Efficient Calculation of Sensor Utility and Sensor Removal in Wireless Sensor Networks for Adaptive Signal Estimation and Beamforming

Alexander Bertrand, *Member, IEEE*, Joseph Szurley, *Member, IEEE*, Peter Ruckebusch, Ingrid Moerman, and Marc Moonen, *Fellow, IEEE*

***Abstract*—Wireless sensor networks are often deployed over a large area of interest and therefore the quality of the sensor signals may vary significantly across the different sensors. In this case, it is useful to have a measure for the importance or the so-called "utility" of each sensor, e.g., for sensor subset selection, resource allocation or topology selection. In this paper, we consider the efficient calculation of sensor utility measures for four different signal estimation or beamforming algorithms in an adaptive context. We use the definition of sensor utility as the increase in cost (e.g., mean-squared error) when the sensor is removed from the estimation procedure. Since each possible sensor removal corresponds to a new estimation problem (involving less sensors), calculating the sensor utilities would require a continuous updating of K different signal estimators (where K is the number of sensors), increasing computational complexity and memory usage by a factor K. However, we derive formulas to efficiently calculate all sensor utilities with hardly any increase in memory usage and computational complexity compared to the signal estimation algorithm already in place. When applied in adaptive signal estimation algorithms, this allows for on-line tracking of all the sensor utilities at almost no additional cost. Furthermore, we derive efficient formulas for sensor removal, i.e., for updating the signal estimator coefficients when a sensor is removed, e.g., due to a failure in the wireless link or when its utility is too low. We provide a complexity evaluation of the derived formulas, and demonstrate the significant reduction in computational complexity compared to straightforward implementations.**

***Index Terms*—LCMV beamforming, MMSE estimation, multi-channel Wiener filtering, sensor subset selection, sensor utility, signal enhancement, signal estimation, wireless sensor networks (WSNs).**

Manuscript received February 16, 2012; revised May 25, 2012 and July 10, 2012; accepted July 10, 2012. Date of publication July 31, 2012; date of current version October 09, 2012. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Joseph Tabrikian. The work of A. Bertrand was supported by a Postdoctoral Fellowship of the Research Foundation—Flanders (FWO). This work was carried out at the ESAT Laboratory of Katholieke Universiteit Leuven, in the frame of K. U. Leuven Research Council CoE EF/05/006 "Optimization in Engineering" (OPTEC) and PFV/10/002 (OPTEC), Concerted Research Action GOA-MaNet, the Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian Federal Science Policy Office IUAP P6/04 (DYSCO, "Dynamical systems, control and optimization," 2007–2011), Research Project IBBT, and Research Project FWO nr. G.0763.12 "Wireless acoustic sensor networks for extended auditory communication." The scientific responsibility is assumed by its authors. The results in Section IV have been published in part in the *Proceedings of the European Signal Processing Conference* (EUSIPCO), Aalborg, Denmark, August 2010, pp. 1092–1096. (see [1]).

A. Bertrand, J. Szurley, and M. Moonen are with the Department of Electrical Engineering ESAT, SCD-SISTA, K.U Leuven, and also with the IBBT-Future Health Department, B-3001 Leuven, Belgium (e-mail: alexander.bertrand@esat.kuleuven.be; joseph.szurley@esat.kuleuven.be; marc.moonen@esat.kuleuven.be).

P. Ruckebusch and I. Moerman are with the Ghent University, INTEC-IBCN, 9050 Ghent, Belgium (e-mail: peter.ruckebusch@intec.ugent.be; ingrid.moerman@intec.ugent.be).

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.2012.2210888

I. Introduction

A wireless sensor network (WSN) [2]–[6] is a low-cost communication network that allows spatially distributed sensing and wireless connectivity in applications with limited power. A WSN consists of randomly deployed embedded wireless devices, called sensor nodes, that are equipped with one or more sensors, a lightweight processor, and a low power radio transceiver. The sensor nodes exchange signals with neighboring nodes or with an external processing unit (a so-called fusion center) in order to perform a network-wide task such as signal estimation, detection, localization, etc. WSNs enable numerous new applications but the limited nature of the devices has to be taken into account in the algorithm design. As the sensor nodes in a WSN usually have limited power sources (e.g., batteries or energy harvesting), energy efficiency is of great importance, both in terms of processing power and network lifetime. Furthermore, the communication resources are also often limited, and therefore protocols and algorithms for WSNs pursue a minimal data exchange.

A. Contribution

In this paper, we consider the case where a WSN is used for adaptive signal estimation or beamforming, where the goal is to recover an unknown desired signal from the noisy sensor signals collected by the WSN. By using a WSN, a large area can be covered to obtain more spatial information, which may result in an improved estimation performance compared to a system with a small local sensor array. However, coverage of a large area also implies that the quality of the sensor signals may vary significantly over the different sensors in the WSN. Therefore, some sensor nodes may be more useful than others. Due to the limited energy and communication resources, it is important to identify the 'usefulness' or utility of each sensor.

A natural definition of sensor utility is the increase in cost (e.g., mean-squared error) when the sensor is removed (backward mode), or the decrease in cost if a sensor is added (forward mode) [1], [7], [8]. This definition of utility is adopted in this paper for four different signal estimation or beamforming algorithms:

1053-587X/$31.00 © 2012 IEEE

1) Linearly constrained minimum variance (LCMV) beamforming [9]
2) Linear minimum mean-squared error (LMMSE) signal estimation, a.k.a. multi-channel Wiener filtering (MWF) [10]
3) Distortion-weighted LMSSE[1] (DW-LMMSE) [11]–[13]
4) Rank-1 DW-LMMSE, a.k.a. Rank-1 MWF (R1-MWF) [8], [12].

It is noted that the well-known minimum variance distortionless response (MVDR) beamformer [9] is a special case of the LCMV beamformer and the R1-MWF, and is therefore also implicitly incorporated in this list. We only treat sensor removal (backward mode). Sensor addition is tougher, and beyond the scope of this paper. It is noted that, for LMMSE signal estimation, sensor addition has been addressed in [1].

We consider *adaptive* signal estimation and beamforming algorithms, where we assume that the sensor signal statistics may change over time due to movement of sources or sensors. This also means that the sensor utilities may change over time, which requires an on-line calculation of the sensor utilities. Assuming there are K sensors, this would require a continuous updating of K different signal estimators (one for each sensor removal), since the optimal estimators for each set of $K-1$ sensors are different from the optimal estimator with K sensors. This increases the computational complexity and memory requirements by a factor K, e.g., assuming the optimal signal estimator can be computed with $O(K^3)$ complexity, then calculating all sensor utilities has $O(K^4)$ complexity. Our main contribution of this paper is to provide efficient formulas to calculate the sensor utilities for the above four signal estimation algorithms. We show that, with the signal estimation already in place, it is possible to track all the sensor utilities simultaneously with $O(K)$ complexity, i.e., with hardly any additional computational effort.

Finally, we also provide computationally efficient formulas to remove a sensor from the current estimator[2]. This is interesting for backwards greedy sensor subset selection algorithms (see Section I-B). Furthermore, WSNs often suffer from link failures, e.g., due to power shortage or interference. The signal estimation algorithm must then be able to swiftly adapt to these link failures to maintain sufficient estimation quality. Due to the low complexity of the sensor removal procedure described here, sensor nodes are able to react very quickly to link failures, which is important in real-time applications with high data rates such as in, e.g., WSN's for real-time audio acquisition [15]–[20].

B. The Use of Utility Measures in Practice

The earlier mentioned utility measure can be used (and has been used) in several contexts, e.g., for sensor subset selection, resource allocation, topology selection, compression or source coding, etc. In this subsection, we briefly address some of these techniques.

So-called sensor subset selection (SSS) algorithms aim at selecting those sensor nodes that yield a significant contribution to the signal estimation process, while putting the other sensor nodes to sleep [1], [7], [8], [21]–[24]. This allows a more efficient allocation of communication resources, and it reduces interference. Furthermore, since sleeping nodes consume less energy, SSS usually also prolongs the lifetime of the network, assuming that the SSS algorithm itself has low computational complexity. Selecting the optimal set of sensors for signal estimation or beamforming is a non-trivial task since it is a combinatorial optimization problem which is typically NP-hard [7], [25]. It is more difficult than, e.g., the notorious knapsack problem due to the fact that the utility of each sensor depends on the other sensors that are in the selected subset (e.g., sensor A may be useful if combined with sensor B, but useless if sensor B is not in the selected set of sensors). The SSS problem has been addressed in many other works, and many selection techniques and heuristics have been described [1], [7], [21]–[23], [26], [27] (non-exhaustive). However, for real-time processing with limited computational power, a greedy subset selection is usually a good choice because of its simplicity and sufficient accuracy [1], [7], [22], [23], [26]. The greedy approach assumes that each sensor has a utility measure that quantifies its contribution to the overall estimation, as envisaged in this paper.

Sensor utility measures are also important in bandwidth constrained WSNs where each node can only transmit a subset of its available sensor signals (now assuming a node may have multiple sensors). This is for instance the case in wireless binaural hearing aids (HAs) with multiple microphones, where each hearing aid can only transmit a single microphone signal to the contra-lateral HA [17], [28]–[30]. Sensor utility measures can also be used as a source coding tool, to select signal components that contribute most to the estimation, and leave out those signal components that have no significant contribution, to reduce the communication bandwidth over the wireless links [13]. Sensor utilities can also provide useful cross-layer information for resource allocation in WSNs, e.g., to determine the transmission bandwidth that should be allocated to each sensor (assigning more bandwidth for sensors with a high utility). They can also be used in topology selection or network clustering. For example, sensors with a high utility can be placed in the center of the network such that their observations spread fast, or links can be pruned between sensors that do not contribute much to each other's estimation problem. The latter eliminates redundant transmissions, and results in (possibly unconnected) clusters containing nodes that significantly contribute from each other.

C. Outline

The paper is organized as follows. In Section II, we provide the general problem statement. The sensor removal procedure and sensor utility calculations for LCMV, LMMSE, DW-LMMSE and R1-MWF are then derived in Sections III–VI, respectively. In Section VII, we provide a comparison of the complexity and sensitivity of the derived formulas. Conclusions are drawn in Section VIII.

[1]This signal estimation algorithm allows to control the trade-off between distortion and noise reduction. This is, i.a., used in speech enhancement literature where it is often referred to as speech distortion weighted MWF (SDW-MWF) [11].

[2]It is noted that sensor removal in the case of LCMV beamforming has been addressed earlier in [14]. However, we provide a more efficient scheme based on variables that are readily available in an adaptive LCMV implementation.

II. Problem Statement and Notation

We consider a WSN with K sensors and, without loss of generality (w.l.o.g.), we assume that all sensor signals are centralized in a fusion center for processing. However, the results in this paper can be equally applied to the distributed case where each sensor node solves a local signal estimation problem, as in [16]–[20], [31]–[33]. Sensor $k \in \{1, \ldots, K\}$ collects samples $y_k[t]$ of a complex valued sensor signal y_k, where $t \in \mathbb{N}$ is the discrete time index. For the sake of an easy exposition, we will mostly omit the discrete time index t, unless we specifically refer to a particular sample. Note that, throughout this paper, all signals are assumed to be complex valued to permit frequency-domain descriptions, e.g., when using a short-time Fourier transform (STFT). We assume that all sensor signals are realizations of stationary and ergodic stochastic processes. However, in practice, the stationarity and ergodicity assumption can be relaxed to short-term stationarity and ergodicity, in which case the theory should be applied to finite signal segments. We define $\mathbf{y}$ as the K-channel signal gathered at the fusion center in which all signals $y_k, \forall\, k \in \{1, \ldots, K\}$, are stacked.

We will consider four different signal estimation or beamforming algorithms (LCMV, LMMSE, DW-LMMSE and R1-MWF), where the goal is to estimate a complex valued desired signal d from the sensor signals $\mathbf{y}$ (we do not impose any data model[3] on $\mathbf{y}$). We consider linear signal estimation, where a linear estimator $\mathbf{w}$ produces an estimate $\hat{d} = \mathbf{w}^H\mathbf{y}$, where the superscript H denotes the conjugate transpose operator. The four signal estimation algorithms considered here, are based on different optimization problems with different cost functions $J(\mathbf{w})$, each one defining a different optimal estimator $\mathbf{w}^{\mathrm{opt}}$.

The goal of this paper is to determine the contribution of each sensor to the estimation process, referred to as sensor utility. A natural way of defining the utility of a particular sensor is to evaluate the increase of the cost when this sensor is removed, after recomputing the new optimal estimator [1], [7]. For sensor k, the utility is then computed as

$$U_k \triangleq J_{-k}(\mathbf{w}_{-k}^{\mathrm{opt}}) - J(\mathbf{w}^{\mathrm{opt}}) \tag{1}$$

where J_{-k} denotes the cost function where sensor k is removed from the estimation process, and $\mathbf{w}_{-k}^{\mathrm{opt}}$ is the corresponding optimal estimator. Note that $\mathbf{w}_{-k}^{\mathrm{opt}}$ is not equal to $\mathbf{w}^{\mathrm{opt}}$ with the k-th entry removed, i.e., a completely new estimator has to be computed, based on the cost function J_{-k}.

In all signal estimation or beamforming algorithms considered here, an inversion of a sample correlation matrix is required. For example, the computation of an LCMV beamformer requires the inverse of a sample correlation matrix corresponding to the sensor signal correlation matrix $\mathbf{R}_{yy} \triangleq E\{\mathbf{y}\mathbf{y}^H\}$, where $E\{.\}$ denotes the expected value operator. When K is large, computing this matrix inverse[4] is computationally expensive, i.e., $O(K^3)$, and should be avoided in applications with high data rates. Therefore, we assume in the sequel that the inverse of the sample correlation matrix is computed recursively. Let $\mathbf{R}_{yy}[t]$ denote the estimate[5] of $\mathbf{R}_{yy}$ at time t. Assuming stationarity and ergodicity, this matrix estimate can be computed based on temporal averaging. A common technique is to use an exponential window, in which case the matrix $\mathbf{R}_{yy}[t]$ is recursively updated based on a convex combination with a forgetting factor $0 < \lambda < 1$, i.e.,

$$\mathbf{R}_{yy}[t] = \lambda\mathbf{R}_{yy}[t-1] + (1-\lambda)\mathbf{y}[t]\mathbf{y}[t]^H. \tag{2}$$

In this case, $\mathbf{R}_{yy}^{-1}[t]$ can be recursively updated by means of the matrix inversion lemma, a.k.a. the Woodbury identity [34], i.e.,

$$\mathbf{R}_{yy}^{-1}[t] = \frac{1}{\lambda}\mathbf{R}_{yy}^{-1}[t-1] - \frac{\mathbf{R}_{yy}^{-1}[t-1]\mathbf{y}[t]\mathbf{y}[t]^H\mathbf{R}_{yy}^{-1}[t-1]}{\frac{\lambda^2}{1-\lambda} + \lambda\mathbf{y}[t]^H\mathbf{R}_{yy}^{-1}[t-1]\mathbf{y}[t]} \tag{3}$$

which has a computational complexity of $O(K^2)$. It is noted that, when (3) is used to update $\mathbf{R}_{yy}^{-1}[t]$, the correlation matrix $\mathbf{R}_{yy}[t]$ itself does not need to be kept in memory.

III. LCMV Beamforming

Linearly constrained minimum variance (LCMV) beamforming is a well-known sensor array processing technique for signal estimation [9]. Its goal is to minimize the output power of a multi-channel filter $\mathbf{w}$, under a set of linear constraints, e.g., to preserve a target source signal d and (fully or partially) cancel interferers. More specifically, the LCMV beamforming problem is formulated as

$$\min_{\mathbf{w}} \quad J(\mathbf{w}) \triangleq E\{|\mathbf{w}^H\mathbf{y}|^2\} \tag{4}$$

$$\text{s.t.} \quad \mathbf{C}^H\mathbf{w} = \mathbf{f} \tag{5}$$

where $\mathbf{C}$ is a $K \times Q$ constraint matrix (with $Q \leq K$) and $\mathbf{f}$ is a $Q \times 1$ vector. The columns of $\mathbf{C}$ usually consist of steering vectors corresponding to Q source signals that are observed at the sensors. The entries of $\mathbf{f}$ usually consist of ones and zeros to preserve an undistorted estimate of the target source signal and fully cancel out the interferers. Assuming that the correlation matrix $\mathbf{R}_{yy}$ has full rank, the solution to this problem is given by [9]:

$$\boxed{\mathbf{w}^{\mathrm{LCMV}} = \mathbf{R}_{yy}^{-1}\mathbf{C}\left(\mathbf{C}^H\mathbf{R}_{yy}^{-1}\mathbf{C}\right)^{-1}\mathbf{f}.} \tag{6}$$

For the sake of an easy exposition, and with a slight abuse of notation, we will mostly omit the superscript 'LCMV' and simply use $\mathbf{w}$ (without superscript) to refer to the LCMV solution (the same holds in Sections V–VIII for the other beamformers or signal estimators).

In Appendix A, a recursive algorithm is described to adaptively update the LCMV solution (6) for each new vector of incoming sensor signal samples $\mathbf{y}[t]$, based on the update (2). The

[3]Only in Sections V and VI, we use a particular (but still quite general) data model to describe $\mathbf{y}$.

[4]In this paper, we assume that the correlation matrix $\mathbf{R}_{yy}$ has full rank, such that its inverse always exists. This is often a reasonable assumption in practice, since sensors usually introduce uncorrelated sensor noise. If this assumption does not hold, the matrix inverse has to be replaced by a (Moore-Penrose) pseudoinverse.

[5]For the sake of an easy exposition, we generally do not introduce separate notations to distinguish between a quantity and its estimate. We assume that it is clear from the context whether the estimate or the true quantity is referred to, even when the discrete time index t is omitted.

overall computational complexity of this algorithm is $O(K^2)$, and it is based on a recursive update of the following variables:

$$\boxed{\begin{array}{ll} \bullet\ \mathbf{R}_{yy}^{-1} & (K\times K)\\ \bullet\ \mathbf{\Gamma}\triangleq\mathbf{R}_{yy}^{-1}\,\mathbf{C} & (K\times Q)\\ \bullet\ \mathbf{L}\triangleq\left(\mathbf{C}^H\,\mathbf{R}_{yy}^{-1}\,\mathbf{C}\right)^{-1} & (Q\times Q)\\ \bullet\ \mathbf{w} & (K\times 1)\end{array}} \tag{7}$$

In the sequel, we assume that the above variables are readily available during operation of the algorithm.

A. Sensor Removal

Assume that the fusion center now only has access to the $(K-1)$-channel signal $\mathbf{y}_{-k}$, which is defined as $\mathbf{y}$ with y_k removed. In this case, the LCMV solution is

$$\mathbf{w}_{-k}=\mathbf{R}_{yy-k}^{-1}\overline{\mathbf{C}}_{-k}\left(\overline{\mathbf{C}}_{-k}^H\mathbf{R}_{yy-k}^{-1}\overline{\mathbf{C}}_{-k}^H\right)^{-1}\mathbf{f} \tag{8}$$

where $\mathbf{R}_{yy-k}\triangleq E\{\mathbf{y}_{-k}\mathbf{y}_{-k}^H\}$ and $\overline{\mathbf{C}}_{-k}$ is the matrix $\mathbf{C}$ with row k removed. However, computing (8) requires $\mathbf{R}_{yy-k}^{-1}$, which is not directly available. If $\mathbf{R}_{yy}$ were available in memory, it is possible to invert its submatrix $\mathbf{R}_{yy-k}$ to obtain $\mathbf{R}_{yy-k}^{-1}$. However, this incurs an $O\left(K^3\right)$ complexity which, when K is large, dominates the overall complexity of the adaptive beamforming algorithm.

In the sequel, we derive an efficient formula to compute the LCMV solution $\mathbf{w}_{-k}$ without knowledge of $\mathbf{R}_{yy}$, and without explicitly computing matrix inversions. It is noted that the sensor removal for LCMV beamforming has been derived in [14] with an $O(K^2+Q^2+KQ)$ complexity[6]. However, assuming that the variables in (7) are known, we will show that it is possible to compute this with an $O(KQ)$ complexity instead.

As explained earlier, we assume that $\mathbf{R}_{yy}^{-1}$ is available. For the sake of an easy exposition, but w.l.o.g., we assume that $k=K$, i.e., the last element of $\mathbf{y}$ is removed. We consider a block partitioning of the inverse correlation matrix

$$\mathbf{R}_{yy}^{-1}=\left[\begin{array}{c|c}\mathbf{A}_k & \mathbf{a}_k\\ \hline \mathbf{a}_k^H & \alpha_k\end{array}\right] \tag{9}$$

where $\mathbf{A}_k$ is a $(K-1)\times(K-1)$ matrix, $\mathbf{a}_k$ is a $(K-1)$-dimensional vector, and α_k is a real-valued scalar. Similarly, we define

$$\mathbf{w}=\left[\begin{array}{c}\overline{\mathbf{w}}_{-k}\\ \hline W_k\end{array}\right] \tag{10}$$

$$\mathbf{C}=\left[\begin{array}{c}\overline{\mathbf{C}}_{-k}\\ \hline \mathbf{c}_k^H\end{array}\right] \tag{11}$$

$$\mathbf{\Gamma}=\left[\begin{array}{c}\overline{\mathbf{\Gamma}}_{-k}\\ \hline \mathbf{g}_k^H\end{array}\right]. \tag{12}$$

Remark: Notice that we introduced the overline notation $\overline{\mathbf{X}}_{-k}$ to denote the matrix $\mathbf{X}$ with row k removed. In the sequel, if we use the notation $\mathbf{X}_{-k}$ (without overline), then we refer to the variable $\mathbf{X}$ as it would be computed in the estimation procedure if sensor k would not be involved. E.g., notice that $\mathbf{w}_{-k}$ refers to the optimal LCMV beamformer if sensor k would not exist, whereas $\overline{\mathbf{w}}_{-k}$ refers to the vector obtained if we remove the k-th entry from $\mathbf{w}$.

Similar to (9), we define the following block partitioning of the (non-inverted) correlation matrix

$$\mathbf{R}_{yy}=\left[\begin{array}{c|c}\mathbf{R}_{yy-k} & \mathbf{r}_k\\ \hline \mathbf{r}_k^H & \rho_k\end{array}\right] \tag{13}$$

where $\mathbf{r}_k$ is a $(K-1)$-dimensional vector, and where ρ_k is a real-valued scalar. By using the matrix inversion lemma, the inverse of this block matrix is obtained as

$$\mathbf{R}_{yy}^{-1}=\left[\begin{array}{c|c}\mathbf{R}_{yy-k}^{-1}+\alpha_k\mathbf{v}_k\mathbf{v}_k^H & -\alpha_k\mathbf{v}_k\\ \hline -\alpha_k\mathbf{v}_k^H & \alpha_k\end{array}\right] \tag{14}$$

with

$$\mathbf{v}_k\triangleq\mathbf{R}_{yy-k}^{-1}\mathbf{r}_k \tag{15}$$

$$\alpha_k\triangleq\frac{1}{\rho_k-\mathbf{r}_k^H\mathbf{v}_k}. \tag{16}$$

By comparing (9) and (14), we find that

$$\mathbf{v}_k=\frac{-1}{\alpha_k}\mathbf{a}_k \tag{17}$$

and

$$\boxed{\mathbf{R}_{yy-k}^{-1}=\mathbf{A}_k-\tfrac{1}{\alpha_k}\mathbf{a}_k\mathbf{a}_k^H.} \tag{18}$$

By making the block matrix product between (14) and (11), we find that

$$\begin{aligned}\mathbf{\Gamma}=\mathbf{R}_{yy}^{-1}\mathbf{C}&=\left[\begin{array}{c}\overline{\mathbf{\Gamma}}_{-k}\\ \hline \mathbf{g}_k^H\end{array}\right]\\ &=\left[\begin{array}{c}\mathbf{R}_{yy-k}^{-1}\overline{\mathbf{C}}_{-k}+\alpha_k\mathbf{v}_k\mathbf{v}_k^H\overline{\mathbf{C}}_{-k}-\alpha_k\mathbf{v}_k\mathbf{c}_k^H\\ \hline -\alpha_k\mathbf{v}_k^H\overline{\mathbf{C}}_{-k}+\alpha_k\mathbf{c}_k^H\end{array}\right].\end{aligned} \tag{19}$$

Let $\mathbf{\Gamma}_{-k}\triangleq\mathbf{R}_{yy-k}^{-1}\overline{\mathbf{C}}_{-k}$, then a comparison between the upper and lower part in (19) yields

$$\boxed{\mathbf{\Gamma}_{-k}=\overline{\mathbf{\Gamma}}_{-k}+\mathbf{v}_k\mathbf{g}_k^H.} \tag{20}$$

We define

$$\mathbf{L}_{-k}\triangleq\left(\overline{\mathbf{C}}_{-k}^H\mathbf{R}_{yy-k}^{-1}\overline{\mathbf{C}}_{-k}\right)^{-1}. \tag{21}$$

A formula to compute $\mathbf{L}_{-k}$ from $\mathbf{L}$ is derived in [14], but we repeat it here for completeness (and use a slightly different derivation). Based on the block partitioning (14) and (11), we find that

$$\begin{aligned}\mathbf{C}^H\mathbf{R}_{yy}^{-1}\mathbf{C}=\overline{\mathbf{C}}_{-k}^H\mathbf{R}_{yy-k}^{-1}\overline{\mathbf{C}}_{-k}\\ +\alpha_k\left(\overline{\mathbf{C}}_{-k}^H\mathbf{v}_k-\mathbf{c}_k\right)\left(\overline{\mathbf{C}}_{-k}^H\mathbf{v}_k-\mathbf{c}_k\right)^H.\end{aligned} \tag{22}$$

By using the notation

$$\tilde{\mathbf{v}}_k\triangleq\left[\begin{array}{c}\mathbf{v}_k\\ \hline -1\end{array}\right] \tag{23}$$

[6]And since $Q\leq K$, an overall $O(K^2)$ complexity.

we find that

$$\mathbf{L}_{-k} = \left(\mathbf{L}^{-1} - \alpha_k \mathbf{C}^H \tilde{\mathbf{v}}_k \tilde{\mathbf{v}}_k^H \mathbf{C}\right)^{-1}. \tag{24}$$

From (11) and the lower part of (19), we find that

$$\tilde{\mathbf{v}}_k^H \mathbf{C} = \frac{-1}{\alpha_k} \mathbf{g}_k^H \tag{25}$$

and therefore

$$\mathbf{L}_{-k} = \left(\mathbf{L}^{-1} - \frac{1}{\alpha_k} \mathbf{g}_k \mathbf{g}_k^H\right)^{-1}. \tag{26}$$

The matrix inversion lemma can again be applied to obtain an $O(Q^2)$ update

$$\boxed{\mathbf{L}_{-k} = \mathbf{L} + \frac{\mathbf{q}_k \mathbf{q}_k^H}{\alpha_k - \mathbf{g}_k^H \mathbf{q}_k}} \tag{27}$$

where

$$\mathbf{q}_k \triangleq \mathbf{L}\mathbf{g}_k. \tag{28}$$

We define the matrix $\mathbf{T}$ as

$$\mathbf{T} \triangleq \mathbf{\Gamma} \mathbf{L} \mathbf{\Gamma}^H. \tag{29}$$

We denote the k-th diagonal element of $\mathbf{T}$ by $\tau_k \triangleq \mathbf{g}_k^H \mathbf{L} \mathbf{g}_k = \mathbf{g}_k^H \mathbf{q}_k$. By using (20) and (27), we find that

$$\mathbf{\Gamma}_{-k} \mathbf{L}_{-k} = \overline{\mathbf{\Gamma}}_{-k} \mathbf{L} + \mathbf{v}_k \mathbf{q}_k^H + \frac{1}{\alpha_k - \tau_k} \left(\overline{\mathbf{\Gamma}}_{-k} \mathbf{q}_k \mathbf{q}_k^H + \tau_k \mathbf{v}_k \mathbf{q}_k^H\right) \tag{30}$$

and by substituting (17), this can be rewritten as

$$\mathbf{\Gamma}_{-k} \mathbf{L}_{-k} = \overline{\mathbf{\Gamma}}_{-k} \mathbf{L} + \frac{1}{\alpha_k - \tau_k} \left(\overline{\mathbf{\Gamma}}_{-k} \mathbf{q}_k - \mathbf{a}_k\right) \mathbf{q}_k^H. \tag{31}$$

Note that the matrix $\mathbf{T}$ does not have to be computed, since we only need a single diagonal entry from this matrix. By multiplying (31) with $\mathbf{f}$, and using the fact that $W_k = \mathbf{q}_k^H \mathbf{f}$, we eventually obtain the following expression to compute the LCMV beamformer without sensor k, based on the current LCMV solution $\mathbf{w}$ with all sensors:

$$\boxed{\mathbf{w}_{-k} = \overline{\mathbf{w}}_{-k} + \frac{W_k}{\alpha_k - \tau_k} \left(\overline{\mathbf{\Gamma}}_{-k} \mathbf{q}_k - \mathbf{a}_k\right).} \tag{32}$$

All variables are readily available, except for $\mathbf{q}_k$ and τ_k, which can both be computed with $O(Q^2)$ complexity. Therefore, and since $Q \leq K$, (32) is an $O(KQ)$ expression.

B. Sensor Utility

In the previous subsection, we have derived an efficient formula to compute the new LCMV solution $\mathbf{w}_{-k}$ for the removal of sensor k, based on the current LCMV solution $\mathbf{w}$ with all sensors. To compute the sensor utility for sensor k, as defined in (1), we have to evaluate the LCMV cost function when this new estimator $\mathbf{w}_{-k}$ is applied. Even though $\mathbf{w}_{-k}$ can be computed with $O(KQ)$ complexity, the evaluation of the cost function $J_{-k}(\mathbf{w}_{-k}) = \mathbf{w}_{-k}^H \mathbf{R}_{yy-k} \mathbf{w}_{-k}$ is still an $O(K^2)$ procedure. Furthermore, if we want to evaluate all sensor utilities, this has to be performed K times, yielding an $O(K^3)$ procedure, which dominates the $O(K^2)$ complexity of the adaptive LCMV algorithm considered here.

In this subsection, we explain how all sensor utilities can be computed simultaneously with only $O(KQ)$ complexity. Since the number of constraints is usually much smaller than the number of sensors ($Q \ll K$), this is negligible compared to the $O(K^2)$ complexity of the adaptive beamforming algorithm. First, by substituting (6) in $J(\mathbf{w}) = \mathbf{w}^H \mathbf{R}_{yy} \mathbf{w}$, we find that

$$J(\mathbf{w}^{\mathrm{LCMV}}) = \mathbf{f}^H \mathbf{L} \mathbf{f}. \tag{33}$$

The sensor utility of sensor k can then be written as

$$U_k = J_{-k}(\mathbf{w}_{-k}^{\mathrm{LCMV}}) - J(\mathbf{w}^{\mathrm{LCMV}}) = \mathbf{f}^H \left(\mathbf{L}_{-k} - \mathbf{L}\right) \mathbf{f}. \tag{34}$$

By substituting (27) in this expression, and using the fact that $W_k = \mathbf{q}_k^H \mathbf{f}$, we find that

$$U_k = \frac{1}{\alpha_k - \tau_k} |W_k|^2. \tag{35}$$

Based on this expression, the sensor utility vector $\mathbf{u} \triangleq [U_1 \; U_2 \; \cdots \; U_K]^T$ containing all sensor utilities can be written as

$$\boxed{\mathbf{u} = \mathbf{\Sigma}^{-1} |\mathbf{w}|^2} \tag{36}$$

where the element-wise operator $|\mathbf{x}|^2$ replaces all elements in the vector $\mathbf{x}$ by their squared absolute value, and with

$$\mathbf{\Sigma} \triangleq \mathcal{D}\{\mathbf{R}_{yy}^{-1}\} - \mathcal{D}\{\mathbf{T}\} \tag{37}$$

where $\mathcal{D}\{\mathbf{X}\}$ denotes the operator that sets all off-diagonal elements of the matrix $\mathbf{X}$ to zero. The only quantity that is not readily available is $\mathcal{D}\{\mathbf{T}\}$. Since this only involves computing the diagonal elements of the matrix $\mathbf{T} = \mathbf{\Gamma} \mathbf{L} \mathbf{\Gamma}^H$, this is an $O(KQ^2)$ procedure since $\mathbf{L}$ and $\mathbf{\Gamma}$ are assumed to be available. However, in Appendix B, a recursive procedure is described that allows to track $\mathcal{D}\{\mathbf{T}\}$ with $O(KQ)$ complexity. It is noted that $\mathbf{\Sigma}$ is a diagonal matrix, and therefore the matrix inversion in (36) actually consists of K scalar inversions. Therefore, the computation of $\mathbf{u}$ has an overall $O(K)$ complexity if we assume that $\mathcal{D}\{\mathbf{T}\}$ is readily available. However, since the latter requires an $O(KQ)$ recursive updating procedure, the computational complexity of computing the LCMV sensor utilities is also $O(KQ)$ (or $O(KQ^2)$ if $\mathcal{D}\{\mathbf{T}\}$ is computed directly without recursion).

IV. Linear MMSE Signal Estimation

In this section, we consider linear MMSE (LMMSE) signal estimation, also often referred to as multi-channel Wiener filtering (MWF). MWF is often used in signal enhancement,

in particular speech enhancement with microphone arrays or acoustic sensor networks [10]–[12], [16], [28], [29].

The goal is to estimate a complex valued desired signal d from the sensor signals $\mathbf{y}$. We consider the general case where d is not an observed signal, i.e., it is assumed to be unknown (e.g., in speech enhancement, d is the speech component in a noisy reference microphone signal [11], [12]). We consider LMMSE signal estimation, i.e., a linear estimator $\hat{d} = \mathbf{w}^H\mathbf{y}$ that minimizes the MSE cost function

$$J(\mathbf{w}) \triangleq E\{|d - \mathbf{w}^H\mathbf{y}|^2\}. \tag{38}$$

The minimizer of (38) is [34]:

$$\boxed{\mathbf{w}^{\mathrm{LMMSE}} = \mathbf{R}_{yy}^{-1}\mathbf{r}_{yd}} \tag{39}$$

with $\mathbf{r}_{yd} \triangleq E\{\mathbf{y}d^*\}$, where d^* denotes the complex conjugate of d. Since d is assumed to be unknown, the estimation of the correlation vector $\mathbf{r}_{yd}$ has to be done indirectly, based on application-specific strategies, e.g., by exploiting the on-off behavior of the target signal source (as in speech enhancement [10]–[12], [16]), by periodic broadcasts of known training sequences, or by incorporating prior knowledge on the signal statistics in case of (partially) static scenarios [32].

Based on the solution (39), we assume an adaptive MWF or LMMSE algorithm that recursively updates the following variables:

$$\boxed{\begin{array}{ll} \bullet\ \mathbf{R}_{yy}^{-1} & (K \times K) \\ \bullet\ \mathbf{r}_{yd} & (K \times 1) \\ \bullet\ \mathbf{w} & (K \times 1) \end{array}} \tag{40}$$

A. Sensor Removal

When sensor k is removed, the optimal LMMSE solution is

$$\mathbf{w}_{-k} = \mathbf{R}_{yy-k}^{-1}\mathbf{r}_{yd-k} \tag{41}$$

where $\mathbf{R}_{yy-k} \triangleq E\{\mathbf{y}_{-k}\mathbf{y}_{-k}^H\}$ and $\mathbf{r}_{yd-k} \triangleq E\{\mathbf{y}_{-k}d^*\}$. For the sake of an easy exposition, but w.l.o.g., we again assume that $k = K$. We again use the block partitioning of $\mathbf{R}_{yy}^{-1}$ and $\mathbf{w}$ defined in (9) and (10), respectively.

Similar to (18), which is repeated here for convenience, $\mathbf{R}_{yy-k}^{-1}$ can be efficiently computed as

$$\boxed{\mathbf{R}_{yy-k}^{-1} = \mathbf{A}_k - \tfrac{1}{\alpha_k}\mathbf{a}_k\mathbf{a}_k^H} \tag{42}$$

and therefore the LMMSE solution with sensor k removed is

$$\mathbf{w}_{-k} = \left(\mathbf{A}_k - \frac{1}{\alpha_k}\mathbf{a}_k\mathbf{a}_k^H\right)\mathbf{r}_{yd-k}. \tag{43}$$

By plugging (9) and (10) into (39) we find that

$$\overline{\mathbf{w}}_{-k} = \mathbf{A}_k\mathbf{r}_{yd-k} + r_{yd}(k)\mathbf{a}_k \tag{44}$$

$$W_k = \mathbf{a}_k^H\mathbf{r}_{yd-k} + \alpha_k r_{yd}(k) \tag{45}$$

where $r_{yd}(k)$ denotes the k-th element of the correlation vector $\mathbf{r}_{yd}$. By comparing (43) with (44) and (45), we find that

$$\boxed{\mathbf{w}_{-k} = \overline{\mathbf{w}}_{-k} - \tfrac{W_k}{\alpha_k}\mathbf{a}_k.} \tag{46}$$

Since all variables in (46) are directly available, expression (46) has only $O(K)$ complexity. Note that this formula is similar to the LCMV sensor removal formula (32), even though the definition of the optimal estimator $\mathbf{w}$ is very different in both cases. Indeed, if we set $\mathbf{C}$ in the LCMV formulation to zero, then both sensor removal formulas (32) and (46) are equivalent.

B. Sensor Utility

By plugging (39) into (38), we find that the MMSE cost is

$$J(\mathbf{w}^{\mathrm{LMMSE}}) = P_d - \mathbf{r}_{yd}^H\mathbf{R}_{yy}^{-1}\mathbf{r}_{yd} \tag{47}$$

$$= P_d - \mathbf{r}_{yd}^H\mathbf{w}^{\mathrm{LMMSE}} \tag{48}$$

with $P_d = E\{|d|^2\}$. With (48) and the definition of U_k as introduced in (1), we find that

$$U_k = \mathbf{r}_{yd}^H\mathbf{w} - \mathbf{r}_{yd-k}^H\mathbf{w}_{-k}. \tag{49}$$

By using (46), and by using the partitioning of $\mathbf{w}$ as defined in (10), we can rewrite (49) as

$$U_k = W_k r_{yd}(k)^* + \frac{W_k}{\alpha_k}\mathbf{r}_{yd-k}^H\mathbf{a}_k. \tag{50}$$

From (45), we find that

$$\mathbf{r}_{yd-k}^H\mathbf{a}_k = W_k^* - \alpha_k r_{yd}(k)^*. \tag{51}$$

By substituting (51) in (50), we find that

$$U_k = \frac{1}{\alpha_k}|W_k|^2. \tag{52}$$

To monitor all the sensor utilities simultaneously, i.e., the utility vector $\mathbf{u} = [U_1\ U_2\ \cdots\ U_K]^T$, it is thus sufficient to monitor the squared components of the current estimator $\mathbf{w}$, normalized by the diagonal elements of the *inverted* correlation matrix $\mathbf{R}_{yy}^{-1}$, i.e.,

$$\boxed{\mathbf{u} = \mathbf{\Lambda}^{-1}|\mathbf{w}|^2} \tag{53}$$

with

$$\mathbf{\Lambda} \triangleq \mathcal{D}\{\mathbf{R}_{yy}^{-1}\}. \tag{54}$$

Formula (53) has $O(K)$ complexity, which is negligible compared to the complexity of the estimator updates (which is $O(K^2)$ due to the update of the inverse correlation matrix $\mathbf{R}_{yy}^{-1}$ based on (3)). Again, notice the similarity with the formula of the LCMV beamformer in (36), i.e., both are the same when setting $\mathbf{C}$ in the LCMV formulation to zero.

V. Distortion-Weighted LMMSE Signal Estimation

In this section, we assume that the sensor signals satisfy the general additive noise model

$$\mathbf{y} = \mathbf{d} + \mathbf{n} \tag{55}$$

where $\mathbf{d}$ is a K-channel desired signal vector (containing K different (filtered) versions of the target source signal(s), as observed by the K sensors) and $\mathbf{n}$ is a K-channel noise vector, which is assumed to be uncorrelated with d and $\mathbf{d}$. W.l.o.g., we also assume that all signals are zero-mean, such that $E\{\mathbf{d}\mathbf{n}^H\}$ is equal to an all-zero matrix. We consider the same MSE cost function as in (38), but we rewrite it as

$$J(\mathbf{w}) = E\{|d - \mathbf{w}^H\mathbf{d}|^2\} + E\{|\mathbf{w}^H\mathbf{n}|^2\}. \tag{56}$$

Note that the first term corresponds to distortion of the desired signal component, whereas the second term corresponds to the actual noise reduction. To obtain a better noise reduction at the price of more distortion (or vice versa), a trade-off parameter μ is introduced in (56) (see also [11]), i.e.,:

$$J(\mathbf{w}) = E\{|d - \mathbf{w}^H\mathbf{d}|^2\} + \mu E\{|\mathbf{w}^H\mathbf{n}|^2\}. \tag{57}$$

The estimator minimizing this cost function is the so-called distortion-weighted LMMSE (DW-LMMSE) estimator given by

$$\boxed{\mathbf{w}^{\text{DW-LMMSE}} = \widetilde{\mathbf{R}}_{yy}^{-1}\mathbf{r}_{yd}} \tag{58}$$

with

$$\widetilde{\mathbf{R}}_{yy} \triangleq \mathbf{R}_{dd} + \mu\mathbf{R}_{nn} \tag{59}$$

where $\mathbf{R}_{dd} \triangleq E\{\mathbf{d}\mathbf{d}^H\}$, $\mathbf{R}_{nn} \triangleq E\{\mathbf{n}\mathbf{n}^H\}$ and $\mathbf{r}_{yd} \triangleq E\{\mathbf{y}d^*\}$. It is noted that, when $\mu = 1$, this is equal to the LMMSE or MWF estimator (39). If $\mu \neq 1$, the computation of DW-LMMSE estimator requires a separate estimate of the noise correlation matrix $\mathbf{R}_{nn}$ and the desired signal correlation matrix $\mathbf{R}_{dd}$. In the sequel, we assume that both can indeed be estimated[7].

Based on the solution (58), we assume an adaptive DW-LMMSE signal estimation algorithm that updates the following variables:

$$\boxed{\begin{array}{ll} \bullet\ \widetilde{\mathbf{R}}_{yy}^{-1} = (\mathbf{R}_{dd} + \mu\mathbf{R}_{nn})^{-1} & (K \times K) \\ \bullet\ \mathbf{r}_{yd} & (K \times 1) \\ \bullet\ \mathbf{w} & (K \times 1) \end{array}} \tag{60}$$

Note that the adaptation of $\widetilde{\mathbf{R}}_{yy}^{-1}$ cannot rely on the recursive update (3) when $\mu \neq 1$, and therefore a full inversion has to be performed in each iteration, which results in a computational complexity of $O(K^3)$.

A. Sensor Removal

The derivation of the sensor removal formula for the DW-LMMSE estimator is almost identical to the derivation in Section IV-A, and is omitted here. Not surprisingly, the resulting formula is identical to (46), with $\mathbf{R}_{yy}$ replaced by $\widetilde{\mathbf{R}}_{yy}$:

$$\boxed{\mathbf{w}_{-k} = \overline{\mathbf{w}}_{-k} - \frac{W_k}{\alpha_k}\widetilde{\mathbf{a}}_k} \tag{61}$$

with

$$\widetilde{\mathbf{R}}_{yy}^{-1} = \left[\begin{array}{c|c} \widetilde{\mathbf{A}}_k & \widetilde{\mathbf{a}}_k \\ \hline \widetilde{\mathbf{a}}_k^H & \widetilde{\alpha}_k \end{array}\right]. \tag{62}$$

B. Sensor Utility

The derivation is almost identical as in Section IV-B, which results in the utility vector

$$\boxed{\mathbf{u} = \widetilde{\mathbf{\Lambda}}^{-1}|\mathbf{w}|^2} \tag{63}$$

with

$$\widetilde{\mathbf{\Lambda}} \triangleq \mathcal{D}\{\widetilde{\mathbf{R}}_{yy}^{-1}\}. \tag{64}$$

VI. Rank-1 Distortion-Weighted LMMSE Signal Estimation

In this section, we assume that the desired component $\mathbf{d}$ in the sensor signals satisfies the rank-1 model

$$\mathbf{y} = \mathbf{d} + \mathbf{n} = \mathbf{h}s + \mathbf{n} \tag{65}$$

where $\mathbf{h}$ is a deterministic steering vector and s is the target source signal. Such models are often encountered in beamforming applications with a single target source. It is noted that the steering vector is not necessarily assumed to be known. Therefore, the aim is to estimate the signal s as it impinges on one of the sensors, referred to as the reference sensor. W.l.o.g. we assume that the reference sensor is the first sensor, and therefore the desired signal is $d = h_1 s = \mathbf{e}_1^T\mathbf{d}$ where h_1 denotes the first element of $\mathbf{h}$ and $\mathbf{e}_1$ is the K-dimensional all-zero vector except for a one in the first entry. Note that whenever the steering vector $\mathbf{h}$ is known, an LCMV beamformer (Section III) can be used instead of the estimator presented in this section.

We consider the same distortion-weighted MSE cost function as in (57), repeated here for convenience:

$$J(\mathbf{w}) = E\{|d - \mathbf{w}^H\mathbf{d}|^2\} + \mu E\{|\mathbf{w}^H\mathbf{n}|^2\}. \tag{66}$$

It can be shown that, under the rank-1 assumption (65), expression (58) can also be written as [12]

$$\boxed{\mathbf{w}^{\text{R1-MWF}} = \mathbf{R}_{nn}^{-1}\mathbf{R}_{dd}\mathbf{e}_1\frac{1}{\mu + \text{Tr}\{\mathbf{R}_{nn}^{-1}\mathbf{R}_{dd}\}}} \tag{67}$$

where $\text{Tr}\{.\}$ denotes the trace operator. We refer to this implementation as the rank-1 multichannel Wiener filter[8] (R1-MWF) [12].

Expression (67) has two major advantages over (58). First, the estimate of the inverse noise correlation matrix can be recursively updated similarly to (3) during noise-only signal segments, which again yields an $O(K^2)$ algorithm (remember that

[7]For example, this is possible in speech enhancement applications, where $\mathbf{R}_{nn}$ can be adaptively estimated during signal segments where the desired speech source is not active, and $\mathbf{R}_{dd}$ can be computed as $\mathbf{R}_{yy} - \mathbf{R}_{nn}$.

[8]It can be shown that, if μ is set to zero, the solution (67) corresponds to the so-called minimum variance distortionless response (MVDR) beamformer based on the single constraint $\mathbf{h}^H\mathbf{w} = h_1^*$ so that the beamformer response is distortionless with respect to the desired signal component in the reference microphone.

this was not possible for the matrix $\widetilde{\mathbf{R}}_{yy}^{-1}$ in Section V). Furthermore, although R1-MWF and (DW)-LMMSE are theoretically equivalent under a rank-1 assumption, (67) is observed to significantly outperform the LMMSE estimator (39) or the DW-LMMSE estimator (58) in practical implementations. A theoretical analysis in [12] has demonstrated that expression (67) is indeed numerically more favorable than (39) or (58).

However, with respect to sensor removal and utility in the R1-MWF, unlike in the DW-LMMSE of Section V, $\widetilde{\mathbf{R}}_{yy}^{-1}$ is not used and hence not directly available. Therefore, the DW-LMMSE expressions for the sensor removal (61) and the sensor utility vector (63) cannot be used here.

For notational convenience, we define

$$\mathbf{D} \triangleq \mathbf{R}_{nn}^{-1}\mathbf{R}_{dd} \tag{68}$$

$$\mathbf{\Lambda}_D \triangleq \mathcal{D}\{\mathbf{D}\} \tag{69}$$

$$\boldsymbol{\lambda}_D \triangleq \mathbf{\Lambda}_D \mathbf{1}_K \tag{70}$$

where $\mathbf{1}_K$ denotes a K-dimensional vector with all entries equal to 1. Since only the trace of $\mathbf{D}$ is required to compute (67), we only need its diagonal elements, i.e., the vector $\boldsymbol{\lambda}_D$, which can be computed in $O(K^2)$. Therefore, the overall complexity of the adaptive algorithm implementing (67) is $O(K^2)$, where we can assume that the following variables are recursively updated:

$$\boxed{\begin{array}{ll} \bullet\ \mathbf{R}_{nn}^{-1} & (K\times K)\\ \bullet\ \mathbf{R}_{dd} & (K\times K)\\ \bullet\ \boldsymbol{\lambda}_D & (K\times 1)\\ \bullet\ \mathbf{w} & (K\times 1)\end{array}} \tag{71}$$

Note that $\mathbf{R}_{dd}$ can be computed as $\mathbf{R}_{dd} = \mathbf{R}_{yy} - \mathbf{R}_{nn}$. Based on the above variables, we will derive formulas for sensor removal and to compute the utility vectors with an $O(K)$ complexity, which is again negligible compared to the $O(K^2)$ complexity of the signal estimation itself.

A. Sensor Removal

We again assume that sensor $k = K$ is removed. In this case, the optimal DW-LMMSE solution is

$$\mathbf{w}_{-k} = \mathbf{R}_{nn-k}^{-1}\mathbf{R}_{dd-k}\mathbf{e}_1 \frac{1}{\mu + \mathrm{Tr}\{\mathbf{R}_{nn-k}^{-1}\mathbf{R}_{dd-k}\}} \tag{72}$$

where $\mathbf{R}_{nn-k} \triangleq E\{\mathbf{n}_{-k}\mathbf{n}_{-k}^H\}$ and $\mathbf{R}_{dd-k} \triangleq E\{\mathbf{d}_{-k}\mathbf{d}_{-k}^H\}$. We again use the block partitioning of $\mathbf{w}$ defined in (10). We define the block partitioning of $\mathbf{R}_{nn}^{-1}$ as

$$\mathbf{R}_{nn}^{-1} = \left[\begin{array}{c|c} \mathbf{B}_k & \mathbf{b}_k \\ \hline \mathbf{b}_k^H & \beta_k \end{array}\right]. \tag{73}$$

We define the variable

$$\boxed{\nu \triangleq \mu + \mathrm{Tr}\{\mathbf{R}_{nn}^{-1}\mathbf{R}_{dd}\} = \mu + \mathrm{Tr}\{\mathbf{D}\} = \mu + \boldsymbol{\lambda}_D^T\mathbf{1}_K} \tag{74}$$

which can be computed with $O(K)$ complexity. We also define

$$\mathbf{D}_{-k} \triangleq \mathbf{R}_{nn-k}^{-1}\mathbf{R}_{dd-k}. \tag{75}$$

It is noted that

$$\nu\,\mathbf{w} = \mathbf{R}_{nn}^{-1}\mathbf{R}_{dd}\mathbf{e}_1 \tag{76}$$

$$(\mu + \mathrm{Tr}\{\mathbf{D}_{-k}\})\,\mathbf{w}_{-k} = \mathbf{R}_{nn-k}^{-1}\mathbf{R}_{dd-k}\mathbf{e}_1. \tag{77}$$

The righthand parts have a similar form as the LMMSE estimators (39) and (41), respectively. Therefore, based on the $O(K)$ formula for the sensor removal in the LMMSE estimator given in (46), we find that

$$\mathbf{w}_{-k} = \left(\overline{\mathbf{w}}_{-k} - \frac{W_k}{\beta_k}\mathbf{b}_k\right)\frac{\nu}{\mu + \mathrm{Tr}\{\mathbf{D}_{-k}\}}. \tag{78}$$

For notational convenience, we define the following variables, which will be used in the sequel:

$$\mathbf{\Lambda}_{nn} \triangleq \mathcal{D}\{\mathbf{R}_{nn}^{-1}\} \tag{79}$$

$$\mathbf{\Lambda}_{dd} \triangleq \mathcal{D}\{\mathbf{R}_{dd}\} \tag{80}$$

$$\mathbf{\Lambda}_{nd} \triangleq \mathbf{\Lambda}_{nn}\mathbf{\Lambda}_{dd}. \tag{81}$$

In [8], an efficient formula is derived to compute $\mathrm{Tr}\{\mathbf{D}_{-k}\}$ with $O(K)$ complexity, based on the variables in (71):

$$\begin{bmatrix} \mathrm{Tr}\{\mathbf{D}_{-1}\} \\ \vdots \\ \mathrm{Tr}\{\mathbf{D}_{-K}\} \end{bmatrix} = \mathrm{Tr}\{\mathbf{D}\}\mathbf{1}_K - \mathbf{\Lambda}_{nd}^{-1}|\boldsymbol{\lambda}_D|^2. \tag{82}$$

Define the diagonal matrix

$$\boxed{\mathbf{\Phi} \triangleq \mathbf{I}_K - \tfrac{1}{\nu}\mathbf{\Lambda}_{nd}^{-1}|\mathbf{\Lambda}_D|^2} \tag{83}$$

which can be computed with $O(K)$ complexity. We denote ϕ_k as the k-th diagonal element of $\mathbf{\Phi}$. Combining (78), (82) and (83), we find that

$$\boxed{\mathbf{w}_{-k} = \tfrac{1}{\phi_k}\left(\overline{\mathbf{w}}_{-k} - \tfrac{W_k}{\beta_k}\mathbf{b}_k\right).} \tag{84}$$

Again, note the similarity between this formula and the sensor removal formulas in the previous sections.

B. Sensor Utility

Since the R1-MWF is a special case of the more general DW-LMMSE signal estimator (58), the sensor utility vector $\mathbf{u}$ of R1-MWF will be equal to the DW-LMMSE utility vector (63). We can therefore start by stating that

$$\mathbf{u} = \widetilde{\mathbf{\Lambda}}^{-1}|\mathbf{w}|^2 = \left(\mathcal{D}\{\widetilde{\mathbf{R}}_{yy}^{-1}\}\right)^{-1}|\mathbf{w}|^2. \tag{85}$$

However, $\widetilde{\mathbf{R}}_{yy}^{-1}$ is not explicitly available here. By using the rank-1 assumption (65), we know that $\mathbf{R}_{dd} = P_s\mathbf{h}\mathbf{h}^H$ with $P_s \triangleq E\{|s|^2\}$ denoting the power of the target source signal, and therefore

$$\widetilde{\mathbf{R}}_{yy}^{-1} = (\mathbf{R}_{dd} + \mu\mathbf{R}_{nn})^{-1} = \left(P_s\mathbf{h}\mathbf{h}^H + \mu\mathbf{R}_{nn}\right)^{-1}. \tag{86}$$

By using the Woodbury identity [34], we can rewrite this as

$$\widetilde{\mathbf{R}}_{yy}^{-1} = \frac{1}{\mu}\left(\mathbf{R}_{nn}^{-1} - \frac{P_s\mathbf{R}_{nn}^{-1}\mathbf{h}\mathbf{h}^H\mathbf{R}_{nn}^{-1}}{\mu + P_s\mathbf{h}^H\mathbf{R}_{nn}^{-1}\mathbf{h}}\right). \tag{87}$$

TABLE I
COMPARISON OF THE NUMBER OF FLOPS FOR SENSOR REMOVAL

LCMV (straightforward)	$\frac{2}{3}(K-1)^3+\frac{2}{3}(Q)^3+2(Q-1)(K-1)^2+2(Q-1)^2(K-1)+2Q^2$
LCMV (with (32))	$(2Q+3)(K-1)+2Q(Q+1)$
LMMSE (straightforward)	$\frac{2}{3}(K-1)^3+2(K-1)^2$
LMMSE (with (46))	$2(K-1)$
DW-LMMSE (straightforward)	$\frac{2}{3}(K-1)^3+2(K-1)^2$
DW-LMMSE (with (61))	$2(K-1)$
R1-MWF (straightforward)	$\frac{2}{3}(K-1)^3+2(K-1)^2+2(K-1)(K-2)+(K-1)$
R1-MWF (with (84))	$3(K-1)+(K+1)+6$

TABLE II
COMPARISON OF THE NUMBER OF FLOPS FOR UTILITY COMPUTATION. THE $*$ DENOTE A NUMBER OF FLOPS REQUIRED TO COMPUTE THE SENSOR REMOVAL OF A SINGLE SENSOR, BASED ON THE CORRESPONDING ROW IN TABLE I

LCMV (straightforward)	$K(4Q+*)$
LCMV (with (36))	$4K$
LMMSE (straightforward)	$K(2K+2(K-1)+*)$
LMMSE (with (53))	$3K$
DW-LMMSE (straightforward)	$K(2K+2(K-1)+*)$
DW-LMMSE (with (63))	$3K$
R1-MWF (straightforward)	$K(2K+2(K-1)+*)$
R1-MWF (with (96))	$11K$

Since

$$P_s\mathbf{h}^H\mathbf{R}_{nn}^{-1}\mathbf{h} = \mathrm{Tr}\{P_s\mathbf{h}^H\mathbf{R}_{nn}^{-1}\mathbf{h}\} \quad (88)$$
$$= \mathrm{Tr}\{\mathbf{R}_{nn}^{-1}P_s\mathbf{h}\mathbf{h}^H\} \quad (89)$$
$$= \mathrm{Tr}\{\mathbf{R}_{nn}^{-1}\mathbf{R}_{dd}\} \quad (90)$$

we can rewrite (87) as

$$\widetilde{\mathbf{R}}_{yy}^{-1} = \frac{1}{\mu}\left(\mathbf{R}_{nn}^{-1} - \frac{1}{\nu}P_s\mathbf{R}_{nn}^{-1}\mathbf{h}\mathbf{h}^H\mathbf{R}_{nn}^{-1}\right). \quad (91)$$

Let δ_k denote the k-th diagonal element of $\mathbf{D}$ and $\mathbf{p}_k \triangleq [\mathbf{b}_k^T \mid \beta_k]^T$ denote the k-th column of $\mathbf{R}_{nn}^{-1}$, then we know from the definition of $\mathbf{D}$ (68) that

$$\delta_k = P_s\mathbf{p}_k^H\mathbf{h}h_k^* \quad (92)$$

with h_k denoting the k-th entry of $\mathbf{h}$. By multiplying (92) with its conjugate, we find that

$$|\delta_k|^2 = P_s^2|h_k|^2\mathbf{p}_k^H\mathbf{h}\mathbf{h}^H\mathbf{p}_k. \quad (93)$$

From this, we find that

$$\mathcal{D}\{P_s\mathbf{R}_{nn}^{-1}\mathbf{h}\mathbf{h}^H\mathbf{R}_{nn}^{-1}\} = \mathbf{\Lambda}_{dd}^{-1}\left|\mathbf{\Lambda}_D\right|^2 \quad (94)$$

where $\mathbf{\Lambda}_{dd}^{-1}$ is defined in (80). Plugging (94) into (91), and using the definitions (79) and (80) yields

$$\mathcal{D}\{\widetilde{\mathbf{R}}_{yy}^{-1}\} = \frac{1}{\mu}\left(\mathbf{\Lambda}_{nn} - \frac{1}{\nu}\mathbf{\Lambda}_{dd}^{-1}\left|\mathbf{\Lambda}_D\right|^2\right). \quad (95)$$

Combining this with the definition of the diagonal matrix $\mathbf{\Phi}$ given in (83), and using expression (85), we eventually find that the sensor utility vector can be computed as

$$\boxed{\mathbf{u} = \mathbf{\Psi}^{-1}|\mathbf{w}|^2} \quad (96)$$

with

$$\mathbf{\Psi} \triangleq \frac{1}{\mu}\mathbf{\Lambda}_{nn}\mathbf{\Phi}. \quad (97)$$

This is again an $O(K)$ formula since all matrices involved are diagonal. It is noted that, if the rank-1 assumption (65) holds, we know by construction that $\mathbf{\Psi}$ is equal to $\widetilde{\mathbf{\Lambda}}$ as defined in (64). If, in addition, $\mu = 1$, we know that $\mathbf{\Psi} = \widetilde{\mathbf{\Lambda}} = \mathbf{\Lambda}$, i.e., the utility vectors (53), (63) and (96) will indeed all be identical.

VII. COMPLEXITY AND SENSITIVITY EVALUATION

A. Analytical Complexity Analysis

In this subsection, we provide a complexity evaluation, based on the number of floating point operations (flops) that are required to compute each formula (i.e., the efficient formulas derived in the paper, and the direct (straightforward) computation of the same variables). A flop can be an addition or a multiplication, i.e., additions and multiplications are treated equally. Furthermore, we assume in the complexity analysis that

- the inversion[9] of a $K \times K$ matrix requires $\frac{2}{3}K^3$ flops
- the multiplication of a $K \times P$ with a $P \times N$ matrix requires $2KPN$ flops.

It is noted that the number of flops as listed above are general worst-case values to give a general idea of the complexity reduction that can be achieved. The algorithms from which these numbers are derived neither exploit possible structure in the matrices (such as symmetry or certain case-specific structures), nor do they exploit advanced techniques such as Strassen's method for matrix multiplication. In general, such efficient implementations only yield a minor reduction in computational complexity, and the general conclusion that is drawn from the complexity analysis below remains valid.

Table I shows the required number of flops for the sensor removal for each of the four signal estimation algorithm. It compares the number of flops between the efficient formulas, i.e., (32), (46), (61) and (84), and the direct (straightforward) computation, i.e., by entirely recomputing the beamformer coefficients, using (6), (39), (58), or (67). Table II shows the same for the utility computation. The $*$ in Table II should be replaced by the number of flops given in Table I (at the corresponding row). This reflects the fact that a straightforward computation of the utility based on (1) requires the computation of a new estimator where one of the sensors is removed, and therefore refers to the first table. The number of flops shown in Table II, for the LCMV sensor utilities computed with (36), assumes that

[9]We assume the use of an LU decomposition, rather than an explicit computation of the matrix inverse (see [34]).

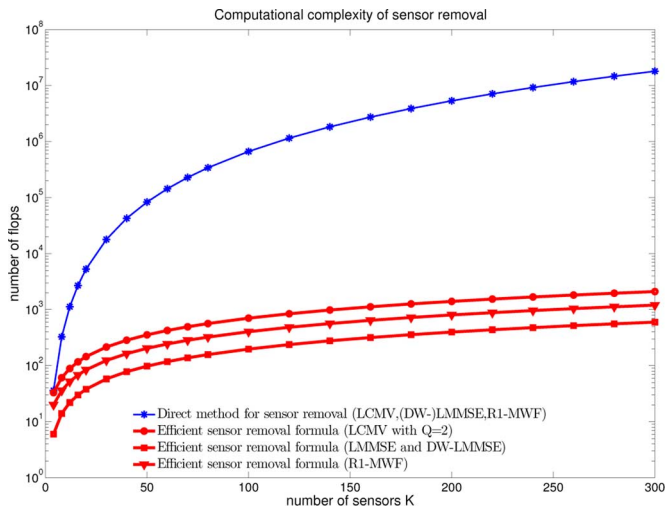

Fig. 1. Number of flops required for sensor removal, as a function of the number of sensors K.

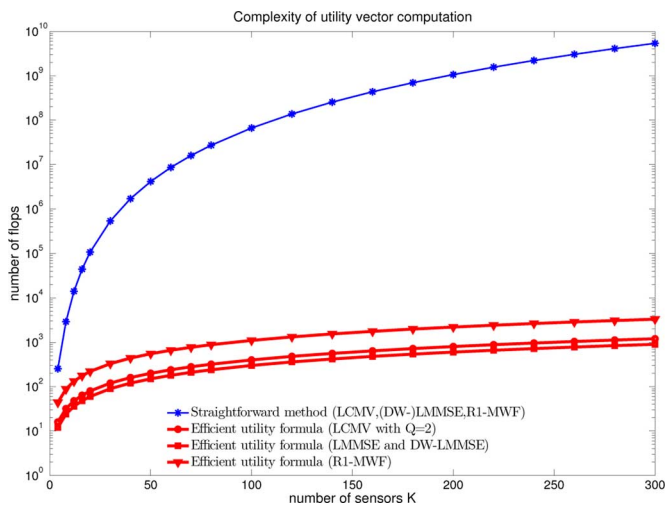

Fig. 2. Number of flops required for utility computation, as a function of the number of sensors K.

$\mathcal{D}\{\mathbf{T}\}$ is readily available, e.g., based on the recursive update given in Appendix B.

To illustrate the computational advantage, Fig. 1 shows the number of flops required for sensor removal, as a function of the number of sensors K, for the four signal estimation algorithms (where Q in LCMV is set to $Q = 2$). It is noted that DW-LMMSE signal estimation has the same complexity as LMMSE signal estimation. Furthermore, the number of flops in the straightforward computation is almost identical for all four algorithms, and therefore only one curve is shown for the sake of intelligibility. Fig. 2 shows the same for utility computation. It is noted that a significant computational advantage has been achieved. In particular for the case of utility computation, the number of computations (for large K) has been reduced by approximately 7 orders of magnitude, i.e., from 10^{10} to 10^3.

Finally, Fig. 3 shows the number of flops required for the sensor removal and utility computation in an LCMV beamformer, as a function of the number of constraints Q for the case where $K = 300$. It shows both the case where $\mathcal{D}\{\mathbf{T}\}$ is readily available (e.g., from the recursive update in Appendix B), or

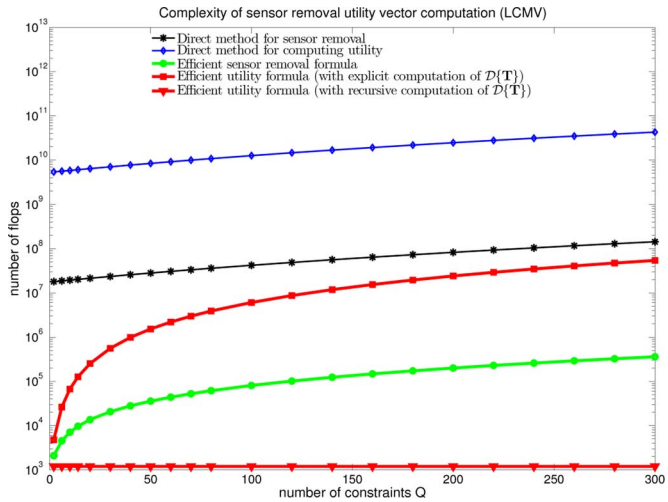

Fig. 3. Number of flops required for utility computation in LCMV, as a function of the number of constraints Q.

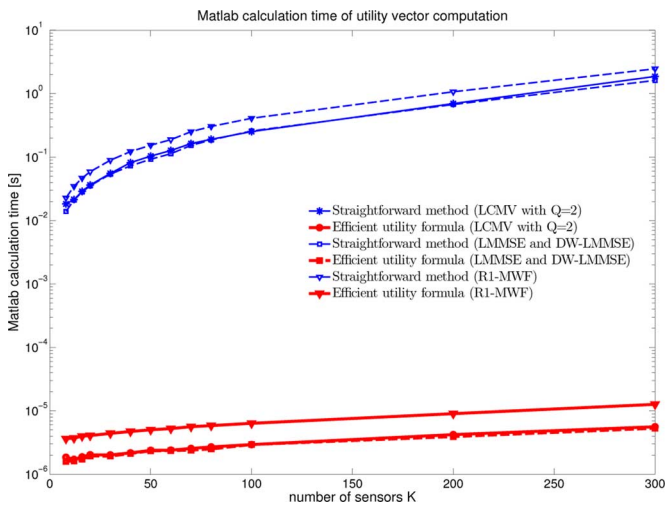

Fig. 4. Utility calculation time as a function of the number of sensors K, averaged over 200 MC trials.

when it has to be computed explicitly. Notice that the computation of the utility vector is independent of Q in the former case, i.e., when $\mathcal{D}\{\mathbf{T}\}$ is assumed to be readily available.

B. Simulation-Based Complexity Analysis

Fig. 4 compares the average Matlab calculation time for the utility computation, as a function of the number of sensors K, for the four signal estimation algorithms (where Q in LCMV is set to $Q = 2$). The results are averaged over 200 Monte Carlo (MC) trials. The calculation time shown in Fig. 4 only incorporate the calculation of the utility vectors, i.e., they do not incorporate the calculations required by the actual beamformer or signal estimators. It is noted that Fig. 4 only gives a rough comparison of the practical calculation time, i.e., it does not necessarily reflect the computation time in fully-optimized implementations on dedicated platforms. The difference between Figs. 2 and 4 may be explained by overhead in the Matlab environment and because certain assumptions were made in the complexity analysis which may not hold in a Matlab implementation (cfr. Section VII-A). Furthermore, the numbers given in

Tables I and II are based on a worst-case analysis without exploiting structure or efficient matrix manipulation algorithms (cfr. Section VII-A).

C. Simulation-Based Sensitivity Analysis

We have demonstrated that the utility vectors can be computed in an efficient way by using the techniques in Sections III–VI. Even though these techniques are theoretically equivalent to the straightforward utility computations based on the utility definition in (1), they may have a different accuracy in case of finite-precision calculations. In this subsection, we provide simulation results that compare the sensitivity to round-off errors of both approaches. For practical reasons, we use a simplistic and inaccurate model to simulate finite-precision (fixed-point) calculations. Therefore, the results only provide a rough comparison, i.e., they do not perfectly reflect the true behavior in finite-precision implementations.

The round-off errors are modeled by quantizers with quantization level δ, i.e., $Q_\delta(x) = \delta(\frac{x}{\delta})_{\text{round}}$, where $(.)_{\text{round}}$ is the operator that replaces its argument by the closest integer number. To model finite-precision calculations, we have introduced such a quantizer before and after each operation that is performed in the utility calculation. For example, a finite-precision implementation for the inversion of matrix $\mathbf{X}$ is modeled by performing the following operations in Matlab:

1) Each element of $\mathbf{X}$ is quantized, yielding the quantized matrix $Q_\delta(\mathbf{X})$.
2) $Q_\delta(\mathbf{X})$ is inverted with Matlab accuracy, yielding $(Q_\delta(\mathbf{X}))^{-1}$.
3) The resulting matrix $(Q_\delta(\mathbf{X}))^{-1}$ is again quantized, yielding $Q_\delta\left((Q_\delta(\mathbf{X}))^{-1}\right)$.

Similar procedures are used to model finite-precision implementations of other high-level operators. It is noted that practical finite-precision implementations will also apply quantizers in each elementary operation inside the operator. Therefore, the total round-off error as provided by our simulations is underestimated. Nevertheless, the simulations provide at least a rough comparison.

We have performed 4000 MC trials, where the geometrically-averaged condition number of $\mathbf{R}_{yy}$ was equal to 9111. We define the average relative error e_r between the finite-precision utility U_{FP} and the ground-truth utility U (which is computed with Matlab accuracy) as

$$e_r = \frac{\sum_{\text{MC trials}} |U_{FP} - U|}{\sum_{\text{MC trials}} U}. \tag{98}$$

It is noted that the ground-truth utility U assumes perfect knowledge of the signal statistics ($\mathbf{R}_{yy}$, $\mathbf{r}_{yd}$, $\mathbf{R}_{nn}$, and $\mathbf{R}_{dd}$), whereas the finite-precision utility U_{FP} also contains estimation errors due to practical estimation of the signal statistics, in addition to the round-off errors in its calculations. To estimate $\mathbf{r}_{yd}$ and $\mathbf{R}_{dd}$, we used similar techniques as in [11], [12].

Fig. 5 shows the relative error e_r as a function of the (log-transformed) quantization level δ, for the four signal estimation algorithms. Since $\mu = 1$, LMMSE and DW-LMMSE are equivalent. It is observed that the utility of the LCMV beamformer is less sensitive to quantization than the other signal estimation algorithms. This is because LCMV does not require the estimation of either $\mathbf{r}_{yd}$ or $\mathbf{R}_{dd}$, which is error-prone since these cannot be computed directly by means of temporal averaging, as the signals d and $\mathbf{d}$ are not observed.

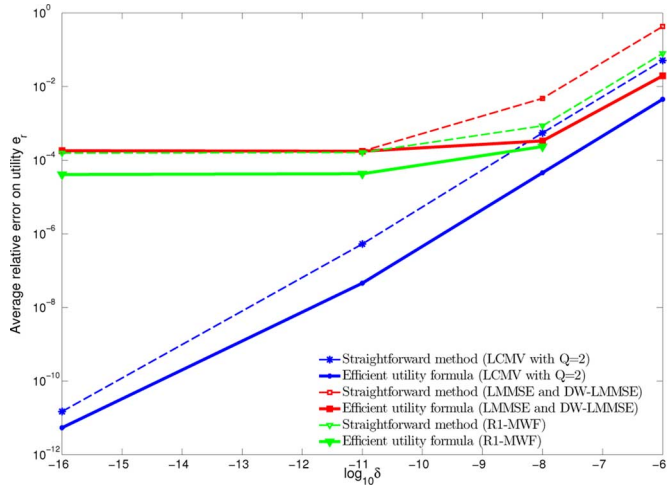

Fig. 5. Average relative error e_r of the utility computations, as a function of the (log-transformed) quantization level δ, averaged over 4000 MC trials.

In fact, for small δ, the estimation errors on $\mathbf{r}_{yd}$ or $\mathbf{R}_{dd}$ in the (DW-)LMMSE and R1-MWF algorithms dominate the round-off errors, which explains the flat curves for small δ. In general, it can be observed that the straightforward computation of utility is significantly more sensitive to round-off errors compared to the efficient formulas. This is not unexpected, since the straightforward approach involves more operations, including matrix inversions, yielding a longer accumulation of round-off errors. It is noted that, in these simulations, the order of magnitude of the ground-truth utility U is approximately 10^{-6}, hence the error e_r will become very large if $\delta > 10^{-6}$.

VIII. Conclusions

In this paper, we have derived efficient formulas for the computation of sensor utilities in four different adaptive signal estimation and beamforming algorithms: LCMV (including MVDR), LMMSE (or MWF), DW-LMMSE and R1-MWF. We have shown that the sensor utilities can be computed with hardly any increase in memory usage and computational complexity compared to the signal estimation or beamforming algorithm already in place. This allows for on-line tracking of all the sensor utilities at almost no additional cost. Furthermore, we have derived efficient formulas that allow to quickly update the signal estimators or beamformers when a sensor signal is removed. Based on numerical simulations, we have made a comparison between the derived formulas and the (direct) straightforward computation, in terms of computational complexity and sensitivity to round-off errors.

Appendix

A. Recursive LCMV

Assume that, at time t, we want to incorporate the new sensor signal observation $\mathbf{y}[t]$, and we have access to the following variables from the previous time step $t-1$:

- $\mathbf{R}_{yy}^{-1}[t-1]$
- $\mathbf{\Gamma}[t-1] = \mathbf{R}_{yy}^{-1}[t-1]\mathbf{C}$
- $\mathbf{L}[t-1] = \left(\mathbf{C}^H\mathbf{\Gamma}[t-1]\right)^{-1}$.

Define

$$\boxed{\begin{array}{l}\bullet\ \mathbf{z}[t] \triangleq \mathbf{R}_{yy}^{-1}[t-1]\,\mathbf{y}[t] \\ \bullet\ \tilde{\mathbf{z}}[t] \triangleq \mathbf{C}^H\,\mathbf{z}[t] \\ \bullet\ \mathbf{x}[t] \triangleq \mathbf{L}[t-1]\,\tilde{\mathbf{z}}[t] \\ \bullet\ \eta[t] \triangleq \frac{\lambda^2}{1-\lambda} + \lambda \mathbf{y}[t]^H\,\mathbf{z}[t] \\ \bullet\ \xi[t] \triangleq \tilde{\mathbf{z}}[t]^H\,\mathbf{x}[t]\end{array}} \tag{99}$$

where λ denotes the forgetting factor as in (3). The recursive update of $\mathbf{R}_{yy}^{-1}$ given in (3) can then be written as

$$\boxed{\mathbf{R}_{yy}^{-1}[t] = \tfrac{1}{\lambda}\mathbf{R}_{yy}^{-1}[t-1] - \tfrac{1}{\eta[t]}\mathbf{z}[t]\mathbf{z}[t]^H.} \tag{100}$$

Using this with the fact that $\mathbf{\Gamma}[t] = \mathbf{R}_{yy}^{-1}[t]\mathbf{C}$, we find that

$$\boxed{\mathbf{\Gamma}[t] = \tfrac{1}{\lambda}\mathbf{\Gamma}[t-1] - \tfrac{1}{\eta[t]}\mathbf{z}[t]\tilde{\mathbf{z}}[t]^H.} \tag{101}$$

By left-multiplying both sides of (101) with $\mathbf{C}^H$, we find that

$$\mathbf{L}^{-1}[t] = \frac{1}{\lambda}\mathbf{L}^{-1}[t-1] - \frac{1}{\eta[t]}\tilde{\mathbf{z}}[t]\tilde{\mathbf{z}}[t]^H. \tag{102}$$

By applying the Woodbury identity, we find that

$$\boxed{\mathbf{L}[t] = \lambda\mathbf{L}[t-1] + \tfrac{\lambda^2}{\eta[t]-\lambda\xi[t]}\mathbf{x}[t]\mathbf{x}[t]^H.} \tag{103}$$

The new LCMV solution can then be computed as

$$\boxed{\mathbf{w}[t] = \mathbf{\Gamma}[t]\left(\mathbf{L}[t]\mathbf{f}\right).} \tag{104}$$

It is noted that the computational complexity of (100), (101), (103) and (104) is $O(K^2)$, $O(KQ)$, $O(Q^2)$ and $O(KQ)$, respectively. Therefore, and since $Q \leq K$, the overall complexity of the recursive algorithm is $O(K^2)$.

B. Recursive Computation of $\mathcal{D}\{\mathbf{T}\}$

Denote $\mathbf{T}[t] = \mathbf{\Gamma}[t]\mathbf{L}[t]\mathbf{\Gamma}[t]^H$ as the value of the matrix $\mathbf{T}$ at sample time t. Using (103), we obtain

$$\mathbf{T}[t] = \lambda\mathbf{\Gamma}[t]\mathbf{L}[t-1]\mathbf{\Gamma}[t]^H + \frac{\lambda^2}{\eta[t]-\lambda\xi[t]}\mathbf{\Gamma}[t]\mathbf{x}[t]\mathbf{x}[t]^H\mathbf{\Gamma}[t]^H. \tag{105}$$

We define

$$\boxed{\begin{array}{l}\bullet\ \mathbf{m}[t] \triangleq \mathbf{\Gamma}[t]\,\mathbf{x}[t] \\ \bullet\ \widetilde{\mathbf{m}}[t] \triangleq \mathbf{\Gamma}[t-1]\,\mathbf{x}[t]\end{array}.} \tag{106}$$

Substituting (101) in (105), and using the notation (106), we obtain

$$\begin{aligned}\mathbf{T}[t] = \lambda\mathbf{T}[t-1] &- \frac{1}{\eta[t]}\left(\mathbf{m}[t]\mathbf{z}[t]^H + \mathbf{z}[t]\mathbf{m}[t]^H\right) \\ &+ \frac{\lambda\xi[t]}{\eta[t]^2}\mathbf{z}[t]\mathbf{z}[t]^H + \frac{\lambda^2}{\eta[t]-\lambda\xi[t]}\widetilde{\mathbf{m}}[t]\widetilde{\mathbf{m}}[t]^H.\end{aligned} \tag{107}$$

Let $\tau_k[t]$ denote the k-th diagonal element of $\mathbf{T}[t]$, then it follows from (107) that

$$\begin{aligned}\tau_k[t] = \lambda\tau_k[t-1] &- \frac{2}{\eta[t]}\mathcal{R}\{m_k[t]z_k[t]^*\} \\ &+ \frac{\lambda\xi[t]}{\eta[t]^2}|z_k[t]|^2 + \frac{\lambda^2}{\eta[t]-\lambda\xi[t]}|\widetilde{m}_k[t]|^2\end{aligned} \tag{108}$$

where $\mathcal{R}\{.\}$ takes the real part of its argument. Note that computing $\tau_k[t]$ has $O(Q)$ complexity, since $m_k[t]$ and $\widetilde{m}_k[t]$ can be computed with $O(Q)$ complexity, and since all other variables are directly available from the recursive LCMV beamformer implementation based on (99). Therefore, the computation of all diagonal entries of $\mathbf{T}[t]$, i.e., the computation of $\mathcal{D}\{\mathbf{T}\}$, has $O(KQ)$ complexity.

REFERENCES

[1] 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.

[2] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the world with wireless sensor networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, 2001, pp. 2033–2036.

[3] S. Haykin and K. J. R. Liu, *Handbook on Array Processing and Sensor Networks*. Hoboken, NJ: Wiley-IEEE Press, 2010.

[4] A. Swami, Q. Zhao, Y.-W. Hong, and L. Tong, *Wireless Sensor Networks: Signal Processing and Communications*. West Sussex, U.K.: Wiley, 2007.

[5] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: A survey," *Comput. Netw.*, vol. 38, no. 4, pp. 393–422, 2002.

[6] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Comput. Netw.*, vol. 52, no. 12, pp. 2292–2330, 2008.

[7] Z. Quan, W. Kaiser, and A. H. Sayed, "Innovations diffusion: A spatial sampling scheme for distributed estimation and detection," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 738–751, Feb. 2009.

[8] J. Szurley, A. Bertrand, and M. Moonen, "Efficient computation of microphone utility in a wireless acoustic sensor network with multichannel Wiener filter based noise reduction," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Kyoto, Japan, Mar. 2012.

[9] B. Van Veen and K. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Mag.*, vol. 5, no. 2, pp. 4–24, Apr. 1988.

[10] S. Doclo and M. Moonen, "GSVD-based optimal filtering for single and multimicrophone speech enhancement," *IEEE Trans. Signal Process.*, vol. 50, no. 9, pp. 2230–2244, Sep. 2002.

[11] S. Doclo, A. Spriet, J. Wouters, and M. Moonen, "Frequency-domain criterion for the speech distortion weighted multichannel Wiener filter for robust noise reduction," *Speech Commun.*, vol. 49, no. 7–8, pp. 636–656, 2007.

[12] B. Cornelis, M. Moonen, and J. Wouters, "Performance analysis of multichannel Wiener filter based noise reduction in hearing aids under second order statistics estimation errors," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 5, pp. 1368–1381, Jul. 2011.

[13] J. Szurley, A. Bertrand, M. Moonen, P. Ruckebusch, and I. Moerman, "Utility based cross-layer collaboration for speech enhancement in wireless acoustic sensor networks," in *Proc. Eur. Signal Process. Conf. (EUSIPCO)*, Barcelona, Spain, Aug. 2011, pp. 235–239.

[14] S. M. Golan, S. Gannot, and I. Cohen, "Low-complexity addition or removal of sensors/constraints in LCMV beamformers," *IEEE Trans. Signal Process.*, vol. 60, no. 3, pp. 1205–1214, 2012.

[15] A. Bertrand, "Applications and trends in wireless acoustic sensor networks: A signal processing perspective," in *Proc. IEEE Symp. Commun. Veh. Technol. (SCVT)*, Nov. 2011.

[16] A. Bertrand and M. Moonen, "Robust distributed noise reduction in hearing aids with external acoustic sensor nodes," *EURASIP J. Adv. Signal Process.*, vol. 2009, no. Article ID 530435, pp. 14–14, 2009.

[17] S. Doclo, T. C. Lawin-Ore, and T. Rohdenburg, "Rate-constrained binaural MWF-based noise reduction algorithms," presented at the ITG Conf. Speech Commun., Bochum, Germany, Oct. 2010.

[18] T. C. Lawin-Ore and S. Doclo, "Analysis of rate constraints for MWF-based noise reduction in acoustic sensor networks," presented at the IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP), Prague, Czech Republic, May 2011.

[19] I. Himawan, I. McCowan, and S. Sridharan, "Clustered blind beamforming from ad-hoc microphone arrays," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 4, pp. 661–676, May 2011.

[20] A. Bertrand, "Signal processing algorithms for wireless acoustic sensor networks," Ph.D. dissertation, KU Leuven (Univ. of Leuven), Leuven, Belgium, 2011.
[21] S. Joshi and S. Boyd, "Sensor selection via convex optimization," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 451–462, Feb. 2009.
[22] H. Wang, G. Pottie, K. Yao, and D. Estrin, "Entropy-based sensor selection heuristic for target localization," in *Proc. 3rd Int. Symp. Inf. Process. Sensor Netw. (IPSN)*, Apr. 2004, pp. 36–45.
[23] M. Shamaiah, S. Banerjee, and H. Vikalo, "Greedy sensor selection: Leveraging submodularity," in *Proc. IEEE Conf. Decision Control (CDC)*, Dec. 2010, pp. 2572–2577.
[24] E. Msechu and G. Giannakis, "Sensor-centric data reduction for estimation with WSNs via censoring and quantization," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 400–414, Jan. 2012.
[25] C. H. Papadimitriou and K. Steiglitz*, Combinatorial Optimization: Algorithms and Complexity*. Englewood Cliffs, NJ: Prentice-Hall, 1982.
[26] J. Liu, J. Reich, and F. Zhao, "Collaborative in-network processing for target tracking," *EURASIP J. Adv. Signal Process. (Special Issues on Sensor Networks)*, vol. 2003, no. 4, pp. 378–391, 2003.
[27] K. Kumatani, J. McDonough, J. Lehman, and B. Raj, "Channel selection based on multichannel cross-correlation coefficients for distant speech recognition," in *Proc. Joint Workshop on Hands-free Speech Commun. Microphone Arrays (HSCMA)*, Jun. 30–1, 2011, pp. 1–6.
[28] S. Srinivasan, "Noise reduction in binaural hearing aids: Analyzing the benefit over monaural systems," *J. Acoust. Soc. of Amer.*, vol. 124, no. 6, pp. EL353–EL359, 2008.
[29] S. Srinivasan and A. C. den Brinker, "Rate-constrained beamforming in binaural hearing aids," *EURASIP J. Adv. Signal Process.*, vol. 2009, pp. 9–9, 2009, Article ID 257197.
[30] O. Roy and M. Vetterli, "Rate-constrained collaborative noise reduction for wireless hearing aids," *IEEE Trans. Signal Process.*, vol. 57, no. 2, pp. 645–657, 2009.
[31] A. Bertrand and M. Moonen, "Distributed node-specific LCMV beamforming in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 60, no. 1, pp. 233–246, 2012.
[32] 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, 2010.
[33] S. M. Golan, S. Gannot, and I. Cohen, "A reduced bandwidth binaural MVDR beamformer," presented at the Int. Workshop Acoust. Echo Noise Control (IWAENC), Tel-Aviv, Israel, Aug. 2010.
[34] G. H. Golub and C. F. van Loan*, Matrix Computations*, 3rd ed. Baltimore, MD: The Johns Hopkins Univ. Press, 1996.



Alexander Bertrand (M'08) received the M.Sc. degree and the Ph.D. degree in electrical engineering, both from KU Leuven, University of Leuven, Belgium, in 2007 and 2011, respectively.

He is currently a Postdoctoral Fellow of the Research Foundation—Flanders (FWO), affiliated with the Electrical Engineering Department (ESAT) of KU Leuven and the IBBT Future Health Department. In 2010, he was a Visiting Researcher at the Adaptive Systems Laboratory, University of California, Los Angeles (UCLA). His research interests are in multi-channel signal processing, ad hoc sensor arrays, wireless sensor networks, distributed signal enhancement, speech enhancement, and distributed estimation.

Dr. Bertrand received a Postdoctoral Fellowship from the Research Foundation—Flanders (FWO) from 2011 to 2014, a Ph.D. scholarship of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen) from 2008 to 2011, and an FWO grant for a Visiting Research Collaboration at UCLA in 2010. He served as a Technical Program Committee (TPC) Member for the European Signal Processing Conference (EUSIPCO) 2012.



Joseph Szurley (M'07) received the B.A. degree in physics from the University of California at Berkeley in 2005 and the M.Sc. degree in electrical engineering (with Hons.) from the California State University Long Beach in 2010. Since 2010 he has been working towards the Ph.D. degree at KU Leuven, Belgium.

In 2003, he worked at Lawrence Berkeley National Laboratory in the Atomic Molecular and Optical Sciences (AMSO) division. From 2004 to 2007, he was a Research Assistant for Takeda Pharmaceuticals. From 2009 to 2010, he worked at Conexant Systemsc. in the product engineering group. His research interests include wireless acoustic sensor networks, cross layer optimization algorithms, and distributed signal enhancement.



Peter Ruckebusch received the M.Sc. degree in computer science from University College Ghent, Ghent, Belgium, in 2008.

From 2008 to 2011, he worked at Ghent University as a scientific staff member of IBCN, a research group of the Information Technology Department INTEC. During this period, he collaborated on various national projects in the domain of wireless networks funded by the IBBT future Internet department. Since 2011, he has been working towards the Ph.D. degree at Ghent University, Ghent, Belgium. His research interests are mainly in the field of ad hoc wireless (sensor) networks with a focus on cross-layer optimization and modular network architectures.



Ingrid Moerman received the Electrical Engineering degree and the Ph.D. degree from the Ghent University, Ghent, Belgium, in 1987 and 1992, respectively.

In 2000, she became a part-time Professor at Ghent University. She is currently a staff member of the research group on Internet Based Communication Networks and Services (IBCN) (www.ibcn.intec.ugent.be), where she is leading the research on mobile and wireless communication networks. In 2006, she joined the Interdisciplinary institute for BroadBand Technology (IBBT), where she is currently coordinating several interdisciplinary research projects. Her main research interests include sensor networks, cooperative and cognitive networks, wireless access, self-organizing distributed networks and experimentally supported research. She is author or coauthor of more than 400 publications in international journals or conference proceedings.



Marc Moonen (M'94–SM'06–F'07) received the Electrical Engineering degree and the Ph.D. degree in applied sciences from KU Leuven, Belgium, in 1986 and 1990, respectively.

Since 2004, he has been a Full Professor at the Electrical Engineering Department of KU Leuven, where he is currently heading a research team working in the area of numerical algorithms and signal processing for digital communications, wireless communications, DSL, and audio signal processing.

Dr. Moonen received the 1994 KU Leuven Research Council Award, the 1997 Alcatel Bell (Belgium) Award (with P. Vandaele), and the 2004 Alcatel Bell (Belgium) Award (with R. Cendrillon), and was a 1997 Laureate of the Belgium Royal Academy of Science. He received a journal Best Paper Award from the IEEE TRANSACTIONS ON SIGNAL PROCESSING (with G. Leus) and from Elsevier *Signal Processing* (with S. Doclo). He was chairman of the IEEE Benelux Signal Processing Chapter from 1998 to 2002 and a member of the IEEE Signal Processing Society Technical Committee on Signal Processing for Communications, and is currently President of the European Association for Signal Processing (EURASIP). He has served as Editor-in-Chief for the *EURASIP Journal on Applied Signal Processing* from 2003 to 2005, and has been a member of the Editorial Board of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II, the *IEEE Signal Processing Magazine*, *Integration—The VLSI Journal*, *EURASIP Journal on Wireless Communications and Networking*, and *Signal Processing*. He is currently a member of the Editorial Board of the *EURASIP Journal on Applied Signal Processing* and Area Editor for Feature Articles in the *IEEE Signal Processing Magazine*.