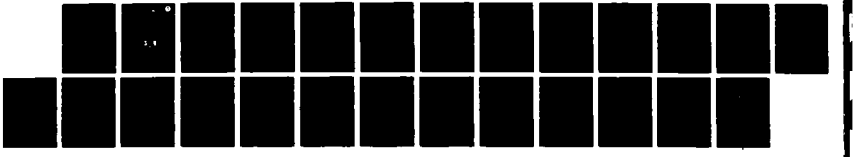
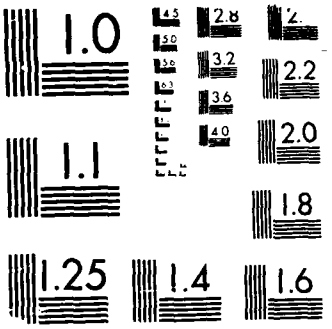
AD-A193 130 GRAM-SCHMIDT IMPLEMENTATION OF A LINEARLY CONSTRAINED ADAPTIVE ARRAY(U) NAVAL RESEARCH LAB WASHINGTON DC K GERLACH 26 FEB 88 1/1

UNCLASSIFIED F/G 17/9 NL





MICROCOPY RESOLUTION TEST CHART

Naval Research Laboratory
Washington, DC 20375-5000

NRL Report 9056

Gram-Schmidt Implementation of a Linearly Constrained Adaptive Array

KARL GERLACH

Target Characteristics Branch
Radar Division

February 26, 1988

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION / AVAILABILITY OF REPORT Approved for public release; distribution unlimited.
2b DECLASSIFICATION / DOWNGRADING SCHEDULE	
4 PERFORMING ORGANIZATION REPORT NUMBER(S) NRL Report 9056	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a NAME OF PERFORMING ORGANIZATION Naval Research Laboratory 6b OFFICE SYMBOL (If applicable) NRL	7a NAME OF MONITORING ORGANIZATION Space and Naval Warfare Systems Command
6c ADDRESS (City, State, and ZIP Code) Washington, DC 20375-5000	7b ADDRESS (City, State, and ZIP Code) Washington, DC 20363-5100
8a NAME OF FUNDING / SPONSORING ORGANIZATION Chief of Naval Research 8b OFFICE SYMBOL (If applicable) CNO	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c ADDRESS (City, State, and ZIP Code) Arlington, VA 22217-5000	10 SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO: 62712N PROJECT NO: (see page ii) TASK NO: WORK UNIT ACCESSION NO: (see page ii)

11 TITLE (Include Security Classification)
Gram-Schmidt Implementation of a Linearly Constrained Adaptive Array

12 PERSONAL AUTHOR(S)
Gerlach, Karl

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
Interim	FROM _____ TO _____	26 February 1988	25

16 SUPPLEMENTARY NOTATION

17 COSATI CODES			18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Adaptive filter, Radar, Adaptive cancellation

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

A Gram-Schmidt (GS) implementation of the linearly constrained adaptive algorithm proposed by Frost is developed. This implementation is shown to be equivalent to the technique developed ~~by Jim, Griffiths, and Buckley~~ whereby the constrained problem is reduced to an unconstrained problem. In addition, analytical results are presented for the convergence rate when the Sampled Matrix Inversion (SMI) algorithm is employed. It had been previously shown that the steady state solution for the optimal weights is identical for both constrained and reduced unconstrained problems. ~~In~~ this report ~~it is~~ shown that if the SMI or GS algorithms are employed, then the transient weighting vector solution for the constrained problem is identical to equivalent transient weighting vector solution for the reduced unconstrained implementation.

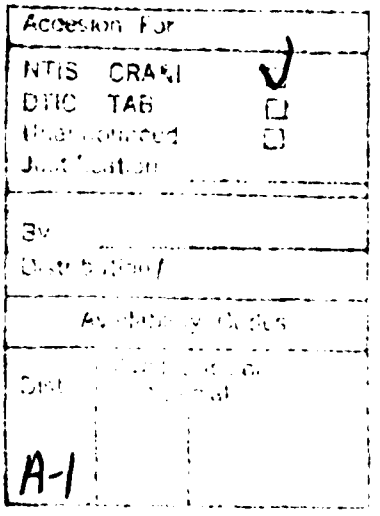
20 DISTRIBUTION / AVAILABILITY OF ABSTRACT ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED
22a NAME OF RESPONSIBLE INDIVIDUAL Dr. Karl Gerlach	22b TELEPHONE (Include Area Code) (202) 767-3599 22c OFFICE SYMBOL 5340.1G

DD Form 1473, JUN 86 Previous editions are obsolete SECURITY CLASSIFICATION OF THIS PAGE

10. SOURCE OF FUNDING NUMBERS

Project No.	Work Unit Accession No.
RS12-131-001	DN480-549
XF12-141-100	DN380-101

CONTENTS



GRAM-SCHMIDT IMPLEMENTATION OF A LINEARLY CONSTRAINED ADAPTIVE ARRAY

I. INTRODUCTION

Unconstrained adaptive antenna arrays for certain external noise interference scenarios can result in the cancellation of a desired signal along with the cancellation of the interfering signals. Frost [1,2] introduced a constrained optimization procedure such that certain main beam antenna properties are maintained during the adaptation process thus preserving the desired signal.

Consider an N input adaptive antenna array as shown in Fig. 1. We define a vector of weights $\mathbf{w}$ as

$$\mathbf{w} = (w_1, w_2, \cdots, w_N)^T,$$

where T denotes the vector transpose operation. Frost [1] shows that if the weights are constrained to satisfy the following linear constraint equation

$$C^t\mathbf{w} = \mathbf{f}, \tag{1}$$

where

M is the number of constraint equations,
C is the $N \times M$ constraint matrix,
$\mathbf{f}$ is the $M \times 1$ column constraint vector,

and t denotes the conjugate transpose vector operation, then the average output noise power residue of $z = \mathbf{w}^t\mathbf{x}$ is minimized if

$$\mathbf{w} = R_{xx}^{-1}C\,(C^tR_{xx}^{-1}C)^{-1}\mathbf{f}, \tag{2}$$

where

$$\mathbf{x} = (x_1, x_2, \cdots, x_N)^T,$$

$R_{xx} = E\{\mathbf{x}\mathbf{x}^t\}$ is the input covariance matrix, and $E\{\cdot\}$ denotes the expected value.

In this report we develop an open-loop Gram-Schmidt (GS) implementation of $\mathbf{w}^t\mathbf{x}$ where $\mathbf{w}$ is defined by Eq. (2) or equivalently an implementation such that

$$z = \mathbf{f}^t(C^tR_{xx}^{-1}C)^{-1}C^tR_{xx}^{-1}\mathbf{x}. \tag{3}$$

The GS implementation of a linearly constrained adaptive array offers many advantages. The GS open-loop technique has been shown to yield superior performance simultaneously in arithmetic

Manuscript approved June 18, 1987.

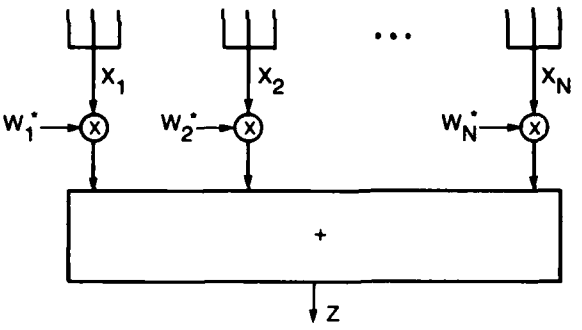


Fig. 1 — Adaptive array

efficiency, stability, and convergence rate [3-7] over other adaptive algorithms. In particular, the stability of the GS algorithm is enhanced because it does not require the calculation of an inverse covariance matrix as does the Sample Matrix Inversion (SMI) algorithm [8]. Also the GS canceller algorithm is very suitable for a non-stationary noise environment because the adaptive weights can be updated in a numerically efficient manner, using "sliding window" or systolic techniques on the input data instead of "batch" processing.

Jim, Griffiths, and Buckley [9,10,11] have shown that the constrained minimization problem can be reduced to an unconstrained implementation called the generalized sidelobe canceller (GSC). In this report, we develop an equivalent implementation of this technique. It had been previously shown that the steady state solution for the optimal weights is identical for both constrained and reduced unconstrained problems. In this report, it is shown that if the SMI or GS algorithms are employed, then the transient weighting vector solution for the constrained problem is identical to equivalent transient weighting vector solution for the reduced unconstrained implementation. In Sections II through V, we develop the basic building blocks for the GS implementation of the linearly constrained adaptive array. In Section VI, the implementation is presented. In Section VII, the special case when there is only one constraint is discussed. In Section VIII the multiple constraint implementation is significantly simplified. In Section IX, the Jim, Griffiths, and Buckley implementation is derived. Finally, in Section X, analytical results are presented for the convergence rate of the constrained minimization implementation when the SMI algorithm is employed.

II. GS CANCELLERS

Consider the general N-input open-loop GS canceller structure as seen in Fig. 2(a). Let $x_1, x_2, \cdots, x_N$ represent the complex data in the 1st, 2nd, Nth channels, respectively. We call the leftmost input (x_1), the main channel, and we call the remaining $N-1$ inputs the auxiliary channels. The canceller operates so as to decorrelate the auxiliary inputs one at a time from the other inputs by using the basic two-input GS processor shown in Fig. 2(b). For example, as seen in Fig. 2(a), in the first level of decomposition, x_N is decorrelated with $x_1, x_2, \cdots, x_{N-1}$. Next, the output channel that results from decorrelating x_N with x_{N-1} is decorrelated with the other outputs of the first level GSs. The decomposition proceeds until a final output channel is generated. If the decorrelation weights in each of the two input GSs are computed from an infinite number of input samples, then this output channel is totally decorrelated with the input: $x_2, x_3, \cdots, x_N$.

Let $x_n^{(m)}$ represent the outputs of the two input GSs on the $m-1$th level. Then outputs of the two-input GSs at the mth level are given by

$$x_n^{(m+1)} = x_n^{(m)} - w_n^{(m)} x_{N-m+1}^{(m)}, \quad \begin{matrix} n = 1, 2, \cdots, N-m, \\ m = 1, 2, \cdots, N-1. \end{matrix} \tag{4}$$

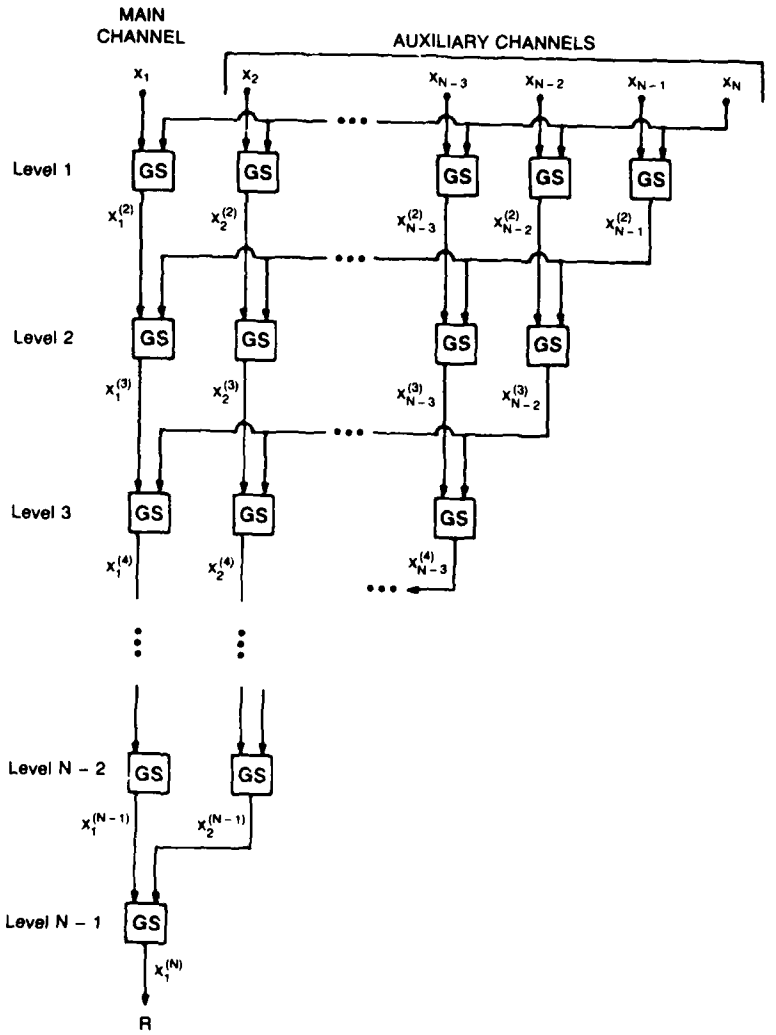


Fig. 2(a) — GS structure

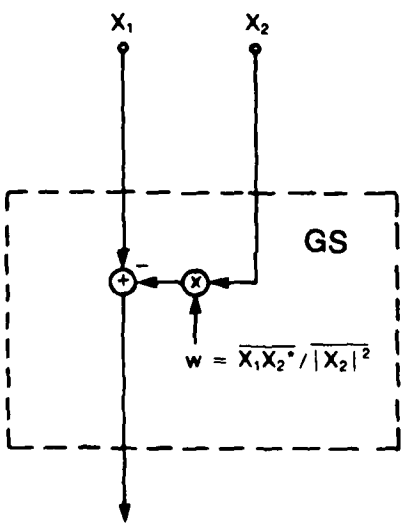


Fig. 2(b) — Basic two-input GS canceller

Note that $x_n^{(1)} = x_n$. The weight $w_n^{(m)}$, seen in Eq. (1), is computed so as to decorrelate $x_n^{(m+1)}$ with $x_{N-m}^{(m)}$. For K input samples per channel, this weight is estimated as

$$w_n^{(m)} = \frac{\sum_{k=1}^{K} x_{N-m+1}^{(m)*}(k) x_n^{(m)}(k)}{\sum_{k=1}^{K} |x_{N-m+1}^{(m)}(k)|^2}, \tag{5}$$

where * denotes the complex conjugate and $|\cdot|$ is the complex magnitude. Here k indexes the time-sampled data.

We simplify the N-input GS canceller structure by the representation as seen in Fig. 3.

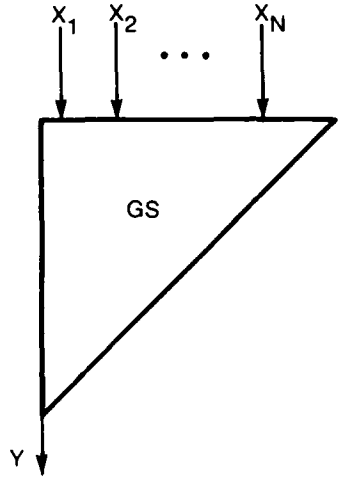


Fig. 3 — GS representation

III. NORMALIZED GS CANCELLER

The GS canceller, as represented in Fig. 3, effectively weights the vector $\mathbf{x}$ with a vector $\mathbf{w} = (w_1, \cdots, w_N)^T$ such that $y = \mathbf{w}^t\mathbf{x}$, where

$$\mathbf{w} = \mu R_{xx}^{-1} \begin{bmatrix} 1 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}, \tag{6}$$

$w_1 = 1$, and μ is a scalar constant to be determined. Because $w_1 = 1$, the scalar constant μ has a specific value. If

$$R_{xx}^{-1} = (r^{(nm)}), \qquad n, m = 1, 2, \cdots, N, \tag{7}$$

where $r^{(nm)}$ are the elements of R_{xx}^{-1}, then we can show

$$\mu = \frac{1}{r^{(11)}}. \tag{8}$$

Hence

$$\mathbf{w} = R_{xx}^{-1} \begin{bmatrix} 1/r^{(11)} \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}. \tag{9}$$

Consider a configuration where we normalize the output y by the average power of y as seen in Fig. 4. Note that the average power after normalization is *not* one. Thus

$$z = \frac{1}{E\{|y|^2\}}\, y = \frac{1}{E\{|y|^2\}}\, \mathbf{w}^t \mathbf{x}. \tag{10}$$

However,

$$\begin{aligned} E\{|y|^2\} &= E\{|\mathbf{w}^t \mathbf{x}|^2\} \qquad (11) \\ &= \mathbf{w}^t E\{\mathbf{x}\mathbf{x}^t\}\mathbf{w} \\ &= \mathbf{w}^t R_{xx} \mathbf{w} \\ &= (1/r^{(11)*}, 0, \cdots 0) R_{xx}^{-1} R_{xx} R_{xx}^{-1} \begin{bmatrix} 1/r^{(11)} \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \\ &= (1/r^{(11)*}, 0, 0, \cdots, 0) R_{xx}^{-1} \begin{bmatrix} 1/r^{(11)} \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix}. \end{aligned}$$

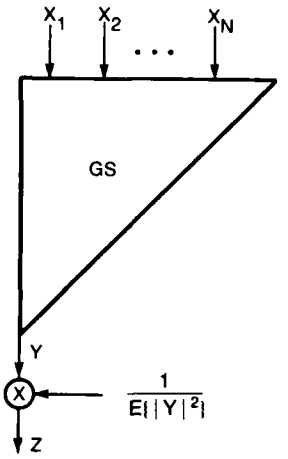


Fig. 4 — Normalized GS canceller

Now since R_{xx} is hermitian, then R_{xx}^{-1} is hermitian and r_{11} is real. Thus

$$E\{|y|^2\} = \frac{1}{r^{(11)}}. \tag{12}$$

Hence

$$z = r^{(11)}\mathbf{w}^t\mathbf{x}, \tag{13}$$

or substituting Eq. (9) into Eq. (13),

$$z = (1, 0, \cdots, 0)\, R_{xx}^{-1} \cdot \mathbf{x}. \tag{14}$$

IV. NORMALIZED FAST ORTHOGONALIZATION NETWORK (FON)

A FON is a numerically efficient implementation of a complete GS network where each input is orthogonalized with every other input [12]. In essence, the FON implements the network seen in Fig. 5. The ordering of the input channels for decorrelation as seen in Fig. 5 was arbitrary. Reference 12 shows that the input channels can be ordered so as to greatly reduce the required number of arithmetic operations. If there were no logic behind choosing the ordering of the input channels, it can be shown that the number of weights that are calculated by using this decorrelation procedure is $0.5N^2$ $(N - 1)$. In Ref. 12 an algorithm was developed that requires approximately $1.5N$ $(N - 1)$ weights for the same decorrelation process.

We represent the FON implementation of Fig. 5 as seen in Fig. 6. Consider an implementation where each output of the FON is normalized with respect to the power of that output as shown in Fig. 7. We call this configuration a normalized FON. If we define $\mathbf{z} = (z_1, z_2, \cdots, z_N)^T$, then we can show that the normalized FON is equivalent to multiplying the vector $\mathbf{x}$ by an $N \times N$ matrix of weights $\mathbf{w}$ such that

$$\mathbf{z} = \mathbf{w}^t\mathbf{x}, \tag{15}$$

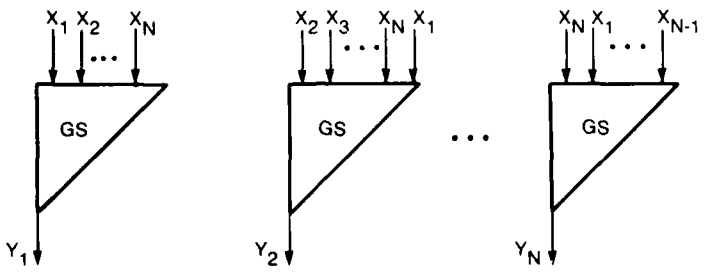


Fig. 5 — Orthogonalization network

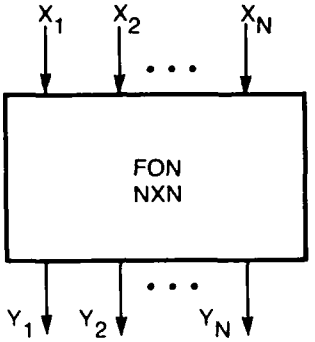


Fig. 6 — FON representation

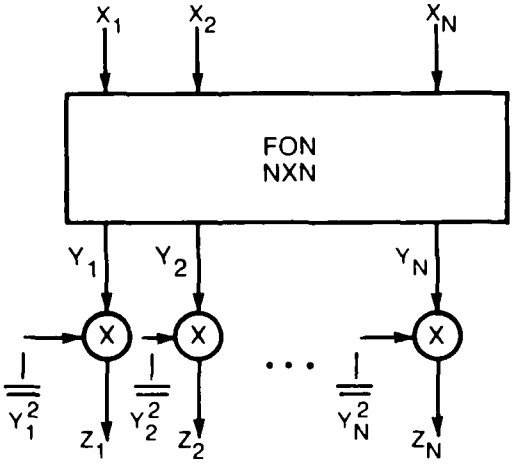


Fig. 7 — Normalized FON

where

$$\mathbf{w} = R_{xx}^{-1}. \tag{16}$$

Note that we used the methodology of the previous section. Hence

$$\mathbf{z} = R_{xx}^{-1}\mathbf{x}. \tag{17}$$

We represent a normalized FON as shown in Fig. 8.

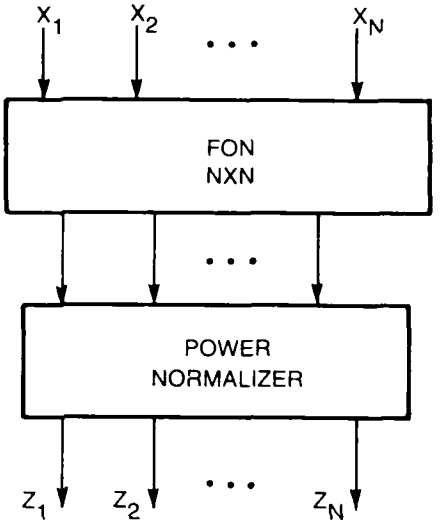


Fig. 8 — Normalized FON representation

V. NORMALIZED REDUCED FONS

A normalized reduced FON orthogonalizes only a fraction of the inputs with respect to one another. For example if we desire only to orthogonalize $x_1, x_2, \cdots, x_M$, where $M < N$, then a reduced FON would efficiently implement the configuration as seen in Fig. 9. If $\mathbf{z} = (z_1, z_2, \cdots, z_M)^T$, then we can show that a normalized reduced FON is equivalent to multiplying the vector $\mathbf{x}$ by an $M \times N$ matrix such that

$$\mathbf{z} = \mathbf{w}^t\mathbf{x} \tag{18}$$

and

$$\mathbf{w} = R_{xx}^{-1}I_{N,M}, \tag{19}$$

where

$$I_{N,M} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \cdot \\ \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ 0 & 0 & 0 \end{bmatrix} = N \times M \text{matrix}.$$

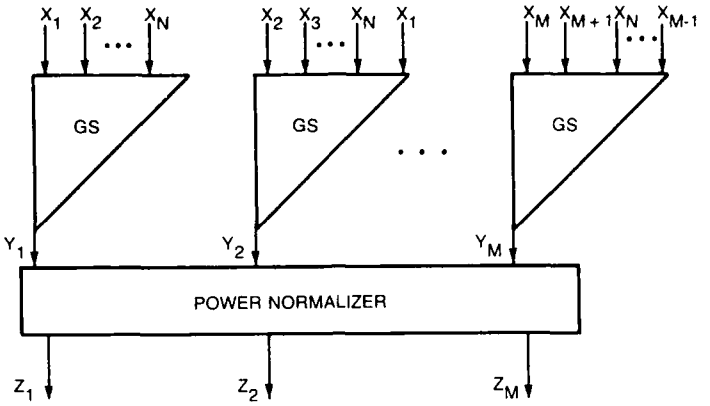


Fig. 9 — Normalized reduced FON

For example, $I_{3,2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$.

Hence

$$\mathbf{z} = I_{M,N} R_{xx}^{-1} \mathbf{x}. \tag{20}$$

We represent a normalized reduced FON as seen in Fig. 10.

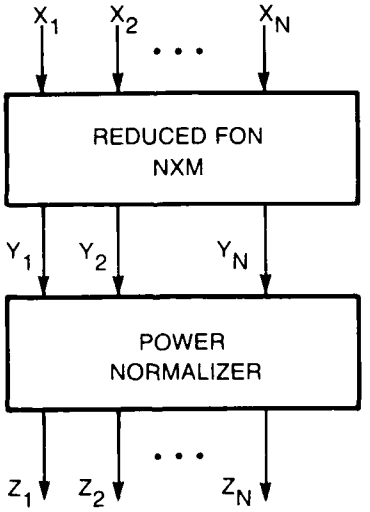


Fig. 10 — Normalized reduced FON representation

VI. GS IMPLEMENTATION

We now present the GS implementation of the linearly constrained adaptive array. First, we define an $N \times N$ augmented matrix, C_{aug}, such that

$$C_{aug} = [C \mid D] \tag{21}$$

where D is an $N \times (N - M)$ matrix such that C_{aug} is nonsingular and hence invertible. We discuss the choice of D in more detail in Section IX.

The GS implementation of the linearly constrained adaptive array is shown in Fig. 11. Note that

$$\begin{aligned} \mathbf{x} &= (x_1, x_2, \cdots, x_N)^T \\ \mathbf{u} &= (u_1, u_2, \cdots, u_N)^T \\ \mathbf{v} &= (v_1, v_2, \cdots, v_M)^T \\ \mathbf{y} &= (y_1, y_2, \cdots, y_M)^T \\ \mathbf{p} &= (p_1, p_2, \cdots, p_M)^T \\ z &= \text{scalar}. \end{aligned} \tag{22}$$

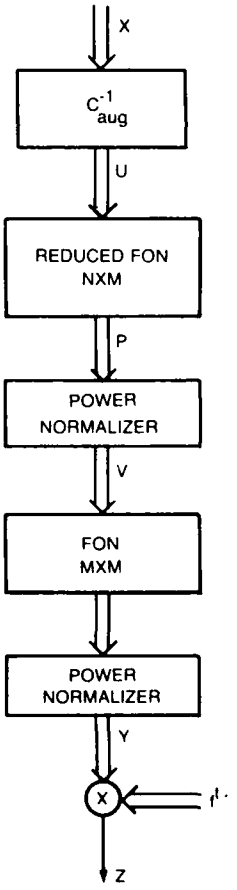


Fig. 11 — GS implementation of a linearly constrained adaptive array

From our preceding discussion in FONs and reduced FONs, we know that

$$\mathbf{u} = C_{aug}^{-1}\mathbf{x}, \tag{23}$$

$$\mathbf{v} = I_{M,N}\, R_{uu}^{-1}\mathbf{u}, \tag{24}$$

$$\mathbf{y} = R_{vv}^{-1}\,\mathbf{v}, \tag{25}$$

$$z = \mathbf{f}^t\,\mathbf{y}. \tag{26}$$

Now

$$R_{uu} = E\{C_{aug}^{-1}\mathbf{x}\cdot(C_{aug}^{-1}\mathbf{x})^t\} = C_{aug}^{-1}\,R_{xx}\,C_{aug}^{t^{-1}}, \tag{27}$$

or

$$R_{uu}^{-1} = C_{aug}^{t}\,R_{xx}^{-1}\,C_{aug}. \tag{28}$$

Hence

$$\mathbf{v} = I_{M,N}\,C_{aug}^{t}\,R_{xx}^{-1}\,C_{aug}\cdot C_{aug}^{-1}\mathbf{x} \tag{29}$$

$$= I_{M,N}\,C_{aug}^{t}\,R_{xx}^{-1}\mathbf{x}.$$

We can show that

$$C^t = I_{M,N}\,C_{aug}^{t}. \tag{30}$$

Thus

$$\mathbf{v} = C^t\,R_{xx}^{-1}\,\mathbf{x}. \tag{31}$$

Now

$$R_{vv} = E\{(C^t R_{xx}^{-1}\mathbf{x})\,(C^t R_{xx}^{-1}\mathbf{x})^t\}$$

$$= C^t R_{xx}^{-1}\cdot E\{\mathbf{x}\mathbf{x}^t\}\,R_{xx}^{-1}C$$

$$= C^t R_{xx}^{-1}\cdot R_{xx}\cdot R_{xx}^{-1}C$$

$$= C^t R_{xx}^{-1}C.$$

Thus

$$R_{vv}^{-1} = (C^t R_{xx}^{-1}C)^{-1}. \tag{33}$$

Hence

$$\mathbf{y} = (C^t R_{xx}^{-1} C)^{-1} \cdot \mathbf{v} \tag{34}$$

$$= (C^t R_{xx}^{-1} C)^{-1} \cdot C^t R_{xx}^{-1} \mathbf{x}.$$

Finally, by using Eq. (26),

$$z = \mathbf{f}^t (C^t R_{xx}^{-1} C)^{-1} C^t R_{xx}^{-1} \mathbf{x}. \tag{35}$$

Note that this is the same weighting of $\mathbf{x}$ as given by Eq. (3).

VII. SINGLE CONSTRAINT IMPLEMENTATION EXAMPLE

Note that when $M = 1$, we set $z = p$ (see Fig. 11) and the function blocks after p are not used. This results because the $M \times M$ FON seen in Fig. 11 is not used, $f = 1$, and data passing through successive power normalizations is unchanged. We can show that

$$z = p = \frac{1}{C^t R_{xx}^{-1} C} C^t R_{xx}^{-1} \mathbf{x}. \tag{36}$$

For this case $C = (c_1, c_2, \cdots, c_N)^T = \mathbf{c}$, a vector. Note that the reduced FON is just a single GS canceller. Hence for a single constraint, the configuration is shown in Fig. 12.

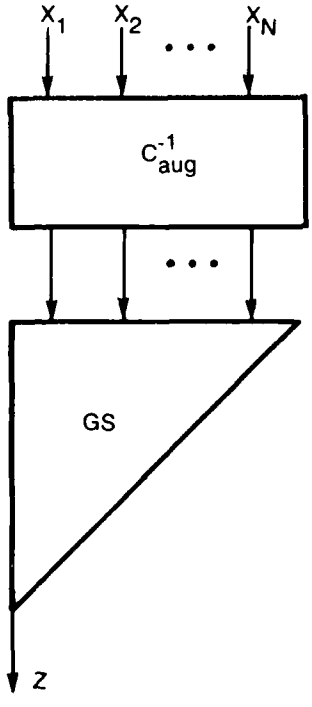


Fig. 12 — GS implementation for a single linear constraint

We arbitrarily set $c_1 = 1$ and augment C as shown below:

$$C_{aug}^{-1} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ c_2 & 1 & 0 & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ c_N & 0 & 0 & \cdots & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -c_2 & 1 & 0 & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ -c_N & 0 & 0 & \cdots & 1 \end{bmatrix}. \tag{37}$$

Hence the single constraint processor of Fig. 12 can be implemented as shown in Fig. 13.

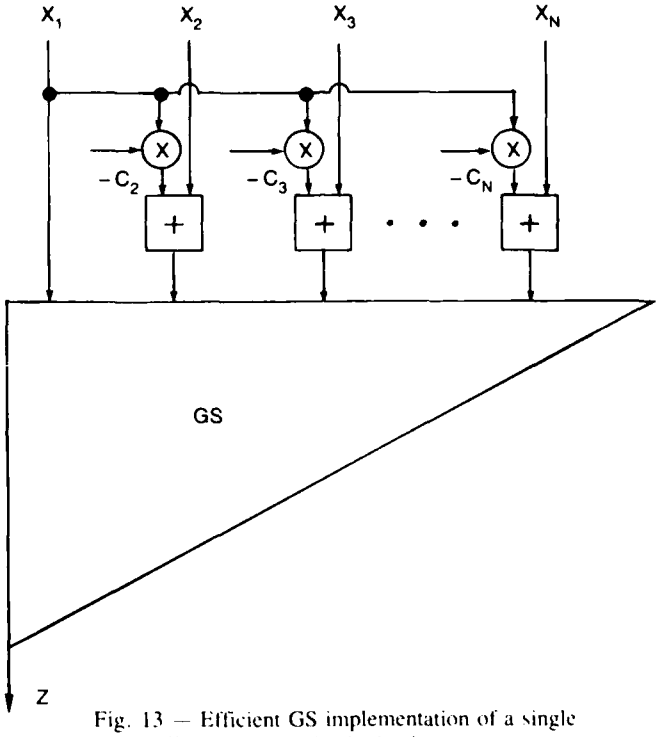


Fig. 13 — Efficient GS implementation of a single linearly constrained adaptive array

VIII. SIMPLIFIED MULTIPLE CONSTRAINT IMPLEMENTATION

The GS structure of the linearly constrained array given in Fig. 11 can be significantly simplified as follows. The $N \times M$ reduced FON structure can be functionally decomposed as shown in Fig. 14. Here, we denote $u_1, u_2, \cdots, u_M$ as the inputs associated with the outputs of the $N \times M$ reduced FON. We call $u_1, u_2, \cdots, u_M$ the primary channels and write them as an M length vector, $\mathbf{u}'$. We also denote $u_{M+1}, u_{M+2}, \cdots, u_N$ as the auxiliary channels and write them as an $N - M$ length vector, $\mathbf{u}_{aux}$. Note that embedded in the $N \times M$ reduced FON is an $M \times M$ FON. Hence Fig. 11 can be redrawn as seen in Fig. 15.

Now it is easy to show that the operation of two successive $M \times M$ normalized FONs is equivalent to multiplying the input data to these FONs by an $M \times M$ identity matrix. Hence, the implementation seen in Fig. 15 reduces to that shown in Fig. 16. Moreover, this structure can be

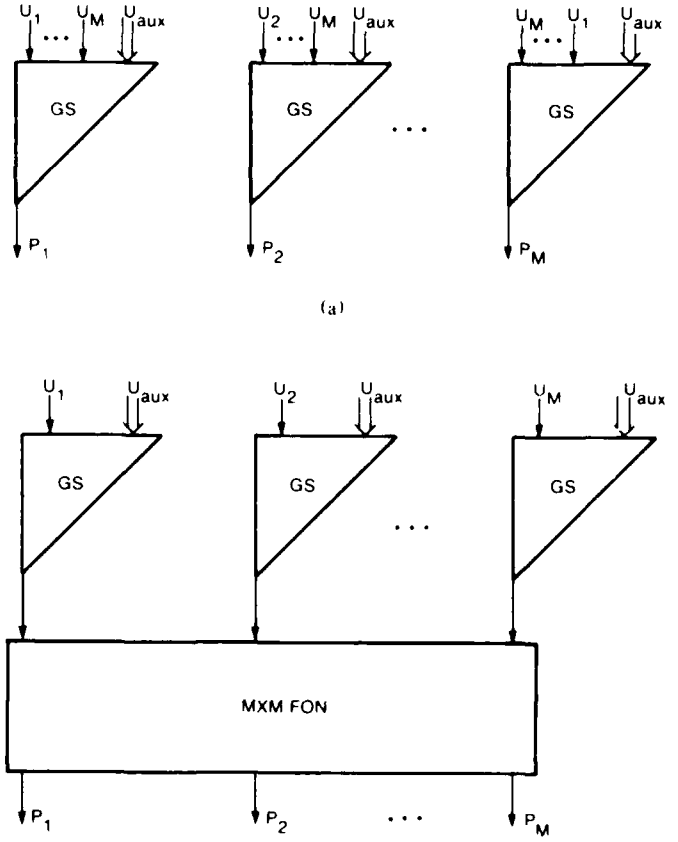


Fig. 14 — Equivalent $N \times M$ FON structures. (a) Reduced $N \times M$ FON; (b) Functional decomposition of reduced $N \times M$ FON

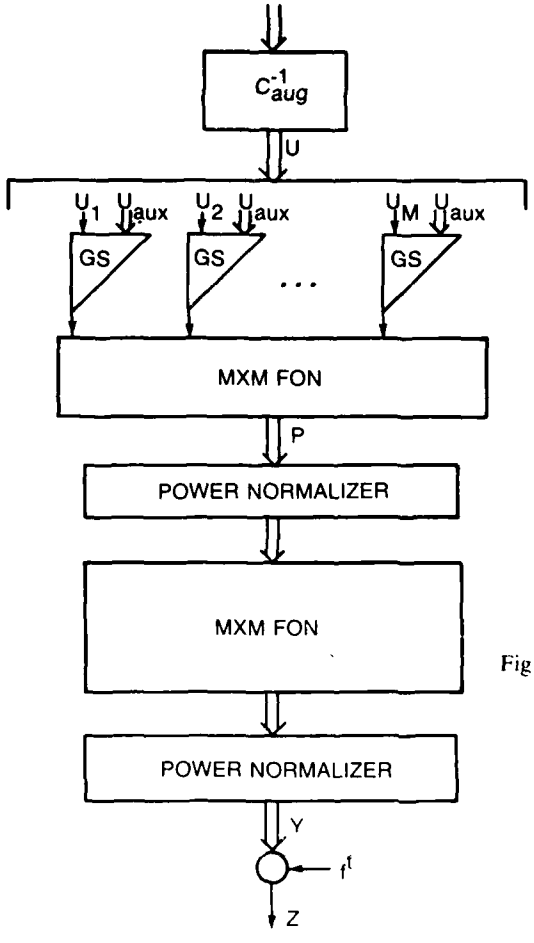


Fig. 15 — Functional equivalent of canceller seen in Fig. 11

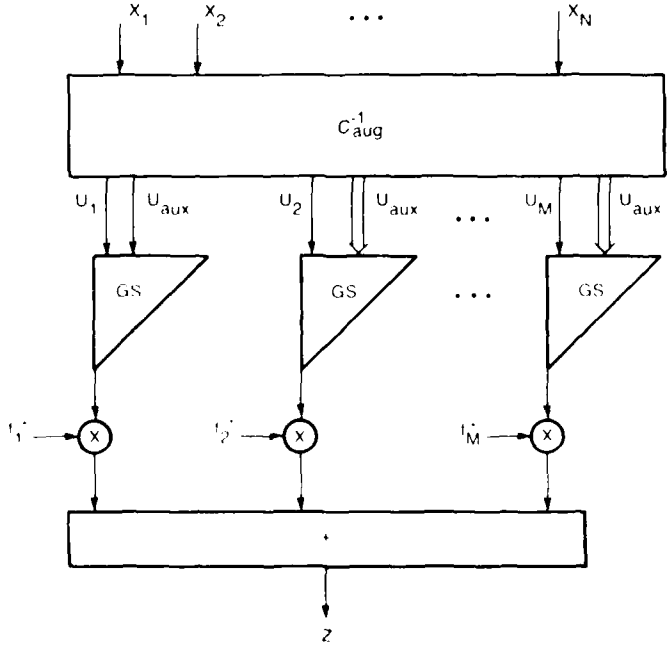


Fig. 16 — Simplified equivalent of canceller seen in Fig. 11

further simplified to the structure shown in Fig. 17, where the main channel input to the GS canceller, u_{in}, is given by

$$u_{in} = \sum_{m=1}^{M} f_m^* u_m . \tag{38}$$

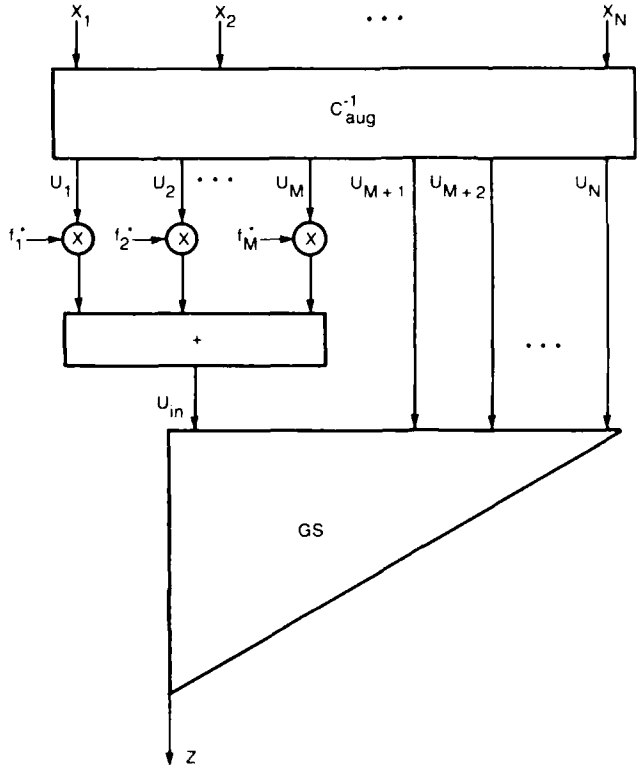


Fig. 17 — Simplified GS implementation of a linearly constrained adaptive array

IX. THE AUGMENTED CONSTRAINT MATRIX

Here we show that the generalized sidelobe canceller (GSC) implementation of a linearly constrained array presented in Refs. 9, 10, and 11 is equivalent to the implementation discussed in Section VIII. We define

$$C_{aug}^{-1} = \left[\begin{array}{c} A \\ \hline B \end{array}\right] \tag{39}$$

where A is an $M \times N$ matrix and B is an $(N - M) \times N$ matrix. Thus

$$C_{aug}^{-1} C_{aug} = \left[\begin{array}{c} A \\ \hline B \end{array}\right] [C \mid D] = \left[\begin{array}{c|c} AC & AD \\ \hline BC & BD \end{array}\right] = \left[\begin{array}{c|c} I_M & \mathbf{0} \\ \hline \mathbf{0} & I_{N-M} \end{array}\right] \tag{40}$$

where I_M and I_{N-M} are $M \times M$ and $(N - M) \times (N - M)$ identify matrices, respectively. As a result

$$AC = I_M , \tag{41}$$

$$AD = \mathbf{0}, \tag{42}$$

$$BC = \mathbf{0}, \tag{43}$$

$$BD = I_{N-M} . \tag{44}$$

The solutions for A and D using pseudoinverses are

$$A = (C^t C)^{-1} C^t + H \tag{45}$$

$$D = B^t (BB^t)^{-1} + G \tag{46}$$

where H is any $M \times N$ matrix satisfying the condition

$$HC = \mathbf{0} \tag{47}$$

and G is any $(N - M) \times N$ matrix satisfying the equation

$$BG = \mathbf{0}. \tag{48}$$

Note from Eq. (43) that rows of B are orthogonal to C, the constraint matrix. In the literature [10,11], B is called the blocking matrix. We can eliminate having to find C_{aug}^{-1} by merely defining a B that satisfies Eq. (42) and an A that is given by Eq. (45) where H satisfies (47).

The linearly constrained canceller now has the form as shown in Fig. 18. If we set $H = 0$ and define w_q to be the quiescent weighting (no external noise, $R_{xx} = I$), then

$$w_q = C^t (C^t C)^{-1} f . \tag{49}$$

Thus the linearly constrained processor can be implemented as shown in Fig. 19, which is identical to the GSC presented in Refs. 9, 10, and 11.

X. CONVERGENCE RATE

One technique for estimating the optimal weighting vector for a linearly constrained adaptive array is the Sampled Matrix Inversion (SMI) algorithm [8]. We will show in this section that this technique has fast convergence properties when applied to an adaptive array with linear constraints. This open loop algorithm is implemented by estimating the input covariance matrix, R_{xx}, using the samples of data in the input channels. The estimated R_{xx} is then substituted into Eq. (2) and the constrained optimal weights, $\mathbf{w}$, are then estimated. Call this estimate $\hat{\mathbf{w}}_{\mathrm{SMI}}$. It is easy to show using an analysis similar to that presented in Section VI that the multiple constraint GS implementation using the same samples as the linearly constrained SMI algorithm yields an exact equivalent estimated linear weighting vector, $\hat{\mathbf{w}}_{\mathrm{GS}}$, as the SMI algorithm; i.e., $\hat{\mathbf{w}}_{\mathrm{GS}} = \hat{\mathbf{w}}_{\mathrm{SMI}}$. This is done by merely substituting $\hat{R}_{xx}$ for $E\{x^* x^T\} = R_{xx}$ in the equations given in Section IV. Hence the GS and SMI implementations of the linearly contrained adaptive array are identical in the transient state as well as the steady state. The convergence rate properties of GS implementation of a linearly constrained array (and also the SMI implementation) as shown in Fig. 18 can be easily analyzed. This is because the open-loop

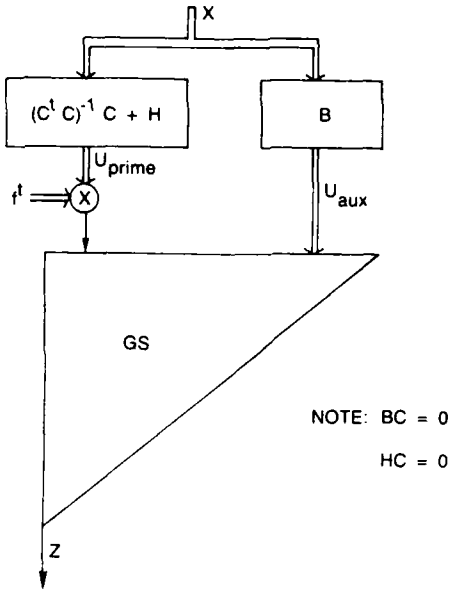


Fig. 18 — Special case of the linearly constrained canceller implementation

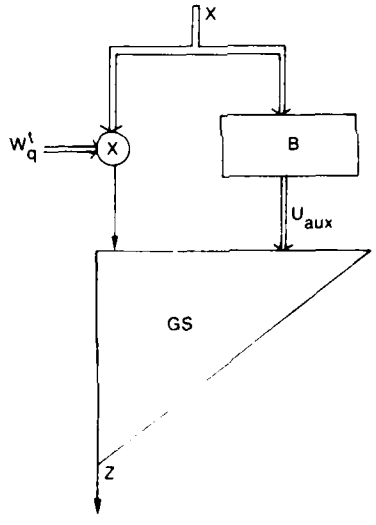


Fig. 19 — GSC configuration

GS canceller is the exact equivalent of the mainbeam gain constrained open-loop SMI algorithm [7] whose convergence properties are well known [8,13]. The gain in the steering vector direction is constrained to equal one, which is equivalent to setting the weight in the main channel equal to one for the GS canceller. We quote these convergence results. Let there be L input channels and K zero mean Gaussian samples per channel, where the samples are independent from time sample to time sample across all channels. Let $\hat{\mathbf{w}}$ be the estimate of the optimum weights $\mathbf{w}_{opt}$ using the SMI algorithm, and let R be the input covariance matrix (including main and auxiliary channels). Define

$$\sigma^2 = \hat{\mathbf{w}}^t \, R \, \hat{\mathbf{w}}, \tag{50}$$

$$\sigma^2_{\min} = E\{\mathbf{w}^t_{opt} \, R \, \mathbf{w}_{opt}\}, \tag{51}$$

and

$$z = \frac{\sigma^2}{\sigma^2_{\min}}. \tag{52}$$

We note that σ^2 is a random variable and the output noise power residue caused by finite sampling when the weights are applied to a data set independent of the data set used to calculate the weights.

Under the conditions stated, Brennan and Reed [13] showed that z has the following probability density function (p.d.f.)

$$p(z) = \begin{cases} \dfrac{K!}{(L-2)!\ (K-L+1)!} \quad \dfrac{(z-1)^{L-2}}{z^{K+1}}, & 1 \le z < \infty. \\ 0, \qquad \text{otherwise.} \end{cases} \tag{53}$$

The mean of z is given by

$$E\{z\} = \frac{K}{K-L+1}. \tag{54}$$

We can apply these results to the adaptive linearly constrained array implementation shown in Fig. 18. If the input channels are zero mean, Gaussian, and independent from time sample to time sample, then the output channels after transformation by the $N \times N$ matrix, C^{-1}_{aug}, are also zero mean Gaussian and independent from time sample to time sample. Moreover, u_{in}, as given by Eq. (38), is also a zero mean Gaussian random variable. Hence the inputs to the GS canceller satisfy the conditions given by the Brennan and Reed analysis [13].

For the constrained implementation, $L = N - M + 1$. Thus if z is the normalized output noise power of the linearly constrained array as defined by Eq. (52), then z has the following p.d.f. and mean

$$p(z) = \begin{cases} \dfrac{K!}{(N-M-1)!\ (K-N+M)!} \quad \dfrac{(z-1)^{N-M-1}}{z^{K+1}}, & 1 \le z < \infty. \\ 0, \qquad \text{otherwise.} \end{cases} \tag{55}$$

$$E\{z\} = \frac{K}{K-N+M}. \tag{56}$$

Let $K_{3\text{dB}}$ be the number of samples needed so that averge output noise power is within 3 dB of the optimum. Using Eq. (56), we can show that

$$K_{3\text{dB}} = 2(N-M). \tag{57}$$

Now for an unconstrained adaptive array with N inputs, it has been shown that $K_{3\text{dB}} = 2N - 2$. Hence, we see from Eq. (57) that a constrained array converges faster for $M \ge 2$ than an unconstrained array under the assumptions previously stated.

XI. REFERENCES

1. O.L. Frost III, "An Algorithm for Linearly Constrained Adaptive Array Processing," *Proc. IEEE* **60**(8), 926-935 (1972).

2. R.A. Monzingo and T.W. Miller, *Introduction to Adaptive Arrays* (John Wiley and Sons, New York, 1980), Chap. 8.

3. B.L. Lewis and F.F. Kretschmer, Jr., published work of limited distribution, dating back to Feb. 1974.

4. W.F. Gabriel, "Building Block for an Orthonormal-Lattice-Filter Adaptive Network," NRL Report 8409, July 1980.

5. M.A. Alam, "Orthonormal Lattice Filter—A Multistage, Multichannel Estimation Technique," *Geophysics* **43**, 1368-1383 (1978).

6. B. Friedlander, "Lattice Filters for Adaptive Processing," *Proc. IEEE* **70**(8), 829-867 (1982).

7. Karl Gerlach and F.F. Kretschmer, Jr., "Convergence Rate of a Gram-Schmidt Canceller," NRL Report 9051.

8. I.S. Reed, J.D. Mallett, and L.E. Brennan, "Rapid Convergence Rate in Adaptive Arrays," *IEEE Trans.* **AES-10**, 853-863 (1974).

9. C.W. Jim, "A Comparison of Two LMS Constrained Optimal Array Structures," *Proc. IEEE* **65**(12), 1730-1731 (1977).

10. L.J. Griffiths and C.W. Jim, "An Alternative Approach to Linearly Constrained Adaptive Beamforming," *IEEE Trans.* **AP-30**, 27-34 (1982).

11. K.M. Buckley and L.J. Griffiths, "An Adaptive Generalized Sidelobe Canceller with Derivative Constraints," *IEEE Trans.* **AP-34**, 311-319 (1986).

12. Karl Gerlach, "Fast Orthogonalization Networks," *IEEE Trans.* **AP-34**, 458-462 (1986).

13. L.E. Brennan and I.S. Reed, "Digital Adaptive Arrays with Weights Computed From and Applied to the Same Data Set," Proceedings of the 1980 Adaptive Antenna Symposium, RADC-TR-378, **1**, Dec. 1980.

END

DATE

FILMED

DTIC

JULY 88