An Alternative Approach to Linearly Constrained Adaptive Beamforming

LLOYD J. GRIFFITHS, SENIOR MEMBER, IEEE, AND CHARLES W. JIM

Abstract—**A beamforming structure is presented which can be used to implement a wide variety of linearly constrained adaptive array processors. The structure is designed for use with arrays which have been time-delay steered such that the desired signal of interest appears approximately in phase at the steered outputs. One major advantage of the new structure is the constraints can be implemented using simple hardware differencing amplifiers. The structure is shown to incorporate algorithms which have been suggested previously for use in adaptive beamforming as well as to include new approaches. It is also particularly useful for studying the effects of steering errors on array performance. Numerical examples illustrating the performance of the structure are presented.**

INTRODUCTION

THIS PAPER describes a simple time-varying beamformer which can be used to combine the outputs of an array of sensors. The beamformer is constrained to filter the "desired" signal with a filter having a prescribed gain and phase response. The "desired" signal is identified by time-delay steering the sensor outputs so that any signal incident on the array from the direction of interest appears as an identical replica at the outputs of the steering delays. All other signals received by the array which do not have this property are considered to be noise and/or interference. The purpose of the beamformer is to minimize the effects of noise and interference at the array output while simultaneously maintaining the prescribed frequency response in the direction of the desired signal.

Beamformers of this type are termed linearly constrained array processors and have been studied by several authors including Levin [1], Lacoss [2], Kobayashi [3], Booker and Ong [4], Frost [5], and Applebaum and Chapman [6]. The last five of these authors describe iterative or continuously adaptive beamformers in which the beamforming coefficients adjust to new values as each new set of samples of array sensor outputs are received. Adaptive methods are of particular interest in those problems in which the interference properties are either spatially or temporarily time varying.

The purpose of this paper is to present the linearly constrained adaptive algorithm, due to Frost [5], using an alternative beamforming model. This presentation illustrates the fundamental properties of the algorithm in an exceedingly simple fashion. It also allows for generalizations not available with Frost's method. The basic structure of the beamforming model has been suggested by Applebaum and Chapman [6]. In this paper we describe the structure in detail and give exact algorithm comparisons for a variety of linearly constrained beamformers. The structure is shown to be a direct consequence of Frost's method. One major advantage of our approach is an assessment of the performance degradation caused by the steering and/or gain errors in the array sensors. In most practical situations the theoretically ideal requirement of an "identical replica" of the desired signal, at the output of each steering delay, is seldom met. The effects of these errors on overall beamformer performance is easily modeled using our approach. For example, it is shown that these effects are particularly detrimental under conditions of high signal-to-noise ratio (SNR).

A second reason for this presentation is to enumerate certain difficulties which may arise with the use of constrained adaptive array processors which do not incorporate Frost's error-correction feature. Of the papers referenced above, four (see [2]–[4] and [7]) use an algorithm based on the gradient projection approach [8]. (Levin's approach was nonadaptive and utilized matrix inversion techniques.)

In this paper we first review Frost's algorithm which is not susceptible to roundoff error and requires relatively few additional computations per adaptive cycle. A simple geometric interpretation illustrating the effects of roundoff errors on his algorithm and on gradient projection is presented. The error-correcting properties of the approach are identified using this illustration.

We then show that the algorithm can be interpreted using a new beamforming model, termed the adaptive sidelobe canceling beamformer. This structure illustrates the constraint features of the algorithm and shows how additional constraints can be added. The error-correcting features are also elucidated. Sidelobe canceling is shown to be closely related to the method of adaptive noise canceling described by Widrow *et al.* [9]. As a consequence results derived in adaptive noise canceling can be applied directly to the linearly constrained adaptive beamformer.

LINEARLY CONSTRAINED ADAPTIVE BEAMFORMING

We denote the sampled output of the mth time-delayed sensor by $x_m(k)$. A total of M sensors are assumed to be present in the assumptions of ideal steering:

$$x_m(k) = s(k) + n_m(k). \tag{1}$$

In this expression $s(k)$ is the desired signal and $n_m(k)$ represents the totality of noise and interference observed at the output of the mth steered sensor. A beamformed output signal $y(k)$ is formed as the sum of delayed and weighted $x_m(k)$. Specifically, if $a_{m,l}$ is used to represent the weight used for the mth channel at delay l, then

$$y(k) = \sum_{m=1}^{M} \sum_{l=-K}^{K} a_{m,l} x_m(k-l). \tag{2}$$

Manuscript received May 19, 1980; revised March 5, 1981. This work was supported in part by the Office of Naval Research, Washington, DC, under Contract N00014-77-C-0592 and by the Electronics System Division (AFSC), Hanscom AFB, MA under Subcontract 14029 with SRI International, Menlo Park, CA.

L. J. Griffiths and C. W. Jim are with the Department of Electrical Engineering, University of Colorado, Boulder, CO 80309.

Note that a total of $2K + 1$ samples are used from each channel and that the zero time reference is at the filter midpoint.

Matrix notation can be used to simplify this notation. We let $\mathbf{A}_l$ and $\mathbf{X}(k-l)$ represent the filter coefficient and signal vectors at the lth delay point, i.e.,

$$\mathbf{A}_l^T = [a_{1,l}, a_{2,l} \cdots a_{M,l}] \tag{3}$$

$$X^T(k-l) = [x_1(k-l), x_2(k-l), \cdots, x_M(k-l)] \tag{4}$$

where superscript T denotes transpose. The output signal of (2) then becomes

$$y(k) = \sum_{l=-K}^{K} \mathbf{A}^T(l)\mathbf{X}(k-l). \tag{5}$$

Under the ideal steering assumption in (1), the signal vector $x(k-l)$ becomes

$$\mathbf{X}(k-l) = s(k-l)\mathbf{1} + \mathbf{N}(k-l) \tag{6}$$

where $\mathbf{1}$ is a column vector of M ones and $\mathbf{N}(k-l)$ is a vector of noise and interference defined in a manner analogous to (4).

Prescribed gain and phase response for the desired signal is ensured by constraining the sums of channel weights at each delay point to be specific values. Thus if $f(l)$ is used to denote the sum for the set of weights at delay l then

$$\mathbf{A}^T(l)\mathbf{1} = f(l). \tag{7}$$

Under this constraint the portion of the output due to desired signal reduces to

$$Y_s(k) = \sum_{l=-K}^{K} f(l)s(k-l). \tag{8}$$

Thus the $f(l)$ represent the impulse response of a finite-duration impulse-response (FIR) filter having length $2K + 1$. One commonly used constraint is that of zero distortion in which $f(l) = \delta(l)$, where $\delta(l)$ is the discrete impulse function. The FIR filter constraint function is normalized such that

$$\mathbf{F}^T\mathbf{1} = 1, \tag{9a}$$

$$\mathbf{F}^T = [f(-k), \cdots, f(k)]. \tag{9b}$$

The objective of linearly constrained adaptive beamforming is then to find filter coefficients $\mathbf{A}(l)$ which satisfy (7) and simultaneously reduce the average value of the square of the output noise component. This is equivalent to finding those coefficients which result in minimum output noise power subject to the constraint of the prescribed desired signal filtering.

In adaptive beamforming the filter coefficients are time varying and change as each new set of samples of sensor outputs is received. Thus if $\mathbf{A}_l(k)$ is used to denote the values at time k the values at the next sampling instant $k + 1$ are computed as

$$\mathbf{A}_l(k+1) = \mathbf{A}_l(k) + \Delta_l(k) \tag{10}$$

this paper we are concerned with Frost's procedure [5], in which

$$\Delta_l(k) = \mu y(k)[q_x(k-l)\mathbf{1} - \mathbf{X}(k-l)] - q_{a,l}(k)\mathbf{1} + \frac{1}{M}f(l)\mathbf{1} \tag{11}$$

and

$$q_x(k-l) = \frac{1}{M}\mathbf{X}^T(k-l)\mathbf{1} \tag{12}$$

$$q_{a,l}(k) = \frac{1}{M}\mathbf{A}_l^T(k)\mathbf{1}. \tag{13}$$

The adaptive step size μ is a scalar which controls both the convergence rate and steady-state noise behavior of the algorithm [9] and is normalized by the total power contained in the beamformer. Thus

$$\mu = \frac{\alpha}{P(k)} \tag{14}$$

$$P(k) = \sum_{m=1}^{M} \sum_{l=-K}^{K} x_m^2(k-l). \tag{15}$$

Convergence of either algorithm is assured if $0 < \alpha < 1$. Other power estimates involving time averaging may be employed without significantly affecting performance.

Frost's procedure differs from that used in gradient projection [7] by the addition of the last two terms in (11). These terms involve a total number of additional $(2K + 1)M$ adds and $2K + 1$ multiples. They are necessary, however, in that they prevent the accumulation of computational errors which may occur on any iteration of the algorithm.

Error Effects in Linearly Constrained Beamforming

The effects of errors may be illustrated by examining the constraints (7) for the adaptive algorithm in (10) and (11). We assume that in the algorithm implementation, the computation of the signal sum $q_x(k-l)$ and the weight sum $a_{a,l}(k)$ in (13) introduced the following errors:

$$q_x(k-l) = \frac{1}{M}\mathbf{X}^T(k-l)\mathbf{1} + \epsilon_x(k) \tag{16a}$$

$$q_{a,l}(k) = \frac{1}{M}[\mathbf{A}_l^T(k)\mathbf{1} + \epsilon_{\mathbf{A}}(k)] \tag{16b}$$

or equivalently, the current weight vector $A_l(k)$ is presumed to be slightly off the constraint, i.e.,

$$\mathbf{A}_l^T(k)\mathbf{1} = f(l) + \epsilon_A(k). \tag{16c}$$

The degree to which the next weight vector fails to meet the constraint can then be computed by solving for $A_l^T(k+1)\mathbf{1}$ in (10) and (11). Thus, using (16),

$$\mathbf{A}_l^T(k+1)\mathbf{1} = f(l) + \epsilon_A(k) + \mu M y(k)\epsilon_x(k)$$

The terms enclosed in $\{\cdot\}$ are produced by error correction position of Frost's algorithm while the first three are due to the gradient projection operator. Thus if a gradient projection adaptation algorithm is employed—as was the case in [2]–[4] and [7]—the constraint error at step $k + 1$ is

$$\epsilon_A(k+1) = \epsilon_A(k) + \mu M y(k) \epsilon_x(k) \quad (18)$$

and with Frost's procedure

$$\epsilon_A(k+1) = \mu M y(k) \epsilon_x(k). \quad (19)$$

The cumulative error effects of gradient projection observed by Shen [7] are due to the first-order difference relationship in (18). If we assume that the driving term $\mu M_y(k) \epsilon_x(k)$ can be modeled as a zero-mean white random process with variance σ_ϵ^2, and that $\epsilon_A(0) = 0$, then the gradient projection constraint error (18) is a Brownian motion [10] or random walk process. Although the mean of the error remains zero, its variance $\sigma_A{}^2(k)$ grows linearly with the number of steps, i.e.,

$$\sigma_A{}^2(k) = k\sigma_\epsilon{}^2 \quad (20a)$$

for gradient projection. With the correction terms, however, the error at each step has constant variance at each iteration,

$$\sigma_A{}^2(k) = \sigma_\epsilon{}^2. \quad (20b)$$

A simple geometric interpretation [5] can also be given for these effects. Consider the geometry associated with the gradient projection algorithm shown in Fig. 1. Coefficient vectors meeting the desired constraint must lie on the planar subspace C defined by the vector $\mathbf{F}$(9b). It is assumed that the coefficient vector $\mathbf{A}_l(k)$ at time k is too long and that the gradient vector produced by the data is $\mathbf{g}_l(k)$ given by

$$\mathbf{g}_l(k) = \mu y(k)\mathbf{X}(k-l). \quad (21)$$

In the gradient projection method the new coefficient vector $\mathbf{A}_l(k)$ is obtained by finding the projection of $g_l(k)$ in the direction of the plane C, and then by adding this projection to the previous vector. As shown by Fig. 1 the resulting new coefficient vector will not lie on the constraint plane, even with an error-free projection operation.

Fig. 2 illustrates the geometry for Frost's approach. In this case the new coefficient vector is found by projecting the sum of the former vector and the gradient in the direction of the constraint plane C. The new coefficient vector $\mathbf{A}_l(k)$ is then the sum of this projected vector and the vector $\mathbf{F}$, which defines C. As shown in the diagram the new coefficients will lie on the constraint plane regardless of the previous error provided that the projection operation is error free. The net error induced by this method is then restricted to the machine quantization error of a single projection operation and accumulation does not occur.

GENERALIZED SIDELOBE CANCELING MODEL

The linearly constrained adaptive algorithm defined by (10)–(13) may be implemented using the structure shown in Fig. 3. Time-delay steering elements $\tau_1, \tau_2, \cdots, \tau_M$ are used to

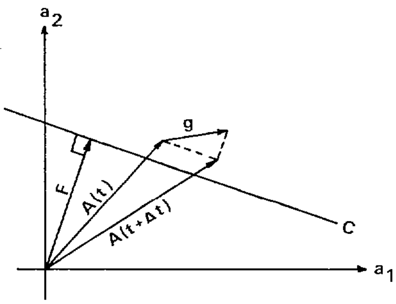

Fig. 1. Geometrical interpretation for gradient projection adaptive algorithm.

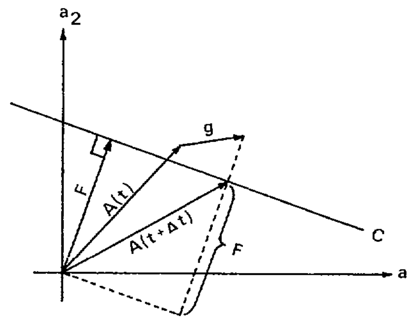

Fig. 2. Geometrical interpretation for linearly constrained error-correcting adaptive algorithm.

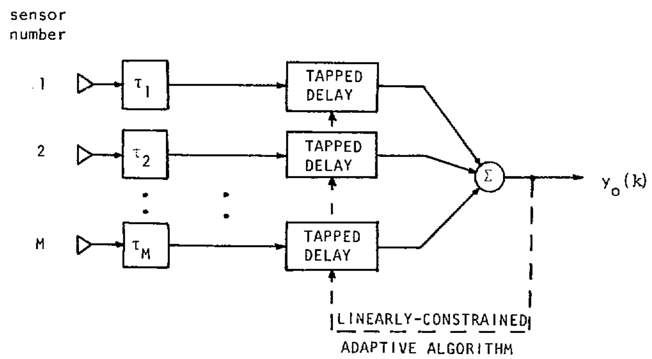

Fig. 3. Direct form implementation of linearly constrained adaptive array processing algorithm.

the beamformer is updated by the adaptive processor, which computes new values using the algorithm. An alternative implementation which achieves precisely the same overall processor can be derived in a simple manner directly from this algorithm. The resulting structure is termed the generalized sidelobe canceling form and is depicted in Fig. 4.

This processor consists of two distinct substructures which are shown as the upper and lower processing paths. The upper or conventional beamformer path consists of a set of fixed amplitude weights $w_{c1}, w_{c2}, \cdots, w_{cM}$ which produce non-adaptive-beamformed signal $y_c(k)$,

$$y_c(k) = \mathbf{W}_c{}^T\mathbf{X}(k) \quad (22)$$

where

$$\mathbf{W}_c{}^T = [w_{c1}, w_{c2}, \cdots, w_{cM}]. \quad (23)$$

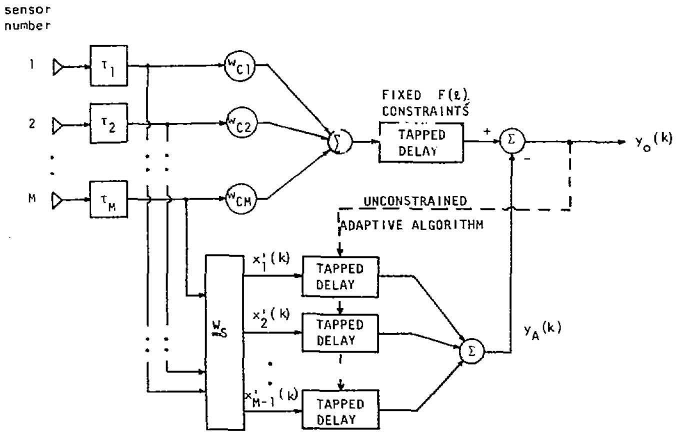

Fig. 4. Generalized sidelobe canceling form of linearly constrained adaptive array processing algorithm.

fixed nonadaptive coefficients. In typical applications the weights $\mathbf{W}_c$ are chosen so as to trade off the relationship between array beamwidth and average sidelobe level [11]. (One widely used method employs Chebyshev polynomials to find the $\mathbf{W}_c$.) For the purpose of this paper, however, any method can be used to choose the weights as the performance of the overall beamformer will be characterized in terms of the specific values chosen. (All w_{ci} are assumed nonzero.) In order to simplify notation the coefficients in $\mathbf{W}_c$ are normalized to have a sum of unity. That is

$$\mathbf{W}_c{}^T\mathbf{1} = 1. \tag{24}$$

The signal $y_c'(k)$ is obtained by filtering $y_c(k)$ and the FIR operator containing the constraint values $f(l)$,

$$y_c'(k) = \sum_{l=-K}^{K} f(l) y_c(k-l). \tag{25}$$

The lower path in Fig. 4 is the sidelobe canceling path. It consists of a matrix preprocessor $\overline{W}_s$ followed by a set of tapped-delay lines, each containing $2K + 1$ weights. The purpose of $\overline{W}_s$ is to block the desired signal $s(k)$ from the lower path. Since $s(k)$ is common to each of the steered sensor outputs (1) blocking is ensured if the rows of $\overline{W}_s$ sum up to zero. Specifically if $\mathbf{X}'(k)$ is used to denote the set of signals at the output of $\overline{W}_s$, then

$$\mathbf{X}'(k) = \overline{W}_s\mathbf{X}(k). \tag{26}$$

In addition, if $\mathbf{b}_m{}^T$ is used to represent the mth row of $\overline{W}_s$, we require that the $\mathbf{b}_m{}^T$ satisfy

$$\mathbf{b}_m{}^T\mathbf{1} = 0, \qquad \text{for all } m, \tag{27}$$

and that the $\mathbf{b}_m$ are linearly independent. As a result $\mathbf{X}'(k)$ can have at most $M - 1$ linearly independent components. Equivalently, the row dimension of $\overline{W}_s$ must be $M - 1$ or less.

The lower path of the generalized sidelobe canceler generates a scalar output $y_A(k)$ as the sum of delayed and weighted elements of $\mathbf{X}'(k)$. Following the notation used to describe the linearly constrained beamformer,

where $\mathbf{X}'$ and $\mathbf{A}'$ are the $M - 1$ dimensional signal and coefficient vectors.

The overall output of the generalized sidelobe canceling structure $y(k)$ is

$$y(k) = y_c'(k) - y_A(k). \tag{29}$$

Because $y_A(k)$ contains no desired signal terms, the response of the processor to the desired signal $s(k)\mathbf{1}$ is that produced only by $y_c'(k)$. Thus from (22)–(25) the output due to the presence of only the desired signal satisfies the constraint defined by (9), regardless of $\mathbf{W}_c$. In addition, since $y_A(k)$ contains only noise and interference terms, finding the set of filter coefficients $\mathbf{A}_l'(k)$ which minimize the power contained in $y(k)$ is equivalent to finding the minimum variance, linearly constrained beamformer. The unconstrained least-mean-square (LMS) algorithm [12] can be employed to adapt the filter coefficients to the desired solution,

$$\mathbf{A}_l'(k) = \mathbf{A}_l'(k) + \mu y(k)\mathbf{X}'(k-l). \tag{30}$$

The step size μ is normalized by the total power contained in the $X'(k - l)$ using methods analogous to those described above.

The algorithm in (30), together with conditions (24) and (27), completely defines the operation of the generalized sidelobe canceling structure. Although it is not obvious, this structure can provide exactly the same filtering operation as the constrained beamformer in Fig. 3, which uses Frost's algorithm. In addition, it can also provide filtering operations which are not the same as Frost's procedure. The key lies with the structure of the blocking matrix $\overline{W}_s$ and the conventional beamformer $\mathbf{W}_c$. If the rows of $\overline{W}_s$ are orthogonal (in addition to satisfying (27)) and if all conventional beamformer weights equal $1/M$, then Frost's method is obtained. Nonorthogonal rows and/or other conventional beamformers produce a processor having the same steady-state performance in a stationary environment, but one which uses a different adaptive trajectory.

The generalized sidelobe canceler separates out the constraint as element $\overline{W}_s$ and an FIR filter. In addition, it provides a conventional beamformer as an integral portion of its structure. Coefficient adaptation is reduced to its simplest possible form: the unconstrained LMS algorithm.

Relationship with Linearly Constrained Beamforming

The structure of the generalized sidelobe canceler can readily be related to the adaptive linearly constrained beamformer. We begin by defining an invertible $M \times M$ matrix $\overline{T}$ as

$$\overline{T} = \begin{bmatrix} \mathbf{W}_c{}^T \\ -\overline{W}_s \end{bmatrix}. \tag{31}$$

The inverse of $\overline{T}$ is guaranteed for $\mathbf{W}_C$ and $\overline{W}_s$ satisfying (24) and (27). In addition, the product $\overline{T}\mathbf{1}$ is a simple unit vector,

$$\overline{T}\mathbf{1} = [1, 0, 0, \cdots, 0]^T. \tag{32}$$

Multiplying Frost's algorithm by this invertible transformation yields

$$\mathbf{B}_l(k+1) = \mathbf{B}_l(k) + \mu y(k)[q_x(k-l)\overline{T}\mathbf{1} - \overline{T}\mathbf{X}(k-l)]$$

The transformed weight vector $B_l(k)$ can be partitioned in a manner analogous to (31) as follows

$$B_l(k) = \begin{bmatrix} b_l'(k) \\ \hdashline \mathbf{B}_l'(k) \end{bmatrix}. \tag{34}$$

With this partitioning, and (32), the transformed algorithm (33) is recognized as two algorithms: one in the scalar $b_l'(k)$ and one in the $M-1$ dimensional vector $B_l'(k)$,

$$b_l'(k+1) = b_l'(k) + \mu y(k)[q_x(k-l) - y_c(k-l)] \tag{35a}$$

$$\mathbf{B}_l'(k+1) = \bar{B}_l'(k) + \mu y(k)\mathbf{X}'(k-l). \tag{35b}$$

These equations may be viewed as an alternative implementation of Frost's procedure. Since $\overline{T}$ is invertible, the output $y(k)$ may be expressed as

$$y(k) = \sum_{l=-K}^{K} [\overline{T}^{-1}\mathbf{B}_l(k)]^T X(k-l). \tag{36}$$

Thus if (35) is used to update the $\mathbf{B}_l(k)$ and the output is computed using (36), this procedure is indistinguishable from the original. Many more computations would be required, however, and the transformed system offers no advantages.

We now consider the simplification which arises when $\overline{T}$ is an orthogonal transformation, i.e., when $\overline{T}^{-1} = \overline{T}^T$. The output equation (36) simplifies to

$$y(k) = \sum_{l=-K}^{K} b_l'(k) y_c(k-l) - \sum_{l=-K}^{K} [B'(k)]^T X'(k-l). \tag{37}$$

Inspection of (35)–(37) shows that the transformed linearly constrained beamformer in this case is identical to the adaptive-sidelobe canceling beamformer, provided that the $b_l'(k)$ satisfy

$$b_l(k) = f(l), \tag{38}$$

for all values of k. Since the $b_l(k)$ must satisfy (35a), this will occur only if they are initialized to the values in (38) and if

$$q_x(k-l) = y_c(k-l). \tag{39}$$

This condition is equivalent to the requirement that

$$\mathbf{W}_c = \frac{1}{M}\mathbf{1} \tag{40}$$

or, equivalently, that all beamformer weights have equal values of $1/M$.

In summary the above discussion has shown that the adaptive-sidelobe canceler will be identical to Frost's algorithm provided that the conventional weights satisfy (40) and that $\overline{T}$ is an orthogonal transformation. (From (31) and (4), this that this is a sufficient condition only, and necessity has not been considered.

Jim [13] has studied the comparison in detail and shown that steady-state performance of the two processors is identical regardless of the structure of $\mathbf{W}_c$ and $\overline{W}_s$, provided that the system operates at full rank. He has also shown that different eigenvalue spectra will be encountered by the adaptive filters in the two systems unless $\mathbf{W}_c$ and $\overline{W}_s$ meet the sufficient equality conditions previously described. As a result the coefficient trajectories and adaptive learning curves will differ.

PROPERTIES AND EXTENSIONS OF ADAPTIVE CONSTRAINED BEAMFORMERS

The previous section has presented a generalized sidelobe canceling structure which can be used to implement the error-correcting linearly constrained adaptive algorithm in (10)–(12). This structure can also be used to both analyze the performance of the algorithm and to suggest generalizations of constrained beamforming. We begin by summarizing the performance characteristics of the algorithm which are readily delineated by the sidelobe canceling model. These properties are then used to extend the concept of linearly constrained adaptive beamforming and to develop new methods for use in array processing.

One key element in the sidelobe canceler is the signal-blocking matrix $\overline{W}_s$. As shown by (27), this matrix is required to have $M-1$ linearly independent rows which sum up to zero. Of the many matrices which can be generated with this property, two possibilities which involve only addition operations are shown below for the case $M = 4$:

$$\overline{W}_s^{(1)} = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & 1 \end{bmatrix} \tag{41}$$

$$\overline{W}_s^{(2)} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \tag{42}$$

In the first matrix the rows are mutually orthogonal and are elements of the binary-valued Walsh functions [14]. The second matrix involves fewer operations and consists of taking the difference between adjacent sensor outputs.

One can interpret the rows of $\overline{W}_s$ as fixed-weight beamformers which are applied to the sensor outputs. The beamformed signals are then the elements of $\mathbf{X}'(k)$ and the constraints in (27) ensure the presence of a spatial null in the broadside direction for each beamformer. Note that $\overline{W}_s^{(1)}$ has a different spatial amplitude response for each row while $\overline{W}_s^{(2)}$ has identical patterns.

The effects of imperfect sensor steering and/or gain variations are easily modeled using the generalized sidelobe canceling structure. For example, gain differences at the outputs of the time-delayed sensors result in a set of received signals $x_m(t)$ given by

$$x_m(t) = s(t)(1 + \epsilon_m) + n_m(t) \tag{43}$$

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

WHAT WILL YOU BUILD? | **sales@docketalarm.com** | **1-866-77-FASTCASE**