathbf{x}_{k,t}\mathbf{x}_{k,t}^H]$ is the sample-based covariance matrix of $\mathbf{x}_{k,t}$.

The unknown $p_{\mathbf{z}_{k,t}}(\mathbf{z}_{k,t})$ can be replaced by the zero mean circular Gaussian pdf $\mathcal{CN}(0, \mathbf{C}_{\mathbf{z}}^{k,t})$, where $\mathbf{C}_{\mathbf{z}}^{k,t} = \mathrm{E}[\mathbf{z}_{k,t}\mathbf{z}_{k,t}^H]$ [4]. $\mathbf{C}_{\mathbf{z}}^{k,t}$ is an unknown nuisance parameter. The model densities are put into the likelihood function which is derived based on the fact that $\mathbf{s}_t$ and $\mathbf{z}_{k,t}$ are independent, by which the contrast function is obtained in the form

$$\mathcal{C}\left(\{\mathbf{w}_k, \mathbf{a}_{k,t}\}_{k,t}\right) = \Bigg\langle \hat{\mathrm{E}}\left[\log f\left(\left\{\frac{\hat{s}_{k,t}}{\hat{\sigma}_{k,t}}\right\}_k\right)\right] - \sum_{k=1}^{K}\log\hat{\sigma}_{k,t}^2 \\ - \sum_{k=1}^{K}\hat{\mathrm{E}}\left[\hat{\mathbf{z}}_{k,t}^H(\mathbf{C}_{\mathbf{z}}^{k,t})^{-1}\hat{\mathbf{z}}_{k,t}\right] + (d-2)\sum_{k=1}^{K}\log|\gamma_{k,t}|^2 \Bigg\rangle_t + C, \tag{6}$$

where $\hat{\cdot}$ denotes the estimate of the argument; C is a constant independent of the parameter vectors.

The estimates of $\mathbf{a}_{k,t}$ and $\mathbf{w}_k$ are sought through finding points maximizing (6). However, since there are many spurious solutions, it is often necessary to impose the orthogonal constraint (OGC). It enforces that $\hat{\mathrm{E}}[\hat{s}_{k,t}\hat{\mathbf{z}}_{k,t}] = 0$ for all k and t. The OGC involves the distortionless constraint and makes $\mathbf{a}_{k,t}$ fully dependent on $\mathbf{w}_k$ through

$$\mathbf{a}_{k,t} = \frac{\widehat{\mathbf{C}}^{k,t}\mathbf{w}_k}{\mathbf{w}_k^H\widehat{\mathbf{C}}^{k,t}\mathbf{w}_k}, \tag{7}$$

or vice versa [4]. For more details, see Section III in [13].

C. QuickIVE

The key approach in the algorithm development is that the gradient and Hessian matrices of (6) w.r.t. $\mathbf{w}_k$ are computed without imposing the OGC. That is, $\mathbf{a}_{k,t}$ are treated as independent variables; only the link $\gamma^{k,t} = (1 - (\mathbf{h}^k)^H\mathbf{g}^{k,t})/(\beta^k)^*$ is considered due to the distortionless constraint. In the following, detailed computations are skipped due to the limited space; they will appear in a forthcoming publication.

After some computations, we obtain

$$\frac{\partial\mathcal{C}}{\partial\mathbf{w}_k^H} = \Big\langle \Re\{\hat{\nu}_{k,t}\}\mathbf{a}_{k,t} - \hat{\mathrm{E}}\left[\phi_k\left(\{\bar{s}_{k,t}\}_k\right)\bar{\mathbf{x}}_{k,t}\right] \Big\rangle_t, \tag{8}$$

where $\bar{s}_{k,t} = \hat{s}_{k,t}/\hat{\sigma}_{k,t}$, $\bar{\mathbf{x}}_{k,t} = \mathbf{x}_{k,t}/\hat{\sigma}_{k,t}$, $\phi_k(\{s_k\}_k) = -\frac{\partial}{\partial s_k}\log f(\{s_k\}_k)$ is the score function; s_k stands for the kth complex argument of $f(\cdot)$, and

$$\hat{\nu}_{k,t} = \hat{\mathrm{E}}[\phi_k(\{\bar{s}_{k,t}\}_k)\bar{s}_{k,t}]. \tag{9}$$

It is necessary that when $N_b \to +\infty$ and when $\mathbf{a}_{k,t}$ and $\mathbf{w}_k$ correspond to their true values, the gradient is zero. This is satisfied when $\hat{\nu}_{k,t} = 1$. We therefore consider a substitution $\phi_k \leftarrow \hat{\nu}_{k,t}^{-1}\phi_k$, which can be justified by a more specific choice of the model density $f(\cdot)$ (in fact, $f(\cdot)$ need not be specified as it is not used explicitly). After this change, the gradient reads

$$\nabla_k = \left\langle \mathbf{a}_{k,t} - \hat{\nu}_{k,t}^{-1}\hat{\mathrm{E}}\left[\phi_k\left(\{\bar{s}_{k,t}\}_k\right)\bar{\mathbf{x}}_{k,t}\right]\right\rangle_t. \tag{10}$$

The Hessian matrices $\frac{\partial \nabla_k^H}{\partial \mathbf{w}_k}$ and $\frac{\partial \nabla_k^T}{\partial \mathbf{w}_k}$ are computed under these additional rules: (a) $\hat{\nu}_{k,t}$ in (10) is treated as a constant, and (b) the analytic forms of the Hessian matrices when $N_b \rightarrow +\infty$ and $\mathbf{a}_{k,t}$ and $\mathbf{w}_k$ correspond to their true values are considered. Then,

$$\mathbf{H}_2^k = \frac{\partial \nabla_k^T}{\partial \mathbf{w}_k} = \left\langle -\nu_{k,t}^{-1}\frac{\partial}{\partial \mathbf{w}_k}\mathrm{E}\left[\phi_k\left(\{\bar{s}_{k,t}\}_k\right)\bar{\mathbf{x}}_{k,t}^T\right]\right\rangle_t. \tag{11}$$

Using Appendix A in [13], it follows that

$$\frac{\partial}{\partial \mathbf{w}_k}\mathrm{E}\left[\phi_k\left(\{\bar{s}_{k,t}\}_k\right)\bar{\mathbf{x}}_{k,t}^T\right] = \frac{\rho_{k,t}}{\sigma_{k,t}^2}(\mathbf{C}^{k,t})^* + \zeta_{k,t}\mathbf{a}_{k,t}^*\mathbf{a}_{k,t}^T, \tag{12}$$

where

$$\rho_{k,t} = \mathrm{E}\left[\frac{\partial \phi_k(\{\bar{s}_{k,t}\}_k)}{\partial s_k^*}\right]. \tag{13}$$

Similarly, it can be shown that the second Hessian matrix $\mathbf{H}_1^k = \frac{\partial \nabla_k^H}{\partial \mathbf{w}_k} = \left\langle \eta_{k,t}\mathbf{a}_{k,t}^*\mathbf{a}_{k,t}^T\right\rangle_t$; $\zeta_{k,t}$ and $\eta_{k,t}$ are constants depending on the distribution of $s_{k,t}$ and on ϕ_k.

We now simplify the algorithm derivation through ignoring the rank-1 terms in $\mathbf{H}_1^k$ and in (12). Hence, we take $\mathbf{H}_1^k = 0$ and $\mathbf{H}_2^k = -\left\langle \frac{\rho_{k,t}}{\nu_{k,t}\sigma_{k,t}^2}(\mathbf{C}^{k,t})^*\right\rangle_t$. By [21], the Newton-Raphson direction is then given by $-((\mathbf{H}_2^k)^*)^{-1}\nabla_k$. Using the sample-based estimates instead of the corresponding expectation values, the update rule for the separating vector $\mathbf{w}_k$ therefore reads

$$\mathbf{w}_k \leftarrow \mathbf{w}_k + \left\langle \frac{\hat{\rho}_{k,t}}{\hat{\nu}_{k,t}^*}\frac{\widehat{\mathbf{C}}^{k,t}}{\hat{\sigma}_{k,t}^2}\right\rangle_t^{-1}\nabla_k. \tag{14}$$

To summarize, the complete update rule in the proposed algorithm proceeds in the following steps. Starting from the given values of $\mathbf{w}^k$, for every $k = 1, \ldots, K$,

1) update $\mathbf{w}^k$ according to (14),
2) to fix the scaling ambiguity, normalize the separating vector as $\mathbf{w}_k \leftarrow \mathbf{w}_k / \sqrt{T\left\langle \sigma_{k,t}^2\right\rangle_t}$,
3) and update the mixing vectors using (7),

until convergence. The stopping criterion as in [13] is used.

III. QuickIVE versus Fixed-Point Algorithms

FastDIVA (Fast Dynamic IVA) is a fixed-point algorithm that has been recently derived in [13]. For $T = 1$ and $K = 1$, FastDIVA coincides with the famous FastICA [19], in the real-valued case, and is closely related to the complex-valued FastICA [22] and FastIVA [20] in the complex-valued case and when $K > 1$, respectively. Its one-unit variant, which performs the extraction of one SOI, updates the separating vectors according to

$$\mathbf{w}_k \leftarrow \mathbf{w}_k - \left\langle \left(\frac{\hat{\nu}_{k,t} - \hat{\rho}_{k,t}}{\hat{\nu}_{k,t}}\right)^* \frac{\widehat{\mathbf{C}}_{k,t}}{\hat{\sigma}_{k,t}^2}\right\rangle_t^{-1}\nabla^k. \tag{15}$$

QuickIVE and One-unit FastDIVA are optimizing the same contrast function given by (6). Provided that both algorithms converge, they must therefore achieve the same asymptotic accuracy in terms of the mean interference-to-signal ratio, as derived in Section IV in [13] and confirmed by the experiments therein. The algorithms differ in the speed of convergence because of the different optimization approaches: In FastDIVA, the gradient and Hessians of (6) are computed when the OGC (7) is imposed while, in QuickIVE, only the weaker distortionless constraint is considered[3]. It means that the mixing and separating vectors are updated simultaneously in FastDIVA, which might explain its faster convergence compared to QuickIVE.

For further comparison, let us consider the real-valued case when $T = 1$ and $K = 1$, which corresponds to the fundamental problem of extraction of one independent component (ICE). Here, the indices k and t and the conjugate operator can be omitted, and, owing to the normalization of the output signal scale, $\hat{\sigma}^2 = 1$. Then, the QuickIVE update rule (14) simplifies to

$$\mathbf{w} \leftarrow \mathbf{w} + \frac{\hat{\nu}}{\hat{\rho}}\widehat{\mathbf{C}}^{-1}\nabla, \tag{16}$$

and the FastDIVA (FastICA) update (15) simplifies to

$$\mathbf{w} \leftarrow \mathbf{w} + \frac{\hat{\nu}}{\hat{\rho} - \hat{\nu}}\widehat{\mathbf{C}}^{-1}\nabla. \tag{17}$$

An important difference of FastDIVA compared to QuickIVE (and also compared to other algorithms) stems from the numerator value $\hat{\rho}-\hat{\nu}$. It can be shown that, for $N \rightarrow +\infty$, this value is zero when the SOI is Gaussian [23], [24]. It means that, if the distribution of the SOI becomes Gaussian for a moment (e.g. in an on-line processing regime), FastDIVA tends to change the direction of the update abruptly, most likely, towards a different independent source.

IV. Experiments

A. Convergence speed

To compare algorithms in terms of the speed of convergence, $K = 6$ complex-valued static ($T = 1$) mixtures of $d = 6$ signals of length $N = 1000$ were generated in 100 trials; the compared algorithms are One-unit FastDIVA[4] [13] (FastDIVA), QuickIVE, OverIVA [5] (equivalent to AuxIVE from [12]), and BOGIVE_w [12], [25]. The algorithms were initialized in a vicinity of the desired SOIs and applied to extract them. The interference-to-signal ratio (ISR) has been evaluated in each iteration. In each trial, all signals were

[3] Interestingly, both computations yield the same gradient but different Hessians; cf. (10) in the present paper and (43) in [13].

[4] Remind that the results by FastDIVA translate to FastICA ($K = T = 1$) and FastIVA ($T = 1$).

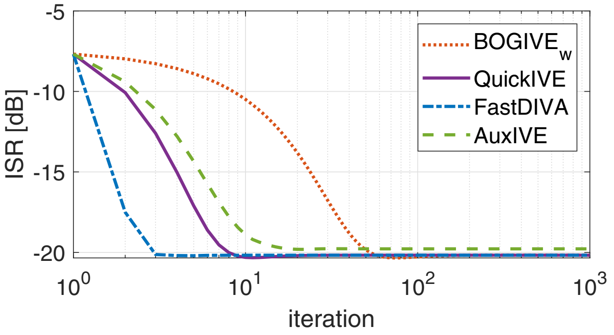

Fig. 1. Averaged Interference-to-Signal Ratio as a function of the iteration index. The compared algorithms are initialized at the same point in the vicinity of the SOIs. The average is taken over 100 independent trials.

generated from the Laplacean distribution, and the SOIs from different mixtures were mixed by a random unitary matrix to make them dependent (but uncorrelated). Then, the signals were mixed by random mixing matrices. In this comparison, the static mixtures are considered, i.e., $T = 1$.

Fig. 1 shows the ISR as a function of the iteration number; the ISR is averaged over trials and all SOIs in the $K = 6$ mixtures. The results show typical speed of convergence: FastDIVA converges in very few iterations; QuickIVE and OverIVA converge in about 10 iterations; BOGIVE_w exhibits the slowest convergence.

It is worth noting that FastDIVA, QuickIVE, and BOGIVE_w finally achieve the same ISR. This is because they use same nonlinearity $\phi_k(\{\bar{s}_{k,t}\}_k) = \frac{\bar{s}^*_{k,t}}{1+\sum_{k=1}^{K}|\bar{s}_{k,t}|^2}$. AuxIVE is using a different nonlinearity, which meets the additional conditions for the auxiliary-function based optimization [18]. Therefore, the final ISR achieved by AuxIVE is different from that of the other algorithms.

B. On-line source extraction

In this experiment, an on-line blind source extraction problem is studied where the SOI is being extracted from a continuously varying real-valued mixture ($K = 1$). In a trial, the mixture of dimension $d = 5$ and length $N = 128000$ is generated: The SOI is Laplacean up to an interval of length 16000 samples beginning at $N/2$, where it is Gaussian. The other four signals are Laplacean. The mixing matrix at sample n is given by $\mathbf{A}_n = [\mathbf{a}_n\, \mathbf{Q}_{\lceil 8*n/N \rceil}]$, where $\lceil x \rceil$ denotes the least integer greater than or equal to x,

$$\mathbf{a}_n = \mathbf{a}_{n-1} + \mathbf{e}_n, \tag{18}$$

$$\mathbf{Q}_i = \mathbf{Q}_{i-1} + \mathbf{E}_i, \tag{19}$$

where the elements of $\mathbf{e}_n$ and $\mathbf{E}_n$ are drawn from $\mathcal{N}(0, \sqrt{0.005})$ and $\mathcal{N}(0, \sqrt{0.01})$; the initial $\mathbf{Q}_1$ and $\mathbf{a}_1$ are generated from $\mathcal{N}(0, 1)$. Hence, the SOI performs a continuous Brownian motion while the background signals perform a step movement after every $N/8$ samples.

The compared algorithms are deployed sequentially batch-by-batch performing one iteration per batch; the batch size is 16000; the shift is 1000. The CSV model is used with $T = 16$ (i.e. the batches are divided into blocks whose length corresponds with the shift length). The methods are initialized by a slightly perturbed true separating vector at time $n = 1$. ISR achieved by the algorithms is evaluated per batch. Its 20%-trimmed mean value computed over 100 trials as a function of n is shown in Fig. 2 together with the number of converged trials: a trial is marked as converged it when ISR is smaller than -3 dB.

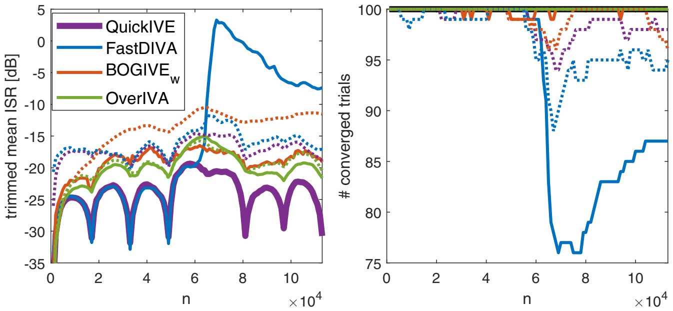

Fig. 2. Trimmed mean of ISR and the number of converged trials as functions of $n = 1, \dots, N$; $N = 128000$; the SOI is Gaussian for $n = N/2, \dots, N/2+15999$. Solid and dotted lines correspond to algorithms' variants, respectively, exploiting CSV with $T = 16$ and the static variants with $T = 1$. For $T = 16$, OverIVA is represented by Block AuxIVE [12]. The significant local minima of ISR (e.g. by QuickIVE) when n is an integer multiple of 16000 correspond to situations when the batch coincides with an interval in which $\mathbf{Q}_i$ is constant (i.e. the background sources are static).

The results reveal the sensitivity of FastDIVA to the Gaussianity of the SOI as its averaged ISR steeply increases at $n = N/2$ and the number of converged trials drops down. In fact, FastDIVA tends to extract a different Laplacean source since $n = N/2$ and keeps extracting it in most subsequent batches. The other methods show lower sensitivity to the Gaussianity of the SOI.

The results also demonstrate the benefit of applying CSV in on-line extraction. Significantly lower ISRs are achieved by the variants of the algorithms where the CSV mixing model is exploited with $T = 16$ compared to when $T = 1$.

C. On-line speech extraction in real environment

We have realized a similar experiment with two speakers in an open-space 12x8x2.6 m room with the reverberation time $\mathrm{T}_{60} \approx 500$ms. A dense microphone array consisting of 48 MEMS microphones arranged in an 8×6 vertical planar grid was used to record the speakers simulated by a loudspeaker positioned at ~ 1 m away from the microphone array. The SOI, female speaker, was positioned at $-30°$ from the front of the array. The female speech, which is 40 s long, involved white Gaussian noise from 15 s through 20 s. This simulated an interval where the SOI is Gaussian as in the previous Section IV-B. An interfering male speaker (IR) was placed at $60°$; its 40 s long utterance was unaltered. Both sources were recorded separately at 48 kHz sampling frequency, downsampled to 16 kHz and mixed with initial SIR $= 5$ dB. The processing was performed in the short-time Fourier transform domain with a frame length of 512 samples and 128 samples overlap.

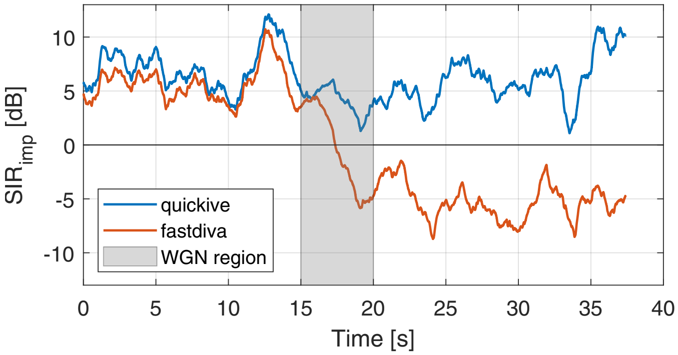

Fig. 3. SIR improvement by QuickIVE and FastDIVA when the speech of the SOI is replaced by white Gaussian noise (WGN) for 5 seconds.

The SOI was extracted by QuickIVE and FastDIVA. Both algorithms used the same settings: 600 frames per batch (corresponding to 4.8 seconds), 300 frames shift, and 300 frames for each CSV block ($T = 2$); each algorithm performed 3 iterations per batch. The algorithms were initialized by an oracle MPDR beamformer. The results in terms of output sample-wise SIR improvement smoothed by a moving-average filter of a 2-second length are shown in Fig. 3.

As expected, both algorithms do not adapt correctly when the SOI is Gaussian. FastDIVA exhibits a fast convergence speed, which causes that the method quickly converges to the IR when the SOI is Gaussian. Furthermore, despite the SOI being dominant in the mixture, FastDIVA keeps converging to the IR for the rest of the recording. QuickIVE converges slightly slower and does not exhibit as high of a sensitivity to the SOI Gaussianity as FastDIVA. This causes QuickIVE to continue the extraction of the SOI after the testing interval.

V. Conclusions

We have proposed the QuickIVE algorithm for independent vector extraction allowing for the CSV mixing model, which is flexible for on-line processing of dynamic mixtures. QuickIVE provides a useful alternative to the fixed-point and auxiliary-function based methods. While its convergence speed is comparable or might be slightly slower, it shows robustness to a short-term Guassianity of the SOI. Compared to OverIVA, AuxIVA or Block AuxIVE, there are no additional requirements on the nonlinearity such as finding an appropriate majorizing function. In the experiment with a dense microphone array, the method shows an efficient performance and a reasonable computational burden because, compared to ICA/IVA methods, it extracts only the desired SOI.

References

[1] P. Comon and C. Jutten, *Handbook of Blind Source Separation: Independent Component Analysis and Applications*. Independent Component Analysis and Applications Series, Elsevier Science, 2010.

[2] T. Kim, H. T. Attias, S.-Y. Lee, and T.-W. Lee, "Blind source separation exploiting higher-order frequency dependencies," *IEEE Transactions on Audio, Speech, and Language Processing*, pp. 70–79, Jan. 2007.

[3] P. J. Huber, "Projection pursuit," *Ann. Statist.*, vol. 13, pp. 435–475, June 1985.

[4] Z. Koldovský and P. Tichavský, "Gradient algorithms for complex non-gaussian independent component/vector extraction, question of convergence," *IEEE Transactions on Signal Processing*, vol. 67, pp. 1050–1064, Feb 2019.

[5] R. Scheibler and N. Ono, "Independent vector analysis with more microphones than sources," in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 185–189, 2019.

[6] R. Ikeshita and T. Nakatani, "Independent vector extraction for joint blind source separation and dereverberation," 2021, arXiv: 2102.04696.

[7] P. Comon, "Independent component analysis, a new concept?," *Signal Processing*, vol. 36, pp. 287–314, 1994.

[8] T. Taniguchi, N. Ono, A. Kawamura, and S. Sagayama, "An auxiliary-function approach to online independent vector analysis for real-time blind source separation," in *2014 4th Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*, pp. 107–111, May 2014.

[9] A. H. Khan, M. Taseska, and E. A. P. Habets, *A Geometrically Constrained Independent Vector Analysis Algorithm for Online Source Extraction*, pp. 396–403. Cham: Springer International Publishing, 2015.

[10] S. H. Hsu, T. R. Mullen, T. P. Jung, and G. Cauwenberghs, "Real-time adaptive eeg source separation using online recursive independent component analysis," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 3, pp. 309–319, 2016.

[11] V. Kautský, Z. Koldovský, P. Tichavský, and V. Zarzoso, "Cramér–Rao bounds for complex-valued independent component extraction: Determined and piecewise determined mixing models," *IEEE Transactions on Signal Processing*, vol. 68, pp. 5230–5243, 2020.

[12] J. Janský, Z. Koldovský, J. Málek, T. Kounovský, and J. Čmejla, "Auxiliary function-based algorithm for blind extraction of a moving speaker," 2020, arXiv: 2002.12619.

[13] Z. Koldovský, V. Kautský, P. Tichavský, J. Čmejla, and J. Málek, "Dynamic independent component/vector analysis: Time-variant linear mixtures separable by time-invariant beamformers," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2158–2173, 2021.

[14] S. Doclo, S. Gannot, M. Moonen, A. Spriet, S. Haykin, and K. R. Liu, "Acoustic beamforming for hearing aid applications," *Handbook on array processing and sensor networks*, pp. 269–302, 2010.

[15] J. Li and P. Stoica, *Robust adaptive beamforming*. Wiley Online Library, 2006.

[16] S. A. Vorobyov, A. B. Gershman, Z.-Q. Luo, and N. Ma, "Adaptive beamforming with joint robustness against mismatched signal steering vector and interference nonstationarity," *IEEE Signal Processing Letters*, vol. 11, no. 2, pp. 108–111, 2004.

[17] N. Amor, J. Čmejla, V. Kautský, Z. Koldovský, and T. Kounovský, "Blind extraction of moving sources via independent component and vector analysis: Examples," in *accepted for 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2021)*, 2021.

[18] N. Ono, "Stable and fast update rules for independent vector analysis based on auxiliary function technique," in *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pp. 189–192, 2011.

[19] A. Hyvärinen, "Fast and robust fixed-point algorithm for independent component analysis," *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.

[20] I. Lee, T. Kim, and T.-W. Lee, "Fast fixed-point independent vector analysis algorithms for convolutive blind source separation," *Signal Processing*, vol. 87, no. 8, pp. 1859–1871, 2007.

[21] H. Li and T. Adalı, "Complex-valued adaptive signal processing using nonlinear functions," *EURASIP Journal on Advances in Signal Processing*, vol. 2008, Feb 2008.

[22] E. Bingham and A. Hyvärinen, "A fast fixed-point algorithm for independent component analysis of complex valued signals," *International Journal of Neural Systems*, vol. 10, pp. 1–8, Feb. 2000.

[23] P. Tichavský, Z. Koldovský, and E. Oja, "Performance analysis of the FastICA algorithm and Cramér-Rao bounds for linear independent component analysis," *IEEE Transactions on Signal Processing*, vol. 54, pp. 1189–1203, April 2006.

[24] P. Tichavský, Z. Koldovský, and E. Oja, *Speed and Accuracy Enhancement of Linear ICA Techniques Using Rational Nonlinear Functions*, pp. 285–292. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007.

[25] Z. Koldovský, J. Málek, and J. Janský, "Extraction of independent vector component from underdetermined mixtures through block-wise determined modeling," in *Proceedings of IEEE International Conference on Audio, Speech and Signal Processing*, vol. 7903–7907, May 2019.