An Algorithm for Linearly Constrained Adaptive Array Processing

OTIS LAMONT FROST, III, MEMBER, IEEE

Abstract—**A constrained least mean-squares algorithm has been derived which is capable of adjusting an array of sensors in real time to respond to a signal coming from a desired direction while discriminating against noises coming from other directions. Analysis and computer simulations confirm that the algorithm is able to iteratively adapt variable weights on the taps of the sensor array to minimize noise power in the array output. A set of linear equality constraints on the weights maintains a chosen frequency characteristic for the array in the direction of interest.**

The array problem would be a classical constrained least-mean-squares problem except that the signal and noise statistics are assumed unknown *a priori*.

A geometrical presentation shows that the algorithm is able to maintain the constraints and prevent the accumulation of quantization errors in a digital implementation.

I. Introduction

THIS PAPER describes a simple algorithm for adjusting an array of sensors in real time to respond to a desired signal while discriminating against noises. A "signal" is here defined as a waveform of interest which arrives in plane waves from a chosen direction (called the "look direction"). The algorithm iteratively adapts the weights of a broad-band sensor array (Fig. 1) to minimize noise power at at the array output while maintaining a chosen frequency response in the look direction.

The algorithm, called the "Constrained Least Mean-Squares" or "Constrained LMS" algorithm, is a simple stochastic gradient-descent algorithm which requires only that the direction of arrival and a frequency band of interest be specified *a priori*. In the adaptive process, the algorithm progressively learns statistics of noise arriving from directions other than the look direction. Noise arriving from the look direction may be filtered out by a suitable choice of the frequency response characteristic in that direction, or by external means. Subsequent processing of the array output may be done for detection or classification.

A major advantage of the constrained LMS algorithm is that it has a self-correcting feature permitting it to operate for arbitrarily long periods of time in a digital computer implementation without deviating from its constraints because of cumulative roundoff or truncation errors.

The algorithm is applicable to array processing problems in geoscience, sonar, and electromagnetic antenna arrays in which a simple method is required for adjusting an array in real time to discriminate against noises impinging on the array sidelobes.

Manuscript received December 23, 1971; revised May 4, 1972. This research is based on a Ph.D. dissertation in the Department of Electrical Engineering, Stanford University, Stanford, Calif.

The author is with ARGOSystems, Inc., Palo Alto, Calif. 94303.

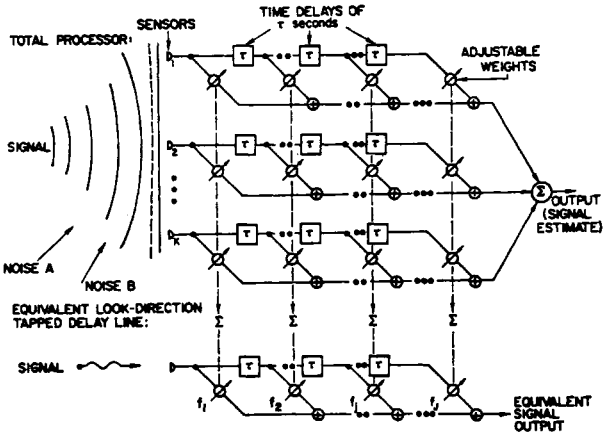

Fig. 1. Broad-band antenna array and equivalent processor for signals coming from the look direction.

Previous Work

Previous work on iterative least squares array processing was done by Griffiths [1]; his method uses an unconstrained minimum-mean-square-error optimization criterion which requires *a priori* knowledge of second-order signal statistics. Widrow, Mantey, Griffiths, and Goode [2] proposed a variable-criterion [3] optimization procedure involving the use of a known training signal; this was an application and extension of the original work on adaptive filters done by Widrow and Hoff [4]. Griffiths also proposed a constrained least mean-squares processor not requiring *a priori* knowledge of the signal statistics [5]; a new derivation of this processor, given in [6], shows that it may be considered as putting "soft" constraints on the processor via the quadratic penalty-function method.

"Hard" (i.e., exactly)-constrained iterative optimization was studied by Rosen [7] for the deterministic case. Lacoss [8], Booker *et al.* [9], and Kobayashi [10] studied "hard"-constrained optimization in the array processing context for filtering short lengths of data. All four authors used gradient-projection techniques [11]; Rosen and Booker correctly indicated that gradient-projection methods are susceptible to cumulative roundoff errors and are not suitable for long runs without an additional error-correction procedure. The constrained LMS algorithm developed in the present work is designed to avoid error accumulation while maintaining a "hard" constraint; as a result, it is able to provide continual filtering for arbitrarily large numbers of iterations.

Basic Principle of the Constraints

The algorithm is able to maintain a chosen frequency response in the look direction while minimizing output noise

power because of a simple relation between the look direction frequency response and the weights in the array of Fig. 1. Assume that the look direction is chosen as the direction perpendicular to the line of sensors. Then identical signal components arriving on a plane wavefront parallel to the line of sensors appear at the first taps simultaneously and parade in parallel down the tapped delay lines following each sensor; however, noise waveforms arriving from other than the look direction will not, in general, produce equal voltage components on any given vertical column of taps. The voltages (signal plus noise) at each tap are multiplied by the tap weights and added to form the array output. Thus as far as the signal is concerned, the array processor is equivalent to a single tapped delay line in which each weight is equal to the sum of the weights in the corresponding vertical column of the processor, as indicated in Fig. 1. These summation weights in the equivalent tapped delay line must be selected so as to give the desired frequency response characteristic in the look direction.

If the look direction is chosen to be other than that perpendicular to the line of sensors, then the array can be steered either mechanically or electrically by the addition of steering time delays (not shown) placed immediately after each sensor.

A processor having K sensors and J taps per sensor has KJ weights and requires J constraints to determine its look-direction frequency response. The remaining $KJ-J$ degrees of freedom in choosing the weights may be used to minimize the total power in the array output. Since the look-direction frequency response is fixed by the J constraints, minimization of the total output power is equivalent to minimizing the non-look-direction noise power, so long as the set of signal voltages at the taps is uncorrelated with the set of noise voltages at these taps. The latter assumption has commonly been made in previous work on iterative array processing [1], [5], [8]–[10]. The effect of signal-correlated noise in the array may be to cancel out all or part of the desired signal component in the array output. Sources of signal-correlated noise may be multiple signal-propagation paths, and coherent radar or sonar "clutter."

It is permissible, and in fact desirable for proper noise cancellation that the voltages produced by the noises on the taps of the array be correlated among themselves, although uncorrelated with the signal voltages. Examples of such noises include waveforms from point sources in other than the look direction (e.g., lightning, "jammers," noise from nearby vehicles), spatially localized incoherent clutter, and self-noise from the structure carrying the array.

Noise voltages which are uncorrelated between taps (e.g., amplifier thermal noise) may be partially rejected by the adaptive array in two ways. As in a conventional nonadaptive array, such noises are eliminated to the extent that signal voltages on the taps are added coherently at the array output, while uncorrelated noise voltages are added incoherently. Second, an adaptive array can reduce the weighting on any tap that may have a disportionately large uncorrelated noise power.

II. Optimum-Constrained LMS Weight Vector

The first step in developing the constrained LMS algorithm is to find the optimum weight vector.

Notation

Notation will be as follows (see Fig. 2):

Every Δ seconds, where Δ may be a multiple of the delay τ between taps, the voltages at the array taps are sampled. The vector of tap voltages at the kth sample is written $X(k)$, where

$$X^T(k) \triangleq [x_1(k\Delta), x_2(k\Delta), \cdots, x_{KJ}(k\Delta)].$$

The superscript T denotes transpose. The tap voltages are the sums of voltages due to look-direction waveforms l and non-look-direction noises n, so that

$$X(k) = L(k) + N(k) \tag{1}$$

where the KJ-dimensional vector of look-direction waveforms at the kth sample is

$$L(k) \triangleq \begin{bmatrix} l(k\Delta) \\ \vdots \\ l(k\Delta) \\ l(k\Delta - \tau) \\ \vdots \\ l(k\Delta - \tau) \\ \vdots \\ l(k\Delta - (J-1)\tau) \\ \vdots \\ l(k\Delta - (J-1)\tau) \end{bmatrix} \begin{matrix} \} K \text{ taps} \\ \\ \\ \} K \text{ taps} \\ \\ \\ \} K \text{ taps} \\ \\ \end{matrix}$$

and the vector of non-look-direction noises is

$$N^T(k) \triangleq [n_1(k\Delta), n_2(k\Delta), \cdots, n_{KJ}(k\Delta)].$$

The vector of weights at each tap is W, where

$$W^T \triangleq [w_1, w_2, \cdots, w_{KJ}].$$

It is assumed for this derivation that the signals and noises are adequately modeled as zero-mean random processes with (unknown) second-order statistics:

$$E[X(k)X^T(k)] \triangleq R_{XX} \tag{2a}$$

$$E[N(k)N^T(k)] \triangleq R_{NN} \tag{2b}$$

$$E[L(k)L^T(k)] \triangleq R_{LL}. \tag{2c}$$

As previously stated, it is assumed that the vector of look-direction waveforms is uncorrelated with the vector of non-look-direction noises, i.e.,

$$E[N(k)L^T(k)] = 0. \tag{3}$$

It is assumed that the noise environment is distributed so that R_{XX} and R_{NN} are positive definite [12].

The output of the array (signal estimate) at the time of kth sample is

$$y(k) = W^TX(k) = X^T(k)W. \tag{4}$$

Using (4) the expected output power of the array is

$$E[y^2(k)] = E[W^TX(k)X^T(k)W] = W^TR_{XX}W. \tag{5}$$

The constraint that the weights on the jth vertical column of taps sum to a chosen number f_j (see Fig. 1) is expressed by

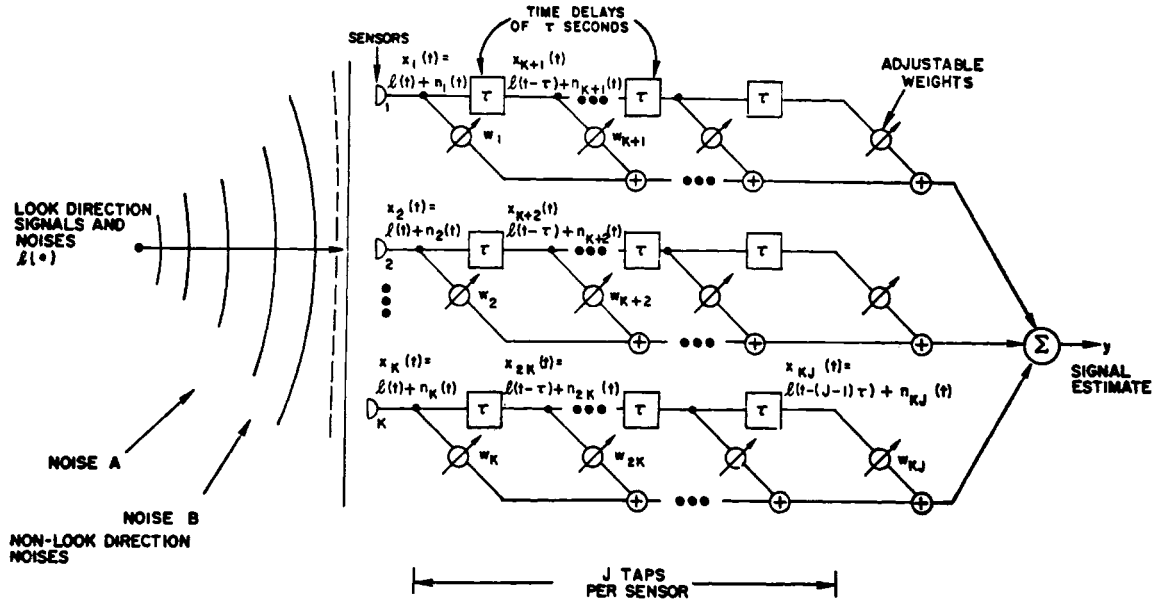

Fig. 2. Signals and noises on the array. Because the array is steered toward the look direction, all beam signal components on any given column of filter taps are identical.

the requirement

$$c_j^T W = f_j, \qquad j = 1, 2, \cdots, J \tag{6}$$

where the KJ-dimensional vector c_j has the form

$$c_j = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \\ \vdots \\ 0 \end{bmatrix} \begin{matrix} \} K \\ \\ \} K \\ \\ \} j\text{th group of } K \text{ elements.} \\ \} K \\ \\ \} K \end{matrix} \tag{7}$$

Constraining the weight vector to satisfy the J equations of (6) restricts W to a $(KJ-J)$-dimensional plane.

Define the constraint matrix C as

$$C \triangleq \underbrace{\begin{bmatrix} c_1 & \cdots & c_j & \cdots & c_J \end{bmatrix}}_{J} \Big\} KJ \tag{8}$$

and define $\mathfrak{F}$ as the J-dimensional vector of weights of the look-direction-equivalent tapped delay line shown in Fig. 1:

$$\mathfrak{F} \triangleq \begin{bmatrix} f_1 \\ \vdots \\ f_j \\ \vdots \\ f_J \end{bmatrix}. \tag{9}$$

By inspection the constraint vectors c_j are linearly independent, hence, C has full rank equal to J. The constraints (6) are now written

$$C^T W = \mathfrak{F}. \tag{10}$$

Optimum Weight Vector

Since the look-direction-frequency response is fixed by the J constraints, minimization of the non-look-direction noise power is the same as minimization of the total output power. The cost criterion used in this paper will be minimization of total array output power $W^T R_{XX} W$. The problem of finding the optimum set of filter weights W_{opt} is summarized by (5) and (10) as

$$\underset{W}{\text{minimize}} \quad W^T R_{XX} W \tag{11a}$$

$$\text{subject to } C^T W = \mathfrak{F}. \tag{11b}$$

This is the constrained LMS problem.

W_{opt} is found by the method of Lagrange multipliers, which is discussed in [13]. Including a factor of $\frac{1}{2}$ to simplify later arithmetic, the constraint function is adjoined to the cost function by a J-dimensional vector of undetermined Lagrange multipliers λ:

$$H(W) = \tfrac{1}{2} W^T R_{XX} W + \lambda^T (C^T W - \mathfrak{F}). \tag{12}$$

Taking the gradient of (12) with respect to W

$$\nabla_W H(W) \; R_{XX} W + C\lambda. \tag{13}$$

The first term in (13) is a vector proportional to the gradient of the cost function (11a), and the second term is a vector

normal to the $(KJ-J)$-dimensional constraint plane defined by $C^TW-\mathfrak{F}=0$ [14]. For optimality these vectors must be antiparallel [13], which is achieved by setting the sum of the vectors (13) equal to zero

$$\nabla_W H(W) = R_{XX}W + C\lambda = 0.$$

In terms of the Lagrange multipliers, the optimal weight vector is then

$$W_{\text{opt}} = -R_{XX}^{-1}C\lambda \tag{14}$$

where R_{XX}^{-1} exists because R_{XX} was assumed positive definite. Since W_{opt} must satisfy the constraint (11b)

$$C^TW_{\text{opt}} = \mathfrak{F} = -C^TR_{XX}^{-1}C\lambda$$

and the Lagrange multipliers are found to be

$$\lambda = -[C^TR_{XX}^{-1}C]^{-1}\mathfrak{F} \tag{15}$$

where the existence of $[C^TR_{XX}^{-1}C]^{-1}$ follows from the facts that R_{XX} is positive definite and C has full rank [6]. From (14) and (15) the optimum-constrained LMS weight vector solving (11) is

$$W_{\text{opt}} = R_{XX}^{-1}C[C^TR_{XX}^{-1}C]^{-1}\mathfrak{F}. \tag{16}$$

Using the set of weights W_{opt} in the array processor of Fig. 2 forms the optimum constrained LMS processor, which is a filter in space and frequency. Substituting W_{opt} in (4), the constrained least squares estimate of the look-direction waveform is

$$y_{\text{opt}}(k) = W_{\text{opt}}{}^TX(k). \tag{17}$$

Discussion

The constrained LMS filter is sometimes known by other names. If the frequency characteristic in the look-direction is chosen to be all-pass and linear phase (distortionless), the output of the constrained LMS filter is the maximum likelihood estimate of a stationary process in Gaussian noise if the angle of arrival is known [15]. The distortionless form of the constrained LMS filter is called by some authors the "Minimum Variance Distortionless Look" estimator, "Maximum Likelihood Distortionless Estimator," and "Least Squares Unbiased Estimator." By suitable choice of $\mathfrak{F}$ a variety of other optimal processors can be obtained [16].

III. The Adaptive Algorithm

In this paper it is assumed that the input correlation matrix R_{XX} is unknown *a priori* and must be learned by an adaptive technique. In stationary environments during learning, and in time-varying environments, an estimate of the optimum filter weights must be recomputed periodically. Direct substitution of a correlation matrix estimate into the optimal-weight equation (16) requires a number of multiplications at each iteration proportional to the cube of the number of weights. The complexity is primarily caused by the required inversion of the input correlation matrix. Recently Saradis *et al.* [17] and Mantey and Griffiths [18] have shown how to iteratively update matrix inversions, requiring only a number of multiplications and storage locations proportional to the square of the number of weights. The gradient-descent constrained LMS algorithm presented here requires only a number of multiplications and storage locations directly proportional to the number of weights. It is therefore simple to implement and, for a given computational cost, is applicable to arrays in which the number of weights is on the order of the square of the number that could be handled by the iterative matrix inversion method and the cube of the number that could be handled by the direct substitution method.

Derivation

For motivation of the algorithm derivation temporarily suppose that the correlation matrix R_{XX} is known. In constrained gradient-descent optimization, the weight vector is initialized at a vector satisfying the constraint (11b), say $W(0)=C(C^TC)^{-1}\mathfrak{F}$, and at each iteration the weight vector is moved in the negative direction of the constrained gradient (13). The length of the step is proportional to the magnitude of the constrained gradient and is scaled by a constant μ. After the kth iteration the next weight vector is

$$\begin{aligned} W(k+1) &= W(k) - \mu\nabla_W H[W(k)] \\ &= W(k) - \mu[R_{XX}W(k) + C\lambda(k)] \end{aligned} \tag{18}$$

where the second step is from (13). The Lagrange multipliers are chosen by requiring $W(k+1)$ to satisfy the constraint (11b):

$$\mathfrak{F} = C^TW(k+1) = C^TW(k) - \mu C^TR_{XX}W(k) - \mu C^TC\lambda(k).$$

Solving for the Lagrange multipliers $\lambda(k)$ and substituting into the weight-iteration equation (18) we have

$$\begin{aligned} W(k+1) = W(k) &- \mu[I - C(C^TC)^{-1}C^T]R_{XX}W(k) \\ &+ C(C^TC)^{-1}[\mathfrak{F} - C^TW(k)]. \end{aligned} \tag{19}$$

The deterministic algorithm (19) is shown in this form to emphasize that the last factor $\mathfrak{F}-C^TW(k)$ is not assumed to be zero, as it would be if the weight vector precisely satisfied the constraint at the kth iteration. It will be shown in Section VI that this term permits the algorithm to correct any small deviations from the constraint due to arithmetic inaccuracy and prevents their eventual accumulation and growth.

Defining the KJ-dimensional vector

$$F \triangleq C(C^TC)^{-1}\mathfrak{F} \tag{20a}$$

and the $KJ\times KJ$ matrix

$$P \triangleq I - C(C^TC)^{-1}C^T \tag{20b}$$

the algorithm may be rewritten as

$$W(k+1) = P[W(k) - \mu R_{XX}W(k)] + F. \tag{21}$$

Equation (21) is a deterministic constrained gradient descent algorithm requiring knowledge of the input correlation matrix R_{XX}, which, however, in the array problem is unavailable *a priori*. An available and simple approximation for R_{XX} at the kth iteration is the outer product of the tap voltage vector with itself: $X(k)X^T(k)$. Substitution of this estimate into (21) gives the stochastic constrained LMS algorithm

$$\boxed{\begin{aligned} W(0) &= F \\ W(k+1) &= P[W(k) - \mu y(k)X(k)] + F \end{aligned}} \tag{22}$$

where $y(k)$ is the array output (signal estimate) defined by (4).

Discussion

The constrained LMS algorithm (22) satisfies the constraint $C^T W(k+1) = \mathcal{F}$ at each iteration, as can be verified by premultiplying (22) by C^T and using (20). At each iteration the algorithm requires only the tap voltages $X(k)$ and the array output $y(k)$; no *a priori* knowledge of the input correlation matrix is needed. F is a constant vector that can be precomputed. One of the two most complex operations required by (22) is the multiplication of each of the KJ components of the vector $X(k)$ by the scalar $\mu y(k)$; the other significant operation is indicated by the matrix $P = I - C(C^T C)^{-1} C^T$. Because of the simple form of C [refer to (7)], multiplication of a vector by P as indicated in (22) amounts to little more than a few additions. Expressed in summation notation the iterative equations for the weight vector components are

$$
\begin{aligned}
w_1(k+1) &= w_1(k) - \mu y(k) x_1(k) - \frac{1}{K} \sum_{j=1}^{K} [w_j(k) - \mu y(k) x_j(k)] + \frac{f_1}{K} \\
&\vdots \\
w_K(k+1) &= w_K(k) - \mu y(k) x_K(k) - \frac{1}{K} \sum_{j=1}^{K} [w_j(k) - \mu y(k) x_j(k)] + \frac{f_1}{K} \\
w_{K+1}(k+1) &= w_{K+1}(k) - \mu y(k) x_{K+1}(k) - \frac{1}{K} \sum_{j=K+1}^{2K} [w_j(k) - \mu y(k) x_j(k)] + \frac{f_2}{K} \\
&\vdots \\
w_{2K}(k+1) &= w_{2K}(k) - \mu y(k) x_{2K}(k) - \frac{1}{K} \sum_{j=K+1}^{2K} [w_j(k) - \mu y(k) x_j(k)] + \frac{f_2}{K} \\
&\vdots \\
w_{JK}(k+1) &= w_{JK}(k) - \mu y(k) x_{JK}(k) - \frac{1}{K} \sum_{j=(J-1)K+1}^{JK} [w_j(k) - \mu y(k) x_j(k)] + \frac{f_J}{K}.
\end{aligned}
$$

These equations can readily be implemented on a digital computer.

IV. Performance

Convergence to the Optimum

The weight vector $W(k)$ obtained by the use of the stochastic algorithm (22) is a random vector. Convergence of the mean weight vector to the optimum is demonstrated by showing that the length of the difference vector between the mean weight vector and the optimum (16) asymptotically approaches zero.

Proof of convergence of the mean is greatly simplified by the assumption (used in [2]) that successive samples of the input vector taken Δ seconds apart are statistically independent. This condition can usually be approximated in practice by sampling the input vector at intervals large compared to the correlation time of the input process plus the length of time it takes an input waveform to propagate down the array. The assumption is more restrictive than necessary, since Daniell [19] has shown that the much weaker assumption of asymptotic independence of the input vectors is sufficient to demonstrate convergence in the related unconstrained least squares problem.

Taking the expected value of both sides of the algorithm (22), using (4), (2a), and the independence assumption yields an iterative equation in the mean value of the constrained LMS weight vector

$$E[W(k+1)] = P\{E[W(k)] - \mu R_{XX} E[W(k)]\} + F. \quad (23)$$

Define the vector $V(k+1)$ to be the difference between the mean adaptive weight vector at iteration $k+1$ and the optimal weight vector (16)

$$V(k+1) \triangleq E[W(k+1)] - W_{\text{opt}}.$$

Using (23) and the relations $F = (I-P) W_{\text{opt}}$ and $P R_{XX} W_{\text{opt}} = 0$, which may be verified by direct substitution of (16) and (20b), an equation for the difference process may be constructed

$$V(k+1) = PV(k) - \mu P R_{XX} V(k). \quad (24)$$

The idempotence of P (i.e., $P^2 = P$), which can be verified by carrying out the multiplication using (20b) and premultiplication of equation (24) by P shows that $PV(k) = V(k)$ for all k, so (24) can be written

$$
\begin{aligned}
V(k+1) &= [I - \mu P R_{XX} P] V(k) \\
&= [I - \mu P R_{XX} P]^{k+1} V(0).
\end{aligned}
$$

The matrix $P R_{XX} P$ determines both the rate of convergence of the mean weight vector to the optimum and the steady-state variance of the weight vector about the optimum. It is shown in [6] that $P R_{XX} P$ has precisely J zero eigenvalues, corresponding to the column vectors of the constraint matrix C; this is a result of the fact that during adaption no movement is permitted away from $(KJ-J)$-dimensional constraint plane. It is also shown in [6, appendix C] that $P R_{XX} P$ has $KJ-J$ nonzero eigenvalues σ_i, $i = 1, 2, \cdots, KJ-J$, with values bounded between the smallest and largest eigenvalues of R_{XX}

$$\lambda_{\min} \leq \sigma_{\min} \leq \sigma_i \leq \sigma_{\max} \leq \lambda_{\max}, \quad i = 1, 2, \cdots, KJ - J$$

where $\lambda_{\min}$ and $\lambda_{\max}$ are the smallest and largest eigenvalues of R_{XX} and $\sigma_{\min}$ and $\sigma_{\max}$ are the smallest and largest nonzero eigenvalues of $P R_{XX} P$.

Examination of $V(0) = F - W_{opt}$ shows that it can be expressed as a linear combination of the eigenvectors of $PR_{XX}P$ corresponding to nonzero eigenvalues. If $V(0)$ is equal to an eigenvector of $PR_{XX}P$, say e_i with eigenvalue $\sigma_i \neq 0$ then

$$\begin{aligned} V(k+1) &= [I - \mu PR_{XX}P]^{k+1} e_i \\ &= [1 - \mu\sigma_i]^{k+1} e_i. \end{aligned}$$

The convergence of the mean weight vector to the optimum weight vector along any eigenvector of $PR_{XX}P$ is therefore geometric with geometric ratio $(1-\mu\sigma_i)$. The time required for the euclidean length of the difference vector to decrease to e^{-1} of its initial value (time constant) is

$$\tau_i = \Delta / \ln (1 - \mu\sigma_i) \cong \Delta / \mu\sigma_i \tag{25}$$

where the approximation is valid for $\mu\sigma_i \ll 1$.

If μ is chosen so that

$$0 < \mu < 1/\sigma_{max}$$

then the length (norm) of any difference vector is bounded between two ever-decreasing geometric progressions

$$\begin{aligned} (1 - \mu\sigma_{max})^{k+1} \|V(0)\| \leq \|V(k+1)\| \\ \leq (1 - \mu\sigma_{min})^{k+1} \|V(0)\| \end{aligned}$$

and so if the initial difference is finite the mean weight vector converges to the optimum, i.e.,

$$\lim_{k \to \infty} \|E[W(k)] - W_{opt}\| = 0$$

with time constants given by (25).

Steady-State Performance—Stationary Environment

The algorithm is designed to continually adapt for coping with nonstationary noise environments. In stationary environments this adaptation causes the weight vector to have a variance about the optimum and produces an additional component of noise (above the optimum) to appear at the output of the adaptive processor.

The output power of the optimum processor with a fixed weight vector (17) is

$$\begin{aligned} E[y_{opt}^2(k)] &= W_{opt}^T R_{XX} W_{opt} \\ &= \mathfrak{F}^T (C^T R_{XX}^{-1} C)^{-1} \mathfrak{F}. \end{aligned}$$

A measure of the fraction of additional noise caused by the adaptive algorithm operating in steady state in a stationary environment is termed "misadjustment" $M(\mu)$ by Widrow

$$M(\mu) \triangleq \lim_{k \to \infty} \frac{E[y^2(k)] - E[y_{opt}^2(k)]}{E[y_{opt}^2(k)]}.$$

By assuming that successive observation vectors [vectors $X(k)$ of tap voltages] are independent and have components $x_1(k), \cdots, x_{KL}(k)$ that are jointly Gaussian distributed, Moschner [20] calculated very tight bounds on the misadjustment, using a method due to Senne [21], [22]. For a convergence constant μ satisfying

$$0 < \mu < \frac{1}{\sigma_{max} + (1/2)\ \mathrm{tr}\ (PR_{XX}P)} \tag{26}$$

the steady-state misadjustment may be bounded by

$$\begin{aligned} \frac{\mu}{2} \frac{\mathrm{tr}\ (PR_{XX}P)}{1 - \frac{\mu}{2}[\mathrm{tr}\ (PR_{XX}P) + 2\sigma_{min}]} \\ \leq M(\mu) \leq \frac{\mu}{2} \frac{\mathrm{tr}\ (PR_{XX}P)}{1 - \frac{\mu}{2}[\mathrm{tr}\ (PR_{XX}P) + 2\sigma_{max}]} \end{aligned} \tag{27}$$

where tr denotes trace.

$M(\mu)$ can be made arbitrarily close to zero by suitably small choice of μ; this means that the steady-state performance of the constrained LMS algorithm can be made arbitrarily close to the optimum. From (25) it seen that such performance is obtained at the expense of increased convergence time.

If μ is chosen to satisfy

$$0 < \mu < \frac{2}{3\ \mathrm{tr}\ (R_{XX})} \tag{28}$$

then it is guaranteed to satisfy (26). Griffiths [1] shows that the upper bound in (28) can be calculated directly and easily from observations since $\mathrm{tr}\ (R_{XX}) = E[X^T(k)X(k)]$, the sum of the powers of the tap voltages.

Steady-State Performance—Nonstationary Environment

A model of the effect of a nonstationary noise environment proposed by Brown [23] is that the steady-state rms change of the optimal weight vector $W_{opt}(k)$ between iterations has magnitude δ, i.e.,

$$\limsup_{k \to \infty} E\|W_{opt}(k+1) - W_{opt}(k)\|^2 = \delta^2.$$

Brown's general results may be applied to the constrained LMS algorithm by restricting the optimal weight vector to have magnitude less than some number $\|W_{max}\|$ and again assuming the successive input vectors are independent with Gaussian-distributed components. For μ small it can be shown [23, p. 47] that the steady-state rms distance of the weight vector from the optimum is bounded by

$$\begin{aligned} &(\limsup_{k \to \infty} E\|W(k) - W_{opt}(k)\|^2)^{1/2} \\ &\leq \frac{\delta + \mu 2^{1/2}[\lambda^*\ \mathrm{tr}^*\ (PR_{XX}P) + \sigma^{*2}]\|W_{max}\|}{1 - \{1 - 2\mu\sigma_* + \mu^2[3\sigma^{*2} + 2\lambda^*\ \mathrm{tr}^*\ (PR_{XX}P)]\}^{1/2}} \end{aligned} \tag{29}$$

where any starred quantities q_* or q^* are taken to bound the corresponding time-varying quantity $q(k)$, i.e., $q_* \leq q(k) \leq q^*$ for all k. In general, the optimum convergence constant μ that minimizes the upper bound (29) for a nonstationary environment is nonzero. This contrasts with the stationary case, in which the best steady-state performance is obtained by making μ as small as possible.

V. Geometrical Interpretation

The constrained LMS algorithm (22) has a simple geometrical interpretation that is useful for visualizing the error-correcting property which keeps the weight vector from deviating from its constraints.

In an error-free implementation of the algorithm, the KJ-dimensional weight vectors satisfy the constraint equation (11b) and therefore terminate on a constraint plane Λ defined

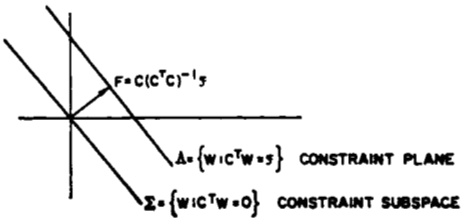

Fig. 3. The $(KJ-J)$-plane Λ and subspace Σ defined by the constraint.

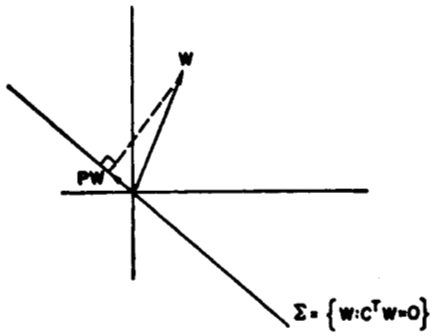

Fig. 4. P projects vectors onto the constraint subspace.

by

$$\Lambda = \{W : C^T W = \mathfrak{F}\}.$$

This $(KJ-J)$-dimensional constraint plane is indicated diagrammatically in Fig. 3.

It is well known [14] that vectors pointing in a direction normal to the constraint plane (but not necessarily normal to the vectors that terminate on that plane) are linear combinations of the constraint matrix column vectors. These vectors have the form CA, where A is a J-dimensional vector determining the linear combination. Thus the vector $F = C(C^TC)^{-1}\mathfrak{F}$, appearing in the algorithm (22) and used as the initial weight vector, points in a direction normal to the constraint plane. F also terminates on the constraint plane since $C^TF = \mathfrak{F}$. Thus F is the shortest vector terminating on the constraint plane (see Fig. 3).

The homogeneous form of the constraint equation

$$C^T W = 0 \tag{30}$$

defines a second $(KJ-J)$-dimensional plane, which includes the zero vector and thus passes through the origin. Such a plane is called a subspace [11] (see Fig. 3).

The matrix P in the algorithm (22) is a projection operator [24]. Premultiplication of any vector by P will annihilate any components perpendicular to Σ, projecting the vector into the constraint subspace (see Fig. 4).

The vector $y(k)X(k)$ in the algorithm is an estimate of the unconstrained gradient. Referring to (12) the unconstrained cost function is $\frac{1}{2}W^TR_{XX}W$. The unconstrained gradient [refer to (13)] is $R_{XX}W$. The estimate of $R_{XX}W$ at the kth iteration, used in deriving (22), is $y(k)X(k)$.

Contours of constant output power (cost) and the optimum constrained weight vector W_{opt} that minimizes the output power are shown in Fig. 5.

The operation of the constrained LMS algorithm is shown in Fig. 6. In this example, the unconstrained negative gradient estimate $-y(k)X(k)$ is scaled by μ and added to the current weight vector $W(k)$. This is an attempt to change the weight vector in a direction that minimizes output power. In

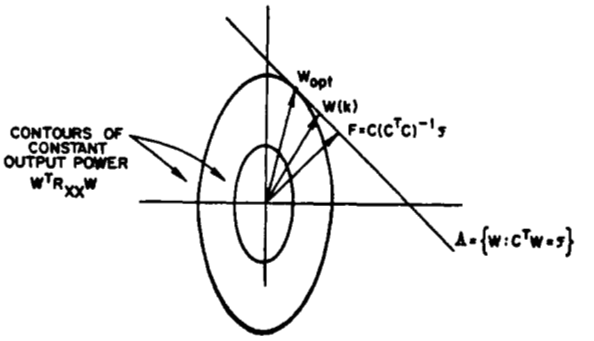

Fig. 5 Example showing contours of constant output power and the constrained weight vector that minimizes output power.

$$W_{opt} = R_{XX}^{-1}C(C^TR_{XX}^{-1}C)^{-1}\mathfrak{F}.$$

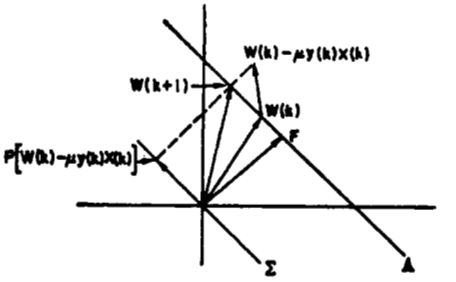

Fig. 6. Operation of the constrained LMS algorithm:

$$W(k+1) = P[W(k) - \mu y(k)X(k)] + F.$$

general, this change moves the resulting vector off the constraint plane. The resulting vector is projected onto the constraint subspace and then returned to the constraint plane by adding F. The new weight vector $W(k+1)$ satisfies the constraint to within the accuracy of the arithmetic used in implementing the algorithm.

VI. Error-Correcting Feature

In a digital-computer implementation of any algorithm, it is likely that small computational errors will occur at each iteration because of truncation, roundoff, or quantization errors. A difficulty in applying the well-known gradient-projection algorithm to the real time array-processing problem is that computational errors causing deviations of the weight vector from the constraint are not corrected [7], [9]. Without additional error-correcting procedures, application of the gradient-projection algorithm is limited to problems requiring few enough iterations that significant deviations from the constraint do not occur. The constrained LMS algorithm, on the other hand, was specifically designed to continuously correct for such errors and prevent them from accumulating. The reason for this characteristic is shown by a geometrical comparison of the two algorithms.

The gradient-projection algorithm may be derived by following the derivation of the constrained LMS algorithm to (19) and dropping the last factor, $\mathfrak{F} - C^TW(k)$. This factor would be equal to zero in a perfect implementation in which the weight vector satisfied the constraint $C^TW(k) = \mathfrak{F}$ at each iteration. The algorithm that results when the term is dropped is

$$W(0) = C(C^TC)^{-1}\mathfrak{F}$$
$$W(k+1) = W(k) - \mu P y(k)X(k). \tag{31}$$

This is a gradient-projection algorithm [11]. It is so named

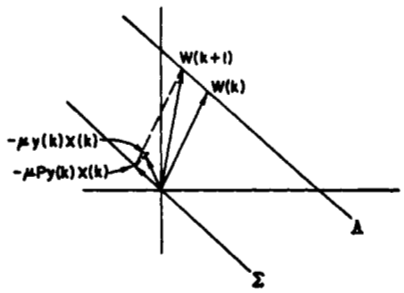

Fig. 7. Operation of the gradient-projection algorithm (31).

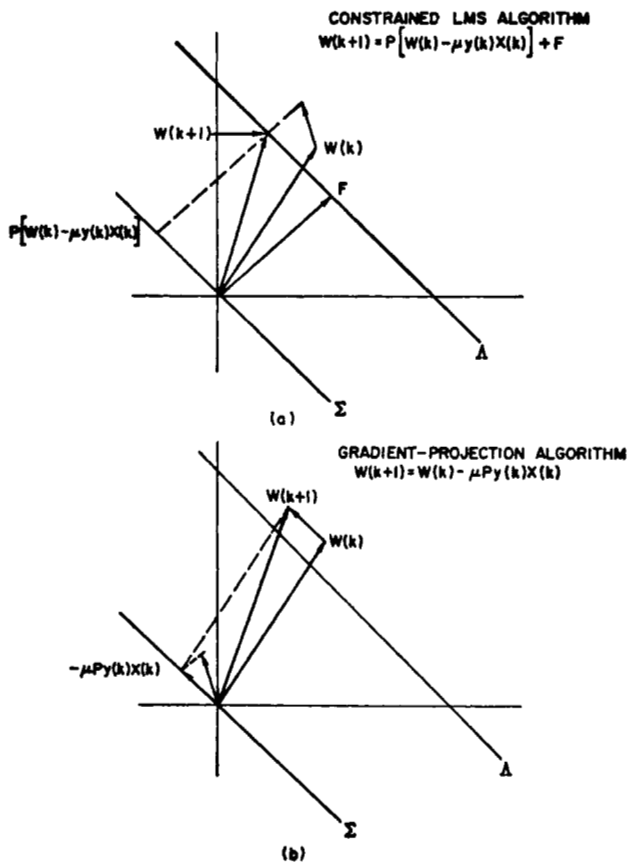

Fig. 8. Error propagation. The constrained LMS algorithm (a) corrects deviations from the constraints while the gradient-projection algorithm (b) allows them to accumulate.

because the unconstrained gradient estimate $y(k)X(k)$ is projected onto the constraint subspace and then added to the current weight vector. Its operation is shown in Fig. 7 (compare with Fig. 6).

A comparison between the effect of computational errors on the gradient-projection algorithm and on the constrained LMS algorithm is shown in Fig. 8. The weight vector is assumed to be off the constraint at the kth iteration because of a quantization error occurring in the previous iteration. It is shown in Fig. 8(a) that the constrained LMS algorithm makes the unconstrained step, projects onto the subspace, and then adds F, producing a new weight vector $W(k+1)$ that satisfies the constraint. The gradient-projection algorithm [Fig. 8(b)], however, projects the gradient estimate onto the subspace and adds the projected vector to the past weight vector, moving parallel to the constraint plane but continuing the error. Note the implicit (incorrect) assumption that $W(k)$ satisfied the constraint, corresponding to the same assumption made in the derivation of the gradient-projection algorithm.

Accumulating errors in the gradient-projection algorithm can be expected to cause the weight vector to do a random walk away from the constraint plane with variance (expected squared distance from the plane) increasing linearly with the number of iterations. By contrast, the expected deviation of the constrained LMS algorithm from the constraint does not grow, remaining at its original value.

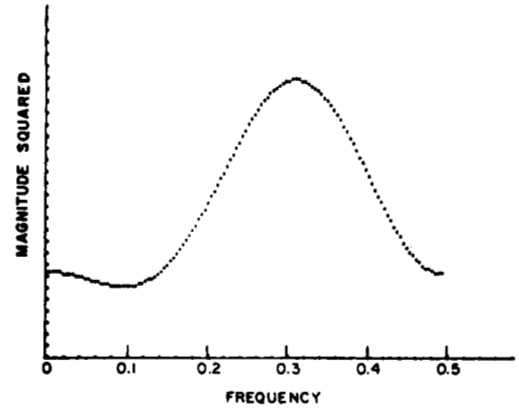

Fig. 9. Frequency response of the processor in the look direction.

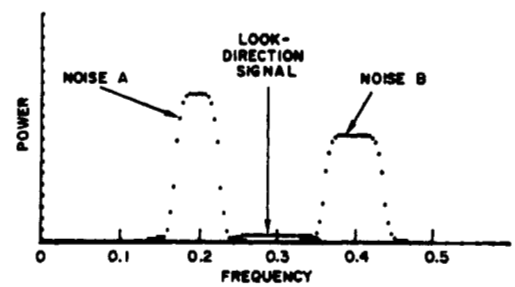

Fig. 10. Power spectral density of incoming signals. See Fig. 2 and Table I for spatial position of noises.

TABLE I

SIGNALS AND NOISES IN THE SIMULATION (SEE FIG. 2)

Source	Power	Direction (0° is normal to array)	Center Frequency (1.0 is $1/\tau$)	Bandwidth
Look-direction signal	0.1	0°	0.3	0.1
Noise A	1.0	45°	0.2	0.05
Noise B	1.0	60°	0.4	0.07
White noise (per tap)	0.1			

VII. SIMULATION

A computer simulation of the processor was made using 6-digit floating point arithmetic on a small computer (the HP-2116). The processor had four sensors on a line spaced at τ-second intervals and had four taps per sensor (thus $KJ = 16$). The environment had three point-noise sources, and white noise added to each sensor. Power of the look direction signal was quite small in comparison to the power of interfering noises (see Table I). The tap spacing defined a frequency of 1.0 (i.e., $f = 1.0$ is a frequency of $1/\tau$ Hz). In the look-direction, foldover frequency for the processor response was $\frac{1}{2}\tau$, or 0.5. All signals were generated by a pseudo-Gaussian gen-

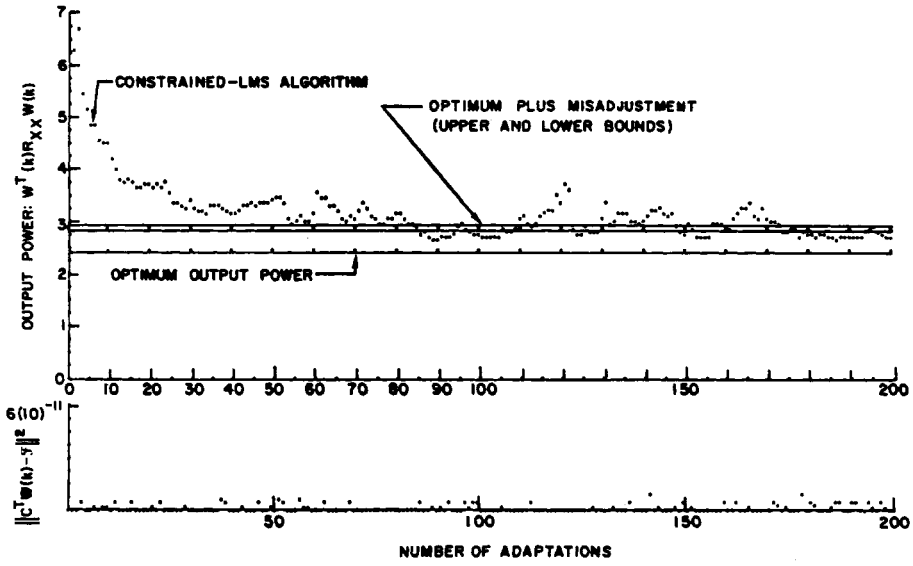

Fig. 11. The output power of the constrained LMS filter (upper graph) decreases as it adapts to discriminate against unwanted noise. Lower curve shows small deviations from the constraint due to quantization.

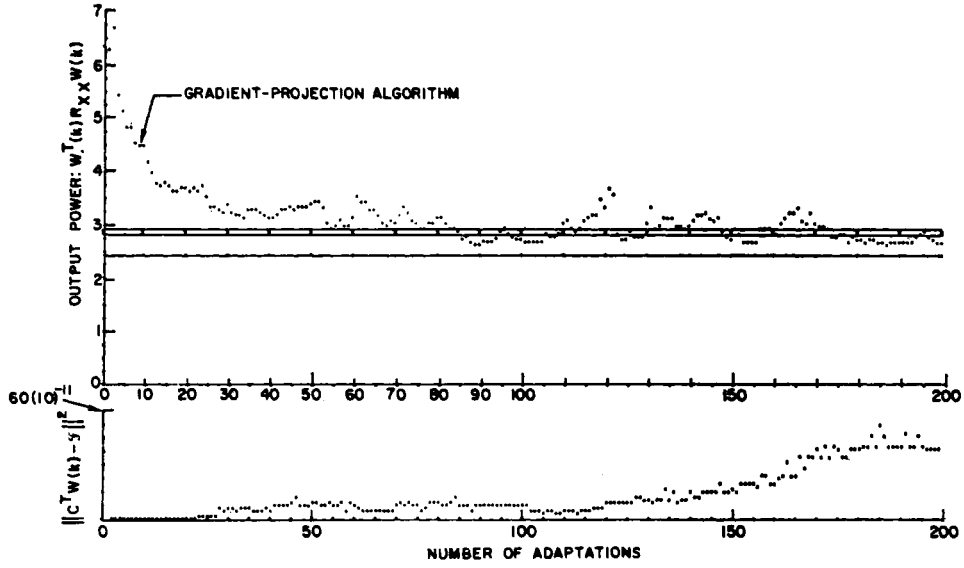

Fig. 12. Output power of the gradient-projection algorithm (upper graph) operated on the same data as the constrained LMS algorithm (c.f., Fig. 11). Lower curve shows that deviations from the constraint tend to increase with time. Note scale.

erator and filtered to give them the proper spatial and temporal correlations. All temporal correlations were arranged to be identically zero for time differences greater than 25τ. The time between adaptations Δ was assumed greater than 58τ, so successive samples of $X(k)$ were uncorrelated.

The look direction filter was specified by the vector $\mathcal{F}^T = [1, -2, 1.5, 2]$ which resulted in the frequency characteristic shown in Fig. 9. The signal and noise spectra are shown in Fig. 10 and their spatial position in Fig. 2.

In this problem, the eigenvalues of R_{XX} ranged from 0.111 to 8.355. The upper permissible bound on the convergence constant μ calculated by (26) was 0.074; a value of $\mu = 0.01$ was selected, which, by (27), would lead to a misadjustment of between 15.2 and 17.0 percent.

The processor was initialized with $W(0) = F = C(C^TC)^{-1}\mathcal{F}$, and Fig. 11 shows performance as a function of time. The upper graph has three horizontal lines. The lower line is the output power of the optimum weight vector. The closely spaced upper two lines are upper and lower bounds for the adaptive processor output power, which is the optimum output power plus misadjustment. The mean steady-state value of the processor's output power falls somewhere between the upper and lower bounds (but may, at any instant fall above or below these bounds). The difference between the initial and steady-state power levels is the amount of undesirable noise power the processor has been able to remove from the output.

A simulation of the gradient projection algorithm (31) on the array problem was made using exactly the same data as used by the constrained LMS algorithm. The results are shown in Fig. 12. The lower part of Fig. 12 shows how the gradient-projection algorithm walks away from the constraint. Note the change in scale. If the errors of the constrained LMS algorithm (Fig. 11) were plotted on the same scale they would not be discernible. The errors of the gradient-projection method are expected to continue to grow.

The fact that the output power of the gradient-projection processor (upper curve, Fig. 12) is virtually identical to the output power of the constrained LMS processor is a result of the fact that the errors have not yet accumulated to the point of moving the constraint a significant radial distance from the origin.

VIII. Limitations and Extension

Application of the constrained LMS algorithm in some array processing problems is limited by the requirement that the non-look-direction noise voltages on the taps be uncorrelated with the look-direction signal voltages.

This restriction is a result of the fact that if the noise voltages are correlated with the signal then the processor may cancel out portions of the signal with them in spite of the constraints. If the source of correlated noise is known, its effect may be reduced by placing additional constraints to minimize the array response in its direction.

Implementation errors, i.e., deviations from the assumed electrical and spatial properties of the array (such as incorrect amplifier gains, incorrect sensor placements, or unpredicted mutual coupling between sensors) may also limit the effectiveness of the processor by permitting it to discriminate against look-direction signals while still satisfying the letter of the constraints. Injection of known test signals into the array may provide information about the signal paths that can be used to compensate, in part, for the errors.

The algorithm may be extended to a more general stochastic constrained least squares problem

$$\min E\{[d(k) - W^T X(k)]^2\} \text{ subject to } C^T W = \mathcal{F} \quad (32)$$

where $d(k)$ is a scalar variable related to the observation vector $X(k)$ and C is a general constraint matrix. The scalar $d(k)$ may be a random variable correlated with $X(k)$ or it may be a known test signal used to compensate for array errors. This would be a classical least squares problem except that the statistics of $X(k)$ and $d(k)$ are assumed unknown *a priori*. The general constrained LMS algorithm solving (32) may be derived similarly to (22) and is

$$W(0) = C(C^T C)^{-1}\mathcal{F}$$

$$W(k+1) = P\{W(k) - \mu[y(k) - d(k)]X(k)\} + F.$$

The general algorithm is applicable to constrained modeling, prediction, estimation, and control. It is discussed in [6].

IX. Conclusion

Analysis and computer simulations have confirmed the ability of the constrained LMS algorithm to adjust an array of sensors in real time to respond to a desired signal while discriminating against noise. Because of a system of constraints on weights in the array, the algorithm is shown to require no prior knowledge of the signal or noise statistics.

A geometrical presentation has shown why the constrained LMS algorithm has an ability to maintain the constraints and prevent the accumulation of quantization errors in a digital implementation. The simulation tests have confirmed the effectiveness of this error-correcting feature, in contrast with the usual uncorrected gradient-projection algorithm. The error-correcting feature and the simplicity of the algorithm make it appropriate for continuous real-time signal estimation and discriminating against noises in a possibly time-varying environment where little *a priori* information is available about the signals or noises. Time constants, steady-state performance, and a proof of convergence are derived for operation of the algorithm in a stationary environment; convergence and steady-state performance in a nonstationary environment are also shown.

A simple extension of the algorithm may be used to solve a general constrained LMS problem, which is to minimize the expected squared difference between a multidimensional filter output and a known desired signal under a set of linear equality constraints.

References

[1] L. J. Griffiths, "A simple adaptive algorithm for real-time processing in antenna arrays," *Proc. IEEE*, vol. 57, pp. 1696–1704, Oct. 1969.

[2] B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, "Adaptive antenna systems," *Proc. IEEE*, vol. 55, pp. 2143–2158, Dec. 1967.

[3] L. J. Griffiths, "Comments on 'A simple adaptive algorithm for real-time processing in antenna arrays' (Author's reply)," *Proc. IEEE* (Lett.), vol. 58, p. 798, May 1970.

[4] B. Widrow and M. E. Hoff, Jr., "Adaptive switching circuits," *IRE WESCON Conv. Rec.*, pt. 4, pp. 96–104, 1960.

[5] L. J. Griffiths, "Signal extraction using real-time adaptation of a linear multichannel filter," Stanford Electron. Lab., Stanford, Calif., Doc. SEL-60-017, Tech. Rep. TR 67881-1, Feb. 1968.

[6] O. L. Frost, III, "Adaptive least squares optimization subject to linear equality constraints," Stanford Electron. Lab., Stanford, Calif., Doc. SEL-70-055, Tech. Rep. TR 6796-2, Aug. 1970.

[7] J. B. Rosen, "The gradient projection method for nonlinear programming, pt. 1: Linear constraints," *J. Soc. Indust. Appl. Math.*, vol. 8, p. 181, Mar. 1960.

[8] R. T. Lacoss, "Adaptive combining of wideband array data for optimal reception," *IEEE Trans. Geosci. Electron.*, vol. GE-6, pp. 78–86, May 1968.

[9] A. H. Booker, C. Y. Ong, J. P. Burg, and G. D. Hair, "Multiple-constraint adaptive filtering," Texas Instruments, Sci. Services Div., Dallas, Tex., Apr. 1969.

[10] H. Kobayashi, "Iterative synthesis methods for a seismic array processor," *IEEE Trans. Geosci. Electron*; vol. GE-8, pp. 169–178, July 1970.

[11] D. G. Luenberger, *Optimization by Vector Space Methods*. New York: Wiley, 1969.

[12] I. J. Good and K. Koog, "A paradox concerning rate of information," *Informat. Contr.*, vol. 1, pp. 113–116, May 1958.

[13] A. E. Bryson, Jr., and Y. C. Ho, *Applied Optimal Control*. Waltham, Mass.: Blaisdell, 1969.

[14] W. H. Fleming, *Functions of Several Variables*. Reading, Mass.: Addison-Wesley, 1965.

[15] E. J. Kelly, Jr., and M. J. Levin, "Signal parameter estimation for seismometer arrays," Mass. Inst. Technol. Lincoln Lab. Tech. Rept. 339, Jan. 1964.

[16] A. H. Nuttall and D. W. Hyde, "A unified approach to optimum and suboptimum processing for arrays," U. S. Navy Underwater Sound Lab., New London, Conn., USL Rep. 992, Apr. 1969.

[17] G. N. Saradis, Z. J. Nikolic, and K. S. Fu, "Stochastic approximation algorithms for system identification, estimation, and decomposition of mixtures," *IEEE Trans. Sys. Sci. Cybern.*, Vol. SSC-5, pp. 8–15, Jan. 1969.

[18] P. E. Mantey and L. J. Griffiths, "Iterative least-squares algorithms for signal extraction," in *Proc. 2nd Hawaii Int. Conf. on Syst. Sci.*, pp. 767–770.

[19] T. P. Daniell, "Adaptive estimation with mutually correlated training samples," Stanford Electron. Labs., Stanford, Calif., Doc. SEL-68-083, Tech. Rep. TR 6778-4, Aug. 1968.

[20] J. L. Moschner, "Adaptive filtering with clipped input data," Stanford Electron. Labs., Stanford, Calif., Doc. SEL-70-053, Tech. Rep. TR 6796-1, June 1970.

[21] K. D. Senne, "Adaptive linear discrete-time estimation," Stanford Electron. Labs., Stanford, Calif., Doc. SEL-68-090, Tech. Rep. TR 6778-5, June 1968.

[22] ——, "New results in adaptive estimation theory," Frank J. Seiler Res. Lab., USAF Academy, Colo., Tech. Rep. SRL-TR-70-0013, Apr. 1970.

[23] J. E. Brown, III, "Adaptive estimation in nonstationary, environments," Stanford Electron. Labs., Stanford, Calif., Doc. SEL-70-056, Tech. Rep. TR 6795-1, Aug. 1970.

[24] D. T. Finkbeiner, II, *Introduction to Matrices and Linear Transformations*. San Francisco, Calif.: Freeman, 1966.