FROST, III, Otis Lamont, 1944-
ADAPTIVE LEAST SQUARES OPTIMIZATION SUBJECT
TO LINEAR EQUALITY CONSTRAINTS.

Stanford University, Ph.D., 1970
Engineering, electrical

ADAPTIVE LEAST SQUARES OPTIMIZATION SUBJECT TO LINEAR EQUALITY CONSTRAINTS

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

By

Otis Lamont Frost, III

August, 1970

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Bernard Widrow

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

Michael A Arbib

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and quality, as a dissertation for the degree of Doctor of Philosophy.

William E. Spicer

Approved for the University Committee
on the Graduate Division:

Lincoln E Moses
Dean of the Graduate Division

ACKNOWLEDGMENT

The author wishes to thank Dr. Bernard Widrow for his guidance, encouragement, and support during this research. Sincere appreciation is given to Dr. Michael Arbib for many stimulating discussions, and to Dr. William Spicer for his helpful comments and suggestions on the presentation of the material.

The author particularly wishes to thank his friends and colleagues at Stanford Electronics Laboratories, Jim Brown, Michel Installé, John Moschner, Lloyd Griffiths, Tom Daniell, Barbara Kenyon, Burwell Goode and many others, for their assistance.

Special thanks go to Judith Ann for her considerable role in the course of this research.

TABLE OF CONTENTS

I. INTRODUCTION

This paper presents a simple algorithm for minimizing a quadratic cost criterion subject to linear equality constraints. The technique, called the "Constrained-Least-Mean Squares" or "Constrained LMS" algorithm is an iterative, stochastic gradient-descent algorithm with low memory requirements. Computationally, it is simple enough that for a variety of practical problems it can be implemented in real time on a small general-purpose computer.

The algorithm is applicable to problems in least squares filtering, estimation, modeling, and others which may properly be viewed as linear-constrained quadratic optimization problems. Specific examples treated in the paper include real-time minimum-variance unbiased estimation, consistent modeling that includes known linear constraints on the model parameters, and real-time processing of data from an array of antennas or other sensors. The constrained least-mean-squares approach is particularly interesting in the estimation and array processing applications because it requires very little *a priori* information for implementation.

The rate of convergence of the algorithm is studied and its steady-state performance is compared with the optimum. A gain constant is shown to control a tradeoff between fastest convergence rate and best steady-state performance. By suitable choice of gain the steady-state performance of the algorithm can be made arbitrarily close to the performance of the optimum least-squares filter.

Previous work on unconstrained least-squares array processing was done by Griffiths [12]; his method requires knowledge of second-order signal statistics. Widrow, et al. [30] proposed a variable-criterion optimization procedure involving the use of a known training signal; this was a direct application of the original work on adaptive filters done by Widrow and Hoff [29]. Griffiths also proposed a constrained least-mean-squares processor not requiring a priori knowledge of the signal statistics [11]; a new derivation of this processor, given in Appendix A, 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 [23] for the deterministic case. Lacoss [14] and Booker [1] studied "hard"-constrained stochastic optimization in the array processing context. All three authors used "gradient projection" techniques; Rosen and Booker correctly indicate 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 is designed to avoid error accumulation while maintaing a "hard" constraint; as a result, it is able to operate continually in order to track an environment that may be slowly time-varying. Discussion of gradient-projection methods and a comparison of the error-correcting properties of the two algorithms is given in Section VII.

In the following section, the general constrained least-mean-squares problem is formulated as a theorem and the optimal solution is derived under the assumption that all the relevant statistics of the problem are known. Several corollaries applying to interesting special cases are drawn. The optimal solution is seen to be computationally difficult, requiring a number of matrix multiplications and inversions. In Section III, the computationally simple Constrained LMS algorithm is derived that converges to the optimal solution while learning the statistics of the problem. This algorithm and studies of its properties is the principal result of this thesis. Special forms of the general algorithm are used to solve particular problems. Remaining sections are concerned with geometrical interpretation of the algorithm, its performance, applications, and computer simulations.

II. CONSTRAINED LEAST-MEAN-SQUARES OPTIMIZATION

A. Notation

In this paper a vector is taken to be a column vector. The superscript T denotes transpose. The expected value of a quantity $\{Q\}$ is denoted by $E\{Q\}$ or $\overline{Q}$. The matrix of correlations between two vectors of random variables, A and B, is written $E\{AB^T\} = R_{AB}$; the vector of correlations between a vector X and a scalar d is written $E\{Xd\} = R_{Xd}$. A vector of zeros of arbitrary dimension is θ and the matrix of zeros is $\underline{\underline{0}}$

B. The General Problem and Optimal Solution

There are two purposes for this section. The first is to define the general constrained least-mean-squares problem and derive the optimal solution. This solution could be obtained directly if one knew the problem statistics beforehand. It will be shown later that the Constrained LMS algorithm converges to this solution and can be used when the problem statistics are unknown. The second purpose is to show that several interesting and important problems can be put in the framework of the general constrained LMS problem and therefore are solvable by the algorithm.

Let X be a vector of n observed data points, $X^T = (x_1, x_2, \ldots, x_n)$, that are drawn from a distribution with $E\{XX^T\} = R_{XX}$. Let d be a random variable correlated with X by an n-dimensional correlation vector R_{Xd}. In this section

R_{XX} and R_{Xd} are assumed known. Let W be an n-dimensional vector of weightings that will be applied to X to estimate d. Let the estimate of d be

$$y \triangleq W^T X , \tag{2.1}$$

and the error between d and the estimate be

$$e \triangleq d-y . \tag{2.2}$$

The constrained least-mean-squares optimization problem is to find the weight vector W_* that minimizes the expected squared error in the estimate,

$$E\{e^2\} = E\{[d-y]^2\} = E[d-W^TX]^2\} \tag{2.3}$$

subject to certain linear equality constraints on W.

The reason for placing constraints on W was suggested in the introduction and will be made clear in the applications. In general, m linear equality constraints (with $n > m$) are placed on W of the form

$$c_i^T W = f_i , \quad i = 1,2,\ldots,m , \tag{2.4}$$

where each c_i is an n-dimensional vector and each f_i is a scalar constant. This is a set of m simultaneous equations which the n components of W must satisfy, but since $m < n$, the equations do not completely determine or totally constrain W. Therefore W can be optimized, to minimize a mean square error, subect to the linear constraint (2.4). It is well known that by requiring W to satisfy $c_i^T W = f_i$ for any single i restricts W to lie in an (n-1)-dimensional hyperplane. Similarly, it is shown in Section IV that constraining

W to satisfy the m equations of (2.4) restricts W to an (n-m)-dimensional plane[†] if the vectors c_i are linearly independent. To express the constraints in matrix notation define

$$C \triangleq \begin{array}{c} \xleftrightarrow{\qquad m \qquad} \\ \left[\begin{array}{cccc} \\ c_1 & c_2 & \cdots & c_m \\ \\ \end{array}\right] \end{array} \Big\updownarrow n \; , \quad \mathcal{F} \triangleq \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix} . \tag{2.5}$$

The constraint matrix C is $(n \times m)$ with $n > m$. It will be assumed that the constraint vectors c_i are linearly independent so that by the definition of rank as the number of linearly independent columns of a matrix, C has full rank equal to m. The constraints (2.4) are now written

$$C^T W = \mathcal{F} \; . \tag{2.6}$$

The problem is summarized and the solution is given in the form of a theorem.

<u>Theorem 1.</u> (Constrained Linear Least-Mean-Squares Optimization) Let d be a random variable and X be an n-dimensional vector of random variables with known correlation matrices

[†] Other names for an "r-dimensional plane" are "linear variety" and "Linear manifold".

$$E\{XX^T\} = R_{XX} \qquad (n \times n)$$

$$E\{Xd\} = R_{Xd} \qquad (n \times 1)$$

and R_{XX} positive-definite. The optimum constrained least-mean-squares weight vector solving

$$\begin{aligned} &\text{minimize} \quad E\{[d - W^T X]^2\} \\ &\text{subject to} \quad C^T W = \mathcal{F} \end{aligned} \tag{2.7}$$

where C is an $(n \times m)$ matrix $(n > m)$ of full rank and $\mathcal{F}$ is an m-vector, is

$$W_* = [I - R_{XX}^{-1} C (C^T R_{XX}^{-1} C)^{-1} C^T] R_{XX}^{-1} R_{Xd} + R_{XX}^{-1} C (C^T R_{XX}^{-1} C)^{-1} \mathcal{F} \, . \tag{2.8}$$

The optimum constrained linear least-mean-squares estimate of d is $y = W_*^T X$.

Proof of Theorem 1.

The proof uses the method of Lagrange multipliers, which is basic to the later development of the major algorithm and another proof. A geometrical interpretation of Lagrange multipliers expressed in the context of this work is presented in Appendix E.

The cost function is

$$\begin{aligned} J(W) &= E\{[d - W^T X]^2\} \\ &= E\{d^2\} - 2\, E\{W^T X d\} + E\{W^T X X^T W\} \\ &= E\{d^2\} - 2 W^T R_{Xd} + W^T R_{XX} W \, . \end{aligned} \tag{2.9}$$

Including a factor of ½ to simplify later arithmetic, adjoin the constraint function to the cost function by a m-dimensional vector of undetermined Lagrange multipliers λ :

$$\begin{aligned} H(W) &= \tfrac{1}{2}J(W) + \lambda^T(C^TW - \mathcal{F}) \\ &= \tfrac{1}{2}[Ed^2 - 2W^TR_{Xd} + W^TR_{XX}W] + \lambda^T(C^TW - \mathcal{F}) \ . \end{aligned} \tag{2.10}$$

The necessary conditions for optimality are

$$\nabla_W H(W) = \theta \ , \tag{2.11}$$

and

$$C^TW = \mathcal{F} \ . \tag{2.12}$$

Taking the gradient of (2.10) with respect to W

$$\nabla_W H(W) = -R_{Xd} + R_{XX}W_* + C\lambda = \theta \tag{2.13}$$

and solving for the optimal weight vector

$$W_* = R_{XX}^{-1}R_{Xd} - R_{XX}^{-1}C\lambda \ , \tag{2.14}$$

where R_{XX}^{-1} exists because R_{XX} was assumed positive definite. Since W_* must satisfy the constraint (2.12)

$$C^TW_* = \mathcal{F} = C^TR_{XX}^{-1}R_{Xd} - C^TR_{XX}^{-1}C\lambda \tag{2.15}$$

and from (2.15) λ is found to be

$$\lambda = [C^TR_{XX}^{-1}C]^{-1}[C^TR_{XX}^{-1}R_{Xd} - \mathcal{F}] \ . \tag{2.16}$$

It is shown in Appendix C that the existence of $[C^TR_{XX}^{-1}C]^{-1}$ follows from the facts that R_{XX} is positive definite and

C has full rank. Substituting the last expression for the Lagrange multipliers into the expression for W_* (2.14) the result follows.

This completes the proof of Theorem 1.

C. Special Cases

A well-known special case of Theorem 1 is the unconstrained least-squares problem.

Corollary 1.1. (Least-Mean-Square Error -- Wiener) The optimum set of weights W_* solving the problem defined by Theorem 1 without constraints, i.e.,

$$\text{minimize} \quad E\{[d - W^T X]^2\} \tag{2.17}$$

with

$$E\{XX^T\} = R_{XX}$$

$$E\{Xd\} = R_{Xd}$$

is

$$W_{*_1} = R_{XX}^{-1} R_{Xd} \ . \tag{2.18}$$

And the best unconstrained estimate of d is $y = W_{*_1}^T X$.

Proof of Corollary 1.1.

Let the constraint matrix C vanish in Theorem 1. See especially Eq. (2.14) of the proof.

This completes the proof of Corollary 1.1.

A second well-known problem that can be formulated as a special case of Theorem 1 is the distortionless least-mean-squares estimation problem that was solved by Gauss.

Corollary 1.2. (Least-Mean-Squares Distortionless Estimate -- Gauss, Markov) Let the data vector X be of the form

$$X = CB + N , \tag{2.19}$$

where C is a known $(n \times m)$ matrix of rank m, and B is an unknown m-dimensional vector with $B^T = \underline{b_1 b_2 \ldots b_m}$. B may be a vector of random variables with unknown mean (so $E\{B\} = \overline{B}$), or it may be a vector of unknown parameters, in which case $E\{B\} = \overline{B} = B$. N is an unknown n-dimensional vector of random variables considered as noise. B and N are uncorrelated, with

$$\begin{aligned} E\{BB^T\} &= R_{BB} && (m \times m) \\ E\{N\} &= \theta && (n \times 1) \\ E\{NN^T\} &= R_{NN} && (n \times n) \\ E\{BN^T\} &= \underline{\underline{0}} && (m \times n) \, , \end{aligned}$$

and R_{NN} is positive definite.

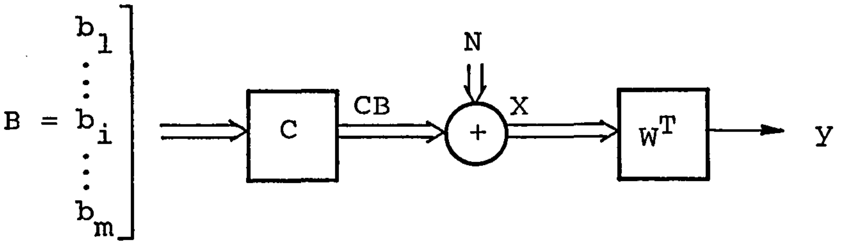

Fig. 2.1. The estimation problem of Corollary 1.2. Thick lines indicate vector-valued quantities. W is chosen so that y is an estimate of the i^{th} component of B, b_i .

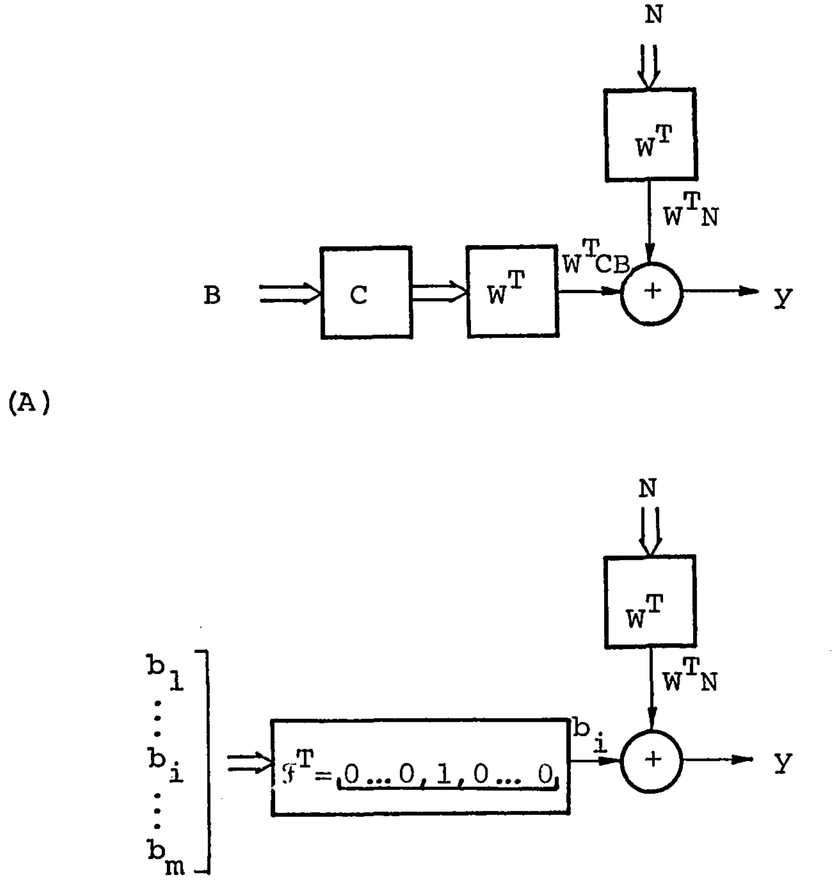

Fig. 2.2. Manipulation of the flow charts from Fig. 2.1 yields (A). Constraining $W^TC = \mathfrak{I}^T$ yields (B), showing that the constraint puts a unity transfer function on b_i and that $y = b_i + W^TN$.

Thus

$$R_{XX} = CR_{BB}C^T + R_{NN} .$$

The problem is to make a linear least-squares estimate of b_i, say $y = W^TX$, that is unbiased (see Fig. 2.1). We wish the estimate of b_i to be corrupted only by the minimum amount of zero-mean noise. The optimum weight vector solving the problem

$$\begin{aligned} &\text{minimize} \quad E\{[b_i - W^TX]^2\} \\ &\text{subject to} \quad E\{W^TX - \overline{b}_i\} = 0 \end{aligned} \tag{2.20}$$

is

$$W_{*_2} = R_{XX}^{-1}C[C^TR_{XX}^{-1}C]^{-1}\mathcal{F} \tag{2.21}$$

where

$$\mathcal{F} = \left.\begin{matrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{matrix}\right] \begin{matrix} \\ \\ \\ \\ \leftarrow i^{th} \text{ position} \\ \\ \\ \\ \end{matrix} . \tag{2.22}$$

and the best unbiased estimate of b_i is $y = W_{*_2}^T X$.

<u>Proof of Corollary 1.2.</u>

The problem (2.20) is put into the form of the problem solved by Theorem 1. Observe that $b_i = \mathcal{F}^TB$. Using (2.19) $X = CB + N$, and the fact that N is zero mean, we have

$$E\{W^TX - \overline{b}_i\} = E\{W^TCB + W^TN - \overline{b}_i\} = E\{W^TCB - \overline{b}_i\} .$$

Now if we require $C^TW = \mathcal{F}$ then

$$E\{W^T X - \overline{b}_i\} = E\{\mathcal{F}^T B - \overline{b}_i\} = 0 \qquad (2.23)$$

and the constraint of (2.20) is satisfied. This is reasonable since the constraint applies a unit transfer function to b_i (see Fig. 2.2).

Further, with the constraint $C^T W = \mathcal{F}$ in force $y = W^T X = W^T CB + W^T N = b_i + W^T N$ and the cost function becomes

$$\begin{aligned} E\{[b_i - y]^2\} &= E\{b_i^2\} - 2E\{b_i y\} + E\{y^2\} \\ &= E\{b_i^2\} - 2E\{b_i(b_i + W^T N)\} + E\{y^2\} \\ &= E\{y^2\} - E\{b_i^2\} \;. \end{aligned} \qquad (2.24)$$

Because $E\{b_i^2\}$ is a constant, the weight vector that minimizes $E\{y^2\}$ also minimizes $E\{y^2\} - E\{b_i^2\}$, so the problem (2.20) reduces to

$$\begin{aligned} &\text{minimize} \quad E\{y^2\} = E\{[W^T X]^2\} \\ &\text{subject to} \quad C^T W = \mathcal{F} \quad , \end{aligned} \qquad (2.25)$$

where C is defined in (2.19), and $\mathcal{F}$ in (2.22). This is a special case of the problem of Theorem 1 with $d = 0$. Since $d = 0$, $E\{Xd\} = R_{Xd} = \theta$ and so the first term of (2.8) vanishes and the optimal solution becomes the second term.

<u>This completes the proof of Corollary 1.2.</u>

In some cases the unbiased estimator may not be the most desirable. Suppose that (as in the array-processing problem discussed later) B is the sum of two vectors

$$B = \mathcal{S} + A \;, \qquad (2.26)$$

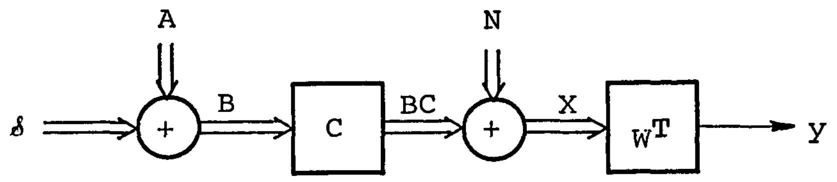

Fig. 2.3. B is a given structure for Corollary 1.3.
B is the sum of signals $\mathcal{S}$ plus noise A .

where $\mathcal{S}$ and A are m-dimensional vectors of random variables which may be statistically correlated. $\mathcal{S}$ is to be thought of as a vector of "signals", one of which we wish to estimate, say s_i . A is to be considered as a vector of additive noise. Note that A and N are both "noise" vectors of different dimension (see Fig. 2.3).

Since the "unbiased estimator" of Corollary 1.2 forms an estimate of b_i , which is equal to s_i plus a_i , it may not be satisfactory as an estimator of s_i alone.

Another approach is to recall from Corollary 1.2 that by suitable choice of the constraints a vector $\mathcal{F}$ can be applied directly to B . Therefore a "filter" vector $\mathcal{F}$ (which may be different from (2.22)) may be designed to use the correlation among the components of $\mathcal{S}$ and the (hopefully different) correlations among the components of A , so that when $\mathcal{F}$ is applied to B it may enhance s_i in the output and discriminate against A . This is exactly analogous to the use of a filter in the frequency domain to pass signals and discriminate against noises.

In the following, it is assumed that $\mathcal{F}$ is a vector chosen by the user. The best choice of $\mathcal{F}$ is a topic with which we do not wish to get deeply involved. An example of a choice of $\mathcal{F}$ is given in Example 3, Section VI. If the weight vector W is constrained to satisfy $C^TW = \mathcal{F}$, then the output is

$$y = W^TX = W^TCB + W^TN = \mathcal{F}^TB + W^TN \ , \qquad (2.27)$$

and output power is

$$E\{y^2\} = \mathscr{S}^T R_{BB}\mathscr{S} + W^T R_{NN} W \ . \tag{2.28}$$

Because B and N are uncorrelated there is no cross term and so long as W satisfies the constraint any permissible variation in W affects only the power of the noise in the output. Thus the "degrees of freedom" of W not constrained by $C^T W = \mathscr{S}$ may be used to minimize the excess noise power in the estimate of s_i .

With the preceding motivation, the problem is set up as a special case of Theorem 1.

Corollary 1.3. (Least-Mean-Squares Filtered Estimate)

Let X be a known n-dimensional vector of observations of the form

$$X = CB + N \ ,$$

where C is a known $(n \times m)$ matrix with $(n > m)$. B and N are unknown vectors of random variables with dimensions m and n respectively. Let B be of the form

$$B = \mathscr{S} + A \ ,$$

where $\mathscr{S}$ and A are m-dimensional vectors of random variables. We wish to form an estimate of the i^{th} element of $\mathscr{S}$, s_i .

$$\begin{aligned} E\{N\} &= \theta \\ E\{NN^T\} &= R_{NN} \\ E\{BN^T\} &= \underline{\underline{0}} \\ E\{XX^T\} &= R_{XX} \ . \end{aligned}$$

Let $\mathcal{F}$ be a given m-dimensional filter vector. The least-mean-squares filtered estimator is the weight vector solving the problem

$$\begin{aligned} &\text{minimize} \quad E\{y^2\} = E\{[W^TX]^2\} \\ &\text{subject to} \quad C^TW = \mathcal{F} \end{aligned} \tag{2.29}$$

and is

$$W_{*_3} = R_{XX}^{-1}C[C^TR_{XX}^{-1}C]^{-1}\mathcal{F} \tag{2.30}$$

The best $\mathcal{F}$-filtered estimate of s_i is $y = W_{*_3}^T X$.

<u>Proof of Corollary 1.3.</u>

Proof follows directly from Theorem 1, with d=0 and hence $R_{Xd} = \theta$. The fact that $y = W_{*_3}^T X$ is an estimate for s_i follows from the above discussion.

<u>This completes the Proof of Corollary 1.3.</u>

<u>Remark</u>: If $\mathcal{F}$ is chosen as in (2.22) (one unit entry and the rest zeros) the solution is the same as the solution of Corollary 1.2.

III. THE ADAPTIVE ALGORITHM

A. The Unknown Statistics Problem

Suppose now that the correlation matrices R_{XX} and R_{Xd} required by Theorem 1 are not known *a priori*. Instead a sequence of observation vectors $\{X(0),X(1),\ldots,X(k),\ldots\}$ is presented, each vector drawn independently from a quasi-stationary ergodic distribution with autocorrelation R_{XX}. A sequence of random variables $\{d(0),d(1),\ldots,d(k),\ldots\}$ which are related to the X's by an unknown correlation vector R_{Xd} is also presented. We wish to minimize the constrained mean square error of the problem of Theorem 1.

An obvious solution is to make estimates of the unknown correlation matrices from observations, e.g.,

$$\hat{R}_{XX}(k) = \alpha\hat{R}_{XX}(k-1) + (1-\alpha)X(k-1)X^T(k-1) \ ,$$

and

$$\hat{R}_{Xd}(k) = \alpha\hat{R}_{Xd}(k-1) + (1-\alpha)X(k-1)d(k-1) \ ,$$

$$0 < \alpha < 1 \ ,$$

and insert these estimates into the expression for the optimal weight vector given by Theorem 1, Eq. (2.8). Inspection of (2.8) shows that because of the number of matrix multiplications and inversions involved, a great deal of computation is required at each iteration by this approach, ultimately limiting the rate at which estimates can be made and the dimensionality of a system of given cost. See Appendix F for an example of the performance of this approach.

The next section describes a computationally simple procedure (the Constrained-LMS algorithm) that converges to the weight vector W_* that solves the problem posed by Theorem 1 without prior knowledge of the correlation matrix R_{XX}. Further, if $d(k)$ is available for training, or is not required for the solution (as in Corollaries 1.2 and 1.3) then the algorithm does not require knowledge of R_{Xd}.

B. Derivation

The Constrained-LMS algorithm is based on a constrained gradient descent, satisfying $C^T W = \mathcal{F}$ at all times while iterating to find a weight vector minimizing the cost function

$$J(W) = \frac{1}{2}E\{[d(k) - W^T X(k)]^2\} = \frac{1}{2}[Ed^2 - 2W^T R_{Xd} + W^T R_{XX} W] \ .$$

For motivation of the derivation, temporarily suppose that R_{XX} and R_{Xd} are known. As in the proof of Theorem 1, form the function $H(W)$ by adjoining the constraint to the cost function by a m-dimensional vector of Lagrange multipliers λ :

$$H(W) = \frac{1}{2}[Ed^2 - 2W^T R_{Xd} + W^T R_{XX} W] + \lambda^T [C^T W - \mathcal{F}] \ . \qquad (3.1)$$

As in Theorem 1, we wish to find a weight vector W_* such that the gradient of H at W_* is θ and W_* satisfies $C^T W = \mathcal{F}$. The gradient descent is initialized by choosing a weight vector $W(0)$ that satisfies the constraint.

The gradient of H with respect to W is

$$\nabla_W H = R_{XX}W - R_{Xd} + C\lambda \ . \tag{3.2}$$

At each iteration the weight vector is moved in the direction of the negative gradient. (Note: a move in the direction of the positive gradient tends to increase a cost function.) The length of the step is proportional to the magnitude of the gradient and scaled by a gain factor μ . At the k^{th} iteration the next weight vector would be

$$\begin{aligned} W(k+1) &= W(k) - \mu\nabla_W H(k) \\ &= W(k) - \mu[R_{XX}W(k) - R_{Xd} + C\lambda(k)] \ . \end{aligned} \tag{3.3}$$

The constrained gradient $[R_{XX}W(k) - R_{Xd} + C\lambda(k)]$ is the unconstrained gradient

$$\nabla_W J = R_{XX}W(k) - R_{Xd} \ , \tag{3.4}$$

plus the term $C\lambda(k)$. As noted in Appendix E and later in Section IV, the vector $C\lambda(k)$ is orthogonal to the constraint. By proper choice of $\lambda(k)$ the component of the unconstrained gradient normal to the constraint (and hence deviating from it) can be exactly cancelled. Thus the Lagrange multipliers are chosen by requiring W(k+1) to satisfy the constraint $\mathcal{F} = C^T W$:

$$\mathcal{F} = C^T W(k+1) = C^T W(k) - \mu C^T R_{XX} W(k) + \mu C^T R_{Xd} - \mu C^T C\lambda(k) \ , \tag{3.5}$$

and solving for the Lagrange multipliers for the k^{th} iteration,

$$\lambda(k) = (C^TC)^{-1}C^TR_{XX}W(k) - \frac{1}{\mu}(C^TC)^{-1}C^TW(k) - (C^TC)^{-1}C^TR_{Xd} \ , \tag{3.6}$$

where it is shown in Appendix C that the existence of $(C^TC)^{-1}$ follows from the fact that C has full rank. Inserting the Lagrange multipliers of (3.6) into the iterative equation (3.3) we have

$$W(k+1) = W(k) - \mu[I - C(C^TC)^{-1}C^T][R_{XX}W(k) - R_{Xd}] + C(C^TC)^{-1}[\mathcal{F} - C^TW(k)] \ . \tag{3.7}$$

The algorithm may be rewritten, defining the n-dimensional vector

$$F = C(C^TC)^{-1}\mathcal{F} \tag{3.8}$$

$$W(k+1) = [I - C(C^TC)^{-1}C^T][W(k) - \mu R_{XX}W(k) + \mu R_{Xd}] + F. \tag{3.9}$$

Equation (3.9) is a deterministic gradient-descent algorithm that converges to the optimal weight vector W_* of Theorem 1 for a suitably small choice of the gain μ (proof given in Section VI). However, it requires knowledge of the correlation matrices R_{XX} and R_{Xd}, which in this study are assumed unavailable a priori. But recall $R_{XX} = E\{X(k)X^T(k)\}$ and $R_{Xd} = E\{X(k)d(k)\}$, so an easily-available and simple approximation for R_{XX} at the k^{th} iteration is the outer product of the observation vector with itself: $X(k)X^T(k)$; likewise if $d(k)$ is available a simple approximation for R_{Xd} at the k^{th} iteration is $X(k)d(k)$.[†] This substitution gives the stochastic algorithm

$$W(k+1) = [I-C(C^TC)^{-1}C^T]\,[W(k) - \mu X(k)X^T(k)W(k) + \mu X(k)d(k)] + F\,, \tag{3.10}$$

which can be simplified using $y(k) = X^T(k)W(k)$ and $e(k) = d(k) - y(k)$ to

$$W(k+1) = [I-C(C^TC)^{-1}C^T][W(k) + \mu e(k)X(k)] + F\ . \tag{3.11}$$

Equation (3.11) is the Constrained-LMS algorithm. It is a stochastic gradient-descent algorithm satisfying the constraint that $C^TW(k) = \mathcal{F}$ at all times (check: $C^TW(k+1) = \mathcal{F}$). At each iteration it requires only the observations $X(k)$ (and $d(k)$ if required). No a priori knowledge of R_{XX} or R_{Xd} is needed. The most comples operation is the multiplication of a constant matrix times a vector, which is a substantial savings over the matrix multiplications and inversions required (either explicitly or implicitly) by a direct implementation of the optimal equations.

The algorithm was derived heuristically. Its convergence to the optimum, rate of convergence, and steady-state

[†] As mentioned previously, better, but more complex, estimates for R_{XX} are available, such as $\frac{1}{k+1}\Sigma X(i)X^T(i)$. See Saradis, et al. [24] for use of this estimate in another algorithm; and Mantey and Griffiths [18] for a closely related estimate. For discussion and use of simpler estimates, see Moschner [20], Lender [15], and Nuttall [21]. The use of $X(k)X^T(k)$ here is a compromise between algorithm complexity and performance and may be changed if desired.

performance are shown in Section V. The next section develops the theory of constrained gradient descent from a geometrical viewpoint.

IV. A GEOMETRICAL VIEW OF THE ALGORITHM

A geometrical interpretation of the Constrained-LMS algorithm (3.11) is now given. Results will be found that permit an easier and more intuitive derivation of the properties of the algorithm than would otherwise be possible. Readers interested in applications may skip to Section VI.

We start from basic definitions.

Definition (Subspace) Let α and β be real scalar numbers. A nonempty subset S of a vector space is called a subspace if every vector of the form $\alpha V + \beta W$ is in S whenever V and W are both in S .

Since a subspace must contain at least one element W , it must also include the zero vector θ because $0 \cdot W = \theta$. Thus every subspace includes the origin.

Let Σ be the set of all n-dimensional weight vectors satisfying the homogeneous form of the constraint equation $C^T W = \theta$.

$$\Sigma \triangleq \{W : C^T W = \theta\} \quad . \tag{4.1}$$

Then we have

Geometrical Property 1. The set $\Sigma = \{W : C^T W = \theta\}$ defined by the homogeneous form of the constraint equation is a subspace.

Proof of Geometrical Property 1.

Let V and Z be vectors in Σ . They must satisfy the equations $C^T V = \theta$ and $C^T Z = \theta$. Therefore for any constants α and β , the vector $Y = \alpha V + \beta Z$ also satisfies $C^T Y = \theta$, so the set Σ is a subspace.

This completes the proof of Geometrical Property 1.

Definition (Linear Variety) A linear variety is a translation of a subspace.

A linear variety L may be expressed by the set equation $L = S + U$, where S is a subspace and U is any vector in the linear variety. The linear variety L is said to be parallel to the subspace S .

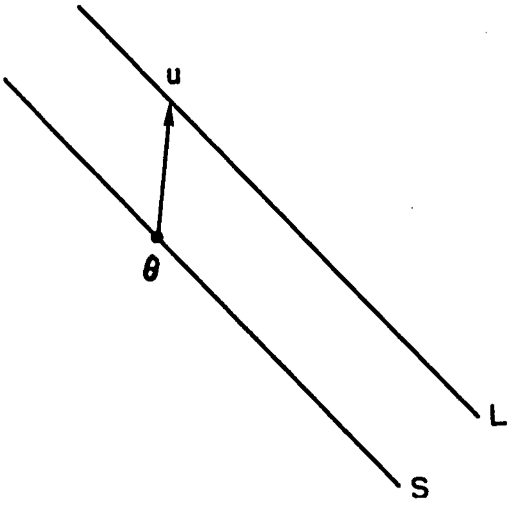

Fig. 4.1. A linear variety and its subspace.

Let Γ be the set of all weight vectors W satisfying the constraint $C^T W = \mathcal{F}$.

$$\Gamma \triangleq \{W : C^T W = \mathcal{F}\} . \tag{4.2}$$

This definition leads to

Geometrical Property 2. The set $\Gamma = \{W : C^T W = \mathcal{F}\}$ defined by the constraint equation is a linear variety parallel to the subspace Σ .

Proof of Geometrical Property 2.

We must show that a vector W is in Γ if and only if it can be written as the sum of a vector in Σ and a translation vector.

(IF) Let the translation vector U be in Γ and Z be any vector in Σ . Then $C^T U = \mathcal{F}$ and $C^T Z = \theta$. Thus $W = Z + U$ satisfies $C^T W = C^T (Z + U) = \mathcal{F}$, so if U is in Γ the sum of any vector in Σ and U is in Γ .

(ONLY IF) Now suppose a vector W in Γ satisfied $C^T W = \mathcal{F}$ but could not be written as the sum of U and a vector in Σ . Then it follows that the vector $W - U$ could not be written as a vector in Σ . But $C^T (W - U) = \mathcal{F} - \mathcal{F} = \theta$ so $W - U$ is in Σ . Contradiction.

This completes the proof of Geometrical Property 2.

Geometrical Property 3. The shortest vector from the origin to the linear variety Γ is the vector $F = C(C^T C)^{-1} \mathcal{F}$, which is orthogonal to Γ .

Proof of Geometrical Property 3.

We want to find the vector W minimizing $\|W\|^2 = W^T W$ while satisfying $C^T W = \mathcal{F}$. Use the method of Lagrange multipliers. Form the function H(W) by adjoining the constraint to the cost criterion:

$$H(W) = \frac{1}{2}[W^TW] + \lambda^T(C^TW - \mathfrak{F}) .$$

A necessary condition for optimality is

$$\nabla_W H = W + C\lambda = \theta ,$$

or

$$W = -C\lambda .$$

Requiring W to satisfy the constraint

$$C^TW = \mathfrak{F}$$

we have

$$-C^TC\lambda = \mathfrak{F} .$$

Solving for λ

$$\lambda = -(C^TC)^{-1}\mathfrak{F} ,$$

and inserting this into the expression for W above

$$W = C(C^TC)^{-1}\mathfrak{F} .$$

This is the vector F appearing in the algorithm (3.11) and defined in (3.8). As a check that F is in Γ note that $C^TF = C^TC(C^TC)^{-1}\mathfrak{F} = \mathfrak{F}$.

We wish to show F is orthogonal to Γ. Vectors parallel to the linear variety itself are the vectors of the parallel subspace Σ. Any vector Z in Σ

is orthogonal to $F = C(C^TC)^{-1}\mathfrak{f}$ since Z satisfies $C^TZ = \theta$ and so the inner product $F^TZ = \mathfrak{f}^T(C^TC)^{-1}C^TZ = 0$.

This completes the proof of Geometrical Property 3.

Note from the above proof that any vector of the form $C\gamma$, where γ is an m-vector, is orthogonal to the constraint variety Γ .

Geometrical properties 1- 3 are illustrated in Fig. 4.2.

The $(n \times n)$ matrix appearing between brackets in the algorithm (3.11) has an interesting geometrical interpretation. Call the matrix P .

$$P \triangleq I - C(C^TC)^{-1}C^T \; . \tag{4.3}$$

The following definition appears in Luenberger [16]:

Definition. Let a vector W have a unique representation as the sum of two vectors, one from subspace Σ and the other from the subspace $\Sigma_\perp$ perpendicular to Σ . Thus let $W = W_\parallel + W_\perp$, where $W_\parallel \in \Sigma$, $W_\perp \in \Sigma_\perp$. The operator $\mathscr{P}$ defined by $\mathscr{P}W = W_\parallel$ is called the projection operator onto Σ .

In other words, a projection operator acts as an identity operator on components in Σ and as a zero operator on components in $\Sigma_\perp$.

Geometrical Property 4. P is a projection operator onto Σ .

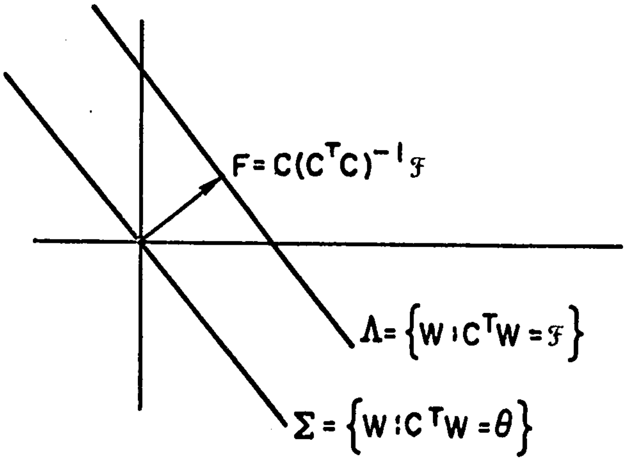
$F = C(C^TC)^{-1}\mathcal{F}$
$\Lambda = \{W : C^TW = \mathcal{F}\}$
$\Sigma = \{W : C^TW = \theta\}$

Fig. 4.2. The linear variety and subspace defined by the constraint.

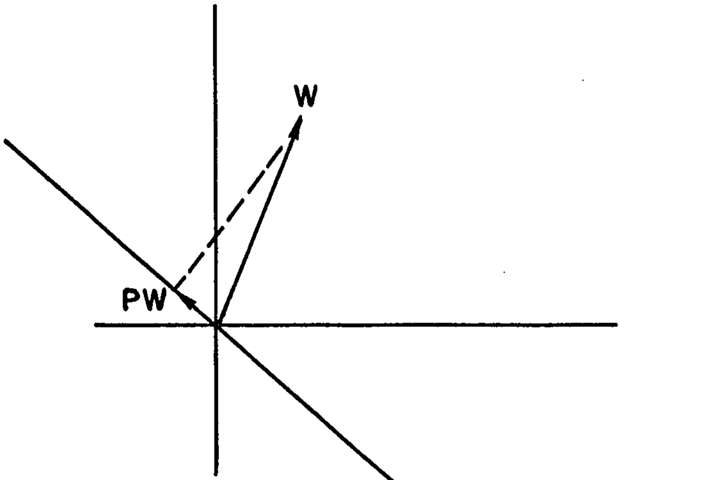

Fig. 4.3. P projects vectors onto Σ .

Proof of Geometrical Property 4.

Any weight vector W can be represented as

$$W = (W - PW) + PW .$$

The vector $PW = [I - C(C^T C)^{-1} C^T]W$ is in Σ since $C^T(PW) = C^T[I - C(C^T C)^{-1} C^T]W = [C^T - C^T]W = \theta$. The vector $(W - PW) = C(C^T C)^{-1} C^T W$ is in $\Sigma_\perp$. This is true because by definition of Σ for all vectors $Z \epsilon \Sigma$, $C^T Z = \theta$; then every vector Z in Σ is orthogonal to $(W - PW)$ since $Z^T C(C^T C)^{-1} C^T W = \theta^T (C^T C)^{-1} C^T W = \theta$. Therefore we may make the identifications: $(W - PW) = W_\perp \epsilon \Sigma_\perp$; $PW = W_\parallel \epsilon \Sigma$; and $W = W_\parallel + W_\perp$, where $W_\parallel$ and $W_\perp$ satisfy the terms of the definition. By the second identification, P is the projection operator onto Σ .

This completes the proof of Geometrical Property 4.

The geometrical interpretation of P is shown in Fig. 4.3.

The algorithm (3.11) may be rewritten in terms of the projection operator:

$$W(k+1) = P[W(k) + \mu e(k)X(k)] + F . \qquad (4.4)$$

It should be mentioned that the vector $-e(k)X(k)$ is an estimate of the unconstrained gradient $\nabla_W J$. The unconstrained gradient, given in (3.4), is $R_{XX}W(k) - R_{Xd}$. Replacing R_{XX} by $X(k)X^T(k)$ and R_{Xd} by $X(k)d(k)$ results in $X(k)X^T(k)W(k) - X(k)d(k) = -e(k)X(k)$, where

$e(k) = d(k) - X^T(k)W(k)$. The algorithm is now considered as a whole.

The algorithm attempts to minimize the cost function $E\{[d(k) - W^TX(k)]^2\}$ by iterating to the optimal weight vector W_* along the constraint. Figure 4.4 shows the position of a hypothetical adaptive weight vector at iteration k and the position of the optimal weight vector.

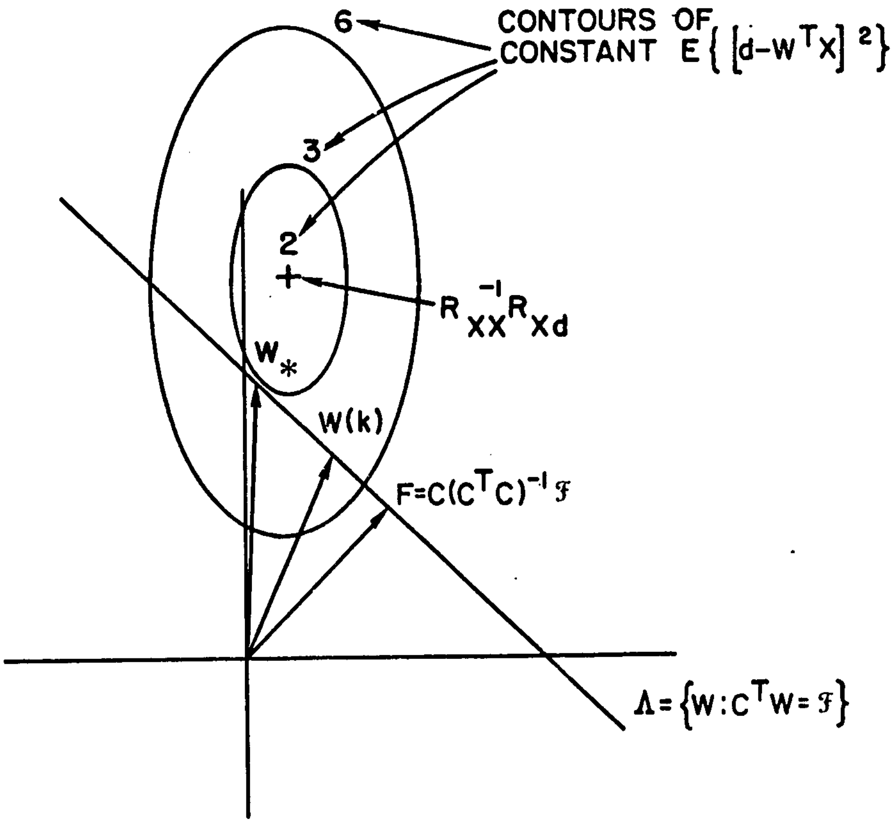
6
CONTOURS OF CONSTANT $E\{[d-W^TX]^2\}$
3
2
$R_{XX}^{-1}R_{Xd}$
W_*
$W(k)$
$F=C(C^TC)^{-1}\mathcal{F}$
$\Lambda=\{W:C^TW=\mathcal{F}\}$

Fig. 4.4. Position of the adaptive weight vector $W(k)$ at the k^{th} iteration and the optimal constrained weight vector W_* .

$$W_* = [I - R_{XX}^{-1}C(C^TR_{XX}^{-1}C)^{-1}C^TR_{XX}^{-1}R_{Xd} + R_{XX}^{-1}C(C^TR_{XX}^{-1}C)^{-1}\mathcal{F} .$$

The operation of the Constrained-LMS algorithm (4.4) is shown in Fig. 4.5. In this example, the unconstrained negative gradient estimate $e(k)X(k)$ is scaled by μ and added to the current weight vector. The resulting vector is projected onto the subspace Σ , producing a vector parallel to the constraint variety Λ . This vector is translated out to the constraint surface by adding it to F , forming the new weight vector $W(k+1)$ satisfying the constraint.

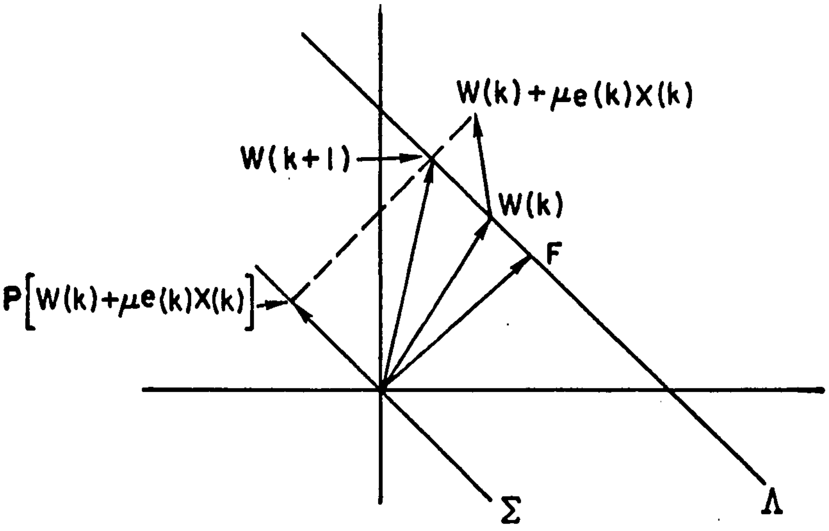

Fig. 4.5. Operation of the Constrained-LMS algorithm.
$W(k+1) = P[W(k)+\mu e(k)X(k)] + F$.

It is now shown that any difference vector between two vectors satisfying the constraint must lie in Σ (see Fig. 4.6). An identity that will be useful in the next section is given.

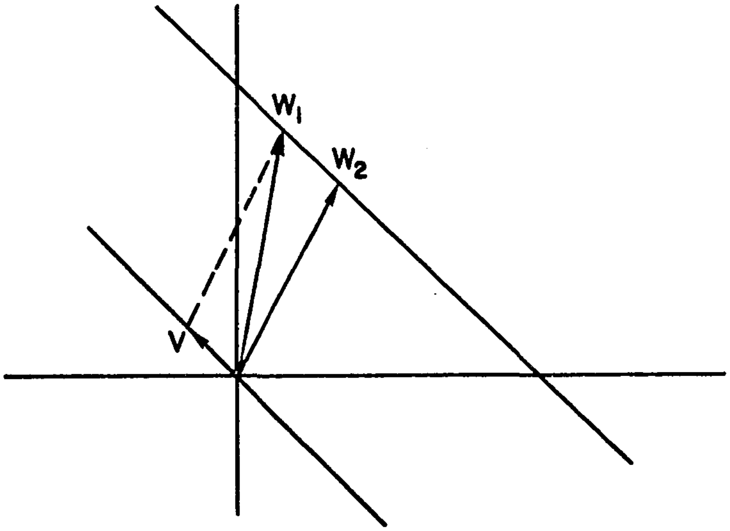

Fig. 4.6. The difference between two vectors satisfying the constraint is in the subspace Σ .

Geometrical Property 5. Let W_1 and W_2 be in Γ and let their difference be $V = W_1 - W_2$. Then V is in the subspace Σ and $PV = V$.

Proof of Geometrical Property 5.

Since W_1 and W_2 are in Γ , $C^T V = C^T W_1 - C^T W_2 = \mathcal{F} - \mathcal{F} = \theta$, so V is in Σ . By definition of a projection operator, if $V \in \Sigma$ then $PV = V$. Algebraically, $PV = [I - C(C^T C)^{-1} C^T] V = V + \theta = V$.

This completes the proof of Geometrical Property 5.

Note also that P is symmetric and idempotent, i.e.,

$$P^T = P \, , \tag{4.5}$$

and

$$P^2 = P \, . \tag{4.6}$$

These are verified by carrying out the operations. The idempotence relation (4.6) for a matrix that is, in general, neither the zero nor the identity operator is interesting because it is impossible in the scalar case. It is a result of the fact (not proven here) that P has only zero and unity eigenvalues.

V. PERFORMANCE

In Part A of this section it is shown that the mean adaptive Constrained-LMS weight vector converges to the optimal constrained weight vector of Theorem 1. Rates of convergence along the eigenvectors of the matrix $PR_{XX}P$ are given. In part B it is shown that the difference in steady-state performance between the algorithm and the optimal estimator can be made arbitrarily small by decreasing the adaptive gain constant μ .

A. Convergence in Mean to the Optimum and Rate of Convergence

The algorithm (4.4) is repeated here in a more convenient form:

$$W(k+1) = P[W(k) - \mu X(k)X^T(k)W(k) + \mu X(k)d(k)] + F \ . \qquad (5.1)$$

Note that the weight vector $W(k)$ is a function of $W(0)$, $\{X(k-1),X(k-2),\ldots,X(0)\}$ and $\{d(k-1),d(k-2),\ldots,d(0)\}$. It was assumed at the beginning of Section III that the observation vectors X are independent,[†] so $X(k)$ is independent of $W(k)$. Taking the expected value of both sides of (5.1) we have an iterative equation in the mean value of the Constrained-LMS weight vector

$$EW(k+1) = P[EW(k) - \mu R_{XX}EW(k) + \mu R_{Xd}] + F \ . \qquad (5.2)$$

[†] This is believed to be an overly-restrictive assumption but greatly simplifies the analysis. For a special case of the algorithm (no constraints), Daniell [6] has shown ϵ-convergence assuming that the X's are only asymptotically independent.

Convergence of the mean is easily established using identities for expressing F and R_{Xd} in terms of the optimal weight vector:

$$F = [I - P]W_* \;, \tag{5.3}$$

$$R_{Xd} = PR_{XX}W_* \;, \tag{5.4}$$

both of which are verified directly using (2.4), (4.3), and (3.8). Let $V(k+1)$ be the difference between the mean adaptive weight vector at iteration $k+1$ and the optimal weight vector:

$$V(k+1) \triangleq EW(k+1) - W_* \;. \tag{5.5}$$

From (5.2)-(5.5) an equation for the difference process may be constructed:

$$\begin{aligned} V(k+1) &= P[EW(k) - \mu R_{XX}EW(k) + \mu PR_{XX}W_*] + [I - P]W_* - W_* \\ &= PV(k) - \mu PR_{XX}V(k) \;. \end{aligned} \tag{5.6}$$

Using $PV = (VP)^T = V$ from Geometrical Property 5 and (4.5) obtain

$$\begin{aligned} V(k+1) &= [I - \mu PR_{XX}P]V(k) \\ &= [I - \mu PR_{XX}P]^{k+1}V(0) \;. \end{aligned} \tag{5.7}$$

The matrix $PR_{XX}P$ is the correlation matrix of projected observations, i.e., $E\{(PX)(PX)^T\}$. The non-zero eigenvalues of this matrix are extremely important in determining both the convergence rate of the algorithm and its steady-state performance relative to the optimum. The matrix being $(n \times n)$ and symmetric is diagonalizable into n orthogonal eigenvectors. It is shown in Appendix C that m of the eigenvectors of $PR_{XX}P$ lie entirely outside the subspace Σ and have zero eigenvalues; the other $(n-m)$ eigenvectors lie entirely within Σ and have strictly non-zero eigenvalues. All of the "action" is in the subspace Σ .

Call the $(n-m)$ non-zero eigenvalues of $PR_{XX}P$ $\sigma_i, i=1,2,\ldots,(n-m)$, and call the n (non-zero) eigenvalues of R_{XX} $\lambda_i, i=1,2,\ldots,n$. To get a feeling for the relationship between the σ's and the λ's , it is proven in Appendix C that the non-zero eigenvalues of $PR_{XX}P$ all fall between the largest and smallest eigenvalues of R_{XX} , that is, for $1 \leq i \leq (n-m)$

$$\lambda_{\min} \leq \sigma_{\min} \leq \sigma_i \leq \sigma_{\max} \leq \lambda_{\max} , \qquad (5.8)$$

where the subscripts min and max denote respectively the smallest and largest members of a set.

Since $V(0)$ is the difference between two vectors satisfying the constraint (5.5), from Geometrical Property 5 $V(0)$ lies entirely within the subspace Σ and may therefore be expressed as a linear combination of eigenvectors of $PR_{XX}P$ corresponding to non-zero eigenvalues. If $V(0)$ is equal to an eigenvector of $PR_{XX}P$, e_i with eigenvalue σ_i then

$$\begin{aligned} V(k+1) &= [I - \mu P R_{XX} P]^{k+1} e_i \\ &= [1 - \mu\sigma_i]^{k+1} e_i \ . \end{aligned} \qquad (5.9)$$

Thus the convergence along any eigenvector of $PR_{XX}P$ is geometric with geometric ratio $[1 - \mu\sigma_i]$ and associated time constant

$$\tau_i = -1/\ln(1 - \mu\sigma_i) \cong 1/\mu\sigma_i \ , \qquad (5.10)$$

where the approximation is valid for $\mu\sigma_i << 1$. It is clear then that if μ is chosen so that

$$0 < \mu < 1/\sigma_{max} \ , \qquad (5.11)$$

then the euclidean norm of the difference vector is bounded between two ever-decreasing geometric progressions

$$[1 - \mu\sigma_{max}]^{k+1} \|V(0)\| \leq \|V(k+1)\| \leq [1 - \mu\sigma_{min}]^{k+1} \|V(0)\| \qquad (5.12)$$

and the expected value of the weight vector converges to the optimum with time constants given by (5.10) if the initial difference is finite.

We emphasize that convergence of the mean shown here is

$$\lim_{k\to\infty} \|EW(k) - W_*\| = 0 \ . \qquad (5.13)$$

B. Steady-state Performance Compared to Optimum

In this subsection the performance of the Constrained-LMS algorithm is compared with the optimum of Theorem 1 after transients have become negligible.

To allow the Constrained-LMS algorithm to operate in quasi-stationary (i.e., slowly time-varying) environments, the adaptive gain μ remains constant during the application of the algorithm. (In stochastic approximation schemes the gain is usually allowed to go to zero as time passes.) As a result of continually adapting, the weight vector has a non-zero variance about its optimal value. In a stationary noise field, the effect of variations about the optimum weight vector is to add a slight additional cost in excess of that achievable by the optimum. (See Brown [2] for results on time-varying noise fields.)

The excess cost normalized by the optimum cost level is a dimensionless quantity called "misadjustment" by Widrow [28] and is a measure of how closely the algorithm's performance achieves the optimal performance. Steady-state misadjustment is

$$M(\mu) = \lim_{k \to 0} \frac{\begin{bmatrix} \text{Cost of} \\ \text{Adaptive Filter} \\ \text{at time } k \end{bmatrix} - \begin{bmatrix} \text{Cost of} \\ \text{Optimal Filter} \end{bmatrix}}{\begin{bmatrix} \text{Cost of} \\ \text{Optimal Filter} \end{bmatrix}} .$$

For the constrained least-mean-squares problem of Theorem 1 the steady-state misadjustment is

$$M(\mu) = \lim_{k\to 0} \frac{E\{[d(k) - W^T(k)X(k)]^2\} \ - \ E\{[d(k) - W_*^T X(k)]^2\}}{E\{[d(k) - W_*^T X(k)]^2\}} \qquad (5.14)$$

Under the assumptions that $d(k)$ and the components of $X(k)$ are jointly Gaussian-distributed and independent from observation to observation it is possible to calculate very tight bounds on $M(\mu)$ by a method due to Moschner [20]. For an adaptive gain constant satisfying

$$0 < \mu < \frac{1}{\sigma_{max} + (1/2)\mathrm{Tr}(PR_{XX}P)}, \qquad (5.15)$$

it is shown in Appendix B that steady-state misadjustment may be bounded by

$$\frac{\mu}{2} \frac{\mathrm{Tr}(PR_{XX}P)}{1 - \frac{\mu}{2}[\mathrm{Tr}(PR_{XX}P) + 2\sigma_{min}]} \leq M(\mu) \leq \frac{\mu}{2} \frac{\mathrm{Tr}(PR_{XX}P)}{1 - \frac{\mu}{2}[\mathrm{Tr}(PR_{XX}P) + 2\sigma_{max}]}$$

$$(5.16)$$

$M(\mu)$ can be made arbitrarily close to zero by suitably small choice of gain constant μ ; this means that the steady-state performance of the Constrained-LMS algorithm can be made arbitrarily close to the optimum. From (5.10) it is seen that such cost performance is obtained at the expense of increased convergence time.

VI. APPLICATIONS

In this section adaptive solutions are given to the problems defined in Theorem 1 and its corollaries. At the same time the performance of each adaptive algorithm is given. The important application to array processing is the main example of this section.

The results of the preceding section are summarized in a companion theorem to Theorem 1:

Theorem 2. (Adaptive Constrained Least-Mean-Squares Optimization) Let $\{d(k)\}$ be a sequence of random variables and $\{X(k)\}$ be a sequence of n-dimensional data vectors of observed random variables. Each vector $X(k)$ is assumed to be produced independently by an unknown ergodic source with unknown correlation matrices

$$E\{X(k)X^T(k)\} = R_{XX} \qquad (n \times n)$$

$$E\{X(k)d(k)\} = R_{Xd} \qquad (n \times 1)$$

and R_{XX} positive definite. The algorithm

$$W(k+1) = P[W(k) + \mu e(k)X(k)] + F \ , \qquad (6.1)$$

where

$$P = [I - C(C^TC)^{-1}C^T] \ ,$$

$$F = C(C^TC)^{-1}\mathcal{F} \ ,$$

$$C^TW(0) = \mathcal{F} \ ,$$

and

$$e(k) = d(k) - W^T(k)X(k) \;,$$

converges in the mean to the optimum weight vector W_* solving the constrained LMS problem defined in Theorem 1:

$$\begin{aligned} &\text{minimize} \quad E\{[(d(k) - W^T X(k)]^2\} \\ &\text{subject to} \quad C^T W = \mathcal{F} \;, \end{aligned} \tag{6.2}$$

if

$$0 < \mu < \frac{1}{\sigma_{\max} + \frac{1}{2}\,\mathrm{Tr}(PR_{XX}P)} \;. \tag{6.3}$$

Further,

i) the convergence time constant of the difference between $EW(k)$ and W_* along the i^{th} eigenvector of $PR_{XX}P$ is

$$\tau_i = \frac{-1}{\ln(1-\mu\sigma_i)} \cong 1/\mu\sigma_i \tag{6.4}$$

where σ_i is the eigenvalue corresponding to the i^{th} eigenvector of $PR_{XX}P$.

ii) Under the additional assumption that variables $d(k)$ and $X(k)$ are jointly Gaussian distributed, the steady-state misadjustment of the adaptive solution can be bounded by (5.16).

<u>Proof of Theorem 2.</u> (See Section V)

<u>This completes the proof of Theorem 2.</u>

Example 1. (Consistent Modeling)

A single-input, single-output system is to be modeled with a tapped-delay-line filter. It is known that the system's steady-state response to a unit step-function input is a particular number α , and this feature is to be incorporated into the model.

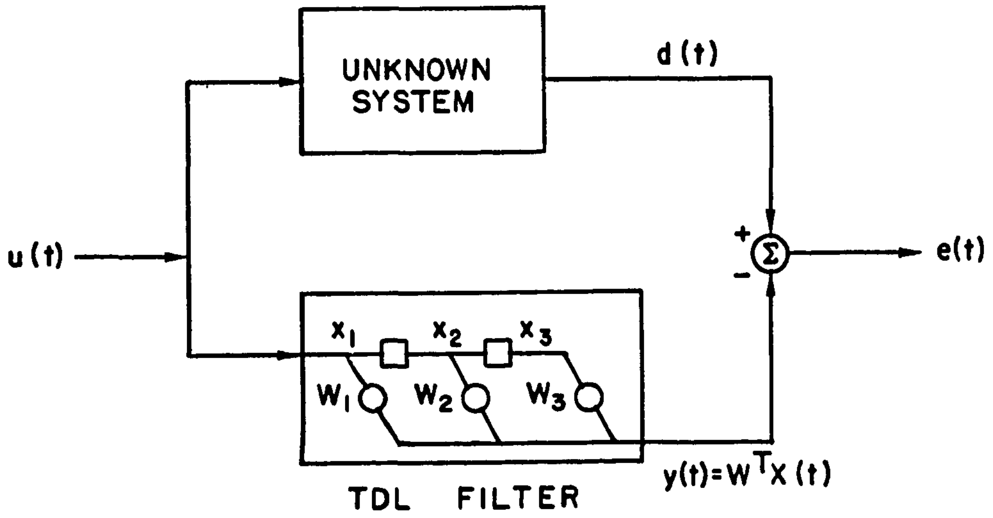

Fig. 6.1. Tapped-delay-line filter modeling a system.

Let the input to the system and model be a random variable $u(t)$. The model is a tapped-delay-line filter with n tap points and a delay of Δ seconds between each tap. Inputs pass down the tapped-delay-line (TDL) filter. Let the states at each tap point be denoted $x_i(t), i=1,2,\ldots,n$.

Thus the first state is equal to the input, $x_1(t) = u(t)$, the second state is equal to the input delayed by Δ seconds, $x_2(t) = u(t-\Delta)$, and so forth. The output $y(t)$ of the TDL filter is a weighted sum of the states. Let the weight on the i^{th} state be w_i and form the n-dimensional vectors of weights $W^T = (w_1, w_2, \ldots, w_n)$ and states $X^T(t) = (x_1(t), x_2(t), \ldots, x_n(t))$. The output of the TDL filter at time t is then $y(t) = W^T X(t)$. The desired output of the model is the output of the unknown system $d(t)$. The error is the difference between the desired output and the actual output: $e(t) = d(t) - y(t)$.

The constraint is now included. If the systems are given a unit step input (i.e., $u(t) = 1$), then after $n\Delta$ seconds the TDL filter will be in steady state, with $X^T(t) = \underline{1}^T = (1, 1, \ldots, 1)$. Thus the constraint that the steady-state response of the TDL filter be a α is equivalent to requiring

$$\underline{1}^T W = \alpha \; . \qquad (6.5)$$

The consistent modeling problem is therefore

$$\begin{aligned} &\text{minimize} \quad E\{e^2(t)\} \\ &\text{subject to} \quad \underline{1}^T W = \alpha \; . \end{aligned} \qquad (6.6)$$

To apply the Constrained LMS algorithm, the state vector and error are sampled at intervals of T seconds. At the k^{th} sampling instant the state vector is $X(kT)$ and the error is $e(kT)$. The time between samples T is made large enough so that $X(kT)$ is essentially independent of $X(jT)$ for $j \neq k$. (As noted in Section V this is not believed to be absolutely necessary). The algorithm is therefore

$$W(k+1) = P[W(k) + \mu e(kT)X(kT)] + \alpha \qquad (6.7)$$

where

$$P = I - \underline{1}(\underline{1}^T\underline{1})^{-1}\underline{1}^T = I - \frac{1}{n}\,\underline{1}\,\underline{1}^T \ ,$$

and the weight vector is assumed to be constant for t in $kT \leq t < (k+1)T$.

A practical matter arises here. It may be difficult to calculate the permissible upper bound on μ given by (6.3), especially if the autocorrelation matrix R_{XX} is not known. An easily measured quantity guaranteed to be no higher than the permissible upper bound is

$$\mu_0 = \frac{1}{\frac{3}{2}\,\mathrm{Tr}(R_{XX})} \ , \qquad (6.8)$$

That is, if μ is chosen to satisfy

$$0 < \mu < \mu_0 \ , \qquad (6.9)$$

then it is guaranteed to satisfy (6.3). Observe that μ_0 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 states.

A special case of the algorithm given in Theorem 2 is the celebrated LMS algorithm. The following is a companion to Corollary 1.1.

<u>Corollary 2.1.</u> (Adaptive Least-Mean-Squares Optimization-- Widrow and Hoff) Let the sequences $\{d(k)\}$, $\{X(k)\}$ and their (unknown) correlation matrices be defined as in Theorem 2. The algorithm

$$W(k+1) = W(k) + \mu e(k)X(k) \tag{6.10}$$

where

$$e(k) = d(k) - W^T(k)X(k) .$$

converges in the mean to the optimum weight vector W_{*1} solving the unconstrained least-mean-squares optimization problem defined in Corollary 1.1:

$$\text{minimize} \quad E\{e^2(k)\} , \tag{6.11}$$

if

$$0 < \mu < \frac{1}{\lambda_{max} + \frac{1}{2}\, \text{Tr}(R_{XX})} . \tag{6.12}$$

Further,

i) the convergence time constant of the difference between $EW(k)$ and W_{*1} along the i^{th} eigenvector of R_{XX} is

$$\tau_i = \frac{-1}{\ln(1-\mu\lambda_i)} \cong 1/\mu\lambda_i , \tag{6.13}$$

where λ_i is the eigenvalue corresponding to the i^{th} eigenvector of R_{XX}.

ii) Under the additional assumption that variables $d(k)$ and $X(k)$ are jointly Gaussian distributed, the steady state misadjustment of the adaptive solution can be bounded by

$$\frac{\mu}{2}\,\frac{\mathrm{Tr}(R_{XX})}{1-\frac{\mu}{2}[\mathrm{Tr}(R_{XX})+2\lambda_{min}]} \le M(\mu) \le \frac{\mu}{2}\,\frac{\mathrm{Tr}(R_{XX})}{1-\frac{\mu}{2}[\mathrm{Tr}(R_{XX})+2\lambda_{max}]} \qquad (6.14)$$

Proof of Corollary 2.1.

The projection operator P of Theorem 2 goes to the identity when all constraints are removed. The vector F vanishes.

This completes the proof of Corollary 2.1.

Uses of linear least-squares algorithms are abundant and well-known so no examples will be given.

The next corollary is a companion to Corollary 1.2.

Corollary 2.2. (Adaptive Least-Mean-Squares Distortionless Estimate) Let the sequences $\{d(k)\}$, $\{X(k)\}$, and their (unknown) correlation matrices be defined as in Theorem 2. Further, let each $X(k)$ be of the form

$$X(k) = CB(k) + N(k) \ , \tag{6.15}$$

where C is a known $(n \times m)$ matrix with $n > m$ and $\{B(k)\}$ is a sequence of unknown m-dimensional vectors. Each $B(k)$ may be a vector of random variables with unknown mean (so $E\{B\} = \overline{B}$), or it may be a vector of unknown parameters, in which case $E\{B\} = \overline{B} = B$. $\{N(k)\}$ is a sequence of unknown n-dimensional zero-mean random vectors considered as noise. $B(k)$ and $N(k)$ are assumed uncorrelated. Let

$$B(k) = \begin{bmatrix} b_1(k) \\ b_2(k) \\ \vdots \\ b_m(k) \end{bmatrix} \quad \text{and} \quad \mathcal{F} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \leftarrow i^{th} \text{ component} \ . \tag{6.16}$$

The algorithm

$$W(k+1) = P[W(k) - \mu y(k)X(k)] + F \tag{6.17}$$

where

$$P = [I - C(C^T C)^{-1} C^T]$$

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

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

$$y(k) = W^T(k) X(k) \ ,$$

converges in the mean to the optimum weight vector W_{*2} solving the least-mean-squares distortionless estimation problem defined in Corollary 1.2:

$$\begin{aligned} &\text{minimize} \quad E\{[b_i - W^T X(k)]^2 \\ &\text{subject to} \quad E\{(\bar{b}_i - W^T X(k))\} = 0 \ , \end{aligned} \tag{6.18}$$

as long as μ satisfies condition (6.3) of Theorem 2.

The convergence rates and misadjustment are the same as those of Eqs. (6.4) and (5.16) of Theorem 2.

Proof of Corollary 2.2.

It was shown in Corollary 1.2 that the problem (6.18) may be reformulated as

$$\begin{aligned} &\text{minimize} \quad E\{[W^T X(k)]^2\} \\ &\text{subject to} \quad C^T W = \mathcal{F} \ . \end{aligned} \tag{6.19}$$

This is just the problem of Theorem 2 with $d(k) = 0$. Accordingly in the Constrained-LMS algorithm (6.1), $d(k)$ is set to zero and the corollary follows. The requirements on μ and performance are unchanged.

This completes the proof of Corollary 2.2.

Example 2. (State Estimation under Uncertainty)

A system is described by the equations

$$\begin{aligned} B(k+1) &= \Phi B(k) + \Gamma U(k) \\ X(k) &= C\,B(k) + N(k) \ , \end{aligned} \tag{6.20}$$

where C is a known $(n \times m)$ matrix with $n > m$. $B(k)$ is the state vector and $U(k)$ is the input vector. $N(k)$ is zero-mean measurement noise. The matrices Φ and Γ are not known; the statistics of $U(k)$ and $N(k)$ are not known. We wish to estimate a component $b_i(k)$ of the state vector. The algorithm (6.17) converges to the best constant linear least squares unbiased estimator of a component of $B(k)$. If an estimate of the entire vector is desired, m algorithms like (6.1) may be used, with the unit entry in a different place in each $\mathcal{F}$ vector.

■

Note that the amount of knowledge required for an estimate of $B(k)$ using the constrained LMS algorithm is a small fraction of that required by the Kalman filter. (The Kalman filter is the optimum unconstrained time-varying linear least-squares estimator for the state vector of the dynamic system (6.20), and requires that all system matrices and correlation matrices be known.)

The next corollary is a companion to Corollary 1.3 .

Corollary 2.3. (Adaptive Least-Mean-Squares Filtered Estimate)

Let the sequence $\{d(k)\}$, $\{X(k)\}$, and their (unknown) correlation matrices be defined as in Theorem 2. Let

$$X(k) = C\,B(k) + N(k)\ , \tag{6.21}$$

$B(k)$ and $N(k)$ be vectors of random variables as in Corollary 1.3. Further, let $B(k)$ be written

$$B(k) = \mathcal{S}(k) + A(k)\ , \tag{6.22}$$

where $\{\mathcal{S}(k)\}$ and $\{A(k)\}$ are sequences of m-dimensional vectors of random variables. We wish to estimate the

i^{th} component of $\mathcal{S}(k)$, $s_i(k)$. A filter vector $\mathcal{F}$ is given, which may be based on as much or as little information about the statistics of $\mathcal{S}(k)$ and $A(k)$ as is known.

The algorithm

$$W(k+1) = P[W(k) - \mu y(k)X(k)] + F , \tag{6.23}$$

where

$$P = [I - C(C^T C)^{-1} C^T]$$

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

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

$$y(k) = W^T(k)X(k) ,$$

converges in the mean to the optimum weight vector W_{*3} solving the least-mean-squares filtering problem given in Corollary 1.3:

$$\begin{aligned} &\text{minimize} \quad E\{[s_i(k) - W^T X(k)]^2\} \\ &\text{subject to} \quad C^T W = \mathcal{F} , \end{aligned} \tag{6.24}$$

as long as μ satisfies condition (6.3) of Theorem 2.

The convergence rates and misadjustment are the same as those of Eqs. (6.4) and (5.16) of Theorem 2.

Proof of Corollary 2.3.

In Corollary 1.3, it is shown that the problem (6.24) may be reformulated as

$$\begin{aligned} &\text{minimize} \quad E\{[W^T X(k)]^2\} \\ &\text{subject to} \quad C^T W = \mathcal{F} \quad . \end{aligned} \qquad (6.25)$$

As before, this is just the problem of Theorem 2 with $d(k) = 0$. The results follow from Theorem 2.

This completes the proof of Corollary 2.3.

The next example is the major example of the paper, and is one of the main reasons for an interest in adaptive constrained least squares optimization: It is shown in this example that adaptive constrained LMS optimization makes possible the near-optimum processing of data from an array of antennas or other sensors with very little a priori information about the signals and noises involved. In contrast, known adaptive processors converging to the optimal unconstrained least-mean-squares filter [12] require knowledge of either the signal or noise statistics.

Example 3. (The Array Processor)

In most applications involving arrays of sensors -- notably sonar, seismology, radio communication, and radio astronomy using antenna arrays -- it is desirable to reduce antenna sensitivity to unwanted signals and noises while processing the signals of interest in real time. For example, arrays of sonar hydrophones provide information about the undersea environment; it may be desirable to listen to signals coming from a particular direction and simultaneously avoid hearing the noise of the sonar ship's own

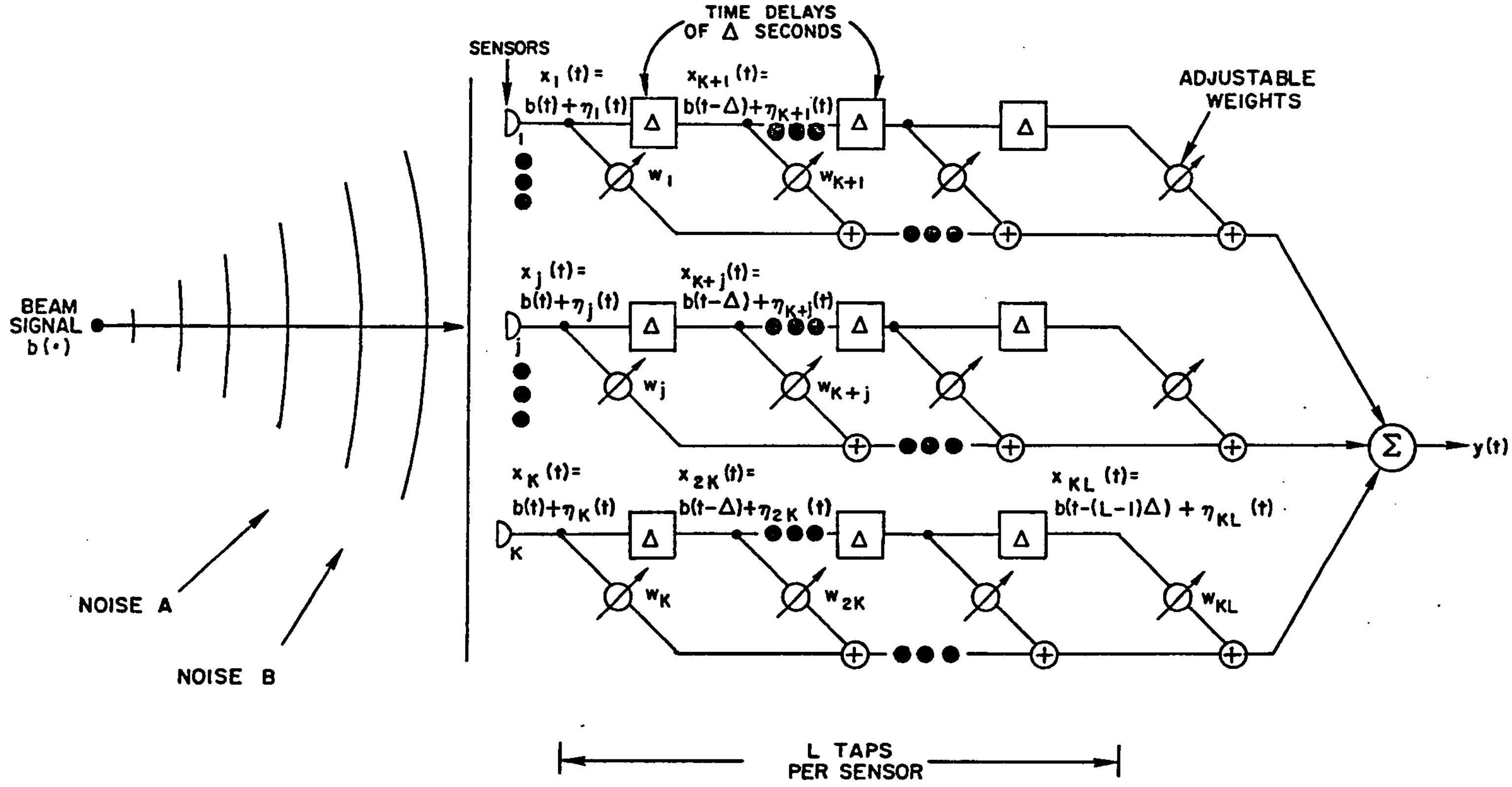

Fig. 6.2. Signals and noises incident on the array. Because the array is steered toward the look direction, all beam signal components on any given column of filter taps are identical.

machinery and screws [22]. In geology, sub-arrays of seismometers are being used in the large-aperture seismic array in Montana [4] to listen to seismic events; such arrays must discriminate against noises emanating from surrounding cities. In radio communications using antenna arrays, it is desirable to receive signals from one direction while ignoring the signals from amateur radio operators and other electromagnetic noises [5]. Radio astronomers using antenna arrays want to look in one part of the sky while discriminating against other radiation sources impinging on antenna sidelobes [25, p. 33]. In all these applications, it is desirable to have a processor that can discriminate against unwanted noises in real time and that requires a minimum of *a priori* information.

An array processor is a filter both in frequency and in space. A typical processor configuration is shown in Fig. 6.2. The array has K sensors with a tapped delay line following each sensor. Each line has L tap points and delays of Δ seconds between taps. Signals and noises impinging on the array are converted to voltages which pass down the tapped delay lines and the weighted sum of these voltages is the output of the filter. By proper choice of weights, the array processor can discriminate against unwanted noises distributed both in frequency and space.

The filter separates desirable signals from other noises. In this example, to be "desirable" a signal must come from a particular chosen direction in space, called the "look direction". All signals coming from other directions, plus any measurement or amplifier noise, are termed "undesirable noise". But not all signals coming from the look direction are desirable; some noise comes from the look direction and is called "look direction noise".

The signal is modeled as a zero-mean random process emanating from the look direction in the far field of the array. It is assumed that the propagating medium is linear, non-dispersive, and that propagation times along the signal phase-front are well enough known that the array can be steered, electrically or mechanically, in the direction of the signal. Sources in the look direction, i.e., desired signal and look direction noise, are assumed to be statistically uncorrelated with noises emanating from other directions. (This rules out multipath.) Finally, all the sensors are assumed to have identical characteristics (but are not necessarily omnidirectional).

It should be mentioned that sonar and seismic signals are generally low frequency (audio or lower) and may be processed in real time using the adaptive algorithm implemented by present-day hardware [13]. In radio-frequency applications, however, the signals must first be demodulated.

As in Example 1, let the observations at each tap points at time t be denoted $x_i(t), i=1,2,\dots,n$. In this case $n = KL$, the number of sensors times the number of taps per sensor. Let the weight on the i^{th} observation be w_i and form the n-dimensional vectors of weights $W^T = (w_1, w_2, \ldots, w_n)$ and observations $X^T(t) = (x_1(t), x_2(t), \ldots, x_n(t))$. The output of the array processor at time t is then $y(t) = W^T X(t)$. Let the signals arriving "in the beam" (i.e., from the look direction) at time t be denoted $b(t)$. Because the array is steered toward the look direction, signals arriving "in the beam" enter each of the K tapped-delay lines simultaneously and parade in parallel down the lines (see Fig. 6.2). All K taps on the first column of taps have the same beam component, $b(t)$, and a different undesirable noise component $\eta_i(t), i=1, \ldots, K$ from noises entering from other directions and amplifier noise; every tap on the second column of taps has the same beam component $b(t-\Delta)$ and a different noise component $\eta_i(t), i=K+1, \ldots, 2K.$, and so forth. Forming the L-dimensional vector of beam signals on each column at time t , $B^T(t)=[b(t), b(t-\Delta), \ldots, b(t-(L-1)\Delta)]$ and the n-dimensional vector of undesirable noises on each tap at time t , $N^T(t)=[\eta_1(t), \eta_2(t), \ldots, \eta_n(t)]$ it is seen from Fig. 6.2 that the vector of observations may be written

$$X(t) = C\,B(t) + N(t) \quad , \qquad (6.26)$$

where

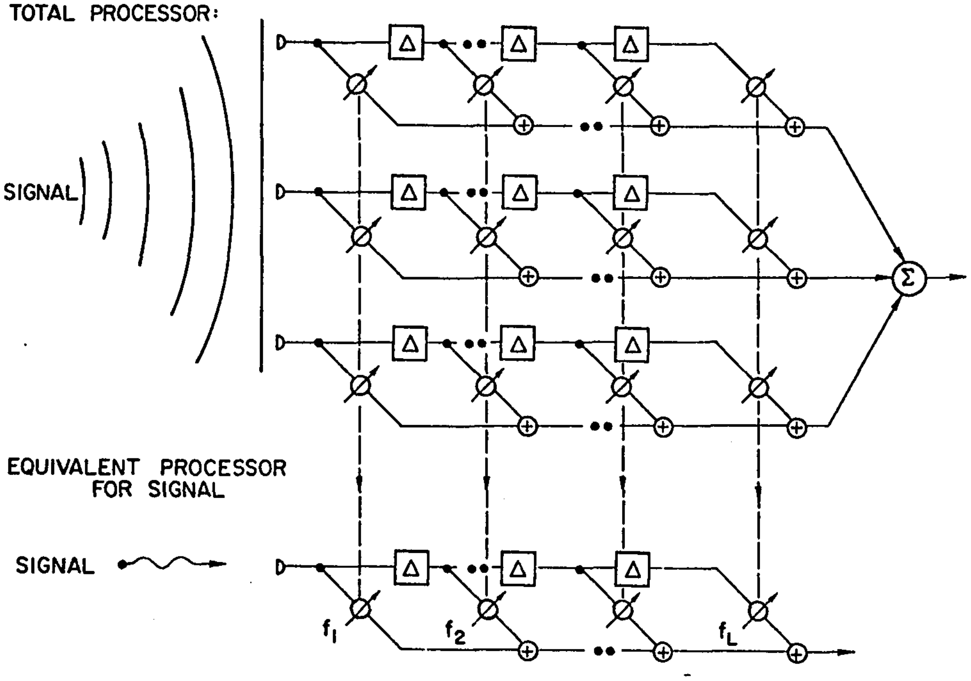

Fig. 6.3 Equivalent processor for signals coming from the look direction.

$$C = \begin{array}{c} K\updownarrow \\ \\ \\ \\ \\ \\ \\ \\ \end{array}\overset{\xleftrightarrow{\qquad\qquad L \qquad\qquad}}{\left[\begin{array}{ccccc} 1 & 0 & 0 & \cdots & 0 \\ \vdots & & & & \\ 1 & 0 & 0 & & \cdot \\ 0 & 1 & 0 & & \cdot \\ & \vdots & & & \cdot \\ 0 & 1 & 0 & & \\ & & \ddots & 0 & 1 \\ & & & \vdots & \vdots \\ 0 & \cdots & & 0 & 1 \end{array}\right]} \updownarrow KL \qquad (6.27)$$

Due to the array steering it is particularly easy to specify the frequency response of the processor in the look direction, since all K taps in every vertical column of taps have identical beam components: The processor output is formed by a weighted sum of the observations; therefore, as far as the beam signal is concerned, the total weighting it receives at each vertical column is the sum of the weights in that column (see Fig. 6.3). For the beam signal, the multichannel processor could be replaced by a single-channel filter. Each weight on the single-channel filter would be equal to the sum of weights on the corresponding column of the multichannel filter.

Let the i^{th} weight on the beam-signal-equivalent filter in Fig. 6.3 be f_i . Let the L-dimensional vector of filter weights be

$$\mathcal{F} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_L \end{bmatrix} . \tag{6.28}$$

Each weight f_i is the sum of the weights in the i^{th} column of the multichannel processor. Referring to the definition of C above, it is seen that this statement is equivalent to

$$\mathcal{F} = C^T W . \tag{6.29}$$

$\mathcal{F}$ determines the transfer function of the processor in the look direction. If, for example,

$$\mathcal{F}^T = \lfloor 0,0,\dots,0,1,0,\dots,0 \rfloor , \tag{6.30}$$

then all frequencies of signals arriving from the look direction in plane waves would be passed equally without attenuation (flat frequency response). Changing any of the zero components would result in a different impulse response and corresponding frequency shaping.

Recall that in the beam signal $b(t)$ there is a component of "desired signal" $s(t)$ and an additive "look direction noise" $a(t)$, i.e., $b(t) = s(t) + a(t)$. It is assumed that from any a priori information he might have about the frequency content of $s(t)$ or $a(t)$, the processor user specifies the look direction frequency response he wants in the form of the vector $\mathcal{F}$. If he has no prior information about the desired signal then a reasonable choice for $\mathcal{F}$ is the all-pass filter (6.30). Notice that specifying the look direction frequency response constrains only L "degrees of freedom" of the n weights in W. The remaining "degrees of freedom" are used by the processor to reduce the power of undesirable noises $N(t)$ in the output. Since the response to the beam signals is constrained and the undesirable noises are assumed uncorrelated with the beam signal, minimizing total processor output power is exactly the same as minimizing noise output power.

The problem is

$$\begin{aligned} &\text{minimize} \quad E\{[W^T X(t)]^2\} \\ &\text{subject to} \quad C^T W = \mathcal{F} \end{aligned} \tag{6.31}$$

and the algorithm is (6.22): $W(k+1) = P[W(k) - \mu y(kT)X(kT] + F$ where T is the time between adaptations, made sufficiently large so that successive vectors X are essentially independent.

In this case P is simple and sparse due to the simple form of the constraint matrix (6.26). The matrix multiplication by P is more simply regarded as a series of additions and scalar multiplications:

$$P = \left[\begin{array}{cccccccccccc}
1 & -\frac{1}{K} & \cdots & -\frac{1}{K} & 0 & & & 0 & & & & 0 \\
-\frac{1}{K} & 1 & & & & & & & & & & \\
\vdots & & \ddots & & & & & & & & & \\
-\frac{1}{K} & -\frac{1}{K} & \cdots & 1 & 0 & 0 & & 0 & & & & \\
0 & 0 & & 0 & 1 & -\frac{1}{K} & \cdots & -\frac{1}{K} & 0 & \cdots & & \vdots \\
 & & & 0 & -\frac{1}{K} & 1 & & & 0 & & & \\
 & & & & \vdots & & \ddots & & & & & \\
\vdots & & & 0 & -\frac{1}{K} & -\frac{1}{K} & \cdots & 1 & 0 & & & \\
 & & & & & & & & \ddots & & & \\
 & & & & & & & & 1 & -\frac{1}{K} & \cdots & -\frac{1}{K} \\
 & & & & & & & & -\frac{1}{K} & 1 & & \\
 & & & & & & & & \vdots & & \ddots & \\
0 & & & & \cdots & & & 0 & -\frac{1}{K} & -\frac{1}{K} & \cdots & 1
\end{array}\right] \qquad (6.32)$$

A computer simulation of the processor was made using a low-precision language (BASIC) on a small computer (the HP 2116). The processor had four sensors on a line spaced at Δ second intervals and had four taps per sensor (thus n=16). The environment had three point noise sources, and white noise added to each sensor. Power of the beam signal was quite small in comparison to the power of interfering noises (see Table 6.1). The tap spacing defined a frequency of

SOURCE	POWER	DIRECTION (0° IS NORMAL TO ARRAY)	CENTER FREQUENCY (1.0 is $1/\Delta$)	BANDWIDTH
Beam 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	-	-	-

Table 6.1. Signals and noises in the simulation

1.0 (i.e., $f = 1.0$ is a frequency of $1/\Delta$ Hz.). In the look direction, foldover frequency for the processor response was $1/2\,\Delta$, or 0.5. All signals were generated by a pseudo-random, pseudo-Gaussian generator and passed through a filter to give them the proper spatial and temporal correlations. All temporal correlations were arranged to be identically zero for time differences greater than $25\,\Delta$. The time between adaptations was assumed greater than $58\,\Delta$, so successive samples of $X(kT)$ were generated independently.

The look direction filter was specified by the vector $\mathcal{f}^T = \lfloor 1,-2,1.5,2 \rfloor$ which resulted in a frequency characteristic shown in Fig. 6.4. The signal and noise spectra are shown in Fig. 6.5 and their spatial position in Fig. 6.2.

In this problem, the eigenvalues of R_{XX} ranged from 0.111 to 8.355. The upper permissible bound on the gain constant μ calculated by (6.3) was .074; a value of $\mu = .01$ was selected, which, by (5.16) would lead to a misadjustment of between 15.2 and 17.0%.

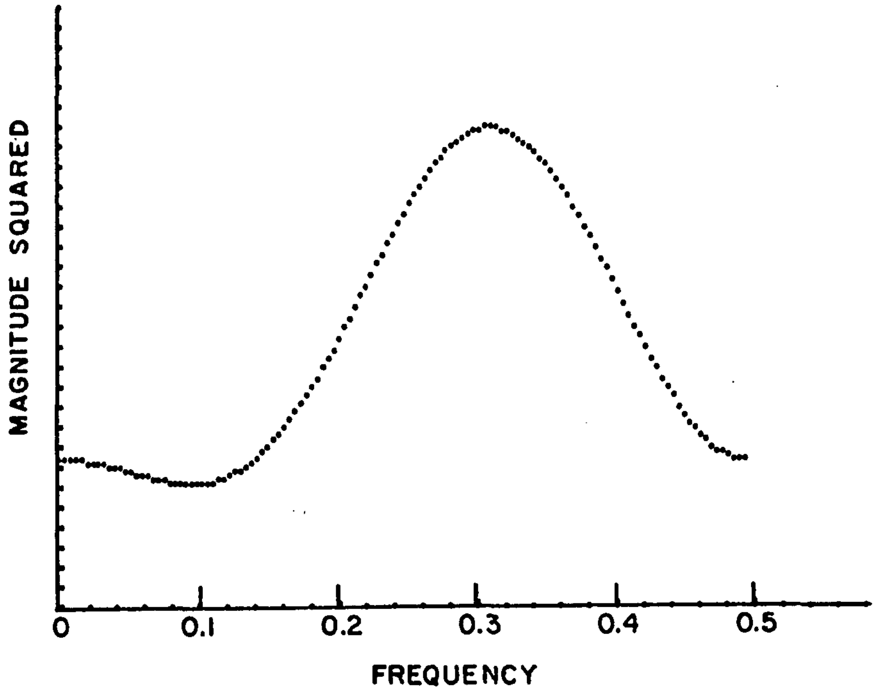

Fig. 6.4. Frequency response of the processor in the look direction.

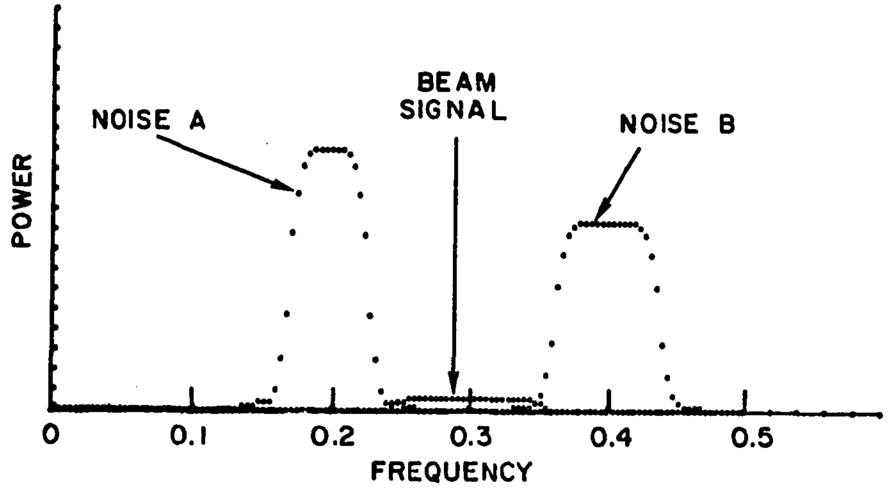

Fig. 6.5. Power spectral density of incoming signals.

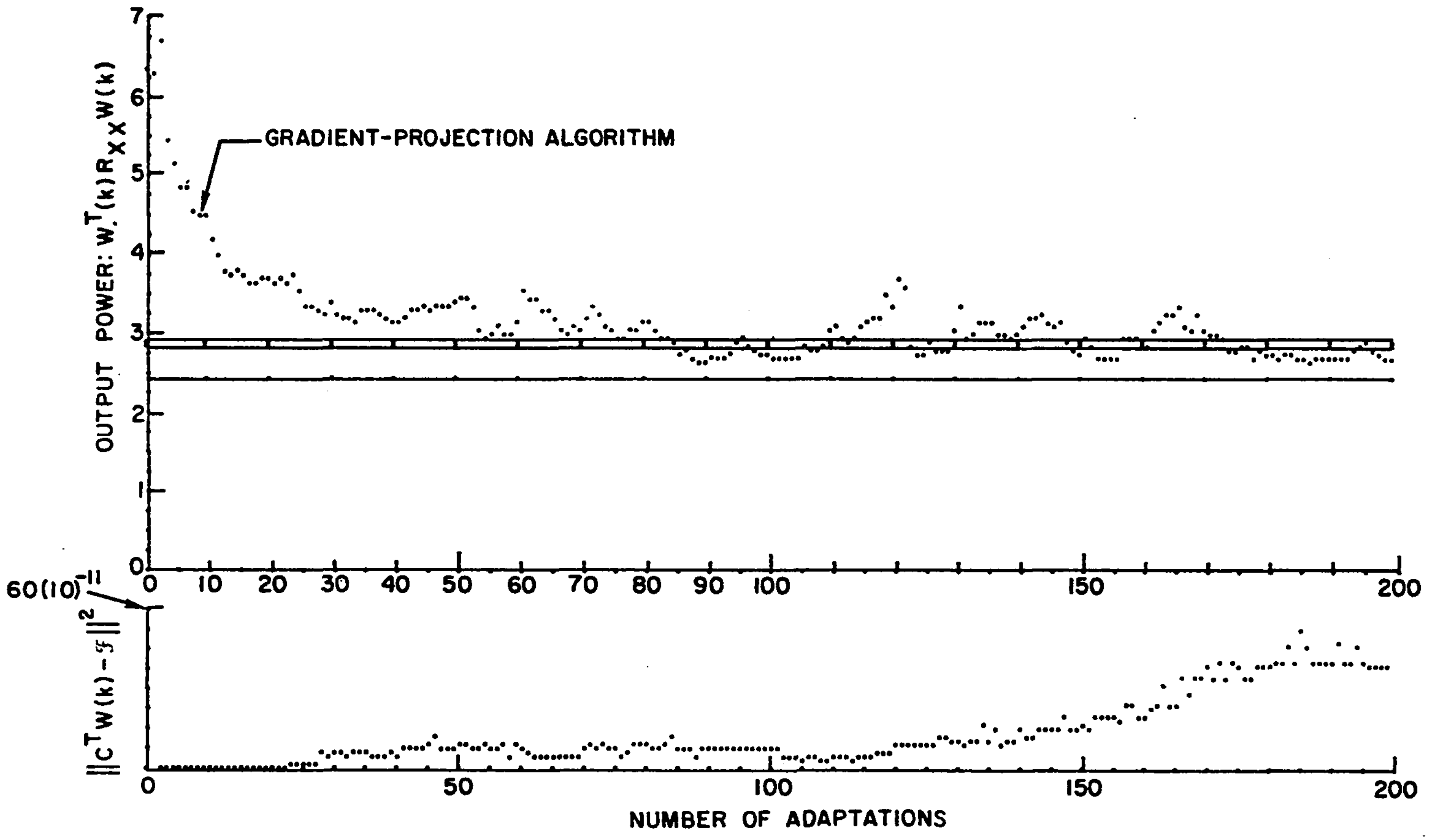

Fig. 6.6. THE OUTPUT POWER OF THE CONSTRAINED-LMS FILTER (upper graph) decreases as it adapts to discriminate against unwanted noise. Lower curve shows small deviations from the constraint due to quantization.

The processor was initialized with $W(0) = F = C(C^TC)^{-1}\mathcal{F}$, and Fig. 6.6 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 optimum output power plus misadjustment. The mean value of the processor's output power falls somewhere between the upper and lower 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.

Although the weight vector is, in theory, constrained to satisfy $C^TW(k) = \mathcal{F}$ at all times, very small deviations occur in an actual implementation due to quantization and computational errors. The lower graph in Fig. 6.6 shows the squared Euclidean distance between the weight vector and the constraint $\|C^TW(k) - \mathcal{F}\|^2$. An error-correcting feature of the Constrained-LMS algorithm prevents the deviations from the constraint from growing.

A last corollary deals with the deterministic constrained least squares problem.

Corollary 2.4. (Gradient-descent, Deterministic Constrained Least Squares) Let R_{XX} be a known $(n \times n)$ matrix and R_{Xd} be a known n-vector. The algorithm

$$W(k+1) = P[W(k) - \mu R_{XX} W(k) + \mu R_{Xd}] + F \qquad (6.33)$$

where

$$P = [I - C(C^T C)^{-1} C^T]$$

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

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

converges deterministically to the solution W_* of the problem

$$\begin{aligned} &\text{minimize} \quad [\alpha - 2W^T R_{Xd} + W^T R_{XX} W] \\ &\text{subject to} \quad C^T W = \mathcal{F} , \end{aligned} \qquad (6.34)$$

where α is any finite constant, as long as μ satisfies (6.3). The convergence time along eigenvectors of $PR_{XX}P$ is given by (6.4) and there is no steady-state misadjustment.

Proof of Corollary 2.4.

This algorithm is the same as the recursive relation (5.2) for the mean weight vector of the stochastic constrained LMS algorithm. Showing that the stochastic algorithm converges in the mean is therefore the same as showing (6.3) converges. Convergence in the mean was proved in Theorem 2.

This completes the proof of Corollary 2.4.

Remark: See Rosen [23] for an alternative solution to this problem.

VII. SENSITIVITY OF ALGORITHMS TO CALCULATION ERRORS

The constrained-LMS algorithm is related to the gradient-projection algorithms due to Rosen [23], Lacoss [14], and Booker [1]. The difference between the Constrained-LMS algorithm and gradient-projection algorithms lies in the way information about the location of the constraint surface is carried. As Fig. 4.5 showed, the Constrained LMS algorithm (3.11) "knows" the orientation of the constraint surface by the matrix C, and its translation from the origin by the vector F. In this section, it is shown that gradient-projection algorithms use only the orientation matrix C; to ensure that the weight vector stays on the constraint surface, they rely exclusively[†] on the fact that the weight vector is initialized on the constraint surface and always moves parallel to it. The gradient-projection method is shown to be sensitive to quantization errors which may cause the weight vector to deviate from the constraint on long runs.

Differences in the algorithms may be traced to Eq. (3.5) of the derivation. If $C^T W(k)$ is replaced by $\mathcal{F}$ in (3.5) and $R_{Xd} = \theta$, the gradient-projection algorithm of Booker results. ($C^T W(k)$ should equal $\mathcal{F}$ if $W(k)$ exactly satisfies

[†] Rosen recognized this problem and suggested using a second algorithm to "reset" the weight vector to the constraint whenever errors became excessive.

the constraint. It is shown in Fig. 6.6 that it may be unreasonable to assume that $W(k)$ is exactly on the constraint at all times. In the derivation of the Constrained-LMS algorithm, the term $C^T W(k)$ was carried instead of replacing it by $\mathcal{F}$. Carrying the term corresponds physically to assuming that $W(k)$ may not precisely satisfy the constraint, perhaps due to the quantization error of a digital implementation.

The algorithm that results from replacing $C^T W(k)$ by $\mathcal{F}$ is

$$W(k+1) = W(k) + \mu P e(k) X(k) \; ; \quad C^T W(0) = \mathcal{F} \; . \qquad (7.1)$$

This is a gradient-projection algorithm. It is so named because the unconstrained gradient is projected onto the constraint subspace and then added to the current weight vector. Its operation is shown in Fig. 7.1 (compare with Fig. 4.5).

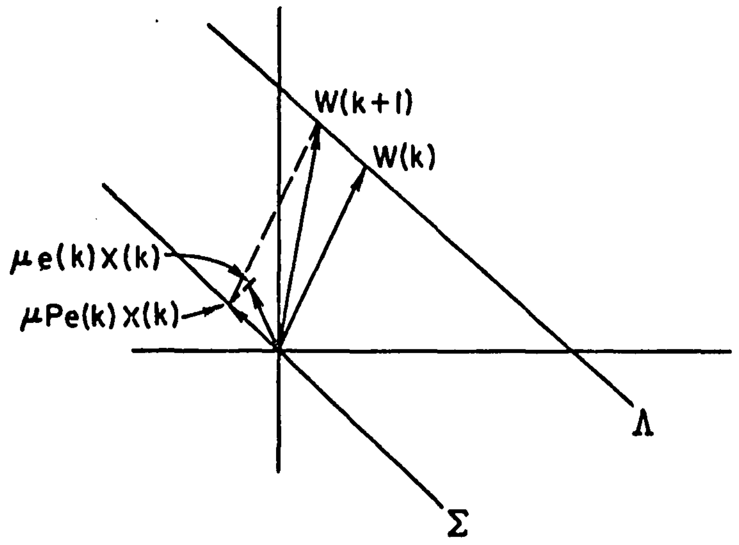

Fig. 7.1. Operation of the gradient-projection algorithm (7.1).

If somewhere in the computation an error occurs due to quantization and the weight vector is a bit off the constraint at time k, Fig. 7.2 shows that the Constrained-LMS algorithm (3.11) will bring the weight vector back to the constraint in the next iteration; however, the gradient-projection algorithm (7.1) assumes that $W(k)$ satisfied the constraint and adds a change parallel to the constraint surface, continuing the error.

An algebraic analysis is obtained by assuming that at each iteration the actual processor introduces a small vector of errors $\xi(k)$ to the weights. The update equations for the two algorithms become

Constrained-LMS:

from (3.11): $$W(k+1) = P[W(k)+\mu e(k)X(k)] + F + \xi(k) \qquad (7.2)$$

Gradient-Projection:

from (7.1): $$W(k+1) = W(k)+\mu Pe(k)X(k) + \xi(k) \ ;$$

$$C^T W(0) = \mathcal{F} \qquad (7.3)$$

Iterating the Constrained-LMS algorithm (7.2) back to the original $W(k+1) = P[W(k) - \mu X(k)X^T(k)W(k) + \mu X(k)d(k)] + F + \xi(k)$

(7.4)

$W(k\text{-}$

$$= \prod^{k} \{P[I-\mu X(i)X^T(i)]\}W(0)$$

(7.4)

$$= \prod_{i=0}^{k} \{P[I-\mu X(i)X^T(i)]\}W(0)$$

$$+ \sum_{i=0}^{k} \prod_{j=i+1}^{k} \left\{P[I-\mu X(j)X^T(j)]\right\} [\mu PX(k)d(k) + F + \xi(i)]$$

(7.5)

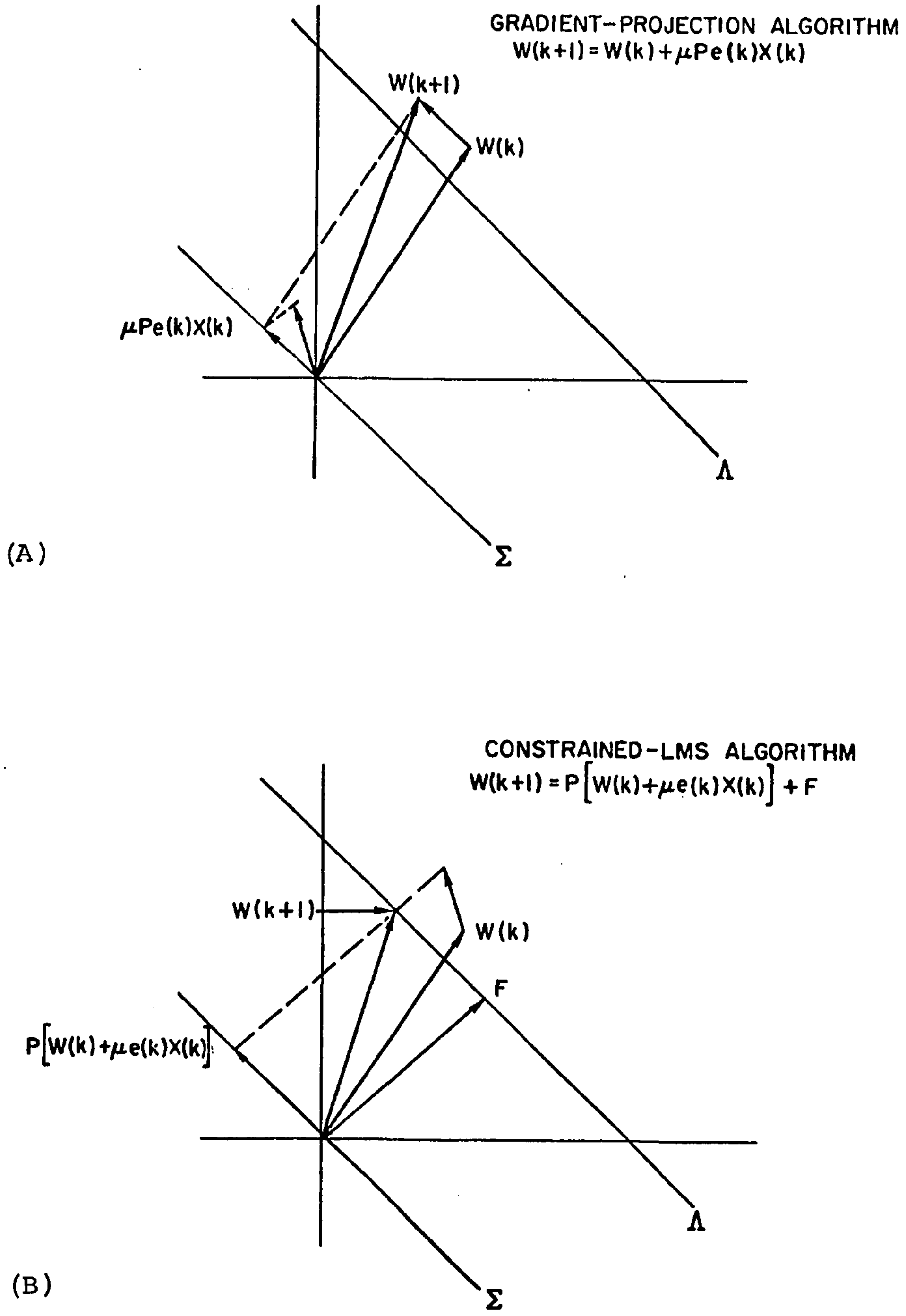
GRADIENT-PROJECTION ALGORITHM
W(k+1) = W(k) + μPe(k)X(k)
W(k+1)
W(k)
μPe(k)X(k)
Λ
(A)
Σ
CONSTRAINED-LMS ALGORITHM
W(k+1) = P[W(k)+μe(k)X(k)] + F
W(k+1)
W(k)
F
P[W(k)+μe(k)X(k)]
Λ
(B)
Σ

Fig. 7.2. Error propagation. The Constrained-LMS algorithm (A) corrects deviations from the constraints while the gradient-projection algorithm (B) allows them to accumulate.

where undefined products are taken to be the identity. Now noting that $C^TP = C^T[I-C(C^TC)^{-1}C^T] = \theta$ and premultiplying by C^T to see how the weight satisfies the constraint we have

$$C^TW(k+1) = C^T[F+\xi(k)] = \mathcal{F} + C^T\xi(k) \ . \tag{5.6}$$

In a perfect implementation the right side of (7.6) would be $\mathcal{F}$. With quantization errors and using the Constrained-LMS algorithm, the weight vector is off the constraint only by a term linear in the last error vector.

Now an error analysis on the gradient-projection algorithm is made. Performing a backward iteration on (7.3) produces

$$W(k+1) = W(k)-\mu P[X(k)X^T(k)W(k) - X(k)d(k)] + \xi(k) \tag{7.7}$$

$$= \prod_{i=0}^{k}\left\{I-\mu PX(i)X^T(i)\right\} W(0)$$

$$+ \sum_{i=0}^{k}\left\{\prod_{j=i+1}^{k} [I-\mu PX(j)X^T(j)]\right\} [\mu PX(i)d(i)+\xi(i)] \tag{7.8}$$

$$C^TW(k+1) = C^TW(0) + C^T\sum_{i=0}^{k} \xi(i) = \mathcal{F} + C^T\sum_{i=0}^{k} \xi(i) \tag{7.9}$$

The last term of (7.9) shows how the algorithm (7.1) accumulates deviations from the constraint.

If the computation errors are modeled as a zero-mean process [27], the gradient-projection algorithm does a random walk, away from the constraint with variance increasing as the number of iterations (see Appendix D).

A simulation of the gradient projection algorithm on the array problem (Example 3) was made, using exactly the same data as used by the Constrained-LMS algorithm. The results are shown in Fig. 7.3. The lower part of Fig. 7.3 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. 6.6) were plotted on the same scale they would not be discernible. Further, the errors of the gradient-projection method are expected to continue to grow.

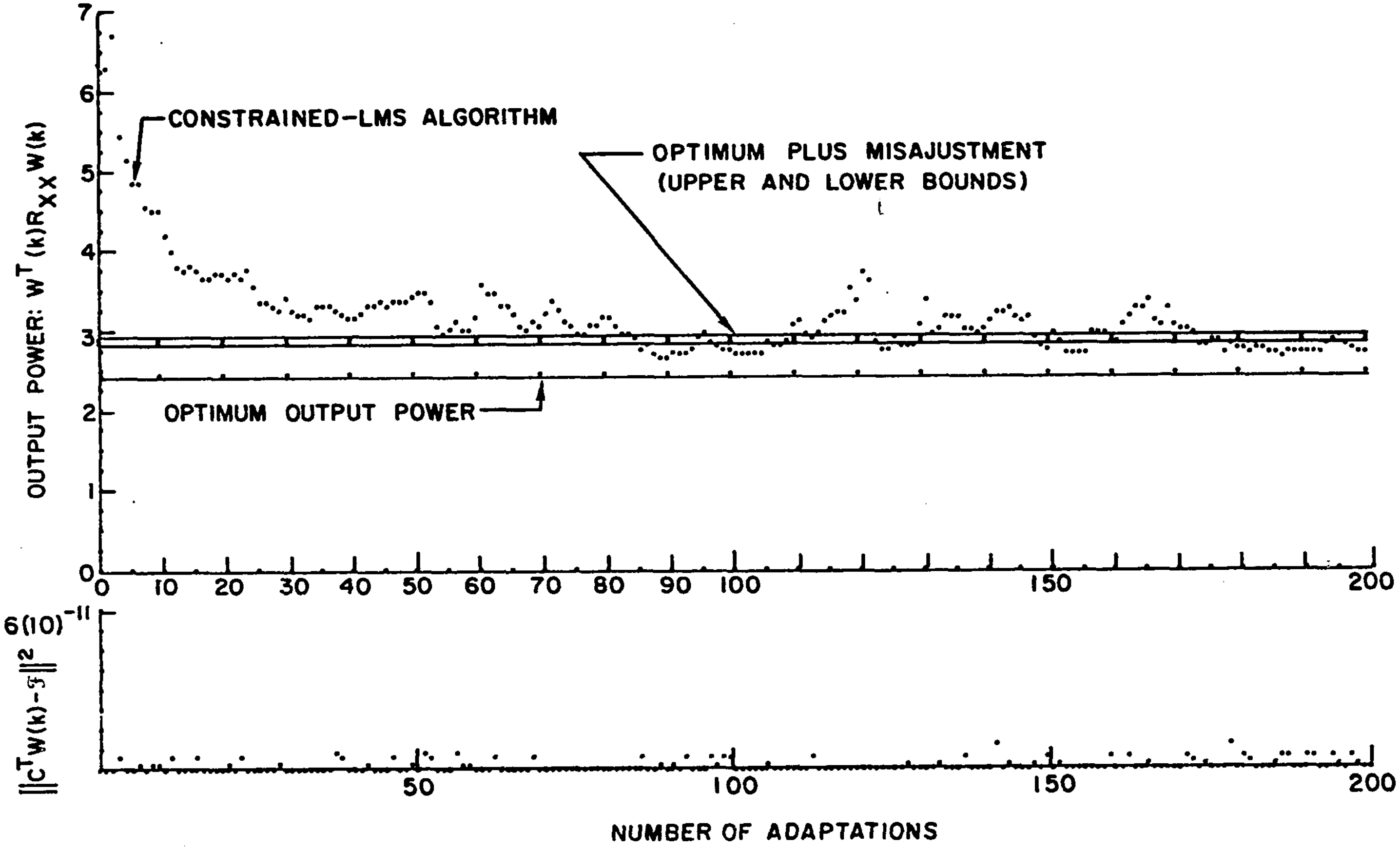

Fig. 7.3. OUTPUT POWER OF THE GRADIENT-PROJECTION ALGORITHM (upper graph) operated on the same data as the Constrained-LMS algorithm (c.f. Fig. 6.6). Lower curves shows that deviations from the constraint tend to increase with time. Note scale.

VIII. SUMMARY

A general algorithm was developed for stochastic linear least-squares optimization subject to linear equality constraints. The algorithm has three major properties: First, it has very modest computational requirements; second, it requires very little *a priori* knowledge; third, it converges to an optimal filter. A fourth property is that the algorithm can operate continuously without wandering from the constraints.

Rate of convergence and steady-state performance of the general algorithm are derived. Special cases of the algorithm are treated, with examples. An important application of the algorithm is the real-time processing of data from an array of sensors.

APPENDIX A

DERIVATION OF GRIFFITHS' MLR ALGORITHM BY THE QUADRATIC PENALTY FUNCTION METHOD

The purpose of this appendix is to show that the Maximum Likelihood Ratio (MLR) algorithm due to Griffiths [11] may be considered as an algorithm solving a least-mean-squares problem subject to "soft" linear equality constraints. This gives a simpler derivation than the original and immediately illuminates some properties of the algorithm that are well-known general properties of quadratic penalty function algorithms. As a side benefit, a general method of generating adaptive algorithms, based on the quadratic penalty function method, is indicated.

The quadratic penalty function method is a way of turning a constrained optimization problem into an easily-solved unconstrained optimization problem. Given a cost function $J(W)$ and a vector-valued constraint function $\Phi(W)=\theta$, the problem

$$\begin{aligned}&\text{minimize} \quad J(W)\\ &\text{subject to} \quad \Phi(W) = \theta\end{aligned} \tag{A.1}$$

is changed to

$$\text{minimize} \quad J(W)+\beta^2\Phi^T(W)\Phi(W) \ . \tag{A.2}$$

As the scalar $\beta\to\infty$ the solution to the unconstrained problem (A.2) goes to the solution of the problem (A.1). The second problem is easily solved by standard unconstrained optimization techniques.

The specific problem considered by Griffiths is the problem of Example 3, Section VI, where $J(W) = W^T R_{XX} W$ and $\Phi(W) = C^T W - \mathcal{F}$. The algorithm is derived by forming the function

$$H(W) = \frac{1}{2} W^T R_{XX} W + \beta^2 (C^T W - \mathcal{F})^T (C^T W - \mathcal{F}) \, , \qquad \text{(A.3)}$$

and taking the gradient with respect to W

$$\nabla_W H = R_{XX} W + \beta^2 C (C^T W - \mathcal{F}) \; . \qquad \text{(A.4)}$$

The iteration is then

$$\begin{aligned} W(k+1) &= W(k) - \mu \nabla_W H \\ &= W(k) - \mu R_{XX} W(k) - \mu\beta^2 C (C^T W(k) - \mathcal{F}) \; . \qquad \text{(A.5)} \end{aligned}$$

R_{XX} is replaced by its estimate, $X(k)X^T(k)$, giving

$$W(k+1) = W(k) - \mu X(k) X^T(k) W(k) - \mu\beta^2 C (C^T W(k) - \mathcal{F}) \; . \qquad \text{(A.6)}$$

This is Griffiths' MLR algorithm.

We infer from this derivation, and well-known properties of penalty function schemes [3], [17] that:

i) the algorithm has an error correcting property, i.e., it will not wander far from the constraint in the sense of the gradient projection algorithm discussed in Section VII.

ii) However, the satisfaction of the constraint is "soft", i.e., for finite values of β the solution of (A.2) will not exactly satisfy the constraint.

iii) Increasing β to cause the weight vector to more nearly satisfy the constraint will increase the convergence time of the algorithm.

APPENDIX B

STEADY-STATE MISADJUSTMENT

Moschner [20] calculated the misadjustment for the algorithm

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

By his method precisely the same results for misadjustment may be obtained for the algorithm

$$W(k+1) = P[W(k) + \mu e(k)X(k)] + F \ , \qquad (B.2)$$

where $e(k) = d(k) - y(k)$ and the optimal weight vector of (B.2) is defined to be W_* of Theorem 1.

A slight improvement in the bounds obtained by Moschner is possible by noting in his equation (D.19) that since $B_n \triangleq E\{V_n V_n^T\}$ and $V_n = PV_n$ by Geometrical Property 5 and $P^2 = P$, then

$$\mathrm{Tr}(PRB_nR) = \mathrm{Tr}(PRPB_nR) \ , \qquad (B.3)$$

and so

$$\sigma_{min}\,\mathrm{Tr}(B_nR) \le \mathrm{Tr}(PRB_nR) \le \sigma_{max}\,\mathrm{Tr}(B_nR) \ , \qquad (B.4)$$

where σ_{min} and σ_{max} are the smallest and largest non-zero eigenvalues of PRP. The result follows by using the above facts in Moschner's derivation.

APPENDIX C

LEMMAS ON QUADRATIC FORMS

Lemma C.1. Let R be an $(n \times n)$ positive-definite matrix and C be an $(n \times m)$ matrix (with $n > m$) having full rank m. Then the $(m \times m)$ matrix $C^T RC$ is positive definite and $(C^T RC)^{-1}$ exists.

Proof of Lemma C.1.

Since R is positive definite then $V^T RV > 0$ for any n-vector $V \neq \theta$. We want to show for any m-vector $U \neq \theta$ that $U^T C^T RCU > 0$, hence, $C^T RC$ is positive definite and its inverse exists.

If the vector $U \neq \theta$, it has rank 1. By Sylvester's inequality [9], the rank of the product of two matrices is not less than the sum of the ranks of the matrices, less their common dimension. Letting $\rho(\cdot)$ denote rank, the rank of the n-vector CU is bounded by

$$\begin{aligned} \rho(CU) &\geq \rho(C) + \rho(U) - m \\ &\geq m + 1 - m \\ &\geq 1 , \end{aligned} \qquad (C.1)$$

from which we conclude CU is not the zero vector. Therefore, letting $V = CU$ we conclude

$$U^T C^T RCU = V^T RV > 0 , \qquad (C.2)$$

for any non-zero vector U so C^TRC is positive definite.

<u>This completes the proof of Lemma C.1.</u>

<u>Remark 1.</u> It follows that if R is positive definite R^{-1} is positive definite and $(C^TR^{-1}C)^{-1}$ exists.

<u>Remark 2.</u> Since the identity matrix is positive definite it follows that $(C^TC)^{-1}$ exists.

<u>Lemma C.2.</u> Let R be a positive-definite $(n \times n)$ matrix. Let $P = [I - C(C^TC)^{-1}C^T]$, where C is $(n \times m)$ with full rank m . Let the subspace Σ be defined as $\Sigma = \{W : C^TW = \mathcal{F}\}$. Then

i) m eigenvectors of PRP lie entirely outside Σ and have zero eigenvalues.

ii) The other $(n - m)$ eigenvectors of PRP lie entirely within Σ and have strictly non-zero eigenvalues.

iii) Let σ_i be the i^{th} non-zero eigenvalue of PRP and λ_j be the j^{th} eigenvalue of R . Then the eigenvalues are related by

$$\lambda_{min} \leq \sigma_{min} \leq \sigma_i \leq \sigma_{max} \leq \lambda_{max} , \qquad (C.3)$$

for all $i = 1,2,\ldots(n-m)$.

Proof of Lemma C.2.

Since PRP is a symmetric $(n \times n)$ matrix, it has n eigenvectors and n eigenvalues. The eigenvectors can be chosen to be orthogonal [7].

i) Since the matrix C has full rank it has m columns of linearly independent n-vectors. Direct calculation shows that $C^T PRP = \theta$, so the m columns of C are eigenvectors of PRP with zero eigenvalues.

ii) There must be $(n-m)$ remaining eigenvectors orthogonal to the columns of C. As shown in Appendix E, the columns of C are vectors normal to the constraint plane Γ and subspace Σ. Therefore, the remaining $(n-m)$ eigenvectors must be in Σ. As shown in Geometrical Property 5 of Section IV, if V is a vector in Σ, then $PV = V$. Therefore if an eigenvector e_i of PRP is in Σ then

$$e_i^T PRPe_i = e_i^T Re_i > 0 \ . \qquad (C.4)$$

Let σ_i be an eigenvalue corresponding to an eigenvector of PRP in Σ. Then by definition

$$PRPe_i = \sigma_i e_i \qquad (C.5)$$

so

$$e_i^T PRPe_i = \sigma_i e_i^T e_i = \sigma_i \ . \qquad (C.6)$$

From (C.4) and (C.6) it follows that

$$\sigma_i > 0 \qquad i=1,2,\ldots,(n-m) . \tag{C.7}$$

iii) It is well known that if e is a unit vector then $e^T Re$ is bounded by

$$\lambda_{min} \leq e^T Re \leq \lambda_{max} , \tag{C.7}$$

where λ_{min} and λ_{max} are respectively the largest and smallest eigenvalues of R . Therefore from (C.4) and (C.6)

$$\lambda_{min} \leq \sigma_i \leq \lambda_{max} . \tag{C.8}$$

The result follows.

This completes the proof of Lemma C.2.

APPENDIX D

EXPECTED DEVIATION FROM THE CONSTRAINT BY THE GRADIENT-PROJECTION ALGORITHM

As an approximation, quantization in the weight vector is modeled as an additive white noise process (see Widrow, [27]); the expected deviation from the constraint by the gradient projection algorithm is computed as a function of time.

Assume that a fixed-point representation for the weights is used; let the quantization size of a single weight be q. Using Widrow's value for the error variance, $q^2/12$, from (7.9) the expected squared Euclidean distance from the constraint at time k is

$$
\begin{aligned}
E\{\|C^T W(k) - \mathcal{F}\|^2\} &= E\{\sum_{i=1}^{k} \xi_i^T C C^T \sum_{j=1}^{k} \xi_j\} \\
&= \mathrm{Tr}(E\{C \sum_{i=1}^{k} \sum_{j=1}^{k} \xi_i \xi_i^T\} C^T) \\
&= \mathrm{Tr}(C \frac{kq^2}{12} I \, C^T) \\
&= \frac{kq^2}{12} \mathrm{Tr}(CC^T) \qquad (D.1)
\end{aligned}
$$

Thus the expected squared distance from the constraint increases linearly with time (approximately).

For the special case of the array problem, with C defined by Eq. (6.27), $\mathrm{Tr}(CC^T) = n$, where n is the number of tap points. Equation (D.1) becomes

$$
E\{\|C^T W(k) - \mathcal{F}\|^2\} = kn \frac{q^2}{12} \,. \qquad (D.2)
$$

APPENDIX E

THE METHOD OF LAGRANGE MULTIPLIERS

Consider the equality-constrained optimization problem

$$\begin{aligned} &\text{minimize} \quad J(W) \\ &\text{subject to} \quad \Phi(W) = \theta \end{aligned} \tag{E.1}$$

where $J(\cdot)$ is a scalar cost function and $\Phi(\cdot)$ is a vector-valued constraint function. In Theorem 1, $J(W) = E\{(d - W^T X)^2\}$ and $\Phi(W) = C^T W - \mathcal{F}$. Let the gradient of the function $J(W)$ with respect to a vector W evaluated at W_o be written $\nabla_W J(W_o)$ where

$$\nabla_W J(W_o) = \left. \begin{matrix} \dfrac{\partial J}{\partial w_1} \\[2ex] \dfrac{\partial J}{\partial w_2} \\ \vdots \\ \dfrac{\partial J}{\partial w_m} \end{matrix} \right]_{W = W_o} . \tag{E.2}$$

A <u>necessary</u> requirement for the optimal solution of (E.1) to be at a point W_o is that the gradient of J with respect to W be normal to the constraint surface[†] at W_o . If the gradient of J at W_o were not normal to the constraint surface then by sliding along the constraint a vector W_1 could be found still satisfying the constraint but having

[†] The constraint surface is understood to be the points satisfying the constraint $\Phi(W) = \theta$.

lower cost, i.e., $J(W_1) < J(W_o)$.

Fleming ([8], p. 126) shows that the normal vectors to any manifold defined by $\Phi(W) = \theta$ is $\nabla_W \Phi$. For example, the gradient of the constraint defined by $\Phi(W) = C^T W - \mathcal{F} = \theta$ is $\nabla_W (C^T W - \mathcal{F}) = C$; therefore, each of the m columns of C is a vector orthogonal to the constraint plane and any linear combination of those vectors is also orthogonal to the plane.

Let λ be an undetermined m-vector of multipliers. The vector $C\lambda$ is a linear combination of the columns of C and so is normal to the constraint plane. Thus another way to express the necessary condition that the gradient of the cost function J be orthogonal to the constraint surface is to say that for some choice of λ the gradient and the normal may be anticollinear, i.e.,(see Fig. E.1)

$$\nabla_W J(W_o) + C\lambda = \theta \ , \tag{E.3}$$

or more generally

$$\nabla_W J(W_o) + \nabla_W \Phi(W_o)\lambda = \theta \ . \tag{E.4}$$

Another way of writing (E.4) analogous to the necessary condition for unconstrained optimality $(\nabla_W J(W_o) = \theta)$ is by defining the function $H(W)$ by "adjoining" the cost function to the constraint function by the Lagrange multipliers,

$$H(W) = J(W) + \lambda^T \Phi(W) \tag{E.5}$$

and requiring

$$\nabla_W H(W) = \theta \ . \tag{E.6}$$

Then since $\nabla_W H(W_o) = \nabla_W H(W_o) = \nabla_W J(W_o) + \nabla_W \Phi(W_o)\lambda$, (E.6) is identical to (E.4) and the necessary conditions become (E.6) and

$$\Phi(W_o) = \theta \ . \tag{E.7}$$

For an excellent discussion of the Lagrange multiplier method in more general applications see Bryson and Ho [3].

APPENDIX F

SIMULATION OF THE DIRECT SUBSTITUTION ALGORITHM

At the beginning of Section III the direct substitution algorithm was suggested: To obtain an estimate of the optimal weight vector, the unknown correlation matrices are estimated and inserted directly into the equation for the optimal weight vector. Although computationally quite difficult (because of the number of matrix inversions and multiplications involved) the direct substitution method offers the possibility of improved performance.

The direct substitution algorithm was simulated on the array-processing problem of Example 3 using exactly the same data as the Constrained-LMS processor. The direct substitution algorithm is

$$\hat{R}_{XX}(k) = \alpha\hat{R}_{XX}(k-1) + (1-\alpha)X(k-1)X^T(k-1) \qquad (F.1)$$

$$W(k) = \hat{R}_{XX}^{-1}(k)C[C^T\hat{R}_{XX}^{-1}(k)C]^{-1}\mathcal{F} \qquad (F.2)$$

where $0 < \alpha < 1$. Equation (F.1) is an exponentially-weighted estimate of the true correlation matrix R_{XX} . Equation (F.2) is the equation for the optimal weight vector for the problem with $\hat{R}_{XX}(k)$ substituted for R_{XX} . The constant α , which controls both rate of convergence and misadjustment, was chosen to be 0.97, a value which experimentally lead to approximately the same misadjustment as the Constrained-LMS processor had in Example 3. $\hat{R}_{XX}(0)$ was initialized to the identity matrix, scaled by the power measured on each tap; in this case the total power on each tap was 2.2 (see Table 6.1), so $\hat{R}_{XX}(0)$ was $2.2\mathbf{I}$. This is a reasonable starting point since the power on a tap is easily measured in a real situation and also a simple calculation shows that if $\hat{R}_{XX}(0)$ is any diagonal matrix then $W(0) = C(C^TC)^{-1}\mathcal{F} = F$. The vector F was also the initial weight vector of the Constrained-LMS algorithm so the two processors essentially start out at the same point and a meaningful comparison is easily obtained.

Results of the simulation are shown in Fig. F.1. Compare with Fig. 6.6. For the same misadjustment, the better processor should have a faster rate of convergence. A careful comparison of Fig. F.1 and 6.6 fails to show conclusively which algorithm has the better performance. For this example at least, the user would have been just as well off to use the simpler Constrained-LMS processor.

Readers interested in the direct substitution method should consult Saradis, *et. al*. [24] and Mantey and Griffiths [18], [19].

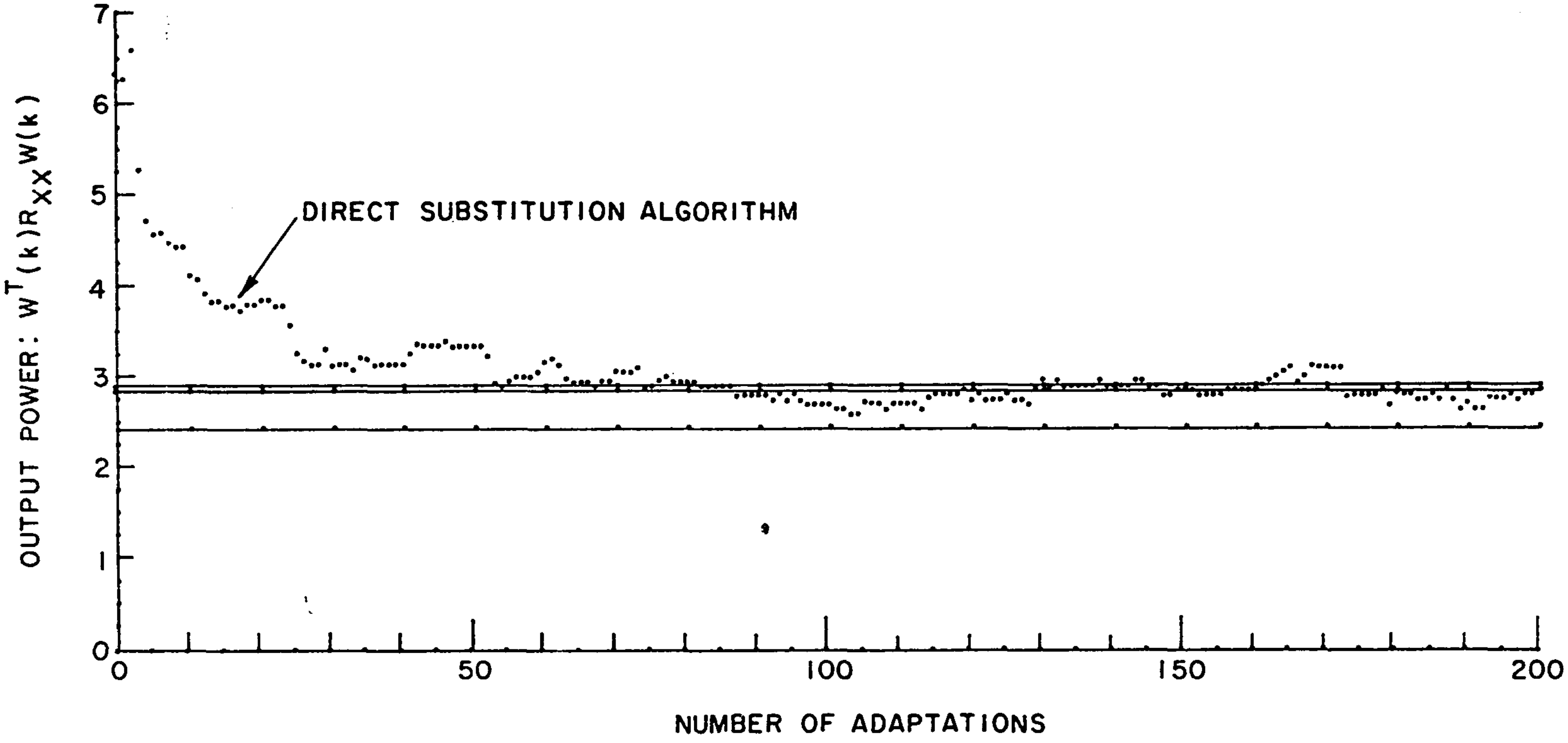

Fig. F.1. Performance of the direct substitution algorithm on the same data used by the Constrained-LMS algorithm (Fig. 6.6). The horizontal lines are retained for reference.

BIBLIOGRAPHY

1. A. Booker, *et al.*, "Multiple-constraint adaptive filtering", Texas Instruments, Science Services Division, Dallas, Texas, Apr 1969

2. J. E. Brown, III, "Adaptive estimation in nonstationary environments", Ph.D. dissertation in preparation, Stanford Electronics Laboratories, Stanford, Calif.

3. A. E. Bryson, Jr. and Y. C. Ho, *Applied Optimal Control*, Blaisdell Publishing Co., Waltham, Mass., 1969

4. J. Capon, *et al.*, "Multidimensional maximum-likelihood processing of a large aperture seismic array", *Proc. IEEE*, *55*, Feb 1967, pp. 142-211.

5. R. E. Collin and F. J. Zucker, *Antenna Theory, Part I*, McGraw-Hill, New York, 1969

6. T. P. Daniell, "Adaptive estimation with mutually correlated training samples", SEL-68-083 (TR No. 6778-4), Stanford Electronics Laboratories, Stanford, Calif., Aug 1968

7. P. M. DeRusso, *et al.*, *State Variables for Engineers*, John Wiley & Sons, New York, 1965

8. W. H. Fleming, *Functions of Several Variables*, Addison-Wesley, Reading, Mass., 1965.

9. F. R. Gantmacher, *The Theory of Matrices*, Chelsea Press, New York, 1959.

10. I. J. Good and K. Doog, "A paradox concerning rate of information", *Information and Control*, *1*, 2, May 1958, pp. 113-126.

11. L. J. Griffiths, "Signal extraction using real-time adaptation of a linear multichannel filter", SEL 68-017, (TR No. 6788-1), Stanford Electronics Laboratories, Stanford, Calif., Feb 1968

12. L. J. Griffiths, "A simple adaptive algorithm for real-time processing in antenna arrays", *Proc. IEEE*, *57*, 10, Oct 1969, p. 1696.

13. R. Kneipfer and R. Hilt, "System description of NUSL's iterative adaptive beamformer (ITAB)", NUSL Tech. Memo. No. 2242-63-70, Naval Undersea Sound Laboratory, Fort Trumbull, New London, Conn., Mar 1970

14. R. T. Lacoss, "Adaptive combining of wideband array data for optimal reception", IEEE Trans. Geoscience Elect., GE-6, 2, May 1968

15. A. Lender, "Decision-directed adaptive equalization technique for high-speed data transmission", Proc 1970 Intl. Conf. on Communications, San Francisco, Jun 1970

16. David G. Luenberger, Optimization by Vector State Methods, John Wiley & Sons, New York, 1969

17. D. G. Luenberger, "Convergence rate of a penalty function scheme", Internal Memo. 69-1, Dept. of Engr-Econ. Systems, Stanford University, Stanford,California

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

19. P. E. Mantey and L. J. Griffiths, Manuscript in preparation.

20. J. L. Moschner, "Adaptive filtering with clipped input data", Ph.D. dissertation, Stanford Electronics Laboratories, Stanford, Calif., Jun 1970

21. A. H. Nuttall, "Theory and application of the separable class of random processes", TR 343, Res. Lab.of Electronics, M.I.T., Cambridge, Mass., 1958

22. A. H. Nuttall and D. W. Hyde, "A unified approach to optimum and suboptimum processing for arrays," USL Rept. No. 992, U. S. Naval Undersea Sound Laboratory, Fort Trumbull, New London, Conn., Apr 1969

23. J. B. Rosen, "The gradient projection method for nonlinear programming, pt. I: Linear constraints", J. Soc. Indust. Appl. Math., 8, 1, Mar 1960, p. 181

24. G. N. Saradis, et al., "Stochastic approximation algorithms for system identification, estimation, and decomposition of mixtures", IEEE Trans. of Syst. Sci. and Cyber., SSC-5, 1, Jan 1969

25. J. L. Steinberg and J. Lequeux, Radio Astronomy, McGraw-Hill, New York (translated by R. N. Bracewell)

26. H. L. Van Trees, Detection, Estimation, and Modulation Theory, Part I, John Wiley & Sons, New York, 1968

27. B. Widrow, "A study of rough amplitude quantization by means of Nyquist sampling theory", IRE Trans. Professional Group on Circuit Theory, CT-3, 4, Dec 1956

28. B. Widrow, "Adaptive filters I: fundamentals", SEL-66-126 (TR No. 6764-6), Stanford Electronics Laboratories, Stanford, Calif., Dec 1966

29. B. Widrow and M. E. Hoff, Jr., "Adaptive switching circuits", IRE WESCON Conv. Rec., Part 4, pp. 96-104, 1960

30. B. Widrow, et al., "Adaptive antenna systems", Proc. IEEE, 55, 12, Dec 1967