

Low-Complexity Addition or Removal of Sensors/Constraints in LCMV Beamformers

Shmulik Markovich-Golan, *Student Member, IEEE*, Sharon Gannot, *Senior Member, IEEE*, and Israel Cohen, *Senior Member, IEEE*

Abstract—We address the application of the linearly constrained minimum variance (LCMV) beamformer in sensor networks. In signal processing applications, it is common to have a redundancy in the number of nodes, fully covering the area of interest. Here we consider suboptimal LCMV beamformers utilizing only a subset of the available sensors for signal enhancement applications. Multiple desired and interfering sources scenarios in multipath environments are considered. We assume that an *oracle* entity determines the group of sensors participating in the spatial filtering, denoted as the active sensors. The oracle is also responsible for updating the constraints set according to either sensors or sources activity or dynamics. Any update of the active sensors or of the constraints set necessitates recalculation of the beamformer and increases the power consumption. As power consumption is a most valuable resource in sensor networks, it is important to derive efficient update schemes. In this paper, we derive procedures for adding or removing either an active sensor or a constraint from an existing LCMV beamformer. Closed-form, as well as generalized sidelobe canceller (GSC)-form implementations, are derived. These procedures use the previous beamformer to save calculations in the updating process. We analyze the computational burden of the proposed procedures and show that it is much lower than the computational burden of the straightforward calculation of their corresponding beamformers.

Index Terms— Beamforming, GSC, LCMV.

I. INTRODUCTION

THE linearly constrained minimum variance (LCMV) beamformer (BF) is a common and powerful scheme for signal enhancement in complicated scenarios, usually involving multiple sources. The LCMV-BF was first introduced by Er and Cantoni [1]. They extended the minimum variance distortion-less response (MVDR)-BF [2], [3], and proposed a BF satisfying a set of linear constraints. Multiple constraints allows for further control of the array beam-pattern, beyond that of a single steer-direction gain constraint. Breed and Strauss [4]

proved that the LCMV extension has also an equivalent generalized sidelobe canceller (GSC) structure, which decouples the constraining and the minimization operations. Affes and Grenier [5] and later Gannot *et al.* [6] reformulated the GSC structure in the frequency domain, extending its application to reverberant environments by handling the more general transfer function (TF). Various strategies for designing the constraints sets exist, several examples are given next. The constraints set can be used for extracting a group of desired speakers out of a mixture of desired and interfering speakers [7]. Another strategy, used for focusing in near-field scenarios, defines a spatial area of interest [8]. Finally, the sensitivity, and robustness of the BF can be controlled by constraining its derivative in certain look directions [9].

Sensor networks deployed over large areas hold great potential for signal processing applications, and call upon applying beamforming techniques. The vast area of deployment allows for a fine spatial resolution, inversely proportional to the effective aperture. Moreover, the large number of sensors improves the ability of the beamformer to cope with multiple sources scenarios, as proposed by Markovich *et al.* [7].

Two major drawbacks result from applying beamforming in distributed sensor networks. The first drawback is the communication bandwidth utilization which increases linearly with the number of sensors, assuming full connectivity of the network (broadcast mechanism is assumed to be available). The second drawback is the growing computational burden for constructing the BF. Several contributions have addressed the problem of reducing the communication bandwidth [10]–[15]. Here we address the computational burden drawback.

Two main causes impose severe complexity constraints in a distributed sensor network. The first cause is energy saving and battery life. Higher energy consumption results from increased computational burden, and is manifested in the large number of mega instruction per second (MIPS). The dynamics of the network and the environment necessitates updating the BF. Recalculation of the BF results in shorter system's lifetime. The computational burden is emphasized in wide-band signal applications in complicated environments such as speech processing in reverberant environments. Dealing with such long room impulse responses (RIRs) requires calculating BF with, respectively, long impulse responses and involves many computations. The second cause stems from the low cost nodes. Complex algorithms require stronger processing units resulting in more expensive nodes, and might prevent deployment of large quantities.

As aforementioned, constructing a BF utilizing all sensor data requires large bandwidth. The contribution of each of the nodes

Manuscript received May 03, 2011; revised July 24, 2011 and November 09, 2011; accepted November 09, 2011. Date of publication December 02, 2011; date of current version February 10, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Donimic K. C. Ho.

S. Markovich-Golan and S. Gannot are with the School of Engineering, Bar-Ilan University, Ramat-Gan 56000, Israel (e-mail: shmulik.markovich@gmail.com; sharon.gannot@biu.ac.il).

I. Cohen is with the Department of Electrical Engineering, Technion, Technion City, Haifa 32000, Israel (e-mail: icohen@ee.technion.ac.il).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TSP.2011.2177829

to the noise reduction task is not equal. Given a bandwidth limitation, a subset of the nodes could be chosen to maximize the noise reduction. Bertrand and Moonen [16] propose an efficient method for updating the multichannel Wiener filter (MWF)-BF corresponding to removal or addition of sensors. They derive equations for efficiently recalculating the MWF-BF based on the previous BF by applying the block-matrix inversion formula [17].

In the current contribution, we address the problem of reducing the computational burden of recalculating the LCMV-BF when modifying the group of sensors which participate in the spatial filtering, denoted as the active sensors or nodes, or when modifying the constraints set. Here we assume that the required BF updates are subjected to a controlling mechanism, referred to as the *oracle*. The decisions of the oracle can be motivated by optimizing the tradeoff between performance and resource usage, handling arbitrary link failures, and also by determining the desired response for the various sources, which result in updating of the constraints set. Updating the configuration of the active sensors could affect the desired constraints set. For example, adding sensors increases the dimension of the received signals and allows for the application of a larger number of constraints. The decision mechanism of the oracle is out of the scope of the current contribution.

In this paper, we propose a set of lower complexity procedures for updating the group of active sensors, and the constraints set to a given LCMV-BF. We derive the updating procedures for both the LCMV closed-form BF and its respective GSC form. The proposed procedures reduce the computational complexity, and are equivalent to the straightforward calculation of the LCMV-BF.

The paper is organized as follows. In Section II the problem is formulated. In Section III four examples for updating procedures of the LCMV-BF are fully derived. Later in the Appendix A all eight updating procedures are summarized. In Section IV we discuss extending the derived algorithms for adding or removing a group of sensors or constraints. The computational complexity of the proposed procedures are analyzed and compared with the complexity of their corresponding straightforward BF in Section V.

II. PROBLEM FORMULATION

Consider P point source signals, some stationary and other nonstationary, denoted by $s^1(\ell, k), \dots, s^P(\ell, k)$, propagating in a multipath environment and impinging on an array comprising M sensors. The problem is formulated using a narrow-band model in the short time Fourier transform (STFT) domain, where ℓ is the frame index and k is the frequency index. From hereon, the frequency notation is omitted for brevity. The application and the calculation of the BF should be interpreted frequency-wise. The TF relating the p th source and the m th sensor is denoted by $h_m^p(\ell)$. Define in vector notation

$$\mathbf{s}(\ell) = [s^1(\ell) \quad \dots \quad s^P(\ell)]^T \quad (1a)$$

$$\mathbf{h}^p(\ell) = [h_1^p(\ell) \quad \dots \quad h_M^p(\ell)]^T; \quad p = 1, \dots, P \quad (1b)$$

$$\mathbf{H}(\ell) = [\mathbf{h}^1(\ell) \quad \dots \quad \mathbf{h}^P(\ell)]. \quad (1c)$$

The received signals vector and its covariance matrix are given by

$$\mathbf{z}(\ell) = \mathbf{H}\mathbf{s}(\ell) + \mathbf{v}(\ell) \quad (2a)$$

$$\begin{aligned} \Phi &= \mathbf{E}[\mathbf{z}(\ell)\mathbf{z}^\dagger(\ell)] \\ &= \mathbf{H}\Sigma\mathbf{H}^\dagger + \Phi_{vv} \end{aligned} \quad (2b)$$

where $\mathbf{v}(\ell)$ denotes the total received interferences vector of the noncoherent signals, Σ denotes the diagonal covariance matrix of the coherent sources (assuming they are statistically independent), and $\Phi_{vv} = \mathbf{E}[\mathbf{v}(\ell)\mathbf{v}^\dagger(\ell)]$ is the covariance matrix of $\mathbf{v}(\ell)$.

Consider a general P th-order constraints set

$$\mathbf{C} = [\mathbf{c}^1 \quad \dots \quad \mathbf{c}^P] \quad (3a)$$

$$\mathbf{g} = [g^1 \quad \dots \quad g^P]^T. \quad (3b)$$

The optimization criterion of the LCMV-BF is given by

$$\mathbf{w} = \arg \min_{\mathbf{C}^\dagger \mathbf{w} = \mathbf{g}} \mathbf{w}^\dagger \Phi \mathbf{w}. \quad (4)$$

The closed-form LCMV-BF is described in Section II-A, and the efficient GSC implementation is described in Section II-B.

A. Closed-Form LCMV-BF

This BF form is obtained by solving (4) directly using Lagrange multipliers. The closed-form LCMV-BF solution to the problem is given by

$$\mathbf{w} = \Phi^{-1} \mathbf{C} \mathbf{Q}^{-1} \mathbf{g} \quad (5)$$

where \mathbf{Q} is defined as follows:

$$\mathbf{Q} = \mathbf{C}^\dagger \Phi^{-1} \mathbf{C}. \quad (6)$$

We denote the solution in (5) as the straightforward LCMV (SF-LCMV). Its computational complexity is mainly dominated by the two matrix inversion Φ^{-1} , and \mathbf{Q}^{-1} .

In the following sections, we derive algorithms for updating an existing LCMV-BF. We consider two types of updates. The first type is sensor updates and the second type is constraint set updates. For each type of update we derive two procedures. The first procedure, denoted as the incremental procedure, refers to adding either a sensor or a constraint to an existing BF. The second procedure, denoted as the decremental procedure, refers to removing either a sensor or constraint from an existing BF. The derived procedures reduce the dimensions of the matrices to be inverted, and hence reduce the computational complexity substantially.

B. GSC-Form LCMV-BF

This BF form is obtained by splitting the applied filters into two components, i.e., $\mathbf{w} = \mathbf{w}_\parallel - \mathbf{w}_\perp$. The components \mathbf{w}_\parallel and \mathbf{w}_\perp lie in the column-subspace of the constraint matrix \mathbf{C} and its complement null-subspace, respectively. The GSC formulation of the problem is given by

$$\mathbf{w} = \mathbf{w}_\parallel - \mathbf{B}\mathbf{f} \quad (7a)$$

$$\mathbf{w}_\parallel = \mathbf{C}\mathbf{R}^{-1}\mathbf{g} \quad (7b)$$

$$\mathbf{f} = (\mathbf{B}^\dagger \Phi \mathbf{B})^{-1} \mathbf{B}^\dagger \Phi \mathbf{w}_\parallel \quad (7c)$$

$$\mathbf{R} = \mathbf{C}^\dagger \mathbf{C}. \quad (7d)$$

The GSC form is decomposed into two branches. The upper branch, also known as the quiescent BF, is denoted by \mathbf{w}_\parallel . It is responsible for maintaining the constraints set. The lower branch is comprised of two parts: the blocking matrix (BM) and the subsequent noise canceller (NC) denoted by \mathbf{B} and \mathbf{f} , respectively. The objective of the BM is to block the signals arriving from the constraints set subspace and generate $M - P$ interference-only reference signals. Its dimensions are $M \times (M - P)$, and it can be calculated, for example, by applying the singular value decomposition (SVD) to the constraints matrix \mathbf{C} [18]. We will assume that all the columns of the BM are orthogonal, as in [7]. Note that an orthogonal BM can always be constructed. The NC uses the reference signals from the output of the BM to estimate the noise component at the output of the quiescent BF and therefore reduce its level. We denote the GSC-form BF in (7) as the straightforward GSC (SF-GSC) BF.

The computational complexity of the SF-GSC BF is mainly dominated by the SVD used for constructing the BM, and by the matrix inversion \mathbf{R}^{-1} . In the following sections, we derive algorithms for updating an existing GSC-BF. We consider two types of updates. The first type is sensor updates and the second type is constraint set updates. For each type of update we derive incremental and decremental procedures which circumvent the SVD and the matrix inversion, and hence reduce the computational complexity substantially. The NC is usually implemented as an adaptive NC (ANC) using the least mean squares (LMS) algorithm [6]. The LMS algorithm consumes $\mathcal{O}(M)$ operations per frequency bin. Due to its low complexity and adaptive nature, it is unnecessary to formulate an update procedure to the ANC.

Please note that in the following sections some notations may be re-defined for brevity. Explicitly, when considering sensor addition or removal, a constraints set comprising P constraints is assumed. Also, when considering constraint addition or removal, an array comprising M sensors is assumed.

III. LOW-COMPLEXITY BEAMFORMER UPDATING METHODS

Algorithms for adding or removing a single constraint to the LCMV-BF and the associated GSC implementation are now derived. The algorithms are denoted by $[S \setminus C]U[I \setminus D] - [GSC \setminus LCMV]$, where $S \setminus C$ stands for sensor or constraint, respectively, U stands for update, $I \setminus D$ stands for incremental or decremental, respectively, and $GSC \setminus LCMV$ stands for the GSC or the closed-form implementations, respectively. For example, the sensor update incremental closed-form implementation algorithm is denoted by SUI-LCMV. For brevity we do not derive all eight algorithms in details. Instead, we chose to elaborate on the derivation of four representative procedures, namely the SUI-LCMV, SUI-GSC, CUI-GSC and the SUD-GSC in the following sub-sections. The derivation of the other algorithms is based on the same methods. A summary of all eight algorithms is given in Appendix A.

A. Derivation of the SUI-LCMV Algorithm

Assume an $M - 1$ sensors and P constraints LCMV-BF is active

$$\mathbf{w} = \Phi^{-1} \mathbf{C} \mathbf{Q}^{-1} \mathbf{g} \quad (8)$$

where \mathbf{C} is the $(M - 1) \times P$ constraints matrix, \mathbf{g} is the $P \times 1$ desired response vector, and the constraints set is

$$\mathbf{C}^\dagger \mathbf{w} = \mathbf{g}. \quad (9)$$

Now, a new sensor (indexed M) becomes available. Define the augmented constraints set

$$\dot{\mathbf{C}} = \begin{bmatrix} \mathbf{C} \\ \dot{\mathbf{c}}^\dagger \end{bmatrix} \quad (10)$$

where $\dot{\mathbf{c}}$ is a $P \times 1$ vector extending the constraints set to M sensors.

The covariance matrix of the M sensors is given by

$$\dot{\Phi} = \begin{bmatrix} \Phi & \dot{\phi} \\ \dot{\phi}^\dagger & \dot{\sigma}^2 \end{bmatrix}. \quad (11)$$

Applying the block matrix inversion formula [17], the inverse of the covariance matrix equals

$$\dot{\Phi}^{-1} = \left[\begin{array}{c|c} \Phi^{-1} + \dot{\epsilon} \Phi^{-1} \dot{\phi} \dot{\phi}^\dagger \Phi^{-1} & -\dot{\epsilon} \Phi^{-1} \dot{\phi} \\ \hline -\dot{\epsilon} \dot{\phi}^\dagger \Phi^{-1} & \dot{\epsilon} \end{array} \right] \quad (12)$$

where

$$\dot{\phi} = \mathbb{E} [[z_1 \ \cdots \ z_{M-1}]^T z_M^*] \quad (13a)$$

$$\dot{\sigma}^2 = \mathbb{E} [z_M z_M^*] \quad (13b)$$

$$\dot{\epsilon} = (\dot{\sigma}^2 - \dot{\phi}^\dagger \Phi^{-1} \dot{\phi})^{-1}. \quad (13c)$$

Considering the definition of \mathbf{Q} in (6), the updated $\dot{\mathbf{Q}}$ in terms of \mathbf{Q} is given by:

$$\begin{aligned} \dot{\mathbf{Q}} &= \dot{\mathbf{C}}^\dagger \dot{\Phi}^{-1} \dot{\mathbf{C}} \\ &= [\mathbf{C}^\dagger | \dot{\mathbf{c}}] \\ &\quad \cdot \left[\begin{array}{c|c} \Phi^{-1} + \dot{\epsilon} \Phi^{-1} \dot{\phi} \dot{\phi}^\dagger \Phi^{-1} & -\dot{\epsilon} \Phi^{-1} \dot{\phi} \\ \hline -\dot{\epsilon} \dot{\phi}^\dagger \Phi^{-1} & \dot{\epsilon} \end{array} \right] \\ &\quad \cdot \begin{bmatrix} \mathbf{C} \\ \dot{\mathbf{c}}^\dagger \end{bmatrix} \end{aligned} \quad (14)$$

$$= \mathbf{Q} + \dot{\epsilon} \dot{\mathbf{q}} \dot{\mathbf{q}}^\dagger \quad (15)$$

where

$$\dot{\mathbf{q}} = \mathbf{C}^\dagger \Phi^{-1} \dot{\phi} - \dot{\mathbf{c}}. \quad (16)$$

Applying the Woodbury identity [19] to the inverse of (15), $\dot{\mathbf{Q}}^{-1}$ equals

$$\dot{\mathbf{Q}}^{-1} = \mathbf{Q}^{-1} - \frac{\mathbf{Q}^{-1} \dot{\mathbf{q}} \dot{\mathbf{q}}^\dagger \mathbf{Q}^{-1}}{\dot{\epsilon}^{-1} + \dot{\mathbf{q}}^\dagger \mathbf{Q}^{-1} \dot{\mathbf{q}}}. \quad (17)$$

Finally, the updated BF, $\dot{\mathbf{w}}$, is given in terms of the previous BF terms, \mathbf{w} , \mathbf{Q} , \mathbf{Q}^{-1} , Φ , and Φ^{-1} , by substituting (12), (17) in (5)

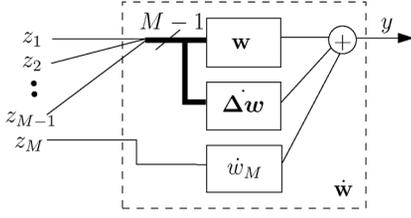


Fig. 1. Block-diagram of the SUI-LCMV procedure.

$$\dot{\mathbf{w}} = \begin{bmatrix} \mathbf{w} + \Delta \mathbf{w} \\ \dot{w}_M \end{bmatrix} \quad (18a)$$

$$\Delta \mathbf{w} = - \frac{\dot{\mathbf{q}}^\dagger \mathbf{Q}^{-1} \mathbf{g}}{\dot{\epsilon}^{-1} + \dot{\mathbf{q}}^\dagger \mathbf{Q}^{-1} \dot{\mathbf{q}}} \Phi^{-1} \mathbf{C} \mathbf{Q}^{-1} \dot{\mathbf{q}} - \dot{w}_M \Phi^{-1} \dot{\phi} \quad (18b)$$

$$\dot{w}_M = - \dot{\epsilon} (\dot{\phi}^\dagger \Phi^{-1} \mathbf{C} \mathbf{Q}^{-1} - \dot{\mathbf{c}}^\dagger \mathbf{Q}^{-1}) \cdot \left(\mathbf{I}_{P \times P} - \frac{\dot{\mathbf{q}} \dot{\mathbf{q}}^\dagger \mathbf{Q}^{-1}}{\dot{\epsilon}^{-1} + \dot{\mathbf{q}}^\dagger \mathbf{Q}^{-1} \dot{\mathbf{q}}} \right) \mathbf{g} \quad (18c)$$

A block-diagram of the SUI-LCMV algorithm is depicted in Fig. 1. The procedure is summarized in Alg. 1.

B. Derivation of the SUI-GSC Algorithm

Similarly to (7), suppose now that an $M-1$ sensors GSC BF maintaining P constraints is given by

$$\mathbf{w} = \mathbf{w}_{\parallel} - \mathbf{B} \mathbf{f} \quad (19a)$$

$$\mathbf{w}_{\parallel} = \mathbf{C} \mathbf{R}^{-1} \mathbf{g} \quad (19b)$$

$$\mathbf{f} = (\mathbf{B}^\dagger \Phi \mathbf{B})^{-1} \mathbf{B}^\dagger \Phi \mathbf{w}_{\parallel} \quad (19c)$$

$$\mathbf{R} = \mathbf{C}^\dagger \mathbf{C} \quad (19d)$$

where we assume that the ANC has converged to \mathbf{f} . The latter filter is the appropriate Wiener filter for estimating the noise component at the output of the quiescent BF based on the noise references at the output of the BM. We further assume that \mathbf{B} is an appropriate $(M-1) \times (M-1-P)$ BM. The BM can be calculated for example by using the SVD of \mathbf{C} [18]. We assume that the BM is orthogonal, i.e., $\mathbf{B}^\dagger \mathbf{B} = \mathbf{I}_{(M-1-P) \times (M-1-P)}$.

Consider adding the M th sensor and updating the BF. The updated constraints set is defined as in (10). The updated $\dot{\mathbf{R}}$ matrix is given by substituting (10) in (19d)

$$\dot{\mathbf{R}} = \mathbf{R} + \dot{\mathbf{c}} \dot{\mathbf{c}}^\dagger. \quad (20)$$

Applying the Woodbury identity to the inverse of (20), $\dot{\mathbf{R}}^{-1}$ is given by

$$\dot{\mathbf{R}}^{-1} = \mathbf{R}^{-1} - \frac{\mathbf{R}^{-1} \dot{\mathbf{c}} \dot{\mathbf{c}}^\dagger \mathbf{R}^{-1}}{1 + \dot{\mathbf{c}}^\dagger \mathbf{R}^{-1} \dot{\mathbf{c}}}. \quad (21)$$

The M sensors quiescent BF is given, similarly to (19b), by replacing \mathbf{C} and \mathbf{R}^{-1} with $\dot{\mathbf{C}}$ and $\dot{\mathbf{R}}^{-1}$ from (10), (21):

$$\dot{\mathbf{w}}_{\parallel} = \begin{bmatrix} \mathbf{w}_{\parallel} + \Delta \mathbf{w}_{\parallel} \\ \dot{w}_{\parallel M} \end{bmatrix} \quad (22a)$$

$$\dot{w}_{\parallel M} = \frac{\dot{\mathbf{c}}^\dagger \mathbf{R}^{-1} \mathbf{g}}{1 + \dot{\mathbf{c}}^\dagger \mathbf{R}^{-1} \dot{\mathbf{c}}} \quad (22b)$$

$$\Delta \mathbf{w}_{\parallel} = - \dot{w}_{\parallel M} \mathbf{C} \mathbf{R}^{-1} \dot{\mathbf{c}}. \quad (22c)$$

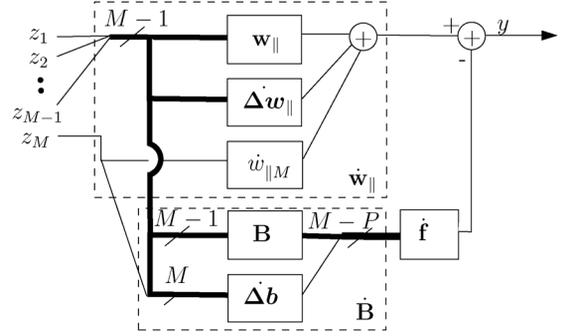


Fig. 2. Block-Diagram of the SUI-GSC procedure.

Next, we address the problem of updating the BM. Since we added the M th sensor, there should be $M-P$ signals at the output of the BM. The updated BM, $\dot{\mathbf{B}}$, should block the signal subspace, i.e., $\dot{\mathbf{B}}^\dagger \dot{\mathbf{C}} = \mathbf{0}$. The first $M-P-1$ reference signals are equivalent to the older ones. This can be verified by adding a row of zeros to \mathbf{B} , i.e. $[\mathbf{0}_{1 \times (M-P-1)} \mathbf{B}^\dagger]^\dagger \dot{\mathbf{C}} = \mathbf{B}^\dagger \mathbf{C} = \mathbf{0}_{(M-P-1) \times P}$. We suggest to use

$$\Delta \dot{\mathbf{b}} = \frac{[\dot{\mathbf{c}}^\dagger \mathbf{R}^{-1} \mathbf{C}^\dagger - 1]^\dagger}{\|[\dot{\mathbf{c}}^\dagger \mathbf{R}^{-1} \mathbf{C}^\dagger - 1]\|} \quad (23)$$

as the $(M-P)$ th column of the updated BM. $\Delta \dot{\mathbf{b}}$ is orthogonal to the first $M-P-1$ columns of $\dot{\mathbf{B}}$ since

$$\begin{bmatrix} \mathbf{B} \\ \mathbf{0}_{1 \times (M-P-1)} \end{bmatrix}^\dagger \Delta \dot{\mathbf{b}} = \frac{\mathbf{B}^\dagger \mathbf{C} \mathbf{R}^{-1} \dot{\mathbf{c}}}{\|[\dot{\mathbf{c}}^\dagger \mathbf{R}^{-1} \mathbf{C}^\dagger - 1]\|} - \mathbf{0}_{(M-P-1) \times 1} = \mathbf{0}_{(M-P-1) \times 1}$$

where the last transition is again due to $\mathbf{B}^\dagger \mathbf{C} = \mathbf{0}_{(M-P-1) \times P}$. $\Delta \dot{\mathbf{b}}$ is also orthogonal to $\dot{\mathbf{C}}$ since

$$\begin{aligned} \Delta \dot{\mathbf{b}}^\dagger \dot{\mathbf{C}} &= \frac{[\dot{\mathbf{c}}^\dagger \mathbf{R}^{-1} \mathbf{C}^\dagger - 1]^\dagger [\dot{\mathbf{C}}^\dagger]}{\|[\dot{\mathbf{c}}^\dagger \mathbf{R}^{-1} \mathbf{C}^\dagger - 1]\|} \\ &= \frac{\dot{\mathbf{c}}^\dagger \mathbf{R}^{-1} \mathbf{C}^\dagger \mathbf{C} - \dot{\mathbf{c}}^\dagger}{\|[\dot{\mathbf{c}}^\dagger \mathbf{R}^{-1} \mathbf{C}^\dagger - 1]\|} \\ &= \mathbf{0}_{1 \times P} \end{aligned}$$

where the last transition is due to the definition of \mathbf{R} in (19d). Therefore, augmenting \mathbf{B} by $\Delta \dot{\mathbf{b}}$ is a proper BM of P constraints

$$\dot{\mathbf{B}} = \begin{bmatrix} \mathbf{B} \\ \mathbf{0}_{1 \times M-1-P} \end{bmatrix} \Delta \dot{\mathbf{b}}. \quad (24)$$

After updating the quiescent BF and the BM, another reference signal is added. In the general case the new reference signal and the previous reference signals are correlated. Therefore, not only the NC filter of the new reference signal needs to be determined, but also the NC filters of the previous reference signals need to be adjusted. As mentioned earlier, we rely on the low complexity and fast convergence of the LMS algorithm for updating the NC coefficients. The resulting NC after convergence is given by substituting (11), (22a), (24) in (19c)

$$\dot{\mathbf{f}} = (\dot{\mathbf{B}}^\dagger \Phi \dot{\mathbf{B}})^{-1} \dot{\mathbf{B}}^\dagger \Phi \dot{\mathbf{w}}_{\parallel}. \quad (25)$$

A block-diagram of the SUI-GSC algorithm is depicted in Fig. 2. The algorithm is summarized in Alg. 5.

C. Derivation of the CUI-GSC Algorithm

Suppose that an M sensors $P-1$ constraints GSC BF is given by

$$\mathbf{w} = \mathbf{w}_{\parallel} - \mathbf{B}\mathbf{f} \quad (26)$$

$$\mathbf{w}_{\parallel} = \mathbf{C}\mathbf{R}^{-1}\mathbf{g} \quad (27)$$

$$\mathbf{f} = (\mathbf{B}^{\dagger}\Phi\mathbf{B})^{-1}\mathbf{B}^{\dagger}\Phi\mathbf{w}_{\parallel} \quad (28)$$

$$\mathbf{R} = \mathbf{C}^{\dagger}\mathbf{C} \quad (29)$$

where \mathbf{C} is the $M \times (P-1)$ constraints matrix, \mathbf{g} is the $(P-1) \times 1$ desired response vector and \mathbf{B} is an appropriate $M \times (M-P+1)$ BM. As was previously stated, we assume that the BM is orthogonal, i.e. $\mathbf{B}^{\dagger}\mathbf{B} = \mathbf{I}_{(M-P+1) \times (M-P+1)}$.

Consider adding the P th constraint and updating the BF. The updated constraints set is

$$\ddot{\mathbf{C}} = [\mathbf{C}|\ddot{\mathbf{c}}] \quad (30a)$$

$$\ddot{\mathbf{g}} = [\mathbf{g}^{\dagger}|\ddot{g}^*]^{\dagger}. \quad (30b)$$

Updating the matrix \mathbf{R} in (29) with the P th constraint yields

$$\ddot{\mathbf{R}} = \left[\begin{array}{c|c} \mathbf{R} & \ddot{\mathbf{r}} \\ \hline \ddot{\mathbf{r}}^{\dagger} & \|\ddot{\mathbf{c}}\|^2 \end{array} \right] \quad (31)$$

where $\ddot{\mathbf{r}} = \mathbf{C}^{\dagger}\ddot{\mathbf{c}}$. The inverse of $\ddot{\mathbf{R}}$ is given by applying the block matrix inversion formula

$$\ddot{\mathbf{R}}^{-1} = \left[\begin{array}{c|c} \mathbf{R}^{-1} + \ddot{\rho}\mathbf{R}^{-1}\ddot{\mathbf{r}}\ddot{\mathbf{r}}^{\dagger}\mathbf{R}^{-1} & -\ddot{\rho}\mathbf{R}^{-1}\ddot{\mathbf{r}} \\ \hline -\ddot{\rho}\ddot{\mathbf{r}}^{\dagger}\mathbf{R}^{-1} & \ddot{\rho} \end{array} \right] \quad (32)$$

where

$$\ddot{\rho} = (\|\ddot{\mathbf{c}}\|^2 - \ddot{\mathbf{r}}^{\dagger}\mathbf{R}^{-1}\ddot{\mathbf{r}})^{-1}. \quad (33)$$

The updated quiescent BF designed to maintain the P constraints set is given by substituting the updated values of $\ddot{\mathbf{R}}^{-1}$, $\ddot{\mathbf{C}}$ and $\ddot{\mathbf{g}}$ from (32), (30a), (30b) in (27)

$$\ddot{\mathbf{w}}_{\parallel} = \mathbf{w}_{\parallel} + \Delta\ddot{\mathbf{w}}_{\parallel} \quad (34a)$$

$$\Delta\ddot{\mathbf{w}}_{\parallel} = \ddot{\rho}(\ddot{g} - \ddot{\mathbf{c}}^{\dagger}\mathbf{w}_{\parallel})(\mathbf{I} - \mathbf{C}\mathbf{R}^{-1}\mathbf{C}^{\dagger})\ddot{\mathbf{c}}. \quad (34b)$$

Next, we update the BM. Notice that the rank of the BM equals the number of sensors minus the number of constraints (assuming the constraints set are linearly independent), i.e. $M-P+1$. Therefore, the rank of the BM corresponding to the modified constraints set is smaller by one than that of the former BM. Hence, we would like to reduce the dimensions of the current BM to $M \times (M-P)$ such that its columns are an orthogonal set and that

$$\ddot{\mathbf{B}}^{\dagger}\ddot{\mathbf{C}} = \mathbf{0}_{(M-P) \times P}. \quad (35)$$

The new constraint vector $\ddot{\mathbf{c}}$ can be written as a combination of two components

$$\ddot{\mathbf{c}} = (\mathbf{I} - \mathbf{B}\mathbf{B}^{\dagger})\ddot{\mathbf{c}} + \mathbf{B}\mathbf{B}^{\dagger}\ddot{\mathbf{c}}. \quad (36)$$

The first component lies in the $P-1$ constraints subspace, and the second component lies in its corresponding null-subspace, hence spanned by the columns of \mathbf{B} . The new BM $\ddot{\mathbf{B}}$ should block both \mathbf{C} and $\mathbf{B}\mathbf{B}^{\dagger}\ddot{\mathbf{c}}$, the component of $\ddot{\mathbf{c}}$ not spanned by

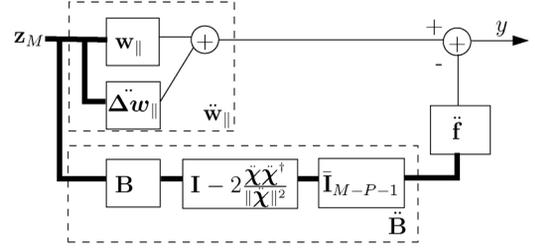


Fig. 3. Block-diagram of the CUI-GSC procedure.

the columns of \mathbf{C} . This can be obtained by: 1) rotating the current BM such that all but one of its columns are orthogonal to the second component of $\ddot{\mathbf{c}}$; 2) deleting that column. The Householder transformation [20] can be applied to satisfy both requirements. The transformed BM is given by

$$\tilde{\mathbf{B}} = \mathbf{B} \left(\mathbf{I}_{(M-P+1) \times (M-P+1)} - \frac{2\ddot{\chi}\ddot{\chi}^{\dagger}}{\|\ddot{\chi}\|^2} \right) \quad (37)$$

where $\ddot{\chi}$ is defined as

$$\ddot{\chi} = \frac{\ddot{\mathbf{b}}}{\|\ddot{\mathbf{b}}\|} + \exp(j\angle\ddot{b}_{M-P+1})\mathbf{i}_{M-P+1} \quad (38)$$

$\angle(\cdot)$ denotes the angle extraction of a complex number $\mathbf{i}_{M-P+1} = [\mathbf{0}_{1 \times (M-P+1)} | 1]^T$, $\ddot{\mathbf{b}}$ is the projection of $\ddot{\mathbf{c}}$ onto \mathbf{B} , i.e. $\ddot{\mathbf{b}} = \mathbf{B}^{\dagger}\ddot{\mathbf{c}}$, and \ddot{b}_{M-P+1} is the last entry of $\ddot{\mathbf{b}}$. It follows that:

$$\tilde{\mathbf{B}}^{\dagger}\mathbf{C} = \mathbf{0}_{(M-P+1) \times (P-1)} \quad (39a)$$

$$\tilde{\mathbf{B}}^{\dagger}\ddot{\mathbf{c}} = -\exp(j\angle\ddot{b}_{M-P+1})\mathbf{i}_{M-P+1}. \quad (39b)$$

Note that since the Householder transformation is unitary, the rotated basis remains orthogonal. The orthogonality property of $\tilde{\mathbf{B}}$ is imperative for assuring that all columns of $\tilde{\mathbf{B}}$ but the last one are orthogonal to $\mathbf{B}\mathbf{B}^{\dagger}\ddot{\mathbf{c}}$. Finally, the updated BM is obtained by deleting the last column of $\tilde{\mathbf{B}}$:

$$\ddot{\mathbf{B}} = \tilde{\mathbf{B}}\mathbf{I}_{M-P}, \quad (40)$$

where \mathbf{I}_m is an $(m-1) \times m$ matrix constructed by removing the last row of the identity matrix $\mathbf{I}_{m \times m}$.

In a similar manner to the NC update of the SUI-GSC procedure in Section III-B, the updated NC filters after convergence are given in a vector form by substituting (40), (2b), (34a) in (28)

$$\ddot{\mathbf{f}} = (\ddot{\mathbf{B}}^{\dagger}\Phi\ddot{\mathbf{B}})^{-1}\ddot{\mathbf{B}}^{\dagger}\Phi\ddot{\mathbf{w}}_{\parallel}. \quad (41)$$

A block-diagram of the CUI-GSC algorithm is depicted in Fig. 3. The algorithm is summarized in Alg. 7.

D. Derivation of the SUD-GSC Algorithm

Suppose that an M sensors and P constraints GSC-BF is given by

$$\dot{\mathbf{w}} = \dot{\mathbf{w}}_{\parallel} - \dot{\mathbf{B}}\dot{\mathbf{f}} \quad (42a)$$

$$\dot{\mathbf{w}}_{\parallel} = \dot{\mathbf{C}}\dot{\mathbf{R}}^{-1}\dot{\mathbf{g}} \quad (42b)$$

where $\dot{\mathbf{C}}$, $\dot{\mathbf{R}}$ are defined as in (10) and (20), respectively, and $\dot{\mathbf{B}}$ is an $M \times (M-P)$ orthogonal BM. Now, consider that the M th

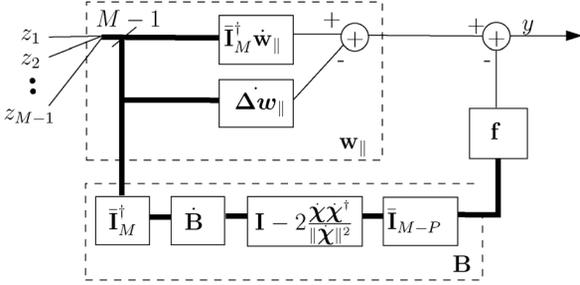


Fig. 4. Block-diagram of the SUD-GSC procedure.

sensor becomes unavailable. Here we derive the equations for updating the BF using its previous value. The updated \mathbf{R}^{-1} is given by applying the Woodbury identity to the inverse of (20)

$$\mathbf{R}^{-1} = \dot{\mathbf{R}}^{-1} + \frac{\dot{\mathbf{R}}^{-1} \dot{\mathbf{c}} \dot{\mathbf{c}}^\dagger \dot{\mathbf{R}}^{-1}}{1 - \dot{\mathbf{c}}^\dagger \dot{\mathbf{R}}^{-1} \dot{\mathbf{c}}}. \quad (43)$$

Substituting (43), (10) in (42b) yields

$$\mathbf{w}_{\parallel} = \bar{\mathbf{I}}_M^\dagger \dot{\mathbf{w}}_{\parallel} - \Delta \dot{\mathbf{w}}_{\parallel} \quad (44a)$$

$$\Delta \dot{\mathbf{w}}_{\parallel} = -\dot{w}_{\parallel M} \mathbf{C} \mathbf{R}^{-1} \dot{\mathbf{c}} \quad (44b)$$

$$\dot{w}_{\parallel M} = \dot{\mathbf{i}}_M^\dagger \dot{\mathbf{w}}_{\parallel}. \quad (44c)$$

Next we address updating the BM. We apply the Householder transformation step and diagonalize the last row of $\dot{\mathbf{B}}$. Define

$$\dot{\boldsymbol{\chi}} = \dot{\mathbf{B}}^\dagger \dot{\mathbf{i}}_M + \|\dot{\mathbf{B}}^\dagger \dot{\mathbf{i}}_M\| \exp(-j\angle \dot{B}_{M,M-P}) \dot{\mathbf{i}}_{M-P} \quad (45)$$

where $\dot{B}_{M,M-P}$ is the $(M, M-P)$ entry in $\dot{\mathbf{B}}$. The rotated BM is given by

$$\tilde{\mathbf{B}} = \dot{\mathbf{B}} \left(\mathbf{I} - \frac{2\dot{\boldsymbol{\chi}}\dot{\boldsymbol{\chi}}^\dagger}{\|\dot{\boldsymbol{\chi}}\|^2} \right). \quad (46)$$

It can be verified that the last row of $\tilde{\mathbf{B}}$ equals

$$\dot{\mathbf{i}}_M^\dagger \tilde{\mathbf{B}} = \left[\mathbf{0}_{1 \times (M-P-1)} \mid -\|\dot{\mathbf{B}}^\dagger \dot{\mathbf{i}}_M\| \exp(j\angle \dot{B}_{M,M-P}) \right]. \quad (47)$$

Since the Householder transformation is unitary, the orthogonality of the BM is kept. The rotated matrix keeps blocking the original constraints matrix, i.e. $\tilde{\mathbf{B}}^\dagger \dot{\mathbf{C}} = \mathbf{0}_{(M-P) \times P}$. Finally, the updated $(M-1) \times (M-P-1)$ dimensional BM is obtained by deleting the last row and last column of $\tilde{\mathbf{B}}$

$$\mathbf{B} = \bar{\mathbf{I}}_M^\dagger \tilde{\mathbf{B}} \bar{\mathbf{I}}_{M-P} \quad (48)$$

and the NC is given after convergence by (7c).

A block-diagram of the SUD-GSC algorithm is depicted in Fig. 4. The algorithm is summarized in Alg. 6.

IV. ADDING/REMOVING A GROUP OF SENSORS/CONSTRAINTS

In a partially connected sensor network, some nodes are accessible only indirectly through some other nodes. In these networks a change in a single link can affect the activity of multiple nodes. Here we discuss adding or removing multiple nodes or multiple constraints to an LCMV-BF.

Two basic methods are used by the algorithms derived in this paper. The first method is the block matrix inversion formula

TABLE I
COMPLEXITY OF BASIC OPERATIONS

Operation	Computations
Matrix mult. $(m \times n) \cdot (n \times p)$	mnp
Matrix inversion $n \times n$	$\frac{2}{3}n^3$
SVD $m \times n$	$4m^2n + 8mn^2 + 9n^3$

[17] which is used for inverting an $m \times m$ matrix based on the already calculated inverse of an $(m-1) \times (m-1)$ submatrix. Two strategies can be adopted in the application of the block matrix inversion formula in cases of $k > 1$ sensor/constraint updates. One strategy utilizes k sequential updates as derived previously. An alternative strategy uses the more general version of the block matrix inversion formula. Namely, the inverse of the $(m-k) \times (m-k)$ sub-matrix is utilized in the inversion of the $m \times m$ matrix. The latter strategy results in more cumbersome expressions. As both strategies involve equivalent computational burden, the sequential strategy of multiple updates is preferred.

The second method used in this paper, is the Householder transformation step [20], which is used for rotating an orthogonal basis such that all of its new basis vectors but one are orthogonal to a predefined vector. A sequence of Householder transformation steps can be applied for multiple sensor/constraint updates. The detailed derivation of these algorithms, as well as their complexity analysis, is out of the scope of the current contribution.

V. COMPLEXITY EVALUATION

In this section we compare the complexity of the straightforward LCMV-BF closed-form and GSC form implementations with their updated form counterparts. Opposed to the straightforward BFs, the updating procedures rely on calculation results of previous BFs, and therefore impose memory requirements. We consider both computational complexity and memory requirements. The computational analysis is based on the complexity of basic operations [21] defined in Table I.

A summary of the complexity of the compared BF is given in Table II. The proposed updating procedures reduce the computational complexity of the SF-LCMV implementation, which is $\mathcal{O}(M^3 + M^2P)$, to $\mathcal{O}(M^2 + MP)$ while increasing the memory requirement to $\mathcal{O}(M^2 + P^2)$. Similarly, regarding the GSC implementation, the updating procedures reduce the computational complexity from $\mathcal{O}(P^3 + M^2P + P^2M)$ in the straightforward implementation to $\mathcal{O}(M^2 + MP)$, while increasing the memory requirements to $\mathcal{O}(P^2)$. Please note that the computational complexity of the LCMV and the GSC updating procedures is similar, whereas the memory requirement of the GSC procedures is much lower than its LCMV form counterparts. The number of computations versus the number of sensors while the number of constraint is fixed to $P = 10$ is depicted in Figs. 5 and 6 for the closed-form and for the GSC form implementations, respectively. It is evident that the updating procedures impose a lower computational burden. The number of computations versus the number of constraints while the number of sensors is fixed to $M = 20$ is depicted in Figs. 7 and 8. Again, it is evident that the updating procedures impose a lower computational burden. It is interesting to note that the number of computations of the

TABLE II
NUMBER OF COMPUTATIONS AND MEMORY USAGE OF VARIOUS CLOSED-FORM
AND GSC-FORM LCMV-BF

BF	Computations	Memory
SF-LCMV	$\frac{2}{3}M^3 + \frac{2}{3}P^3 + 2M^2P + MP + P^2$	0
SF-GSC	$9\frac{2}{3}P^3 + 4M^2P + 9MP^2 + MP + P^2$	0
SUI-LCMV	$4M^2 + 2MP + 5P^2 - 6M + 11P + 2$	$2M^2 + 2P^2$
SUD-LCMV	$4M^2 + 2MP + 5P^2 - 5M + 5P + 1$	$2M^2 + 2P^2$
SUI-GSC	$MP + 5P^2 + 4M + 2P - 2$	$2P^2$
SUD-GSC	$2M^2 + 7P^2 - 3MP + 5M - 2P - 2$	$2P^2$
CUI-LCMV	$2M^2 + 2MP + 3P^2 + 2M - 4P + 1$	$2M^2 + 2P^2$
CUD-LCMV	$2M^2 + 2MP + 2P^2 + 2M - 2P$	$2M^2 + 2P^2$
CUI-GSC	$4M^2 - 5MP + 6P^2 + 12M - 12P + 6$	$2P^2$
CUD-GSC	$3MP + 3P^2 + 3M - 4P$	$2P^2$

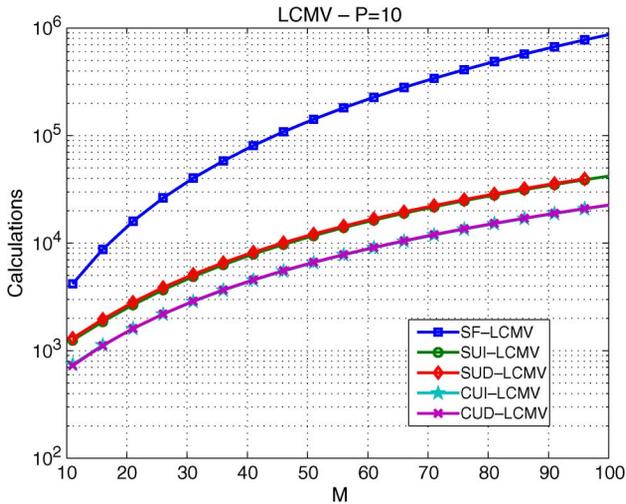


Fig. 5. Number of computations versus M for LCMV-BF with $P = 10$.

CUI-GSC and the SUD-GSC is not monotonically increasing with P . This is attributed to the fact that the dimensions of the BM are reversely proportional to the number of constraints. In many applications, the number of constraints can be increased with the number of available sensors. In Figs. 9 and 10 the computational complexity is depicted versus the number of sensors, while the number of constraints is set to $P = \lfloor \frac{1}{2}M \rfloor$. The complexity reduction is evident from these figures as well.

The overall computational saving is proportional to the BF update rate, whereas the memory complexity is fixed and considerably low. In a dynamically changing network a substantial computational saving is expected. Please notice that even in the case of a single update of the BF, less computations are required when using the proposed updating procedures than in the straightforward recalculation.

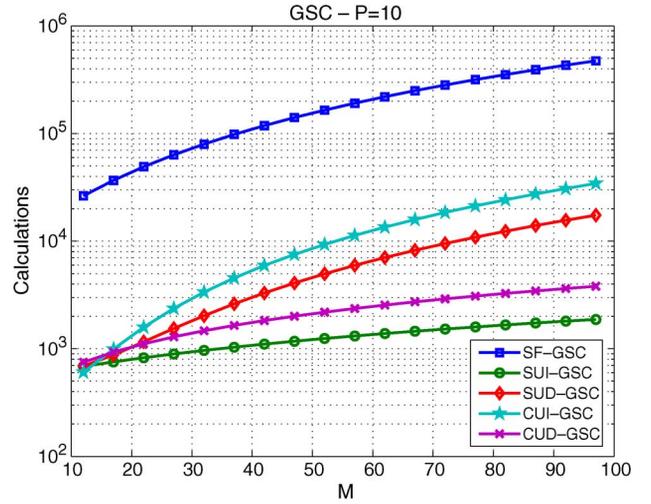


Fig. 6. Number of computations versus M for GSC-BF with $P = 10$.

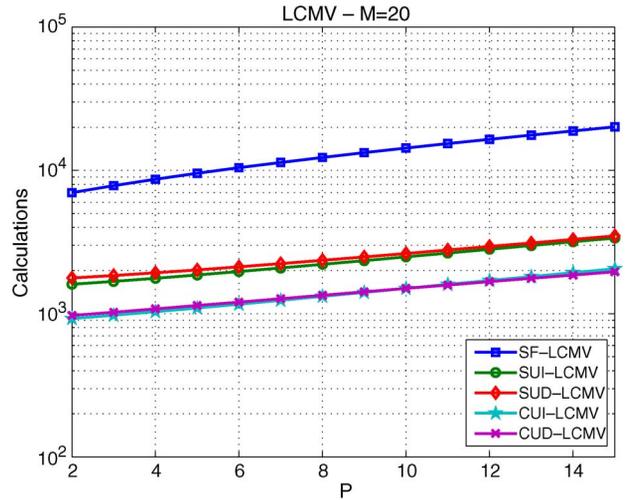


Fig. 7. Number of computations versus P for LCMV-BF with $M = 20$.

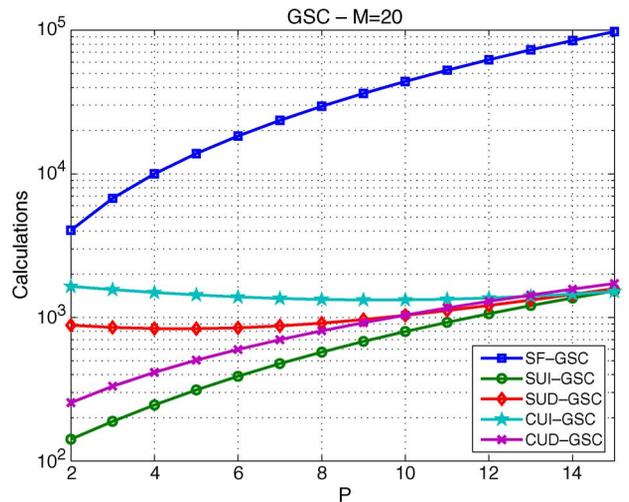


Fig. 8. Number of computations versus P for GSC-BF with $M = 20$.

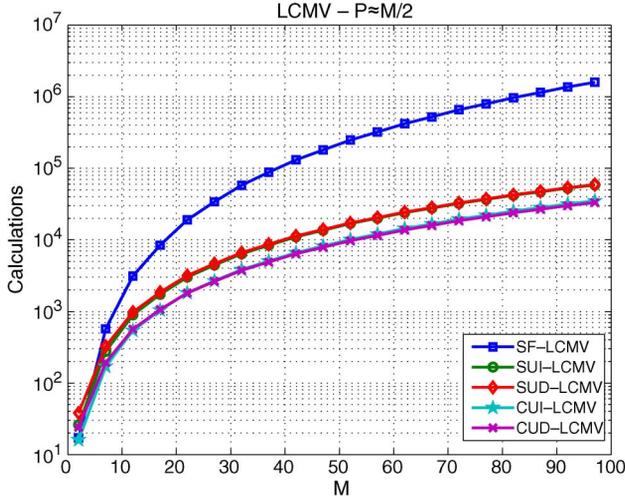


Fig. 9. Number of computations versus M for LCMV-BF with $P = \lfloor \frac{1}{2}M \rfloor$.

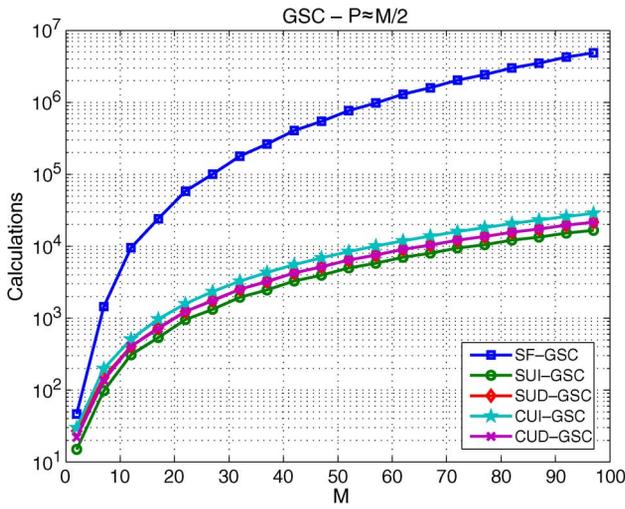


Fig. 10. Number of computations versus M for GSC-BF with $P = \lfloor \frac{1}{2}M \rfloor$.

VI. CONCLUSION

Procedures for adding/removing active sensors or constraints to/from an existing LCMV-BF have been derived. Different procedures were derived for both the closed-form and GSC-form implementations. These procedures use the information of the former BF and save calculations, at the expense of some memory requirements. The computational burden of the proposed procedures was analyzed and compared with the computational burden of their corresponding straightforward BF recalculation. It is evident from the comparison that the number of computations in the proposed procedures is much lower than in straightforward calculation, while the increase in the memory complexity is considerably low. The proposed procedures are beneficial in sensor network applications, where the dynamics of the network and of the environment require frequent updates of the BF, whereas the computational capability is often limited.

APPENDIX A ALGORITHMS SUMMARY

In Section III we derived the SUI-LCMV, SUI-GSC, CUI-GSC, and SUD-GSC algorithms. The derivation was based on

matrix algebra, the Woodbury identity, the block matrix inversion formula and the Householder transformation. We use similar methods to derive the rest of the algorithms, namely the incremental or decremental updates of either the number of sensors or the number of constraints for the GSC or the closed-form implementations. We therefore omit the derivation of the rest of the algorithms for brevity. Instead, in the following, we summarize all the proposed low-complexity beamformer updating methods. The sensor updating algorithms SUI-LCMV, SUD-LCMV, SUI-GSC, SUD-GSC, and the constraint updating algorithms CUI-LCMV, CUD-LCMV, CUI-GSC, CUD-GSC are summarized in Algs. 1, 2, 5, 6, and Algs. 3, 4, 7, 8, respectively.

Algorithm 1: Summary of the SUI-LCMV procedure

input: $\mathbf{C}, \mathbf{g}, \mathbf{w}, \Phi^{-1}, \sigma^2, \mathbf{Q}^{-1}, \dot{\phi}, \dot{\mathbf{c}}$

output: $\dot{\mathbf{C}}, \dot{\mathbf{w}}, \dot{\Phi}^{-1}, \dot{\mathbf{Q}}^{-1}$

begin

$$\dot{\mathbf{C}} = [\mathbf{C}^\dagger | \dot{\mathbf{c}}]^\dagger$$

$$\dot{\epsilon} = (\dot{\sigma}^{-2} - \dot{\phi}^\dagger \Phi^{-1} \dot{\phi})^{-1}$$

$$\dot{\mathbf{q}} = \mathbf{C}^\dagger \Phi^{-1} \dot{\phi} - \dot{\mathbf{c}}$$

$$\dot{\Phi}^{-1} = \begin{bmatrix} \Phi^{-1} + \dot{\epsilon} \Phi^{-1} \dot{\phi} \dot{\phi}^\dagger \Phi^{-1} & -\dot{\epsilon} \Phi^{-1} \dot{\phi} \\ -\dot{\epsilon} \dot{\phi}^\dagger \Phi^{-1} & \dot{\epsilon} \end{bmatrix}$$

$$\dot{\mathbf{Q}}^{-1} = \mathbf{Q}^{-1} - \frac{\mathbf{Q}^{-1} \dot{\mathbf{q}} \dot{\mathbf{q}}^\dagger \mathbf{Q}^{-1}}{\dot{\epsilon}^{-1} + \dot{\mathbf{q}}^\dagger \mathbf{Q}^{-1} \dot{\mathbf{q}}}$$

$$\dot{w}_M = -\dot{\epsilon} (\dot{\phi}^\dagger \Phi^{-1} \mathbf{C} \mathbf{Q}^{-1} - \dot{\mathbf{c}}^\dagger \mathbf{Q}^{-1}) \cdot \left(\mathbf{I}_{P \times P} - \frac{\dot{\mathbf{q}} \dot{\mathbf{q}}^\dagger \mathbf{Q}^{-1}}{\dot{\epsilon}^{-1} + \dot{\mathbf{q}}^\dagger \mathbf{Q}^{-1} \dot{\mathbf{q}}} \right) \mathbf{g}$$

$$\dot{\Delta} \mathbf{w} = -\frac{\dot{\mathbf{q}}^\dagger \mathbf{Q}^{-1} \mathbf{g}}{\dot{\epsilon}^{-1} + \dot{\mathbf{q}}^\dagger \mathbf{Q}^{-1} \dot{\mathbf{q}}} \Phi^{-1} \mathbf{C} \mathbf{Q}^{-1} \dot{\mathbf{q}} - \dot{w}_M \Phi^{-1} \dot{\phi}$$

$$\dot{\mathbf{w}} = \begin{bmatrix} \mathbf{w} + \dot{\Delta} \mathbf{w} \\ \dot{w}_M \end{bmatrix}$$

end

Algorithm 2: Summary of the SUD-LCMV procedure

input: $\dot{\mathbf{C}}, \mathbf{g}, \dot{\mathbf{w}}, \Phi^{-1}, \dot{\mathbf{Q}}^{-1}, \sigma^2, \dot{\phi}$

output: $\mathbf{C}, \mathbf{w}, \Phi, \mathbf{Q}^{-1}$

begin

$$\mathbf{C} = \bar{\mathbf{I}}_M^\dagger \dot{\mathbf{C}}$$

$$\Phi^{-1} = \bar{\mathbf{I}}_M^\dagger \dot{\Phi}^{-1} \bar{\mathbf{I}}_M - \frac{\bar{\mathbf{I}}_M^\dagger \dot{\Phi}^{-1} \mathbf{i}_M \mathbf{i}_M^\dagger \dot{\Phi}^{-1} \bar{\mathbf{I}}_M}{\mathbf{i}_M^\dagger \dot{\Phi}^{-1} \mathbf{i}_M}$$

$$\dot{\epsilon} = (\dot{\sigma}^2 - \dot{\phi}^\dagger \Phi^{-1} \dot{\phi})^{-1}$$

$$\dot{\mathbf{q}} = \mathbf{C}^\dagger \Phi^{-1} \dot{\phi} - \dot{\mathbf{c}}$$

$$\mathbf{Q}^{-1} = \dot{\mathbf{Q}}^{-1} + \frac{\dot{\mathbf{Q}}^{-1} \dot{\mathbf{q}} \dot{\mathbf{q}}^\dagger \dot{\mathbf{Q}}^{-1}}{\dot{\epsilon}^{-1} - \dot{\mathbf{q}}^\dagger \dot{\mathbf{Q}}^{-1} \dot{\mathbf{q}}}$$

$$\dot{w}_M = -\dot{\epsilon} (\dot{\phi}^\dagger \Phi^{-1} \mathbf{C} \mathbf{Q}^{-1} - \dot{\mathbf{c}}^\dagger \mathbf{Q}^{-1}) \cdot \left(\mathbf{I}_{P \times P} - \frac{\dot{\mathbf{q}} \dot{\mathbf{q}}^\dagger \mathbf{Q}^{-1}}{\dot{\epsilon}^{-1} - \dot{\mathbf{q}}^\dagger \mathbf{Q}^{-1} \dot{\mathbf{q}}} \right) \mathbf{g}$$

$$\dot{\Delta} \mathbf{w} = -\frac{\dot{\mathbf{q}}^\dagger \mathbf{Q}^{-1} \mathbf{g}}{\dot{\epsilon}^{-1} - \dot{\mathbf{q}}^\dagger \mathbf{Q}^{-1} \dot{\mathbf{q}}} \Phi^{-1} \mathbf{C} \mathbf{Q}^{-1} \dot{\mathbf{q}} - \dot{w}_M \Phi^{-1} \dot{\phi}$$

$$\mathbf{w} = \bar{\mathbf{I}}_M^\dagger \dot{\mathbf{w}} - \dot{\Delta} \mathbf{w}$$

end

Algorithm 3: Summary of the CUI-LCMV procedure**input:** $\mathbf{C}, \mathbf{g}, \mathbf{w}, \Phi^{-1}, \mathbf{Q}^{-1}, \check{\mathbf{c}}, \check{g}$ **output:** $\check{\mathbf{C}}, \check{\mathbf{g}}, \check{\mathbf{w}}, \check{\mathbf{Q}}^{-1}$ **begin**

$$\check{\mathbf{C}} = [\mathbf{C}|\check{\mathbf{c}}]$$

$$\check{\mathbf{g}} = [\mathbf{g}^\dagger|\check{g}^*]^\dagger$$

$$\check{\mathbf{q}} = \mathbf{C}^\dagger \Phi^{-1} \check{\mathbf{c}}$$

$$\check{\eta} = (\check{\mathbf{c}}^\dagger \Phi^{-1} \check{\mathbf{c}} - \check{\mathbf{q}}^\dagger \mathbf{Q}^{-1} \check{\mathbf{q}})^{-1}$$

$$\check{\mathbf{Q}}^{-1} = \left[\begin{array}{c|c} \mathbf{Q}^{-1} + \check{\eta} \mathbf{Q}^{-1} \check{\mathbf{q}} \check{\mathbf{q}}^\dagger \mathbf{Q}^{-1} & -\check{\eta} \mathbf{Q}^{-1} \check{\mathbf{q}} \\ \hline -\check{\eta} \check{\mathbf{q}}^\dagger \mathbf{Q}^{-1} & \check{\eta} \end{array} \right]$$

$$\check{\Delta} \mathbf{w} = \check{\eta} (\check{g} - \check{\mathbf{c}}^\dagger \mathbf{w}) \Phi^{-1} (\mathbf{I} - \mathbf{C} \mathbf{Q}^{-1} \mathbf{C}^\dagger \Phi^{-1}) \check{\mathbf{c}}$$

$$\check{\mathbf{w}} = \mathbf{w} + \check{\Delta} \mathbf{w}$$

end**Algorithm 4: Summary of the CUD-LCMV procedure****input:** $\check{\mathbf{C}}, \check{\mathbf{g}}, \check{\mathbf{w}}, \Phi^{-1}, \check{\mathbf{Q}}^{-1}$ **output:** $\mathbf{C}, \mathbf{g}, \mathbf{w}, \mathbf{Q}^{-1}$ **begin**

$$\mathbf{C} = \check{\mathbf{c}} \bar{\mathbf{I}}_P$$

$$\mathbf{g} = \bar{\mathbf{I}}_P^\dagger \check{\mathbf{g}}$$

$$\mathbf{Q}^{-1} = \bar{\mathbf{I}}_P^\dagger \check{\mathbf{Q}}^{-1} \bar{\mathbf{I}}_P - \frac{\bar{\mathbf{I}}_P^\dagger \check{\mathbf{Q}}^{-1} \mathbf{i}_P \mathbf{i}_P^\dagger \check{\mathbf{Q}}^{-1} \bar{\mathbf{I}}_P}{\mathbf{i}_P^\dagger \check{\mathbf{Q}}^{-1} \mathbf{i}_P}$$

$$\check{\mathbf{q}} = \mathbf{C}^\dagger \Phi^{-1} \check{\mathbf{c}}$$

$$\check{\eta} = (\check{\mathbf{c}}^\dagger \Phi^{-1} \check{\mathbf{c}} - \check{\mathbf{q}}^\dagger \mathbf{Q}^{-1} \check{\mathbf{q}})^{-1}$$

$$\check{\Delta} \mathbf{w} = \check{\eta} (\check{g} - \check{\mathbf{c}}^\dagger \mathbf{w}) \Phi^{-1} (\mathbf{I} - \mathbf{C} \mathbf{Q}^{-1} \mathbf{C}^\dagger \Phi^{-1}) \check{\mathbf{c}}$$

$$\mathbf{w} = \check{\mathbf{w}} - \check{\Delta} \mathbf{w}$$

end**Algorithm 5: Summary of the SUI-GSC procedure****input:** $\mathbf{C}, \mathbf{g}, \mathbf{w}_\parallel, \mathbf{R}^{-1}, \mathbf{B}, \check{\mathbf{c}}$ **output:** $\check{\mathbf{C}}, \check{\mathbf{w}}_\parallel, \check{\mathbf{R}}^{-1}, \check{\mathbf{B}}$ **begin**

$$\check{\mathbf{C}} = [\mathbf{C}^\dagger|\check{\mathbf{c}}]^\dagger$$

$$\check{\mathbf{R}}^{-1} = \mathbf{R}^{-1} - \frac{\mathbf{R}^{-1} \check{\mathbf{c}} \check{\mathbf{c}}^\dagger \mathbf{R}^{-1}}{1 + \check{\mathbf{c}}^\dagger \mathbf{R}^{-1} \check{\mathbf{c}}}$$

$$\check{w}_{\parallel M} = \frac{\check{\mathbf{c}}^\dagger \mathbf{R}^{-1} \mathbf{g}}{1 + \check{\mathbf{c}}^\dagger \mathbf{R}^{-1} \check{\mathbf{c}}}$$

$$\check{\Delta} \mathbf{w}_\parallel = -\check{w}_{\parallel M} \mathbf{C} \mathbf{R}^{-1} \check{\mathbf{c}}$$

$$\check{\mathbf{w}}_\parallel = [(\mathbf{w}_\parallel + \check{\Delta} \mathbf{w}_\parallel)^T | \check{w}_{\parallel M}]^T$$

$$\check{\Delta} \mathbf{b} = \frac{[\check{\mathbf{c}}^\dagger \mathbf{R}^{-1} \mathbf{C}^\dagger - 1]^\dagger}{\|[\check{\mathbf{c}}^\dagger \mathbf{R}^{-1} \mathbf{C}^\dagger - 1]\|}$$

$$\check{\mathbf{B}} = \left[\begin{array}{c|c} \mathbf{B} & \\ \hline \mathbf{0}_{1 \times M-1-P} & \check{\Delta} \mathbf{b} \end{array} \right]$$

end**Algorithm 6: Summary of the SUD-GSC procedure****input:** $\check{\mathbf{C}}, \mathbf{g}, \check{\mathbf{w}}_\parallel, \check{\mathbf{R}}^{-1}, \check{\mathbf{B}}$ **output:** $\mathbf{C}, \mathbf{w}_\parallel, \mathbf{R}^{-1}, \mathbf{B}$ **begin**

$$\mathbf{C} = \bar{\mathbf{I}}_M^\dagger \check{\mathbf{C}}$$

$$\mathbf{R}^{-1} = \check{\mathbf{R}}^{-1} + \frac{\check{\mathbf{R}}^{-1} \check{\mathbf{c}} \check{\mathbf{c}}^\dagger \check{\mathbf{R}}^{-1}}{1 - \check{\mathbf{c}}^\dagger \check{\mathbf{R}}^{-1} \check{\mathbf{c}}}$$

$$\check{w}_{\parallel M} = \bar{\mathbf{I}}_M^\dagger \check{\mathbf{w}}_\parallel$$

$$\check{\Delta} \mathbf{w}_\parallel = -\check{w}_{\parallel M} \mathbf{C} \mathbf{R}^{-1} \check{\mathbf{c}}$$

$$\mathbf{w}_\parallel = \bar{\mathbf{I}}_M^\dagger \check{\mathbf{w}}_\parallel - \check{\Delta} \mathbf{w}_\parallel$$

$$\check{\chi} = \check{\mathbf{B}}^\dagger \mathbf{i}_M + \|\check{\mathbf{B}}^\dagger \mathbf{i}_M\| \exp(-j \angle \check{B}_{M, M-P}) \mathbf{i}_{M-P}$$

$$\mathbf{B} = \bar{\mathbf{I}}_M^\dagger \check{\mathbf{B}} \left(\mathbf{I} - \frac{2\check{\chi} \check{\chi}^\dagger}{\|\check{\chi}\|^2} \right) \bar{\mathbf{I}}_{M-P}$$

end**Algorithm 7: Summary of the CUI-GSC procedure****input:** $\mathbf{C}, \mathbf{g}, \mathbf{w}_\parallel, \mathbf{R}^{-1}, \mathbf{B}, \check{\mathbf{c}}, \check{g}$ **output:** $\check{\mathbf{C}}, \check{\mathbf{g}}, \check{\mathbf{w}}_\parallel, \check{\mathbf{R}}^{-1}, \check{\mathbf{B}}$ **begin**

$$\check{\mathbf{C}} = [\mathbf{C}|\check{\mathbf{c}}]$$

$$\check{\mathbf{g}} = [\mathbf{g}^\dagger|\check{g}^*]^\dagger$$

$$\check{\mathbf{r}} = \mathbf{C}^\dagger \check{\mathbf{c}}$$

$$\check{\rho} = (\|\check{\mathbf{c}}\|^2 - \check{\mathbf{r}}^\dagger \mathbf{R}^{-1} \check{\mathbf{r}})^{-1}$$

$$\check{\mathbf{R}}^{-1} = \left[\begin{array}{c|c} \mathbf{R}^{-1} + \check{\rho} \mathbf{R}^{-1} \check{\mathbf{r}} \check{\mathbf{r}}^\dagger \mathbf{R}^{-1} & -\check{\rho} \mathbf{R}^{-1} \check{\mathbf{r}} \\ \hline -\check{\rho} \check{\mathbf{r}}^\dagger \mathbf{R}^{-1} & \check{\rho} \end{array} \right]$$

$$\check{\Delta} \mathbf{w}_\parallel = \check{\rho} (\check{g} - \check{\mathbf{c}}^\dagger \mathbf{w}_\parallel) (\mathbf{I} - \mathbf{C} \mathbf{R}^{-1} \mathbf{C}^\dagger) \check{\mathbf{c}}$$

$$\check{\mathbf{w}}_\parallel = \mathbf{w}_\parallel + \check{\Delta} \mathbf{w}_\parallel$$

$$\check{\mathbf{b}} = \mathbf{B}^\dagger \check{\mathbf{c}}$$

$$\check{\chi} = \frac{\check{\mathbf{b}}}{\|\check{\mathbf{b}}\|} + \exp(j \angle \check{b}_{M-P+1}) \mathbf{i}_{M-P+1}$$

$$\check{\mathbf{B}} = \mathbf{B} \left(\mathbf{I}_{(M-P+1) \times (M-P+1)} - \frac{2\check{\chi} \check{\chi}^\dagger}{\|\check{\chi}\|^2} \right) \bar{\mathbf{I}}_{M-P+1}$$

end**Algorithm 8: Summary of the CUD-GSC procedure****input:** $\check{\mathbf{C}}, \check{\mathbf{g}}, \check{\mathbf{w}}_\parallel, \check{\mathbf{R}}^{-1}, \check{\mathbf{B}}$ **output:** $\mathbf{C}, \mathbf{g}, \mathbf{w}_\parallel, \mathbf{R}^{-1}, \mathbf{B}$ **begin**

$$\mathbf{C} = \check{\mathbf{C}} \bar{\mathbf{I}}_P$$

$$\mathbf{g} = \bar{\mathbf{I}}_P^\dagger \check{\mathbf{g}}$$

$$\mathbf{R}^{-1} = \bar{\mathbf{I}}_P^\dagger \check{\mathbf{R}}^{-1} \bar{\mathbf{I}}_P - \frac{\bar{\mathbf{I}}_P^\dagger \check{\mathbf{R}}^{-1} \mathbf{i}_P \mathbf{i}_P^\dagger \check{\mathbf{R}}^{-1} \bar{\mathbf{I}}_P}{\mathbf{i}_P^\dagger \check{\mathbf{R}}^{-1} \mathbf{i}_P}$$

$$\check{\Delta} \mathbf{w}_\parallel = \|\check{\mathbf{C}} \bar{\mathbf{R}}^{-1} \mathbf{I}_P\|^{-2} \check{\mathbf{C}} \bar{\mathbf{R}}^{-1} \mathbf{i}_P (\check{\mathbf{C}} \bar{\mathbf{R}}^{-1} \mathbf{i}_P)^\dagger \check{\mathbf{w}}_\parallel$$

$$\mathbf{w}_\parallel = \check{\mathbf{w}}_\parallel - \check{\Delta} \mathbf{w}_\parallel$$

$$\mathbf{B} = \left[\check{\mathbf{B}} \left| \frac{\check{\mathbf{C}} \bar{\mathbf{R}}^{-1} \mathbf{i}_P}{\|\check{\mathbf{C}} \bar{\mathbf{R}}^{-1} \mathbf{i}_P\|} \right. \right]$$

end

REFERENCES

- [1] M. Er and A. Cantoni, "Derivative constraints for broad-band element space antenna array processors," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 31, no. 6, pp. 1378–1393, Dec. 1983.
- [2] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proc. IEEE*, vol. 57, no. 8, pp. 1408–1418, Aug. 1969.
- [3] O. L. Frost, III, "An algorithm for linearly constrained adaptive array processing," *Proc. IEEE*, vol. 60, no. 8, pp. 926–935, Aug. 1972.
- [4] B. R. Breed and J. Strauss, "A short proof of the equivalence of LCMV and GSC beamforming," *IEEE Signal Process. Lett.*, vol. 9, no. 6, pp. 168–169, Jun. 2002.
- [5] S. Affes and Y. Grenier, "A signal subspace tracking algorithm for microphone array processing of speech," *IEEE Trans. Speech Audio Process.*, vol. 5, no. 5, pp. 425–437, Sep. 1997.
- [6] S. Gannot, D. Burshtein, and E. Weinstein, "Signal enhancement using beamforming and nonstationarity with applications to speech," *Signal Process.*, vol. 49, no. 8, pp. 1614–1626, Aug. 2001.
- [7] S. Markovich, S. Gannot, and I. Cohen, "Multichannel eigenspace beamforming in a reverberant noisy environment with multiple interfering speech signals," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 17, no. 6, pp. 1071–1086, Aug. 2009.
- [8] Y. Zheng, R. Goubran, and M. El-Tanany, "Robust near-field adaptive beamforming with distance discrimination," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 5, pp. 478–488, Sep. 2004.
- [9] H. Cox, R. M. Zeskind, and M. M. Owen, "Robust adaptive beamforming," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 35, no. 10, Oct. 1987.
- [10] S. Doclo, M. Moonen, T. V. den Bogaert, and J. Wouters, "Reduced-bandwidth and distributed MWF-based noise reduction algorithms for binaural hearing aids," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 17, no. 1, pp. 38–51, Jan. 2009.
- [11] A. Bertrand and M. Moonen, "Distributed adaptive node-specific signal estimation in fully connected sensor networks—Part I: Sequential node updating," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5277–5291, Oct. 2010.
- [12] A. Bertrand and M. Moonen, "Distributed adaptive node-specific signal estimation in fully connected sensor networks—Part II: Simultaneous and asynchronous node updating," *IEEE Trans. Signal Process.*, vol. 58, no. 10, pp. 5292–5306, Oct. 2010.
- [13] S. Markovich Golan, S. Gannot, and I. Cohen, "A reduced bandwidth binaural MVDR beamformer," in *Proc. Int. Workshop on Acoust. Echo and Noise Contr. (IWAENC)*, Tel Aviv, Israel, Aug. 2010.
- [14] A. Bertrand and M. Moonen, "Distributed LCMV beamforming in wireless sensor networks with node-specific desired signals," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Prague, Czech Republic, May 2011.
- [15] T. C. Lawin-Ore and S. Doclo, "Analysis of rate constraints for MWF-based noise reduction in acoustic sensor networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, May 2011.
- [16] A. Bertrand and M. Moonen, "Efficient sensor subset selection and link failure response for linear MMSE signal estimation in wireless sensor networks," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Aalborg, Denmark, Aug. 2010, pp. 1092–1096.
- [17] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.
- [18] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 5, no. 2, pp. 4–24, Apr. 1988.
- [19] M. A. Woodbury, "Inventing modified matrices," *Statist. Res. Group, Memo*, no. 42, 1950, Princeton Univ., Princeton, NJ.
- [20] A. S. Householder, "Unitary triangularization of a nonsymmetric matrix," *J. ACM*, vol. 5, pp. 339–342, Oct. 1958.
- [21] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: The Johns Hopkins Univ. Press, 1996.



Shmulik Markovich-Golan (S'09) received the B.Sc. (*cum laude*) and M.Sc. degrees in electrical engineering from the Technion—Israel Institute of Technology, Haifa, in 2002 and 2008, respectively.

He is currently pursuing the Ph.D. degree at the Engineering Faculty, Bar-Ilan University, Israel. His research interests include multichannel signal processing, distributed sensor networks, speech enhancement using microphone arrays, and distributed estimation.



Sharon Gannot (S'92–M'01–SM'06) received the B.Sc. degree (*summa cum laude*) from the Technion—Israel Institute of Technology, Haifa, in 1986 and the the M.Sc. (*cum laude*) and Ph.D. degrees from Tel-Aviv University, Israel, in 1995 and 2000, respectively, all in electrical engineering.

In 2001, he held a Postdoctoral position with the Department of Electrical Engineering (ESAT-SISTA), K.U. Leuven, Belgium. From 2002 to 2003, he held a research and teaching position with the Faculty of Electrical Engineering, Technion-Israel Institute of Technology. Currently, he is an Associate Professor with the School of Engineering, Bar-Ilan University, Israel, where he is heads the Speech and Signal Processing Laboratory. His research interests include parameter estimation, statistical signal processing, especially speech processing using either single- or multimicrophone arrays.

Prof. Gannot is the recipient of the Bar-Ilan University Outstanding Lecturer award for 2010. He is currently an Associate Editor of the IEEE TRANSACTIONS ON SPEECH, AUDIO AND LANGUAGE PROCESSING. He served as an Associate Editor of the *EURASIP Journal of Advances in Signal Processing* between 2003–2001, and as an Editor of two special issues on "Multi-microphone Speech Processing" of the same *Journal*. He also served as a Guest Editor of the *EL-SEVIER Speech Communication* journal and a reviewer of many IEEE journals and conferences. He is a member of the Audio and Acoustic Signal Processing (AASSP) Technical Committee of the IEEE since January 2010. He is also a member of the Technical and Steering Committee of the International Workshop on Acoustic Echo and Noise Control (IWAENC) since 2005 and the General Co-Chair of IWAENC held in Tel-Aviv during August 2010. He will serve as the General Co-Chair of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA) in 2013.



Israel Cohen (M'01–SM'03) received the B.Sc. (*summa cum laude*), M.Sc., and Ph.D. degrees in electrical engineering from the Technion—Israel Institute of Technology, Haifa, in 1990, 1993, and 1998, respectively.

From 1990 to 1998, he was a Research Scientist with RAFAEL Research Laboratories, Haifa, Israel Ministry of Defense. From 1998 to 2001, he was a Postdoctoral Research Associate with the Computer Science Department, Yale University, New Haven, CT. In 2001, he joined the Electrical Engineering Department, Technion, where he is currently an Associate Professor. His research interests are statistical signal processing, analysis and modeling of acoustic signals, speech enhancement, noise estimation, microphone arrays, source localization, blind source separation, system identification, and adaptive filtering.

Prof. Cohen received in 2005 and 2006 the Technion Excellent Lecturer awards. He served as Associate Editor of the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING and IEEE SIGNAL PROCESSING LETTERS, and as Guest Editor of a special issue of the *EURASIP Journal on Advances in Signal Processing* on Advances in Multimicrophone Speech Processing and a special issue of the *EURASIP Speech Communication Journal* on Speech Enhancement. He is a coeditor of the Multichannel Speech Processing section of the *Springer Handbook of Speech Processing* (New York: Springer, 2007).