ADAPTIVE BEAMFORMING VIA TWO-DIMENSIONAL COSINE TRANSFORM

Jiajin An and Benoit Champagne

INRS-Télécommunications, Université du Québec, 16 Place du Commerce
Verdun, Québec, Canada H3E 1H6

Abstract

In this paper, we present a new transform domain least-mean-square (LMS) algorithm for weight adaptation in the Griffiths-Jim's generalized sidelobe canceller (GSC). In this algorithm we consider the set of tap-input vectors at the output of the blocking matrix as a two-dimensional array and map the whole array to transform domain via a full two-dimensional orthogonal transform. A self-orthogonalizing LMS algorithm is then used for weight adaptation in the transform domain. The purpose of the 2-dimensional orthogonal transform is to remove both the autocorrelation of the signal within individual channels and the crosscorrelation between adjacent channels. As a result the proposed algorithm exhibits faster convergence rate than previous algorithms and is therefore well suited to real-time processing of non-stationary array signals.

I. Introduction

The purpose of an adaptive beamformer is to adjust the weights of a linear array processor in real time to respond to a signal coming from a desired direction while discriminating against noises from other directions.

Early work on adaptive beamforming was done by Frost [1]. His method, called "Constrained LMS" algorithm, is a simple constrained stochastic gradient-descent least-mean-square (LMS) algorithm with self-correcting capability. The algorithm attempts to minimize noise power at the array output while using a set of linear constraints to maintain a chosen frequency response in the direction of interest (look-direction). The generalized sidelobe canceller (GSC), due to Griffiths and Jim [2], represents an alternative formulation of the constrained LMS problem. Essentially, the GSC is a mechanism for transforming a constrained minimization problem into an unconstrained one. In [2], it is shown that the GSC is identical to Frost's algorithm under certain conditions. In both Frost's algorithm and the GSC, a vector of time-domain tap inputs is used to update the weights at each iteration. As a result, these algorithms suffer from a major drawback which is common to most time-domain implementations of LMS type adaptive algorithms, that is, their convergence rate decreases as the condition number (i.e., the ratio of maximum to minimum eigenvalues) of the input autocorrelation matrix increases.

Support for this work was provided by NSERC operating grant OGP0105533.

Recently, Chen and Fang [3] have applied the self-orthogonalizing frequency-domain LMS algorithm [4], called FLMS, to Griffiths-Jim's GSC in order to accelerate its convergence rate for real-time adaptive processing of array signals. Computer simulations illustrate that their algorithm exhibits much faster convergence rate and better performance of nulling jammers than that of Griffiths-Jim's GSC. However, their approach does not exploit the full benefit of the transform domain LMS algorithm. Indeed, each tapped-delay line at the output of the blocking matrix is transformed individually with a one-dimensional DFT. This corresponds to diagonalizing only the principal submatrices of the full data autocorrelation matrix. As a result the crosscorrelation between adjacent channels is not removed and the resulting matrix controlling the adaptation process is not fully diagonal.

In this paper, we make further improvement on Chen-Fang's FLMS beamforming algorithm by exploiting the correlation which exists between adjacent channels at the output of the blocking matrix in the GSC. In the proposed algorithm, we view the set of tap inputs along these parallel channels as a two-dimensional array and map the whole array to a transform domain via a full two-dimensional orthogonal transform. Following this transformation, a self-orthogonalizing LMS algorithm is used for weight adaptation.

II. Chen-Fang's FLMS algorithm

The FLMS adaptive beamforming algorithm proposed by Chen and Fang is shown in Fig. 1. It consists of an array of K sensors attached to delay elements, which are used to steer the array in the desired look direction.

The upper part is a conventional beamformer followed by a fixed target signal filter, the purpose of which is to control the frequency response of the beamformer in the look-direction. The lower part is the sidelobe canceller. It consists of $K-1$ subtractors followed by a set of $K-1$ tapped-delay lines (TDL), each with $L-1$ tap-delays. The time-domain tap-input vector $X_i(n)$ of the ith TDL is transformed into the frequency-domain vector $U_i(n)$ by means of the L-point discrete Fourier transform (DFT)

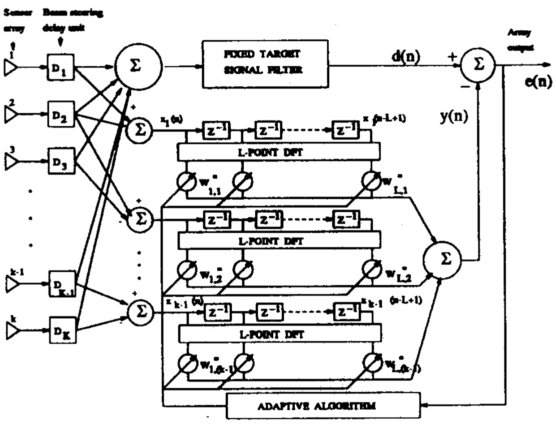

Fig. 1. Frequency-domain GSC.

matrix $\mathbf{D}$,

$$U_i(n) = \mathbf{D}X_i(n) \tag{2.1}$$

where

$$X_i(n) \triangleq [\, x_i(n) \quad x_i(n-1) \quad \ldots \quad x_i(n-L+1)\,]^T$$

$$U_i(n) \triangleq [\, u_{0,i}(n) \quad u_{1,i}(n) \quad \ldots \quad u_{(L-1),i}\,]^T .$$

The complete frequency-domain $(K-1)L \times 1$ tap-input vector and the complete frequency-domain weight vector of all the TDLs at the nth iteration are defined as, respectively,

$$U(n) \triangleq [\, U_1^T(n) \quad U_2^T(n) \quad \ldots \quad U_{K-1}^T(n)\,]^T ,$$

$$W(n) \triangleq [\, W_1^T(n) \quad W_2^T(n) \quad \ldots \quad W_{K-1}^T(n)\,]^T ,$$

where

$$W_i \triangleq [\, w_{1i}(n) \quad w_{2i}(n) \quad \ldots \quad w_{Li}(n)\,]^T .$$

From (2.1), it is clear that

$$U(n) = \begin{bmatrix} \mathbf{D} & & & \\ & \mathbf{D} & & \\ & & \ddots & \\ & & & \mathbf{D} \end{bmatrix} X(n). \tag{2.3}$$

Finally, the output of the complete frequency-domain filter is

$$y(n) = W^H(n)U(n). \tag{2.2}$$

The weight vector $W(n)$ is continually updated so as to minimize the power of the error $e(n) = d(n) - y(n)$ between the output of the conventional beamformer, $d(n)$, and that of the sidelobe cancelling network, $y(n)$. To this end, a self-orthogonalizing adaptive LMS algorithm with accelerated convergence rate is used [4]. In the ideal form of this modified LMS algorithm, the gradient estimate is premultiplied by the inverse of the correlation matrix of the complete frequency-domain tap-input vector $U(n)$, so that the weights are updated as

$$W(n+1) = W(n) + 2\gamma \mathbf{R}_{UU}^{-1} U(n) e^*(n). \tag{2.4}$$

In practice, the inverse of the correlation matrix $\mathbf{R}_{UU}$ is unknown and must be estimated from the data. In the FLMS beamforming algorithm, this matrix is estimated as follows:

- Step 1: Assume $\mathbf{R}_{UU}$ is approximately diagonal so that $\mathbf{R}_{UU}$ can be estimated by a diagonal matrix, i.e.,

$$\mathbf{R}_{UU} \approx diag(r_{01} \ldots r_{(L-1)1} \cdots r_{0(K-1)} \cdots r_{(L-1)(K-1)}) \tag{2.5}$$

  where $r_{lk} \triangleq E[u_{lk}(n)u_{lk}^*(n)]$ $(0 \le l \le L-1, 1 \le k \le K-1)$ is the power of the lth frequency component of the kth TDL. Since $\mathbf{R}_{UU}$ is assumed to be diagonal, the computation of its inverse is straightforward.

- Step 2: Estimate the value of r_{lk} recursively as

$$\hat{r}_{lk}(n) = \beta \hat{r}_{lk}(n-1) + (1-\beta) u_{lk}(n) u_{lk}^*(n) \tag{2.6}$$

  where β is a smoothing constant.

- Step 3: The desired estimate of the inverse correlation matrix $\mathbf{R}_{UU}^{-1}$ is given by

$$\hat{\mathbf{R}}_{UU}^{-1} = diag(1/\hat{r}_{01}, \ldots, 1/\hat{r}_{(L-1)(K-1)}).$$

III. The new 2D-DCT-LMS algorithm

The original purpose of Chen-Fang's FLMS algorithm is to accelerate the convergence rate of the conventional GSC by first removing the correlation between the tap-inputs via an orthogonal transform and then using the self-orthogonalizing LMS algorithm in the transform domain. We note, however, that contrarily to the assumption in (2.5), the frequency-domain correlation matrix $\mathbf{R}_{UU}$ is not diagonal in general. Indeed, even if the optimal Karhunen-Loeve transform (KLT) is used in (2.1) instead of the DFT, $\mathbf{R}_{UU}$ will not be diagonal because of cross-correlation between adjacent TDLs in Fig. 1. This is expalined below.

Suppose we use the L-point KLT transform in (2.1). Then, we have

$$U(n) = \begin{bmatrix} \mathbf{L}_1 & & & \\ & \mathbf{L}_2 & & \\ & & \ddots & \\ & & & \mathbf{L}_{K-1} \end{bmatrix} X(n) \tag{3.1}$$
$$= [\, U_1^T(n) \quad U_2^T(n) \quad \ldots \quad U_{K-1}^T(n)\,]^T$$

where $\mathbf{L}_i$ is KLT transform matrix of the ith TDL, and

$$U_i(n) = \mathbf{L}_i X_i(n).$$

Now,

$$\mathbf{R}_{UU} \triangleq E[U(n)U^H(n)] = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} & \dots & \mathbf{R}_{1(K-1)} \\ \mathbf{R}_{21} & \mathbf{R}_{22} & \dots & \mathbf{R}_{2(K-1)} \\ & & \dots & \\ \mathbf{R}_{(K-1)1} & \mathbf{R}_{(K-1)2} & \dots & \mathbf{R}_{(K-1)(K-1)} \end{bmatrix} \tag{3.2}$$

where

$$\mathbf{R}_{ij} \triangleq E[U_i(n)U_j^H(n)] = \mathbf{L}_i E[X_i(n)X_j^H(n)]\mathbf{L}_j^H$$

is the cross-correlation matrix of $U_i(n)$ and $U_j(n)$ (in the transform-domain).

When $i = j$, $\mathbf{R}_{ij}$ in (3.2) is the autocorrelation matrix of the ith transform domain tap-input vector. By definition of the KLT transform, this matrix is diagonal. However, when $i \neq j$, there is no reason to believe that $\mathbf{R}_{ij}$ is a zero-matrix. Indeed, it is not difficult to imagine situations where there exists a strong correlation between the transform domain vectors $U_i(n)$ and $U_j(n)$. As a result, it follows that $\mathbf{R}_{UU}$ is generally not a diagonal matrix. Without the full diagonality of $\mathbf{R}_{UU}$ in (2.4), the advantages of the FLMS algorithm appear to be limited. In fact, it should be possible to exploit the cross-correlation between adjacent TDLs to further improve the convergence rate of the FLMS algorithm.

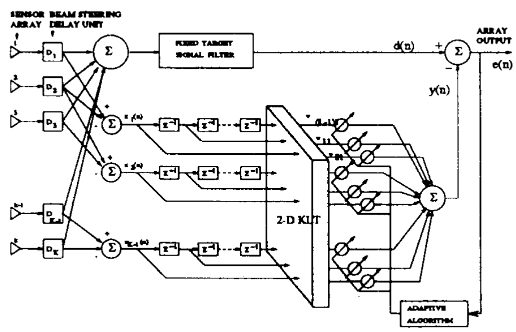

Fig. 2. 2-D KLT transform GSC.

In the single channel case, the purpose of the self-orthogonalizing transform-domain LMS algorithm is to remove the temporal correlation of the TDL tap inputs in order to accelerate the convergence rate. In this respect, the KLT transform is optimal in the sense that the transform coefficients are completely uncorrelated. In the multi-channel case under consideration here, i.e. beamforming, there exist both temporal and spatial correlation between the tap-inputs on the various TDLs. In this case, the optimal transform should remove both types of correlation. In principle, this can be achieved by applying a full orthogonal transform which diagonalizes all the tap inputs simultaneously. Instead of transforming each TDL individually as in the FLMS algorithm, we therefore propose the following approach: consider the set of tap-input vectors at the output of the blocking matrix in Fig. 1 as a two-dimensional array and map the whole array to the transform domain via a full two-dimensional KLT transform. A block diagram of the proposed approach, called 2-D KLT transform GSC, is shown in Fig. 2.

The complete set of tap-input vectors at the output of the subtraction network, represented by the $L \times (K-1)$ matrix

$$\mathbf{X}(n) \triangleq [X_1(n) \quad X_2(n) \quad \dots \quad X_{K-1}(n)],$$

is first transformed into another $L \times (K-1)$ matrix $\mathbf{V}(n)$ via a full 2-D KLT transform:

$$\mathbf{V}(n) \triangleq [V_1(n) \quad V_2(n) \quad \dots \quad V_{K-1}(n)] \triangleq \begin{bmatrix} v_{01}(n) & v_{02}(n) & \dots & v_{0(K-1)}(n) \\ v_{11}(n) & v_{12}(n) & \dots & v_{1(K-1)}(n) \\ & & \dots & \\ v_{(L-1)1}(n) & v_{(L-1)2}(n) & \dots & v_{(L-1)(K-1)}(n) \end{bmatrix} = \mathbf{L}_2[\mathbf{X}(n)] \tag{3.3}$$

where the operator $\mathbf{L}_2$ represents the 2-D KLT transform (this operator can be thought of as a 4-dimensional matrix). Now, all the transform coefficients are uncorrelated, that is,

$$E[v_{ij}(n)v_{kl}^*(n)] = \lambda_{ij}\delta_{ik}\delta_{jl}. \tag{3.4}$$

The output of the sidelobe cancelling network, $y(n)$, is obtained as a weighted sum of the elements of the transform-domain matrix $\mathbf{V}(n)$. The complete transform-domain weight vector of the adaptive filter at the nth iteration is represented by

$$Z(n) \triangleq [Z_1^T(n) \quad Z_2^T(n) \quad \dots \quad Z_{K-1}^T(n)]^T$$

where

$$Z_i(n) \triangleq [z_{1i}(n) \quad z_{2i}(n) \quad \dots \quad z_{Li}(n)]^T$$

is the weight vector associated with $V_i(n)$. Then, the output $y(n)$ of the cancelling network is given by

$$y(n) = Z^H(n)V(n) \tag{3.5}$$

where

$$V(n) \triangleq [V_1^T(n) \quad V_2^T(n) \quad \dots \quad V_{K-1}^T(n)]^T$$

is the complete transform-domain tap-input vector.

Now, if the self-orthogonalizing LMS algorithm is used to update the weight vector $Z(n)$, we have:

$$Z(n+1) = Z(n) + 2\gamma\mathbf{R}_{VV}^{-1}V(n)e^*(n) \tag{3.6}$$

where $\mathbf{R}_{VV}$ is the autocorrelation matrix of the vector $V(n)$, i.e.

$$\mathbf{R}_{VV} = E[V(n)V^H(n)]$$

which is a $L(K-1) \times L(K-1)$ matrix. From (3.4), it follows that $\mathbf{R}_{VV}$ is diagonal, i.e.

$$\mathbf{R}_{VV} = diag\left[\lambda_{01}, ..., \lambda_{(L-1)1}, ..., \lambda_{0(K-1)}, ..., \lambda_{(L-1)(K-1)}\right]. \quad (3.7)$$

As a result, the calculation of the inverse matrix $\mathbf{R}_{VV}^{-1}$ in (3.6) is straightforward.

We note that the KLT transform is signal-dependent and consequently, its implementation in real time applications poses practical problems. In such cases, it is preferable to use suboptimal transforms such as the 2-dimensional discrete cosine or Fourier transforms (2D-DCT and 2D-DFT, respectively), which are not signal-dependent and for which fast algorithms are available. When such transforms are used, $\mathbf{R}_{VV}$ is only approximately diagonal. However, as our experience indicates, the advantages offered by the 2D-KLT (e.g., improve convergence rate) are preserved.

Finally, as with the FLMS algorithm, only an estimate of $\mathbf{R}_{VV}^{-1}$ can be used in (3.6). Such an estimate can be obtained by following Steps 2 and 3 in Section II.

IV. Simulation results

To compare the performance of the proposed 2D-DCT-LMS algorithm with that of Chen-Fang's FLMS, the same simulated conditions as in [3] were used. These conditions are briefly described below.

The target signal and the three interfering jammers are narrow-band plane waves impinging upon a linear, uniform sensor array from different directions. These signals are monitored in the presence of a white background noise process with variance $\sigma_w^2 = 0.1$. The fixed target signal filter is all-pass with a gain of $1/K$. Additional specifications are given below, where f_i and θ_i denote the normalized frequency and incident angle (relative to broadside) of the plane wave signals, respectively:

- Array: number of sensors $K = 17$, tap length $L = 8$.
- Target signal: $f_0 = 0.1$, $\theta_0 = 0°$, SNR = 10 dB.
- Jammer 1: $f_1 = 0.3$, $\theta_1 = 34°$, JNR1 = 20 dB.
- Jammer 2: $f_2 = 0.4$, $\theta_2 = -49°$, JNR2 = 40 dB.
- Jammer 3: $f_3 = 0.25$, $\theta_3 = -24°$, JNR3 = 30 dB.

Synthetic signals with the above specifications were generated and processed with the 2D-DCT-LMS and the FLMS algorithms. The step size parameter γ governing the weight adaptation of the FLMS algorithm in (2.4) was set to $\gamma = 1.2 \times 10^{-3}$ and that of the 2D-DCT-LMS in (3.6) was set to $\gamma = 1.116 \times 10^{-3}$. With these values, the same steady-state mean-square-error (MSE) is obtained at the output of both adaptive beamformer.

The output waveforms of the two algorithms are shown in Fig.3. The average learning curves of the GSC using the FLMS and the new 2D-DCT-LMS algorithms are shown in Fig.4. From these results, it is apparent that the 2D-DCT-LMS algorithm converges faster than the FLMS algorithm.

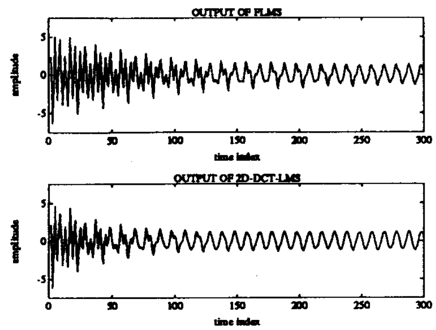

Fig. 3. Comparison of the waveforms.

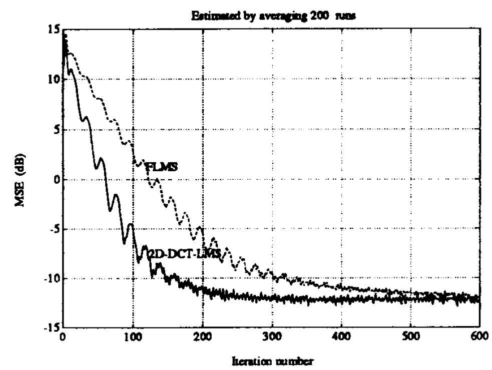

Fig. 4. Comparison of learning curves.

References

[1] O. L. Frost, III, "An algorithm for linearly-constrained adaptive array processing", *Proc. IEEE*, vol. 60, no. 8, pp. 926-935, Aug. 1972.

[2] L. J. Griffiths and C. W. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Trans. Antennas And Propag.*, vol. AP-30, pp. 27-34, Jan. 1982.

[3] Y. H. Chen and H. D. Fang, "Frequency-domain implementation of Griffiths-Jim adaptive beamformer," *J. Acoust. Soc. Am.*, vol. 91, no. 6, pp. 3354-3366, June 1992.

[4] R. D. Gitlin and F. R. Magee, Jr., "Self-orthogonalizing adaptive equalization algorithms," *IEEE Trans. Commun.*, vol. COM-25, pp. 666-672, July 1977.