RL-TR-92-327
In-House Report
December 1992

THE DYNAMIC BEHAVIOR OF CONSTRAINED ADAPTIVE ARRAY SENSOR PROCESSORS

Jay Scott Goldstein, 1Lt, USAF

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Materiel Command
Griffiss Air Force Base, New York

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

RL-TR-92-327 has been reviewed and is approved for publication.

APPROVED:

ANTHONY S. SZALKOWSKI
Actg Ch, Communications Transmission Division
Command, Control and Communications Directorate

FOR THE COMMANDER:

JOHN A. GRANIERO
Chief Scientist
Command, Control and Communications Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1992	In-House

4. TITLE AND SUBTITLE
THE DYNAMIC BEHAVIOR OF CONSTRAINED ADAPTIVE ARRAY SENSOR PROCESSORS

5. FUNDING NUMBERS
PE - 63401F
PR - 3784
TA - 00
WU - 01

6. AUTHOR(S)
Jay Scott Goldstein, 1st Lt, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Rome Laboratory (C3BA)
525 Brooks Road
Griffiss AFB NY 13441-4505

8. PERFORMING ORGANIZATION REPORT NUMBER
RL-TR-92-327

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Laboratory (C3BA)
525 Brooks Road
Griffiss AFB NY 13441-4505

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Rome Laboratory Project Engineer: 1st Lt Jay Scott Goldstein/C3BA/(315)330-3091

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
The transient behavior of constrained wideband adaptive array sensors is examined with the goal of improving the convergence properties while simultaneously satisfying a requirement for limited computational complexity. The multichannel wideband adaptive filter is derived for the tapped-delay-line, frequency domain, lattice and Gram-Schmidt structures. The mean-square error for each structure is estimated and compared over varying degrees of difficulty with respect to the signal environment. A new multichannel wideband extension of the frequency domain structure is presented, and the performance is demonstrated to surpass that of the conventional tapped-delay-line structure. It is shown via simulation that utilizing the Discrete Cosine Transform, the new structure is capable of converging at speeds which rival the more computationally expensive least square methods at no increase in the required number of adaptive coefficients. The multichannel wideband lattice structure is shown to perform better than the other structures and the least squares method of Sample Matrix Inversion.

14. SUBJECT TERMS
Adaptive filters, adaptive arrays, constrained processors, least squares algorithms, multichannel filtering, orthogonal signals.

15. NUMBER OF PAGES
132

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev 2-89)
Prescribed by ANSI Std Z39-18
298-102

Table of Contents

List of Tables

List of Figures

I. INTRODUCTION

The research described in this report is concerned with the transient behavior of linearly constrained wideband adaptive array sensors. The intent of this analysis is to develop a computationally inexpensive constrained sensor which is capable of quick convergence in a dynamic signal environment. Various structures will be examined and their transient performance will be evaluated through simulation.

The use of multichannel space-time processors has a proven value in the detection and estimation of signals which are received at spatially separated sensors. The benefit of utilizing such an array of receiving sensor elements to improve signal reception has long been recognized in the fields of communications [1], radar [2], sonar [3,4] and seismology [5].

In a dynamic signal environment it is desirable to have the processor sense the presence of interference noise sources and automatically adapt itself in order to both suppress the interference and enhance the desired signal reception. The manner in which these dual functions are realized is through the use of an adaptive control system which updates the parameters of an array processor in order to minimize some performance index. This process is depicted in figure 1. The goal of the entire system is to produce an output which is the best estimate of a particular waveform of the composite observation data in some statistical sense. This must be accomplished with little or no a priori knowledge of the signal environment. The first adaptive array research can be traced to the late 1950's and early 1960's [6,7,8,9]. Much research has been done on adaptive sensors since these early

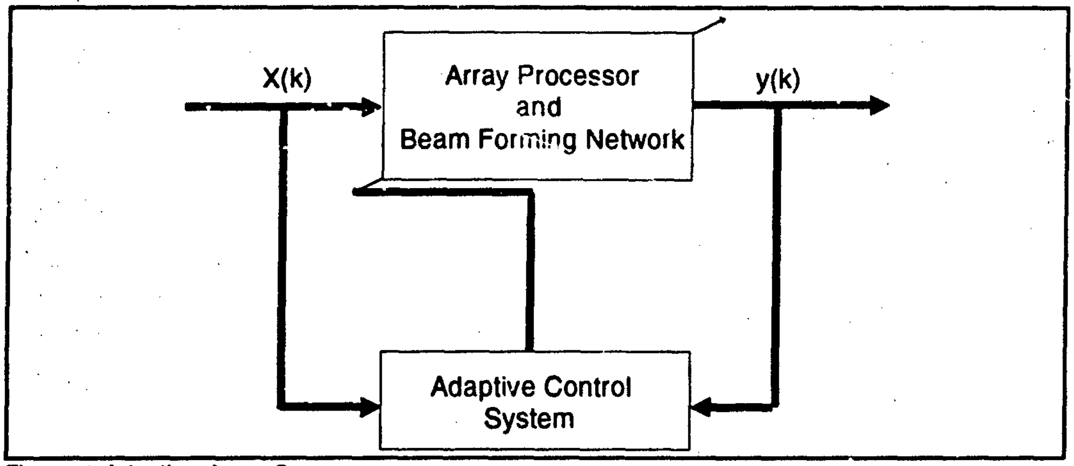

Figure 1 Adaptive Array Sensor

developments, and the use of adaptive arrays have been incorporated into many modern signal processing systems.

Thus, the problem at hand is characterized by a need to optimize the reception of one or more desired signals when multiple desired and undesired wideband directional sources impinge upon an array of passive receiving sensor elements. The traditional solution to this problem is the utilization of an adaptive tapped-delay-line filter following each element of the array. The rationale for this solution is based upon the stability properties of the finite impulse response class of filters coupled with the ability of the tapped-delay-line filter to process signals which encompass an appreciable bandwidth. The choice of adaptive algorithm for updating the coefficients of the adaptive filters in this study is restricted to the computationally modest stochastic gradient class.

The difficulty encountered with the traditional method of processing described above is that a dependency exists between the speed of convergence of the adaptive processor and the range of the eigenvalues of the observation data correlation matrix. The methodology which this research follows is to utilize different adaptive structures in an attempt to transform the observation data into a domain such that the resulting transformed correlation matrix exhibits a smaller eigenvalue spread or the aforementioned dependency is relieved.

A. Organization of the Report

This section will introduce the organization of the research to be presented. Section I.B reviews estimation theory and derives the discrete-time Wiener filter. In Section I.C, we extend the previous results to the multichannel wideband array case. Section I.D then derives and analyzes the classical least-mean square (LMS) algorithm.

One short coming of the early LMS adaptive array systems was the degradation of the desired signal while attempting to minimize interference in the receiving sensor sidelobes. Through the imposition of hard constraints on certain aspects of the processor we can guarantee some desirable responses regardless of the external environment.

The constraint of interest in this research is realized by defining the frequency response of the processor in the direction of the desired signal. Through the enforcement of this frequency response in the desired signal direction, one guarantees that the adaptation process can not cause its degradation.

The second chapter derives three linearly constrained wideband adaptive array sensors with tapped-delay-line structure. The first of these, termed the direct form, was originally conceived by Frost [11]. The second form is a partitioned realization which is shown to be identical to the direct form. This form, introduced by Griffiths [15], is derived solely to facilitate the development of the third form. The final form, termed the Generalized Sidelobe Canceller (GSC), was first presented by Applebaum and Chapman [12], and later extended by Griffiths [13]. The GSC form is then used extensively in this research.

The third chapter is concerned with replacing the GSC tapped-delay-line filter structure with orthogonal transform domain filter structures. The Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), lattice and Gram-Schmidt structures are examined, and a new frequency subband normalization algorithm is introduced. Simulation results are presented to compare the different structures.

The fourth chapter provides an overall comparison of the different structures considered, examines the results of the last chapter, presents the conclusions of this research and identifies areas for further research.

B. Estimation Theory and Wiener Filtering

This section considers discrete time estimation theory and derives the scalar form of the Wiener filter. The derivation in this section follows Gelb [32] and Widrow [20]. For a more thorough treatment and derivation in continuous time, one is referred to Van Trees [16]. In simple terms, estimation is concerned with the use of information derived from observations in order to make decisions about parameters of interest that are optimal in some sense. The adaptive sensor problem is to estimate a signal of interest s(k), which is observed only in the presence of additive noise n(k). That is to say, given a received observation signal sequence x(k) such that

$$x(k) = s(k) + n(k)\,,\quad k=1,2,... \tag{1-1}$$

we desire to process it in order to obtain an estimate $y(k) = \hat{s}(k)$. The observation data x(k) is a random variable whose statistics are formed from those of both the signal and the noise. The processing or filtering is shown in figure 2. In section A of this chapter we mentioned

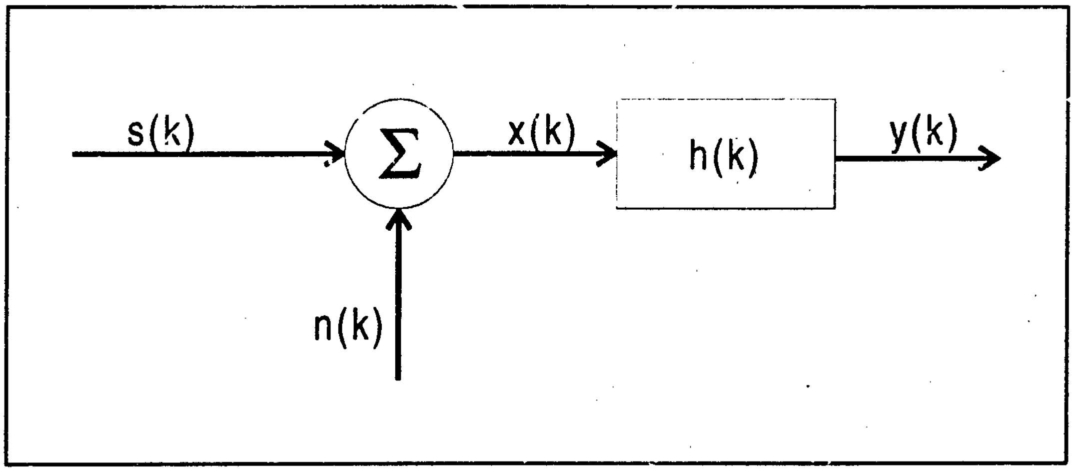

Figure 2 Filtering Operation

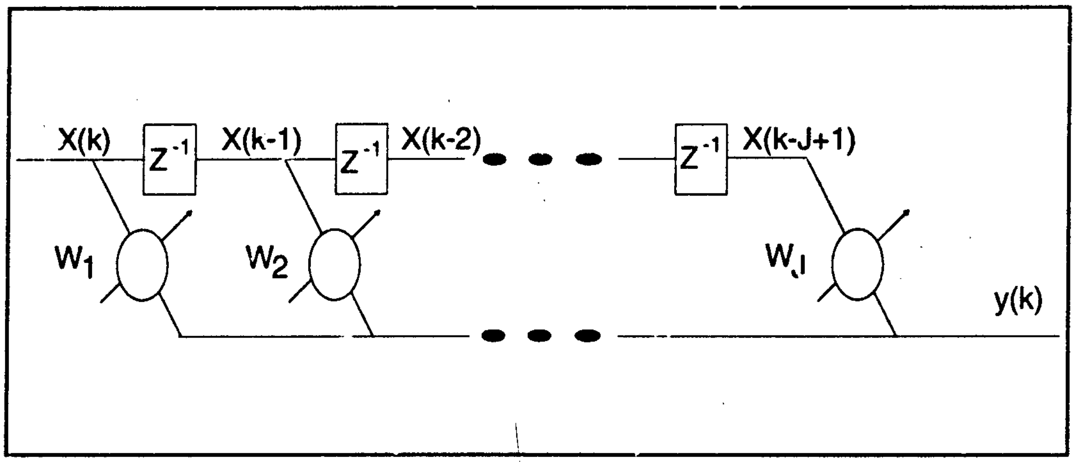

Figure 3 Tapped-Delay-Line FIR Filter

that we were interested in an array processor which had a finite impulse response. The FIR filter is represented by its impulse response denoted h(k) on the box in figure 2.

Consider the linear tapped-delay-line (TDL) filter shown in figure 3. This filter clearly has a finite duration impulse response characterized by the sequence

$$h(k) = \mathbf{W} \tag{1-2}$$

where W is a J-dimensional vector and the filter is therefore termed a FIR filter of order J. The notation Z^{-1} in the boxes of figure 2 represent a unit time delay. We assume that the filter is driven by the random process x(k) and that this process is wide-sense stationary; it is characterized by a mean value which is independent of time

$$E[x(k)] = \alpha \tag{1-3}$$

where α is a constant (assumed zero for simplicity) and a correlation function

$$E[x(m)x(n)] \equiv r_{xx}(m,n) = r_{xx}(m-n) = r_{xx}(\tau) \tag{1-4}$$

with τ being an integer value. The filter output may be expressed as the convolution sum

$$y(k) = \sum_{m=1}^{J} w_m x(k-m+1) \tag{1-5}$$

We now assume for the following derivation that there is some desired reference signal, d(k), available which represents the desired system output. Then the residual or error signal $\varepsilon(k)$ is defined as

$$\varepsilon(k) = d(k) - y(k) \tag{1-6}$$

We desire to find the optimal values of the filter coefficients W_i which minimize this error signal in some statistical sense. In Wiener filter theory [19] the performance function that is used to optimize the filter coefficients is the mean-square value of the error signal.

The mean-square value of equation (1-6) is represented as

$$E[\varepsilon^2(k)] = E[d^2(k)] - 2\,E[d(k)\,y(k)] + E[y^2(k)] \tag{1-7}$$

and through the substitution of equation (1-5) into equation (1-7) we may write

$$E[\varepsilon^2(k)] = E[d^{\,2}(k)] - 2\sum_{m=1}^{J} w_m\, E[d(k)\,x(k-m+1)] + \sum_{m=1}^{J}\sum_{n=1}^{J} w_m\, w_n\, E[x(k-m+1)\,x(k-n+1)] \tag{1-8}$$

We now assume that the medium through which the signal propagates to reach the sensor is linear and time-invariant. Then the desired signal component s(k) of the observation data x(k) is related through a linear time-invariant transformation to the desired reference signal d(k). Furthermore, we then assume that x(k) and d(k) are jointly stationary. This means that the expression in equation (1-8) is the sum of the mean-squared value of the desired response, a function of the cross-correlation of the desired response with the observation signal and a function of the correlation of the received observation signal. Thus, equation (1-8) may be written as

$$E[\varepsilon^2(k)] = E[d^{\,2}(k)] - 2\sum_{m=1}^{J} w_m\, r_{xd}(m-1) + \sum_{m=1}^{J}\ \sum_{n=1}^{J} w_m\, w_n\, r_{xx}(n-m) \tag{1-9}$$

where the cross-correlation and input correlation are, respectively

$$r_{xd}(m-1) \equiv E[d(k)\, x(k-m+1)] \tag{1-10}$$

$$r_{xx}(n-m) \equiv E[x(k-m+1)\, x(k-n+1)] \tag{1-11}$$

It is convenient to now change notation from scalar representation to matrix form, and define the observation vector and weight vector as

$$X(k) = \begin{bmatrix} x(k) \\ x(k-1) \\ \cdot \\ \cdot \\ \cdot \\ x(k-J+1) \end{bmatrix} \quad W(k) = \begin{bmatrix} w_1(k) \\ w_2(k) \\ \cdot \\ \cdot \\ \cdot \\ w_J(k) \end{bmatrix} \tag{1-12}$$

where we have explicitly shown the time-dependency of the filter coefficients. Suppressing the matrix time-varying notation and recognizing from equation (1-5) that the output may be expressed as $y = W^T X$, equation (1-9) is equivalent to

$$E[\varepsilon^2(k)] = E[d^{\,2}(k)] - 2\, R_{xd}^T W + W^T R_{xx} W \tag{1-13}$$

where the cross-correlation vector $\boldsymbol{R}_{xd}$ and the correlation matrix $\boldsymbol{R}_{xx}$ are formed in a manner analogous to the scalar case in equations (1-10) and (1-11):

$$\boldsymbol{R}_{xd} = \boldsymbol{E}\begin{bmatrix} d(k)x(k) \\ d(k)x(k-1) \\ \cdot \\ \cdot \\ \cdot \\ d(k)x(k-J+1) \end{bmatrix} = \begin{bmatrix} r_{xd}(0) \\ r_{xd}(1) \\ \cdot \\ \cdot \\ \cdot \\ r_{xd}(J-1) \end{bmatrix} \tag{1-14}$$

$$\boldsymbol{R}_{xx} = \boldsymbol{E}\begin{bmatrix} x^2(k) & x(k)x(k-1) & \cdot\ \cdot\ \cdot & x(k)x(k-J+1) \\ x(k-1)x(k) & x^2(k-1) & \cdot\ \cdot\ \cdot & x(k-1)x(k-J+1) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ x(k-J+1)x(k) & x(k-J+1)x(k-1) & \cdot\ \cdot\ \cdot & x^2(k-J+1) \end{bmatrix} \tag{1-15}$$

$$= \begin{bmatrix} r_{xx}(0) & r_{xx}(-1) & \cdot\ \cdot\ \cdot & r_{xx}(1-J) \\ r_{xx}(1) & r_{xx}(0) & \cdot\ \cdot\ \cdot & r_{xx}(2-J) \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ r_{xx}(J-1) & r_{xx}(J-2) & \cdot\ \cdot\ \cdot & r_{xx}(0) \end{bmatrix}$$

From equation (1-13) we can see that the mean-square error is quadratic in terms of the weight vector. This dependency results in a bowl-shaped performance surface. This surface must be concave upward since a negative mean-square error is not physically realizable. The objective is to design the filter such that we operate on the bottom of this performance surface. Since the surface is quadratic, we seek the sole global extreme value.

The gradient of the performance surface can be obtained by differentiating equation (1-13)

$$M_W = \frac{\partial E[\varepsilon^2]}{\partial W} = 2\boldsymbol{R}_{xx}\boldsymbol{W} - 2\boldsymbol{R}_{xd} \tag{1-16}$$

Setting this vector equal to zero to find the extremum yields

$$R_{xx}W = R_{xd} \tag{1-17}$$

and we find the optimal weight vector is

$$W_{opt} = R_{xx}^{-1}R_{xd} \tag{1-18}$$

which is the Wiener-Hopf equation in matrix form. Examining equation (1-17) in scalar form, the optimal weight values are given by the solution to the equation

$$\sum_{m=1}^{J} w_{opt_m} r_{xx}(m-n) = r_{xd}(n-1)\,, \quad n=1,2,\ldots,J \tag{1-19}$$

A tapped-delay-line filter whose impulse response is defined by equation (1-18) or (1-19) is said to be optimal in a mean-square sense. The filter output realized by W_{opt} is denoted y_{opt}, and is the best estimate (in a mean-square sense) of the desired response given the observation input. This may be expressed as

$$y_{opt}(k) = \hat{d}(k|k,k-1,\ldots,k-J+1) = \sum_{m=1}^{J} w_{opt_m} x(k-m+1) \tag{1-20}$$

Therefore, using (1-10) and (1-11), equation (1-19) can be written as

$$\sum_{m=1}^{J} w_{opt_m} E[x(k-m+1)x(k-n+1)] = E[d(k)x(k-n+1)], \quad n=1,2,\ldots J \tag{1-21}$$

This is equivalent to the statement

$$E[(d(k)-\sum_{m=1}^{J} w_{opt_m} x(k-m+1))x(k-n+1)] = 0\,, \quad n=1,2,\ldots J \tag{1-22}$$

and finally, using (1-20)

$$E[(d(k) - y_{opt}(k))x(k-n+1)] = E[\varepsilon_{opt}(k)x(k-n+1)] = 0\,, \quad n=1,2,\ldots J \tag{1-23}$$

where ε_{opt} is defined as

$$\varepsilon_{opt}(k) = d(k) - y_{opt}(k) \tag{1-24}$$

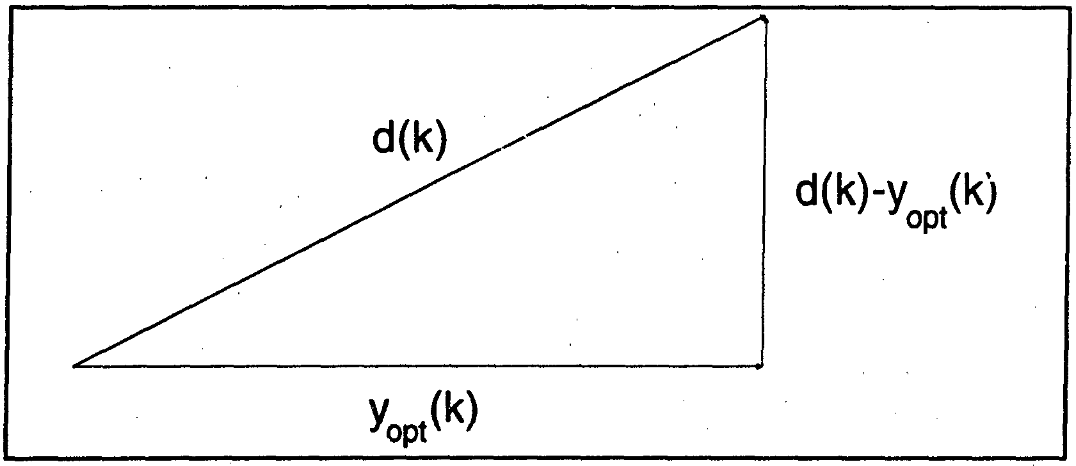

Figure 4 Orthogonality Condition

Two important results now appear from this derivation. First, equation (1-23) states that the error signal and any of the observation signals are orthogonal in the optimal filter. Second, the error signal ε_{opt} and the output y_{opt} are orthogonal since

$$E[\varepsilon_{opt}(k)y_{opt}(k)] = E[\varepsilon_{opt}(k)\sum_{m=1}^{J} w_{opt_m}x(k-m+1)] = \sum_{m=1}^{J} w_{opt_m} E[\varepsilon_{opt}(k)x(k-n+1)] = 0 \tag{1-25}$$

This condition, depicted in figure 4, ensures that the error signal vector is minimum.

C. The Multichannel Array Sensor

We now extend the results of the previous section to the multichannel case of interest. The wideband multichannel model is depicted in figure 5 for an array composed of K sensor elements and J taps per element. It was mentioned in section I.B that signals whose spectrum can not be adequately characterized by a single frequency must be processed by a filter which is capable of realizing a broadband frequency response. If the TDL filter tap spacing is sufficiently close and the number of taps is large, then the filter will approximate an ideal

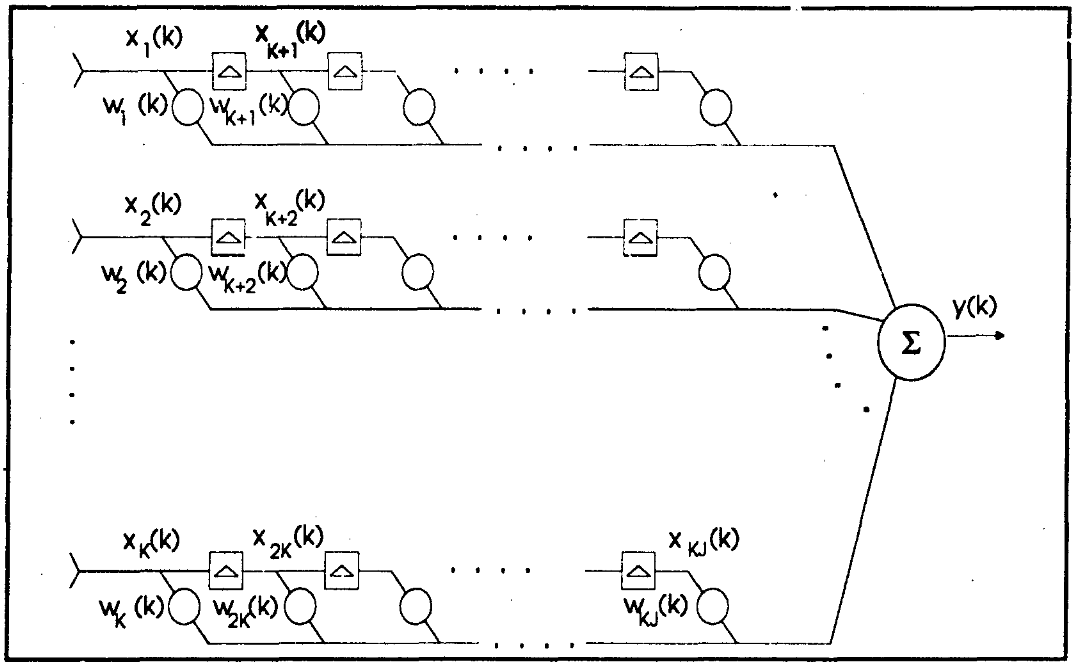

Figure 5 Wideband Multichannel Array

filter which exhibits exact control of gain and phase at each frequency of interest. The sampling theorem [49] may be used to define the filter bandwidth. Consider a continuous input signal which is sampled by one TDL filter. The sample sequence defined by the signals appearing at the TDL taps uniquely characterize the corresponding waveform from which it was generated provided that the continuous signal is bandlimited with its highest frequency component f_{max} less than or equal to one-half the sample frequency corresponding to the time delay Δ, or $f_{max} \leq 1/2\,\Delta$. The total bandwidth of a bandlimited signal is $2f_{max}$, so that a TDL can uniquely characterize any continuous signal having a bandwidth less than or equal to $1/\Delta$ Hz; the signal bandwidth of the TDL filter.

The received observation data for the array in figure 5 is the sum of the directional signals impinging upon the array and the thermal noise present on each element. The signals are assumed to have been produced by sources in the far field which propagate through the medium surrounding the sensor. We now define the KJ-dimensional vectors $X(k)$ and $W(k)$ as

$$X(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ x_{KJ}(k) \end{bmatrix} = \begin{bmatrix} X_t(k) \\ X_t(k-\Delta) \\ \vdots \\ X_t(k-(J-1)\Delta) \end{bmatrix} \qquad W(k) = \begin{bmatrix} w_1 \\ w_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ w_{KJ} \end{bmatrix} \tag{1-26}$$

where $X_t(k-l\Delta)$ for $l = 0,1,..,J-1$ is the K-dimensional vector of the observation signals present at the column of weights following the *l-th* delay. The signals are assumed to be plane waves, so that if we let $\boldsymbol{u}$ representing the directional signal unit vector, $\boldsymbol{r}$ be the sensor coordinate vector and v denote the propagation velocity, then the intersensor delay may be written as

$$\tau_i = \frac{\boldsymbol{u} \cdot \boldsymbol{r}}{v} \tag{1-27}$$

where the $(\cdot)$ operator is the standard vector inner product. Assuming that the reference element is the top sensor of figure 5, then we may express the K dimensional vector of directional signals $S_t(k)$ as

$$S_t(k) = \begin{bmatrix} s_1(k) \\ s_2(k) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ s_K(k) \end{bmatrix} = \begin{bmatrix} s_1(k) \\ s_1(k-\tau_i) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ s_1(k-(K-1)\,\tau_i) \end{bmatrix} \tag{1-28}$$

For the K-dimensional vector of array observation signals $X_t(k)$ the correlation matrix $\overleftrightarrow{R}_{xx}(\tau)$ is of dimension K x K and is given by

$$\overleftrightarrow{R}_{xx}(\tau) = E[X_t(k)X_t^T(k-\tau)] \tag{1-29}$$

For the KJ-dimensional vector of all observation signals $X(k)$ the correlation matrix is of dimension KJ x KJ, and is given by

$$R_{xx}(\tau) = E[X(k)X^T(k-\tau)] \tag{1-30}$$

which may be written as

$$R_{xx}(\tau) = \begin{bmatrix} \overleftrightarrow{R}_{xx}(\tau) & \overleftrightarrow{R}_{xx}(\tau+\Delta) & \cdot & \cdot & \overleftrightarrow{R}_{xx}(\tau+(J-1)\Delta) \\ \overleftrightarrow{R}_{xx}(\tau-\Delta) & \overleftrightarrow{R}_{xx}(\tau) & & & \cdot \\ \cdot & & \cdot & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & \cdot & \cdot \\ \cdot & & & & \cdot \\ \overleftrightarrow{R}_{xx}(\tau-(J-1)\Delta) & \cdot & \cdot & \cdot & \overleftrightarrow{R}_{xx}(\tau) \end{bmatrix} \tag{1-31}$$

We assume that the signal and noise components of the observation data are independent. Then

$$R_{xx}(\tau) = R_{ss}(\tau) + R_{nn}(\tau) \tag{1-32}$$

where $R_{SS}(\tau)$ and $R_{nn}(\tau)$ are the KJ x KJ dimensional correlation matrices of the received desired signal component and noise component, respectively, of the observation data:

$$R_{ss}(\tau) = E[S(k)S^T(k-\tau)] \tag{1-33}$$

$$R_{nn}(\tau) = E[N(k)N^T(k-\tau)] \tag{1-34}$$

At a zero time shift, it is well known [11,20,33] that for the case of interest R_{xx} and R_{nn} are positive definite matrices and R_{SS} is generally at least positive semi-definite. For the remainder of this research, any second moment not explicitly containing a time delay argument will be meant to denote the second moment at zero time delay.

The optimal weight vector in the minimum mean-square error sense for the wideband multichannel array of figure 5 is given by the Wiener-Hopf equation

$$W_{opt} = R_{xx}^{-1} R_{xd} \tag{1-35}$$

where R_{xx} is now of dimension KJ x KJ as shown in equation (1-31) and R_{xd} is a KJ x 1 vector formed in a manner analogous to equation (1-14), but using the vector $\mathbf{X}(k)$ given in equation (1-26). This is the same solution as that of Widrow [20, equation 2.17].

D. The LMS Algorithm

The requirement which exists for the array processor is to solve the Wiener-Hopf equation for the optimal weight vector. This solution requires knowledge of both R_{xx} and R_{xd}. In the problems of interest, the correlation matrix R_{xx} is unknown while, in general, the cross-correlation vector R_{xd} may not be available. One method of obtaining the solution would be the direct estimation of these values followed by their substitution into the Wiener-Hopf equation. Monzingo and Miller [34] describe the drawbacks of this approach. In summary: potentially serious computational problems arise in computing and inverting R_{xx}; the number of measurements and computations needed to accurately estimate the elements of R_{xx} and R_{xd} is large and requires repetition upon change of the input signal statistics; and the implementation of a direct solution requires highly accurate estimates and results in an open loop control.

Another method of solving the Wiener-Hopf equation is to solve for the optimal weight vector iteratively through a gradient search procedure. The LMS algorithm [23] is one of the family of gradient search techniques for descending towards the performance surface minimum. Not having a priori knowledge of W_{opt}, we begin at some arbitrary weight value $W(0)$ and estimate the gradient at this point. We choose the next weight value to be the value of the current weight plus an increment proportional to the negative slope estimated. This procedure leads to the iterative procedure

$$W(k+1) = W(k) + \mu(-\hat{M}_W(k)) \tag{1-36}$$

where $\hat{M}_W(k)$ is the estimate of the gradient at time k and μ is the step size of the incremental walk to the bottom of the performance bowl. It will soon be seen that μ also controls the stability and the rate of convergence of the algorithm.

The key to the LMS algorithm is that it views $\varepsilon^2(k)$ to be an estimate of the mean-square error $E[\varepsilon^2(k)]$. Thus, the gradient estimate is given by

$$\hat{M}_W(k) = \frac{\partial \varepsilon^2(k)}{\partial W(k)} = 2\varepsilon(k)\frac{\partial \varepsilon(k)}{\partial W(k)} = 2\varepsilon(k)\frac{\partial [d(k) - W^T(k)X(k)]}{\partial W(k)} = -2\varepsilon(k)X(k) \tag{1-37}$$

yielding the LMS algorithm

$$W(k+1) = W(k) + 2\,\mu\varepsilon(k)X(k) \tag{1-38}$$

The LMS algorithm is now demonstrated through a simple narrowband scalar example motivated by Widrow [20]. The purpose of this example is to provide a graphical understanding of the gradient search technique used in the LMS algorithm and to examine the effects of noise in the development. Consider a tapped-delay-line filter with one delay and two taps (order J=2). The input signal is a sinusoid given by $s(k) = \sin(\frac{2\pi k}{5})$ and the desired signal is $d(k) = 2\cos(\frac{2\pi k}{5})$. The observation data x(k) was then formed in both the case where n(k)=0 and for n(k) being a zero-mean gaussian random variable with power $P_n = 0.01$. The correlation matrix and the cross-correlation vector were computed from equations (1-14) and (1-15). The weight vector was found from (1-18), and the performance surface given by (1-13) was then plotted as shown in figure 6. The LMS algorithm was then executed through 500 iterations with a step size $\mu = 0.05$ and an initial condition of $W(0) = \mathbf{0}$. The contour plot in figures 7 and 8 depict the noise free and additive noise performance surface searches, respectively. Figures 9 and 10 show the weight vector transients. Figures 11 and 12 present plots of the estimated mean-square error, termed 'learning curves' by Widrow. Figures 13 and 14 show the estimation error, a measure of the convergence of the adaptive algorithm, for the no noise and noisy cases, respectively. The reduction of the area under the learning curve and the reduction of estimation error are the two key indicators of the dynamic response or transient behavior of the adaptive filter.

The convergence of the LMS algorithm to the optimal weight vector solution is now considered. The derivation in this section follows that of Widrow [20]. As previously mentioned, the LMS algorithm utilizes the square error as an estimate of the mean-square error. From equation (1-37) we see that this leads to an unbiased estimate when the weight vector is held constant.

$$E[\hat{M}_W(k)] = -2E[(d(k) - y(k))X(k)] = 2\left(R_{xx}W - R_{xd}\right) = M_W(k) \tag{1-39}$$

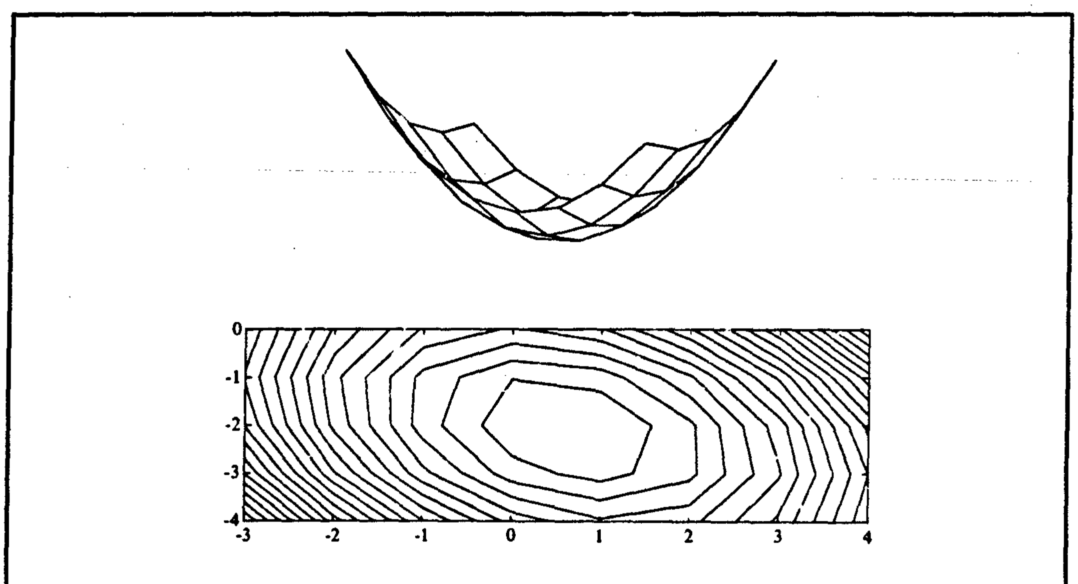

Figure 6 Performance Surface

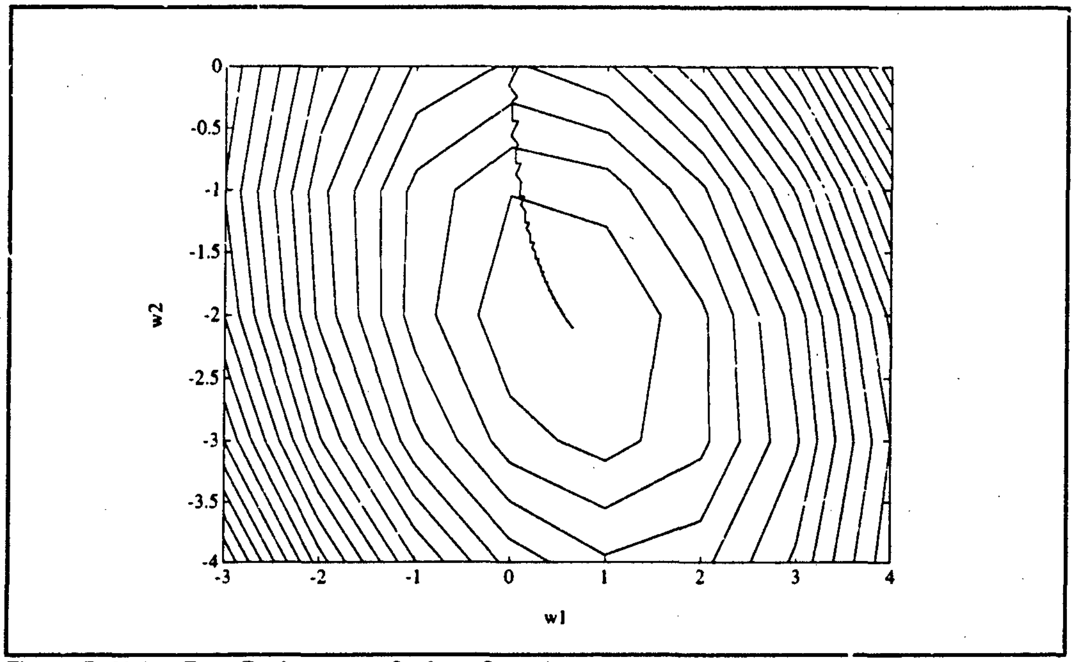

Figure 7 Noise Free Performance Surface Search

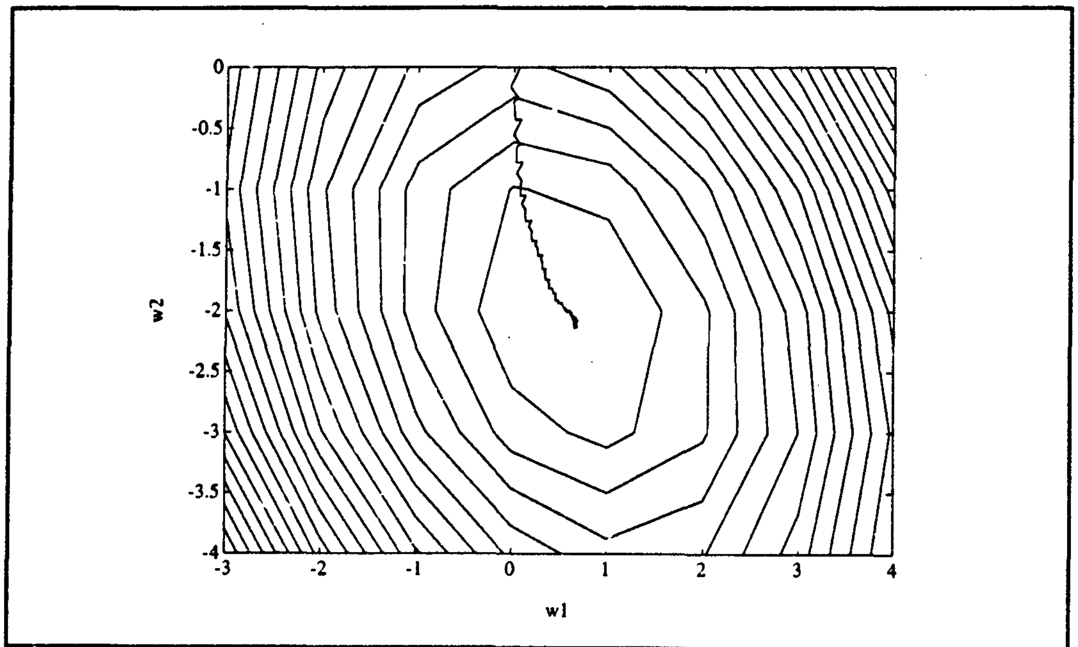

Figure 8 Noisy Performance Surface Search

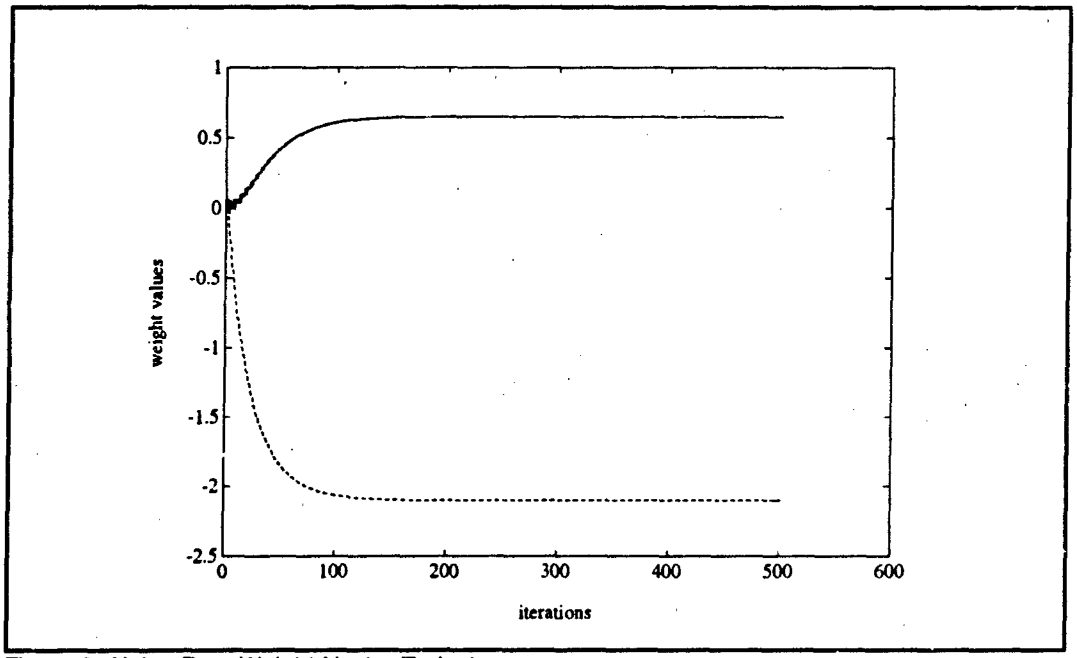

Figure 9 Noise Free Weight Vector Trajectory

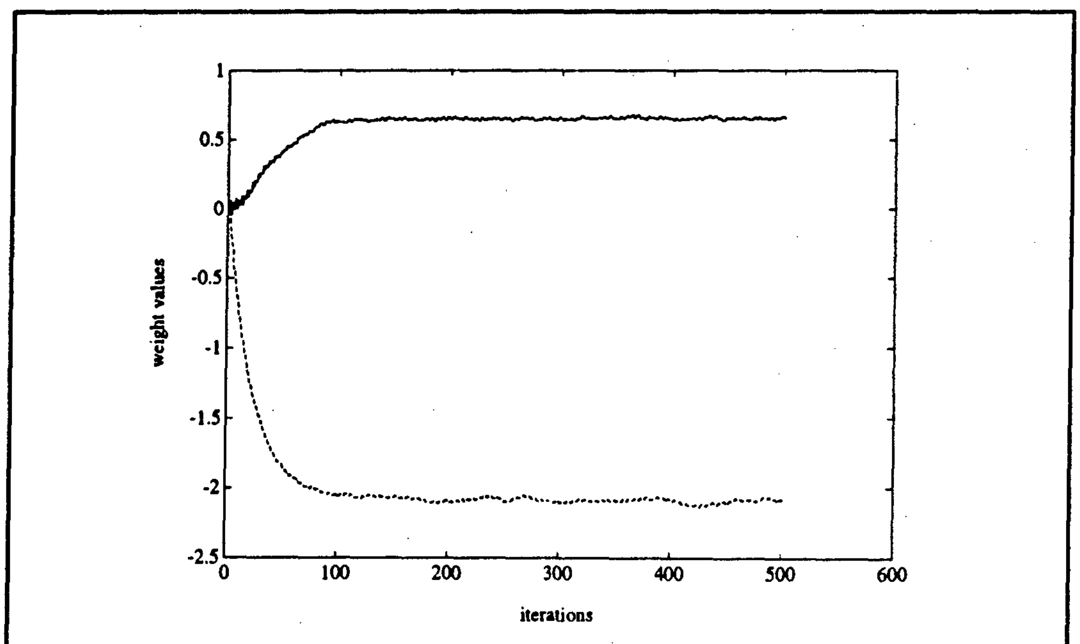

Figure 10 Noisy Weight Vector Trajectory

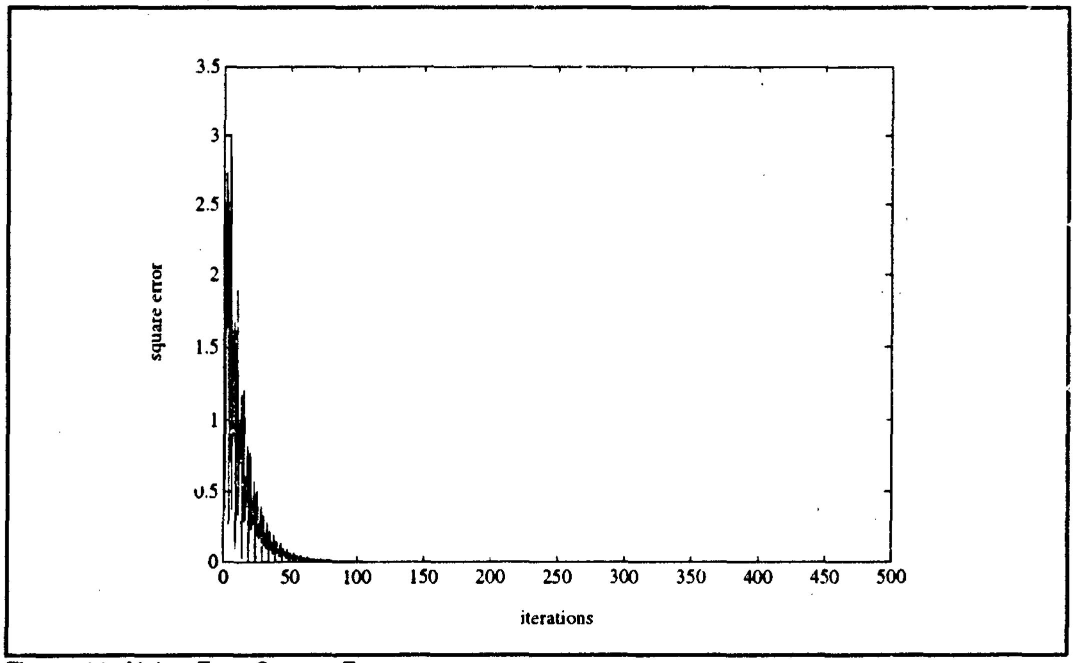

Figure 11 Noise Free Square Error

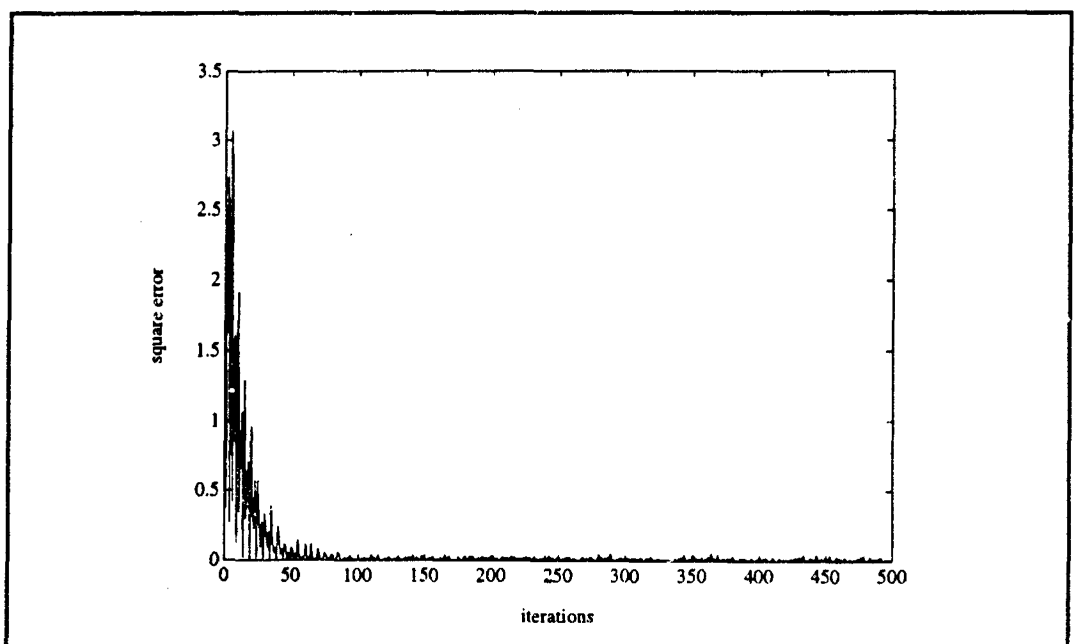

Figure 12 Noisy Square Error

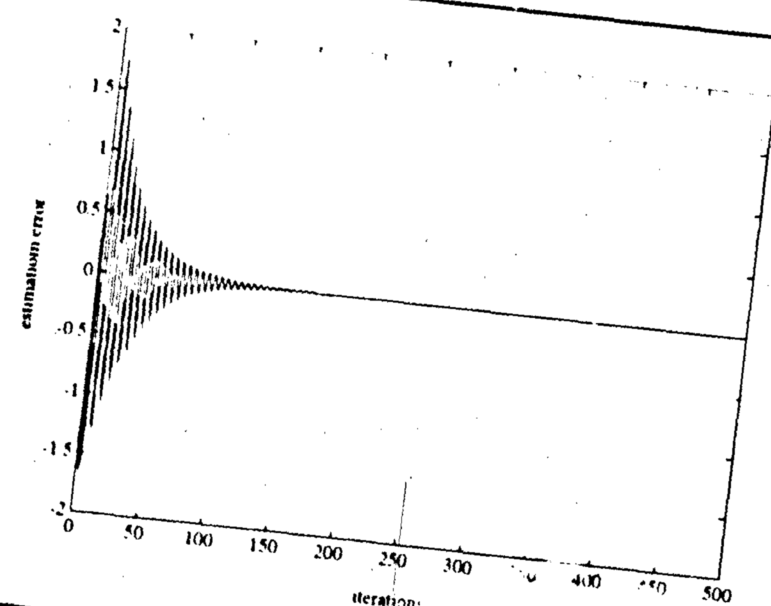

Figure 13 Noise Free Esimation Error

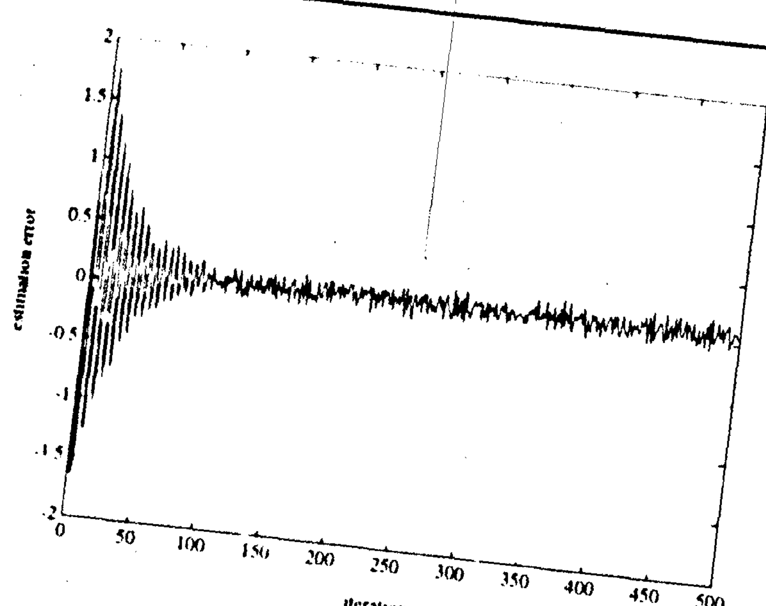

Figure 14 Noisy Estimation Error

Since the weight vector is not constant but changes with each iteration, we now examine the dynamic weight convergence. It is assumed in the following development that successive observation data vectors are independent, which allows the weight vector to be treated as though it was independent of the observation data process. It is noted that while this may be unnecessary, it will be used to simplify the following derivation. Then the expected value of equation (1-38) yields

$$\boldsymbol{E}[\boldsymbol{W}(k+1)] = \boldsymbol{E}[\boldsymbol{W}(k)] + 2\mu\left(\boldsymbol{R}_{xd} - \boldsymbol{R}_{xx}\boldsymbol{E}[\boldsymbol{W}(k)]\right) \tag{1-40}$$

Rearranging equation (1-18) or (1-35) to yield $\boldsymbol{R}_{xd} = \boldsymbol{R}_{xx}\boldsymbol{W}_{opt}$ we find

$$\boldsymbol{E}[\boldsymbol{W}(k+1)] = (\boldsymbol{I} - 2\mu\boldsymbol{R}_{xx})\boldsymbol{E}[\boldsymbol{W}(k)] + 2\mu\boldsymbol{R}_{xx}\boldsymbol{W}_{opt} \tag{1-41}$$

Proceeding with the derivation at hand, we define the weight error vector to be the translation vector $\boldsymbol{T}$

$$\boldsymbol{T}(k) = \boldsymbol{W}(k) - \boldsymbol{W}_{opt} \tag{1-42}$$

In order to diagonalize $(\boldsymbol{I} - 2\mu\boldsymbol{R}_{xx})$, the coefficient matrix of $\boldsymbol{E}[\boldsymbol{W}(k)]$ in (1-41), we define the unitary matrix $\boldsymbol{Q}$ which performs a rotation upon $\boldsymbol{T}$ (introducing the new vector $\boldsymbol{V}$) and the similarity operation upon the correlation matrix

$$\boldsymbol{T} = \boldsymbol{Q}\boldsymbol{V} \tag{1-43}$$

$$\boldsymbol{R}_{xx} = \boldsymbol{Q}\,\Lambda\,\boldsymbol{Q}^{T} \tag{1-44}$$

where Λ is a square diagonal matrix whose element on the *l-th* row is the *l-th* eigenvalue of the correlation matrix. The *l-th* column of $\boldsymbol{Q}$ is the corresponding eigenvector. Using the Wiener-Hopf equation and substituting (1-42) into (1-41), the weight vector error equation becomes

$$\boldsymbol{E}[\boldsymbol{T}(k+1)] = (\boldsymbol{I} - 2\mu\boldsymbol{R}_{xx})\boldsymbol{E}[\boldsymbol{T}(k)] \tag{1-45}$$

where the simplification falls out of the algebra upon the expansion. Utilizing our transformations defined in (1-43) and (1-44), we may express this as

$$E[V(k+1)] = \left(Q^{-1} I Q - 2\mu Q^{-1} R_{xx} Q\right) E[V(k)] = (I - 2\mu\Lambda) E[V(k)] \quad (1\text{-}46)$$

Finally, through the iteration of (1-46), we find the solution

$$E[V(k)] = (I - 2\mu\Lambda)^k V(0) \quad (1\text{-}47)$$

The weight convergence question is now answered by considering whether the error vector $W(k) - W_{opt}$ converges to zero. Mathematically, this is equivalent to

$$\lim_{k \to \infty} (I - 2\mu\Lambda)^k = 0 \quad (1\text{-}48)$$

or, since both matrices inside the parenthesis are diagonal,

$$\lim_{k \to \infty} (1 - 2\mu\lambda_i)^k = 0 \quad (1\text{-}49)$$

where λ_i denotes the *i-th* eigenvalue of the observation correlation matrix. Thus, for convergence, the step size μ must be chosen such that

$$0 < \mu < \frac{1}{\lambda_{max}} \quad (1\text{-}50)$$

where λ_{max} is the largest eigenvalue of R_{xx}. The translation vector $V(k)$ thus obeys a trajectory which is the sum of n modes, where the correlation matrix is of dimension n x n, and the *i-th* mode is proportional to $(1 - 2\mu\lambda_i)^k$. The speed of convergence is governed by μ. If the step size is too large for equation (1-50) to be satisfied, then one or more modes of the translation vector will be larger than unity in magnitude and the error will increase in time. For a fixed step size, the speed of convergence is dominated by the slowest mode. The eigenvalue spread of the correlation matrix (the ratio of largest to smallest eigenvalues or the condition number of the matrix) is therefore an indicator of the convergence speed of the LMS algorithm. The larger that the eigenvalue spread of the observation correlation matrix is, the slower the convergence of the algorithm.

The above results can be used to better understand figures 7 through 14. Comparing figures 7 and 8, it is apparent that the noisy gradient estimate does not immediately approach the minimum mean-square error solution, but walks around the bottom of the bowl. This causes the weight jitter seen in comparing the noisy weight vector transients of figure 9 to those of figure 10. From equation (1-49), we see that the learning curves in figures 11 and 12 should decay according to geometric ratios of the form $(1 - 2\mu\lambda_i)$, yielding a time constant for the *i-th* mode of

$$\tau_i = \frac{1}{4\mu\lambda_i} \tag{1-51}$$

where we have used the convention of Widrow that the time constant of the mean-square error learning curve is one-half that of the geometric decay [14]. The noisy estimate of the mean-square error causes the jitter shown in figure 12. The estimation error in figures 13 and 14 depict the amount of time it takes for the adaptive filter to learn the amplitude difference and phase shift between the desired signal and the observed signal and, in figure 14, the error is at first sinusoidal and then becomes increasingly random.

The requirements of the LMS algorithm are evident through the examination of equations (1-39) and (1-40) and the fact that the LMS approximation is accomplished by estimating the unknown average values with the available present values; in essence dropping the expectation operator. Specifically, a desired signal is required and the correlation matrices are approximated by

$$\hat{R}_{xx}(k) = X(k)X^T(k) \tag{1-52}$$

$$\hat{R}_{xd}(k) = X(k)d(k) \tag{1-53}$$

The desired signal requirement has been addressed by Widrow, who derived training schemes to provide this signal [17], and Compton, who demonstrated that the algorithm can be successfully applied to communications when the desired information carrying signal is unknown but some of the characteristics of it are available [18].

Griffiths modified the LMS algcrithm through recognizing that a priori knowledge of the desired signal correlation function, its direction of arrival and the array geometry allow the received cross correlation vector to be defined [33]. This relieves the algorithm of the desired signal requirement and allows the vector $\boldsymbol{R}_{xd}$ to be formed off-line. The only real-time estimate that must be formed in Griffiths' LMS scheme is the correlation matrix $\boldsymbol{R}_{xx}$.

The principal algorithm of interest in this research is a linearly constrained LMS algorithm. This algorithm, presented in the next chapter, will be seen to utilize a directional constraint in order to completely relieve any requirement of a desired signal or its statistical characterization. In fact, the desired signal d(k) is taken to be identically zero. The correlation matrix $\boldsymbol{R}_{xx}$, as with Griffiths' algorithm, is the only required real-time estimate and the necessary a priori information is simply the direction of arrival of the desired signal.

II. CONSTRAINED PROCESSORS WITH TAPPED-DELAY-LINE STRUCTURE

The first constrained adaptive array to be derived is termed the "direct form". A second form, which is a partitioned constrained processor, will then be developed from the direct form. This processor and the direct form will be shown to have identical performance. A third form, referred to as the "Generalized Sidelobe Canceller" or GSC will be shown to be a partitioned processor which utilizes a methodology to separate the constraint from the adaptive beamformer in a manner which results in an unconstrained adaptation.

A. Direct Form Processor

The derivation in this section follows that of Frost [11]. The geometrical interpretation and portions of the algorithm development have been expanded. A direct form constrained processor with K sensors and J taps per sensor is displayed in figure 15. We assume that the array has been electronically pre-steered so as to be parallel to the desired signal's wavefront through the use of the shown steering time delays. This is referred to as a signal aligned array.

The received signal X(t) is a composite of the desired signal S(t) and noise N(t). The noise itself may be a composite of passive and active noises. An example of passive noises would be the thermal noise present on each element, while an example of active noises might be hostile jammers. The signals are sampled and processed so that the received signal at the *k-th* sample may be written as:

$$X(k) = S(k) + N(k) \tag{2-1}$$

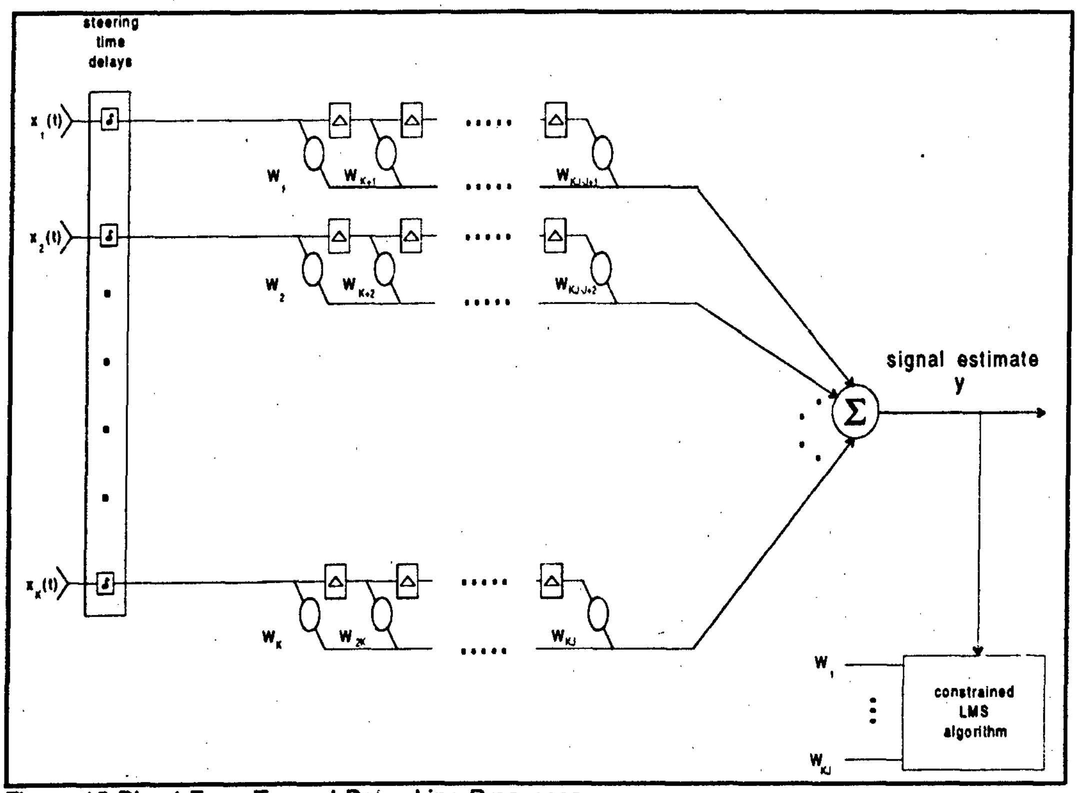

Figure 15 Direct Form Tapped-Delay-Line Processor

The signals in this derivation are assumed to be realizations of zero mean stochastic processes with unknown second order statistics. The notation that will be used to describe the signal correlation matrices is:

$$\begin{aligned} R_{xx} &\equiv E[X(k)X^T(k)] \\ R_{nn} &\equiv E[N(k)N^T(k)] \\ R_{ss} &\equiv E[S(k)S^T(k)] \end{aligned} \tag{2-2}$$

The desired signal S(k) is assumed to be uncorrelated with the noise N(k).

The signal **X**(k) impinges on the sensor array and arrives at each element at a different time determined by the array spacing and the composite signals direction of arrival. Since the array is assumed signal aligned, the look direction waveforms S(k) are the same down each column of the array.

The KJ dimensional stacked vectors of look direction waveforms $S(k)$, noises $N(k)$ and weights $W_{DF}(k)$ may be written as:

$$
S(k)=\begin{bmatrix} s(k) \\ s(k) \\ \cdot \\ \cdot \\ s(k) \\ s(k-\Delta) \\ s(k-\Delta) \\ \cdot \\ \cdot \\ s(k-\Delta) \\ \cdot \\ \cdot \\ \cdot \\ s(k-(J-1)\Delta) \\ s(k-(J-1)\Delta) \\ \cdot \\ \cdot \\ s(k-(J-1)\Delta) \end{bmatrix},\quad N(k)=\begin{bmatrix} n_1(k) \\ n_2(k) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ n_{KJ}(k) \end{bmatrix},\quad W_{DF}(k)=\begin{bmatrix} w_{DF_1}(k) \\ w_{DF_2}(k) \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ w_{DF_{KJ}}(k) \end{bmatrix} \tag{2-3}
$$

where Δ is the delay time introduced between successive taps in the array. The subscript DF will be suppressed for the remainder of section A, and will only be used in subsequent sections to clarify the array form being referenced as needed.

1) <u>The Adaptive Algorithm:</u> The algorithm for adapting the direct form weights must be capable of maintaining a chosen frequency response in the look direction while minimizing the output power in other directions; power which is due to undesirable noises. As previously shown, the desired signal produces identical components on each column of taps while the noise arriving from other directions will not in general produce equal

components at the tap inputs. The received composite signal at each tap is multiplied by a corresponding weight and summed to produced the array output. It is therefore evident that under these conditions the multichannel processor appears as a single channel tapped-delay-line with respect to the desired signal. This equivalent single channel filter has weights which are equal to the sum of the weights in the corresponding vertical column of the multichannel processor. This is shown in figure 16 for a three tap three channel array. Through constraining the sums of the weights in each of the J vertical columns to have some value f_j, we have in fact fixed the frequency response of the processor in the look direction. The cost of demanding this response is the loss of J degrees of freedom. Thus, only KJ-J degrees of freedom in choosing the weight values may be used to minimize the total output power from the array processor. Minimization of the total array output power subject to our

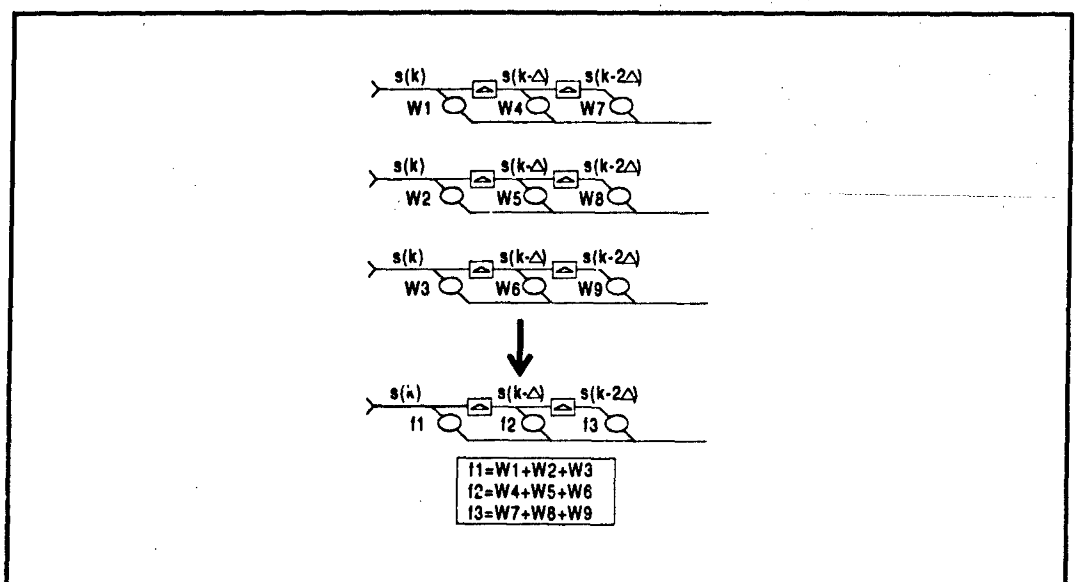

Figure 16 Equivalent Look Direction Processor

constraint is equivalent to minimizing all non-look direction noises as long as the desired signal is uncorrelated with the noise.

In terms of the signal and weight vectors, the look direction constraint may be expressed as

$$C^T W = F \tag{2-4}$$

where the KJxJ matrix **C** and the J vector **F**

$$C = \begin{bmatrix} \mathbf{1}_k & & & \\ & \mathbf{1}_k & \mathbf{0} & \\ & & \cdot & \\ & \mathbf{0} & \cdot & \\ & & & \mathbf{1}_k \end{bmatrix} \qquad F = \begin{bmatrix} f_i \\ \cdot \\ \cdot \\ \cdot \\ f_J \end{bmatrix} \tag{2-5}$$

effectively define the sums of the weights on each column of taps.

The output of the processor at the *k-th* sample is

$$y(k) = W^T(k)\ X(k) \tag{2-6}$$

and the expected value of the array output power is

$$E[y^2(k)] = W^T R_{xx} W \tag{2-7}$$

The problem at hand is to find the optimum weight vector **W** that will minimize the array output power (a scalar performance index)

$$L = \frac{1}{2} W^T R_{xx} W \tag{2-8}$$

subject to the constraint

$$g = C^T W - F = 0 \tag{2-9}$$

The constrained optimization problem can be reduced to an unconstrained problem through the use of Lagrange multipliers [10]. The Lagrangian for this system is

$$H = L + \lambda^T g = \frac{1}{2} W^T R_{xx} W + \lambda^T (C^T W - F) \tag{2-10}$$

where $\lambda \in R^J$ is an undetermined vector Lagrange multiplier The necessary conditions for optimality are

$$H_W = \frac{\partial H}{\partial W} = 0 \quad \text{and} \quad C^T W = F \tag{2-11}$$

Taking the gradient of the Lagrangian

$$H_W = R_{xx} W + C \lambda = 0 \tag{2-12}$$

and solving for the optimal weight vector yields

$$W_{opt} = -R_{xx}^{-1} C \lambda \tag{2-13}$$

Since W_{opt} must satisfy the constraint

$$-C^T R_{xx}^{-1} C \lambda = F \tag{2-14}$$

we find that the optimal value of the Lagrange multiplier is

$$\lambda = -[C^T R_{xx}^{-1} C]^{-1} F \tag{2-15}$$

and the optimal weight vector may be written as

$$W_{opt} = R_{xx}^{-1} C \, [C^T R_{xx}^{-1} C]^{-1} F \tag{2-16}$$

The constrained least squares estimate of the look direction signal is given by

$$y_{opt}(k) = W_{opt}^T X(k) \tag{2-17}$$

If we now assume for the time being that R_{xx} is known, we may derive the deterministic constrained least mean square (CLMS) algorithm. The initial weight vector is to be initialized on the constraint plane and subsequently moved in the direction of the negative gradient at each iteration. The adaptive step size used to walk down the performance surface is proportional to the magnitude of the gradient. The weight state and costate equations are given by

$$W(k+1) = W(k) - \mu H_W = W(k) - \mu [R_{xx} W(k) + C \lambda(k)] \tag{2-18}$$

$$\lambda(k) = \frac{-1}{\mu}(C^T C)^{-1}(F - C^T W(k)) - (C^T C)^{-1} C^T R_{xx} W(k) \tag{2-19}$$

The term $F - C^T W(k)$ in the costate equation permits the algorithm to make corrections for small deviations on the constraint, preventing large errors due to the accumulation of slight trajectory flaws. Substituting the weight costate equation into the weight state equation yields the CLMS algorithm

$$W(k+1) = P\,[\,W(k) - \mu R_{xx} W(k)\,] + Q \tag{2-20}$$

where

$$P = I_{KJ} - C\,(C^T C)^{-1} C^T \tag{2-21}$$

and

$$Q = C\,(C^T C)^{-1} F \tag{2-22}$$

This deterministic solution requires the a priori knowledge of the signal correlation matrix R_{xx}. The stochastic CLMS algorithm is obtained by estimating the correlation matrix at each iteration. A readily available estimate is formed at the *k-th* iteration by

$$\hat{R}_{xx} = X(k)\,X^T(k) \tag{2-23}$$

where we note that other estimates of the correlation matrix may have been used in place of equation (2-23). This form of the estimate, which is made up of the outer product of the available tap voltage vector, is the simplest, and is consistent with the LMS algorithm. Thus, from equations (2-20) and (2-23), we find that the weight update equation may now be written

$$W(k+1) = P\,[\,W(k) - \mu\, y(k)\, X(k)\,] + Q \tag{2-24}$$

For later comparison, a signal flow block diagram is presented in figure 17. Note that the pre-steering filters for the signal aligned array are not shown in block diagrams. For

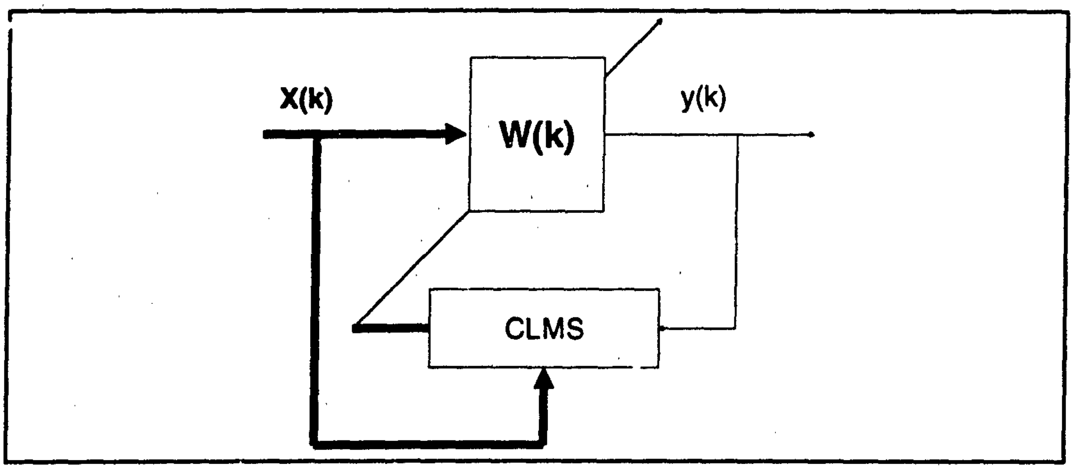

Figure 17 Direct Form Block Diagram

the remainder of this research we will be solely concerned with linearly constrained Minimum Variance Distortionless Response (MVDR) adaptive array sensors. In terms of the direct form processor, this may be realized by enforcing the condition that the vector $\boldsymbol{F}$ be composed of one reference element equal to unity and the remaining elements equal to zero. All of the results presented, however, will be general in nature and not reliant upon this emphasis unless explicitly stated.

2) Geometrical Interpretation: The geometrical analysis of the CLMS algorithm requires a number of definitions and propositions. These are presented in appendix I. The CLMS algorithm will now be shown to have a very simple representation. In addition, the relationship between the CLMS algorithm and the standard LMS algorithm will become evident.

We now consider the geometrical interpretation of the CLMS algorithm. Consider the diagram shown in figure 18. The subspace Σ is that subspace which satisfies the homogeneous form of the constraint equation. Thus, Σ is the nullspace of the matrix $\boldsymbol{C}$ defined in equation (2-5). This subspace will be termed the homogeneous constraint

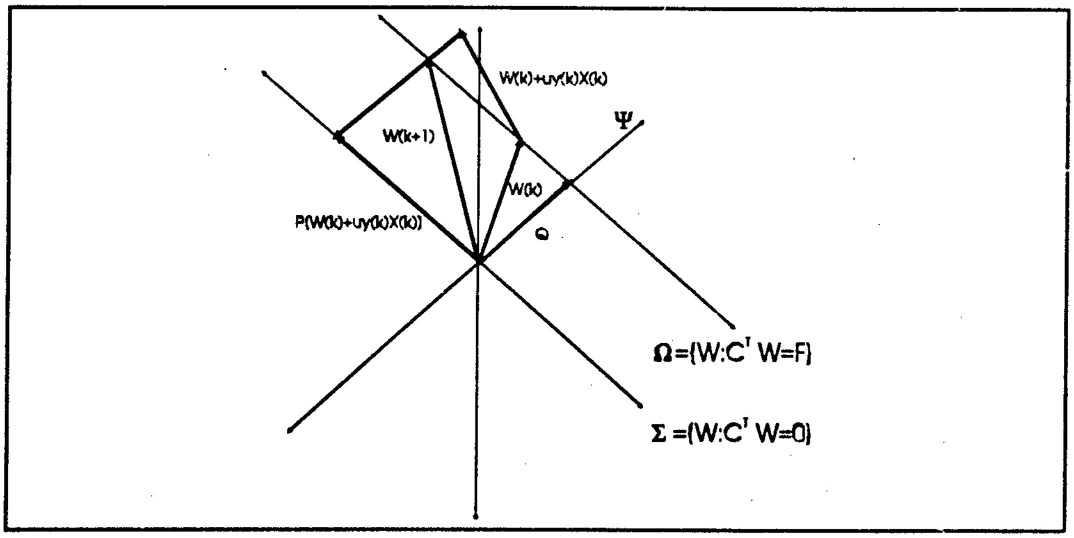

Figure 18 Geometrical View of CLMS Algorithm

subspace. The constraint equation, equation (2-4), ensures that the vector W_{opt} terminates on the (KJ-J) dimensional constraint surface **Ω**. The shortest weight vector from the origin to the constraint surface **Ω** is found by minimizing the norm of the weight vector $|W|^2$ subject to $C^T W = F$. The method of Lagrange multipliers is utilized, and the Lagrangian is

$$\Psi = \frac{1}{2} W^T W + \lambda^T (C^T W - F) \qquad (2\text{-}25)$$

The necessary conditions for optimality are

$$Hw = \frac{\partial H}{\partial W} = 0 \quad \text{and} \quad C^T W = F \qquad (2\text{-}26)$$

Taking the gradient of the Lagrangian and setting it equal to zero yields

$$W_{opt} = -C\,\lambda \qquad (2\text{-}27)$$

Since W_{opt} must satisfy the constraint, we find $-C^T C\,\lambda = F$ and the weight vector costate equation is

$$\lambda = -(C^T C)^{-1} F \qquad (2\text{-}28)$$

Substituting the weight costate equation into the equation for W_{opt} yields the vector which we seek:

$$W_{opt} = C\,(C^T C)^{-1} F \tag{2-29}$$

This vector is identical to the vector Q in the CLMS algorithm, equation (2-22).

It can easily be verified that Q is orthogonal to any vector z in Σ by examining the inner product

$$Q^T z = F^T (C^T C)^{-1} C^T z \tag{2-30}$$

and noting that $C^T z = 0$ by definition.

Consider the projection operation (defined in definition A1-6 and characterized by propositions A1-2 and A1-3 of appendix I). A vector W may be decomposed as the sum of one vector in Σ and one vector from the orthogonal space spanned by the J linearly independent columns of the constraint matrix C, termed the constraint subspace Ψ. The projection operator of interest acts as an identity operator on components within Σ and as an annihilator operator on components in Ψ.

The matrix P in equation (2-21) can be seen to be a projection operator onto Σ by noting that

$$C^T (PW) = C^T [I - C (C^T C)^{-1} C] W = 0 \tag{2-31}$$

and therefore $PW \in \Sigma$. We also note that

$$(I - P) W = W - PW = C (C^T C)^{-1} C^T W = Q \in \Psi \tag{2-32}$$

We now rewrite the CLMS algorithm (20) for convenience:

$$W (k + 1) = P \, [W (k) - \mu\, y (k) X (k)] + Q \tag{2-33}$$

The term inside the bracket is the standard LMS algorithm [23] with the desired signal d(k)=0, and the expression $y(k)X(k)$ is an estimate of the unconstrained negative gradient at time k. The CLMS algorithm thus computes the LMS estimate and projects the resulting vector onto the subspace Σ. The next weight vector $\boldsymbol{W}(k+1)$ is then formed by translating the projected vector onto the constraint surface Ω through the vector addition with $\boldsymbol{Q}$. This is depicted in figure 18.

3) Convergence of the CLMS algorithm: The weight vector $\boldsymbol{W}(k)$ in the CLMS algorithm is a function of $\boldsymbol{W}(0)$ and the sequence $\{\boldsymbol{X}(k)\}$. Throughout this development we have assumed that the observation vectors $\boldsymbol{X}(k)$ are independent. We note that Frost [11] has pointed out that this may be unnecessary and Daniell [21] has shown ε-convergence based on only asymptotic independence. Utilizing this assumption, $\boldsymbol{X}(k)$ is independent of $\boldsymbol{W}(k)$ and the expected value of the CLMS algorithm may be written as

$$E[\boldsymbol{W}(k+1)] = \boldsymbol{P}\left(E[\boldsymbol{W}(k)] - \mu R_{xx} E[\boldsymbol{W}(k)] \right) + \boldsymbol{Q} \tag{2-34}$$

From proposition A1-2 of appendix I, we may express $\boldsymbol{Q}$ as

$$\boldsymbol{Q} = (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{W}_{opt} \tag{2-35}$$

Let $\varphi(k+1)$ denote the difference between the mean adaptive weight vector at time k+1 and the optimal weight vector:

$$\boldsymbol{\varphi}(k+1) = E[\boldsymbol{W}(k+1)] - \boldsymbol{W}_{opt} \tag{2-36}$$

Proposition II-1: Let $\boldsymbol{W}_{\exp}$ and $\boldsymbol{W}_{opt}$ be elements of Ω with the difference vector $\boldsymbol{\varphi} = \boldsymbol{W}_{\exp} - \boldsymbol{W}_{opt}$. Then $\boldsymbol{\varphi} \in \Sigma$ and $\boldsymbol{P\varphi} = \boldsymbol{\varphi}$.

Proof of Proposition II-1: Since $\boldsymbol{W}_{\exp}$, $\boldsymbol{W}_{opt} \in \Omega$, then

$$\boldsymbol{C}^T\boldsymbol{\varphi} = \boldsymbol{C}^T\boldsymbol{W}_{\exp} - \boldsymbol{C}^T\boldsymbol{W}_{opt} = \boldsymbol{F} - \boldsymbol{F} = 0 \tag{2-37}$$

and $\boldsymbol{\varphi} \in \Sigma$. By proposition A1-2 and definition A1-6 of appendix I, if $\boldsymbol{\varphi} \in \Sigma$ then $\boldsymbol{P\varphi} = \boldsymbol{\varphi}$. This may be shown algebraically:

$$P\,\varphi = [\,I - C\,(\,C^TC\,)^{-1}C^T\,]\;\varphi = \varphi - 0 = \varphi.$$

With the results from proposition II-1, we may rewrite the iterative difference process as

$$\varphi\,(k+1) = P\,(E[\,W\,(k)] - uR_{xx}\,E[\,W\,(k)]\,) + [\,I - P\,]W_{opt} - W_{opt} \tag{2-38}$$

$$= P\,\varphi(k) - uP\,R_{xx}\,\varphi(k) \;\; = [I - u\,P\,R_{xx}\,P\,]\varphi(k) \;\; = [I - u\,P\,R_{xx}\,P\,]^{k+1}\varphi(0) \tag{2-39}$$

where we have used the fact that, from equations (2-16) and (2-21), $PR_{xx}W_{opt} = 0$.

The matrix $P\,R_{xx}P$ is the correlation matrix of the projected observations. It is the non-zero eigenvalues of this matrix which determine both the convergence rate of the CLMS algorithm and the performance of its steady-state with respect to the optimum. This projected correlation matrix $P\,R_{xx}P \in R^{n\,x\,n}$ and is symmetric. Hence, it is diagonalizable into n orthogonal eigenvectors.

Proposition II-2: m of the n eigenvectors of the matrix $P\,R_{xx}P$ are outside of the subspace Σ and have zero eigenvalues. The remaining n-m eigenvectors have non-zero eigenvalues and lie within Σ.

Proof of Proposition II-2: The matrices $P = I - C\,(\,C^TC\,)^{-1}C^T$ and C have full rank. Therefore, C has m columns of linearly independent vectors. It is evident that the product $C^TP\,R_{xx}\,P = 0$ since $P\left(R_{xx}\,P\,\right) \in \Sigma$. Thus, m columns of C are eigenvectors of $P\,R_{xx}P$ with zero eigenvalues.

The columns of C are orthogonal to Σ. Therefore, the remaining (n-m) eigenvectors must be in Σ. From proposition II-2, if $\varphi \in \Sigma$ then $P\,\varphi = \varphi$. Thus, if v_i is an eigenvector of $P\,R_{xx}P$ in Σ, then

$$v_i^TP\,R_{xx}\,P\,v_i = v_i^TR_{xx}\,v_i > 0 \tag{2-40}$$

furthermore, if σ_i is an eigenvalue corresponding to $\boldsymbol{v}_i \in \Sigma$, then $\boldsymbol{P}\boldsymbol{R}_{xx}\boldsymbol{P}\boldsymbol{v}_i = \sigma_i \boldsymbol{v}_i$ and

$$\boldsymbol{v}_i\boldsymbol{P}\boldsymbol{R}_{xx}\boldsymbol{P}\boldsymbol{v}_i = \sigma_i \boldsymbol{v}_i^T\boldsymbol{v}_i = \sigma_i \tag{2-41}$$

We now consider the relationship between the eigenvalues of the constrained system and those of the unconstrained correlation matrix. We continue the notation used to denote the (n-m) eigenvalues of $\boldsymbol{P}\,\boldsymbol{R}_{xx}\boldsymbol{P}$ as σ_i and will denote the n non-zero eigenvalues of $\boldsymbol{R}_{xx}$ as λ_i. The well known result, generally referred to as Rayleigh's Theorem [22] states:

$$\lambda_{min} \le \sigma_{min} \le \sigma_i \le \sigma_{max} \le \lambda_{max} \tag{2-42}$$

The difference vector $\varphi(0)$, defined in the previous section, lies entirely within Σ and may therefore be considered as a linear combination of the eigenvectors $\boldsymbol{v}_i$ which correspond to the (n-m) non-trivial eigenvalues of $\boldsymbol{P}\,\boldsymbol{R}_{xx}\boldsymbol{P}$. Thus,

$$\varphi(k+1) = [\,\boldsymbol{I} - \mu\boldsymbol{P}\boldsymbol{R}_{xx}\boldsymbol{P}\,]^{k+1}\,\boldsymbol{v}_i = [\,1 - \mu\,\sigma_i\,]^{k+1}\boldsymbol{v}_i \tag{2-43}$$

The convergence along any eigenvector $\boldsymbol{v}_i$ is then geometric with ratio $[\,1 - \mu\sigma_i]$ and associated time constant

$$\tau_i = \frac{-1}{\ln(1-\mu\sigma_i)} \tag{2-44}$$

if $\mu\sigma_i << 1$, then $\dfrac{-1}{\ln(1-\mu\sigma_i)} \rightarrow \dfrac{1}{\mu\sigma_i}$, and it becomes evident that if μ is chosen so that

$$0 \le \mu \le \frac{1}{\sigma_{max}} \tag{2-45}$$

then the norm of the difference vector is bounded between two monotonically decreasing geometric progressions

$$[\,1 - \mu\,\sigma_{max}\,]^{k+1}\|\,\varphi(0)\| \le \|\,\varphi(k+1)\| \le [\,1 - \mu\,\sigma_{min}]^{k+1}\|\,\varphi(0)\| \tag{2-46}$$

Therefore, if the initial difference is finite, then $E[W]$ converges to the optimum with the time constants for the geometric ratio given in equation (2-44) above. That is to say, the weight vector converges in the sense that

$$\lim_{k\to\infty} | E[W(k) - W_{opt}] | = 0 \tag{2-47}$$

The operation of the CLMS algorithm in a quasi-stationary environment is now considered. The algorithm step size μ is assumed to remain constant during this development. The weight vector adaptation results in a non-zero variance about its optimal value. This adds an additional cost, termed 'misadjustment" by Widrow [14]. This dimensionless quantity, denoted $M(\mu)$, is a measure of how closely the algorithm approaches the optimal performance.

$$M(\mu) = \lim_{k\to 0} \frac{E\left[\left(W^T(k)X(k)\right)^2\right] - E\left[\left(W_{opt}^T X(k)\right)^2\right]}{E\left[\left(W_{opt}^T X(k)\right)^2\right]} \tag{2-48}$$

Frost [11] has shown that the steady-state misadjustment may be bounded by

$$\frac{\mu}{2}\left(\frac{trace\,(PR_{xx}P)}{1 - \frac{\mu}{2}[\,trace\,(PR_{xx}P) + 2\sigma_{\min}]}\right) \le M(\mu) \le \frac{\mu}{2}\left(\frac{trace\,(PR_{xx}P)}{1 - \frac{\mu}{2}[\,trace\,(PR_{xx}P) + 2\sigma_{\max}]}\right) \tag{2-49}$$

Thus, $M(\mu)$ can be made arbitrarily close to zero by choosing a small step size. The fundamental trade off is cost performance at the expense of increased convergence time.

B. Partitioned Form Constrained Processor

The algorithm for the direct form TDL structure clearly shows two discernible events taking place. The first event is the adaptive process which determines the LMS solution for the adaptivity. The second event is the enforcement of the constraint each iteration, which is both deterministic and decided upon before implementation. This

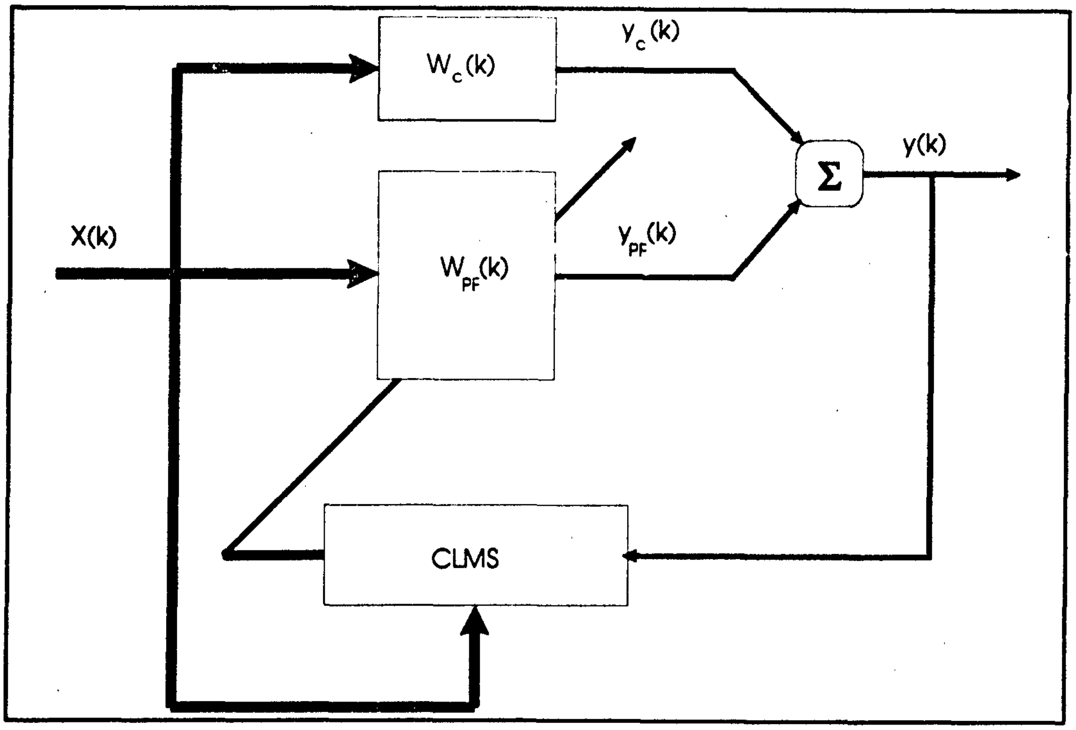

Figure 19 Partitioned CLMS Processor

algorithm partitioning was also depicted in the preceding geometrical interpretation and figure 18.

Noting this separation in the algorithm, it would seem reasonable that one could change the form of the processor in order to partition the two aforementioned events. This is the approach that was taken by Griffiths [15] and now explained. The adaptive section of the processor would operate in the subspace Σ and the non-adaptive section would perform the translation to the subspace Ω, satisfying the constraint. Thus, this structure separates the conventional beamformer from the adaptive processor. As shown in figure 19, the partitioned processor described above is implemented so that

$$C^T W_c = F \tag{2-50}$$

and

$$C^T W_{PF} = 0 \tag{2-51}$$

where W_{PF} represents the adaptive weights in a manner analogous to section II.A.1 and W_C is a conventional non-adaptive beamforming filter. Equation (2-51) ensures that the desired signal is eliminated from the adaptive processor input. Equations (2-50) and (2-51) can be seen to satisfy equation (2-4) by considering the global partitioned weight vector

$$W_g = \begin{bmatrix} W_c \\ \hline W_{PF} \end{bmatrix} \tag{2-52}$$

and noting that

$$C^T W_g = \begin{bmatrix} F \\ \hline 0 \end{bmatrix} \tag{2-53}$$

The partitioned processor shown in figure 19 thus utilizes the CLMS algorithm:

$$\min_{W_{PF}} \; W_{PF}^T R_{xx} W_{PF} \tag{2-54}$$

subject to the constraint

$$C^T W_{PF} = 0 \tag{2-55}$$

The optimal adaptive weight vector $W_{PF_{opt}}$ for the partitioned processor is now derived. The output of the processor is

$$y = \left(W_c^T - W_{PF}^T\right) X \tag{2-56}$$

and the Lagrangian for this system is

$$H = \frac{1}{2}\left(W_c^T - W_{PF}^T\right) R_{xx} \left(W_c^T - W_{PF}^T\right) + \lambda^T \left(C^T W_{PF} \right) \tag{2-57}$$

The necessary conditions for optimality are

$$H_{W_{PF}} = \frac{\partial H}{\partial W_{PF}} = 0 \quad \text{and} \quad C^T W_{PF} = 0 \tag{2-58}$$

Taking the gradient of the Lagrangian

$$H_{W_{PF}} = -R_{xx}\left(W_c - W_{PF}\right) + C\,\lambda = 0 \tag{2-59}$$

and solving for the optimal weight vector yields

$$W_{PF_{opt}} = W_c - R_{xx}^{-1} C\,\lambda \tag{2-60}$$

Since W_{opt} must satisfy the constraint

$$C^T\left(W_c - R_{xx}^{-1}\,C\,\lambda\right) = 0 \tag{2-61}$$

we find that the optimal value of the Lagrange multiplier is

$$\lambda = [\,C^T R_{xx}^{-1}\,C\,]^{-1}\,C^T W_c \tag{2-62}$$

and the optimal weight vector may be written as

$$W_{PF_{opt}} = W_c - R_{xx}^{-1}C\,[\,C^T R_{xx}^{-1} C\,]^{-1} C^T W_c \tag{2-63}$$

which can be seen to be equivalent to equation (2-16) by considering $\left(W_c - W_{PF_{opt}}\right)$. It is apparent that the array transient response is unchanged from that of the direct form due to the partitioned form's dependence upon the eigenvalues of the same matrix $\boldsymbol{PR_{xx}P}$. Since the partitioned processor has the same transient behavior and the optimal weight vector for the processor is equivalent to the direct form, the two processors have identical performance.

C. Generalized Sidelobe Canceler (GSC)

The GSC form was first introduced by Applebaum and Chapman [12] for narrowband signals in a radar context. The derivation in this section will follow the extensions made by Griffiths [13] for wideband signals. The iterative equation (2-24) may be partitioned into a scalar nonadaptive equation and a matrix adaptive equation. To derive this particular form, we will begin with the direct form processor CLMS algorithm, and partition it in a manner which will utilize a matrix filter to ensure that the adaptive processor operates on the homogeneous constraint subspace Σ. This means that the processor does not require the constraint $C^T W = 0$ for the adaptive section. In other words, the constrained processor operates with an unconstrained adaptive algorithm.

We note that equation (2-24) can be written for the *l-th* column as

$$W(k+1) = W(k) + \mu y(k)\left[\frac{1}{K}X^T(k-l)\mathbf{1}\mathbf{1} - X(k-l)\right] - \frac{1}{K}\left[W^T(k)\mathbf{1}\mathbf{1} - f_l\mathbf{1}\right] \tag{2-64}$$

We proceed by defining a K-dimensional constraint vector W_C for the MVDR processor such that

$$W_c = \frac{1}{K}\mathbf{1} \tag{2-65}$$

and a (K-1) x J dimensional signal blocking matrix W_S composed of linearly independent rows $r(i)$ such that

$$r^T(i)\mathbf{1} = 0 \qquad i = 1 \ldots K-1 \tag{2-66}$$

Then we introduce the invertible transformation matrix T such that

$$T = \begin{bmatrix} \hat{W}_c^T \\ -\hat{W}_s \end{bmatrix} \tag{2-67}$$

where the KJ-J x KJ matrix $\hat{W}_S$ and the KJ x 1 vector $\hat{W}_C$ are given by

$$\hat{W}_s = \begin{bmatrix} W_s & 0 & 0 & \ldots & 0 \\ 0 & W_s & 0 & \ldots & 0 \\ 0 & 0 & W_s & \ldots & 0 \\ \cdot & \cdot & \cdot & \ldots & \cdot \\ \cdot & \cdot & \cdot & \ldots & \cdot \\ \cdot & \cdot & \cdot & \ldots & \cdot \\ 0 & 0 & 0 & \ldots & W_s \end{bmatrix} \qquad \hat{W}_c = \begin{bmatrix} W_c \\ 0 \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \tag{2-68}$$

and finally, from (2-65) and (2-66) we note

$$T\,\mathbf{1} = \begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \tag{2-69}$$

Multiplying the stochastic CLMS algorithm by the transformation matrix $\boldsymbol{T}$, using equation (2-69), and defining

$$a(k) = \hat{W}_c^T W_{DF}(k) \tag{2-70}$$

$$\boldsymbol{B}(k) = -\hat{W}_s W_{DF}(k) \tag{2-71}$$

we find that equation (2-64) can be expressed as

$$\left[\begin{array}{c} a(k+1) \\ \hline \boldsymbol{B}(k+1) \end{array}\right] = \left[\begin{array}{c} a(k) \\ \hline \boldsymbol{B}(k) \end{array}\right] + \mu y(k) \left[\left(\begin{array}{c} \frac{1}{K} X^T 1 \\ \hline 0 \end{array} \right) - \left(\begin{array}{c} W_c^T X \\ \hline -W_s X \end{array} \right) \right] - \frac{1}{K} \left[\begin{array}{c} W_{DF}^T(k)\,1 - f_l \\ \hline 0 \end{array} \right] \tag{2-72}$$

Substituting equations (2-65) and (2-4) into equation (2-72) yields the nonadaptive scalar equation

$$a(k+1) = a(k) \tag{2-73}$$

and the adaptive matrix equation

$$\boldsymbol{B}(k+1) = \boldsymbol{B}(k) + \mu y(k) \boldsymbol{X}_s(k) \tag{2-74}$$

Equation (2-74) is the standard unconstrained LMS algorithm and $\boldsymbol{X}_S$ is defined by

$$\hat{\boldsymbol{X}}_s(k) = \hat{W}_s \boldsymbol{X}(k) \quad or \quad \boldsymbol{X}_s(k) = \boldsymbol{W}_s \boldsymbol{X}_l(k - l\Delta) \tag{2-75}$$

where $\boldsymbol{X}_l(k-l\Delta)$ was defined in equation (1-26).

As long as the nonadaptive scalar weights $a(k)$ satisfy

$$a_l(k) = f_l \tag{2-76}$$

then equations (2-73) and (2-74) define the stochastic CLMS algorithm with the constraint explicitly separated from the adaptive beamformer. This is the Generalized Sidelobe Canceller (GSC) form of the CLMS array, and is shown in figure 20, where for the directional constraint of interest $\bar{K} = K-1$.

The LMS algorithm in equation (2-74) is a direct function of the CLMS algorithm weight vector in equations (2-24) and (2-64). In other words, equation (2-74) presents a

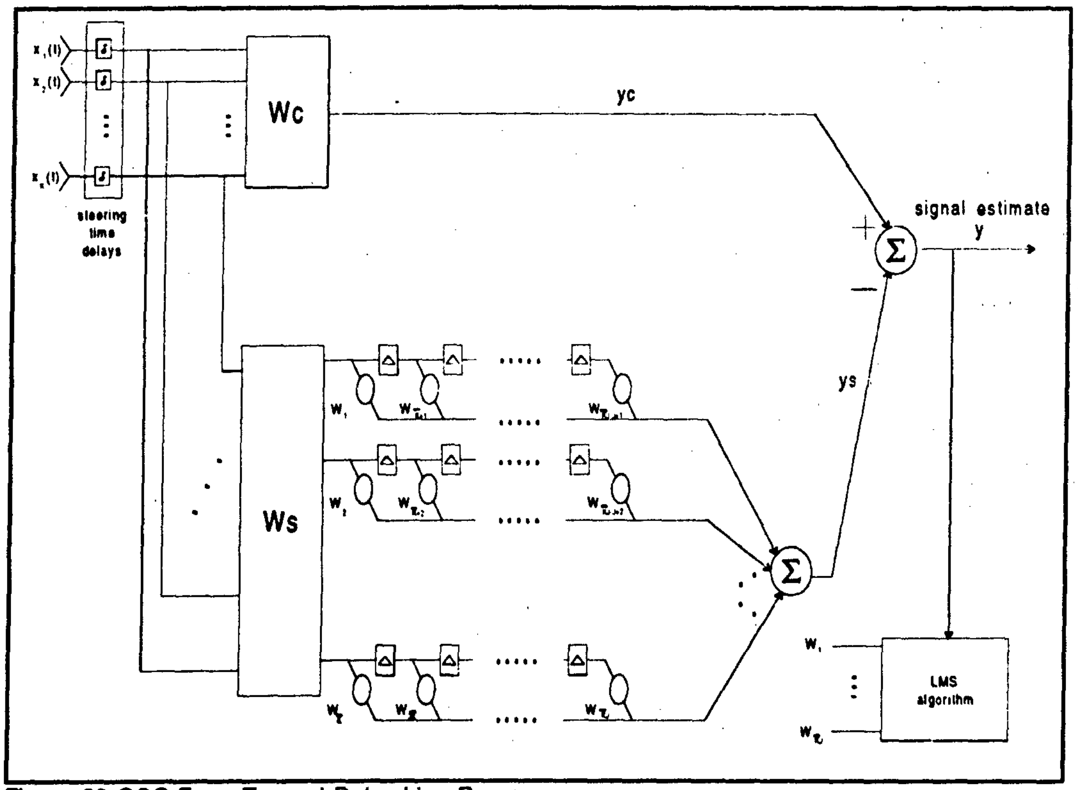

Figure 20 GSC Form Tapped-Delay-Line Processor

method of representing the CLMS algorithm with respect to the GSC form. In general, the GSC algorithm weights are not a function of the CLMS algorithm, but are updated by equation (2-74) such that the W_i in figure 20 are not constrained

$$W(k) \neq W_{DF}(k) \tag{2-77}$$

and the nonadaptive weights shown satisfy equation (2-76).

The question of interest is the behavior of the GSC form as a function of the signal blocking matrix W_s and the unconstrained weight vector $\hat{W}_{GSC}(k) = W(k)$. It is shown in appendix II that the optimal weight vector $\hat{W}_{GSC_{opt}}$ is equivalent to the other two forms considered when equations (2-65) and (2-66) are satisfied, ie:

$$\hat{W}_{GSC_{opt}} = \left(\hat{W}_s R_{xx} \hat{W}_s^T\right)^{-1} \hat{W}_s R_{xx} \hat{W}_c = W_{PF_{opt}} = W_{DF_{opt}} \tag{2-78}$$

This leads to the conclusion that as long as the signal blocking matrix forces the adaptive processor to function in the subspace Σ (blocks the look-direction signal) and the nonadaptive weights enforce the constraint, then the steady-state solution is equivalent across all three processor forms. It should be noted that this equivalent steady-state response is achieved through not only an unconstrained adaptation, but also with a reduced order adaptive processor. For the directional constraint of interest, the processor need only operate on the transformed observation vector $\hat{X}_S$ of dimension (KJ-J) x 1. The transient behavior of the GSC form, however, may or may not be identical.

Thus, the GSC can provide filtering operations which are either identical or different to those of the CLMS array depending on the structure of the signal blocking matrix $\boldsymbol{W_S}$ and the constraint vector $\boldsymbol{W_C}$. If the linearly independent rows of $\boldsymbol{W_S}$ satisfy equation (2-66), are orthogonal

$$\boldsymbol{r}^T(i)\,\boldsymbol{r}(j)=0, \qquad i \neq j \tag{2-79}$$

and $\boldsymbol{W_C}$ satisfies equation (2-65), then the stochastic CLMS processor is obtained, as detailed in appendix II and explained from a geometrical point of view below. Furthermore, if $\boldsymbol{W_S}$ satisfies equation (2-66) but not equation (2-79) and/or $\boldsymbol{W_C}$ does not satisfy equation (2-65), then a processor is formed which will have the same steady-state performance, but different transient trajectories.

The fact that the GSC form reduces to the CLMS form if equation (2-79) is enforced is evident by considering the geometrical interpretation of section II.A.2. With a fixed $\boldsymbol{W_C}$, it is clear that the orthogonality condition for the rows of $\boldsymbol{W_S}$ is equivalent to decomposing the GSC such that

$$\hat{\boldsymbol{W}}_c = \boldsymbol{Q} \tag{2-80}$$

$$\hat{W}_s \hat{W}_{GSC}(k) = -P[\hat{W}_{GSC}(k) + \mu y(k) X(k)] \quad (2\text{-}81)$$

where **P** and **Q** are defined in equations (2-21) and (2-22), and the geometrical relationship was shown in figure 18. The vector $\hat{W}_C = Q$ can not be changed in order to keep the desired MVDR processor. Since Q spans Ψ, in our search for quicker transient behavior we are restricted to examine relationships which are non-orthogonal combinations of Q with $\hat{W}_S \in \Sigma$. The weight vector at each iteration must still be an element of Ω.

As an example, assume a linear array with K=J=4 and W_C satisfying equation (2-65). Then if we choose a W_{S_1} such that it is composed of mutually orthogonal row vectors which are binary Walsh functions[24]

$$W_{s_1} = \begin{bmatrix} 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \quad (2\text{-}82)$$

then we obtain a processor identical to the CLMS direct form. By contrast, if we choose a non-orthogonal W_{S_2} such that it forms the sum and difference of the adjacent channels

$$W_{s_2} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \quad (2\text{-}83)$$

then we obtain the processor described by Applebaum and Chapman [12].

The capability of finding a W_S which implements a rectangular quadratic transformation $R_{XS} = \hat{W}_S R_{XX} \hat{W}_S^T$ such that the resulting matrix has a smaller eigenvalue spread than the original correlation matrix $PR_{XX}P$ is intriguing. This capability means that the GSC adaptive processor may converge faster than the CLMS form. The quickest convergence possible would be realized if the rectangular quadratic transformation equalized the eigenvalues of the input correlation matrix. However, due to the form of $\hat{W}_S$, given in equation (2-68), it is not in general possible to determine a W_S which satisfies the requirement of equation (2-66) and equalizes the eigenvalues of the resulting matrix $\boldsymbol{R}_{XS}$. The best that can be achieved is to find a signal blocking matrix $\boldsymbol{W}_S$ which causes the smallest eigenvalue spread of $\boldsymbol{R}_{XS}$. The dilemma is that even this requires unavailable a priori information about the original correlation matrix itself. Thus, while one may guarantee identical transient behavior for the GSC and the direct form processor, there is no deterministic method available to ensure the capability of a better dynamic response.

The utility of the GSC form, however, lies in the fact that the matrix filter partitioning permits the use of other adaptive structures to replace the tapped-delay-line unconstrained processor and reduces both the dimensionality of the adaptive weight vector and the algorithmic complexity. It is precisely these capabilities which the remainder of this research will concentrate on, with the goal of reducing the adaptive processor convergence time.

D. Normalization of the Step Size Gain

The CLMS and the LMS algorithms used in the direct form and GSC form arrays, respectively, utilize a constant step size gain denoted μ. This gain is dependent upon the input signal power as described in section II.A.3, where the CLMS misadjustment was derived. This dependence is undesirable seeing as μ remains constant while the signal power changes over time. This undesirable dependency may also be seen by considering an increase Δs in the input signal. For the LMS algorithm, this is equivalent to increasing the gain from μ to $\mu(\Delta s)^2$; which is the same increase observed in the eigenvalues of the signal correlation matrix. If the signal power grows too large the adaptation algorithm may become unstable.

One method of solving the problem described above is to implement a time varying step size gain. The use of time-varying step sizes for the LMS algorithm has been considered by many authors for the narrowband single channel TDL adaptive filter [50,51,52,53]. The development in this section follows the development in Honig and Messerschmitt [53] with the utilization of different parameters and extensions to the multichannel wideband case of interest. The step-size should be normalized to the input signal power at each iteration. Thus, it is desirable to change the CLMS and LMS algorithms in equations (2-24) and (2-74) to the form

$$\boldsymbol{W}_{CLMS}(k+1) = \boldsymbol{P}\left[\boldsymbol{W}_{CLMS}(k) - \mu(k)\, y(k)\, \boldsymbol{X}(k)\right] + \boldsymbol{Q} \tag{2-84}$$

$$\boldsymbol{W}_{LMS}(k+1) = \boldsymbol{W}_{LMS}(k) + \mu(k)\, y(k)\, \boldsymbol{X}_l(k) \tag{2-85}$$

where $\mu(k)$ is the aforementioned time-varying step size. The magnitude of the step size each iteration will be approximately the same on average if at each sample it is normalized by an estimate of the input signal power. For the l-th TDL, this may be expressed as

$$\mu_l(k) = \frac{\alpha}{\hat{\sigma}_l^2(k)} \tag{2-86}$$

where $\hat{\sigma}_l^2(k)$ is a measure of the power content of the received signal at the l-th channel during the k-th iteration and α is a scalar.

One choice for forming the power estimate is through the use of a single pole low-pass filter. This is equivalent to using an exponentially weighted time average of the input signal power above each tap of the channel, and may be written as

$$\hat{\sigma}_l^2(k) = \beta\, \hat{\sigma}_l^2(k-1) + \frac{(1-\beta)}{J} X_l^T X_l \tag{2-87}$$

for the l-th channel. The selection of β such that $0<\beta<1$ controls the bandwidth of the filter and the resulting power averaging time. Let $\alpha = 1$, then the selection of β such that $\beta = 1$

yields

$$\lim_{k \to \infty} E\left[\frac{1}{\mu(k)}\right] = \frac{E\,[X^T(k)\,X(k)]}{1-\beta} \tag{2-88}$$

We now consider setting $\alpha = 1 - \beta$. This is equivalent to letting

$$E\,[\mu(k)] \approx \frac{1}{E\,[\sigma_x^2(k)]} \tag{2-89}$$

which is reasonable as long as $(1-\beta)$ is small enough to smooth out the statistical fluctuations in the step size so that $\mu(k)$ becomes virtually independent of the observation data $X(k)$. If the step size at k=0 is initialized at $\mu(0) = \dfrac{(1-\beta)}{X^T(0)\,X(0)}$, then the expected value of the step size will approximate the expected value of $\dfrac{(1-\beta)}{\hat{\sigma}_x^2(k)}$ for all k. The time constant in equation (2-44) will become

$$\tau_i \approx \frac{E\,[X^T(k)X(k)]}{(1-\beta)\,\lambda_i} = \frac{\sum_{i=1}^{N} \lambda_i}{N\,(1-\beta)\,\lambda_i} \tag{2-90}$$

which shows the proportionality to the ratio $\dfrac{\lambda_{avg}}{\lambda_i}$. Therefore, a change in the input signal variance causes a much less dramatic change in the convergence speed and the misadjustment is reduced.

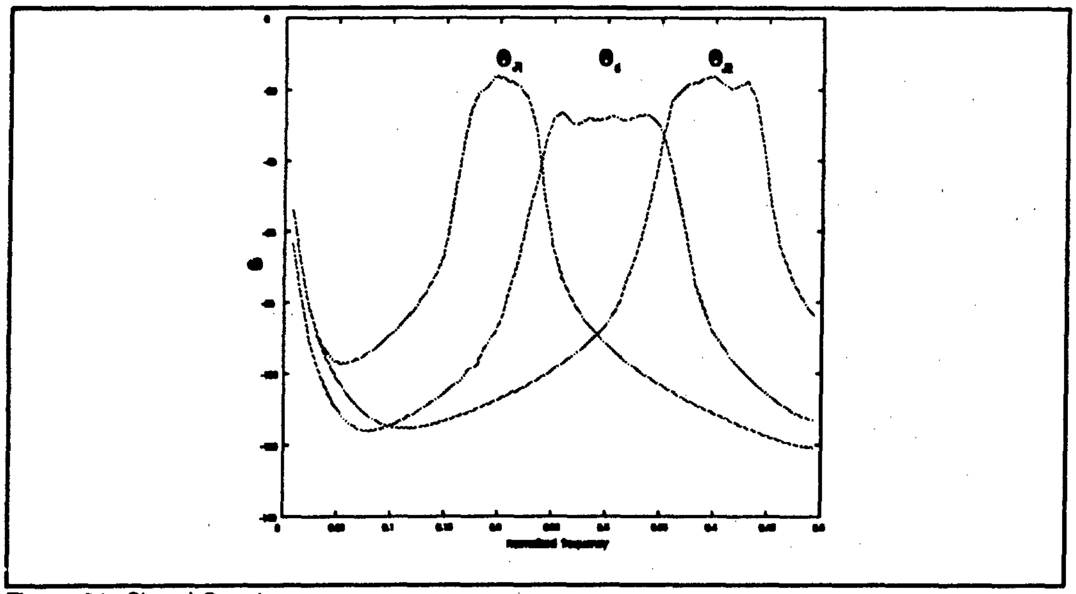

Figure 21 Signal Spectrum

Table 1 Signal Characteristics				
SOURCE	Θ	POWER	CENTER FREQUENCY	BANDWIDTH
desired signal	0°	0.1	$0.3 f_0$	0.1
jammer #1	45°	1.0	$0.2 f_0$	0.05
jammer #2	-60°	1.0	$0.4 f_0$	0.07
white noise per tap	N/A	0.1	N/A	N/A

E. Example

The example considered in this section is based on a linear array geometry and is similar to that presented by Frost [11]. The array is composed of four sensors equispaced at half-wavelength. Each tapped-delay-line filter has four taps. The tap spacing Δ corresponds to a frequency of $f_0 = 1.0$ ($f_0 = 1$ defines a frequency of $1/\Delta$ Hz). The signal environment is characterized by one desired signal and two active jammers. The signals are assumed to emanate from sources in the far field of the array and impinging from directions θ_d, θ_{J_1}, and θ_{J_2}. The propagation medium is assumed to be linear and non-dispersive. The

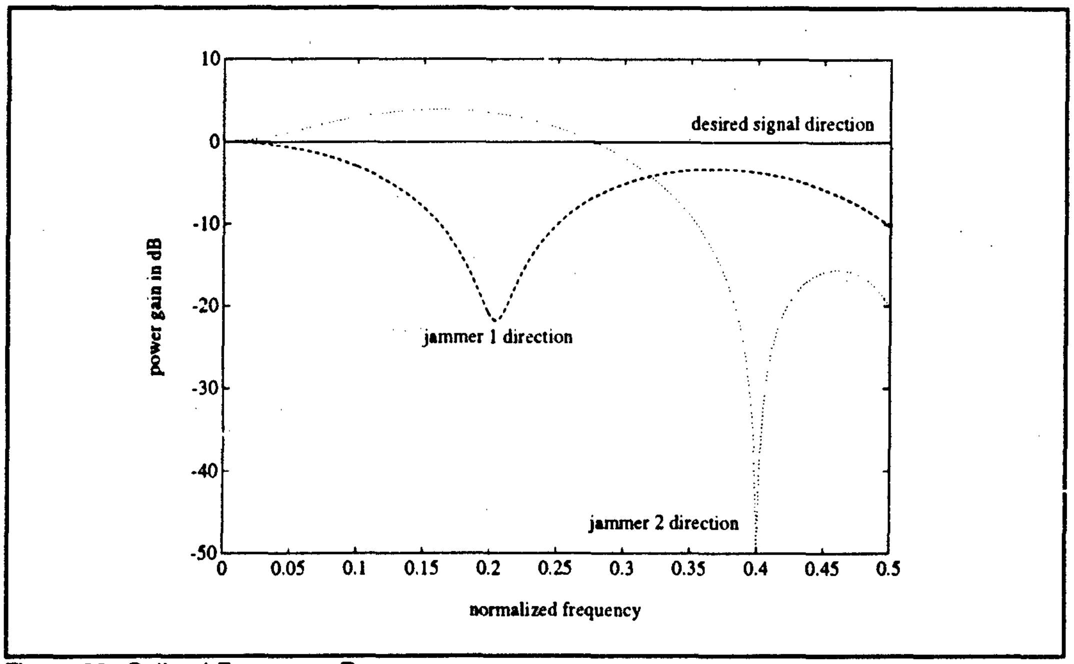

Figure 22 Optimal Frequency Response

look-direction signal and noises are assumed to be statistically uncorrelated with the non look-direction signals, explicitly ruling out multipath. The signal environment is described in table 1 and figure 21. The power spectral density depicted in figure 21 is a plot of normalized frequency versus power in dB.

The vector of look-direction filter coefficients $\boldsymbol{F}$ is designed to provide a distortionless response. This desired response in combination with our directional constraint yields the aforementioned linearly constrained minimum variance distortionless response (MVDR) processor. The optimal weight vector given by equation (2-16) produces the frequency response shown in figure 22. This figure is actually three steady state plots of normalized frequency versus power gain superimposed upon one another. The first plot depicts the distortionless look direction frequency response, while the second and third plots are the frequency responses in the directions of jammer 1 and jammer 2, respectively.

The ensemble characteristics with the known correlation matrix of the direct form and GSC form TDL structure are now examined. This type of computer investigation is termed an analysis model, as opposed to a simulation which will be addressed later. The magnitude of the step size determines the speed of convergence and the steady state mean-square error of the CLMS and LMS algorithms. The upper bound, given in equations (2-45) and (1-50), require a priori knowledge of the data correlation matrix. A step size which is near the upper bound may lead to overshoot with data fluctuations and the mean-square error increases proportionally with the step size magnitude. A step size which is too small will increase the amount of time required to convergence. For this example, the magnitude of the step size is taken to be

$$\frac{1}{10\,trace(R)} \tag{2-91}$$

where **R** is the relevant correlation matrix. It is noted that this step size is the value recommended by Wid~ow [20, p. 106].

Figure 23 shows the ensemble average weight transients resulting from iteratively applying the CLMS algorithm in equation (2-20). Figure 24 depicts the ensemble mean square error or learning curves. Since the LMS algorithm's desired reference signal d(k) is equal to zero in the constrained algorithms, the mean-square error is equivalent to the ensemble output power; that is, $\boldsymbol{E}[\varepsilon^2] = \boldsymbol{E}[(d(k) - y(k))^2] = \boldsymbol{E}[y^2(k)]$. The GSC form requires the choice of signal blocking matrix be made for further analysis. The choice of W_{s_1} defined in equation (2-82) yields a performance identical to that of the direct form, and is therefore not presented. Figures 25 and 26 present the ensemble weight transients and output power for the signal blocking matrix W_{s_2} defined in equation (2-83). For this example, the

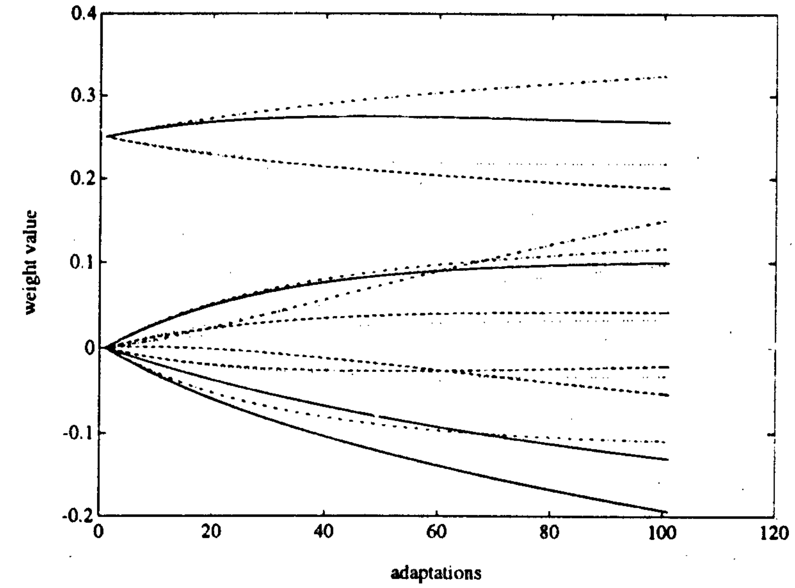

Figure 23 Ensemble TDL Direct Form Weight Vector Transients

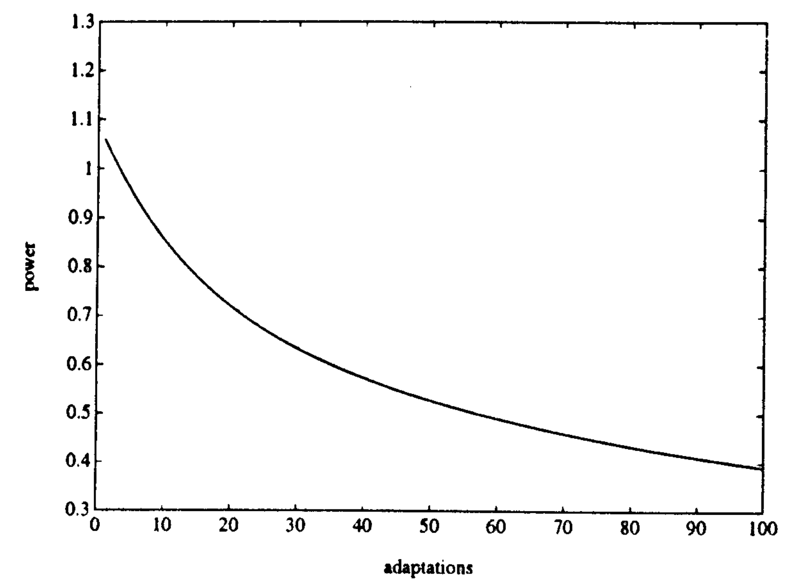

Figure 24 Ensemble TDL Direct Form Learning Curve

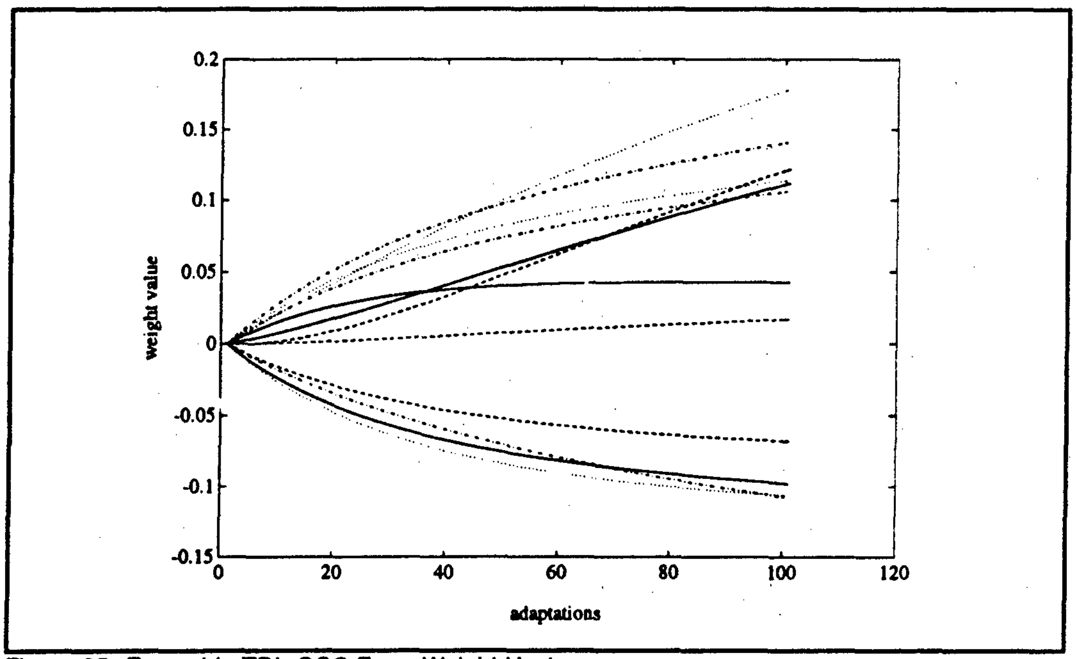

Figure 25 Ensemble TDL GSC Form Weight Vector
Transients for Signal Blocking Matrix Ws2

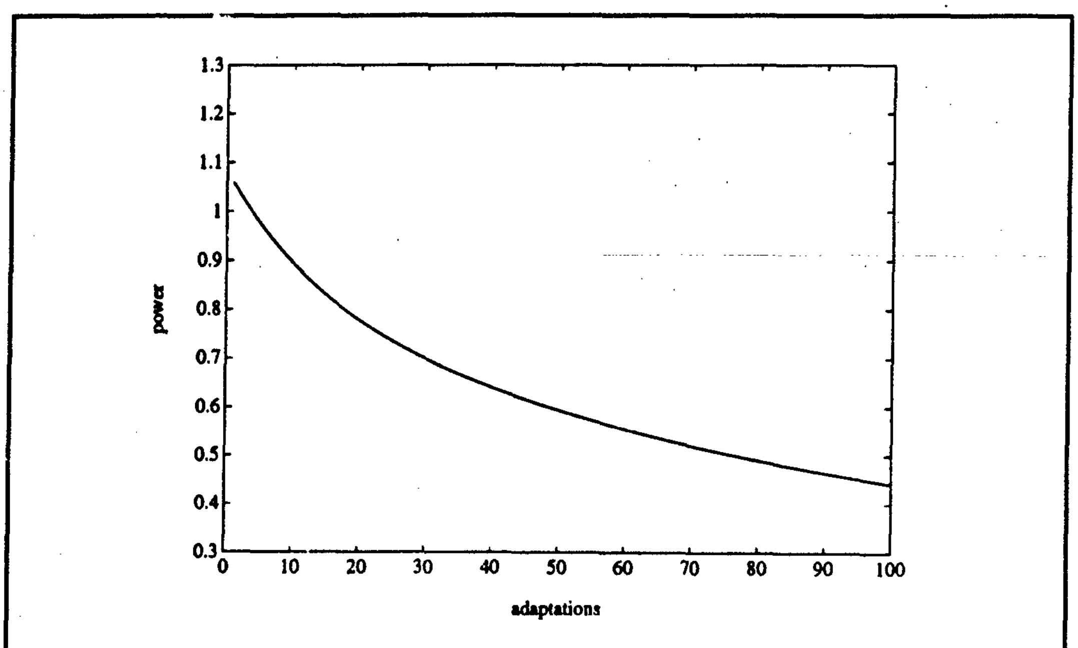

Figure 26 Ensemble TDL GSC Form Learning Curve
for Signal Blocking Matrix Ws2

eigenvalues of the observation correlation matrix $\boldsymbol{R}_{xx}$ range from 0.100 to 10.056. The eigenvalue spread of the direct form quadratic matrix function $\boldsymbol{P}\boldsymbol{R}_{xx}\boldsymbol{P}$ is 75.447. The eigenvalue spread of the GSC quadratic matrix function $\hat{W}_{S2}\boldsymbol{R}_{xx}\hat{W}_{S2}^{T}$ is 57.8019, depicting the situation where the GSC performance will exceed that of the direct form.

The simulation model analyzes the actual behavior of the adaptive processor. This is accomplished through generating all propagating signals and simulating the sensor noise at each element. The signals impinge upon the array as they would in a true deployed system, and the statistics are data driven. This differs from the analysis model, where simply an investigation and iteration of equations given the correlation matrix is performed.

The simulation of the tapped-delay-line structure for both the direct form and GSC form is now considered. The propagating signals are modeled as zero-mean stochastic processes with Gaussian distributions. The signals are filtered to provide the spectral and spatial characteristics described in table 1 and figure 21. The constant step size simulations are based on an LMS gain as described in equation (2-91). The normalized step size simulations are formulated as described in section D of this chapter.

The simulation executed one hundred adaptations over a one hundred independent realizations of the input process. The same observation process at the sensor inputs were used for each form to provide a meaningful comparison. The direct form constant step size mean weight vector trajectories, ensemble averaged output power and frequency response evaluated at the mean value of $W_{DF}(101)$ are displayed for this realization in figures 27, 28 and 29. The analogous plots for the GSC are presented in figures 30,31 and 32.

The normalized step size simulations are presented for the same observation process realization generated above. It is noted that the ensemble statistics are not available due to

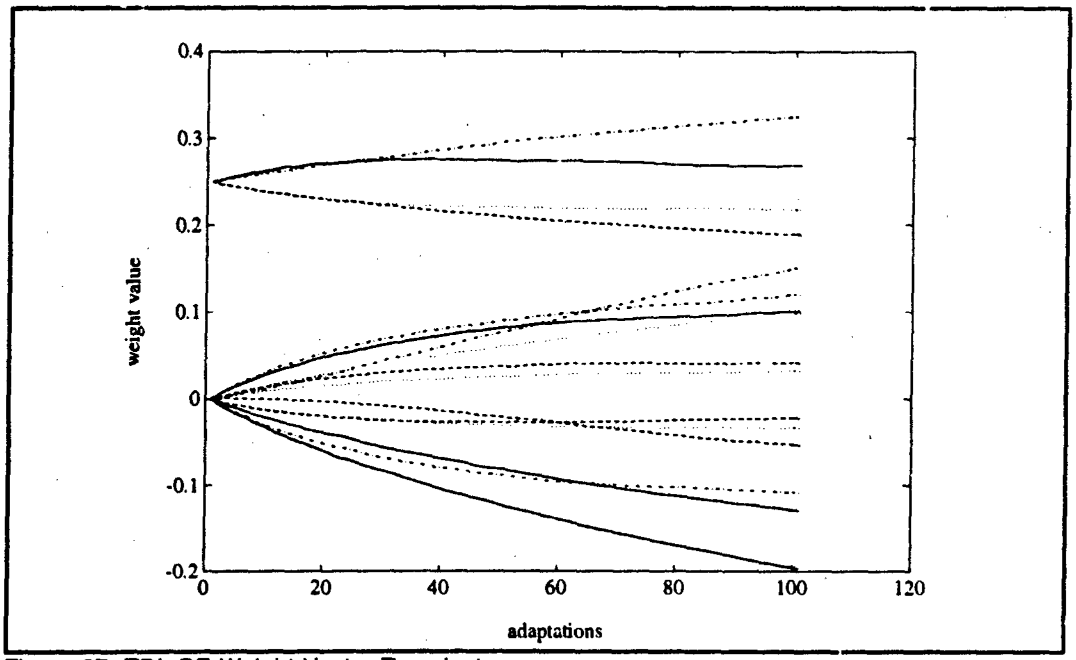

Figure 27 TDL DF Weight Vector Transients

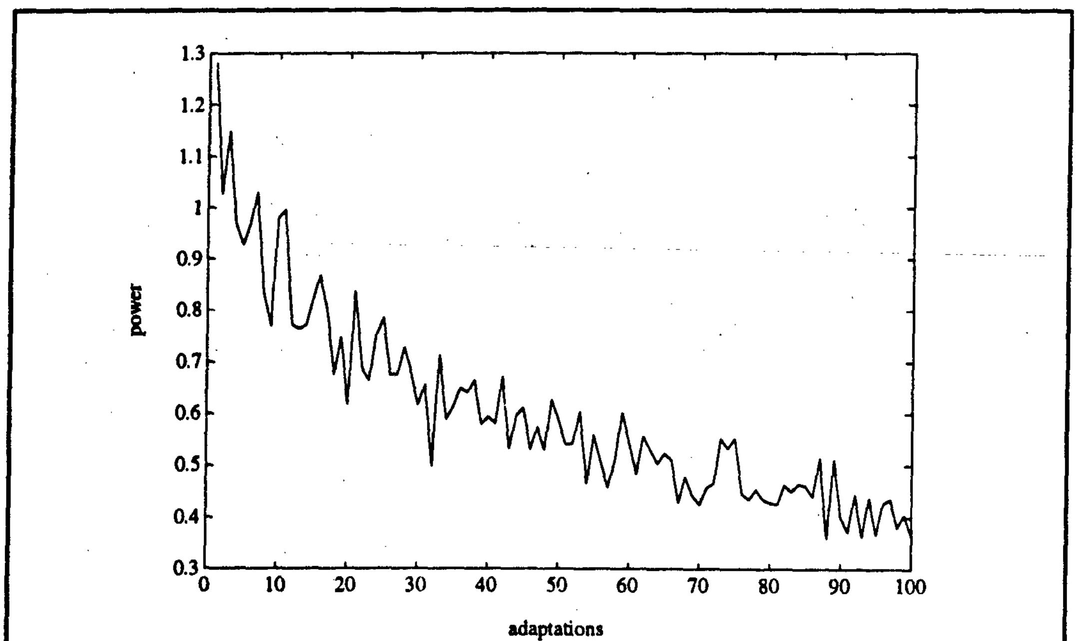

Figure 28 TDL DF Learning Curve

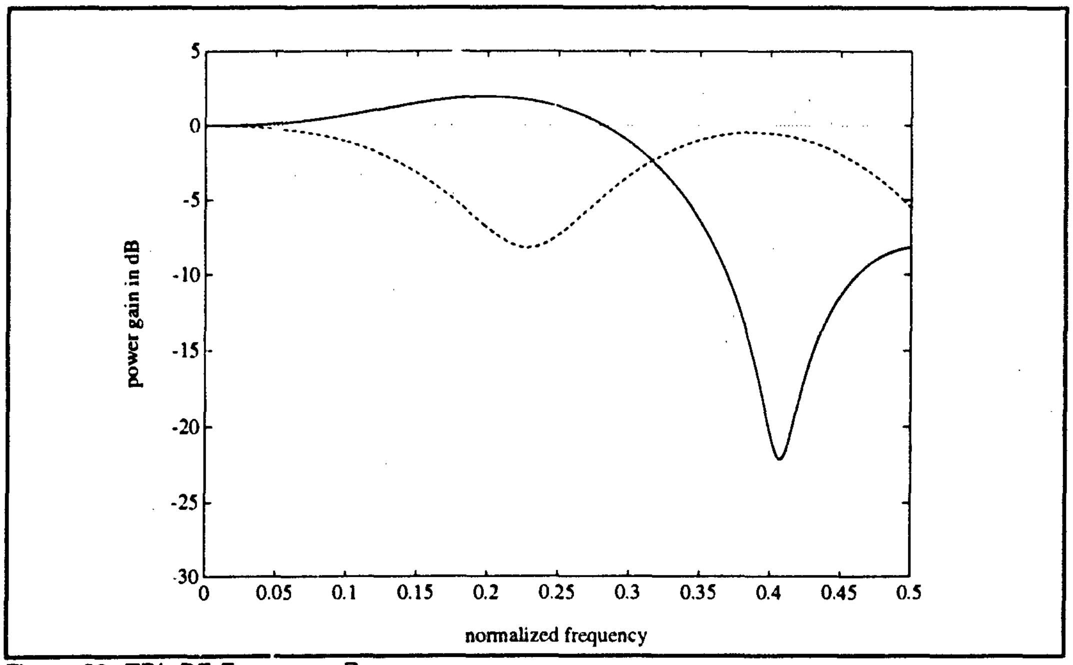

Figure 29 TDL DF Frequency Response

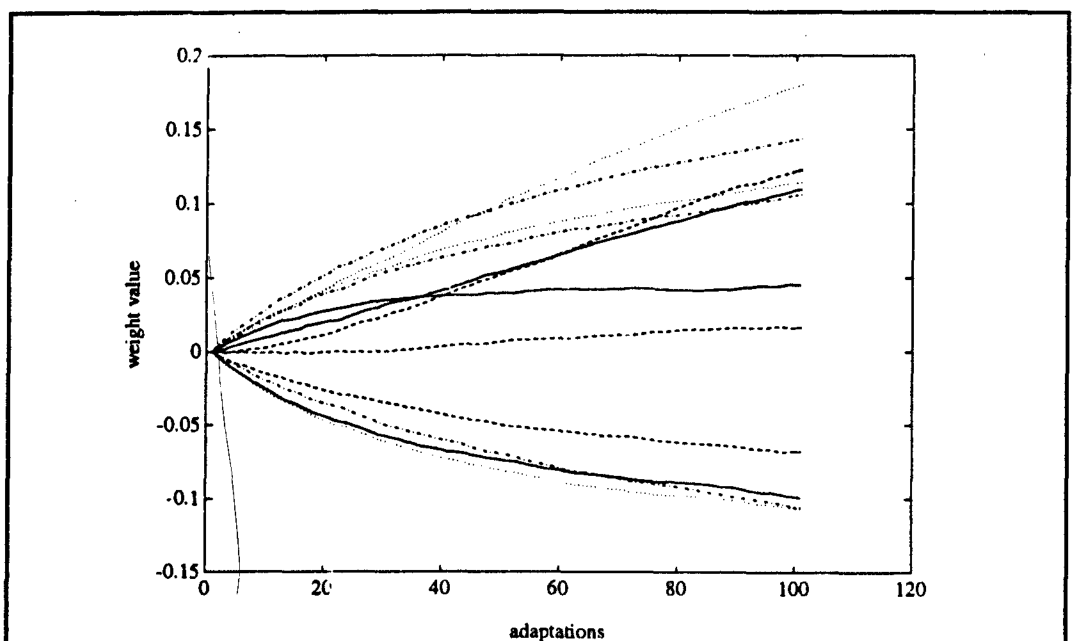

Figure 30 TDL GSC Ws2 Weight Vector Transients

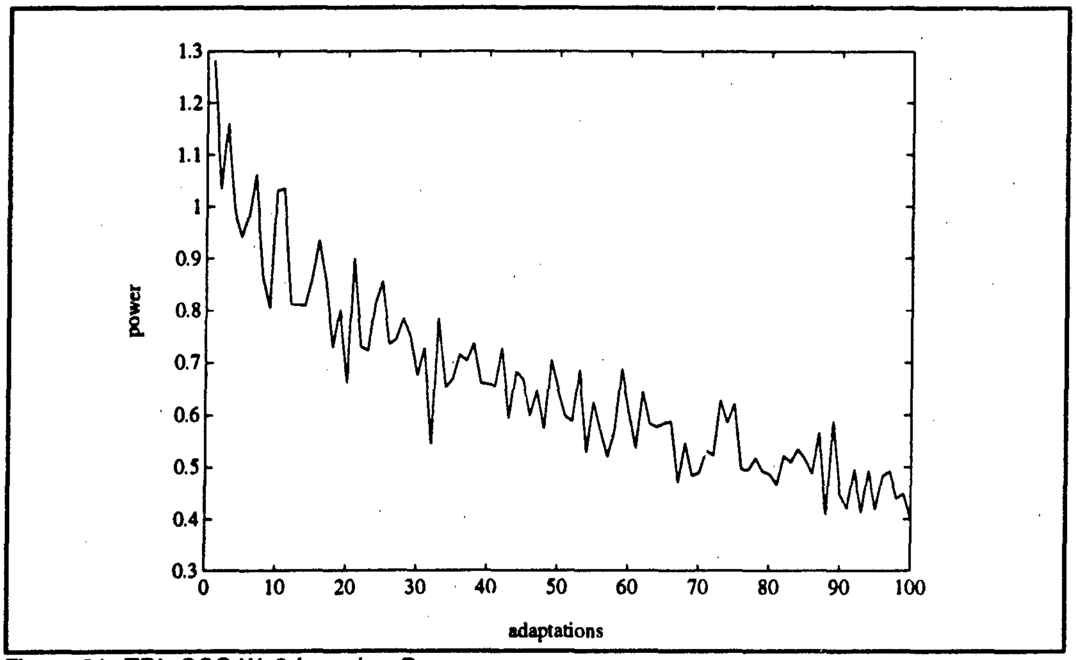

Figure 31 TDL GSC Ws2 Learning Curve

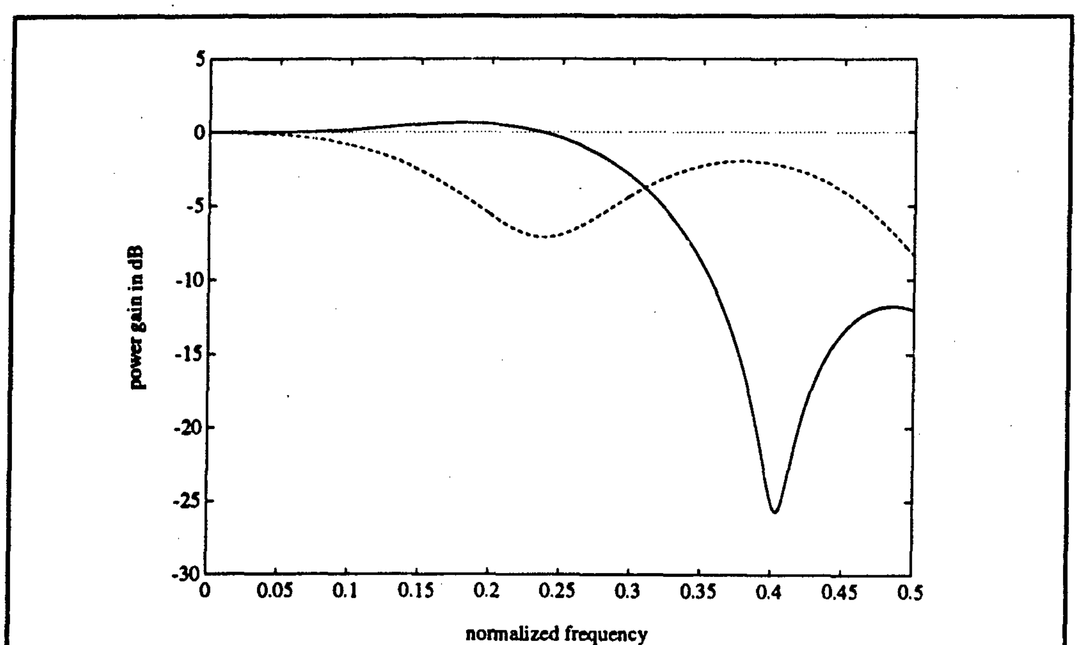

Figure 32 TDL GSC Ws2 Frequency Response

the algorithm's new dependency upon the data variance. Approximations to the ensemble performance are made over one hundred adaptations of the observation process realizations.

The time-varying LMS algorithm begins with an initial condition of the input signal variance, which must be large for fast convergence and subsequently decrease in magnitude for minimum steady state mean-square error. In accordance with the earlier discussion on convergence, too small of an initial variance estimate leads to a large initial step size and a corresponding initial overshoot. A value of the initial variance estimate which yields similar performance to that exhibited by the fixed step size of equation (2-91) has been experimentally determined to be

$$\hat{\sigma}^2_{X_l}(0) = \frac{2\, \mathbf{X}_l^T \mathbf{X}_l}{J\, trace(\mathbf{R})} \tag{2-92}$$

where **R** is the relevant correlation matrix for the form under consideration.

The plots of weight vector dynamics, temporally averaged learning curves and frequency responses are displayed in figures 33, 34 and 35 for the direct form; figures 36, 37 and 38 for the GSC W_{s2} form. The learning curves are roughly equivalent to the constant gain simulations. The frequency response graphs depict that the ability to adjust the step size independently for each element based upon the input variance provides more accurate placement of the nulls as well as deeper attenuation. The problem of determining a step size gain based upon misadjustment is relieved by the step size normalization.

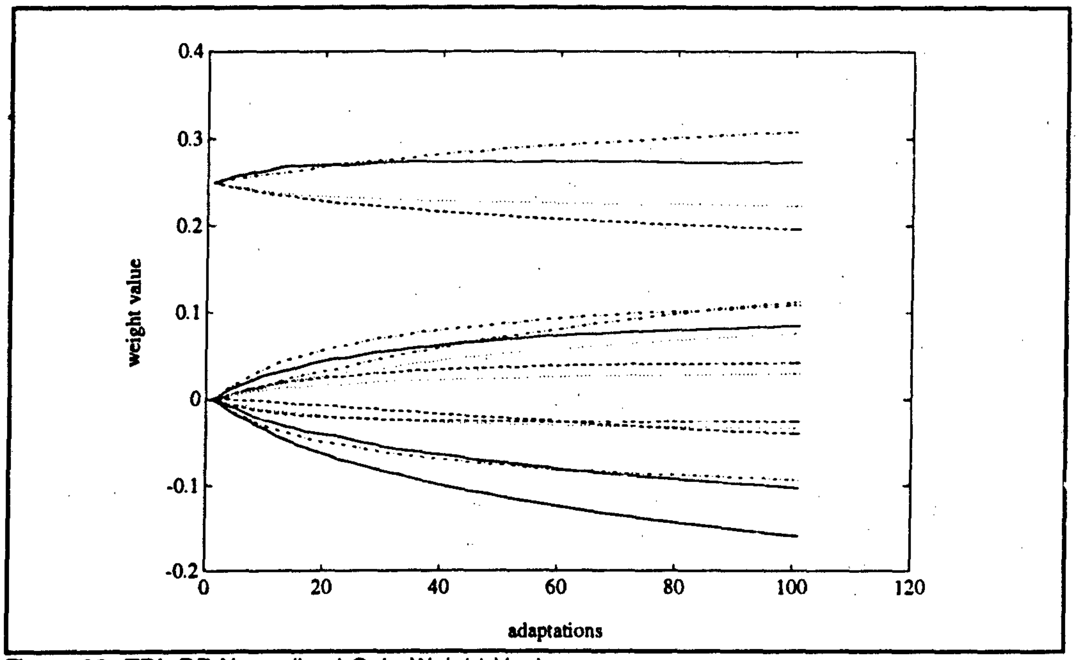

Figure 33 TDL DF Normalized Gain Weight Vector Transients

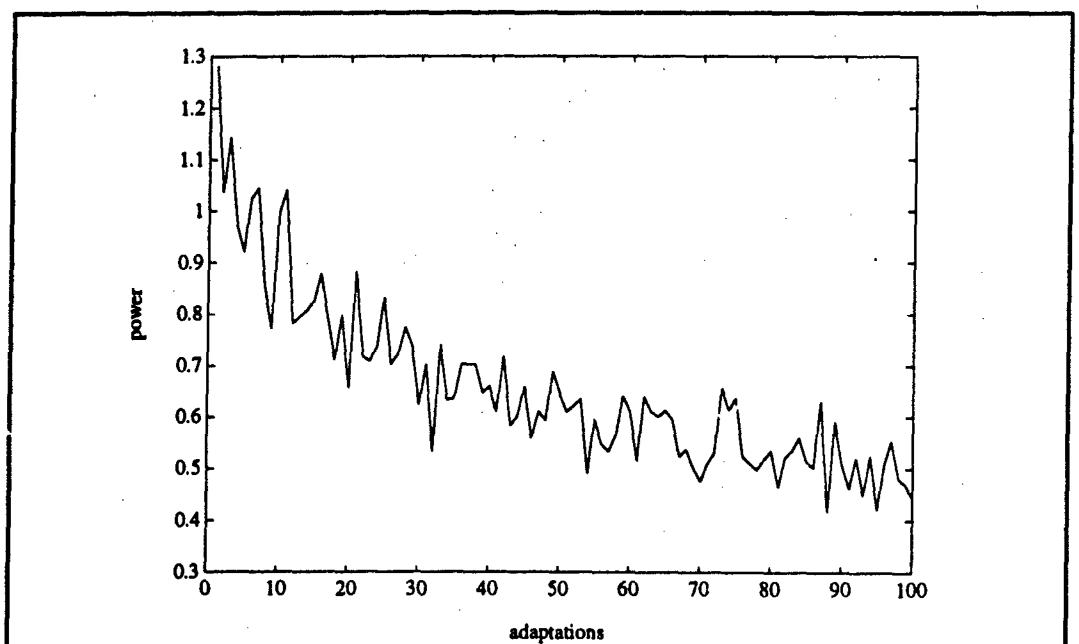

Figure 34 TDL DF Normalized Gain Learning Curve

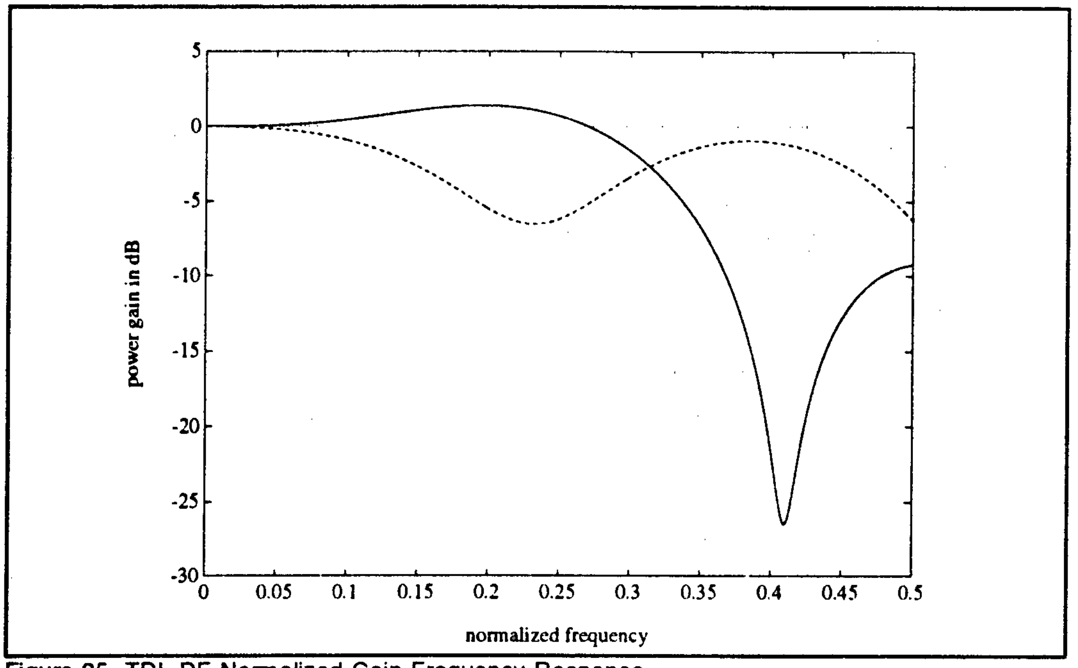

Figure 35 TDL DF Normalized Gain Frequency Response

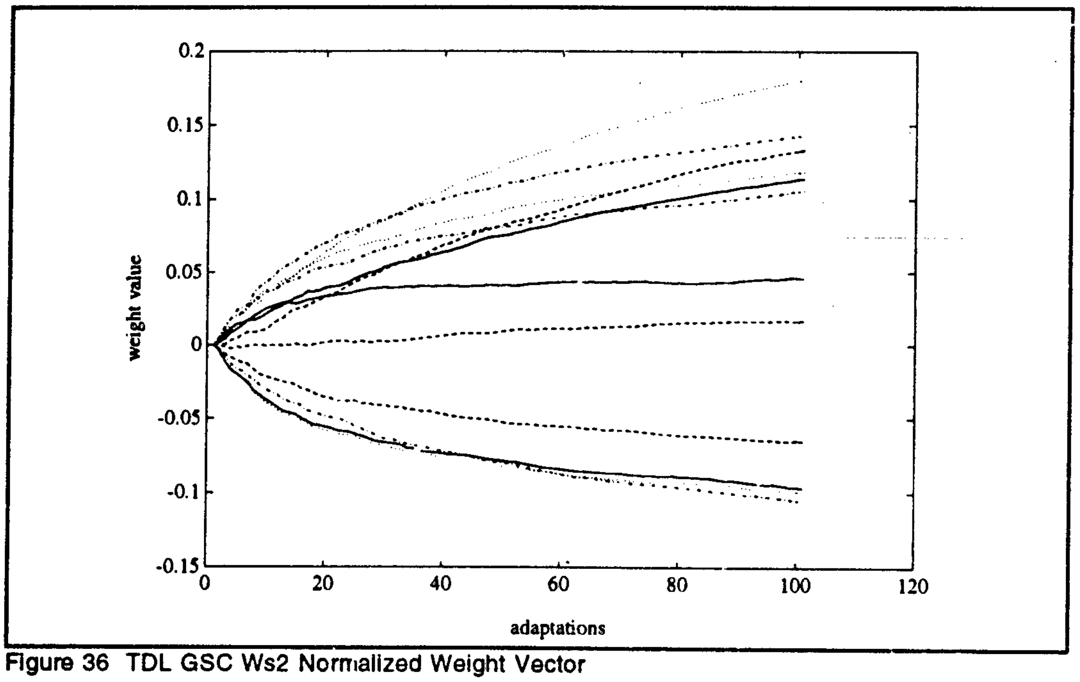

Figure 36 TDL GSC Ws2 Normalized Weight Vector Transients

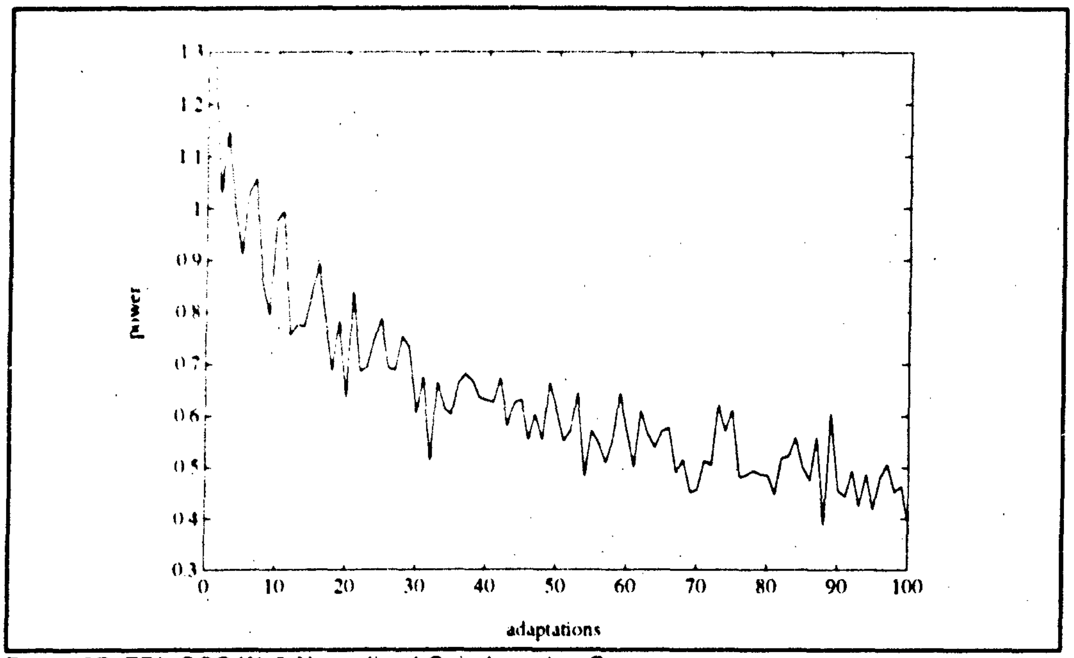

Figure 37 TDL GSC Ws2 Normalized Gain Learning Curve

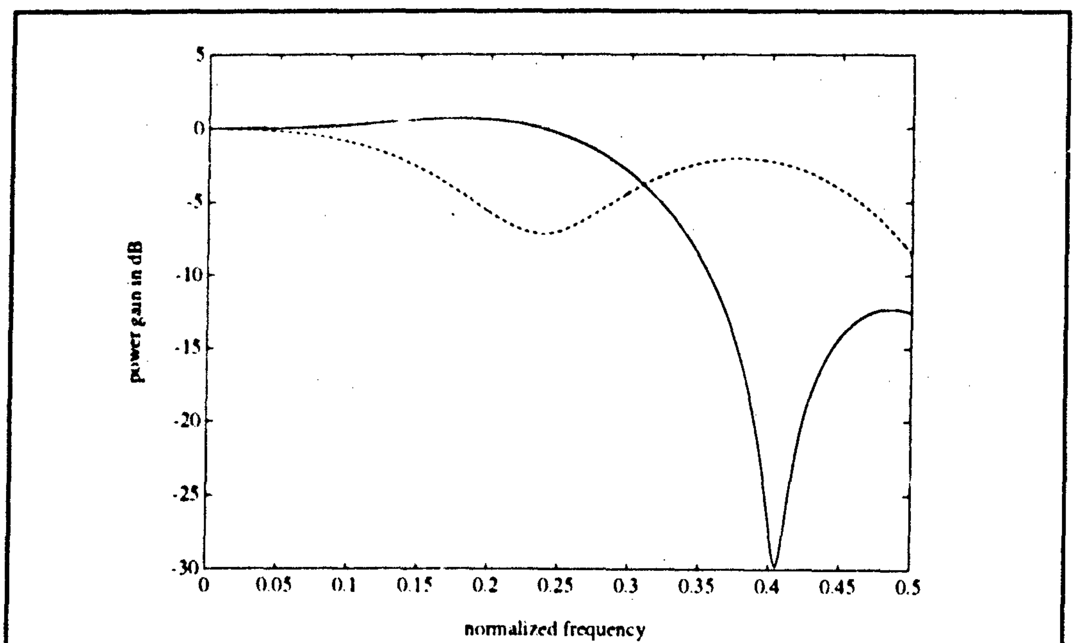

Figure 38 TDL GSC Ws2 Normalized Gain Frequency Response

F. Conclusions

The GSC form and the direct form TDL structure adaptive array sensor have identical dynamic behavior if the LMS and CLMS step sizes are constants which provide the same level of misadjustment, the GSC form constraint enforced through the conventional beamforming matrix W_C is equivalent to that of the direct form algorithmic constraint and the GSC form signal blocking matrix W_S is composed of orthogonal rows which map the constraint nullspace. Since the emphasis in the examples was a MVDR array which provides a distortionless look direction response, the constraint was consistent across all forms and examples. The difficulty in choosing a signal blocking matrix composed of nonorthogonal rows which would consistently provide a better dynamic behavior was discussed at the end of section II.C. It was found that the GSC form signal blocking matrix implementing an adjacent element subtraction led to a quicker convergence for the signal and array geometry presented in the examples of this section.

The LMS algorithm with a time-varying step size presents two benefits which are of importance in this study. First, the choice of step size selection is simplified due to its update as a function of input signal variance (or equivalently, the eigenvalues of the input process correlation matrix). Second, and most important, this normalization of the step size leads to a reduction in the dependency between the speed of convergence and the eigenstructure of the relevant correlation matrix. This is realized due to the fact that the algorithm can now update each weight independently with separately valued time-varying step sizes. Thus, each mode of the algorithm can adapt at its own speed. The correlation matrices which then determine the speed of convergence are then given by $\boldsymbol{P}\,\boldsymbol{\Omega}_{DF}(k)\,\boldsymbol{R}_{xx}\,\boldsymbol{P}$

and $\hat{W}_S \, \Omega_{GSC}(k) \, R_{xx} \, \hat{W}_S^T$ where $\Omega_F(k)$ is a diagonal matrix of the proper dimension for form F whose elements at time k are given by the individual step sizes presented in equations (2-86) and (2-87). For the example presented, the eigenvalue spread of the normalized algorithms and that of the standard algorithm were nearly identical. This is believed to be due to the modest eigenvalue spread produced by this example's geometry.

III. CONSTRAINED PROCESSORS WITH ORTHOGONAL FILTER STRUCTURE

This chapter investigates the performance of the GSC form linearly constrained wideband adaptive array sensor with the adaptive processor replaced by an orthogonal filter structure. The motivation for utilizing an orthogonal filter realization of the adaptive processor is that we desire to obtain a new set of data vectors which exhibit minimum correlation to provide as the input to the adaptive filter.

Consider the GSC form array with a single distortionless constraint as shown in figure 20 of the last chapter. The operations considered in this chapter are realized by transforming the data present on each tap in the figure prior to weighting. This delay-and-transform operation is conveniently represented, in general, by the transformation matrix $\boldsymbol{Q}$. This structure is depicted in figure 39. It will be seen that the elements of the

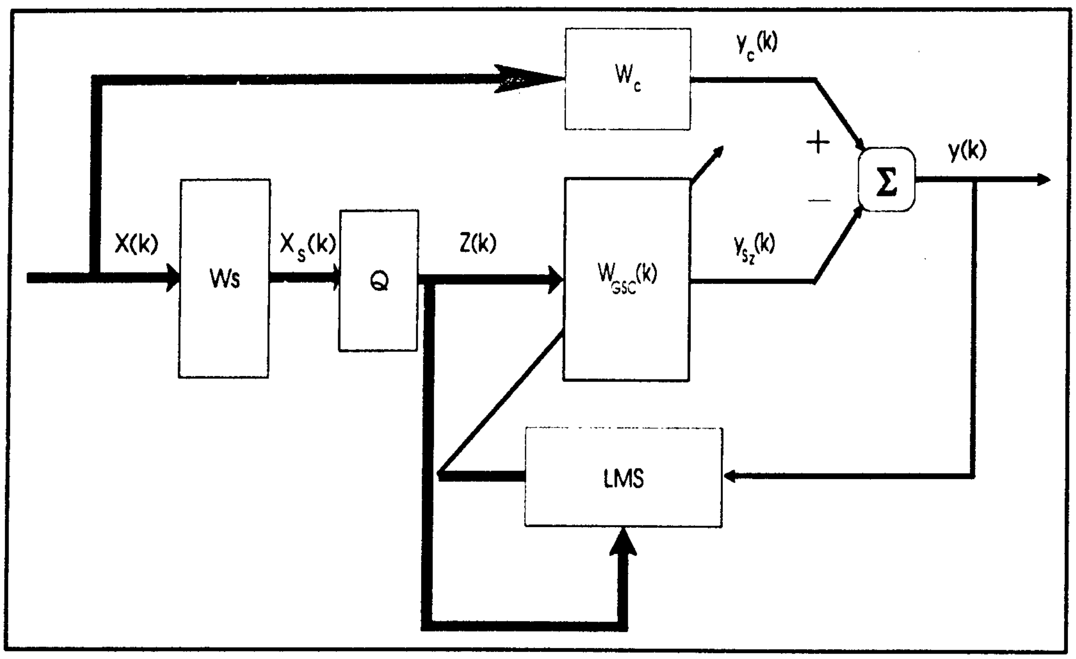

Figure 39 Transform Domain GSC Processor Structure

transformation matrix may be adaptive or fixed. The goal of this transformation, as stated initially, is to provide a less correlated input $\boldsymbol{Z}(k)$ to the adaptive weight vector $\boldsymbol{W}_{GSC}(k)$.

This chapter will first consider the case of a fixed orthogonal transform structure in section III.A, where the Discrete Fourier Transform (DFT) and the Discrete Cosine Transform (DCT) with a normalized step size will be extended to the multichannel case of interest. Linear prediction will be reviewed in section III.B to provide the foundation for the derivation of the lattice filter structure in section III.C. The converged lattice structure will be related to the Gram-Schmidt orthogonalization process, and the multichannel adaptive lattice structure will then be examined as a replacement for the TDL processor. The Gram-Schmidt orthogonal structure will then be derived directly in section III.D, and the characteristics of this structure in the GSC form processor will be examined. Simulations in section III.E will then be used to compare the performance of these structures

A. Fixed Orthogonal Transform Domain Structure

Very early in the history of adaptive array research many investigators examined frequency domain LMS filters [31,35,36]. The frequency domain transformation is usually implemented with $\boldsymbol{Q}$ being an invertible matrix composed of the DFT or the DCT coefficients which operate in an identical manner upon each TDL. The frequency domain has an intuitive appeal as a method of improving the performance of wideband adaptive arrays. The transformation from the temporal domain to the frequency domain results in frequency subbanding, in effect reducing the wideband problem to discrete frequency bins. The initial research in this area was limited to the analysis of the LMS algorithm with a constant step size. Compton [29] then published a report which showed that the frequency domain structure performance was identical to that of the tapped-delay-line processor, again utilizing a fixed step size in the LMS algorithm. Subsequently, many other researchers began examining the use of transform domain adaptive filtering for narrowband single channel applications which considered the use of time-varying LMS step sizes [28,30,40,41,42,43,44,45,46,47].

Section III.A.1 will present the equivalence of invertible linear transforms for completeness. This equivalence directly applies to the DFT and DCT processors with a constant step size and demonstrates that the resulting array transient and steady-state behavior is unchanged by the transformation. Section III.A.2 will then be concerned with the main contribution of this section, the wideband multichannel extension of the transform domain filtering research of Narayan [28], Lee [30], Clark [40] and Jenkins [42]. Section III.A.3 will present the DCT, and section III.A.4 will consider the convergence of these transform domain algorithms.

1) The DFT Transform: In order to facilitate the following development, we briefly return to the TDL structure GSC in order to abbreviate notation. Define the quadratic correlation matrix function and the quiescent response vector as

$$R_{x_s} = \hat{W}_s R_{xx} \hat{W}_s^T \tag{3-1}$$

$$R_{xd} = \hat{W}_s R_{xx} \hat{W}_c \tag{3-2}$$

Utilizing this notation, the optimal GSC weight vector in equation (2-78) may be written in a form similar to that of the Wiener-Hopf equation which was introduced in equation (1-18)

$$W_{opt} = R_{x_s}^{-1} R_{xd} \tag{3-3}$$

The optimal value of the GSC lower path output can be written as

$$y_{s_{opt}} = W_{opt}^T \hat{X}_s = \hat{X}_s^T W_{opt} \tag{3-4}$$

Now, extending the derivation of Compton [29], consider the GSC form array with an invertible transform Q introduced after the signal blocking matrix and before the adaptive processor. We note that any reversible operation can not affect the performance of the array. This situation is clear by considering the new data vector Z, which becomes the input to the adaptive processor, where

$$Z(k) = Q\hat{X}_s(k) \tag{3-5}$$

The transform domain optimai weight vector is given by

$$W_{Q_{opt}} = (QR_{x_s}Q^T)^{-1}QR_{xd} \tag{3-6}$$

and the output of the transform domain structure's lower path is

$$\begin{aligned} y_{s_{Z_{opt}}}(k) &= Z^T W_{Q_{opt}} \\ &= \hat{X}_s^T Q^T W_{Q_{opt}} \\ &= \hat{X}_s^T Q^T (QR_{x_s}Q^T)^{-1} QR_{xd} \\ &= \hat{X}_s^T R_{x_s}^{-1} R_{xd} \\ &= y_{s_{opt}}(k) \end{aligned} \tag{3-7}$$

In fact, it is easily shown that optimal weight vector of the two structures are related by

$$W_Q = [Q^T]^{-1} W \tag{3-8}$$

Hence, the transform domain GSC processor with an invertible transform and the TDL GSC ($Q = I$) both converge to the Wiener-Hopf solution.

Consider the symmetrical DFT which is implemented upon the signals present at each channel of the blocking matrix output. For the l-th channel,

$$Z_{ln} = \sum_{m=1}^{J} X_{s_m} e^{-j\left(\frac{2\pi(m-1)(n-1)}{J}\right)} \quad l=1,K \quad n=1,J \tag{3-9}$$

or, equivalently, for the stacked (K-1)xJ dimensional data vector X at time k

$$Z(k) = Q_{DFT}\hat{X}_s(k) \tag{3-10}$$

and Q_{DFT} is simply the rank (K-1)xJ matrix of exponential coefficients which realize equation (3-9). This algorithm has the additional benefit of not requiring an inverse transform to obtain the time domain output [24].

Due to the symmetry of the DFT matrix, we may write equation (3-8) as

$$W_{QDFT} = \bar{Q}_{DFT}^{-1} W \tag{3-11}$$

and the weight vector behind each element is simply the inverse DFT of the TDL structure weight vector.

The transform relation of equation (3-11) depicts that the steady state value of the DFT and the TDL structure are the same. Following Compton [29], we now show through the analysis of the correlation matrix eigenstructure that the transient behavior of the two arrays are also identical for a fixed step size. For the present time, we will be concerned solely with the frequency domain transform, and therefore the subscript DFT will be suppressed.

Define the transform domain correlation matrix as

$$R_{zz} = \bar{Q} R_{x_s} Q^T \tag{3-12}$$

where the operator $\bar{(\cdot)}$ denotes the complex conjugate. Since R_{x_s} and R_{xx} are both assumed to be positive definite and Hermitian, R_{x_s} has a complete set of orthonormal eigenvectors whose corresponding eigenvalues are real and positive. Denote the *i-th* eigenvalue of R_{x_s} as λ_i and the *i-th* eigenvector as φ_i. Then the orthonormal condition may be explicitly written as

$$\varphi_i^T \varphi_j = \delta_{ij} \tag{3-13}$$

We proceed by defining the transform eigenvector

$$\varphi_{z_i} = \sqrt{J}\, Q^{-1} \varphi_i \tag{3-14}$$

and note that

$$\varphi_{z_i}^T \varphi_{z_j} = \varphi_i^T \sqrt{J}\, [Q^{-1}]^T \sqrt{J}\, Q^{-1} \varphi_i \tag{3-15}$$

Utilizing the symmetry of the DFT and the fact that the transformation matrix is realized in a block diagonal form due to the same transform taking place on each element, we find

$$Q^{-1} = \frac{1}{J} \bar{Q} \tag{3-16}$$

$$[Q^{-1}]^T = \frac{1}{J} Q \tag{3-17}$$

and equation (3-15) becomes

$$\varphi_{z_i}^T \varphi_{z_j} = J \varphi_i^T \frac{1}{J} Q Q^{-1} \varphi_i = \delta_{ij} \tag{3-18}$$

Now, consider the eigenvector equation

$$R_{x_t} \varphi_i = \lambda_i \varphi_i \tag{3-19}$$

Substituting equation (3-14) into (3-19) and multiplying by Q^{-1} yields

$$Q^{-1} R_{x_t} Q\, \varphi_{z_i} = \lambda_i \varphi_{z_i} \tag{3-20}$$

and using the symmetry of the DFT, equations (3-16) and (3-20) can be written as

$$(\bar{Q} R_{x_t} Q^T) \varphi_{z_i} = J \lambda_i \varphi_{z_i} \tag{3-21}$$

so that it is evident that

$$R_{zz} \varphi_{z_i} = J \lambda_i\, \varphi_{z_i} \tag{3-22}$$

Then each eigenvalue of R_{zz} is simply J times the corresponding eigenvalue of R_{x_t}, and we conclude that the eigenvalue spread for the frequency domain structure and the TDL structure GSC are identical.

2) <u>The DFT Frequency Domain Structure with Subband Normalization:</u> Both Compton [29] and Lee and Un [30] have examined the performance of the DFT algorithm

presented in equation (3-21). Compton concluded that the TDL and DFT structures always perform identically in his analysis of single and multiple channel adaptive filters. His work, however, did not consider normalization of the adaptive step size. Lee and Un realized the possibility of achieving better convergence properties through the normalization of the step size, as have Narayan et al. [28]. However, both of the latter authors restricted their analysis to single channel filters and hence, were not able to realize the normalization conditions for the adaptive algorithm step size which is now presented to yield speed of convergence improvement in the adaptive array sensor problem. Following the notation of Narayan, we define these conditions and present a method of multichannel variance averaging which results in better dynamic behavior while achieving the same steady state Wiener solution.

Consider the normalized step size for the unconstrained LMS algorithm

$$\boldsymbol{W}(k+1) = \boldsymbol{W}(k) + \mu(k)y(k)\boldsymbol{X}_s(k) \tag{3-23}$$

where at time k

$$\mu(k) = \frac{1-\beta}{\hat{\sigma}^2_{x_s}(k)} \tag{3-24}$$

is a diagonal matrix composed of the averaged signal variances. We note that the averaging operation for the TDL structure was a multichannel extension of the only one which can be considered in the single channel case; averaging over the channel. Thus, for the *n-th* channel

$$\hat{\sigma}^2_{x_{s_n}}(k) = \beta\,\hat{\sigma}^2_{x_{s_n}}(k-1) + \frac{(1-\beta)}{J}\boldsymbol{X}^T_{s_n}(k)\boldsymbol{X}_{s_n}(k) \tag{3-25}$$

The steady state convergence of the normalized step size LMS algorithm and the variance estimate of equations (3-24) and (3-25) were considered in section D of chapter II.

It would seem reasonable to conclude that better transient behavior would occur when the weights were able to adapt in each frequency bin through a normalized step size in which the variance was averaged over the power present in that frequency bin. An algorithm which accomplishes this utilizes the estimate

$$\mu(k) = \frac{1-\beta}{\hat{\sigma}^2_{z_{f_n}}} \tag{3-26}$$

$$\hat{\sigma}^2_{z_{f_n}}(k) = \beta\, \hat{\sigma}^2_{z_{f_n}}(k-1) + \frac{(1-\beta)}{K-1} Z^T_{f_n}(k)\, Z_{f_n}(k) \tag{3-27}$$

and the weight state update

$$\mathbf{W}_{DFT}(k+1) = \mathbf{W}_{DFT}(k) + \mu(k)\, y(k)\, \mathbf{Z}(k) \tag{3-28}$$

where f_n is the *n-th* frequency bin.

3) <u>The Discrete Cosine Transform:</u> The DCT has the computational advantage of using only real numbers to provide a transform of the input data. Further, this transform was chosen for comparison to the DFT since recent articles in the literature [24, 28,44] reported that the narrowband single channel DCT adaptive filter provided better results than the DFT and other orthogonal transform filters for a class of data used in speech related applications.

The DCT orthogonal transform for the *l-th* channel is given by

$$Z_{ln} = \begin{cases} \dfrac{\sqrt{2}}{J} \displaystyle\sum_{m=1}^{J} X_{s_m} & l=1 \quad n=1,J \\ \dfrac{2}{J} \displaystyle\sum_{m=1}^{J} X_{s_m} \cos\left(\dfrac{\pi(2m+1)(n-1)}{2J}\right) & l=2,K \quad n=1,J \end{cases} \tag{3-29}$$

which is represented at time k by the (K-1)xJ dimensional vector

$$\mathbf{Z}(k) = Q_{DCT}\hat{\mathbf{X}}_s(k) \tag{3-30}$$

and Q_{DCT} is simply the rank (K-1)xJ matrix of real coefficients which realize equation (3-29).

4) <u>The Transform Domain Convergence:</u> The speed of convergence of the weight vector $\mathbf{W}_{DFT}(k)$ is a function of the eigenvalue spread of the correlation matrix

$$\mu R_{zz} = \mu [Q R_{x_s} Q^T] \tag{3-31}$$

We now assume without any loss of generality that the observation process variance is unity. Since the trace of a matrix is the sum of its spectrum, for any square matrix $\boldsymbol{R}$ we can say

$$\lambda_{R_{\max}} \leq trace(R) \tag{3-32}$$

Similarly, the determinant of a matrix is the product of its spectrum. For any positive definite hermitian matrix whose rank is greater than two, it can be shown [28] that

$$\lambda_{R_{\min}} \geq \det(P) \tag{3-33}$$

Therefore, an upper bound for the eigenvalue spread can be expressed as

$$\Gamma(R) = \frac{trace(R)}{\det(R)} \tag{3-34}$$

From equation (3-31), the trace and determinant are expanded as

$$trace(\mu\, \boldsymbol{R}_{zz}) = trace(\boldsymbol{R}_{x_s}) = KJ \tag{3-35}$$

$$\det(\mu\, \boldsymbol{R}_{zz}) = \det(\mu)\det(\boldsymbol{R}_{x_s}) \tag{3-36}$$

and from equation (3-34), the upper bound is found to be

$$\Gamma(\mu\, \boldsymbol{R}_{zz}) = \frac{KJ}{\det(\mu)\, \det(\boldsymbol{R}_{x_s})} = \frac{1}{\det(\mu)}\, \Gamma(\boldsymbol{R}_{x_s}) \tag{3-37}$$

Since the input process variance was assumed to be unity, the determinant of μ will be less than or equal to unity. Thus,

$$\Gamma(\mu\, \boldsymbol{R}_{zz}) \leq \Gamma(\boldsymbol{R}_{x_s}) \tag{3-38}$$

The equality condition holds when μ is normalized through the averaging of element variances and the subband normalization of equations (3-26) through (3-28) improve the transient characteristics of the sensor, as will be shown in the simulations at the end of this chapter.

B. Linear Prediction

The lattice structure to be derived solves the adaptive array sensor problem by performing two optimum estimation operations jointly. The first is linear prediction, which is used to transform the correlated inputs into a corresponding sequence of uncorrelated backward error predictions. The second estimation is the familiar optimum filtering operation which produces the estimate of the desired response, or equivalently, the array output. We now derive the optimum forward and backward scalar linear predictors. This development follows Haykin [27].

1) Forward Linear Prediction: The forward linear prediction problem is concerned with predicting a future value of a stationary discrete-time stochastic process given a set of past sample values of the process. Consider the time series {x(k), x(k-1),...x(k-J)} which is composed of J+1 samples. The operation of linear prediction makes an estimate of x(k) given the sample values x(k-1) through x(k-J). Let χ_{k-1} denote the J-dimensional space spanned by {x(k-1), x(k-2),...x(k-J)} and $\hat{x}(k \mid \chi_{k-1})$ denote the predicted value of x(k) given this set of samples. The predicted value may, in general, be expressed as some function Θ of the given samples

$$\hat{x}(k \mid \chi_{k-1}) = \Theta(x(k-1), X(k-2), \ldots x(k-J)) \tag{3-39}$$

and is termed linear prediction when the function Θ consists of a linear combination of the samples in the form

$$\hat{x}(k \mid \chi_{k-1}) = \sum_{n=1}^{J} w_{on}\, x(k-n) \tag{3-40}$$

The forward prediction error equals the difference between the actual sample value x(k) at time k and its predicted value $\hat{x}(k \mid \chi_{k-1})$. The forward prediction error is denoted $f_l(k)$ and given by

$$f_l(k) = x(k) - \hat{x}(k \mid \chi_{k-1}) \tag{3-41}$$

where the subscript l signifies the order

The single channel forward prediction operation is depicted in figure 40. The predictor consists of J unit-delays and J tap weights w_{ol}, $l=1,2,\ldots J$ which are assumed to be optimized in the mean-square sense and fed with the respective delayed samples of the observation process. The resultant output is the predicted value of x(k) given by equation (3-40). Then, we may write equation (3-41) as

$$f_J(k) = x(k) - \sum_{n=1}^{J} w_{on}\, x(k-n) \tag{3-42}$$

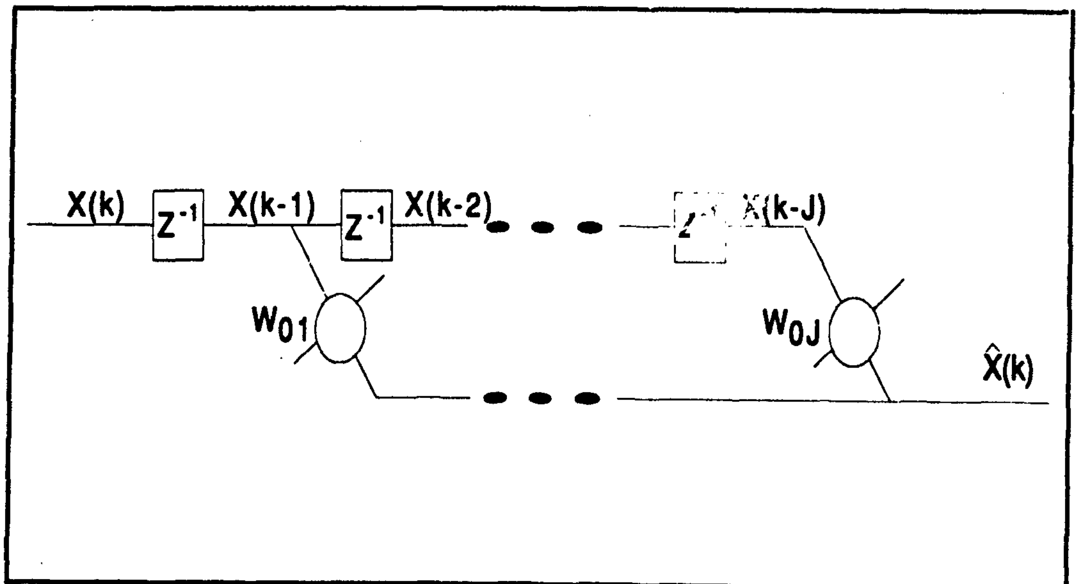

Figure 40 Forward Predictor

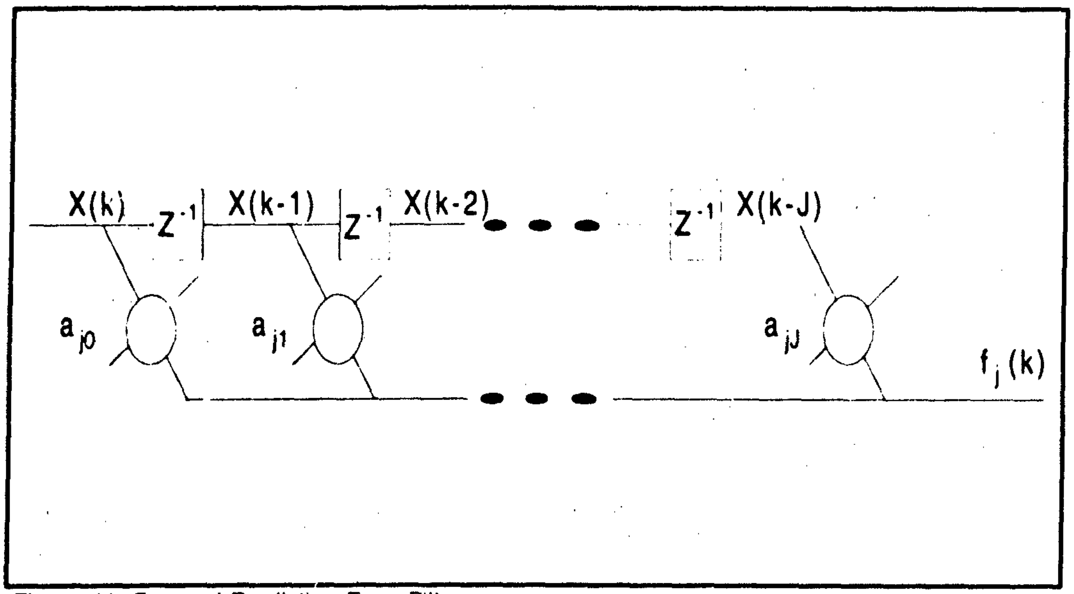

Figure 41 Forward Prediction Error Filter

Let a_{jn} , n=0,1,.. J denote the tap weight values of a new TDL filter which are related to the tap weights of the forward prediction filter as follows:

$$a_{jn} = \begin{cases} 1 & n=0 \\ -w_{on} & n=1,2,...J \end{cases} \tag{3-43}$$

Then, equation (3-42) may be expressed as

$$f_J(k) = \sum_{n=0}^{J} a_{jn}\, x(k-n) \tag{3-44}$$

which yields the filter depicted in figure 41 and is termed a forward prediction-error filter.

2) Backward Linear Prediction: We may also operate on the time series {x(k), x(k-1),...x(k-J+1)} to make a prediction of the sample x(k-J). Let χ_k denote the J-dimensional space spanned by {x(k), x(k-1),...x(k-J+1)}. Then we may write

$$\hat{x}(k-J \mid \chi_k) = \sum_{n=1}^{J} g_n\, x(k-n+1) \tag{3-45}$$

to represent a linear prediction of the sample x(k-J), where $\boldsymbol{g}$ is the J-dimensional vector of tap weights which are also assumed to be optimized in the mean-square sense. In the case of backward prediction, the desired response is given by d(k)=x(k-J) and the backward error equals the difference between the actual sample value x(k-J) and its predicted value $\hat{x}(k-J \mid \chi_k)$. The backward prediction error is given by $b_J(k)$ where

$$b_J(k) = x(k-J) - \hat{x}(k-J \mid \chi_k) \tag{3-46}$$

and, from equation (3-45), we may write

$$b_J(k) = x(k-J) - \sum_{n=1}^{J} g_n\, x(k-n+1) \tag{3-47}$$

Defining the tap weights of the backward prediction-error filter in terms of the corresponding backward predictor weights as

$$c_{Jn} = \begin{cases} -g_{k+1} & n=0,1,\ldots J-1 \\ 1 & n=J \end{cases} \tag{3-48}$$

we may write equation (3-47) as

$$b_J(k) = \sum_{n=0}^{J} c_{Jn}\, x(k-n) \tag{3-49}$$

where the backward predictor and backward prediction-error filter are depicted in figures 42 and 43, respectively.

3) <u>The Solution and Relationship of Prediction Weight Vectors:</u> Assuming stationarity, the correlation matrix for both the forward and backward processes may be expressed as

$$\boldsymbol{R} = E[\boldsymbol{x}\boldsymbol{x}^T] \tag{3-50}$$

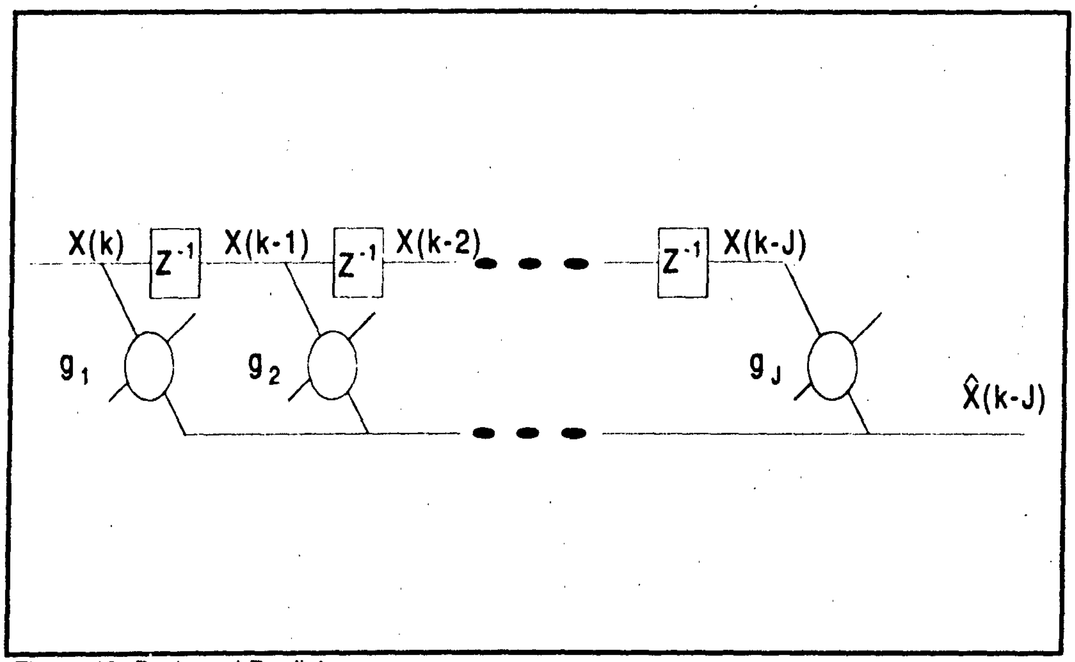

Figure 42 Backward Predictor

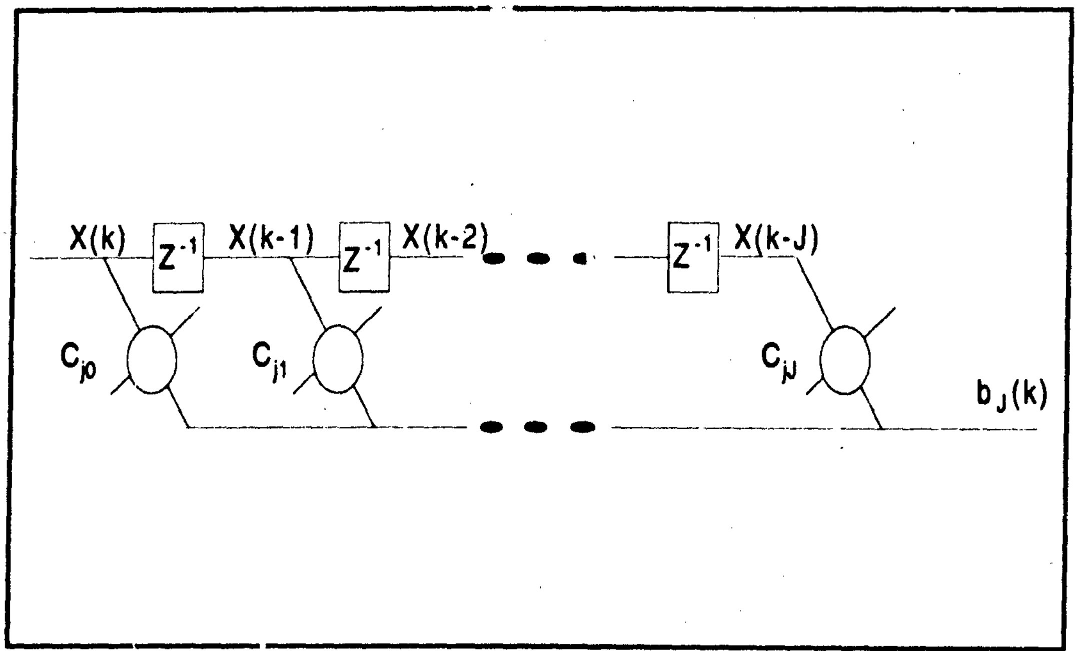

Figure 43 Backward Prediction Error Filter

where $\boldsymbol{x}$ is the J-dimensional vector composed of the observation process samples. The cross-correlation vectors may be formulated for the forward and backward predictors as $\boldsymbol{r}^f$ and $\boldsymbol{r}^b$, respectively:

$$\boldsymbol{r}^f = \boldsymbol{E}\begin{bmatrix} x(k-1)\,x(k) \\ x(k-2)\,x(k) \\ \cdot \\ \cdot \\ x(k-J)\,x(k) \end{bmatrix} = \begin{bmatrix} r(-1) \\ r(-2) \\ \cdot \\ \cdot \\ r(-J) \end{bmatrix} \tag{3-51}$$

$$\boldsymbol{r}^b = \boldsymbol{E}\begin{bmatrix} x(k)\,x(k-J) \\ x(k-1)\,x(k-J) \\ \cdot \\ \cdot \\ x(k-J+1)\,x(k-J) \end{bmatrix} = \begin{bmatrix} r(J) \\ r(J-1) \\ \cdot \\ \cdot \\ r(1) \end{bmatrix} \tag{3-52}$$

The solutions to the forward and backward linear prediction problems are given by the Wiener-Hopf equation as

$$\boldsymbol{W}_o = \boldsymbol{R}^{-1}\boldsymbol{r}^f \tag{3-53}$$

$$\boldsymbol{g} = \boldsymbol{R}^{-1}\boldsymbol{r}^b \tag{3-54}$$

Denoting the vector formed by reversing the elements of the vector $\boldsymbol{g}$ as $\boldsymbol{g}^B$, we note from equations (3-51), (3-52), (3-53) and (3-54) that

$$\boldsymbol{g}^B = \boldsymbol{W}_o \tag{3-55}$$

and the ensemble error variances for the forward and backward predictors are identical since $(\boldsymbol{r}^b)^T\boldsymbol{g} = (\boldsymbol{r}^f)^T\boldsymbol{g}^B$ and

$$\boldsymbol{E}[|b_J|^2] = r(0) - (\boldsymbol{r}^b)^T\boldsymbol{g} = r(0) - (\boldsymbol{r}^f)^T\boldsymbol{g}^B = r(0) - (\boldsymbol{r}^f)^T\boldsymbol{W}_o = \boldsymbol{E}[|f_J|^2] \tag{3-56}$$

C. The Lattice Filter Structure

We know extend the results of the previous section to the vector case of interest and the consider the GSC data vector $\boldsymbol{X}_s$ derived in the last chapter. The lattice filter solves the prediction problem by finding orthogonal bases for the subspaces χ_{k-1} and χ_k. The Wiener-Hopf solution for the TDL structure optimum filter derived in chapter I determined the weighting coefficients associated with each basis vector of the subspace of past observations such that the prediction error was orthogonal with respect to that subspace. The lattice structure differs in that one first constructs an orthogonal basis of the subspace of past observations, and then projects the vector $X_S(k)$ successively onto the orthogonal basis vectors. Consequently, since the projections are formed onto the orthogonal basis vectors, successive stages of the lattice are decoupled. Hence, one may increase the order of the filter by adding additional stages to the lattice while the original lower order predictor remains optimal in the expanded structure. Thus, it is no longer necessary to use the fixed-order Wiener-Hopf equation to determine the optimal filter coefficients.

1) The Optimal Lattice Filter: The lattice filter structure is derived by employing a recursive formulation of the Gram-Schmidt orthogonalization procedure for orthogonal projections. Following Strobach [26], we denote the inner product of two vectors A and B as $<A , B>$. Let the (K-1) dimensional vector at stage n, F_n, be the complement of the orthogonal projection of a vector X_s onto the subspace χ_n denoted

$$F_n = X_s < \chi_{n-1} , X_s(n) > = X_s < \chi_n > \tag{3-57}$$

with the property

$$F_n^T X_s(n) = 0 \qquad 1 \le n \le J \tag{3-58}$$

then the orthogonal complement F_n of the n-th order projection can be constructed order-recursively from the orthogonal complement F_{n-1} of the (n-1)-th order projection using the recursion formula

$$X_s < \chi_{n-1}, X_s(n) > = X_s < \chi_{n-2}, X_s(n-1) > + K_n X_s(n) < \chi_{n-2}, X_s(n-1) >. \tag{3-59}$$

or, equivalently,

$$F_n = F_{n-1} + K_n X_s(n) < \chi_{n-1} > \tag{3-60}$$

where the (K-1) x (K-1) matrix K_n is given by

$$K_n = (X_s(n) < \chi_{n-1} > X_s^T(n) < \chi_{n-1} >)^{-1} X_s(n) < \chi_{n-1} > X_s^T < \chi_{n-1} > \tag{3-61}$$

This can be proved by considering $F_n(K_n)$ as a vector constructed by the linear combination

$$F_n(K_n) = X_s < \chi_{n-1} > + K_n X_s(n) < \chi_{n-1} > \tag{3-62}$$

Then, F_n is orthogonal with respect to the subspace χ_{n-1} extended by the vector $X_s(n)$ and, equivalently, is orthogonal to χ_{n-1} and $X_s(n) < \chi_{n-1} >$ if and only if the parameter K_n is adjusted such that the Euclidean norm of $F_n(K_n)$ attains a minimum. This follows directly from the geometrical considerations of the Wiener-Hopf solution and leads to a least-squares determination of K_n via the approach

$$\min_{K_n} F_n^T F_n \tag{3-63}$$

Substituting equation (3-62) into (3-63), taking the gradient and setting it equal to zero gives

$$\frac{\partial F_n}{\partial K_n} = 2 X_s^T < \chi_{n-1} > X_s(n) < \chi_{n-1} > + 2 K_n X_s^T(n) < \chi_{n-1} > X_s(n) < \chi_{n-1} > = 0 \tag{3-64}$$

which yields equation (3-61) and determines K_n such that F_n is orthogonal with respect to the extended subspace spanned by χ_{n-1} and $X_s(n) < \chi_{n-1} >$, or equivalently, with respect to the subspace χ_n.

The vector $X_s(k)$ may be projected successively onto the components of the subspace of past observations as follows:

$$
\begin{gathered}
F_0(k) = X_s(k) \\
F_1(k) = X_s(k) \langle X_s(k-1) \rangle \\
F_2(k) = X_s(k) \langle X_s(k-1)\, ,\, X_s(k-2) \rangle \\
\vdots \\
F_J(k) = X_s(k) \langle X_s(k-1)\, ,\, X_s(k-2)\, ,\, \ldots, X_s(k-J) \rangle
\end{gathered}
\tag{3-65}
$$

Similarly, we may successively construct an orthogonal basis of the same subspace as

$$
\begin{gathered}
B_0(k) = X_s(k) \\
B_1(k) = X_s(k-1) \langle X_s(k) \rangle = X_s(k-1) \langle B_0(k) \rangle \\
B_2(k) = X_s(k-2) \langle X_s(k)\, ,\, X_s(k-1) \rangle = X_s(k-2) \langle B_0(k)\, ,\, B_1(k) \rangle \\
\vdots \\
B_J(k) = X_s(k-J) \langle X_s(k)\, ,\, X_s(k-1)\, ,\, \ldots, X_s(k-J+1) \rangle
\end{gathered}
\tag{3-66}
$$

These nonrecursive decompositions are a consequence of applying the Gram-Schmidt orthogonalization procedure. They can be made recursive by applying equations (3-59), (3-60) and (3-61) to the last terms in equations (3-65) and (3-66):

$$F_n(k) = X_s(k) \langle X_s(k-1), \ldots, X_s(k-n+1) \rangle + K_n^f(k) X_s(k-n) \langle X_s(k-1), \ldots, X_s(k-n+1) \rangle \tag{3-67}$$

$$B_n(k) = X_s(k-n) \langle X_s(k-1), \ldots, X_s(k-n+1) \rangle + K_n^b(k) X_s(k) \langle X_s(k-1), \ldots, X_s(k-n+1) \rangle \tag{3-68}$$

Equations (3-67) and (3-68) may be expressed in terms of the orthogonalized vectors $F_{n-1}(k)$ and $B_{n-1}(k)$ at stage n-1, establishing the recursive laws

$$F_n(k) = F_{n-1}(k) + K_n^f(k) B_{n-1}(k-1) \tag{3-69}$$

$$B_n(k) = B_{n-1}(k-1) + K_n^b(k) F_{n-1}(k) \tag{3-70}$$

with the initial condition

$$F_0(k) = B_0(k) = X_s(k) \tag{3-71}$$

From equation (3-61), the matrices $K_n^f(k)$ and $K_n^b(k)$, termed the forward and backward reflection coefficient matrices, respectively, can be defined as

$$K_n^f(k) = -\left(B_{n-1}(k-1)\, B_{n-1}^T(k-1)\right)^{-1} \left(F_{n-1}(k)\, B_{n-1}^T(k-1)\right)^T \tag{3-72}$$

$$K_n^b(k) = -\left(F_{n-1}(k)\, F_{n-1}^T(k)\right)^{-1} \left(F_{n-1}(k)\, B_{n-1}^T(k-1)\right) \tag{3-73}$$

It is noted that equations (3-72) and (3-73) are identical to those presented by Griffiths [39, equations 11a and 11b].

The preceding derivation was concerned solely with the optimal predictors. The symmetry of the autocorrelation function showed that the optimal backward predictor coefficients are the mirror image of the optimal forward prediction coefficients and that the backward and forward prediction errors have the same norm (the lengths are the same, but the error signals themselves are not). The backward prediction errors are orthogonal to each other and time-shifted versions of both the forward and backward prediction errors are orthogonal. Thus, the generation of a sequence of backward prediction errors by a lattice filter consisting of n-stages is equivalent to a Gram-Schmidt orthogonalization process applied recursively to a corresponding sequence of input samples. Haykin [27, pp. 173-178] shows that this transformation of the tap-input vector $X_s(k)$ into the backward prediction-error vector $B_J(k)$ can be accomplished through the premultiplication of the input vector by a lower triangular matrix L (where, from the preceding section, $Q = L$) with 1's along the diagonal. The non-zero elements along each row of the matrix L are defined by the tap weights of the backward prediction-error filter whose order corresponds to the position of the pertinent row in the matrix. This matrix L, where we may explicitly write

$$B_J(k) = L\, X_s(k) \tag{3-74}$$

is nonsingular and hence, there is a one-to-one correspondence between the input vector and the backward prediction-error vector. It is again emphasized that these properties are applicable only to the optimal predictors.

2) <u>The Adaptive Lattice Structure GSC Form Processor:</u> The GSC form array presented in Chapter II will now be examined with a data dependent adaptive lattice filter structure replacing the TDL processor. This section follows the work of Griffiths [37,38,39] and Lee, Chang, Cha, Kim and Youn [48]. The all zero fixed coefficient lattice filter has the same transfer function as the fixed coefficient TDL filter, and the scalar filter coefficient conversions are presented clearly in Oppenheim and Schafer [25]. The lattice filter, as derived above, achieves the transfer function through an orthogonalization procedure. This property of the lattice structure will be shown to be capable of providing desirable convergence properties in the adaptive multichannel structure. The recursive form of the lattice structure GSC is presented in figure 44.

The basic adaptive lattice stage represented by each box in figure 44 is shown in figure 45. The delayed observation data sequence $\boldsymbol{X}(k-l)$ is transformed into the orthogonal sequence $\boldsymbol{B}_l(k)$ through the Gram-Schmidt type relations described in equations (3-69), (3-70) and (3-71) and, with a slight change of notation, repeated below:

$$\boldsymbol{B}_o(k) = \boldsymbol{F}_o(k) = \boldsymbol{X}_s(k) \qquad (3\text{-}75)$$

$$\boldsymbol{B}_l(k) = \boldsymbol{B}_{l-1}(k-1) - \boldsymbol{W}_l^b(k)\,\boldsymbol{F}_{l-1}(k) \qquad (3\text{-}76)$$

$$\boldsymbol{F}_l(k) = \boldsymbol{F}_{l-1}(k) - \boldsymbol{W}_l^f(k)\,\boldsymbol{B}_{l-1}(k-1) \qquad (3\text{-}77)$$

These order update equations relate the higher order forward and backward prediction errors to lower order prediction errors. The signal $\boldsymbol{B}_l(k)$ is the backward residual at stage l, and $\boldsymbol{F}_l(k)$ is the forward residual at stage l. Both of the residual vectors at time k are of dimension $(K\text{-}1) \times 1$. The backward and forward reflection coefficient matrices $\boldsymbol{W}_l^b(k)$ and $\boldsymbol{W}_l^f(k)$ are of dimension $(K\text{-}1) \times (K\text{-}1)$ and are commonly termed partial correlation (PARCOR) coefficients. The residual vectors in equations (3-76) and (3-77) are

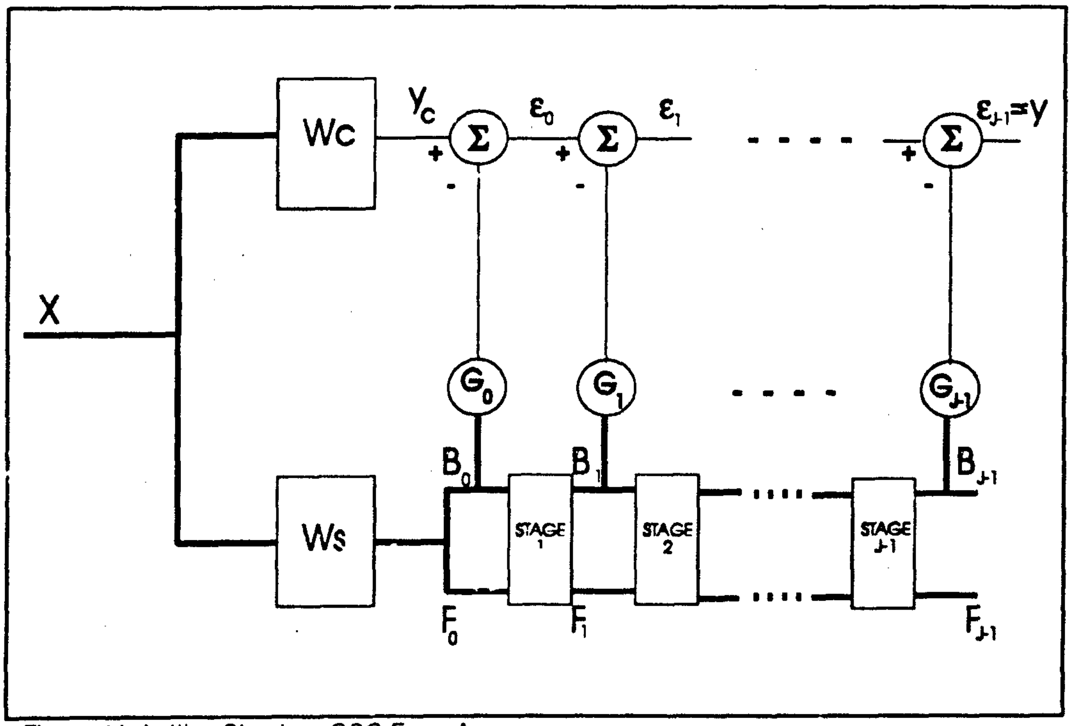

Figure 44 Lattice Structure GSC Form Array

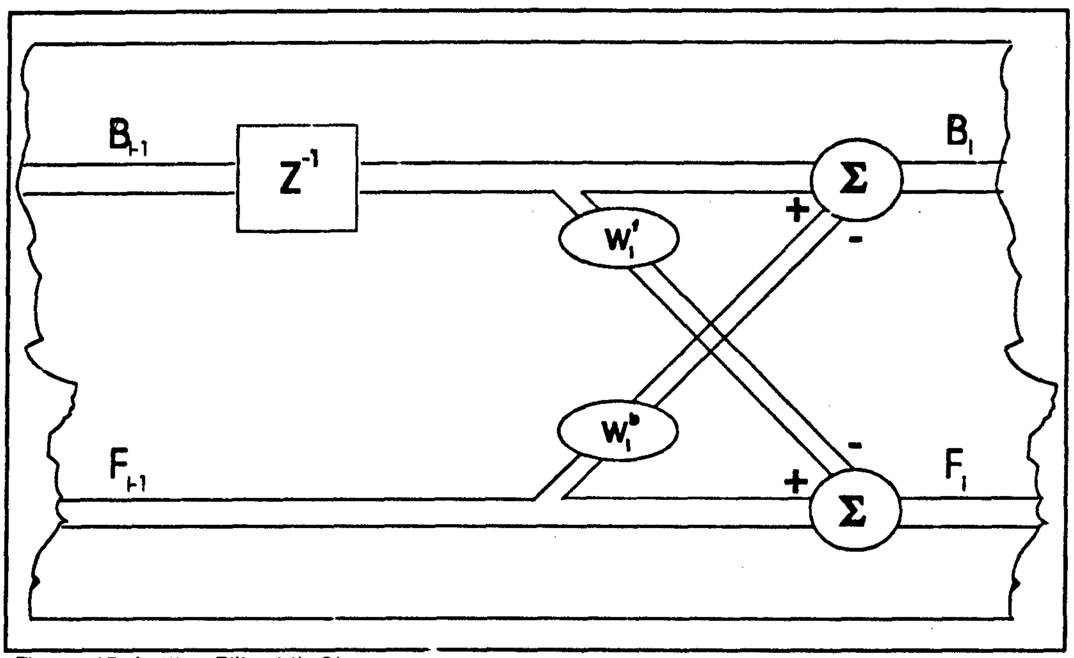

Figure 45 Lattice Filter l-th Stage

recursively updated through the use of the LMS algorithm to minimize their mean squared norm value:

$$W_l^{f^T}(k+1) = W_l^{f^T}(k) + \mu_l^f(k)\, B_{l-1}(k-1)\, F_l^T(k) \tag{3-78}$$

$$W_l^{b^T}(k+1) = W_l^{b^T}(k) + \mu_l^b(k)\, F_{l-1}(k)\, B_l^T(k) \tag{3-79}$$

The optimal PARCOR coefficients are independent of the filter order, so that the PARCOR values in any one stage do not depend on those of other stages.

The Gram-Schmidt type of orthogonalization which the lattice filter stages form may increase the speed of adaptation in subsequent stages. The residual becomes increasingly white as the order of the filter increases. The backward residuals from stage to stage are orthogonal after the PARCOR coefficients converge, resulting in the aforementioned overall convergence rate increase.

Consider the lattice filter structure implementation of the GSC shown in figure 44. The J coefficient vectors $G_l(k)$ are of dimension (K-1) and utilize the LMS algorithm to minimize the mean squared value of the l-th stage error signal $\varepsilon_l(k)$:

$$\varepsilon_o(k) = y_c(k) - G_o^T(k)\, B_o(k) \tag{3-80}$$

$$\varepsilon_l(k) = \varepsilon_{l-1}(k) - G_l^T(k)\, B_l(k) \tag{3-81}$$

$$G_l(k+1) = G_l(k) + \mu_l^g(k)\, \varepsilon_l(k)\, B_l(k) \tag{3-82}$$

The time-varying step size gains are normalized to the input signal variance, and are diagonal matrices given by

$$\mu_l^g(i,j,k) = \begin{cases} \dfrac{1-\beta}{\beta\, \sigma_{B_l}(i,k) + \dfrac{(1-\beta)}{K-1} B_{l_i}^2(k)} & i=j \\ 0 & else \end{cases} \tag{3-83}$$

$$\mu_l^f(i\ i,k) = \begin{cases} \dfrac{1-\beta}{\beta\,\sigma_{B_{l-1}}(i,k-1) + \dfrac{(1-\beta)}{K-1} B_{(l-1)_i}^2(k-1)} & i=j \\ 0 & else \end{cases} \tag{3-84}$$

$$\mu_l^b(i,j,k) = \begin{cases} \dfrac{1-\beta}{\beta\,\sigma_{F_{l-1}}(i,k) + \dfrac{(1-\beta)}{K-1} F_{(l-1)_i}^2(k)} & i=j \\ 0 & else \end{cases} \tag{3-85}$$

Once the PARCOR coefficients have converged, the convergence rate of the lattice structure GSC estimation weight vector $\boldsymbol{G}_l(k)$ is no longer dependent on the eigenvalue spread of the correlation matrix $\boldsymbol{R}_{X_s}$, but upon the (K-1) x (K-1) dimensional correlation matrices of the forward and backward prediction errors. It is this property which provides the capability for a faster convergence rate which can not be achieved with the corresponding TDL processor.

The process of generating the backward prediction error process from the observation process can in general be represented by the matrix operator $\boldsymbol{Q} = \boldsymbol{L}$ where

$$\hat{\boldsymbol{B}}(k) = \boldsymbol{L}(k)\,\hat{\boldsymbol{X}}_s(k) \tag{3-86}$$

where the matrix $\boldsymbol{L}(k)$ is data dependent and changes with each adaptation in accordance with equation (3-76). Once the PARCOR matrices converge, this transformation takes on the lower diagonal form mentioned in section III.C.1 and the input to the conventional weighting structure $\boldsymbol{G}(k) = [\boldsymbol{G}_0(k)\ \boldsymbol{G}_1(k) \ldots \boldsymbol{G}_{J-1}(k)]^T$ is orthogonal due to the realization of a Gram-Schmidt transformation.

D. The Gram-Schmidt Orthogonal Structure

The direct implementation of the Gram-Schmidt algorithm serves as an alternative method of realizing a completely orthogonal signal set to serve as the input to the adaptive processor. The direct Gram-Schmidt orthogonal structure utilizing the LMS algorithm was first developed by Griffiths [39] and utilized a constant step size. This was later modified by Lee et al. [48] to include both a time-varying step size and an escalator realization, where the unit lower triangular transform (ULT) factorization is utilized. Following Griffiths [39], the structure may be realized in the form of figure 39, where the matrix $\boldsymbol{Q}$ is composed of time-varying coefficients and the (K-1)J outputs satisfy

$$E[z_m(k)\, z_n(k)] = 0 \qquad m \neq n \tag{3-87}$$

The matrix $\boldsymbol{Q}$ is lower triangular and composed of elements $q_{i,j}$ which may be represented as

$$\boldsymbol{Q} = \begin{bmatrix} 1 & 0 & . & . & 0 \\ q_{2,1} & 1 & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & 1 & 0 \\ q_{(K-1)J,1} & . & . & q_{(K-1)J,(K-1)J-1} & 1 \end{bmatrix} \tag{3-88}$$

The orthogonalization procedure generates the orthogonal output $Z_m(k)$ via the recursive relationship

$$y_{m,1}(k) = \hat{X}_{s_m}(k) \tag{3-89}$$

$$y_{m,m}(k) = z_m(k)$$

$$z_m(k) = \hat{X}_{s_m}(k) - \sum_{n=1}^{m-1} c_{m,n}(k)\, z_n(k) \qquad 2 \leq m \leq (K-1)J$$

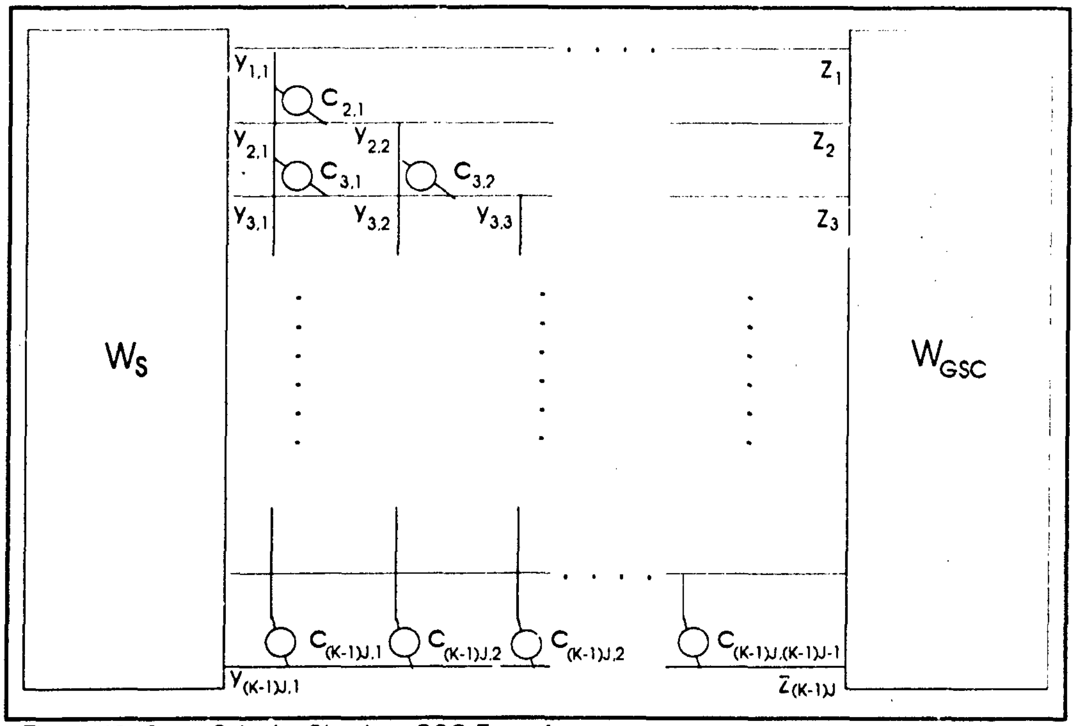

Figure 46 Gram-Schmidt Structure GSC Form Array Processor Lower Path

where the value of $c_{m,n}$ is chosen to minimize the local values $E[y^2_{m,n+1}(k)]$ shown in figure 46. In conjunction with the method of gradient descent, we may write

$$c_{m,n}(k+1) = c_{m,n}(k) + \mu \frac{\partial\, y^2_{i,j+1}(k)}{\partial\, c_{m,n}(k)} \tag{3-90}$$

This result may be achieved through the use of the LMS algorithm to update the adaptive coefficients $c_{m,n}$

$$c_{m,n}(k+1) = c_{m,n}(k) + \mu_n(k)\, y_{m,n+1}(k)\, z_n(k) \tag{3-91}$$

where $\mu_n(k)$ is the time varying step sized formed in the same manner earlier established in this research,

$$\mu_n(k) = \frac{(1-\beta)}{\hat{\sigma}^2_{z_n}(k)} \tag{3-92}$$

The matrix Q in equation (3-88) is then given by $[I + C]^{-1}$, where C is lower triangular with zeros on the diagonal and elements $c_{m,n}$. The form of the Gram-Schmidt structure, presented in figure 46, depicts the generation of the matrix C. The orthogonality in this structure is complete after the convergence of the adaptive coefficients via the LMS algorithm in equation (3-91).

E. Example and Transient Analysis

Thus far in this research, we have derived and examined the TDL structure, DFT and DCT orthogonal transform structure, lattice structure and Gram-Schmidt orthogonal structure GSC form linearly constrained MVDR adaptive array.

This section will be explicitly concerned with evaluating the transient behavior of the adaptive structures under consideration. The ensemble mean-square error for each case will be estimated and compared. The performance as a function of computational complexity will be examined and related to the more computationally expensive least squares techniques, utilizing Sample Matrix Inversion (SMI) [34] as a reference.

The first example to be considered is a continuation of the the simulation from the second chapter. A second example is then generated, where the array to be considered for the simulation is composed of ten linear sensors (K=10) equispaced at half-wavelength. The sampled signals are delayed via an FIR filter of order eight (J=8). The tap spacing in both examples defines a frequency of $f_0 = 1 = \frac{1}{\Delta}$. There is one desired signal, in whose direction the array is assumed to be pre-steered. The look direction sensor noise is omitted from both of these examples.

The first example examines the transient behavior of the array via simulation for the TDL, DFT, DCT, lattice and Gram-Schmidt structures GSC Ws2 form constrained adaptive array. The propagating signal descriptions remain unchanged from table I in the example of Chapter II. The behavior of the structures are characterized by the estimated mean-square error and estimated mean transform domain weight transients. The mean-square error estimate for each structure was formed by averaging the output power of each processor over two hundred independent simulations (consisting of three hundred adaptations each), and the mean weight vector values were similarly averaged. Again, the same observation data was provided to each adaptive filter structure during the independent simulations for consistency.

The graphs in figures 47 and 48 present the ensemble weight vector trajectories and learning curve for the TDL structure. The graphs presented in figures 49 - 50, 51 - 52, 53 - 54 and 55 - 56 depict the analogous results for the DFT, DCT, lattice and Gram-Schmidt structures. The benefit of using of an orthogonal transform is readily apparent by the better mean-square error performance of all such structures compared to the TDL in figure 48. The behavior of the time-varying orthogonal lattice and Gram-Schmidt structures appear nearly identical, as was expected by their derivations. The DCT frequency domain structure performs nearly as well as the time-varying orthogonal structures while the DFT structure's performance is only slightly better than that of the TDL. To better depict the situation, figure 57, 58, 59 and 60 present the TDL, DFT, DCT and lattice structure's performance (dotted lines) versus that of the Gram-Schmidt (solid line), respectively.

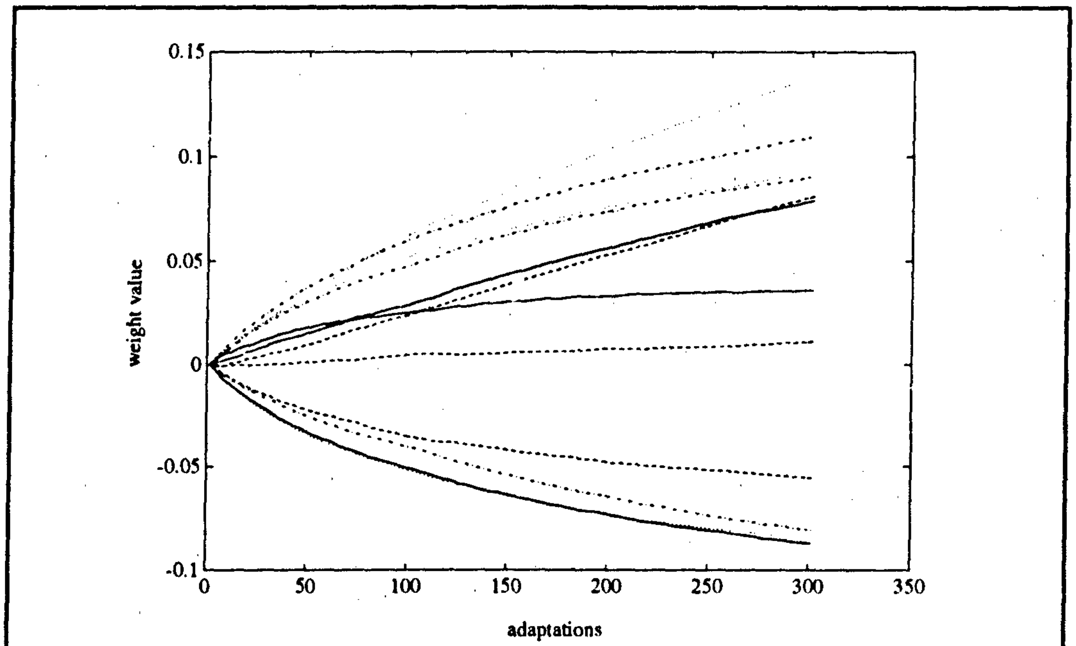

Figure 47 Mean Weight Transient Estimate for TDL GSC

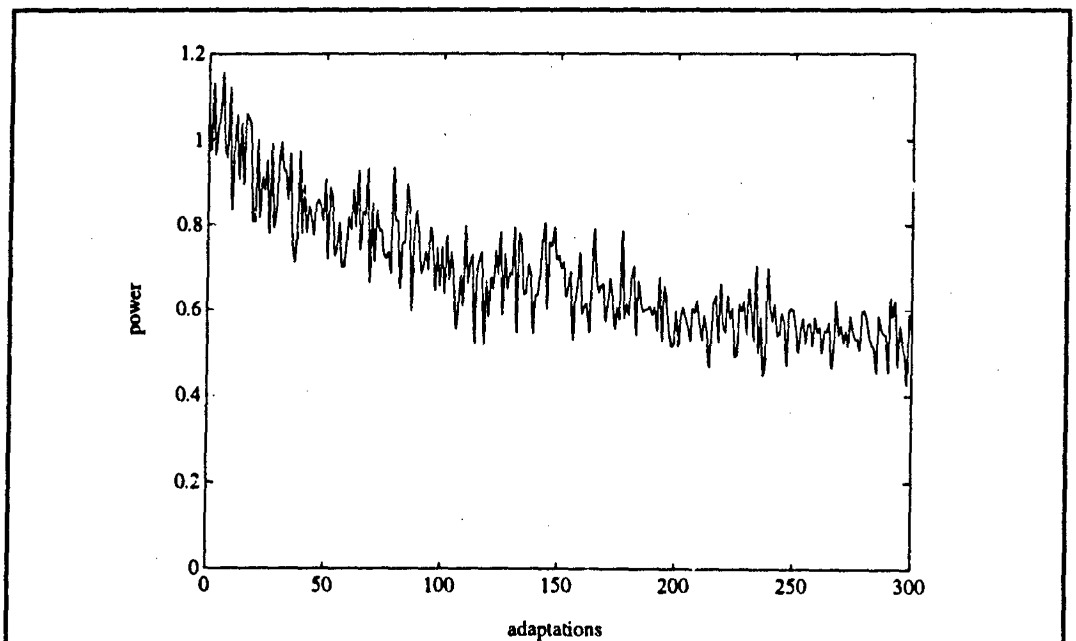

Figure 48 MSE Estimate for TDL GSC

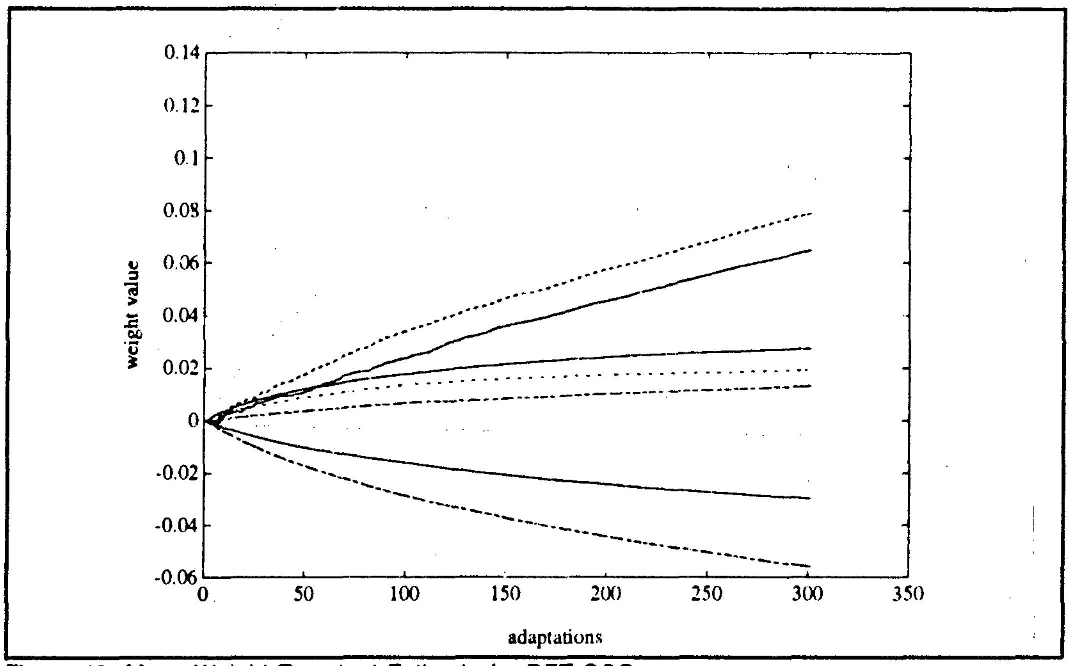

Figure 49 Mean Weight Transient Estimate for DFT GSC

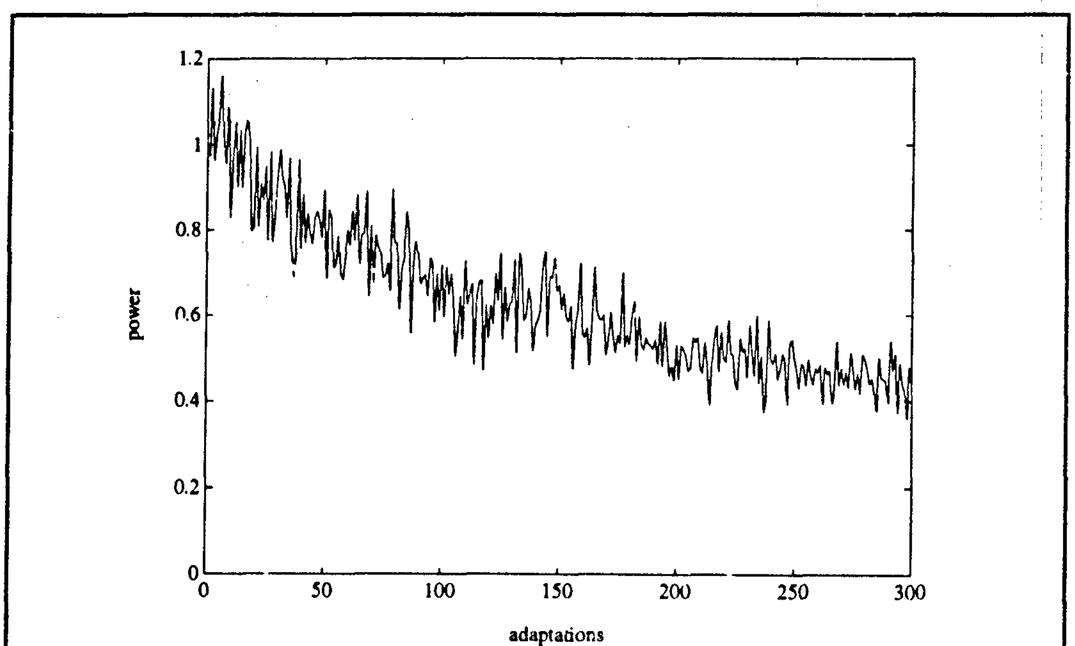

Figure 50 MSE Estimate for DFT GSC

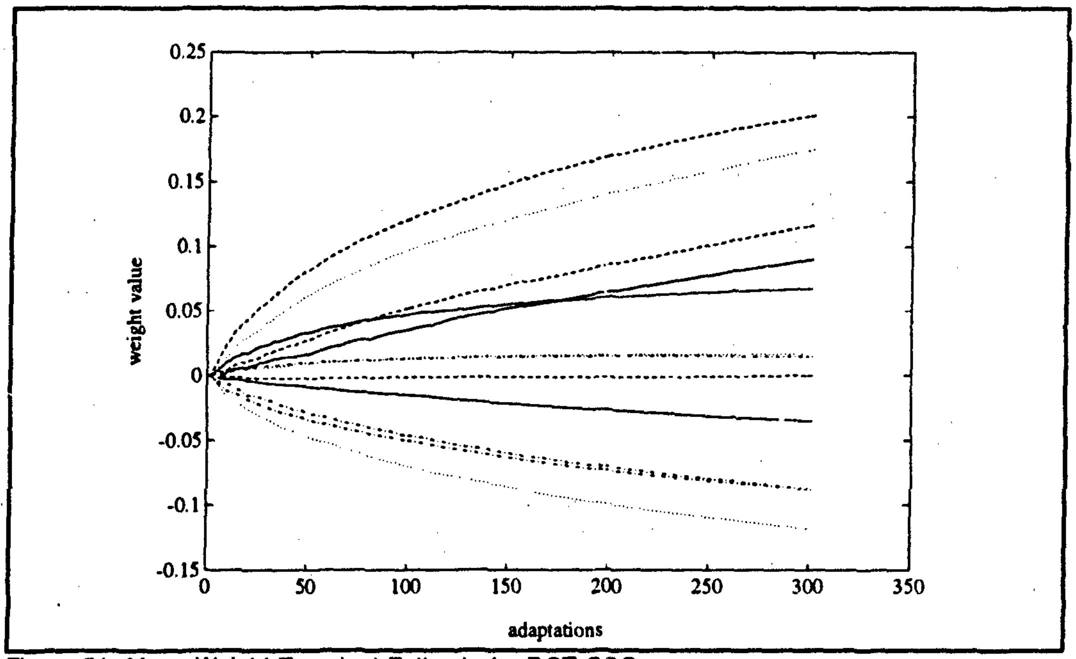

Figure 51 Mean Weight Transient Estimate for DCT GSC

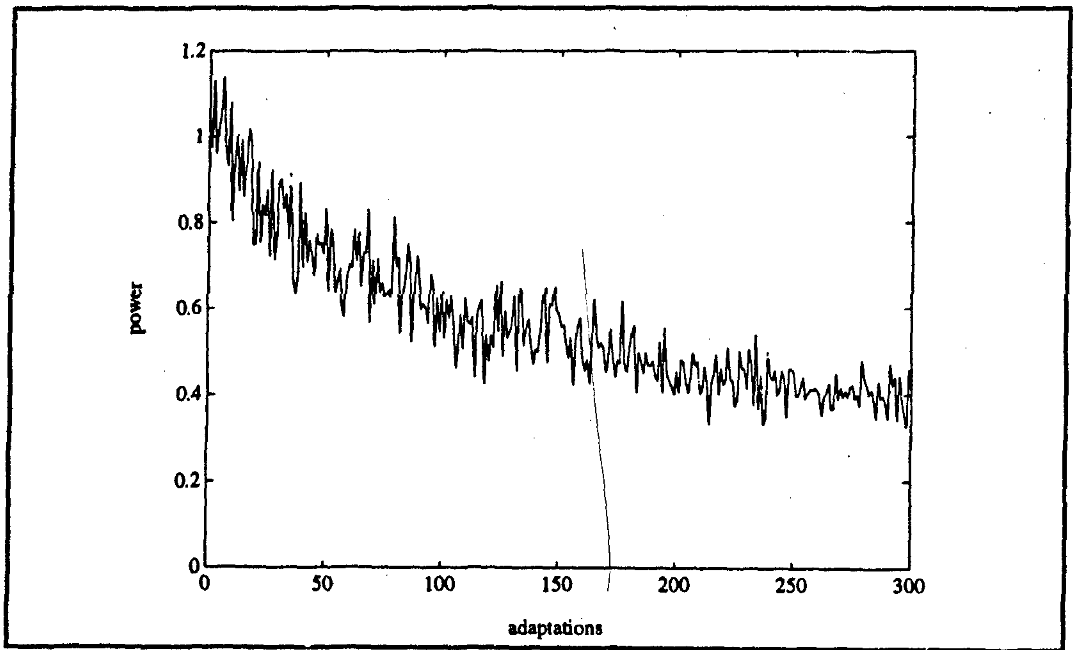

Figure 52 MSE Estimate for DCT GSC

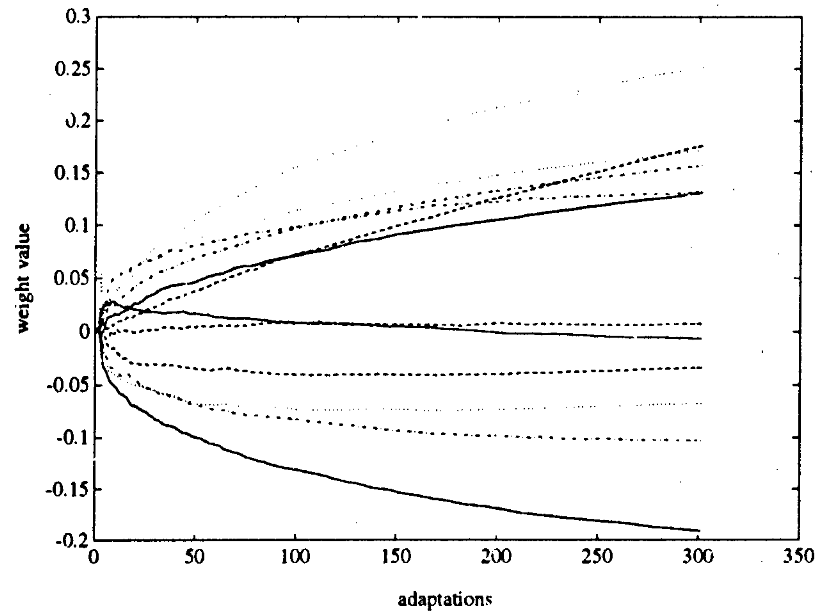

Figure 53 Mean Weight Transient Estimate for Lattice GSC

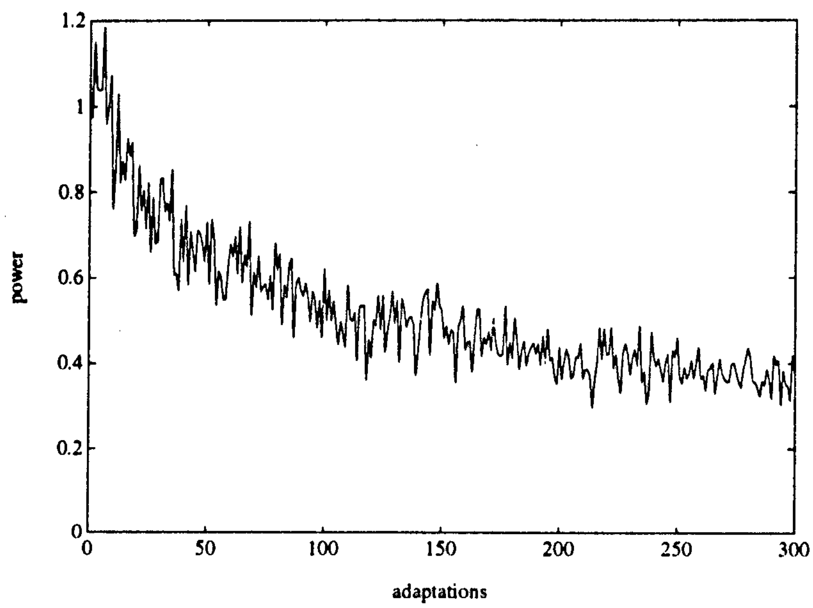

Figure 54 MSE Estimate for Lattice GSC

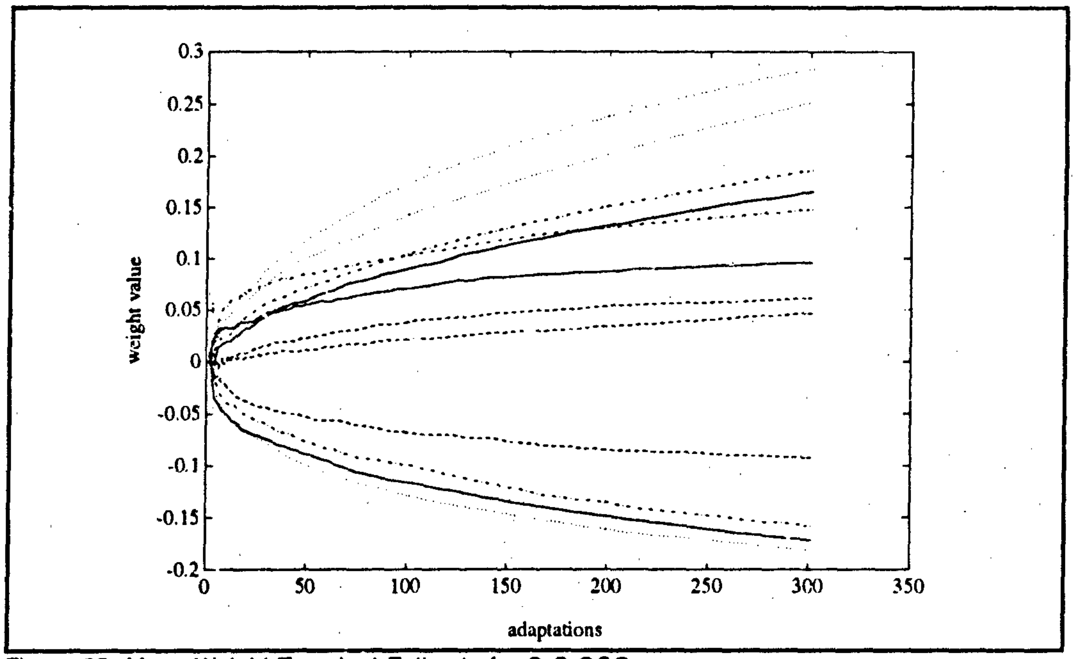

Figure 55 Mean Weight Transient Estimate for G-S GSC

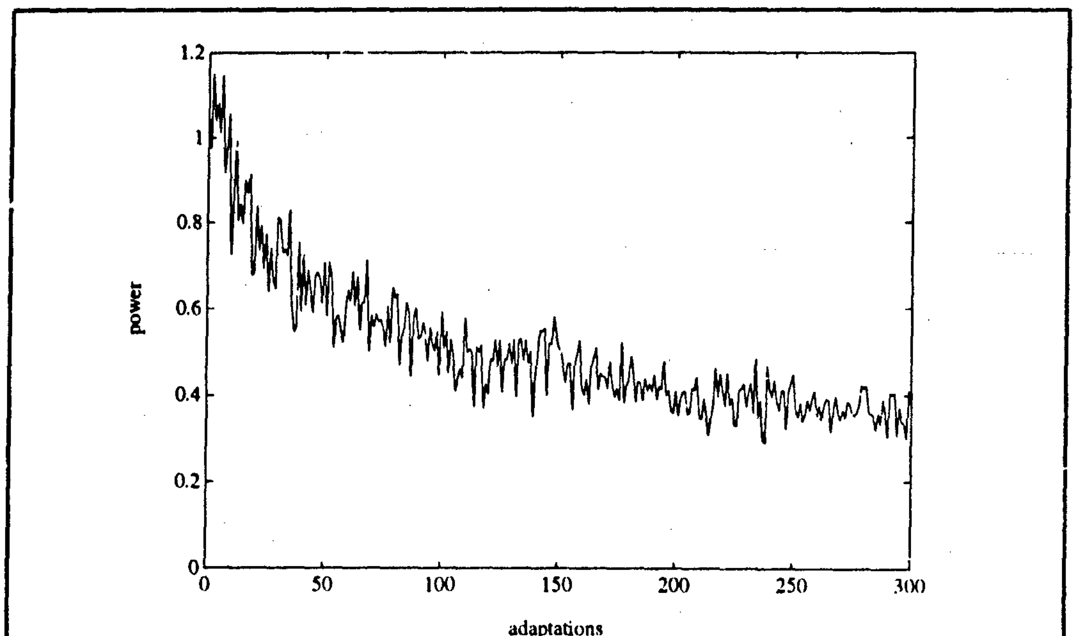

Figure 56 MSE Estimate for G-S GSC

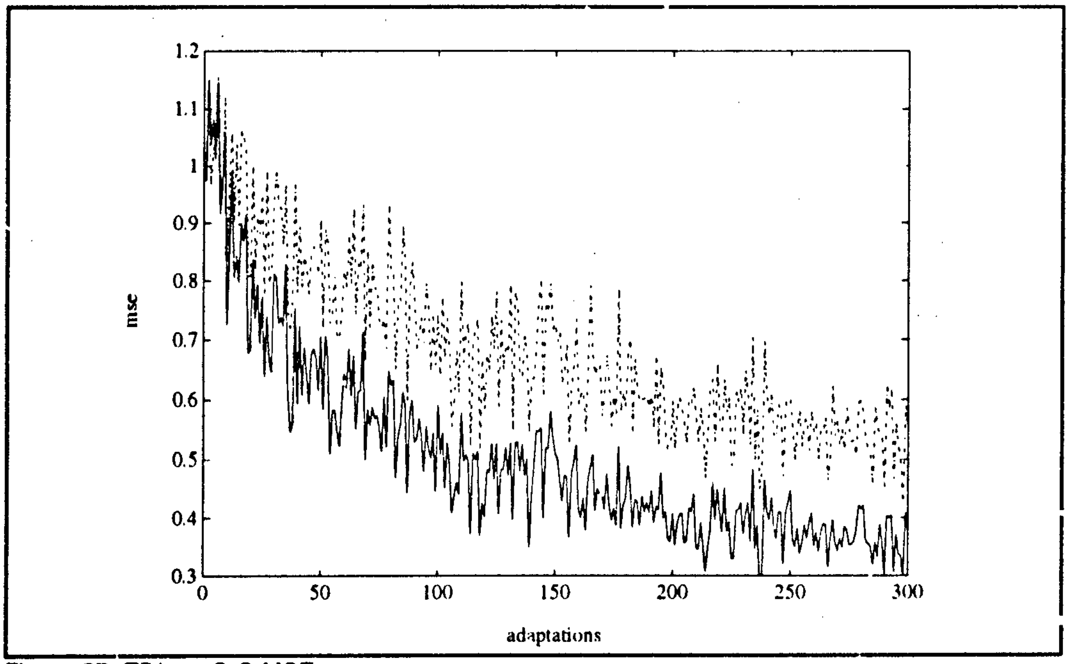

Figure 57 TDL vs G-S MSE

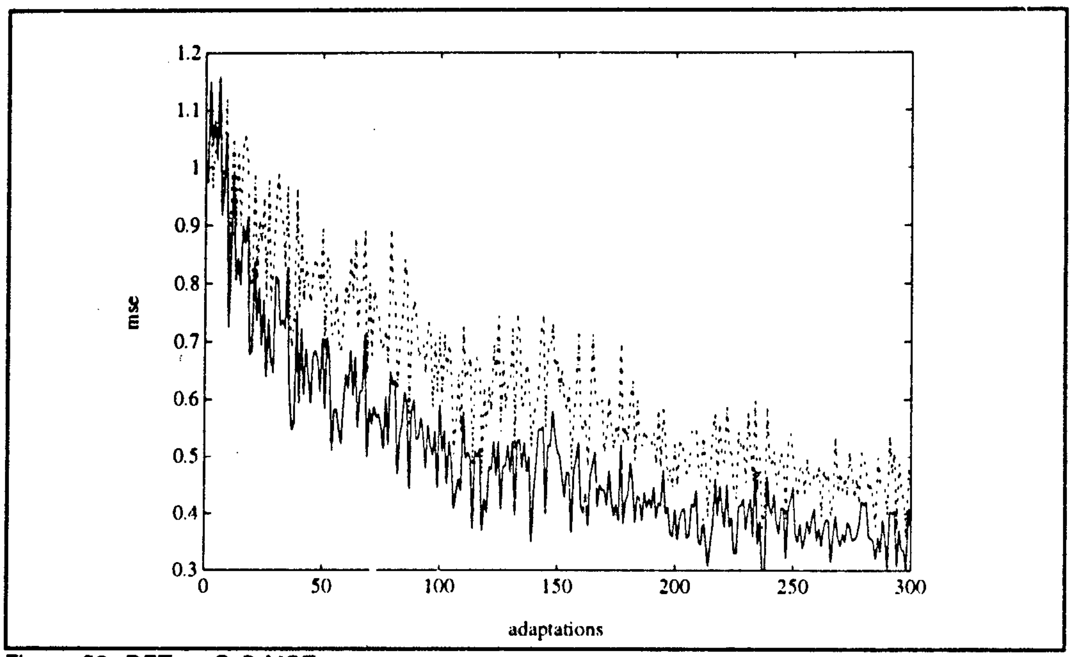

Figure 58 DFT vs G-S MSE

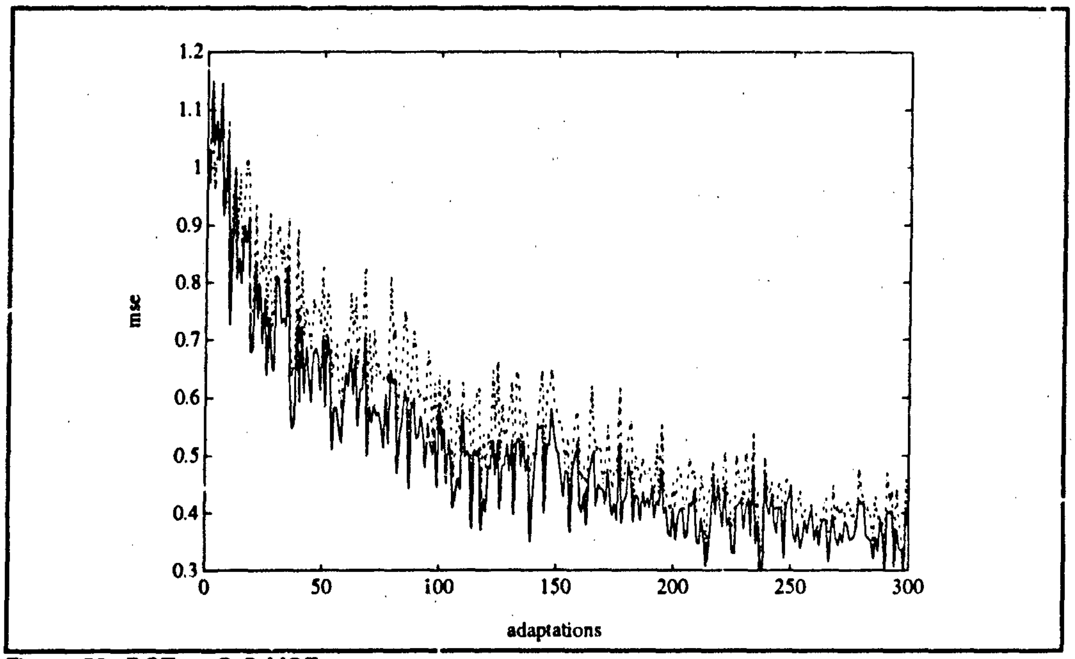

Figure 59 DCT vs G-S MSE

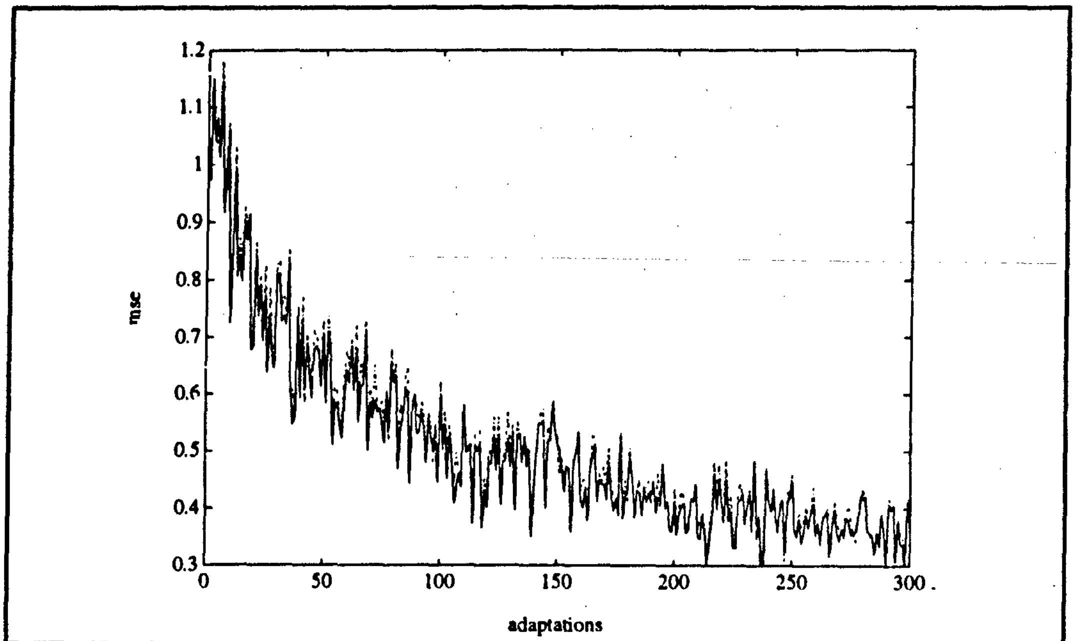

Figure 60 Lattice vs G-S MSE

The normalized time-varying step-sizes for each structure were initialized to the value $\frac{1}{\mathbf{Z}^T(0)\,\mathbf{Z}(0)}$ for both the TDL and frequency domain structures, where $\mathbf{Z}$ is the transform domain data vector (the transform is $\mathbf{Q}=\mathbf{I}$ for the TDL), and to the value $\frac{1}{\mathbf{Z}^2(0)}$ for both the lattice and Gram-Schmidt structures. While this conveniently removes the necessity of choosing an initial step size / power estimate, it does result in the minor MSE overshoot present in the figures.

The mean-square error performance of the second example is now considered, where the array consists of ten sensors and eight taps per sensor. The signal characteristics for the ten sensor array are described in table 2. It is noted that jammer #2 is now centered at the same frequency as the desired signal and that it has a larger bandwidth.

Table 2 Signal Characteristics

SOURCE	Θ	POWER	CENTER FREQUENCY	BANDWIDTH
desired signal	0°	0.001	$0.3 f_0$	0.1
jammer #1	-12.56°	1.0	$0.4 f_0$	0.05
jammer #2	-16.56°	5.0	$0.3 f_0$	0.15
jammer #3	25.58°	10.0	$0.2 f_0$	0.07

The mean-square error performance of the TDL, DFT, DCT, Lattice and Gram-Schmidt structures are depicted in figures 61, 62, 63, 64 and 65, respectively. These results were generated by averaging the mean-square error of two hundred independent simulations consisting of five hundred adaptations each. The performance of the TDL, DFT, DCT and lattice structures (dotted line) versus that of the Gram-Schmidt (solid line) are presented in figures 66, 67, 68 and 69.

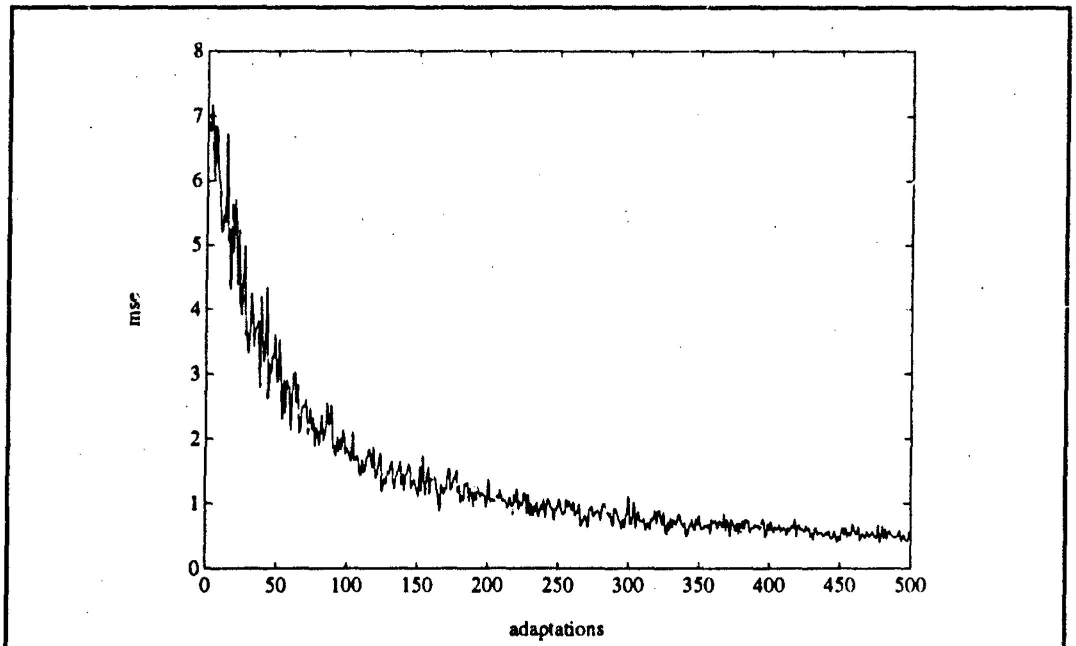

Figure 61 MSE Estimate for TDL GSC

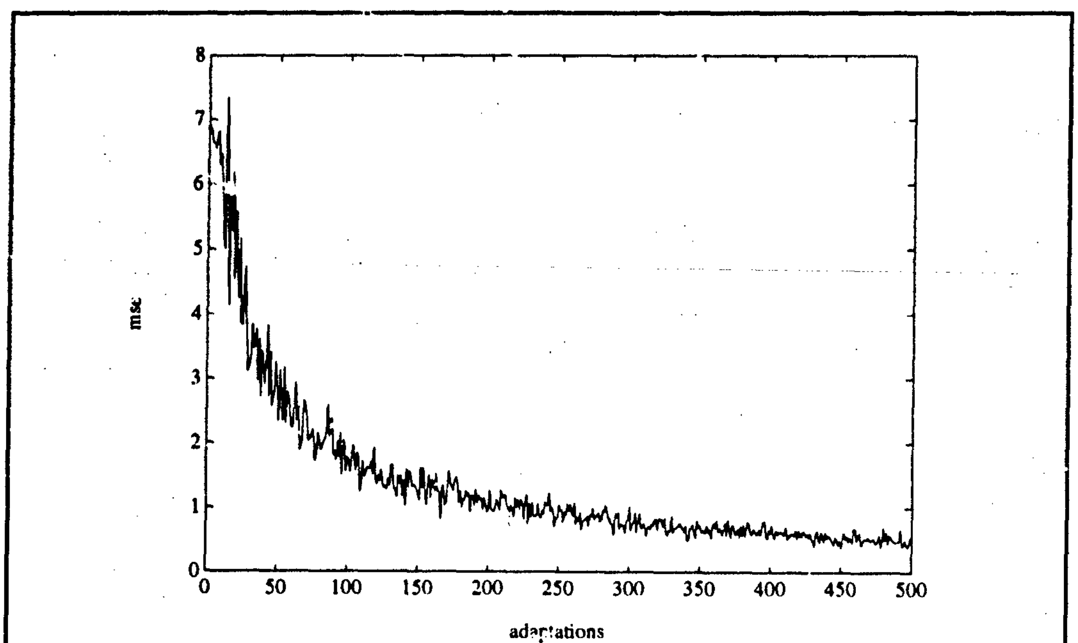

Figure 62 MSE Estimata for DFT GSC

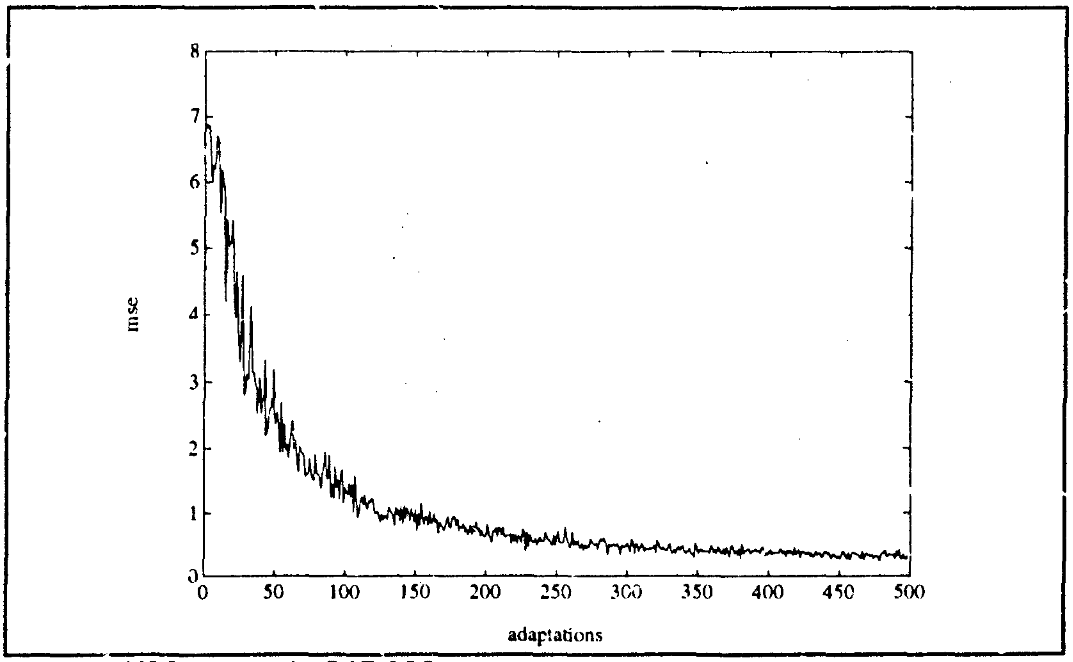

Figure 63 MSE Estimate for DCT GSC

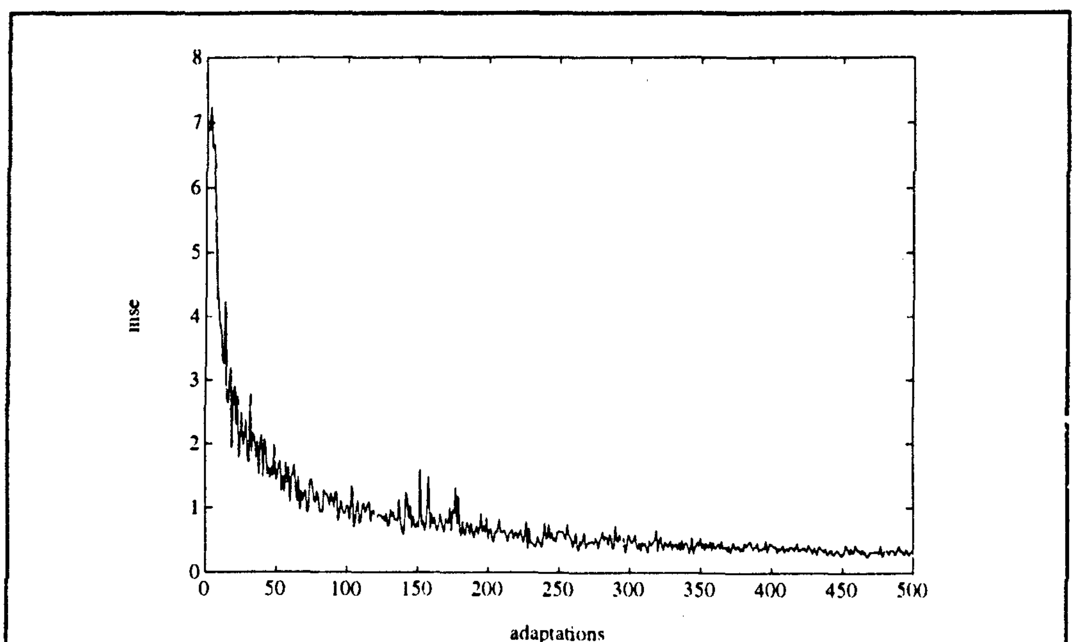

Figure 64 MSE Estimate for Lattice GSC

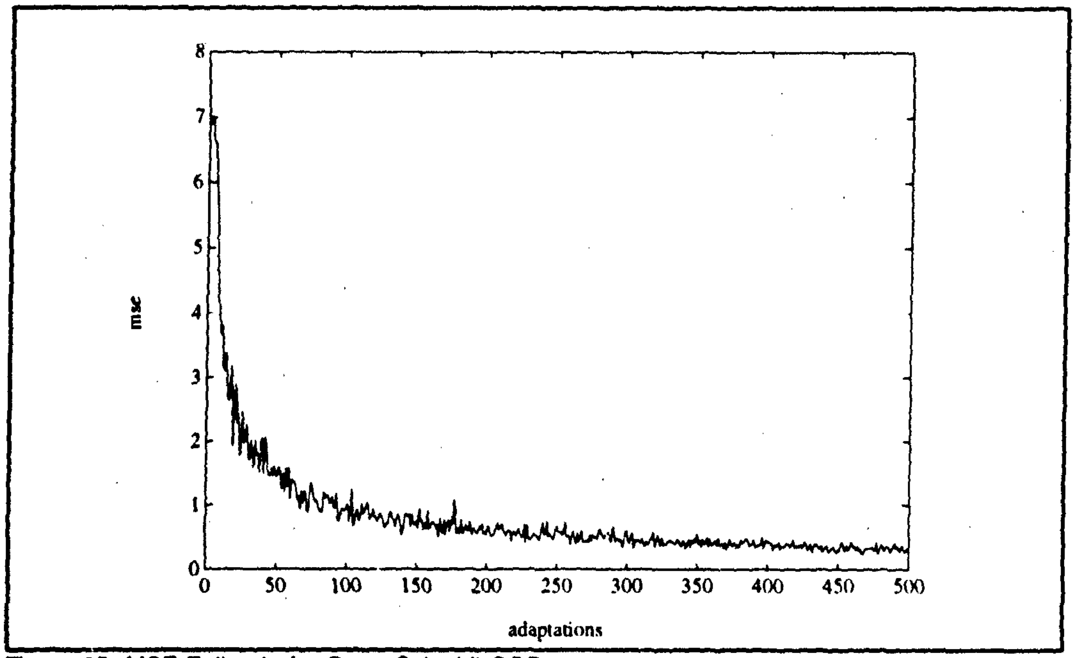

Figure 65 MSE Estimate for Gram-Schmidt GSC

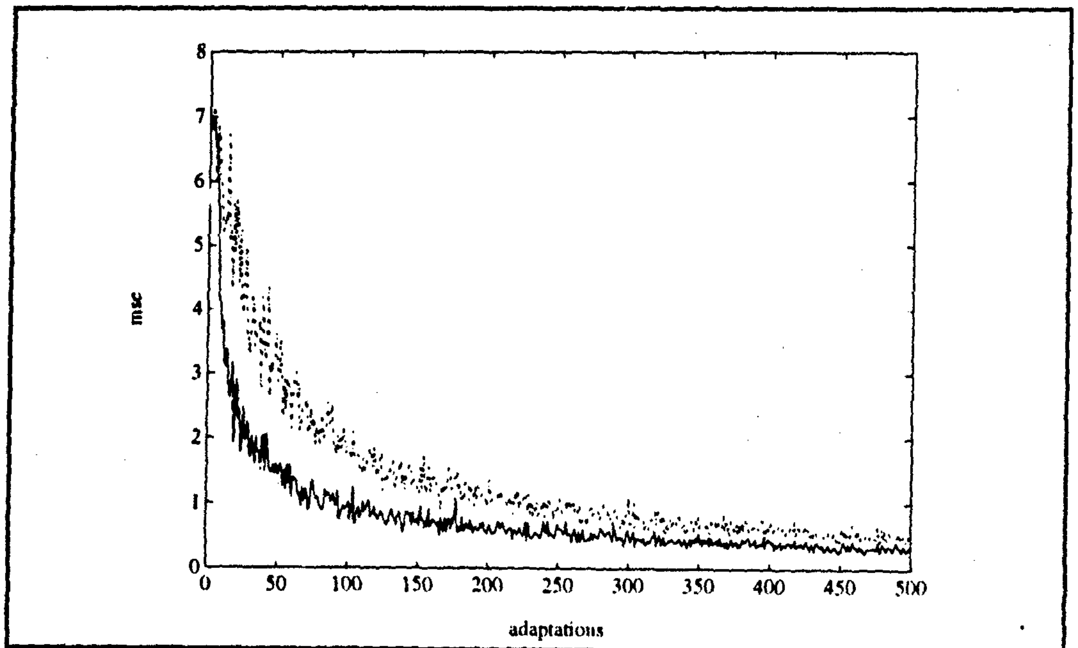

Figure 66 TDL vs G-S MSE

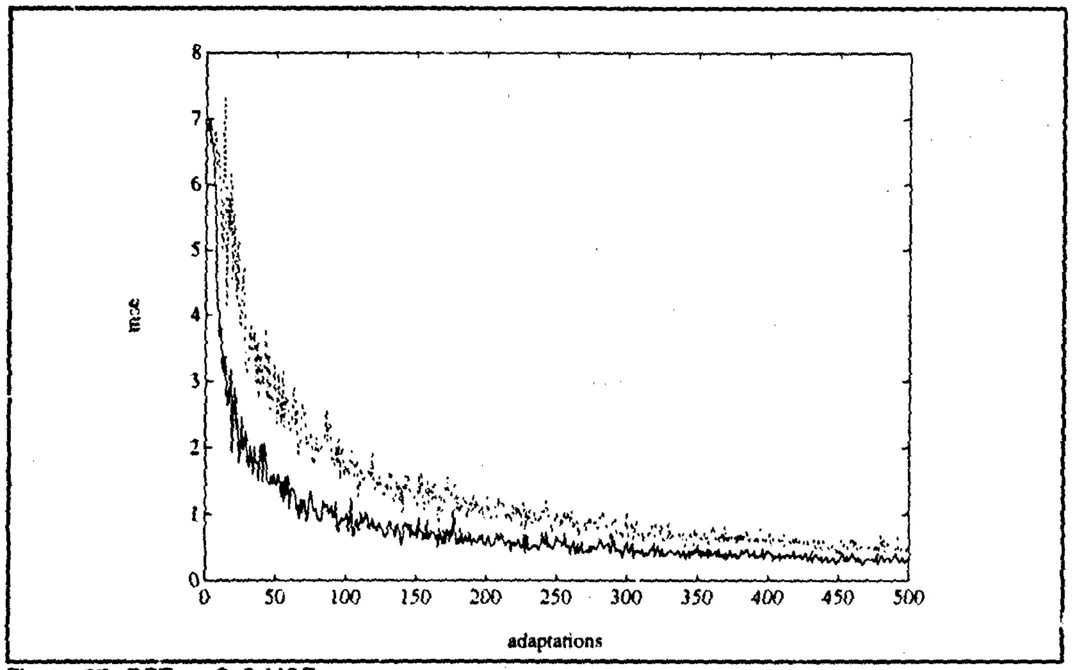

Figure 67 DFT vs G-S MSE

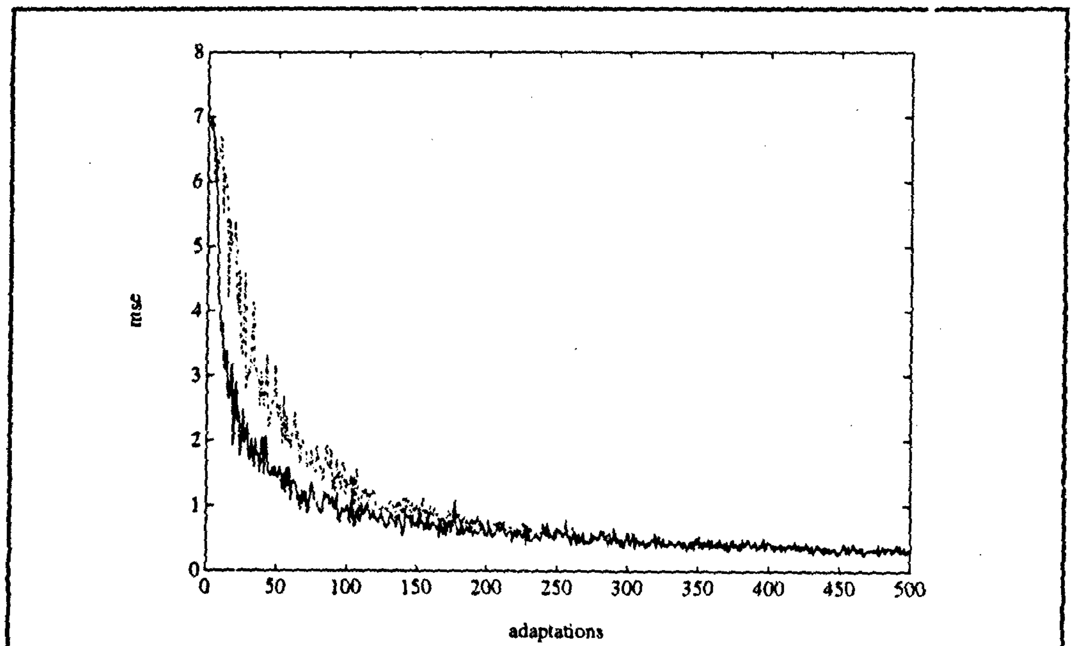

Figure 68 DCT vs G-S MSE

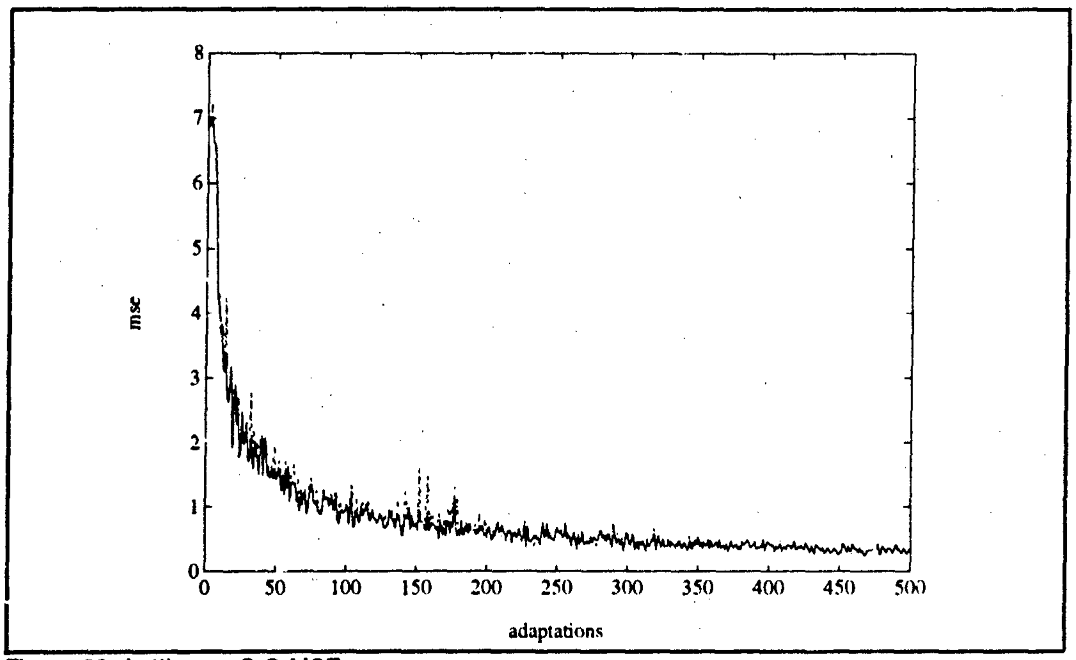

Figure 69 Lattice vs G-S MSE

The results are very similar to those of the last example. The performance of the DFT is not much better than that exhibited by the TDL structure. The DCT structure's performance is considerably better than the TDL. The lattice and the Gram-Schmidt structure's performance is nearly identical. It is noted that the DCT structure achieves the same mean-square error performance as the Gram-Schmidt after approximately two hundred adaptations, while both the DFT and the TDL structures do not attain that level throughout the five hundered adaptations period.

These results may be viewed as presenting a graphical representation of the capability each structure has to provide an uncorrelated signal set to the processor. The DCT and DFT both incorporate data-independent transforms and therefore are not able to change in time with the input process. Thus, given that the LMS step sizes are computed equivalently across the different structures, the performance increase of these transform domain structures over the TDL is limited by the capability of the fixed transform to produce a diagonal correlation matrix. The Gram-Schmidt structure is data dependent, and continuously attempts to provide an orthogonal output based on the input process. The lattice structure maps the input data to an orthogonal basis through independent stages, as described in section III.C. Therefore, after the convergence of the PARCOR coefficients, the lattice structure also produces a completely orthogonal output.

The computational requirements of the different structures are now compared. The measure of computational complexity used will be the number of adaptive coefficients required for the realization of each structure. Since the LMS algorithm is being used for all structures, the number of required operations (multiplications and additions) for each coefficient will be the same, except for the DFT, where the operations are complex. Thus, this measure is reasonable and provides a comparable quantity.

The required number of adaptive coefficients for the structures is presented in table 3. For the array used in the second example, K=10 and J=8 so that the TDL, DFT and DCT

Table 3 Computational Requirements in Terms of Adaptive Coefficients

STRUCTURE	ADAPTIVE COEFFICIENTS
TDL	$(K-1)J$
Frequency Domain	$(K-1)J$
Lattice	$(K-1)J+2(K-1)^2(J-1)$
Gram-Schmidt	$(K-1)J+[(K-1)J][(K-1)J-1]/2$

structures required 72 coefficients, the lattice required 1,206 and the Gram-Schmidt structure required 2,628 adaptive coefficients.

Sample matrix inversion is a weight determination approach, and Gram-Schmidt an algorithm for solving the SMI or least squares problem [55]. Thus, the Gram-Schmidt structure provides a means of realizing the SMI algorithm. Gerlach [56] and Youn [55] recently commented on the convergence behavior of these two algorithms, and both agree that they are numerically equivalent assuming infinite numerical accuracy. Therefore, a direct comparison can be made of the computational requirements (in terms of the number of needed adaptive coefficients) with respect to the least squares algorithms, where the LMS version of the Gram-Schmidt structure presented here is the lower bound in terms of required operations.

IV. CONCLUSIONS

The purpose of this research was to investigate methods of improving the transient response of constrained adaptive array sensor processors while simultaneously satisfying a requirement for limited computational resources. The adaptive processor using the LMS algorithm provides the smallest computational requirements. This processor was developed for the standard TDL structure in chapter I and investigated in terms of the constrained array sensor problem in chapter II. It was shown that the tradeoff for computational simplicity was a transient behavior dependency upon the eigenstructure of the correlation matrix which described the signal and array geometry. This same problem motivated the development of more expensive least squares techniques which led to a solution exhibiting independence of the correlation matrix eigenstructure. Thus, the course of action undertaken in this research was to investigate methods of improving the convergence properties of the LMS array processor in order to gain performance similar to that of the least squares algorithms while maintaining computational simplicity.

The development and utilization of the GSC form linearly constrained MVDR adaptive array sensor in chapter II allowed the adaptive processor to be realized in an unconstrained manner while the overall solution satisfied the constraint. This feature of the GSC provided the motivation to replace the standard TDL structure processor with an orthogonal filter structure. In chapter III, the DFT and DCT frequency domain structures, the lattice structure and a direct implementation of the Gram-Schmidt structure were investigated. The results of the simulations presented in chapter III clearly depict the advantages of orthogonal structures for the linearly constrained MVDR processor.

The fixed transform frequency domain structures have been shown to provide an improvement in the transient behavior of the adaptive array at no increase in the (K-1)J adaptive coefficient computational requirements. However, this structure may still be dependent upon the eigenvalue spread of the correlation matrix. Furthermore, there is no a priori method of knowing which fixed transform will provide the best results for any given observation process. In general, the use of the frequency domain structures will provide some benefit in the arrays dynamic behavior as long as subband normalization is used, and the benefit may be great depending upon the effectiveness of the transform and the interference dynamic range.

The DCT structure processor provided an effective orthogonal transform which, with the use of subband normalization, led to a transient behavior which was extremely close in performance to that of the lattice and the Gram-Schmidt structure. Furthermore, as depicted in table 3, the structure's real valued DCT transform required only one matrix multiplication more computational complexity than the TDL and no additional adaptive coefficients.

The transient performance of the lattice structure is greatly improved over that of the TDL and frequency domain structures. If the increased computational requirements of $(K-1)J+2(K-1)^2(J-1)$ are acceptable, then the lattice structure is the processor of choice. It is noted that this computational complexity is less than that required for least squares methods. The convergence time of PARCOR coefficients in the lattice structure are dependent upon the eigenstructure of independent backward prediction error correlation matrices of smaller-order than the original correlation matrix. Once these coefficients converge, a completely orthogonal process serves as input to the standard LMS adaptive processor. Since the PARCOR coefficients are similarly updated via the LMS algorithm, the computational increase was able to be directly compared in table 2 of the last chapter.

The Gram-Schmidt structure's performance was seen to be the best. It provides an orthogonal output to the adaptive filter via a direct orthogonalization process. The Gram-Schmidt structure, however, suffers from a large computational burden in adapting the [(K-1)J][(K-1)J-1]/2 LMS coefficients required for orthogonalization.

In those applications where the DCT performs as well as in the examples of Chapter III, the DCT frequency domain structure should be the processor of choice for performance versus complexity. The lattice structure provides the best overall performance for cost, providing a nearly equivalent behavior to the Gram-Schmidt structure. Either of these latter two structures will provide a transient performance that is numerically equivalent with the least-squares techniques.

Areas for further research include studying the behavior of these structures when the input process is non-stationary, investigating the utility of an adaptive GSC blocking matrix in order to serve as a pre-processor and provide a more uncorrelated input to the adaptive filter structure (especially for the case of the structure employing a fixed frequency domain transformation), analysis of other orthogonal transforms in the frequency domain GSC structure, and sensitivity analysis via derivative or soft constraints to the LMS algorithm. Furthermore, the analysis of the Short-Time Fourier / cosine Transform and the application of wavelet transforms to the frequency domain structure are recommended in order to provide a capability to track the input process statistics in a manner similar to that achieved by the lattice and Gram-Schmidt structures.

REFERENCES

1. Ma, M.T., "Theory and Application of Antenna Arrays," John Wiley & Sons, Inc., New York, 1974.

2. Haykin, S., "Performance Analysis of a Radar Signal Processing System with Continuous Electronic Array Scanning," Information Science, Volume 13, 1977, pp.201-227.

3. Van Trees, H.L., "Optimum Processing for Passive Sonar Arrays," IEEE Ocean Electronics Symposium, August 29-31, 1966, Honolulu, HI, pp. 41-65.

4. Schultheiss, P.M. and Tuteur, F.B., "Optimum and Suboptimum Detection of Directional Gaussian Signals in an Isotropic Gaussian Noise Field," IEEE Trans. Military Electronics, July-October 1965, pp. 197-208.

5. Capon, J., Greenfield, R.J., and Kolker, R.J., "Multidimensional Maximum Likelihood Processing of a Large Aperture Seismic Array," Proceedings of the IEEE, February 1967, pp. 192-217.

6. Van Atta, L.C., "Electromagnetic Reflection," U.S. Patent 2908002, October 6, 1959.

7. Howells, P.W., "Intermediate Frequency Sidelobe Canceller," U.S. Patent 3202990, August 24, 1965.

8. Applebaum, S.P., "Adaptive Arrays," Syracuse University Research Corp., Rep. SPL TR66-1, August 1966.

9. Widrow, B., "Adaptive Filters I: Fundamentals," Stanford University Electronics Laboratories., Stanford, CA, Rep. SEL-66-126, TR 6764-6, December 1966.

10. Luenberger, D.G., "Optimization by Vector Space Methods," John Wiley & Sons, Inc., 1969.

11. Frost, O.L., "An Algorithm for Linearly Constrained Adaptive Array Processing," Proceedings of the IEEE, Vol 60, No. 8, August, 1972.

12. Applebaum, S.P. and Chapman, D.J., "Adaptive Arrays with Main Beam Constraints," IEEE Trans. Antennas and Propagation, Vol AP-24, September 1976, pp. 650-661.

13. Griffiths, L.J., "An Adaptive Beamformer Which Implements Constraints Using an Auxiliary Array Preprocessor," Aspects of Signal Processing, Part 2, G. Tacconi, ed., D. Reidel Publishing Company, Dordrecht-Holland, 1977, pp. 517-522.

14. Widrow, B., McCool, J.M., Larimore, M.G. and Johnson, C.R., "Stationary and Non-Stationary Learning Characteristics of the LMS Adaptive Filter," Proceedings of the IEEE, Vol 64, No. 8, August 1976, pp. 1151-1161.

15. Griffiths, L.J., "Adaptive Monopulse Beamforming," Proceedings of the IEEE, Vol 64, No. 8, August 1976, pp.1260-1261.

16. Van Trees, H.L., "Detection, Estimation, and Modulation Theory: part 1," John Wiley & Sons, Inc., New York, 1968.

17. Widrow, B., Mantey, P.E., Griffiths, L.J., and Goode, B.B, "Adaptive Antenna Systems," Proceedings of the IEEE, Vol 12, No. 12, December 1967, pp. 2143-2159.

18. Compton, R.T., "Adaptive Antennas," Prentice Hall, Inc., New Jersey, 1988.

19. Wiener, N., "The Extrapolation, Interpolation, and Smoothing of Stationary Time Series," John Wiley & Sons Inc., New York, 1949.

20. Widrow, B. and Stearns, S.D., "Adaptive Signal Processing," Prentice Hall, Inc., New Jersey, 1985.

21. Daniell, T.P., "Adaptive Estimation with Mutually Correlated Training Samples," Stanford Electronics Laboratories, Stanford, CA, Rep. SEL-68-083, TR 6778-4, August 1968.

22. Lancaster,P. and Tismenetsky, M., "The Theory of Matrices," Academic Press, Inc., San Diego, CA, 1985.

23. Widrow, B. and Hoff, M.E., "Adaptive Switching Circuits," IRE WESTCON Record, Pt. 4, 1960, pp.96-104.

24. Ahmed, N. and Rao, K.R., "Orthogonal Transforms for Digital Signal Processing," Springer-Verlag, Berlin GDR, 1975.

25. Oppenheim, A.V. and Schafer, R.W., "Discrete-Time Signal Processing," Prentice Hall, Inc., Englewood Cliffs, NJ, 1989.

26. Strobach, P., "Linear Prediction Theory," Springer-Verlag, Berlin GDR, 1990.

27. Haykin, S.S., "Adaptive Filter Theory," Prentice Hall, Inc., Englewood Cliffs, NJ, 1986.

28. Narayan, S.S., Peterson, A.M. and Narasimha, M.J., "Transform Domain LMS Algorithm," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-31, No. 3, June 1983, pp.609-615.

29. Compton, R.T., "On the Equivalence Between Tapped Delay-Line and FFT Processing in Adaptive Arrays," Ohio State University ElectroScience Laboratory, Final Technical Report 717253-5, June 1986.

30. Lee, J.C. and Un, C.K., "Performance of Transform-Domain LMS Adaptive Filters," IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-34, No. 3, June 1986, pp.499-510.

31. Widrow, B., McCool, J. and Ball, M., "The Complex LMS Algorithm," Proceedings of the IEEE, April 1975, pp.719-720.

32. Gelb, Arthur, editor, "Applied Optimal Estimation," MIT Press, Cambridge, MA., 1974.

33. Griffiths, L.J., "A Simple Adaptive Algorithm for Real-Time Processing in Antenna Arrays," Proceedings of the IEEE, October 1969, pp. 1696-1704.

34. Monzingo, R.A. and Miller, T.W., "Introduction to Adaptive Arrays," John Wiley & Sons, New York, NY, 1980.

35. Dentino, M., McCool, J. and Widrow, B., "Adaptive Filtering in the Frequency Domain," Proceedings of the IEEE, Vol 66, No 12, December 1978, pp.1658-1659.

36. Bershad, N.J. and Feintuch, P.L., "Analysis of the Frequency Domain Adaptive Filter," Proceedings of the IEEE, Vol 67, No 12, December 1979, pp. 1658-1659.

37. Griffiths, L.J., "A Continuously Adaptive Filter Implemented As a Lattice Structure," Proceedings of ICASSP, Hartford, CT, May 1977, pp.683-688.

38. Griffiths, L.J., "An Adaptive Lattice Structure for Noise-Cancelling Applications," Proceedings of ICASSP, Tulsa, OK, April 1978, pp. 87-90.

39. Griffiths, L.J., "Adaptive Structures for Multiple-Input Noise Cancelling Applications," Proceedings of ICASSP, Washington, DC, April 1979, pp. 925-928.

40. Clark, G.A., Soderstrand, M.A. and Johnson, T.G., "Transform Domain Adaptive Filtering Using a Recursive DFT," Proceedings of ISCAS, 1985, pp. 1113-1116.

41. Jenkins, W.K. and Kreidle, J.R., "Adaptive Digital Filters Using the Walsh-Hadamard Transform," Proceedings of ISCAS, 1986, pp. 875-878.

42. Jenkins, W.K. and Marshall, D.F., "On The Analysis and Design of Unitary Transformations for Adaptive FIR Digital Filters," Proceedings of the 29th Mid-West Symposium on Circuits and Systems, 1986, pp. 534-537.

43. Yassa, F.F., "Optimizing the Convergence Factor for the Frequency Domain Adaptive Algorithm," Proceedings of the 18th Asilomar Conference, 1985, pp. 61-66.

44. Ahmed, N., Natarajan, T. and Rao, K.R., "Discrete Cosine Transform," IEEE Transactions on Computers, January 1974, p. 90-93.

45. Soo, J.S. and Pang, K.K., "A Multi-Step Size Frequency Domain Adaptive Filter for Stationary and Nonstationary Signals," Proceedings of ICASSP, 1989, pp. 924-927.

46. Bershad, N.J. and Feintuch, P.L., "A Normalized Frrequency Domain LMS Adaptive Algorithm," IEEE Transactions on ASSP, Vol ASSP-34, No 3, June 1986, pp. 452-461.

47. Florian, S. and Bershad. N.J., "A Weighted Normalized Frequency Domain LMS Adaptive Algorithm," IEEE Transactions on ASSP, Vol 36, No 7, July 1988, pp. 1002-1007.

48. Lee, B.H., Chang, B.K., Cha, I.W., Kim, W.K. and Youn, D.H., "Realization of a Generalized Sidelobe Canceller," IEEE Transactions on Circuits and Systems, Vol CAS-34, No 7, July 1987, pp. 759-764.

49. Oliver, R.M., Pierce, J.R. and Shannon, C.E., "The Philosophy of Pulse Coded Miodulation," Proceedings of the IRE, Vol 36, No 11, November 1948, pp. 1324-1331.

50. Evans, J.B. and Liu, B., "Variable Step Size Methods for the LMS Adaptive Algorithm," Proceedings of ISCAS, 1987, pp. 422-425.

51. Gitlin, R.D., Mazo, J.E. and Taylor, M.G., "On the Design of Gradient Algorithms for Digitally Implemented Adaptive Filters," IEEE Transactions on Circuit Theory," Vol CT-20, No 2, March 1973, pp. 125-136.

52. Xue, P. and Liu, B., "On the Convergence of a Variable Step Size Adaptive Filter," Proceedings of ISCAS, 1985, pp. 1661-1662.

53. Honig, M.L. and Messerschmitt, D.G., "Adaptive Filters," Kluwer Academic Publishers, Hingham, MA, 1984.

54. Jim, Charles W., "A Comparison of Two LMS Constrained Optimal Array Structures," Proceedings of the IEEE, Vol 65, No 12, December 1977, pp. 1730-1731.

55. Yuen, Stanley M., "Comments on Convergence Properties of Gram-Schmidt and SMI Adaptive Algorithms," IEEE Transactions on Aerospace and Electronic Systems, Vol 27, No 6, November 1991, pp. 897-899.

56. Gerlach, Karl and Kretschmer, Frank E., "Convergence Properties of Gram-Schmidt and SMI Adaptive Algorithms," IEEE Transactions on Aerospace and Electronic Systems, Vol 26, No 1, January 1990, pp. 44-56.

APPENDIX I

The basic reference for this appendix is Lancaster [18]. Throughout this section, Γ will denote a general field and $\boldsymbol{R}$ will denote the set of real numbers.

Definition A1-1: A subspace

Let ξ denote a linear space over a field Γ and consider a nonempty subset ξ_o of the elements from ξ. The operations of addition and scalar multiplication are defined for all elements of ξ and, in particular, for all elements belonging to ξ_o. If these operations are closed in ξ_o so that for scalars α, γ and vectors $\boldsymbol{A}$, $\boldsymbol{B}$: $\boldsymbol{A} \in \xi_o$, $\boldsymbol{B} \in \xi_o$

$$\alpha \boldsymbol{A} + \gamma \boldsymbol{B} \in \xi_o$$

then we say that ξ_o is a subspace of ξ.

If ξ is a linear space, it is readily verified that if 0 is the zero element of ξ, then the singleton $\{0\}$ and the whole space ξ are subspaces of ξ. These are the trivial subspaces. It is important to note that the zero element of ξ is necessarily the zero element of any subspace ξ_o of ξ. This can be shown to be true by considering $\alpha \boldsymbol{A} = \boldsymbol{0}$ where $\alpha = 0$ is scalar and $\boldsymbol{A} \in \xi_o$.

Definition A1-2: A null space or kernel

The set of all vectors $\boldsymbol{W}$ such that $\boldsymbol{A}^T \boldsymbol{W} = \boldsymbol{0}$ is the nullspace or kernel of the matrix $\boldsymbol{A}$ and is written $Ker\,\boldsymbol{A}$.

Proposition A1-1: Let $\boldsymbol{A}^T \in \Gamma^{m \times n}$. Then the set of all solutions of the homogeneous equation $\boldsymbol{A}^T \boldsymbol{W} = \boldsymbol{0}$ forms a subspace of Γ^n which, by definition A1-2, is $\boldsymbol{\Sigma} = ker\,\boldsymbol{A}$.

Proof of Proposition A1-1: Let $\{Y, Z : Y \in \Sigma, Z \in \Sigma\}$. Then the vectors must satisfy $A^T Y = 0$ and $A^T Z = 0$. For any scalars α and γ the vector $A^T(\alpha Y + \gamma Z) = 0$. Hence, $\Sigma = ker\, A$ is a subspace.

Definition A1-3: A range space or image

A dual concept to that of the null space is the range space or image of a matrix A, denoted $\mathrm{Im}\, A$. Let $A^T \in \Gamma^{m \times n}$, then

$$\mathrm{im}\, A \equiv \{y \in \Gamma^m : y = A^T W \text{ for } some\, W \in \Gamma^n\}.$$

Definition A1-4: A span of a subspace

Any subspace containing the elements $\{a_i\}_{i=1}^n$ must also contain all elements of the form $\sum_{i=1}^n \alpha_i a_i$ for any $\alpha_i \in \Gamma$. This implies that the set of all linear combinations over Γ of the elements $\{a_i\}_{i=1}^n$ belonging to a linear space ξ generates a subspace ξ_o of ξ.

It can be seen that the subspace ξ_o above is the minimal subspace of ξ containing $\{a_i\}_{i=1}^n$ in the sense that $\xi_1 \supset \xi_o$ for any subspace ξ_1 which also contains $\{a_i\}_{i=1}^n$. This minimal subspace is called the linear hull or span of $\{a_i\}_{i=1}^n$ over Γ. Thus,

$$span\{a_i\}_{i=1}^n \equiv \{a \in \xi : a = \sum_{i=1}^n \alpha_i a_i, \{a_i\} \in \Gamma\}$$

Definition A1-5: Idempotent Matrix

The matrix A is said to be idempotent if

$$A^2 = A$$

which infers that for any positive integer i and idempotent matrix A, $A^i = A$.

Proposition A1-2: If P is an idempotent matrix, then:

1. $I - P$ is idempotent.

2. $\mathrm{Im}\,(I - P) = ker\,P$

3. $ker\,(I - P) = \mathrm{Im}\,P$

Proof of Proposition A1-2:

1. $(I - P)^2 = I - 2P + P^2 = I - P$

2. if $y \in \mathrm{Im}(I - P)$, then $y = (I - P)\,x$ for some $x \in \Gamma^n$. Therefore,

$$Py = P\,(I - P)\,x = (P - P^2)\,x = 0$$

and $y \in ker\,P$. Conversely, if $Py = 0$, then $(I - P)y = y$ and $y \in \mathrm{Im}\,(I - P)$ so that $\mathrm{Im}\,(I - P) = ker\,P$.

3. similar to the above argument, let $y \in ker\,(I - P)$. Then $Py = y$ since $(I - P)y = y - Py = 0$ and $y \in \mathrm{Im}\,P$ so that $ker\,(I - P) = \mathrm{Im}\,P$.

Proposition A1-3: If P is idempotent, then $ker\,P + \mathrm{Im}\,P = \Gamma^n$

Proof of Proposition A1-3: For any $x \in \Gamma^n$ we can write $x = x_1 + x_2$, where $x_1 = (I - P)x$ and $x_2 = Px$. Note that $x_1 \in ker\,P$ while $x_2 \in \mathrm{Im}\,P$. Hence, the whole space is defined as $\Gamma^n = ker\,P + \mathrm{Im}\,P$. Furthermore, if $x \in ker\,P \cap \mathrm{Im}\,P$, it must be the zero element.

Definition A1-6: A projection matrix

An idempotent matrix is a projection matrix. Each idempotent matrix P generates two unique mutually complementary subspaces $\xi_1 = ker\,P$ and $\xi_2 = \mathrm{Im}\,P$ and their sum is the entire space. Thus $\boldsymbol{P}$ performs the projection of the space Γ^n on the subspace ξ_2 parallel to ξ_1, or onto ξ_2 along ξ_1.

APPENDIX II

This appendix presents the proof that the GSC processor with any form of a spatial matrix filter W_S satisfying

$$W_sC = \mathbf{0} \tag{A2-1}$$

$$rank\,(W_s) = (K-1)\,J \tag{A2-2}$$

where the constraint matrix C is defined in equation (2-5), will yield an optimal processor which is equivalent to both the partitioned form and the direct form CLMS processors under the same constraint. This derivations follows directly from Jim [54].

The problem may be stated as follows:

Given the KJ x J dimensional constraint matrix C; the relationship $W_SC = \mathbf{0}$, resulting from the fact that look direction signals are eliminated from the GSC lower path; the KJ x KJ non-singular matrix $[W_S^T \;\; C]$, which spans the entire signal space since the rank of C is J and the rank of W_S is $(K\text{-}1)J$; and the KJ x KJ non-singular symmetric matrix R_{XX}; then, from equations (2-63) and (2-78),

$$I - W_s^T\,(W_sR_{xx}W_s^T)^{-1}W_sR_{xx} = R_{xx}^{-1}C\,(C^TR_{xx}^{-1}C)^{-1}C^T \tag{A2-3}$$

The proof is now developed through the existence of an orthogonal non-singular transformation matrix T such that

$$TC = \begin{bmatrix} \mathbf{0} \\ V \end{bmatrix} = \hat{C} \tag{A2-4}$$

Then,

$$(W_sT^{-1})\begin{bmatrix} \mathbf{0} \\ V \end{bmatrix} = \mathbf{0}\,, \quad W_sT^{-1} = [\,U \;\; \mathbf{0}\,] = \hat{W} \tag{A2-5}$$

We now let

$$TR_{xx}T^T = \begin{bmatrix} A & B \\ B^T & D \end{bmatrix} = \hat{R} \tag{A2-6}$$

and consider an operation

$$(T^T)^{-1}\,[I - W_s^T\,(W_s R_{xx} W_s^T)^{-1} W_s R_{xx} = R_{xx}^{-1} C\,(C^T R_{xx}^{-1} C)^{-1} C^T]\,(T^T) \tag{A2-7}$$

this implies

$$I - (W_s T^{-1})^T [(W_s T^{-1})(TR_{xx}T)(W_s T^{-1})^T]^{-1} (W_s T^{-1})(TR_{xx}T^T) = (TR_{xx}T)^{-1}(TC)\,[(TC^T(TR_{xx}T^T)^{-1} - (TC)]^{-1}(C^T T^T) \tag{A2-8}$$

which may be written as

$$I - \hat{W}^T\,(\hat{W}\hat{R}\hat{W}^T)^{-1}\hat{W}\hat{R} = \hat{R}^{-1}\hat{C}\,(\hat{C}^T\hat{R}^{-1}\hat{C})^{-1}\hat{C}^T \tag{A2-9}$$

To show that equation (A2-9) is valid, let $\hat{R}^{-1}$ be given by the block partition form matrix

$$\hat{R}^{-1} = \begin{bmatrix} X & Y \\ Y^T & Z \end{bmatrix} \tag{A2-10}$$

Then

$$\begin{gathered} X = (A - BD^{-1}B)^{-1} \\ Y = -A^{-1}BZ \\ Z = D^{-1} + D^{-1}B^T(A - BDB^T)^{-1}BD^{-1} = (D - B^T A^{-1} B)^{-1} \end{gathered} \tag{A2-11}$$

Furthermore,

$$\begin{aligned} \hat{R}^{-1}\hat{C}\,(\hat{C}^T\hat{R}^{-1}\hat{C})\,\hat{C}^T &= \begin{bmatrix} XV \\ YV \end{bmatrix} \begin{bmatrix} V^T & ZV \end{bmatrix}^{-1} [0 \;\; V^T] \\ &= \begin{bmatrix} 0 & YZ^{-1} \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} 0 & -A^{-1}B \\ 0 & I \end{bmatrix} \end{aligned} \tag{A2-12}$$

and

$$\hat{W}^T(\hat{W}\hat{R}\hat{W}^T)^{-1}\hat{W}\hat{R} = \begin{bmatrix} U^T \\ 0 \end{bmatrix} [U\ AU^T]^{-1} [UA\ \ UB] \tag{A2-13}$$

$$= \begin{bmatrix} I & A^{-1}B \\ 0 & 0 \end{bmatrix}$$

Therefore, equation (A2-3) holds and the proof is complete. Jim [54] also notes that there is no further restrictions on possible forms of the matrix filter $\boldsymbol{W_S}$ other than satisfying equations (A2-1) and (A2-2). If $\boldsymbol{W_S}$ is valid, then so is any non-singular transformation of $\boldsymbol{W_S}$.

*U.S. GOVERNMENT PRINTING OFFICE: 1993-710-093-60149

MISSION

OF

ROME LABORATORY

Rome Laboratory plans and executes an interdisciplinary program in research, development, test, and technology transition in support of Air Force Command, Control, Communications and Intelligence (C^3I) activities for all Air Force platforms. It also executes selected acquisition programs in several areas of expertise. Technical and engineering support within areas of competence is provided to ESD Program Offices (POs) and other ESD elements to perform effective acquisition of C^3I systems. In addition, Rome Laboratory's technology supports other AFSC Product Divisions, the Air Force user community, and other DOD and non-DOD agencies. Rome Laboratory maintains technical competence and research programs in areas including, but not limited to, communications, command and control, battle management, intelligence information processing, computational sciences and software producibility, wide area surveillance/sensors, signal processing, solid state sciences, photonics, electromagnetic technology, superconductivity, and electronic reliability/maintainability and testability.

END

FILMED

DATE: 4-93

DTIC