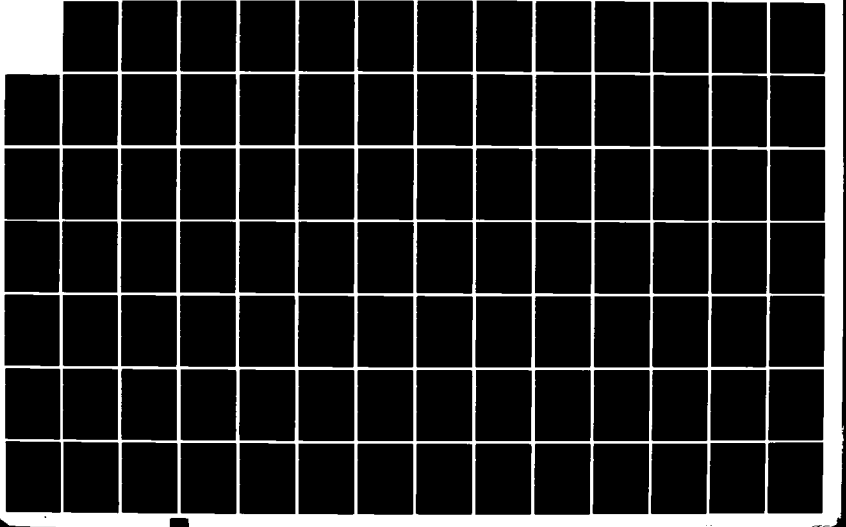
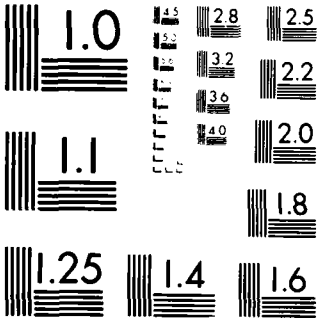
AD-A149 341 RESEARCH ON ADAPTIVE ANTENNA TECHNIQUES VI(U) STANFORD UNIV CA INFORMATION SYSTEMS LAB Y L SU ET AL. SEP 84 N00019-83-C-0287 1/2

UNCLASSIFIED F/G 17/4 NL





RESEARCH ON ADAPTIVE ANTENNA TECHNIQUES VI

FINAL REPORT

by

Young-Lim Su
Bernard Widrow

September 1984

Information Systems Laboratory
Department of Electrical Engineering
Stanford University
Stanford, CA 94305

This research was supported by the
Naval Air Systems Command of the Department of Defense
Under Contract N00019-83-C-0287

The views and conclusions contained in this document
are those of the authors and should not be interpreted as
necessarily representing the official policies, either expressed or implied
of the Naval Air Systems Command or the U. S. Government.

ABSTRACT

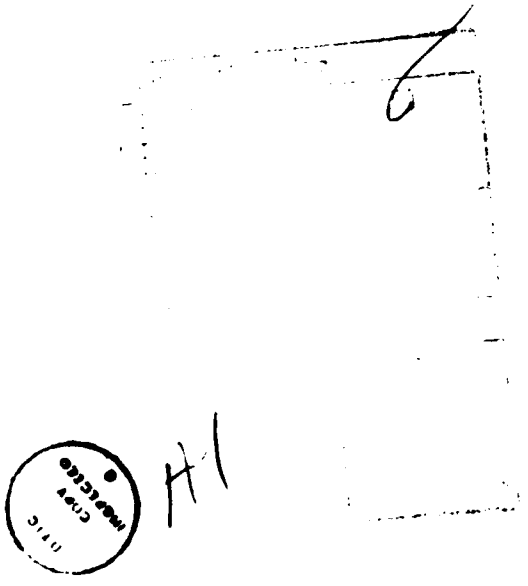
Signal cancellation is an effect which occurs in conventional adaptive arrays. This effect manifests itself as a loss of information in the desired signal. ~~In~~ this report, we will present two new adaptive array techniques to combat signal cancellation. These two new techniques are known as the frequency-hop spread spectrum approach and the parallel spatial processing approach.

The frequency-hop spread spectrum technique makes use of frequency-discrimination to combat jammer interference. Using the desired signal's frequency-hop nature, we can remove the signal from the adaptation process in a manner that eliminates signal cancellation. When the spread-spectrum technique and the spatial-discrimination inherent in adaptive arrays are combined, a system results with an interference rejection capability greater than either of the two alone. Several effective schemes and simulations ~~will be~~ presented.

The second technique makes use of spatial smoothing and parallel structure to eliminate signal cancellation. We will show that this new scheme results in a maximum-likelihood estimate of the desired signal in a spatial averaging sense. Simulation results are presented to illustrate the effectiveness of this proposed technique for combating signal cancellation.

TABLE OF CONTENTS

LIST OF FIGURES



I. INTRODUCTION

Adaptive arrays have been the subject of extensive investigation for the past two decades. They can reduce the receiver's vulnerability to interference when used in radar, sonar, seismic, and communication systems. The principal motivation for this widespread interest in adaptive arrays is their ability to sense and to automatically suppress the interference while simultaneously enhancing desired signal reception without prior knowledge of the signal/interference environment. Perhaps the easiest way to visualize the operation of an adaptive array is to consider the response in terms of the array beam sensitivity pattern. Interference suppression is obtained by appropriately steering a beam pattern null and reducing sidelobe levels in the directions of interference sources, while desired signal reception is maintained by preserving proper mainlobe features. An adaptive array system therefore relies heavily on spatial characteristics to improve the output signal-to-noise ratio (SNR). Since it is possible to form very deep nulls over a certain frequency band, very strong interference suppression can be achieved. This exceptional interference suppression capability is a principal advantage of adaptive arrays over other techniques.

In the early 1960s the key capability of adaptive interference nulling was recognized and developed by Howells [1.1-1.2]. Subsequently, Applebaum [1.3] established the control law associated with the Howells adaptive nulling scheme by analyzing an algorithm that maximizes a generalized SNR. Concurrently, Widrow *et.al.* [1.4] applied the technique of self-training or self-optimizing

control to adaptive arrays. This self-optimizing control work established the least mean square error (LMS) algorithm that is based on the method of steepest descent. The Applebaum and the Widrow algorithm are very similar, and both converge toward the optimum Wiener solution. Developments in seismic and acoustic array work commenced at about the same time, so papers describing applications of seismic arrays and hydrophone arrays appeared during the late 1960s. Capon *et.al.* [1.5] and Lacoss [1.6] addressed adaptive signal processing in seismic arrays. while Shor [1.7] worked with hydrophone arrays.

The original Howells-Applebaum sidelobe-canceller exploited the differing signal-to-jammer ratios in a directive primary antenna and an omni-directional auxiliary antenna to avoid seriously attenuating desired radar signals. Widrow introduced a pilot signal to control beamformer response in specified look directions. Griffiths [1.8] devised a different soft-constraint technique that involved statistical characterization of the desired signals. Frost [1.9-1.10] developed a constrained least-mean-square (LMS) algorithm that assured exact conformance with some prespecified look-direction response. More recently, Griffiths and Jim [1.11-1.12] contributed a structure called the "generalized sidelobe canceller," which provided an alternative method of realizing hard constraints. In the past few years Zahm and Gabriel [1.13-1.16] developed a generalization of the soft-constraint method. Chestek [1.17] brought together much of the earlier work on soft-constraint methods by combining soft linear constraints with a mean-square-error criterion.

Although a number of various processing techniques for adaptive arrays have

been described, these techniques can be simply categorized as implementing either hard or soft constraints. The Frost beamformer is an example of a hard-constraint adaptive array, whereas the adaptive sidelobe canceller is a soft-constraint adaptive array.

1.1 Background

Many adaptive array issues arise in applications of direction finding and adaptive beamforming. The preservation of desired signal from the adaptive arrays is one of the issues of great concern. Research over the last few years have been mainly directed toward achieving satisfactory SNR performance and yielding highly refined adaptive systems that can overcome most forms of clutter and jamming.

In any adaptive array applications, assumptions are always made for the desired signal and the interference. For example, the strength of the desired signal is small compared to the interference, or the signal is statistically uncorrelated with the interference. In fact, most algorithms will *not* work if the desired signal is correlated with interference. This limitation is a severe obstacle for adaptive arrays in many applications where multi-path propagations or smart jamming problems exist. Frost pointed out in his paper [1.10] that a linearly constrained adaptive array may cancel out portions of the desired signal with jammers present in spite of the constraints. This cancellation of signal occurs when the jammers are correlated with the desired signal.

Recently, Widrow *et.al.* [1.18-1.19] have found that a sinusoidal jammer sitting in the frequency band of the desired signal can easily destroy the "quality" of the output response in any form of adaptive array. Since then, Widrow *et.al.* have explored and studied the so called "signal cancellation effect" in adaptive arrays, and proposed Duvall's beamformer and the spatial dither algorithm to combat the signal cancellation phenomena. More recently, Shan [1.20] suggested a spatial smoothing algorithm for adaptive beamforming to breakup the coherent signal cancellations.

The purpose of this report is to study two new approaches for avoiding signal cancellation in adaptive arrays. The first approach is based on frequency-hop spread spectrum systems. The second one is based on spatial processing with parallel structures.

1.2 Outline of Chapters

Chapter II illustrates signal cancellation and provides insight from various perspectives of adaptive arrays. With that background, we then propose two different approaches to eliminate signal cancellation. They are the frequency-hop spread spectrum approach and the parallel spatial processing approach. Chapter III and IV focus on the frequency-hop spread spectrum approach. Chapter V concentrates on the parallel spatial processing approach.

Chapter III first gives a brief introduction to frequency-hop spread spectrum systems and provides the necessary background about the whole system that incorporating with frequency-hop adaptive arrays. Several different schemes are

then developed for frequency-hop adaptive arrays. Simulation results are presented to verify the effectiveness of these algorithms in preventing signal cancellation.

It will be seen in Chapter III that frequency-hop notch filters are always required in frequency-hop adaptive arrays. Chapter IV then studies several types of frequency-hop notch filters. Transient performance and convergence are investigated. Comparisons of bandwidth and frequency response for filters are made to provide general insight into their use in frequency-hop adaptive arrays.

Chapter V gives an introductory background and reviews previous work in the field of spatial processing. Then, a spatial processing algorithm with parallel array structures is developed. This algorithm results in a maximum likelihood estimate of desired signal in a spatial averaging sense. Analysis and simulation results that show the performance of the algorithm are provided.

II. SIGNAL CANCELLATION IN ADAPTIVE ARRAYS

This chapter first illustrates the problem of signal cancellation in adaptive arrays, then provides insight into the nature of this effect, and finally discusses general approaches for eliminating signal cancellation. We will focus on the case of narrow-band signals.

The chapter consists of four sections: Section 2.1 demonstrates signal cancellation in adaptive arrays. Section 2.2 examines how this effect occurs in the Frost beamformer, sometimes called a hard-constraint adaptive array. In Section 2.3, the same effect is investigated for the adaptive sidelobe canceller, which is a soft-constraint adaptive array. Based on the insight gained in last two sections, Section 2.4 discusses and proposes cures for preventing signal cancellation in adaptive arrays.

2.1 Introduction

Conventional adaptive arrays are known to be very effective in suppressing directional jammers. This can be achieved by forming spatial nulls in the directions of these jammers, provided that the desired signal and jammers are uncorrelated. The nulls are created by weighting the received jammer components in a manner that the jammers but not desired signal are cancelled at the array output. This optimal weighting is frequently referred to as the Weiner solution. Improper weighting may, however, cause partial or total cancellation of the desired signal components at the array output. This phenomenon is called

signal cancellation, and it may seriously degrade the performance of adaptive arrays. Several researchers [1.10, 2.1] have observed and reported such signal cancellation effects in adaptive array systems.

Recently, Widrow *et.al.* [1.18-1.19] have demonstrated signal cancellation in the Frost beamformer. The Frost beamformer uses a constrained LMS algorithm to minimize its output power. In this demonstration, a unit-gain constraint is imposed in the look direction of the Frost beamformer. Suppose that a desired broadband signal is arriving from the look direction. This desired signal should appear at the array output after going through a constrained unit gain. Now if a sinusoidal jammer arrives off the look direction, this jamming sinusoid should be rejected by the adaptive array. When both the jammer and the desired signal are present, however, minimizing the total output power will cause the sinusoidal jammer to be modulated so that it cancels some components of the desired signal close to the jammer frequency [2.2]. Figure 2.1 shows the spectra of both a desired broadband signal and the received array output. The jammer is a sinusoid, and its spectrum is a line sitting at a normalized frequency of 0.25. Notice that the signal components around the jammer's frequency in Figure 2.1 have been cancelled at the array output. If the jammers consist of a sum of sinusoids at spaced frequencies within the passband of the desired signal, the output spectrum will be notched at each of the jammer frequencies as shown in Figure 2.2. This effect results in a loss of information from the desired signal, and could be troublesome in anti-jamming or spread spectrum communications.

If a fast adaptation rate is employed, the jammer modulation is more

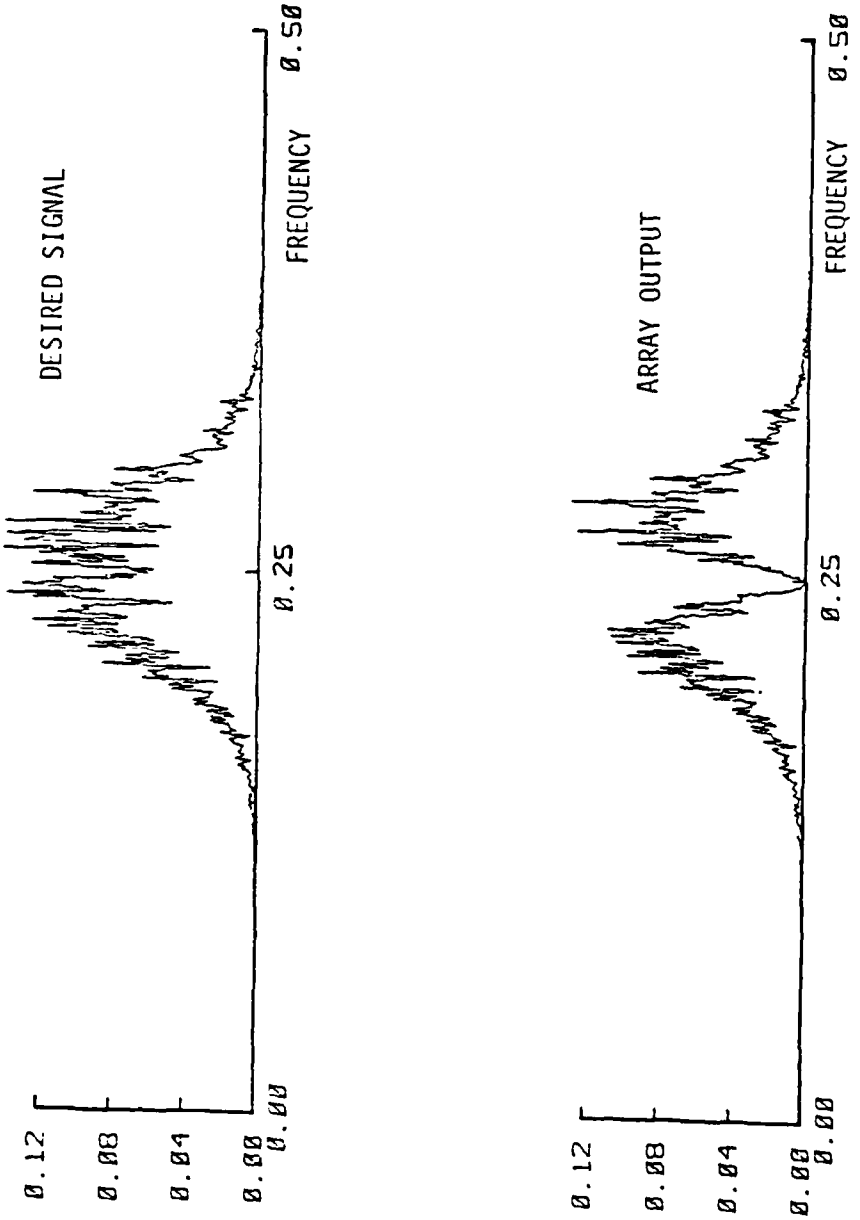


Figure 2.1. Power spectra of a desired broadband signal and an array output when a sinusoidal jammer is present at frequency 0.25.

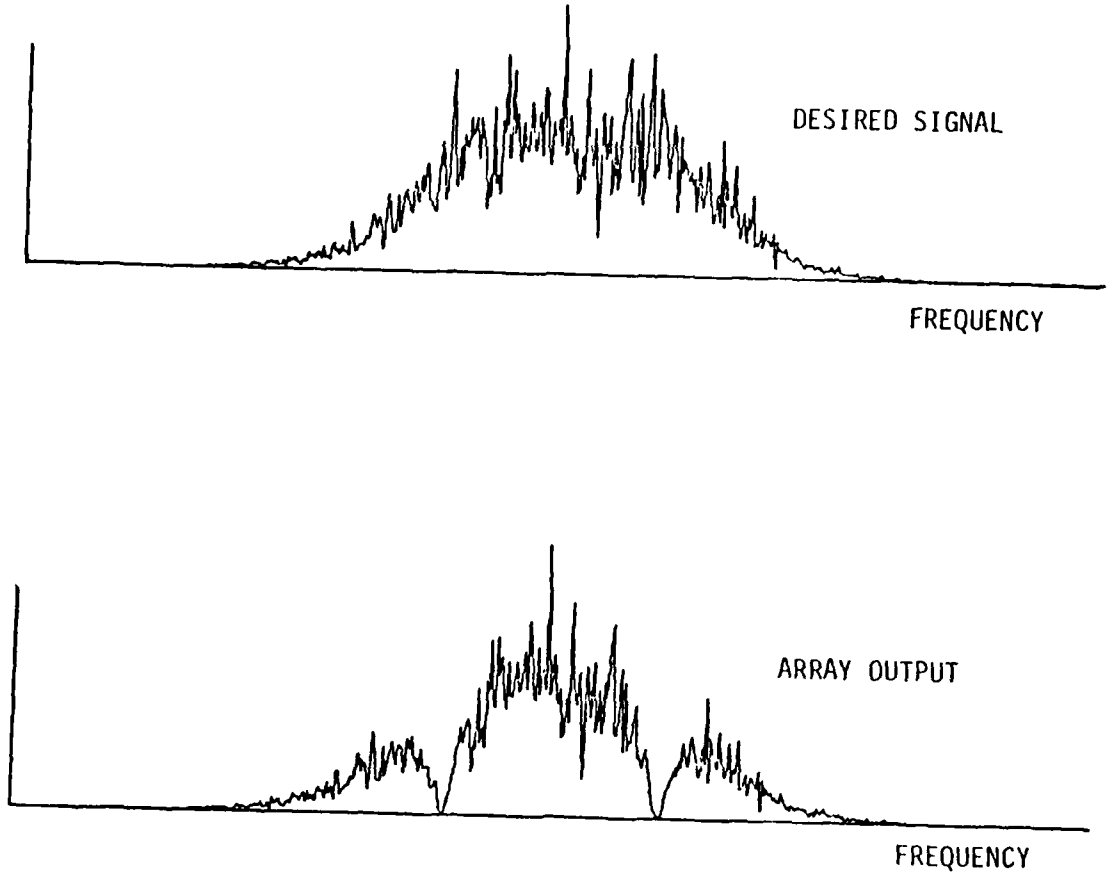


Figure 2.2. Power spectra of a desired signal and an array output when sinusoidal jammers are present at two frequencies.

effective and complete desired signal cancellation can occur. In such situations, signal cancellation can be induced even with a broadband jammer. Widrow recognized that this signal cancellation can be related to the adaptive noise cancelling problems [2.3]. In adaptive noise cancelling, an adaptive noise canceller is able to create a null at the frequency of a sinusoidal reference signal. By virtue of this nulling ability, the adaptive noise canceller performs as a linear, time-invariant, notch filter. The step size μ of the LMS adaptive algorithm controls the width of the notch. The larger the step size, the wider the notch [2.4]. This is similar to the problem of wide-band signal cancellation. Figure 2.3 compares the output spectra showing different cases of wide-band signal cancellation for different values of μ. Both spectra have same frequency null centered at the jammer's frequency, but the larger the value of μ, the wider the bandwidth of the cancellation notch. The step size μ obviously plays a similar role in both adaptive noise cancelling and signal cancellation. That is, the faster the adaptation rate, the wider the cancellation width, since faster adaptation allows for higher frequency modulation of the jammer.

If the desired signal is a narrow-band signal sitting at the same frequency as the jammer, the output of the adaptive array may fall to zero, as shown in Figure 2.4. The convergent beam pattern could thus form a null in a false direction as shown in Figure 2.5. In this situation, the adaptive array completely fails to perform as a receiving array.

As another example, it has been found in [2.2] that signal cancellation can occur even if the jammer and desired signal are sinusoids with slightly different

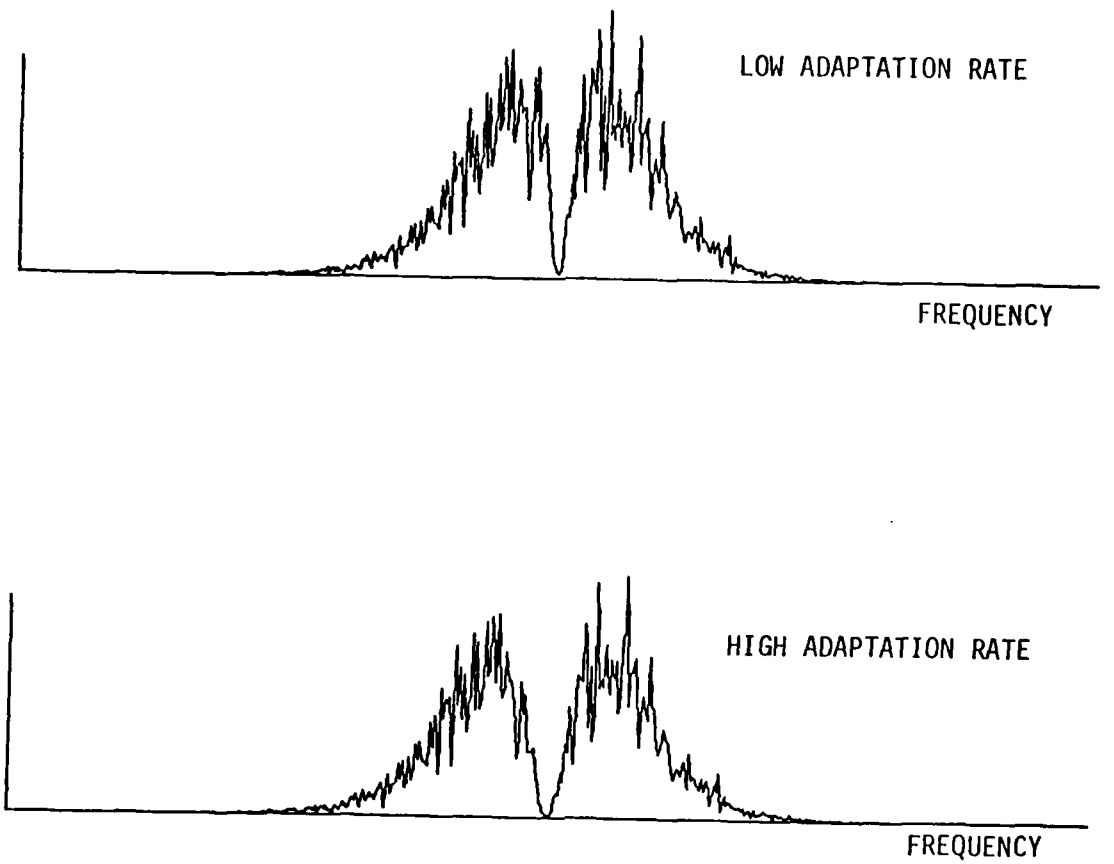


Figure 2.3. Power spectra of the array output using different adaptation rates.

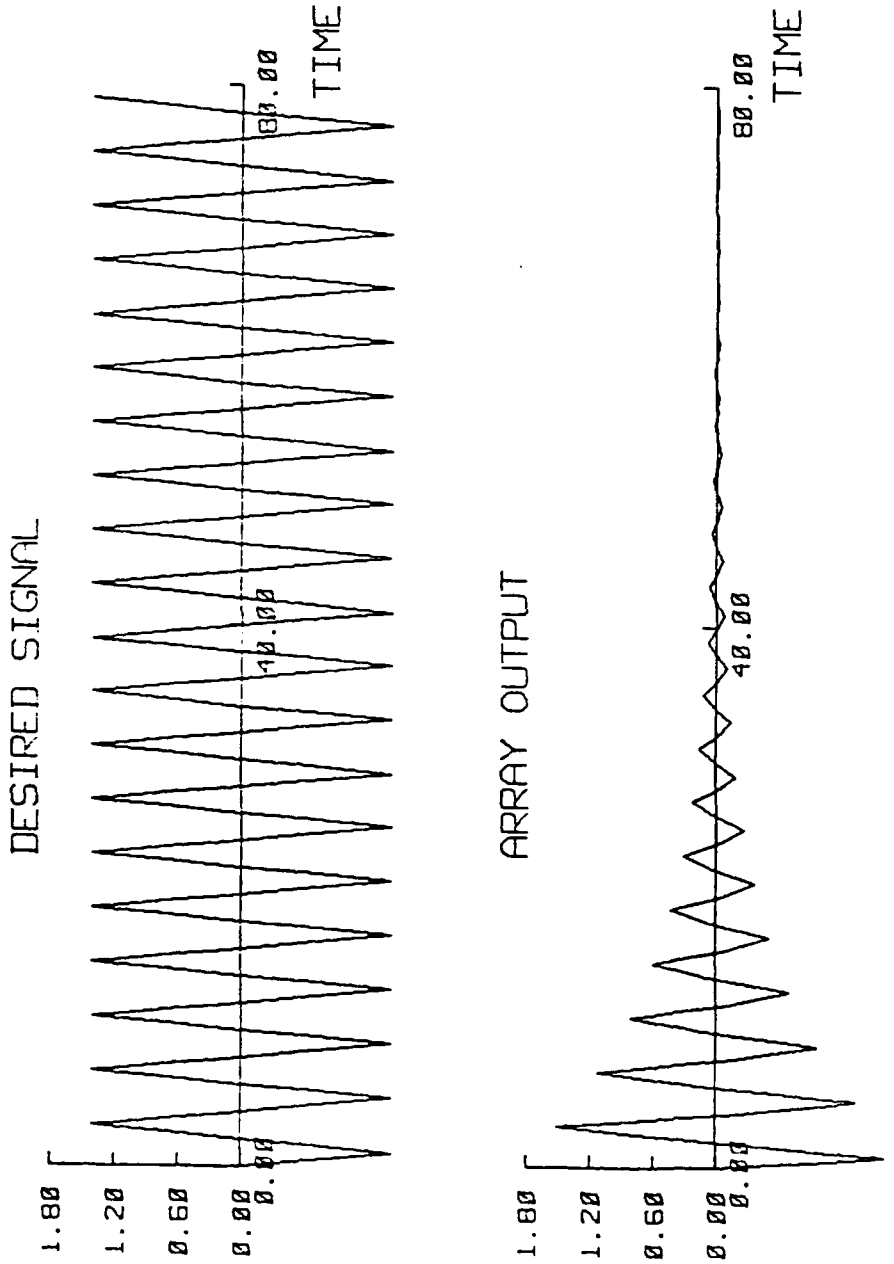


Figure 2.4. Time domain waveforms of a desired sinusoidal signal and an array output when signal cancellation occurs.

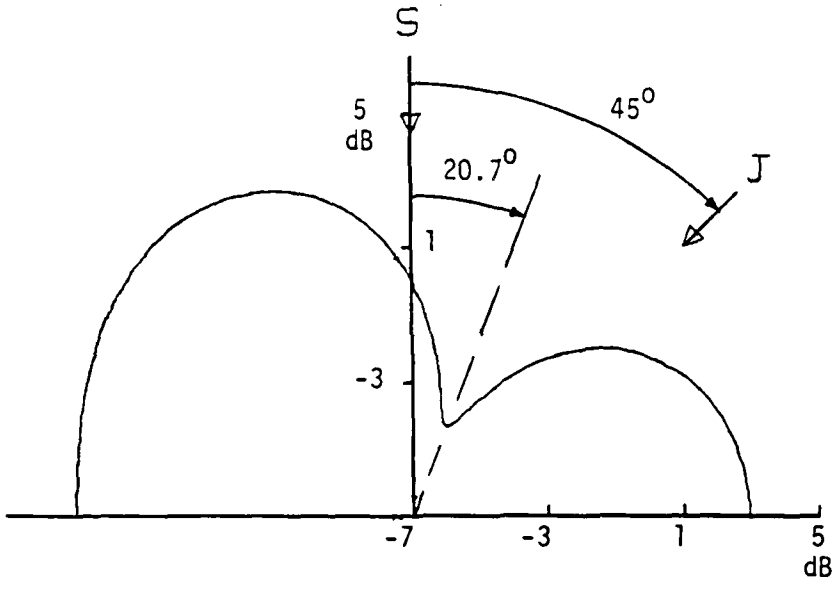


Figure 2.5. A converged array beam pattern when signal cancellation occurs.

frequencies. The adaptive weights are modulated during the adaptation process. This weight performance is far from that of the Weiner solution. This signal cancellation due to the weight modulation is described by Widrow *et.al.* [1.18], and is also referred to as the non-Weiner signal cancellation.

Another form of signal cancellation takes place in adaptive arrays that have "soft" omnidirectional or look-direction constraints. The Howells-Applebaum sidelobe canceller is an example of such an array [2.1]. An adaptive array algorithm which uses soft constraints may null the desired signal if its power is much greater than the power of jammers. This is because nulling the high-power desired signal minimizes output power more than nulling the low-power jammers. This effect is referred to as Weiner signal cancellation, since it is a property of the converged Weiner solution. Notice that non-Weiner signal cancellation occurs in any form of adaptive arrays, but Weiner signal cancellation occurs only in "soft-constraint" adaptive arrays.

2.2 Signal Cancellation in Frost Adaptive Beamformers

In this section, we investigate signal cancellation in the Frost adaptive beamformer. The Frost beamformer minimizes its total output power by employing a constrained least-mean-square criterion. Since its structure can easily impose a linear constraint in the look direction, sometimes it is called a hard-constraint adaptive array. A typical constraint is one that forces the beamformer to form a unit gain and zero phase over a certain frequency band in the look direction.

The most important point regarding signal cancellation is the quality of the adaptive array output. To demonstrate how this effect can occur, consider a simple two-element Frost beamformer as shown in Fig. 2.6. Suppose a sinusoidal desired signal is arriving from the look direction, and a jammer at the same frequency as the desired signal and with a fixed phase shift is arriving from the off-look direction. Let the desired signal S and the jammer J be the following,

$$\begin{aligned} S &= Ae^{jwt} \\ J &= Be^{jwt+j\phi} \quad , \end{aligned} \tag{2.1}$$

where A and B are the corresponding amplitude of the signal and the jammer, ϕ is a constant phase difference between S and J, and w is the angular frequency. In the sinusoidal case, sometimes a phasor diagram may be useful in explanation, and we will use it to explain the false beamforming later.

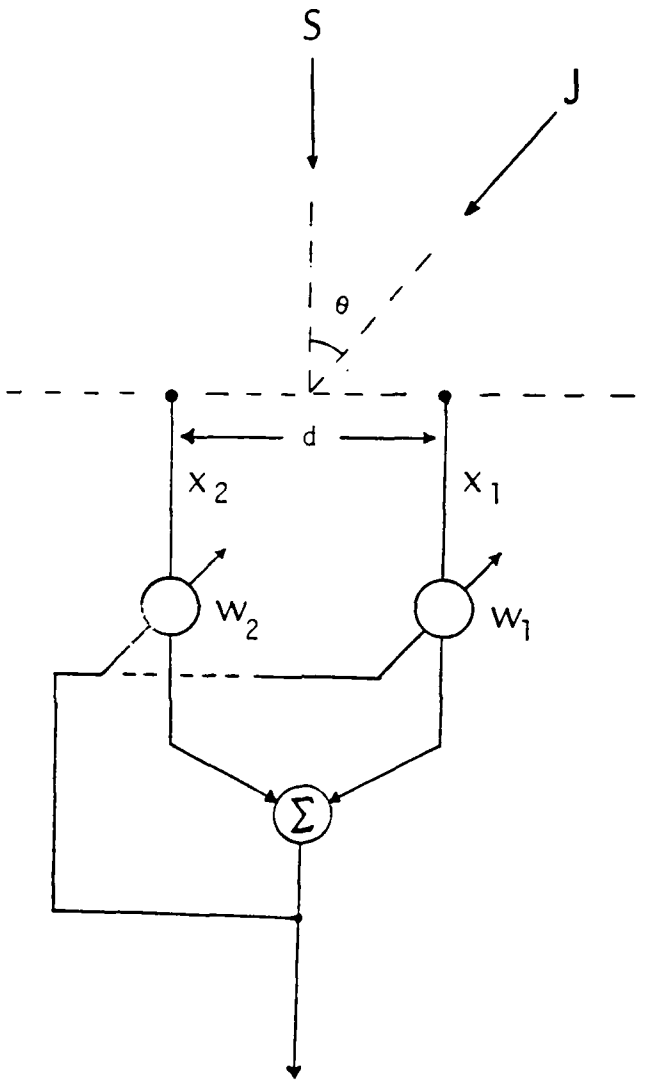


Figure 2.6. A simple two-element Frost adaptive beamformer.

In Figure 2.6, the receiving element #1 receives both the desired signal and jammer as

$$X_1 = Ae^{jwt} + Be^{jwt+j\phi} \quad , \tag{2.2}$$

and the element #2 receives the same signal plus the delayed jammer as

$$X_2 = Ae^{jwt} + Be^{jwt+j\phi-jw\Delta} \quad . \tag{2.3}$$

where

$$\begin{aligned} \Delta &= \frac{d\sin\theta}{c} \\ d &= \text{the inter-element distance} \\ c &= \text{the speed of propagation} \\ \theta &= \text{the jammer's incident angle from broadside} \quad . \end{aligned}$$

Denote the weight vector and the received signal vector as

$$\begin{aligned} W &= [\, W_1 \;\; W_2 \,]^T \\ X &= [\, X_1 \;\; X_2 \,]^T \quad . \end{aligned} \tag{2.4}$$

The beamformer output is thus given by

$$\begin{aligned} \mathbf{y} &= W^T X = X^T W \\ &= W_1 X_1 + W_2 X_2 \quad . \end{aligned} \tag{2.5}$$

The adaptive weights are complex, and the complex algorithm for linearly constrained adaptive beamformers [2.5] is used. For a detailed analysis, please refer to Appendix A. The constraint in the look direction is set to unity gain and zero phase from zero frequency to half the sampling rate. Thus the Frost algorithm can be expressed as the following.

$$\begin{aligned} &\min_{W} \ |\mathbf{y}|^2 \\ &\text{subject to} \quad W_1 + W_2 = 1 \quad , \end{aligned} \tag{2.6}$$

or is equivalently given by

$$\begin{aligned} &\min_{W} \ | \ W_1 X_1 + W_2 X_2 \ | \\ &\text{subject to} \quad W_1 + W_2 = 1 \quad . \end{aligned} \tag{2.7}$$

Substituting X_1, X_2 and W_1 into (2.7) yields an unconstrained minimization problem as follows.

$$\min_{W_2} \ | e^{jwt} | \cdot | A + Be^{j\phi}[1 - W_2 + W_2 e^{-jw\Delta}] | \quad . \tag{2.8}$$

Solving (2.8), one easily finds that the optimal weight W_2^* is

$$W_2^* = \frac{1}{1 - e^{-jw\Delta}} + \frac{Ae^{-j\phi}}{B(1 - e^{-jw\Delta})} \quad . \tag{2.9}$$

Note that the optimal solution results in a zero output when the adaptive processor reaches steady state. *i.e.*,

$$\mathbf{y}_{\min}(\infty) = W^{*T} X = 0 \quad .$$

Of course this is undesirable. Ideally the output should be the desired signal only, with no added coherent jammer. By this criterion, the optimal solution $W_{2\,opt}$ should be

$$W_{2\,opt} = \frac{1}{1 - e^{-jw\Delta}} \quad . \tag{2.10}$$

Comparing (2.9) and (2.10), we may see that there are two ways to force the weights to the optimal solution in coherent jamming environment. The first one

is to set A zero, or to eliminate the desired signal in the adaptive processor. The second one is to make $A << B$, which means the signal power should be much smaller than the jammer power.

In his master-slave beamformer, Duvall [2.2] applied the first idea to remove desired signal from adaptation. Since no desired signal is involved due to inter-element subtraction in Duvall's master beamformer, the influence over weight settings will be dominated by the jammers. The adaptive weights therefore reach an optimal solution which cancels the jammers only.

Now consider the case when $A << B$, which means a very strong jammer is present, then $W_2^* \cong W_{2\,opt}$. Note that the output $\mathbf{y}$ equals to $W^T X$. This still results in a zero output, even though the weight is very close to the optimal solution. This is better explained from the perspective of covariance space. Shan [2.6] has shown that in a coherent signaling environment the sample covariance matrix has a zero eigenvalue. Thus minimization with respect to the weights will steer the weight vector to align with the eigenvector corresponding to this zero eigenvalue. The output of the beamformer hence falls down to zero.

To understand the false nulling phenomenon of the beam pattern, it is helpful to consider a phasor diagram as shown in Figure 2.7. In this phasor diagram, $\vec{OQ}$ and $\vec{OR}$ are the jammer components received by the element #1 and #2, respectively. The angle $\angle QOR$ represents the phase delay $w\Delta$ between the jammer components at element #1 and #2. $\vec{PO}$ is the desired signal received by both elements. An ideal adaptive beamformer should form a null in a direction such that the phase delay is $w\Delta$.

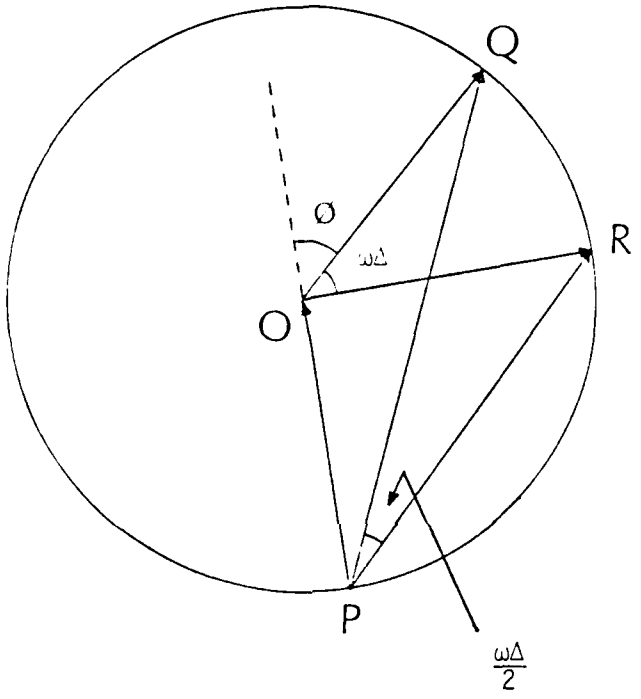


Figure 2.7. A phasor diagram explain the false nulling.

Let the length represent the amplitude. For a far-field planewave jammer, each element receives equal jammer amplitude, namely $|\vec{OQ}| = |\vec{OR}|$. The received amplitude may vary from element to element for a near-field jammer. Without loss of generality, suppose both the jammer and the desired signal have equal power intensity. In other words, $|\vec{PO}| = |\vec{OQ}| = |\vec{OR}|$. For the coherent jamming situation, both the jammer and the desired signal have the same frequency w. The relative phase difference ϕ between signal and jammer is a fixed constant. In the phasor diagram, this means $\vec{OQ}$ and $\vec{OR}$ are rotating about point O with angular speed w, and $\vec{PO}$ is rotating about point P with the same angular speed. The relative phase difference ϕ between signal and jammer should not be confused with the phase delay $w\Delta$ between the jammer components at elements #1 and #2.

One can easily see that $\vec{PQ}$ is the phasor superposition of $\vec{PO}$ and $\vec{OQ}$, whereas $\vec{PR}$ is the phasor superposition of $\vec{PO}$ and $\vec{OR}$. In other words, $|\vec{PQ}|$ actually represents X_1 as received at element #1, and $|\vec{PR}|$ represents X_2 as received at element #2. Since the signal and the jammer have the same frequency w, both $\vec{PQ}$ and $\vec{PR}$ are then rotating about point P with the same angular speed w. Besides, the phase delay $\angle QPR$ between $\vec{PQ}$ and $\vec{PR}$ has been fixed, and it is easy to verify that $\angle QPR = w\Delta/2$ by geometrical identities. This means that the phase delay between antenna element #1 and antenna element #2 is changed to another fixed value, which is $w\Delta/2$ instead of $w\Delta$. Note that this is virtually equivalent to the following scenario: A near-field jammer, with no desired signal arrives in a direction for which the inter-element

phase delay is $w\Delta/2$. For a near-field jammer, the array elements should receive jammer components with significant attenuation. Since it "looks" like a near-field jammer, the adaptive beamformer, subject to the minimization algorithm, will always adapt to minimize the beamformer output power. Therefore, forming a null in a wrong direction with phase delay $w\Delta/2$, rather than $w\Delta$, will still achieve the power minimization. This accounts for the false nulling of the adaptive array. Again, note that the desired signal has been cancelled, and hence can not be recovered at the array output.

Simulations with the Frost beamformer in Fig. 2.6 were conducted to verify the above argument. The inter-element distance was half a wavelength. A coherent jammer as well as a desired signal are received by the adaptive beamformer. Both have equal power intensity 1. The desired signal is from broadside and the jammer is in a direction 45 ° from broadside. The output of the beamformer is shown in Figure 2.4. The resultant beam pattern in Figure 2.5 has a null in a direction 20.7 ° from the broadside. By the above false nulling argument, one could verify that

$$w \sin 20.7^\circ = \frac{w \sin 45^\circ}{2} \quad . \tag{2.11}$$

Hence, the adaptive array finds that forming a null at 20.7 ° from broadside can minimize array output more than forming a null at 45 ° from broadside. This false nulling is not easily seen when the signal power and the jammer power are not of the same order of magnitude, especially when the jammer power is much stronger than the signal power. In such a situation, $|\vec{PO}|$ is very small compared to $|\vec{OQ}|$ or $|\vec{OR}|$, and the fixed phase delay $\angle QPR$ between X_1 and

X_2 will be very close to $w\Delta$. The resultant null of the converged beam pattern will be very close to the jammer's bearing. Even though the beam pattern looks correct in this case, the beamformer output still falls to zero which leads to difficulty in recovering the desired signal.

2.3 Signal Cancellation in Adaptive Sidelobe Cancellers

In last section, we demonstrated that the desired signal was cancelled by the jammers in the "hard-constraint" adaptive arrays. In this section, we will show that the same cancellation can occur in a "soft-constraint" adaptive array. The adaptive sidelobe canceller is an example of the "soft-constraint" adaptive arrays. To show how signal cancellation can occur, consider a simple two-element adaptive sidelobe canceller as shown in Figure 2.8. Both receiving elements are assumed omni-directional. The adaptive weight is complex and is updated by the complex LMS algorithm [2.7]. Again the signal and the jammer are coherent as in (2.1). The receiving elements #1 and #2 receive X_1 and X_2 as in (2.2) and (2.3), respectively. The array output $\mathbf{y}$ is given by

$$\mathbf{y} = X_1 - WX_2 \quad .$$

Mathematically, the adaptive algorithm can be expressed as follows,

$$\min_{W} |\mathbf{y}|^2 \quad .$$

This appears as an unconstrained minimization problem, and the minimum solution for the complex weight is

$$\begin{aligned} W_{min} &= \frac{A + Be^{j\phi}}{A + Be^{j\phi - jw\Delta}} \quad . \\ &= e^{-jw\Delta} + \frac{A(1-e^{-jw\Delta})}{A + Be^{j\phi - jw\Delta}} \quad . \end{aligned} \tag{2.12}$$

Again, this minimal solution results in a zero array output, when the adaptive processor reaches steady state. The desired signal is totally cancelled out by the jammer, and the adaptive array fails to perform as a receiving array.

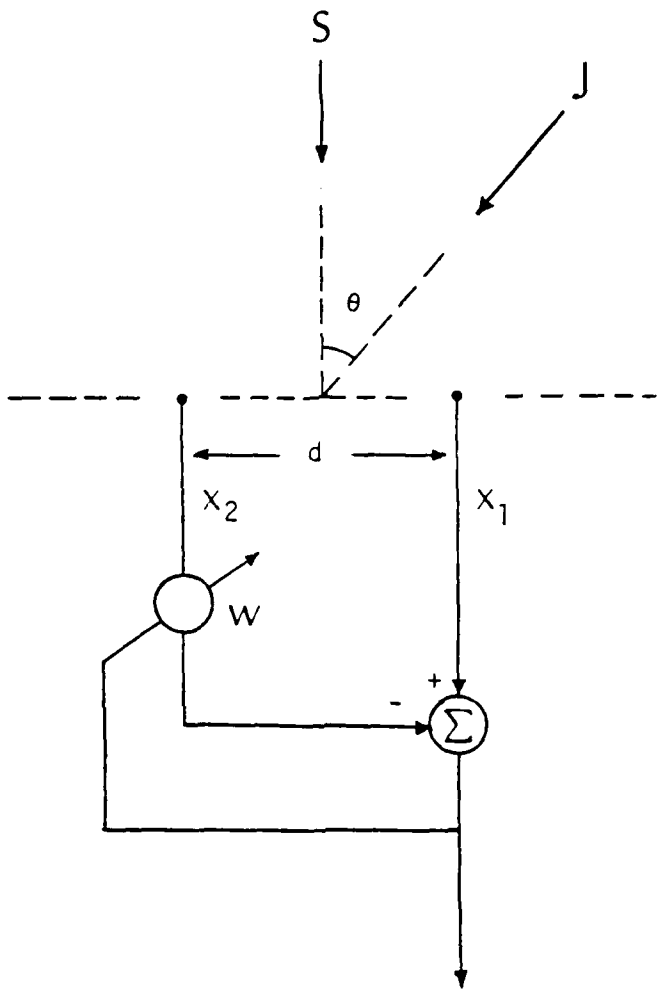


Figure 2.8. A simple two-element adaptive sidelobe canceller.

Usually, the adaptive sidelobe canceller is very effective when jamming power is much higher than signal power. The optimal solution should be able to suppress the jammer only and is given by

$$W_{opt} = e^{-jw\Delta} \quad . \tag{2.13}$$

Comparing (2.12) and (2.13), it shows that the weight can be the optimal solution if A is zero. This implies that the removal of desired signal from adaptation process is a key point to combat signal cancellation in adaptive arrays.

2.4 Cures for Signal Cancellation

In this section we first discuss the nature of signal cancellation, and then propose several cures to eliminate such a negative effect in adaptive arrays.

From Section 2.2 and 2.3, one can easily see that signal cancellation arises because of the following: First, the power minimization algorithm is used. Second, non-zero correlation between signal and jammer exists. Unfortunately, these scenarios happen all the time. Any adaptive array using the mean square error (MSE) minimization criteria exhibits such "signal cancellation" phenomena. Furthermore, Duvall showed in his thesis that signal cancellation can occur even when the jammer is uncorrelated with the desired signal, and the weights are also modulated during the adaptation process. More details about the weight-modulation effect in the hard-constraint adaptive array are described in Duvall [2.2].

As we have seen, there are several symptoms present if signal cancellation occurs during the adaptation process. First of all, the adaptive weights are modulated. Secondly, the output spectrum is distorted. Thirdly, the beam pattern may show false nulling. The key point concerns the interaction of the jammer with the signal during the adaptation process. Any properly designed preprocess that can separate the desired signal from the jammer in some manner will essentially eliminate signal cancellation. By such a preprocess, the adaptive array would be able to null the jammer only, and to recover the desired signal as well. The preprocess has to utilize the *a priori* information either from the signal or from the jammer.

If the direction of the desired signal is known, Duvall has developed a master-slave beamformer to combat signal cancellation in his thesis [2.2]. The idea of Duvall's beamformer is to preprocess the desired signal prior to the adaptive process. In his method, inter-element subtraction is employed to remove the desired signal before the adaptive processor. By taking the signal out of the adaptation process, the signal/jammer interaction in the adaptive process is essentially eliminated. Notice that the inter-element subtraction can be applied only when identical antenna elements are used.

Another idea is to preprocess the jammers so as to break up the signal/jammer correlation. To do so, spatial discrimination can be applied to such preprocessing. Widrow has suggested a spatial dither method, which is also called the "3/4in-plywood" method, to break up the signal/jammer correlation [1.18].

So far, all the methods proposed by Widrow and Duvall are suggested for the hard-constraint adaptive arrays. In such an adaptive array, the signal's direction is assumed known. Since signal cancellation also exists in soft-constraint adaptive arrays, more powerful methods to eliminate this effect should be devised. The remainder of this section suggests other approaches to eliminate signal cancellation in adaptive arrays. Two approaches are proposed.

The first approach is based on frequency-hop spread spectrum techniques. Chapter III and Chapter IV investigate and discuss the effectiveness of such an approach. In this approach, the desired signal is assumed to be frequency-hopping. The only information about the desired signal known to the adaptive

arrays is the pseudo-random (PN) code. The term "pseudo-random" is used specifically to mean random in appearance but reproducible by deterministic means. This approach is very suitable for soft-constraint adaptive arrays, or for applications where the signal's direction is unknown.

Another approach is based on parallel spatial processing techniques. This approach is investigated in Chapter V. In this part, we assume the direction of the desired signal is known. The idea is to apply spatial smoothing in a direction orthogonal to the look direction, so that any off-look jammer will be spatially smoothed out.

III. FREQUENCY-HOP ADAPTIVE ARRAYS

In Chapter II we demonstrated the phenomenon of signal cancellation, which exists in most adaptive arrays. We briefly mentioned two new cures for signal cancellation: one is the frequency-hop approach; the other is the spatial smoothing approach. Chapter III will investigate the first approach.

The chapter is organized in seven sections: Section 3.1 gives a brief introduction to frequency-hop spread-spectrum techniques and their integration with adaptive arrays. In Section 3.2, we describe an overall system structure for frequency-hop adaptive arrays. From Section 3.3 to Section 3.6, we present four different adaptive array schemes to combat signal cancellation, when the desired signal is known to be a frequency-hop spread-spectrum signal. In Section 3.3, the first scheme is called frequency-hop adaptation algorithm, which can be used in adaptive sidelobe cancellers. In Section 3.4, we discuss the filtered-X filtered-ϵ algorithm, which is also used in adaptive sidelobe cancellers. In Section 3.5, we discuss the master-slave adaptive sidelobe canceller. Section 3.6 presents a master-slave Frost adaptive beamformer when the signal's direction is known. Finally, Section 3.7 describes coherent detection, and compares the results of these new schemes with the results of existing techniques.

3.1 Introduction

Spread-spectrum techniques are often used to neutralize large number of interferences and jammers from interfering with signal transmission. This spread-spectrum processing [3.1-3.3] encodes the signals in a manner which makes them resistant to unauthorized detection, demodulation, and interference. It is also well known that adaptive arrays are able to suppress directional jammers by forming spatial nulls in the direction of jammers. Adaptive arrays utilize spatial-domain information to discriminate between the desired signal and jammers. Spread-spectrum techniques utilize frequency-domain information to discriminate between the desired signal and jammers. In many anti-jamming applications, a combination of spread-spectrum processing and adaptive array processing, rather than either separately, constitutes the most robust and effective anti-jamming protection. In such situations, if spread-spectrum techniques effectively interface with adaptive array systems, they represent a noteworthy advantage.

Integration of spread-spectrum techniques with adaptive arrays has been reported by Compton [3.4], and Winter [3.5]. Digital communications are always employed in such a system. Compton applied the direct sequence spread-spectrum methods to Widrow's LMS adaptive arrays. The same experiments using direct sequence techniques are described by Winter, except that four phases are used to increase the data transmission rate.

3.1.1 Frequency-Hop Spread-Spectrum Techniques

One of the common spread-spectrum techniques utilizes frequency hopping. With the frequency-hop spread spectrum technique, the desired signal can be binary-phase-shift keyed (BPSK) or quadrature-phase-shift keyed (QPSK). The center frequency of the desired signal is hopping corresponding to a previously arranged pseudo-random (PN) code. For an authorized receiver, the signal can be easily recovered by tuning in accord with the known pseudo-random frequency schedule. To unauthorized receivers, the emanating signals looks like white noise. Only one frequency is used at a time. The resultant signal spectrum is spread over a large bandwidth, a bandwidth that is typically ten to several hundred times larger than the signal information bandwidth [3.6-3.7].

The fact that frequency hopping does not provide instantaneous coverage of the broad signal band leads to the consideration of the rate at which the hops occur. Clearly, the faster the hopping, the more nearly the frequency-hop approximates true spectrum spreading. Two basic characterizations of frequency hopping are fast frequency hop and slow frequency hop. These are distinguished from one another by the amount of time spent at each discrete frequency before hopping to the next. Sometimes the number of bits per hop is used to distinguish slow frequency hop from fast frequency hop. The two types of frequency hopping are briefly discussed below.

When slow frequency hop is employed, the carrier frequency remains constant for time periods far in excess of the time span of the data bits. This usually allows many data bits to be transmitted at each frequency, and the resulting transmitter and receiver equipment is simpler and less expensive than

that for a faster frequency hop. The disadvantage of slow frequency hop is that an enemy can implement smart jammers that defeat the anti-jamming protection in many instances. This can be accomplished by providing the jammer with a search receiver that scans the signal frequency band and locates the transmission; then the jammer's power can be concentrated at the frequency where the signal is being transmitted. If the jammer can adapt quickly enough, it may be able to follow the slow frequency hop.

For fast frequency hop, as the name implies, it involves very rapid returning of the signal and very short dwell time at each frequency. Generally, a fast hop is applied to defeat the smart jammer's attempt to measure signal frequency and tune the interference to the portion of the band. To defeat this tactic, the signal must be hopped to a new frequency before the jammer can complete its measurement and effect interference. Smart jammers, therefore, are forced to jam only a fraction of the total hopped band, since they only need to interfere with enough of the hops to decrease the SNR.

It is well known that signal cancellation effects exist in most of the conventional forms of adaptive array. This effect occurs when a jammer is sitting inside the signal's frequency band. This motivates us to use the frequency-hop spread-spectrum technique as an approach to the elimination of signal cancellation in adaptive arrays. In addition, combining the spatial-discrimination capability of adaptive arrays with frequency-hop techniques yields a system whose interference rejection capability is far greater than that of either of the two techniques used alone.

3.2 Overall System Structure

In this section we describe an overall system structure for frequency-hop adaptive arrays to provide a background of the whole system concept.

Figure 3.1 shows the overall system diagram. This system consists of an array of antenna elements, an adaptive array processor, a set of local oscillators, and a match filter. The match filter system consists of an "integrate and dump" and a decision maker, or a threshold detector. The antenna elements are omni-directional. The adaptive array processor is used to suppress the jammer and to receive the desired signal as well. Following the processor output, there is a set of local oscillators which mixes and decodes the signal from the array output. After mixing and decoding, the match filter is used to recover the signal's information.

Several assumptions are made for the frequency-hop adaptive array system. First of all, the desired signal is assumed to be a frequency-hopping signal. Second, digital data transmission is employed. Third, fast frequency hop is used. By this fast frequency hop, several frequency hops are possible during the time span of one data bit. The signal's direction may or may not be known.

The system can be used in many anti-jamming applications when the desired signal is known as a frequency-hop spread-spectrum signal. This will not work when signal cancellation occurs in adaptive array processors. New schemes for adaptive arrays are presented in the following sections.

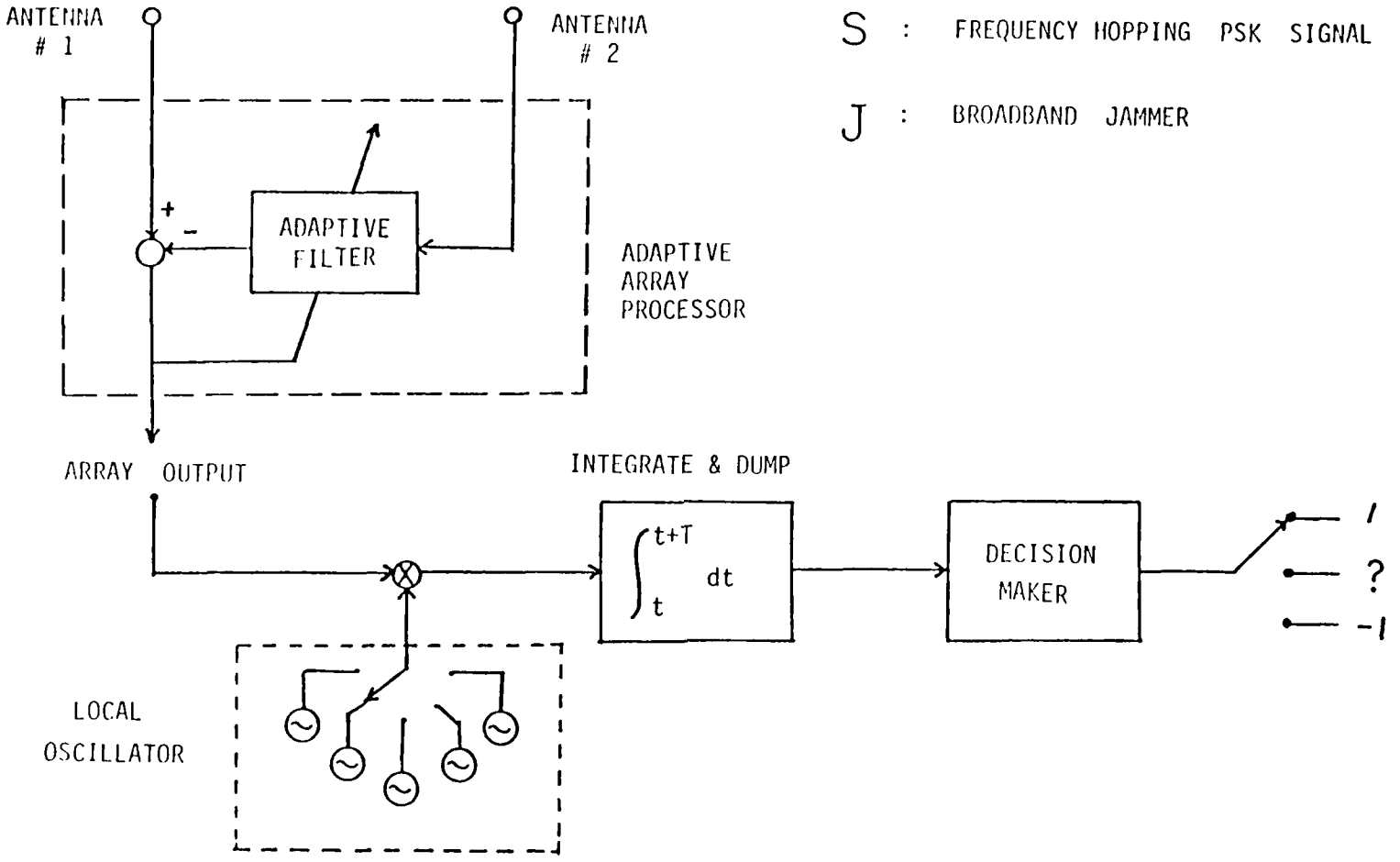


Figure 3.1. The overall system diagram for frequency-hop adaptive array.

3.3 Frequency-Hop Adaptation Algorithm

In this section, we propose a new adaptive array technique to prevent signal cancellation in frequency-hopped communication systems. This technique, termed the "frequency-hop adaptation algorithm," is applicable to both soft and hard-constraint adaptive arrays. The new algorithm utilizes a special filter structure which is called the frequency domain, complex, LMS adaptive filter.

3.3.1 Frequency Domain LMS Adaptive Filters

A conventional LMS adaptive filter is shown in Figure 3.2. The input goes through the tapped-delay line (TDL), and then is multiplied by the adjustable weights and summed to form the output. The weights adapt to match the desired response in a least-mean-square sense. Basically, this is a time-varying finite-impulse-response (FIR) filter. The frequency response of the filter depends on those weights of the tapped-delay line. Each weight has effectiveness over the entire frequency band. This filter structure, however, is not the most appropriate when only certain portion of the frequency response need to adapt.

Various structures and algorithms have been proposed for frequency-domain adaptive filters [3.8-3.11]. Among these filters, a structure suggested by Narayan [3.11] is very compatible with frequency-hop spread spectrum techniques. Horowitz and Senne [3.12] have applied frequency-domain filtering to adaptive array processing.

Figure 3.3 shows the diagram of Narayan's frequency-domain adaptive filter. The adaptive weights are complex, and are updated by the complex LMS

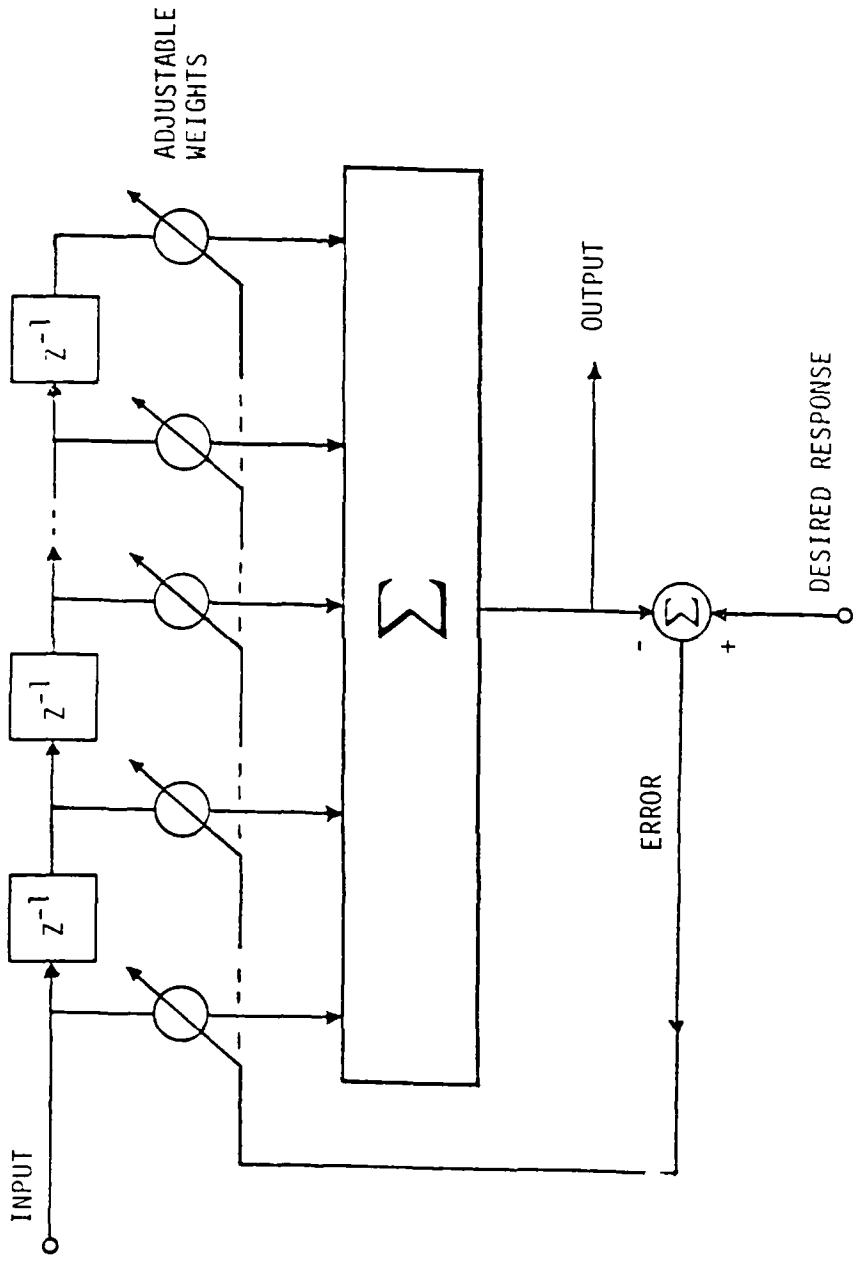


Figure 3.2. A conventional LMS adaptive filter.

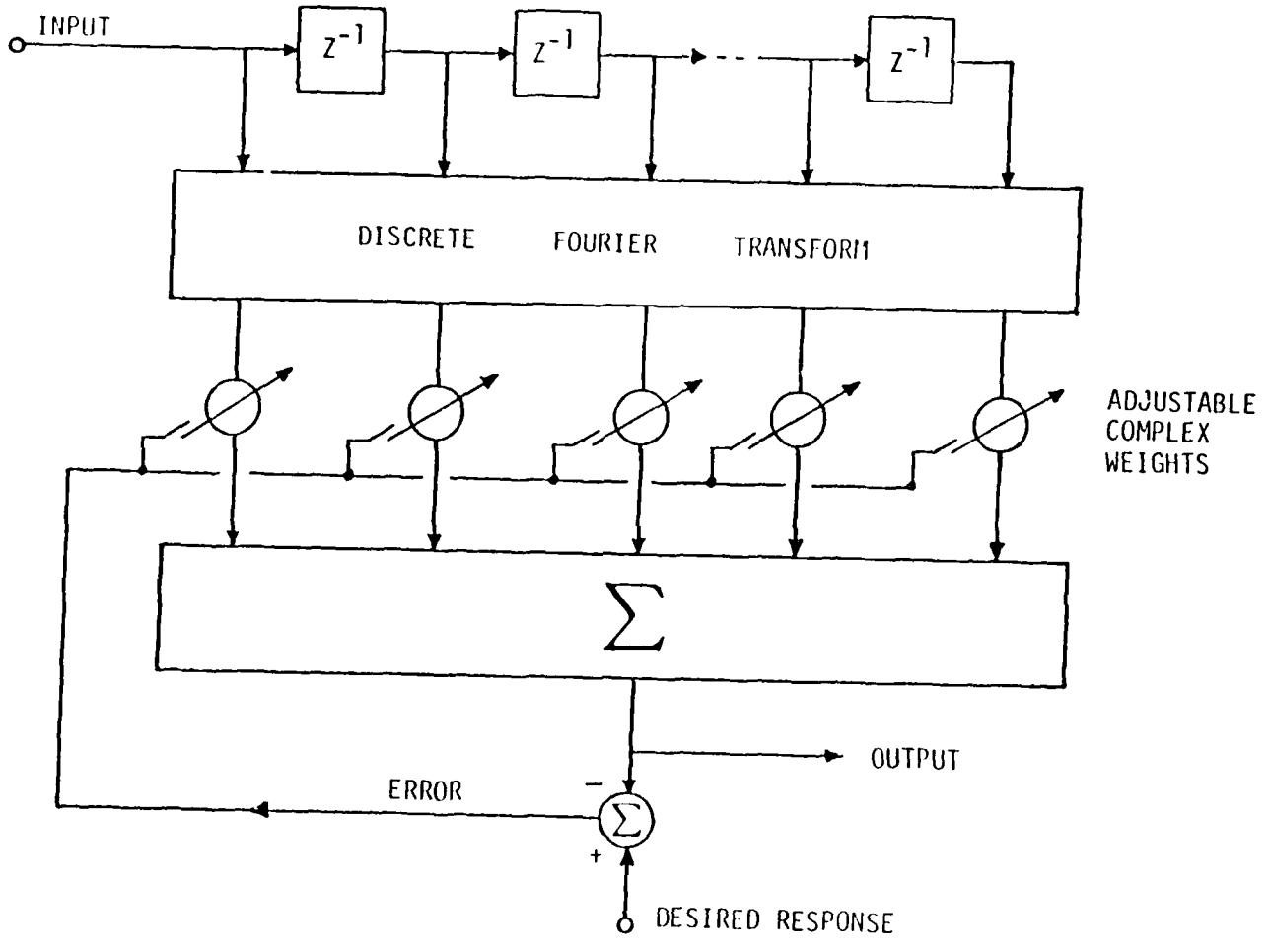


Figure 3.3. A frequency-domain LMS adaptive filter.

algorithm [2.7]. The input feeds through the tapped-delay line and undergoes the discrete Fourier transformation (DFT). The complex output bins of the DFT are weighted by adaptive weights and summed to give the filter output.

Comparing Figure 3.3 with Figure 3.2, it is very clear that the only difference between frequency-domain and conventional adaptive filters is the discrete Fourier transform in between the tapped-delay line and the adaptive weights. It is well known that the DFT can be interpreted as a bank of uniformly spaced band-pass filters. This implies that the weights in the frequency-domain LMS adaptive filter are adapting at the output of a bank of band-pass filters. Each individual weight now has the ability to control the gain and phase of the frequency response within a narrow range of its assigned frequency.

3.3.2 Frequency-Hop Adaptation Algorithm

With the background of frequency-domain adaptive filtering, we present an adaptive array scheme as shown in Figure 3.4. The antenna elements are omni-directional. Following the antenna elements is a frequency-domain adaptive filter. The system is operating with frequency-hopped BPSK signals in the presence of jammers. The desired signal is hopping from one frequency bin to another. Notice that the adaptive array scheme in Figure 3.4 will perform as a two-element Howells-Applebaum adaptive sidelobe canceller if the weights are adapted by the conventional LMS algorithm. This is true even though a frequency-domain LMS adaptive filter is used. Weiner and non-Weiner signal

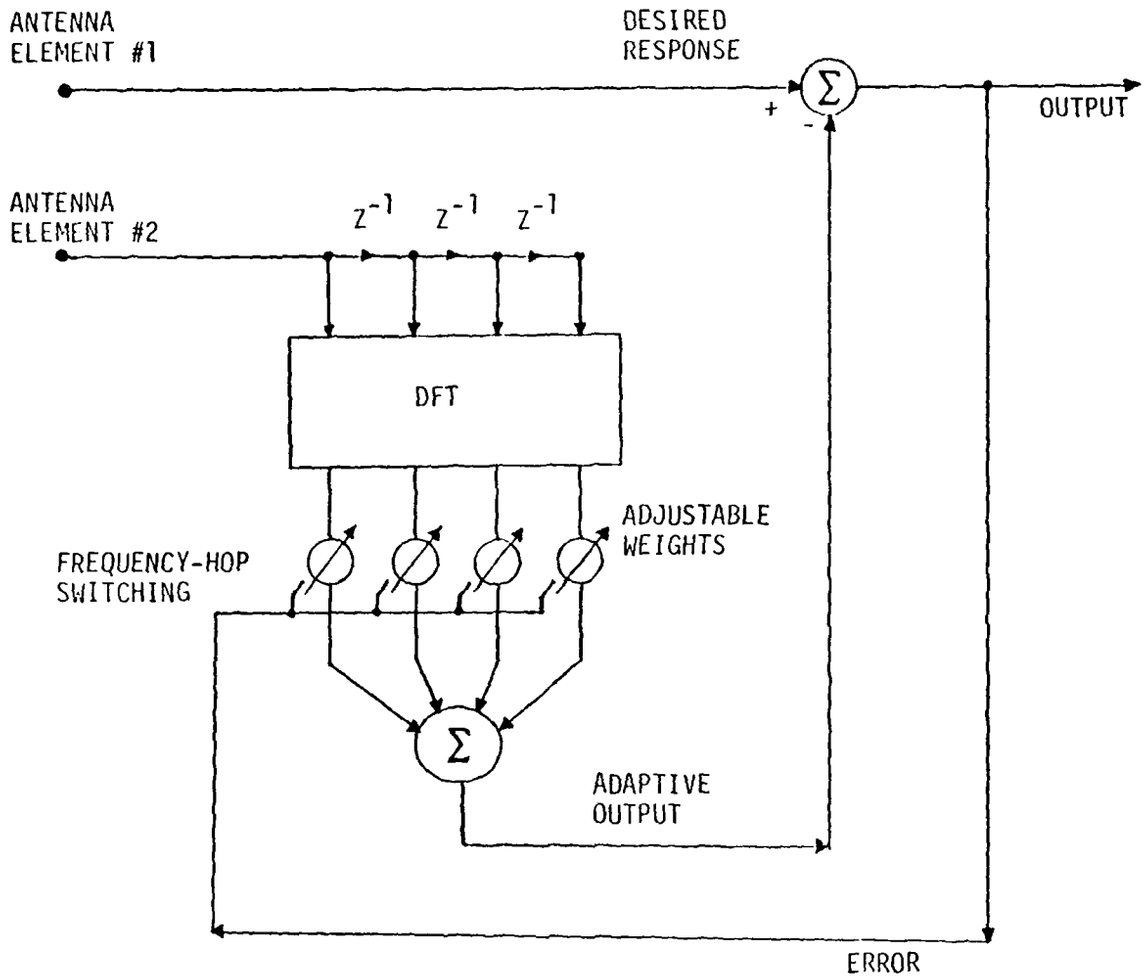


Figure 3.4. A frequency-hop adaptive array using *frequency-hop adaptation algorithm.*

cancellation still occur in such a Howells-Applebaum sidelobe canceller, and the weights are modulated during the adaptation process.

With the structure of the frequency-domain LMS adaptive filter, we are able to use a so called *frequency hop* adaptation algorithm. In this algorithm, the idea of eliminating signal cancellation is to stop the weights from modulation. Now the frequency-hop adaptation algorithm is given as follows: All frequency bins except the one which contains the current signal are adapted. Meanwhile, it "freezes" the weight which contains the current signal while adapting the rest of weights to minimize the array output power. This contrasts with existing techniques which adapt all complex weights corresponding to all the frequency bins.

Mathematically, the weights of frequency-hop adaptation algorithm can be described as

$$W(k+1) = W(k) + 2 \cdot \mu \cdot \epsilon(k) \cdot A(k) \cdot X(k) \quad , \tag{3.1}$$

where $A(k)$ is a time-varying diagonal matrix and can be expressed as follows,

$$A(k) = \begin{bmatrix} 1 & 0 & \cdot & \cdot \\ 0 & 1 & \cdot & \cdot \\ \cdot & \cdot & 0 & \cdot \\ \cdot & \cdot & \cdot & 1 \end{bmatrix} \quad .$$

Note that all but one of the diagonal entries of $A(k)$ are unity. The position of the zero diagonal entry is hopping according to the signal's PN frequency code.

In the algorithm, only the selected complex weight corresponding to the current instantaneous frequency is temporarily frozen, and the rest of the complex weights are adapted in a conventional way to minimize the noise, to null

jammers in the sense of mean square error. Since the desired signal always appears through a frozen (not adapting at that moment) weight, the weight modulation effect will never act on the signal. Consequently, degrees of freedom in the weights are never used to cancel the desired signals. Thus signal cancellation can be prevented. And since the adaptive process adapts all frequency bins except one at a time, inhibiting adaptation at the current frequency bin will not significantly slow the rate of convergence.

It remains to demonstrate the effectiveness in suppressing the jammer. This is easily understood from the point of view that the desired signal is transparent to the frequency-hop adaptation algorithm. No matter whether the desired signal is present or not, all the weights on the average will still adapt to minimize the output power. In the situation when the jammer is present only, it works just as a conventional adaptive sidelobe canceller.

We leave the simulations to Section 3.5 and 3.6, since many of them are similar in terms of sensitivity in beam pattern and frequency response.

3.3.3 Discussion and Conclusions

The frequency-hop technique is very compatible with Narayen's frequency domain LMS algorithm when adapted in accord with "frequency-hop adaptation algorithm" as given in (3.1). Basically, this algorithm is a filtered-X LMS algorithm. The algorithm makes use of the frequency-domain adjustability to freeze the adaptive weight which corresponds to the desired signal frequency bin. By freezing the selected weight, the modulation effect on the signal is stopped.

and signal cancellation thus can be eliminated.

There are several points that should be taken into account. First of all, the array output is directly used as the feed-back error in the adaptation process. Minimizing the output power is the performance criterion. This however introduces a performance limitation associated with signal power. When a high-power signal is present, this signal will feed back to the adaptive processor, and cause a high misadjustment which corresponds to noisy weights. This results in a noisy recovered signal.

Second, the DFT always introduces inherent leakage effects, *i.e.*, the energy in the main band of the frequency response "leaks" into the sidebands, obscuring and distorting other sidebands responses. Besides, the frequency-band resolution of the DFT is limited by the length of the window. Normally, in frequency hop environments, thousands of frequency bands are required. This implies that the number of taps in the tapped delay line, and hence the number point of DFT, should be at least around a few thousand for a good frequency resolution. If so, the DFT processor will be expensive and complicated. Also the transient response due to the associated long tapped delay line will become very critical when a high hopping rate for the signal is desired. Chapter IV will relate more details about leakage effect, frequency resolution, and transient performance.

3.4 Filtered-X Filtered-ϵ LMS Algorithm

In this section, we present a second adaptive array technique, termed the filtered-X filtered-ϵ LMS algorithm, to combat signal cancellation in soft-constraint adaptive arrays. This algorithm shows that signal power will have no effect on the weight setting. The new algorithm shares common merits with the so called "instrumental variable" (IV) method in the field of recursive identification [3.13]. In this scheme we also introduce a filter called the frequency-hop notch filter. Ideally, this filter should have a flat frequency response and a linear phase except that it can form notches at the specific frequencies in accord with a known frequency code.

3.4.1 Structure and Algorithm

Figure 3.5 shows a modified Howells-Applebaum sidelobe canceller for use with the "filtered-X, filtered-ϵ LMS algorithm." Again the antenna elements are omni-directional, and will receive the signal as well as the jammer. The desired signal is a frequency-hopping signal. Only one frequency is used at a time, and the signal direction may or may not be known.

The key idea embodied in this modified sidelobe canceller is the removal of the desired signal from the adaptation feedback (the error). Since the desired signal is hopping from bin to bin, a signal-free error can be formed by filtering the array output through a frequency-hop notch filter. This signal-free error is then used as a measure of the performance criterion. Due to the removal of desired signal from adaptation feedback, signal power should have little or no

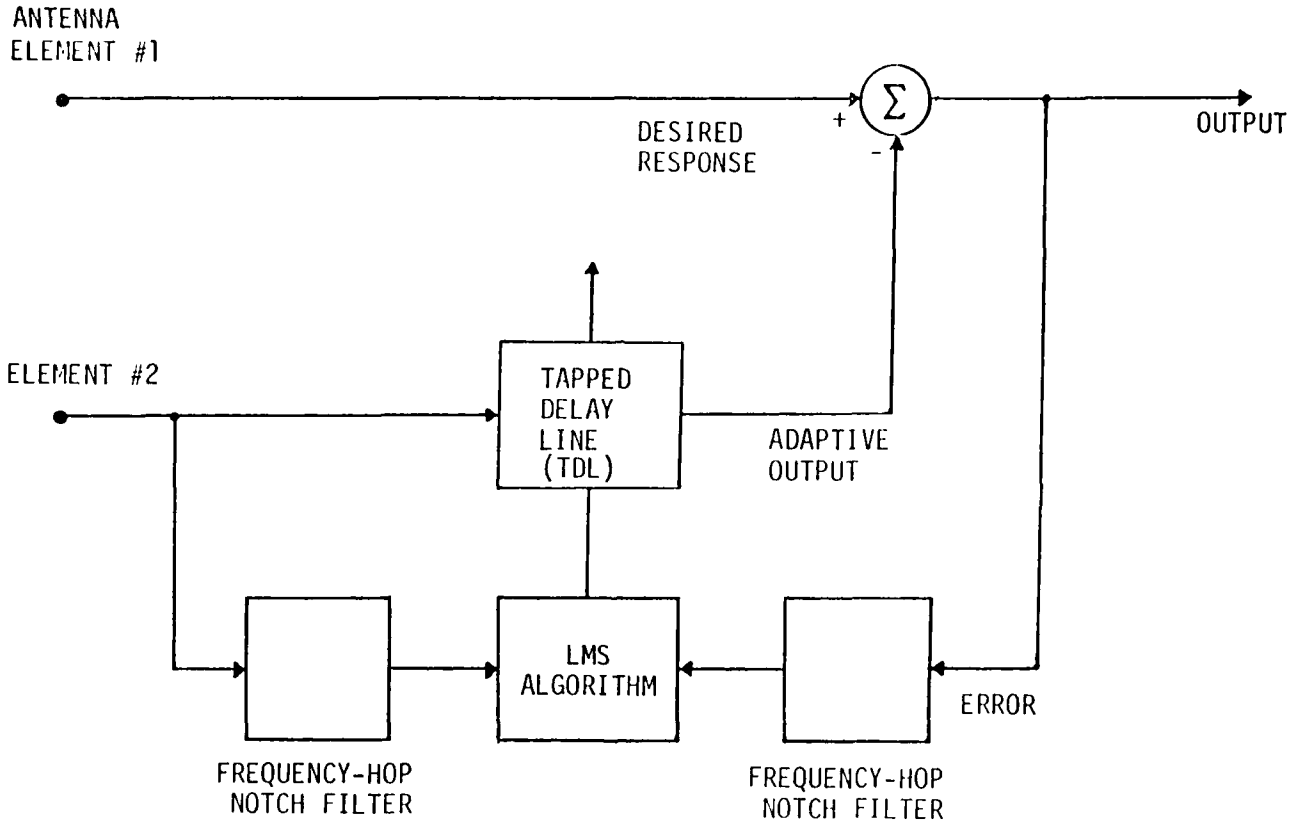


Figure 3.5. A frequency-hop adaptive array using filtered-X filtered-ε LMS algorithm.

effect on the optimal solution of the weights. Direct application of the LMS algorithm using this filtered error to update the weight vector may result in instability [3.14]. To avoid such instability, the X vector is filtered in the same manner as the error, and this provides the same phase delay needed by the LMS algorithm.

As the usual way with a Howells-Applebaum sidelobe canceller, the jammer and the signal components from element #2 will go through the tapped-delay line as well as the frequency-hop notch filter. The error is also obtained by notch-filtering the array output. The weights of the tapped-delay line are adapted by a modified LMS algorithm, which is referred as the "filtered-X, filtered-ϵ LMS algorithm." A conventional LMS algorithm generates a filter output as the inner product of the current weight vector and the current signal vector, and then updates the next adaptive weight vector by using the current error and the current signal vector. For the "filtered-X, filtered-ϵ LMS algorithm," the adaptive filter output is the same inner product of the current weight vector and the current signal vector, except that the next weight vector is updated by the filtered error and the filtered signal vector.

Mathematically, the conventional LMS algorithm updates the weights vector as follows,

$$W(k+1) = W(k) + 2 \cdot \mu \cdot \epsilon(k) \cdot X(k) \quad .$$

The filtered-X, filtered-ϵ LMS algorithm updates the weights vector according to

$$W(k+1) = W(k) + 2 \mu \cdot \epsilon'(k) \cdot X'(k) \quad .$$

where $X'(k)$ represents the filtered X vector and $\epsilon'(k)$ represents the filtered error. The output is the same inner product of the weight vector W and the unfiltered X vector, *i.e.*,

$$\mathbf{y}(k) = W^T(k)X(k) \quad .$$

This modified algorithm was first presented by Widrow *et.al.* [3.15-3.16], and is somewhat similar to the "instrumental variable" (IV) method [3.13]. The IV method decorrelates an estimate of X vector and the system noise so as to overcome the convergence problems in recursive identifications. The detailed block diagram of the "filtered-X, filtered-ϵ LMS algorithm" is shown in Figure 3.6.

We need to demonstrate that this modified algorithm will still be capable of removing jammers from the array output. Assume that the spectrum of the jammer is constant relative to the time-variation of the frequency-hop desired signal. Suppose a given frequency bin of the "unfiltered" error contains the jammer plus an occasional burst of desired signal. The same given frequency bin of the "filtered" error thus will contain jammer alone when the frequency-hop notch filter is tuned elsewhere and will be zero during the time when the notch filter is tuned to that bin. On the average, however, the jammer will be present in that frequency bin and will make itself apparent to the adaptive process in which the goal is to reduce the power of the filtered error.

The frequency-hop notch filters are synchronously notching at the signal's frequency. Since unbalanced delay of filtering might result in instability in the LMS adaptive algorithm, the same filtering is then required to balance delay and

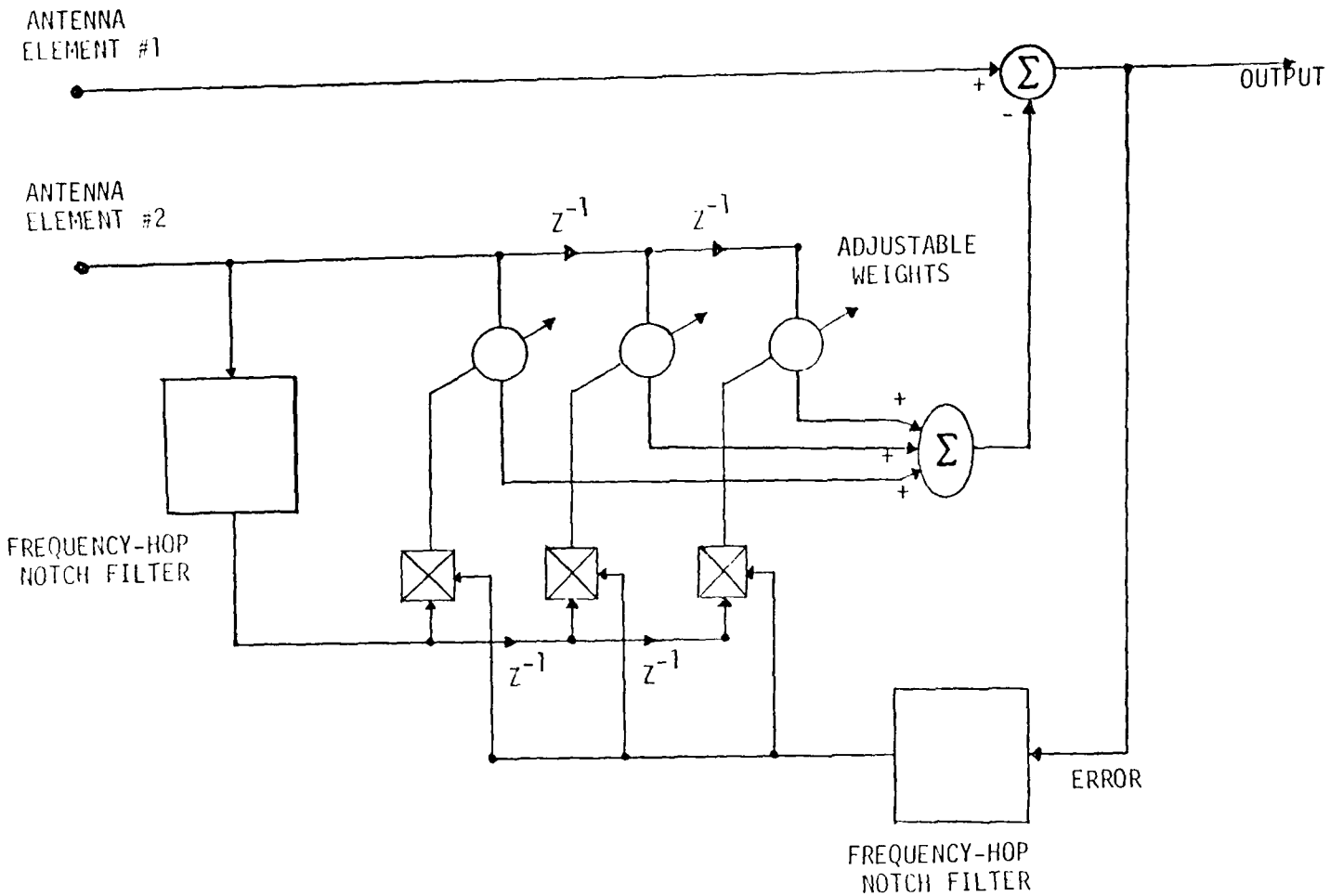


Figure 3.6. The detailed block diagram of the filtered-X filtered-ε LMS algorithm.

to stablize the LMS algorithm. More details about the structure of frequency-hop notch filters are left for Chapter IV.

Notice that the desired signal components are all notched out from the error and the element #2 by the frequency-hop notch filters, and the jammers and white input noise will control those adaptive weights of the tapped delay line. This implies that signal power would have no effect on the adaptive weights. The signal and jammer will not interact with each other during the adaptation process. This prevents signal cancellation phenomena of both Wiener and non-Wiener types in conventional Howells-Applebaum sidelobe cancellers. From the aspect of the sensitivity pattern, the adaptive array should create nulls in the directions of jammers. The beam pattern corresponding to the Wiener solution would be the same, with and without the signal.

3.4.2 Discussion and Conclusions

In frequency-hop spread spectrum systems, normally high-Q and frequency-tunable filters are used to reduce jamming effects outside the desired signal's frequency band. This kind of filtering, however, would not reduce the jammer power level inside the signal's frequency band. Adaptive arrays can attenuate directional jammers, but they can also cause signal cancellation problems. The modified scheme in this section is able to reduce the jammer in the signal band by forming a deep null in the direction of the jammer. In addition, this scheme also helps in preventing signal cancellation.

By separating the desired signal from the jammer during the adaptation

process, the existing signal cancellation problems with both the Weiner and non-Wiener solutions in a Howells-Applebaum sidelobe cancelling array can be essentially eliminated.

There are two remarks about the filter-X, filtered-ϵ LMS algorithm. First of all, the desired signal is removed from the vector X and the error, hence the weights are dominated by jammers and white noises only. Second, the same notch filtering structures are used to provide phase delay balance when employing this special algorithm.

3.5 Master-Slave Adaptive Sidelobe Canceller

In this section, we introduce another adaptive array technique that utilizes conventional LMS algorithm and frequency-hop nature of the desired signal to prevent signal cancellation. We call this scheme the master-slave adaptive sidelobe canceller. The basic idea is to remove the desired signal from the inputs of adaptive processor (known as the master processor). This is accomplished by prefiltering the signal at each array element with a frequency-hop notch filter, notching in accord with the signal's frequency code. Because the notch filter is continuously hopping from bin to bin, jammers will pass through to the master processor inputs and the adaptive algorithm will attempt to eliminate them. The set of weights derived from this master processor are then copied into a slave processor. The slave processor containing the desired signal as well as the jammers will recover the desired signal while simultaneously suppressing the jammers. Provided the prefiltering is done identically on each element of the array, nulls formed in the slave processor will be in the same direction as nulls formed in the master processor.

3.5.1 Structure and Algorithm

Figure 3.7 shows a block diagram of the master-slave adaptive sidelobe canceller. This scheme consists of two separate processors; a master processor operating on prefiltered array signals and a slave processor operating on the original array signals. The two frequency-hop notch filters have the same structures and are "hopping" in a manner that rejects the desired signal while

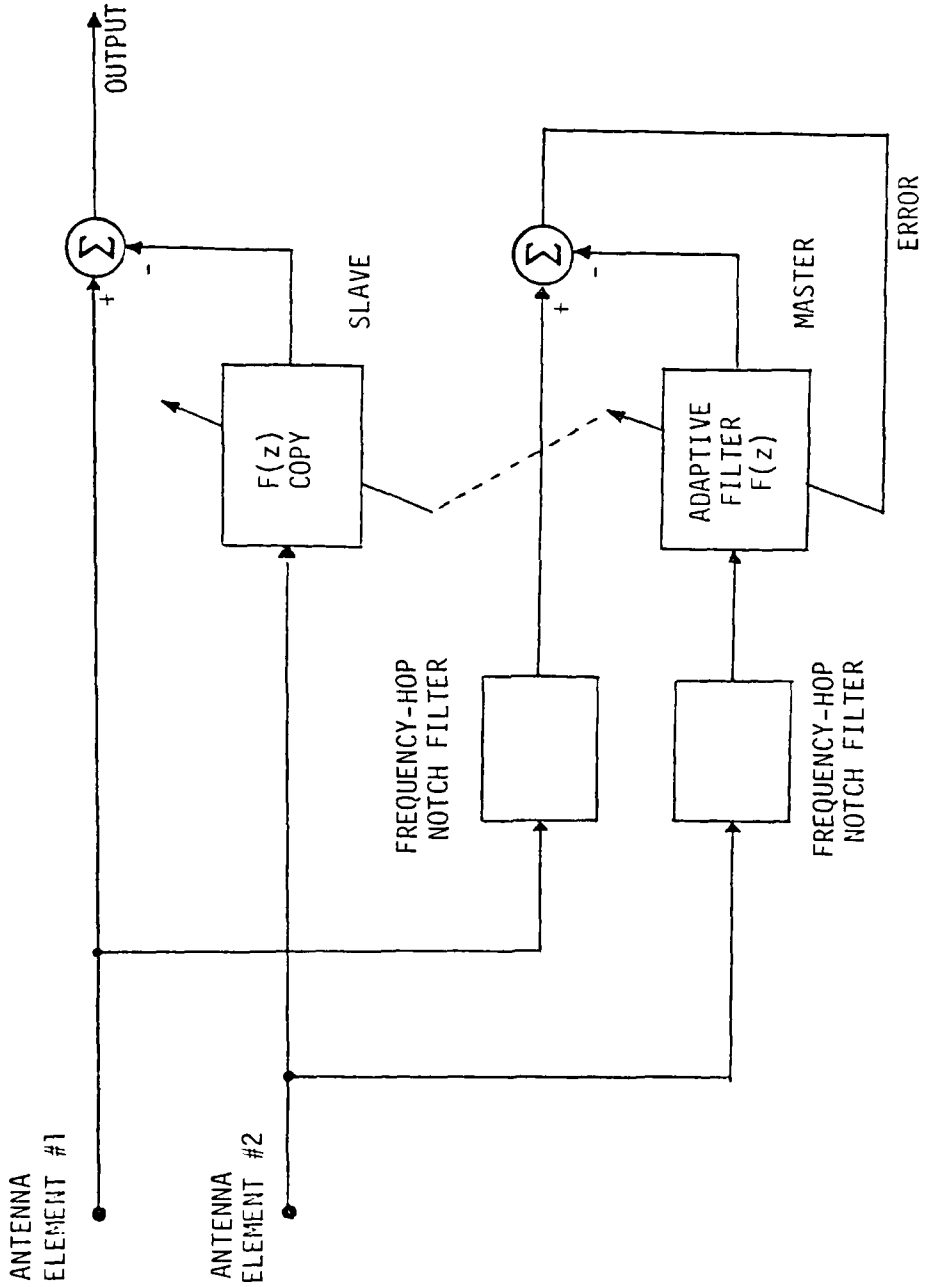


Figure 3.7. An adaptive sidelobe canceller using frequency-hop notch prefiltering.

passing out-of-band signals. This filtering operation removes the desired signal from the adaptive master processor inputs. The output of the master processor will contain no desired signal and is used only in the adaptation feedback. The weights derived from the master processor are then copied into the slave processor. The inputs of the slave processor come directly from the array elements and are used to generate the useful array output.

Because the desired signal has been removed from the master processor, it can no longer affect the adaptive weights. Non-Weiner signal cancellation results from an interaction between the desired signal and jammers during the adaptive process. Obviously removal of the desired signal from the adaptive algorithm will eliminate non-Weiner signal cancellation. Weiner signal cancellation results from the inability of the adaptive algorithm to distinguish between the desired signal and the jammer. A high-power desired signal would be nulled as if it were a jammer. A Howells-Applebaum sidelobe canceller will experience this type of signal cancellation. By removing the desired signal from the inputs to the adaptive process, both Weiner and non-Weiner signal cancellation can be eliminated.

It remains to demonstrate that the presence of frequency-hop notch prefiltering will not degrade the ability of the adaptive array to null jammers. This is most easily demonstrated by visualizing an equivalent jamming scenario. Assume for the moment that the only signals received by the adaptive array originate from either the jammers or the desired signal. That is, assume there is no ambient thermal noise. Then the effect of the frequency-hop notch filters can

be "pushed" into the source of both the jammers and the desired signal to generate the equivalent signal environment shown in Figure 3.8. The notch filters have been moved from the processor inputs to each directional signal source. Since the frequency notch hops from bin to bin in accord with the hopping of the desired signal, the desired signal is eliminated from the conceptual signal environment as indicated by the large X. The equivalent jammer is generated by filtering the original jammer through the frequency-hop notch filter as shown. Since the adaptive weights in the master processor are affected only by this equivalent jammer, conceptually the master processor will see no desired signal present. By this means it aids all Howells-Applebaum sidelobe cancellers, since the criterion of power minimization applies only to the jammers but not to the desired signals.

A typical frequency-hop spread spectrum signal will hop among 100 different frequency bins spending approximately 10 μsec (the chip duration) in each bin [3.6]. On the average the frequency-hop signal will spent only 1% of its time in any given frequency bin. Provided the time constant of the adaptive algorithm is chosen to be at least several times the chip duration, (for example, 100μsec), the average effect of the notch filter is to slightly reduce the apparent power of the jammer. If the time constant of the adaptive algorithm is comparable to or smaller than the chip duration, then the apparent jammer will be nonstationary and the adaptive weights will try and track the time varying situation.

Figure 3.9 shows an improved methods for generating the overall system

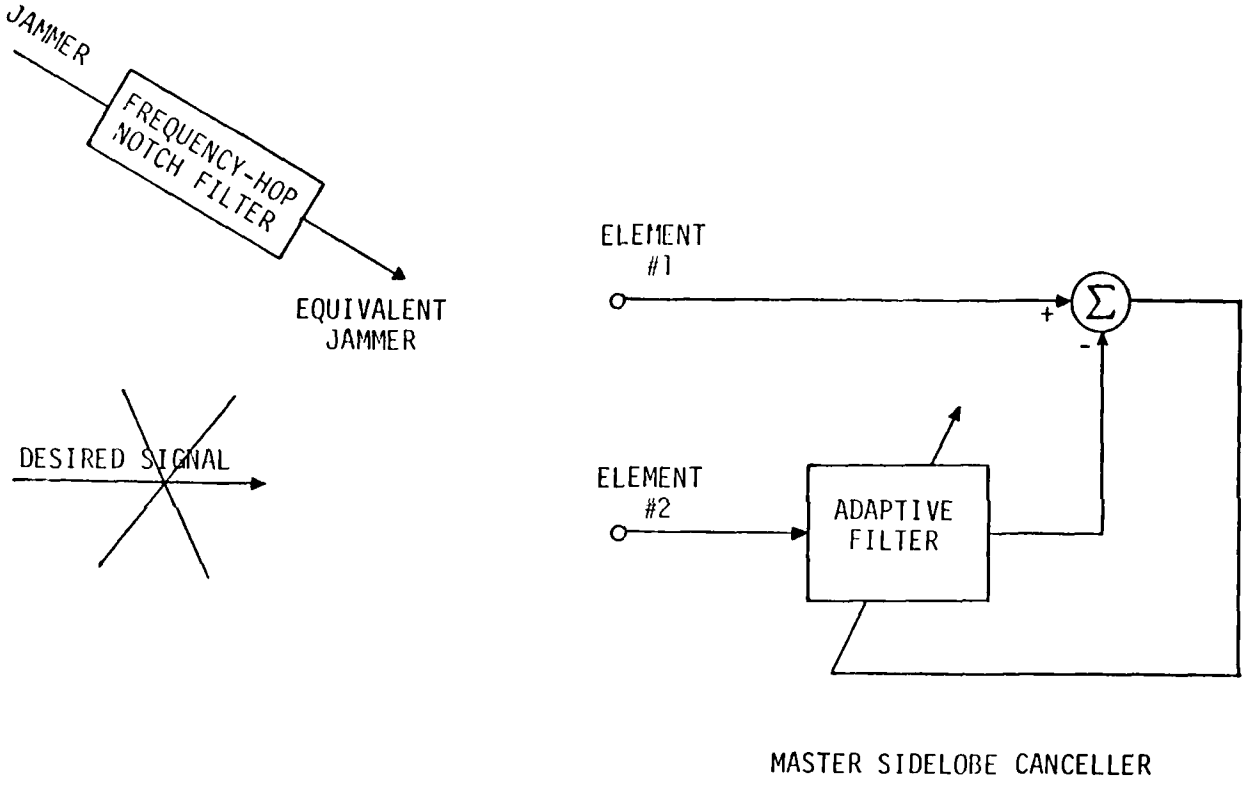


Figure 3.8. A conceptual system diagram generated by pushing the notch filter into the directional impinging sources.

output. Instead of taking the output of the slave processor alone, the difference between the output of the slave and the master is taken. Generating the array output in this manner helps reduce the out-of-band jammer components and is essentially equivalent to filtering the slave processor output with a bandpass filter centered at the current frequency-hop bin, as shown in Figure 3.10. This is best explained by noting that the output of master processor contains a residual amount of the jammer component outside the signal band while the output of slave processor contains almost the same amount of out-of-band jammer residual in addition to the desired signal. By subtracting the two outputs, the out-of-band jammers in the master processor output cancels the out-of-band jammers in the slave processor output, effectively creating a bandpass filter on the output of the slave processor, as illustrated in Figure 3.10.

3.5.2 Simulation Results

Simulations of the two-element adaptive array shown in Figure 3.7 and Figure 3.9 have been conducted. Assume that a BPSK frequency-hop signal is emanating from one direction, and a broadband jammer from another direction. A broadband jammer was generated by passing uncorrelated noise through a Butterworth bandpass filter. The notch filters are tuned to the center frequency of the desired signal.

For the first experiment, the power of the jammer was set to 100 and the power of the desired signal was set to 1. Also, the desired signal was "frozen" to one frequency, that is it was not allowed to hop. Power spectra at various stages

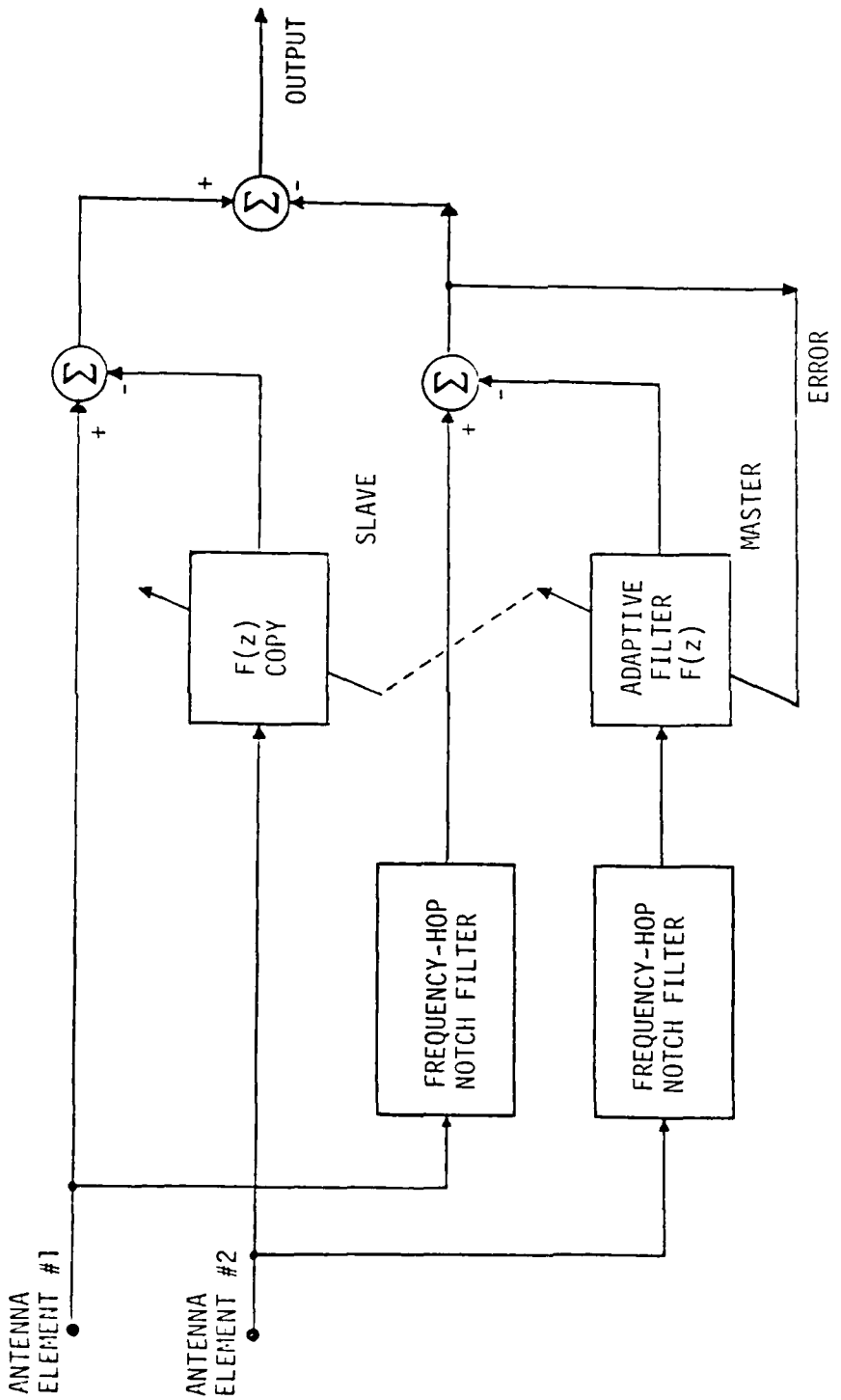


Figure 3.9. An improved method for generating the useful system output.

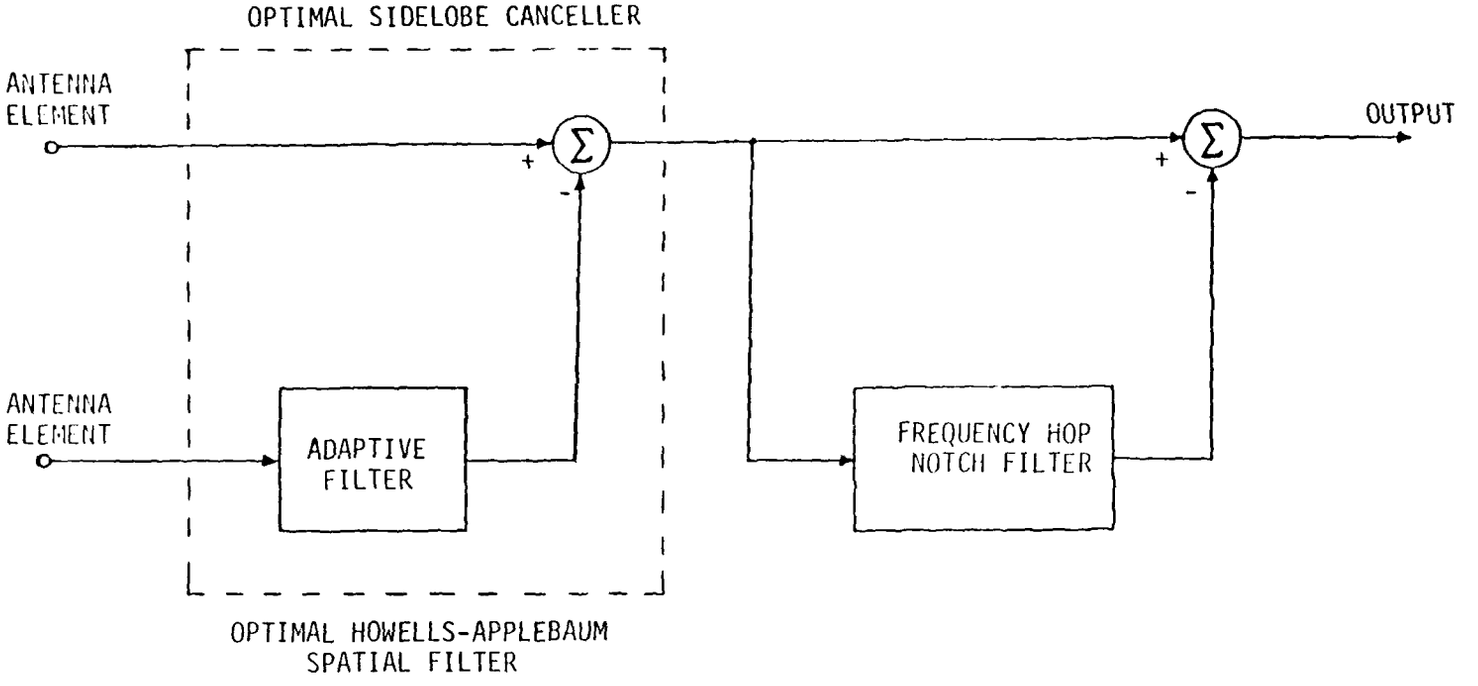


Figure 3.10. An equivalent scheme of the improved sidelobe canceller when the master processor reaches steady state.

of the processor are shown in Figure 3.11. The top plot shows the spectrum of the signal received by one array element. Without spatial processing, this input signal is dominated by the jammer and the sinusoidal signal can not be seen. The middle plot shows the spectrum at the output of the master processor after convergence and the bottom plot shows the spectrum of the output of the slave processor. Notice that the sinusoidal desired signal is well above the background interference level in the slave processor output, whereas this desired signal has been notched out of the master output. In addition, the interference has been greatly reduced by the 10-to-1 scale difference of the input signal and master output spectral plots.

Figure 3.12 shows the time domain waveforms at the slave processor output and the output after differencing the master and slave processor output. For this experiment the desired signal was not frozen and was giving rise to the three distinct frequency hops shown. At the slave processor output, the periodic nature of the desired signal can been seen but it appear to be a rather noisy sinusoid. After subtraction however, almost all of the out-of-band jammer has been removed and the sinusoidal desired signal appears very clean.

Apparently, identical structures are required for those two frequency-hop notch filters to preserve the relative phase. The nulls formed in master processor will be the same as nulls formed in slave processor.

Assume for the moment that the frequency-hop notch filters are not present in Figure 3.7. In this case, if the desired signal was of a power much greater than that of the jammer sources, the adaptive array would form a null in the direction

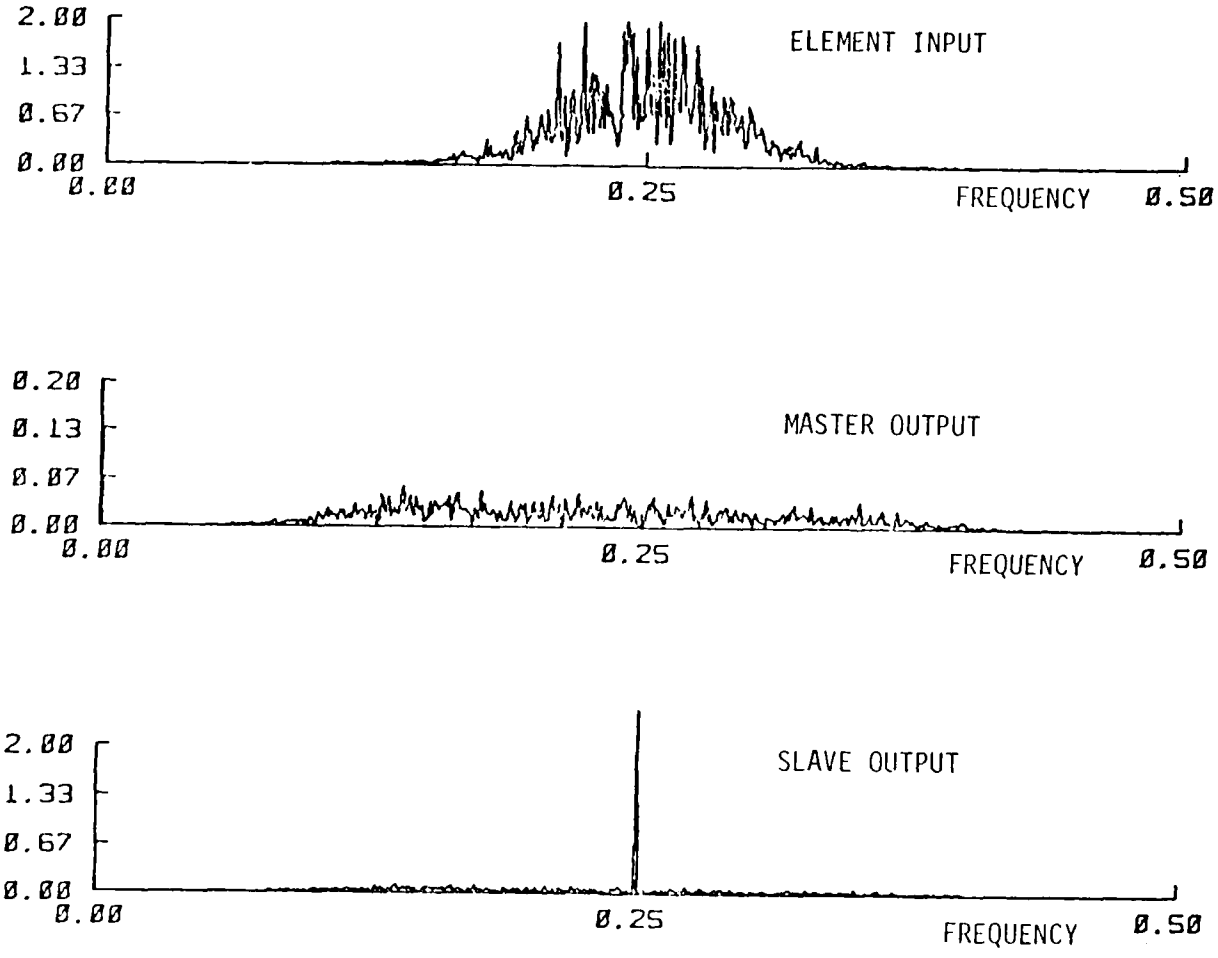


Figure 3.11. Power spectra corresponding to the element input, the master output, and the slave output.

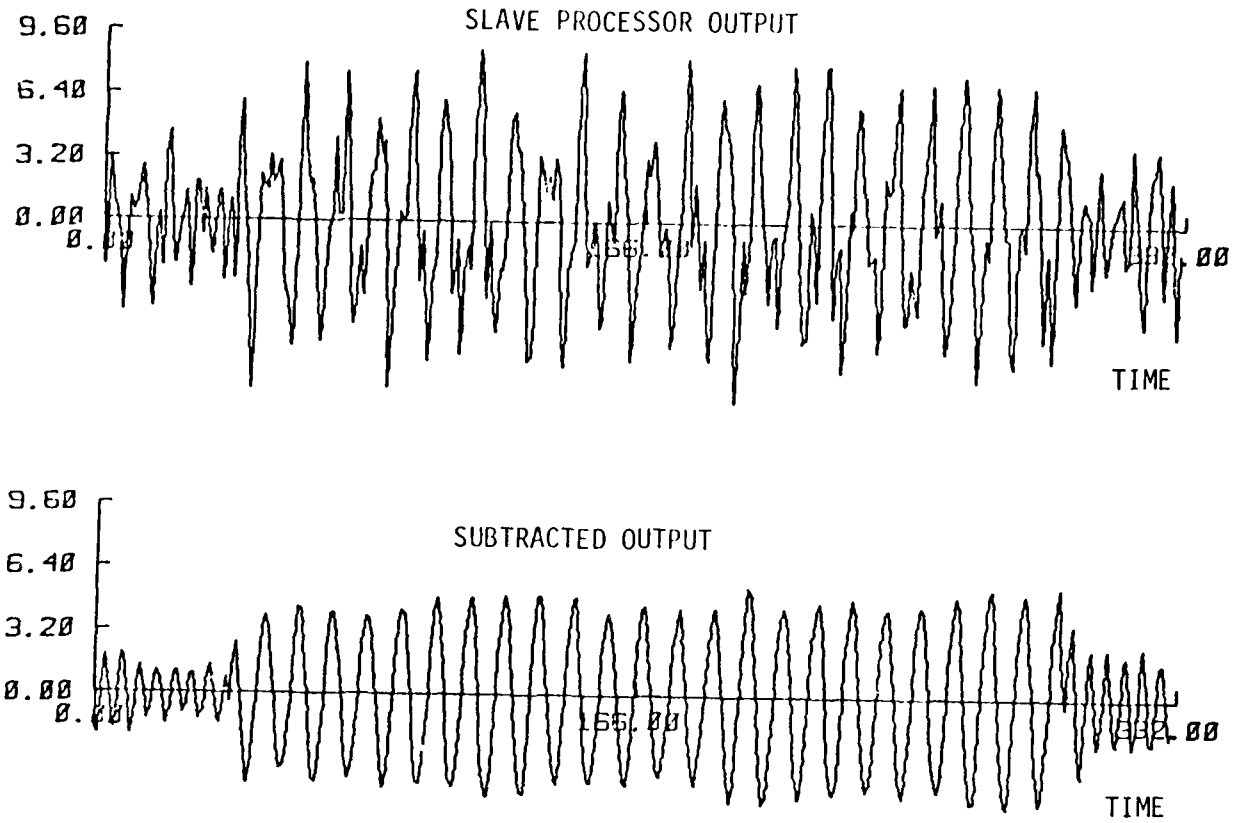


Figure 3.12. Time domain waveforms at the output of the slave processor and after subtraction of the master and the slave outputs.

of the desired signal. However, with the notch prefilters present, the desired signal is removed from the input to the adaptive master processor and therefore the power of the desired signal can not affect the weights. To illustrate this, two experiments were conducted each with identical parameters except for the signal power. In one experiment the signal power was set to 1 while the other the signal power was set to 40. Again, the desired signal was frozen to one frequency. After convergence, the array beam patterns were evaluated at the frequency of the desired signal and plotted in Figure 3.13. As anticipated, the two sensitivity patterns are identical.

To illustrate the wideband performance of the adaptive array, Figure 3.14 shows the array's frequency response in the direction of both the signal and the jammer. The horizontal axis is normalized frequency. The desired signal was hopping among 128 different frequency bins and covered a normalized frequency range between 0.15 and 0.35. Notice that the response in the signal direction is fairly flat over this bandwidth and the response in the jammer direction is virtually zero over the jammer bandwidth.

For the above experiment, the time constant of the adaptive algorithm was set to several times the bit duration. Thus the adaptive weights converged to a constant value. Had the time constant been set to a value less than the bit duration, the weights would have attempted to track the time-varying jammer spectrum. At any given instant, the frequency response in the jammer direction would have been virtually zero over the bandwidth of the jammer except possibly in the current frequency-hop bin. This could occur because the notch filters

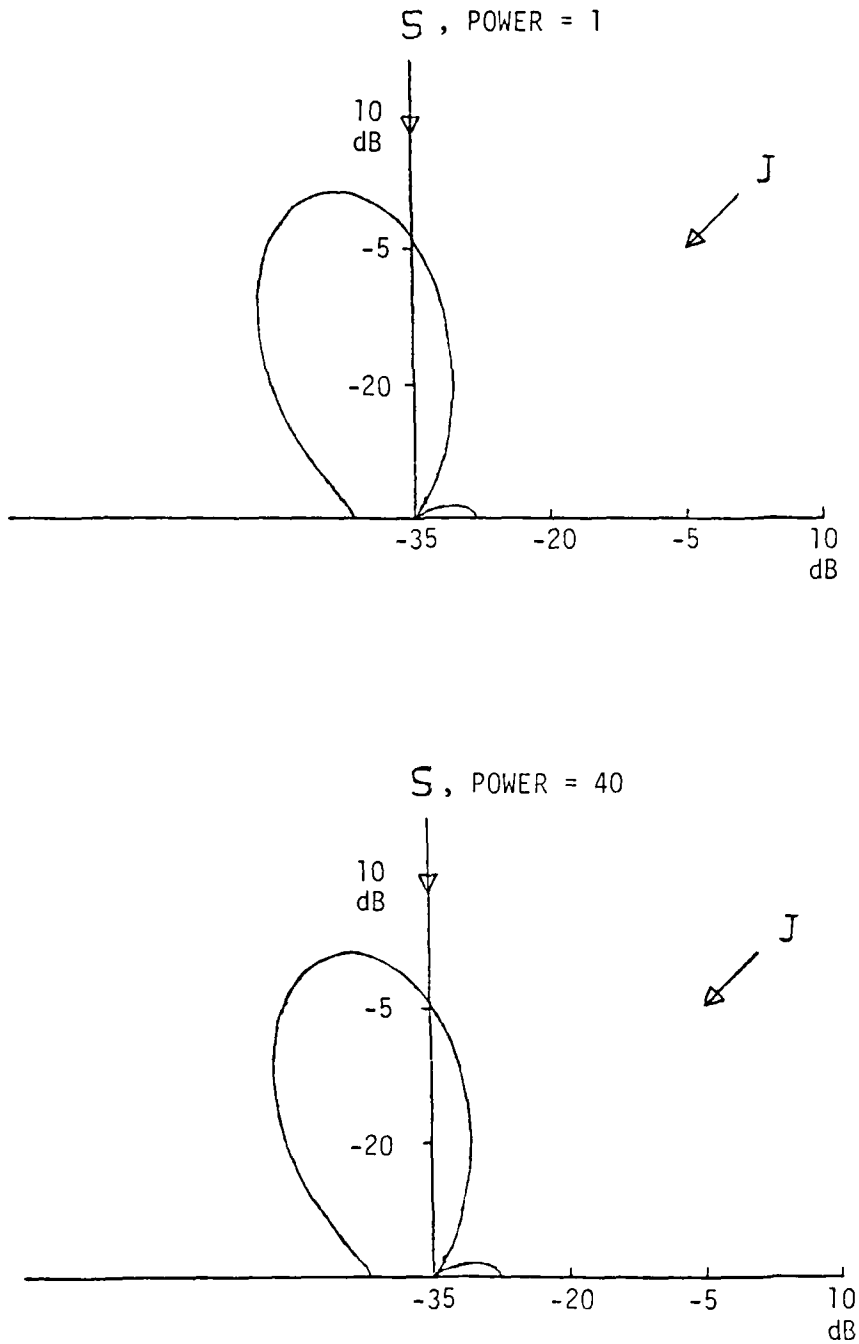
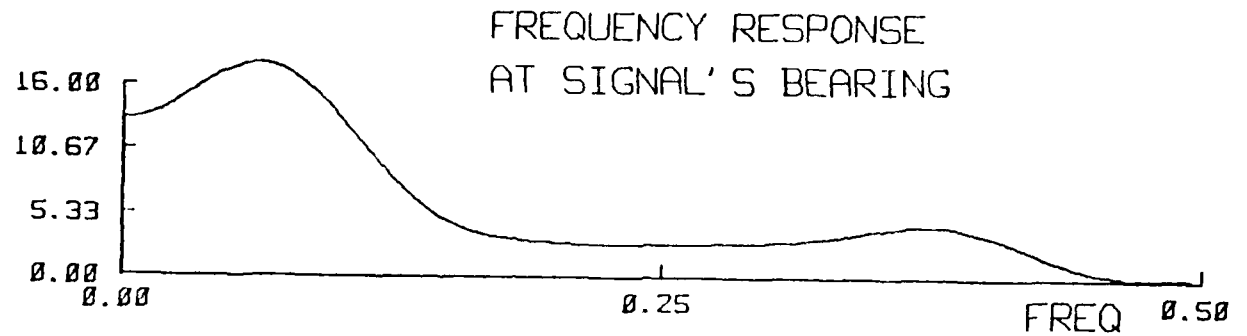


Figure 3.13. Comparison of the array beam pattern for two different power levels of the signal.



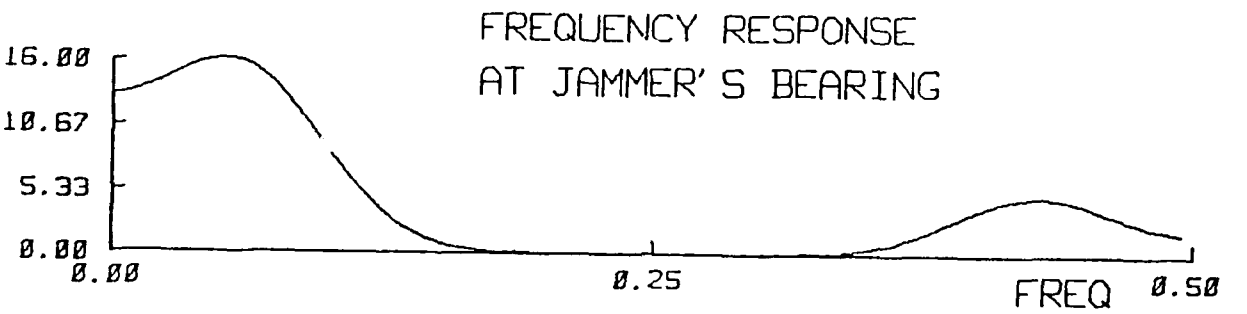


Figure 3.14. The frequency responses in the desired signal direction and in the jammer direction.

remove any jammer component in the current frequency-hop bin making them invisible to the master adaptive processor. However, since the bandwidth of the notch filters is very narrow compared to the bandwidth of the jammer and the array's frequency response prefers to be continuous, the response in the frequency-hop bin should be about the same as the response immediate outside the bin. Thus a wideband null is formed in the direction of the jammer over the entire jammer bandwidth.

3.5.3 Discussion and Conclusions

The master-slave adaptive sidelobe canceller can eliminate signal/jammer interactions by using frequency-hop notching. Conventional LMS algorithm is used in this scheme. The improved scheme results in better frequency filtering. It can also be decomposed into two filtering stages; the first spatial filtering, the second frequency filtering. This scheme still results in a soft-constraint adaptive array. The gain in the signal's direction is determined by the jamming situation.

3.6 Master-Slave Frost Adaptive Beamformer

In this section we propose a master-slave Frost adaptive beamformer to prevent signal cancellation in adaptive arrays. The key idea is to remove the desired signal from the master adaptive beamformer by frequency-hop notch filtering. This scheme basically is similar to the scheme in previous section, except that it can retain an assigned gain or a linear constraint in the look direction.

Figure 3.15 illustrates a block diagram of the master-slave Frost adaptive beamformer. There are two identical Frost beamformers used in this scheme. One is the master, and the other is the slave. The master beamformer copies the weights into the slave one. The Frost beamformer imposes a linear constraint in the desired look direction. The array elements received the jammer as well as the desired signal. A bank of frequency-hop notch filters are used to notch out the desired signal from the receiving array elements.

Since the jammers are transparent to the frequency-hop notch filters, they will pass through to the master beamformer inputs and this master beamformer will attempt to eliminate the jammers. By copying the weights from the master into the slave beamformer, this slave beamformer containing signal and jammers will recover the signal while simultaneously suppressing the jammers. The look-direction constraints are sustained as usual, and the jammers are nulled. By removing the desired signal from the master beamformer, signal cancellation can be eliminated.

Simulations for a two-element master-slave Frost adaptive beamformer are

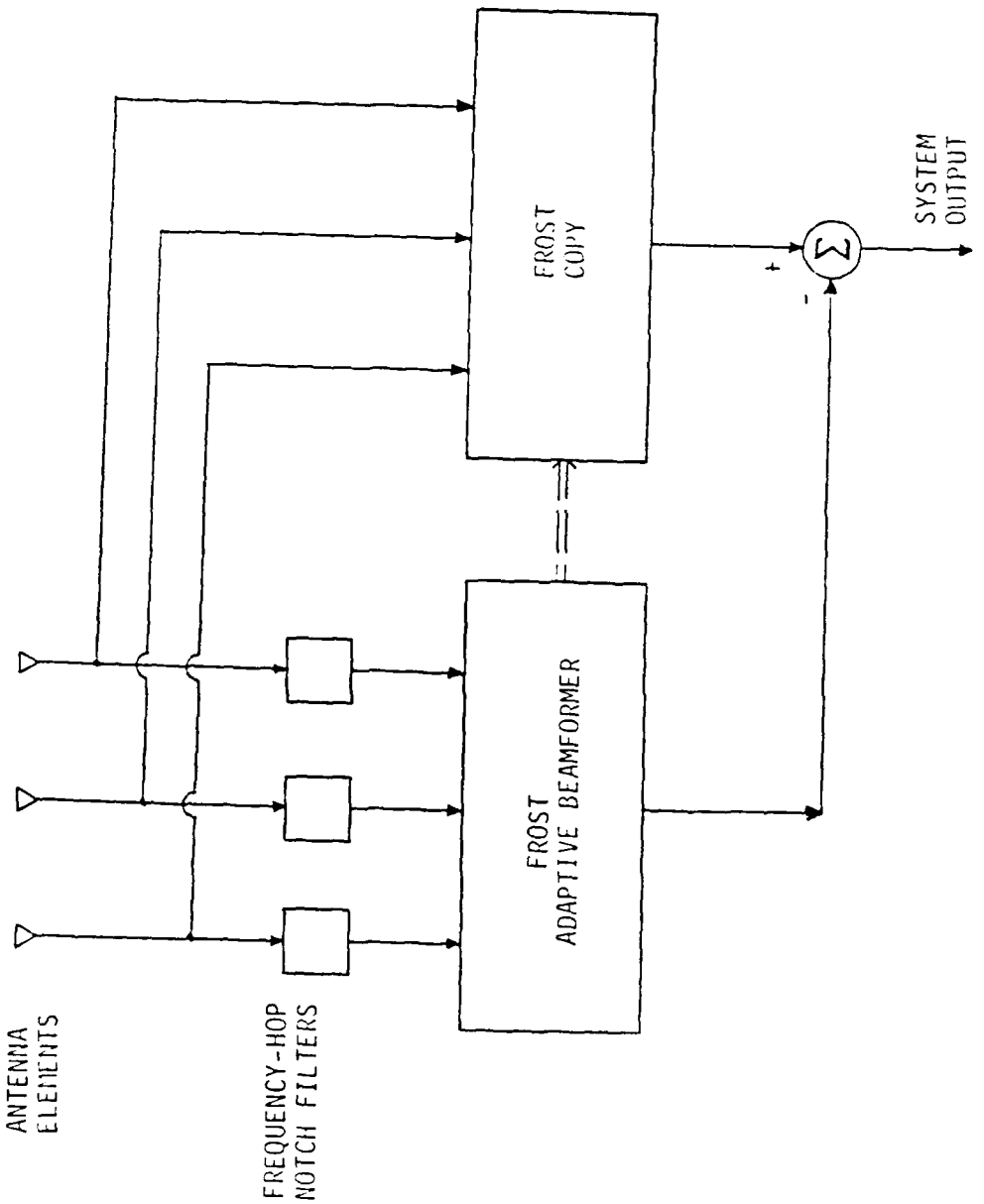


Figure 3.15. A master-slave Frost adaptive beamformer.

conducted to verify its effectiveness. In this experiment, a unit gain with zero phase constraint is imposed in the signal direction. Figure 3.16 shows a beam pattern of this modified beamformer as the adaptive process converges. A wideband null in the direction of the jammer is about 20 db below the sensitivity in the signal direction. Notice that the sensitivity in the signal direction is constrained to 0 db.

Figure 3.17 shows the frequency responses in the directions of both the signal and the jammer. Again note that the frequency response in the signal direction is sustained to unity gain with zero phase, and the frequency response in the jammer direction forms a flat null over the bandwidth of the jammer.

Figure 3.18 compares both the time domain waveforms of the slave output and the system output when the signal is hopping. The system output results in a cleaner waveform than the output of slave beamformer. The system output is generated by subtracting the master output from slave output, and this subtraction results in a band-pass filtering as was seen before. The idea is illustrated for the Frost beamformer in Figure 3.19.

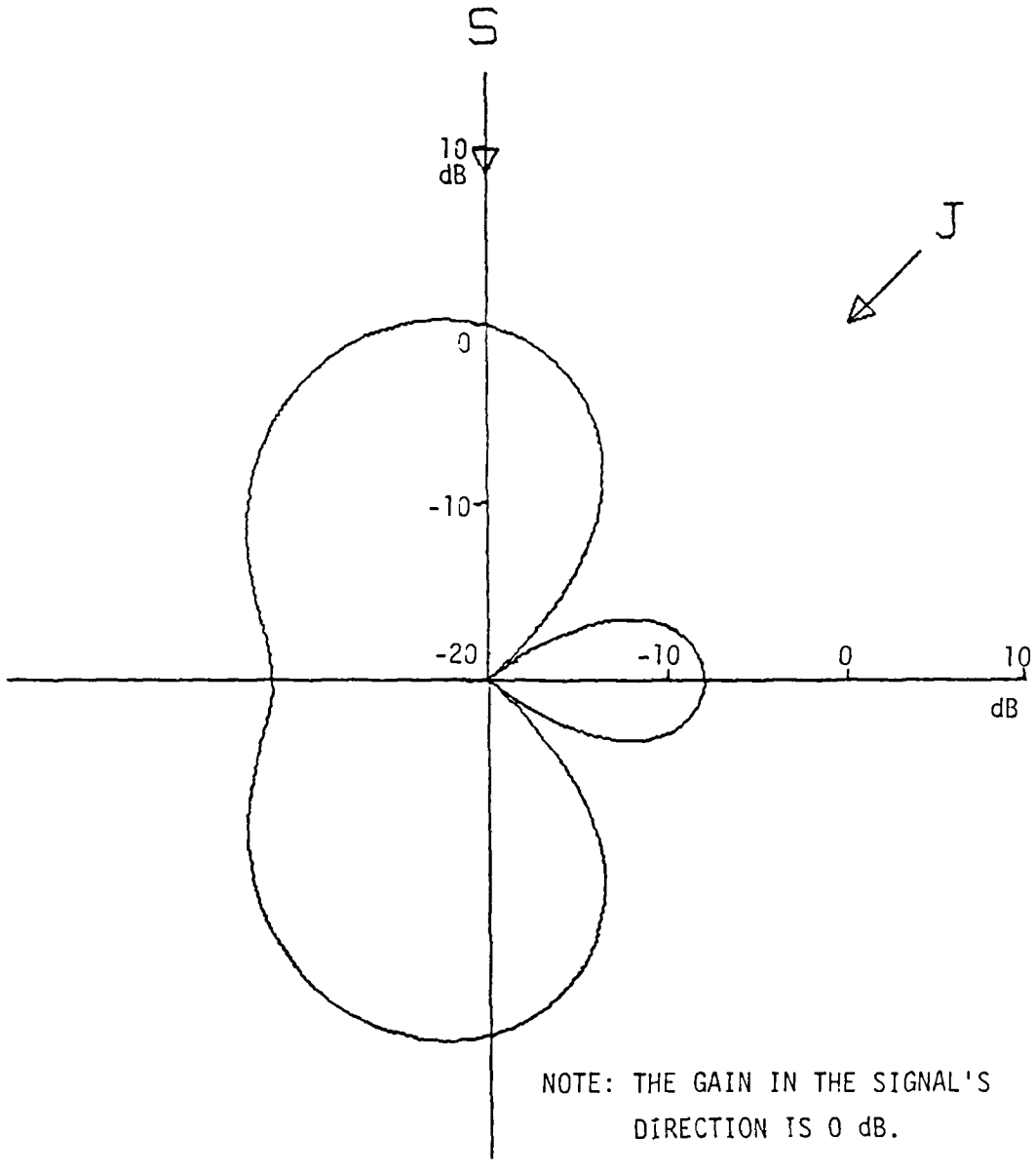
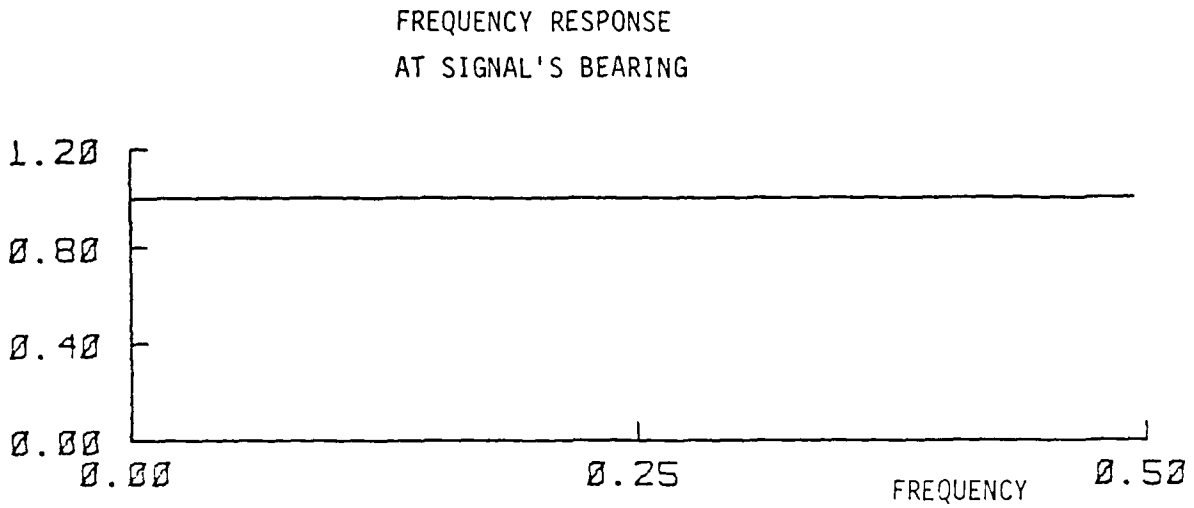


Figure 3.16. A converged beam pattern for the master-slave Frost beamformer.



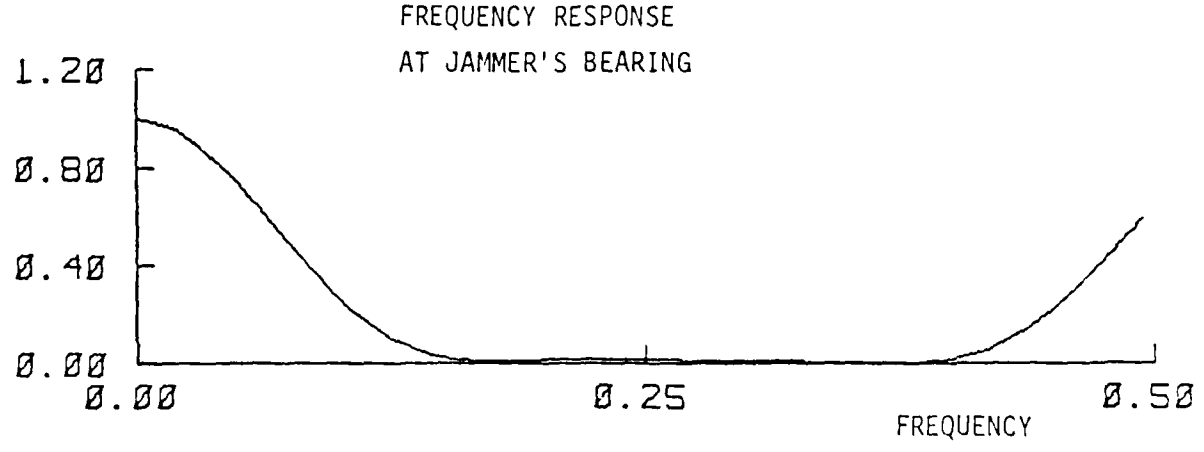


Figure 3.17. The frequency responses of the master-slave Frost beamfromer in both the directions of signal and jammer.

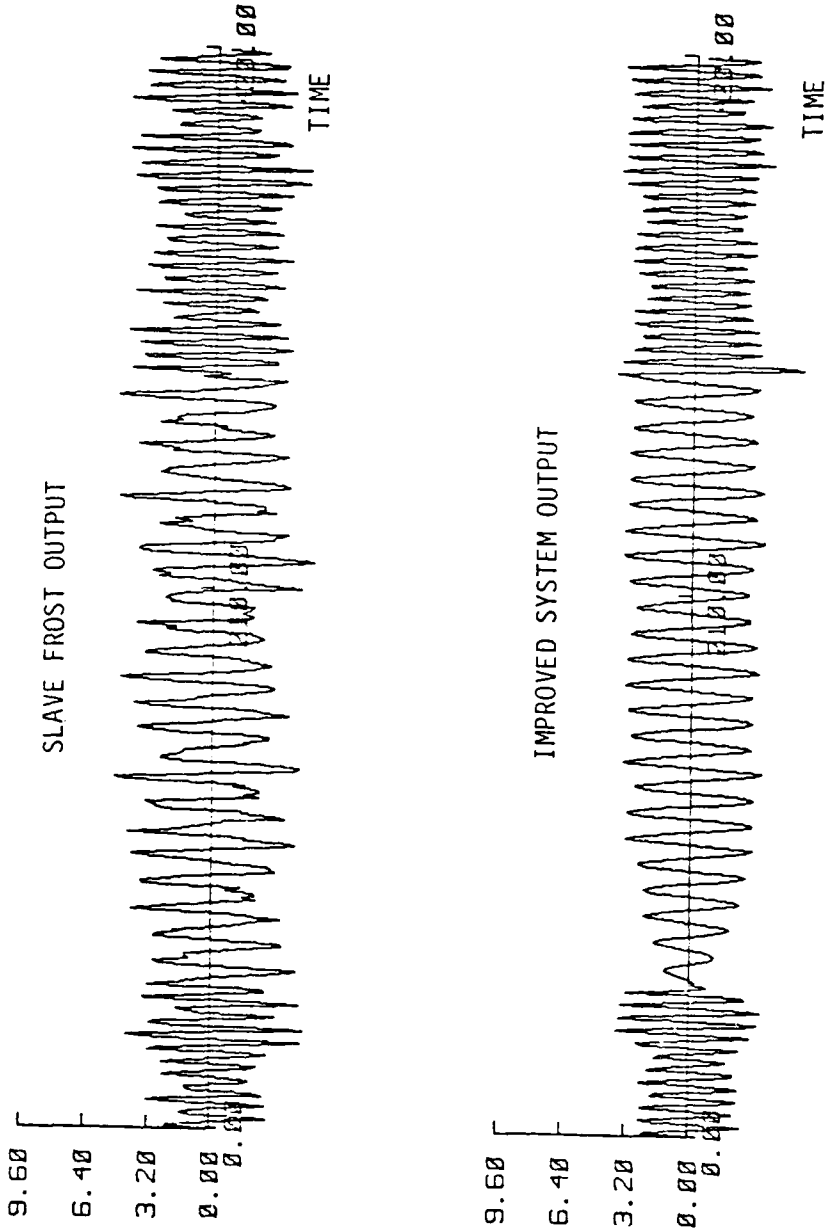


Figure 3.18. Time domain waveforms at the output of slave Frost beamformer and the improved system output.

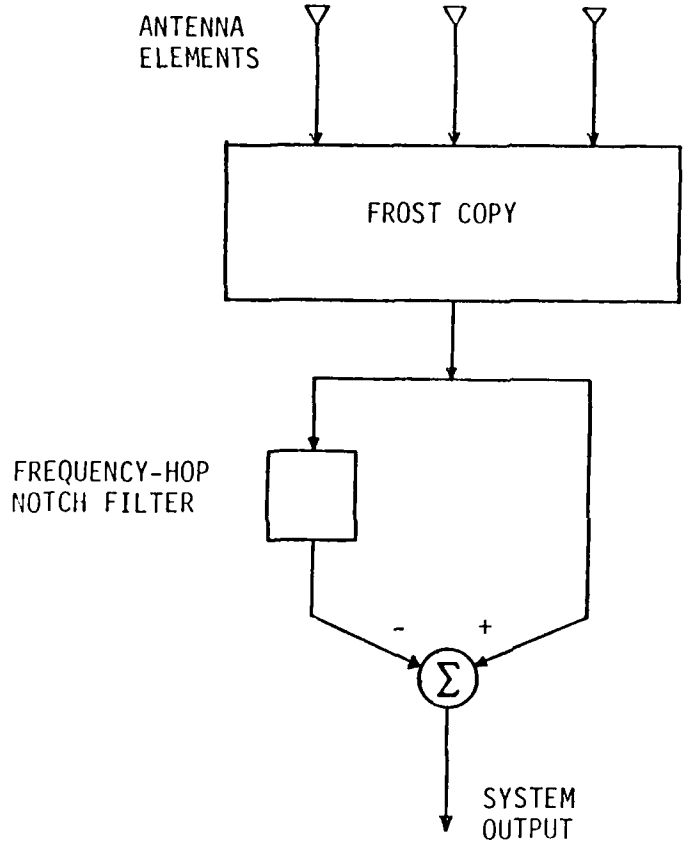


Figure 3.19. An equivalent scheme of the master-slave Frost beamformer when the master Frost beamformer reaches steady state.

3.7 Coherent Detection Results

So far, we have suggested several effective methods for adaptive arrays to combat signal cancellation. The comparisons and discussions are focused on the output of adaptive arrays. In this section, we first discuss a correlation technique to detect transmitted data from the output of frequency-hop adaptive arrays, and then compare the results for frequency-hop adaptive arrays and conventional adaptive arrays.

As previously shown in Figure 3.1, the array output is mixed or multiplied by a sinusoid generated from the local oscillator. This local oscillator is operating at the known hopping frequencies and mixed with proper phase shifts for different frequencies. Then it performs an "integrate and dump" operation within a certain time interval, the chip period T. This correlation technique is referred to as coherent detection or matched filtering. The decision maker following the integrate-and-dump consists of a set of thresholds to detect if the binary data is either + 1 or -1, or 0 when no signal is transmitted.

The following is a brief analysis of the coherent detection. Consider a frequency-hop digital communication system, where the signal frequency at any given time is assumed known at the receiving site, and only one bit of information is transmitted at each frequency at a time. The system operates with constant envelope, binary phase-shift-keyed signals of the form

$$s(t) = A \cos\{ \omega t + \phi(t) \},$$

where

A = a constant amplitude

ω = hopping frequency known by the pseudo-random code generator

$$\phi(t) = \begin{cases} 0 & \text{for binary } +1 \\ \pi & \text{for binary } -1 \end{cases} .$$

By using the coherent detection method, we are able to recover the original transmitted data. Now let the array output be

$$\mathbf{y}(t) = \hat{s}(t) + \hat{n}(t)$$

where

$$\hat{s}(t) = \begin{cases} \alpha \cos[\omega t + \phi(t) + \eta(\omega)] & \text{with presence of signal} \\ 0 & \text{with absense of signal} \end{cases}$$

$\eta(\omega)$ = the phase shift of adaptive filter

$$E[\hat{n}(t)] = 0 \quad .$$

the output of the integrate-and-dump is described as

$$Z(t) = \int_{t}^{t+T} \mathbf{y}(t) \cos[\omega t + \eta(\omega)]\, dt$$

$$= \int_{t}^{t+T} \frac{1}{2} \alpha \cos \phi(t)\, dt + \int_{t}^{t+T} \frac{1}{2} \alpha \cos[2\omega t + \phi(t) + 2\eta(\omega)]\, dt$$

$$+ \int_{t}^{t+T} \hat{n}(t) \cos[\omega t + \eta(\omega)]\, dt \quad .$$

Since the second and the third terms will be averaged out to be zero, the output of the integrate-and-dump would be

$$Z(t) \approx \frac{1}{2} \alpha \cos \phi(t) T,$$

or

$$Z(t) = \begin{cases} \frac{1}{2} \alpha T & \text{if } \phi(t) = 0 \\ \frac{-1}{2} \alpha T & \text{if } \phi(t) = \pi \\ 0 & \text{if no signal} \end{cases}.$$

Figure 3.20 shows two output plots of the integrate-and-dump. The top plot corresponds to a conventional adaptive array, whereas the bottom plot corresponds to one of the suggested frequency-hop adaptive arrays. An encoded frequency-hop signal is used in the presence of a jamming situation. The transmitted data sequence is 1, -1, 1, 1, -1, -1, then nothing. It is clear from the top plot that signal cancellation appears in the conventional adaptive array, and that sometimes this cancellation effect can be severe enough to cause false data decoding. For the suggested adaptive arrays, the integrate-and-dump integrates very clearly and steadily. This demonstrates the effectiveness of frequency-hop adaptive arrays.

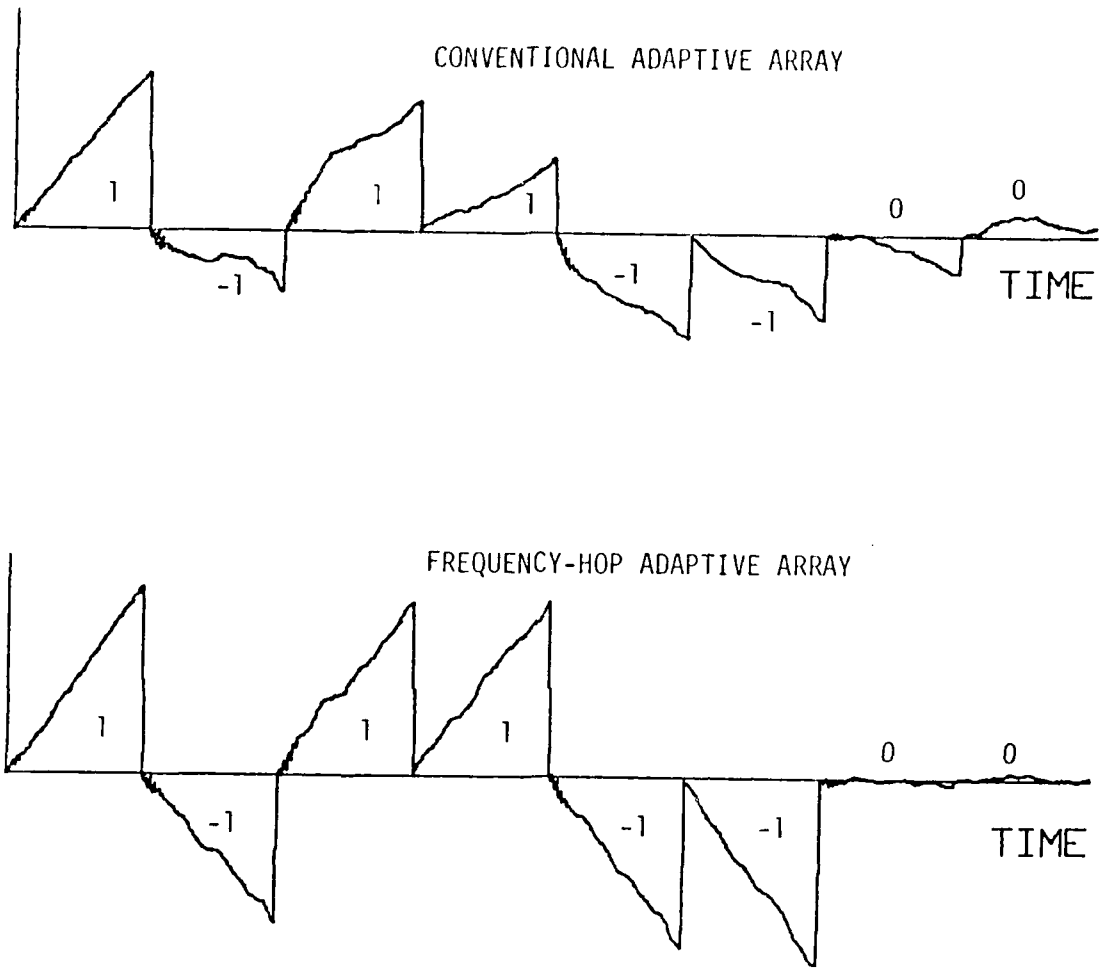


Figure 3.20. Comparison of the "integrate and dump" output for using conventional adaptive array and the frequency-hop adaptive array.

IV. FREQUENCY HOP NOTCH FILTERS

In Chapter III we suggested several effective schemes for adaptive arrays to eliminate signal cancellation. The frequency-hop notch filters are always required in these schemes. In Chapter IV, we present two methods to implement a frequency-hop notch filter. One is based on the structure of DFT, and the other is based on the structure of adaptive noise cancelling.

This chapter has four sections. Section 4.1 gives a brief introduction of notch filters. Section 4.2 discusses the DFT frequency-hop notch filter. Section 4.3 discusses the adaptive frequency-hop notch filter. The properties of frequency-hop notch filters such as transient response, band width, and spectral shaping are investigated and compared in Section 4.4.

4.1 Introduction

There are a variety of notch filters available for frequency-hop spread spectrum systems. Most of these notch filters fall into three categories: the all-zero type, the pole-zero type, and the all-pole type. An all-pole filter always requires an infinite order realization to create a stop band or a notch in the frequency response. Thus, it is not practical to implement an all-pole notch filter for frequency-hop systems. For this reason, the all-zero and the pole-zero types will be considered here. An all-zero notch filter can be implemented by using the structure of the discrete Fourier transform (DFT). Since the DFT can be regarded as a bank of band-pass filters [4.1], one may create a band-reject or

notch filter by rejecting a selected frequency band. For pole-zero notch filters, a simple design can be had which is based on the structure of adaptive noise canceller. Glover [2.4, 4.2] showed such a structure as a two-pole-two-zero notch filter centered at the frequency of a reference signal.

When used in frequency-hop adaptive arrays, some important factors of the filter must be considered. These include transient response, band width, spectral shaping, and feasibility. The data transmission rate, for example, strongly depends on the acquisition time in such a spread spectrum system. The acquisition time is increased by a slow transient response of the notch filter. Hence, the transient response of the filter plays an important role in determining the data transmission rate. The information bandwidth can also affect the data transmission rate. Spectral shaping may cause signal distortion and decrease the signal-to-noise ratio (SNR).

4.2 DFT Notch Filters

In this section, we analyze the frequency response of the DFT notch filter, then develop its transfer function, and finally discuss the limitations of this filter.

Consider the DFT notch filter shown in Fig. 4.1. The input signal feeds into a tapped-delay-line (TDL) to create an N-element vector consisting of delayed signal values. The discrete Fourier transform is applied to this vector to generate N output bins. Only the switch of a selected output bin is open. The rest of the other switches are closed, and the sum of the N-1 remaining DFT output bins forms the output of the notch filter. Notice that if *all* the DFT output bins including the selected one are summed, the filter will have a transfer function of unity. In other words, if all the DFT output bins are summed together, the filter's output is the same as the filter's input. This is intuitively clear, because the sum of all DFT output bins transforms into the first bin of the inverse DFT, and the first bin of the inverse DFT is exactly the input signal. The filter's output, therefore, is equal to the filter's input, only if all the switches are closed. When the switch of a selected bin is open, the filter's transfer function would be unity minus the transfer function from the input to the selected output bin. It is well known that the DFT can be viewed as a bank of band-pass filters, and that the center frequencies of these filters are uniformly spaced between zero and the sampling rate. When this selected frequency band is removed from the output, a band-reject or notch filter results. The selected bin which was not included in the output sum, then, can be used to select the notch's frequency band.

Let the input signal to the TDL in Fig. 4.1 be given by $x(n)$. By definition,

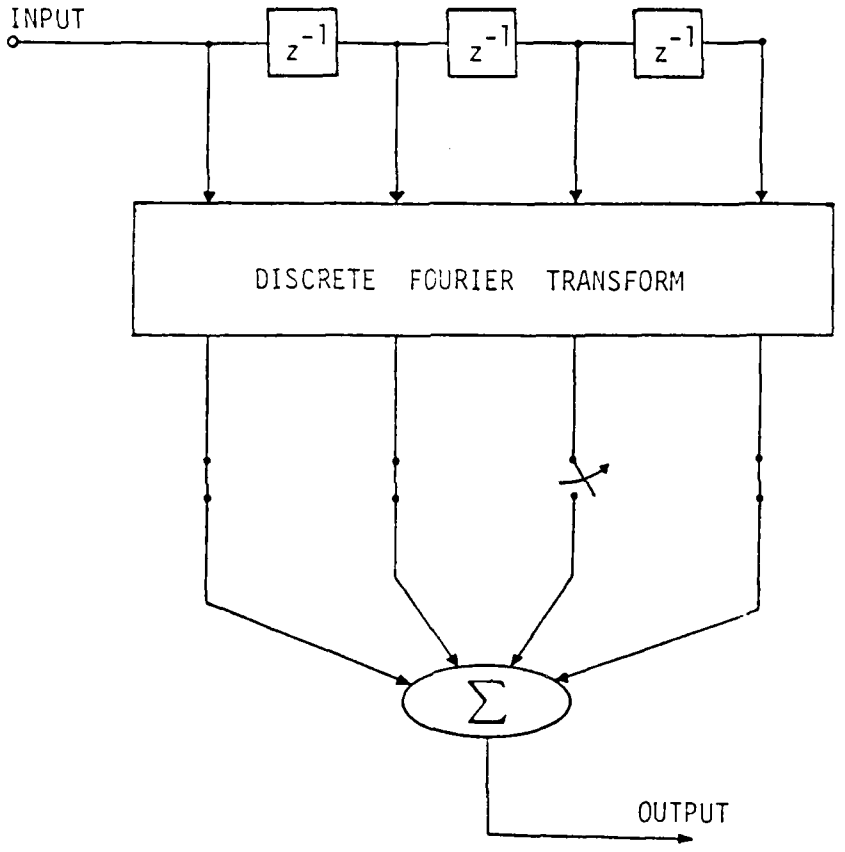


Figure 4.1. A DFT-based frequency-hop notch filter.

the m^{th} output bin of the DFT is

$$y_m(n) = \sum_{k=0}^{N-1} x(n-k) W^{mk} \tag{4.1}$$

where

$$W = e^{-j2\pi/N} \quad . \tag{4.2}$$

Eqn. (4.1) is a convolution between the input and the filter's impulse response. Thus, it is convenient to express (4.1) in terms of the Z-transform of $x(n)$ and $y_m(n)$, $X(z)$ and $Y_m(z)$. By taking the Z-transform of (4.1), we have

$$H_m(z) = \frac{Y_m(z)}{X(z)} = \sum_{k=0}^{N-1} z^{-k} \cdot W^{mk} \quad . \tag{4.3}$$

Through some simple algebraic manipulation, we can rewrite (4.3) as follows,

$$H_m(z) = \frac{1-z^{-N}}{1-W^m z^{-1}} \qquad m = 0, 1, \cdots, N-1 \quad . \tag{4.4}$$

Note that (4.4) represents the transfer function of the filter from input to the m^{th} DFT output bin. Evaluating $H_m(z)$ with $z = e^{-j\omega}$ gives the frequency response of the m^{th} filter. As previously mentioned, this is a band-pass filter with center frequency $w = 2\pi m/N$. The bandwidth of the band-pass filter is

$$BW = \frac{2\pi}{N \cdot T} \quad , \tag{4.5}$$

where T is the sampling period. Notice that the bandwidth depends on the value of N, the size of the DFT. The larger the value of N, the smaller the bandwidth. If all the transfer functions in (4.4) are summed, *i.e.*, all the output switches are closed, then we will have

$$H(z) = \sum_{m=0}^{N-1} H_m(z)$$
$$= \sum_{m=0}^{N-1} \sum_{k=0}^{N-1} z^{-k} W^{mk}$$
$$= N \quad . \tag{4.6}$$

Eqn. (4.6) shows that when all switches of the output bins are closed, the transfer function of the filter, as shown in Figure 4.1, is nothing but a constant gain which can be normalized to be unity. As all the output bins except the m^{th} one are summed to form the output, the transfer function of the filter will be

$$H_m(z) = N - \frac{1-z^{-N}}{1-W^m z^{-1}}$$
$$= \frac{(N-1) - NW^m z^{-1} - z^{-N}}{1-W^m z^{-1}} \quad . \tag{4.7}$$

When evaluated with $z=e^{-j\omega}$, the filter's frequency response results in a narrow band rejection centered at normalized frequency $w = 2\pi m/N$. Figure 4.2 is the frequency response of a notch filter based on the structure of the DFT. The gain is normalized to be unity, and the frequency of the notch for the case illustrated is a quarter of the sampling frequency. For this frequency response, an infinite null is created at the selected notch frequency. In addition, the passband has a ripple response, and this response may introduce signal distortion.

It appears that this filter structure does not result in a perfect notch filter. There are several inherent performance limitations of the DFT approach [13]. One of them is that of frequency resolution. The frequency resolution is roughly the reciprocal of the time interval over which the sampled data is available. A second limitation is due to the leakage of the DFT. Implicit windowing of the

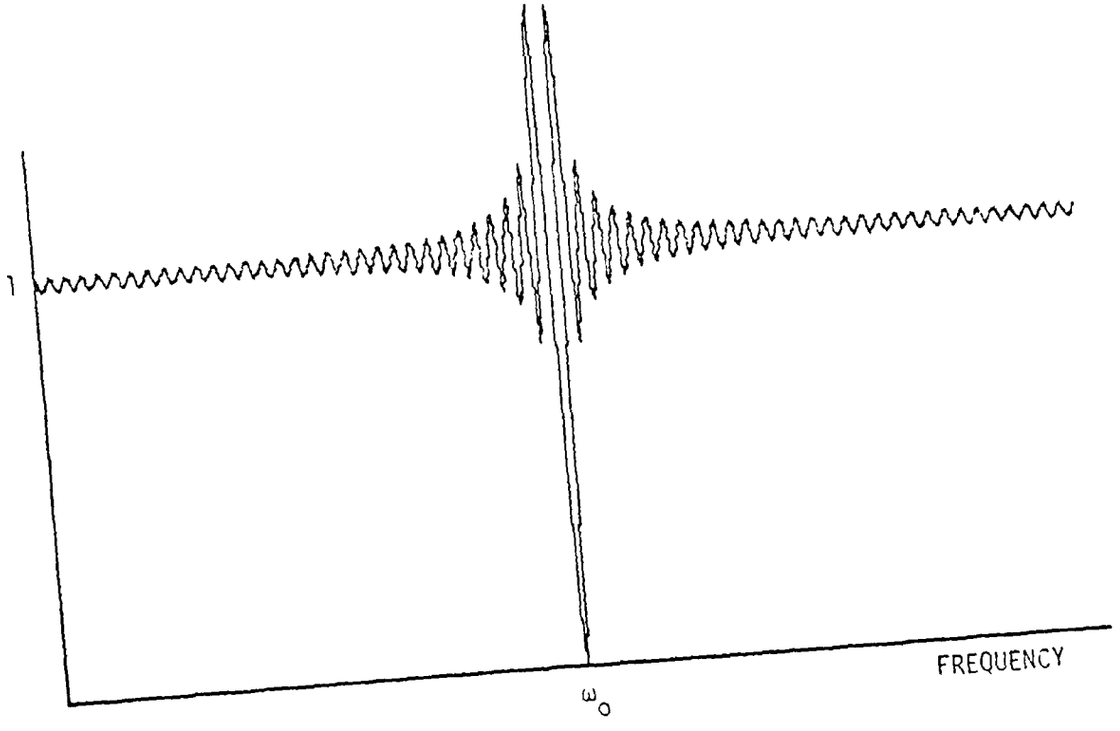


Figure 4.2. The frequency response of a DFT-based notch filter.

data always occurs when processing with the DFT, and the windowing effect manifests itself as "leakage" in the spectral domain. In other words, energy in the main lobe of a spectral response "leaks" into the sidelobes, creating signal distortion.

4.3 Adaptive Notch Filters

Adaptive notch filters can be realized by adaptive noise cancelling techniques. These techniques have been used for a variety of applications in speech processing, array processing, and communication systems. The concepts of adaptive noise cancelling are first described and derived by Widrow *et.al.* [2.3] and later extended by Glover [4.2]. Figure 4.3 shows the structure of a noise canceller with two adaptive weights. The error criterion minimizes the output power of the noise canceller. The weights are adjusted by the LMS algorithm. The primary input is assumed to be any kind of signal. With a sinusoidal reference input, the adaptive noise canceller sums up the weighted in-phase and quadrature-phase reference components, and then subtracts the sum from the primary input. Glover has shown that there is a transfer function from the primary input to the noise canceller output. This transfer function performs as a notch filter nulling at the frequency of the reference sinusoid, and it can be expressed as follows,

$$H(z) = \frac{z^2 - 2\cos(w_o T)\, z + 1}{z^2 - 2(1-\mu)\cos(w_o T)\, z + 1-2\mu} \quad , \tag{4.8}$$

where μ is the adaptive algorithm step size, w_o is the frequency of the reference input, and T is the sampling period. Equation (4.8) describes the transfer function of a second-order notch filter. This notch filter has two zeros on the unit circle at the frequency of the reference sinusoid and two poles located along the same angle as the zeros but at a radius, $1-\mu$, somewhat less than one. The bandwidth of the notch filter is

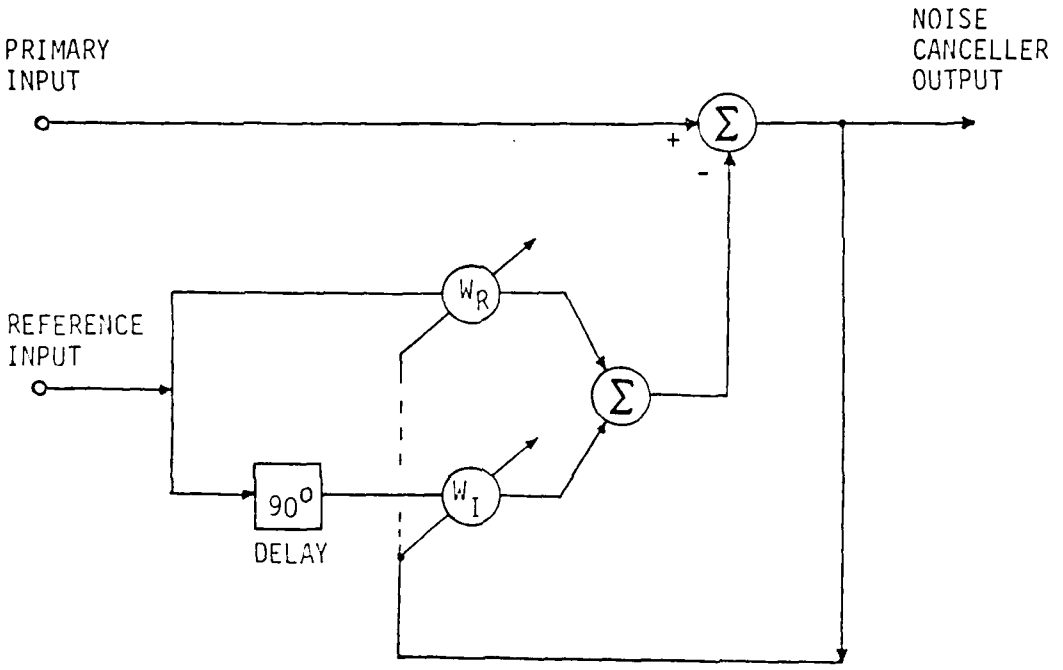


Figure 4.3. A simple two-weight adaptive noise canceller.

$$BW = \frac{2\mu}{T} \quad . \tag{4.9}$$

Notice that the step size μ of the LMS algorithm controls both the radii of the two poles and the bandwidth of the notch filter. Figure 4.4 shows the frequency responses of the adaptive noise canceller in Figure 4.3. The reference input is a sinusoid with frequency w_o and unity amplitude. Figure 4.4a and 4.4b correspond to the large and the small value of μ, respectively. The value of μ does not affect the notch's frequency, but it does affect the bandwidth of the notch. The larger the value of μ, the wider the bandwidth of notch. Notice that both frequency responses have an infinite null at the frequency of reference sinusoid. In addition, the passband has a flat response with unity gain. In other words, the adaptive noise canceller can function as a frequency-controllable notch filter by controlling the frequency of the reference input.

When used in frequency-hop spread spectrum receivers, this form of notch filter offers easy control of bandwidth, an infinite null, and the capability of adaptively tracking the exact frequency of the reference signal. Figure 4.5 shows a method of implementing a frequency-hop notch filter based on adaptive noise cancelling. A set of local oscillators are available, and each tunes to the center frequency of one bin of the frequency-hopped signal. These local oscillators can be used as the sinusoidal reference inputs to an adaptive noise canceller as shown in Figure 4.3. By sequencing through the different oscillator signals, the adaptive noise canceller forms a frequency-hop notch filter.

As mentioned before, the transient response is a crucial factor in frequency-hop adaptive arrays. Time-domain analysis, therefore, is necessary for studying

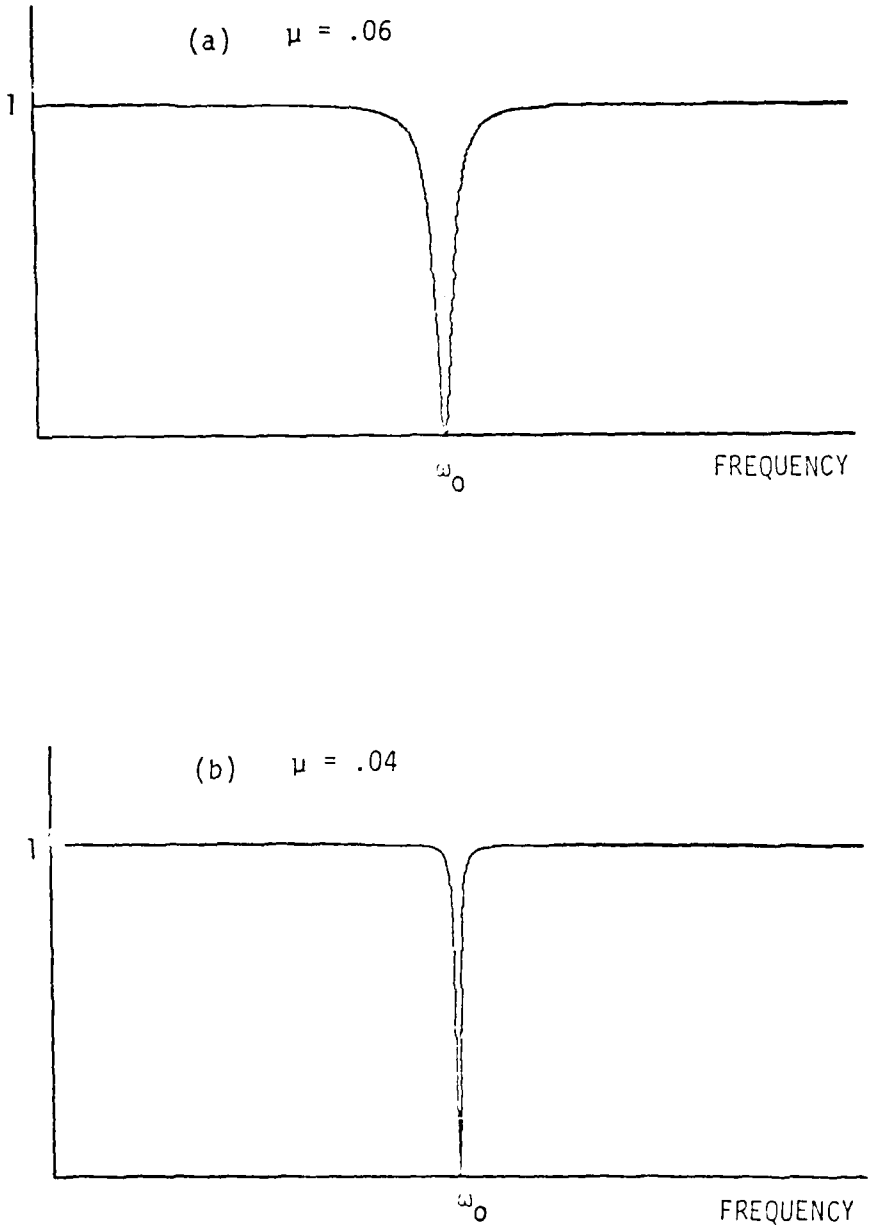


Figure 4.4. The frequency responses of the adaptive noise canceller with different values of step size.

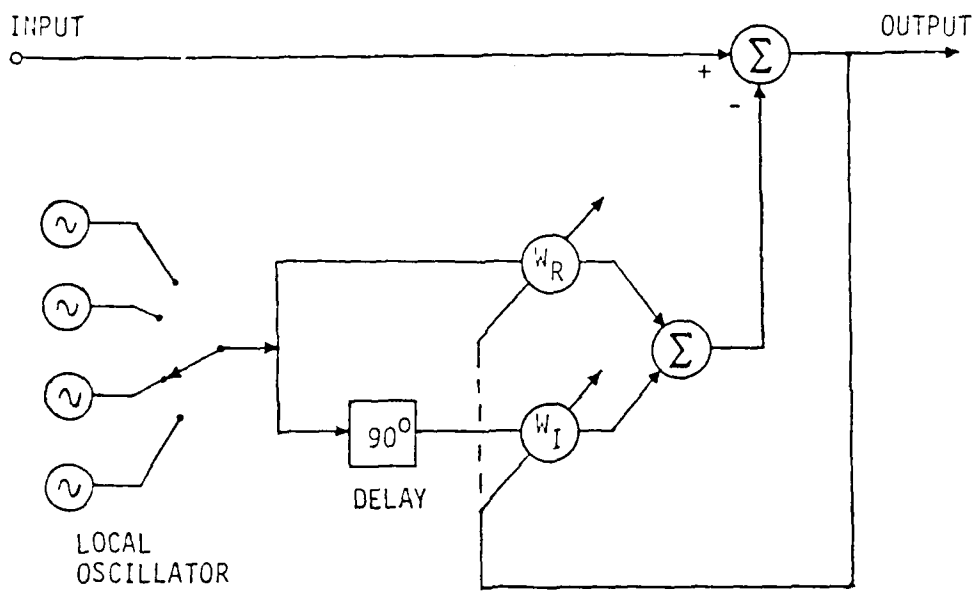


Figure 4.5 A detailed structure of the frequency-hop adaptive notch filter.

the time constant, the transient response, etc. The following is a state-space analysis for the adaptive noise canceller. We define the weight vector as the state vector, and will show that the difference equations of the adaptive noise canceller describe a linear time-invariant system.

Consider a two-weight adaptive noise canceller in Figure 4.3, where the primary input is denoted d_k and the reference input is a pure sinusoid. The canceller output ϵ_k is as follows,

$$\begin{aligned} \epsilon_k &= d_k - y_k \\ y_k &= W_R(k)\ cos(wkT) + W_I(k)\ sin(wkT) \end{aligned} \quad . \tag{4.10}$$

The least mean square (LMS) algorithm is given by

$$\begin{aligned} W_R(k+1) &= W_R(k) + 2\,\mu\,\epsilon_k\ cos(\ wkT\ ) \\ W_I(k+1) &= W_I(k) + 2\,\mu\,\epsilon_k\ sin(\ wkT\ ) \end{aligned} \quad , \tag{4.11}$$

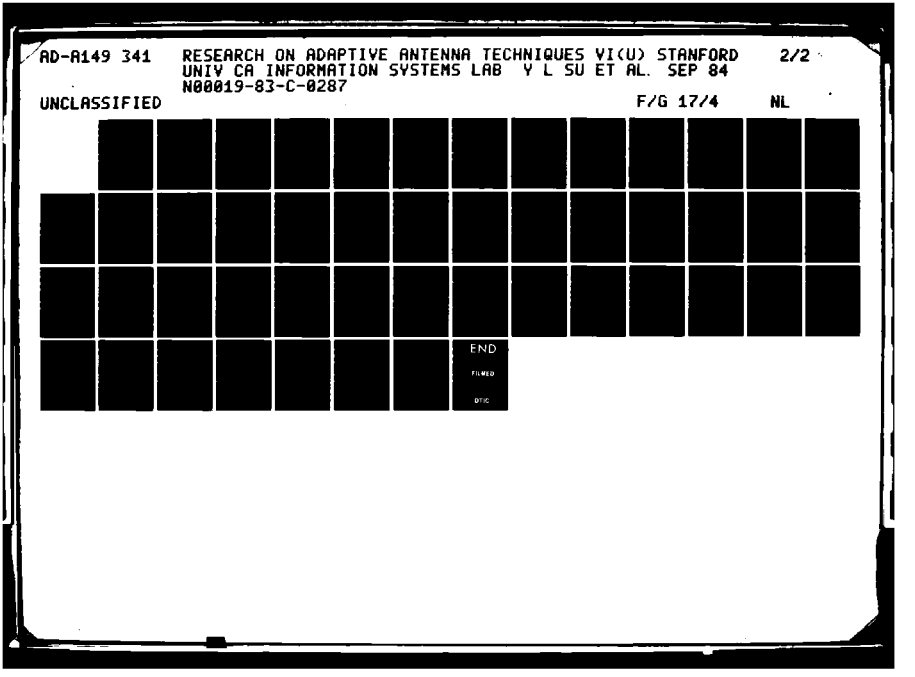
where μ is a constant step size. Substituting of (4.10) into (4.11), yields

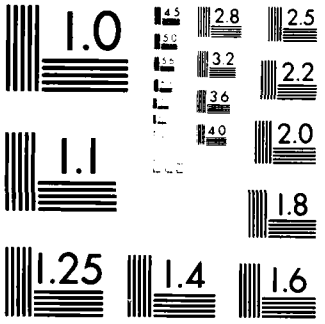
$$\begin{bmatrix} W_R(k+1) \\ W_I(k+1) \end{bmatrix} = \begin{bmatrix} 1 - 2\mu cos^2(wkT) & -2\mu cos(wkT)sin(wkT) \\ -2\mu cos(wkT)sin(wkT) & 1 - 2\mu sin^2(wkT) \end{bmatrix} \begin{bmatrix} W_R(k) \\ W_I(k) \end{bmatrix}$$

$$+ \ 2\,\mu \begin{bmatrix} cos(wkT) \\ sin(wkT) \end{bmatrix} \cdot d_k \tag{4.12}$$

$$\epsilon_k = d_k - \begin{bmatrix} cos(wkT) \\ sin(wkT) \end{bmatrix}^T \begin{bmatrix} W_R(k) \\ W_I(k) \end{bmatrix} \tag{4.13}$$

Notice that (4.12) is a *state* equation, and (4.13) is an *output* equation. These two equations form a *normal* equation, or a state-space representation of the adaptive


AD-A149 341
RESEARCH ON ADAPTIVE ANTENNA TECHNIQUES VI(U) STANFORD
UNIV CA INFORMATION SYSTEMS LAB Y L SU ET AL. SEP 84
N00019-83-C-0287
2/2
UNCLASSIFIED
F/G 17/4
NL
END
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

noise canceller with input d_k and output ϵ_k. Although the transition matrix in (4.12) is a function of the timing index k, it will be shown that the adaptive noise canceller is a linear and time-invariant system. Since the transition matrix is symmetric, it can be decomposed into the form $A\ \Lambda\ A^{-1}$, where

$$A(k) = \begin{bmatrix} cos(wkT) & -sin(wkT) \\ sin(wkT) & cos(wkT) \end{bmatrix} \quad , \tag{4.14}$$

and

$$\Lambda = \begin{bmatrix} 1-2\mu & 0 \\ 0 & 1 \end{bmatrix} . \tag{4.15}$$

The matrix Λ is a diagonal matrix, and A is a rotation matrix with the following properties.

$$A(k) = R^k$$

$$R \triangleq \begin{bmatrix} cos(wT) & -sin(wT) \\ sin(wT) & cos(wT) \end{bmatrix} \tag{4.16}$$

$$A^{-1}(k) = A^T(k) = A(-k) \quad .$$

By the sampling theorem, the sampling frequency must be at least twice the cutoff frequency of the sampled signal. This implies that

$$0 < wT < \pi \quad . \tag{4.17}$$

Premultiplying (4.12) by $A^{-1}(k)$, we have

$$A^{-1}(k)\ W(k+1) \;=\; \Lambda\, A^{-1}(k)\ W(k) \;+\; 2\mu\ d_k \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad . \tag{4.18}$$

Define the weight vectors $W(k)$, $V(k)$ as the following,

$$W(k) \;\triangleq\; \begin{bmatrix} W_R(k) \\ W_I(k) \end{bmatrix}$$

$$V(k) \;\triangleq\; A^{-1}(k-1)\cdot W(k) \quad , \tag{4.19}$$

then (4.12) and (4.13) can be rewritten as

$$\begin{aligned} V(k+1) &= F\cdot V(k) \;+\; G\cdot d_k \\ \epsilon_k &= H\cdot V(k) \;+\; d_k \\ W(k) &= A(k-1)\cdot V(k) \quad , \end{aligned} \tag{4.20}$$

where

$$\begin{aligned} F &\;\triangleq\; \Lambda\cdot R^{-1} \\ G &\;\triangleq\; 2\mu \begin{bmatrix} 1 & 0 \end{bmatrix}^T \\ H &\;\triangleq\; \begin{bmatrix} -1 & 0 \end{bmatrix}^T \quad . \end{aligned} \tag{4.21}$$

Notice that R, defined in (4.16), is a constant rotation matrix and its element values depend only on the relative sampling frequency wT. Since the matrices F, G, and H are all constant, the system described by (4.20) is a linear time-invariant system. This is true for the adaptive noise canceller only when the reference input is a pure sinusoidal signal.

The complete response of the system consists of both the zero-state response and the zero-input response. By setting initial weights $W(0)$ zero, we can

directly solve the zero-state response, which is

$$V(k) = \sum_{n=0}^{k-1} F^{k-1-n} G d_n \quad . \tag{4.22}$$

By zeroing the primary input d_k, we may expand (4.20) for zero-input response. This means that,

$$V(k) = F^k \cdot V(0) \quad . \tag{4.23}$$

Thus, the complete response will be the superposition of zero-input response and zero-state response, *i.e.*,

$$V(k) = F^k \cdot V(0) + \sum_{n=0}^{k-1} F^{k-1-n} G d_n \quad .$$

By (4.19) and (4.20), we can explicitly express the complete response in closed form as

$$\begin{aligned} W(k) &= A(k-1)\, V(k) \ , \\ &= R^{k-1} F^k R\ W(0) + R^{k-1} \sum_{n=0}^{k-1} F^{k-1-n} G d_n \end{aligned} \tag{4.24}$$

$$\epsilon_k = H \cdot W(k) + d_k \quad , \tag{4.25}$$

where F, G, H, and R are defined in (4.16) and (4.21).

The transition matrix F plays an important role in the transient analysis of the adaptive noise canceller. Its eigenvalues determine the character of the transient response. Through the eigenvalues, the stability and the time constant of the response can be investigated. According to (4.21), the transition matrix F is

$$F \triangleq \Lambda \cdot R^{-1}$$

$$= \begin{bmatrix} 1-2\mu & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} cos(wT) & sin(wT) \\ -sin(wT) & cos(wT) \end{bmatrix}$$

$$= \begin{bmatrix} (1-2\mu)cos(wT) & (1-2\mu)sin(wT) \\ -sin(wT) & cos(wT) \end{bmatrix} \quad . \tag{4.26}$$

The eigenvalues of F shall satisfy the following equation,

$$\lambda^2 - 2(1-\mu)cos(wT)\,\lambda + 1-2\mu = 0 \quad . \tag{4.27}$$

Notice that these eigenvalues of F are the same as the poles of the filter's transfer function in (4.8). Solving (4.27) for the eigenvalues, we have

$$\lambda_i = (1-\mu)cos(wT) \pm \sqrt{(1-\mu)^2cos^2(wT) - (1-2\mu)} \quad . \tag{4.28}$$

There are two possibilities for the eigenvalues. One is the real case, and the other is the complex case. For the case of real eigenvalues, the term inside the square root must be greater than or equal to zero. This implies that

$$(1-\mu)^2 \cos^2(wT) \geq (1-2\mu) \quad ,$$

or

$$\mu \geq \frac{sin(wT)}{1 + sin(wT)} \quad . \tag{4.29}$$

For stability, the absolute values of both eigenvalues in (4.28) must be less than one so that transients will die out. This implies that

$$1 - |\lambda_i|^2 > 0 \quad , \tag{4.30}$$

or equivalently

$$2(1-\mu)\,(\,\alpha \pm \beta\,) \;>\; 0 \quad ,$$

where

$$\alpha \;=\; 1 - (1-\mu)cos^2(wT)$$

$$\beta \;=\; cos(wT)\,\sqrt{(1-\mu)^2\cos^2(wT) - (1-2\mu)}$$

As long as (4.17) and (4.29) hold, we can prove that

$$(\,\alpha \pm \beta\,) \;>\; 0 \quad .$$

Thus, eqn. (4.30) implies that μ must be less than one for the stability requirement. Combining the stability requirement and the real eigenvalue criterion in (4.29), we will have

$$\frac{sin(wT)}{1 + sin(wT)} \;\leq\; \mu \;<\; 1 \quad . \tag{4.31}$$

In other words, if μ is in the region of (4.31), the adaptive noise canceller will be stable with an overdamped transient response of the adaptive weights.

For the case of complex conjugate eigenvalues, the term inside the square root in (4.28) should be less than zero. This implies that

$$\mu \;<\; \frac{sin(wT)}{1 + sin(wT)} \quad . \tag{4.32}$$

For stability, the modulus of the eigenvalues should be less than one to result in a stable filter. In other words,

$$|\,\lambda_i\,| \;=\; |\,1 - 2\mu\,|^{\frac{1}{2}} \;<\; 1 \quad . \tag{4.33}$$

From (4.32) and (4.33), the value of μ for a stable adaptive noise canceller with underdamped transients should be

$$0 < \mu < \frac{sin(wT)}{1 + sin(wT)} \quad . \tag{4.34}$$

Combining (4.31) and (4.34) yields a stable range of the step size :

$$0 < \mu < 1 \quad . \tag{4.35}$$

Figure 4.6 shows the stable region of μ versus the relative frequency wT. The region is partitioned into two subregions by the curve

$$\mu = \frac{sin(wT)}{1 + sin(wT)} \quad . \tag{4.36}$$

Region I corresponds to the overdamped transient response of the set of weights, whereas region II the underdamped transient response.

Now we can relate the state-space analysis of the adaptive noise canceller with the property of the adaptive notch filter. Notice that for a second-order notch filter, the poles are complex conjugates. These complex poles can be controlled by selecting μ in region II, which corresponds to an underdamped transient response.

Let the time constant of the adaptive notch filter be denoted as τ. Considering the eigenvalues in (4.33), we should have

$$| \lambda |^{\tau} = | 1 - 2\mu |^{\frac{\tau}{2}} = e^{-1} \quad ,$$

or

$$\tau = \frac{-2}{\ln | 1 - 2\mu |} \quad . \tag{4.37}$$

If μ is very small, the time constant will be

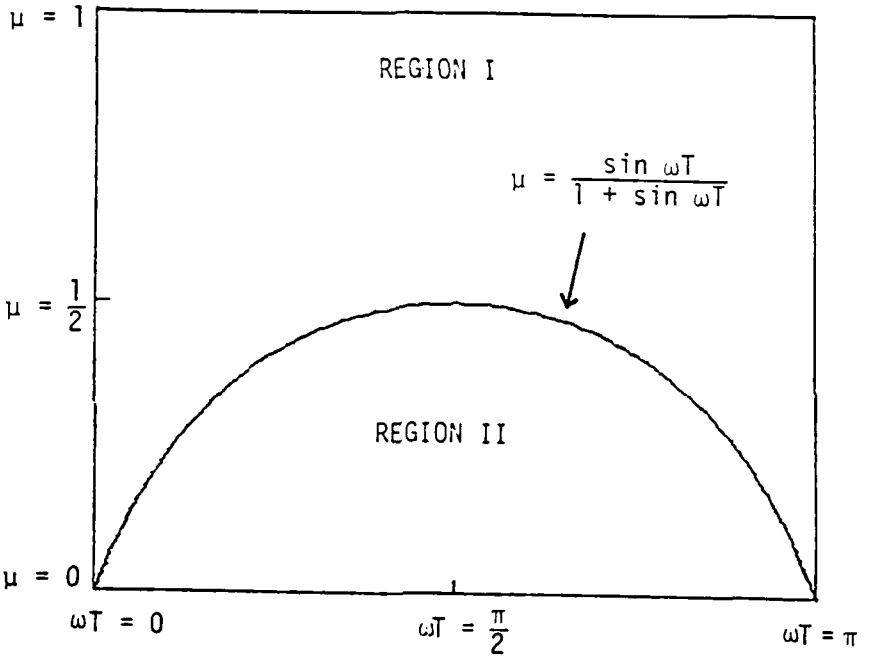


Figure 4.6 A stable region of the step size for the adaptive notch filter.

$$\tau = \frac{1}{\mu} \quad . \tag{4.38}$$

Notice that the time constant is a function of μ only, and that it is inversely proportional to the bandwidth of the notch as in (4.8). In other words, the smaller the value of μ, the longer the transient response and the narrower the notch.

4.4 Comparison and Discussion

In comparing the performance of the two suggested frequency-hop notch filters, several issues must be considered. These include transient performance, notch bandwidth, spectral shaping, and flexibility.

In frequency-hop spread spectrum adaptive arrays, high data transmission rate is desired. The data transmission rate, however, is closely related to the transient response of the frequency-hop notch filter. For each frequency hop, if the transient response dies out very slowly, the required acquisition time will be longer. Thus, long transient response slows down the data transmission rate.

Suppose the input to the notch filter consists of a frequency-hop signal and white noise,

$$d_k = a\ cos(wkT) + n_k \quad .$$

The frequency w is a function of time, and is randomly hopping. Consider the moment that the signal and the filter's notch are hopping into a new frequency band. Since the DFT notch filter is an FIR filter, the residuals of the last frequency hop will still remain in the tapped-delay-line of the filter. These residuals will then appear at the output of the notch filter until they are shifted out of the tapped-delay-line. In other words, the transient response will completely die out after a number of iterations equal to the length of the tapped-delay-line.

The transient performance of the adaptive notch filter depends on its eigenvalues or poles. As mentioned in Section 4.1, the adaptive notch filter is an IIR filter, and the time constant can be a measure of its dominant poles.

According to (4.37), the time constant is a function of μ, which is

$$\tau = \frac{-2}{\ln|\ 1 - 2\mu\ |} \approx \frac{1}{\mu} \quad . \tag{4.37}$$

The larger the time constant, the longer the transient response. Figure 4.7 shows the transient responses for various notch filters. The signal is hopping from one frequency bin to another, and the frequency-hop notch filter is notching accordingly. Figure 4.7a corresponds to a DFT notch filter. Since it is an all-zero filter, the transient response completely dies out after N cycles, where N is the order of the filter. Figure 4.7b and 4.7c correspond to adaptive notch filters with different values of μ. The larger the value of μ, the faster the transient response dies out. The value of μ, of course, should be in the stable region.

The bandwidth of the notch filter should be small, if high frequency resolution is desired. The frequency resolution of the DFT is proportional to the number of frequency bins. The number of frequency bins is the same as the number of taps in the tapped-delay-line. This implies that the DFT notch filter must have a large number of taps in order to have a small bandwidth. This large number of taps, however, will cause a long transient response. Explicitly, the bandwidth and the time constant of the DFT notch filter are

$$BW_{DFT} = \frac{2\pi}{N \cdot T} \quad ; \quad \tau_{DFT} = N \quad .$$

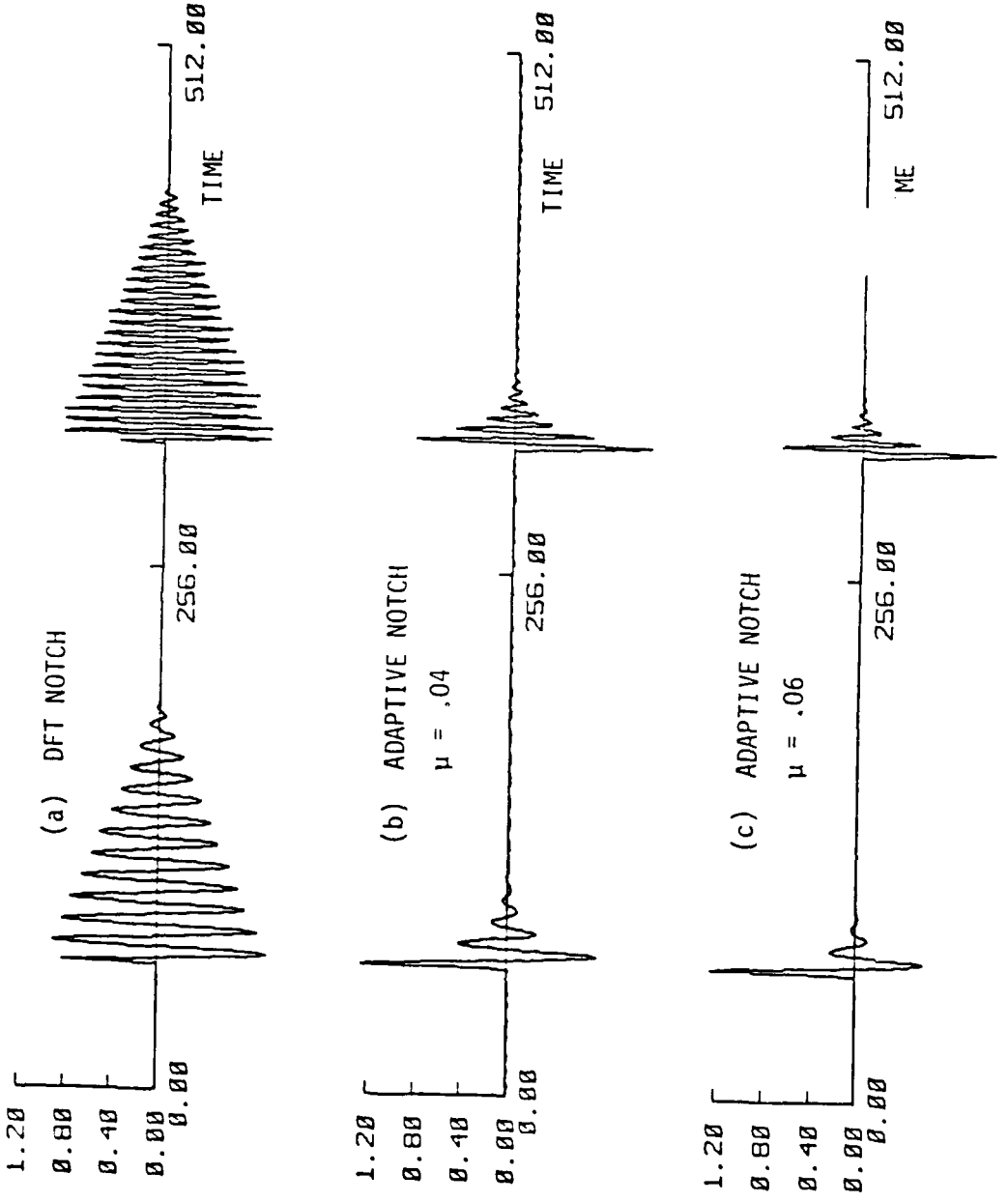


Figure 4.7. The transient responses of the DFT-based notch filter and the adaptive notch filter.

The bandwidth and the time constant of the adaptive notch filter, according to (4.9) and (4.38), are

$$BW_{adaptive} = \frac{2\mu}{T} \quad ; \quad \tau_{adaptive} = \frac{1}{\mu} \quad . \tag{4.9}$$

To have a small bandwidth, μ must be small. This small value of μ however may result in a slow speed of transient performance. Notice that high frequency resolution will cause long transient performance no matter which filter is used. Nonetheless, the adaptive notching may be more flexible than the DFT notching. For adaptive notching, the step size μ can be easily adjusted to control the bandwidth or the transient performance. For DFT notching, to adjust the bandwidth or the transient performance means to change the length of the tapped-delay-line and the length of DFT.

Another comparison concerns the frequency response outside the notch, *i.e.*, the spectral shaping. When processing with the DFT, implicit data windowing always occurs. This time-domain windowing introduces the passband leakage effect. This inherent leakage effect then introduces distortion for the passband signals. With the adaptive notch filter, the frequency response outside the notch band is fairly smooth and flat. Thus, passband distortion will be negligible. In this respect, the adaptive notching is clearly superior to the DFT notching.

In conclusion, for both the adaptive notch filter and the DFT notch filter, we can easily select the notch's frequency by controlling the reference frequency and the switched output bin, respectively. Data transmission rate and distortion level, however, can be very crucial in frequency-hop adaptive arrays. As such adaptive notching performs better than DFT notching, since the adaptive notch

filter has a flatter passband response than the DFT notch filter. In addition, it is easier to control the notch bandwidth and the time constant when using the adaptive notch filter.

V. PARALLEL SPATIAL SMOOTHING

In Chapter III we applied the frequency-hop approach to eliminate signal cancellation in adaptive arrays. This approach is very effective when the desired signal is known to be a frequency-hop spread spectrum signal. In some applications the desired signal is unknown, and the available *a priori* information is the direction of the desired signal. Spatial smoothing techniques can be applied with adaptive arrays in these situations. In Chapter V, we present a method based on a spatial smoothing technique to combat signal cancellation. Our only assumption is that the direction of the desired signal is known. We emphasize the Frost adaptive beamformer. Basically, this algorithm has parallel structure and requires the same computation power as any conventional beamformer. Furthermore, when the adaptive processor reaches the minimum of the performance surface, the total system output will be a maximum-likelihood estimate of the desired signal in a spatial averaging sense.

This chapter is organized in four sections: Section 5.1 gives a brief introduction and reviews previous work on spatial smoothing techniques. Section 5.2 presents a parallel spatial processing algorithm as a cure for signal cancellation. Section 5.3 analyzes the proposed algorithm. Finally, Section 5.4 compares the experimental results of both the proposed algorithm and the previous methods.

5.1 Introduction

As previously shown in Chapter II, signal cancellation phenomena exist in many conventional adaptive arrays. These effects can result in a signal loss in the case of narrow-band signals, or significant signal distortion in the case of wide-band signals.

Duvall proposed a composite beamformer to prevent signal cancellation when the signal direction is known. This beamformer however requires identical array elements to perform the inter-element subtraction which removes the desired signal from the adaptive processor. Since identical array elements sometimes are not available, the spatial smoothing techniques here are used as an alternative to combat signal cancellation.

Previous work using spatial smoothing techniques to combat signal cancellation is due to Widrow [5.1] and Shan [1.20]. These methods are briefly discussed below.

5.1.1 Spatial Dither Algorithms

The spatial dither algorithm was first proposed to prevent signal cancellation by Professor Widrow at Stanford University [5.1]. This algorithm applies locally controlled modulation to jammers arriving at angles other than the look direction, while leaving the signal from the look direction unmodulated and undistorted. The effect is to cause jammers arriving off the look direction to be spread spectrally, thereby reducing jammer power intensity.

To visualize the spatial dither algorithm Widrow explained a so called

"3/4-in plywood" approach as shown in Figure 5.1. The elements of an adaptive array are attached to a piece of plywood that provides a rigid support, so that the entire array can be moved mechanically. The idea of this spatial dither algorithm is to randomly dither in directions which are orthogonal to the look direction. Far-field emanations arriving from the look direction will be undistorted by the mechanical motion, while emissions from sources off the look direction will be randomly modulated. Through this random modulation on the jammer, the array can break up the signal/jammer correlation.

Although the mechanical motion is somewhat not compatible with electronic processing, the spatial dither algorithm provides a profound basis for adaptive arrays to combat signal cancellation.

5.1.2 Spatial Smoothing Algorithm

Recently Shan proposed another spatial smoothing approach to eliminate signal cancellation. He first showed that in a coherent signaling environment the sample covariance matrix has some zero eigenvalues. Minimization of mean square error with respect to the weights will steer the weight vector to align with an eigenvector corresponding to a zero eigenvalue. The output of the beamformer hence falls down to zero. With his spatial smoothing method, the array will be able to restore full rank to the sample covariance matrix.

Figure 5.2 shows a picture of Shan's spatial smoothing algorithm. Auxiliary antenna elements are used, and all the elements are partitioned into several groups as shown in Figure 5.2. Note that all the groups for a given snapshot in

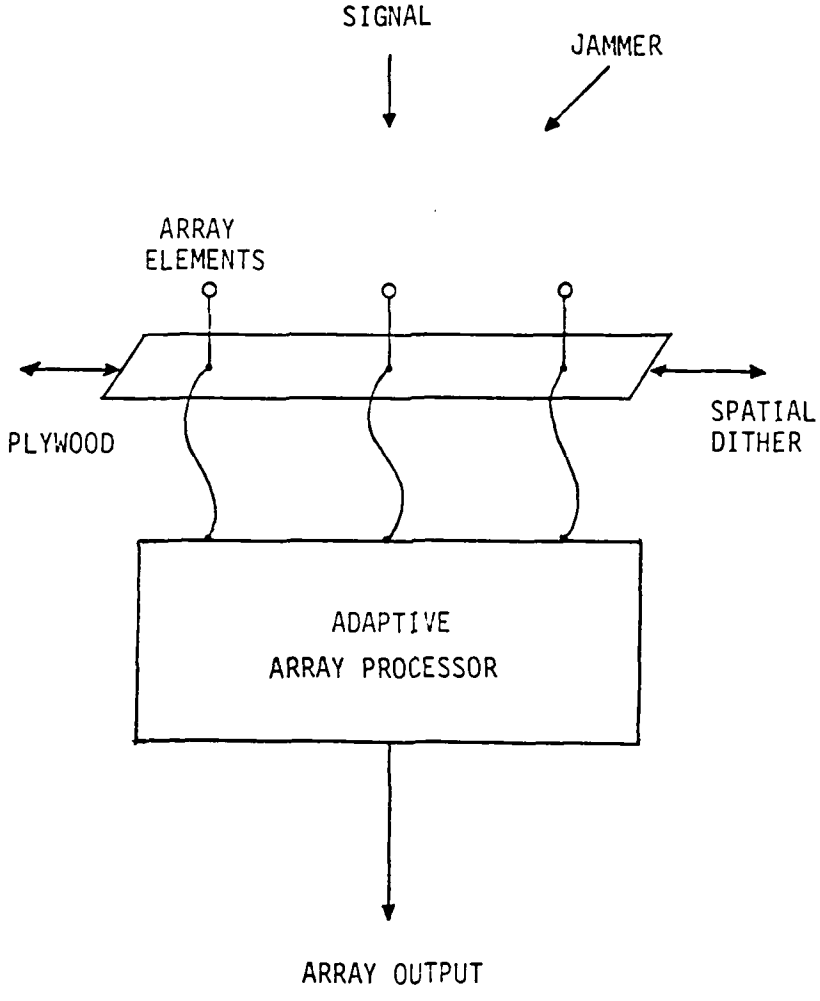


Figure 5.1. Widrow's mechanical spatial dither algorithm ("3/4-inch plywood").

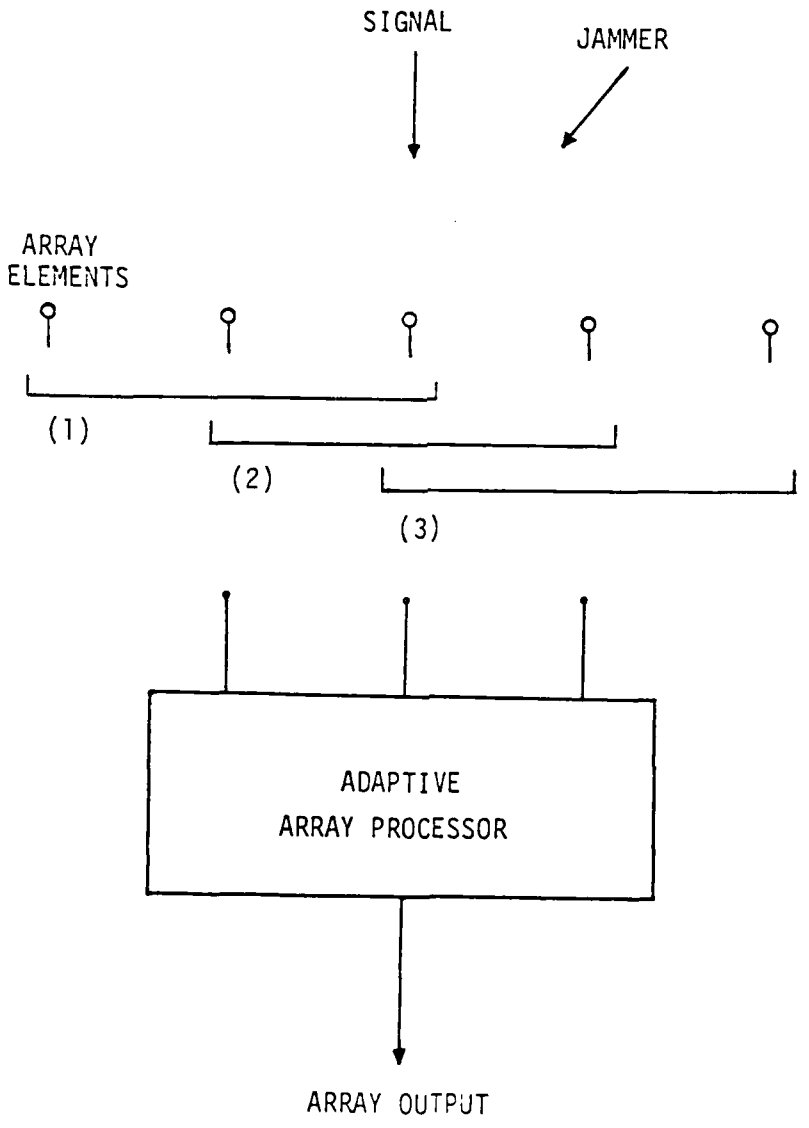


Figure 5.2. Shan's spatial smoothing algorithm.

time still contain the same signal, but the jammers in different groups are in different phase relations. The algorithm then runs these groups one by one into the adaptive array processor. Different running sequences may result in different spatial smoothings, but the effect in breaking up the signal-jammer correlation is still the same.

This method is found effective in applications to direction finding and adaptive beamforming. For many signal cancellation problems, however, the "quality" of the array output rather than its output power is of great concern. The recovered signal however is still sensitive to the adaptation rate, and another form of signal distortion can result from using a high adaptation rate. For each snapshot, this method requires a considerable amount of computation to achieve spatial smoothing.

5.2 Parallel Spatial Processing Algorithm

In this section we present an approach called the "parallel spatial processing algorithm" to combat signal cancellation. In this algorithm, a number of sub-beamformers having the same structures as above are used. These sub-beamformers are arranged in a parallel way. Due to its parallel structure, the algorithm will require the same computation power for each snapshot as any conventional adaptive beamformer.

Figure 5.3 illustrates a general block diagram of the algorithm. It consists of a linear array with L equal-distance elements. These L elements are partitioned into N groups, where N is the number of sub-beamformers. Each sub-beamformer has M input elements. The input elements of adjacent sub-beamformers could be partially overlapping. If the adjacent sub-beamformers have overlapping input elements, every sub-beamformer should do the overlapping in the same way. This implies that the total number of elements in the linear array should be less than or equal to $M \cdot N$. Since every sub-beamformer has the same structure, each one can share the same set of weights.

The parallel spatial processing algorithm is given as follows: For the first snapshot, we use the first sub-beamformer to update the weights and then copy the weights into the rest of the sub-beamformers. For the second snapshot, we use the second sub-beamformer to update the weights and then copy the weights into the rest of the sub-beamformers. So the adaptation process is sequentially propagating one by one along the sub-beamformers. After the adaptation reaches the last sub-beamformer, it restarts from the first one. Meanwhile, for each

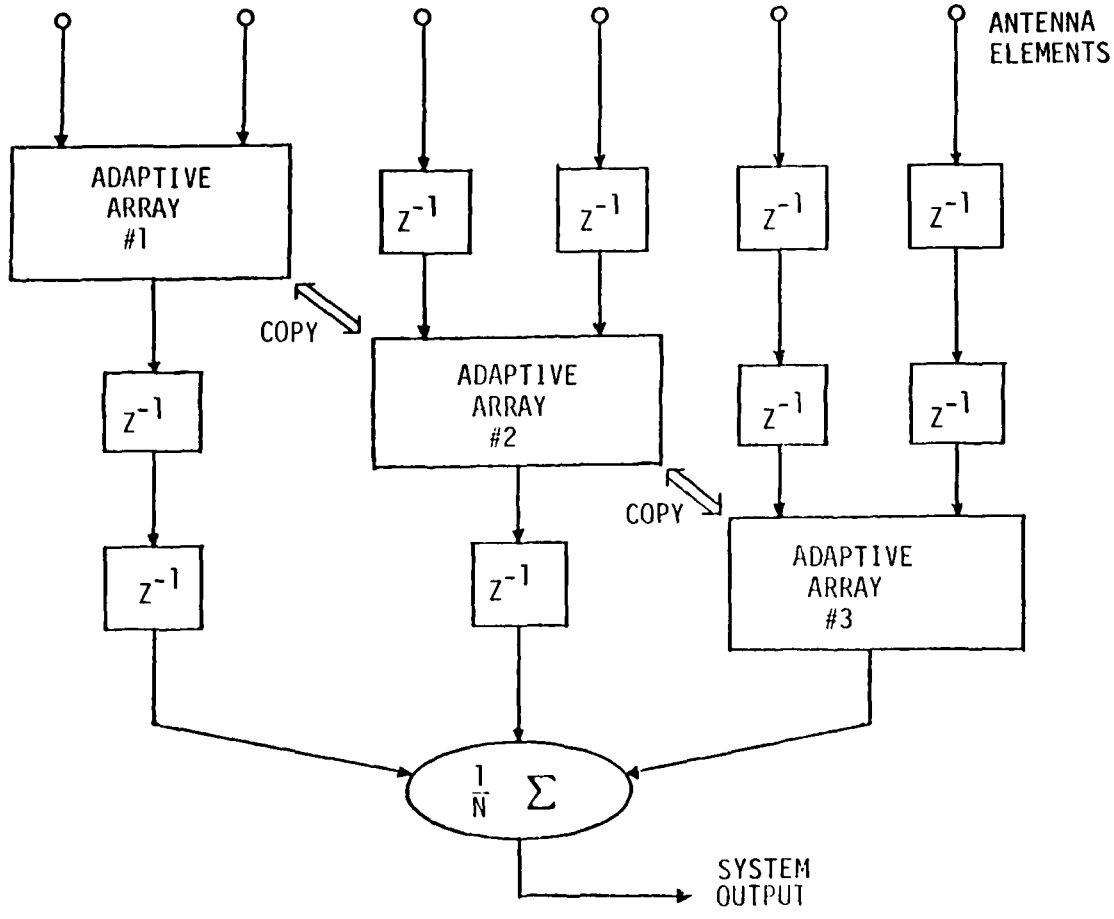


Figure 5.3. A general block diagram for the "parallel spatial processing algorithm."

snapshot, every sub-beamformer uses the same set of weights to yield its own output. The system output is then generated by averaging the various delayed outputs of all these sub-beamformers.

Basically, this shares common merits with Shan's spatial smoothing algorithm. The weight propagation from one sub-beamformer to another will incorporate spatial smoothing as well as time averaging in the sample covariance matrix. Thus, the rank of the signal space would be restored when coherent situations take place. With the parallel structure, the algorithm also provides a better estimate of the desired signal.

Analysis in the next section will show that the algorithm results in a maximum likelihood estimate of the desired signal in a spatial averaging sense. In addition, the algorithm only takes one adaptation to generate one system data output. This contrast to N adaptations in Shan's method. As the name of the algorithm implies, the set of weights is spatially propagated and updated along the sub-beamformers, and the received signals are processed in parallel to produce the system output.

5.3 Analysis

In this section an example with N Frost sub-beamformers is analyzed to give a general insight into the proposed algorithm.

Each sub-beamformer has M input elements. The adjacent sub-beamformers have $M-1$ overlapping elements. This means, there are total $M+N-1$ elements in the linear array. Suppose the desired signal and the jammer are impinging on the array; the signal is from the look direction and the jammer is from an off-look direction. Since the elements of the linear array are equally spaced, each element receives

$$X_m(k) = Ae^{jwkT} + Be^{jwkT+jo+j(m-1)w\Delta}, \quad m = 1, 2, \cdots, M+N-1 \quad (5.1)$$

Denote the signal vector received at the n^{th} Frost sub-beamformer by

$$Z_n(k) \triangleq [\, X_m(k-n+1) \;\; X_{m+1}(k-n+1) \;\cdots\; X_{m+M-1}(k-n+1) \,]^T , \quad (5.2)$$

where m is the labelling number of the first element of the n^{th} sub-beamformer. Mathematically, the algorithm can be expressed as the following,

$$\begin{aligned} \mathbf{y}_n(k) &= W^T(k)\, Z_n(k) \\ W(k+1) &= P\,[W(k) + \mu \mathbf{y}_n(k)\bar{Z}_n(k)] + F \end{aligned} , \quad (5.3)$$

where

$$\begin{aligned} k &= \text{the discrete time index} \\ n &= Mod(k,N)+1 \\ \mathbf{y}_n(k) &= \text{the output of the } n^{th} \text{ Frost sub-beamformer} \\ \bar{Z}_n(k) &= \text{the complex conjugate of } Z_n(k) \\ P,\, F &= \text{the constant vectors of the Frost algorithm} \end{aligned}$$

The system output is generated by averaging the various delayed outputs of each

of the sub-beamformers and is given by

$$\mathbf{y}(k) = \frac{1}{N}\left(\mathbf{y}_N(k) + \mathbf{y}_{N-1}(k-1) + \cdots + \mathbf{y}_1(k-N+1) \right) \quad . \tag{5.4}$$

The linear constraints in the weights is expressed as the following,

$$\sum_{i=1}^{N} W_i(k) = 1 \quad , \qquad \text{for any } k \quad . \tag{5.5}$$

It is easy to find from (5.2)-(5.3) that the output of each sub-beamformer is

$$\mathbf{y}_n(k) = \sum_{i=1}^{M} W_i(k) X_{i+n-1}(k-n+1) \qquad n = 1, 2, \cdots, N \quad . \tag{5.6}$$

For the $k+n-1^{th}$ time instant, one may have

$$\mathbf{y}_n(k+n-1) = \sum_{i=1}^{M} W_i(k+n-1) X_{i+n-1}(k) \qquad n = 1, 2, \cdots, N \quad . \tag{5.7}$$

The overall system output $\mathbf{y}(k+N-1)$ is

$$\begin{aligned} \mathbf{y}(k+N-1) &= \frac{1}{N}\left[\mathbf{y}_1(k) + \mathbf{y}_2(k+1) + \cdots + \mathbf{y}_N(k+N-1) \right] \\ &= \frac{1}{N} \sum_{n=1}^{N} \mathbf{y}_n(k+n-1) \quad . \end{aligned} \tag{5.8}$$

Substituting (5.1), and (5.7) into (5.8), one has

$$\begin{aligned} \mathbf{y}(k+N-1) &= Ae^{jwkT} + Be^{jwkT+j\phi} \cdot \frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{M} W_i(k+n-1) e^{j(i+n-2)w\Delta} \\ &= Ae^{jwkT} + Be^{jwkT+j\phi} \cdot \frac{1}{N} \sum_{n=1}^{N} \alpha(n+k)\, e^{j(n-1)w\Delta} \end{aligned} \tag{5.9a}$$

where

$$\alpha(n+k) = \sum_{i=1}^{M} W_i(k+n-1) e^{j(i-1)w\Delta} \quad . \tag{5.9b}$$

Notice that (5.9) can be a quality measure of the system output. According to

this equation, the system output contains the desired signal plus a coherent jammer which is multiplied by a spatial averaging term. This spatial averaging term may determine whether the whole adaptive beamformer could recover the desired signal or not.

Another interesting thing is that the weights are modulated by the spatial frequency $e^{j(i-1)w\Delta}$ as shown in (5.9b). This modulated term $\alpha(n+k)$ shown in (5.9b) is a function of the time index k. As the adaptive process reaches the minimum of the performance surface, it is very likely that

$$W_i(k+n-1) \cong W_i(k+N-1) \qquad n = 1, 2, \cdots, N-1 \quad .$$

Thus, it is easy to obtain the following

$$\begin{aligned} \mathbf{y}(k+N-1) &= Ae^{jwkT} \\ &+ Be^{jwkt+j\phi} \cdot \alpha(k+N) \cdot \left[\frac{1}{N} \sum_{n=1}^{N} e^{j(n-1)w\Delta} \right] \quad . \end{aligned} \tag{5.10}$$

There are two factors in (5.10) which can modify the jammer. The first factor is a function of time as shown in (5.9b), and is subject to the least-mean-square criterion and the linear constraint. The second term is given as

$$\frac{1}{N} \sum_{n=1}^{N} e^{j(n-1)w\Delta} \quad ,$$

which is the summation of N uniformly-spaced terms on the unit circle. Notice that this results in a very small value, close to zero, and it also asymptotically approaches zero as N goes to infinity. When the adaptive process reaches steady state, the coherent jamming effect will be greatly reduced by such a modification. Therefore, if a large number of sub-beamformers are used, it is easy to get

$$\lim_{N\to\infty} \mathbf{y}(k+N-1) = Ae^{jwkT} \quad . \tag{5.11}$$

If the desired signal is stationary, the expected value of the system output will be a minimum-variance estimate of the signal. Capon *et.al.* [1.5] showed that a minimum-variance estimate is equivalent to the maximum-likelihood estimate. Since the spatial summation factor asymptotically approaches zero, the system output hence is a maximum-likelihood estimate of the desired signal in a spatial averaging sense. To make the spatial summation factor close to zero, the number of sub-beamformers, N, should be large enough so that the terms $e^{j(n-1)u\Delta}$ span the unit circle. This implies that if the incident angle of the jammer from broadside is very small, then a large number of sub-beamformers are required. Finally, the signal estimate appears at the system output with a delay of $N-1$ sampling periods.

Although the analysis is based on the Frost linearly constrained beamformer, any other known adaptive beamformer can be used as the sub-beamformer of the parallel spatial processing algorithm. The spatial averaging effect on the jammer from an off-look direction can still be achieved.

5.4 Simulation Results

Experiments were conducted for the parallel spatial processing algorithm. The structure in Figure 5.3 with four Frost sub-beamformers was simulated in a coherent signaling environment. Each sub-beamformer had three input elements. The adjacent sub-beamformers had two overlapping elements. In other words, the linear array made a total of six elements. Each element was assumed omni-directional, and the inter-element distance was half wave-length. The ambient white noise was assumed negligible. The constraint was set up to be unit gain and zero phase over the frequency band from zero to half the sampling rate in the desired direction. The initial quiescent beam pattern of the proposed scheme is shown in Figure 5.4. In this quiescent beam pattern, some inherent nulls exist in the off-look direction, and sometimes these nulls are referred to as grating nulls. The constraints in the look direction were still preserved.

Now suppose a desired sinusoidal signal arrived from the look direction, and a coherent jammer arrived 45 ° off the look direction. Both the signal and the jammer had equal power intensity of 1. Figure 5.5 shows a beam pattern of the proposed adaptive beamformer when the adaptation process converged. A sharp null with a depth of nearly -70 db was formed in the incoming direction of the jammer. The linear constraint in the look direction was still preserved at unity. The beam pattern resulted as desired. We leave the output to the last experiment.

The next experiment was similar to the first one except that the desired signal now was a wide-band signal and the jammer was still a sinusoid at the

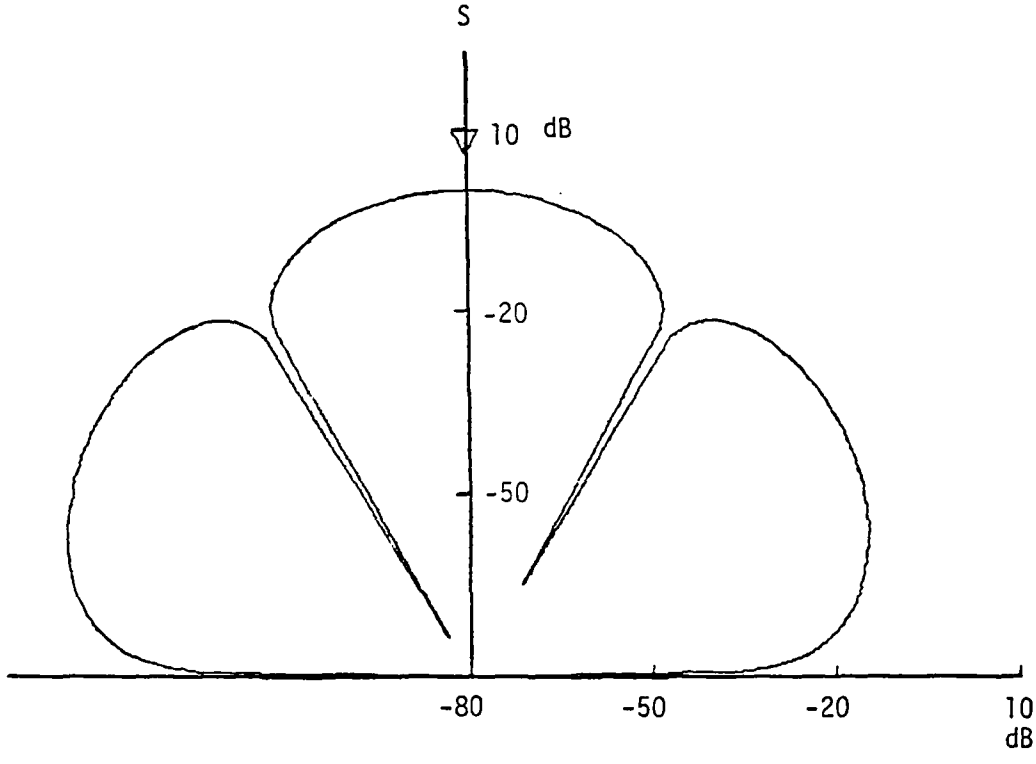


Figure 5.4. A quiescent beam pattern of the proposed scheme using the "parallel spatial processing algorithm."

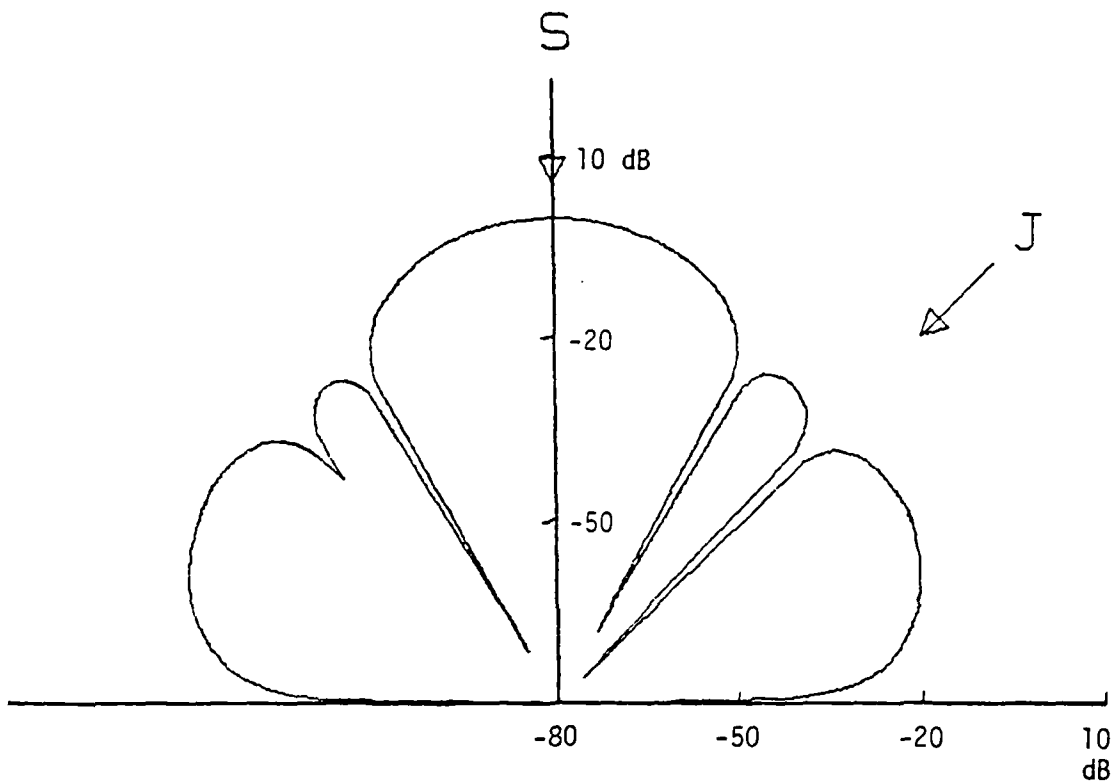


Figure 5.5. A converged beam pattern of the proposed scheme using the "parallel spatial processing algorithm."

center frequency of the signal band. Figure 5.6a shows the power spectrum of the desired signal. Usually, the output spectrum of the Frost beamformer was as shown in Figure 5.6b, where signal cancellation occurred in the jamming frequency band. In contrast, the signal-cancellation-free output spectrum for the parallel processing structure is as shown in Figure 5.6c. One can easily see that the original signal spectrum was recovered without any signal cancellation effect. Figure 5.7 shows the corresponding time waveforms of Figure 5.6. The proposed scheme obviously resulted in a better replica of the desired signal than the conventional Frost beamformer. Note that the output of this parallel spatial processing algorithm was delayed for several sampling periods in contrast to the desired signal. Besides, the transient response of the adaptive process died out after about 60 adaptations.

The final experiment compared the output qualities for both the proposed method and Shan's spatial smoothing method. The desired signal, shown in Figure 5.8a, was set to be of unit amplitude. A strong, coherent jammer arrived off the look direction. Both methods were tested by using the same Frost sub-beamformers running at a high adaptation rate. The beamformer output of Shan's spatial smoothing method is shown in Figure 5.8b, and the output of the proposed method is shown in Figure 5.8c. Apparently, Shan's method introduced some amplitude and phase distortions. For the proposed method, the desired signal was recovered without any distortion, but with a delay of several sampling periods.

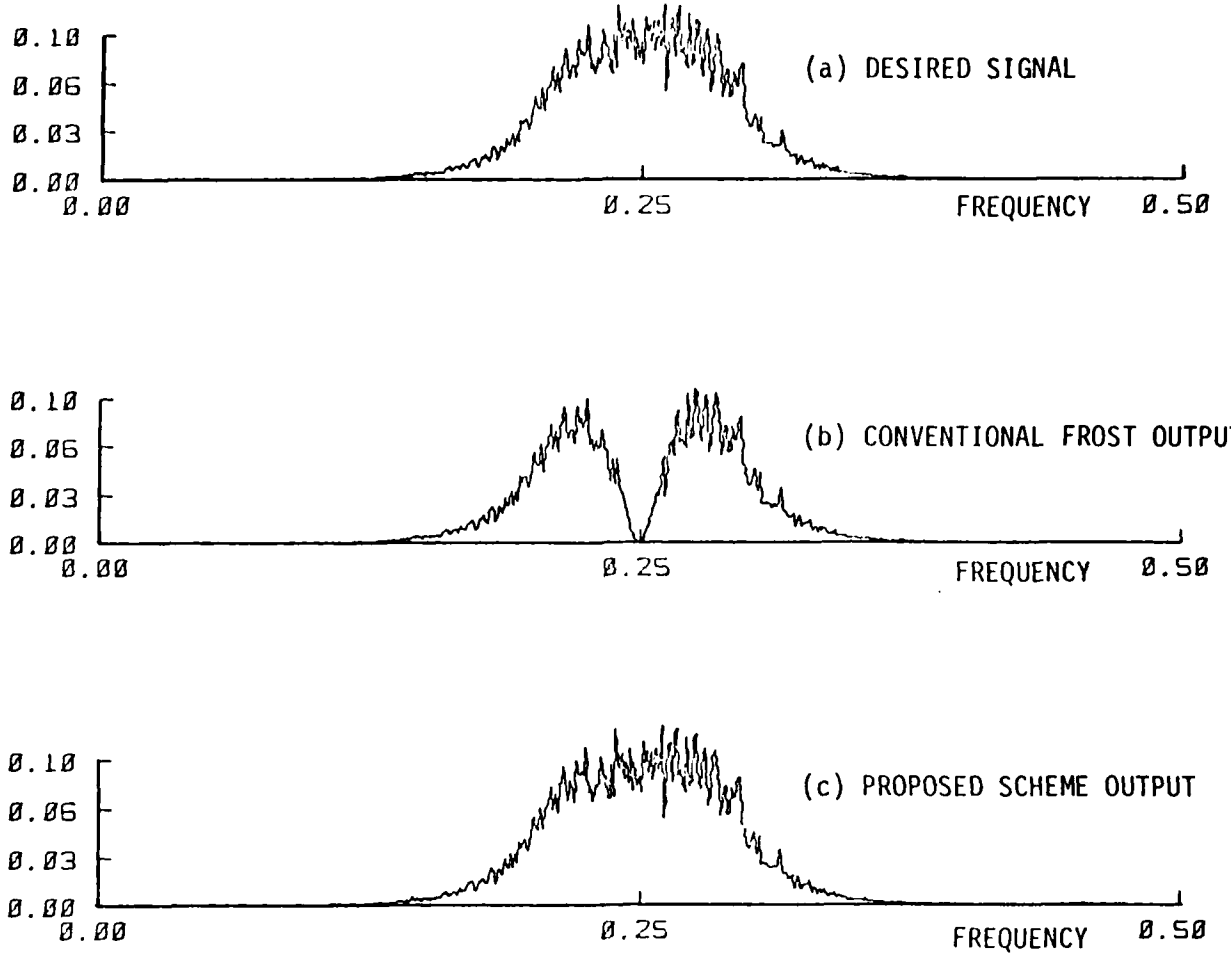


Figure 5.6. Power spectra of a desired signal, the output of a conventional Frost beamformer, and the output of the proposed scheme.

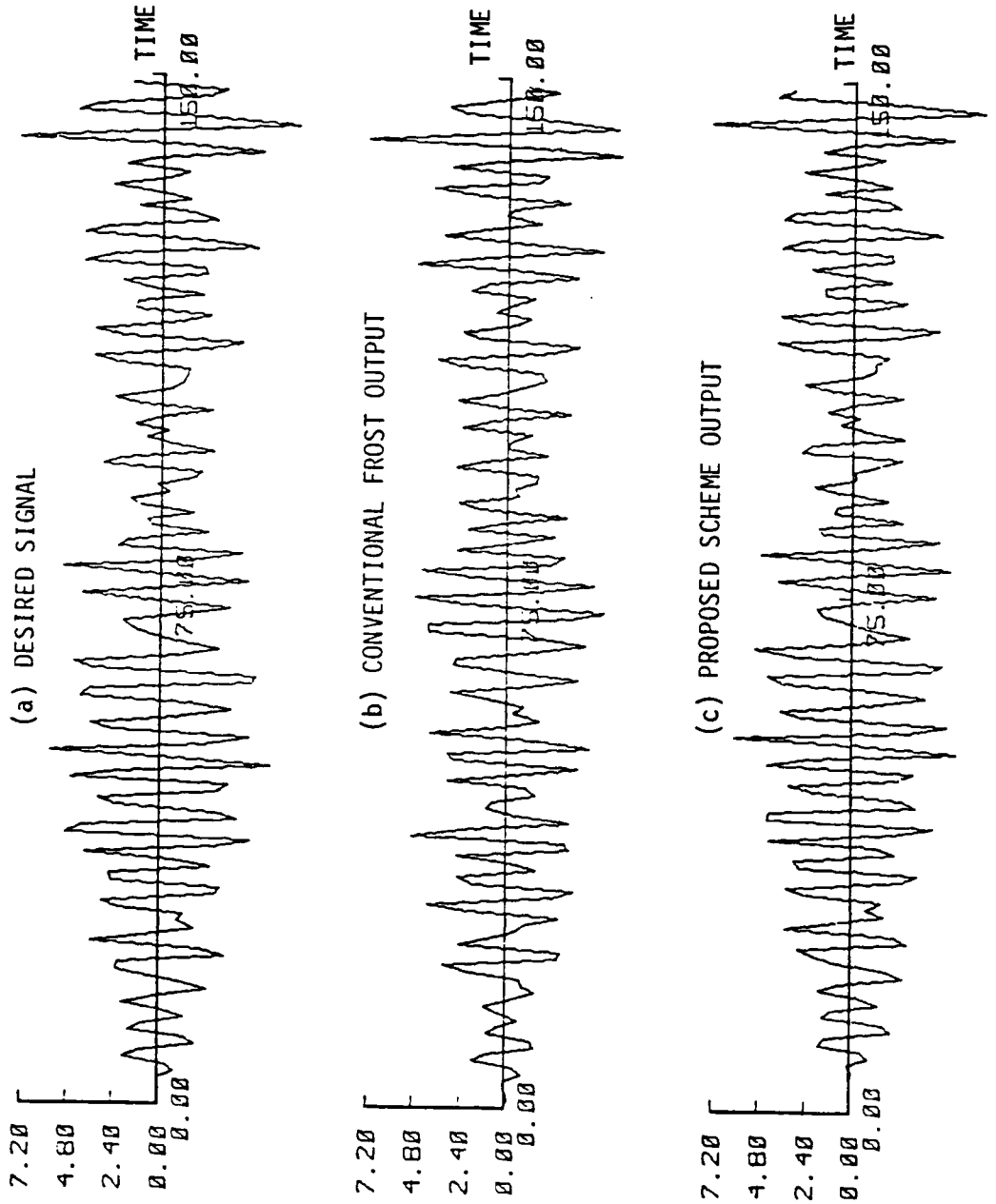


Figure 5.7. The time domain waveforms corresponding to Figure 5.6.

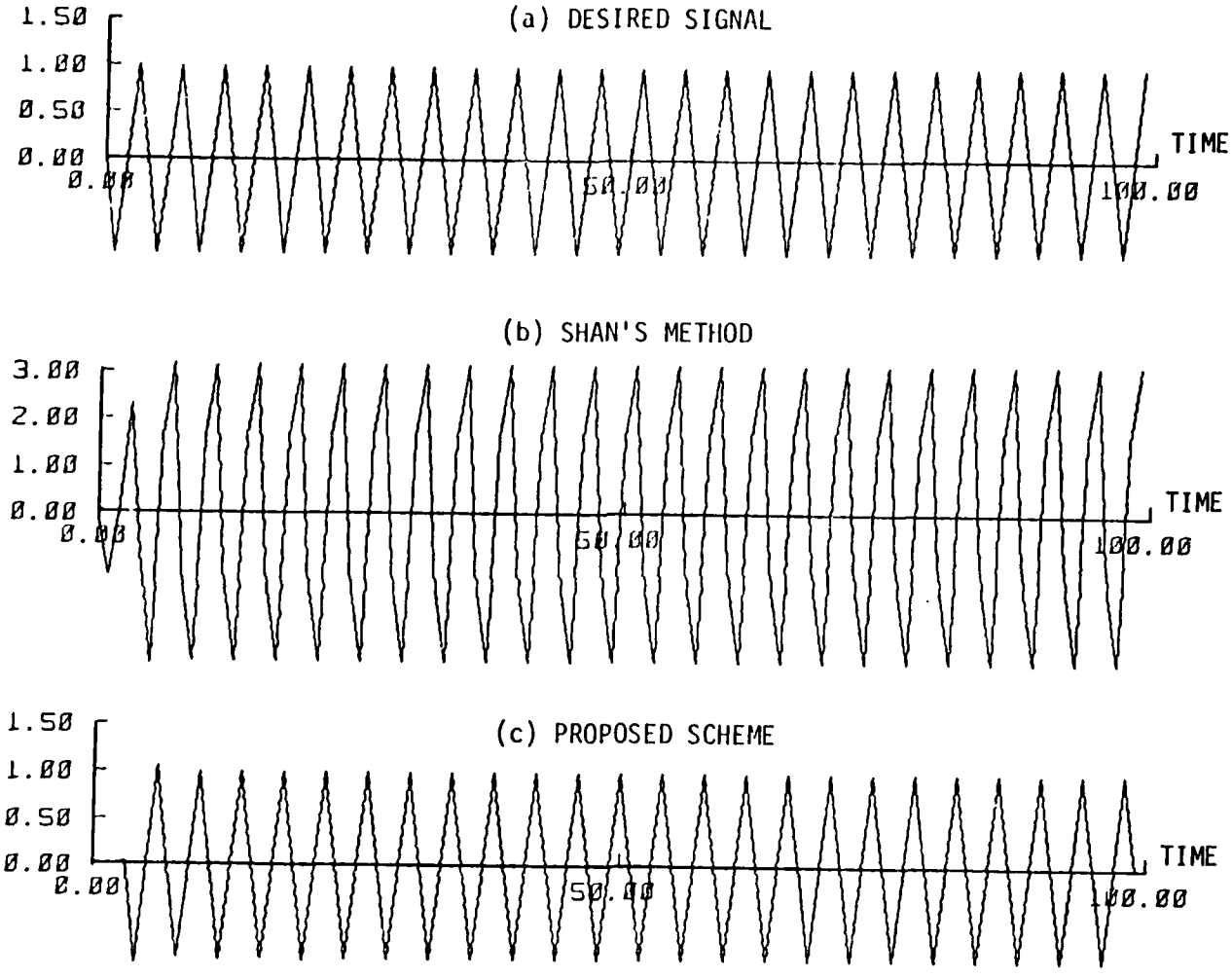


Figure 5.8. Time domain waveforms of the desired signal, the output of Shan's method, and the output of the proposed scheme.

5.4.1 Conclusions

The "parallel spatial processing" algorithm for adaptive arrays was proposed to combat signal cancellation effects in coherent jamming environments. The effectiveness of this algorithm is verified by several computer simulations. The algorithm requires the same computation power as conventional adaptive arrays, although it also requires additional array sensing elements. Analysis shows that the system output results in a maximum-likelihood estimate of the desired signal in a spatial averaging sense.

APPENDIX A

A COMPLEX ALGORITHM FOR LINEARLY CONSTRAINED ADAPTIVE ARRAYS

A.1 Introduction

The so-called "Constrained LMS" algorithm, a simple stochastic gradient descent algorithm with a linear constraint on the adaptive weights has been applied to a variety of problems in geoscience, seismology, and antenna arrays [A.1-A.4]. Algorithms of this type have been devised by Frost [1.10] for implementation with real signals. Following Frost, the adaptive array processor of Figure A.1 has N tapped-delay-lines and L taps per TDL for a total of NL adjustable weights. The NL-dimensional sample vector X at the time of the k^{th} adaptation is

$$X(k) \triangleq [x_1(k\Delta) \;\; x_2(k\Delta) \;\; \cdots \;\; x_{NL}(k\Delta)]^T \quad . \tag{A.1}$$

The NL-dimensional weight vector W is

$$W \triangleq [w_1 \;\; w_2 \;\; \cdots \;\; w_{NL}]^T \quad . \tag{A.2}$$

The output of the array at the time of the k^{th} adaptation is

$$y(k) = W^T X(k) = X(k)^T W \tag{A.3}$$

and the expected output power of the array is

$$E[y^2(k)] = E[W^T X(k) X(k)^T W] = W^T R_{xx} W \tag{A.4}$$

where

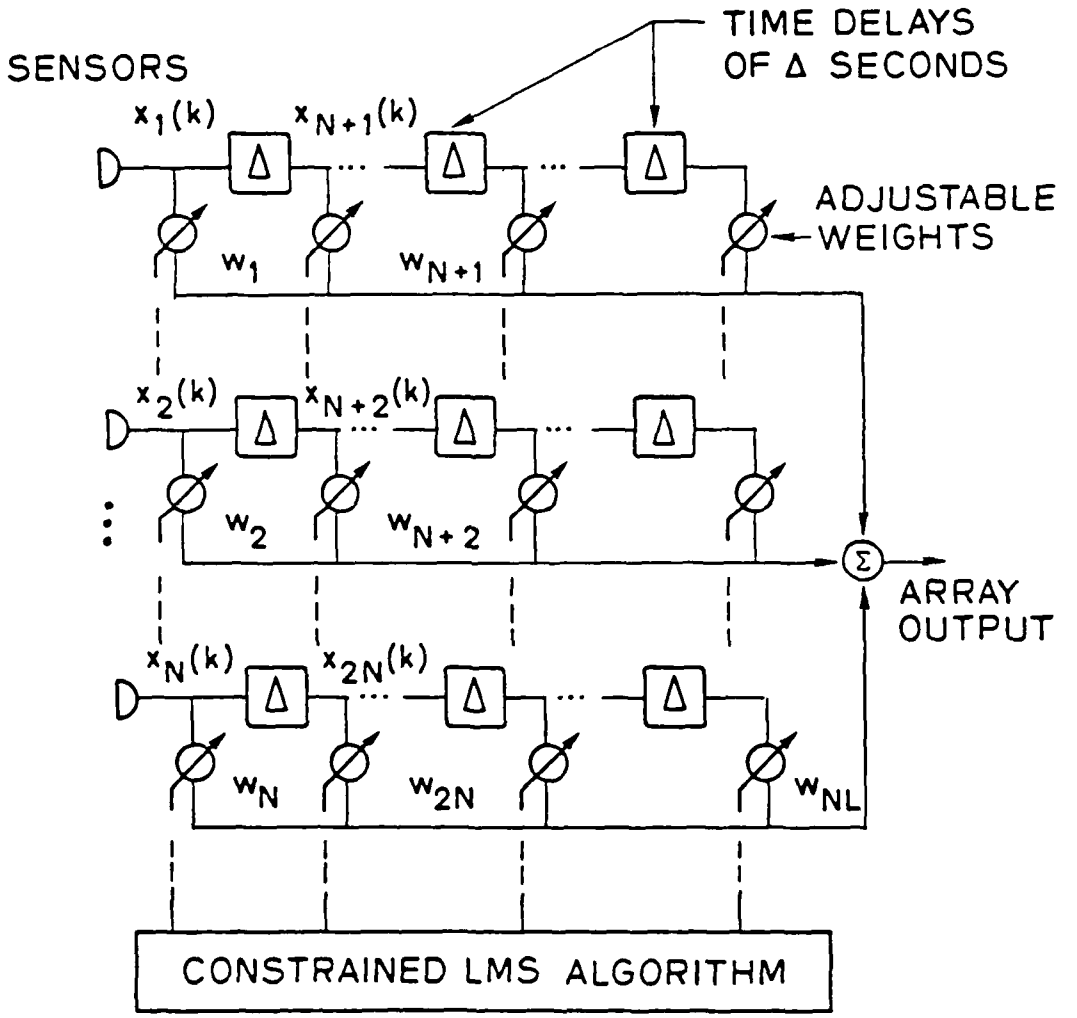


Figure A.1. A detailed structure of the Frost adaptive array.

$$R_{xx} \triangleq E[X(k)X(k)^T] \tag{A.5}$$

is positive definite.

The constraint for desired frequency response characteristic of the target signal is defined in such a way that the linear combinations of the weights on the j^{th} vertical column of taps sums to a constant number f_j as shown in Figure A.2. The requirement is thus given by

$$C_j^T W = f_j \qquad\qquad j = 1, 2, \cdots, L \tag{A.6}$$

where the NL-dimensional vector C_j has the form

$$C_j = [\underbrace{0...0}_{(j-1)N} \quad \underbrace{1...1}_{N} \quad \underbrace{0...0}_{(L-j)N}]^T \qquad j = 1, 2, ..., L \quad . \tag{A.7}$$

Furthermore, we define

$$C \triangleq [C_1 \;..\; C_j \;..\; C_L] \tag{A.8}$$

$$f \triangleq [f_1 \;..\; f_j \;..\; f_L]^T \quad . \tag{A.9}$$

The constraints (A.6) are now rewritten as

$$C^T W = f \quad . \tag{A.10}$$

With (A.4) and (A.10), the optimal weight vector W_{opt} may be obtained by minimizing $W^T R_{xx} W$ over W subject to the constraints $C^T W = f$.

For deterministic gradients, assuming that $C^T W(k+1) = f$, the constrained LMS algorithm, which has been derived in [1.10], may be expressed as

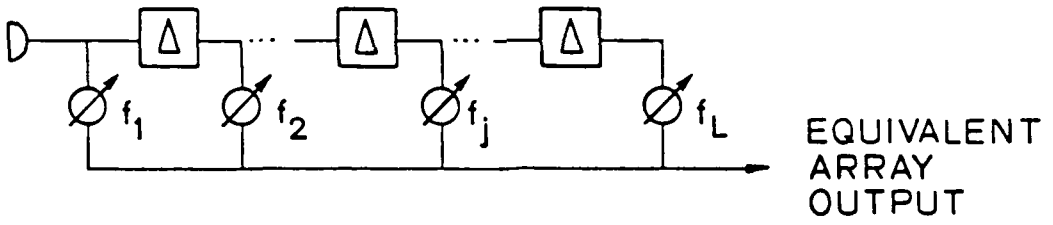


Figure A.2. The equivalent linear constraint filter.

$$\begin{aligned} W(k+1) &= W(k) - \mu \nabla_W H(W(k)) \\ &= W(k) - \mu[R_{xx} W(k) + C\lambda(k)] \\ &= P[W(k) - \mu R_{xx} W(k)] + F \quad , \end{aligned} \tag{A.11}$$

where

$$\begin{aligned} H(W) &\triangleq \frac{1}{2} W^T R_{xx} W + \lambda^T (C^T W - f) \\ P &\triangleq I - C(C^T C)^{-1} C^T \\ F &\triangleq C(C^T C)^{-1} f \\ \mu &= \text{a constant convergence factor} \quad . \end{aligned} \tag{A.12}$$

For stochastic gradients using real data, substituting R_{xx} with $X(k)X(k)^T$ gives the constrained LMS algorithm as

$$\begin{aligned} W(0) &= F \\ W(k+1) &= P[W(k) - \mu y(k) X(k)] + F \quad . \end{aligned} \tag{A.13}$$

A.2 Derivation of Complex Algorithm

Some applications of adaptive arrays require phase delay for the constrained algorithm. For example, complex weights will be needed to render quadrature phase shift for a narrow-band signal at an intermediate frequency. The complex constrained LMS algorithm therefore must be capable of adapting the real and imaginary parts of W simultaneously, minimizing in some sense both $R_e\{y(k)\}$ and $I_m\{y(k)\}$. The complex sample vector $X(k)$ and the complex weight vector W are given by

$$X(k) \triangleq X_R(k) + iX_I(k) \tag{A.14}$$

$$W \triangleq W_R + iW_I \tag{A.15}$$

where the subscriptions R and I denote the real part and imaginary part, respectively. The complex output is correspondingly given by

$$\begin{aligned} y(k) &\triangleq y_R(k) + iy_I(k) \\ &= X(k)^T W = W^T X(k) \quad . \end{aligned} \tag{A.16}$$

According to (A.16) the expected output power of the array is

$$E[\,|\,y(k)\,|^2\,] = E[W^T X(k)\overline{X}(k)^T\,\overline{W}] = W^T R_{xx}\,\overline{W} \tag{A.17}$$

where

$$\begin{aligned} |\,y(k)\,|^2 &= y_R^2(k) + y_I^2(k) \\ R_{xx} &\triangleq E[X(k)\overline{X}(k)^T] \quad . \end{aligned} \tag{A.18}$$

Note that for complex signal cases the covariance matrix R_{xx} is Hermitian and positive definite. Also the complex constraints are given by

$$C^T W = f \tag{A.19}$$

where

$$\begin{aligned} f &= f_R + if_I \\ C &= \text{same as (A.8)} \quad . \end{aligned}$$

The problem of finding the optimum complex weight vector W_{opt} is now formulated as

$$\underset{W}{\text{minimize}} \quad W^T R_{xx}\,\overline{W}$$

subject to $C^T W = f$.

Note that all multiplies and adds are complex. For simplicity, we may form the cost function $H(W)$ by introducing an L-dimensional complex vector of Lagrange multipliers $\lambda = \lambda_R + i\lambda_I$ such that

$$\begin{aligned} H(W) &= \frac{1}{2} W^T R_{xx} \overline{W} + \lambda_R^T (C^T W_R - f_R) + \lambda_I^T (C^T W_I - f_I) \\ &= \frac{1}{2} [W_R^T + iW_I^T] R_{xx} [W_R - iW_I] \\ &+ \lambda_R^T (C^T W_R - f_R) + \lambda_I^T (C^T W_I - f_I) . \end{aligned} \tag{A.20}$$

For gradient descent technique [A.5-A.6], we have

$$\nabla_W H(W) = \nabla_{W_R} H(W) + i \nabla_{W_I} H(W) \tag{A.21}$$

where the gradients of $H(W)$ with respect to the real components and the imaginary components of the weight vector, respectively, are as follows

$$\begin{aligned} \nabla_{W_R} H(W) &= \frac{1}{2} [R_{xx} W_R + R_{xx}^T W_R + iR_{xx}^T W_I - iR_{xx} W_I] + C\lambda_R \\ &= \frac{1}{2} [R_{xx} \overline{W} + R_{xx}^T W] + C\lambda_R \end{aligned} \tag{A.22}$$

$$\begin{aligned} \nabla_{W_I} H(W) &= \frac{1}{2} [iR_{xx} W_R - iR_{xx}^T W_R + R_{xx} W_I + R_{xx}^T W_I] + C\lambda_I \\ &= \frac{i}{2} [R_{xx} \overline{W} - R_{xx}^T W] + C\lambda_I \quad . \end{aligned} \tag{A.23}$$

Since R_{xx} is Hermitian, $R_{xx}^T = \overline{R}_{xx}$. For a constant μ, the constrained LMS algorithm for the $(k+1)^{th}$ adaptation complex weight vector is

$$\begin{aligned} W(k+1) &= W(k) - \mu \nabla_W H(W(k)) \\ &= W(k) - \mu [\overline{R}_{xx} W(k) + C\lambda(k)] \quad . \end{aligned} \tag{A.24}$$

Solving for the Lagrange multipliers vector $\lambda(k)$ by equating $f = C^T W(k+1)$, and substituting into Eq. (A.24), we get

$$W(k+1) = P[W(k)-\mu\overline{R}_{xx}W(k)] + F \tag{A.25}$$

where

$$P = I - C(C^TC)^{-1}C^T$$
$$F = C(C^TC)^{-1}f \quad .$$

Equation (A.25) is a deterministic gradient descent algorithm requiring *a priori* information of the statistics of the signals and noises. For a stochastic gradient descent technique, *a priori* information is neither available nor necessary. Substituting R_{xx} for $X(k)\overline{X}(k)^T$ results in the complex constrained LMS algorithm

$$W(0) = F$$
$$W(k+1) = P[W(k)-\mu y(k)\overline{X}(k)] + F \quad . \tag{A.26}$$

REFERENCES

[1.1] P. W. Howells, "Intermediate Frequency Side-lobe Canceller," U.S. Patent 3,202,990, Aug. 24, 1965 (filed May 4, 1959).

[1.2] P. W. Howells, "Explorations in Fixed and Adaptive Resolution at GE and SURC," *IEEE Trans. Antenna and Propaga.*, vol. AP-24, pp. 575-584, Sept. 1976.

[1.3] Applebaum, "Adaptive arrays," Syracuse Univ. Res. Corp., Rep. SPL TR 66-1, Aug. 1966.

[1.4] B. Widrow, P. E. Mantey, L. J. Griffiths, and B. B. Goode, "Adaptive Antenna Systems," *Proc. IEEE*, vol. 55, no. 12, pp. 2143-2159, Dec. 1967.

[1.5] Capon et. al., "Multi-dimensional Maximum Likelihood processing of a large aperture seismic array," *Proc. IEEE*, vol. 55, no. 2, pp. 192-211, Feb. 1967.

[1.6] R. T. Lacoss, "Adaptive combining of wideband array data for optimal reception," *IEEE Trans. Geosci. Electron.*, vol. GE-6, no. 2, pp. 78-86, May 1968.

[1.7] S. W. W. Shor, "Adaptive technique to discriminate against coherent noise in a narrow-band system," *J. Acoust. Soc. Amer.*, vol. 39, January 1966.

[1.8] L. J. Griffiths, "A Simple Adaptive Algorithm for Real-time Processing in Antenna Arrays," *Proc. IEEE*, vol. 57, no. 10, pp. 1696-1704, Oct. 1969.

[1.9] O. L. Frost, III, "Adaptive Least Squares Optimization Subject to Linear Equality Constraints," Ph.D. Dissertation, Stanford University, Stanford, CA, Aug. 1970.

[1.10] O. L. Frost, III, "An algorithm for linearly constrained adaptive array processing," *Proc. IEEE*, vol. 60, no. 8, pp. 926-935, Aug. 1972.

[1.11] C. W. Jim, "A Comparison of Two LMS Constrained Optimal Array Structures," *Proc. IEEE*, vol. 65, pp. 1730-1731, Dec. 1977.

[1.12] L. J. Griffiths and C. W. Jim, "An Alternative Approach to Linearly Constrained Adaptive Beamforming," *IEEE Trans. Ant. and Prop.*, vol. AP-30, no. 1, pp. 27-34, Jan. 1982.

[1.13] C. L. Zahm, "Application of Adaptive Arrays to Suppress Strong Jammers in the Presence of Weak Signals," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-9, no. 2, pp. 260-271, March 1973.

[1.14] W. F. Gabriel, "Adaptive arrays-An introduction," *Proc. IEEE*, vol. 64, pp. 239-272,Feb. 1976.

[1.15] Special issue on adaptive antennas, *IEEE Trans. Antennas Propagat.*, vol. AP-24, no. 5, Sept. 1976.

[1.16] W. F. Gabriel, "Spectral analysis and adaptive array supperresolution technique," *Proc. IEEE* vol. 68, no. 6, pp. 654-666, June 1980.

[1.17] R. A. Chestek, "The Addition of Soft Constraints to the LMS Algorithm," Ph.D. Dissertation, Stanford University, Stanford, CA, May 1979.

[1.18] B. Widrow *et.al.*, "Signal Cancellation Phenomena in Adaptive Antennas: Causes and Cures," *IEEE Trans. Antennas Propaga.*, vol. AP-30, no. 3, May 1982.

[1.19] K. Duvall, "Research on Adaptive Antenna Techniques V," Final Report, Naval Air Systems Command under Contract N00019-82-C0189, Sept. 1983.

[1.20] T. J. Shan and T. Kailath, "Adaptive beamforming for coherent signals and interference," *Proc. of II IEEE Workshop* on Spectral Estimation, Tampa, FL, Nov, 1983.

[2.1] S. P. Applebaum, and D. J. Chapman, "Adaptive Arrays with Main Beam Constraints," *IEEE Trans. Antennas Propaga.*, vol. AP-24, no. 5, pp. 650-662, Sept. 1976.

[2.2] K. Duvall, "Signal Cancellation in Adaptive Arrays: The Phenomena and a Remedy," Ph.D Thesis, Dept. of Elec. Eng., Stanford University, Sept. 1983.

[2.3] B. Widrow, *et.al.*, "Adaptive Noise Cancelling: Principles and Applications," *Proc. IEEE* vol. 63, no. 12, December 1975.

[2.4] J. R. Glover, "Adaptive Noise Cancelling of Sinusoidal Interference," Ph.D Thesis, Dept. of Elec. Eng., Stanford University, December 1975.

[2.5] Y. L. Su, "A Complex Algorithm for Linearly Constrained Adaptive Arrays," *IEEE Trans. Antennas Propaga.*, vol. AP-31, no. 4, July 1983.

[2.6] T. J. Shan and T. Kailath, "Adaptive beamforming for coherent signals and interference," to be published in *IEEE Trans. Acoust., Speech, Signal Processing,*

[2.7] B. Widrow, J. McCool, and M. Ball, "The Complex LMS Algorithm," *Proc. IEEE* vol. 63, pp. 719-720, April 1975.

[3.1] R. C. Dixon, "Spread Spectrum Techniques," The IEEE Press.

[3.2] R. A. Scholtz, "The Origins Of Spread-Spectrum Communications," *IEEE Trans. Commun.*, vol. COM-30, no. 5, pp. 822-854, May 1982.

[3.3] R. L. Pickholtz, D.L. Schilling, and L. B. Milstein, "Theory of Spread-Spectrum Communications -- A Tutorial," *IEEE Trans. Commun.*, vol. COM-30, no. 5, pp. 855-884, May 1982.

[3.4] R. T. Compton, Jr., "An Adaptive Array in a Spread-Spectrum Communication System," *Proc. IEEE* vol. 66, no. 3, pp. 289-298, March 1978.

[3.5] J. H. Winters, "Spread Spectrum in a Four-Phase Communication System Employing Adaptive Antennas," *IEEE Trans. Commun.*, vol. COM-30, no. 5, pp. 929-936, May 1982.

[3.6] C. E. Cook and H. S. Marsh, "An Introduction to Spread Spectrum," IEEE Communication Magazine, Vol. 21, no. 2, pp. 8-16, March 1983.

[3.7] M. Spellman, "A Comparison Between Frequency Hopping and Direct Spread PN As Antijam Techniques," IEEE Communication Magazine, Vol. 21, no. 2, pp. 37-51, March 1983.

[3.8] M. Dentino, J McCool, and B. Widrow, "Adaptive Filtering in the Frequency Domain," *Proc. IEEE* vol. 66, no. 12, pp. 1658-1659, December 1978.

[3.9] N. J. Bershad and P. L. Feintuch, "Analysis of the Frequency Domain Adaptive Filter," *Proc. IEEE* vol. 67, no. 12, pp. 1658-1659, December 1979.

[3.10] R. R. Bitmeaad and B. D. O. Anderson, "Adaptive Frequency Sampling Filters," *IEEE Trans. Acoust. Speech Signal Processing,* vol. ASSP-29, no. 3, pp. 684-694, June 1981.

[3.11] S. S. Narayan, A. M. Peterson, and M. J. Narasimha, "Transformation Domain LMS Algorithm," *IEEE Trans. Acoust. Speech Signal Processing,* vol. ASSP-31, no. 3, pp. 609-615, June 1983.

[3.12] L. L. Horowitz and K. D. Senne, "Performance Advantage of Complex LMS for Controlling Narrow-Band Adaptive Arrays," *IEEE Trans. Circuit and Systems,* vol. CAS-28, no. 6, pp. 562-576, June 1981.

[3.13] L. Ljung, and T. Soderstrom, "Theory and Practice of Recursive Identification," 1983, The MIT Press.

[3.14] S. Shaffer, "Adaptive Inverse-Model Control," Ph.D Thesis, Dept. of Elec. Eng., Stanford University, August 1982.

[3.15] B. Widrow, J. McCool, and B. Medoff, "Adaptive Control by Inverse Modelling," Conf. Rec. of 12th Asilomar Conference on Circuits, Systems, and Computers, pp. 90-94, Nov. 1978.

[3.16] B. Widrow, D. Shur, and S. Shaffer, "On Adaptive Inverse Control," Conf. Rec. of 15th Asilomar Conference on Circuits, Systems, and Computers, pp. 185-189, Nov. 1981.

[4.1] J. Makhoul, "Linear Prediction: A Tutorial Review," *Proc. IEEE,* vol. 63, no. 4, pp. 561-580, April 1975.

[4.2] J. R. Glover, Jr., "Adaptive Noise Cancelling Applied to Sinusoidal Interferences," *IEEE Trans. on Acoustics, Speech, and Signal Proc.*, vol. ASSP-25, pp. 484-491, Dec. 1977.

[4.3] S. M. Kay and S. L. Marple, Jr., "Spectrum Analysis--A Modern Perspective," *Proc. IEEE,* vol. 69, no. 11, Nov. 1981.

[5.1] Widrow *et.al.*, "Research on Adaptive Antenna Techniques IV," Final Report, Naval Air System Command under Contract N00019-80-C-0483, Sept. 1980.

[A.1] B. S. Byun and A. F. Gangi, "A Constraint-Elimination Technique for Linearly Constrained Array Processing," *IEEE Trans. Geosci. Electron.*, vol. GE-1P, pp. 8-15, Jan. 1981.

[A.2] W. W. Shen, "A Constrained Minimum Power Adaptive Beamformer with Time-varying Adaption Rate," *Geophysics*, vol. 44, pp. 1088-1096, June 1979.

[A.3] L. E. Brennan and I. S. Reed, "Theory of Adaptive Radar," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-9, pp. 237-252, March 1973.

[A.4] R. T. Compton, "An Experimental Four-Element Adaptive Array," *IEEE Trans. Antennas Propaga.*, vol. AP-24, no. 5, pp. 697-706, Sept. 1976.

[A.5] B. Widrow, "Adaptive Filters," in Aspects of Network and System Theory, R. E. Kalman and N. DeClaris, Eds. New York: Holt, Rinehart, and Winston, pp. 563-587, 1970.

[A.6] B. Widrow and J. McCool, "A Comparison of Adaptive Algorithms Based on the Methods of Steepest Descent and Random Search," *IEEE Trans. Antennas Propaga.*, vol. AP-24, no. 5, pp. 615-637, Sept. 1976.

END

FILMED

2-85

DTIC