Unsupervised Bayesian Estimation and Tracking of Time-Varying Convolutive Multichannel Systems

Herbert Buchner
Department of Engineering, University of Cambridge
Cambridge, UK
hb444@cam.ac.uk

Karim Helwani
Amazon Inc.
Sunnyvale, CA, USA
karim.helwani@ieee.org

Simon Godsill
Department of Engineering, University of Cambridge
Cambridge, UK
sjg30@cam.ac.uk

***Abstract*—In this paper we focus on Bayesian blind and semi-blind adaptive signal processing based on a broadband MIMO FIR model (e.g., for blind source separation (BSS) and blind system identification (BSI)). Specifically, we study in this paper a framework allowing us to systematically incorporate various types of prior knowledge: (1) source signal statistics, (2) deterministic knowledge on the mixing system, and (3) stochastic knowledge on the mixing system. In order to exploit all possible types of source signal statistics (1), our considerations are based on TRINICON, a previously introduced generic framework for broadband blind (and semi-blind) adaptive MIMO signal processing. The motivation for this paper is threefold: (a) the extension of TRINICON to Bayesian point estimation to address (3) in addition to (1), and (b) more specifically to unify system-based blind adaptive MIMO signal processing with the tracking of time-varying scenarios, and finally (c) to show how the Bayesian TRINICON-based tracking can be formulated as a sequence estimation approach on arbitrary partly smooth manifolds. As we will see in this paper, the Bayesian approach to incorporate stochastic priors and the manifold learning approach to exploit deterministic system knowledge (2) complement one another very efficiently in the context of TRINICON.**

***Index Terms*—Bayesian learning, tracking, manifolds, convolutive BSS, blind system identification**

I. Introduction

The analysis of scenes using sensor arrays (e.g., based on multiple channels of acoustic signals, sonar signals, or radar signals) is a fundamental problem in signal processing with many potential applications and use cases, e.g., for localization/object tracking, source separation/extraction, characterization of the environment, and signal enhancement/dereverberation. In general, the received sensor signals may consist of a mixture of components from multiple simultaneously active sources or objects of interest and various interfering signals located at different spatial positions. Additionally, there can be multipath and/or dispersive propagation of the various wave components and hence we often have to deal with convolutive signal mixtures, e.g., acoustic signals in reverberant indoor environments. In other words, assuming the propagation medium is linear, we have a matrix of convolutive propagation paths from the multiple sources to the multiple employed sensors. Since typically neither the original source signals nor the convolutive propagation paths are known *a priori*, we generally treat in this paper this overall array-based scene analysis problem as a *blind estimation problem*.

The special focus of this paper is on (1) solving the above-mentioned array signal processing problems using blind broadband adaptive MIMO (Multiple-Input and Multiple-Output) systems with explicit FIR models, and (2) to consider specifically the case of *time-varying MIMO mixing systems* within a *Bayesian framework*. In practice, time-varying mixing systems are very common, as they describe

- moving sources,
- moving reflectors/scattering objects,
- moving sensor array,

or a combination thereof. Accordingly, from an algorithmic point of view, this paper will bring together elements from Bayesian tracking (e.g., [1], [2]) and elements from blind adaptive signal processing (e.g., [3]–[6]). Apart from the high-dimensionality of this tracking problem (on the order of the filter length L) compared to common 2D or 3D tracking problems, another important challenge from the signal-processing point of view is the blindness (well known tracking algorithms, such as Kalman filters or particle filters, are generally to be regarded as supervised algorithms). Moreover, an important objective in this paper is to make use of TRINICON ('TRIple-N ICA for CONvolutive mixtures'), a previously introduced generic concept for broadband adaptive MIMO filtering, e.g., [4]–[6], using the technique of independent component analysis (ICA), e.g., [3], and in the *Bayesian TRINICON* framework, as presented in this paper, all the essential structures and degrees of freedom (see the references for more details) should be maintained.

The three main properties of TRINICON, making it an attractive vehicle for the present paper are: Its rigorous broadband formalism, its configurability to all the above tasks (blind, semi-blind, and also supervised [7], [8]), and its inherent exploitation of all possible statistical properties of the source signals ('TRIple-N'). Due to these properties, we can make use of the various relationships to most of the known blind, semi-blind, and supervised adaptive signal processing algorithms via the well defined degrees of freedom, and the fact that it has already led to various efficient time-

Work done while K.H. was with Huawei European Research Center, Munich, Germany. We thank Huawei ERC for the support.

domain and frequency-domain realizations, such as the blind unconstrained frequency-domain adaptive filter (now widely popular as *Independent Vector Analysis*) [4], [6].

Finally, an additional focus (3) of this paper is to formulate the (TRINICON-based) estimation and tracking on an arbitrary partly smooth manifold $\mathcal{M}$ in a way as shown previously in [9] for the non-Bayesian case. As shown in [9], being able to choose arbitrary partly smooth manifolds (a manifold is a topological space that is *locally* Euclidean [10], [11]) is a powerful approach to constrain the search space according to requirements of the desired application, or to further increase the convergence speed, or to reduce the computational complexity, or a combination thereof. Hence, it also allows us to extend the framework to the various semi-blind algorithms (including supervised algorithms as a special case), such as adaptive beamforming. Moreover, for blind adaptive filtering the *natural* manifold is of special importance (e.g., natural gradient descent [12]), and the general manifold-based approach in this paper will show how to incorporate this concept into the Bayesian tracking algorithms.

Generally, one of the main advantages of working with broadband MIMO FIR models is that we can, in principle, avoid any signal distortions in the source separation and signal enhancement tasks. Moreover, once we have obtained a robust estimate of the MIMO demixing system, we can infer most of the other relevant information for scene analysis, as mentioned above. A prominent example is acoustic source localization which is typically based on certain intermediate quantities, such as direction(s) of arrival (DOA) or time difference(s) of arrival (TDOA) in the classical approaches. It can be shown that these classical approaches can be related directly as special cases to the more *'holistic' approach* outlined above (see Fig. 1), which explicitly takes into account the reverberant environment and the multiple, possibly simultaneously active sources (e.g., [4] and references therein). Similar

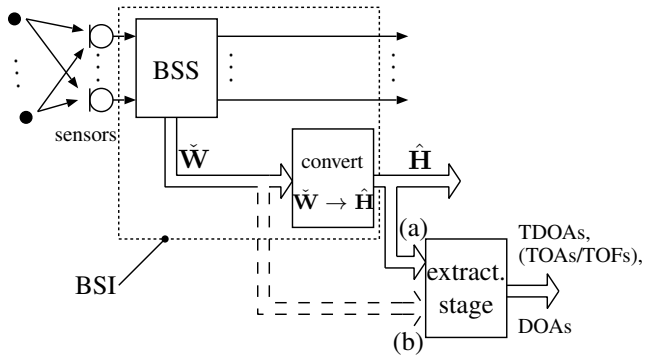

Fig. 1. Using blind source separation(BSS)/blind system identification (BSI) for source localization in convolutive environments.

relationships as in the forward adaptive filter problems (separation/interference cancellation, system identification) can also be established for inverse adaptive filter problems, such as blind dereverberation, e.g., [13].

Among the various known classes of Bayesian estimation and tracking algorithms (e.g., [1], [2]), we focus in this paper on Bayesian point estimators. This can be motivated by robustness and complexity considerations due to the typically very high dimensionality of the state space in our application, which is given by the filter length. Probably the best known and most popular classical tracking algorithm is the Kalman filter (KF) [14], [15], and its variants, such as the Extended Kalman filter (EKF) and the Unscented Kalman filter (UKF) [1], [2]. This class of algorithms can either be derived from a full Bayesian approach (under Gaussian assumptions, specified by mean and covariance) or *equivalently* as a point estimator of the mean vector (where the covariance estimate is obtained in the algorithm as a by-product). Coming from the field of adaptive array processing, and for the purpose of unifying these ideas with TRINICON, the latter approach is probably the more illustrative one in this paper. Indeed, the original paper by Thiele [14], refering directly to the earlier work by Gauss on the least-squares (LS) estimator, already describes this algorithm as a *dynamical model-based regularized LS problem* (directly reflected by the title of [14]), where the additive regularization term to the LS cost function is the *log prior* according to Bayes' rule. For adaptive signal processing in the special case of supervised adaptation, the (nonregularized) TRINICON-based adaptation was first contrasted with the Kalman-based (i.e., dynamically regularized RLS) adaptation in [16].

In this paper, we will confirm experimentally, that this additive regularization formulation also holds for the TRINICON optimization criterion, so that this idea of model-based regularization also carries over to the general blind and semi-blind broadband adaptive MIMO signal processing. A more detailed theoretical derivation of the Bayesian TRINICON optimization criterion can be found in [17]. Some early regularized ICA algorithms (not convolutive, not dynamically regularized, ICA only based on nongaussianity, not the full empirical risk minimization) can be found, e.g., in [18], [19]. Based on the Bayesian TRINICON optimization criterion, i.e., a suitably regularized version of the TRINICON criterion, and the manifold-based framework [9], the derivation of the generic algorithm in this paper follows from a *sequence estimation* in a similar way as shown in [20], by optimizing w.r.t. a *state vector* which collects the history (concatenation) of all sets of previous demixing filters.

II. General MIMO setup and notation

In this paper, we denote the original source signals by $s_q(n)$, $q = 1, \ldots, Q$ and the captured sensor signals by $x_p(n)$, $p = 1, \ldots, P$. We describe the MIMO mixing system by length-M FIR filters, where $h_{qp,\kappa}$, $\kappa = 0, \ldots, M-1$ denote the coefficients of the FIR filter model from the q-th source signal $s_q(n)$ to the p-th sensor signal $x_p(n)$. We assume throughout this paper that $Q \leq P$. According to the optimization criterion, we are interested in finding a corresponding length-L FIR demixing system with coefficients $w_{pq,\kappa}$. This yields the output signals $y_q(n)$. As a compact formulation

of the set of demixing filter coefficients and mixing filter coefficients we form the $PL \times Q$ demixing coefficient matrix

$$\check{\mathbf{W}} = \begin{bmatrix} \mathbf{w}_{11} & \cdots & \mathbf{w}_{1Q} \\ \vdots & \ddots & \vdots \\ \mathbf{w}_{P1} & \cdots & \mathbf{w}_{PQ} \end{bmatrix} \tag{1}$$

and the corresponding $QM \times P$ mixing coefficient matrix $\check{\mathbf{H}}$, respectively, where

$$\mathbf{h}_{qp} = [h_{qp,0}, \ldots, h_{qp,M-1}]^{\mathrm{T}}, \tag{2}$$

$$\mathbf{w}_{pq} = [w_{pq,0}, \ldots, w_{pq,L-1}]^{\mathrm{T}} \tag{3}$$

denote the coefficient vectors of the FIR subfilters of the MIMO systems, and superscript $^{\mathrm{T}}$ denotes transposition of a vector or a matrix. The downwards pointing hat symbol on top of $\mathbf{W}$ in (1) serves to distinguish this *condensed* matrix from the corresponding larger matrix structure $\mathbf{W}$ as introduced below. The rigorous distinction between these different matrix structures is also an essential aspect of the general TRINICON framework, as shown later.

To model the time-varying nature of $\check{\mathbf{W}}$, we consider in this paper the following class of *state-space models* with *block time-index* m and a vector-valued function $\mathbf{g}(\cdot,\cdot)$:

$$\mathrm{vec}\,\check{\mathbf{W}}(m+1) = \mathbf{g}\left(\mathrm{vec}\,\check{\mathbf{W}}(m), \mathbf{u}(m)\right) + \mathbf{z}(m), \tag{4}$$

where $\mathbf{u}(m)$ denotes some possible innovation term, and $\mathbf{z}(m)$ denotes the process noise. As an important special case, this model includes the linear Gauss-Markov model with a (possibly time-varying) transition matrix $\mathbf{A}(m)$,

$$\mathrm{vec}\,\check{\mathbf{W}}(m+1) = \mathbf{A}(m)\,\mathrm{vec}\,\check{\mathbf{W}}(m) + \mathbf{u}(m) + \mathbf{z}(m), \tag{5}$$

where the process noise $\mathbf{z}(m)$ is assumed to be gaussian, described by a covariance matrix $\mathbf{Q}(m) = \mathcal{E}\{\mathbf{z}(m)\mathbf{z}^{\mathrm{T}}(m)\}$.

In the following sections, without loss of generality (i.e., equivalently to saying that $Q \leq P$), we assume $Q = P$ sources which *may or may not all be simultaneously active at a given time instant*.

III. TRINICON AS A GENERAL FRAMEWORK FOR BROADBAND SIGNAL PROCESSING ON CONVOLUTIVE MIXTURES AND EXTENSION TO THE BAYESIAN CASE

In the following, to simplify the derivation, we work with an *expanded state vector* $\mathrm{vec}\,\check{\boldsymbol{\theta}}(m)$ of size $(m+1)P^2L \times 1$ which captures the whole state evolution, i.e., the *sequence of coefficient vectors* $\mathrm{vec}\,\check{\mathbf{W}}(i)$ from block time index $i = 0$ up to and including the current block time index $i = m$:

$$\begin{aligned} \mathrm{vec}\,\check{\boldsymbol{\theta}}(m) &:= \left[\mathrm{vec}^{\mathrm{T}}\,\check{\mathbf{W}}(0), \ldots, \mathrm{vec}^{\mathrm{T}}\,\check{\mathbf{W}}(m)\right]^{\mathrm{T}} \\ &= \begin{bmatrix} \mathrm{vec}\,\check{\boldsymbol{\theta}}(m-1) \\ \mathrm{vec}\,\check{\mathbf{W}}(m) \end{bmatrix}. \end{aligned} \tag{6}$$

Taking into account the whole sequence $\check{\boldsymbol{\theta}}(m)$ for the optimization process rather than only the current coefficient matrix $\check{\mathbf{W}}(m)$, and using the dynamical model (4) in this way as a prior on $\check{\boldsymbol{\theta}}(m)$ are essentially the keys for obtaining the filter equations with tracking capability. Based on this state vector, we now formulate a criterion that is minimized with respect to the *sequence* of filter coefficient vectors up to block m, i.e., w.r.t. the expanded state vector $\mathrm{vec}\check{\boldsymbol{\theta}}(m)$:

$$\begin{aligned} \mathcal{J}(m, \boldsymbol{\theta}(m)) &= \sum_{i=0}^{m} \beta(i,m) \tilde{\mathcal{J}}(i, \boldsymbol{\theta}) \\ &= \sum_{i=0}^{m} \beta(i,m) \left[\tilde{\mathcal{J}}_0(i, \boldsymbol{\theta}) + \tilde{\mathcal{J}}_{\mathrm{R}}(i, \boldsymbol{\theta})\right], \end{aligned}$$

where $\beta(i,m)$ is a weighting function defining different classes of algorithms [8] and allowing for online, offline, or block-online algorithms [6]. Similar to the distinction between $\check{\mathbf{W}}(m)$ and $\mathbf{W}(m)$, we also distinguish between $\check{\boldsymbol{\theta}}(m)$ and $\boldsymbol{\theta}(m)$ as explained below. In this paper we choose

$$\beta(i,m) = \alpha \cdot \lambda^{m-i} \tag{7}$$

with $0 < \lambda \leq 1$ and an arbitrary real constant $\alpha > 0$.

The two terms $\tilde{\mathcal{J}}_0$ and $\tilde{\mathcal{J}}_{\mathrm{R}}$ in

$$\tilde{\mathcal{J}}(m, \boldsymbol{\theta}(m)) = \tilde{\mathcal{J}}_0(m, \boldsymbol{\theta}(m)) + \tilde{\mathcal{J}}_{\mathrm{R}}(m, \boldsymbol{\theta}(m)) \tag{8a}$$

denote the current data-based term and the regularization term, respectively, as discussed in the following subsections.

An important feature of the choice (7) for $\beta(i,m)$ is that it allows a recursive computation as

$$\mathcal{J}(m, \boldsymbol{\theta}(m)) = \lambda \mathcal{J}(m-1, \boldsymbol{\theta}(m-1)) + \alpha \tilde{\mathcal{J}}(m, \boldsymbol{\theta}(m)). \tag{8b}$$

1) Regularization Term: We first consider the regularization term $\tilde{\mathcal{J}}_{\mathrm{R}}(m, \boldsymbol{\theta}(m))$. As it can be interpreted as a logarithmized prior on the convolutive demixing system using the state-space model, we write

$$\begin{aligned} &\tilde{\mathcal{J}}_{\mathrm{R}}(m, \boldsymbol{\theta}(m)) \quad = \\ &= f\left(\mathrm{vec}\check{\mathbf{W}}(m) - \mathbf{g}\left(\mathrm{vec}\check{\mathbf{W}}(m-1), \mathbf{u}(m-1)\right)\right) \\ &= f\left(\mathrm{vec}\check{\mathbf{W}}(m) - \mathbf{g}\left(\mathbf{W}^{01}_{P^2L \times mP^2L}\mathrm{vec}\,\check{\boldsymbol{\theta}}(m-1), \mathbf{u}(m-1)\right)\right) \end{aligned} \tag{9}$$

with a potential function $f(\cdot)$ and the window matrix $\mathbf{W}^{01}_{P^2L \times mP^2L} := [\mathbf{0}, \ldots, \mathbf{0}, \mathbf{I}]$. The special case of the Gauss-Markov regularizer follows for

$$\mathbf{g}(\bullet, \bullet) = \mathbf{A}(m-1) \bullet + \bullet, \tag{10a}$$

$$f(\cdot) = \| \bullet \|^2_{\mathbf{Q}^{-1}(m-1)} = \bullet^{\mathrm{T}} \mathbf{Q}^{-1}(m-1) \bullet . \tag{10b}$$

For the recursive calculation of $\mathcal{J}$ we assume $\mathbf{A}(-1) = \mathbf{0}$ and $\mathbf{u}(-1) =: \mathbf{u}_0$ as initial values for $i = 0$.

2) Data Term: Various approaches exist to estimate the demixing matrix $\check{\mathbf{W}}$ by utilizing the fundamental source signal properties nongaussianity, nonwhiteness, and nonstationarity [3] which were all combined in TRINICON using the following data term [5]:

$$\begin{aligned} &\tilde{\mathcal{J}}_0(m, \mathbf{W}(m)) \quad = \\ &- \frac{1}{N} \sum_{j=mN_L}^{mN_L+N-1} \left\{\log(\hat{p}_{s,PD}(\mathbf{y}(j))) - \log(\hat{p}_{y,PD}(\mathbf{y}(j)))\right\}, \end{aligned} \tag{11}$$

where $\hat{p}_{s,PD}(\cdot)$ and $\hat{p}_{y,PD}(\cdot)$ are assumed or estimated PD-variate source model (i.e., desired) pdf and output pdf, respectively. The index m denotes the block time index for a block of N output samples shifted by N_L samples relatively to the previous block (e.g., $N_L = L$ as a common choice).

To introduce an algorithm for broadband processing of convolutive mixtures, we formulate the convolution of the FIR demixing system of length L and with D time lags for each demixing filter output channel in the following matrix form [5]:

$$\mathbf{y}(n) = \mathbf{W}^{\mathrm{T}}\mathbf{x}(n), \tag{12}$$

where n denotes the time index, and

$$\begin{aligned}
\mathbf{x}(n) &= [\mathbf{x}_1^{\mathrm{T}}(n), \ldots, \mathbf{x}_P^{\mathrm{T}}(n)]^{\mathrm{T}}, && (13)\\
\mathbf{y}(n) &= [\mathbf{y}_1^{\mathrm{T}}(n), \ldots, \mathbf{y}_P^{\mathrm{T}}(n)]^{\mathrm{T}}, && (14)\\
\mathbf{x}_p(n) &= [x_p(n), \ldots, x_p(n-2L+1)]^{\mathrm{T}}, && (15)\\
\mathbf{y}_q(n) &= [y_q(n), \ldots, y_q(n-D+1)]^{\mathrm{T}}. && (16)
\end{aligned}$$

The parameter D in (16), $1 \leq D < L$, denotes the number of time lags taken into account to exploit the nonwhiteness of the source signals. $\mathbf{W}_{pq}$, $p = 1, \ldots, P$, $q = 1, \ldots, P$ denote $2L \times D$ *Sylvester matrices* that contain all coefficients of the respective filters in each column by successive shifting, i.e., the first column reads $\left[\mathbf{w}_{pq}^{\mathrm{T}}, 0, \ldots, 0\right]^{\mathrm{T}}$, the second column $\left[0, \mathbf{w}_{pq}^{\mathrm{T}}, 0, \ldots, 0\right]^{\mathrm{T}}$, etc. Finally, the $2PL \times PD$ matrix $\mathbf{W}$ combines all Sylvester matrices $\mathbf{W}_{pq}$.

The rigorous distinction between the different matrix structures $\breve{\mathbf{W}}$ and $\mathbf{W}$ due to the nonwhiteness for $D > 1$ is also an essential aspect of the general TRINICON framework and leads to an important building block whose actual implementation is fundamental to the properties of the resulting algorithm, the so-called *Sylvester constraint* ($\mathcal{SC}$) on the coefficient update, formally introduced in [6]. The Euclidean gradient of the data term can then be written as

$$\nabla_{\breve{\mathbf{W}}}\tilde{\mathcal{J}}_0(m, \mathbf{W}) = \mathcal{SC}\left\{\nabla_{\mathbf{W}}\tilde{\mathcal{J}}_0(m, \mathbf{W})\right\}, \tag{17}$$

or using a fixed matrix $\mathbf{K}_{\mathcal{SC}}$ (e.g., [9]),

$$\nabla_{\mathrm{vec}\breve{\mathbf{W}}}\tilde{\mathcal{J}}_0(m, \mathbf{W}) = \mathbf{K}_{\mathcal{SC}}\nabla_{\mathrm{vec}\mathbf{W}}\tilde{\mathcal{J}}_0(m, \mathbf{W}), \tag{18}$$

where

$$\nabla_{\mathbf{W}}\tilde{\mathcal{J}}_0(m, \mathbf{W}) = \sum_{j=iN_L}^{iN_L+N-1}\left[\mathbf{x}(j)\boldsymbol{\Phi}_{s,PD}^{\mathrm{T}}(\mathbf{y}(j)) - \left(\mathbf{W}^{\mathrm{T}}\right)^{+}\right], \tag{19}$$

with $\cdot^{+}$ denoting the pseudoinverse of a matrix, and with the generalized score function $\boldsymbol{\Phi}_{s,PD}(\mathbf{y}(j))$ according to [13]. As an example, for convolutive blind source separation, the desired pdf is factorized w.r.t. the output channels. This leads to a concatenation of P individual score functions $\boldsymbol{\Phi}_{q,D}$ of dimension D each, and for the special case of second-order statistics, i.e., multivariate Gaussian source models [6], they read $\boldsymbol{\Phi}_{q,D}(\mathbf{y}_q(j)) = \mathbf{R}_{\mathbf{y}_q\mathbf{y}_q}^{-1}(i)\mathbf{y}_q(j)$.

IV. Generic Newton-type algorithm for sequence estimation on arbitrary partly smooth manifolds for time-varying convolutive mixing systems

Having defined the Bayesian TRINICON optimization criterion and the related quantities in the previous section, we now derive a generic estimation and tracking algorithm.

In [9] we presented a generic TRINICON-based Newton-type algorithm on arbitrary partly smooth manifolds, with multiple possible iterations per block. It was shown to contain several important optimization techniques as special cases [9], including the *natural gradient* which is of special importance for *unsupervised estimation* (see Sect. I). Since a manifold $\mathcal{M}$ is defined as a topological space that is locally Euclidean, local parameterizations in a Euclidean *tangent space* always exist. Hence, the idea in [9] was to perform each update step $\Delta\mathbf{T}(m)$ in the current tangent space, followed by the application of the local map $\mathbf{W}^{\ell} = \varphi_{\mathbf{W}^{\ell-1}}(\Delta\mathbf{T})$ back to the coefficient space.

In this paper, we apply this strategy for the more general case of *sequence estimation* by formulating the algorithm in terms of the corresponding *state vectors* for the coefficient matrices and the tangent space updates $\Delta\mathbf{T}(m)$ analogously to (6), i.e.,

$$\mathrm{vec}\,\Delta\overline{\mathbf{T}}(m) := \left[\mathrm{vec}^{\mathrm{T}}\,\Delta\mathbf{T}(0), \ldots, \mathrm{vec}^{\mathrm{T}}\,\Delta\mathbf{T}(m)\right]^{\mathrm{T}}. \tag{20}$$

The resulting algorithm can be simplified significantly in a number of steps, the most important steps resulting from picking only the last subvector of the new state vector, and by spliting the algorithm into a model-based *prediction part* and a data-based *correction part*, as shown in the remainder of this Section. As we will see, it turns out that this prediction-correction formulation is indeed made possible for arbitrary partly smooth manifolds by the chosen state-space model (4). Further algorithm formulations resulting from this framework can be found in [17].

Figure 2 outlines the basic idea for the TRINICON-based adaptation on an arbitrary manifold which will form the basis for the following mathematical developments in this section. Note that in the context of TRINICON and adaptive MIMO

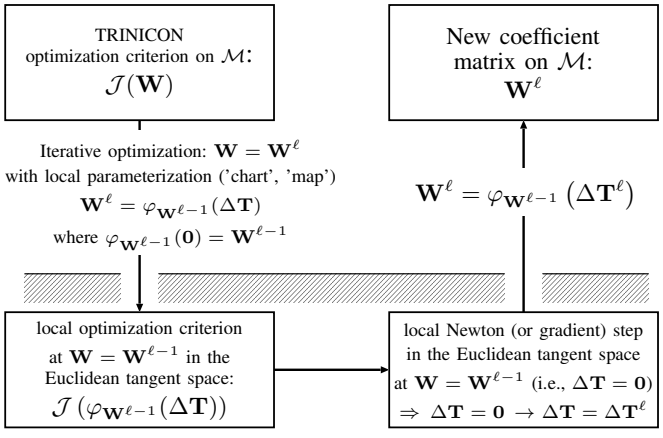

Fig. 2. Basic approach for TRINICON-based optimization on an arbitrary partly smooth manifold $\mathcal{M}$.

systems we have to deal with matrix-valued manifolds.

According to [9], the key quantities for the manifold-based algorithms are the description of the new coefficients by the (potentially nonlinear) map $\mathbf{W}^{\ell} = \varphi_{\mathbf{W}^{\ell-1}}(\Delta\mathbf{T})$, and the resulting *directional derivative* $\boldsymbol{\Xi}$ of the map w.r.t. the tangent vector. For *sequence estimation* we formulate these quantities in terms of the extended state vectors containing all the history. Hence, adapted from [9], the set of coefficient update equations, depending on the manifold $\mathcal{M}$, reads:

$$\overline{\boldsymbol{\Xi}}\left(m,\boldsymbol{\theta}^{\ell-1}(m)\right) := \left.\frac{\partial \mathrm{vec}^{\mathrm{T}}\left(\boldsymbol{\varphi}_{\boldsymbol{\theta}^{\ell-1}(m)}\right)}{\partial \mathrm{vec}\left(\Delta\overline{\mathbf{T}}\right)}\right|_{\Delta\overline{\mathbf{T}}=\mathbf{0}}, \quad (21a)$$

Euclidean Hessian:

$$\mathbf{P}_{\boldsymbol{\theta}}\left(m,\boldsymbol{\theta}^{\ell-1}(m)\right) = \sum_{i=0}^{\infty}\beta(i,m)\cdot\frac{1}{N}\sum_{j=iN_L}^{iN_L+N-1} (\mathbf{I}\otimes\mathbf{x}(j))\left.\frac{\partial\left(\boldsymbol{\Phi}_{\mathrm{s},PD}^{\mathrm{T}}(\mathbf{y}(j))-\boldsymbol{\Phi}_{\mathrm{y},PD}^{\mathrm{T}}(\mathbf{y}(j))\right)}{\partial\mathbf{y}}\right|_{\mathbf{W}=\mathbf{W}^{\ell-1}(m)}(\mathbf{I}\otimes\mathbf{x}^{\mathrm{T}}(j)) + \text{regularization term}, \quad (21b)$$

Hessian in the tangent space (omitting here the Christoffel symbols):

$$\mathbf{P}_{\Delta\overline{\mathbf{T}}}\left(m,\boldsymbol{\theta}^{\ell-1}(m)\right) = = \overline{\boldsymbol{\Xi}}\left(m,\boldsymbol{\theta}^{\ell-1}(m)\right)\mathbf{P}_{\boldsymbol{\theta}}\left(m,\boldsymbol{\theta}^{\ell-1}(m)\right)\overline{\boldsymbol{\Xi}}^{\mathrm{T}}\left(m,\boldsymbol{\theta}^{\ell-1}(m)\right), \quad (21c)$$

Update in the tangent space:

$$\begin{aligned}\mathrm{vec}\Delta\overline{\mathbf{T}}^{\ell}(m) &= -\boldsymbol{\mu}\mathbf{P}_{\Delta\overline{\mathbf{T}}}^{-1}\sum_{i=0}^{\infty}\beta(i,m)\,\overline{\boldsymbol{\Xi}}\left(m,\boldsymbol{\theta}^{\ell-1}(m)\right) \\ &\quad\cdot\mathrm{vec}\left\{\nabla_{\boldsymbol{\theta}(m)}\tilde{\mathcal{J}}(m,\boldsymbol{\theta}(m))\right\}\Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{\ell-1}(m)}, \quad (21d)\\ \check{\boldsymbol{\theta}}^{\ell}(m) &= \overline{\mathcal{SC}}\left\{\boldsymbol{\varphi}_{\boldsymbol{\theta}^{\ell-1}(m)}\left(\Delta\overline{\mathbf{T}}^{\ell}(m)\right)\right\}. \quad (21e)\end{aligned}$$

Similar to the definition of the state vector in (6), we have used here the corresponding definition in the tangent space:

$$\begin{aligned}\mathrm{vec}\,\Delta\overline{\mathbf{T}}(m) &:= \left[\mathrm{vec}^{\mathrm{T}}\,\Delta\mathbf{T}(0),\ldots,\mathrm{vec}^{\mathrm{T}}\,\Delta\mathbf{T}(m)\right]^{\mathrm{T}} \\ &= \begin{bmatrix}\mathrm{vec}\,\Delta\overline{\mathbf{T}}(m-1)\\ \mathrm{vec}\,\Delta\mathbf{T}(m)\end{bmatrix}, \quad (22)\end{aligned}$$

as introduced above.

To perform block-online adaptation, where the superscript ℓ denotes an index of possible offline iterations ($\ell = 1,\ldots,\ell_{\max}$) within each block m, the above adaptation procedure is performed in the following scheme:

$$\begin{aligned}\mathrm{vec}\check{\boldsymbol{\theta}}^{0}(m) &:= \begin{bmatrix}\mathrm{vec}\check{\boldsymbol{\theta}}(m-1)\\ \mathrm{vec}\check{\mathbf{W}}^{0}(m)\end{bmatrix}, \quad (23a)\\ &\text{For}\quad \ell=1,\ldots,\ell_{\max}: \\ &\qquad\text{calculate}\ \check{\boldsymbol{\theta}}^{\ell}(m)\text{using procedure (21)}\\ \check{\boldsymbol{\theta}}(m) &:= \check{\boldsymbol{\theta}}^{\ell_{\max}}(m). \quad (23b)\end{aligned}$$

Due to the requirement of increasing dimensionality of the state space vector with the increasing block index m according to (6), the new initial vector $\mathrm{vec}\check{\boldsymbol{\theta}}^{0}(m)$ is obtained in (23a) by appending a new length-LPQ vector $\mathrm{vec}\check{\mathbf{W}}^{0}(m)$. An appropriate way of choosing $\mathrm{vec}\check{\mathbf{W}}^{0}(m)$ will be discussed below. Note that in the classical algorithms (including the Kalman Filter), $\ell_{\max} = 1$ which leads to the familiar link between ℓ and time index m.

A. Simplification of the exact sequence estimation algorithm on arbitrary partly smooth manifolds

We now significantly simplify the generic sequence estimation algorithm (21) in several steps. The first step is to pick the last subvector of $\mathrm{vec}\check{\boldsymbol{\theta}}^{\ell}(m)$. For online/block-online estimation, this last subvector contains the current parameters of interest:

$$\mathrm{vec}\,\check{\mathbf{W}}^{\ell}(m) = \mathbf{K}_{\mathcal{SC}}\,\mathrm{vec}\,\boldsymbol{\varphi}_{\mathbf{W}^{\ell-1}(m)}\left(\Delta\mathbf{T}^{\ell}(m)\right). \quad (24)$$

Hence, it must hold

$$\mathrm{vec}\,\boldsymbol{\varphi}_{\boldsymbol{\theta}^{\ell-1}(m)}\left(\Delta\mathbf{T}^{\ell}(m)\right) = \begin{bmatrix}\mathrm{vec}\,\boldsymbol{\varphi}_{\mathbf{W}^{\ell-1}(0)}\left(\Delta\mathbf{T}^{\ell}(0)\right)\\ \vdots\\ \mathrm{vec}\,\boldsymbol{\varphi}_{\mathbf{W}^{\ell-1}(m)}\left(\Delta\mathbf{T}^{\ell}(m)\right)\end{bmatrix}, \quad (25)$$

and if the map $\boldsymbol{\varphi}_{\mathbf{W}^{\ell-1}(m)}\left(\Delta\mathbf{T}^{\ell}(m)\right)$ only depends on $\Delta\mathbf{T}(m)$, we infer from (21a) that $\overline{\boldsymbol{\Xi}}\left(m,\boldsymbol{\theta}^{\ell-1}(m)\right)$ must be a block-diagonal matrix

$$\overline{\boldsymbol{\Xi}}\left(m,\boldsymbol{\theta}^{\ell-1}(m)\right) = = \mathrm{Bdiag}\left\{\boldsymbol{\Xi}\left(0,\mathbf{W}^{\ell-1}(0)\right),\ldots,\boldsymbol{\Xi}\left(m,\mathbf{W}^{\ell-1}(m)\right)\right\}, \quad (26)$$

where

$$\boldsymbol{\Xi}\left(m,\mathbf{W}^{\ell-1}(m)\right) = \left.\frac{\partial \mathrm{vec}^{\mathrm{T}}\left(\boldsymbol{\varphi}_{\mathbf{W}^{\ell-1}(m)}\right)}{\partial \mathrm{vec}\left(\Delta\mathbf{T}(m)\right)}\right|_{\Delta\mathbf{T}(m)=\mathbf{0}}. \quad (27)$$

Having replaced (21e) by the much simpler equation (24), we now turn our attention to the update step (21d) which requires the calculation of the gradient and Hessian w.r.t. the state vector. Both of them can be significantly simplified. It can be shown that the gradient $\mathrm{vec}\,\nabla_{\boldsymbol{\theta}(m)}\tilde{\mathcal{J}}(m,\boldsymbol{\theta}(m))$ in (21d) can be expressed using the matrix $\mathbf{K}_{\mathcal{SC}}$ analogously to (18) as

$$\mathrm{vec}\,\nabla_{\boldsymbol{\theta}(m)}\tilde{\mathcal{J}}(m,\boldsymbol{\theta}(m)) = \left(\mathbf{I}\otimes\mathbf{K}_{\mathcal{SC}}^{\mathrm{T}}\right)\mathrm{vec}\,\nabla_{\check{\boldsymbol{\theta}}(m)}\tilde{\mathcal{J}}(m,\boldsymbol{\theta}(m)). \quad (28)$$

Moreover, according to (6) the gradient of $\tilde{\mathcal{J}}(m,\boldsymbol{\theta}(m))$ w.r.t. $\mathrm{vec}\,\check{\boldsymbol{\theta}}(m)$ on the right hand side of (28) is written as

$$\mathrm{vec}\nabla_{\check{\boldsymbol{\theta}}(m)}\tilde{\mathcal{J}}(m,\boldsymbol{\theta}(m)) = \begin{bmatrix}\mathrm{vec}\nabla_{\check{\boldsymbol{\theta}}(m-1)}\\ \mathrm{vec}\nabla_{\check{\mathbf{W}}(m)}\end{bmatrix}\tilde{\mathcal{J}}(m,\boldsymbol{\theta}(m)). \quad (29)$$

As the data term $\tilde{\mathcal{J}}_0$ in (8a) only depends on the current vector $\mathrm{vec}\check{\mathbf{W}}(m)$, this expression simplifies to

$$\mathrm{vec}\nabla_{\check{\boldsymbol{\theta}}(m)}\tilde{\mathcal{J}}(m,\boldsymbol{\theta}(m)) = \begin{bmatrix}\mathrm{vec}\nabla_{\check{\boldsymbol{\theta}}(m-1)}\tilde{\mathcal{J}}_{\mathrm{R}}(m,\boldsymbol{\theta}(m))\\ \mathrm{vec}\nabla_{\check{\mathbf{W}}(m)}\tilde{\mathcal{J}}_{\mathrm{R}}(m,\boldsymbol{\theta}(m)) + \mathrm{vec}\nabla_{\check{\mathbf{W}}(m)}\tilde{\mathcal{J}}_{0}(m,\mathbf{W}(m))\end{bmatrix}.$$

Finally, as we will illustrate next, a deliberate choice of the initial vector $\mathrm{vec}\check{\mathbf{W}}^0(m)$ in (23a) allows for a further simplification of of the gradient. For a variety of practically relevant dynamical model-based regularizers $\tilde{\mathcal{J}}_{\mathrm{R}}(m,\boldsymbol{\theta}(m))$ according to Sect. III-1, the vector $\mathrm{vec}\check{\mathbf{W}}^0(m)$ can be chosen such that the two gradients of $\tilde{\mathcal{J}}_{\mathrm{R}}(m,\boldsymbol{\theta}(m))$ become equal to zero. It can be shown that this is indeed the case for the family of regularizers (9), and the choice

$$\mathrm{vec}\,\check{\mathbf{W}}^0(m) = \mathbf{g}\left(\mathrm{vec}\,\check{\mathbf{W}}(m-1), \mathbf{u}(m-1)\right). \tag{30}$$

Due to the similarity to (4) this initialization step (30) can be understood as a *model-based prediction step*. In summary, with this specific choice of $\mathrm{vec}\check{\mathbf{W}}^0(m)$ we can express the gradient (29) in the very simple form

$$\mathrm{vec}\nabla_{\check{\boldsymbol{\theta}}(m)}\tilde{\mathcal{J}}(m,\boldsymbol{\theta}(m)) = \begin{bmatrix} \mathbf{0} \\ \mathrm{vec}\nabla_{\check{\mathbf{W}}(m)}\tilde{\mathcal{J}}_0(m,\mathbf{W}(m)) \end{bmatrix}. \tag{31}$$

Having simplified the gradient, we now focus on the Hessian $\mathbf{P}_{\Delta\overline{\mathbf{T}}}$ in the update equation (21d). Due to the simple structure of the gradient (31), and, as we are ultimately interested in the new estimate $\mathrm{vec}\,\Delta\mathbf{T}^\ell(m)$, i.e., the lowermost subvector of $\mathrm{vec}\,\Delta\overline{\mathbf{T}}^\ell(m)$, we are only interested in the lower right corner of the inverse Hessian $\mathbf{P}^{-1}_{\Delta\overline{\mathbf{T}}}$. Specifically, using (31) and the block-diagonal structure of $\overline{\boldsymbol{\Xi}}\cdot\left(\mathbf{I}\otimes\mathbf{K}^{\mathrm{T}}_{\mathcal{SC}}\right)$ in (28) and (26), respectively, and the properties of $\mathbf{K}_{\mathcal{SC}}$, the lowermost subvector of (21d) simplifies to

$$\begin{aligned}\mathrm{vec}\Delta\mathbf{T}^\ell(m) = &-\boldsymbol{\mu}\Big[\mathbf{P}^{-1}_{\Delta\overline{\mathbf{T}}}\Big]_{\substack{\text{lower}\\ \text{right}}}\sum_{i=0}^{\infty}\beta(i,m)\boldsymbol{\Xi}\left(m,\mathbf{W}^{\ell-1}(m)\right)\\ &\cdot\mathrm{vec}\left\{\nabla_{\mathbf{W}(m)}\tilde{\mathcal{J}}_0(m,\mathbf{W}(m))\right\}\Big|_{\mathbf{W}=\mathbf{W}^{\ell-1}(m)},\end{aligned} \tag{32}$$

As shown in Appendix A, this lower right corner is equal to the inverse of the *Schur complement* $\mathbf{F}_{22}$ of the matrix

$$\begin{aligned}\mathbf{P}_{\Delta\overline{\mathbf{T}}}(m,\boldsymbol{\theta}(m)) &= \overline{\boldsymbol{\Xi}}\,\mathbf{P}_{\boldsymbol{\theta}}(m,\boldsymbol{\theta}(m))\,\overline{\boldsymbol{\Xi}}^{\mathrm{T}}\\ &= \overline{\boldsymbol{\Xi}}\,\nabla_{\mathrm{vec}\boldsymbol{\theta}(m)}\nabla^T_{\mathrm{vec}\boldsymbol{\theta}(m)}\mathcal{J}(m,\boldsymbol{\theta}(m))\,\overline{\boldsymbol{\Xi}}^{\mathrm{T}}\\ = \begin{bmatrix}\overline{\boldsymbol{\Xi}}'\left(\lambda\mathbf{P}_{\boldsymbol{\theta}}(m-1,\boldsymbol{\theta}(m-1))+\mathbf{A}_{\boldsymbol{\theta}\boldsymbol{\theta}}\right)\overline{\boldsymbol{\Xi}}'^{\mathrm{T}} & \overline{\boldsymbol{\Xi}}'\,\mathbf{A}_{\boldsymbol{\theta}\mathbf{W}}\,\boldsymbol{\Xi}^{\mathrm{T}}\\ \boldsymbol{\Xi}\,\mathbf{A}^{\mathrm{T}}_{\boldsymbol{\theta}\mathbf{W}}\,\overline{\boldsymbol{\Xi}}'^{\mathrm{T}} & \boldsymbol{\Xi}\left(\mathbf{A}_{\mathbf{WW}}+\mathbf{B}\right)\boldsymbol{\Xi}^{\mathrm{T}}\end{bmatrix},\end{aligned}$$

where we used the recursion (8b) and the short-hand notations [based on (26)]

$$\begin{aligned}\boldsymbol{\Xi} &= \boldsymbol{\Xi}(m,\boldsymbol{\theta}(m)),\\ \overline{\boldsymbol{\Xi}} &= \overline{\boldsymbol{\Xi}}(m,\boldsymbol{\theta}(m)),\\ \overline{\boldsymbol{\Xi}}' &= \overline{\boldsymbol{\Xi}}(m-1,\boldsymbol{\theta}(m-1)),\end{aligned}$$

and the definitions

$$\begin{aligned}&\mathbf{P}_{\boldsymbol{\theta}}(m-1,\boldsymbol{\theta}(m-1))\\ &\quad= \nabla_{\mathrm{vec}\boldsymbol{\theta}(m-1)}\nabla^T_{\mathrm{vec}\boldsymbol{\theta}(m-1)}\mathcal{J}(m-1,\boldsymbol{\theta}(m-1))\end{aligned} \tag{33a}$$

$$\mathbf{A}_{\boldsymbol{\theta}\boldsymbol{\theta}} := \alpha\nabla_{\mathrm{vec}\boldsymbol{\theta}(m-1)}\nabla^T_{\mathrm{vec}\boldsymbol{\theta}(m-1)}\tilde{\mathcal{J}}_{\mathrm{R}}(m,\boldsymbol{\theta}(m)), \tag{33b}$$

$$\mathbf{A}_{\boldsymbol{\theta}\mathbf{W}} := \alpha\nabla_{\mathrm{vec}\boldsymbol{\theta}(m-1)}\nabla^T_{\mathrm{vec}\mathbf{W}(m)}\tilde{\mathcal{J}}_{\mathrm{R}}(m,\boldsymbol{\theta}(m)), \tag{33c}$$

$$\mathbf{A}_{\mathbf{WW}} := \alpha\nabla_{\mathrm{vec}\mathbf{W}(m)}\nabla^T_{\mathrm{vec}\mathbf{W}(m)}\tilde{\mathcal{J}}_{\mathrm{R}}(m,\boldsymbol{\theta}(m)), \tag{33d}$$

$$\mathbf{B} := \alpha\nabla_{\mathrm{vec}\mathbf{W}(m)}\nabla^T_{\mathrm{vec}\mathbf{W}(m)}\tilde{\mathcal{J}}_0(m,\mathbf{W}(m)). \tag{33e}$$

Based on these definitions, we can now express the *inverse Schur complement*

$$\left[\mathbf{P}^{-1}_{\Delta\overline{\mathbf{T}}}\right]_{\substack{\text{lower}\\ \text{right}}} = \mathbf{F}^{-1}_{22} =: \tilde{\mathbf{P}}_{\Delta\mathbf{T}}\left(m\,|\,\mathbf{W}^{\ell-1}(m)\right)$$

according to (49) as follows:

$$\begin{aligned}\tilde{\mathbf{P}}_{\Delta\mathbf{T}}\left(m\,|\,\mathbf{W}^{\ell-1}(m)\right) = &\left(\boldsymbol{\Xi}\left[\mathbf{A}_{\mathbf{WW}}\right.\right.\\ -\mathbf{A}^{\mathrm{T}}_{\boldsymbol{\theta}\mathbf{W}}&\overline{\boldsymbol{\Xi}}'^{\mathrm{T}}\left[\lambda\overline{\boldsymbol{\Xi}}'\mathbf{P}_{\boldsymbol{\theta}}(m-1,\boldsymbol{\theta}(m-1))\overline{\boldsymbol{\Xi}}'^{\mathrm{T}}+\overline{\boldsymbol{\Xi}}'\mathbf{A}_{\boldsymbol{\theta}\boldsymbol{\theta}}\overline{\boldsymbol{\Xi}}'^{\mathrm{T}}\right]^{-1}\\ &\left.\left.\overline{\boldsymbol{\Xi}}'\mathbf{A}_{\boldsymbol{\theta}\mathbf{W}}\right]\boldsymbol{\Xi}^{\mathrm{T}}+\boldsymbol{\Xi}\,\mathbf{B}\,\boldsymbol{\Xi}^{\mathrm{T}}\right)^{-1}\end{aligned} \tag{33g}$$

Remark on the notation of the inverse schur complement: The argument in $\tilde{\mathbf{P}}_{\Delta\mathbf{T}}\left(m\,|\,\mathbf{W}^{\ell-1}(m)\right)$ means that it is calculated at block m and *taking into account* the current data term $\tilde{\mathcal{J}}_0(m,\mathbf{W}^{\ell-1}(m))$ of the cost function, based on all previous blocks, up to and *including* the current block m with coefficient matrix $\mathbf{W}^{\ell-1}(m)$. [In addition to this *a-posteriori inverse schur complement* we will later in Sect. IV-C also introduce an *a-priori inverse schur complement* which does *not* take into account the data term $\tilde{\mathcal{J}}_0$ of the current block m but only a model-based prediction instead.]

Note that so far the update equations (32), (33) are valid for arbitrary cost functions and for arbitrary state-space models [except the initialization step (30)] and on arbitrary partly smooth manifolds. As a simple illustrative example, the conventional memoryless regularizers, e.g., the ℓ_{pq}-based regularizers according to [21], follow as special case for $\mathbf{A}_{\boldsymbol{\theta}\mathbf{W}} = \mathbf{0}$ (i.e., memoryless) and $\boldsymbol{\Xi} = \mathbf{I}$ (i.e., Euclidean) so that $\tilde{\mathbf{P}}_{\Delta\mathbf{T}}\left(m\,|\,\mathbf{W}^{\ell-1}(m)\right) = \left(\mathbf{A}_{\mathbf{WW}}+\mathbf{B}\right)^{-1}$. In the following subsections, we show some important step-by-step specializations for dynamical state-space models.

B. Generic Newton Algorithm with 1st-order Nonlinear and Additive Nongaussian Regularizer

We now specialize the above generic algorithm by applying the regularizer (9) based on the nonlinear dynamical state-space model, while the manifold $\mathcal{M}$ is still assumed to be arbitrary partly smooth. By a somewhat tedious but straightforward calculation we obtain in this case for (33b)-(33d)

$$\begin{aligned}\mathbf{A}_{\boldsymbol{\theta}\boldsymbol{\theta}} &= \alpha\left(\mathbf{I}\otimes\mathbf{K}^{\mathrm{T}}_{\mathcal{SC}}\right)\mathbf{W}^{01}_{mP^2L\times P^2L}\mathbf{G}^{\mathrm{T}}(m-1)\\ &\quad\mathbf{F}(m-1)\mathbf{G}(m-1)\mathbf{W}^{01}_{P^2L\times mP^2L}\left(\mathbf{I}\otimes\mathbf{K}_{\mathcal{SC}}\right),\end{aligned} \tag{34a}$$

$$\begin{aligned}\mathbf{A}_{\boldsymbol{\theta}\mathbf{W}} &= -\alpha\left(\mathbf{I}\otimes\mathbf{K}^{\mathrm{T}}_{\mathcal{SC}}\right)\mathbf{W}^{01}_{mP^2L\times P^2L}\mathbf{G}^{\mathrm{T}}(m-1)\\ &\quad\mathbf{F}(m-1)\mathbf{K}_{\mathcal{SC}},\end{aligned} \tag{34b}$$

$$\mathbf{A}_{\mathbf{WW}} = \alpha\mathbf{K}^{\mathrm{T}}_{\mathcal{SC}}\mathbf{F}(m-1)\mathbf{K}_{\mathcal{SC}}, \tag{34c}$$

respectively, where $\mathbf{F}$ denotes the Hessian of f, and $\mathbf{G}$ denotes the Jacobian of $\mathbf{g}$, i.e.,

$$\begin{aligned}&\mathbf{F}(m-1) =\\ &= \nabla_{\mathbf{z}}\nabla^{\mathrm{T}}_{\mathbf{z}}f(\mathbf{z})\big|_{\mathbf{z}=\mathrm{vec}\,\check{\mathbf{W}}(m)-\mathbf{g}\left(\mathbf{W}^{01}_{P^2L\times mP^2L}\,\mathrm{vec}\,\check{\boldsymbol{\theta}}(m-1),\mathbf{u}(m-1)\right)},\end{aligned} \tag{35a}$$

$$\mathbf{G}(m-1) = \nabla_{\mathbf{z}}\mathbf{g}^{\mathrm{T}}(\mathbf{z},\mathbf{u}(m-1))\big|_{\mathbf{z}=\mathbf{W}^{01}_{P^2L\times mP^2L}\,\mathrm{vec}\,\check{\boldsymbol{\theta}}(m-1)}. \tag{35b}$$

It should be emphasized that this result also contains the *Gauss-Markov regularizer* as a special case for the functions $\mathbf{g}(\cdot,\cdot)$ and $f(\cdot)$ according to (10) leading to the particularly simple expressions

$$\mathbf{F}(m-1) = \mathbf{Q}^{-1}(m-1), \tag{36a}$$
$$\mathbf{G}(m-1) = \mathbf{A}(m-1). \tag{36b}$$

C. Generic Algorithm in Prediction-Correction Formulation

The specialization (34a)-(34c) and the block-matrix relationship

$$\overline{\boldsymbol{\Xi}}' \left(\mathbf{I} \otimes \mathbf{K}_{\mathcal{SC}}^{\mathrm{T}}\right) \mathbf{W}_{mP^2L\times P^2L}^{01} = \mathbf{W}_{mP^2L\times P^2L}^{01} \boldsymbol{\Xi}' \mathbf{K}_{\mathcal{SC}}^{\mathrm{T}}, \tag{37}$$

with the further short-hand notation

$$\boldsymbol{\Xi}' = \boldsymbol{\Xi}\left(m-1, \mathbf{W}^{\ell-1}(m-1)\right),$$

allow us to reformulate the inverse Schur complement (33g) in a more familiar and efficient version using the matrix inversion lemma:

$$\begin{aligned}\tilde{\mathbf{P}}_{\Delta\mathbf{T}}^{-1}\left(m \,|\, \mathbf{W}^{\ell-1}(m)\right) = \\ = \boldsymbol{\Xi}\,\mathbf{K}_{\mathcal{SC}}^{\mathrm{T}}\Big[\frac{1}{\alpha}\mathbf{F}^{-1}(m-1) + \frac{1}{\lambda}\mathbf{G}(m-1)\,\mathbf{K}_{\mathcal{SC}}\,\boldsymbol{\Xi}'^{\mathrm{T}} \\ \tilde{\mathbf{P}}_{\Delta\mathbf{T}}\left(m-1 \,|\, \mathbf{W}^{\ell-1}(m-1)\right) \\ \boldsymbol{\Xi}'\,\mathbf{K}_{\mathcal{SC}}^{\mathrm{T}}\,\mathbf{G}^{\mathrm{T}}(m-1)\Big]^{-1}\mathbf{K}_{\mathcal{SC}}\,\boldsymbol{\Xi}^{\mathrm{T}} \\ +\alpha\,\boldsymbol{\Xi}\,\nabla_{\mathrm{vec}\,\mathbf{W}}\nabla_{\mathrm{vec}\,\mathbf{W}}^{\mathrm{T}}\tilde{\mathcal{J}}_0\left(m, \mathbf{W}^{\ell-1}(m)\right)\,\boldsymbol{\Xi}^{\mathrm{T}}. \end{aligned} \tag{39}$$

Taking a closer look at (39), we see that the expression within the square brackets only depends on the quantities $\mathbf{F}$ and $\mathbf{G}$ of the dynamical model-based regularizer, and *only* this expression, while the remaining part of (39) contains a *correction* by the current data term $\tilde{\mathcal{J}}_0$ of the cost function. We therefore split this equation into two equations, and denote the expression within brackets as the *a-priori inverse Schur complement* $\tilde{\mathbf{P}}_{\Delta\check{\mathbf{T}}}\left(m \,|\, \mathbf{W}^{\ell-1}(m-1)\right)$. We can interpret it as a *prediction step*, purely based on the dynamical state-space model.

Analogously, a similar prediction-correction pair is given for the filter coefficient matrix $\check{\mathbf{W}}$ by the purely model-based initialization step (30) and the filter update (32) / (24), respectively.

In the following pseudocode we summarize the generic Bayesian TRINICON algorithm on arbitrary partly smooth manifolds, based on the nonlinear state space model (4) with multiple iterations in Prediction-Correction formulation. It should be noted that this algorithm can be further developed into a Prediction-Innovation-Gain-Correction formulation analogously to the known classical Kalman-type tracking algorithms (see [17] for details).

Definitions:

$$\mathbf{F}(m-1) = \nabla_{\mathbf{z}}\nabla_{\mathbf{z}}^{\mathrm{T}} f(\mathbf{z})\Big|_{\mathbf{z}=\mathrm{vec}\,\check{\mathbf{W}}(m)-\mathbf{g}\left(\mathrm{vec}\,\check{\mathbf{W}}(m-1),\mathbf{u}(m-1)\right)}$$
$$\mathbf{G}(m-1) = \nabla_{\mathbf{z}}\mathbf{g}^{\mathrm{T}}(\mathbf{z}, \mathbf{u}(m-1))\Big|_{\mathbf{z}=\mathrm{vec}\,\check{\mathbf{W}}(m-1)}$$
$$\boldsymbol{\Xi}' = \boldsymbol{\Xi}\,(m-1, \mathbf{W}(m-1)) = \left.\frac{\partial \mathrm{vec}^{\mathrm{T}}\left(\boldsymbol{\varphi}_{\mathbf{W}(m-1)}\right)}{\partial \mathrm{vec}\,(\Delta\mathbf{T}(m-1))}\right|_{\Delta\mathbf{T}=\mathbf{0}}$$

Predictions:

$$\mathrm{vec}\,\check{\mathbf{W}}^0(m) = \mathbf{g}\left(\mathrm{vec}\,\check{\mathbf{W}}(m-1), \mathbf{u}(m-1)\right)$$

$$\begin{aligned}\tilde{\mathbf{P}}_{\Delta\check{\mathbf{T}}}\left(m \,|\, \mathbf{W}(m-1)\right) = \\ = \frac{1}{\alpha}\mathbf{F}^{-1}(m-1) + \frac{1}{\lambda}\mathbf{G}(m-1)\,\mathbf{K}_{\mathcal{SC}}\,\boldsymbol{\Xi}'^{\mathrm{T}} \\ \tilde{\mathbf{P}}_{\Delta\mathbf{T}}\left(m-1 \,|\, \mathbf{W}(m-1)\right)\,\boldsymbol{\Xi}'\,\mathbf{K}_{\mathcal{SC}}^{\mathrm{T}}\,\mathbf{G}^{\mathrm{T}}(m-1)\end{aligned}$$

Corrections:

For $\ell = 1, \ldots, \ell_{\max}$:

$$\boldsymbol{\Xi} = \boldsymbol{\Xi}\left(m, \mathbf{W}^{\ell-1}(m)\right) = \left.\frac{\partial \mathrm{vec}^{\mathrm{T}}\left(\boldsymbol{\varphi}_{\mathbf{W}^{\ell-1}(m)}\right)}{\partial \mathrm{vec}\,(\Delta\mathbf{T}(m))}\right|_{\Delta\mathbf{T}(m)=\mathbf{0}}$$

$$\begin{aligned}\tilde{\mathbf{P}}_{\Delta\mathbf{T}}^{-1}\left(m \,|\, \mathbf{W}^{\ell-1}(m)\right) = \boldsymbol{\Xi}\,\mathbf{K}_{\mathcal{SC}}^{\mathrm{T}}\tilde{\mathbf{P}}_{\Delta\check{\mathbf{T}}}^{-1}\left(m \,|\, \mathbf{W}(m-1)\right)\mathbf{K}_{\mathcal{SC}}\,\boldsymbol{\Xi}^{\mathrm{T}} \\ +\alpha\,\boldsymbol{\Xi}\,\nabla_{\mathrm{vec}\,\mathbf{W}}\nabla_{\mathrm{vec}\,\mathbf{W}}^{\mathrm{T}}\tilde{\mathcal{J}}_0\left(m, \mathbf{W}^{\ell-1}(m)\right)\,\boldsymbol{\Xi}^{\mathrm{T}}\end{aligned}$$

$$\begin{aligned}\mathrm{vec}\Delta\mathbf{T}^{\ell}(m) = -\boldsymbol{\mu}\tilde{\mathbf{P}}_{\Delta\mathbf{T}}\left(m \,|\, \mathbf{W}^{\ell-1}(m)\right)\sum_{i=0}^{\infty}\beta(i,m)\,\boldsymbol{\Xi} \\ \cdot\,\mathrm{vec}\left\{\nabla_{\mathbf{W}(m)}\tilde{\mathcal{J}}_0(m, \mathbf{W}(m))\right\}\Big|_{\mathbf{W}=\mathbf{W}^{\ell-1}(m)}\end{aligned}$$

$$\check{\mathbf{W}}^{\ell}(m) = \mathcal{SC}\left\{\boldsymbol{\varphi}_{\mathbf{W}^{\ell-1}(m)}\left(\Delta\mathbf{T}^{\ell}(m)\right)\right\}$$

endfor

$$\check{\mathbf{W}}(m) = \check{\mathbf{W}}^{\ell_{\max}}(m)$$
$$\tilde{\mathbf{P}}_{\Delta\mathbf{T}}\left(m \,|\, \mathbf{W}(m)\right) = \tilde{\mathbf{P}}_{\Delta\mathbf{T}}\left(m \,|\, \mathbf{W}^{\ell_{\max}-1}(m)\right)$$

V. Application to broadband BSS for time-varying convolutive mixing systems

To evaluate the effectiveness of the Bayesian TRINICON approach with state-space model, we consider blind source separation of speech signals in a time-varying reverberant environment. Our measurements were performed in a real office environment (with curtains and carpet). The reverberation time T_{60} was approximately 300ms and the sampling rate was 16kHz. We placed two (spatially fixed) speech sources (using loudspeakers) and two (spatially fixed) microphones in this room. To simulate the time-variance of the 2×2 acoustic mixing system in a reproducible way, we placed a rotating panel ($\approx 2m^2$) on a stepper motor nearby the microphones. Using the stepper motor, this panel was rotated in 0.2^0-steps so that in total, we obtained a set of 900 measured 2×2 MIMO systems for the entire rotation by 180^0.

We evaluate the BSS separation performance in terms of the signal-to-interference ratio (SIR) improvements on the outputs of the demixing system, or, more precisely, the arithmetic average between the SIR improvements in the two output signals $y_1(n)$ and $y_2(n)$. Note that the precise SIR improvements can be calculated only in simulations when the measured *mixing* systems are available as a reference. Hence, in the case of time-varying mixing systems we used our entire set of 900 measured MIMO systems as described above in order to calculate the SIR improvements, i.e., to show the tracking performance during the online BSS operation.

Figure 3 shows the SIR improvements obtained by two different algorithms in the time-varying environment after

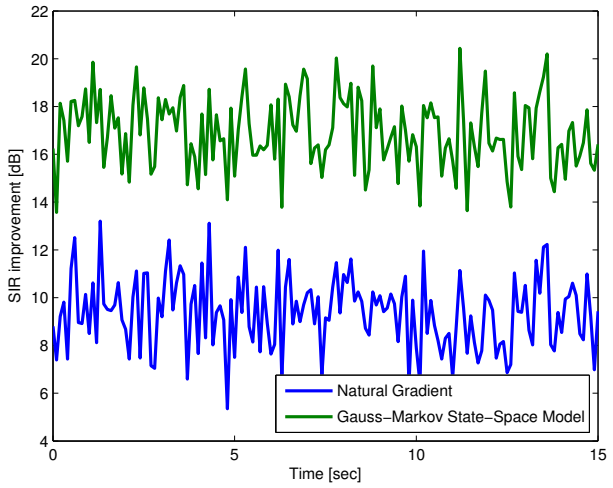

Fig. 3. Simulation results for blind source separation in a continuously time-varying environment.

reaching their steady-state performance (i.e., after their initial convergence). The lower (blue) SIR improvement curve shows the performance of the natural gradient-based broadband BSS algorithm (with a fixed Gaussian prior for the demixing system). We regard this as a state-of-the-art baseline algorithm (e.g., [4]–[6]). The natural manifold is defined by the map [9]

$$\varphi_{\mathbf{W}^{\ell-1}}(\Delta\mathbf{T}) = \mathbf{W}^{\ell-1} + \mathbf{W}^{\ell-1}\Delta\mathbf{T}. \tag{46}$$

The upper (green) SIR improvement curve was obtained using a corresponding broadband BSS algorithm with a Gauss-Markov state-space model leading to the novel blind Kalman-like BSS algorithm (also on the natural manifold defined by (46)). In this case, the Kalman gain computation (including the inverse Hessian) was performed in the frequency domain for computational efficiency.

VI. Conclusions

In this paper we have presented a unified Bayesian adaptive signal processing and tracking framework. The resulting SIR performance for the BSS example in time-varying environments shows a clear improvement over the baseline algorithm.

Appendix A
Block Matrix Inversion

For conformably partitioned 2-by-2 block matrices, i.e., all submatrices exhibit compatible, the following relation holds (under the assumption of invertibility of $\mathbf{A}_{11}$ and $\mathbf{A}_{22}$):

$$\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{F}_{11}^{-1} & -\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\mathbf{F}_{22}^{-1} \\ -\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{F}_{11}^{-1} & \mathbf{F}_{22}^{-1} \end{bmatrix} \tag{47}$$

where

$$\mathbf{F}_{11} = \mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}, \tag{48}$$

$$\mathbf{F}_{22} = \mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12} \tag{49}$$

are the so-called Schur complements.

References

[1] Y. Bar-Shalom, P.K. Willett, and X. Tian, *Tracking and Data Fusion: A Handbook of Algorithms*, YBS Publishing, 2011.

[2] A.J. Haug, *Bayesian Estimation and Tracking: A Practical Guide*, John Wiley & Sons, New York, NY, 2012.

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

[4] H. Buchner, R. Aichner, and W. Kellermann, "TRINICON-based blind system identification with application to multiple-source localization and separation," in *Blind Speech Separation*, S. Makino, T.-W. Lee, and S. Sawada, Eds., pp. 101–147. Springer, Berlin, Sept. 2007.

[5] H. Buchner, R. Aichner, and W. Kellermann, "TRINICON: A versatile framework for multichannel blind signal processing," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, Montreal, Canada, May 2004, vol. 3, pp. 889–892.

[6] H. Buchner, R. Aichner, and W. Kellermann, "Blind source separation for convolutive mixtures: A unified treatment," in *Audio Signal Processing for Next-Generation Multimedia Communication Systems*, J. Benesty and Y. Huang, Eds., pp. 255–293. Kluwer Academic Publishers, Boston, Apr. 2004.

[7] H. Buchner and W. Kellermann, "A fundamental relation between blind and supervised adaptive filtering illustrated for blind source separation and acoustic echo cancellation," in *Proc. Joint Workshop on Hands-Free Speech Communication and Microphone Arrays (HSCMA)*, Trento, Italy, May 2008.

[8] S. Haykin, *Adaptive Filter Theory*, Prentice Hall Inc., Englewood Cliffs, NJ, 4th edition, 2002.

[9] H. Buchner, "A systematic approach to incorporate deterministic prior knowledge in broadband adaptive MIMO systems," in *Proc. Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, USA, Nov. 2010.

[10] W.M. Boothby, *An Introduction to Differentiable Manifolds and Riemannian Geometry*, Academic Press, New York, 1986.

[11] M. Spivak, *Calculus on Manifolds: A Modern Approach to Classical Theorems of Advanced Calculus*, Harper Collins Publishers, 1965.

[12] S.-I. Amari, A. Cichocki, and H.H. Yang, "A new learning algorithm for blind signal separation," *Advances in neural information systems, 8*, pp. 757–763, 1996.

[13] H. Buchner and W. Kellermann, "TRINICON for dereverberation of speech and audio signals," in *Speech Dereverberation*, P.A. Naylor and N.D. Gaubitch, Eds., pp. 311–385. Springer, London, Jul. 2010.

[14] T.N. Thiele, "Om anvendelse af mindste kvadraters methode i nogle tilfaelde, hvor en komplikation af visse slags uensartede tilfaldige fejlkilder giver fejlene en 'systematisk' karakter," *Det Kongelige Danske Videnskabernes Selskab, naturvidenskabelig og matematisk Afdeling*, vol. 12, pp. 381–408, 1880, In Danish. (On the application of the method of least squares in such cases where certain kinds of random sources of errors give the errors a 'systematic' character.).

[15] R.E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME*, vol. 82, pp. 35–45, 1960.

[16] G. Enzner, H. Buchner, A. Favrot, and F. Kuech, "Acoustic echo control," in *Academic Press Library in Signal Processing*, R. Chellappa and S. Theodoridis, Eds. Elsevier/Academic Press, 2014.

[17] H. Buchner, K. Helwani, and S. Godsill, "Blind signal processing for time-varying convolutive mixing systems based on sequence estimation on partly smooth manifolds," *ArXiv*, 2018.

[18] K.H. Knuth, "A bayesian approach to source separation," in *Proc. Int. Conf. on Independent Component Analysis and Blind Signal Separation (ICA)*, Jan. 1999, pp. 283–288.

[19] R. Choudrey and S.J. Roberts, "Learning hierarchical dynamics using independent component analysis," in *Proc. Int. Symposium on Independent Component Analysis and Blind Signal Separation (ICA)*, Nara, Japan, Apr. 2003, pp. 797–802.

[20] A.H. Jazwinski, *Stochastic Processes and Filtering Theory*, Academic Press, San Diego, CA, 1970.

[21] K. Helwani, H. Buchner, and S. Spors, "Multichannel adaptive filtering with sparseness constraints," in *Proc. Int. Workshop on Acoustic Echo and Noise Control (IWAENC)*, Aachen, Germany, Sep. 2012.