A Systematic Approach to Incorporate Deterministic Prior Knowledge in Broadband Adaptive MIMO Systems

Herbert Buchner
Deutsche Telekom Laboratories, Technische Universität Berlin, Ernst-Reuter-Platz 7, 10587 Berlin, Germany
E-Mail: hb@buchner-net.com

***Abstract*—Various approaches for incorporating prior system knowledge into adaptive filtering algorithms exist, e.g., using constrained adaptation. Moreover, also the basic setup of the adaptation problem, e.g., whether it is supervised or blind, can be considered as prior system knowledge. In this paper, we consider a systematic approach to incorporate such deterministic prior knowledge in broadband adaptive MIMO systems by optimizing the coefficients in arbitrary partly smooth manifolds. The resulting generic set of update equations explicitly shows all the available degrees of freedom for a top-down algorithm design. Using practically relevant examples, we show how both well-known and novel algorithms for various applications can be derived using the framework.**

I. Introduction

In adaptive filtering, a set of coefficients in the form of a vector or a matrix is continuously optimized based on received input signals, certain requirements on the desired output signals, and a certain cost function. The particular form of the requirements on the desired output signals and the choice of the cost function generally depend on the type of adaptive filter problem, e.g., supervised adaptive filtering directly utilizes given desired output signals, e.g., [1], while in blind problems we have to resort to stochastic requirements, such as the mutual statistical independence between the output signals for blind source separation (BSS), e.g., [2], [3]. Without any further constraints, the associated search process for the optimal coefficients is typically performed in the Euclidean space, i.e., the gradient directly corresponds to the partial derivatives w.r.t. the filter coefficients, e.g., [1]. Based on the gradient of the cost function, gradient descent adaptation is one of the simplest techniques for adaptive filtering. On the other hand, more advanced adaptation schemes are given by Newton-type algorithms which in general also require the second derivatives of the cost function. Most of the well known adaptation algorithms (including the gradient-descent-based adaptation) can be deduced as approximations of the Newton algorithm.

In adaptive filtering, a variety of different ways to incorporate certain prior knowledge on the system into the adaptation process have been proposed in order to

- constrain the search space according to the requirements of the desired application,
- increase the convergence speed,
- reduce the computational complexity,

or a combination thereof. Some popular examples are the incorporation of linear constraints, e.g., as used in adaptive beamforming [4], [5], [6], the use of the so-called natural gradient in the context of adaptive multiple-input and multiple output (MIMO) filtering [7], or adaptive filtering in transform domains, e.g., the DFT domain, which results from the assumption of an FIR structure [8]. Moreover, it can be shown that the class of supervised adaptive filtering algorithms indeed follows as a special case of the more general broadband adaptive MIMO algorithms originally introduced for blind adaptation problems, if we incorporate additional knowledge on a specialized system structure [9].

In this paper, we consider the adaptation in arbitrary partly smooth manifolds which promises to be one of the most general approaches to incorporate prior knowledge on the search space. Our considerations are based on TRINICON ('TRIple-N ICA for CONvolutive mixtures'), a previously introduced generic concept for broadband adaptive MIMO filtering, e.g., [10], [3], [11], using the technique of independent component analysis (ICA), e.g., [2]. Based on TRINICON, which has so far been presented mainly for the Euclidean search space (with, e.g., the natural gradient and the adaptation in the DFT domain etc. as additional, but more or less heuristically motivated extensions), we derive in this paper a generic set of broadband MIMO update equations in arbitrary partly smooth manifolds. In general, a manifold is a topological space that is *locally* Euclidean [12], [13]. Smooth manifolds (also called differentiable manifolds) are manifolds for which overlapping *charts*, i.e., local parameterizations, relate smoothly to each other. (A smooth manifold with a metric in order to measure lengths and angles is called Riemannian manifold.) In order to devise the generic equations on a certain manifold $\mathcal{M}$, we equip the TRINICON optimization criterion with certain local parameterizations leading to the corresponding local optimization criteria in the Euclidean tangent space. Such local parameterizations always exist since a manifold consists of subsets of the multidimensional real space glued together. We then derive a local Newton step based on the TRINICON criterion in the Euclidean tangent space. The new coefficient matrix on the manifold $\mathcal{M}$ after the adaptation step is then obtained by again applying the local mapping function.

As we will see, the various degrees of freedom in order to select both well known and novel improved algorithms, such as the so-called Sylvester constraint [11], generalize well to the case of arbitrary partly smooth manifolds. As simple, but practically important examples, we will show that the above-mentioned examples, i.e., adaptation with linear constraints and adaptive beamforming, the natural gradient, and the supervised adaptive filtering algorithms indeed follow rigorously as special cases of the novel generic algorithm. A significant advantage of this deductive approach is that by relatively simple specializations, we immediately have all of the different techniques for incorporating prior system knowledge available for the various adaptive filtering applications, i.e., for both supervised and blind adaptive filtering problems. The systematic approach also greatly simplifies the integration of different techniques into a common adaptation algorithm.

II. General MIMO Setup and Notation

In broadband signal aquisition by sensor arrays, such as in hands-free speech communication scenarios, the original source signals $s_q(n)$, $q = 1, \ldots, Q$ are filtered by a linear MIMO system (e.g., a reverberant room) before they are captured as sensor signals $x_p(n)$, $p = 1, \ldots, P$. In this paper, we describe this 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 $x_p(n)$ according to Fig. 1. Moreover, we assume throughout this paper that $Q \leq P$. According to a certain optimization criterion, we are interested in finding a corresponding length-L FIR demixing system with coefficients $w_{pq,\kappa}$ by adaptive signal processing. This yields the output signals $y_q(n)$. As a compact

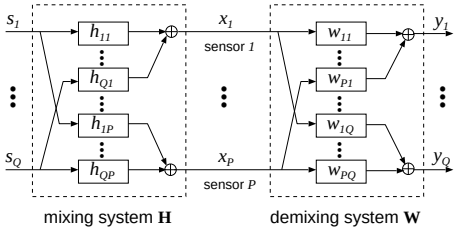

Fig. 1. General setup for MIMO signal processing.

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

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

and the corresponding $QM \times P$ mixing coefficient matrix $\breve{\mathbf{H}}$, 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.

III. A BRIEF RECAPITULATION OF TRINICON AND COEFFICIENT UPDATES IN THE EUCLIDEAN SPACE

In this section we first give a brief overview of the essential elements of TRINICON for the coefficient adaptation. Thereby, we restrict the presentation here to time-domain coefficient updates in the Euclidean space.

A. Optimization Criterion

Various approaches exist to estimate the demixing matrix $\breve{\mathbf{W}}$ by utilizing the following fundamental source signal properties [2] which were all combined in TRINICON:

(i) Nongaussianity is exploited by using higher-order statistics for ICA. The minimization of the mutual information (MMI) among the output channels can be regarded as the most general approach to separation problems [2]. To obtain an estimator that is also suitable for inverse problem, TRINICON uses the Kullback-Leibler divergence (KLD) [14] between a certain *desired* joint pdf (essentially representing a hypothesized stochastic source model as shown below) and the joint pdf of the actually estimated output signals.

(ii) Nonwhiteness is exploited by simultaneous minimization of output cross-relations over multiple time-lags. We therefore consider multivariate pdfs, i.e., 'densities covering D time-lags'.

(iii) Nonstationarity is exploited by simultaneous minimization of output cross-relations at different time-instants. We assume ergodicity within blocks of length N so that the ensemble average is replaced by time averages over these blocks.

Throughout this section, we formulate the framework for $Q = P$ sources without loss of generality. In practice, the current number of simultaneously active sources is allowed to vary throughout the application and only the conditions $Q \leq P$ (for separation only) and $Q < P$ (for deconvolution), respectively, have to be fulfilled.

To introduce an algorithm for broadband processing of convolutive mixtures, we first formulate the convolution of the FIR demixing system of length L in the following matrix form [10]:

$$\mathbf{y}(n) = \mathbf{W}^{\mathrm{T}}\mathbf{x}(n), \tag{4}$$

where n denotes the time index, and

$$\mathbf{x}(n) = [\mathbf{x}_1^{\mathrm{T}}(n), \ldots, \mathbf{x}_P^{\mathrm{T}}(n)]^{\mathrm{T}}, \tag{5}$$

$$\mathbf{y}(n) = [\mathbf{y}_1^{\mathrm{T}}(n), \ldots, \mathbf{y}_P^{\mathrm{T}}(n)]^{\mathrm{T}}, \tag{6}$$

$$\mathbf{x}_p(n) = [x_p(n), \ldots, x_p(n-2L+1)]^{\mathrm{T}}, \tag{7}$$

$$\mathbf{y}_q(n) = [y_q(n), \ldots, y_q(n-D+1)]^{\mathrm{T}}. \tag{8}$$

The parameter D in (8), $1 \leq D < L$, denotes the number of time lags taken into account to exploit the nonwhiteness of the source signals as shown below. $\mathbf{W}_{pq}$, $p = 1, \ldots, P$, $q = 1, \ldots, P$ 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 $\mathbf{W}_{pq}$.

Based on the KLD, the following cost function was introduced in [10] taking into account all three fundamental signal properties (i)-(iii):

$$\mathcal{J}(m, \mathbf{W}) = -\sum_{i=0}^{\infty} \beta(i,m) \frac{1}{N} \cdot \sum_{j=iN_L}^{iN_L+N-1} \left\{ \log(\hat{p}_{s,PD}(\mathbf{y}(j))) - \log(\hat{p}_{y,PD}(\mathbf{y}(j))) \right\}, \tag{9}$$

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 L samples relatively to the previous block. Furthermore, β is a window function allowing for online, offline, or block-online algorithms [3].

An *alternative formulation of the second term in the optimization criterion* (9) is obtained by using the mapping between the output pdf and the input pdf of the demixing filter which plays an important role for the following considerations in this paper. This mapping can be expressed as follows, e.g., [15]:

$$\hat{p}_{y,PD}(\mathbf{y}) = \frac{\hat{p}_{x_{PD},PD}(\mathbf{x}_{PD})}{|\det\{\mathbf{V}^{\mathrm{T}}\mathbf{W}\}|} \tag{10}$$

with the window matrix $\mathbf{V} = \mathrm{Bdiag}\{\tilde{\mathbf{V}}, \ldots, \tilde{\mathbf{V}}\}$, where $\tilde{\mathbf{V}} = \left[\mathbf{I}_{D\times D},\ \mathbf{0}_{D\times(2L-D)}\right]^{\mathrm{T}}$.

B. Euclidean Gradient-Based Coefficient Updates

In this subsection we concentrate on iterative Euclidean gradient-based block-online coefficient updates which can be written in the general form

$$\breve{\mathbf{W}}^0(m) := \breve{\mathbf{W}}(m-1), \tag{11a}$$

$$\breve{\mathbf{W}}^\ell(m) = \breve{\mathbf{W}}^{\ell-1}(m) - \mu\Delta\breve{\mathbf{W}}^\ell(m),\ \ \ell = 1, \ldots, \ell_{\max}, \tag{11b}$$

$$\breve{\mathbf{W}}(m) := \breve{\mathbf{W}}^{\ell_{\max}}(m), \tag{11c}$$

where μ is a stepsize parameter, and the superscript index ℓ denotes an iteration parameter to allow for multiple iterations ($\ell = 1, \ldots, \ell_{\max}$)

within each block m. The matrix $\check{\mathbf{W}}$ consists of the first column of each submatrix $\mathbf{W}_{pq}$ without the L zeros.

Obviously, when calculating the gradient of $\mathcal{J}(m,\mathbf{W})$ w.r.t. $\check{\mathbf{W}}$ explicitly, we are confronted with the problem of the different matrix formulations $\mathbf{W}$ and $\check{\mathbf{W}}$. The larger dimensions of $\mathbf{W}$ are a direct consequence of taking into account the nonwhiteness signal property by choosing $D > 1$. The rigorous distinction between these different matrix structures 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 [3]. Using the Sylvester constraint operator the gradient descent update can be written as

$$\Delta\check{\mathbf{W}}^{\ell}(m) = \mathcal{SC}\left\{\nabla_{\mathbf{W}}\mathcal{J}(m,\mathbf{W})\right\}|_{\mathbf{W}=\mathbf{W}^{\ell}(m)}. \tag{12}$$

Depending on the particular realization of ($\mathcal{SC}$), we are able to select both, well known and also novel improved adaptation algorithms [15]. In [11] an explicit formulation of a *generic* Sylvester constraint was derived based on the chain rule to further formalize and clarify this concept:

$$\left[\Delta\check{\mathbf{w}}_{pq}^{\ell}(m)\right]_i = \sum_{k,j}\left[\Delta\mathbf{W}_{pq}^{\ell}(m)\right]_{kj}\delta_{k,(i+j-1)}. \tag{13}$$

Here, δ_{ab} denotes the Kronecker symbol.

It can be shown [15] that by taking the gradient of $\mathcal{J}(m)$ with respect to the demixing filter matrix $\check{\mathbf{W}}(m)$ according to (12), we obtain the following generic gradient descent-based TRINICON update rule in the Euclidean space:

$$\Delta\check{\mathbf{W}}^{\ell}(m) = \frac{1}{N}\sum_{i=0}^{\infty}\beta(i,m)\,\mathcal{SC}\left\{\sum_{j=iN_L}^{iN_L+N-1}\left[\mathbf{x}(j)\boldsymbol{\Phi}_{s,PD}^{\mathrm{T}}(\mathbf{y}(j)) - \left(\left(\mathbf{W}^{\ell-1}(m)\right)^{\mathrm{T}}\right)^{+}\right]\right\}, \tag{14a}$$

with $\cdot^{+}$ denoting the pseudoinverse of a matrix, and with the generalized score function

$$\boldsymbol{\Phi}_{s,PD}(\mathbf{y}(j)) = -\frac{\partial\log\hat{p}_{s,PD}(\mathbf{y}(j))}{\partial\mathbf{y}(j)} - \frac{1}{N}\sum_{r}\sum_{i_1,i_2,\ldots}\frac{\partial\mathcal{G}_{s,i_1,i_2,\ldots}^{(r)}}{\partial\mathbf{y}}\sum_{j=iN_L}^{iN_L+N-1}\frac{\partial\log\hat{p}_{s,PD}}{\partial\mathcal{Q}_{s,i_1,i_2,\ldots}^{(r)}} \tag{14b}$$

resulting from the hypothesized source model $\hat{p}_{s,PD} = \hat{p}_{s,PD}(\mathbf{y},\boldsymbol{\mathcal{Q}}_s^{(1)},\boldsymbol{\mathcal{Q}}_s^{(2)},\ldots)$ with certain stochastic model parameters $\boldsymbol{\mathcal{Q}}_s^{(r)}$, $r = 1,2,\ldots$ (the calligraphic symbols denote multidimensional arrays) given by their elements $\mathcal{Q}_{s,i_1,i_2,\ldots}^{(r)}$ in the generic form $\mathcal{Q}_{s,i_1,i_2,\ldots}^{(r)}(i) = \frac{1}{N}\sum_{j=iN_L}^{iN_L+N-1}\left\{\mathcal{G}_{s,i_1,i_2,\ldots}^{(r)}(\mathbf{y}(j))\right\}$ with certain nonlinear functions $\mathcal{G}_{s,i_1,i_2,\ldots}^{(r)}(\mathbf{y})$, $r = 1,2,\ldots$. A well known special case of such a parameterization is the estimate of the correlation matrix $\mathbf{R}_{\mathbf{yy}}(i) = \frac{1}{N}\sum_{j=iN_L}^{iN_L+N-1}\left\{\mathbf{y}(j)\mathbf{y}^{\mathrm{T}}(j)\right\}$. The filter coefficients and the stochastic model parameters are estimated in an alternating way.

C. Euclidean Newton-Based Coefficient Updates

This subsection presents the main ideas of the extension of the previously described class of gradient-descent-based coefficient update rules to the so-called Newton-type update rules for TRINICON-based adaptation algorithms. The main advantage of the Newton-type adaptation algorithms is its *quadratic convergence rate* compared to the linear convergence rate of the gradient-based algorithms. In contrast to the gradient descent-based optimization approach, the Newton method directly aims at finding the numerical solution of a generally nonlinear set of equations. In our case, this set of equations is a matrix equation of the form

$$\mathbf{Q}\left(\check{\mathbf{W}}\right) = \mathbf{0}, \tag{15}$$

which arises from the condition

$$\nabla_{\check{\mathbf{W}}}\mathcal{J}\left(\mathbf{W}\right) = \frac{\partial\mathcal{J}\left(\mathbf{W}\right)}{\partial\check{\mathbf{W}}} = \mathbf{0} \tag{16}$$

for minimization of the TRINICON optimization criterion w.r.t. the filter coefficients. Analogously to the so-called normal equation for solving least-squares problems, we denote (16) as the *TRINICON normal equation.*

The Newton-Raphson method is based on a local linearization of the function $\mathbf{Q}\left(\check{\mathbf{W}}\right)$ after (15) using a Taylor expansion around a certain point $\check{\mathbf{W}}'$:

$$\mathbf{Q}\left(\check{\mathbf{W}}\right) = \mathbf{Q}\left(\check{\mathbf{W}}\right)\big|_{\check{\mathbf{W}}'} + \left[\frac{\partial}{\partial\check{\mathbf{W}}}\mathbf{Q}^{\mathrm{T}}\left(\check{\mathbf{W}}\right)\right]_{\check{\mathbf{W}}'}^{\mathrm{T}}\left(\check{\mathbf{W}}-\check{\mathbf{W}}'\right)+\ldots = \mathbf{0}. \tag{17}$$

Setting $\check{\mathbf{W}} = \check{\mathbf{W}}^{\ell}(m)$ and $\check{\mathbf{W}}' = \check{\mathbf{W}}^{\ell-1}(m)$, considering $\mathbf{Q} = \nabla_{\check{\mathbf{W}}}\mathcal{J}$, and neglecting the residual term, we obtain from (17) the iterative update rule

$$\check{\mathbf{W}}^{\ell}(m) = \check{\mathbf{W}}^{\ell-1}(m) - \left[\nabla_{\check{\mathbf{W}}}\nabla_{\check{\mathbf{W}}}^{\mathrm{T}}\mathcal{J}\left(m,\mathbf{W}^{\ell-1}(m)\right)\right]^{-1} \cdot\nabla_{\check{\mathbf{W}}}\mathcal{J}\left(m,\mathbf{W}^{\ell-1}(m)\right). \tag{18}$$

Comparing (18) with (11) and (12) we see that the Newton-Raphson update can be considered as an extension of the standard gradient-based update. The additional quantity $\nabla_{\check{\mathbf{W}}}\nabla_{\check{\mathbf{W}}}^{\mathrm{T}}\mathcal{J}\left(\mathbf{W}(m-1)\right)$ in the Newton-Raphson update is called the $Hessian$. Note that in general, the Hessian in (18) is described by a multidimensional array with four indices. To simplify the handling, we introduce the vec operator allowing an equivalent formulation involving only regular matrices. Moreover, after introducing a relaxation factor (or matrix in general) $\boldsymbol{\mu}$ and a regularization matrix $\boldsymbol{\delta}_{\mathbf{P}}$, we obtain the following general expression for the update:

$$\mathrm{vec}\check{\mathbf{W}}^{\ell}(m) = \mathrm{vec}\check{\mathbf{W}}^{\ell-1}(m) - \boldsymbol{\mu}(m,\ell)\left[\mathbf{P}_{\check{\mathbf{W}}}\left(m,\check{\mathbf{W}}^{\ell-1}(m)\right)+\boldsymbol{\delta}_{\mathbf{P}}(m,\ell)\right]^{-1} \cdot\mathrm{vec}\left(\nabla_{\check{\mathbf{W}}}\mathcal{J}\left(m,\mathbf{W}^{\ell-1}(m)\right)\right). \tag{19}$$

It can be shown that the Hessian can then be expressed as

$$\mathbf{P}_{\check{\mathbf{W}}}\left(m,\check{\mathbf{W}}^{\ell-1}(m)\right) = \sum_{i=0}^{\infty}\beta(i,m)\cdot\frac{1}{N}\sum_{j=iN_L}^{iN_L+N-1} \mathbf{K}_{\mathcal{SC}}(\mathbf{I}\otimes\mathbf{x}(j))\,\frac{\partial\left(\boldsymbol{\Phi}_{\mathrm{s},PD}^{\mathrm{T}}(\mathbf{y}(j))-\boldsymbol{\Phi}_{\mathrm{y},PD}^{\mathrm{T}}(\mathbf{y}(j))\right)}{\partial\mathbf{y}}\Big|_{\mathbf{W}=\mathbf{W}^{\ell-1}(m)}(\mathbf{I}\otimes\mathbf{x}^{\mathrm{T}}(j))\mathbf{K}_{\mathcal{SC}}^{\mathrm{T}}, \tag{20}$$

where the matrix $\mathbf{K}_{\mathcal{SC}}$ is a fixed matrix consisting of ones and zeros so that

$$\mathrm{vec}\check{\mathbf{W}} = \mathrm{vec}\,\mathcal{SC}\{\mathbf{W}\} = \mathbf{K}_{\mathcal{SC}}\,\mathrm{vec}\mathbf{W}. \tag{21}$$

IV. Example: TRINICON for Blind Source Separation

Broadband blind source separation algorithms constitute a class of algorithms that requires a minimum amount of prior knowledge on the involved source signal characteristics and on the MIMO mixing system. Hence, in this sense, the BSS algorithms can be regarded as prototype algorithms for the general class of separation and identification algorithms. In the later sections we will gradually introduce more prior knowlege on the system structure. In BSS, the aim is to

achieve statistical independence between the output channels. Hence, the desired pdf is factorized w.r.t. the output channels, i.e.,

$$\hat{p}_{s,PD}(\mathbf{y}(j)) \overset{(\mathrm{BSS})}{=} \prod_{q=1}^{P} \hat{p}_{y_q,D}(\mathbf{y}_q(j)), \tag{22}$$

so that the desired score function simplifies to

$$\boldsymbol{\Phi}_{s,PD}(\mathbf{y}) \overset{(\mathrm{BSS})}{=} \left[\boldsymbol{\Phi}_{1,D}^{\mathrm{T}}(\mathbf{y}_1), \ldots, \boldsymbol{\Phi}_{P,D}^{\mathrm{T}}(\mathbf{y}_P)\right]^{\mathrm{T}}. \tag{23}$$

In other words, for each output channel the score function can be obtained individually from a certain choice of pdf. For illustration, the special case of algorithms based on second-order statistics (SOS) is obtained from choosing multivariate Gaussian source models leading to [3]

$$\boldsymbol{\Phi}_{q,D}(\mathbf{y}_q(j)) = \mathbf{R}_{\mathbf{y}_q\mathbf{y}_q}^{-1}(i)\mathbf{y}_q(j). \tag{24}$$

V. General Coefficient Update on Arbitrary Partly Smooth Manifolds

In adaptive filtering the Euclidean geometry is by far the most widely assumed topology for the coefficient optimization, mostly due to its conceptual simplicity. However, other topologies may be more suitable to obtain a good match with the respective optimization problem at hand. The adaptation on arbitrary partly smooth manifolds considered in this section can be considered as one of the most general concepts. A manifold $\mathcal{M}$ is an arbitrary topological space that is locally Euclidean [12], [13]. Hence, local parameterizations in a Euclidean *tangent space* always exist. These local parameterizations are called *charts* or *maps*. Multiple maps glued together to form an arbitrary partly smooth manifold consisting of subsets of the multidimensional real space are also called *atlas*.

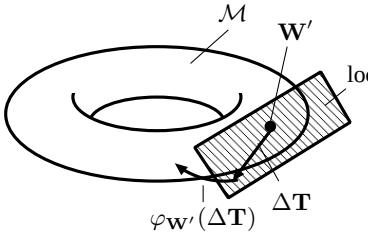

Fig. 2. Example for a two-dimensional manifold $\mathcal{M}$.

Potential advantages of choosing a suitable map (or, more generally, a suitable atlas) in the context of adaptive filtering are a further improvement of convergence speed and/or a decreased computational complexity. However, the most important aspect is that most - if not all - practically relevant types of deterministic prior system information, e.g., given by various types of constraints, transform domains, or specialized system structure, can be described by the adaptation on a suitably designed manifold. In particular, any constrained optimization problem in the Euclidean space can be thought of as unconstrained optimization problem on a special manifold.

Figure 3 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 systems we have to deal with matrix-valued manifolds.

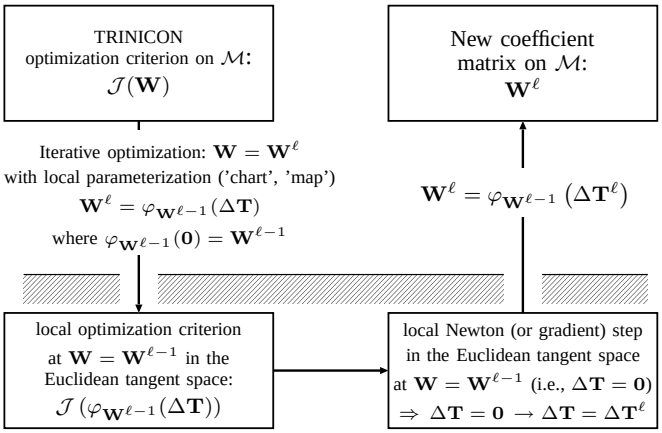

Fig. 3. Basic approach for TRINICON-based optimization on an arbitrary partly smooth manifold $\mathcal{M}$.

A. Normal Equation and Newton Update on Manifolds

In general, the optimization problem w.r.t. the coefficient matrix $\check{\mathbf{W}}$ on a manifold $\mathcal{M}$ can be expressed as

$$\check{\mathbf{W}} = \arg \min_{\check{\mathbf{W}} \in \mathcal{M}} \mathcal{J}\left(\check{\mathbf{W}}\right). \tag{25}$$

Using a *local* parameterization around a certain point $\check{\mathbf{W}}'$, i.e., using the matrix-valued map $\check{\boldsymbol{\varphi}}_{\check{\mathbf{W}}'}\left(\Delta\check{\mathbf{T}}\right)$, where $\check{\boldsymbol{\varphi}}_{\check{\mathbf{W}}'}(\mathbf{0}) = \check{\mathbf{W}}'$, this translates into the local optimization problem

$$\Delta\check{\mathbf{T}} = \arg \min_{\Delta\check{\mathbf{T}} \in \mathbb{R}^{L'P' \times Q'}} \mathcal{J}\left(\check{\boldsymbol{\varphi}}_{\check{\mathbf{W}}'}\left(\Delta\check{\mathbf{T}}\right)\right). \tag{26}$$

In an analogous way as shown in (17) for the Euclidean case, a locally applied Taylor approximation in the tangent space yields

$$\begin{aligned}\mathrm{vec}\left(\nabla_{\Delta\check{\mathbf{T}}}\mathcal{J}\left(\check{\boldsymbol{\varphi}}_{\check{\mathbf{W}}}(\Delta\check{\mathbf{T}})\right)\right) &= \mathrm{vec}\left(\nabla_{\Delta\check{\mathbf{T}}}\mathcal{J}\left(\check{\boldsymbol{\varphi}}_{\check{\mathbf{W}}'}(\Delta\check{\mathbf{T}})\right)\Big|_{\Delta\check{\mathbf{T}}=\mathbf{0}}\right) \\ &+ \left[\frac{\partial}{\partial \mathrm{vec}\left(\Delta\check{\mathbf{T}}\right)}\left(\frac{\partial\mathcal{J}\left(\check{\boldsymbol{\varphi}}_{\check{\mathbf{W}}'}(\Delta\check{\mathbf{T}})\right)}{\partial \mathrm{vec}\left(\Delta\check{\mathbf{T}}\right)}\right)^{\mathrm{T}}\right]^{\mathrm{T}}_{\Delta\check{\mathbf{T}}=\mathbf{0}} \mathrm{vec}\left(\Delta\check{\mathbf{T}}\right) = \mathbf{0}.\end{aligned} \tag{27}$$

Note that here, ∇ still denotes the *Euclidean* gradient (which exclusively consists of partial derivatives with uniform weights) since the tangent space is an affine space. From the second part of the representation (27) we readily obtain the following local Newton step in the tangent space of the manifold $\mathcal{M}$ at point $\check{\mathbf{W}}' = \check{\mathbf{W}}^{\ell-1}$:

$$\begin{aligned}\mathrm{vec}\Delta\check{\mathbf{T}}^{\ell}(m) &= -\mathbf{P}^{-1}_{\Delta\check{\mathbf{T}}}\left(m, \check{\mathbf{W}}^{\ell-1}(m)\right) \\ &\quad \mathrm{vec}\left(\nabla_{\Delta\check{\mathbf{T}}}\mathcal{J}\left(\check{\boldsymbol{\varphi}}_{\check{\mathbf{W}}^{\ell-1}(m)}(\Delta\check{\mathbf{T}})\right)\Big|_{\Delta\check{\mathbf{T}}=\mathbf{0}}\right)\end{aligned} \tag{28a}$$

with the *Hessian* $\mathbf{P}_{\Delta\check{\mathbf{T}}}\left(m, \check{\mathbf{W}}^{\ell-1}(m)\right)$. As in the Euclidean case, the Hessian can be formulated in several ways, e.g.,

$$\begin{aligned}&\mathbf{P}_{\Delta\check{\mathbf{T}}}\left(m, \check{\mathbf{W}}^{\ell-1}(m)\right) = \\ &\quad \left[\frac{\partial}{\partial \mathrm{vec}\left(\Delta\check{\mathbf{T}}\right)}\mathrm{vec}^{\mathrm{T}}\left(\frac{\partial\mathcal{J}\left(\check{\boldsymbol{\varphi}}_{\check{\mathbf{W}}^{\ell-1}(m)}(\Delta\check{\mathbf{T}})\right)}{\partial\Delta\check{\mathbf{T}}}\right)\right]_{\Delta\check{\mathbf{T}}=\mathbf{0}}\end{aligned} \tag{28b}$$

Finally, the new filter coefficients are then calculated using the map:

$$\check{\mathbf{W}}^{\ell}(m) = \check{\boldsymbol{\varphi}}_{\check{\mathbf{W}}^{\ell-1}(m)}\left(\Delta\check{\mathbf{T}}^{\ell}(m)\right). \tag{28c}$$

B. Sylvester Constraint on Manifolds and Further Generalization

To further develop the TRINICON coefficient update on Manifolds based on the general set of equations (28a)-(28c) we need to introduce the Sylvester Constraint ($\mathcal{SC}$). Introducing the Sylvester Constraint correctly in a rigorous way into the general update in arbitrary manifolds seems to be a non-trivial task. Fortunately, as we will see next in this section, the operator-based framework introduced in [3], [11] (see (12), (13)) already provides all the necessary tools to carry over the Sylvester Constraint to the optimization on arbitrary manifolds. To begin with, we first consider the gradient analogously to [11].

1) Sylvester Constraint for the Gradient on Manifolds: Let $W_{kj}^{KJ} = [\mathbf{W}]_{kj}^{KJ}$ denote the kj-th component of the *Sylvester matrix* for the KJ-th channel corresponding to the KJ-th submatrix. As shown in Sect. III-B, the Euclidean gradient of $\mathcal{J}$ w.r.t. these components is transformed by ($\mathcal{SC}$) to the Euclidean gradient w.r.t. the components $\check{W}_m^{MN} = \left[\check{\mathbf{W}}\right]_m^{MN}$ of the *downsized matrix*. The main ingredient towards the definition of the Sylvester operator was the observation that this can be expressed concisely by applying the chain rule for matrix derivatives. For the general case of arbitrary manifolds we exploit another important observation namely that the tangent space onto a arbitrary manifold forms an affine space which allows us to use the Euclidean gradient in the tangent space. This fact allows again for a straightforward application of the chain rule for matrix derivatives. In contrast to the Euclidean case we now apply it twice so that the gradient (as it appears in (28a)) reads

$$\left.\frac{\partial \mathcal{J}}{\partial(\Delta\check{T}_m^{MN})}\right|_{\Delta\check{\mathbf{T}}=\mathbf{0}} = \sum_{k,j,K,J}\sum_{r,R,S} \frac{\partial \mathcal{J}}{\partial(W'^{KJ}_{kj})} \left.\frac{\partial(W'^{KJ}_{kj})}{\partial\left([\check{\boldsymbol{\varphi}}_{\check{\mathbf{W}}'}]_r^{RS}\right)}\right|_{\Delta\check{\mathbf{T}}=\mathbf{0}} \left.\frac{\partial\left([\check{\boldsymbol{\varphi}}_{\check{\mathbf{W}}'}]_r^{RS}\right)}{\partial(\Delta\check{T}_m^{MN})}\right|_{\Delta\check{\mathbf{T}}=\mathbf{0}}.$$

Since $\check{\boldsymbol{\varphi}}_{\check{\mathbf{W}}'}(\mathbf{0}) = \check{\mathbf{W}}'$, we can rewrite and simplify this equation. After a straightforward calculation, we obtain

$$\left.\frac{\partial \mathcal{J}}{\partial(\Delta\check{T}_m^{MN})}\right|_{\Delta\check{\mathbf{T}}=\mathbf{0}} = \sum_{r,R,S}\left[\mathcal{SC}\left\{\frac{\partial \mathcal{J}}{\partial \mathbf{W}'}\right\}\right]_r^{RS} \left.\frac{\partial\left([\check{\boldsymbol{\varphi}}_{\check{\mathbf{W}}'}]_r^{RS}\right)}{\partial(\Delta\check{T}_m^{MN})}\right|_{\Delta\check{\mathbf{T}}=\mathbf{0}}. \quad (29)$$

Here we have used exactly the same definition of the Sylvester operator as in the Euclidean case. Applying the vec operator on this expression as required for (28a) yields

$$\mathrm{vec}\left(\nabla_{\Delta\check{\mathbf{T}}}\mathcal{J}\left(\check{\boldsymbol{\varphi}}_{\check{\mathbf{W}}^{\ell-1}(m)}(\Delta\check{\mathbf{T}})\right)\Big|_{\Delta\check{\mathbf{T}}=\mathbf{0}}\right) = \left.\frac{\partial \mathrm{vec}^{\mathrm{T}}\left(\check{\boldsymbol{\varphi}}_{\check{\mathbf{W}}'}\right)}{\partial \mathrm{vec}\left(\Delta\check{\mathbf{T}}\right)}\right|_{\Delta\check{\mathbf{T}}=\mathbf{0}} \mathrm{vec}\left(\mathcal{SC}\left\{\frac{\partial \mathcal{J}}{\partial \mathbf{W}'}\right\}\right). \quad (30)$$

Equation (30) allows for an interesting *first illustration*. The gradient vector of $\mathcal{J}$ on the right hand side is exactly the same as in the Euclidean case. The corresponding expression (30) on arbitrary manifolds exhibits an additional matrix

$$\check{\boldsymbol{\Xi}}\left(m, \check{\mathbf{W}}'(m)\right) := \left.\frac{\partial \mathrm{vec}^{\mathrm{T}}\left(\check{\boldsymbol{\varphi}}_{\check{\mathbf{W}}'}\right)}{\partial \mathrm{vec}\left(\Delta\check{\mathbf{T}}\right)}\right|_{\Delta\check{\mathbf{T}}=\mathbf{0}}. \quad (31)$$

Obviously, this additional matrix depends on the current point $\check{\mathbf{W}}'$ and on the structure of the underlying manifold $\mathcal{M}$. In the **Euclidean space as a special case** the map $\check{\boldsymbol{\varphi}}_{\check{\mathbf{W}}'}$ has the simple form

$$\check{\mathbf{W}} = \check{\boldsymbol{\varphi}}_{\check{\mathbf{W}}'}^{\mathrm{Euclid}}(\Delta\check{\mathbf{W}}) = \check{\mathbf{W}}' + \Delta\check{\mathbf{T}}. \quad (32)$$

It is easy to verify that in this case the additional matrix $\check{\boldsymbol{\Xi}}$ on the right hand side of (30) is equal to

$$\check{\boldsymbol{\Xi}}^{\mathrm{Euclid}} = \mathbf{I}. \quad (33)$$

Moreover, noting that the gradient descent coefficient update follows from the Newton update by setting the Hessian equal to $\mathbf{I}$, the familiar gradient-based update in the Euclidean case follows easily as a special case from the general equations (28a) (note the minus sign in addition to the expression (30)) and (28c). $\diamond$

2) Generalized Manifold Formulation Integrating the Sylvester Constraint: Although (30) may already be seen as a final expression of the gradient in the tangent space, a further generalization integrating the Sylvester constraint into the manifold is possible as shown next in this paragraph. Later we will see that such a generalization is in fact desirable in order to obtain an even more flexible expression, particularly for the Newton update and also for maps other than the Euclidean map (32).

Having introduced the general notion of an optimization on manifolds on the one hand, and the Sylvester constraint on the other hand, we can render the optimization criterion $\mathcal{J}$ w.r.t. the filter coefficients concisely as

$$\mathcal{J} = \mathcal{J}\left(m, \mathcal{S}\left\{\check{\boldsymbol{\varphi}}_{\check{\mathbf{W}}'}\left(\check{\mathbf{0}}\right)\right\}\right), \quad (34)$$

where $\mathcal{S}\{\cdot\}$ denotes an operator generating a Sylvester matrix. Obviously, this can be written equivalently as

$$\mathcal{J} = \mathcal{J}\left(m, \boldsymbol{\varphi}_{\mathcal{S}\left\{\check{\mathbf{W}}'\right\}}(\mathbf{0})\right), \quad (35)$$

where we have introduced the new matrix function

$$\boldsymbol{\varphi}_{\mathbf{W}'}(\Delta\mathbf{T}) = \mathcal{S}\left\{\check{\boldsymbol{\varphi}}_{\mathcal{SC}\{\mathbf{W}'\}}\left(\widetilde{\mathcal{SC}}\left\{\Delta\mathbf{T}\right\}\right)\right\}. \quad (36)$$

Equations (34)-(36) show that the Sylvester constraint already forms a special manifold even in the Euclidean case. Thus, in fact we have already worked with a special kind of manifold above in Sect. III-B.

The new operator $\widetilde{\mathcal{SC}}\{\cdot\}$ in (36) linking $\mathbf{T}$ and $\check{\mathbf{T}}$ denotes the *counterpart of the Sylvester operator $\mathcal{SC}\{\cdot\}$ in the tangent space*, i.e., in analogy to (12) we write

$$\Delta\check{\mathbf{T}} = \widetilde{\mathcal{SC}}\left\{\Delta\mathbf{T}\right\} = \widetilde{\mathcal{SC}}\left\{\Delta\mathbf{T}\right\}. \quad (37)$$

In general, the distinction between $\widetilde{\mathcal{SC}}\{\cdot\}$ and $\mathcal{SC}\{\cdot\}$ is necessary since the projection of a Sylvester matrix into the corresponding tangent space of the chosen manifold $\mathcal{M}$ does *not* necessarily[1] exhibit a Sylvester structure. The particular detailed definition of $\widetilde{\mathcal{SC}}\{\cdot\}$ depends on the chosen manifold $\mathcal{M}$ and it is therefore another degree of freedom for the coefficient optimization (examples, such as the so-called natural gradient-based coefficient update will be shown below). Despite of this flexibility, many of the known properties of $\mathcal{SC}\{\cdot\}$ carry over to the operator $\widetilde{\mathcal{SC}}\{\cdot\}$ in the tangent space. In particular, when using $\widetilde{\mathcal{SC}}\{\cdot\}$ in conjunction with the vec operator, we can also introduce a corresponding *Sylvester constraint matrix $\mathbf{K}_{\widetilde{\mathcal{SC}}}$ in the tangent space*, defined by the derivative

$$\mathbf{K}_{\widetilde{\mathcal{SC}}} = \frac{\partial \mathrm{vec}^{\mathrm{T}}\left\{\Delta\mathbf{T}\right\}}{\partial \mathrm{vec}\left\{\Delta\check{\mathbf{T}}\right\}}, \quad \mathbf{K}_{\widetilde{\mathcal{SC}}} = \frac{\partial \mathrm{vec}^{\mathrm{T}}\left\{\Delta\mathbf{T}\right\}}{\partial \mathrm{vec}\left\{\Delta\check{\mathbf{T}}\right\}}. \quad (38)$$

Moreover, in the same way as shown for the coefficient space, the *Sylvester generator in the tangent space*

$$\Delta\mathbf{T} = \widetilde{\mathcal{S}}\left\{\Delta\check{\mathbf{T}}\right\}, \quad \Delta\mathbf{T} = \widetilde{\mathcal{S}}\left\{\Delta\check{\mathbf{T}}\right\} \quad (39)$$

[1] In the special case of an Euclidean manifold after (32), $\widetilde{\mathcal{SC}}\{\cdot\}$ is obviously equal to $\mathcal{SC}\{\cdot\}$.

is introduced as a complementary operation, and the corresponding *Sylvester generator matrix* reads

$$\mathbf{K}_{\tilde{\mathcal{S}}} = \frac{\partial \mathrm{vec}^{\mathrm{T}}\left\{\Delta\check{\mathbf{T}}\right\}}{\partial \mathrm{vec}\left\{\Delta\mathbf{T}\right\}}, \quad \mathbf{K}_{\tilde{\mathcal{S}}} = \frac{\partial \mathrm{vec}^{\mathrm{T}}\left\{\Delta\check{\mathbf{T}}\right\}}{\partial \mathrm{vec}\left\{\Delta\mathbf{T}\right\}}. \tag{40}$$

Having introduced the formulation (35) with the generalized matrix function $\boldsymbol{\varphi}_{\mathbf{W}'}$, we next write the set of equations (28a)-(28c) in an even more general way by means of $\boldsymbol{\varphi}$ instead of $\breve{\boldsymbol{\varphi}}$. It can be shown that we can formulate the Hessian in terms of Sylvester matrices rather than the downsized coefficient matrices without loss of generality. Analogously, we formulate here the Hessian in terms of the matrix $\Delta\mathbf{T}$. The set of update equations can then be formulated as

$$\begin{aligned}\mathrm{vec}\Delta\mathbf{T}^{\ell}(m) &= -\mathbf{P}_{\Delta\mathbf{T}}^{-1}\left(m, \mathbf{W}^{\ell-1}(m)\right) \\ &\quad \mathrm{vec}\left(\nabla_{\Delta\mathbf{T}}\mathcal{J}\left(\boldsymbol{\varphi}_{\mathbf{W}^{\ell-1}(m)}(\Delta\mathbf{T})\right)\Big|_{\Delta\mathbf{T}=\mathbf{0}}\right)\end{aligned} \tag{41a}$$

with the *Hessian*

$$\begin{aligned}&\mathbf{P}_{\Delta\mathbf{T}}\left(m, \mathbf{W}^{\ell-1}(m)\right) = \\ &= \left[\frac{\partial}{\partial \mathrm{vec}\left(\Delta\mathbf{T}\right)}\mathrm{vec}^{\mathrm{T}}\left(\frac{\partial\mathcal{J}\left(\boldsymbol{\varphi}_{\mathbf{W}^{\ell-1}(m)}(\Delta\mathbf{T})\right)}{\partial\Delta\mathbf{T}}\right)\right]_{\Delta\mathbf{T}=\mathbf{0}}\end{aligned} \tag{41b}$$

and the actual coefficient update

$$\mathbf{W}^{\ell}(m) = \boldsymbol{\varphi}_{\mathbf{W}^{\ell-1}(m)}\left(\Delta\mathbf{T}^{\ell}(m)\right). \tag{41c}$$

In addition, in order to finally obtain the downsized coefficient matrix $\check{\mathbf{W}}^{\ell}(m)$, we simply apply the Sylvester operator to the result of (41c):

$$\check{\mathbf{W}}^{\ell}(m) = \mathcal{SC}\left\{\mathbf{W}^{\ell}(m)\right\}. \tag{41d}$$

The last two steps can be justified by applying $\mathcal{SC}\{\cdot\}$ on both sides of (36),

$$\mathcal{SC}\left\{\boldsymbol{\varphi}_{\mathbf{W}'}\left(\Delta\mathbf{T}\right)\right\} = \mathcal{SC}\left\{\mathcal{S}\left\{\breve{\boldsymbol{\varphi}}_{\mathcal{SC}\{\mathbf{W}'\}}\left(\widetilde{\mathcal{SC}}\left\{\Delta\mathbf{T}\right\}\right)\right\}\right\}, \tag{42}$$

so that

$$\mathcal{SC}\left\{\boldsymbol{\varphi}_{\mathbf{W}'}\left(\Delta\mathbf{T}\right)\right\} = \breve{\boldsymbol{\varphi}}_{\mathcal{SC}\{\mathbf{W}'\}}\left(\widetilde{\mathcal{SC}}\left\{\Delta\mathbf{T}\right\}\right) = \check{\mathbf{W}}, \tag{43}$$

i.e.,

$$\check{\mathbf{W}} = \mathcal{SC}\left\{\boldsymbol{\varphi}_{\mathbf{W}'}\left(\Delta\mathbf{T}\right)\right\}. \tag{44}$$

This equation corresponds to (41c) and (41d).

C. TRINICON Gradient Calculation on Manifolds and Gradient Coefficient Update on Manifolds

Based on the general set of equations (41a)-(41d) we now derive the TRINICON gradient on Manifolds and the corresponding gradient-based coefficient update on arbitrary partly smooth manifolds. The result of this subsection will also act as an intermediary result towards the more general TRINICON-based Newton coefficient update treated in the next subsection.

To begin with, we formulate the gradient in (41a) similarly as in (30) by applying the chain rule again:

$$\begin{aligned}&\frac{\partial\mathcal{J}}{\partial(\Delta T_{mn}^{MN})}\bigg|_{\Delta\mathbf{T}=\mathbf{0}} = \\ &= \sum_{k,j,K,J}\frac{\partial\mathcal{J}}{\partial\left([\boldsymbol{\varphi}_{\mathbf{W}'}]_{kj}^{KJ}\right)}\bigg|_{\Delta\mathbf{T}=\mathbf{0}}\frac{\partial\left([\boldsymbol{\varphi}_{\mathbf{W}'}]_{kj}^{KJ}\right)}{\partial(\Delta T_{mn}^{MN})}\bigg|_{\Delta\mathbf{T}=\mathbf{0}}. \\ &= \sum_{k,j,K,J}\frac{\partial\mathcal{J}}{\partial(W'^{KJ}_{kj})}\frac{\partial\left([\boldsymbol{\varphi}_{\mathbf{W}'}]_{kj}^{KJ}\right)}{\partial(\Delta T_{mn}^{MN})}\bigg|_{\Delta\mathbf{T}=\mathbf{0}}.\end{aligned} \tag{45}$$

The gradient thus reads

$$\begin{aligned}&\mathrm{vec}\left(\nabla_{\Delta\mathbf{T}}\mathcal{J}\left(\boldsymbol{\varphi}_{\mathbf{W}^{\ell-1}(m)}\left(\Delta\mathbf{T}\right)\right)\Big|_{\Delta\mathbf{T}=\mathbf{0}}\right) = \\ &= \frac{\partial\mathrm{vec}^{\mathrm{T}}\left(\boldsymbol{\varphi}_{\mathbf{W}^{\ell-1}(m)}\right)}{\partial\mathrm{vec}\left(\Delta\mathbf{T}\right)}\bigg|_{\Delta\mathbf{T}=\mathbf{0}}\mathrm{vec}\left(\frac{\partial\mathcal{J}}{\partial\mathbf{W}}\Big|_{\mathbf{W}=\mathbf{W}^{\ell-1}(m)}\right).\end{aligned} \tag{46}$$

This equation again traces back the gradient in the tangent space to the partial derivative w.r.t. the filter coefficient matrix and a premultiplied matrix

$$\boldsymbol{\Xi}\left(m, \mathbf{W}^{\ell-1}(m)\right) := \frac{\partial\mathrm{vec}^{\mathrm{T}}\left(\boldsymbol{\varphi}_{\mathbf{W}^{\ell-1}(m)}\right)}{\partial\mathrm{vec}\left(\Delta\mathbf{T}\right)}\bigg|_{\Delta\mathbf{T}=\mathbf{0}} \tag{47a}$$

depending on the manifold $\mathcal{M}$. The partial derivative w.r.t. the coefficient matrix corresponds to the *Euclidean* gradient as already presented in Sect. III-B (equation (14a) without the $\mathcal{SC}$ operator). Thus, based on (46), (41a), (41c), and (41d), we can now summarize the **general TRINICON gradient-based coefficient update rule on an arbitrary partly smooth manifold** $\mathcal{M}$ as follows[2]:

$$\begin{aligned}\mathrm{vec}\Delta\mathbf{T}^{\ell}(m) &= -\frac{\boldsymbol{\mu}}{N}\sum_{i=0}^{\infty}\beta(i,m)\,\boldsymbol{\Xi}\left(m, \mathbf{W}^{\ell-1}(m)\right) \\ &\quad \cdot\sum_{j=iN_L}^{iN_L+N-1}\mathrm{vec}\left\{\mathbf{x}(j)\boldsymbol{\Phi}_{\mathrm{s},PD}^{\mathrm{T}}(\mathbf{y}(j)) - \left(\mathbf{W}^{\mathrm{T}}\right)^{+}\Big|_{\mathbf{W}=\mathbf{W}^{\ell-1}(m)}\right\},\end{aligned} \tag{47b}$$

$$\check{\mathbf{W}}^{\ell}(m) = \mathcal{SC}\left\{\boldsymbol{\varphi}_{\mathbf{W}^{\ell-1}(m)}\left(\Delta\mathbf{T}^{\ell}(m)\right)\right\}. \tag{47c}$$

D. TRINICON Hessian Calculation on Manifolds and Newton Coefficient Update on Manifolds

In (41b) we have introduced the Hessian $\mathbf{P}_{\Delta\mathbf{T}}\left(m, \mathbf{W}^{\ell-1}(m)\right)$ in the tangent space. In the same way as we have traced back the gradient in the tangent space to the partial derivative w.r.t. the filter coefficient matrix $\mathbf{W}$ and the premultiplied matrix $\boldsymbol{\Xi}$ after (47a), we now reformulate the Hessian. A detailed derivation (not shown here for brevity) yields the expression

$$\begin{aligned}&\mathbf{P}_{\Delta\mathbf{T}}\left(m, \mathbf{W}^{\ell-1}(m)\right) = \\ &= \boldsymbol{\Xi}\left(m, \mathbf{W}^{\ell-1}(m)\right)\mathbf{P}_{\mathbf{W}}\left(m, \mathbf{W}^{\ell-1}(m)\right)\boldsymbol{\Xi}^{\mathrm{T}}\left(m, \mathbf{W}^{\ell-1}(m)\right) \\ &\quad + \boldsymbol{\Xi}\left(m, \mathbf{W}^{\ell-1}(m)\right)\sum_{i}\left(\left[\mathrm{vec}^{\mathrm{T}}\left(\frac{\partial\mathcal{J}}{\partial\mathbf{W}}\right)\right]_i\Big|_{\mathbf{W}=\mathbf{W}^{\ell-1}(m)}\right. \\ &\quad \left.\frac{\partial^2\left[\mathrm{vec}\left(\boldsymbol{\varphi}_{\mathbf{W}^{\ell-1}(m)}\right)\right]_i}{\partial\mathrm{vec}^{\mathrm{T}}\left(\Delta\mathbf{T}\right)\partial\mathrm{vec}\left(\boldsymbol{\varphi}_{\mathbf{W}^{\ell-1}(m)}\right)}\bigg|_{\Delta\mathbf{T}=\mathbf{0}}\right),\end{aligned} \tag{48}$$

where $\mathbf{P}_{\mathbf{W}}\left(m, \mathbf{W}^{\ell-1}(m)\right)$ corresponds to the *Euclidean Hessian* as already introduced explicitly for the TRINICON criterion above. The derivative $\partial\mathcal{J}/\partial\mathbf{W}$ in the second term of (48) corresponds to the *Euclidean gradient* that also has already been shown above for TRINICON. All other quantities in the expression (48) exclusively depend on the chosen manifold $\mathcal{M}$: Both the matrix $\boldsymbol{\Xi}$ according to (47a), and the last derivative on the right-hand side of (48) are derived

[2]Another degree of freedom would be to decouple the map in (47c) from the one in (47a) in order to allow more approximations of this generic set of equations. This could be done by formally replacing $\varphi_{\mathbf{W}'}(\cdot)$ in (47c) by another function $\psi_{\mathbf{W}'}(\cdot)$. In the following, we omit this further generalization for brevity.

directly from the map $\boldsymbol{\varphi}_{\mathbf{W}^{\ell-1}(m)}$. Note that this last derivative in (48) is in fact a quantity with three indices rather than an ordinary matrix. It should be noted that in the special case of so-called Riemannian manifolds[3] in the field of differential geometry and tensor analysis, there is a close relation to the so-called *Christoffel symbols* [12], [13]. Note that **in the Euclidean case**, these quantities, and thus the second term in (48) are zero[4]. Moreover, since in this case $\check{\boldsymbol{\Xi}} = \check{\boldsymbol{\Xi}}^{\text{Euclid}} = \mathbf{I}$, we have $\boldsymbol{\Xi} = \mathbf{K}_{\tilde{\mathcal{S}}}\mathbf{K}_{\mathcal{SC}} = \mathbf{K}_{\mathcal{S}}\mathbf{K}_{\mathcal{SC}}$, the Hessian (48) simplifies in the Euclidean case to

$$\begin{aligned}\mathbf{P}_{\Delta\mathbf{T}}\left(m,\mathbf{W}^{\ell-1}(m)\right) &= \mathbf{K}_{\mathcal{S}}\mathbf{K}_{\mathcal{SC}}\mathbf{P}_{\mathbf{W}}\left(m,\mathbf{W}^{\ell-1}(m)\right)\mathbf{K}_{\mathcal{SC}}^{\mathrm{T}}\mathbf{K}_{\mathcal{S}}^{\mathrm{T}} \\ &= \mathbf{K}_{\mathcal{S}}\mathbf{P}_{\check{\mathbf{W}}}\left(m,\check{\mathbf{W}}^{\ell-1}(m)\right)\mathbf{K}_{\mathcal{S}}^{\mathrm{T}}. \qquad (49)\end{aligned}$$

VI. Some Important Special Cases of Manifolds

Having derived the generic expressions for TRINICON-based adaptation on arbitrary partly smooth manifolds, we now discuss some illustrative examples for special choices of maps $\boldsymbol{\varphi}_{\mathbf{W}^{\ell-1}(m)}(\Delta\mathbf{T})$. The discussions in this section are mainly based on (47a)-(47c) but the extension to Newton-based updates with the Hessian (48) is straigthforward.

A. Euclidean Space and Riemannian Space

As already mentioned, the adaptation in the Euclidean space is given by the map

$$\mathbf{W} = \boldsymbol{\varphi}_{\mathbf{W}'}^{\text{Euclid}}(\Delta\mathbf{T}) = \mathbf{W}' + \Delta\mathbf{T}, \qquad (50)$$

which is easily verified by plugging it into (47a)-(47c). It should be emphasized that the additive structure of (50) represents a fundamental feature of the Euclidean space. In the Euclidean space, the shortest connection between two points is given by a straight line. In other words, in the Euclidean case the adaptation of the filter coefficients corresponds to a pure *translation*, as represented in (11b). On arbitrary manifolds this is in general not the case. Hence, (47a)-(47c) are not restricted to pure translations. For instance, in the Riemannian space, the shortest connection between two points is given by a *geodesic line*[5]. The corresponding map is known as the *exponential map* [12], [13], i.e., $\mathbf{W} = \boldsymbol{\varphi}_{\mathbf{W}'}^{\exp}(\Delta\mathbf{T})$.

B. TRINICON Natural Newton and Natural Gradient Updates

The *natural gradient* [7], also called *relative gradient* [16], [17], is based on the assumption that the set of matrices representing possible demixing systems form a so-called Lie group. In other words, the natural gradient is based on the prior knowledge that we have an invertible MIMO system, and exploiting this prior knowlege by the natural gradient leads to the so-called equivariance property. Keeping the requirement of equivariance for convolutive mixtures, it was shown that the natural gradient can also be formulated compactly using Sylvester matrices, e.g., [18]. The generic formulation for convolutive mixtures is given by

$$\Delta\check{\mathbf{W}} = \mathcal{SC}\left\{\mathbf{W}\mathbf{W}^{\mathrm{T}}\frac{\partial\mathcal{J}}{\partial\mathbf{W}}\right\}. \qquad (51)$$

[3]In the Riemannian case the manifold is equipped with a metric which allows us to formally define lengths and angles in the manifold.

[4]This fact is also well known in the Riemannian case: The Euclidean geometry precisely is the special case of the Riemannian geometry where the corresponding term is equal to zero.

[5]In general, geodesic lines can be calculated by the calculus of variations using a given metric tensor. However, it should be noted that for arbitrary and/or high-dimensional geometries this is a non-trivial task and the existence of closed-form solutions is not always guaranteed.

Note that unlike the adaptation in the Riemannian space, the natural gradient-based adaptation still contains the translation (11b). It can easily be verified using (47a)-(47c) that the natural gradient-based adaptation is given by the map

$$\mathbf{W} = \boldsymbol{\varphi}_{\mathbf{W}'}^{\text{Natural}}(\Delta\mathbf{T}) = \mathbf{W}' + \mathbf{W}'\Delta\mathbf{T}. \qquad (52)$$

Note that using the same map, it is straightforward to derive a *natural Newton algorithm*/*relative Newton algorithm*. Moreover, in the same way as for the natural gradient it can be shown that the natural Newton algorithm also exhibits the equivariance property.

C. Linearly Constrained TRINICON-Based Adaptation

As mentioned above, any *constrained optimization problem* in the Euclidean space can be thought of as *unconstrained optimization problem* on a special manifold. Specifically, the constrained problem

$$\check{\mathbf{W}} = \arg\min_{\check{\mathbf{W}}\in\mathbb{R}^{LP\times P}} \mathcal{J}\left(\check{\mathbf{W}}\right) \quad \text{subject to } \mathbf{G}(\check{\mathbf{W}}) = \mathbf{0} \qquad (53)$$

can be thought of as the unconstrained problem

$$\check{\mathbf{W}} = \arg\min_{\check{\mathbf{W}}\in\mathcal{M}} \mathcal{J}\left(\check{\mathbf{W}}\right), \qquad (54)$$

where $\mathcal{M} = \left\{\check{\mathbf{W}} \mid \mathbf{G}(\check{\mathbf{W}}) = \mathbf{0}\right\}$.

As an example, we now consider the *linearly constrained* problem (which should otherwise be Euclidean). In practice, linearly constrained adaptive filtering has a particularly prominent role in the field of adaptive beamforming as the linear constraint allows to incorporate prior information on the direction of the incoming source signals relative to the sensor array, e.g., [4], [5], [6].

As an *approach* to determine the corresponding map, we are searching for the *linear map*

$$\mathbf{W} = \boldsymbol{\varphi}_{\mathbf{W}'}^{\text{lin}}(\Delta\mathbf{T}) = \mathbf{A}\mathbf{W}' + \mathbf{B}\Delta\mathbf{T} + \tilde{\mathbf{F}} \qquad (55)$$

that satisfies the linear constraint

$$\mathbf{C}^{\mathrm{T}}\mathbf{W} = \mathbf{F} \qquad (56)$$

in the otherwise Euclidean space. The assumed Euclidean requirement leads to $\mathbf{B} = \mathbf{A}$. Moreover, plugging (55) into (56) and equating coefficients on both sides of the resulting expression finally leads to the map

$$\mathbf{W} = \boldsymbol{\varphi}_{\mathbf{W}'}^{\text{LC}}(\Delta\mathbf{T}) = \mathbf{P}\mathbf{W}' + \mathbf{P}\Delta\mathbf{T} + \tilde{\mathbf{F}}, \qquad (57a)$$

where

$$\mathbf{A} = \mathbf{P} = \mathbf{I} - \mathbf{C}\left[\mathbf{C}^{\mathrm{T}}\mathbf{C}\right]^{-1}\mathbf{C}^{\mathrm{T}}, \qquad (57b)$$

$$\tilde{\mathbf{F}} = \mathbf{C}\left[\mathbf{C}^{\mathrm{T}}\mathbf{C}\right]^{-1}\mathbf{F}. \qquad (57c)$$

Again, it should be emphasized that (57) is valid for all possible gradient-based algorithms, Newton-based algorithms, or Quasi-Newton algorithms.

For example, the resulting gradient-based coefficient update is obtained by plugging (57) into the generic set of equations (47a)-(47c). Noting that $\mathbf{P}\mathbf{P} = \mathbf{P}$, this leads to

$$\check{\mathbf{W}} = \mathcal{SC}\left\{\mathbf{P}\left[\mathbf{W}' - \mu\frac{\partial\mathcal{J}}{\partial\mathbf{W}'}\right] + \tilde{\mathbf{F}}\right\}. \qquad (58)$$

Considering the structure of this coefficient update equation, we readily see that (58) together with (57b) and (57c) is precisely the *TRINICON-based generalization of the so-called Frost beamformer*, which in its original form represents an adaptive gradient-descent solution of the linearly constrained minimum variance (LCMV) problem [4], [5], [6]. Note that in contrast to the original approach

in [4], the approach based on manifolds can straightforwardly be generalized to Newton-based optimization procedures. Moreover, it should be noted that there is a tight relation between the Frost beamformer and the so-called generalized sidelobe canceller (GSC) structure [19] which can also be exploited in the more general case of TRINICON-based adaptation. Using more advanced optimization criteria instead of the original minimum-variance criterion promises a more robust performance w.r.t. the well known signal cancellation problem in adaptive beamforming, e.g., [20].

D. Linearly Constrained Natural Gradient and Linearly Constrained Natural Newton Adaptation

As a simple example for a manifold combining various constraints, we consider the linearly constrained adaptation problem from Sect. VI-C using natural gradient/natural Newton-based adaptation after Sect. VI-B. In this case, the corresponding manifold is given by the map

$$\mathbf{W} = \varphi^{\mathrm{LC}}_{\mathbf{W}'}(\Delta\mathbf{T}) = \mathbf{P}\mathbf{W}' + \mathbf{P}\mathbf{W}'\Delta\mathbf{T} + \tilde{\mathbf{F}}. \tag{59}$$

It can be verified that the resulting coefficient matrix fulfills both the linear condition (56) and the equivariance property of the natural gradient/natural Newton-based adaptation.

E. Supervised Adaptive Filtering

In Sect. VI-C we have seen that the known structures for adaptive beamforming and related generalizations follow rigorously from TRINICON-based adaptation in combination with certain choices of manifolds for exploiting prior knowledge on the direction of the incoming source signals relative to the sensor array.

Similarly, in [9] the supervised adaptive filter theory [1] was related rigorously to the blind adaptive filter theory. Based on the problem of system identification it was shown that the supervised algorithms for single-input and single-output systems (SISO) can be derived from TRINICON in combination with a certain prior knowledge on the structure of the mixing system after Fig. 1. Hence, we can expect that the supervised algorithms are obtained even more systematically by choosing a certain manifold which we confirm in the following.

According to [9], due to the prior knowledge the solution of the demixing system $\check{\mathbf{W}}$ is constrained as follows:

$$\begin{bmatrix} \mathbf{w}_{11} & \mathbf{w}_{12} \\ \mathbf{w}_{21} & \mathbf{w}_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{1}_1 & \mathbf{0} \\ -\hat{\mathbf{h}} & \mathbf{1}_1 \end{bmatrix}, \tag{60}$$

where $\hat{\mathbf{h}}$ denotes the estimate of the unknown SISO system. Hence, the supervised adaptive filter algorithms can be derived by suitably picking the lower left submatrix of the TRINICON-based coefficient updates [9]. Obviously, a suitable map taking into account this knowledge is given by

$$\begin{aligned} \mathbf{W} &= \varphi^{\mathrm{superv.,Euclid}}_{\mathbf{W}'}(\Delta\mathbf{T}) \\ &= \begin{bmatrix} \tilde{\mathbf{V}} & \mathbf{0} \\ \mathbf{W}'_{21} & \tilde{\mathbf{V}} \end{bmatrix} + \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{V}} \end{bmatrix} \Delta\mathbf{T} \begin{bmatrix} \tilde{\mathbf{V}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \end{aligned} \tag{61}$$

An even more compact representation which directly picks the lower left submatrix $\mathbf{W}_{\mathrm{superv}}$ is given by the following map:

$$\begin{aligned} \mathbf{W}_{\mathrm{superv}} &:= -\hat{\mathbf{H}} = \varphi^{\mathrm{superv.,Euclid}}_{\hat{\mathbf{H}}'}(\Delta\mathbf{T}) \\ &= -\hat{\mathbf{H}}' + \underbrace{\begin{bmatrix} \mathbf{0} & \tilde{\mathbf{V}} \end{bmatrix}}_{=:\mathbf{G}_{01}} \Delta\mathbf{T} \underbrace{\begin{bmatrix} \tilde{\mathbf{V}} \\ \mathbf{0} \end{bmatrix}}_{=:\mathbf{G}_{10}}. \end{aligned} \tag{62}$$

Using the matrix relation $\mathrm{vec}(\mathbf{ABC}) = (\mathbf{C}^{\mathrm{T}} \otimes \mathbf{A})\,\mathrm{vec}(\mathbf{B})$, where $\otimes$ denotes the Kronecker product, we obtain from (47a) the constant windowing matrix $\boldsymbol{\Xi}\left(m, \mathbf{W}^{\ell-1}(m)\right) = \mathbf{G}_{10} \otimes \mathbf{G}_{01}^{\mathrm{T}}$. This directly leads to the supervised adaptive filtering algorithms according to [9], including the Newton-type algorithms (such as the well known recursive least-squares algorithm).

VII. Conclusions

In this paper we introduced generic expressions for TRINICON-based adaptation on arbitrary partly smooth manifolds. It was demonstrated that this concept adds another important degree of freedom in the general framework so that nearly arbitrary deterministic prior **system** information can be incorporated.

References

[1] S. Haykin, *Adaptive Filter Theory*, 4th ed. Englewood Cliffs, NJ: Prentice Hall Inc., 2002.

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

[3] 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. Boston: Kluwer Academic Publishers, Apr. 2004, pp. 255–293.

[4] O. Frost, "An algorithm for linearly constrained adaptive array processing," *Proceedings of the IEEE*, vol. 60, no. 8, pp. 926–935, Aug. 1972.

[5] D. Johnson and D. Dudgeon, *Array Signal Processing: Concepts and Techniques*. Prentice Hall, 1993.

[6] B. van Veen and K. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Magazine*, pp. 4–24, Apr. 1988.

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

[8] J. Shynk, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Processing Magazine*, pp. 14–37, Jan. 1992.

[9] 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. Workshop Hands-Free Speech Commun. and Microphone Arrays*, Trento, Italy, May 2008.

[10] 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)*, vol. 3, Montreal, Canada, May 2004, pp. 889–892.

[11] ——, "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. Berlin: Springer, Sept. 2007, pp. 101–147.

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

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

[14] T. Cover and J. Thomas, *Elements of Information Theory*. New York: Wiley & Sons, 1991.

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

[16] J.-F. Cardoso and B. Laheld, "Equivariant adaptive source separation," *IEEE. Trans. Signal Processing*, vol. 44, no. 12, pp. 3017–3030, Dec. 1996.

[17] J.-F. Cardoso, "Blind signal separation: statistical principles," *Proc. IEEE*, vol. 86, pp. 2009–2025, Oct. 1998.

[18] H. Buchner, R. Aichner, and W. Kellermann, "A generalization of a class of blind source separation algorithms for convolutive mixtures," in *Proc. Int. Symposium on Independent Component Analysis and Blind Signal Separation (ICA)*, Nara, Japan, Apr. 2003, pp. 945–950.

[19] L. Griffiths and C. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Trans. Antennas Propagation*, vol. 30, no. 1, pp. 27–34, Jan. 1982.

[20] B. Widrow, K. Duvall, R. Gooch, and W. Newman, "Signal cancellation phenomena in adaptive antennas: Causes and cures," *IEEE Trans. Antennas and Propagation*, vol. 30, no. 3, pp. 469–478, May 1982.