1980-12

Adaptive image processing and implementation by multiple microcomputer system.

Amir, Haim

Monterey, California. Naval Postgraduate School

http://hdl.handle.net/10945/17505

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ADAPTIVE IMAGE PROCESSING AND IMPLEMENTATION
BY MULTIPLE MICROCOMPUTER SYSTEM

by

Haim Amir

December 1980

Thesis Advisor: T. F. Tao

Approved for public release; distribution unlimited.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Adaptive Image Processing and Implementation by Multiple Microcomputer System		5. TYPE OF REPORT & PERIOD COVERED Ph.D. Thesis: December 1980 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Haim Amir		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE December 1980 13. NUMBER OF PAGES 265
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Adaptive spatial filter
Conjugate gradient
Nonlinear search
Microcomputer
Random priority controller

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This thesis has two parts, both related to the development of smart sensor systems. The first part is a theoretical development of two families of adaptive spatial filters for suppressing background clutters in infrared images and based on the minimization of mean squared error or the maximization of signal to noise ratio criterion. Seven different nonlinear search techniques have been developed for the adaptation process. They have been

applied to two real world infrared test images and exhibit fast convergence rate with no misadjustment. The second part is an experimental development of a multiple microcomputer system which can be a candidate for an on-board processor system. A multiple star, multiple cluster architecture was developed whose intercommunication is managed by a three-level control including central controller, distributed controller and random priority controller. The adaptive spatial filter has been successfully implemented on this system using partitioning for parallel computing.

Approved for public release; distribution unlimited.

Adaptive Image Processing and Implementation by Multiple Microcomputer System

by

Haim Amir
Lieutenant, Israeli Navy
B.S., Technion Israel Institute of Technology, 1972

Submitted in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

from the

NAVAL POSTGRADUATE SCHOOL
December 1980

ABSTRACT

This thesis has two parts, both related to the development of smart sensor systems. The first part is a theoretical development of two families of adaptive spatial filters for suppressing background clutters in infrared images and based on the minimization of mean- squared error or the maximization of signal to noise ratio criterion. Seven different nonlinear search techniques have been developed for the adaptation process. They have been applied to two real world infrared test images and exhibit fast convergence rate with no misadjustment. The second part is an experimental development of a multiple microcomputer system which can be a candidate for an on-board processor system. A multiple star, multiple cluster architecture was developed whose intercommunication is managed by a three level control including central controller, distributed controller and random priority controller. The adaptive spatial filter has been successfully implemented on this system using partitioning for parallel computing.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADRDC	-	Advance read command
ADWTC	-	Advance write command
ANREQ	-	Any request
BHD	-	Bus hold
BPRN	-	Bus priority in
BPRN*	-	Advance bus priority in
BREQ	-	Bus request
CIC	-	Coincidence inhibiter circuit
CLK1,2	-	Clock 1,2
CLPRN	-	Cluster priority in
CLREQ	-	Cluster request
CSPE	-	Cluster star PRN chain enable
CSRA	-	Cluster star request arbiter
DAC	-	Deadlock avoidance circuit
EXREQ	-	External request
ICAAM	-	Intra cluster advance activities monitor
INH	-	Inhibit
IREQ	-	Internal request
MSABT	-	Most significant address bits (5 out of 20)
OSA	-	On board SBC arbiter
PRE	-	Priority Enable
SBC	-	Single board computer
STPRN	-	Star priority in
STREQ	-	Star request
SSEC	-	Star switch enable circuit

ACKNOWLEDGEMENTS

Many individuals and groups have made it possible for me to be able to present this thesis. This involved much time, finances, and instructions as well as encouragement and patience.

I will always be grateful to the Israeli Navy for opening the doors for me to come to America to the Naval Postgraduate School to pursue this course of study.

I would like to thank my parents for their love and training in my early years and even now. My wife, Michal, and my son, Yaniv, have been most encouraging and patient with me and sacrificed some of their desires to allow me to complete my research. That was not an easy task and I appreciate their willingness to encourage me in this way.

My committee members - Professor T. F. Tao, Professor G. L. Sackman, Professor U. R. Kodres, Professor J. P. Powers, Professor B. O. Shubert and Professor H. A. Titus - who worked with me, spent much time and brought me helpful suggestions as well as clarifying many situations for me. This allowed me freedom to bring my project to completion. Professor Sackman was very gracious in allowing me to use his computer at all times of the day or night. My Advisor, Professor Tao, spent endless hours with me to complete this research. Commander McGonigal and Lt. Commander Harvey were there when I needed them during my time of adjusting and facing difficult situations. I would also like to thank Chuck Sproull, Kurt Holmquist and Bill Cousain Simon for their assistance.

As I look back over the years I would like to thank Mrs. Atara who constantly encouraged me to move ahead in my education.

My neighbors, Bob and Ruby Ramsay, became like parents to us and their family our family, adding pleasure to our stay in the United States.

Last but not least, I wish to thank Miss Schow who typed this thesis for me.

I. INTRODUCTION

A. OBJECTIVES

1. Dual Objectives of this Thesis

This thesis consists of two closely related studies.

a. The first study is the theoretical development of adaptive image processing algorithms for enhancement of "target signal" to "clutter noise" ratio in images. It will be used in the first step of a multiple-stage image processing program for detection of dim targets in noisy infrared images.

b. The second study is an experimental development of a multiple microcomputer system for implementation of these adaptive image processing algorithms.

These two studies belong to two different technical areas. Either topic could be the subject of one thesis project. However, they are investigated together in this thesis because of the special nature of a new emerging field which inspired the research undertaken by this project. This new field is sometimes known as the "Smart Sensors" [1, 2, 3]. Its developments got into high gear only in the late 1970's when advances in two integrated circuit fields, VLSI digital electronics and mosaic optical sensor arrays, were joined together to develop new optical sensors which also have sophisticated on-board signal/data processing capabilities.

In other words, they are SMART-SENSORS. Their importance is closely associated with the coexistence of "sensing" and "processing" capabilities on a small volume, light weight, low power platform. Therefore, the successful development of "smart sensor" systems includes not only new signal/data processing algorithms to provide the needed "smartness" but also efficient implementation by signal/data processors whose size, weight, power and performance are compatible with the requirements of on-board equipment in many practical military systems.

2. Multi-Dimensional "Smart Sensor" Signal Processing

In most optical smart sensor systems, signals of interest are in the form of images. If the field of view of the sensor platform is not stabilized, or locked onto a target, successive frames of images are not registered. Signal processing can only use single frames of an image. Therefore, the signal is two dimensional in terms of the spatial variables x and y. If sensors in several spectral bands are available and well registered spatially the signals are three dimensional in terms of variables, x, y and λ.

In many other smart sensor systems, the field of view of the sensor platform either does not change (as in a synchronous orbit satellite with staring sensors) or is stabilized (as in aircraft with step-staring sensors) or is locked onto a target (as in missiles after they have already acquired a target). In these cases, successive images

are registered. Both single frames of images and multiple frames of images are available for signal processing. The signal is then three dimensional in terms of x, y and t. In addition, if multi-spectral sensors are registered, the signal is four dimensional in terms of x, y, t and λ.

Therefore, signal processing operations required for smart sensors are often multi-dimensional. This thesis is concerned with adaptive spatial filters processing infrared images. This type of spatial filter should be distinguished from the majority of image processing methods which are concerned with the image itself as the signal of interest. Our primary goal is concentrated in the targets. The image itself, often called the background clutter, is considered as noise and must be suppressed so that dim target signals can be revealed to allow the application of a threshold to initiate the detection process. In addition to the clutter, the image may include other noise and man-made interference and jamming also, which are all treated as noise. Only targets are considered as signals.

3. Multiple Stages "Smart Sensor" Signal Processing

To accomplish the objectives of most smart sensor systems in detecting, tracking and recognizing very dim targets deeply buried in noise, a multiple stage image processing approach is generally needed (Table I.1).

TABLE I.1

IMAGE PROCESSING STAGES

Objective in Various Stages		Processing
Enhancement	Pre-threshold	Hard Limiting Adaptive Filtering
Detection	Threshold	Adaptive threshold Target Acquisition
Tracking Recognition	Post-threshold	Kalman Tracker Target Recognition

For more detail, see Chapter III.B.2.

This thesis will concentrate on the development of new adaptive filter techniques which will be used in the "Enhancement" stage to improve the "target signal" to "background clutter noise" ratio by either suppressing the background clutter or enhancing the target signal, or both.

B. STATISTICAL IMAGE PROCESSING TECHNIQUES FOR ENHANCEMENT OF "TARGET SIGNAL" TO "BACKGROUND NOISE" RATIO IN INFRARED IMAGES

1. Introduction

Although the responsibility of detecting very dim targets is shared by several steps of image processing in pre-threshold, threshold and post-threshold stages, the "enhancement" step before thresholding plays a very important role because it is necessary to improve the "target signal" to "clutter noise" ratio to approximately one before a

threshold operation can be applied. Otherwise, there will be too many false alarms collected by the thresholding step, which makes post-threshold signal processing difficult. Therefore, in theoretical developments of new image processing techniques for smart sensors, a great deal of attention is given to background clutter suppression techniques for enhancement of the signal to noise ratio before the thresholding step.

We have made a survey of these techniques and present them in several classifications in Table I.2. First, they are classified as nonadaptive, open loop adaptive and closed loop adaptive. By "nonadaptive," we refer to those approaches whose filters are not designed by using the image characteristics. However, in two adaptive cases, the filters are tailor-designed based on the characteristic learned from the images being processed. In the open loop adaptive case, the filter is not able to update or correct itself when the characteristics of the image are changed. The image properties must be "relearned" before a redesign of the filter can be made. In the closed loop adaptive case, a feedback process is provided between the filter output and the input to the design process. In this way, any change in the image characteristics will result in an increase of the output error which is used to automatically update and correct the filter design.

TABLE 1. 2

FOCAL PLANE PROCESSING TECHNIQUES FOR BACKGROUND CLUTTER SUPPRESSION

FOCAL PLANE PROCESSING ALGORITHMS				ACTIVE GROUPS	
NONADAPTIVE	DETERMINISTIC	SPATIAL	1st order, 2nd order (Laplacian) 4th order nonrecursive spatial filter	MIT Lincoln* Laboratory	
		TEMPORAL	Frame to frame differencing: (Nonrecursive temporal filter) 1st and 2nd differencing 3rd differencing	Grumman Rockwell Hughes	
		SPATIAL-TEMPORAL	Three dimensional spatial-temporal filter by variational method	Rockwell	
			Pseudo-reticle nonrecursive spatial filter followed by recursive temporal bandpass or highpass filter	Optical Science	
		SPATIAL-SPECTRAL	Nonrecursive spatial filter followed by two color discrimination	MIT Lincoln* Laboratory	
OPEN LOOP ADAPTIVE	DETERMINISTIC	SPATIAL	Background normalization (Localized adaotive threshold)	General* Electric	
		TEMPORAL	Bandpass filter followed by adaptive threshold	Aerojet * ElectroSystems	
			2nd, 3rd order recursive temporal highpass filter	Rockwell	
	STATISTICAL	SPATIAL	Minimization of mean square error: Recursive Kalman filter (spatial)	Grumman, NPGS	
			Nonrecursive Wiener filter(soatial)	Lockheed	NPGS
			Maximization of signal to noise ratio: Nonrecursive spatial match filter	MIT Lincoln* Laboratory	NPGS
		TEMPORAL	Maximization of Likelihood ratio	Aerospace Corp	
			Minimization of mean square error: Nonrecursive temooral Wiener filter	Lockheed	NPGS
			Recursive temporal Kalman filter		
			Maximization of signal to noise ratio	Hughes	NPGS
		SPATIAL-TEMPORAL	Minimization of mean square error		NPGS
			Maximization of sional to noise ratio		NPGS
CLOSED LOOP ADAPTIVE	STATISTICAL	SPATIAL	Minimization of mean square error: Nonrecursive spatial filter		NPGS
			Maximization of signal to noise ratio		NPGS
		TEMPORAL	Minimization of mean sauare error		NPGS
			Maximization of signal to noise ratio		NPGS

* Techniques developed for tactical systems

These approaches are further classified as deterministic and statistical. In deterministic cases, the filter design is based on non-statistical properties of the image, such as its frequency characteristics. In statistical cases, the filter design is based on statistical properties of the image, such as its autocorrelation or power spectral density.

Furthermore, they are classified according to the types of signal processing operations used: spatial, temporal, spectral or some of their combinations.

2. Open Loop Adaptive Filter

In our research group, several nonrecursive adaptive open loop adaptive filters have been developed. D. Bar Yehoshua [4] first developed the nonrecursive statistical spatial filters designed by a minimization of mean squared error criterion using theoretically generated images based on both the first and second order Markov models. These images are all assumed to have zero mean. D. Hilmers [5] extended these spatial filters to process real world images which have non-zero mean. Further, he extended the same concept to nonrecursive statistical temporal filters. B. Evenor [6] made two additional extensions. First, he developed the design procedures for spatial filters based on the maximization of signal to noise ratio. Second, he developed a closed loop adaptive spatial filter by extending the LMS (least mean square) algorithm used by many one dimensional adaptive filter researchers. It will be discussed further in the next section.

Using several real world infrared test images, these open loop adaptive filters have been found to be very effective in suppressing background clutter for point targets. However, they are not responsive to any change in the characteristics of the image being processed.

3. Closed Loop Adaptive Filter and this Thesis

The realization of this lack of true adaptive capability led to the study of B. Evenor [6] who developed the non-recursive closed loop adaptive spatial filter based on the "LMS" algorithm, and tested this approach by theoretically generated image using Markov models. However, it was discovered that the LMS algorithm is actually a simplified version of a more general and powerful family of closed loop adaptive filters. It was decided that the first part of this thesis would be to develop such a general adaptive filter approach which includes:

- Two optimization criteria:
  Minimization of mean square error
  Maximization of signal to noise ratio
- General adaptation equation using gradient search models
- A family of nonlinear searching techniques to carry out the adaptation process.

The details of this theoretical study will be presented in Chapter II.

C. IMPLEMENTATION OF THE IMAGE PROCESSING PROGRAM BY A MULTIPLE MICROCOMPUTER SYSTEM

1. Introduction

A parallel effort has been made in the investigation of practical implementation of these statistical nonadaptive image processing algorithms developed in our research group. G. Hilimitzas [7] first investigated the execution speed and accuracy of these image processing algorithms on a main frame computer, IBM 360/67.

2. Microcomputer Implementation

D. Becker [8] investigated the performance of implementation of the nonadaptive image processing algorithms on one 16 bit LSI-11 microcomputer and a combination of this LSI-11 microcomputer and a microcomputer compatible CDA-MSP-3 array processor. It was found that using high order language programming and floating point data format, today's microcomputer implementation is still in its infancy. Its execution speed is slow and not anywhere near any real time processing requirements. Improvements in microcomputer implementation by using assembly language programming, integer data format and improved programming on array processor are currently being developed.

3. Multiple Microcomputer Implementation and this Thesis

It is obvious that to achieve real time image processing performance using microcomputers, several improvements should be considered simultaneously. First, the processing

capability of individual microcomputers must be improved by more imaginative programming and by using attached special processors, such as the array processor. Second, and probably much more important, is to take advantage of the rapidly increasing number of microcomputers affordable in a system by cleverly orchestrating them into an effective concurrent parallel and pipeline execution of the whole image processing program. The advantages offered by the type of multiple microcomputer approaches do not stop at faster execution only, but also include multi-tasking, higher reliability because of better fault tolerance. It was decided that to fully meet the needs of new research for the successful development of a smart sensor, a second part of this thesis should address the implementation issue of image processing algorithms by a multiple microcomputer system. Its details will be presented in Chapter III.

D. SCOPE AND EXTENSION OF THIS THESIS

It should be strongly emphasized that although this thesis specifically developed a family of adaptive spatial filters for the enhancement of target signal to noise ratio of images and a multiple microcomputer system for the implementation of the image processing, the motivation of this thesis is to contribute to the development of smart sensor systems. Therefore, the adaptive filter concepts and design techniques are not limited to spatial filters only. They can

be readily extended to a wide class of problems of poor signal to noise ratios. The implementation issue is not limited to adaptive filter processing only. The multiple microcomputer system is designed to implement not only the mission signal processing but also a host of other signal/ data processing tasks for management, command, control and communication functions.

II. ADAPTIVE IMAGE PROCESSING

A. INTRODUCTION

1. General

The idea of an adaptive filter is inherently attractive. It does not take any stretch of imagination to see a myriad of advantages offered by an adaptive filter which can automatically update itself when it is not performing according to an optimum criterion. The development of adaptive filters started in the early 1960's when it was extended from the sampled data control system [9] and when it was developed for adaptive antenna applications [10]. In ensuing years, a large number of investigations were made for applications in antennas [11], noise cancellation [12] and a variety of filtering applications [13-48].

It is natural that adaptive filter concepts are very attractive for the objective of this thesis--to detect very dim targets deeply buried in infrared background clutter. However, a survey of adaptive filter research published in the 70's reveals the following facts:

a. Practically all of the past adaptive filter research dealt with one dimensional problems.

b. LMS (least mean square) error has been the most widely used criterion. Very little attention has been given to other criteria, such as the maximization of output signal to noise ratio which is probably better suited for detection problems.

c. Very little attention has been given to the convergence speed issue of adaptive filters.

Therefore, we decided to address these three issues and develop new adaptive image processing techniques which are multi-dimensional, using either the mMSE (minimization of mean square error) or the MSNR (maximization of signal to noise ratio) criterion, and using a family of nonlinear convergence techniques developed in the optimization field to search for the extremum in the adaptive process.

However, the basic concept of the adaptive filter and the traditional LMS approach will be briefly reviewed first as a starting point to introduce new techniques developed in this thesis.

2. Basic Concepts of Adaptive Filters

The basic concepts of an adaptive filter can be described concisely as follows:

The filter is represented by a vector $\underline{H}$. In an adaptive filter, $\underline{H}$ is updated in successive iteration steps described by a subscript as $\underline{H}_K$, $\underline{H}_{K+1}$. A correction term, $\underline{\Delta H}_K$, is generated in each iteration step such that

$$\underline{H}_{K+1} = \underline{H}_K + \underline{\Delta H}_K \qquad (1.0)$$

The iteration steps are carried out to optimize a selected performance function until the filter converges to its steady state which also corresponds to the reaching of an extremum of the performance function surface.

The filter $\underline{H}$ could be a temporal filter or a spatial filter. It could be a recursive filter, also called infinite impulse response (IIR) and zero/pole filter, or a nonrecursive filter, also called finite impulse response (FIR), and all zero filter.

The performance function could be the mean square error, or the output signal to noise ratio, or other functions such as the likelihood ratio. The optimization objective could be either the minimization or maximization.

In this thesis, two dimensional spatial filters are considered. They are the nonrecursive type. Two types of cost functions are used. Their optimization objectives are shown in the following table.

TABLE II.0

OBJECTIVE FUNCTIONS

Adaptive filter	Performance Function	Optimization Goal
mMSE	Mean Square Error	Minimization
MSNR	Output Signal to Noise Ratio	Maximization

Let us consider a nonrecursive spatial filter of a filter area of 3 by 3 pixels which has nine filter coefficients. The cost function is a surface in a nine dimensional space. The goal of the iterative adaptation procedure is to search for the coordinates (filter coefficient space) for the extreme point (either a minimum or a maximum) of the performance function surface.

3. Traditional Approach - LMS Algorithm

An overwhelmingly large portion of the past adaptive filter studies followed the approach originated by Professor B. Widrow [14], and commonly known as the LMS (least mean square) algorithm.

The performance function used in this approach is the "mean square error." The optimization goal is "minimization." Prof. Widrow proposed that the adaptation term $\Delta\underline{H}$ be expressed as:

$$\Delta\underline{H} = 2\mu\varepsilon\underline{X}$$

where $\underline{X}$ = signal being processed

2μ = a constant, called adaptive gain

ε = adaptation error = $d - H^T X$

d = reference (or desired signal)

$\underline{H}$ = filter coefficient vector.

The adaptation equation is then

$$\underline{H}_{K+1} = \underline{H}_K + 2\mu\varepsilon\underline{X}$$

A steepest descent search technique is then used for performing the adaptation steps.

Although this traditional LMS approach has been used by most of the adaptive filter researchers, it is not without certain drawbacks which will be briefly described as follows.

The adaptation equation used in the traditional approach can be considered as a special case of a more general adaptation equation,

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \underline{G}_K$$

an equation commonly used in the field of optimization. The term $\underline{G}_K$ is sometimes called the "gradient" meaning the gradient of the performance function surface. The term α_K is sometimes called the "step size" meaning the displacement in the vector space $\underline{H}$. The optimization procedure at iteration step K+1 gave a filter vector $\underline{H}_{K+1}$ which is closer to the optimal vector $\underline{H}$* than previous filter vectors. Therefore, Prof. Widrow's imaginative proposal can be interpreted as the following two assumptions:

$$\underline{G}_K \leftrightarrow 2\varepsilon\underline{X}$$

$$\alpha_K \leftrightarrow \mu = \text{a constant.}$$

These two bold assumptions probably have resulted in several inherent limitations.

a. Because the gradient $\underline{G}_K$ is not tailored to the performance function, convergence could be slow. Further, the steady state filter result may not yield the best estimation. Possibly, a steady state misadjustment could exist [24].

b. Because the step size α_K is assumed to be a constant, the adaptation procedure may never reach a steady state.

4. This Thesis Research

In view of the results of the survey and review of the status of the adaptive filter approach as presented above, we identified a series of research problems which must be

investigated in order to develop adaptive image processing techniques for suppressing background clutter in infrared images and for helping the detection of dim targets.

First, we must extend the one dimensional adaptive filter techniques based on the mMSE criterion to two dimensions.

Second, we should develop an adaptive filter based on the MSNR criterion which is presented in section B.

Third, we should develop a new adaptive equation which is more responsive to the performance function in order to improve convergence speed and to minimize steady state misadjustment. In other words, the adaptive equation is in the form of

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \underline{G}_K$$

The step size α_K and gradient $\underline{G}_K$ will not take the form of 2μ and εX as is customarily done in practically all of the past adaptive filter studies based on the LMS algorithm.

Fourth, we will investigate a variety of non-linear gradient techniques to search for the minimum in the case of mMSE filter and the maximum in the case of MSNR filter. They are derived and presented in sections C and D, respectively.

The results of applying these adaptive spatial filters to two real infrared images will be presented in section F.

B. DERIVATION OF OPTIMIZATION CRITERIA

1. Performance Function I - mMSE

The performance function based on the mMSE criteria is derived along with the nonrecursive spatial/temporal filter. The nonrecursive spatial and temporal filters are described by a set of filter coefficients, vector $\underline{H}$ over the area of a "search-box"[1]. The observed signal in the "ith" "search-box" is represented by the signal vector $\underline{X}_i$. The estimated target intensity within the search-box $\hat{S}_i$ is obtained by the linear filter

$$\hat{S}_i = \underline{H}^T \underline{X}_i \tag{2.00}$$

This process is carried out throughout the whole image.

The nonrecursive filter is represented by the vector[2]

$$\underline{H}^T = [H(1), H(2), \ldots, H(N)] \tag{2.01}$$

where N is the number of pixels in the filter "search-box". The image signal within the "ith" filter "search-box" is described by the vector:

$$\underline{X}_i^T = [X_i(1), X_i(2), \ldots, X_i(N)] \tag{2.02}$$

Throughout this thesis, matrices will be denoted by a "~" under the symbol. Vectors will be denoted by a "_" under the symbol.

The estimation error is defined as:

[1] See Fig. 2.0

[2] T denotes the transpose of the vector.

$$\varepsilon_i \triangleq \hat{S}_i - S_i \tag{2.03}$$

where S_i is the signal and $\hat{S}_i$ the estimated signal in the "search-box".

The mMSE (minimization of mean square error) performance function is defined as:

$$J \triangleq E[\varepsilon_i \cdot \varepsilon_i^T] \tag{2.04}$$

where E[•] denotes the expected value. Substitution of (2.00) and (2.03) into (2.04) gives:

$$\begin{aligned} J &= E[(\underline{H}^T\underline{X}_i - S_i)(\underline{H}^T\underline{X}_i - S_i)^T] \\ &= E[\underline{H}^T\underline{X}_i\underline{X}_i^T\underline{H} - 2\underline{H}^T\underline{X}_iS_i + S_i^2] \end{aligned} \tag{2.05}$$

Since the filter value is fixed for an image, it can be moved out of the expectation operation to give:

$$J = \underline{H}^T E[\underline{X}_i\underline{X}_i^T]\underline{H} - 2 \cdot \underline{H}^T \cdot E[\underline{X}_iS_i] + E[S_i^2] \tag{2.06}$$

In order to simplify (2.06), the following terms are defined:

(1) The autocorrelation matrix $\underset{\sim}{R}_{XX}$ of the observed image is:

$$\underset{\sim}{R}_{XX} \triangleq E[\underline{X}_i\underline{X}_i^T] \tag{2.07}$$

Being a correlation matrix, it is a symmetric and positive definite matrix.

(2) The cross correlation vector between the observed signal and the target signal of interest is:

$$\underline{R}_{XS} \triangleq E[\underline{X}_iS_i] \tag{2.08}$$

(3) The mean square value of the target signal is:

$$d \triangleq E[S_i^2] \tag{2.09}$$

Substitution of (2.07) through (2.09) into (2.06) gives:

$$J = \underline{H}^T \underset{\sim}{R}_{XX}\underline{H} - 2\underline{H}^T \underline{R}_{XS} + d \qquad (2.10)$$

Equation (2.10) is the performance function of the mMSE criteria. It is a quadratic function in terms of the filter vector $\underline{H}$.

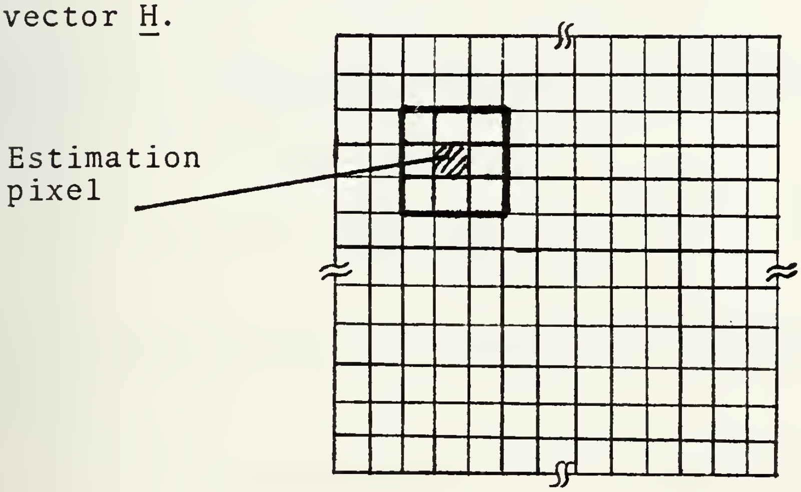


Figure 2.0 Search box.

Theorem 2.01

The performance function (2.10) is a unimodal (i.e., has a single minimum) function if the autocorrelation matrix $\underset{\sim}{R}_{XX}$ is positive definite.

Proof

The stationary points of the function (2.10) are found by setting the gradient of (2.10) with respect to $\underline{H}$ to zero.

$$\nabla_{\underline{H}} J = 2(\underset{\sim}{R}_{XX}\underline{H}^* - \underline{R}_{XS}) = 0 \qquad (2.11)$$

Since $\underset{\sim}{R}_{XX}$ is a symmetric positive definite matrix, its inverse exists. Therefore

$$\underline{H}^* = \underset{\sim}{R}_{XX}^{-1} \cdot \underline{R}_{XS} \qquad (2.12)$$

Equation (2.12) is the optimum filter vector which minimizes the cost function (2.10). In order to prove that the cost function is minimized for $\underline{H}^*$, the second gradient of (2.10) with respect to $\underline{H}$ is taken.

$$\nabla_{\underline{H}}(\nabla_{\underline{H}} J) = \underset{\sim}{R}_{XX} \tag{2.13}$$

Since $\underset{\sim}{R}_{XX}$ is positive definite, the cost function is minimized. The minimum value is

$$J_{min} = d - \underline{R}_{XS}^T \, \underset{\sim}{R}_{XX}^{-1} \cdot \underline{R}_{XS} \tag{2.14}$$

It is obtained by substituting the optimum filter vector $\underline{H}^*$ to (2.10).

The second derivative of the cost function I, as described in (2.13), is called the Hessian matrix.

If the autocorrelation matrix is singular, the cost function (2.10) is no longer unimodal because (2.11) can be set to zero for an infinite number of filter vectors $\underline{H}$.

It can be shown [49] that for such a case, a minimal solution can be obtained [50, 51] by using the pseudo inverse of $\underset{\sim}{R}_{XX}$.

$$\underline{H}^* = (\underset{\sim}{R}_{XX} \, \underset{\sim}{R}_{XX}^T)^{-1} \cdot \underset{\sim}{R}_{XX} \cdot \underline{R}_{XS}$$

The solution is not unique.

2. Performance Function II - MSNR

The observed signal in the "search-box" is represented by the vector $\underline{X}$. Let us assume that the target signal vector $\underline{S}$ and the clutter noise vector $\underline{N}$ are additive:

$$\underline{X} = \underline{S} + \underline{N} \tag{2.15}$$

Applying the linear filter $\underline{H}$ to the input signal vector $\underline{X}$, we obtain:

$$\begin{aligned} \underline{H}^T\underline{X} &= \underline{H}^T(\underline{S} + \underline{N}) \\ &= \underline{H}^T\underline{S} + \underline{H}^T\underline{N} \end{aligned} \tag{2.16}$$

Let us define the following terms:

$$S_o \triangleq \underline{H}^T\underline{S} = \text{target signal after filtering} \tag{2.17}$$

$$N_o \triangleq \underline{H}^T\underline{N} = \text{clutter noise after filtering} \tag{2.18}$$

The output signal to clutter noise ratio is then defined as:

$$J \triangleq \frac{\text{The Power in the filter image } \underline{H}^T\underline{X} \text{ due to target signal}}{\text{The Power in the filter image } \underline{H}^T\underline{X} \text{ due to clutter noise}} \tag{2.19}$$

$$J = \frac{E[S_o^2]}{E[N_o^2]} \tag{2.20}$$

Where E[$\cdot$] denotes the expected value, substitution of (2.17) and (2.18) into (2.20) gives:

$$J = \frac{E[(\underline{H}^T\underline{S})^2]}{E[(\underline{H}^T\underline{N})^2]} = \frac{E[\underline{H}^T\underline{S}\underline{S}^T\underline{H}]}{E[\underline{H}^T\underline{N}\underline{N}^T\underline{H}]} \tag{2.21}$$

The filter vector $\underline{H}$ can be taken out of the expectation operation.

$$J = \frac{\underline{H}^T E[\underline{SS}^T]\underline{H}}{\underline{H}^T E[\underline{NN}^T]\underline{H}} \tag{2.22}$$

Let us define the signal autocorrelation matrix as:

$$\underset{\sim}{R}_{SS} \triangleq E[\underline{SS}^T] \tag{2.23}$$

and the clutter noise autocorrelation matrix as:

$$\underset{\sim}{R}_{NN} \triangleq E[\underline{NN}^T] \tag{2.24}$$

$\underset{\sim}{R}_{NN}$ and $\underset{\sim}{R}_{SS}$ are symmetric and positive definite. Substitution of (2.23) and (2.24) in (2.22) yields:

$$J = \frac{\underline{H}^T \underset{\sim}{R}_{SS}\underline{H}}{\underline{H}^T \underset{\sim}{R}_{NN}\underline{H}} \tag{2.25}$$

The performance function J in (2.25) is the performance function of the MSNR criteria.

The filter vector $\underline{H}$ is obtained by maximizing J in (2.25) with respect to the filter vector $\underline{H}$.

Theorem 2.02

The maximum of the objective function (2.25) is equal to the largest eigenvalue of the matrix $\underset{\sim}{R}_{NN}^{-1} \cdot \underset{\sim}{R}_{SS}$, and the optimum filter $\underline{H}^*$ is the corresponding eigenvector.

Proof

The proof is based on the Cauchy-Schwarz inequality by finding the upper bound of J.

Since the autocorrelation matrix $\underset{\sim}{R}_{NN}$ is symmetric and positive definite, there exists a square nonsingular matrix $\underset{\sim}{V}$ which satisfies the relation [52].

$$\underset{\sim}{R}_{NN} = \underset{\sim}{V}^T \underset{\sim}{V} \tag{2.26}$$

Substitution of (2.26) into (2.25) and using the fact that

$$\underset{\sim}{V}^{-1} \cdot \underset{\sim}{V} = \underset{\sim}{V}^T \cdot \underset{\sim}{V}^{T^{-1}} = \underset{\sim}{I} \tag{2.27}$$

gives

$$J = \frac{\underline{H}^T (\underset{\sim}{V}^T \underset{\sim}{V}^{T^{-1}}) \underset{\sim}{R}_{SS} (\underset{\sim}{V}^{-1} \underset{\sim}{V}) \underline{H}}{\underline{H}^T (\underset{\sim}{V}^T \underset{\sim}{V}) \underline{H}} \tag{2.28}$$

Let us define the normalized vector $\underline{W}$ as:

$$\underline{W} \overset{\Delta}{=} \frac{\underset{\sim}{V}\underline{H}}{\sqrt{(\underset{\sim}{V}\underline{H})^T \cdot (\underset{\sim}{V}\underline{H})}} = \frac{\underset{\sim}{V}\underline{H}}{\| \underset{\sim}{V}\underline{H} \|} \tag{2.29}$$

which also satisfies the normalization condition,

$$\underline{W}^T \underline{W} = 1 \tag{2.30}$$

Substitution of (2.29) and (2.30) into (2.28) gives

$$J = \underline{W}^T \underset{\sim}{V}^{T^{-1}} \underset{\sim}{R}_{SS} \underset{\sim}{V}^{-1} \underline{W} \tag{2.31}$$

Let us define the matrix $\underset{\sim}{P}$ as:

$$\underset{\sim}{P} \overset{\Delta}{=} \underset{\sim}{V}^{T^{-1}} \underset{\sim}{R}_{SS} \underset{\sim}{V}^{-1} \tag{2.32}$$

Equation (2.31) becomes

$$J = \underline{W}^T \underset{\sim}{P} \underline{W} \tag{2.33}$$

Using the Schwarz inequality, we obtain

$$(\underline{W}^T \underline{\underset{\sim}{P}\underline{W}})^2 \le (\underline{W}^T \underline{W}) \cdot (\underline{W}^T \underset{\sim}{P}^T \underset{\sim}{P} \underline{W}) \tag{2.34}$$

Since the left side of the inequality is equal to J^2, the right side of (2.34) is the upper bound of J^2. The performance

function J reaches its maximum when the equality holds, which occurs when:

$$\underline{W} = \alpha \cdot \underset{\sim}{P}\,\underline{W} \tag{2.35}$$

where α is a constant. Substituting (2.29) and (2.32) into (2.35) obtains (2.36).

$$\frac{\underset{\sim}{V}\,\underline{H}}{\sqrt{(\underset{\sim}{V}\underline{H})^T\ (\underset{\sim}{V}\underline{H})}} = \alpha \cdot \underset{\sim}{V}^{T^{-1}}\,\underset{\sim}{R}_{SS}\underset{\sim}{V}^{-1} \cdot \frac{\underset{\sim}{V}\,\underline{H}}{\sqrt{(\underset{\sim}{V}\underline{H})^T\ (\underset{\sim}{V}\underline{H})}} \tag{2.36}$$

Multiplying (2.36) by

$$\frac{1}{\sqrt{(\underset{\sim}{V}\underline{H})^T\ (\underset{\sim}{V}\underline{H})}} \cdot \underset{\sim}{V}^T\ , \tag{2.37}$$

we obtain:

$$\underset{\sim}{V}^T\underset{\sim}{V} \cdot \underline{H} = \alpha \cdot \underset{\sim}{V}^T\underset{\sim}{V}^{T^{-1}} \cdot \underset{\sim}{R}\,\underset{\sim}{V}^{-1}\,\underset{\sim}{V}\,\underline{H} \tag{2.38}$$

Substituting (2.26) and (2.27) in (2.38), we get:

$$\underset{\sim}{R}_{NN}\,\underline{H} = \alpha \cdot \underset{\sim}{R}_{SS} \cdot \underline{H} \tag{2.39}$$

Since $\underset{\sim}{R}_{NN}$ is a positive definite matrix, its inverse $\underset{\sim}{R}_{NN}^{-1}$ exists. Multiplying (2.39) by $\frac{1}{\alpha} \cdot \underset{\sim}{R}_{NN}^{-1}$, we obtain:

$$(\underset{\sim}{R}_{NN}^{-1} \cdot \underset{\sim}{R}_{SS} - \frac{1}{\alpha} \cdot \underset{\sim}{I}) \cdot \underline{H}^* = \underline{0} \tag{2.40}$$

where $\underset{\sim}{I}$ is the identity matrix.

Equation (2.40) is called the generalized eigenvalue eigenvector problem [52].

Substituting the $\underline{H}^*$ of (2.40) into (2.25), we obtain the maximum value of J. One can see that

$$J_{max} = \left(\frac{1}{\alpha} \right)_{max} \qquad (2.41)$$

In other words, J_{max} is the largest eigenvalue of the matrix $\underset{\sim}{R}_{NN}^{-1} \underset{\sim}{R}_{SS}$, and $\underline{H}^*$ is the corresponding eigenvector. The noise correlation matrix $\underset{\sim}{R}_{NN}$ can be obtained by assuming some target signal of interest $\underline{S}$ and using the observed signal $\underline{X}$ in the following way (the signal and noise are assumed additive).

$$\underset{\sim}{R}_{NN} = E[(\underline{X} - \underline{S})(\underline{X} - \underline{S})^T]$$

(Q.E.D.)

Theorem 2.03

The performance function J in (2.25) is in general a multi-modal function.

Proof

Based on theorem (2.02), the stationary points of the performance function J satisfy the eigenvector equation (2.40):

$$(\underset{\sim}{R}_{NN}^{-1} \underset{\sim}{R}_{SS} - \frac{1}{\alpha} \cdot I)\,\underline{H}^* = 0$$

In general, this equation has n different solutions, because the matrix $\underset{\sim}{R}_{NN}^{-1} \underset{\sim}{R}_{SS}$ in general can have n distinct eigenvalues, and thus n corresponding eigenvectors. So, in general, the performance function can have one absolute maximum and n-1 local smaller maxima.

Theorem 2.04

The performance function J is a unimodal (has a single maximum) if the matrices $\underset{\sim}{R}_{SS}$ and $\underset{\sim}{R}_{NN}$ are defined as in (2.23) and (2.24).

Proof

The proof is based on the fact that $\underset{\sim}{R}_{SS}$ is a dyad. Use equation (2.40):

$$(\underset{\sim}{R}_{NN}^{-1} \ \underset{\sim}{R}_{SS} - \frac{1}{\alpha} \cdot \underset{\sim}{I}) \cdot \underline{H} = 0$$

The matrix $\underset{\sim}{R}_{SS}$, being a dyad, can be written as:

$$\underset{\sim}{R}_{SS} = \underline{r} \cdot \underline{r}^T \qquad (2.42)$$

where $\underline{r}$ is a vector.

As mentioned before for the nontrivial solution of (2.40), the performance function

$$J = \frac{1}{\alpha} \qquad (2.43)$$

Using (2.42) and (2.43) in (2.40), we obtain

$$\underset{\sim}{R}_{NN}^{-1} \cdot \underline{r}\,\underline{r}^T \cdot \underline{H}^* = J\underline{H}^* \qquad (2.44)$$

Separating (2.44) into a product of a vector and a constant, we obtain:

$$(\underset{\sim}{R}_{NN}^{-1}\underline{r})(\underline{r}^T\,\underline{H}^*) = J \cdot \underline{H}^* \qquad (2.45)$$

For generality, a constant β can be used in the left side of (2.45) to give:

$$(\beta \cdot \underset{\sim}{R}_{NN}^{-1} \cdot \underline{r})(\frac{1}{\beta} \cdot \underline{r}^T\underline{H}^*) = J \cdot \underline{H}^* \qquad (2.46)$$

Comparing both sides of (2.46), we get:

$$J_{max} = \frac{1}{\beta} \cdot \underline{r}^T\underline{H}^* \qquad (2.47)$$

$$\underline{H}^* = \beta \cdot \underset{\sim}{R}_{NN}^{-1} \cdot \underline{r} \qquad (2.48)$$

Equation (2.47) shows that if $\underset{\sim}{R}_{SS}$ is a dyad, the performance function J has a unique stationary point

where it reaches its maximum. The general eigenvalue problem has a single non zero eigenvalue J_{max}, and a corresponding eigenvector: $\underline{H}^* = \beta \cdot \underset{\sim}{R}_{NN}^{-1} \cdot \underline{r}$.

(Q.E.D.)

C. DERIVATION OF SEARCHING TECHNIQUES FOR EXTREMUM: GRADIENT SEARCH METHODS FOR THE MINIMUM OF THE mMSE PERFORMANCE FUNCTIONS

1. Steepest Descent Method (SD) and the Best Step Adaptation Gain

The steepest descent method is a gradient method which uses the Jacobian gradient ($\underline{G} = \nabla_{\underline{H}} J$) of the performance function J to determine a suitable direction of search. Gradient methods which use the Jacobian to determine the direction search are called first order methods. Gradient methods for optimization are based on the Taylor expansion of the performance function J, as given below:

$$J(\underline{H} + \Delta\underline{H}) \simeq J(\underline{H}) + \underline{G}^T \cdot \Delta\underline{H} + \tfrac{1}{2}\Delta\underline{H}^T \underset{\sim}{A} \cdot \Delta\underline{H} \qquad (2.49)$$

where $\underline{G}$ is the Jacobian gradient of J and $\underset{\sim}{A}$ is the matrix of second order partial derivatives called the Hessian matrix. Equation (2.49) can be written in the form:

$$J(\underline{H} + \Delta\underline{H}) \simeq J(\underline{H}) + \Delta J \qquad (2.50)$$

The steepest descent uses only the Jacobian, so

$$\Delta J = \underline{G}^T \cdot \Delta\underline{H} \qquad (2.51)$$

In order to minimize the performance function J, we want to generate a descending sequence of J which finally converges

to the minimum of J, J*. In other words, we want a negative ΔJ, but:

$$\Delta J = \| \underline{G} \| \cdot \| \Delta \underline{H} \| \cdot \cos \Phi$$

where ϕ is the angle between the two vectors, $\underline{G}$ and $\Delta\underline{H}$. For maximum reduction of the cost function J, $\phi = \pi$ (2.52)

From (2.52), it is obvious that the change $\Delta\underline{H}$ in the filter vector $\underline{H}$ should be in the direction of the negative gradient - $\underline{G}$. This direction is called the steepest descent direction.

The steepest descent step $\Delta\underline{H}$ can be written in the form:

$$\Delta\underline{H} = - \alpha \cdot \underline{G} \tag{2.53}$$

where $-\underline{G}$ is called the step direction gradient and α the step size. In adaptive filter terminology, α is called the adaptation gain.

In order to generate an iterative method, one can represent the filter vector $\underline{H} + \Delta\underline{H}$ as $\underline{H}_{K+1}$ and $\underline{H}$ as $\underline{H}_k$. Thus,

$$H_{K+1} = H_K + \Delta\underline{H} \tag{2.54}$$

Substituting (2.53) in (2.54), we obtain:

$$\underline{H}_{K+1} = \underline{H}_K - \alpha_K \cdot \underline{G}_K \tag{2.55}$$

Equation (2.55) is called the steepest descent iterative method. For simplicity and without losing generality, the negative sign will be included in α_K. Thus (2.55) becomes:

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \, \underline{G}_K \tag{2.56}$$

If very small values of $\{\alpha_K\}$ are selected, the sequence $\{\underline{H}_K\}$ will converge very slowly. In order to increase the speed of convergence substantially, we chose the step sizes which provide the biggest descent each step. This concept is called the "best step". The adaptation gain α_K is picked to minimize $J(\underline{H}_{K+1})$. This choice of α_K constitutes a one dimensional minimization of the performance function $J(\underline{H}_{K+1})$.

Lemma 2.05

Let $J(\underline{H}_K)$ be the performance function to be minimized. Let the filter vector $\underline{H}_{K+1}$ be updated by the steepest descent method (2.56), then the "best step" towards the minimum of J is obtained in every iteration if the adaptation gain satisfies the relationship:

$$\underline{G}_{K+1}^T \cdot \underline{G}_K = 0 \tag{2.57}$$

where $\underline{G}_K$ is the Jacobian gradient of J with respect to the filter vector $\underline{H}$.

Proof

The performance function $J(\underline{H}_{K+1})$ can be expressed as:

$$J(\underline{H}_{K+1}) = J(\underline{H}_K + \alpha_K \, \underline{G}_K) \tag{2.58}$$

The task is to find α_K which minimizes $J(\underline{H}_{K+1})$ by setting the derivative of $J(\underline{H}_{K+1})$ with respect to α_K to zero.

$$\frac{dJ(\underline{H}_{K+1})}{d\alpha_K} = 0 \tag{2.59}$$

but H_{K+1} is a function of α_K as shown in (2.56). Thus (2.59) becomes:

$$\frac{d}{d\alpha_K}[J(\underline{H}_{K+1})] = (\nabla_{\underline{H}_{K+1}} J)^T \cdot \frac{d}{d\alpha_K}(\underline{H}_{K+1}) = 0 \tag{2.60}$$

Since $\underline{G}_{K+1} = \nabla_{\underline{H}_{K+1}} J$ and $\frac{d}{d\alpha_K}(\underline{H}_{K+1}) = \underline{G}_K$, (2.60) becomes:

$$\underline{G}_{K+1}^T \; \underline{G}_K = 0 \tag{2.61}$$

From (2.61), the best step concept requires orthogonality between the two gradient vectors, $\underline{G}_{K+1}$ and $\underline{G}_K$.

(Q.E.D.)

Up to this point, the cost function J was not specified, and the derivation of the steepest descent was made for any continuous differentiable function.

The mMSE performance function as given by [2.10] can be written as:

$$J_K = \underline{H}_K^T \underset{\sim}{R}_{XX} \underline{H}_K - 2\underline{H}_K^T \underline{R}_{XS} + d \tag{2.62}$$

The gradient $\underline{G}_K$ of J_K with respect to $\underline{H}_K$ is given by:

$$\underline{G}_K \triangleq \nabla_{\underline{H}_K} J_K = 2(\underset{\sim}{R}_{XX} \underline{H}_K - \underline{R}_{XS}) \tag{2.63}$$

From (2.63) and (2.56), $\underline{G}_{K+1}$ can be expressed as:

$$\begin{aligned}\underline{G}_{K+1} &= 2(\underset{\sim}{R}_{XX}\ \underline{H}_{K+1} - \underline{R}_{XS}) \qquad (2.64)\\ &= 2(\underset{\sim}{R}_{XX}(\underline{H}_K + \alpha_K\ \underline{G}_K) - \underline{R}_{XS})\\ &= 2(\underset{\sim}{R}_{XX}\ \underline{H}_K - \underline{R}_{XS}) + 2 \cdot \underset{\sim}{R}_{XX}\ \alpha_K \cdot \underline{G}_K\\ &= \underline{G}_K + 2\alpha_K \cdot \underset{\sim}{R}_{XX} \cdot \underline{G}_K\end{aligned}$$

Lemma (2.05) is used in (2.64) to compute the best step α_K.

Since $\underline{G}_{K+1}^T \cdot \underline{G}_K = 0$, Lemma (2.05)

$$(\underline{G}_K + 2\alpha_K \cdot \underset{\sim}{R}_{XX}\ \underline{G}_K)^T \cdot \underline{G}_K = 0, \quad \text{see (2.64)}$$

$$\underline{G}_K^T\ \underline{G}_K + 2\alpha_K \cdot \underline{G}_K^T\ \underset{\sim}{R}_{XX}^T\ \underline{G}_K = 0 \qquad (2.65)$$

$\underset{\sim}{R}_{XX}$ is a symmetric matrix, thus $\underset{\sim}{R}_{XX}^T = \underset{\sim}{R}_{XX}$. Using this fact in (2.65), we obtain

$$\alpha_K = -\frac{1}{2} \cdot \frac{\underline{G}_K^T\ \underline{G}_K}{\underline{G}_K^T\ \underset{\sim}{R}_{XX}\ \underline{G}_K} \qquad (2.66)$$

Equation (2.66) is the equation of the best step for the steepest descent method.

Combining the results from (2.56), (2.63) and (2.66), we obtain the steepest descent adaptive filter:

[Step 1] Set a starting filter vector $\underline{H}_0$, stopping bound (i.e., max. acceptable adaptation error) ε, the correlation matrix $\underset{\sim}{R}_{XX}$ of the observed signal, and the cross correlation vector $\underline{R}_{XS}$.

[Step 2] Compute the gradient:

$$G_K = 2(\underset{\sim}{R}_{XX}\ \underline{H}_K - \underline{R}_{XS})$$

[Step 3] Compute the adaptation gain:

$$\alpha_K = -\frac{1}{2} \cdot \frac{\underline{G}_K^T\ \underline{G}_K}{\underline{G}_K^T\ \underset{\sim}{R}_{XX}\ \underline{G}_K}$$

[Step 4] Update the filter vector:

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \cdot \underline{G}_K$$

[Step 5] Test for stopping condition:

If $\underline{G}_K^T\ \underline{G}_K \leq \varepsilon$, then terminate. Otherwise, go to step 2.

The stopping criteria is chosen as $\underline{G}_K^T\ \underline{G}_K \leq \varepsilon$ because the performance function is unimodal (has a single stationary point), and we are looking for the stationary point which in fact satisfies the vanishing of the gradient.

2. Accelerated Steepest Descent Method (ASD)

The accelerated steepest descent method was first introduced in 1964 by Shah, Buehler and Kempthose [53] Its purpose was to accelerate the convergence of the standard steepest descent method. Its concept was incorporated in an algorithm which converges to the minimum of any n dimensional quadratic function in no more than 2•n-1 steps. Practically, this algorithm is not very efficient because of its sensitivity to error propagation. For large n, the error propagation affected the convergence rate and the method sometimes

converges as slowly or even more slowly than the steepest descent method.

The adaptation gain of the ASD method is computed using Lemma 2.05 and the fact that the adaptive filter $\underline{H}_{K+1}$ is updated by the iterative equation:

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \cdot \underline{V}_K \tag{2.67}$$

From Lemma (2.05) and (2.67),

$$\underline{G}_{K+1}^T \cdot \underline{V}_K = 0 \tag{2.68}$$

but $\quad \underline{G}_{K+1} = 2(\underset{\sim}{R}_{XX}\, \underline{H}_{K+1} - \underline{R}_{XS}) \qquad$ [see (2.63)]

$$= 2(\underset{\sim}{R}_{XX}(\underline{H}_K + \alpha_K\, \underline{V}_K) - \underline{R}_{XS})$$

$$= 2(\underset{\sim}{R}_{XX}\, \underline{H}_K - \underline{R}_{XS}) + 2\alpha_K\, \underset{\sim}{R}_{XX}\underline{V}_K$$

$$= \underline{G}_K + 2\alpha_K\, \underset{\sim}{R}_{XX}\, \underline{V}_K \tag{2.69}$$

Using (2.69) and (2.68), we obtain:

$$(\underline{G}_K + 2\alpha_K\, \underset{\sim}{R}_{XX}\, \underline{V}_K)^T \cdot \underline{V}_K \tag{2.70}$$

$$\alpha_K = -\frac{1}{2} \cdot \frac{\underline{G}_K^T\, \underline{V}_K}{\underline{V}_K^T\, \underset{\sim}{R}_{XX}\, \underline{V}_K} \tag{2.71}$$

The accelerated steepest descent adaptive filter is carried out by the following steps:

[Step 1] Set a starting filter vector, $\underline{H}_0 = \underline{H}_1$, stopping bound ε, the correlation matrix $\underset{\sim}{R}_{XX}$ and the cross correlation vector $\underline{R}_{XS}$, and the gradient $\underline{G}_0$.

[Step 2] Compute the gradient $\underline{G}_K$ of J.

$$\underline{G}_K = 2(\underset{\sim}{R}_{XX}\,\underline{H}_K - \underline{R}_{XS})$$

[Step 3] Compute the step direction vector $\underline{V}_K$.

$$\underline{V}_K = \begin{cases} -\underline{G}_K & \text{for } K = 2, 4, 6 \ldots. \\ \underline{H}_K - \underline{H}_{K-2} & \text{for } K = 3, 5, 7 \ldots. \end{cases}$$

[Step 4] Compute the adaptation gain α_K.

$$\alpha_K = -\frac{1}{2} \cdot \frac{\underline{G}_K^T\,\underline{V}_K}{\underline{V}_K^T\,\underset{\sim}{R}_{XX}\,\underline{V}_K}$$

[Step 5] Update the filter vector $\underline{H}_K$.

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K\,\underline{V}_K$$

[Step 6] Test for stopping condition.

If $\underline{G}_K^T \cdot \underline{G}_K \leq \varepsilon$, terminate. Otherwise go to step 2.

3. Amir's Method (AMM)

This method was suggested by this author at the beginning of the research. The purpose was to derive a method which will converge faster than the steepest descent method. Experiments showed that the AMM method converges approximately

three times faster than the SD method as shown in Fig. 2.6a. This method is a non-conjugate gradient method and is not as fast as the conjugate gradient methods. But it can replace the SD method as a robust and faster method.

The AMM gradient search method was designed based on the fact that the gradient of a unimodal performance function vanishes only once, at the stationary point of the performance function, which is the extremum point we are looking for.

The adaptation procedure is derived in the following.

The functional Ψ_K is defined as:

$$\Psi_K = \underline{G}_K^T \underline{G}_K \tag{2.71-1}$$

where $\underline{G}_K$ is the gradient of the performance function J, as in (2.63).

The adaptive filter $\underline{H}_{K+1}$ is updated as given in (2.56) for the SD method. The adaptation gain α_K is computed according to the "best step" concept, to minimize Ψ_{K+1}. Using (2.64), we obtain:

$$\begin{aligned} \Psi_{K+1} &= \underline{G}_{K+1}^T \underline{G}_{K+1} \\ &= (\underline{G}_K + \alpha_K \underset{\sim}{R}_{XX} \underline{G}_K)^T \cdot (\underline{G}_K + \alpha_K \underset{\sim}{R}_{XX} \underline{G}_K) \\ &= \underline{G}_K^T (\underset{\sim}{I} + \alpha_K \underset{\sim}{R}_{XX}) (\underset{\sim}{I} + \alpha_K \underset{\sim}{R}_{XX}) \underline{G}_K \end{aligned} \tag{2.71-2}$$

$$\Psi_{K+1} = \underline{G}_K^T (\underset{\sim}{I} + 2\alpha_K \underset{\sim}{R}_{XX} + \alpha_K^2 \underset{\sim}{R}_{XX}^2) \underline{G}_K \tag{2.71-3}$$

In order to get the best step, we take the derivative of Ψ_{K+1} with respect to the adaptation gain α_K and set it to zero:

$$\frac{d}{d\alpha_K}\Psi_{K+1} = \underline{G}_K^T\,(\cdot\, 2\underset{\sim}{R}_{XX} + 2\alpha_K \underset{\sim}{R}_{XX}^2)\,\underline{G}_K = 0 \qquad (2.71\text{-}4)$$

$$= 2\underline{G}_K^T\,\underset{\sim}{R}_{XX}\,\underline{G}_K + 2\alpha_K\,\underline{G}_K^T\,\underset{\sim}{R}_{XX}^2\,\underline{G}_K = 0$$

Solve (2.71-4) and get:

$$\alpha_K = -\frac{\underline{G}_K^T\,\underset{\sim}{R}_{XX}\,\underline{G}_K}{\underline{G}_K^T\,\underset{\sim}{R}_{XX}^2\,\underline{G}_K} \qquad (2.71\text{-}5)$$

The AMM adaptive filter is implemented by the following steps:

[Step 1] Set initial filter vector $\underline{H}_o$, the stopping bound ε, the correlation matrix $\underset{\sim}{R}_{XX}$ and the cross-correlation vector $\underline{R}_{XS}$.

[Step 2] Compute the gradient $\underline{G}_K$ of the performance function J.

$$\underline{G}_K = 2(\underset{\sim}{R}_{XX}\,\underline{H}_K - \underline{R}_{XS})$$

[Step 3] Compute the adaptation gain α_K:

$$\alpha_K = -\frac{\underline{G}_K^T\,\underset{\sim}{R}_{XX}\,\underline{G}_K}{\underline{G}_K^T\,\underset{\sim}{R}_{XX}^2\,\underline{G}_K}$$

[Step 4] Update the filter vector $\underline{H}_K$

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K\,\underline{G}_K$$

[Step 5] Test for stopping condition.

If $\Psi_K \leq \varepsilon$, then terminate, otherwise go to step 2.

4. Fletcher-Reeve Conjugate Gradient Method (CGF)

The Fletcher-Reeves conjugate gradient (CGF) was first introduced in 1964 by Fletcher and Reeves [69]. The method is similar to the pioneering work of Hestenes and Stiefel [54]. The CGF method uses conjugate vectors as step direction.

Definition

The vectors $\underline{V}_i$, $\underline{V}_j$ are said to be "conjugate" with respect to the matrix $\underset{\sim}{R}_{XX}$ if they satisfy the following condition:

$$\underline{V}_i^T \underset{\sim}{R}_{XX} \underline{V}_j = 0 \quad \text{for } i \neq j \text{ and } \underline{V}_i, \underline{V}_j \neq 0.$$

The importance of this method is its fast convergence rate for quadratic functions like (Eq. 2.10). This method is proved to converge in n steps apart from rounding errors where n is the dimension of the filter vector.

The adaptation gain of the CGF method is computed using Lemma 2.05 and the fact that the adaptive filter $\underline{H}_{K+1}$ is updated by the iterative equation in (2.67). Following the equations (2.67) up to (2.71) in a similar way, we obtain the adaptation gain as:

$$\alpha_K = -\frac{1}{2} \cdot \frac{\underline{G}_K^T \underline{V}_K}{\underline{V}_K^T \underset{\sim}{R}_{XX} \underline{V}_K} \qquad (2.71\text{-}6)$$

The step direction vector $\underline{V}_K$ is computed by the following iterative procedure [55].

$$\underline{V}_{K+1} = - \underline{G}_{K+1} + \beta_K \cdot \underline{V}_K \tag{2.73}$$

$$\beta_K = \frac{\underline{G}_{K+1}^T \cdot \underline{G}_{K+1}}{\underline{G}_K^T \; \underline{G}_K} \tag{2.74}$$

The method of CGF was once applied to the Rosenbrock function [54]. The performance result was poor. Subsequently, it was suggested to restart the method every n iterations, where n is the dimension of the vector $\underline{H}$. This thesis confirmed that the convergence of this method for our two performance functions (2.10) and (2.25) is faster if this method of restarts is used.

The CGF adaptive filter is carried out in the following steps:

[Step 1] Select a starting filter vector $\underline{H}_0$, the stopping bound ε, the auto-correlation matrix $\underset{\sim}{R}_{XX}$ and the cross-correlation vector $\underline{R}_{XS}$.

[Step 2] Compute the gradient $\underline{G}_K$ of the performance function J.

$$\underline{G}_K = \nabla J = 2(\underset{\sim}{R}_{XX} \; \underline{H}_K - \underset{\sim}{R}_{XS})$$

[Step 3] Compute the step direction vector $\underline{V}_K$.

$$\underline{V}_K = \begin{cases} - \underline{G}_K & \text{if } K \text{ Mod } n = 0 \\ - \underline{G}_K + \dfrac{\underline{G}_K^T \; \underline{G}_K}{\underline{G}_{K+1}^T \; \underline{G}_{K-1}} \cdot \underline{V}_{K-1} & \text{else,} \end{cases}$$

[Step 4] Compute the adaptation gain:

$$\alpha_K = -\frac{1}{2} \cdot \frac{\underline{G}_K^T \underline{V}_K}{\underline{V}_K^T \underset{\sim}{R}_{XX} \underline{V}_K}$$

[Step 5] Update the filter vector $\underline{H}_{K+1}$.

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \underline{V}_K$$

[Step 6] Test for stopping condition.

If $\underline{G}_K^T \underline{G}_K \leq \varepsilon$, terminate the adaptation. Otherwise go to step 2.

5. Pollack-Rebiere Conjugate Gradient Method (CGP)

The Pollack-Rebiere conjugate gradient CCGP method is similar to the CGF method. The difference is in the computation of the search direction when K Mod n ≠ 0. In [56], Powell gave a theoretical reason for favoring the Pollack-Rebiere algorithm. In this thesis, the author found the CGP method more efficient and converging faster than the CGF method. (See Section F).

The search direction of the CGP method is given by the following expression:

$$\underline{V}_K = -\underline{G}_K + \frac{\underline{G}_K^T (\underline{G}_K - \underline{G}_{K-1})}{\underline{G}_{K-1}^T \underline{G}_{K-1}} \cdot \underline{V}_{K-1} \qquad (2.75)$$

The CGP adaptive filter is carried out in the following steps:

[Step 1] Select a starting filter vector $\underline{H}_0$, the stopping bound ε, the auto-correlation matrix $\underset{\sim}{R}_{XX}$ and the cross-correlation vector $\underline{R}_{XX}$.

[Step 2] Compute the gradient $\underline{G}_K$ of the performance function J.

$$\underline{G}_K = \nabla J = 2(\underset{\sim}{R}_{XX}\ \underline{H}_K - \underline{R}_{XS})$$

[Step 3] Compute the step direction vector $\underline{V}_K$.

$$\underline{V}_K = \begin{cases} -\ \underline{G}_K & \text{if K Mod n} = 0 \\ -\ \underline{G}_K + \dfrac{\underline{G}_K^T\ (\underline{G}_K - \underline{G}_{K-1})}{\underline{G}_{K-1}^T \cdot \underline{G}_{K-1}} \cdot \underline{V}_{K-1} & \text{else} \end{cases}$$

[Step 4) Compute the adaptation gain.

$$\alpha_K = -\ \frac{1}{2}\ \frac{\underline{G}_K^T \cdot \underline{V}_K}{\underline{V}_K^T\ \underset{\sim}{R}_{XX}\ \underline{V}_K}$$

[Step 5] Update the filter vector $\underline{H}_{K+1}$.

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \cdot \underline{V}_K$$

[Step 6] Test for stopping condition.

If $\underline{G}_K^T\ \underline{G}_K \leq \varepsilon$, terminate the adaptation. Otherwise go to step 2.

6. Davidon-Fletcher-Powell Variable Metric Method (DFP)

One of the most efficient searching methods is the Davidon-Fletcher-Powell (DFP). It was developed by Fletcher and Powell [57] from the variable metric method due to Davidon [54,58]. The variable metric term was coined by Davidon to describe methods which at the K iteration utilize the increment of the form

$$\Delta \underline{H} = - \alpha_K \underset{\sim}{A}_K \underline{G}_K \tag{2.76}$$

and update the metric-correction transformation $\underset{\sim}{A}_K$ from iteration to iteration. The DFP method updates the metric $\underset{\sim}{A}_K$ by the iterative expression:

$$\underset{\sim}{A}_{K+1} = \underset{\sim}{A}_K + \alpha_K \frac{\underline{V}_K \underline{V}_K^T}{\underline{V}_K^T \underline{P}_K} - \frac{\underset{\sim}{A}_K \underline{P}_K \underline{P}_K^T \underset{\sim}{A}_K}{\underline{P}_K^T \underset{\sim}{A}_K \underline{P}_K} \tag{2.77}$$

where:

$$\underline{V}_K = - \underset{\sim}{A}_K \underline{G}_K \tag{2.78}$$

$$\underline{P}_K = \underline{G}_{K+1} - \underline{G}_K \tag{2.79}$$

Fletcher and Powell proved that for a general function J that a positive definite $\underset{\sim}{A}_K$ implies $\underset{\sim}{A}_{K+1}$ is also positive definite [58]. For the performance function J given in (2.10), it can be shown [59] that the set $\{\alpha_K \underset{\sim}{A}_K \cdot \underline{G}_K\}$ is a set of conjugate directions so the DFP exhibits quadratic termination in n steps.

The adaptation gain of the DFP adaptive filter based on the best step concept introduced in (Lemma 2.05).

Using the filter update of the DFP method:

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \underline{V}_K \tag{2.80}$$

The adaptation gain is found to be:

$$\alpha_K = - \frac{1}{2} \cdot \frac{\underline{V}_K^T \underline{G}_K}{\underline{V}_K^T \underset{\sim}{R}_{XX} \underline{V}_K}$$

The adaptive filter designed by the DFP method is carried out in the following steps:

[Step 1] Select a starting filter vector $\underline{H}_o$, the starting correction metric $\underset{\sim}{A}_o = \underset{\sim}{I}$ (where $\underset{\sim}{I}$ is the identity matrix, the gradient $\underline{G}_o$, the stopping bound ε, the autocorrelation matrix $\underset{\sim}{R}_{XX}$, and the crosscorrelation vector $\underline{R}_{XS}$.

[Step 2] Compute the step direction vector $\underline{V}_K$:

$$\underline{V}_K = - \underset{\sim}{A}_K \, \underline{G}_K$$

[Step 3] Compute the adaptation gain.

$$\alpha_K = - \frac{1}{2} \cdot \frac{\underline{V}_K^T \, \underline{G}_K}{\underline{V}_K^T \, \underset{\sim}{R}_{XX} \, \underline{V}_K}$$

[Step 4] Update the filter vector $\underline{H}_K$.

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \, \underline{V}_K$$

[Step 5] Compute the gradient $\underline{G}_{K+1}$ of the function J.

$$\underline{G}_{K+1} = 2(\underset{\sim}{R}_{XX} \, \underline{H}_{K+1} - \underline{R}_{XS})$$

[Step 6] Compute the vector $\underline{P}_K$.

$$\underline{P}_K = \underline{G}_{K+1} - \underline{G}_K$$

[Step 7] Update the variable metric $\underset{\sim}{A}_K$.

$$\underset{\sim}{A}_{K+1} = \underset{\sim}{A}_K + \frac{\underline{V}_K^T \, \underline{V}_K}{\underline{V}_K^T \, \underline{P}_K} - \frac{\underset{\sim}{A}_K \, \underline{P}_K \, \underline{P}_K^T \, \underset{\sim}{A}_K}{\underline{P}_K^T \, \underset{\sim}{A}_K \, \underline{P}_K}$$

[Step 8] Test for stopping criteria.

If $\underline{G}_{K+1} \cdot \underline{G}_{K+1} \leq \varepsilon$, terminate the adaptation, otherwise go to step 2.

D. DERIVATION OF SEARCHING TECHNIQUES FOR EXTREMUM, GRADIENT SEARCH METHODS FOR THE MAXIMUM OF THE MSNR PERFORMANCE FUNCTION

1. Approximation for Best Step Adaptation Gain

The maximization of signal to noise ratio performance function J, as defined in (2.25), is a non-linear performance function of the filter vector $\underline{H}$. The function J being non-quadratic introduces new difficulties. The methods which have been theoretically proved to converge in N steps for quadratic cases like the mMSE, no longer converge as fast. The adaptation gain can no longer be efficiently computed by the best step concept because of the large amount of computation required to obtain the best step. In order to make this gradient search method efficient, the adaptation gain is approximated by the "best step" concept to generate a nondecreasing sequence of performance functions $\{J_K\}$ which finally converges to the maximum of J.

Lemma 2.06

Let the performance function J be defined as in (2.25), and the adaptive filter be updated according to (2.67), then the best step adaptation gain at iteration step K satisfies the relation:

$$\alpha_K = - \frac{\underline{H}_K^T (\underset{\sim}{R}_{SS} - J_{K+1} \cdot \underset{\sim}{R}_{NN}) \cdot \underline{V}_K}{\underline{V}_K^T (\underset{\sim}{R}_{SS} - J_{K+1} \underset{\sim}{R}_{NN}) \underline{V}_K} \tag{2.81}$$

Proof

The proof is based on the Lemma 2.05. Using (2.67) and Lemma 2.05, we obtain:

$$\underline{G}_{K+1}^T \cdot \underline{V}_K = 0 \tag{2.82}$$

where $\underline{G}_{K+1}$ is the gradient of the performance function J at the K+1 iteration step, and $\underline{V}_K$ is the step direction (search direction) vector. But according to (2.25),

$$J_{K+1} = \frac{\underline{H}_{K+1}^T \underset{\sim}{R}_{SS} \underline{H}_{K+1}}{\underline{H}_{K+1} \underset{\sim}{R}_{NN} \underline{H}_{K+1}}$$

The gradient of J_{K+1} with respect to $\underline{H}_{K+1}$ is:

$$\underline{G}_{K+1} \triangleq \nabla_{\underline{H}_{K+1}} J = \frac{2}{\underline{H}_{K+1}^T \underset{\sim}{R}_{NN} \underline{H}_{K+1}} \cdot (\underset{\sim}{R}_{SS} - J_{K+1} \underset{\sim}{R}_{NN}) \cdot \underline{H}_{K+1} \tag{2.83}$$

Using (2.83) in (2.82), we obtain:

$$\frac{2}{\underline{H}_{K+1} \underset{\sim}{R}_{NN} \underline{H}_{K+1}} \cdot \underline{H}_{K+1}^T \cdot (\underset{\sim}{R}_{SS}^T - J_{K+1} \underset{\sim}{R}_{NN}^T) \cdot \underline{V}_K = 0 \tag{2.84}$$

But according to (2.67)

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \cdot \underline{V}_K$$

and $\underset{\sim}{R}_{SS}$, $\underset{\sim}{R}_{NN}$ being symmetric and positive definite gives:

$$(\underline{H}_K^T + \alpha_K \underline{V}_K^T)(\underset{\sim}{R}_{SS} - J_{K+1} \cdot \underset{\sim}{R}_{NN}) \cdot \underline{V}_K = 0 \qquad (2.85)$$

$$\therefore \underline{H}_K^T (\underset{\sim}{R}_{SS} - J_{K+1} \cdot \underset{\sim}{R}_{NN})\underline{V}_K + \alpha_K \cdot \underline{V}_K^T (\underset{\sim}{R}_{SS} - J_{K+1}\underset{\sim}{R}_{NN})\underline{V}_K = 0$$

So,

$$\alpha_K = - \frac{\underline{H}_K^T (\underset{\sim}{R}_{SS} - J_{K+1} \underset{\sim}{R}_{NN}) \underline{V}_K}{\underline{V}_K^T (\underset{\sim}{R}_{SS} - J_{K+1} \underset{\sim}{R}_{NN}) \underline{V}_K} \qquad (2.86)$$

Q.E.D.

The adaptation gain α_K in (2.86) cannot be obtained because it is a function of J_{K+1} which itself cannot be computed without α_K. Thus the "best step" concept introduces a nonlinear problem for the MSNR performance function. In order to overcome this problem of solving a nonlinear equation in each iteration, the adaptation gain will be approximated by using J_K instead of J_{K+1}. Since J_K is obtained one step prior to α_K, J_{K+1} does not need to be solved. Now we must prove that this choice of adaptation gain for the MSNR performance function will generate a nondecreasing sequence $\{J_K\}$ which eventually converges to the optimum J_{max}.

<u>Lemma 2.07</u>

Let the performance function J be defined as in (2.25) and the filter $\underline{H}_K$ be updated by (2.67), let the adaptation gain α_K be given by

$$\alpha_K = - \frac{\underline{H}_K^T (\underset{\sim}{R}_{SS} - J_K \underset{\sim}{R}_{NN}) \underline{V}_K}{\underline{V}_K^T (\underset{\sim}{R}_{SS} - J_K \underset{\sim}{R}_{NN}) \underline{V}_K} \qquad (2.87)$$

Then it will generate a nondecreasing sequence $\{J_K\}$ which converge to the maximum J.

Proof

Using (2.25), we obtain:

$$J_{K+1} = \frac{\underline{H}_{K+1}^{T} \; \underset{\sim}{R}_{SS} \; \underline{H}_{K+1}}{\underline{H}_{K+1}^{T} \; \underset{\sim}{R}_{NN} \; \underline{H}_{K+1}} \tag{2.88}$$

Substitution of (2.67) in (2.88) gives:

$$\begin{aligned} J_{K+1} &= \frac{(\underline{H}_K + \alpha_K \underline{V}_K)^T \underset{\sim}{R}_{SS} (\underline{H}_K + \alpha_K \underline{V}_K)}{(\underline{H}_K + \alpha_K \underline{V}_K)^T \underset{\sim}{R}_{NN} (\underline{H}_K + \alpha_K \underline{V}_K)} \\ &= J_K \frac{1 + 2\alpha_K \cdot \dfrac{\underline{V}_K^T \underset{\sim}{R}_{SS} \underline{H}_K}{\underline{H}_K^T \underset{\sim}{R}_{SS} \underline{H}_K} + \alpha_K^2 \; \dfrac{\underline{V}_K^T \underset{\sim}{R}_{SS} \underline{V}_K}{\underline{H}_K^T \underset{\sim}{R}_{SS} \underline{H}_K}}{1 + 2\alpha_K \cdot \dfrac{\underline{V}_K^T \underset{\sim}{R}_{NN} \underline{H}_K}{\underline{H}_K^T \underset{\sim}{R}_{NN} \underline{H}_K} + \alpha_K^2 \; \dfrac{\underline{V}_K^T \underset{\sim}{R}_{NN} \underline{V}_K}{\underline{H}_K^T \underset{\sim}{R}_{NN} \underline{H}_K}} \end{aligned} \tag{2.89}$$

Equation (2.89) is simplified due to the fact that $\underset{\sim}{R}_{SS}$, $\underset{\sim}{R}_{NN}$ are symmetric and positive definite.

In order to obtain a non-decreasing sequence $\{J_K\}$, we must satisfy: $J_{K+1} \geq J_K$, but the sequence $\{J_K\}$ is positive, so:

$$\frac{J_{K+1}}{J_K} \geq 1 \tag{2.90}$$

In order for (2.80) to satisfy (2.90), it can be seen that:

$$1 + 2\alpha_K \cdot \frac{\underline{V}_K^T \underset{\sim}{R}_{SS}\, \underline{H}_K}{\underline{H}_K^T \underset{\sim}{R}_{SS}\, \underline{H}_K} + \alpha_K^2 \frac{\underline{V}_K^T \underset{\sim}{R}_{SS}\, \underline{V}_K}{\underline{H}_K^T \underset{\sim}{R}_{SS}\, \underline{H}_K} \geq 1 + 2\alpha_K \frac{\underline{V}_K \underset{\sim}{R}_{NN}\, \underline{H}_K}{\underline{H}_K^T \underset{\sim}{R}_{NN}\, \underline{H}_K}$$

$$+ \alpha_K^2 \frac{\underline{V}_K^T \underset{\sim}{R}_{NN}\, \underline{H}_K}{\underline{H}_K^T \underset{\sim}{R}_{NN}\, \underline{H}_K} \qquad (2.91)$$

Using (2.91), (2.25) and the fact that $\underset{\sim}{R}_{NN}$, $\underset{\sim}{R}_{SS}$ are positive definite matrices, we obtain:

$$2 \cdot \underline{V}_K^T \underset{\sim}{R}_{SS}\, \underline{H}_K + \alpha_K \cdot \underline{V}_K^T \underset{\sim}{R}_{SS}\, \underline{V}_K \geq J_K(2\, \underline{V}_K^T \underset{\sim}{R}_{NN}\, \underline{H}_K + \alpha_K \cdot \underline{V}_K^T \underset{\sim}{R}_{NN}\, \underline{V}_K) \qquad (2.92)$$

$$\alpha_K \cdot \underline{V}_K^T (\underset{\sim}{R}_{SS} - J_K \underset{\sim}{R}_{NN}) \underline{V}_K \geq - 2 \cdot \underline{V}_K^T (\underset{\sim}{R}_{SS} - J_K \underset{\sim}{R}_{NN}) \underline{H}_K$$

$$\alpha_K \geq -2 \cdot \frac{\underline{V}_K^T (\underset{\sim}{R}_{SS} - J_K \underset{\sim}{R}_{NN}) \underline{H}_K}{V_K^T (\underset{\sim}{R}_{SS} - J_K \underset{\sim}{R}_{NN}) V_K} \qquad (2.93)$$

So the adaptation gain given in (2.87) generates a non-decreasing sequence $\{J_K\}$ because it satisfies (2.93).

Q.E.D.

Lemma 2.08

Let the performance function J be defined as in (2.25) and the filter vector $\underline{H}_K$ being updated by (2.67), then for each iteration step K, the gradient $\underline{G}_K$ of J is orthogonal to the filter vector $\underline{H}_K$ regardless of the adaptation gain.

Proof

The performance function given by (2.25) is

$$J_K = \frac{\underline{H}_K^T \underset{\sim}{R}_{SS} \underline{H}_K}{\underline{H}_K^T \underset{\sim}{R}_{NN} \underline{H}_K}$$

The gradient of J_K with respect to the filter vector $\underline{H}_K$ is given by

$$\underline{G}_K \triangleq \nabla_{H_K} J = \frac{2}{\underline{H}_K^T \underset{\sim}{R}_{NN} \underline{H}_K} \cdot (\underset{\sim}{R}_{SS} - J_K \cdot \underset{\sim}{R}_{NN}) \cdot \underline{H}_K \tag{2.83}$$

From (2.83), it follows that:

$$\underline{H}_K^T \cdot \underline{G}_K = \frac{2}{\underline{H}_K^T \underset{\sim}{R}_{NN} \underline{H}_K} \cdot \underline{H}_K^T (\underset{\sim}{R}_{SS} - J_K \underset{\sim}{R}_{NN}) \cdot \underline{H}_K \tag{2.94}$$

$$\underline{H}_K^T \underline{G}_K = 2 \cdot \left(\frac{\underline{H}_K^T \underset{\sim}{R}_{SS} \underline{H}_K}{\underline{H}_K^T \underset{\sim}{R}_{NN} \underline{H}_K} - J_K \frac{\underline{H}_K^T \underset{\sim}{R}_{NN} \underline{H}_K}{\underline{H}_K^T \underset{\sim}{R}_{NN} \underline{H}_K} \right) \tag{2.95}$$

Using (2.25) in (2.95), we obtain:

$$\underline{H}_K^T \underline{G}_K = 2 \cdot (J_K - J_K) = 0$$

Therefore, $\underline{H}_K^T \underline{G}_K = 0$ (2.96)

Thus the filter vector H_K at iteration step K is orthogonal to the gradient $\underline{G}_K$ of the performance function J.

Q.E.D.

2. Steepest Descent Method (SD)

The steepest descent (SD) method as described for the quadratic mMSE perform function can be used here for the

MSNR performance function with some exceptions:

The concept of the "best step" is used by an approximation of the best adaptation gain. The gradient of the performance function with respect to the filter vector is a function of the performance function. Thus, successive values of performance function must be computed.

The adaptation equation used here is identical to (2.56).

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \cdot \underline{G}_K$$

The adaptation gain is obtained from (2.87) by replacing the step direction vector $\underline{V}_K$ with the gradient $\underline{G}_K$ (the direction of the SD).

The adaptation gain obtained is:

$$\alpha_K = - \frac{\underline{H}_K^T (\underset{\sim}{R}_{SS} - J_K \underset{\sim}{R}_{NN}) \underline{G}_K}{\underline{G}_K^T (\underset{\sim}{R}_{SS} - J_K \underset{\sim}{R}_{NN}) \underline{G}_K} \tag{2.97}$$

The matrix $\underset{\sim}{Q}_K$ is defined as:

$$\underset{\sim}{Q}_K \triangleq \underset{\sim}{R}_{SS} - J_K \underset{\sim}{R}_{NN} \tag{2.98}$$

Substituting (2.98) into (2.99), we obtain:

$$\alpha_K = - \frac{\underline{H}_K^T \underset{\sim}{Q}_K \cdot \underline{G}_K}{\underline{G}_K^T \underset{\sim}{Q}_K \cdot \underline{G}_K} \tag{2.99}$$

The adaptive filter designed by the SD method is carried out in the following steps:

[Step 1] Select a starting filter vector $\underline{H}_o$, and a stopping bound δ.

[Step 2] Compute the performance function J_K as in (2.25)

$$J_K = \frac{\underline{H}_K^T \underset{\sim}{R}_{SS} \underline{H}_K}{\underline{H}_K^T \underset{\sim}{R}_{NN} \underline{H}_K}$$

[Step 3] Compute the gradient $\underline{G}_K \triangleq \nabla_{\underline{H}_K} J_K$:

$$G_K = \frac{2}{\underline{H}_K^T \underset{\sim}{R}_{NN} \underline{H}_K} \cdot \underset{\sim}{Q}_K \cdot \underline{H}_K$$

where $\underset{\sim}{Q}_K$ is given by (2.98).

[Step 4] Compute the adaptation gain:

$$\alpha_K = - \frac{\underline{H}_K^T \underset{\sim}{Q}_K \underline{G}_K}{\underline{G}_K^T \underset{\sim}{Q}_K \underline{G}_K}$$

[Step 5] Update the filter vector $\underline{H}_K$

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \underline{G}_K$$

[Step 6] Test for stopping condition.

If $\frac{||\underline{H}_{K+1} - \underline{H}_K||}{||\underline{H}_K||} \leq \delta$, then terminate the adaptation. Otherwise, go to step 2.

The stopping condition is different from the one used for the mMSE criteria because in this case the gradient $\underline{G}_K$ is a nonlinear function of the filter $\underline{H}_K$ and when $\underline{H}_K \rightarrow \underline{\infty}$ the gradient $\underline{G}_K \rightarrow \underline{0}$ (use (2.83) to verify). Thus, the gradient does not necessarily vanish at the stationary point, but can vanish when the system diverges.

3. Accelerated Steepest Descent Method (ASD)

The ASD method derived in (II.C.2) is applied in this section with some modifications to design an adaptive filter which maximizes the performance function J in (2.25). The adaptive filter is updated according to (2.67).

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \cdot \underline{V}_K$$

The step direction vector $\underline{V}_K$ is computed from the filter vector $\underline{H}_K$ and the gradient $\underline{G}_K$ of the performance function J_K.

$$\underline{V}_K = \begin{cases} -\underline{G}_K & \text{for } K = 2, 4, 6 \ldots \\ H_K - H_{K-2} & K = 3, 5, 7 \ldots \end{cases}$$

The gradient $\underline{G}_K$ is obtained from (2.83) and the adaptation gain from (2.87) and Lemma 2.07.

The adaptive filter designed by the ASD method is carried out in the following steps:

[Step 1] Set a starting filter vector $\underline{H}_o = \underline{H}_1$, stopping bound δ and compute the performance function J_o and the gradient $\underline{G}_o$.

[Step 2] Compute the performance function value at iteration step K.

$$J_K = \frac{\underline{H}_K^T \underset{\sim}{R}_{SS} \underline{H}_K}{\underline{H}_K^T \underset{\sim}{R}_{NN} \underline{H}_K}$$

[Step 3] Compute the gradient $\underline{G}_K$ of J_K with respect to $\underline{H}_K$.

$$\underline{G}_K = \frac{2}{\underline{H}_K^T \underset{\sim}{R}_{NN} \underline{H}_K} \quad \underset{\sim}{Q}_K \cdot \underline{H}_K$$

where $\underset{\sim}{Q}_K$ is given by (2.98)

[Step 4] Compute the step direction vector $\underline{V}_K$.

$$\underline{V}_K = \begin{cases} -\underline{G}_K & \text{for } K = 2, 4, 6 \ldots \\ \underline{H}_K - \underline{H}_{K-2} & \text{for } K = 3, 5, 7 \ldots \end{cases}$$

[Step 5] Compute the adaptation gain:

$$\alpha_K = -\frac{\underline{H}_K^T \underset{\sim}{Q}_K \underline{G}_K}{\underline{G}_K^T \underset{\sim}{Q}_K \underline{G}_K}$$

[Step 6] Update the filter vector:

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \cdot \underline{V}_K$$

[Step 7] Test for stopping condition:

If $\frac{\| \underline{H}_{K+1} - \underline{H}_K \|}{\| \underline{H}_K \|} \leq \delta$ then terminate the adaption, otherwise go to step 2.

4. <u>Fletcher-Reeves Conjugate Gradient Method (CGF)</u>

The Fletcher-Reeves conjugate gradient (CGF) method is applied to the MSNR adaptive filter in a similar way as for the mMSE adaptive filter. However, the nonlinear MSNR performance function requires more computation and does not use the true "best step" but an approximation. The "restart" concept was used and found to be able to accelerate the convergence speed.

The adaptive filter based on the CGF method is updated by the following iterative scheme:

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \underline{V}_K \qquad (2.100)$$

The step direction vector $\underline{V}_K$ is obtained as in (II.C.4) by the expression:

$$\underline{V}_K = \begin{cases} -\underline{G}_K & \text{if } K \text{ Mod } n = 0 \\ -\underline{G}_K + \dfrac{\underline{G}_K^T \underline{G}_K}{\underline{G}_{K-1}^T \underline{G}_{K-1}} \cdot \underline{V}_{K-1} \end{cases} \qquad (2.101)$$

The adaptation gain α_K is obtained from Lemma (2.07) and given by:

$$\alpha_K = -\frac{\underline{H}_K^T (\underset{\sim}{R}_{SS} - J_K \cdot \underset{\sim}{R}_{NN}) \underline{V}_K}{\underline{V}_K^T (\underset{\sim}{R}_{SS} - J_K \cdot \underset{\sim}{R}_{NN}) \underline{V}_K} \qquad (2.102)$$

Using definition (2.90), we obtain:

$$\alpha_K = -\frac{\underline{H}_K^T \underset{\sim}{Q}_K \underline{V}_K}{\underline{V}_K^T \underset{\sim}{Q}_K \underline{V}_K} \qquad (2.103)$$

The adaptive filter designed by the CGF method is carried out in the following steps:

[Step 1] Select a starting filter vector $\underline{H}_o$ and a stopping bound δ.

[Step 2] Compute the performance function J_K.

$$J_K = \frac{\underline{H}_K^T \underset{\sim}{R}_{SS} \underline{H}_K}{\underline{H}_K^T \underset{\sim}{R}_{NN} \underline{H}_K}$$

[Step 3] Compute the gradient $\underline{G}_K$ of J_K with respect to $\underline{H}_K$.

$$\underline{G}_K = \frac{2}{\underline{H}_K^T \underset{\sim}{R}_{NN} \underline{H}_K} \cdot \underset{\sim}{Q}_K \cdot \underline{H}_K$$

where $\underset{\sim}{Q}_K$ is given by (2.98).

[Step 4] Compute the step direction vector $\underline{V}_K$.

$$\underline{V}_K = \begin{cases} -\underline{G}_K & \text{if K Mod n = 0.} \\ -\underline{G}_K + \dfrac{\underline{G}_K^T \underline{G}_K}{\underline{G}_{K-1}^T \underline{G}_{K-1}} \cdot \underline{V}_{K-1} & \text{Else,} \end{cases}$$

[Step 5] Compute the adaptation gain.

$$\alpha_K = - \frac{\underline{H}_K^T \underset{\sim}{Q}_K \underline{V}_K}{\underline{V}_K^T \underset{\sim}{Q}_K \underline{V}_K}$$

[Step 6] Update the filter vector $\underline{H}_K$.

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \underline{V}_K$$

[Step 7] Test for stopping condition.

If $\left\| \dfrac{\underline{H}_{K+1} - \underline{H}_K}{\| \underline{H}_K \|} \right\| \le \delta$ then terminate the adaptation.

Otherwise go to step 2.

5. Pollack-Rebiere Conjugate Gradient Method (CGP)

The CGP method is similar to the CGF method. The only difference is the way the step direction is computed.

The CGP method uses the following expression to compute the step direction vector $\underline{V}_K$.

$$\underline{V}_K = \begin{cases} -\underline{G}_K & \text{if K Mod n = 0} \\ -\underline{G}_K + \dfrac{\underline{G}_K^T (\underline{G}_K - \underline{G}_{K-1})}{\underline{G}_{K-1}^T \underline{G}_{K-1}} \cdot \underline{V}_{K-1} & \text{else,} \end{cases}$$

All the rest is identical to the CGF method. However, this method was found to converge much faster than the CGF for all the images tested in this thesis.

The adaptive filter designed by the CGP method is carried out in the following steps:

[Step 1] Select a starting filter vector $\underline{H}_o$ and a stopping bound δ.

[Step 2] Compute the performance function J_K.

$$J_K = \frac{\underline{H}_K^T \underset{\sim}{R}_{SS} \underline{H}_K}{\underline{H}_K^T \underset{\sim}{R}_{NN} \underline{H}_K}$$

[Step 3] Compute the gradient $\underline{G}_K$ of J_K with regard to $\underline{H}_K$.

$$\underline{G}_K = \frac{2}{\underline{H}_K \underset{\sim}{R}_{NN} \underline{H}_K} \underset{\sim}{Q}_K \underline{H}_K$$

where Q_K is given by (2.98)

[Step 4] Compute the step direction vector $\underline{V}_K$.

$$\underline{V}_K = \begin{cases} -\underline{G}_K & \text{if K Mod n = 0} \\ -\underline{G}_K + \dfrac{\underline{G}_K^T (\underline{G}_K - \underline{G}_{K-1})}{\underline{G}_{K-1}^T \cdot \underline{G}_{K-1}} \cdot \underline{V}_{K-1} & \text{Else,} \end{cases}$$

[Step 5] Compute the adaptation gain.

$$\alpha_K = - \frac{\underline{H}_K^T \underset{\sim}{Q}_K \underline{V}_K}{\underline{V}_K^T \underset{\sim}{Q}_K \underline{V}_K}$$

[Step 6] Update the filter vector $\underline{H}_K$.

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \underline{V}_K$$

[Step 7] Test for stopping condition:

If $\frac{\| \underline{H}_{K+1} - \underline{H}_K \|}{\| \underline{H}_K \|} \leq \delta$ then terminate the adaptation, otherwise go to step 2.

6. Davidon-Fletcher-Powell Variable Metric Method (DFP)

The DFP method is applied to the MSNR adaptive filter in a similar way as for the mMSE adaptive filter. The major difference is the approximation of the adaptation gain and the need to evaluate the performance function at every iteration step K.

The adaptive filter based on the DFP method is updated by the following iterative scheme:

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \cdot \underline{V}_K$$

The step direction vector $\underline{V}_K$ is obtained by the variable metric:

$$\underline{V}_K = - \underset{\sim}{A}_K \cdot \underline{G}_K \qquad (2.105)$$

The adaptation gain is obtained from Lemma (2.07):

$$\alpha_K = - \frac{\underline{H}_K^T \underset{\sim}{Q}_K \underline{V}_K}{\underline{V}_K^T \underset{\sim}{Q}_K \underline{V}_K} \qquad (2.106)$$

where $\underset{\sim}{Q}_K$ is given by (2.98). The metric $\underset{\sim}{A}_K$ is updated by the DFP iterative procedure:

$$\underset{\sim}{A}_{K+1} = \underset{\sim}{A}_K + \alpha_K \cdot \frac{\underline{V}_K \underline{V}_K^T}{\underline{V}_K^T \underline{P}_K} - \frac{\underset{\sim}{A}_K \underline{P}_K \underline{P}_K^T \underset{\sim}{A}_K}{\underline{P}_K^T \underset{\sim}{A}_K \underline{P}_K} \qquad (2.107)$$

The vector $\underline{P}_K$ in (2.107) is defined as:

$$\underline{P}_K \triangleq \underline{G}_{K+1} - \underline{G}_K \qquad (2.108)$$

The adaptive filter designed by the DFP method is carried out in the following steps:

[Step 1] Select a starting filter vector $\underline{H}_o$, the starting correction metric $\underset{\sim}{A}_o = \underset{\sim}{I}$ (where $\underset{\sim}{I}$ is the identity matrix), compute the gradient $\underline{G}_o$ of the performance function as before, set the stopping bound δ.

[Step 2] Compute the step direction vector $\underline{V}_K$.

$$\underline{V}_K = - \underset{\sim}{A}_K \underline{G}_K$$

[Step 3] Compute the adaptation gain α_K.

$$\alpha_K = - \frac{\underline{H}_K^T \underset{\sim}{Q}_K \underline{V}_K}{\underline{V}_K^T \underset{\sim}{Q}_K \underline{V}_K}$$

[Step 4] Update the filter vector $\underline{H}_K$.

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \underline{V}_K$$

[Step 5] Compute the performance function J_{K+1}.

$$J_{K+1} = \frac{\underline{H}_{K+1}^{T} \underset{\sim}{R}_{SS} \, \underline{H}_{K+1}}{\underline{H}_{K+1}^{T} \underset{\sim}{R}_{NN} \, \underline{H}_{K+1}}$$

[Step 6] Compute the gradient $\underline{G}_{K+1}$ of the performance function J_{K+1} with respect to the filter vector $\underline{H}_{K+1}$.

$$\underline{G}_{K+1} = \frac{2}{\underline{H}_{K+1}^{T} \underset{\sim}{R}_{NN} \, \underline{H}_{K+1}} \cdot \underset{\sim}{Q}_{K+1} \cdot \underline{H}_{K+1}$$

where $\underset{\sim}{Q}_{K+1}$ is given by updating (2.98).

[Step 7] Compute the vector $\underline{P}_K$ by (2.108).

$$\underline{P}_K = \underline{G}_{K+1} - \underline{G}_K$$

[Step 8] Update the correction matrix A_{K+1} by (2.107).

$$\underset{\sim}{A}_{K+1} = \underset{\sim}{A}_K + \alpha_K \cdot \frac{\underline{V}_K \, \underline{V}_K^{T}}{\underline{V}_K^{T} \, \underline{P}_K} - \frac{\underset{\sim}{A}_K \, \underline{P}_K \, \underline{P}_K^{T} \, \underset{\sim}{A}_K}{\underline{P}_K^{T} \, \underset{\sim}{A}_K \, \underline{P}_K} \qquad (2.107)$$

[Step 9] Test for stopping condition.

This step can be done after step 4 to save some extra computations but was placed here to follow the consistent pattern as all other methods.

If $\frac{\| \underline{H}_{K+1} - \underline{H}_K \|}{\| \underline{H}_K \|} \leq \delta$ then terminate the adaptation procedure, otherwise go to step 2.

7. Amir's Transform Method (AT)

From test results, which will be shown in (2.17 – 2.28), it was observed that a faster convergence method will be helpful for designing the MSNR adaptive filter. Both the conjugate gradient method by Pollack and the variable metric method following Davidon do not exhibit the same convergence speed as for the quadratic mMSE case. The reason for this slower convergence for the MSNR performance function is the nonlinear nonquadratic performance function as shown in (2.25). It was then decided to derive a method tailored for this performance function. The derivation of this method is based on the generalized eigenvalue/eigenvector problem introduced by the stationary points of the performance function J in (2.25). The stationary point of the performance function J in (2.25) satisfies (2.40) which can be written in the form:

$$(\underset{\sim}{R}_{NN}^{-1} \underset{\sim}{R}_{XX} - J_{max} \cdot I) \cdot \underline{H}^* = 0 \qquad (2.109)$$

where $\underline{H}^*$ is the optimal filter vector which maximizes the performance function J.

The optimal filter $\underline{H}^*$ satisfies

$$\underline{H}^* = \frac{1}{J_{max}} \cdot \underset{\sim}{R}_{NN}^{-1} \cdot \underset{\sim}{R}_{SS} \cdot \underline{H}^* \qquad (2.110)$$

From equation (2.110), it is obvious that an adaptive filter designed by using the transform matrix $\frac{1}{J} \cdot \underset{\sim}{R}_{NN}^{-1} \cdot \underset{\sim}{R}_{SS}$ for updating the filter will satisfy (2.110) if it converges to the optimum.

In order to accelerate the convergence of such an adaptive filter, a gradient search is added to update the filter vector. The steepest descent search direction is adopted. The "best step" concept is used partially to compute the adaptation gain.

At iteration step K + 1, the filter update equation is described by:

$$\underline{H}_{K+1} = \frac{1}{J_K} \cdot \underset{\sim}{R}_{NN}^{-1} \cdot \underset{\sim}{R}_{SS} \cdot \underline{H}_K + \alpha_K \cdot \underline{G}_K \qquad (2.111)$$

The transform matrix $\underset{\sim}{M}_K$ is defined as:

$$\underset{\sim}{M}_K \triangleq \frac{1}{J_K} \cdot \underset{\sim}{R}_{NN}^{-1} \cdot \underset{\sim}{R}_{SS} \qquad (2.112)$$

Using (2.112) in (2.111), we obtain:

$$\underline{H}_{K+1} = \underset{\sim}{M}_K \cdot \underline{H}_K + \alpha_K \cdot \underline{G}_K \qquad (2.113)$$

The adaptation gain for the AT method can be obtained by Lemma 2.05.

$$\underline{G}_{K+1}^T \cdot \underline{G}_K = 0 \qquad (2.57)$$

From (2.83) and (2.98), the gradient $\underline{G}_{K+1}$ of J_{K+1} is:

$$\underline{G}_{K+1} = \frac{2}{\underline{H}_{K+1}^T R_{NN} \underline{H}_{K+1}} \cdot \underset{\sim}{Q}_{K+1} \cdot \underline{H}_{K+1} \qquad (2.114)$$

Using (2.113) and (2.114) in (2.57), we obtain:

$$\left[\frac{2}{\underline{H}_{K+1}^{T} \underset{\sim}{R}_{NN} \underline{H}_{K+1}} \cdot \underset{\sim}{Q}_{K+1} \left(\underset{\sim}{M}_K \underline{H}_K + \alpha_K \underline{G}_K\right)\right]^T \cdot \underline{G}_K = 0 \tag{2.115}$$

$$\left(\underset{\sim}{Q}_{K+1} \underset{\sim}{M}_K \underline{H}_{,} + \alpha_K \underset{\sim}{Q}_{K+1} \underline{G}_K\right)^T \cdot \underline{G}_K = 0 \tag{2.116}$$

(2.116) can be viewed as a dot product between two orthogonal vectors, and the expression can be modified to be:

$$\underline{G}_K^T \cdot \left(\underset{\sim}{Q}_{K+1} \cdot \underset{\sim}{M}_K \underline{H}_K + \alpha_K \underset{\sim}{Q}_{K+1} \underline{G}_K\right) = 0 \tag{2.117}$$

Solving (2.117) for α_K, we obtain:

$$\alpha_K = -\frac{\underline{G}_K^T \cdot \underset{\sim}{Q}_{K+1} \cdot \underset{\sim}{M}_K \cdot \underline{H}_K}{\underline{G}_K^T \underset{\sim}{Q}_{K+1} \underline{G}_K} \tag{2.118}$$

Since $\underset{\sim}{Q}_{K+1}$ is not known, we use Lemma (2.07) to approximate α_K.

$$\alpha_K = -\frac{\underline{G}_K^T \underset{\sim}{Q}_K \underset{\sim}{M}_K \underline{H}_K}{\underline{G}_K^T \underset{\sim}{Q}_K \underline{G}_K} \tag{2.119}$$

In order for the adaptation gain α_K in (2.119) to be acceptable (i.e., the adaptive filter will converge), it must satisfy the condition (2.90).

The adaptive filter designed by the AT method is carried out in the following steps:

[Step 1] Select a starting filter vector $\underline{H}_o$, and a stopping bound δ.

[Step 2] Compute the performance function J_K as in (2.25).

$$J_K = \frac{\underline{H}_K^T \underset{\sim}{R}_{SS} \underline{H}_K}{\underline{H}_K^T \underset{\sim}{R}_{NN} \underline{H}_K}$$

[Step 3] Compute the gradient $\underline{G}_K \triangleq \nabla_{\underline{H}_K} J_K$.

$$\underline{G}_K = \frac{2}{\underline{H}_K^T \underset{\sim}{R}_{NN} \underline{H}_K} \cdot \underset{\sim}{Q}_K \underline{H}_K$$

[Step 4] Compute the adaptation gain.

$$\alpha_K = - \frac{\underline{H}_K^T \underset{\sim}{Q}_K \underline{G}_K}{\underline{G}_K^T \underset{\sim}{Q}_K \underline{G}_K}$$

[Step 5] Update the filter vector $\underline{H}_K$ according to (2.113).

$$\underline{H}_{K+1} = \underset{\sim}{M}_K \underline{H}_K + \alpha_K \underline{G}_K$$

[Step 6] Test for stopping condition.

If $\dfrac{||\underline{H}_{K+1} - \underline{H}_K||}{||\underline{H}_K||} \leq \delta$, then terminate the adaptation, otherwise go to step 2.

E. CONVERGENCE AND CONVERGENCE RATE OF THE GRADIENT METHODS

1. SD Adaptive Filter

Theorem 2.09

For any starting filter vector $\underline{H}_0$, the sequence $\{\underline{H}_i\}$ of the adaptive filter given by (2.56) converges to

the unique optimal solution $\underline{H}^*$ given by (2.12). Furthermore, the rate of convergence satisfies

$$\rho \triangleq \frac{\| \underline{H}_{K+1} - \underline{H}^* \|}{\| \underline{H}_K - \underline{H}^* \|} \leq \frac{1}{\beta} \cdot \frac{1 - C}{1 + C} \tag{2.120}$$

where C is the condition number of the Hessian matrix R_{XX} of the performance function J given in (2.10) and β is a constant. The condition number is defined as

$$C \triangleq \frac{\lambda_S}{\lambda_L} \tag{2.121}$$

where λ_L, λ_S are the largest and smallest eigenvalues of the Hessian matrix $\underset{\sim}{R}_{XX}$.

Proof

The Kantorovich inequality is used to prove the theorem.

The functional Ψ_K is defined as:

$$\Psi_K \triangleq (\underline{H}_K - \underline{H}^*)^T \cdot \underset{\sim}{R}_{XX}(\underline{H}_K - \underline{H}^*) \tag{2.122}$$

where $\underset{\sim}{R}_{XX}$ is the Hessian matrix of the performance function in (2.10). For the adaptive filter $\underset{\sim}{R}_{XX}$ is the correlation matrix of the observed image signal.

$\underline{H}^*$ is the filter which minimizes (2.10). The filter vector $\underline{H}^*$ is called the optimal filter. Ψ is updated at iteration step K + 1 as:

$$\Psi_{K+1} = (\underline{H}_{K+1} - \underline{H}^*)^T \underset{\sim}{R}_{XX}(\underline{H}_{K+1} - \underline{H}^*) \tag{2.123}$$

Using (2.56) to substitute for $\underline{H}_{K+1}$ in (2.133), we have

$$\Psi_{K+1} = (\underline{H}_K + \alpha_K \underline{G}_K - \underline{H}^*)^T \underset{\sim}{R}_{XX}(\underline{H}_K + \alpha_K \underline{G}_K - \underline{H}^*) \tag{2.124}$$

Using the definition (2.122) and the fact that $\underset{\sim}{R}_{XX}$ is a symmetric matrix, we obtain:

$$\Psi_{K+1} = \Psi_K + 2\alpha_K \underline{G}_K^T \cdot \underset{\sim}{R}_{XX}(\underline{H}_K - \underline{H}^*) + \alpha_K^2 \underline{G}_K^T \underset{\sim}{R}_{XX} \underline{G}_K$$

The adaptation gain α_K is given by (2.66), which can be used in (2.125) to obtain:

$$\begin{aligned}\Psi_{K+1} &= \Psi_K + 2\alpha_K \cdot \underline{G}_K^T \underset{\sim}{R}_{XX}(\underline{H}_K - \underline{H}^*) \\ &\quad - \frac{1}{2} \cdot \alpha_K \cdot \frac{\underline{G}_K^T \underline{G}_K}{\underline{G}_K^T \underset{\sim}{R}_{XX} \underline{G}_K} \cdot \underline{G}_K^T \underset{\sim}{R}_{XX} \underline{G}_K \\ &= \Psi_K + 2\alpha_K \underline{G}_K^T \underset{\sim}{R}_{XX}(\underline{H}_K - \underline{H}^*) - \frac{1}{2} \alpha_K \cdot \underline{G}_K^T \underline{G}_K\end{aligned} \tag{2.126}$$

Using (2.63) and (2.12) in (2.126), we obtain:

$$\begin{aligned}2 \cdot \underset{\sim}{R}_{XX}(\underline{H}_K - \underline{H}^*) &= 2(\underset{\sim}{R}_{XX} \underline{H}_K - \underline{R}_{XS} \underbrace{+ \underline{R}_{XS} - \underset{\sim}{R}_{XX} \underline{H}^*}_{0}) \\ &= \underline{G}_K\end{aligned} \tag{2.127}$$

Substitution of (2.127) in (2.126) gives:

$$\Psi_K = \Psi_K + \alpha_K \underline{G}_K^T \cdot \underline{G}_K - \frac{1}{2} \alpha_K \underline{G}_K^T \underline{G}_K$$

$$= \Psi_K + \frac{1}{2} \alpha_K \underline{G}_K^T \underline{G}_K \qquad (2.128)$$

Let us define the vector $\underline{E}_K$ as

$$\underline{E}_K \overset{\Delta}{=} \underline{H}_K - \underline{H}^* \qquad (2.129)$$

Using this definition (2.129) in (2.122), we obtain:

$$\Psi_K = \underline{E}_K^T \underset{\sim}{R}_{XX} \underline{E}_K \qquad (2.130)$$

From (2.127), we obtain:

$$\underline{E}_K = \frac{1}{2} \cdot \underset{\sim}{R}_{XX}^{-1} \underline{G}_K \qquad (2.131)$$

Using (2.131) and the fact that $\underset{\sim}{R}_{XX}$ is symmetric, (2.130) became:

$$\Psi_K = \frac{1}{4} \cdot \underline{G}_K^T \underset{\sim}{R}_{XX} \underline{G}_K \qquad (2.132)$$

Using (2.132) in (2.128), we obtain:

$$\frac{\Psi_{K+1} - \Psi_K}{\Psi_K} = \frac{\frac{1}{2} \alpha_K \cdot \underline{G}_K^T \underline{G}_K}{\frac{1}{4} \cdot \underline{G}_K^T R_{XX}^{-1} \underline{G}_K} \qquad (2.133)$$

Substitution of α_K given by (2.66) in (2.133) gives:

$$\frac{\Psi_{K+1} - \Psi_K}{\Psi_K} = - \frac{\underline{G}_K^T \underline{G}_K}{\underline{G}_K^T \underset{\sim}{R}_{XX}^{-1} \underline{G}_K} \cdot \frac{\underline{G}_K^T \underline{G}_K}{\underline{G}_K^T \underset{\sim}{R}_{XX} \underline{G}_K} \qquad (2.134)$$

Now the Kantorovich inequality is used:

$$\frac{(\underline{G}_K^T \underset{\sim}{R}_{XX} \underline{G}_K)(\underline{G}_K^T \underset{\sim}{R}_{XX}^{-1} \underline{G}_K)}{(\underline{G}_K^T \underline{G}_K)^2} \leq \frac{(\lambda_S + \lambda_L)^2}{4\lambda_S \cdot \lambda_L} \qquad (2.135)$$

where λ_S and λ_L are the smallest and largest eigenvalues of the matrix R_{XX}.

Using (2.135) in (2.134), we obtain:

$$\frac{\Psi_{K+1} - \Psi_K}{\Psi_K} \leq - \frac{4\lambda_S \lambda_L}{(\lambda_S + \lambda_L)^2} \qquad (2.136)$$

$$\frac{\Psi_{K+1}}{\Psi_K} \leq \left(\frac{\lambda_L - \lambda_S}{\lambda_L + \lambda_S} \right)^2 \leq 1 \qquad (2.137)$$

Again, the condition number of the matrix $\underset{\sim}{R}_{XX}$ is defined as

$$C \triangleq \frac{\lambda_S}{\lambda_L} \qquad (2.138)$$

(2.137) became

$$\frac{\Psi_{K+1}}{\Psi_K} \leq \left(\frac{1 - C}{1 + C} \right)^2 \leq 1 \qquad (2.139)$$

Since $\underset{\sim}{R}_{XX}$ is a positive definite matrix, the sequence $\{\Psi_K\}$ is a positive sequence.

Let us define

$$q^2 \triangleq \left(\frac{1 - C}{1 + C}\right)^2 \leq 1 \qquad (2.140)$$

we obtain

$$\Psi_{K+1} \leq \Psi_K \cdot q^2 \qquad (2.141)$$

From (2.141), we can see that when $k \to \infty$, the sequence $\{\Psi_K\}$ converge to zero. The reason is that we have a decreasing positive sequence $\{\Psi_K\}$, thus $\Psi_\infty = 0$. It implies $\underline{E}_\infty = \underline{0}$ (use (2.130) to justify this statement). Since $\underline{E}_K$ is defined as $\underline{H}_\infty - H^*$ in (2.129), we conclude that:

$$\underline{H}_\infty - \underline{H}^* = 0 \quad \text{or} \quad \underline{H}_\infty = \underline{H}^* \tag{2.142}$$

This completed the proof of convergence.

From (2.139), we observe that the rate of convergence of the sequence $\{\Psi_K\}$ is given by (2.140). However, Ψ_K as defined in (2.122) is a quadratic function of the vector $\underline{E}_K = \underline{H}_K - \underline{H}^*$, it satisfies the relation

$$\frac{\Psi_{K+1}}{\Psi_K} = \beta \cdot \frac{\|\underline{H}_{K+1} - \underline{H}^*\|^2}{\|\underline{H}_K - \underline{H}^*\|^2} \leq \left(\frac{1 - C}{1 + C}\right)^2 \tag{2.144}$$

where β is a positive number.

Thus we obtain:

$$\rho = \frac{\|\underline{H}_{K+1} - \underline{H}^*\|}{\|\underline{H}_K - \underline{H}^*\|} \leq \frac{1}{\beta} \cdot \frac{1 - C}{1 + C} \tag{2.145}$$

(Q.E.D.)

Theorem (2.09) proves that the SD method exhibits at least linear convergence.

Definition

An algorithm with the property that $\rho = \frac{\|\underline{H}_{K+1} - \underline{H}^*\|}{\|\underline{H}_K - H^*\|} = \text{constant}$ is said to exhibit linear convergence.

The linear convergence is sometimes called geometric convergence since it follows from the definition that for large K, j

$$\cdot\| \underline{H}_K - \underline{H}^* \| \simeq a^{K-j} \| \underline{H}_j - \underline{H}^* \|$$

The speed of convergence of the SD method is a function of the condition number C. The more ill-conditioned $\underset{\sim}{R}_{XX}$, the slower will be the rate of convergence.

Theorem (2.09) used the mMSE quadratic function. For the MSNR performance function, it was shown (II.E.1) that the sequence $\{\underline{H}_K\}$ generated by the SD method converges. Test results showed that the convergence of SD is slower for MSNR than mMSE.

2. ASD Adaptive Filter

The algorithm is illustrated in Fig. 2.01.

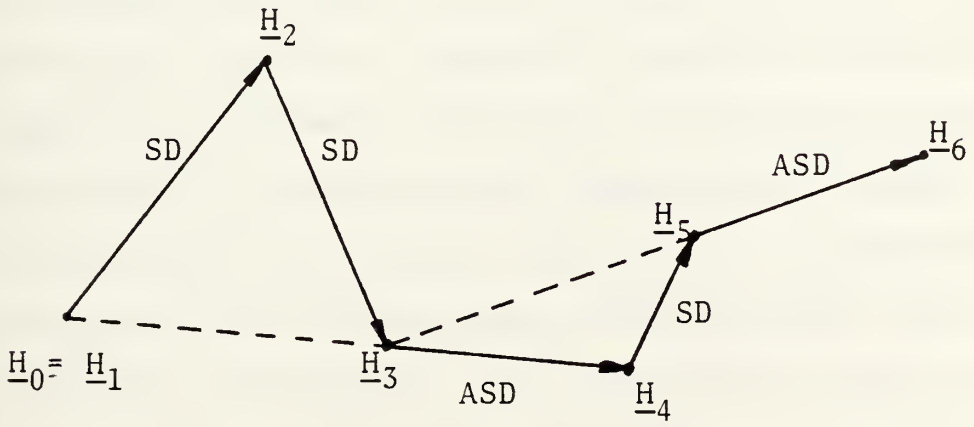


Fig. 2.01 The ASD algorithm.

The steepest descent steps are labeled SD, the accelerated steps are labeled ASD.

Shah, Buehler and Kempthore [53] showed that for an n dimensional quadratic function, the sequence of iterates H_0, H_2, H_4 ... is identical to the full sequence of iterates generated by the conjugate-gradient descent. Since the conjugate gradient descent takes no more than n steps to reach the minimum of the n dimensional quadratic function, the accelerated steepest descent takes no more than (2n-1) steps.

Applying the ASD method to design a multidimensional adaptive filter using real test screen images has shown poor convergence speed for both the mMSE and MSNR performance functions. The reason is due to error propagation. These methods are sensitive to error propagation, which do not satisfy the condition for accelerated convergence.

3. CGF Adaptive Filter

The conjugate gradient methods CGP and CGF exhibit quadratic termination (apart from rounding errors) for the mMSE performance function. Quadratic termination means that for a quadratic performance function it is guaranteed that the minimum will be located exactly (apart from rounding errors) in no more than n steps. However, for nonquadratic functions like (2.25) the conjugate gradient method does not exhibit quadratic termination. For the infinite dimensional case, Daniel [60] showed that the rate of convergence is:

$$\rho = \frac{\| \underline{H}_{K+1} - \underline{H}^* \|}{\| \underline{H}_K - \underline{H}^* \|} \leq \frac{\sqrt{\lambda_L} - \sqrt{\lambda_S}}{\sqrt{\lambda_L} - \sqrt{\lambda_S}}$$

where λ_L, λ_S are the largest and smallest eigenvalues of the Hessian matrix of the performance function J.

Depending on the approach to design, the adaptive filter for nonlinear, nonquadratic performance function, different rates of convergence can be obtained. Some approaches exhibit quadratic convergence (those which approximate the performance function by a Taylor series expansion). Others exhibit superlinear convergence.

Theorem 2.10

Let the performance function J be defined by (2.10) and the adaptive filter be designed using the conjugate gradient method, then the sequence of adaptive filters $\{\underline{H}_K\}$ converges in no more than n steps to the unique minimum $\underline{H}^*$ of the performance function J.

Proof

The proof is based on the fact that both methods, CGF and CGP, are based on the conjugate direction search method which implies that the step direction vector $\underline{V}_K$ is orthogonal to the gradient of the performance function J at iteration step K + 1. This fact is stated as: [55, 61]

$$\underline{G}_{K+1}^T \cdot \underline{V}_K = 0 \tag{2.146}$$

The adaptation equation is:

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \underline{V}_K \tag{2.147}$$

Its expression at the iteration step n can be related to all steps from iteration step K by:

$$\underline{H}_n = \underline{H}_{K+1} + \sum_{i=K+1}^{n-1} \alpha_j \underline{V}_j \tag{2.148}$$

for any $0 \leq K \leq n - 1$.

The gradient of the performance function J at iteration step n is given by:

$$\underline{G}_n = 2(\underset{\sim}{R}_{XX} \underline{H}_n - \underline{R}_{XS}) \tag{2.149}$$

By substituting (2.148) in (2.149), we get:

$$\underline{G}_n = \underline{G}_{K+1} + \sum_{j=K+1}^{n-1} \alpha_j \underset{\sim}{R}_{XX} \underline{V}_j \tag{2.150}$$

Using equation (2.146) in (2.150), we obtain:

$$\underline{V}_K^T \underline{G}_n = \sum_{j=K+1}^{n-1} \underline{V}_K^T \underset{\sim}{R}_{XX} \underline{V}_j \tag{2.151}$$

The method of conjugate gradient is based on generating a conjugate sequence of step direction vector $\{\underline{V}_K\}$.

The conjugacy condition satisfies:

$$\underline{V}_K^T \underset{\sim}{R}_{XX} \underline{V}_j = 0 \qquad \text{for } K \neq j. \tag{2.152}$$

We use (2.152) in (2.151) to show that:

$$\underline{V}_K^T \underline{G}_n = 0 \tag{2.153}$$

The step direction vectors $\underline{V}_o$, $\underline{V}_1, \ldots \underline{V}_{n-1}$ form a complete conjugate basis. Therefore, at iteration step n, the only $\underline{G}_n$ which satisfies (2.153) is $\underline{G}_n = \underline{0}$. (2.154)

But for the quadratic performance function, the gradient vanishes only at the minimum. So we proved that the method converges to the minimum of J in (2.10) in no more than n steps.

Substituting (2.154) in (2.149), we obtain

$$\underset{\sim}{R}_{XX} \underline{H}_n - \underset{\sim}{R}_{XS} = 0 \qquad (2.155)$$

$$\underline{H}_n = \underset{\sim}{R}_{XX}^{-1} \underline{R}_{XS} \qquad (2.156)$$

So

$$\underline{H}_n = \underline{H}^* \qquad (2.157)$$

Thus the filter converges to the unique minimum of J.

Q.E.D.

In practical applications, it was found that the conjugate gradient methods converge sometimes in more than n steps. The reason is the round-off error. The two conditions (2.146) and (2.152) are not satisfied, so the sequence $\{\underline{V}_K\}$ of step directions does not form a complete basis in n iteration steps.

For the MSNR cases, the adaptive filter could not converge as fast as in the mMSE cases because the performance function J in the MSNR is nonquadratic.

4. The DFP Method

The variable metric DFP method exhibits quadratic termination, apart from rounding errors, for the mMSE performance function.

Fletcher and Powell [58] proved for a general performance function J that a positive definite variable metric $\underset{\sim}{A}_K$ implies a positive definite $\underset{\sim}{A}_{K+1}$, updated by (2.77). They showed that for a quadratic function like the mMSE type, successive filter updates $\Delta\underline{H}_0$, $\Delta\underline{H}_1$ ... $\Delta\underline{H}_{n-1}$ form a set of conjugate directions, and $\underset{\sim}{A}_{n-1} = \underset{\sim}{R}_{XX}^{-1}$, so the DFP algorithm exhibits quadratic termination in n steps.

The MSNR performance function is nonquadratic and nonlinear, so the DFP method cannot exhibit quadratic termination. But according to our test results, it is still a fast convergence technique. If the method converges slowly, it is recommended to restart the variable metric every n+1 steps by setting $\underset{\sim}{A}_{K+1} = \underset{\sim}{I}$, to overcome round-off errors.

5. The AT Adaptive Filter

The Amir transform adaptive filter exhibits very fast convergence speed. The reason lies in the way it was designed. Each iterative step uses a transform to satisfy the generalized eigenvalue and eigenvector steady state equation.

Theorem 2.11

Let the adaptive filter be updated by (2.111) and the performance function defined by (2.25). Then the filter $\underline{H}_K$

converges to the unique optimal filter $\underline{H}^*$, if the adaptation gain α_K satisfies condition (2.90).

The adaptive filter $\underline{H}_K$ is updated by (2.111)

$$\underline{H}_{K+1} = \frac{1}{J_K} \cdot \underset{\sim}{R}_{NN}^{-1} \cdot \underset{\sim}{R}_{SS} \cdot \underline{H}_K + \alpha_K \underline{G}_K \tag{2.111}$$

Substituting (2.83) for $\underline{G}_K$ in (2.111), and defining

$$\beta_K \triangleq \underline{H}_K^T \underset{\sim}{R}_{NN} \underline{H}_K \tag{2.158}$$

we obtain:

$$\underline{H}_{K+1} = \frac{1}{J_K} \underset{\sim}{R}_{NN}^{-1} \underset{\sim}{R}_{SS} \underline{H}_K + \frac{2\alpha_K}{K} \cdot (\underset{\sim}{R}_{SS} - J_K \underset{\sim}{R}_{NN}) \underline{H}_K \tag{2.159}$$

Rearranging (2.159) gives:

$$\underline{H}_{K+1} = [\frac{1}{J_K} \underset{\sim}{R}_{NN}^{-1} \underset{\sim}{R}_{SS} + \frac{2J_K\alpha_K}{\beta_K} \underset{\sim}{R}_{NN}^{-1} (\frac{1}{J_K} \underset{\sim}{R}_{NN}^{-1} \underset{\sim}{R}_{SS} - \underset{\sim}{I})] \cdot \underline{H}_K \tag{2.160}$$

Subtracting $\underline{H}_K$ from both sides, we obtain:

$$\underline{H}_{K+1} - \underline{H}_K = (\underset{\sim}{I} + \frac{2\alpha_K J_K}{\beta_K} \cdot \underset{\sim}{R}_{NN}^{-1})(\frac{1}{J_K} \underset{\sim}{R}_{NN}^{-1} \underset{\sim}{R}_{SS} - \underset{\sim}{I}) \underline{H}_K \tag{2.161}$$

$\underset{\sim}{I}$ is the identity matrix.

Let us define the matrix $\underset{\sim}{Z}_K$ as:

$$\underset{\sim}{Z}_K = \underset{\sim}{I} + \frac{2 \;_K J_K}{K} \underset{\sim}{R}_{NN}^{-1} \tag{2.162}$$

Since $\underset{\sim}{R}_{NN}$, $\underset{\sim}{R}_{SS}$ are positive definite and α_K, J_K, β_K are positive numbers, thus $\underset{\sim}{Z}_K$ is positive definite. Since α_K, J_K, β_K are bounded, the norm of the matrix $\underset{\sim}{Z}_K$ is bounded.

In other words, there exists a positive number   such that:

$$\| \underset{\sim}{Z}_K \| \leq \lambda \tag{2.163}$$

Taking the norm of (2.161) and using the inequality,

$$\| \underset{\sim}{A} \cdot \underset{\sim}{B} \| \leq \| \underset{\sim}{A} \| \cdot \| \underset{\sim}{B} \| \tag{2.164}$$

where $\underset{\sim}{A}$, $\underset{\sim}{B}$ are matrices, and combining with (2.163), we obtain:

$$\| \underline{H}_{K+1} - \underline{H}_K \| \leq \lambda \cdot \| \frac{1}{J_K} \cdot \underset{\sim}{R}_{NN}^{-1} \underset{\sim}{R}_{SS} - \underset{\sim}{I} \| \cdot \| \underline{H}_K \| \tag{2.165}$$

which turns out to be

$$\frac{\| \underline{H}_{K+1} - \underline{H}_K \|}{\| \underline{H}_K \|} \leq \lambda \cdot \| \frac{1}{J_K} \cdot \underset{\sim}{R}_{NN}^{-1} \underset{\sim}{R}_{SS} - \underset{\sim}{I} \| \tag{2.166}$$

If the convergence error ε_K is defined as

$$\varepsilon_K \triangleq \frac{\| \underline{H}_{K+1} - \underline{H}_K \|}{\| \underline{H}_K \|} \tag{2.167}$$

$$\therefore \varepsilon_K \leq \lambda \cdot \| \frac{1}{J_K} \underset{\sim}{R}_{NN}^{-1} \underset{\sim}{R}_{XX} - \underset{\sim}{I} \| \tag{2.168}$$

The largest eigenvalue of $\frac{1}{J_K} \underset{\sim}{R}_{NN}^{-1} \underset{\sim}{R}_{SS} - \underset{\sim}{I}$ is

$$\frac{J_{max}}{J_K} - 1 \tag{2.169}$$

where J_{max} is the largest eigenvalue of $\underset{\sim}{R}_{NN}^{-1} \underset{\sim}{R}_{SS}$.

But

$$\| \frac{1}{J_K} \cdot \underset{\sim}{R}_{NN}^{-1} \underset{\sim}{R}_{SS} - \underset{\sim}{I} \| = \frac{J_{max}}{J_K} - 1 \tag{2.170}$$

So

$$\varepsilon_K \leq \lambda \cdot \left(\frac{J_{max}}{J_K} - 1 \right) \qquad (2.171)$$

The adaptation gain α_K is designed to satisfy condition (2.90) which states that:

$$J_{K+1} \geq J_K > 0$$

Updating (2.171) and using condition (2.90), we have:

$$\text{I.} \qquad \varepsilon_{K+1} \leq \lambda \cdot \left(\frac{J_{max}}{J_{K+1}} - 1 \right)$$

$$\text{II.} \qquad \varepsilon_K \leq \lambda \cdot \left(\frac{J_{max}}{J_K} - 1 \right) \qquad (2.172)$$

$$\text{III.} \qquad J_K \leq J_{K+1}$$

Thus, the sequence $\{\varepsilon_K\}$ converges to zero, because J_{max} is the maximum value of the unimodal performance function, and the sequence $\{J_K\}$ is an ascending sequence bounded by the upper bound J_{max},

so

$$\lim_{\to\infty} \varepsilon_K = \lambda \cdot \lim_{K=\infty} \left(\frac{J_{max}}{J_K} - 1 \right) = \lambda \cdot \left(\frac{J_{max}}{J_{max}} - 1 \right)$$

$$= 0 \qquad (2.173)$$

This proves that the filter converges. At the convergence point, (2.170) satisfies

$$\left\| \frac{1}{J_\infty} \underset{\sim}{R}_{NN}^{-1} \underset{\sim}{R}_{SS} - \underset{\sim}{I} \right\| = 0 \qquad (2.174)$$

or, in the vector form

$$\left(\frac{1}{J_\infty} \cdot \underset{\sim}{R}_{NN}^{-1} \underset{\sim}{R}_{SS} - \underset{\sim}{I}\right) \cdot \underline{H}_\infty = 0 \qquad (2.175)$$

(2.175) is the equation for the stationary points of the performance function J. Thus, $J_\infty = J_{max}$ and correspondingly $\underline{H}_\infty = \underline{H}^*$.

So the adaptive filter converges to the unique optimum.

(Q.E.D.)

F. PRESENTATION OF RESULTS

1. Organization of Results

The performances of both mMSE and MSNR nonrecursive adaptive spatial filters have been extensively evaluated on two real world infrared images, shown in Fig. 2.1 and 2.2. Before the detailed presentation of these results, a detailed description is given of how the evaluations are organized.

(a) Filter type:

- Nonrecursive adaptive spatial filter
- Search box (filter size) 3 by 3 pixels with the estimation pixel in the middle of the filter

(b) Optimization criterion and performance function:

- mMSE: Minimization of mean square error
- MSNR: Maximization of signal to noise ratio

(c) Adaptation equation:

1. LMS approach:

$$\underline{H}_{K+1} = \underline{H}_K + 2\mu\varepsilon\underline{X}$$

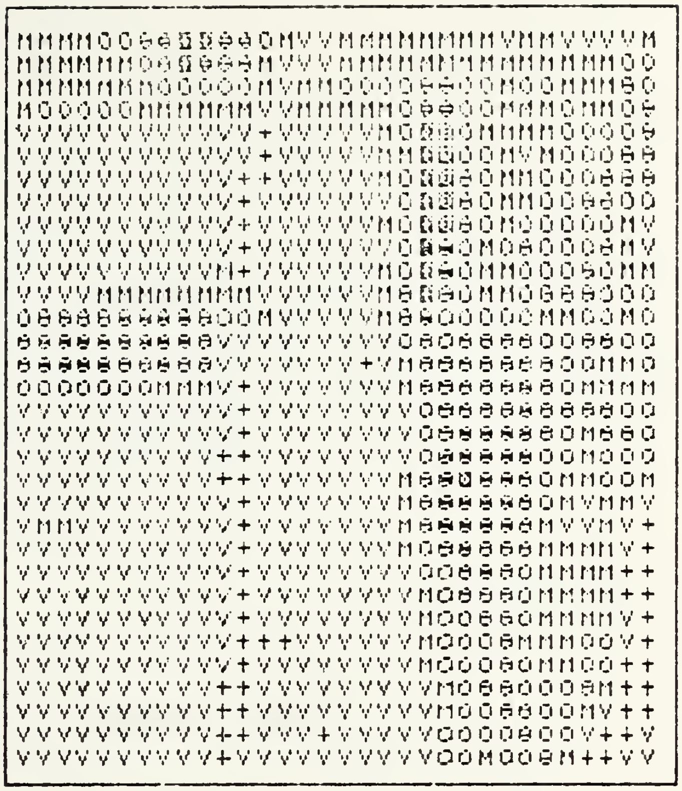


Fig. 2.1 A 9 level computer print of Indiana infrared test image.

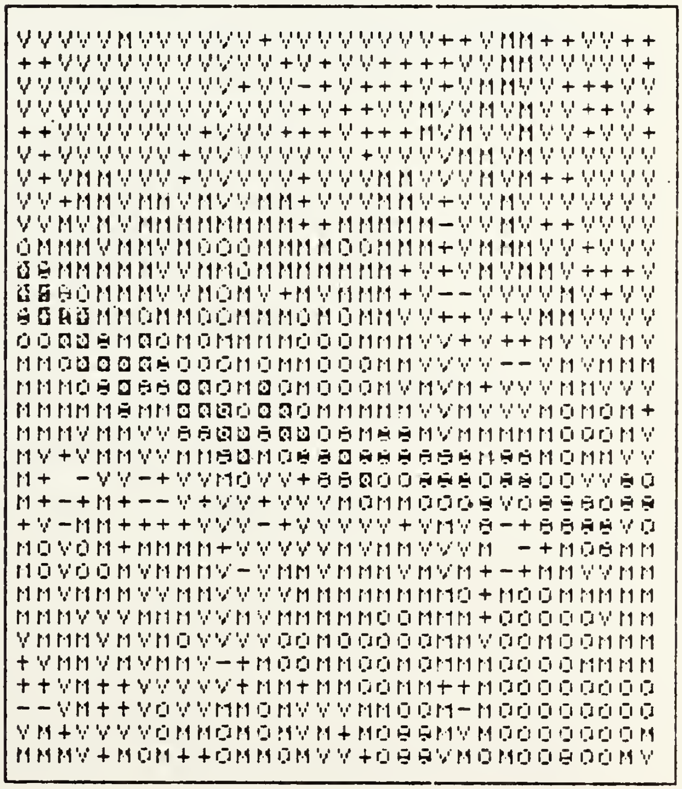


Fig. 2.2 A 9 level computer print of China Lake infrared test image.

2. Gradient approaches:

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \underline{G}_K$$

3. Conjugate gradient approaches:

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \underline{V}_K$$

4. Variable metric approach:

$$\underline{H}_{K+1} = \underline{H}_K + \alpha_K \underline{V}_K$$

5. Amir's transform approach:

$$\underline{H}_{K+1} = \underline{M}_K + \alpha_K \underline{G}_K$$

(d) Search methods:

1. LMS approach:

Steepest descent method

2. Gradient approaches:

Steepest descent method
Accelerated steepest descent method
Amir's method (apply only to mMSE case)

3. Conjugate gradient approaches:

Fletcher-Reeves method
Pollack method

4. Variable metric approaches:

Davidon-Fletcher-Powell method

5. Amir's transform approach:

Apply only to MSNR case

(e) Test images used:

1. Indiana image (Fig. 2.1):

32 x 32 pixels
Blue spike infrared spectral band

An image taken from a city in Indiana and used quite extensively as a standard test image for high altitude downward looking infrared surveillance system.

2. China Lake image (Fig. 2.2):

   32 x 32 pixels

   Thermal infrared band in 10-13μ range

   An image taken from a desert area in China Lake, California with a highway in the picture. It has been used as one of the standard test images for short distance side looking infrared target acquisition system.

(f) Performance evaluation:

The performance of the adaptive filters is presented in four different ways, all as a function of the number of iterative steps N.

1. Filter coefficients as a function of N. (9 coefficients for a 3 x 3 spatial filter)

2. Output variance as a function of N.

3. Processing gain as a function of N. The processing gain is defined as follows:

$$PG = 10 \log \frac{m_i^2 + \sigma_i^2}{m_0^2 + \sigma_0^2}$$

where m_i, m_0 = means of the input and filtered images respectively.

σ_i^2, σ_0^2 = variances of the input and filtered images respectively.

4. Output signal to noise ratio (used only in MSNR cases):

   Output SNR of the filtered image is defined as follows:

$$SNR_0 = \frac{\underline{H}^T \underset{\sim}{R}_{SS} \underline{H}}{\underline{H}^T \underset{\sim}{R}_{NN} \underline{H}}$$

where $\underline{H}$ = the filter vector

$\underset{\sim}{R}_{SS}$ = target signal correlation matrix

$\underset{\sim}{R}_{NN}$ = clutter noise correlation matrix

2. Results of mMSE Adaptive Spatial Filters I - Indiana Image

The test results of adaptive filters based on the mMSE criterion and using Indiana test image are presented in the following figures:

Fig. 2.3 - LMS approach, steepest descent method

Fig. 2.4 - Gradient approach, steepest descent method

Fig. 2.5 - Gradient approach, accelerated steepest descent method

Fig. 2.6 - Gradient approach, Amir's method

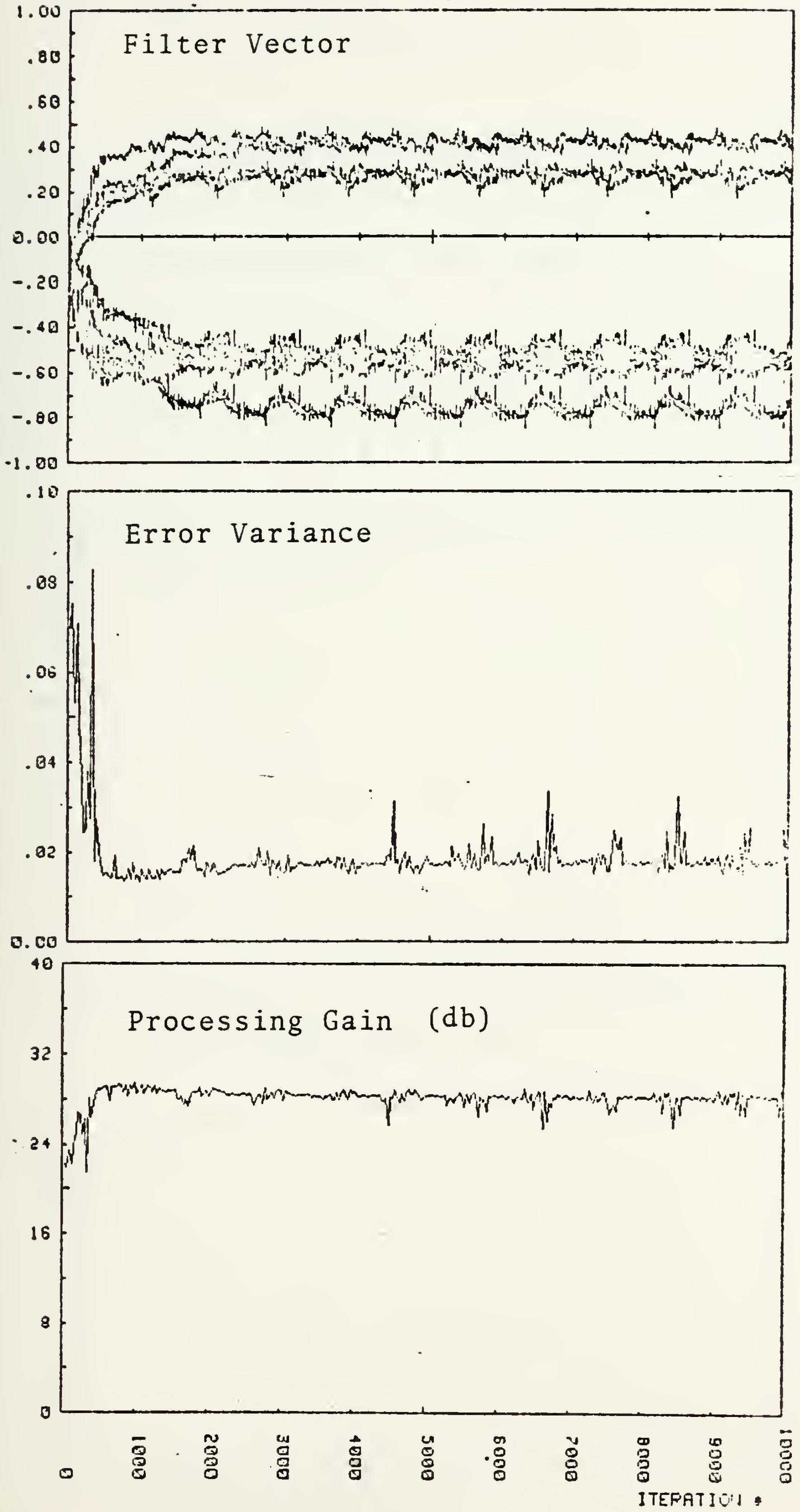
Fig. 2.7 - Conjugate gradient approach, Fletcher-Reeves method

Fig. 2.8 - Conjugate gradient approach, Pollack method

Fig. 2.9 - Variable metric approach, Davidon-Fletcher-Powell method.

In each figure three results - the nine filter coefficients, output variance and processing gain - are presented as a function of iteration steps.

LMS Algorithm



Filter Vector

Fig. 2.3a

Error Variance

Fig. 2.3b

Processing Gain (db)

Fig. 2.3c

Steepest Descent Method - mMSE

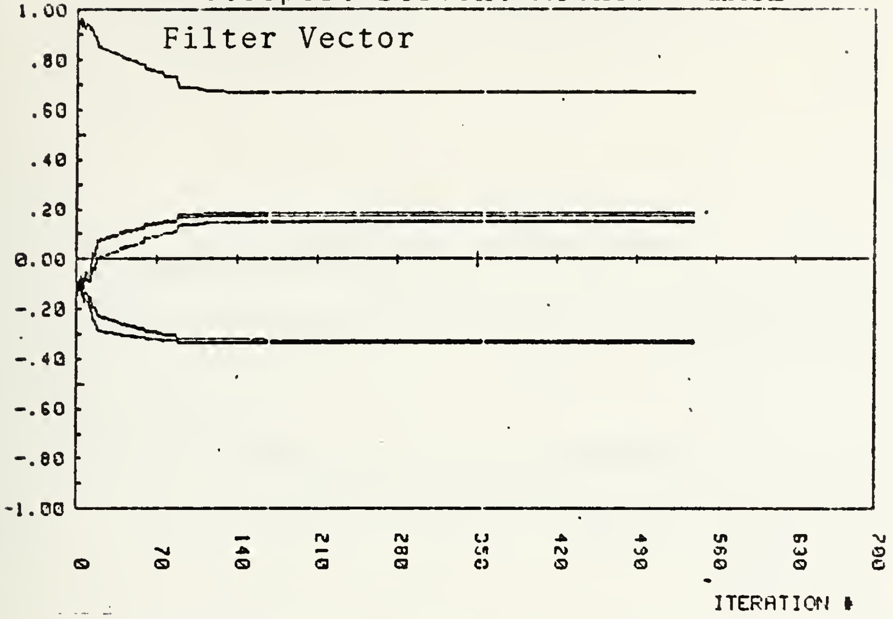


Fig. 2.4a

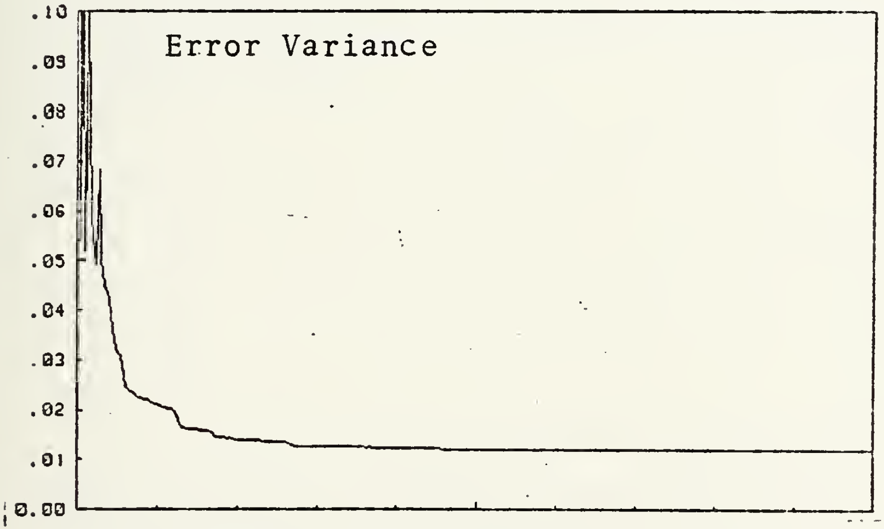


Fig. 2.4b

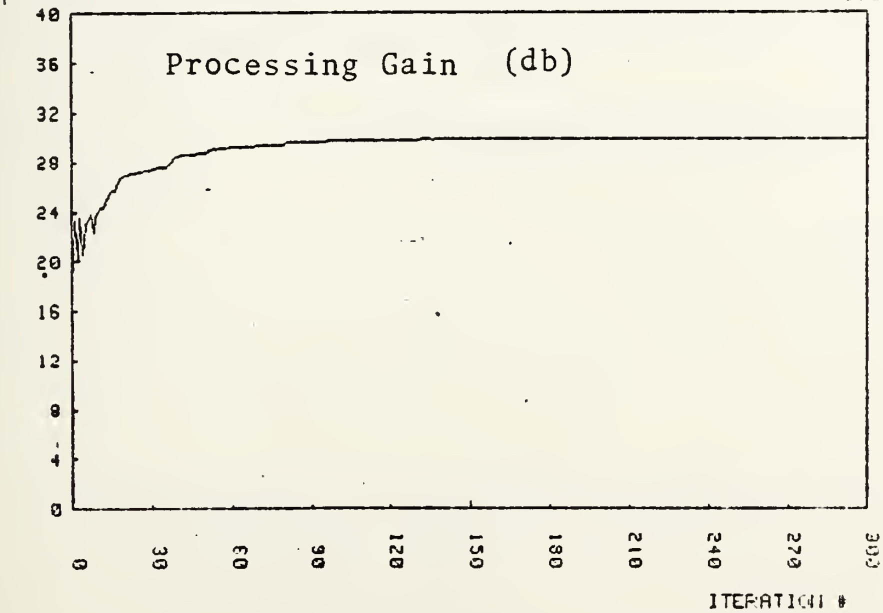


Fig. 2.4c

Accelerated Steepest Descent - mMSE

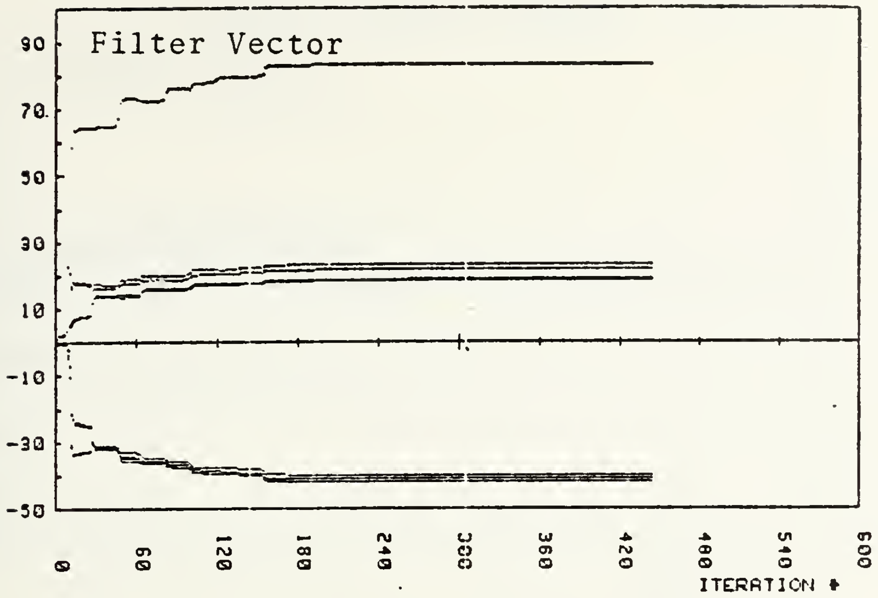


Fig. 2.5a

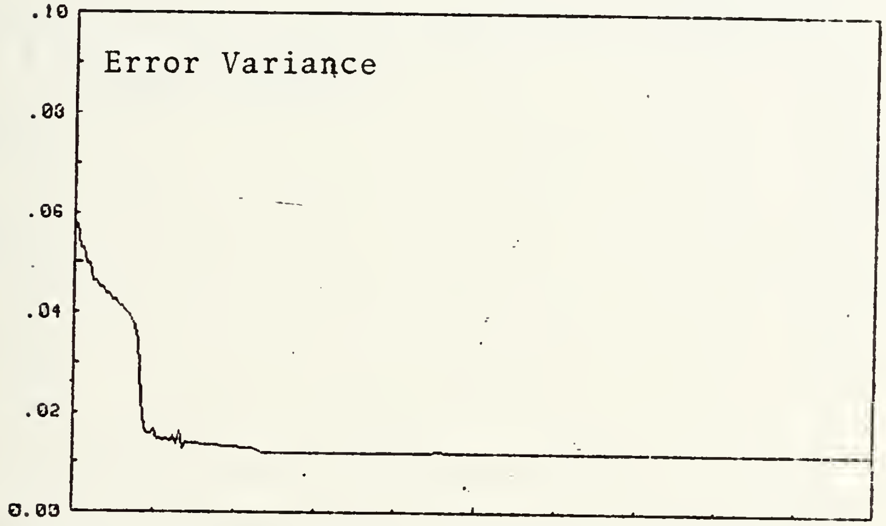


Fig. 2.5b

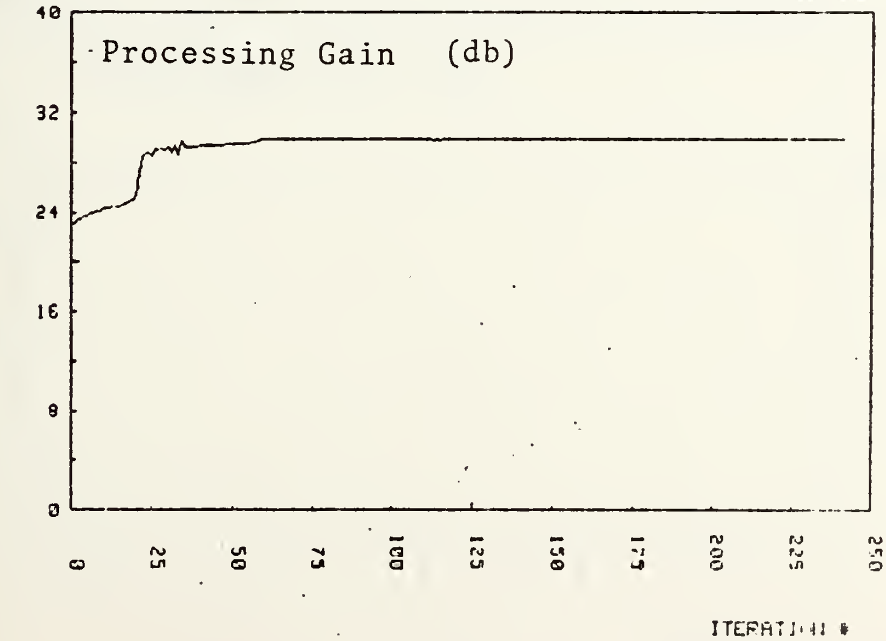


Fig. 2.5c

Amir's Method - mMSE

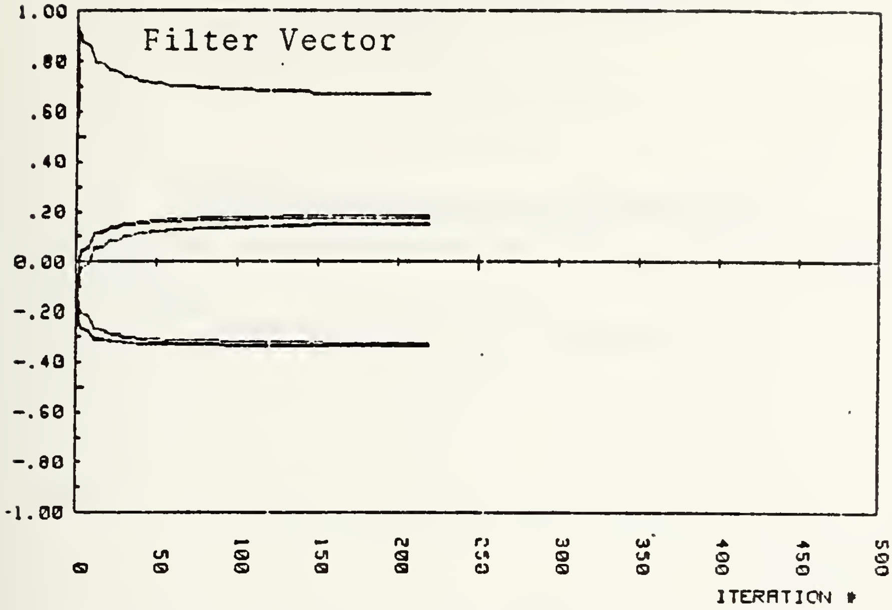


Fig. 2.6a

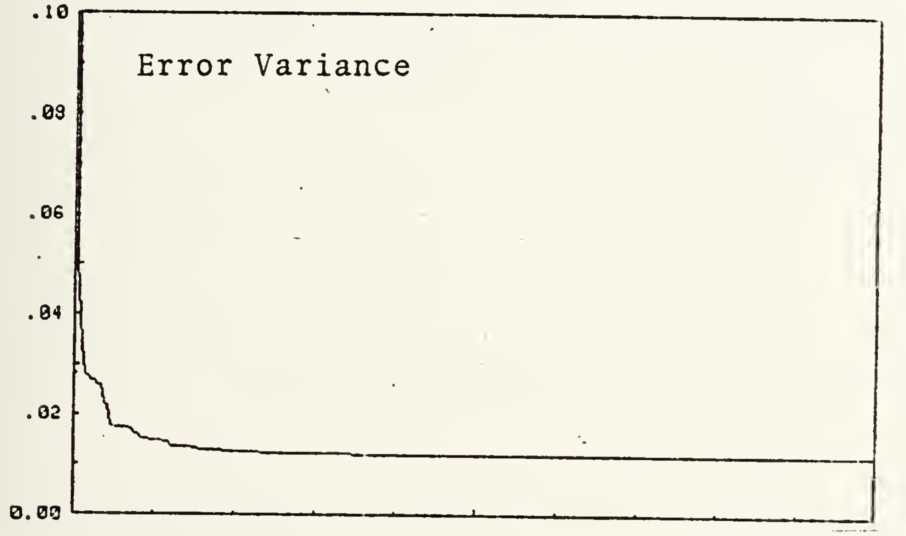


Fig. 2.6b

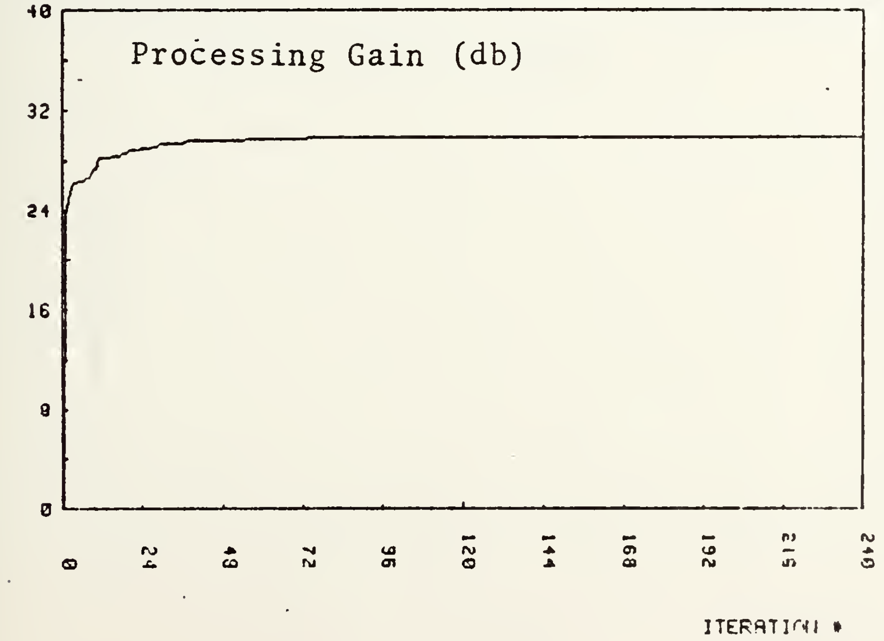


Fig. 2.6c

Fletcher-Reeves Method - mMSE

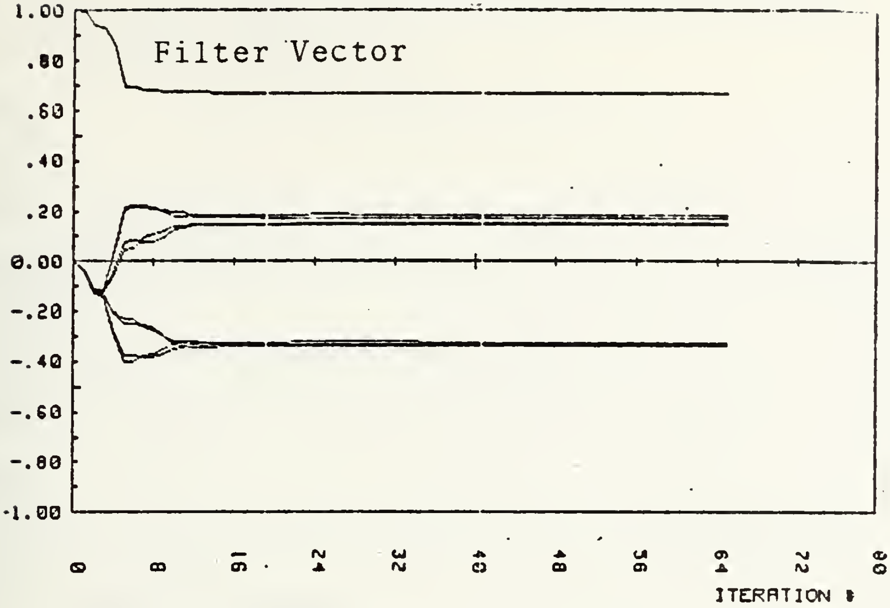


Fig. 2.7a

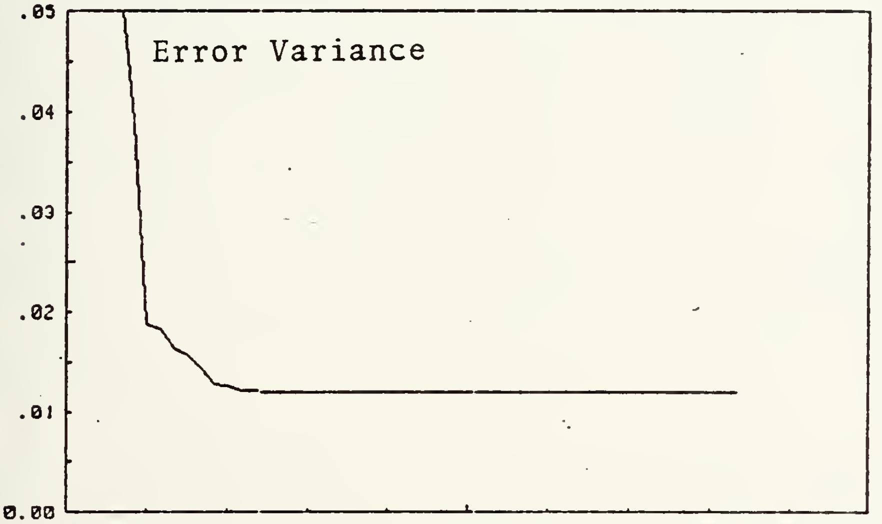


Fig. 2.7b

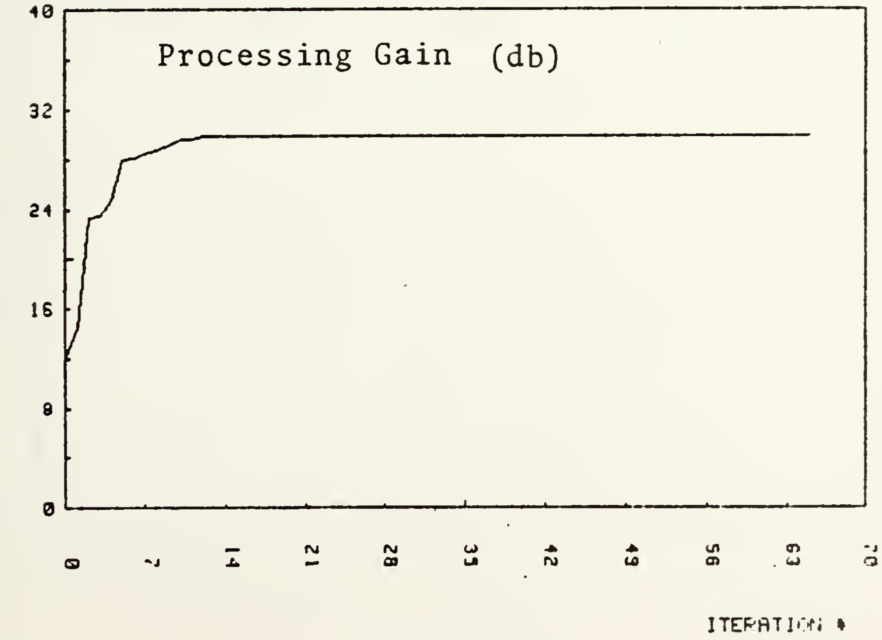


Fig. 2.7c

Pollack Method - mMSE

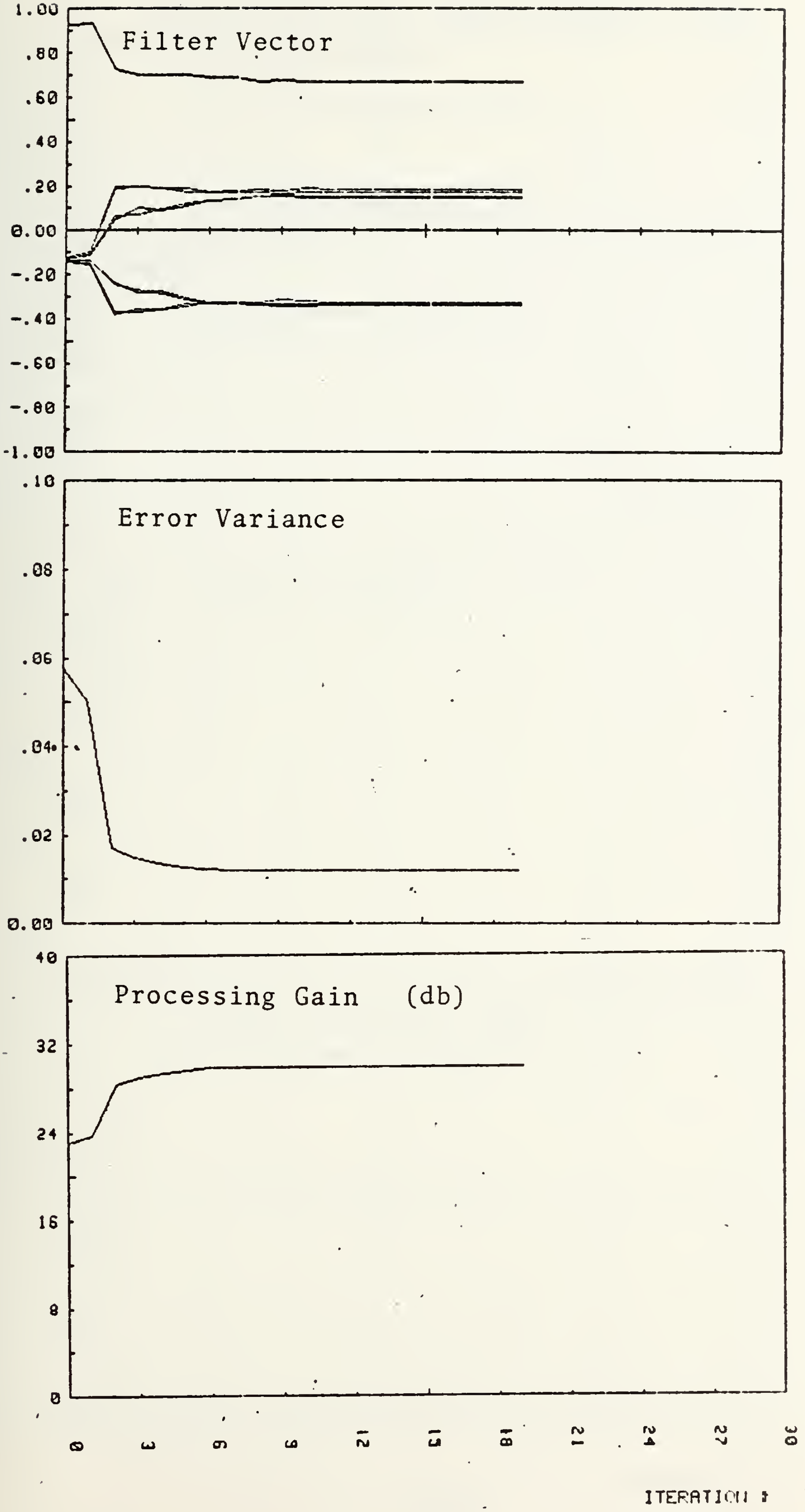


Fig. 2.8a

Fig. 2.8b

Fig. 2.8c

Davidon-Fletcher-Powell Method - mMSE

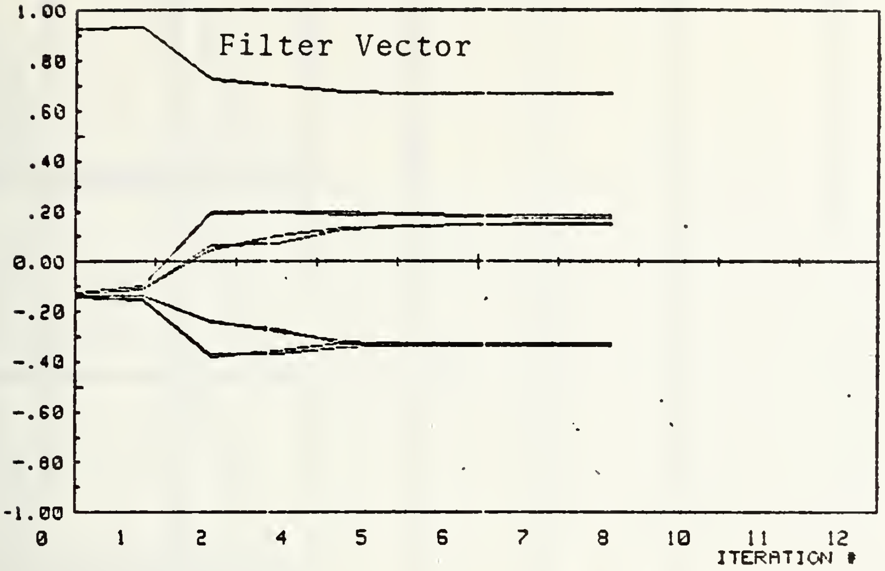


Fig. 2.9a

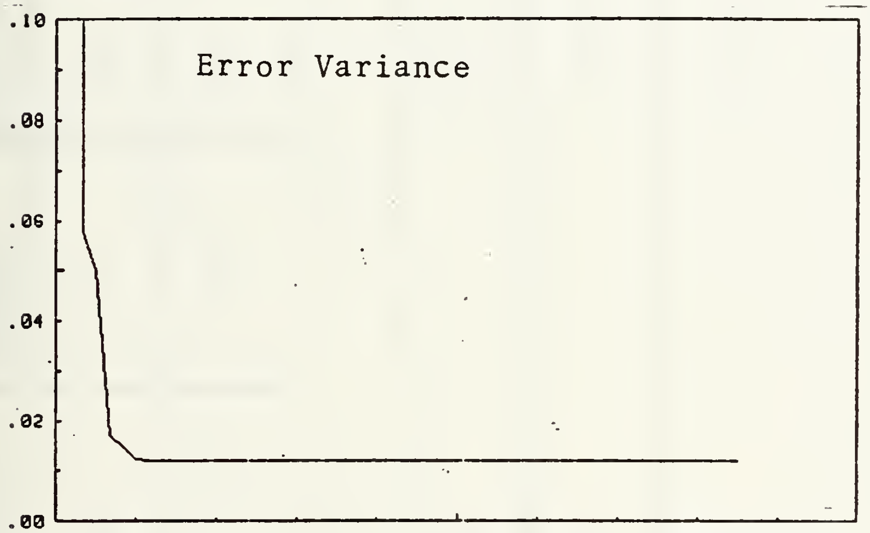


Fig. 2.9b

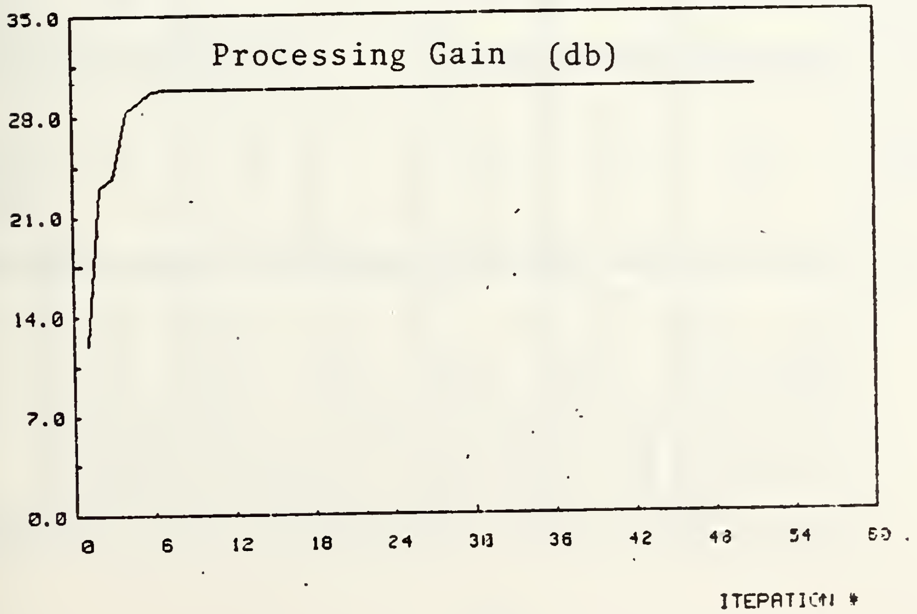


Fig. 2.9c

Table [illegible] Results of mMSE Adaptive Spatial Filter (Indiana Image)

	Least Mean Square	Steepest Descent	Accelerated Steepest Descent	Amir's Method	Fletcher Reeves	Pollack	Davison Flectcher Powell
Number of Adaptation Steps	17795	541	445	220	66	10	9
Adaptation Error At Stop	$8.64x10^{-16}$	$4.7x10^{-11}$	$-1.24x10^{-10}$	$9.58x10^{-11}$	$7.24x10^{-11}$	$1.65x10^{-11}$	$4.6x10^{-19}$
Stopping Error	$1.5x10^{-11}$						
Mean of Filtered Image	$-1.28x10^{-3}$	$6.495x10^{-5}$*					
Variance of Filtered Image	$1.76x10^{-2}$	$1.2x10^{-2}$*					
Processing Gain of Adaptive Filter(db)	28.213	29.874 *					

Test Image: Blue Spike Indiana Infrared Image
32x32 pixels
Before Filtering: Mean = 3.3039 , Variance = 0.74111

*These results are exactly the same as that of the optimum mMSE filter.

The following additional numerical results are presented in Table II-1:

- Processing gain
- Mean of the filtered image
- Variance of the filtered image
- Number of iteration steps to go below the prescribed error
- Actual adaptation error when the adaptation is stopped.

a. Discussion

These results will be discussed in several groups.

(1) LMS Approach and Steepest Descent Method. This approach is the two dimensional extension of the most widely used adaptive filter technique. In Fig. 2.3, we can see that as the adaptation took close to one thousand steps to reach the minimum of the output variance and the maximum of the processing gain. However, the adaptation never achieved a steady state, even up to 10,000 steps of iteration.

Further, there is a steady state deviation from the optimum output variance. It is known as the "misadjustment" which commonly occurs in the traditional adaptive filter approach (23).

We believe these problems are the consequences of the basic assumptions of this LMS algorithm. The reasons probably are not obvious if we follow the traditional adaptive concept which was initiated by Prof. Widrow using the error signal concept in control, $\varepsilon = H^T X - d$. The filtered output

H^TX is compared with a desirable result, d. Their difference, ε, is used together with a constant, but adjustable, adaptive gain, 2μ, to form a correction term, $\Delta \underline{H}$, for the filter coefficients to approach the optimization goal, which is the minimization of mean square error.

On the other hand, if we consider the adaptation procedure as an optimization process, then, the adaptation equation takes the form of

$$\underline{H}_{K+1} = \underline{H}_K + \Delta \underline{H}_K = \underline{H}_K + \alpha_K \, \underline{G}_K \quad .$$

where $\underline{G}_K$ is called the "gradient," α_K is called the "step size." The concept of gradient means the gradient of the performance function surface, J. The product of adaptation step size α_K and the gradient $\underline{G}_K$ is the correction term $\Delta \underline{H}$.

It is postulated that the assumptions made by the LMS approach are not directly responsive to the goal of adaptation because the error term H^TX - d is not directly related to the minimization of the performance function. Further, the assumption that the adaptive gain 2μ, which corresponds to the concept of step size in optimization, is constant, does not coincide with the fact that the iterative steps toward optimization usually take place in varying step sizes. These problems contributed to the slow convergence, and the steady state misadjustment in the LSM adaptive spatial filters.

We developed several adaptive filters using gradient methods developed in the optimization field. Their results are discussed in the following.

(2) Gradient Approaches. Three different methods were developed. Their results are shown in Figures 2.4, 2.5, and 2.6 for the steepest descent (SD), accelerated steepest descent (ASD) and Amir's (AMM) methods respectively.

The reasoning described above is quite convincingly supported by the following observations:

a. The convergence of adaptation is faster. It took 541, 445, and 220 steps for the SD, ASD and AMM methods to reach the stopping condition of adaptation less than 1.5 x 10^{-11} as shown in Table II-1.

b. The adaptation procedure indeed reached steady state once the adaptation error is less than the stopping condition.

c. The steady state error is smaller than that of the LMS algorithm as shown in Table II-1. In fact, the output variance is equal to that of the optimum filter.

(3) Conjugate Gradient Approaches. Two different methods were developed. Their results are shown in Figures 2.7 and 2.8 for the Fletcher-Reeves (CGF) and the Pollack (CGP) methods respectively.

Again, the improvements are clearly seen. In fact, they are even better than the gradient methods. The convergence took only 66 and 10 steps for CGF and CGP methods to reach below the stopping condition of 1.5 x 10^{-11}. At the same time, the output variance is the same.

(4) Variable Metric Approach. Results of this approach, which is extended from the one dimensional work of

Davidon-Fletcher-Powell are shown in Fig. 2.9. Again, the improvements are clearly seen. The background suppression result is the same measured by the output variance and processing gain. But the convergence speed is even better and took only 9 iteration steps to reach below the stopping condition.

3. Results of mMSE Adaptive Spatial Filter II - China Lake Images

The test results of adaptive filters based on the mMSE criterion and using the China Lake test image are presented in the following figures:

Fig. 2.10 - LMS approach, steepest descent method

Fig. 2.11 - Gradient approach, steepest descent method

Fig. 2.12 - Gradient approach, accelerated steepest descent method

Fig. 2.13 - Gradient approach, Amir's method

Fig. 2.14 - Conjugate gradient approach, Fletcher-Reeves method

Fig. 2.15 - Conjugate gradient approach, Pollack method

Fig. 2.16 - Variable metric approach, Davidon-Fletcher-Powell method

In each figure, three results are presented as functions of iteration steps: filter coefficients, output variance and processing gain.

Further, additional results are summarized and presented in Table II-2:

- Processing gain
- Mean of the filtered image

- Variance of the filtered image
- Number of iteration steps to go below the prescribed stopping error
- Actual adaptation error when the adaptation is stopped.

a. Discussion

The results, using the China Lake image, are generally similar to that using the Indiana image. Only the important features will be summarized below.

(1) LMS Approaches. The adaptation based on the LMS approach again show three problems: slow convergence, never reached steady state, and misadjustment.

(2) New Approaches Developed in this Thesis. All new approaches achieve the same steady state performance equal to that of the optimum filter as shown in Table II.2:

Mean of the filtered image.......... = $6.495 \cdot 10^{-5}$
Variance of the filtered image...... = $1.2 \cdot 10^{-2}$

However, they converge to the steady state value with much less numbers of steps, as shown in Table II.2 also.

Therefore, test results on the China Lake image again demonstrated the improvements in adaptive filters using the approaches suggested in this thesis.

It is interesting to note that the effectiveness of background clutter suppression in the case of the China Lake image are not as good as that in the case of the Indiana image. For example, the processing gain for

the China Lake image is (19.32) db compared with (29.874) db for the Indiana image. We believe this difference is related to the spatial correlation of the image. The higher the correlation, the better is the background clutter suppression. The Indiana image is more spatially correlated than the China Lake image.

LMS Algorithm

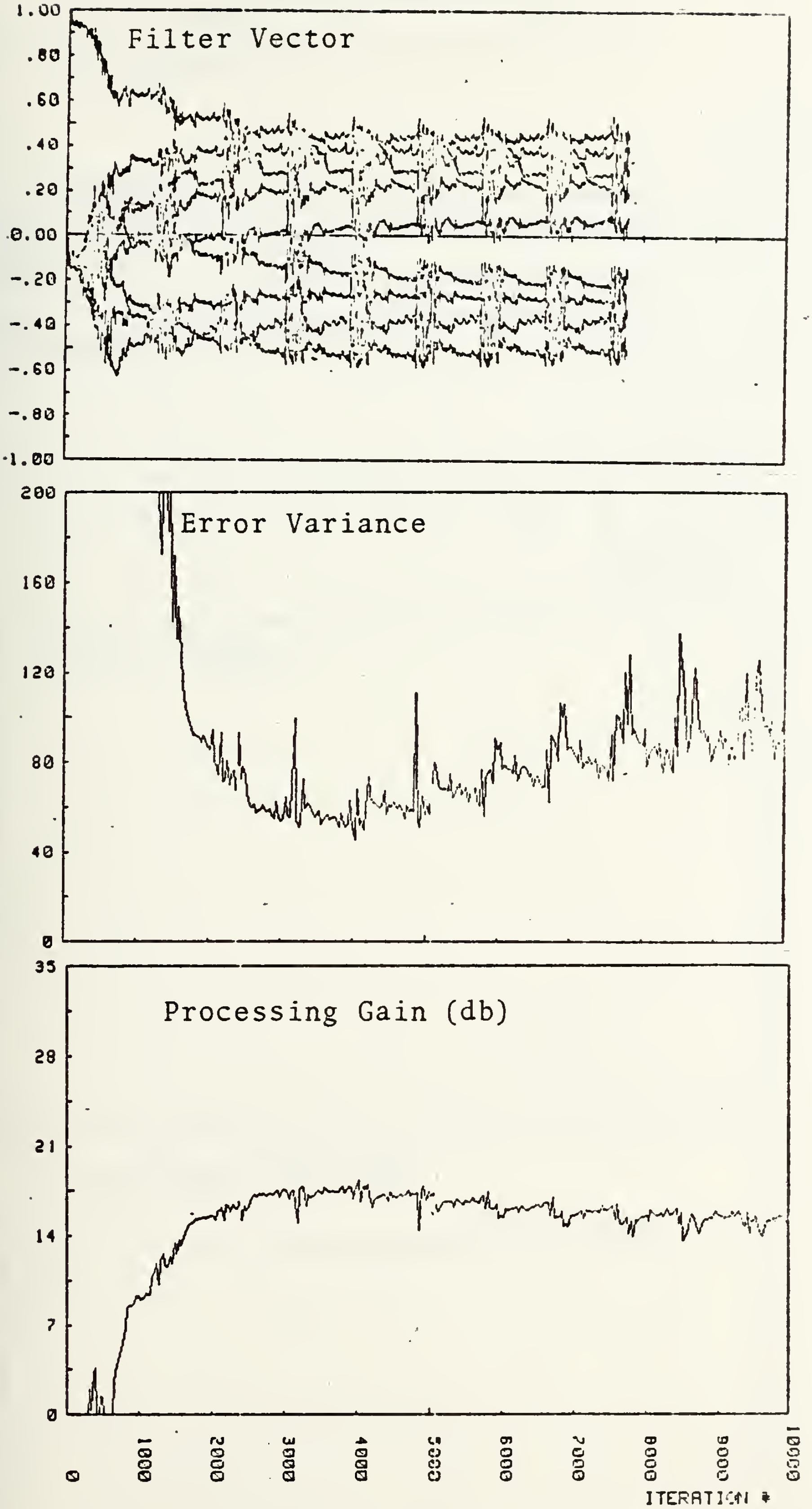


Fig. 2.10a

Fig. 2.10b

Fig. 2.10c

Steepest Descent - mMSE

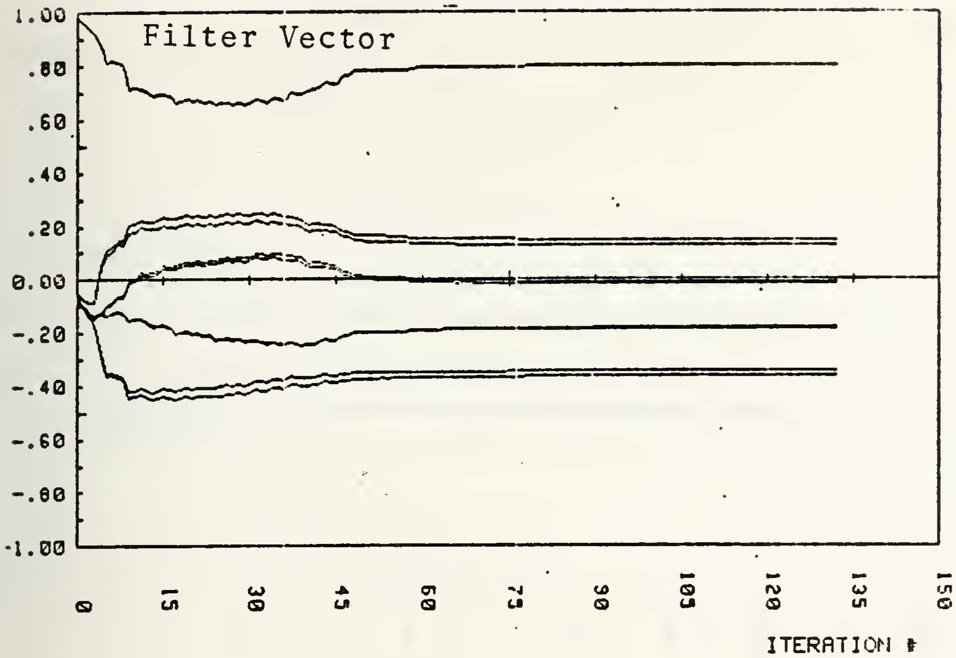


Fig. 2.11a

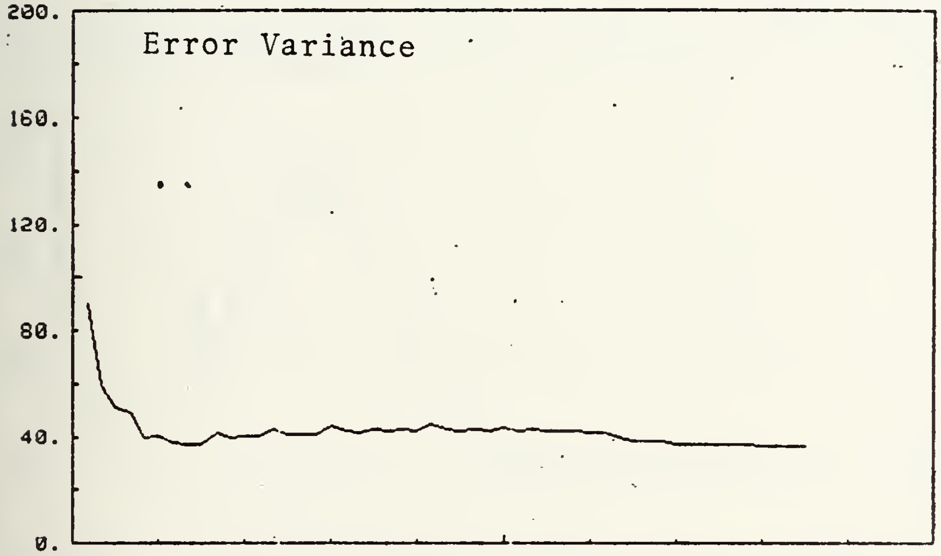


Fig. 2.11b

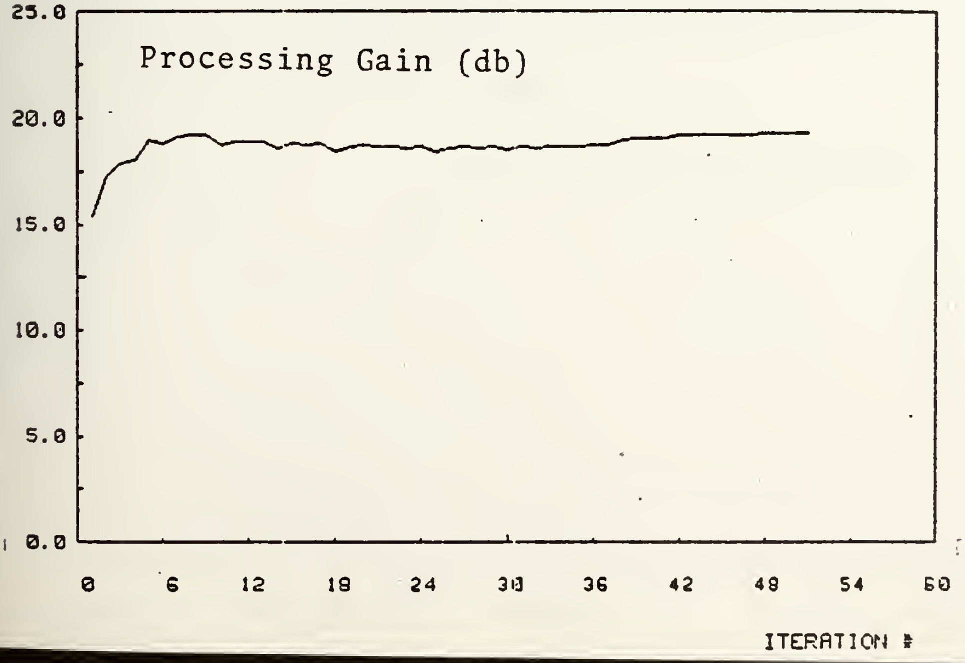


Fig. 2.11c

Accelerated Steepest Descent - mMSE

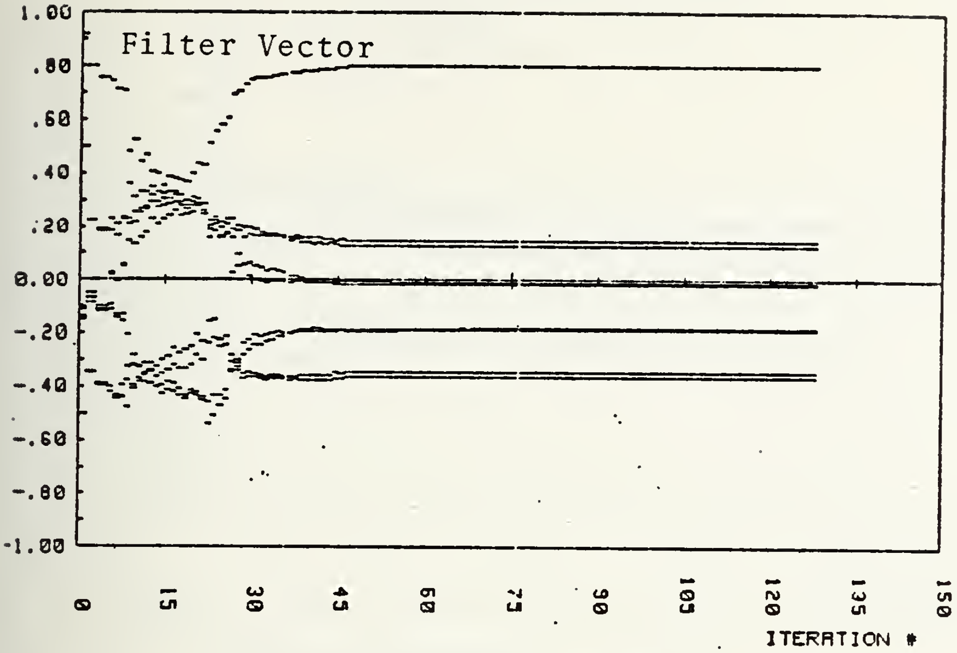


Fig. 2.12a

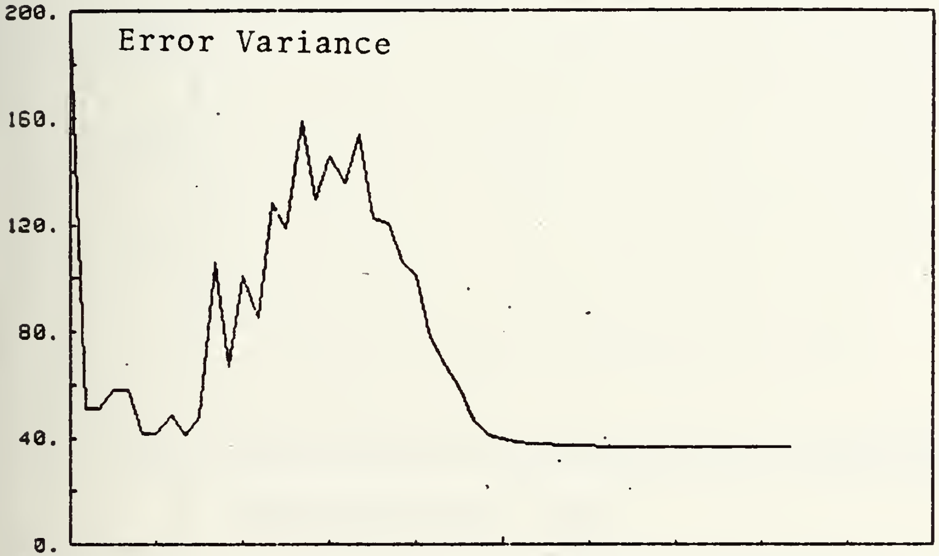


Fig. 2.12b

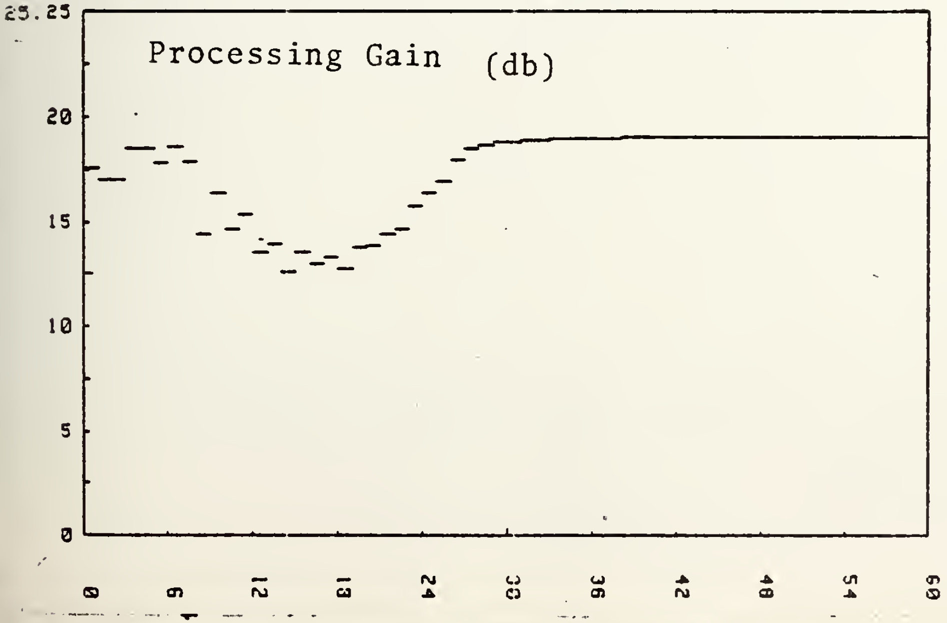


Fig. 2.12c

Amir's Method - mMSE

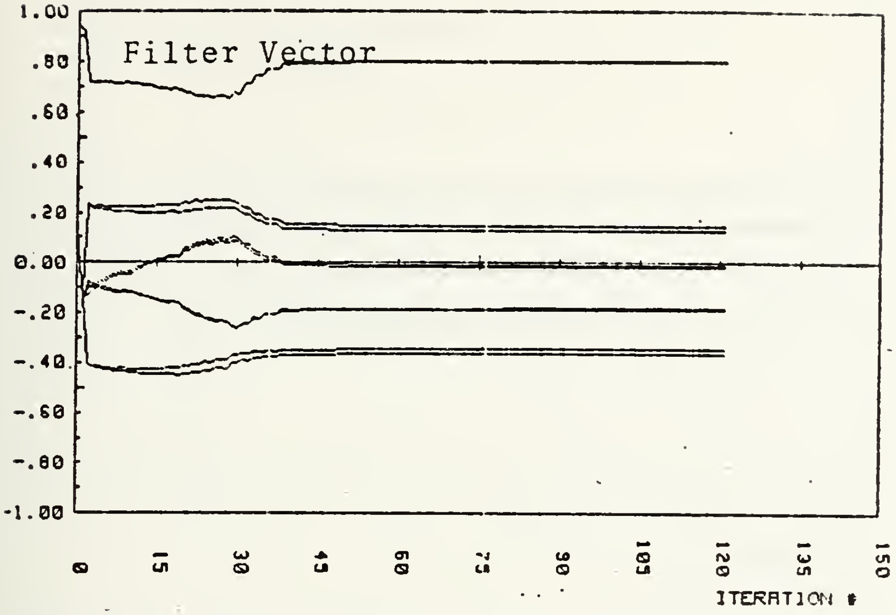


Fig. 2.13a

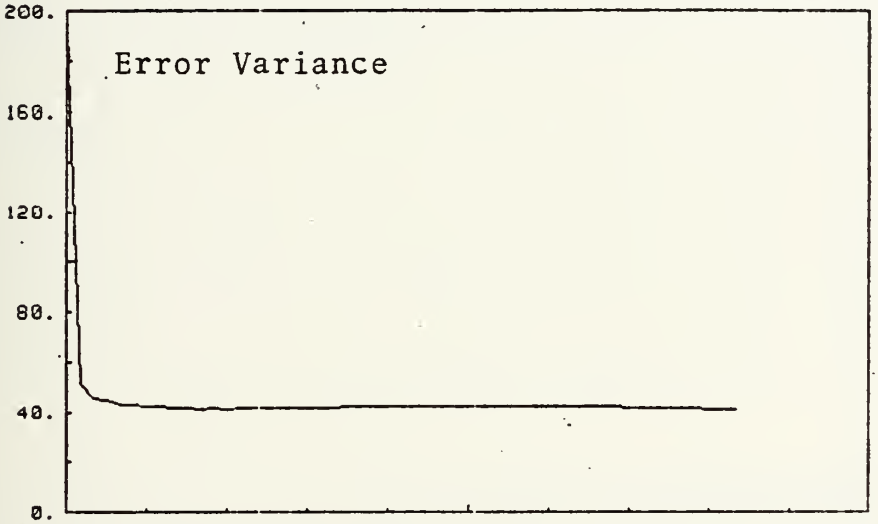


Fig. 2.13b

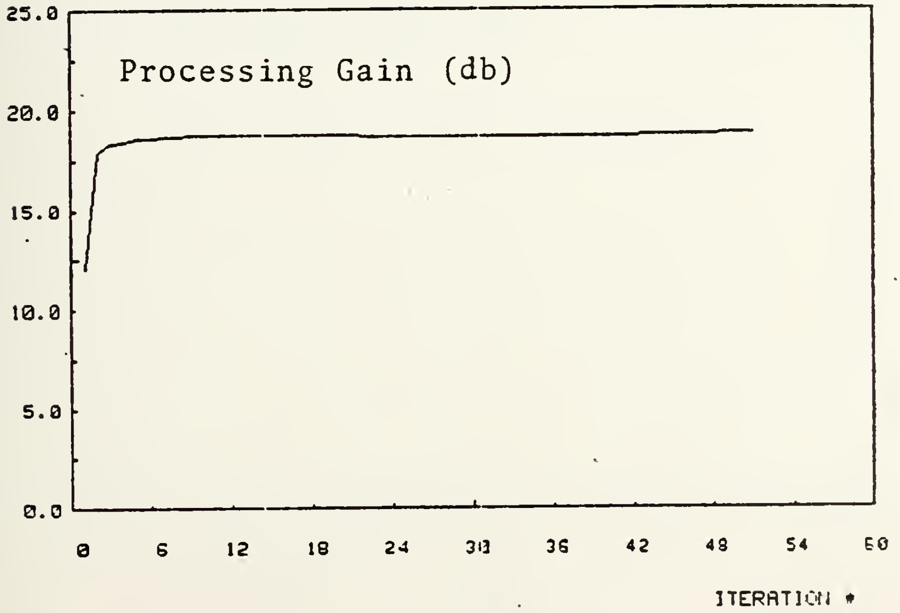


Fig. 2.13c

Fletcher-Reeves Method - mMSE

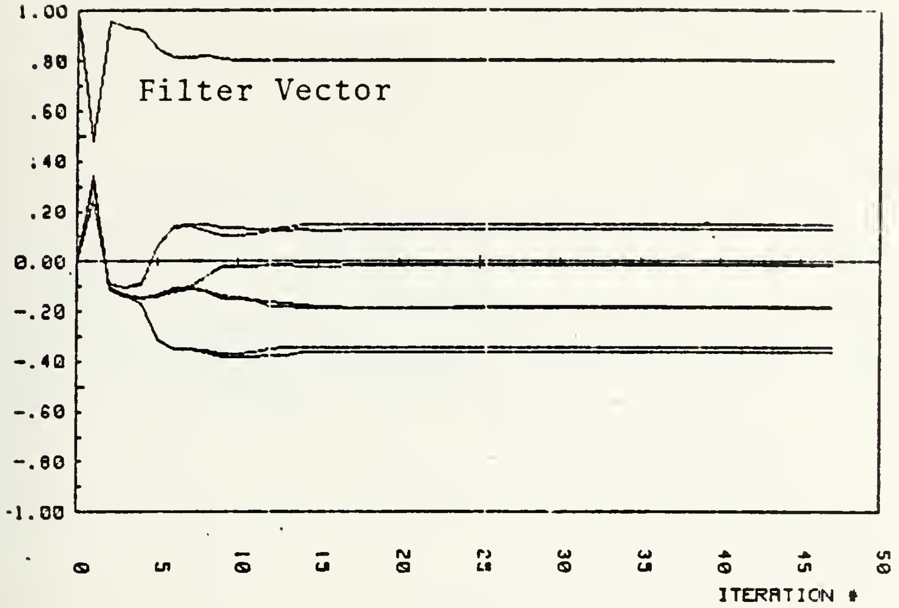


Fig. 2.14a

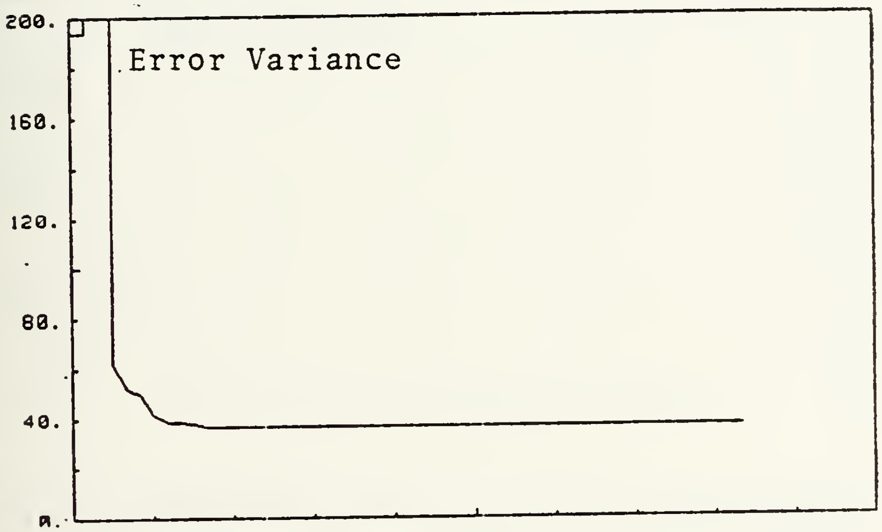


Fig. 2.14b

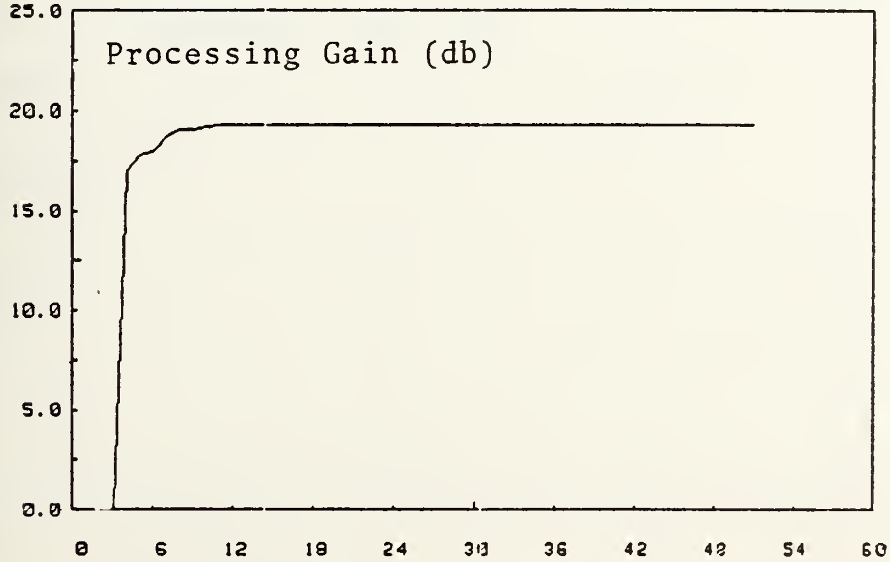


Fig. 2.14c

Pollack Method - mMSE

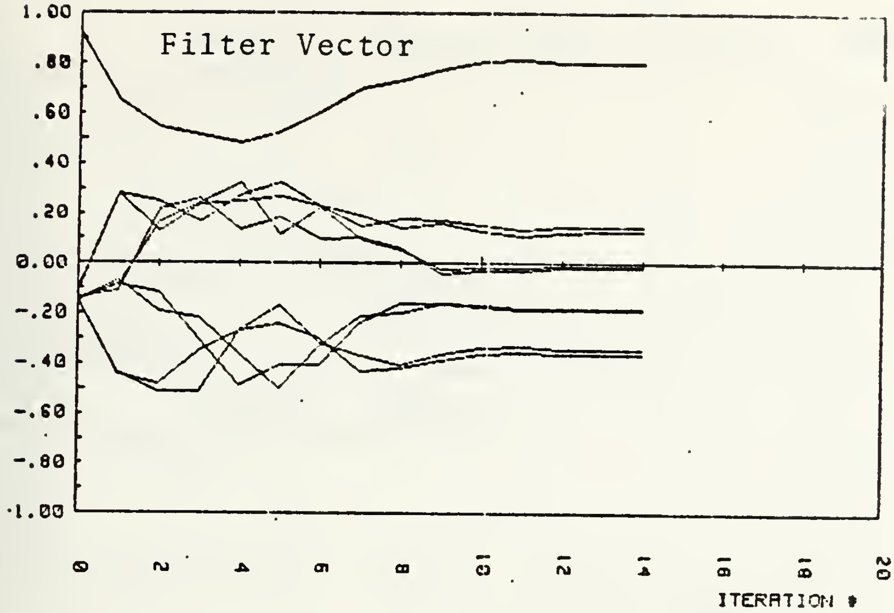


Fig. 2.15a

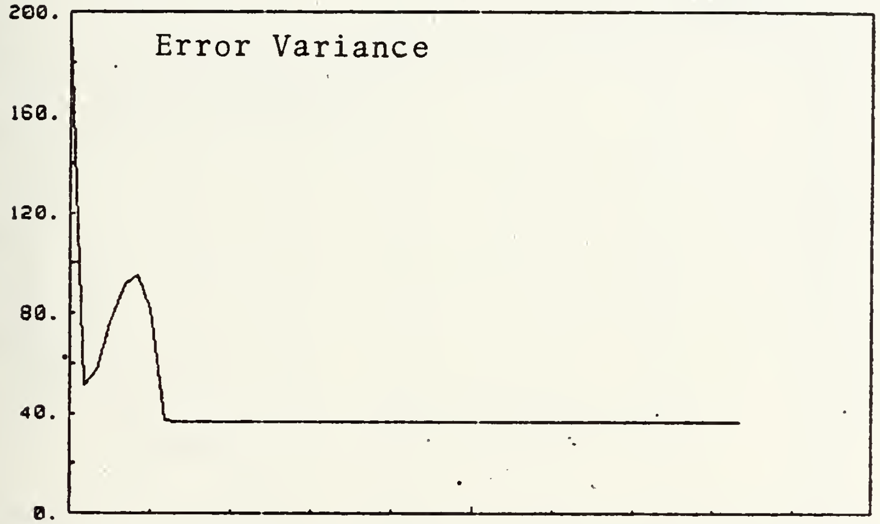


Fig. 2.15b

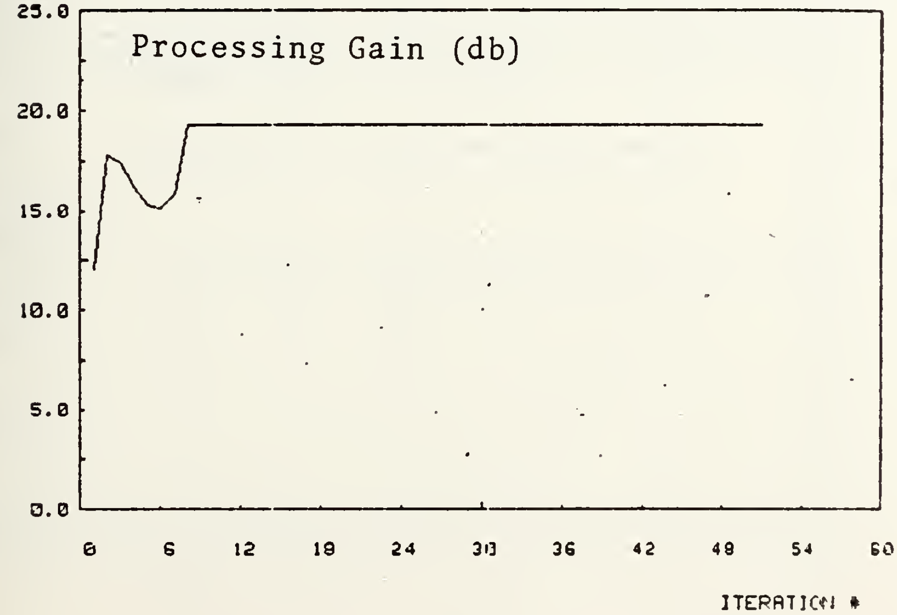


Fig. 2.15c

Davidon-Fletcher-Powell Method - mMSE

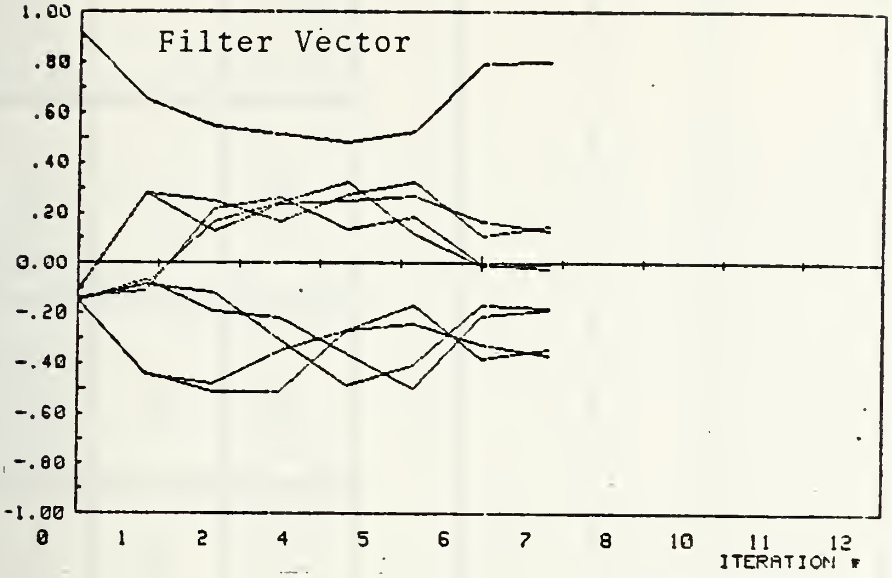


Fig. 2.16a

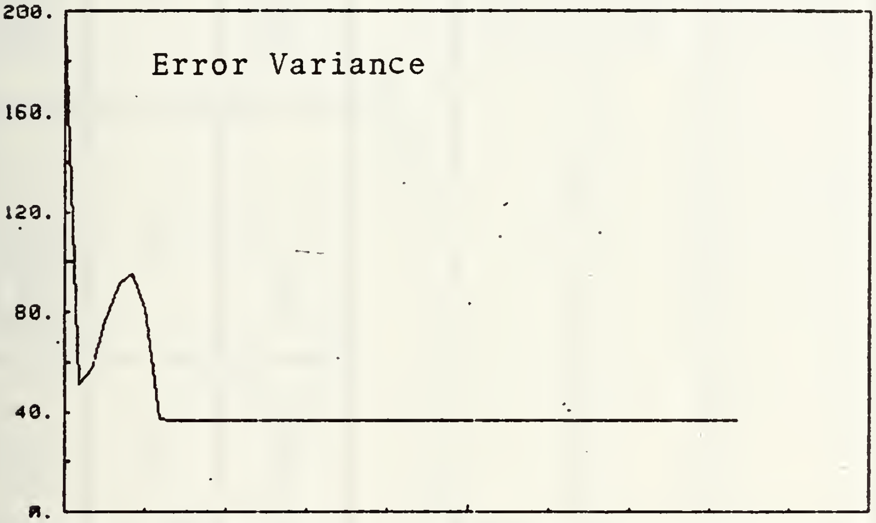


Fig. 2.16b

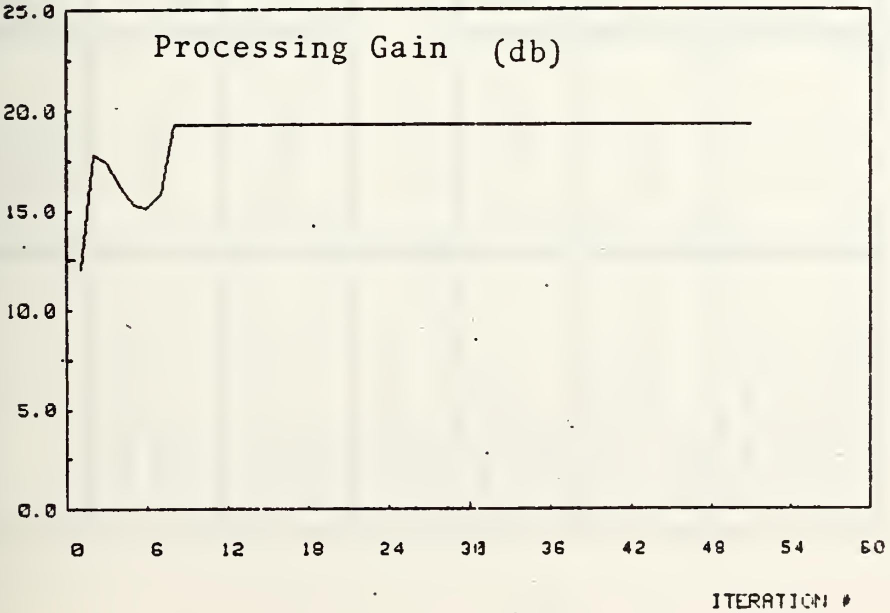


Fig. 2.16c

Table II.2 Results of mMSE Adaptive Spatial Filter (China Lake Image)

	Least Mean Square	Steepest Descent	Accelerated Steepest Descent	Amir's Method	Fletcher Reeve's	Pollack	Davison Fletcher Powell
Number of Adaptation Steps	7780	133	129	122	48	15	8
Adaptation Error At Stop	$-3.18x10^{-11}$	$5.15x10^{-11}$	$-4x10^{-14}$	$8.96x10^{-11}$	$4.8x10^{-11}$	$2.6x10^{-15}$	$1.01x10^{-17}$
Stopping Error	$1.5x10^{-11}$						
Mean of Filtered Image	$-7.31x10^{-2}$	$-3.2x10^{-2}$*					
Variance of Filtered Image	117.87	36.42*					
Processing Gain of Adaptive Filter(db)	14.22	19.32*					

Test Image: China Lake Infrared Image in the 10-13 Micron Band
32x32 pixels
Before Filtering: Mean = 54.02
Variance = 193.464

* These results are exactly the same as that of the optimum mMSE filter.

4. Results of MSNR Adaptive Spatial Filter I - Indiana Image

The test results of MSNR adaptive spatial filters using Indiana test image are presented in the following figures.

Fig. 2.17 - Gradient approach, steepest descent method

Fig. 2.18 - Gradient approach, accelerated steepest descent approach

Fig. 2.19 - Conjugate gradient approach, Fletcher-Reeves method

Fig. 2.20 - Conjugate gradient approach, Pollack method

Fig. 2.21 - Variable metric approach, Davidon-Fletcher-Powell method

Fig. 2.22 - Amir's transform approach.

In each figure, four results are presented as functions of iteration steps: filter coefficients, output variance, processing gain and output signal to noise ratio.

Further, additional numerical results are summarized and presented in Table II-3.

Output signal to noise ratio

Processing gain

Mean of filtered image

Variance of filtered image

Number of iteration steps to reach below the prescribed stopping error

Actual adaptation error.

Discussion:

a. In the mMSE adaptive filter study, we first presented the results of adaptive filter design by the LMS

algorithm because it is the most frequently used method. We extended it to two dimensions and used it as a benchmark for comparison. For the MSNR criterion, we have not yet found any past study of adaptive filter using this method. Therefore, comparisons of convergence results are based on several methods developed in this thesis study.

b. However, we can compare the background clutter suppression results - of the mMSE and MSNR adaptive filters. For point targets, their steady state filter coefficients are the same if the coefficient of the estimation pixel are all normalized to unity. Therefore, the statistical properties of the filtered image are the same, i.e., the error variance and the mean of the image after processing by the two types of filters are identical. For the Indiana image, the mean and variance of the unfiltered and filtered images are.

	Before filtering	After filtering
mean	3.30397	0.00006495
variance	0.74111	0.012

c. The convergence speeds are different, as shown in Table II.3. For a stopping condition of 10^{-15}, the numbers of iteration steps to reach below this condition are:

SD = 739	CGF = 76	DFP = 25
ASD = 739	CGP = 76	AT = 2

Fig. 2.17a Steepest Descent Method - MSNR

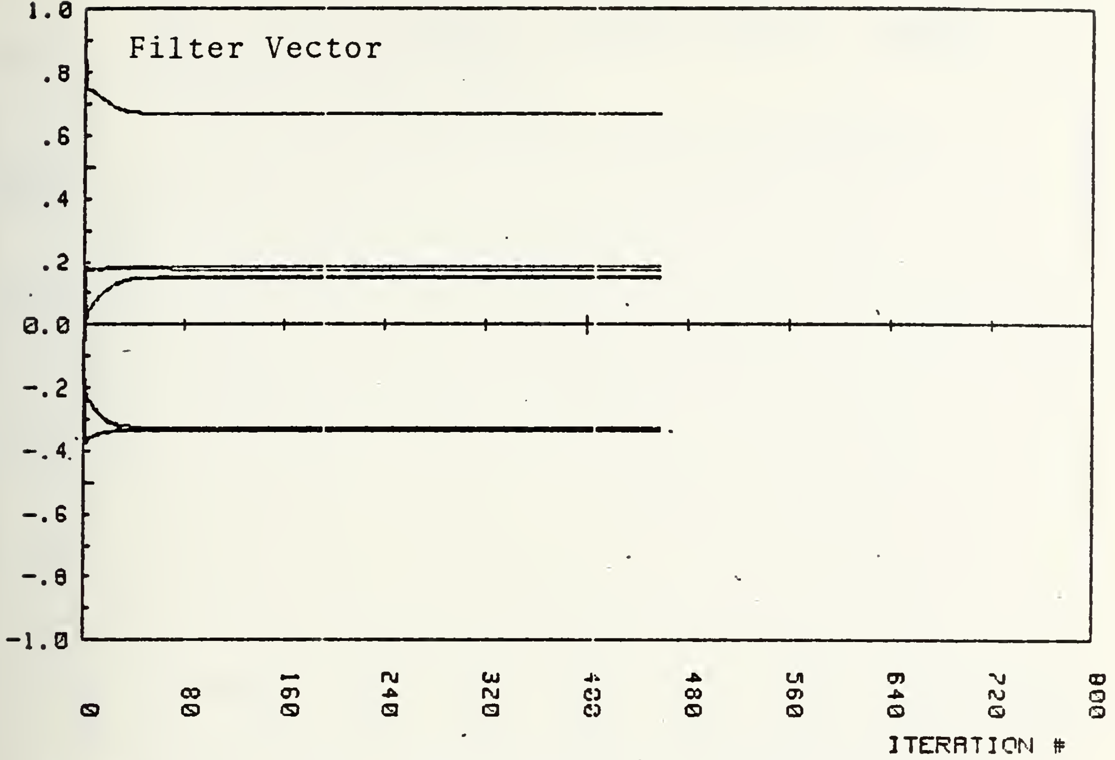


Fig. 2.17b Steepest Descent Method - MSNR

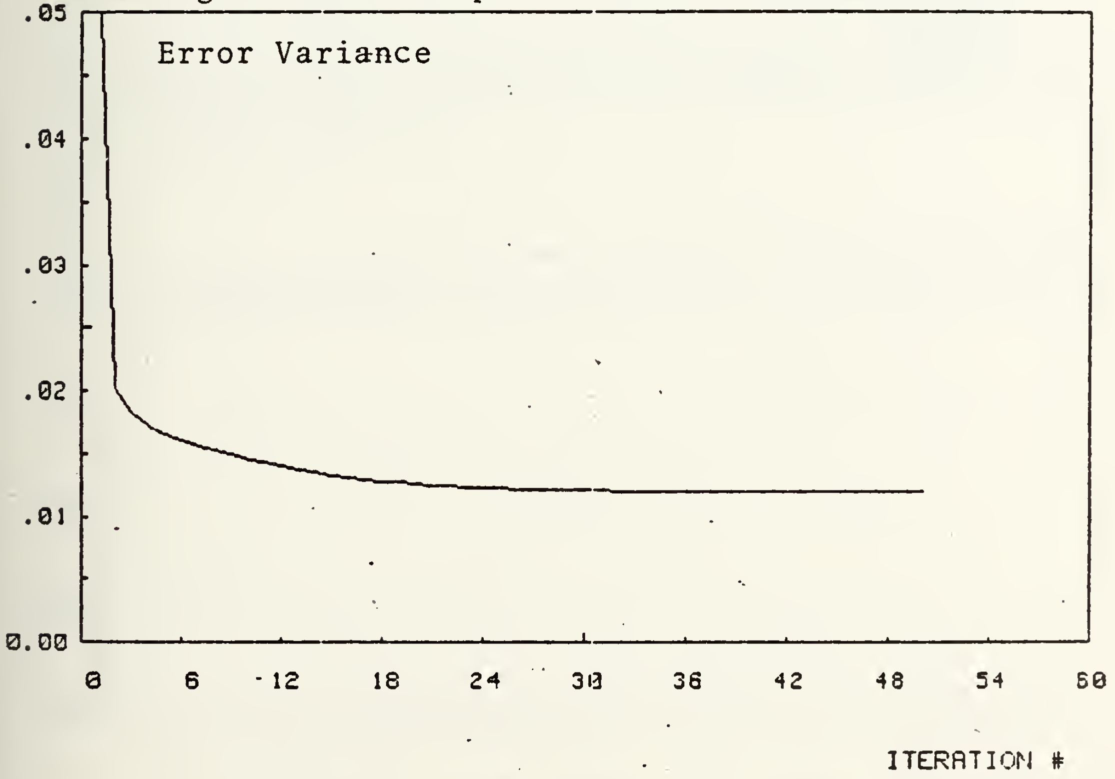


Fig. 2.17c Steepest Descent Method - MSNR

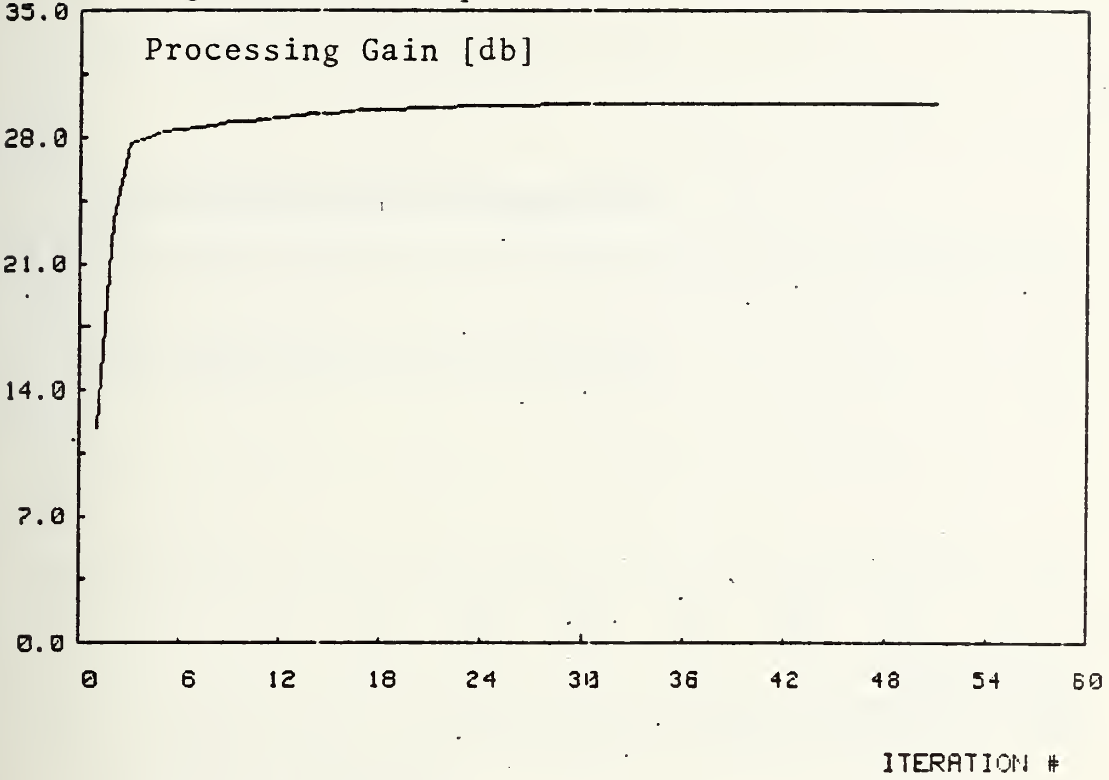


Fig. 2.17d Steepest Descent Method - MSNR

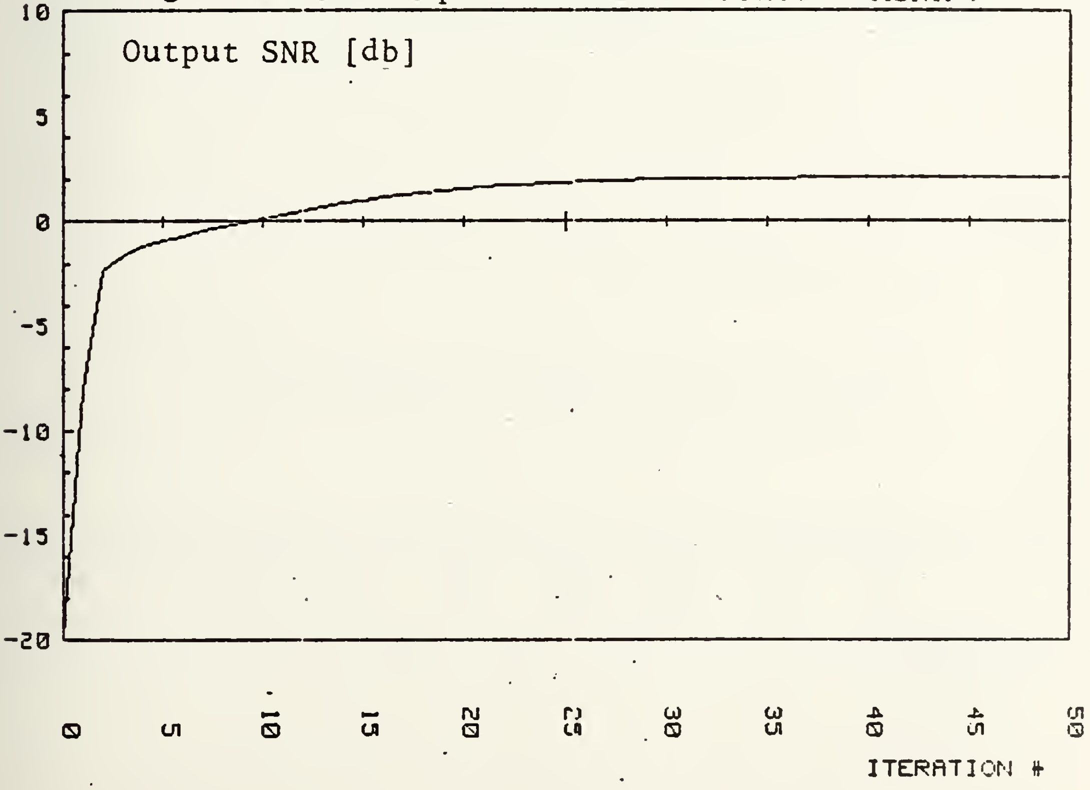


Fig. 2.18a Accelerated Steepest Descent - MSNR

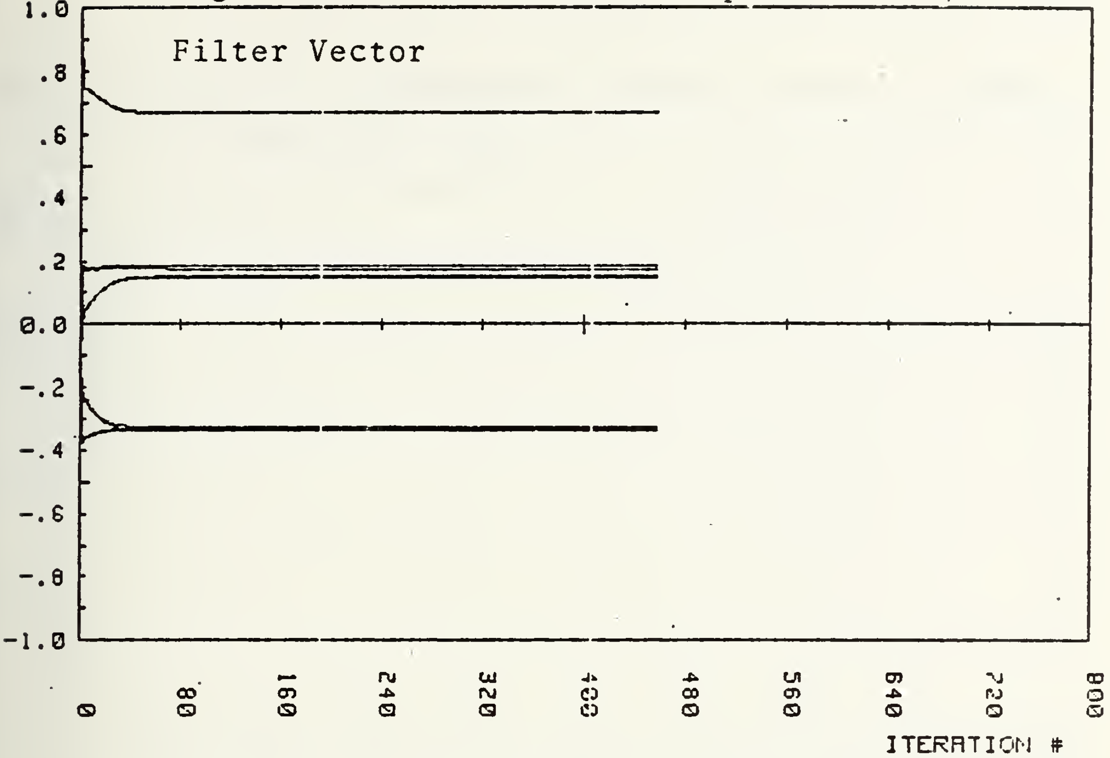


Fig. 2.18b Accelerated Steepest Descent - MSNR

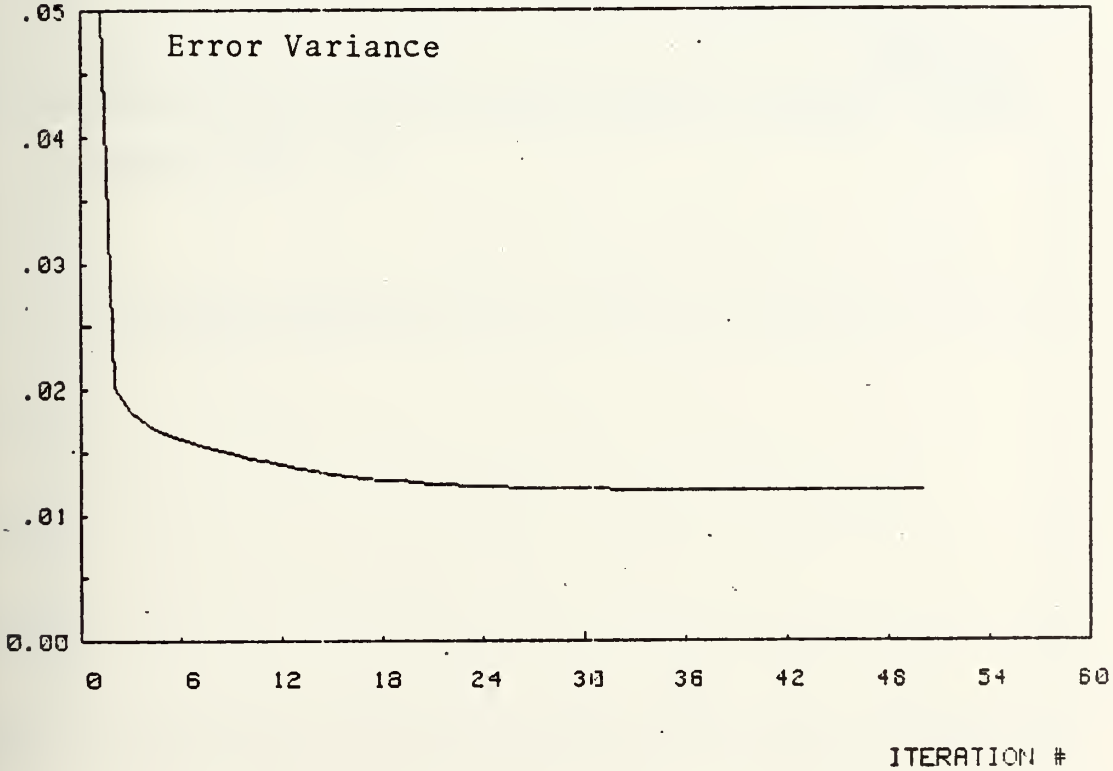


Fig. 2.18c Accelerated Steepest Descent - MSNR

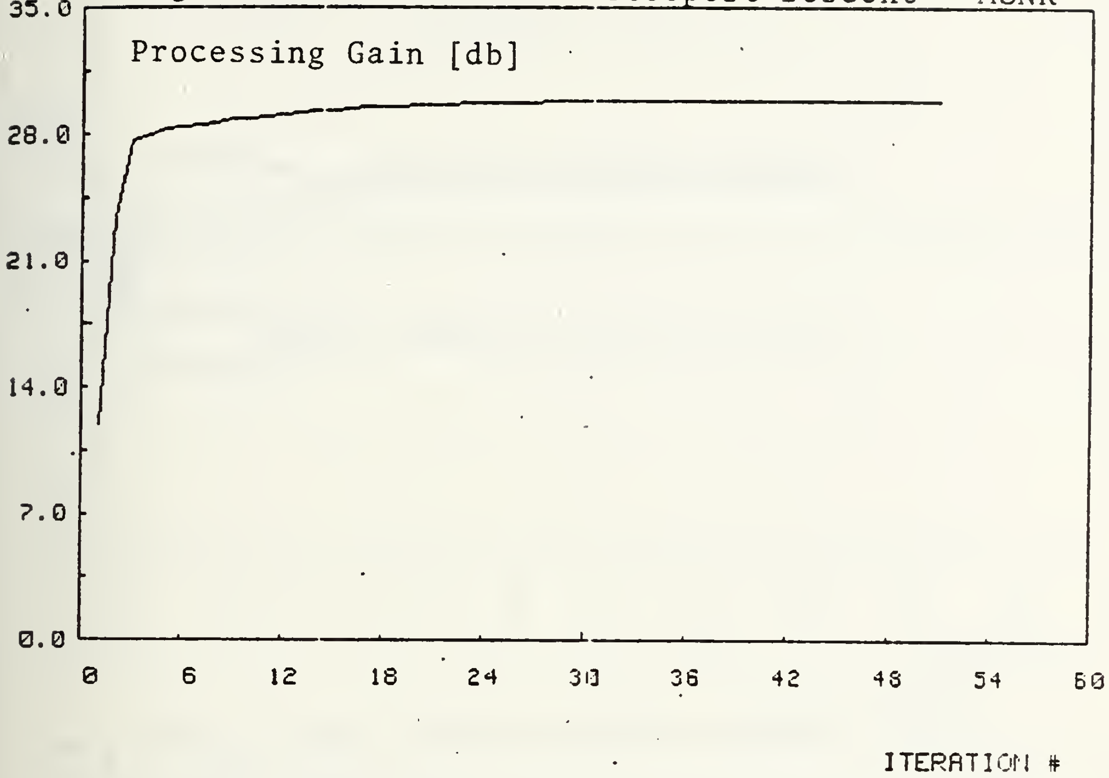


Fig. 2.18d Accelerated Steepest Descent - MSNR

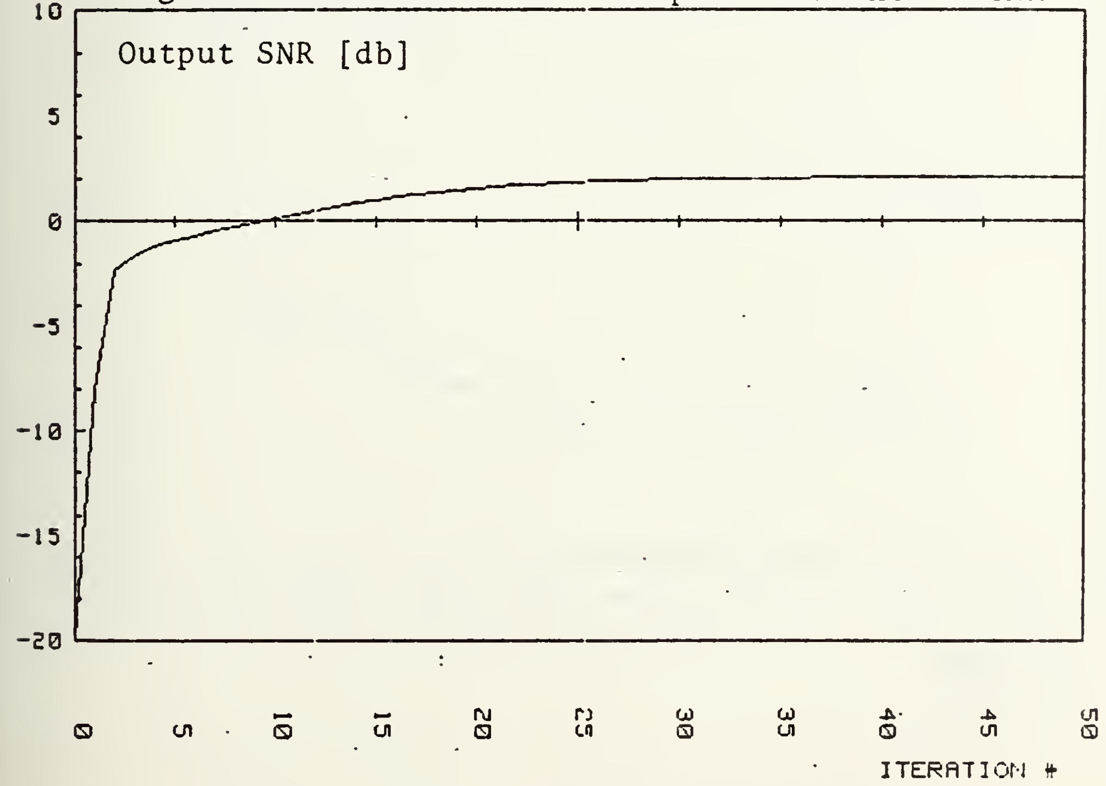


Fig. 2.19a Fletcher-Reeves Method - MSNR

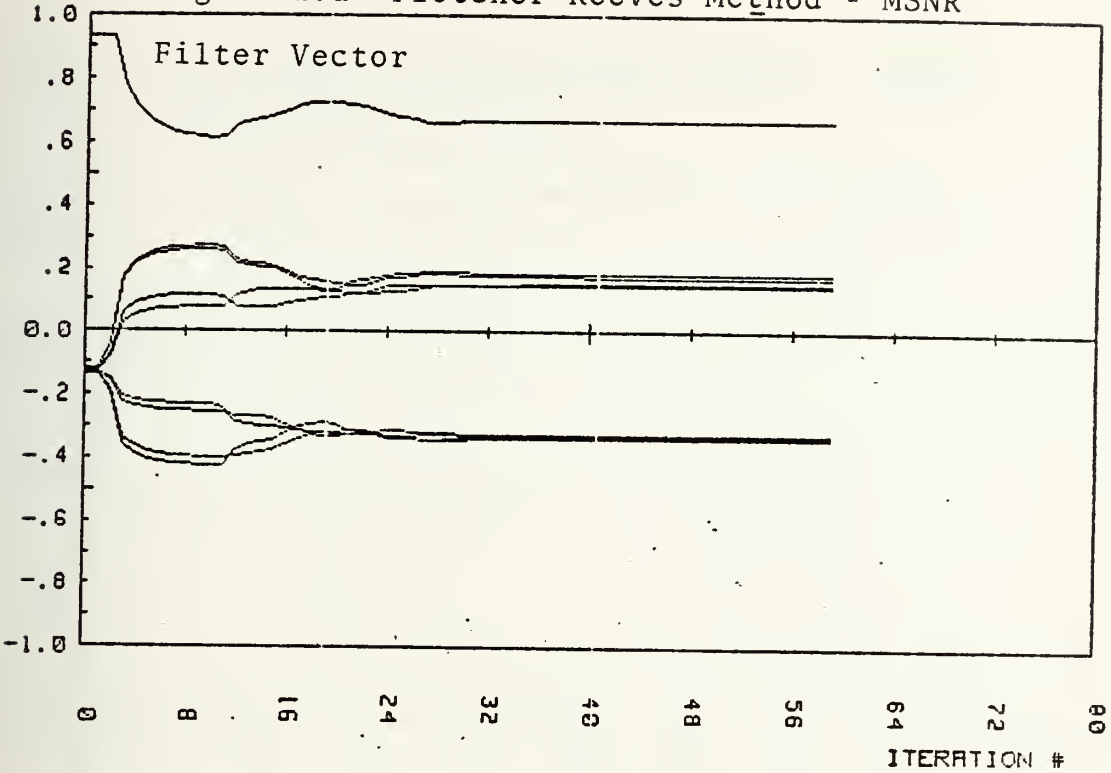


Fig. 2.19b Fletcher-Reeves Method - MSNR

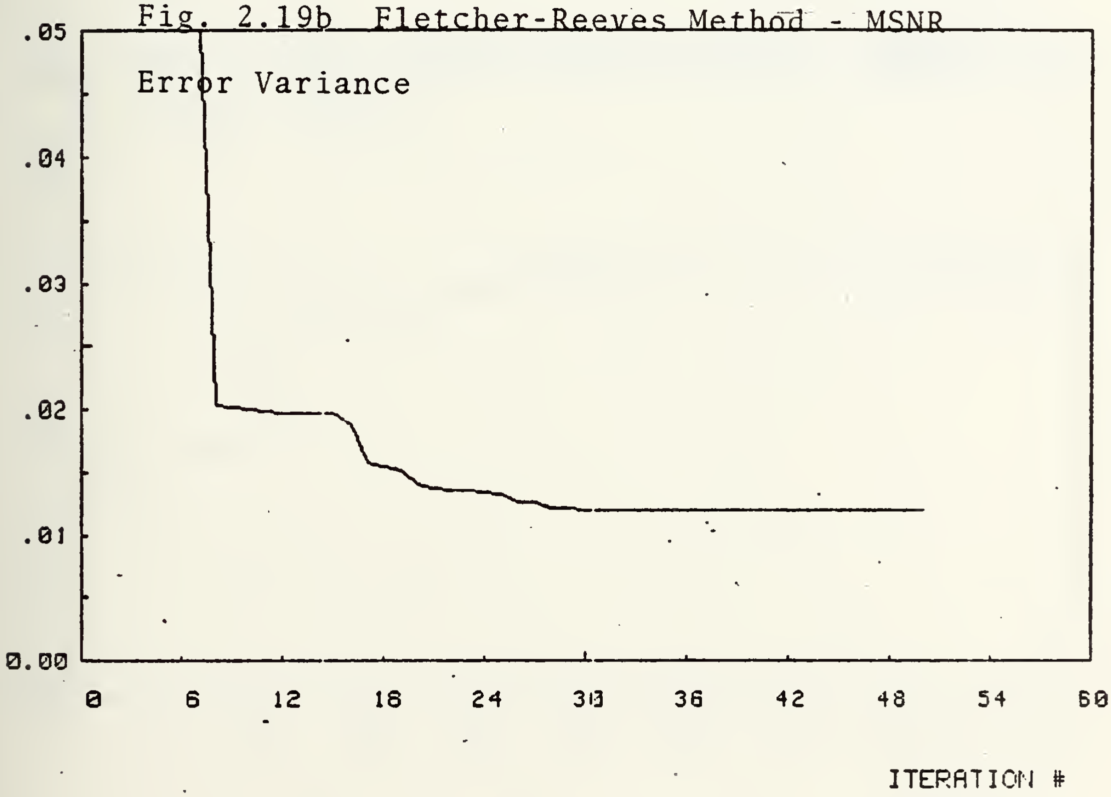


Fig. 2.19c Fletcher-Reeves Method - MSNR

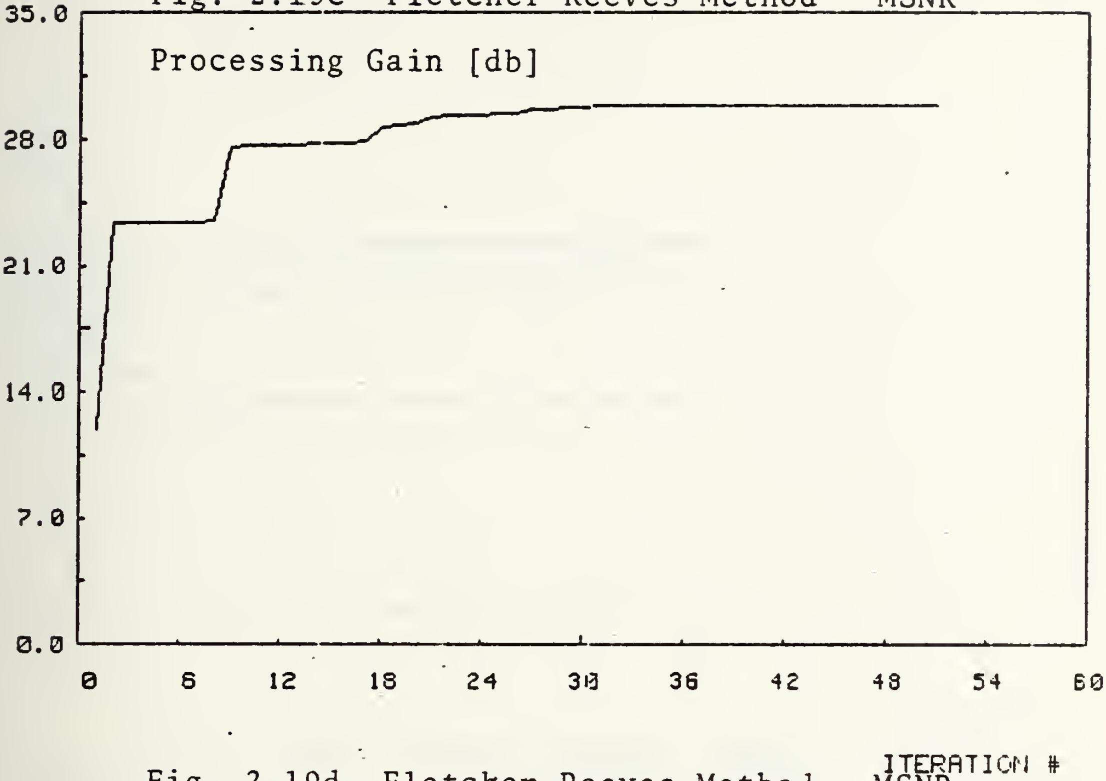


Fig. 2.19d Fletcher-Reeves Method - MSNR

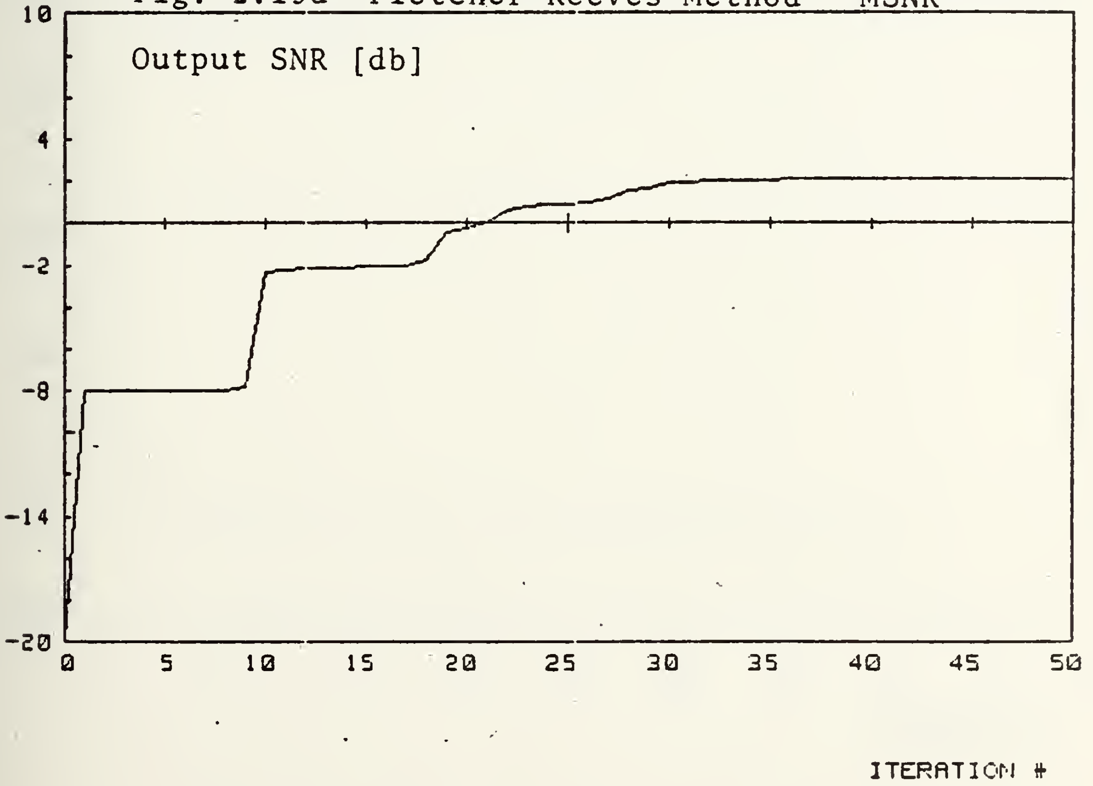


Fig. 2.20a Pollack Method - MSNR

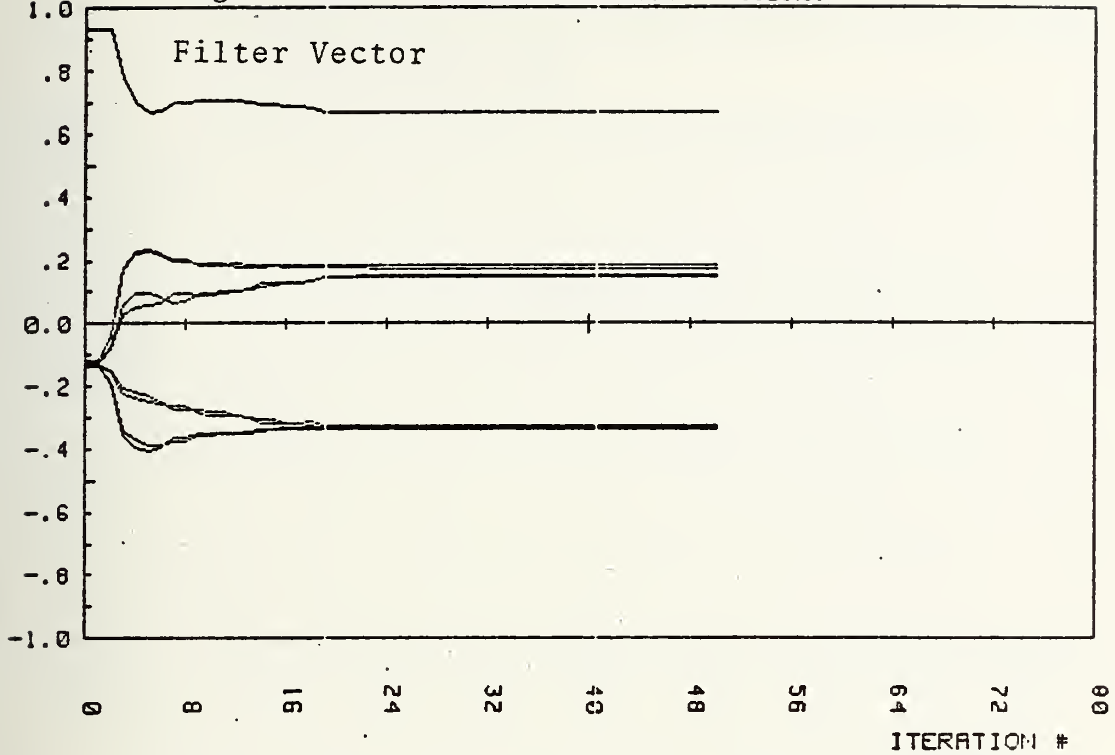


Fig. 2.20b Pollack Method - MSNR

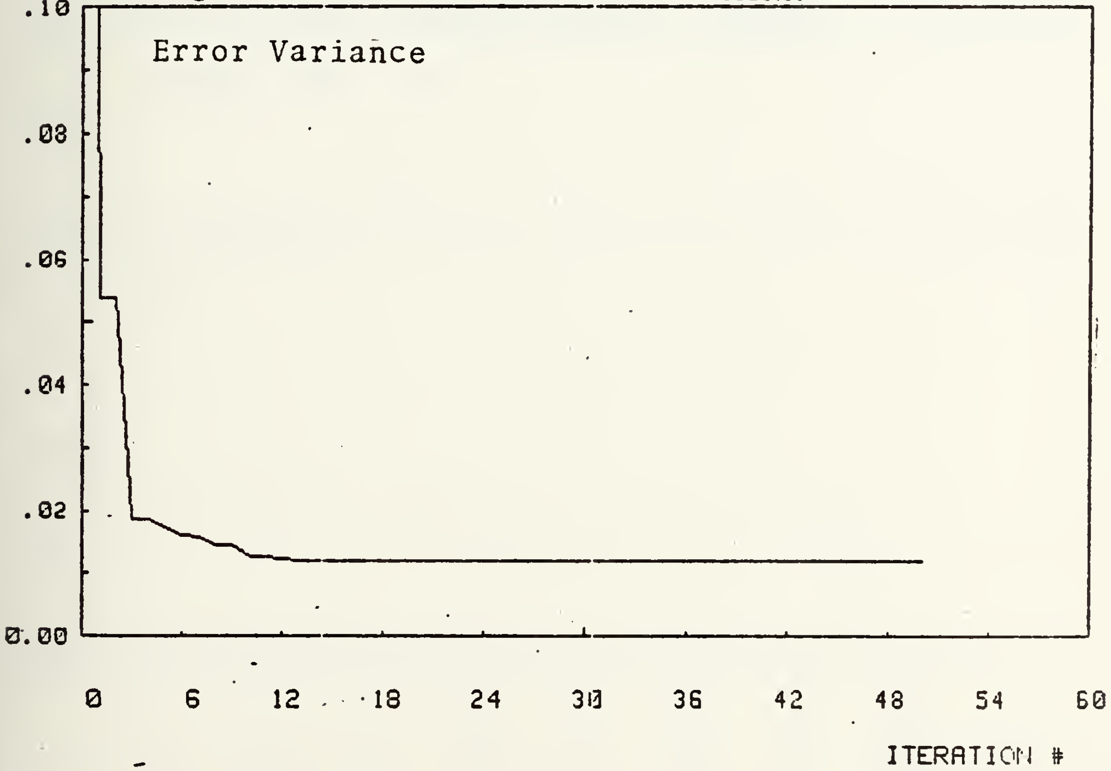


Fig. 2.20c Pollack Method - MSNR

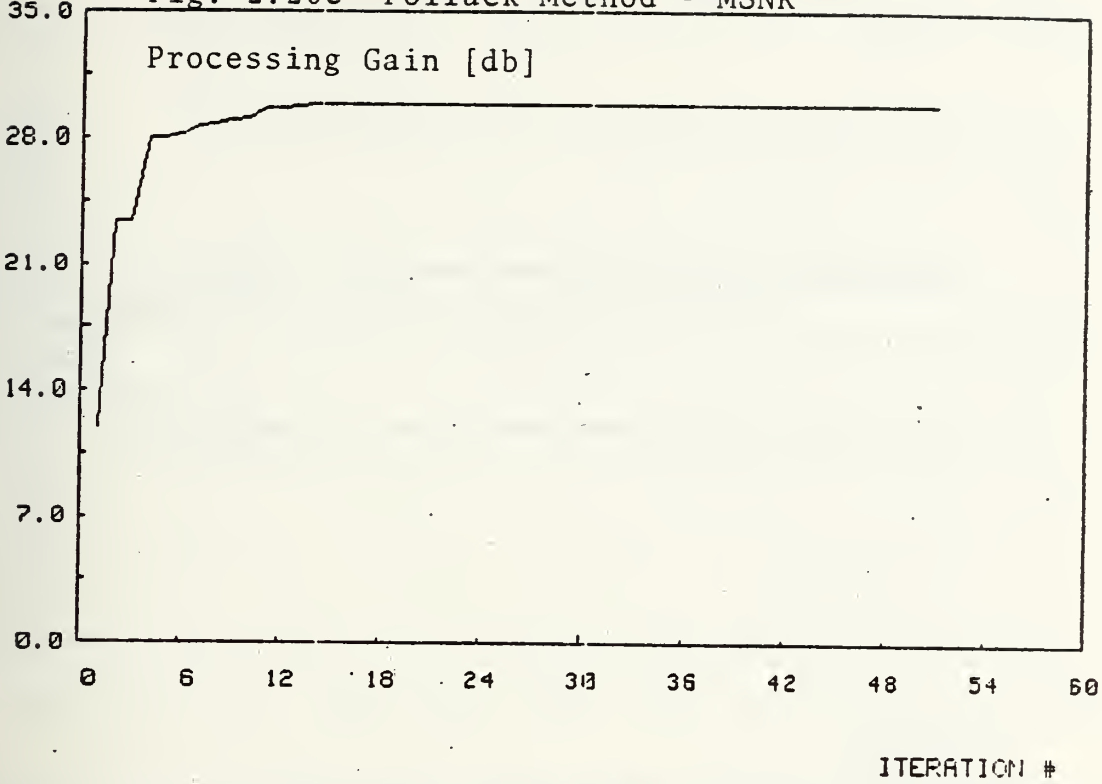


Fig. 2.20d Pollack Method - MSNR

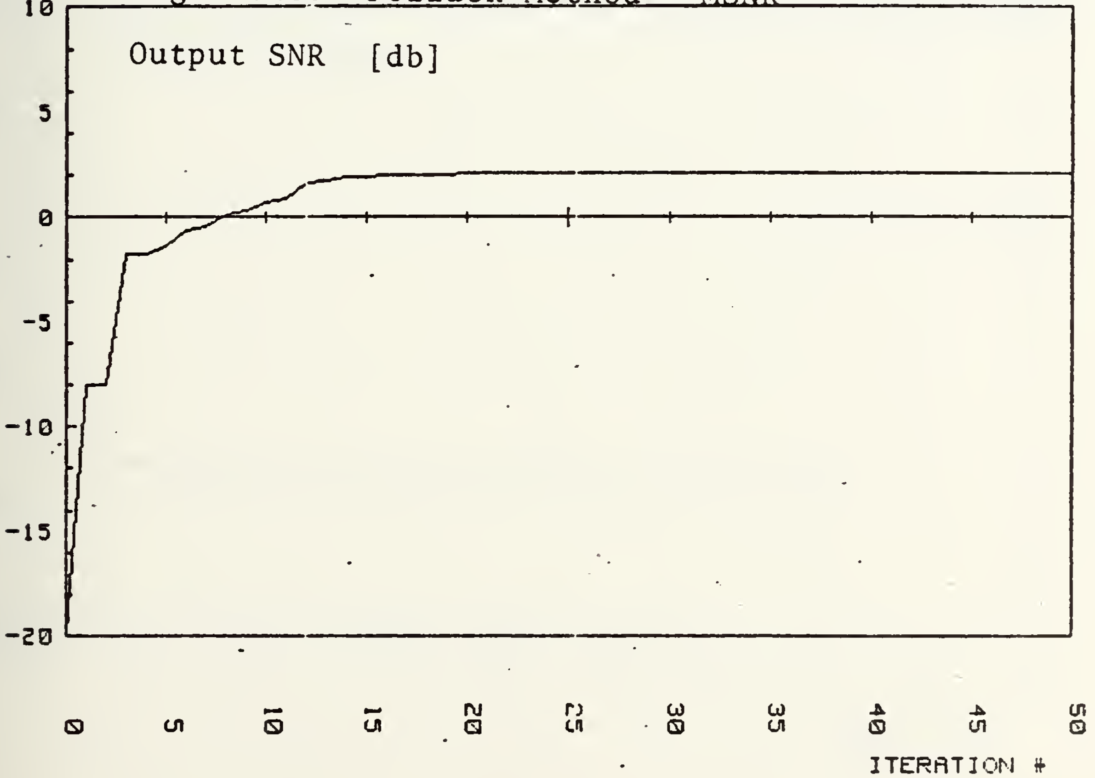


Fig. 2.21a Davidon-Fletcher-Powell Method - MSNR

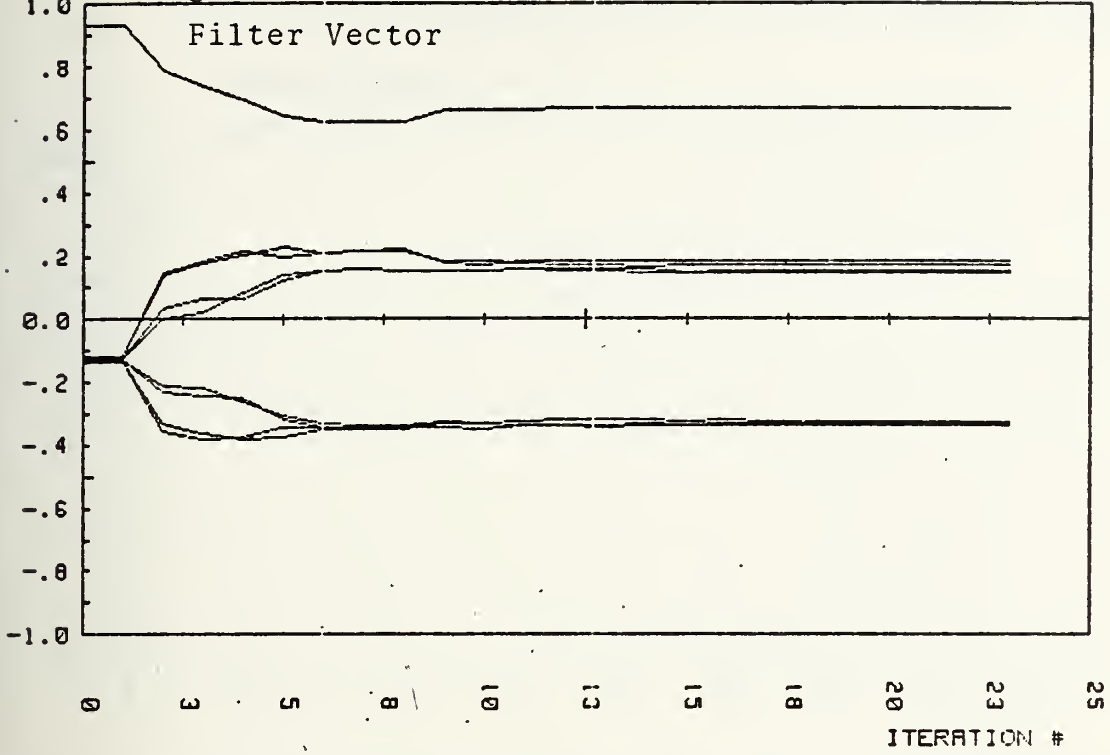


Fig. 2.21b Davidon-Fletcher-Powell Method - MSNR

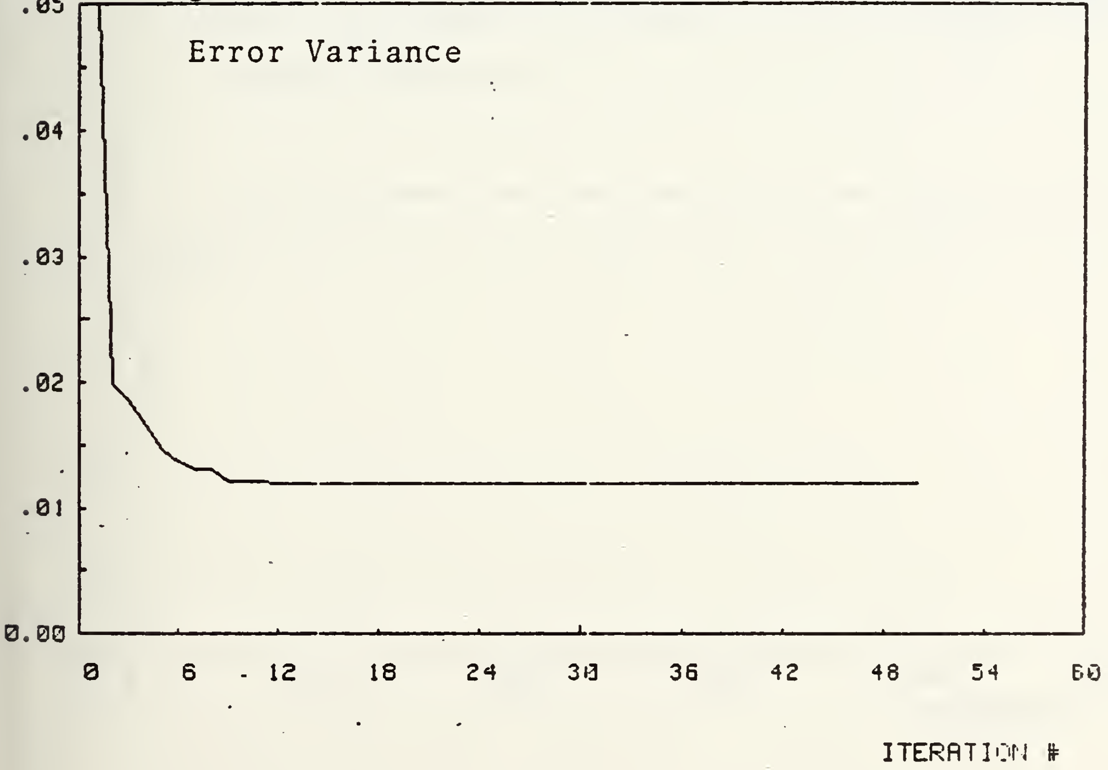


Fig. 2.21c Davidon-Fletcher-Powell Method - MSNR

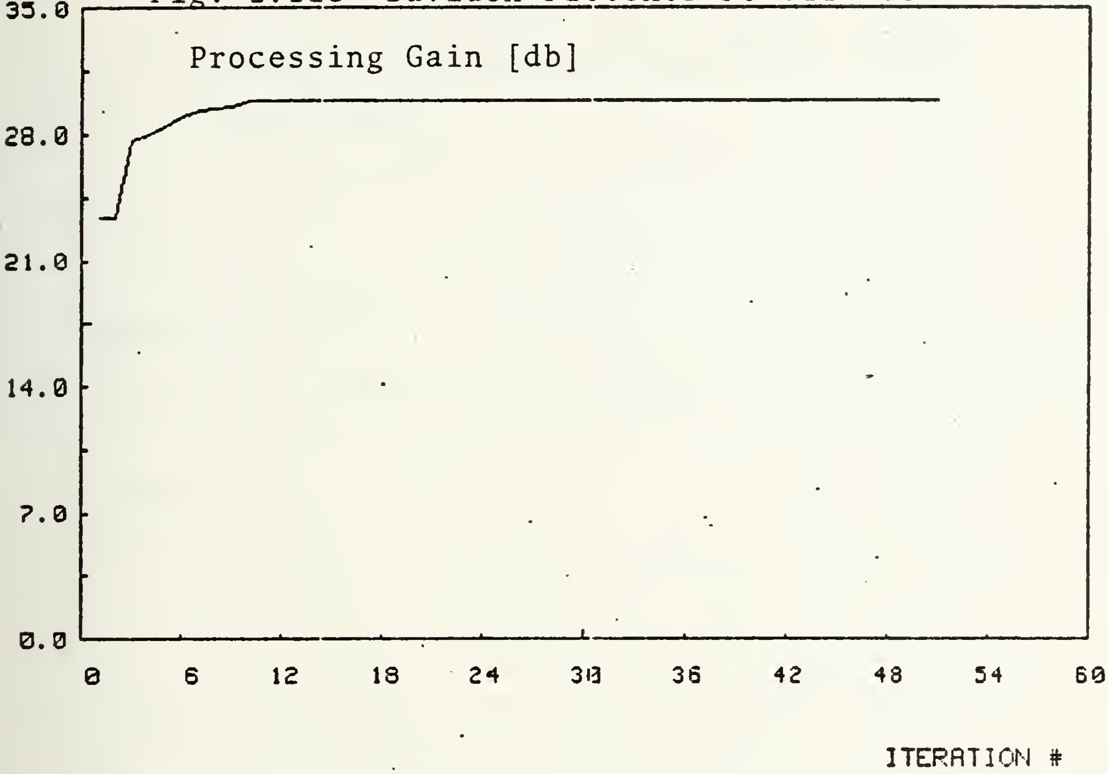


Fig. 2.21d Davidon-Fletcher-Powell Method - MSNR

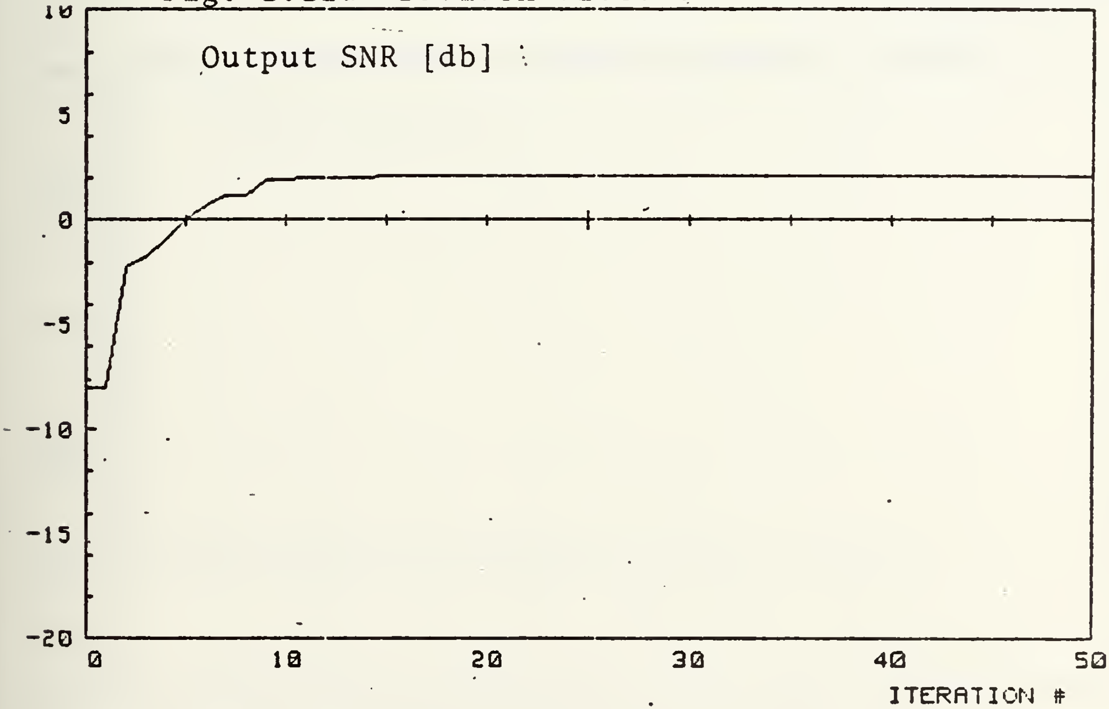


Fig. 2.22a Amir's Transform Method - MSNR

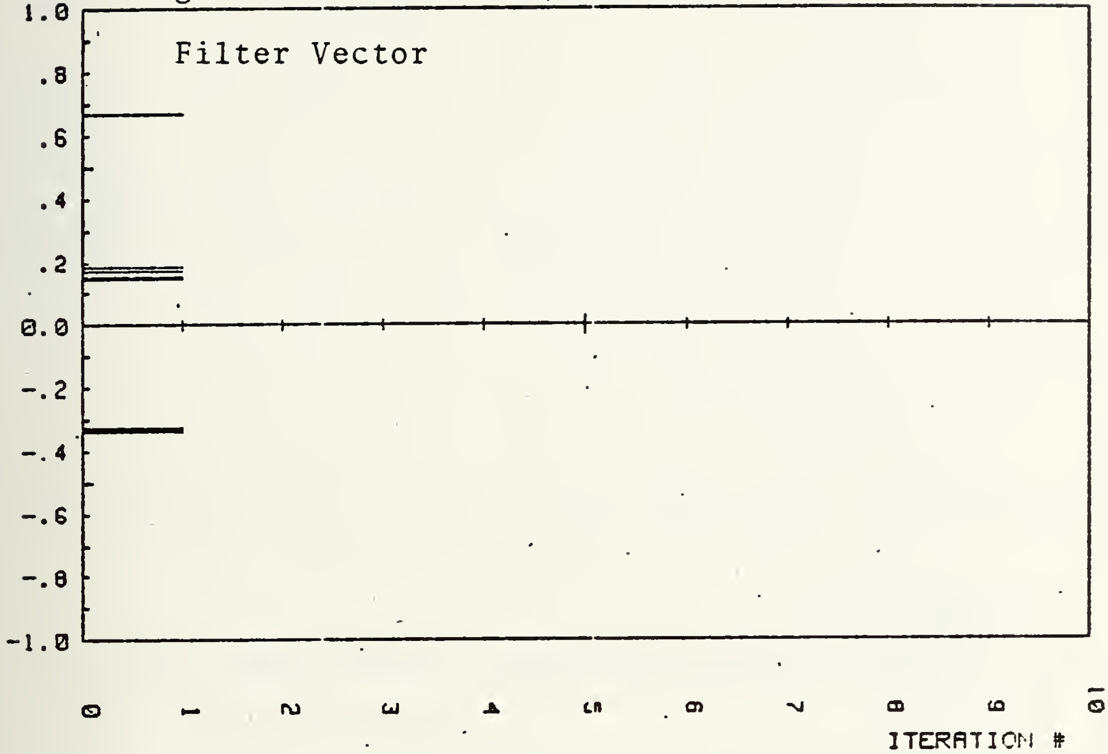


Fig. 2.22b Amir's Transform Method - MSNR

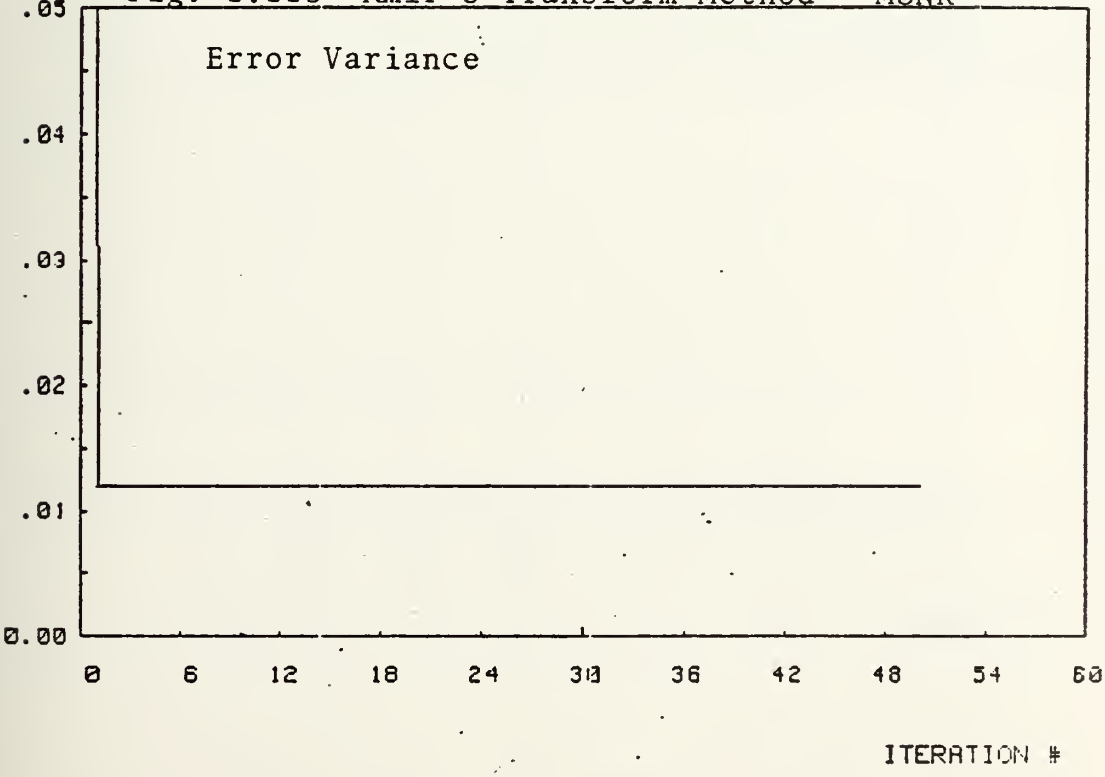


Fig. 2.22c Amir's Transform Method - MSNR

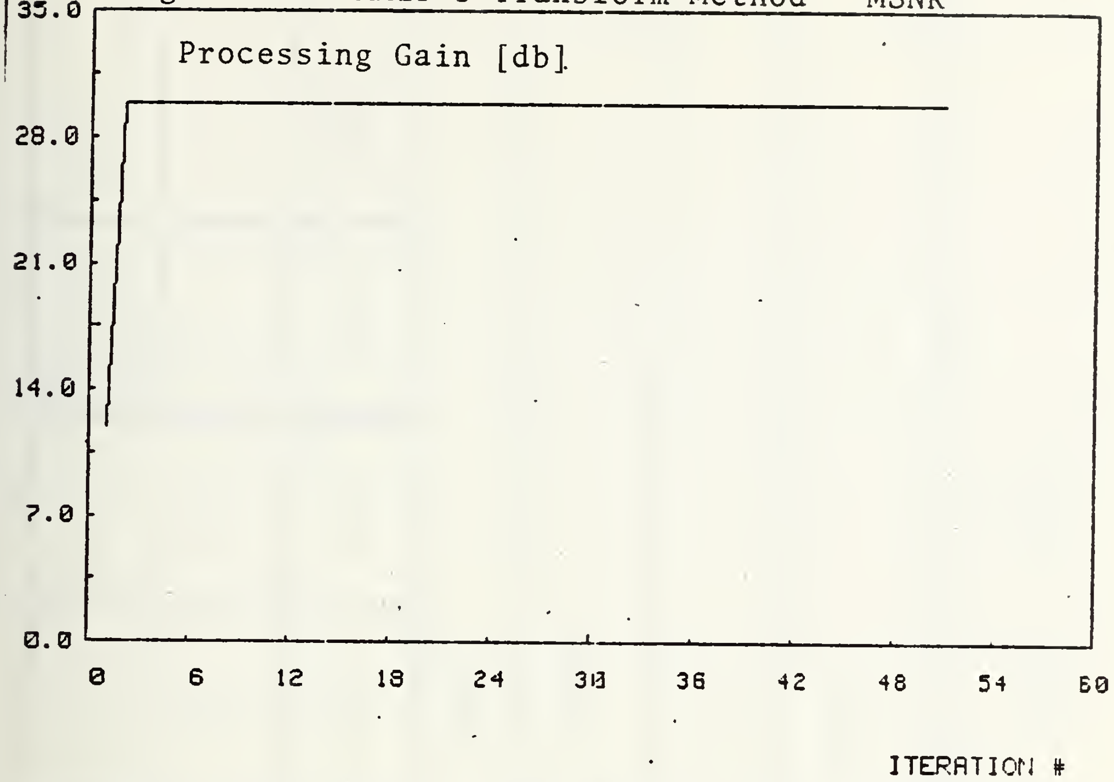


Fig. 2.22d Amir's Transform Method - MSNR

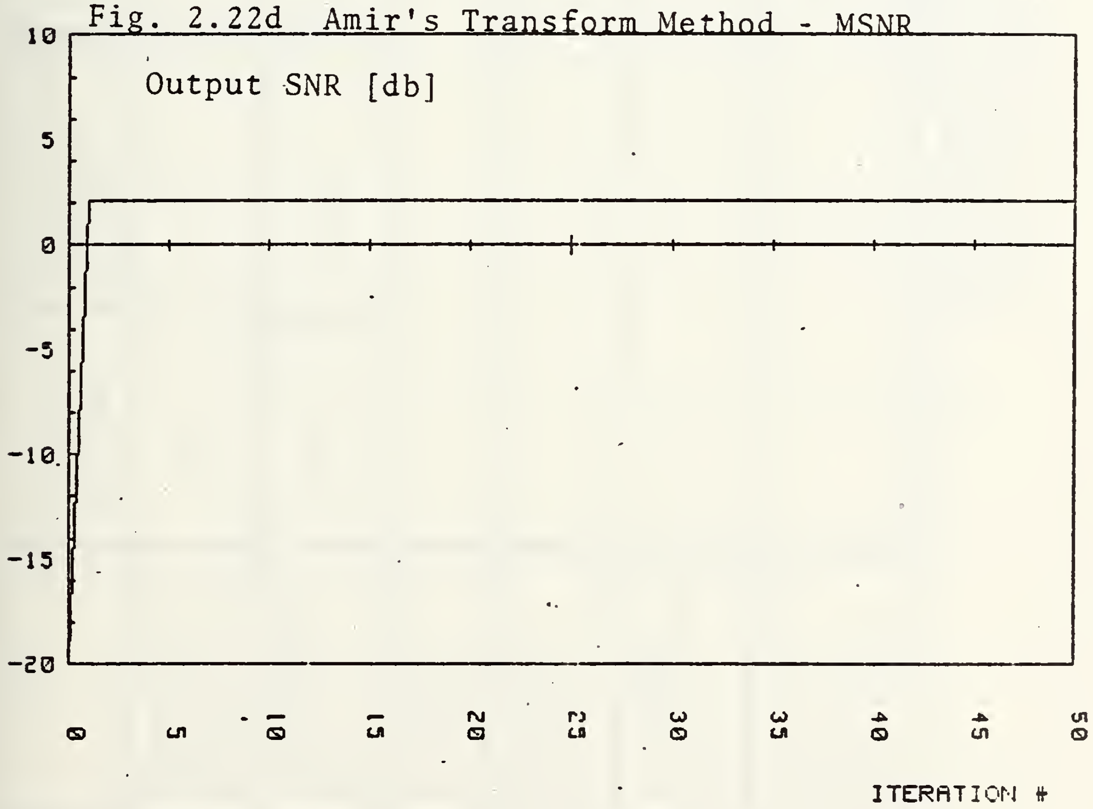


Table II-3 Results of MSNR Adaptive Spatial Filter (Indiana Image)

	Steepest Descent	Accelerated Steepest Descent	Fletcher Reeve's	Pollack	Davidon Fletcher Powell	Amir's Transformation
Number of Adaptation Steps	457	457	59	50	23	1
Adaptation Error At Stop	$9.99\text{x}10^{-11}$	$9.99\text{x}10^{-11}$	$4.54\text{x}10^{-12}$	$3.5\text{x}10^{-11}$	$2.94\text{x}10^{-11}$	$3.98\text{x}10^{-21}$
Stopping Error	$1.5\text{x}10^{-11}$					
Mean of Filtered Image	$6.495\text{x}10^{-5}$					
Variance of Filtered Image	$1.2\text{x}10^{-2}$					
Processing Gain of Adaptive Filter(db)	29.874					

Test Image: Blue Spike Indiana Infrared Image
32x32 Pixels
Before Filtering: Mean = 3.3039
Variance = 0.74111

The same trend in mMSE cases is found for the MSNR cases. The variable metric method (DFP) is faster than the conjugate gradient methods (CGF, CGP) which are faster than the gradient methods (SD, ASD).

It is important to point out that the transform method (AT) which does not have a corresponding method in the mMSE cases has the fastest convergence speed. It took only two steps compared with the twenty-five steps required for the variable metric method to reach below the stopping condition.

5. Results of MSNR Adaptive Spatial Filters II - China Lake Image

Test results of MSNR adaptive spatial filters using the China Lake test image are presented in the following figures:

Fig. 2.23 - Gradient approach, steepest descent method

Fig. 2.24 - Gradient approach, accelerated steepest descent method

Fig. 2.25 - Conjugate gradient approach, Fletcher-Reeves method

Fig. 2.26 - Conjugate gradient approach, Pollack method

Fig. 2.27 - Variable metric approach, Davidon-Fletcher-Powell method

Fig. 2.28 - Amir's transform method.

Several numerical results are presented in Table II.4:

Output signal to noise ratio

Processing gain

Mean of filtered image

Variance of filtered image

Number of iteration steps to reach below the prescribed stopping error

Actual adaptation error.

Discussion

a. Gradient approaches have not been included in these presentations because their convergence speeds are not as fast as the conjugate gradient, variable metric and Amir's transform methods.

b. Again, the Amir transform method has the fastest convergence speed. It only took three steps to reach below the stopping condition compared with fifteen steps required by the next fastest method, the variable metric method.

c. Based on the experience using the Indiana image and the China Lake image, the comparative behaviors among these methods are similar.

Fig. 2.23a Steepest Descent Method - MSNR

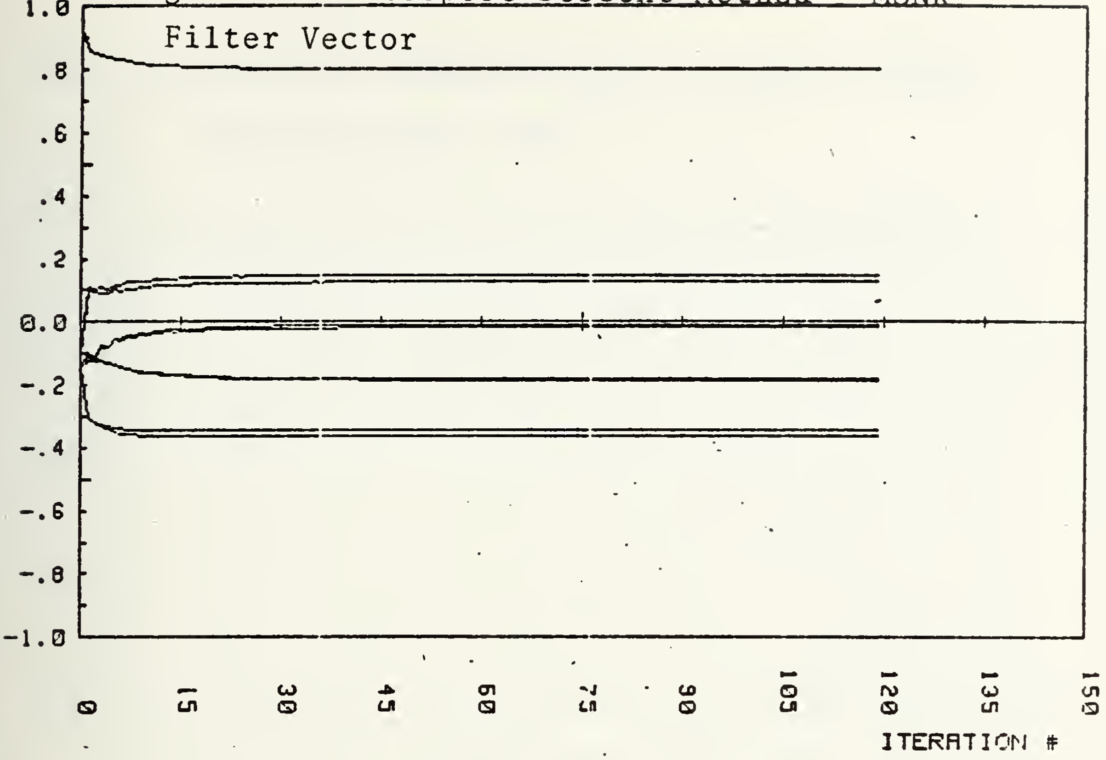


Fig. 2.23b Steepest Descent Method - MSNR

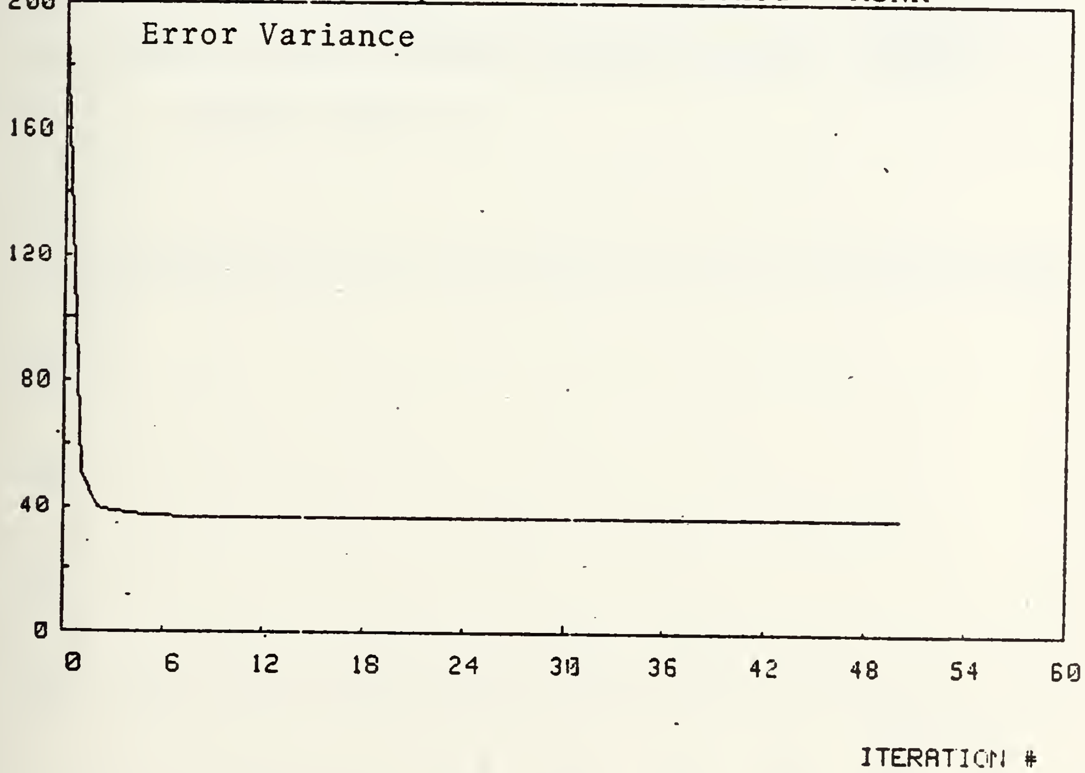


Fig. 2.23c Steepest Descent Method - MSNR

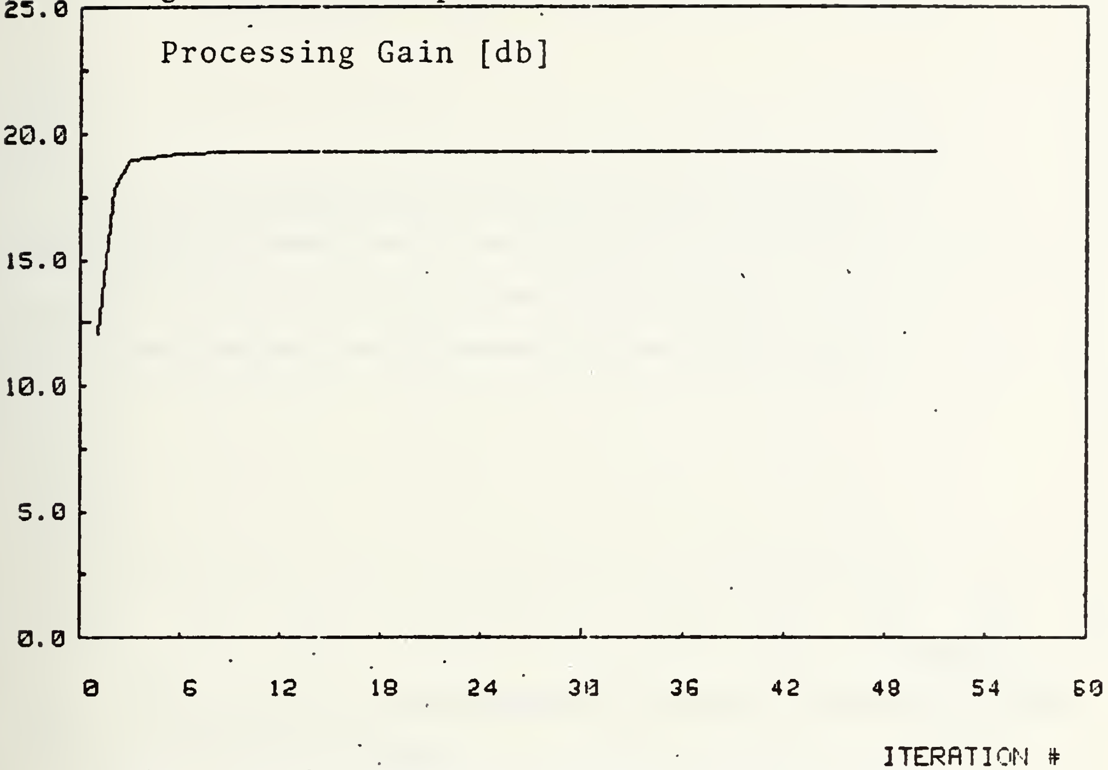


Fig. 2.23d Steepest Descent Method - MSNR

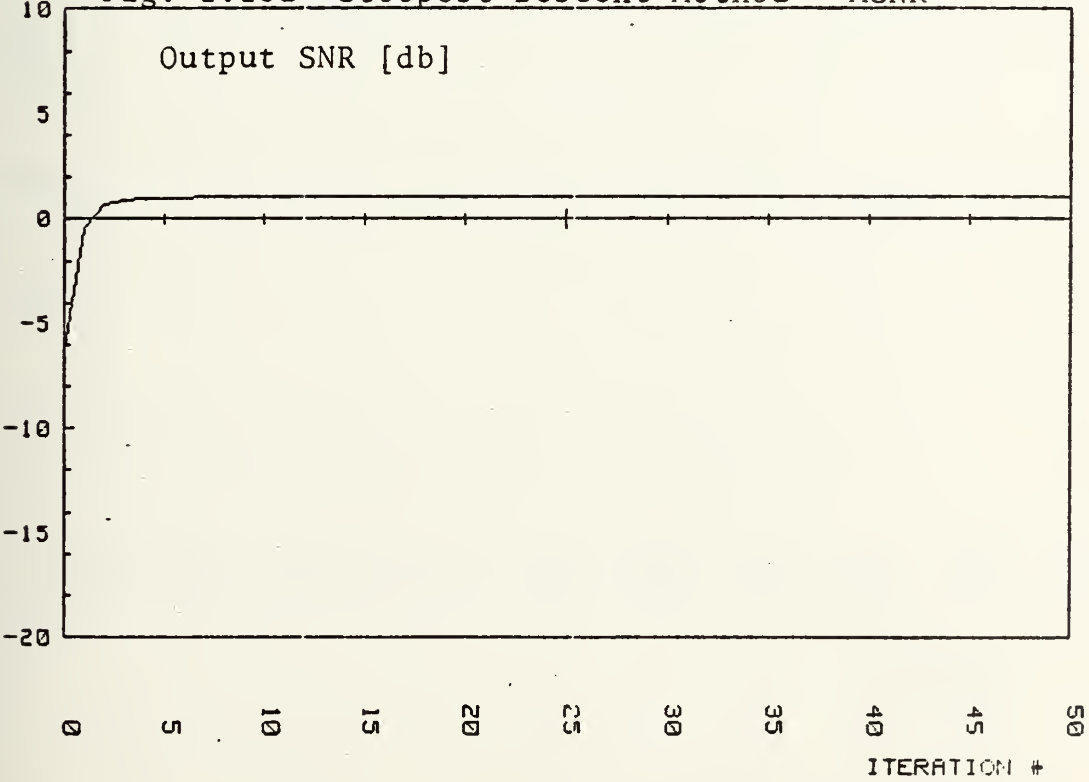


Fig. 2.24a Accelerated Steepest Descent Method - MSNR

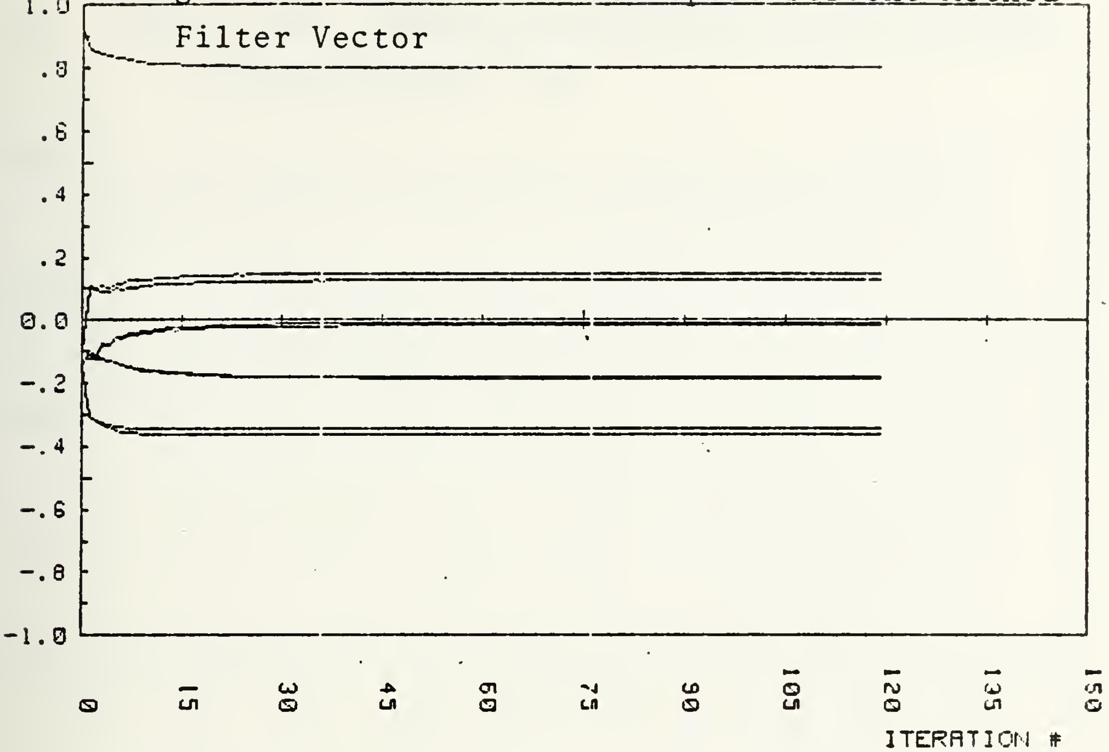


Fig. 2.24b Accelerated Steepest Descent Method - MSNR

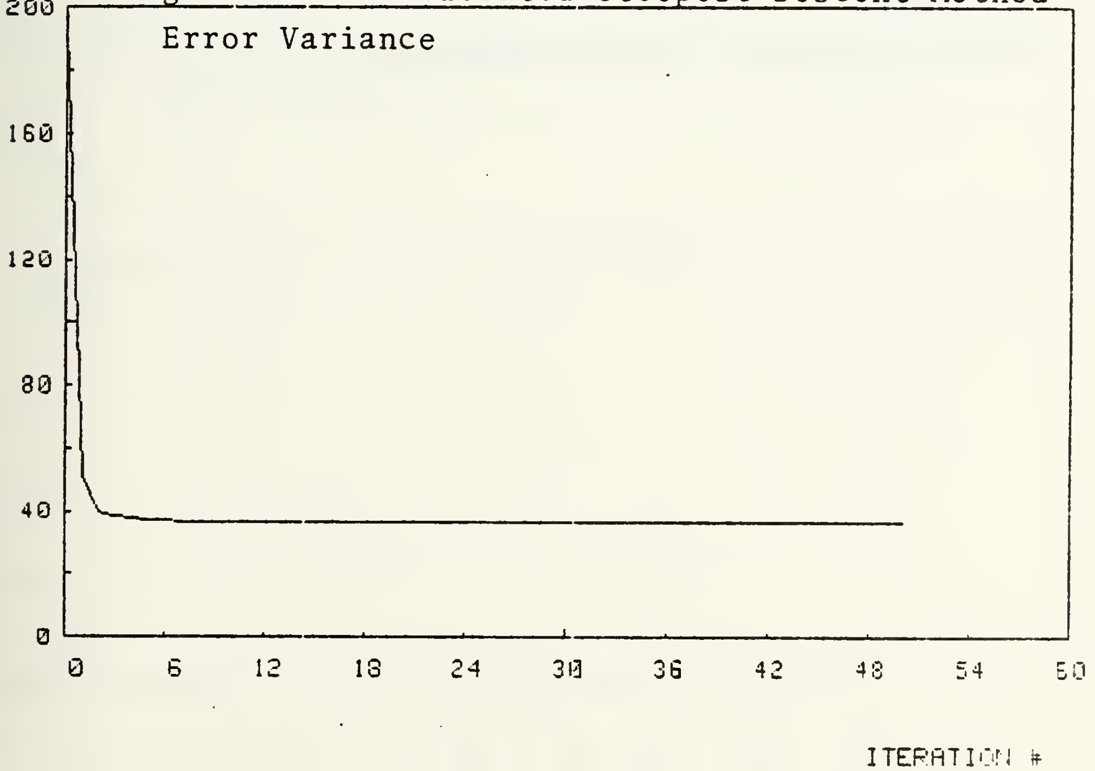


Fig. 2.24c Accelerated Steepest Descent Method - MSNR

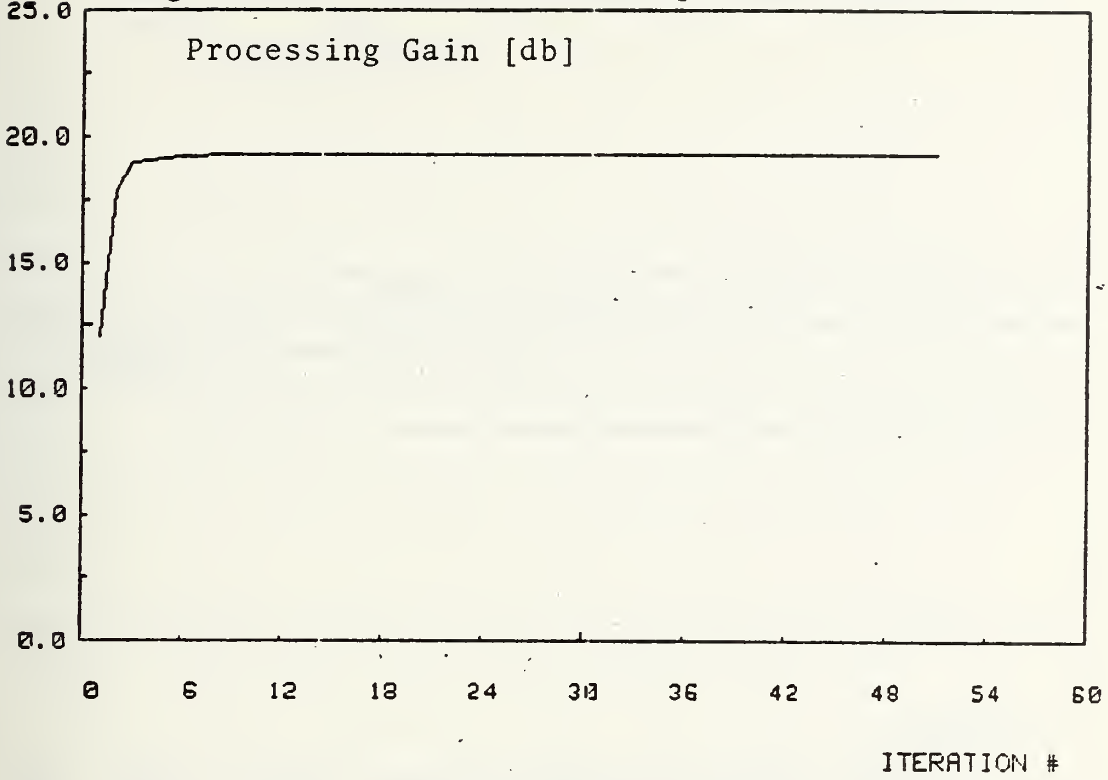


Fig. 2.24d Accelerated Steepest Descent Method - MSNR

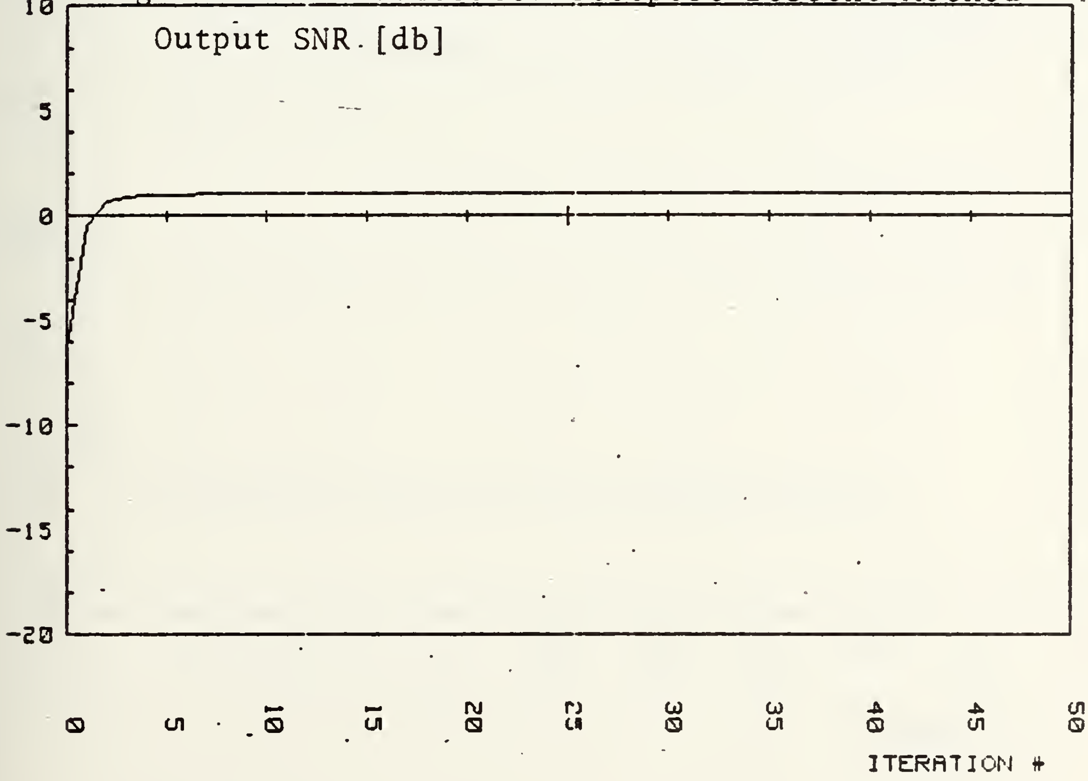


Fig. 2.25a Fletcher-Reeves Method - MSNR

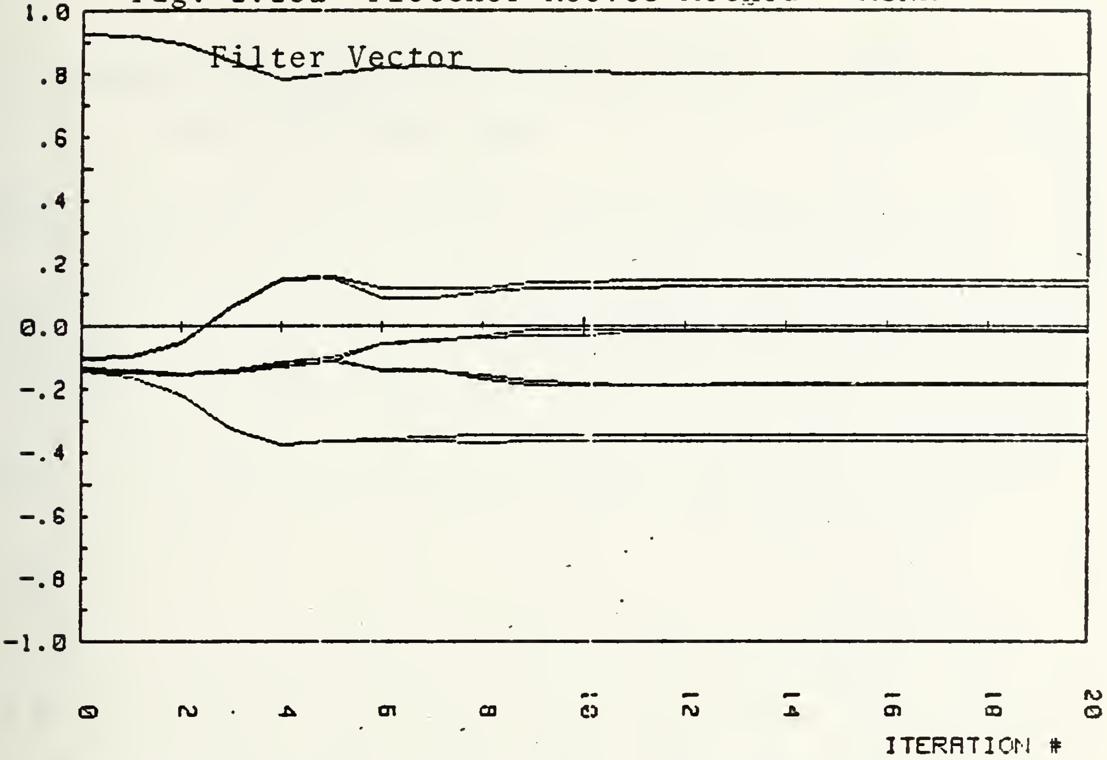


Fig. 2.25b Fletcher-Reeves Method - MSNR

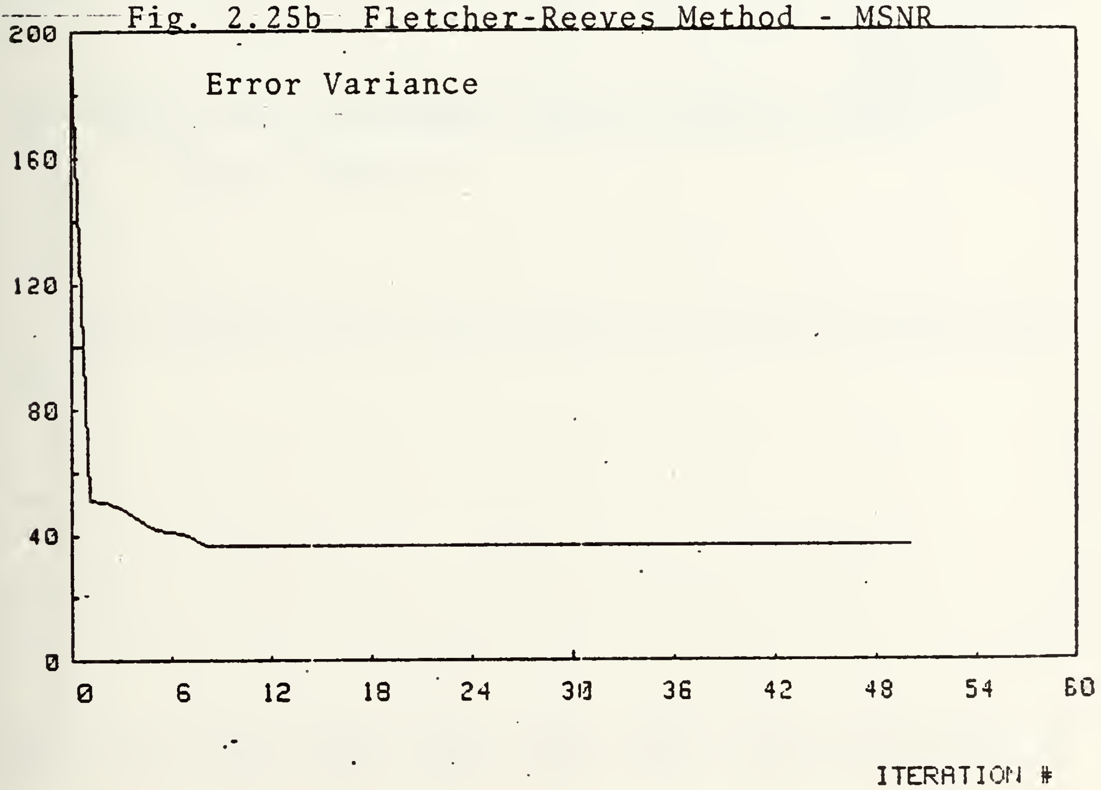


Fig. 2.25c Fletcher-Reeves Method - MSNR

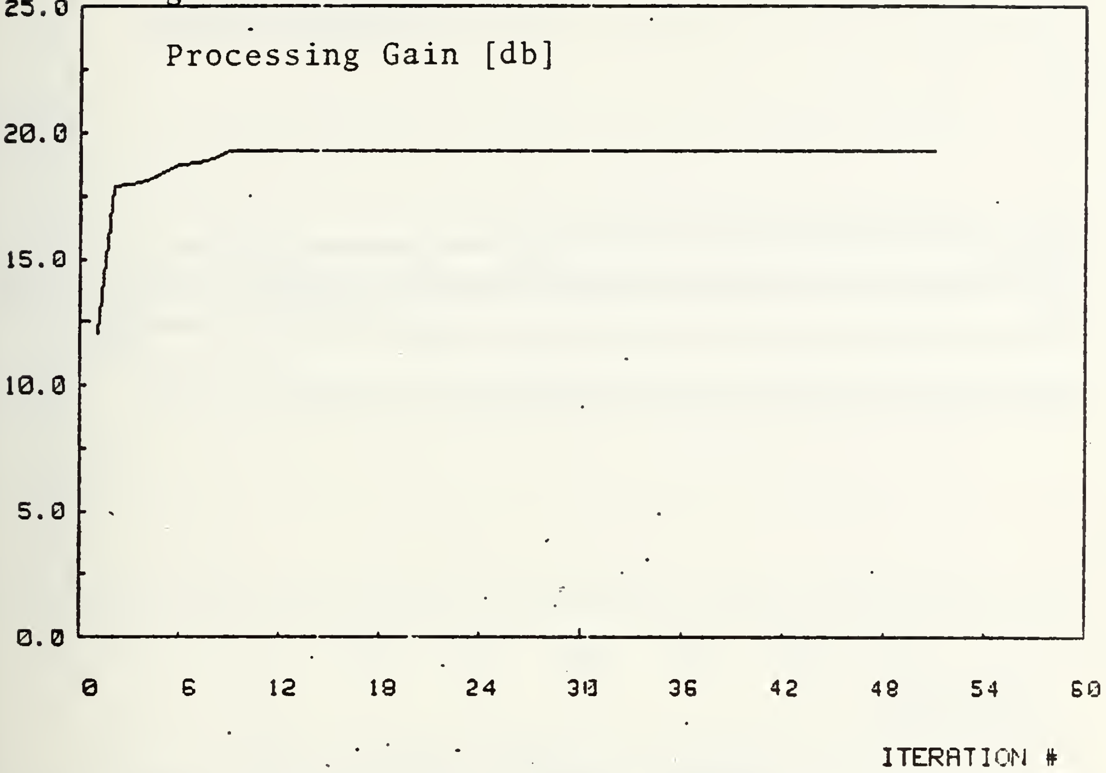


Fig. 2.25d Fletcher-Reeves Method - MSNR

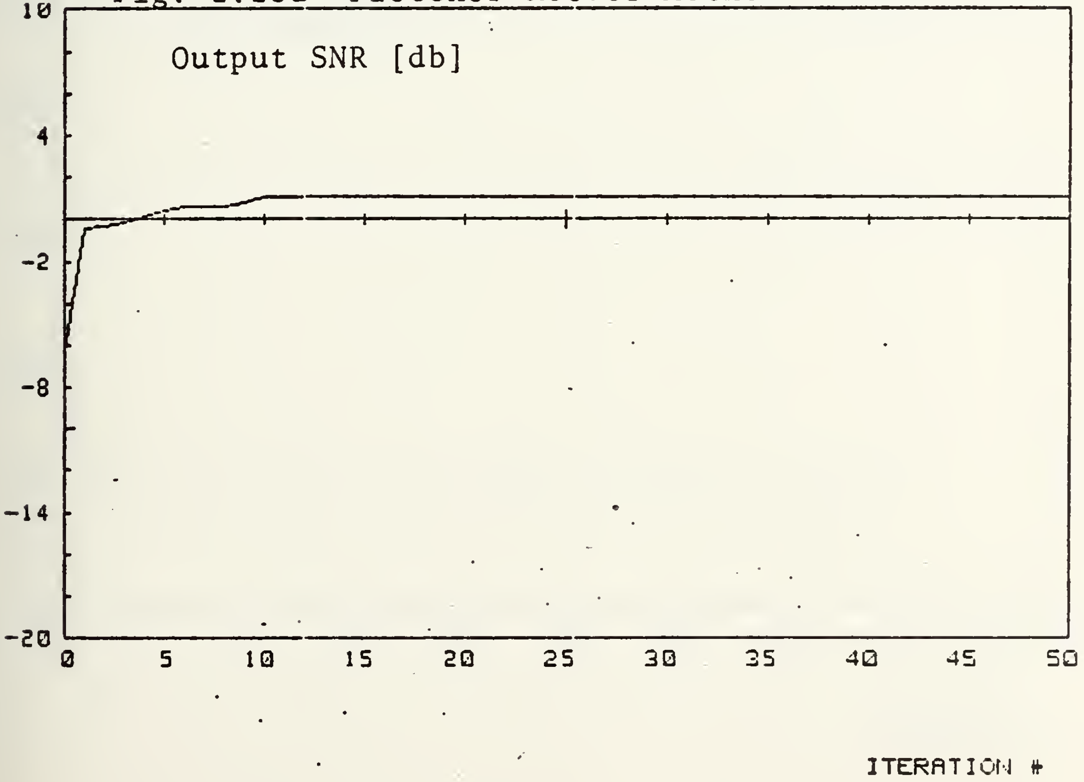


Fig. 2.26a Pollack Method - MSNR

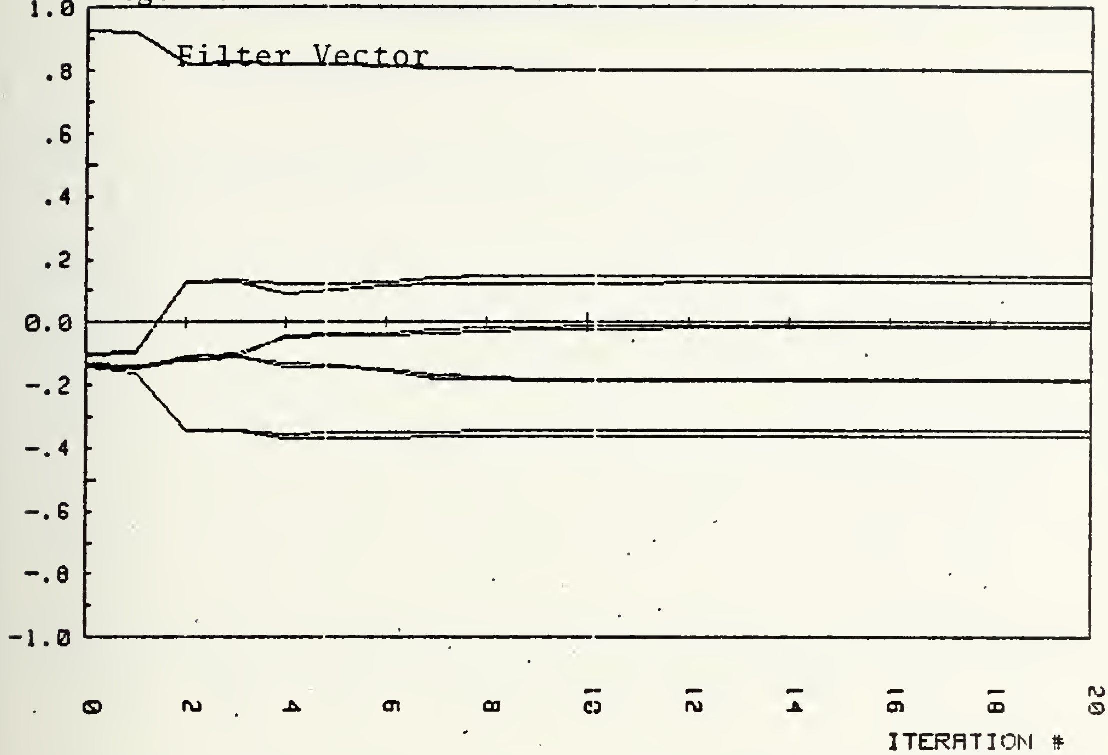


Fig. 2.26b Pollack Method - MSNR

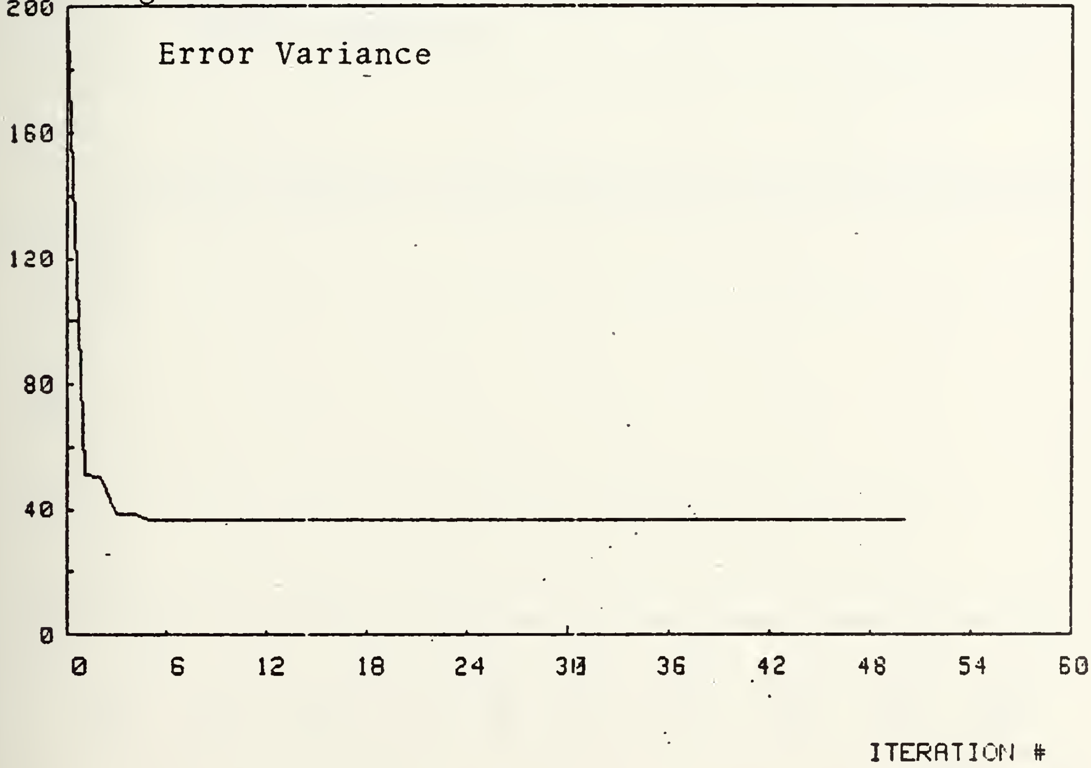


Fig. 2.26c Pollack Method - MSNR

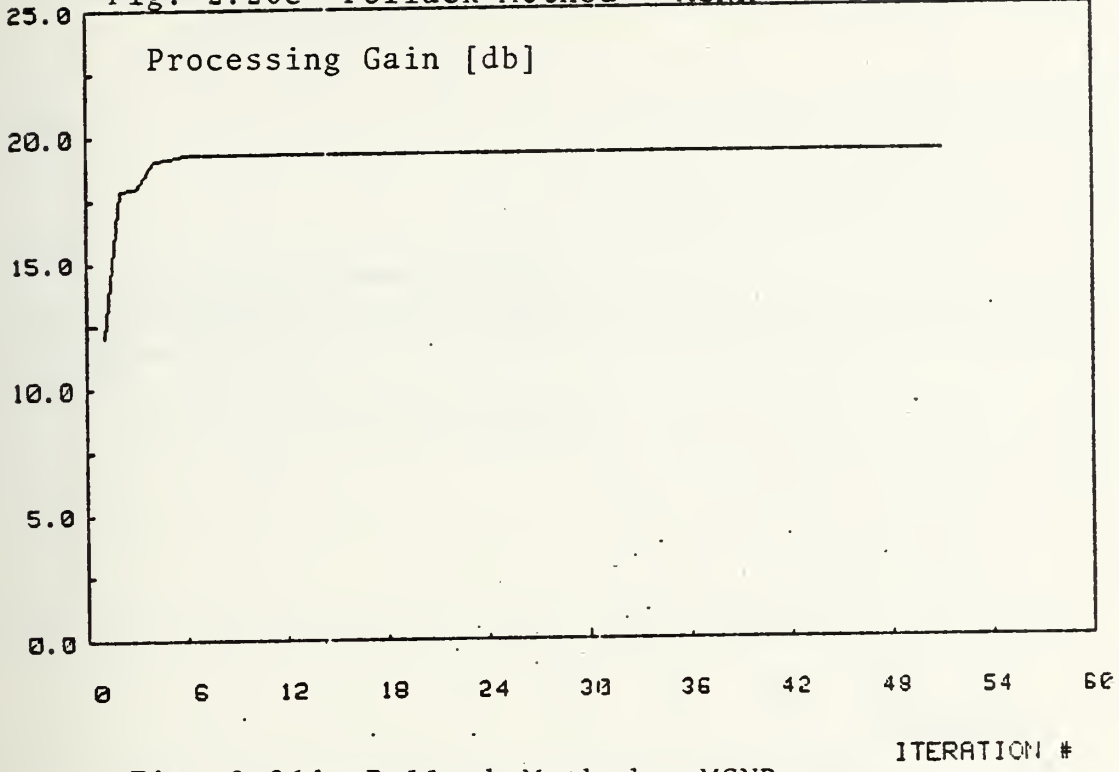


Fig. 2.26d Pollack Method - MSNR

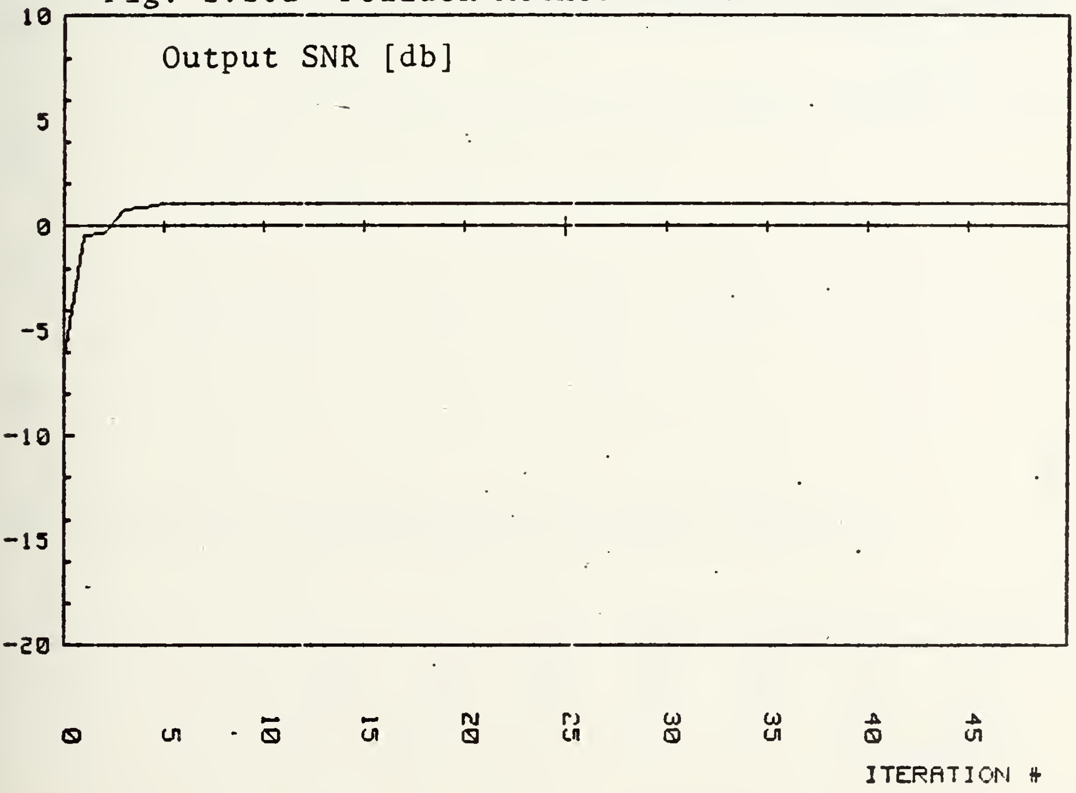


Fig. 2.27a Davidon-Fletcher-Powell Method - MSNR

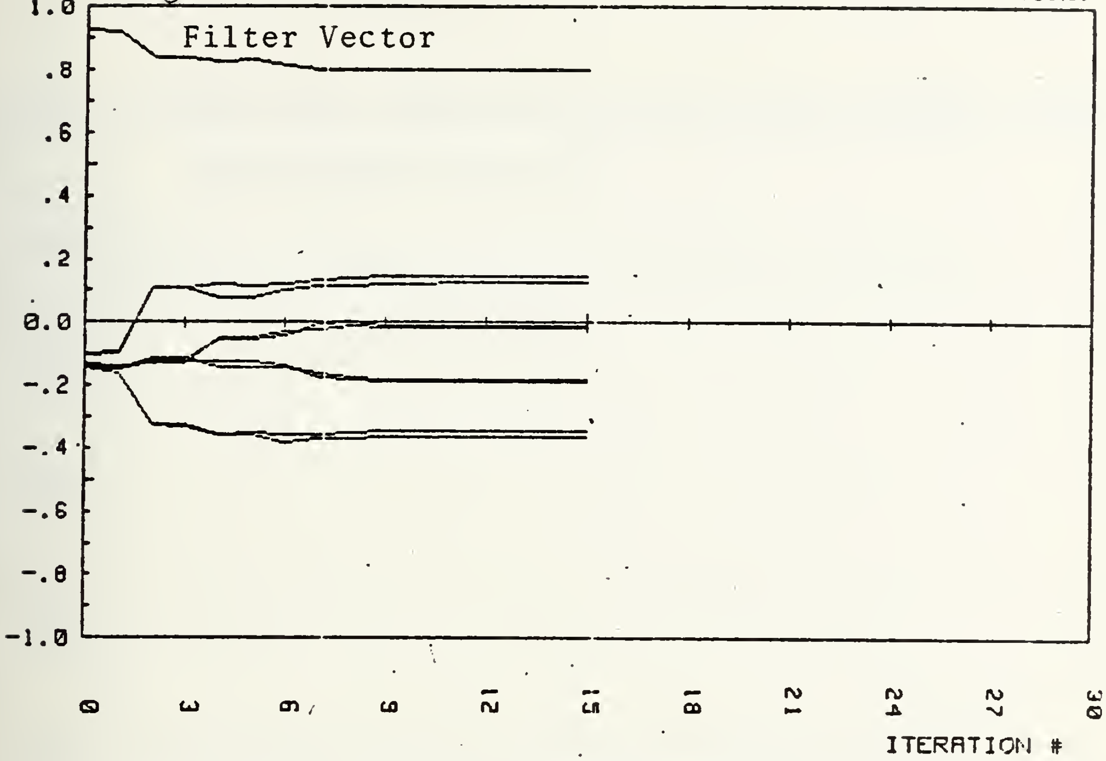


Fig. 2.27b Davidon-Fletcher-Powell Method - MSNR

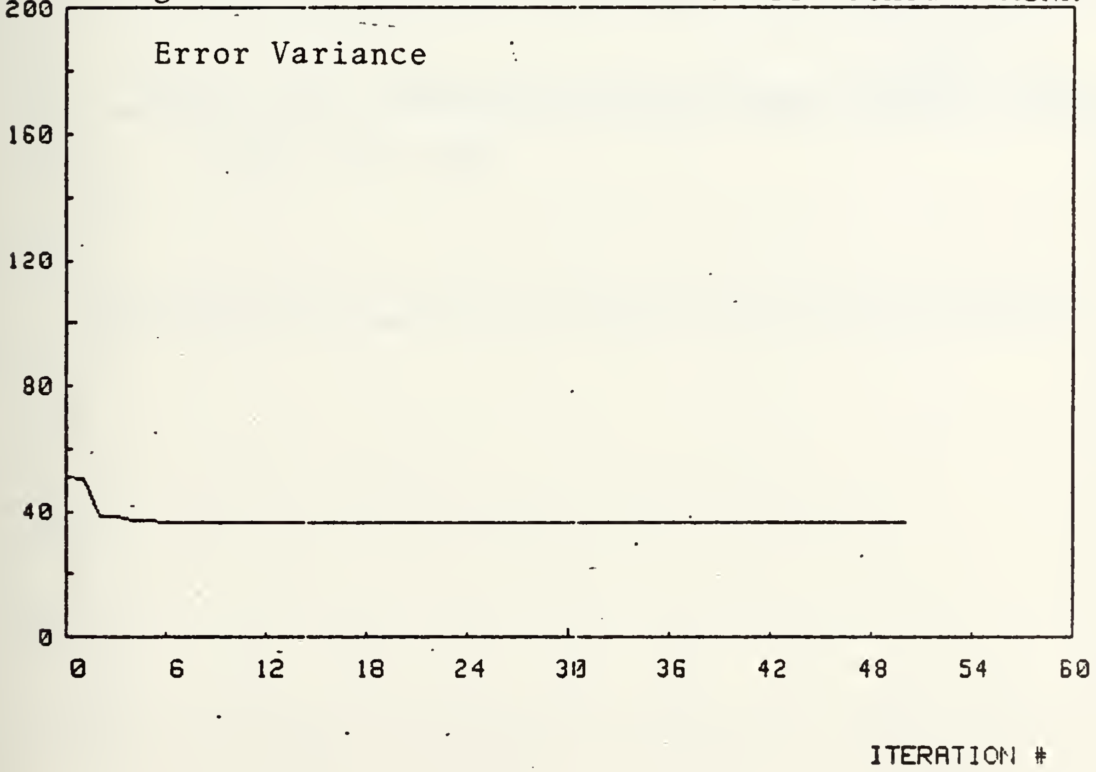


Fig. 2.27c Davidon-Fletcher-Powell Method - MSNR

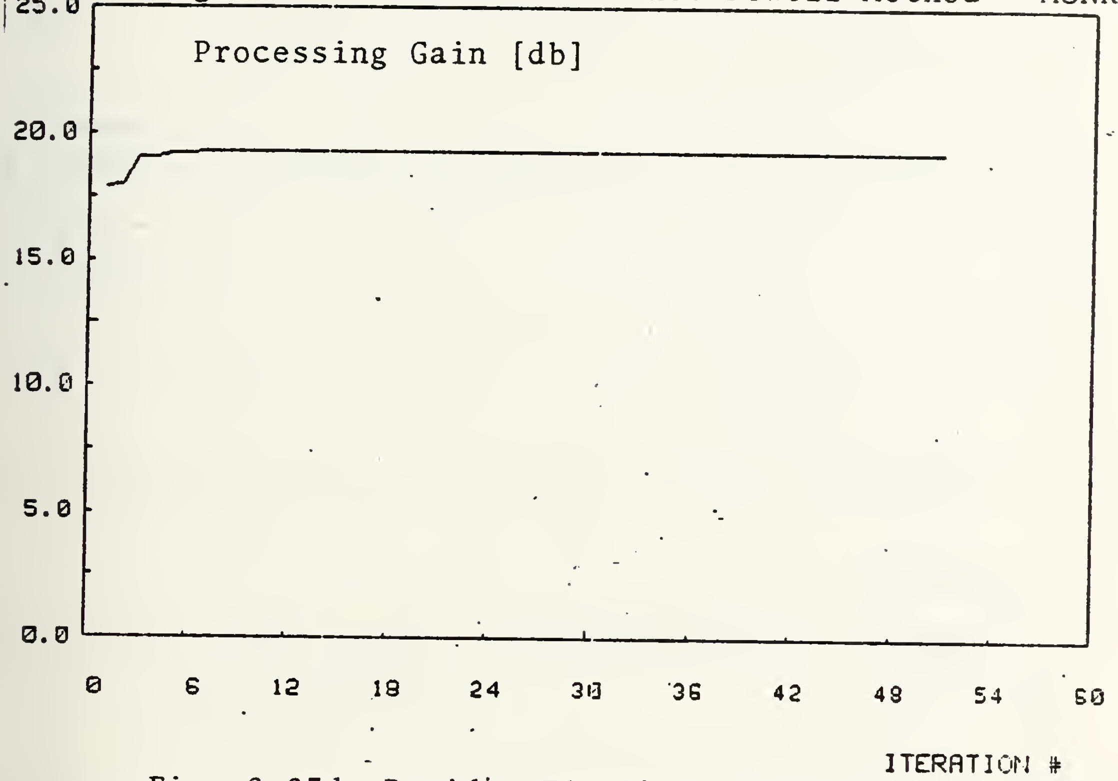


Fig. 2.27d Davidon-Fletcher-Powell Method - MSNR

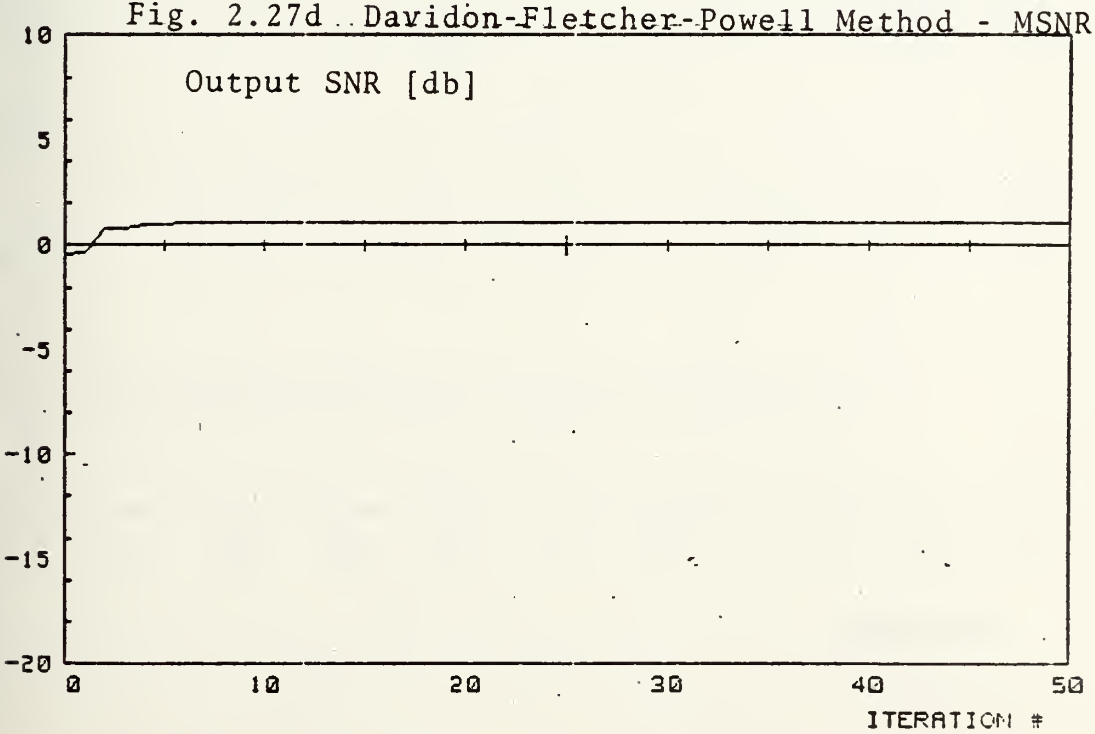


Fig. 2.28a Amir's Transform Method - MSNR

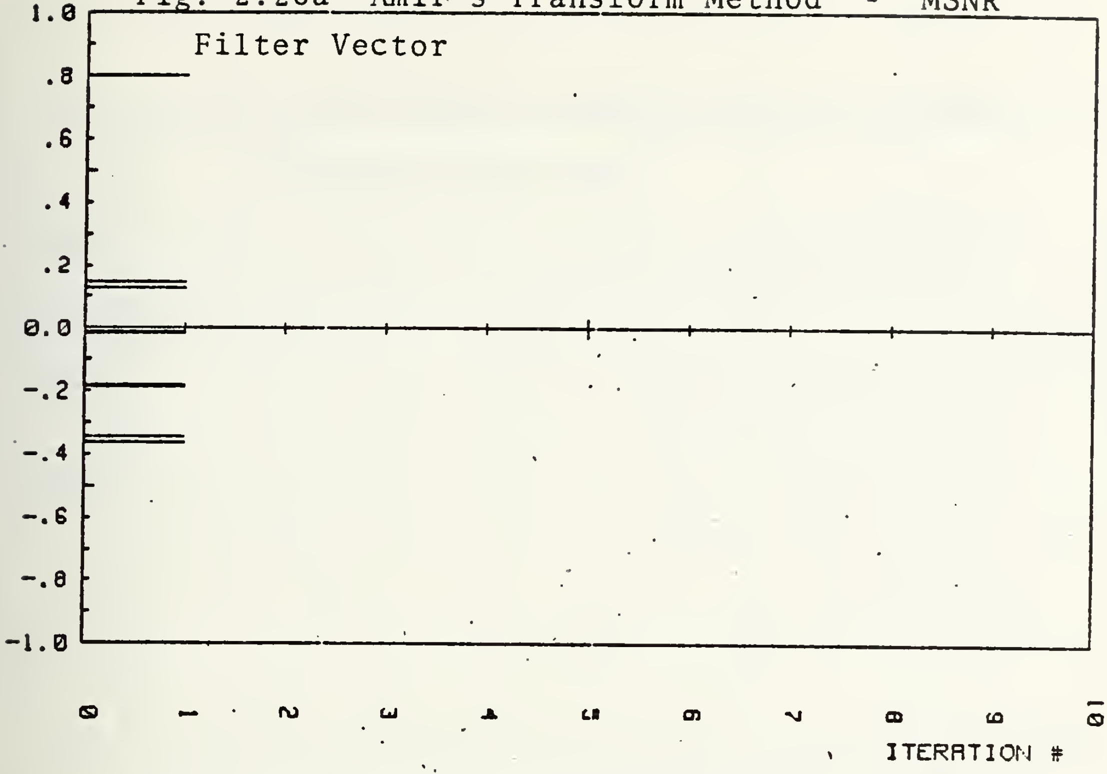


Fig. 2.28b Amir's Transform Method - MSNR

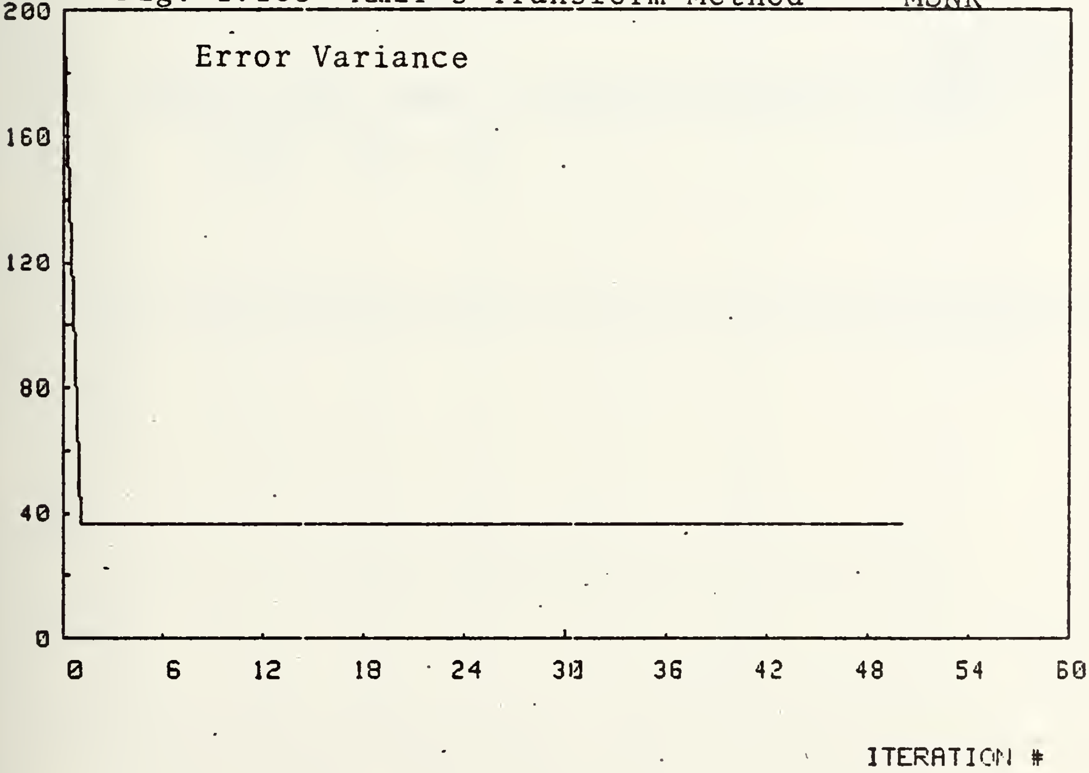


Fig. 2.28c Amir's Transform Method - MSNR

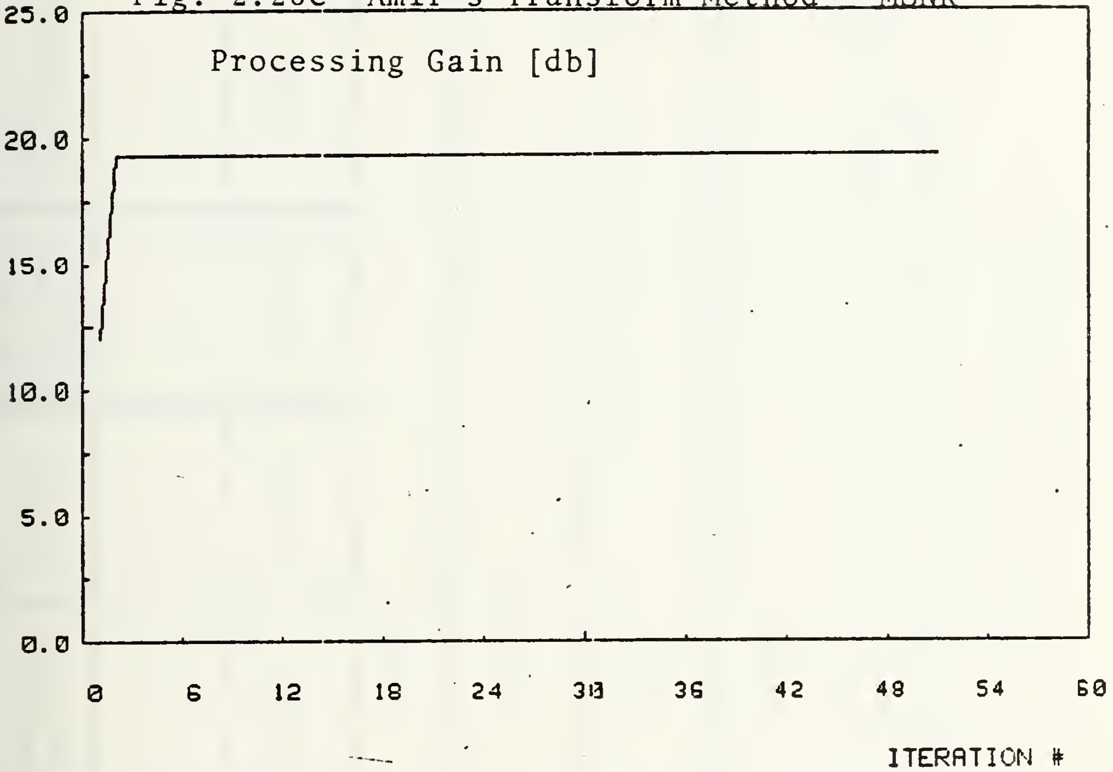
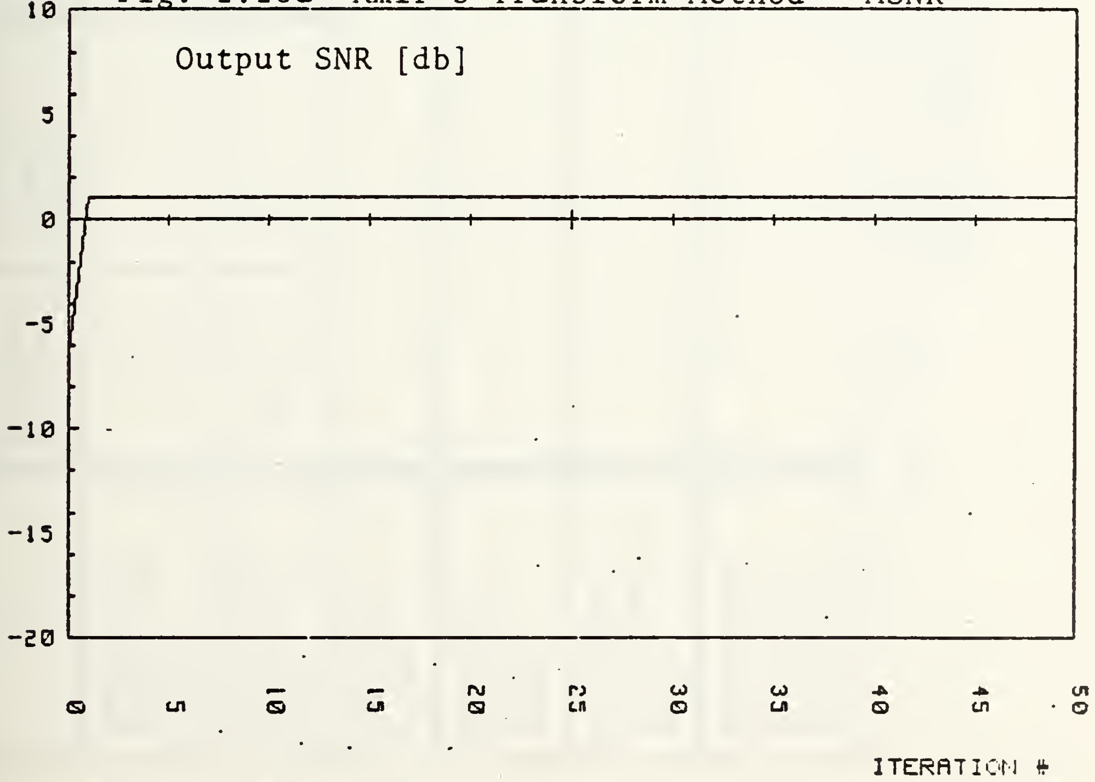


Fig. 2.28d Amir's Transform Method - MSNR



	Steepest Descent	Accelerated Steepest Descent	Fletcher Reeve	Pollack	Davison Fletcher Powell	Amir's Transformation
Number of Adaptation Steps	69	69	20	20	14	1
Adaptation Error at Stop	$9x10^{-11}$	$9x10^{-11}$	$9.15x10^{-11}$	$1.78x10^{-11}$	$2.06x10^{-11}$	$9.57x10^{-25}$
Stopping Error	$1.5x10^{-11}$					
Mean of Filtered Image	$-3.2x10^{-2}$					
Variance of Filtered Image	36.422					
Processing Gain of Adaptive Filter(db)	19.32					

Test Image: China Lake Infrared Image in the 10-13 Micron Band
32x32 pixels
Before Filtering: Mean = 54.02
Variance = 193.464

III. THE MULTIPLE MICROCOMPUTER SYSTEM

A. INTRODUCTION

1. General

Signal processing algorithms are usually developed on main frame computers. The transfer of these algorithms to on-board processors in practical systems is, in general, not an easy task because there are many constraints in real systems such as the processing speed, weight, volume, power, fault tolerance and others. This thesis undertook both the theoretical development task and the practical implementation investigation. Specifically, this chapter will present the second part of this thesis which considers the implementation of adaptive image processing algorithms developed in the last chapter by a multiple microcomputer system using concurrent parallel and pipeline processing.

It is important to point out that the digital computer is not the only technique for real time implementation. Depending on the amount and rate of signal data; precision and dynamic range requirements; need of programmability and several oth er factors, different approaches of signal formats, device technologies, signal/data processor architectures should be considered. In many cases, combinations of analog, sampled analog and digital processing approaches using optical,

electronic and acoustical devices probably will offer cost-effective and optimum performance [178-180]. However, with the rapid advances of VLSI/VHSIC technologies in both increasing speed and decreasing power, size and cost, the importance of digital electronic implementation in the form of distributed processing using multiple processors has been increasing at a rapid rate and will undoubtedly play a more and more important role in real on-board implementation. This part of the thesis is to investigate and develop the feasibility and potential of using multiple microcomputer systems for real implementation.

2. Multiple Processor Developments

Multiple microcomputer systems are a subset of larger families of multiple processor systems whose developments were started over twenty years ago. It was obvious for a long time that several processors are better than one. However, how should they be connected together and effectively used has not been obvious at all. The answer depends on many factors. First, what is the objective? Is it real time processing, fault tolerance, multiple users, security, or some combinations of these? Second, what are the available technologies in both hardware and software? Third, what are the constraints in cost, weight, volume, development time, available manpower? The answers have been very different depending on many of these factors. We can identify several major areas of multiple processor developments since the early 1960's.

a. Supercomputers [151, 152]

The first area can be generally called the "supercomputers." Several processors were connected in different ways to offer parallel processing [153-155], pipeline processing [156-158] or combined parallel/pipeline processing capabilities. In some cases, specially designed signal processors, called array processors, are connected to a host computer to offer very fast data crunching capabilities. In most of these cases, the basic processing elements to form the multiple processor systems are special arithmatic or signal processing units, not stand-alone computers. Their inter-communications and the signal flow are usually fixed in the design stage to achieve very fast computing speed and are not changed during operation. Several representative systems are listed in Table III-1. Their common objective is "fast computation" and "high throughput." The processing elements are tightly coupled.

b. Computer Networks [161, 162]

The second area can be generally called the "computer network." Several processors are connected together for intercommunication and resource sharing. The basic processing elements are mainly stand-alone computers. A problem is usually not partitioned and performed concurrently on several processing elements. The system is, in general, loosely coupled. The communication is carried out by messages with appropriate synchronization codes at the beginning and the ending of the message.

c. Ultra-Reliable Fault Tolerant or Highly Available, Graceful Degrading Computers [166, 167]

The third area can be generally called "Fault Tolerant or Highly Available" computers. Multiple processing elements have been connected in different ways to offer either fail-soft, fail-safe or graceful degradation capabilities. In most fault tolerant computers, the redundancy and/or sparing are usually made at the building block levels, such as the CPU, RAM, I/O ports, etc. to make a very reliable and fault tolerant single computer [168]. The intercommunications among the elements are generally fixed. In recent years, because of the steady decrease of the cost of a computer, the basic processing elements in a multiple processor system are a small number of stand-alone computers [169-171]. These systems started a new direction in the multiple processor developments because the intercommunications among the processing elements are no longer fixed. The processing tasks can be flexibly assigned to different processors. This dynamic assignment, or allocation capability, also allows a new system/software approach to fault tolerance and fault repair.

3. Multiple Microcomputer System Developments

The rapid advance of low cost and small microcomputers has extended the development described above into a new dimension because a large number of microcomputers, instead of just a few, can conceivably be interconnected into a system. Not only can its fault tolerance capability be further

increased, the computational or signal processing capability can also be much enhanced by providing concurrent parallel and pipeline processing capabilities.

The beginning of the multiple minicomputer system development was started at the Carnegie Mellon University in their Cmmp system [172]. Although it used PDP-11 minicomputers, its tightly coupled architecture and dynamic memory allocation concept allowed a relatively large number of processing elements to join together into a single system. This development was soon followed by a tightly coupled multiple microcomputer project, CM* [173], also at Carnegie Mellon University. Since that time, several tightly coupled systems have been proposed [174 to 183]. Some of them have gone beyond the conceptualization stage and started serious hardware/software development efforts. However, none has reached the operational stage at this writing.

At the same time, another direction of multiple microcomputer development has been pursued toward the "computer network" objective [184-188]. These systems can be distinguished from the developments described above in the following major aspects:

- Different types of processing elements are used. In other words, they are "heterogeneous."
- The processing elements are loosely coupled.
- The bandwidth of the intercommunication buses is relatively low.

4. This Thesis Research

The second part of this thesis research is to develop a multiple microcomputer system and to investigate its feasibility in implementing real time on-board signal/data processing for a smart sensor system. It is similar to a number of multiple microcomputer systems in development in the past three to four years which permit up to 16 microcomputers to be interconnected in some ways to perform computations. However, their objectives, architectures, intercommunication concepts, controllers, hardware buses and processing elements, software operating system, etc. are quite different.

This thesis project is presented by highlighting the following features:

a. Its objectives are to provide a multiple tasking system including fast image/signal processing capability and other more moderate speed but highly reliable signal/data processing capability for system management, command and control.

b. Some of the signal/data processing tasks will be performed by tightly coupled processors. But the processors performing other tasks do not have to be all tightly coupled together. Therefore, a mixed tightly and loosely coupled system is envisioned.

c. A part of the system must perform some critical tasks which require ultra-reliability. Other parts of the system only require fail-soft and graceful degradation performance.

In any event, a dynamic allocation capability is required which allows flexible assignment of microcomputers to perform various tasks, which provides some fault tolerance.

For these requirements, a multiple star/multiple cluster system of 16 bit microcomputers was developed. Its general concept and philosophy was developed by a top-down system design procedure which will be presented in the next Section, III.B. It will be explained how a choice was made considering several alternatives and seven important issues related to the system. In Section III.C, detailed implementation of these choices will be presented by describing the principles and circuits of this multiple microcomputer system in five categories:

System architecture
Processing resources
Intercommunication network
Intercommunication procedures
Multibus communication.

The performance of this system is described in Section III.E.

B. DESIGN CONSIDERATIONS FOR THIS MULTIPLE MICROCOMPUTER SYSTEM

1. Introduction

Although only two large multiple microcomputer systems and one multiple minicomputer system have appeared in the literature and reached operational status, a large number of different architectures have been proposed and some are in the process of being implemented. The three operational

systems are all from the Carnegie-Mellon University: CM* [172], Cvmp [191] and Cmmp [173]. There are now many options for the hardware and software design of a multiple microcomputer system.

This thesis took a top-down system design approach to reach the choices made for the design of our system. This design process is presented in several steps in this section to explain the general idea and philosophy of this system. In the next section, the detailed design of various parts will be described.

2. Architecture

This thesis is primarily concerned with the implementation of adaptive image processing. It is important, however, to realize that the adaptive filter is only one part of a longer end-to-end image processing program for detecting, tracking and recognizing targets in noisy images. The adaptive spatial filter is used to enhance the target signal to noise ratio by suppressing the background clutter which may be enhanced by additional image processing techniques, such as the adaptive temporal filters. The clutter suppression stage is followed by thresholding, target acquisition, recognition and tracking stages. These signal processing operations are quite different. For example, adaptive spatial filters require the computation of statistical image characteristics, solving matrix equations. Adaptive thresholding requires the comparison and rearrangement of real numbers.

Target acquisition usually involves pattern tests of numbers based on spatial, temporal and/or spectral information. Therefore, although each individual signal processing stage requires real-time or fast execution speed, different signal processing stages do not depend on one another during processing. Furthermore, it is important to realize that processing of target signals for the mission objective is only one part, although a very important part, of the total signal/data processing requirements for the whole system. There are processing functions such as management, control, communication and others which must also be implemented. The nature and requirements of their processing operations are quite different and vary over a wide range. Some do not need high processing speed but demand very high reliability. Others do limited computation but handle large amounts of data. In general, the signal/data processing requirements of many systems cover a wide range. Therefore, we designed an architecture which has several levels of coupling among processing elements.

At the first level, special processors may be directly coupled to a microcomputer. At the second higher level, several microcomputers are connected to the same system bus in parallel and form a "cluster." A microcomputer can communicate with any other microcomputer on the same bus or within the same cluster directly through common memory. It is a tightly coupled, bus oriented multiple microcomputer architecture.

At a higher level, the third level, four clusters are connected by way of a "complete star" bus switch network and form a "star." The communication of microcomputers between two clusters is accomplished by way of the switch network. Therefore, they are not as tightly coupled as microcomputers within a cluster because there will be more overhead in intercluster communication than intracluster communication. However, it was found that using specially designed controllers for the intercluster communication, the access time was increased by only 9%. This data is presented in Section III.E. Therefore, we can consider that microcomputers in different clusters within the same star are still tightly coupled. At the next higher level, the fourth level, several "stars" are connected together by linking nearest neighboring "stars" through a bus switch to form a "lattice network." The intercommunication between microcomputers from two stars are similar to that within a star, involving one central controller and two distributed controllers. The overhead is practically the same. Therefore, from the intercommunication viewpoint, microcomputers from two stars, and also throughout the systems, are practically all tightly coupled. However, through programming, they may be used either in tight coupling, loose coupling or any combinations in between to suit the requirements of the applications.

3. Intercommunication and Control

Because of the hierarchical structure of the architecture, the intercommunication processes and their controls are also hierarchical and are distributed. They are hierarchical because there are three levels of controls as shown in Table III.1.

At the lowest level of intracluster communication, no bus switch is needed. A Random Priority Controller (RPC) is used for arbitration. Only a small portion of the distributed controller is used, mainly to check if requests outside the cluster have been granted. At the next higher level of intercluster communication, the intrastar bus switch is used. Arbitration is accomplished by both distributed controller and RPC. Only a portion of central controller is used to grant the intercluster request. At the highest level, both interstar and intrastar bus switches may be used and all controllers, central, distributed and random priority, are in full action.

Further, the controllers are distributed because there are four identical RPC and distributed controllers, one in each cluster. Although there is only one central controller, it consists of four identical units, one for each cluster. The advantages of such a distributed control system are: (1) Parallel control actions which enhance the speed of "request arbitration." (2) Improved fault tolerance because the control actions are shared between four separate units

Table III.1 Componensts Used In Distributed Intercommunication and Control

Communication	Bus Switch	Central Controller	Distributed Controller	Random Priority Controller
Inter-Star	ISBSW SBSW	X	X	X
Intra-Star/Inter-Cluster	SBSW	X	X	X
Intra-Star/Intra-Cluster			X	X

ISBSW : Inter-Star Bus Switch

SBSW : Star Bus Switch (Intra-Star)

and should one malfunction, the other three can still continue their functions.

4. Hardware Implementation of Controllers

Controller circuits can be implemented in several ways:

a. Microprocessor control
b. Bit slice processor control
c. Digital logic circuit control.

Two performance characteristics should be considered in their choice and design: programmability and speed. The microprocessor approach has the most versatile programmability but the slowest speed. The digital hardware approach has very limited programmability but the fastest speed of the three. An estimate has been made to compare their speeds.

In our design, the primary goal is to offer the fastest response and arbitration of requests and communication speed. Therefore, we chose the digital logic circuit approach. Great care was given to the design of controller concepts and circuits, to avoid unexpected changes. Further, Schottky and low power Schottky chips are used due to their speed and power trade-offs. CMOS chips were found to be too slow and do not have adequate driving capability.

5. Priority Resolver

There are several ways to arbitrate multiple requests or to resolve priorities:

Fixed priority
  Serial (Daisy chain)
  Parallel
Rotating priority
FIFO
Random priority

There are two primary requirements for a priority resolver circuit: uniform and fast resolution of bus requests. In this system, an Intel Multibus is used as the system bus with 10 MHZ bus clock frequency. We decided to design a priority resolver circuit to arbitrate 8 SBCs within one bus clock.

The fixed priority approach was not selected because it was unable to arbitrate multiple bus requests and grant their usages uniformly. Test results showed that in our tightly coupled environments, two SBCs are able to share the bus adequately. More than two SBCs produce unacceptable delays.

Rotating priority is much faster than the fixed priority approach. It is able to arbitrate multiple requests and does grant their bus usages uniformly. However, it was not our final choice because the random priority approach was found to be faster. This is because in the rotating priority approach, every bus request line is tested serially (in a rotating manner) whether there are request signals on these lines or not. In the worst case, the rotating priority resolver grants the bus after N searches, where N is the number of SBCs being arbitrated by the resolver.

First in-first out (FIFO) is a resolver approach which requires memory. Because of the time needed to reference the memory, it is not possible to build a FIFO resolver to arbitrate 8 bus requests within 100 nsec, the bus clock period. With current technology, a fast FIFO arbiter probably requires more than 300 nsec.

The random priority resolver is designed based on the binary tree synchronous selector concept. Consider our case of 8 SBCs in a cluster. Three-stage selection is used. During the first stage, four out of eight lines are checked simultaneously. In the second stage, two out of these four lines are checked simultaneously again. The final bus grant is made in the third stage. In other words, the time for searching and resolving the bus requests is $\log_2 N$, which is faster than the rotating priority resolver. Test results have shown that the random priority resolver is able to arbitrate multiple bus requests and grant their bus usages uniformly as demonstrated in Fig. 3.17. Four SBCs simultaneously sharing the bus in a tightly coupled environment are taken for the test case. These four SBCs were programmed to request the bus usage almost 100% of the time. The BPRN signals of these four SBCs are shown. A low signal of BPRN indicates that its SBC is using the system bus. The fact that none of these four traces showed any long periods of bus usage or bus waiting demonstrated that the random priority resolver is able to arbitrate very heavy bus requests by these four SBCs and grant

bus usage to them "uniformly." It was found that, on the average, a "bus request" is granted in about 60 nsec.

6. Bus Switches

Bus switches are one of the most important parts of a multiple microcomputer system because they provide the interconnection means among the processing resources. There are two aspects of the "bus switch" problem: bus switch network and the individual bus switch link.

Many switch networks have been investigated, some predated the computer developments [195]. A small number of them have been considered in the multiple microcomputer development: cross-bar, banyan, hyperconcentrator, simple ring, etc.

A combined approach was selected including two levels of switching networks because of the consideration of multi-task signal/data processing requirements in a typical system. At the higher level, many stars are interconnected in a lattice architecture. Interstar bus switches are provided between neighboring nodes. At the lower level, four clusters are included in each "star" node. They are interconnected by a "complete star bus switch" network. The complete star switch is chosen for two reasons. First, the coupling within a star should be as tight as possible. The complete star switch allows us to connect two clusters by the shortest link. Second, if a link failed, the complete switch gives us two choices to connect two clusters by way of a third cluster via two links, thus providing some fault tolerance.

The important part of the individual bus switch link is the switches themselves. For the Intel Multibus, we found that 58 of the 86 lines should be switched. There are several choices for the switches:

Bidirectional: MOS types of switches, such as CMOS, VMOS and DMOS.

Unidirectional: Bipolar types such as Schottky, low power Schottky and ECL types; Optoelectronic types.

Optoelectronic types of switches were not chosen because they are slow, on the order of 10 μsec. Very fast switching speeds on the order of several tens of nanosec are required because today's Multibus is running at 10 MHZ which corresponds to a clock period of only 100 nsec. CMOS, VMOS and DMOS switches could provide such switching speeds. However, they do not have enough driving capabilities for the 15 ma or more required by many of the control and address signals of the microcomputer. Therefore, these MOS switches were not chosen, although their bidirectional feature and the low power characteristics of the CMOS switches are extremely attractive and reliable. We chose the low power Schottky switches because of their speed and driving capability. A typical performance is shown in Fig. 3.15 which shows the waveform of an address signal before and after the switch. It can be seen that not only is the delay short, on the order of 25 nsec, but also the waveform is improved by the switch because of its good driving capability of up to 50 ma. It was tested with a minimum load resistor of 50 ohms and maximum capacities

of 270 pf and the switch continued to function satisfactorily up to 45 MHZ. One disadvantage is the need to use two back-to-back switch circuits for a bidirectional switching of each signal. Therefore, a special circuit was designed to provide not only the "enable" signal but also the "direction."

7. Processing Elements

There are two major types of processing elements on the system bus: general purpose microcomputers and special purpose processors which can further be separated into two subcategories. One is a special purpose processor like an array processor which can perform several signal processing operations such as fast Fourier transform, correlation, convolution, finite impulse filtering, infinite impulse filtering, etc. The second type is a special purpose processor which is designed to perform only one signal processing operation such as FFT.

a. General Purpose Microcomputer

It was decided that all general purpose microcomputers used in our system should be treated homogeneously. This is necessary because two major principles of our operating system are based on the "virtual processor" [189] and "dynamic process allocation" [190] concepts which require homogeneous processing elements.

b. Special Purpose Processors

It was decided that special purpose processors could not be treated in the same way as the microcomputers.

However, it has not been decided at this time exactly how these special purpose processors should be handled. There are two important alternatives. In one case, a special purpose processor is treated as an I/O port managed by the operating system. In the other case, a special purpose processor can be operated in a "slave" mode on the system bus.

8. Mode of Data Transfer

The basic mode of data transfer in most of the multiple processor systems is based on the "message transfer" communication. However, a basic philosophy of our operating system is the "loop free" structure which requires frequent synchronization primitive references. In other words, the operating system program on a microcomputer needs to reference synchronization primitives located in either internal or external global memories. These "references" are executed via the system bus. If the data transfer is "message" based, the synchronization of processes could be delayed because the system bus is being occupied by a long message transfer. In order to avoid such a delay, it was decided that the basic mode of data transfer should be based on the "word transfer." This allows several microcomputers to reference their synchronization primitives and other data in an "interleave" mode.

However, the transfer of data in "blocks" is possible if required. This is accomplished by a special feature of the Intel 16 bit 8086 microprocessor which can generate a bus lock signal of a duration specified by software. This bus

lock signal holds the bus for the completion of the block transfer. Thus, data transfer by "messages based communication" is possible as well.

C. DESCRIPTION OF THIS MULTIPLE MICROCOMPUTER SYSTEM

1. Introduction

In the last section, we have presented the reasons for choosing the specific approaches for various parts of our multiple microcomputer system based on a top-down design procedure to meet the requirements of this type of smart sensor systems. In this section, more detailed description will be given to explain how those choices are implemented. The presentation will be made in five major categories:

System architecture (Section C.2)

Processing resources (Section C.3)

Intercommunication network (Section C.4)

Intercommunication procedures among resources
in different clusters and stars (Section C.5)

Multibus communication (Section C.6)

Performance of this multiple microcomputer system will be presented in Section D.

2. System Architecture

The topology of this system consists of many "star" nodes interconnected by links to nearest neighbor stars. A two dimensional example is shown in Fig. 3.1. Each star has four links connected to its four neighbors. The links are bidirectional system buses with a bus switch, called

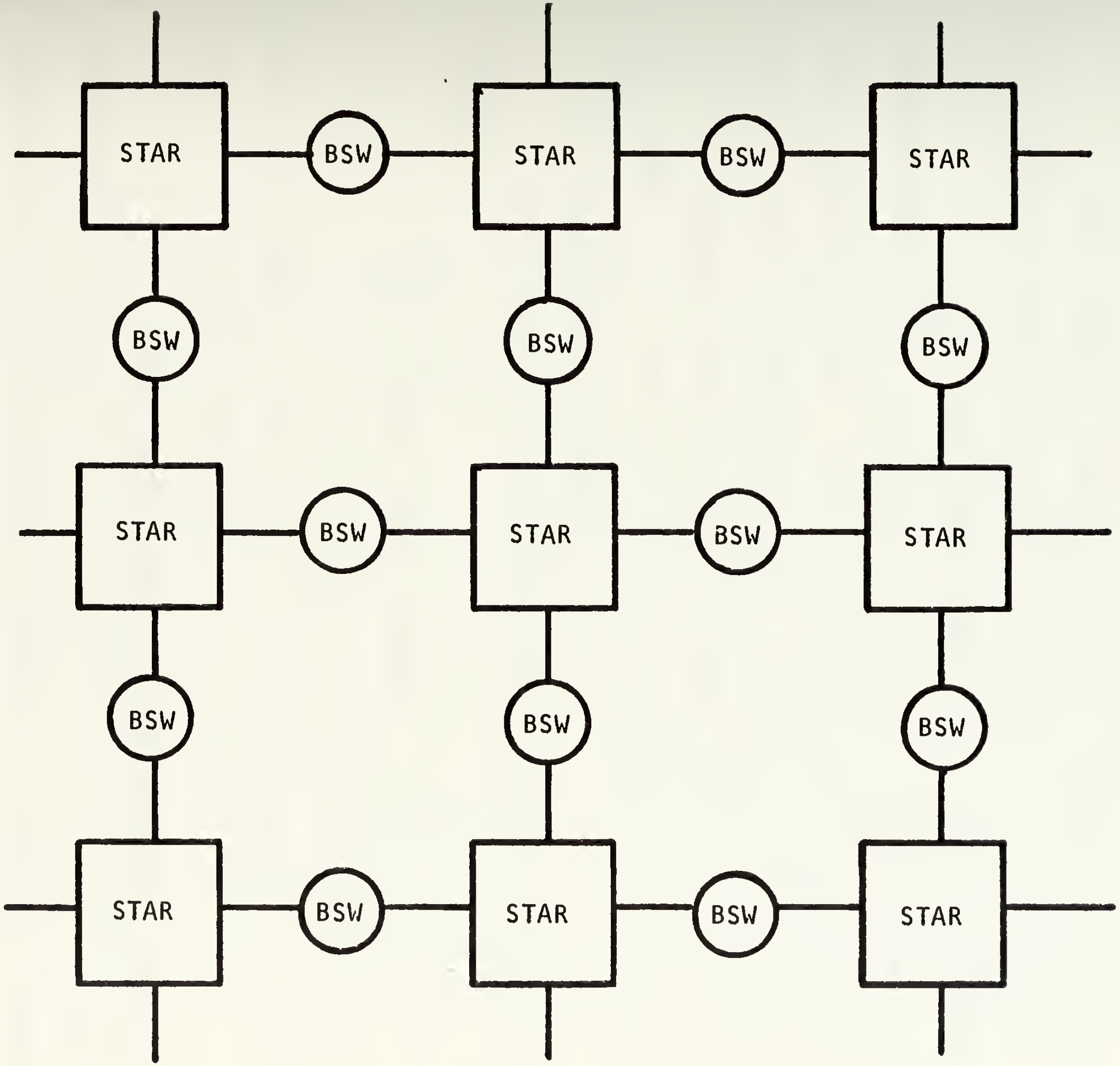


Figure 3.1 Two Dimensional Lattice Architecture of Multiple Star Multiple Microcomputer System

"inter-star bus switch" (ISBSW). The "bus switch" consists of 60 bidirectional switches for 60 signal lines. Two types of switches have been investigated: one with latches and one without latches for the signal lines.

Each "star" consists of four clusters interconnected by a complete star "bus-switch network." Each "cluster" consists of up to eight microcomputers. Other processing elements and one or more RAM boards are also connected onto the system Multibus. Fig. 3.2 depicts the topology of a single star with four clusters. In this example, the bus switch network consists of six bidirectional system buses, each with a bus switch interconnected as shown in Fig. 3.7.

3. Processing Resources

Two types of processing resources are used in this system.

a. Basic Processing Elements - SBC 8612A

Intel's 16 bit single board microcomputers, SBC 8612A, are used as the basic processing elements. A block diagram of the SBC 8612A is shown in Fig. 3.3.

(1) The Single Board Microcomputer SBC-8612A. The iSBC 8612A Single Board Computer is a 16 bit single board computer, a complete computer system on a single printed circuit assembly. The iSBC 8612A board includes a 16 bit central processing unit (CPU) up to 32K bytes of dynamic RAM, a serial communications interface, three programmable parallel I/O ports, three programmable timers, priority interrupt control,

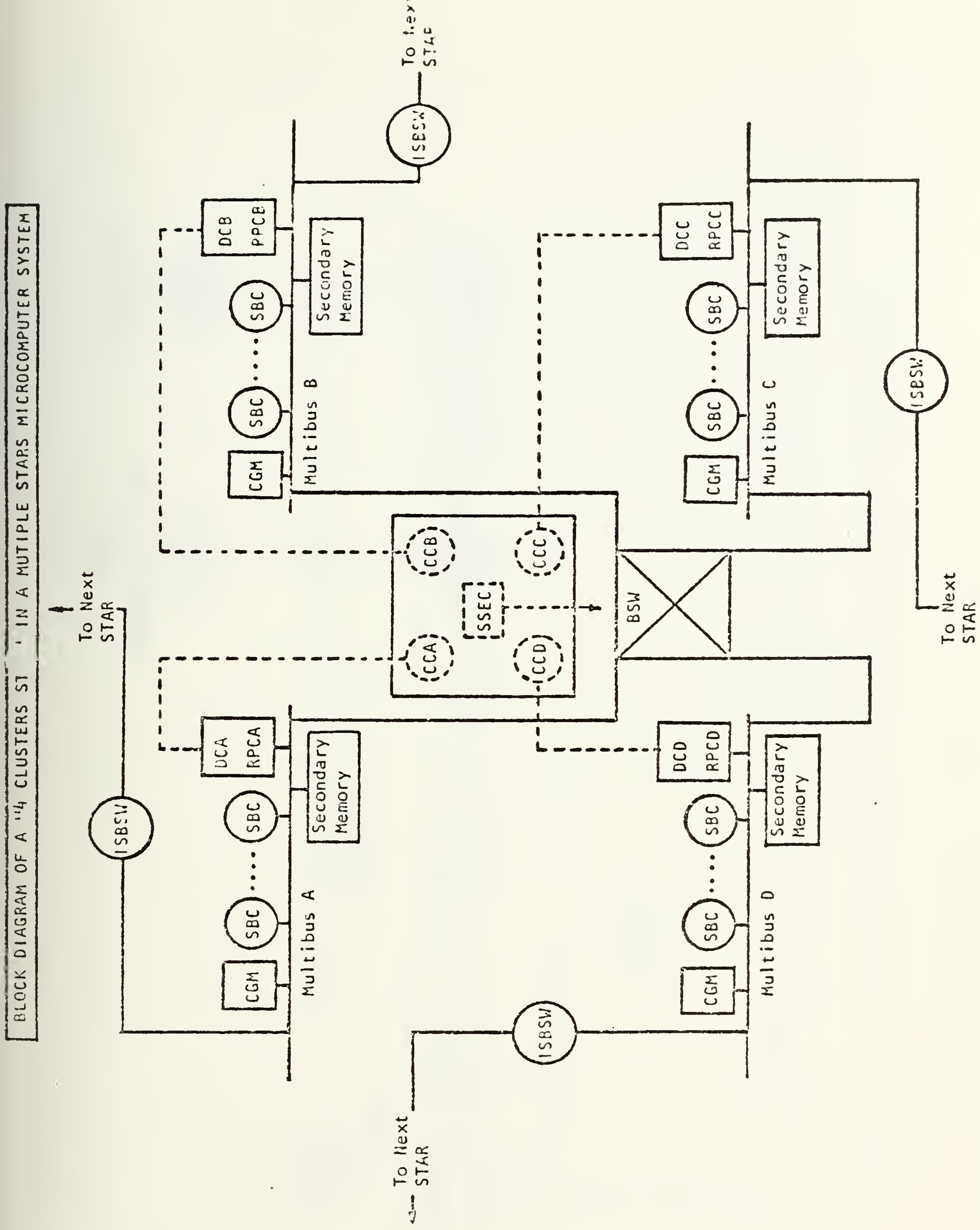


Figure 3.2 The topology of a single star.

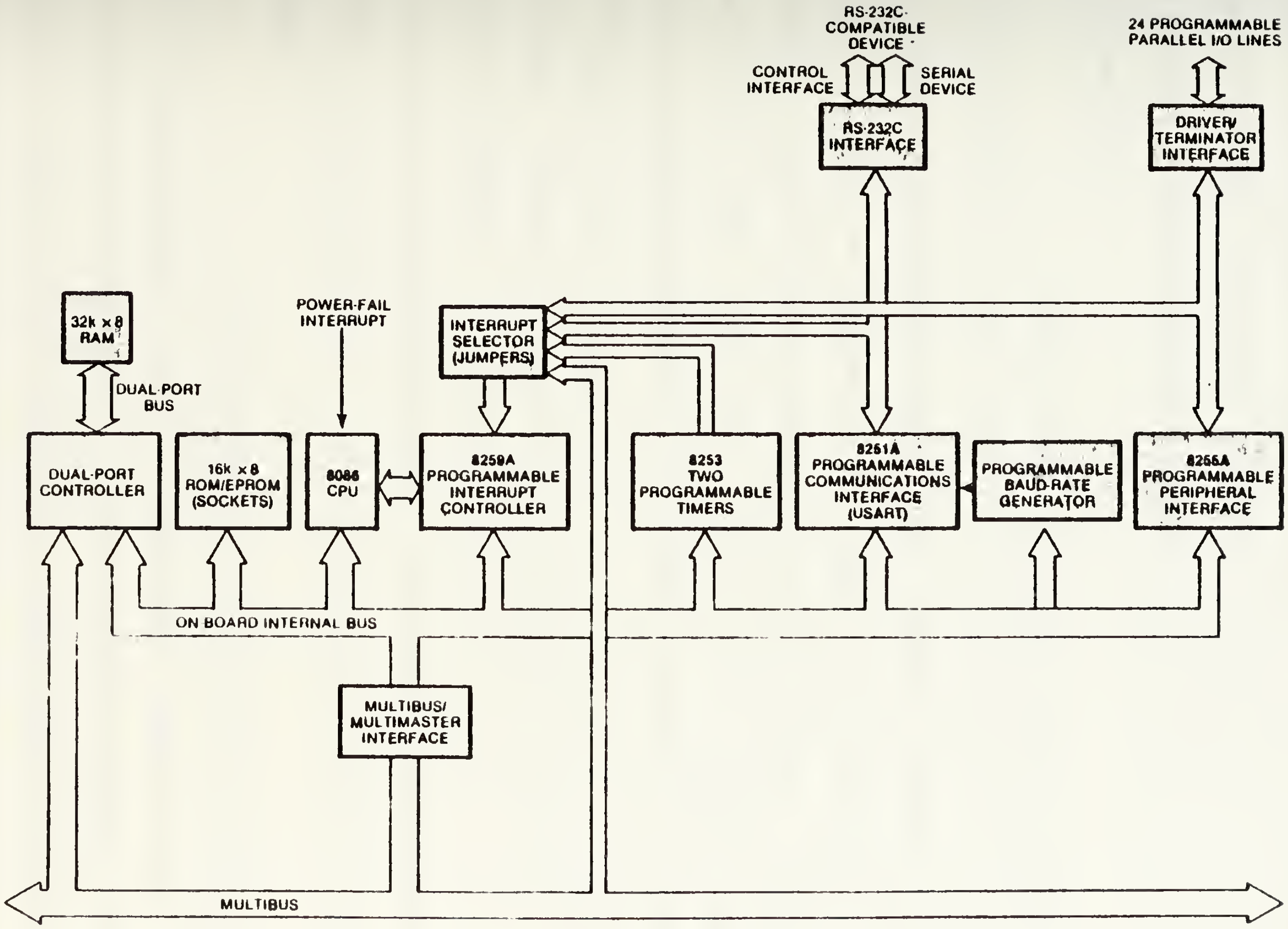


Figure 3.3 Architecture of the Intel Single Board Computer 8612

Multibus interface control logic, and bus expansion drivers for interface with other Multibus interface-compatible expansion boards. Also included is dual port control logic to allow the iSBC 8612A board to act as a slave RAM device to other Multibus interface masters in the system. Provision is made for user installation of up to 16K bytes of read only memory.

The iSBC 8612A Single Board Computer is controlled by an Intel 8086 16 bit microprocessor (CPU). The 8086 CPU includes four 16 bit general purpose registers that may also be addressed as eight 8 bit registers. In addition, the CPU contains two 16 bit pointer registers and two 16 bit index registers. Four 16 bit segment registers allow extended addressing to a full megabyte of memory. The CPU instruction set supports a wide range of addressing modes and data transfer operations, signed and unsigned 8 bit and 16 bit arithmetic including hardware multiply and divide, and logical and string operations. The CPU architecture features dynamic code relocation, reentrant code, and instruction look-ahead.

The iSBC 8612A board has an internal bus for all on-board memory and I/O operations and accesses the system bus (Multibus interface) for all external memory and I/O operations. Hence, local (on-board) operations do not involve the Multibus interface making the Multibus interface available for true parallel processing when several bus masters

(e.g., DMA devices and other single board computers) are used in a multimaster scheme.

Dual port control logic is included to interface the dynamic RAM with the Multibus interface so that the iSBC 8612A board can function as a slave RAM device when not in control of the Multibus interface. The CPU has priority when accessing on-board RAM. After the CPU completes its read or write operation, the controlling bus master is allowed to access RAM and complete its operation. Where both the CPU and the controlling bus master have the need to write or read several bytes or words to or from on-board RAM, their operations are interleaved. For CPU access, the on-board RAM addresses are assigned from the bottom up of the 1 megabyte address space; i.e., $00000\text{-}07FFF_H$. The slave RAM address decode logic includes jumpers and switches to allow positioning the on-board RAM into any 128K segment of the 1 megabyte system address space.

The slave RAM can be configured to allow either 8K, 16K, 24K, or 32K access by another bus master. If the iSBC 300 Multimodule RAM option is installed, the memory increments are 16K, 32K, 48K, or 64K. Thus, the RAM can be configured to allow other bus masters to access a segment of the on-board RAM and still reserve another segment strictly for on-board use. The addressing scheme accommodates both 16 bit and 20 bit addressing.

Four IC sockets are included to accommodate up to 16K bytes of user-installed read only memory. Configuration jumpers allow read only memory to be installed in 2K, 4K, or 8K increments.

The iSBC 8612A board includes 24 programmable parallel I/O lines implemented by means of an Intel 8255A Programmable Peripheral Interface (PPI). The system software is used to configure the I/O lines in any combination of unidirectional input/output and bidirectional ports. The I/O interface may be customized to meet specific peripheral requirements and, in order to take full advantage of the large number of possible I/O configurations, IC sockets are provided for interchangeable I/O line drivers and terminators. Hence, the flexibility of the parallel I/O interface is further enhanced by the capability of selecting the appropriate combination of optional line drivers and terminators to provide the required sink current, polarity, and drive/termination characteristics for each application. The 24 programmable I/O lines and signal ground lines are brought out to a 50 pin edge connector (J1) that mates with flat, woven, or round cable.

The RS232C compatible serial I/O port is controlled and interfaced by an Intel 8251A USART (Universal Synchronous/Asynchronous Receiver/Transmitter) chip. The USART is individually programmable for operation in most synchronous or asynchronous serial data transmission formats (including IBM Bi-Sync).

In the synchronous mode the following are programmable:

a. Character length,
b. Sync character (or characters), and
c. Parity.

In the asynchronous mode the following are programmable:

a. Character length,
b. Baud rate factor (clock divide ratios of 1, 16, or 64),
c. Stop bits, and .
d. Parity.

In both the synchronous and asynchronous modes, the serial I/O port features half- or full-duplex, double buffered transmit and receive capability. In addition, USART error detection circuits can check for parity, overrun, and framing errors. The USART transmit and receive clock rates are supplied by a programmable baud rate/time generator. These clocks may optionally be supplied from an external source. The RS232C command lines, serial data lines, and signal ground lines are brought out to a 50 pin edge connector (J2) that mates with flat or round cable.

Three independent, fully programmable 16 bit interval timer/event counters are provided by an Intel 8253 Programmable Interval Timer (PIT). Each counter is capable of operating in either BCD or binary modes; two of these counters are available to the system's designer to generate accurate time intervals under software control. Routing for the outputs and gate/trigger inputs of two of these counters

may be independently routed to the 8259A Programmable Interrupt Controller (PIC). The gate/trigger inputs of the two counters may be routed to I/O terminators associated with the 8255A PPI or as input connections from the 8255A PPI. The third counter is used as a programmable baud rate generator for the serial I/O port. In utilizing the iSBC 8612A board, the systems designer simply configures, via software, each counter independently to meet system requirements. Whenever a given time delay or count is needed, software commands to the 8253 PIT to select the desired function. The contents of each counter may be read at any time during system operation with simple operations for event counting applications, and special commands are included so that the contents of each counter can be read "on the fly."

The iSBC 8612A board provides vectoring for bus vectored (BV) and non-bus vectored (NBV) interrupts. An on-board Intel 8259A Programmable Interrupt Controller (PIC) handles up to eight NBV interrupts. By using external PICs slaved to the on-board PIC (master), the interrupt structure can be expanded to handle and resolve the priority of up to 64 BV sources.

The PIC, which can be programmed to respond to edge-sensitive or level-sensitive inputs, treats each "true" input signal condition as an interrupt request. After resolving the interrupt priority, the PIC issues a single interrupt request to the CPU. Interrupt priorities are

independently programmable under software control. The programmable interrupt priority modes are:

(a) Nested Priority. Each interrupt request has a fixed priority: input 0 is highest, input 7 is lowest.

(b) Fully Nested Priority. This mode is the same as the nested mode, except that when a slave PIC is being serviced, it is not locked out from the master PIC priority logic and when exiting from the interrupt service routine, the software must check for pending interrupts from the slave PIC just serviced.

(c) Auto-Rotating Priority. Each interrupt request has equal priority. Each level, after receiving service, becomes the lowest priority level until the next interrupt occurs.

(d) Specific Priority. Software assigns lowest priority. Priority of all other levels is in numerical sequence based on lowest priority.

(e) Special Mask. Interrupts at the level being serviced are inhibited, but all other levels of interrupts (higher and lower) are enabled.

(f) Poll. The CPU internal interrupt enable is disabled. Interrupt service is achieved by programmer initiative using a Poll command.

The CPU includes a non-maskable interrupt (NMI) and a maskable interrupt (INTR). The NMI interrupt is

intended to be used for catastrophic events such as power outages that require immediate action of the CPU. The INTR interrupt is driven by the 8259A PIC which, on demand, provides an 8 bit identifier of the interrupting source. The CPU multiplies the 8 bit identifier by four to derive a pointer to the service routine for the interrupting device.

Interrupt requests may originate from 18 sources without the necessity of external hardware. Two jumper-selectable interrupt requests can be automatically generated by the Programmable Peripheral Interface (PPI) when a byte of information is ready to be transferred to the 8086 CPU (i.e., input buffer is full) or a byte of information has been transferred to a peripheral device (i.e., output buffer is empty). Two jumper-selectable interrupt requests can be automatically generated by the USART when a character is ready to be transferred to the 8086 CPU (i.e., receive channel buffer is full) or when a character is ready to be transmitted (i.e., transmit channel data buffer is empty). A jumper-selectable interrupt request can be generated by two of the programmable counters and eight additional interrupt request lines are available to the user for direct interfaces to user designated peripheral devices via the Multibus interface. One interrupt request line may be jumper routed directly from a peripheral via the parallel I/O driver/terminator section and one power fail interrupt may be input via auxiliary connector P2.

The iSBC 8612A board includes the resources for supporting a variety of original equipment manufacturer system requirements. For those applications requiring additional processing capacity and the benefits of multiprocessing (i.e., several CPUs and/or controllers logically sharing systems tasks with communication over the Multibus interface), the iSBC 8612A board provides full bus arbitration control logic. This control logic allows up to three bus masters (e.g., combination of iSBC 8612A board, DMA controller, diskette controller, etc.) to share the Multibus interface in serial (daisy-chain) fashion or up to 16 bus masters to share the Multibus interface using an external parallel priority resolving network.

The Multibus interface arbitration logic operates synchronously with the bus clock, which is derived either from the iSBC 8612A board or can be optionally generated by some other bus master. Data, however, are transferred via a handshake between the controlling master and the addressed slave module. This arrangement allows different speed controllers to share resources on the same bus, and transfers via the bus proceed asynchronously. Thus, the transfer speed is dependent on transmitting and receiving devices only. This design prevents slower master modules from being handicapped in their attempts to gain control of the bus, but does not restrict the speed at which faster modules can transfer data via the same bus. The most obvious

applications for the master-slave capabilities of the bus are multiprocessor configurations, high speed direct memory access (DMA) operations, and high speed peripheral control, but are by no means limited to these three.

Adding the optional iSBC 300 Multimodule RAM to the iSBC 8612A board, allows the on-board RAM to be expanded by 32K (for an on-board total of 64K). If the optional iSBC 340 Multimodule EPROM is installed on the iSBC 8612A board, the amount of on-board ROM/EPROM can be expanded by 16K (for an on-board total of 32K).

b. Special Processing Elements

Special purpose processing elements will also be used in this system to enhance processing capabilities. Typical examples are array processors, FFT, correlators, etc. However, they have not been included in this thesis project.

c. Memories

Three types of memories are provided.

(1) Secondary Memory. It consists of two magnetic cartridge hard discs and a dual drive floppy diskette system. The magnetic hard disc is manufactured by the DYNEX Company and has a storage capacity of 10 megabytes. This hard disc system is connected to the system Multibus, thus allows fast data transfer rate and has DMA capability. Its interface to the Multibus is made by the Interphase Corp. The dual floppy diskette drive is a part of the Intel MDS-220 development system.

(2) Primary Memory. It consists of dynamic RAM and EPROM (Erasable Programmable Read Only Memory). The EPROMs reside in each SBC (8K byte to 16K byte per SBC). It can be used as the monitor storage, and to store part of the operating system. The RAMs reside in two types of physical locations. The first location is on each SBC and has a capacity up to 64K bytes. The second type of location is on separate RAM boards. A 128K byte RAM board developed by the MUPRO Company is used. The RAM in the SBC is a dual ported RAM which can be shared with other SBCs via the Multibus interface. Part or all of the dual ported RAM can be made accessible only to the on-board CPU; in other words, made "private" and "unshared" to the SBC. The stand-alone RAM boards are shared with other SBCs via the Multibus interface.

d. Memory Hierarchy

The primary memory of this type is partitioned according to the following hierarchical scheme.

A) Private Unshared Memory - RAMs available on each SBC which can be accessed only by the on-board CPU.

B) Internal Global Shared Memory - Internal global shared RAM available on each SBC and special RAM boards. The on-board RAM in the SBC is a dual ported RAM and can be accessed by any SBC which is a member of that cluster (unaccessible to PE in other clusters). See Section C.3.a.1.

C) External Global Shared Memory - External global shared RAMs reside in special RAM boards and/or in dual ported RAM of the SBCs. These memories can be accessed by any SBCs in the same "star," and any SBCs in the corresponding clusters in neighboring stars.

Using this memory hierarchy, the total address space can be expanded from the physical memory address space of each CPU. The 8086 microprocessor has 20 address lines so its physical address space is (2^{20}) = 1,048,576 bytes, or 1M bytes.

In this implementation, the total address space (memory space) for a single star is partitioned in the following way:

(1) Private Memory

6 μC in each cluster	8 μC in each cluster
$2 \cdot 65{,}536 + 4 \cdot (65{,}536 - 8{,}192)$	$4 \cdot 64K + 4 \cdot (64K - 8K)$
= 360,448 bytes/cluster	= 480K bytes/cluster
$2 \cdot 64K + 4 \cdot (64K - 8K)$	= 491,520 bytes/cluster
= 352 Kbytes/cluster[3]	

(2) Internal Global Memory

6 μC/CL	8 μC/CL
1 M bytes $\cdot \frac{3}{4}$	1 M bytes $\cdot \frac{3}{4}$
= 768K byte/cluster	= 768K byte/cluster
= 786,432 bytes/cluster	= 786,432 bytes/cluster

(3) External Global Memory

6 μC/CL	8 μC/CL
32K byte/cluster	32K bytes/cluster
= 32,768 bytes/cluster	= 32,768 bytes/cluster

As described before, a "star" consists of four clusters, thus the total memory space for a single star is:

[3] 1 K bytes = 1024 bytes.

6 μC/CL	8 μC/CL
$4 \cdot (352K + 768K + 32K)$	$4 \cdot (480K + 768K + 32K)$
= 4,608K bytes/star	= 5,120K bytes/star
= 4,718,592 bytes/star	= 5,242,880 bytes/star

This expanded memory space can be determined in general as:

MS = Memory space

CL = Number of clusters in a "star"

PM = Private memory. In K bytes.

GIM = Global internal memory. In K bytes.

GEM = Global external memory. In K bytes.

N = Number of SBCs.

$$MS = CL \cdot \sum_{i=1}^{N} PM_i + GIM + GEM \qquad (3.0)$$

If all SBCs are assigned the same amount of private memory, then (3.0) becomes

$$MS = CL \cdot (N \cdot PM + GIM + GEM) \qquad (3.1)$$

The reason for computing the memory space for 6 microcomputers and for 8 microcomputers in a cluster is mainly because of power supply considerations. The available power supply can handle up to 6 SBCs in a cluster. However, the controller for intercommunication is designed for 8 SBCs.

4. Intercommunication Network

In order to establish fast, reliable and high of fault toleran communication among SBCs of different clusters and stars, three level communication controllers

were designed, built and tested. They include a combination of random priority, distributed, and central controllers as shown in Fig. 3.4 for a single star. Each cluster has its own distributed controller. Each star has four such controllers. The four clusters share one central controller. The four distributed controllers are identical, and have some degree of programmability..

a. Distributed Controllers (DC)

A block diagram of the distributed controller is depicted in Fig. 3.5. It resides on a single board located in each cluster. Its primary functions are the following:

1) Arbitration among Internal/External bus requests from within and outside the cluster.
2) Priority resolving.
3) Inter-cluster advance activities monitoring.
4) Interacting with the central controller.
5) Deadlock avoidance.

b. Random Priority Controller (RPC)

The RPC is a bus contention resolver based on a binary tree approach. The RPC accepts up to eight "Bus Requests" (BREQ) and issues a single "Bus Priority In" (BPRN) signal. BREQ is a signal generated by the bus arbiter which resides on-board the SBC to indicate that this particular SBC requires the control of the cluster system bus (Multibus) for one or more data transfers. BPRN is a signal generated by the RPC to indicate to the requesting SBC that control of the

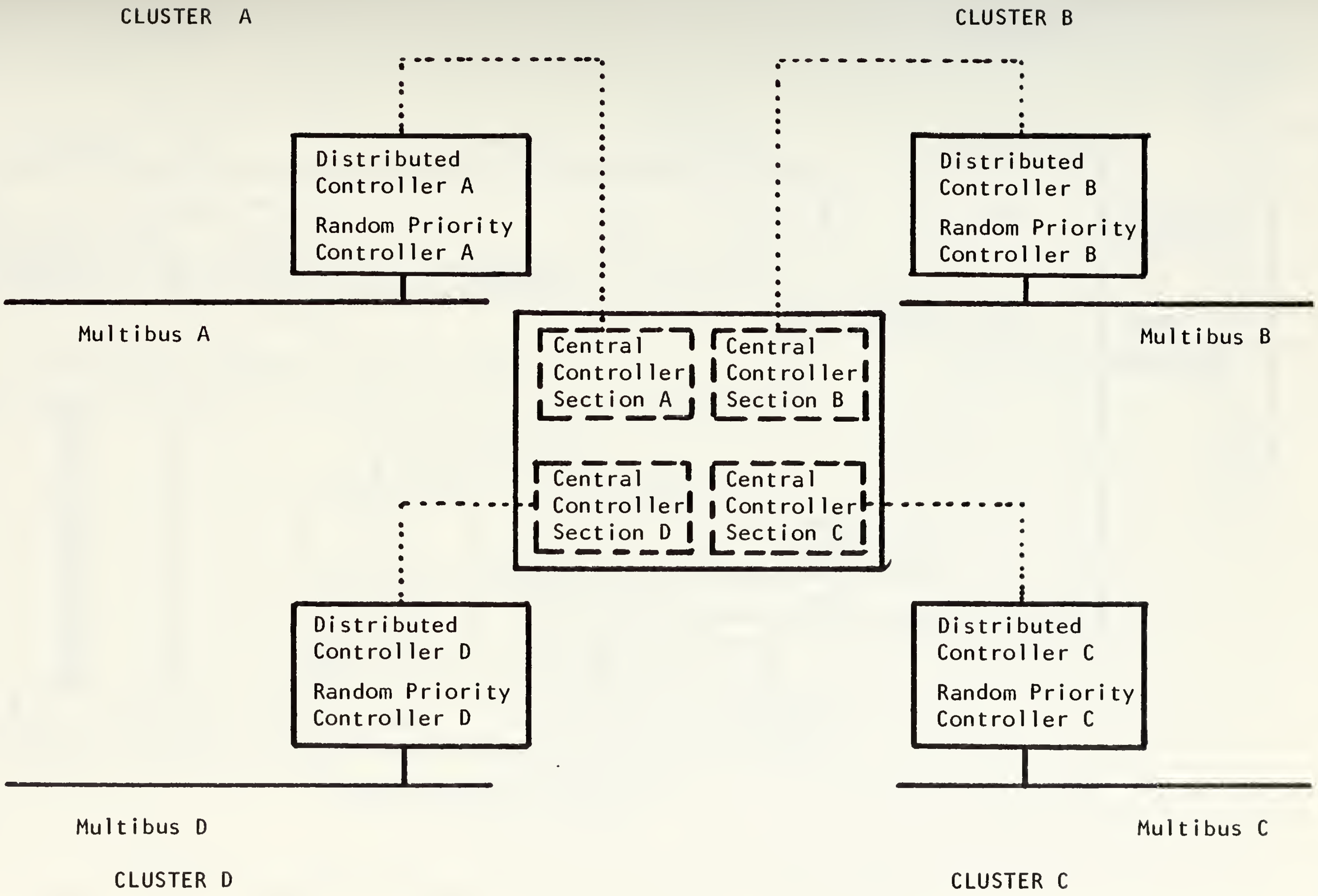

CLUSTER A
CLUSTER B
Distributed Controller A
Random Priority Controller A
Distributed Controller B
Random Priority Controller B
Multibus A
Multibus B
Central Controller Section A
Central Controller Section B
Central Controller Section D
Central Controller Section C
Distributed Controller D
Random Priority Controller D
Distributed Controller C
Random Priority Controller C
Multibus D
Multibus C
CLUSTER D
CLUSTER C

Figure 3.4 Diagram of a Three Level Control for a Four Clusters Multiple Microcomputer System

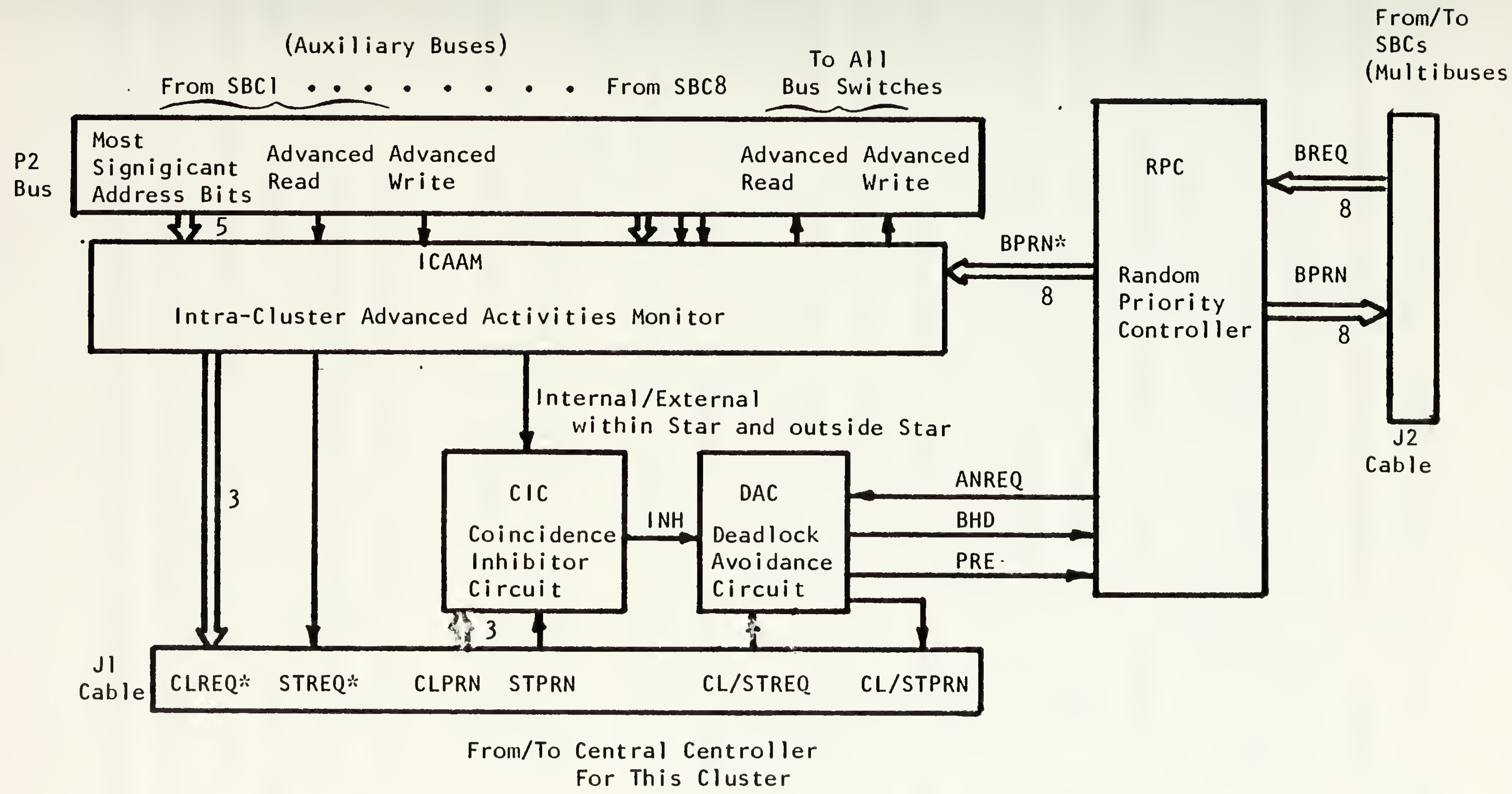

(Auxiliary Buses)
From SBC1 • • • • • • • • • From SBC8
To All Bus Switches
From/To SBCs (Multibuses
P2 Bus
Most Signigicant Address Bits
Advanced Read
Advanced Write
Advanced Read
Advanced Write
5
ICAAM
Intra-Cluster Advanced Activities Monitor
BPRN*
8
RPC
Random Priority Controller
BREQ
8
BPRN
8
J2 Cable
Internal/External within Star and outside Star
3
CIC
Coincidence Inhibitor Circuit
INH
DAC
Deadlock Avoidance Circuit
ANREQ
BHD
PRE
3
J1 Cable
CLREQ*
STREQ*
CLPRN
STPRN
CL/STREQ
CL/STPRN
From/To Central Centroller For This Cluster

Figure 3.5 Schematic Diagram of Distributed Controller

cluster bus is granted. Prior to issuance of a BPRN, the RPC generates an "advanced bus priority in" signal (intra-cluster advance activities monitor BPRN*) which is sent to the ICAAM as a "port selector" signal. This signal starts a chain of logical activities which eventually causes the DAC (deadlock avoidance circuit) to send two signals, i.e., BHD (bus hold) and PRE (priority enable) to the RPC. When the appropriate BHD and PRE are received by the RPC, it will generate the BPRN signal. BHD is a positive logic signal which enables the tristate output of the RPC to allow BPRN* to propagate and become a BPRN signal, when the PRE signal is enabled. If BHD goes low, it disables all PRN*. PRE is a negative logic signal which is generated in the DAC circuit. When the PRE signal is generated, it disables requests from other clusters and enables the output driver of the RPC to send the BPRN.

The RPC has an internal clock to synchronize its arbitration function. More details can be found in Section C.4.b.

ICAAM (Intra-Cluster Advance Activities Monitor) has a multiplexer which selects two signals, MSBT (most significant address bits, 5 bits out of 20) and ADRDC/ADWTC (advance read command/advance write command) when a BPRN* is received from the RPC. By analysing the MSBT, the ICAAM generates a bus request of one of the following types:

1) Intra-cluster bus request. It is a request for the system bus in the same cluster only. In response to this request, the ICAAM generates a IREQ signal.

2) Inter-cluster bus request. It is one out of four cluster requests generated by the ICAAM of the distributed controller. Each CLREQ* requests three resources: one system bus of the requesting cluster, one system bus of the requested cluster and one inter-connecting bus switch. Following a CLREQ*, the ICAAM also creates an EXREQ for the CIC (coincidence inhibit circuit).

3) Inter-star bus request. This request, labeled STREQ*, involves three resources: the system bus of a cluster in the requesting star, the system bus of the corresponding cluster in the requested star, and the inter-connecting bus switch between these two stars. Following a STREQ* signal, the ICAAM also creates an EXREQ for the CIC.

The ICAAM also generates an advanced read command (ADRDC) or advance write command (ADWTC) before the corresponding read command (MRDC) or write command (MWTC) is generated by the bus controller of the requesting SBC. This is done by monitoring the activities of the CPU of the requesting SBC before the CPU grants the system bus. Those signals are needed to determine the direction of the drivers in the bus switch in advance, so that all switching transients are settled before a data transfer takes place.

CIC (Coincidence Inhibiter Circuit) - The CIC accepts five signals as inputs: one STPRN (star priority in), three (cluster priority in) from the central controller and one IREQ/EXREQ from ICAAM. It generates one output signal INH (inhibit) for the DAC (deadlock avoidance circuit). The primary function of the CIC is to inhibit a BPRN from the RPC

in case that a CLREQ* or STREQ* were issued by the ICAAM, until either a CLPRN or a STPRN is granted by the central controller to the CIC. The necessity of this signal INH is to prevent the system bus to be tied down in waiting until the inter-cluster request is granted and allow efficient bus usage and reduce bus contention.

DAC (Deadlock Avoidance Circuit). A "deadlock" is a situation in which two processes are unknowingly waiting for resources that are held by each other and thus unavailable [192]. More details can be found in Section C.5.d.,e. The primary function of the DAC is to prevent deadlock. Its principle is similar to the "Suspend" Lock method [Ref. 193]. The DAC accepts four input signals: ANREQ (any request), INH, STREQ, CLREQ and generates three signals: BHD (bus hold), PRE (priority enable) and CL/STPRN. Three cases will be described to explain the operations of DAC depending on the occurrence of either the CLREQ (or STREQ) and the INH signals.

(Case 1) - A CLREQ (or STREQ) occurs prior to the INH signal, the CL/STPRN signal will be granted. In this case, BHD will go low and PRE high, thus freezing the selected request in the RPC, disabling the BPRN* which will release all the resources held by the appropriate SBC via the BPRN* signal (ICAAM, CCU-I). About 30 nsec later, a CL/STPRN will be generated by the DAC. This allows the appropriate processing element to grant the system bus.

(Case 2) - A CLREQ (or STREQ) signal occurs after the INH signal, the CL/STPRN signal will be blocked. It indicates that the system bus is in use. In this case, BHD is high and PRE goes low, BPRN will be granted.

(Case 3) - If the INH signal and CLREQ (STREQ) signal occur simultaneously within a time window of 15 nsec, the CLREQ (or STREQ) signal will be blocked as before. In case of any occurrence of a transient CL/STPRN signal, the "GLITCH KILLER" will suppress it and prevent the transient from propagating to the central controller.

c. Central Controller (CC)

The central controller is a single board controller, which consists of two clocks and four identical units, each corresponding to one cluster in the star. The primary functions of the CC are:

1) To arbitrate among different CLREQ and STREQ to a single cluster.

2) Enable and disable the CL/STPRN signal chain.

3) Enable and disable the appropriate bus switch links of the complete star switch.

A block diagram of the CC is presented in Fig. 3.6.

CLK-1 - Clock 1 is the main clock of the central controller, Its frequency is 30 MHZ. It is used to synchronize and enable the arbitration function of the CSRA (cluster/star request arbitor) and the four-phase clock, CLK-2.

CLK-2 - Clock 2 is a four-phase, anti-coincidence clock. Its input is CLK-1 which generates four clocks, one

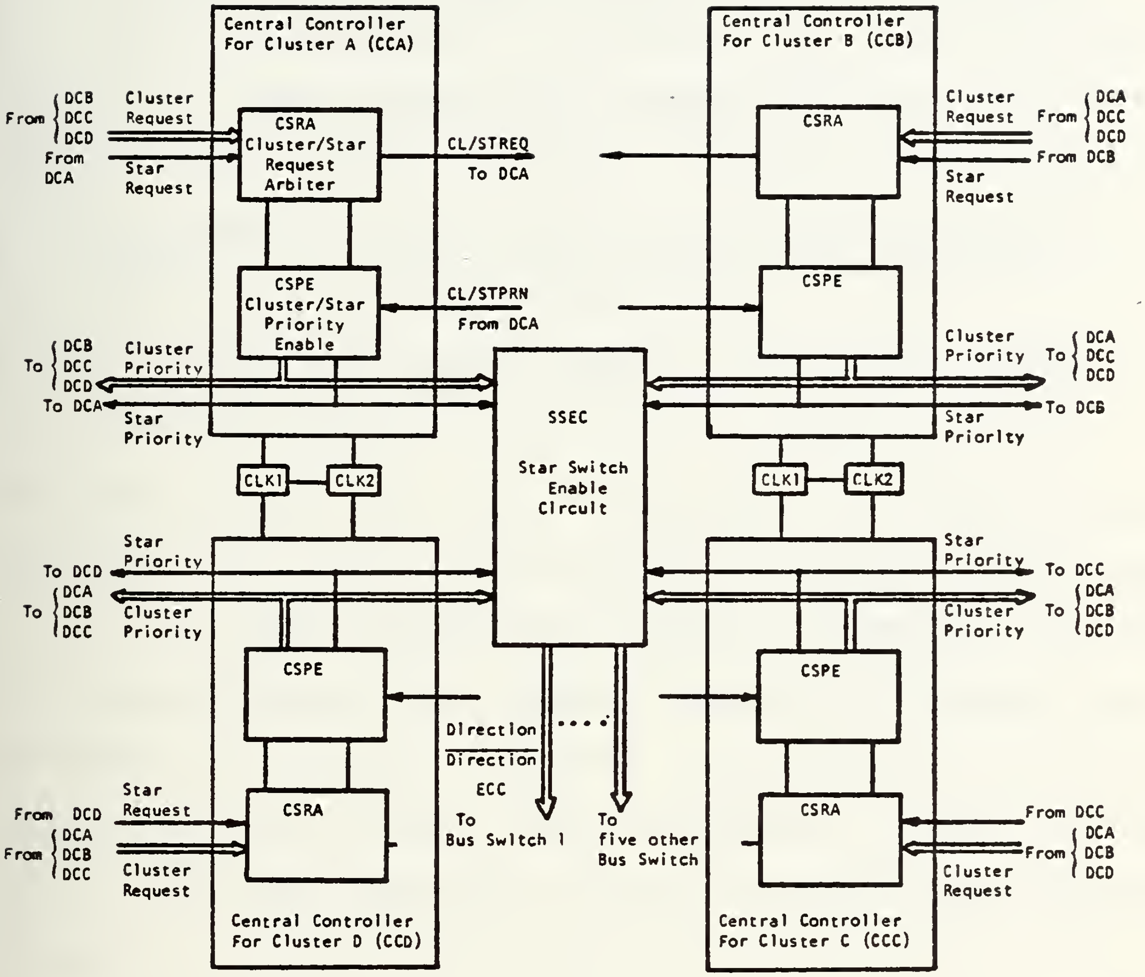


Figure 3.6 A block diagram of the central controller.

each for four CSRAs. The functions of the four-phase clock are:

1) To synchronize the CLREQ (or STREQ) chain action via the CSRA in order to prevent deadlocks. The deadlock avoidance method used in this implementation is similar to the "spinning lock" method [192]. The spinning lock is rotating at a frequency of 3.75 MHZ (30/8 MHZ).

CSRA (Cluster/Star Request Arbiter) - The CSRA is a rotating priority resolver. Its primary functions are:

1) To arbitrate among requests from three other clusters within the same star and from the corresponding cluster in the neighboring star.

2) To enable the selected request, after being synchronized with the spinning lock, to propagate to the requested cluster.

The CSRA accepts four different requests to a single cluster and grants one of them according to a rotating priority scheme.

CSPE (Cluster/Star Priority In Enable) - the CSPE is a demultiplexer whose primary function is to enable the CL/STPRN chain action. The CSPE is synchronized by the CSRA. When a CLPRN is received from the requested cluster, the CSPE will enable the CLPRN chain action to the selected requesting cluster.

SSEC (Star Switch Enable Circuit) - The SSEC consists of a set of six drivers. It accepts the different CLPRNs and generates two signals, ECC, DIR, $\overline{\text{DIR}}$. ECC is a negative logic signal which enables one of the bus switch links corresponding to the CLPRN signal. DIR is a signal which sets the requesting direction of the drivers in the selected link of the "complete star" bus switch. $\overline{\text{DIR}}$ is

the inverted DIR signal. The SSEC is responsible for the enabling of the six different links of the complete star bus switch as depicted in Fig. 3.7.

5. Intercommunication Procedures Among Resources

Communication among the resources of this system is governed by the following basic concepts: Explicitly segmented memory; unshared local and shared global internal/external memory hierarchy, asynchronous process structure and a design decision that each single board computer is allowed to use the system bus for transfer of only one word of data and then must release the system bus to other SBCs except when a prefix lock is executed by software. A software lock will grant the bus to that SBC for any length of time needed by that SBC. In general, this feature is not required frequently so the operating system will not normally be delayed waiting for the system bus to be released in order to test a semaphore, or any other synchronization primitives.

In order to provide effective communication among all processing elements (within a single cluster, among different clusters in a single "star," and among "stars") and to arbitrate the contention of bus usage (in star bus switch and inter-star bus switches), we have developed an intercommunications system managed by distributed and central controllers, as described in Chapter III.D.4.,5.

In order to describe the communication protocol among different SBCs, a two "star" system is chosen - STAR-1, STAR-2

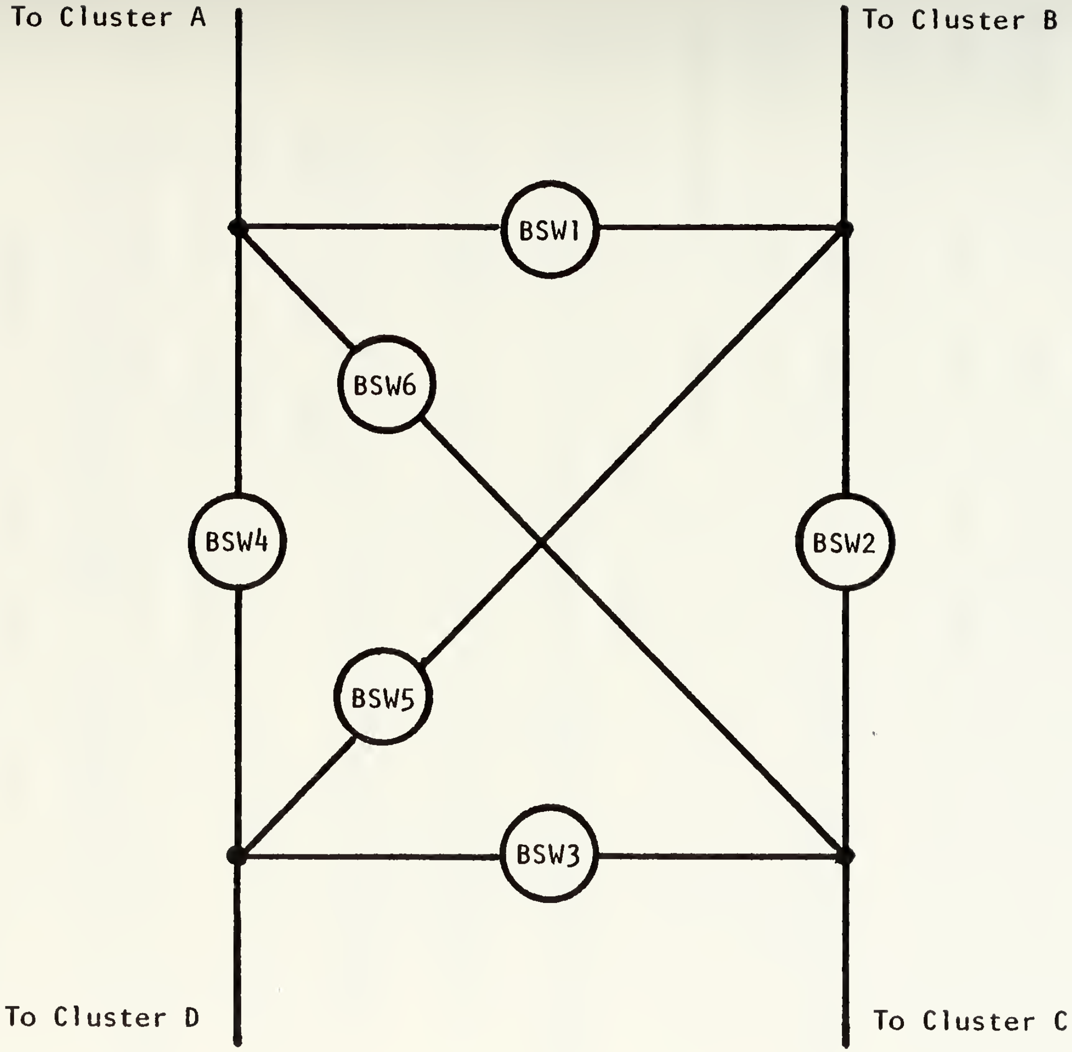


Figure 3.7 Diagram for the "Complete Star" Bus Switch Network
(Example: 4 Clusters, 6 Bus Switches)

as depicted in Fig. 3.8. Several examples of different types of communication are presented.

a. Example #1 - Intra-Cluster Communication

Intra-cluster communication is accomplished by means of data transfer via the cluster Multibus. This type of communication does not involve the central controller or any bus switch. The distributed controller resident in the specific cluster and on-board SBCs are the controllers of this communication link.

For example, let us assume SBC-1 in cluster A1 requests some information from SBC-2 in the same cluster. The sequence of events (Fig. 3.9) is:

a) SBC-1 generates BREQ signal.

b) The RPC of the distributed controller will grant the request and generates a BPRN* signal.

c) The ICAAM of the distributed controller will generate an IREQ signal, for the inhibiter.

d) From the IREQ, the "IHC" generates an inhibit signal which causes the DAC to send appropriate BHD and PRE signals.

e) These two signals are sent to the RPC to close the chain and a BPRN is generated.

f) The BPRN signal is applied to the arbiter circuit of the corresponding SBC. From this point, a regular Multibus transfer is executed.

These six events are necessary to establish any intra-cluster communication. But they are not sufficient. The following conditions corresponding to the requests from other clusters and stars must be examined:

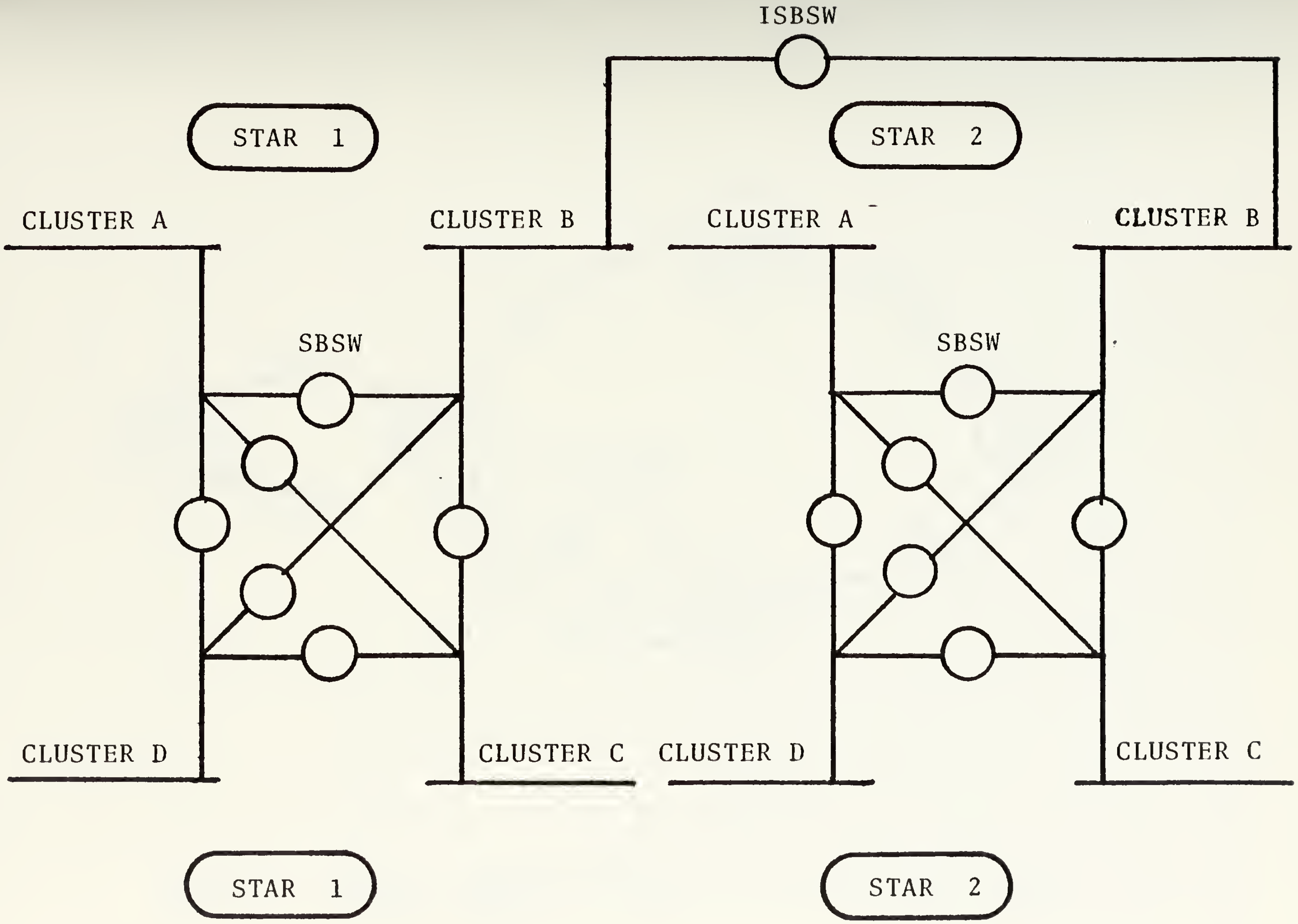


Figure 3.8 Diagram Showing Inter-Star and Intra-Star Interconnections Using Bus Switches

ISBSW : Inter-Star Bus Switch
SBSW : Star Bus Switch (Intra-Star)

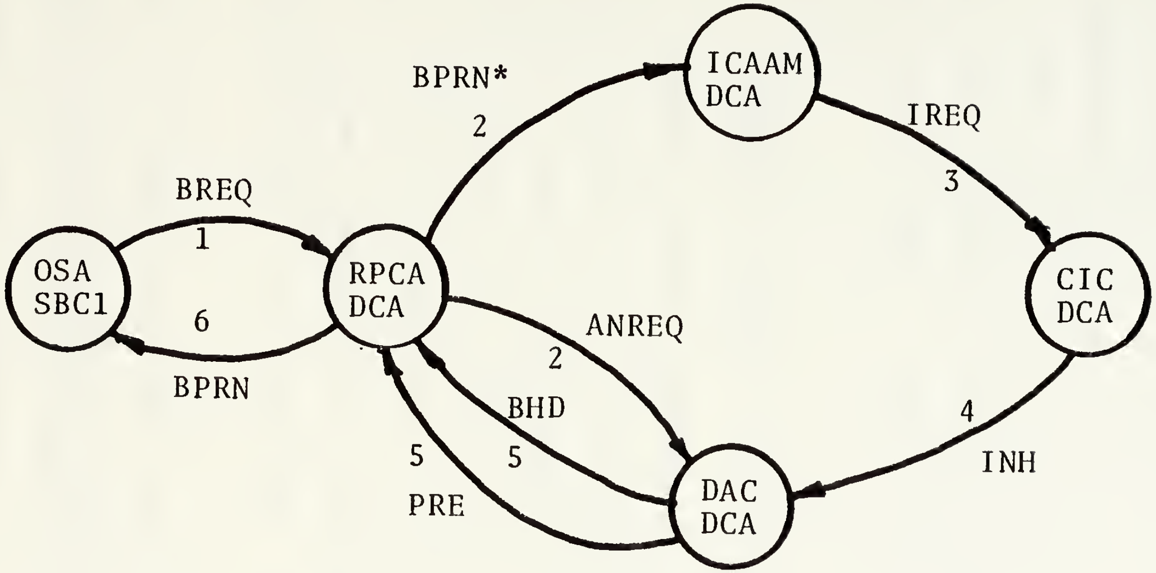


Figure 3.9 State Diagram of Intra-Cluster Communication (Intra-Star)

1) Is there any other cluster in process of communication with this cluster?

2) Is there any other star in process of communication with this cluster?

For simplicity of this example, we assumed that no external requests were involved in the process of intracluster communication.

Upon termination of the data transfer via the system bus, SBC-1 releases its BREQ signal which releases all sources held by SBC-1. The average time of word transfer is 1.65 μsec.

b. Example 2 - Inter-Cluster Communication (within a Star)

Inter-cluster communication is accomplished by means of data transfer via two clusters' system buses (Multibus) and the bus switch interconnecting those two clusters. This type of communication involves all controllers, the star bus switch, and the on-board SBC arbiter. (See Fig. 3.10).

Assume that SBC-1 in cluster A1 requests some information from SBC-1 in cluster B1. The sequence of events is:

1) SBC-1 of A1 generates BREQ signal.

2) The RPC of the distributed controller in cluster A1 locks on the request and generates a BPRN* signal.

3) The BPRN* signal is applied to the ICAAM of the distributed controller.

4) The ICAAM generates two signals: CLREQ-B1, which propagates to the rotating priority arbiter of the central controller unit B and "EXREQ" which is applied to the "CIC" coincidence inhibiter of the distributed controller of cluster A1.

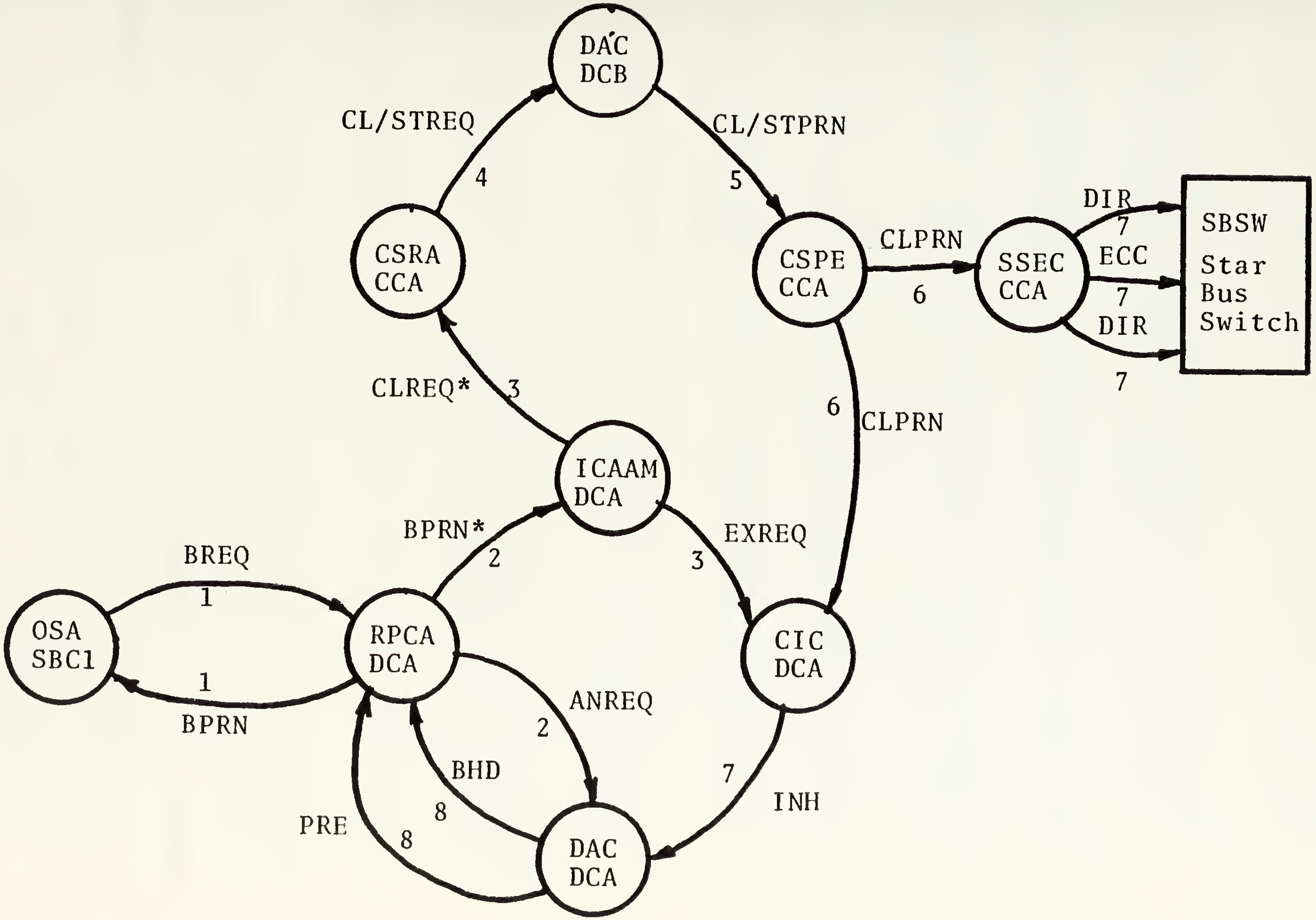


Figure 3.10 State Diagram of Inter-Cluster Communication (Intra-Star)

5) The "CIC" coincidence inhibiter generates an appropriate INH signal which will cause the distributed controller in cluster A to wait for a CLPRN from the demultiplexer of the central controller, unit B.

6) The "cluster/star request arbiter" in the central controller locks on the CLREQ-B1 signal and waits for the spinning lock to enable the CLREQ chain action and locks on the request.

7) The CLREQ signal is applied to the DAC of the distributed controller of cluster B1.

8) The DAC of the distributed controller of cluster B1 generates a CLPRN signal which is applied to the demultiplexer of unit B of the central controller.

9) The central controller enables the CLPRN signal to the "DAC" of the distributed controller in cluster A which generates appropriate BHD and PRE signals.

10) The BHD and PRE signals are applied to the ROC and closes the chain action. The RPC then generates the BPRN signal.

11) The BPRN signal is applied to the on-board SBC-1 arbiter which starts the regular Multibus communication.

12) After the event #9, a parallel process is initialized. This process is the bus switch enable. Two signals, DIR and ECC, are sent to the bus switch which links the buses of cluster A1 and cluster B1.

13) Those two signals prepare the switch for the coming data transfer.

The initialization of the bus switch terminates 200 nsec before the transfer of data via the bus (switch). This feature makes the bus switch transparent to the requesting cluster, and both clusters are linked on a longer system bus for the time the transfer takes place. SBC 1 in cluster A1 can use the "longer" system bus (two system buses and the plus switch) for more than one word transfer, if this feature

is requested by a software bus lock instruction from SBC 1. Termination of this process is started by releasing the BREQ signal by SBC-1 of cluster A1. This event releases all resources held by SBC 1 of cluster A1.

The sequence of events described in this example is necessary for this type of communication. Other external events were not introduced in order to simplify the example. This sequence of events takes place in an average time of 2.1 μsec.

c. Example #3 - Inter-Star Communication

Inter-star communication is accomplished by means of data transfer via the system buses of two clusters and the bus switch interconnecting these two clusters. This type of communication involves all controllers, and the bus switch interconnecting the two clusters. The sequence of events is similar to the previous example. Instead of the CLREQ signal, a STREQ signal is applied to the central controller. The responding signal is STPRN. (See Fig. 3.11).

Examples 1, 2, and 3 described a case of separable communication levels. In a real application, the situation can be more complicated. For example, a simultaneous combination of the three different examples is possible. In such a case, deadlocks could occur frequently [193]. In order to prevent those deadlocks, two methods of deadlock avoidance are used.

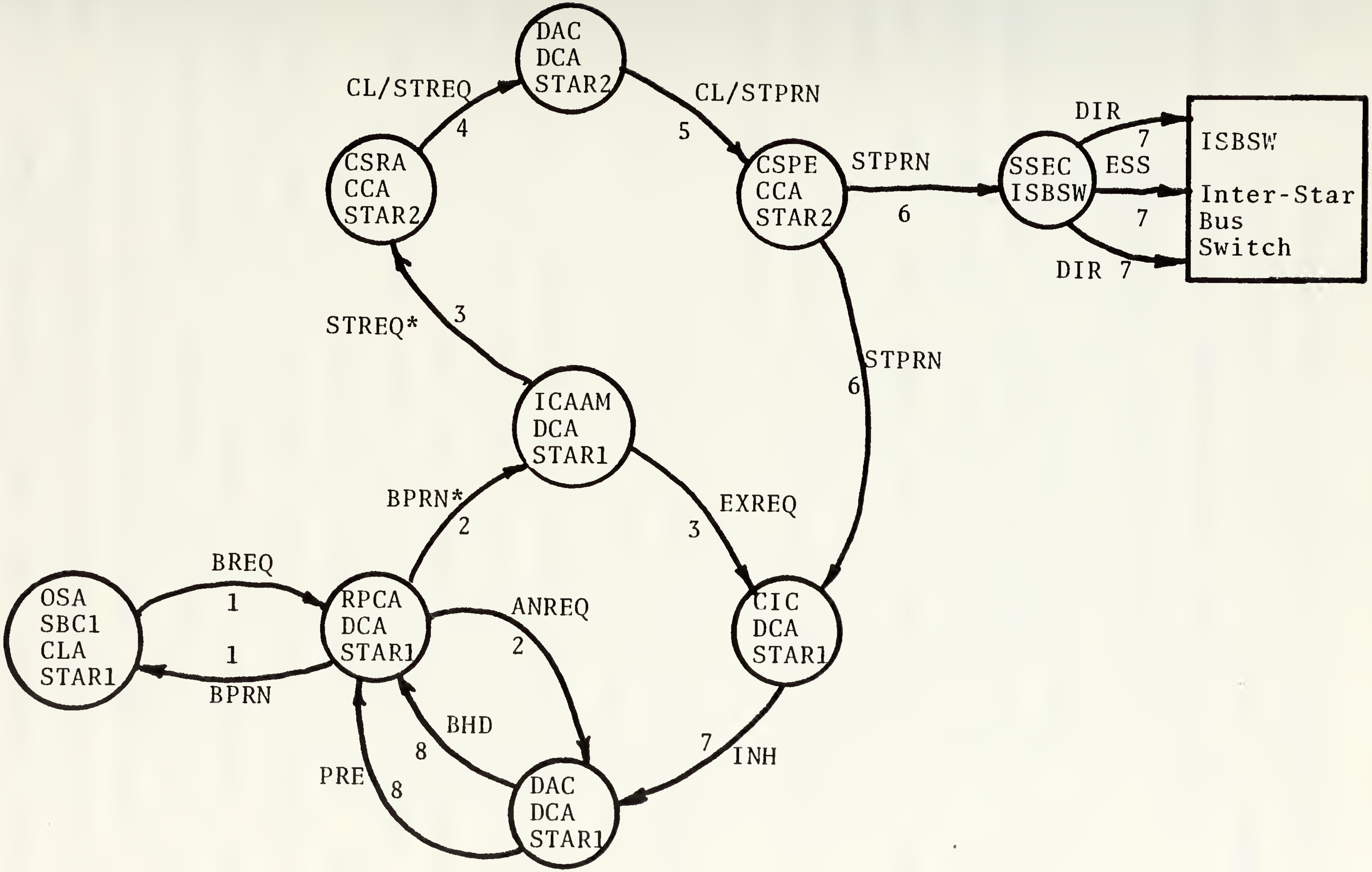


Figure 3.11 State Diagram of Inter-Star Communication

"Suspend Lock" - This method is implemented in the DAC of the distributed controller. In order to explain how this method works, the following example is used.

d. Example #4 - Deadlock Avoidance I - Suspend Lock

SBC-i in cluster A1 of star 1 requests SBC-j in cluster A2 of star 2 (process P1, and SBC-k in cluster A2 of star-2 requests SBC-ℓ in cluster A1 of star 1 (process P2), $\{8 \geq i,j,k,\ell \geq 1\}$. Let's assume that in time T_o the two request processes P1 and P2 progress to state No. 3 (Fig. 3.12). At this point of execution, the processes P1, P2 are holding the following resources:

P1: {RPC-DC-A1, ICAAM-DC-A1, CSRA/CCB1, DAC-A1, CIC-A1}

P2: {RPC-DC-A2, ICAAM-DC-A2, CSRA/CCA2, DAC-A2, CIC-A2}

At this point of execution, each process requests the DAC located in the other distributed controller. But the two DACs are held by the requesting processes and are unavailable. It seems that we have a deadly embrace situation (deadlock).

The DAC is designed to avoid such a case. One of the DAC (which will be called the first DAC depending upon the time of arrival of the requests) will suspend the lock of the second DAC, by releasing some of the resources that are held by the second requesting process. This way the first requesting process will be advanced while the second will be suspended and wait for the first process to terminate. This deadlock could happen if the suspend lock method is not

used when the two requesting clusters are located in different stars because the two spinning locks of the two central controllers are not synchronized. Therefore, the spinning lock function is limited for inter-star communication. This is the reason for having two types of deadlock avoidance methods. The suspend lock method is used to prevent deadlock for inter-star communication. The issue of synchronizing the spinning locks of the different central controllers of a multi-star system is not desirable for fault tolerance, and sometimes it may not be possible to synchronize them.

The second method of deadlock avoidance is the "spinning lock" method. This method is used to prevent deadlocks which may occur in inter-cluster or intra-cluster communication within the same star. If for any reason this method fails to prevent a deadlock, the "suspend lock" method will take over and prevent the deadlock. The reason for using two different methods is to reduce the overhead created by the suspend method and to increase fault tolerance.

CLK-2 in the central controller is a four-phase anti-coincidence clock as shown in Fig. 3.22. This clock is the "spinning lock" generator.

e. Example #5 - Deadlock Avoidance II - Spinning Lock (Fig. 3.12)

Let us assume that SBC-i in cluster A requests SBC-j in cluster B and SBC-k in cluster B requests SBC-ℓ in cluster A. These requests are all for SBCs residing in the same "star." If the two requests are sent simultaneously to

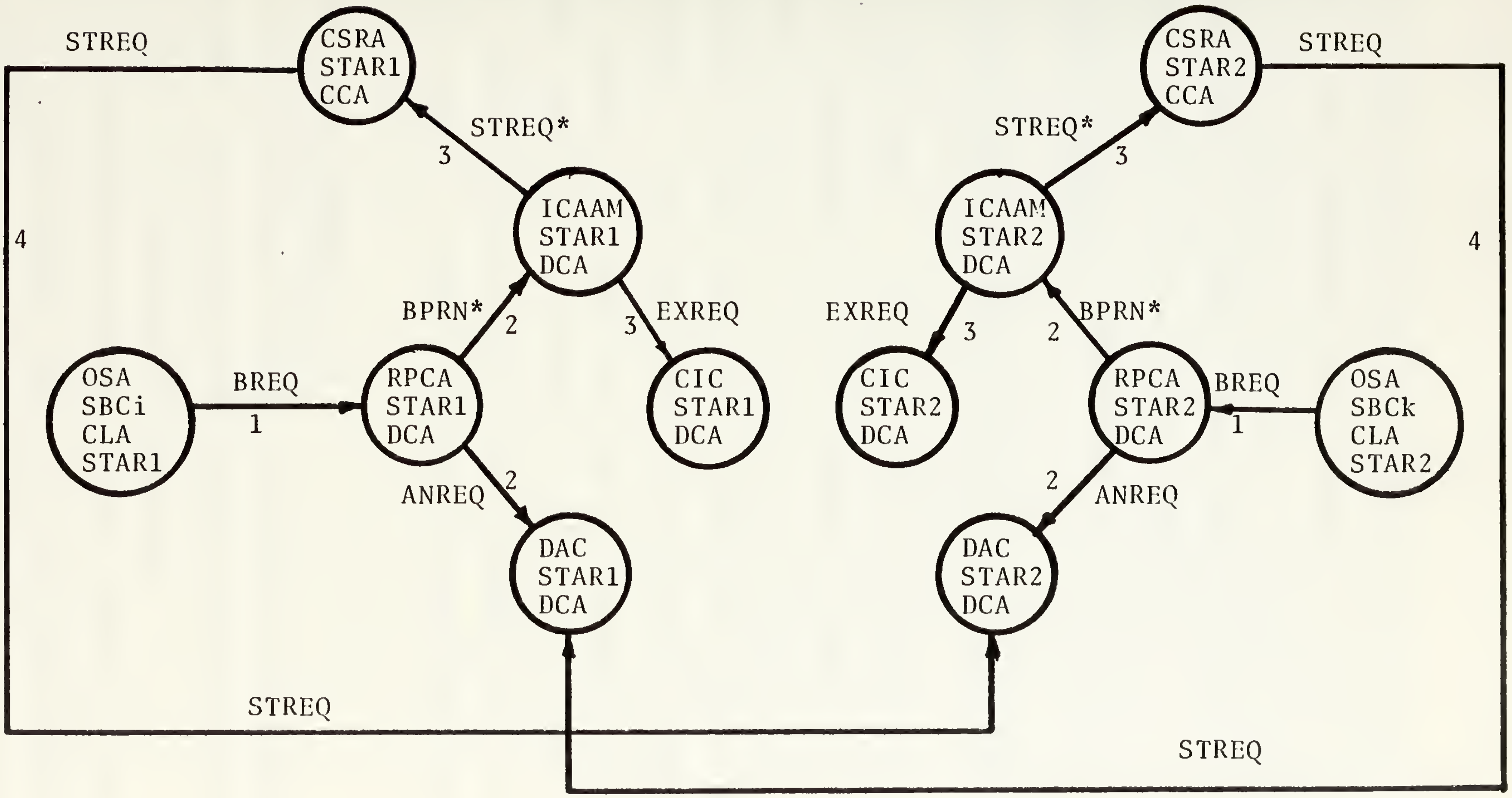


Figure 3.12 State Diagram for A Deadlock Example

the CSRA of CCA and CSRA of CCB, respectively, of the central controller, they eventually will progress to the deadlock condition as explained in Example #4. In order to prevent such possibility, the CSRA of the central controller is designed with two "lock in request" phases.

1) The first phase is implemented by the rotating priority arbiter.

2) The request selected by the first arbiter propagates to the "spinning lock" circuit which will lock on the request only when CLK-2 goes low.

CLK has four phases. Since only one goes low at any given time, it is impossible for both requests to leave the central controller at the same time to the distributed controller of the requested cluster and thus eliminates the race condition and deadlock. A race condition occurs when the scheduling of two processes is so critical that the various orders of scheduling them result in different processing [192]. The minimum time difference caused by the spinning lock to the requesting process is equal to the anti-coincidence time t_{ac} of CLK-2 (Fig. 3.22).

6. Multibus Communication

Two arbitration circuits are used in the Multibus communication: the on-board SBC arbiter called Bus Arbiter and the RPC of the distributed controller.

The Bus Arbiter provides several resolving techniques based on a priority concept that at a given time one SBC will have priority above all the rest. The RPC can be regarded as

a parallel priority resolver. A parallel priority resolving technique has a separate bus request BREQ line for each arbiter on the system bus (Multibus). Several BREQ lines enter to the RPC input. For each BREQ line, there is a corresponding BPRN (bus priority in) line at the output of the RPC. Only one BPRN signal can be activated at any given time. This signal BPRN is returned to the highest priority requesting bus arbiter. The bus arbiter receiving priority (BPRN active low) then allows its associated SBC onto the multi-master system bus, as soon as the bus becomes available (i.e., it is no longer busy). When one bus arbiter gains priority over another arbiter, it cannot immediately seize the bus. It must wait until the present bus occupant completes its transfer cycle. Upon completing its transfer cycle, the present bus occupant recognizes that it no longer has priority (BPRN goes high) and surrenders the bus, releasing the Busy signal. Busy is an "active low" signal line which goes to every bus arbiter on the system bus and is tied with other busy signals by a "OR" gate. When the "Busy" goes high, the arbiter which presently has bus priority (BPRN active low) then seizes the bus and pulls "Busy" low to keep other arbiters off the bus. (See waveform timing diagram, Fig. 3.13.) Note that all multi-master system bus transactions are synchronized to the bus clock (BCLK). This gives to the parallel priority resolving circuit time to settle and make a correct decision. Fig. 3.14 depicts the interconnections between the bus arbiters and the RPC.

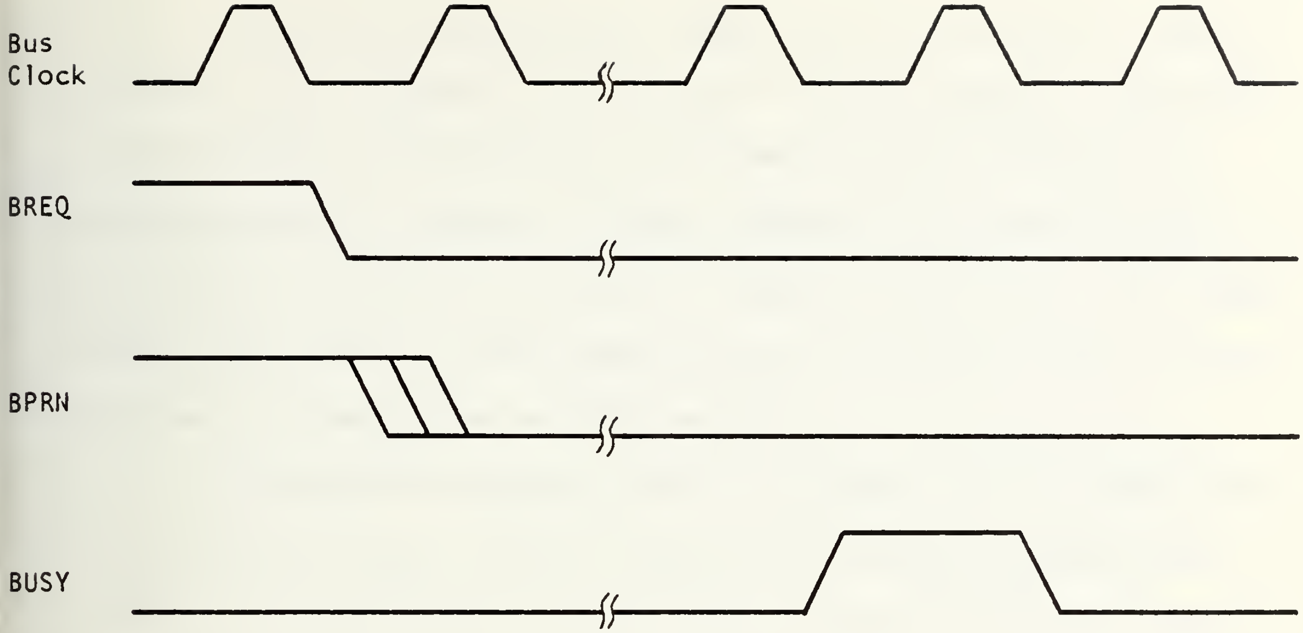


Fig. 3.13 Timing Diagram of Bus Arbiter and Random Priority Controller

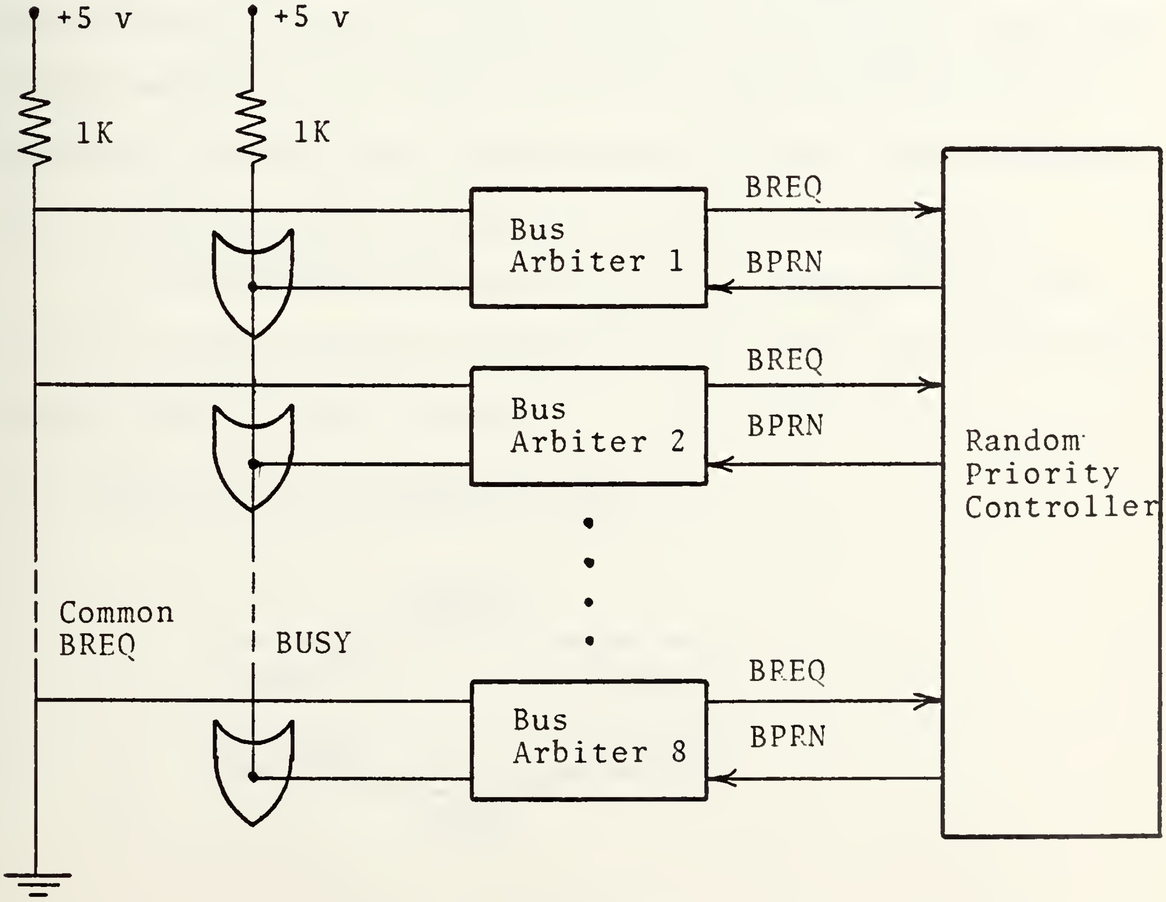


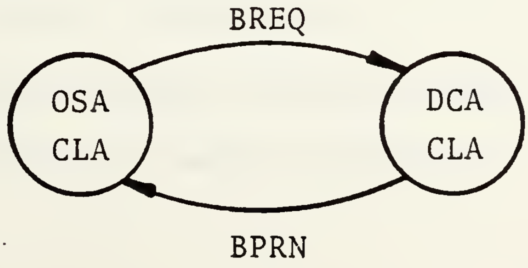
Fig. 3.14 Interconnection of Random Priority Controller and Bus Arbiters

In our configuration, every master currently using the bus will surrender the bus upon completing its transfer cycle (unless a bus lock is executed). This property is accomplished by tying all CBREQ (common bus request) lines o of all bus arbiters to ground. CBREQ is an active low signal which indicates to the current master on the bus that the bus has been requested by another master.

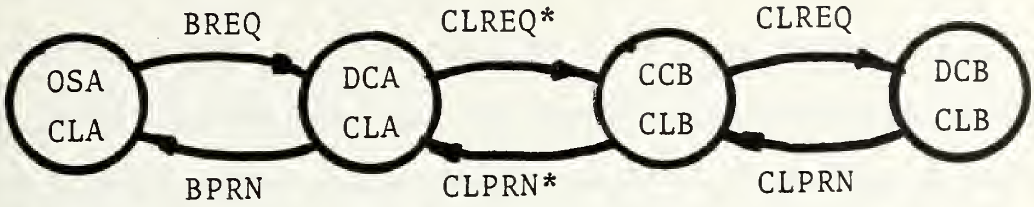
Two other signals, LOCK and CRQLCK, lend to the flexibility of the bus arbiter within the system configuration. LOCK is a signal generated by the processor to prevent the bus arbiter from surrendering the multi-master system bus to any other master, either higher or lower priority. CRQLCK (common request lock) serves to prevent the bus arbiter from surrendering the bus to a lower priority bus master when conditions warrant it. LOCK is used for implementing software semaphores for critical code section and real time critical events (such as memory refresh or hard disc transfer).

In the three different types of communications we referred to the term PRN and REQ chains. The following state diagrams depict those chains:

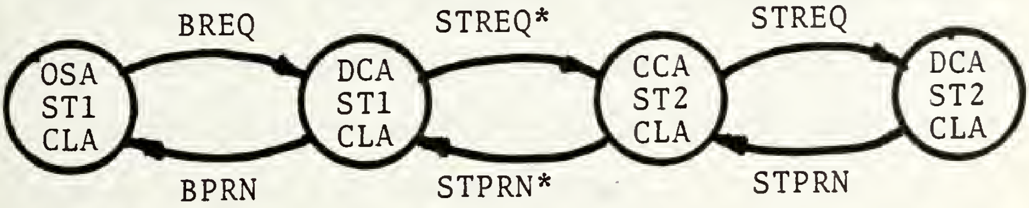
1) Intra-cluster communications



2) Inter-cluster communications



3) Inter-star communication



D. PRESENTATION OF RESULTS

1. Introduction

The important hardware components developed in this thesis to support this multiple microcomputer system are the following:

Interconnection:

Intra-cluster -- Multibus

Inter-cluster -- Complete-Star Bus Switch Network

Intercommunication Control (three levels):

Random-Priority Controller

Distributed Controller

Central Controller

In this section, we will present representative test results to answer two major questions.

1) Did our design work?

2) How well did it work?

Since the Multibus is developed by Intel and is well documented [196], we decided not to report its operations here. We will describe the operational results of the bus switch and the three levels of intercommunication control.

How well they work together in a computational environment will be reported in Chapter IV where the implementation of an adaptive spatial filter on the multiple microcomputer system will be described.

2. Bus Switches

The function of a bus switch is to transmit a signal from the Multibus in one cluster to the Multibus in another cluster. For four clusters, the "complete star bus switch network" designed has six branches of bus switches as shown in Fig. 3.7. Although the Intel's Multibus has 86 lines, we decided that only 58 of them need to be switched to facilitate communication between two SBCs from different clusters. Therefore, one "bus switch" includes appropriate circuits to transmit 58 signals, including data, address and control signals.

Four figures will be used to describe the behavior of the bus switch. The first three figures are used to show the improvement of signal waveform before and after the bus switch. The signals shown are the following:

One data bit - Fig. 3.15a
One address bit - Fig. 3.15b
One control signal - Fig. 3.15c

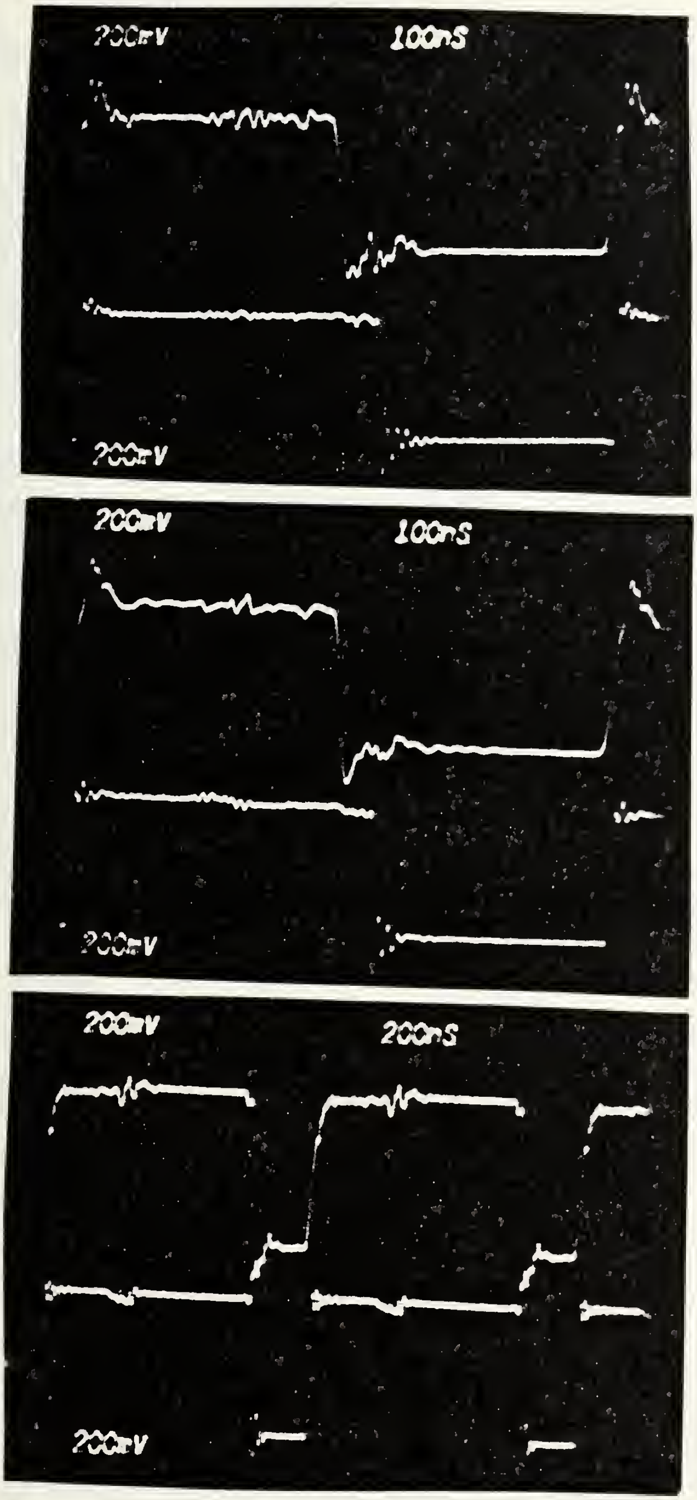
Each figure consists of two traces. The top trace shows the waveform before the switch. The lower trace shows the waveform after the switch. It can be seen that in all three cases the waveforms after the switch are better because their rise times are all shorter, giving a sharper pulse. It is interesting to note the noise appearing on these three signals. They are typical in the real operational environment. It should be noted that the control signal in Fig. 3.15c is the Acknowledge Signal (XACK) generated by the SBC requesting the use of the system bus.

The behavicr of the bus switch is described also by Fig. 3.20 which shows the delay of the switch. Again, the top trace is before the switch, the bottom trace is after the switch. The delay is no more than 25 nsec.

These four figures demonstrated that our bus switches are adequate to provide communication between two Multibuses running at 10 MHZ.

3. Random Priority Controllers (RPC)

The function of random priority controllers is to arbitrate the requests of bus usage from many SBCs, either from the same cluster or from several clusters. If an SBC from another cluster wants the Multibus to communicate either with another SBC or with the Global RAM, two higher level controllers - the central controller and two distributed



A data Bit

Fig. 3.15a

An Address bit

Fig. 3.15b

A control signal "Acknowledge" (XACK)

Fig. 3.15c

Figure 3.15 The input and output waveforms of three selected signals to demonstrate the performance of bus switch

Top trace: Input to the bus switch
Bottom trace: Output of the bus switch

controllers associated with this cluster and the other cluster where the requesting SBC resides - must also participate in the control function. However, the control ultimately came to the RPC because it is the circuit which grants the bus usage signal, BPRN (Bus Priority In). One RPC is used for every Multibus. So there are four RPCs in each star.

The behavior of our RPC will be described by four figures using the BPRN signals (Bus Priority In) of the SBCs requesting the bus. A BPRN low signal means the SBC has been granted the bus and is using it.

a. Sharing of the Multibus by Two SBCs.

Fig. 3.16 shows BPRNs of two SBCs. The bus usage pattern was created by software. Each unit of low BPRN represents a transfer of one word. If there is no request of bus usage by other SBCs, the SBC currently using the bus will hold, as shown by the BPRN low signal for a longer period of time. The figure shows the interleaving of bus usages by these two SBCs, indicating that the RPC works rapidly and efficiently to serve these two SBCs.

b. Slow-Down of Bus Release Due to Refresh of Dynamic RAM

However, we discovered that the SBC using the bus may not release the bus after its one word of transfer, as shown by a wide gap in Fig. 3.17, although the other SBC was requesting the bus. We discovered that this is the nature of Intel's 8612 design. When the dynamic RAM is being

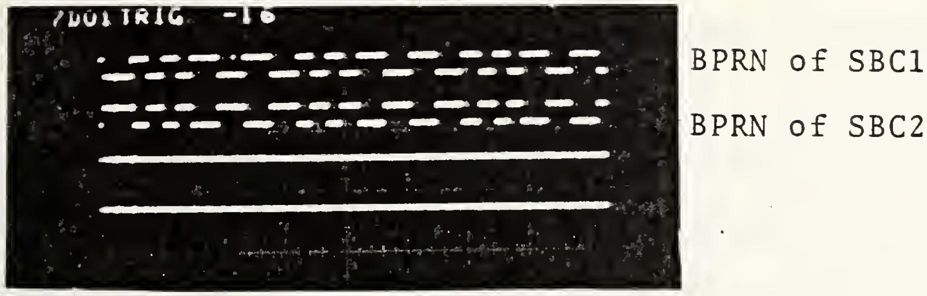


Figure 3.16 Bus Priority In signals of two SBCs to demonstrate the arbitration of their usage of the bus by the random priority controller

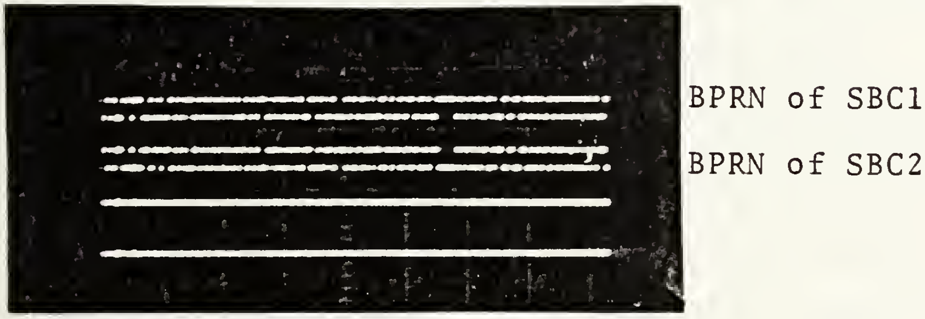


Figure 3.17 Bus Priority In signals of two SBCs to demonstrate the effect of dynamic RAM refresh on the bus usage

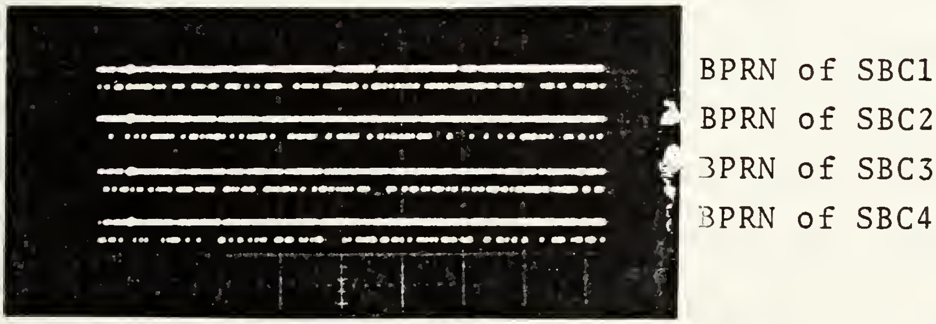
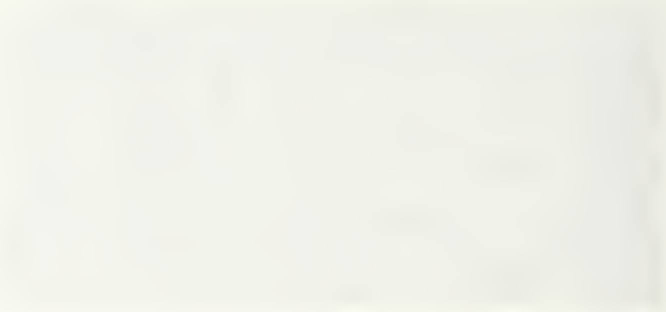
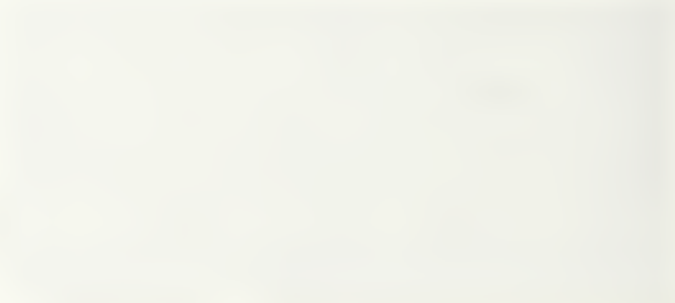


Figure 3.18 Bus Priority In signals of four SBCs to demonstrate the arbitration of their usage of the bus by the random priority controller





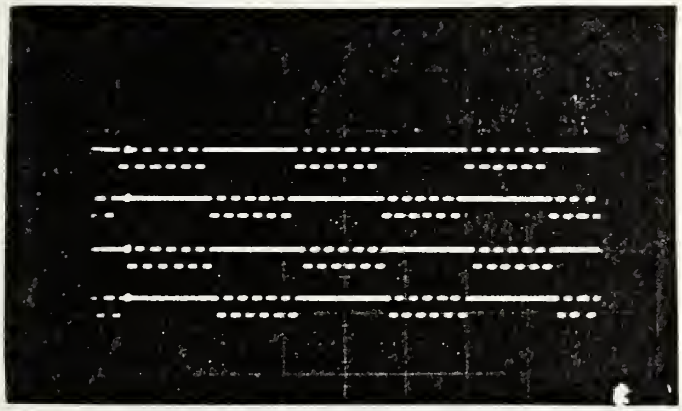
refreshed, the SBC will not release the bus. This is a drawback we cannot do anything about except to redesign the 8612 SBC.

c. Sharing of Multibus by Four SBCs

Fig. 3.21 shows the BPRN signals of four SBCs. Their general patterns are similar, in the sense that there is no large gap in any one of these traces indicating no SBC is dominating the bus and none is being left out either. This "uniform" and "equal" treatment of all SBCs requesting the bus is exactly what the RPC is designed to do.

d. Behavior of RPC When the Bus is Saturated

We prepared the most severe test for the RPC by programming four SBCs requesting the bus all the time. Of course, in real applications, this condition should never be allowed to happen. It represents very poor application programming. However, it is a tough test for the RPC. Fig. 3.19 shows the BPRN of four SBCs. The interleaving of bus usage is no different from the previous three figures. However, it is important to note that the bus was first shared by SBC1 and SBC3 for 12 transfers and then shared by SBC2 and SBC4 for another 12 transfers, followed by the repetition of such a pattern. Two important properties caused this pattern. First, the RPC is designed based on a binary tree selection. Therefore, only two SBCs will be granted first, followed by another pair. Second, the 12 transfers between SBC1 and SBC3 are determined by the basic design of the 8686 instruction queue which has a FIFO queue of six instructions.



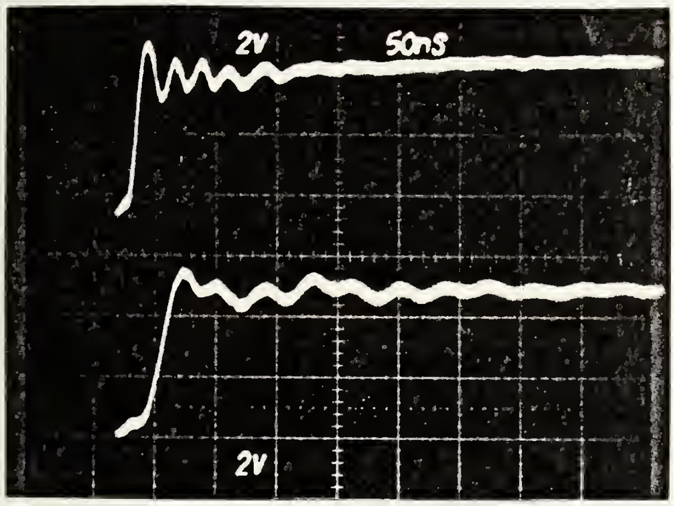
BPRN of SBC1

BPRN of SBC2

BPRN of SBC3

BPRN of SBC4

Figure 3.19 Bus Priority In signals of four SBCs which request the bus usage 100% of the time to demonstrate the function of random priority controller



Input signal to a bus switch

Output signal waveform from a bus switch

Figure 3.20 Waveforms of input and output signal of a bus switch to demonstrate the operation of the switch

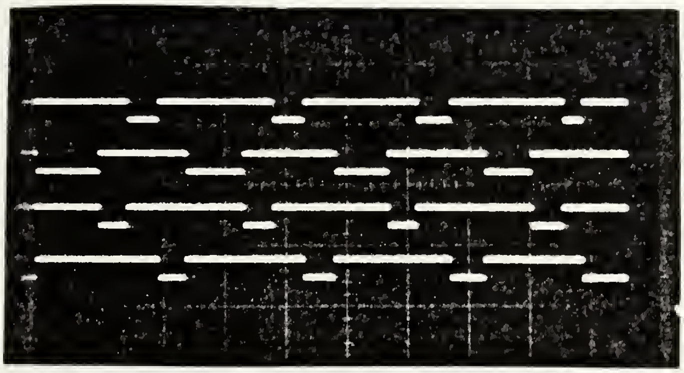


Figure 3.21 Bus Priority In signal of four microcomputers requesting 20% usage of the Multibus to demonstrate the operation of the random priority controller in this example of heavy bus requests (80% bus request)

This demonstration clearly indicated that our RPC is able to arbitrate four SBCs under the most demanding bus contention situation which should never be allowed to occur in real application.

4. Central Controller

The function of the central controller is to arbitrate requests for inter-cluster and inter-star communication. It works jointly with the distributed controllers to search, select and synchronize these requests. Although there is only one central controller for a star, it has four sections, one for each cluster in the star.

The important components of each section in the central controller are CSRA and CSPE. All four sections are synchronized by two clocks: CLK1 for the searching and selecting of requests, CLK2 for their synchronization.

Two figures will be used to demonstrate their operations.

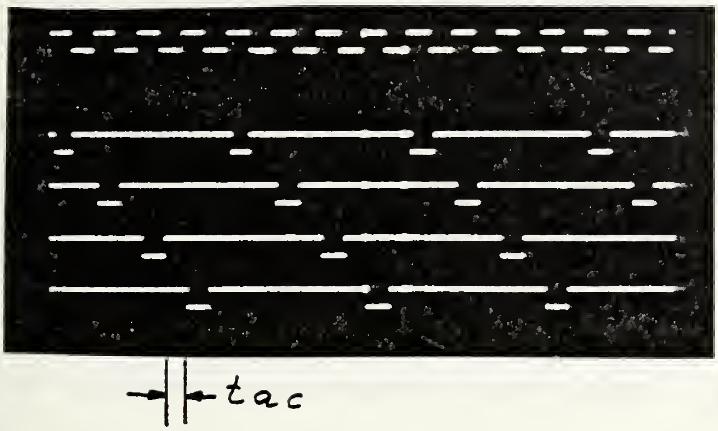
a. Searching/Selecting Clock (CLK1) and Synchronization Clock (CLK2)

These two clocks are the heart beats of the inter-communication network. It should be realized that CLK2 is not independent because it is generated from CLK1. Fig. 3.22 shows their mutual relationship. The third trace is CLK1. Below it are the four-phase CLK2 signals for four clusters. It is important to note that there is no overlap among them. This is to avoid any undesirable coincidence. CLK1 is at a higher clock frequency such that all requests from other

clusters and stars are searched and selected at adequate rates. Once a request is selected, it is synchronized by CLK2 and sent on to the appropriate cluster.

b. Searching and Selection of Requests

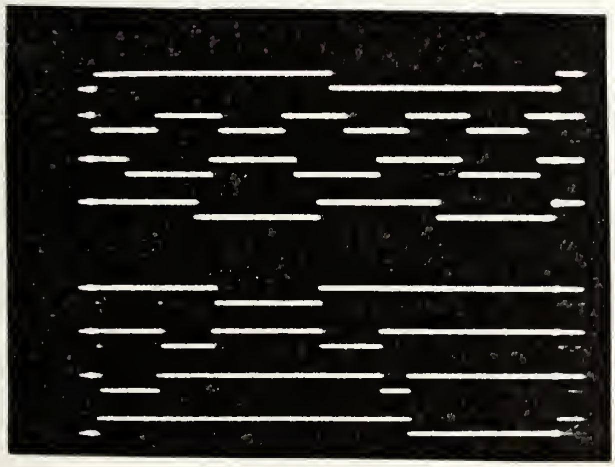
Fig. 3.23 shows the functions of CSRA and CSPE circuits of the central controller A. Four signals are shown in the top half of the figure representing three cluster requests from clusters B, C, D and from the cluster A of another star, respectively. The lower half of this figure shows the cluster or star grant signals to another star, cluster D, C and B, respectively. It is important to note that these CLPRN (or STPRN) signals do not overlap although the request signals do overlap. It can be seen that cluster C sent its CLREQ first and got its CLPRN. However, cluster D sent its CLREQ before cluster C finishes its request. Such an occasion is generally not allowed in real application because any SBC is allowed to transfer one word of data and must release the bus only if a software bus lock is ordered. However, this test is to challenge the ability of the central controller. In this case, the CSRA/CSPE of the CCA will allow the cluster A to complete its request period and then award a CLPRN to cluster D. This figure clearly demonstrated that with a mix of cluster request signals from three clusters and one star, some with overlap, some without overlap, the central controller is able to take in these requests, sort them out, select one at a time and award "cluster grant" appropriately.



CLK1: For Searching
and Selection

CLK2: 4 Phase Clock
For Synchronization

Figure 3.22 Two Clocks In Central Controller For Searching/Selection and Synchronization of Requests From Stars and Clusters



Four Request Signals To CSRA:

From DCB
From DCC
From DCD
From Star A

Four Priority In Signals From CSPE:

To Star A
To DCD
To DCC
To DCB

Figure 3.23 Demonstration of the Functions of CSRA and CSPE Circuits in the Central Controller (Section A for Cluster A) Input to CSRA, Output from CSPE

Of course, this is not the completion of the intercommunication task. The CLPRN will be sent to the distributed controller to initiate further control actions to complete the total task of communication between two SBCs.

5. Distributed Controller

The function of the distributed controller is the same as that of the central controller. They must work with the RPC to complete the intercommunication. The central controller is located away from the Multibus and also controls the operations of all bus switches. The distributed controller is mounted on the Multibus. Therefore, we have four distributed controllers in a star. The important components of each distributed controller are:

ICAAM (Intra-cluster advanced activities monitor)
CIC (Coincidence inhibitor circuit)
DAC (Deadlock avoidance circuit)

Four figures will be used to demonstrate their operations. Eight control signals in the distributed controller are used in these figures.

BREQ
CLREQ*
Internal/External Signal
Inhibit
PRE
BHD
CLPRN
BPRN

The first and eighth control signals, BREQ and BPRN, are two of the most important ones because they are directly connected to the SBCs. We must remember that all the buses,

switches, controllers are supporting circuits to help the SBCs to compute, to talk among themselves efficiently. The SBCs are the originators and receivers of the data and communication and control signals.

a. Intra-Cluster Communication

Fig. 3.24 shows the sequence of events in a test case where one SBC in a cluster wants to talk to another SBC in the same cluster.

It can be seen that CLREQ* (second trace) is high, which means no request from another cluster. CLPRN (7th trace) is therefore also high, i.e., no cluster priority signal is granted by the central controller.

It is interesting to notice the small delays between BREQ, PRE and BPRN.

b. Inter-Cluster/Intra-Star Communication

Fig. 3.25 shows the sequence of events in a test case where an SBC in one cluster wants to talk to an SBC in another cluster within the same star.

There are several interesting points when this case is compared with the intra-cluster case:

- Both BREQ and CLREQ* exist.
- Inhibit signal is active to prevent any premature generation of BPRN.
- CLPRN is also active to respond to the CLREQ*.

It is clearly seen that this inter-cluster communication has been correctly handled by the distributed controller.

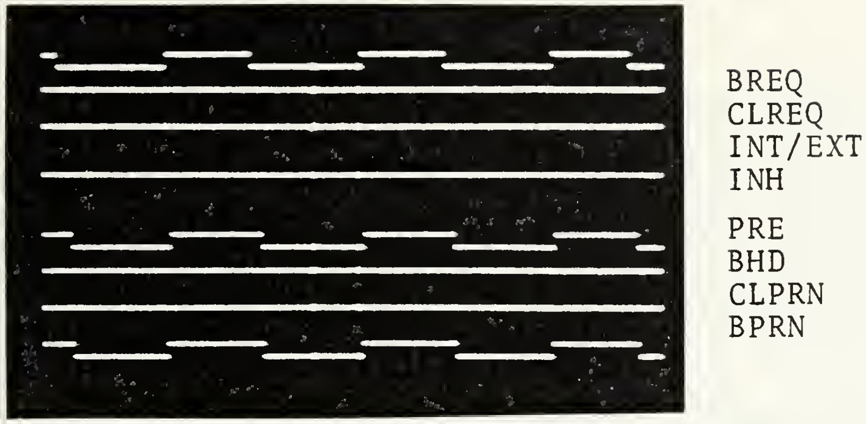


Figure 3.24 Eight Control Signals to Demonstrate The Function of Distributed Controller For Arbitration of Intra-Star and Intra-Cluster Communication

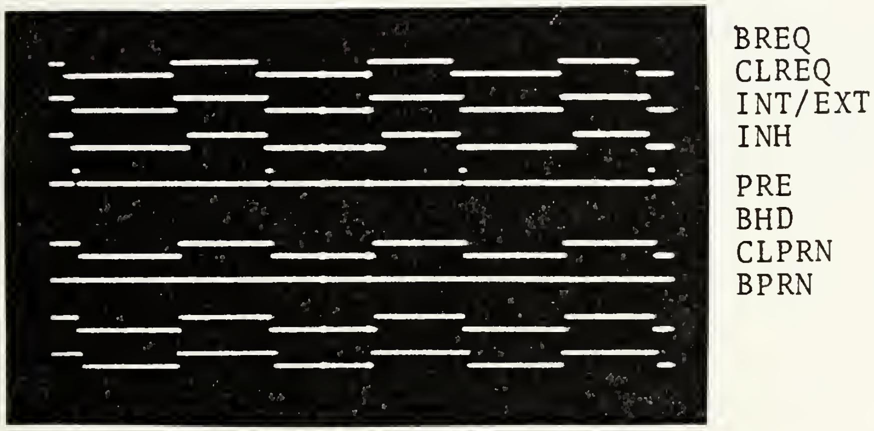


Figure 3.25 Eight Control Signals to Demonstrate The Function of Distributed Controller For Arbitration of Intra-Star and Inter-Cluster Communication

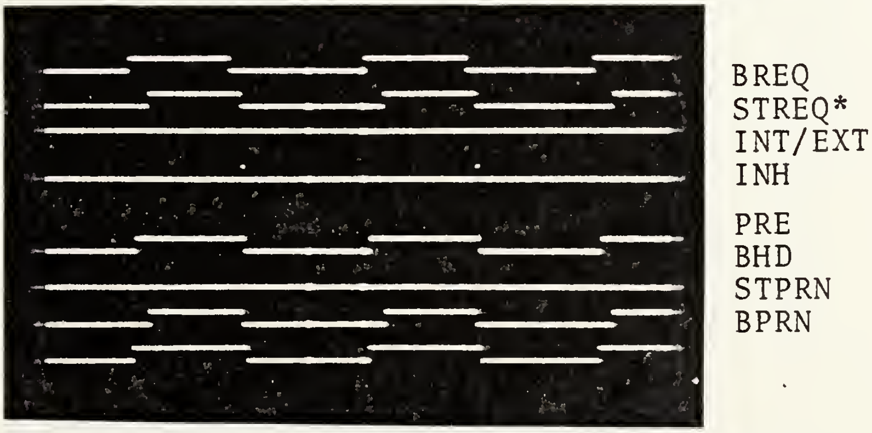


Figure 3.26 Eight Control Signals to Demonstrate The Function of Distributed Controller For Arbitration of Inter-Star Communication

c. Inter-Star Communication

Figure 3.26 shows the sequence of events in a test case where an SBC in one cluster of a star wants to talk to an SBC in the corresponding cluster of a neighboring star. They are quite similar to the inter-cluster/intra-star case in Fig. 3.25 with several changes.

> The second trace is now the STREQ* instead of the CLREQ* signal.
>
> The seventh trace is now the STPRN signal instead of the CLPRN signal.

The rest of the signals behave quite similarly. It shows that requests from a cluster in the same star and from a neighboring star are treated quite the same.

IV. IMPLEMENTATION OF ADAPTIVE FILTER ON MULTIPLE MICROCOMPUTER SYSTEM

A. INTRODUCTION

1. Selection of Microcomputer

The goal of this thesis research was to eliminate the gap between the theoretical development of image processing algorithms and the experimental development of their implementation on some processor systems which are good candidates for practical applications.

In this thesis, a multiple microcomputer system was chosen as the processor system candidate.

It should be recognized that only during the past two to three years have 16 bit microcomputers been seriously considered for signal processing implementations. Although 8 bit microcomputers have been investigated for performing signal processing operations, the motivations of these studies are mainly to explore what can the 8 bit microcomputers do for signal processing. For serious implementations, bit slice microprocessors have always been the favored approach which can be designed to emulate 16 bit, 32 bit or even longer word computers. However, 16 bit microcomputers are being supported with more and more powerful hardware and software and are approaching low-end minicomputer performance.

To examine the signal processing performance of today's 16 bit MOS microcomputer, we coded the statistical

3 x 3 spatial filter on one main frame computer, IBM 360/67 and two 16 bit microcomputers, DEC LSI-11 and Intel 8612, using high order programming languages and single precision numerical data format. Fortran is used for the IBM and DEC computers. PLM86 is used for the Intel computer. The execution times expressed in seconds are shown in Table IV.1 for comparison.

TABLE IV.1

IMAGE PROCESSING EXECUTION TIME
(in seconds)

Image Processing Operations	IBM 360/67	DEC LSI-11	Intel 8612	
	Fortran	Fortran	PLM86	Macro
	Single Precision	Single Precision	Single Precision	Integer
Spatial Statistics Calculation	4.07	25.46	334.25	0.72
Spatial Filter Design	0.0047	0.24	2.82	
Perform Spatial Filter	0.98	5.62	79.8	0.47

It can be seen that LSI-11 has better floating point computation support today than Intel's 8612 which took 13 to 14 times longer than the LSI-11 to perform these image processing operations. The LSI-11 itself took approximately 6 times longer than the IBM 360/67. Based on this comparison, the LSI-11 should be chosen as the 16 bit microcomputer candidate. However, Intel's 8612 was selected because of its larger physical memory addressing space and its system Multibus support which are much better suited for multiple microcomputer system development.

Further, two of the three spatial filter modules were coded in assembly language and a 32 bit integer data format on the 8612. It was found that the execution times are quite short, suggesting that even today's Intel 16 bit microcomputer, without the assistance of hardware arithmatic devices, can perform these rather sophisticated image processing operations very well if compared with the main frame computer IBM 360/67. More specifically, it took 0.72 seconds to compute the autocorrelation matrix elements for the 3 x 3 spatial filter, averaged over the 32 x 32 image, and 0.47 seconds to perform this 3 x 3 spatial filtering over the image.

2. Implementation

In this chapter we will present the implementation results of our adaptive filter on the multiple microcomputer system. In Section B, the performance of spatial filters is discussed. In Section C, the performance of adaptive spatial filters will be discussed.

The functions of various components of the interconnections and communication controllers have been described in previous sections using mainly signals generated by function generators. In this section, a test program was used to test and evaluate the data transfer behaviors of the system. This program is quite straightforward and fetches data from the RAM and displays them on a CRT terminal. However, the locations of the program and data are at different parts of the system to provide a thorough test of the data transfer and bus arbitration behaviors.

Three tests were made.

The objectives of the first two tests are to measure the maximum rate of data transfer on the system bus. For this purpose, both the program and data were stored either in the global RAM located in another slave SBC, as in test case 1, or in the global RAM located in the μPRO RAM board. Therefore, the system bus was used very busily because not only the data must be fetched via the bus, the program itself must be read from the memory external to the testing SBC.

TABLE IV.2

MEMORY ALLOCATION FOR MULTIBUS TEST

Test No.	Location of Program	Location of Data	Remarks
1	Slave SBC	Slave SBC	Program and data being run at maximum rate.
2	μPRO RAM	μPRO RAM	
3	Master SBC	μPRO RAM	Program and data being run at approximately 20% of the maximum rate.

The maximum rates at which this test can run with one to six microcomputers are shown in Table IV.3. Several important facts can be noticed.

(1) The bus transfer rate of each SBC is reduced when more and more SBCs want to use the bus, as it should be.

(2) However, the maximum rate and amount of reduction vary from test to test. For example, in test 1, we

were able to transfer 710 Kbyte/sec at its maximum if only one SBC is using the bus as compared with a maximum of 911 Kbyte/sec rate for one SBC in test case 2. Test 2 showed that it is quicker to get data out of the μPRO than the RAM on a different SBC. This can be explained easily because control on the SBC must decide whether the memory addressed is on-board or off-board. This decision takes time, thus it slows down the transfer rate. When more SBCs were added in these two tests, the transfer rate of every SBC was decreased. However, the rates of decrease were different in Test 1 and Test 2 as shown in Table IV.3. They are also plotted in Fig. 4.1 to give a graphical view. It is obvious that substantial deteriorations of the bus transfer rate took place in these two cases, from 710 Kbyte/sec to 144 Kbyte/sec in Test 1 and from 911 to 167.1 Kbyte/sec in Test 2.

(3) It should be pointed out that such heavy usage of the system bus should be allowed to happen only during tests. If a programmer prepared an application program with such heavy bus usage, he has failed miserably in partitioning his program for parallel and pipeline computation in the multiple microcomputer system.

(4) Therefore, to provide a test more compatible with real operational conditions, Test 3 was prepared which has its program in the RAM of the master SBC and its data in the global RAM in μPRO. Further, it was run at a rate of 194.9 Kbyte/sec on the bus when only one SBC requested

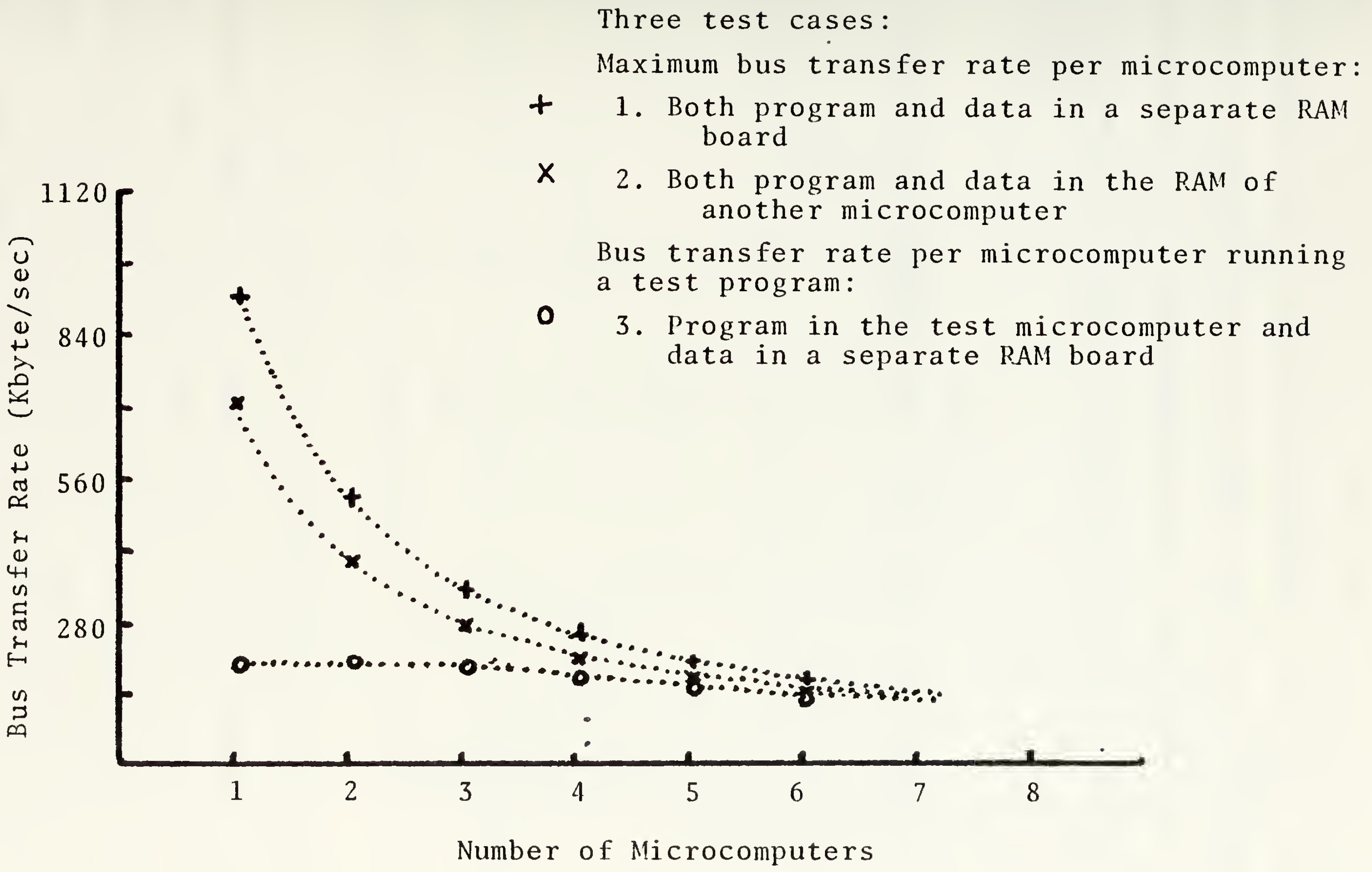


Fig. 4.1 Bus transfer rate per microcomputer of three test cases in a multiple microcomputer system to show the reduction of bus usage due to bus contention by several microcomputers

the bus. It can be seen that the deterioration of the system bus transfer rate is much more moderate, from 194.9 for one SBC to 132 Kbyte/sec for six SBCs. This is a testimony of the ability of the intercommunication controller in treating all SBCs equally without allowing any one SBC to dominate the bus usage.

TABLE IV.3

SYSTEM BUS TRANSFER RATE (Kbyte/sec) FOR EVERY SBC IN THREE MULTIPLE MICROCOMPUTER SYSTEM TESTS

No. of SBCs	Test 1	Test 2	Test 3
1	710	911	194.9
2	400.7	522	188
3	277.7	345.33	184
4	212	255.7	166
5	171.8	202.3	147.9
6	144	167.1	132

(5) Further, the overhead loss of transfer rate in arbitrating the bus usage of several microcomputers is small. Let us consider Test Case 2. The maximum bus transfer rate took place when there were two SBCs using the bus and was 2 x 522 = 1044 Kbyte/sec. When six SBCs were using the bus, the total transfer rate on the bus was 6 x 167.1 = 1002.6 Kbyte/sec. The loss is only (1044 - 1002.6)/1044 = 0.0397, or 3.97%. Of course, each SBC suffered a loss of

(911 - 167.1)/911 = 81.658% in its bus usage rate. It is interesting to note that 167.1 KBS for six SBCs is close to one-sixth of the rate of 911 KBS if one SBC has the system all to itself.

B. IMPLEMENTATION OF 3 x 3 SPATIAL FILTERING ON MULTIPLE MICROCOMPUTER SYSTEM

1. Introduction

Four different implementations were compared. They differed in the manner of storing the programs, variables and data in various parts of the memory hierarchy and some programming skills. For this development, all program and data were stored in RAM on the single board microcomputers. These RAM have been separated into two types:

- Unshared RAM: They are "private" to the microcomputer where the RAM is located.
- Shared RAM: They are "global" and can be accessed by other microcomputers on the same Multibus.

TABLE IV.4

PROGRAM DATA AND VARIABLE ALLOCATION

Implementation	Program	Variables	Data
Case 1	Ideal Case		
Case 2*	Unshared	Unshared	Shared
Case 3	Unshared	Unshared	Shared
Case 4	Unshared	Shared	Shared
Case 5	Shared	Shared	Shared

The results are presented in Fig. 4.2 which expresses the number of frames which can be performed on the 3 x 3 spatial filtering task per second as a function of the number of microcomputers used to partition the spatial filtering into parallel operations. It should be pointed out that the image size is 30 x 30 pixels. The partitioning is to split the image into equal parts for several microcomputers.

The results will be discussed in the following.

a. The first case is not a measured result. It represents the ideal enhancement of computation by using multiple microcomputers. We first measured the execution speed of performing a spatial filter over the whole image by one microcomputer with program, variables and data all in the private unshared RAM of the SBC. There was no bus usage, therefore no overhead due to bus communication. The maximum filtering speed is roughly two thousand pixels processed by this spatial filter per second. For more SBCs, we simply multiply the rate by the number of microcomputers and plotted a "linear enhancement" curve. This represents the ideal case and serves as the goal for our partitioning to approach.

b. Let us start with the case of lowest performance, Case 5. In this case, all program, variables and data were located in the shared memory of another SBC. It obviously required the maximum amount of transfer and system bus usage. It can be seen that the performance saturated quite quickly.

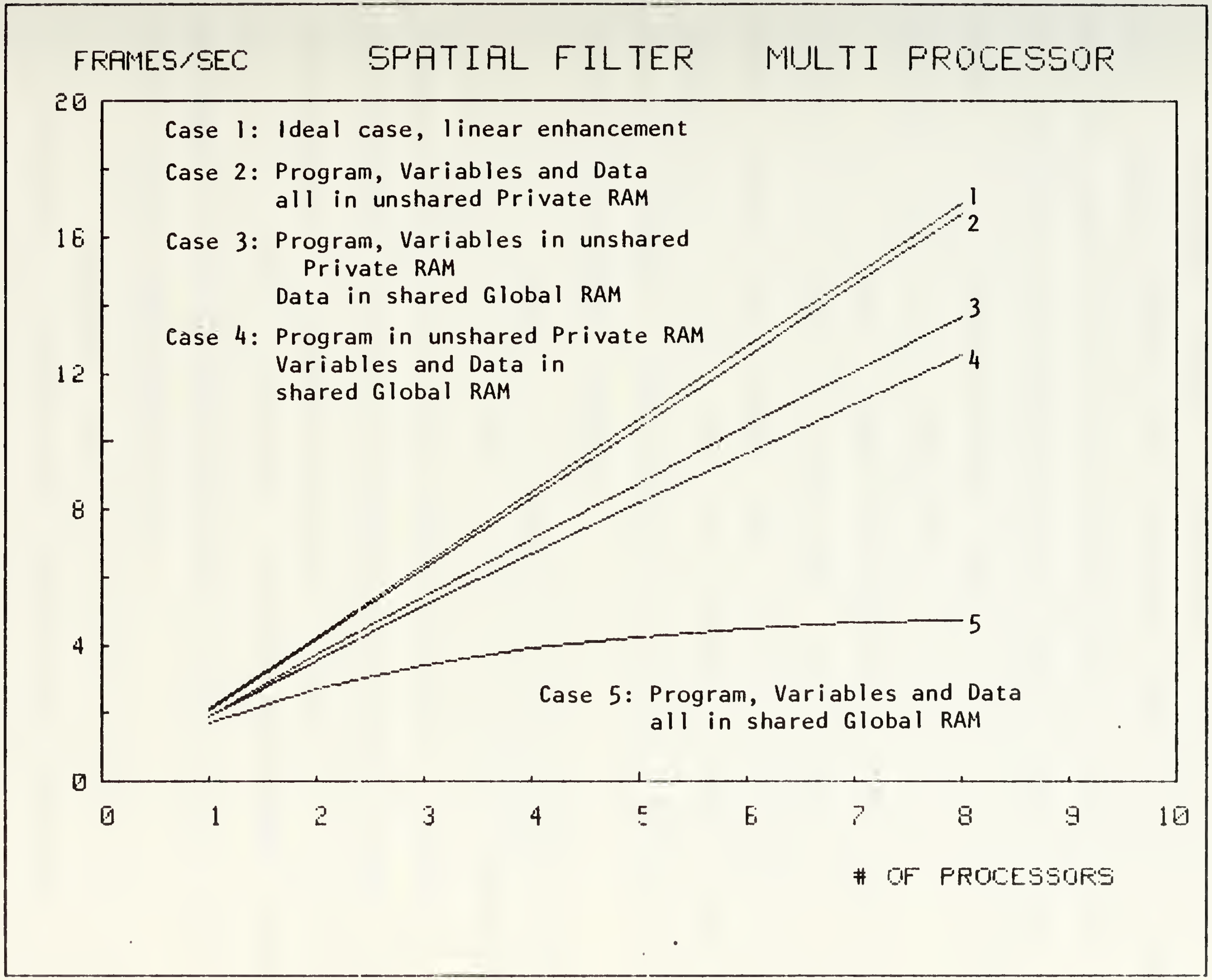


Fig. 4.2 Performance of the Partitioning of a Spatial Filter In a Multiple Microcomputer System (Parallel Processing)

We are obviously wasting the computational power of added microcomputers.

c. Next, in Case 4, where the program was stored in the private memory of the computing SBC, but the variables and data were stored in the global memory of another SBC. The throughput performance improved almost linearly with respect to the number of microcomputers but at a rate lower than the "ideal linear enhancement."

d. In Case 3, both the program and variables were stored in the unshared private RAM. But the data were stored in the global RAM of another SBC. Further improvement was accomplished. However, about 20% of the computing capability was lost because of the overhead for the arbitration of multiple microcomputer requests.

e. In Case 2, the locations of the program, variables and data are the same as in Case 3, but the programming is more clever in the sense that the number of accesses to the system bus by each microcomputer is minimized and, further, the occurrences of these system bus accesses were distributed as evenly in time as possible. It can be seen that the enhancement of total computing power is much closer to the total "ideal linear enhancement" case.

f. In summary, we have used the special case of spatial filtering to explore the behavior and improvement of computing by the multiple microcomputer system. It should be pointed out that although there have been a lot of ideas

in this field, real experience is still very limited. Consequently, there is really no concensus in the philosophy, approaches and methodologies of effective partitioning for parallel and pipeline computing. This thesis is a first step in testing the uncharted water. We only used a spatial filter to test the parallel processing. We have not used a problem to test pipeline processing and combined parallel/pipeline processing yet. Therefore, we do not intend to declare that the experience learned from this spatial filtering established a general methodology for effective partitioning.

But we feel that the following guidelines probably will be helpful when more complex problems will be tested to develop a more thorough philosophy of partitioning:

a) The bus usage should be minimized.

b) The bus usage should be distributed more evenly in time. Concentration of bus usage should be avoided.

g. Meanwhile, it should be pointed out that this implementation of spatial filtering is a test case based on a real computation problem. In addition to the experience learned for partitioning, the successful implementation of the spatial filtering involving up to five microcomputers in parallel processing convincingly proved that the random priority is working correctly.

V. CONCLUSION AND RECOMMENDATIONS

A. CONCLUSION

1. Motivation

This thesis was motivated by the needs of new smart sensor developments. With the anticipation of new sensitive and large mosaic optical sensor arrays and very sophisticated signal/data processing capabilities to be offered by VLSI/VHSIC electronics, very ambitious mission objectives of new surveillance, search/track and weapon guidance systems are being proposed and developed, which require new signal processing techniques to accomplish demanding goals. Further, they require very sophisticated processor systems which are powerful enough to implement the new signal processing algorithms and also small and light enough for mounting on platforms of practical systems.

2. Single Objective and Dual Tasks

This thesis has one single objective, to help to make the new "smart sensors" practical, but consists of two tasks to achieve this objective.

a. Develop new adaptive filter techniques to process infrared images for enhancement of "target signal" to "background clutter noise" ratio.

b. Develop a new multiple microcomputer system to implement this type of image processing.

3. Extensions and Contributions

Both studies, although motivated by the development of "infrared smart sensors," are generic and can contribute to broader fields much beyond the image processing problems in infrared smart sensor systems.

4. Results I - Adaptive Filters

The following results have been obtained:

a. Adaptive filter research done in the past was surveyed. It was found that:

- Practically all past research dealt with one dimensional problems, except one by B. Evenor who extended the LMS algorithm to images generated by Markov models.
- Most approaches are based on LMS algorithms.

b. In this thesis the LMS algorithm was extended to process real world infrared images.

c. A new approach to nonrecursive adaptive filters was developed which is similar to searching for the extreme point in optimization problems.

d. Two optimization criteria were considered:

mMSE = minimization of mean square error
MSNR = maximization of signal to noise ratio.

e. Seven different optimization/searching techniques were developed:

- Gradient approaches = $\begin{cases}\text{Steepest descent}\\ \text{Accelerated steepest descent}\\ \text{Amir's method (mMSE only)}\end{cases}$
- Conjugate gradient approaches = $\begin{cases}\text{Fletcher-Reeves}\\ \text{Pollack}\end{cases}$
- Variable metric approach - Davidon-Fletcher-Powell

° Amir's transform approach (MSNR only)

f. These approaches were tested on two infrared test images:

° Indiana - Blue spike band infrared image appropriate for high altitude downward looking infrared sensor systems.

° China Lake - 10-13 micron thermal band infrared image appropriate for shorter distance side-looking infrared sensor systems.

The results are encouraging and showed that these new adaptive filters are effective in suppressing background clutter and enhancing the "target signal" to "clutter noise ratio."

5. Results II - Multiple Microcomputer System

a. The tightly-coupled multiple microcomputer research done in the past was surveyed. It was found that:

° There are many conceptual designs of new multiple microcomputer systems. Only a very small number of these have embarked on actual developments with both hardware and software efforts.

° More loosely coupled multiple microcomputer systems are being developed. They are mostly computer networks.

° There are only two tightly coupled multiple microcomputer systems in operation today based on the survey of the open literature. Both are at Carnegie Mellon University: Cmmp and Cm*. It should be noted that although Cmmp is a multiple minicomputer system, today's 16 bit microcomputers are fast approaching minicomputer performance.

b. Based on an intensive consideration of the requirements of typical new smart sensor systems in not only the mission signal processing area but also in management, control, and communication areas, it was decided that a hierarchical architecture which supports simultaneous tightly and loosely coupled systems is attractive.

c. A multiple star, multiple cluster architecture using commercially developed 16 bit microcomputers was developed. A complete star bus switch network was developed which is managed by a control system consisting of three levels of control: random priority controller, distributed controller, central controller.

d. The basic concept of this hardware architecture has been basically tested by simulated intercommunications. Extensive tests in real signal/data processing environments are awaiting the successful developments of operating systems.

6. ResultsIII - Implementation of Adaptive Spatial Filters on Microcomputers and Multiple Microcomputer Systems

a. The spatial filter program was coded for one main frame, the IBM 360-67, and two 16 bit microcomputers: the DEC LSI-11 and one Intel 8612. The DEC LSI-11 has more mature floating point mathematics software and a hardware arithmetic IC chip, but is not as well suited for multiple microcomputer system development as the Intel 8612, whose floating point software is still very primitive. However, when coded in assembly language, the Intel 8612 performs the spatial filtering faster than the main frame coded in high order language.

b. Implemented by using only one 16 bit 8612 microcomputer, the computation times for the 3 x 3 spatial filter

and a 32 x 32 image have been measured as follows:

Spatial statistics computation	= 0.72 sec.
Adaptive spatial filter design (Conjugate gradient Pollack method)	= 1.0 sec.
Perform spatial filtering	= 0.47 sec.

c. Several ways of using the multiple microcomputer implementation by placing program, variables and data in the unshared private RAM and/or the shared global RAM have been investigated.

It was found that the best enhancement of total execution speed of the spatial filtering is to use more microcomputers by storing the program and variables in the private RAM and the data in the global RAM. The image data is not moved into the microcomputer all at once.. Instead, the data is moved, one at a time, into the private RAM of the microcomputer only moments before it is needed for processing.

B. RECOMMENDATION

1. General

Both topics covered in this thesis are quite new. This research only opens the gate a little into two fields worthy of more investigations. Although this thesis is concerned mainly with the image processing developments and their implementations for infrared smart sensors, the techniques developed are generic and can be applied to much broader fields beyond smart sensors.

2. Adaptive Filters

The new techniques based on the concepts of gradient, optimization search can be applied to most of the adaptive filter research done in the past using the LMS algorithm.

For adaptive image processing applications, they should be used to develop adaptive temporal filters if a series of successive frames of images are rather well registered spatially from frame to frame, although there may be drift, jitter, rotations, etc. between frames.

Testing of these adaptive filters using more challenging real world images which have serious non-stationarity should be performed to give the adaptive filtering techniques some tough challenges. Jamming and interference noises should be considered. The convergence time of the compiled adaptive filter programs should be measured to obtain relative speed of convergence of all the adaptation methods. Adaptive filters for extended targets should be developed.

3. Multiple Microcomputer System

Although the subject of multiple microcomputer systems is not new, there are many unresolved questions that have hardly been touched because of the extensive effort required to make any type of multiple microcomputer system operational. Only two such systems are known to be working today, Cmmp and Cm*, although many system architectures have been proposed and conceptualized. A small number of these have been simulated. A smaller number of them are being emulated. An even

smaller number of them are being built. Simulations and modeling used today for multiple microcomputer systems must be carefully and critically scrutinized for their validity and usefulness. It is extremely important to examine how the intercommunication overhead is modeled and simulated. There is very little first-hand experience in existence today.

Therefore, a wide variety of problems associated with the new multiple microcomputer systems must be researched, examined and answered.

This thesis contributed to the formulation, design, fabrication and test of a multiple microcomputer system which can be used -

1. Not only for developing effective ways of implementing smart sensor image processing, in general, and the adaptive image processing, in particular,

2. But also as a test bed to develop, verify, and improve several basic issues of multiple microcomputer systems. Included were considerations of:

a. Effective and alternative intercommunication for combined tightly and loosely coupled systems.

b. Effective and alternative operating systems for real time signal processing, multi-tasking, multi-users, security, dynamic reconfiguration and fault tolerance.

c. Effective and alternative programming methodologies for partitioning a given problem into a number of modules suitable for combined pipeline and parallel implementation on multiple microcomputer systems.

d. Effective and alternative ways of using the distributed capabilities of multiple microcomputer systems for fault tolerance, self-maintenance error recovery.

LIST OF REFERENCES

1. "Remote Sensing of Earth From Space: Role of Smart Sensors," Ed. R. Breckenridge, Proceedings AIAA Conference, Vol. 67, 1979.

2. "Smart Sensors," Ed. D. Barbe, Proceedings SPIE, Vol. 178, 1979.

3. "Smart Sensors II," ed. D. Barbe, Proceedings SPIE, Vol. 252, 1980.

4. Bar-Yehoshua, D., Two-Dimensional Nonrecursive Filter For Estimation and Detection of Targets, EE Thesis, Naval Postgraduate School, Monterey, CA, 1977.

5. Hilmers, D.C., Spatial-Temporal Filter For Clutter Suppression and Target Detection of Real World Infrared Images, EE Thesis, Naval Postgraduate School, Monterey CA, 1978.

6. Evenor, B., Statistical Nonrecursive Spatial-Temporal Focal Plane Processing For Background Clutter Suppression and Target Detection, Ph.D. Thesis, Naval Postgraduate School, Monterey, CA, 1979.

7. Hilinitzas, G., Image Processing Studies for Detection and Tracking of Dim Targets in Multiple Frames of Infrared Images, M.S. Thesis Naval Postgraduate School, Monterey, CA, December 1979.

8. Becker, D., Microcomputer and Array Processor Based Implementation of Infrared Image Processing, MS Thesis, Naval Postgraduate School, Monterey, CA, 1980.

9. Widrow, B., "Adaptive Sampled-Data-Systems--A Statistical Theory of Adaptation," 1959 IRE WESCON Conv. Record, Pt. 4, pp. 74-85, 1959.

10. IEEE Trans. Antennas and Propagation "Special Issue on Active and Adaptive Antennas," AP-12, March 1964.

11. IEEE Trans. Antennas and Propagation, "Special Issue on Adaptive Antennas," AP-24, September 1976.

12. Widrow, B., Glover, J., McCool, J. et al., "Adaptive Noise Cancelling: Principles and Applications," Proc. IEEE, 63, pp. 1692-1716, 1976.

13. Schumer, M.A. and Steiglitz, K.,"Adaptive Step Size Random Search,"IEEE Trans. on Automatic Control, Vol. AC-13, No. 3, June 1968.

14. Widrow, B., et al.,"Adaptive Antenna Systems," Proc. IEEE, December 1967.

15. Gersho, A., "Adaptation In a Quantized Parameter Space," Proc. Sixth Annual Allerton Conference on Circuit and System Theory, Monticello, Ill., October 1968.

16. Moschner, J.L., Adaptive Filter with Clipped Input Data, Stanford University-Center for Systems Research, Technical Report No. 6796-1, June 1970.

17. Widrow, B., "Adaptive Filters," a chapter from Aspects of Network and System Theory. Edited by R.E. Kalman and N. DeClaris; Holt, Rinehard and Winston, Inc., N.Y., 1970.

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

19. Gran, R., "On the Convergence of Random Search Algorithms in Continuous Time with Applications to Adaptive Control," IEEE Trans. on Systems Man. and Cyber, January 1973.

20. Widrow, B., McCool, J.M., and Ball, M., "The Complex LMS Algorithm," Proc. IEEE, April 1975.

21. White, S.A., "An Adaptive Recursive Digital Filter," Proc. of Asilomar Conf. on Circuit Systems and Computers, Pacific Grove, CA, 1975.

22. Zentner, C.R., Frequency Domain Adaptive Decoupling in Multiple Output Array Processors, Pennsylvania State University, TM 75-223, Contract N00017-73-C-1418, DD-A023137, September 1975.

23. Widrow, B., et al., "Stationary and Non-Stationary Learning Characteristics of the LMS Adaptive Filter," Proc. IEEE, August 1976.

24. Widrow, B., McCool, J.M., "A Comparison of Adaptive Algorithms Based on the Method of Steepest Descent and Random Search," IEEE Trans. on Antennas and Propagation, September 1976.

25. Stearns, S.D., Elliott, G.R., and Ahmed, N., "On Adaptive Recursive Filtering," Proc. of Asilomar Conf. on Circuit Systems and Computers, Pacific Grove, CA, 1976.

26. Feintuch, P.L., "An Adaptive Recursive LMS Filter," Proc. IEEE, November 1976.

27. Elliott, G.R., Jacklin, W.L., and Stearns, S.D., "The Adaptive Digital Filter," Sandia Lab Report Sand 76-0360, Albuquerque, NM, August 1976.

28. Widrow, B., McCool, J.M., "Comments on 'An Adaptive Recursive LMS Filter'" Proc. IEEE, September 1977.

29. McMurray, L.R., "Stability Diagram for an Adaptive Recursive Filter," Proc. of Asilomar Conf. on Circuit Systems and Computers, Pacific Grove, CA, 1977.

30. Dentino, M.J., "Frequency Domain Adaptive Correlator," Proc. of Asilomar Conf. on Circuit Systems and Computers, Pacific Grove, CA, 1977.

31. Elliott, G.R., Stearns, S.D., and Ahmed, N., "Adaptive Transfer Filter Considerations," Asilomar Conf. on Circuit Systems and Computers, Pacific Grove, CA, 1977.

32. Ahmed, N., "A Study of Adaptive Digital Filters," Sandia Lab, Albuquerque, NM, 87115, Sand 77-0102, August 1977.

33. McMurray, L.R., "Adaptive Recursive Filter Turn-on Transient," Proc. of Asilomar Conf. on Circuit Systems and Computers, Pacific Grove, CA, 1978.

34. Parikh, D., Ahmed, N., "On an Adaptive Algorithm for IIR Filters," Proc. IEEE, May 1978.

35. Shensa, M., Time Constants and Learning Curves of LMS Adaptive Filters, Naval Ocean Systems Center, San Diego, CA, Technical Report 312, November 1978.

36. Treichler, J.R., Larimore, M.G., and Johnson, C.R., "Simple Adaptive IIR Filtering," Proc. of the IEEE International Conf. on Acoustics, Speech, and Signal Processing, Tulsa, Oklahoma, April 1978.

37. Zeidler, J.R., et al., "Adaptive Enhancement of Multiple Sinusoids in Uncorrelated Noise, IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. ASSP-26, No. 3, June 1978.

38. Schlunt, R.S., "An Adaptive Matched Filter," 12th Annual Asilomar Conf. on Circuit Systems and Computers, Pacific Grove, CA, November 1978.

39. Widrow, B., McCool, John M., and Medoff, Barry P., "Adaptive Control by Inverse Modeling," Proc. of Asilomar Conf. on Circuit Systems and Computers, Pacific Grove, CA, 1978.

40. Dentino, M.J.; Huey, H.M.; Burdic, W.S; and Zeidler, J., "Statistical Properties of the Adaptive Line Enhanced Processor (ALE)," Proc. of Asilomar Conf. on Circuit Systems and Computers, Pacific Grove, CA, 1978.

41. Sawyers, J.H., "Applying the Maximum Entropy Method to Adaptive Digital Filtering," Proc. of Asilomar Conf. on Circuit Systems and Computers, Pacific Grove, CA, 1978.

42. Treichler, J.R., "Response of the Adaptive Line Enhancer to Chirped and Doppler-Shifted Sinusoids," IEEE Trans. on Acoustics, Speech and Signal Processing, 1979.

43. Larimore, M.G.; Treichler, J.R.; and Johnson, C.R. Jr., "SHARF - An Algorithm for Adapting IIR Digital Filters," IEEE Trans. on Acoustics, Speech and Signal Processing, 1979.

44. Rickard, John T. and Zeidler, James R., "ROC Performamce of the Adaptive Line Enhancer/DFT Detection Processor," Proc. of Asilomar Conf. on Circuit Systems and Computers, Pacific Grove, CA, 1979.

45. Etter, Delores M., "Convergence Properties of the Adaptive Delay Element in Delay-Lock Loops and Frequency Tracking," Proc. of Asilomar Conf. on Circuit Systems and Computers, Pacific Grove, CA, 1979.

46. Rickard, J.T. and Zeidler, J.R., "Second-Order Output Statistics of the Adaptive Line Enhancer," IEEE Trans. on Acoustics, Speech and Signal Processing, Vol. ASSP-27, No. 1, February 1979.

47. Treichler, J.R., "Transient and Convergent Behavior of the Adaptive Line Enhancer," IEEE Trans. on Acoustic Speech and Signal Processing, Vol. ASSP-27, No. 1, February 1979.

48. Griffiths, L.J. and Jim, C.W., "A New Digital Adaptive Beamforming System for Microwave Radar Arrays," Septieme Colloque Sur Le Traitement Du Signal Et Ses Applications, Nice, France, 28 May to 2 June 1979.

49. Sondhi, M.M., "Image Restoration: The Removal of Spatially Invariant Degradations," Proc. IEEE, Vol. 60, No. 7, July 1972, pp. 842-853.

50. Bard, Jonathan, Nonlinear Parameter Estimation. New York: Academic Press, 1974.

51. Artzy, Ehud; Gabor, Tommy E.; and Compoter, T., "Quadratic Optimization for Image Reconstruction," in Graphics and Image Processing, pp. 242-261 (1979).

52. Clarendon, Wilkinson, The Algebraic Eigenvalue Problem. Oxford Press, 1965.

53. Shah, B.V.; Buehler, R.J. and Kempthorne, "Some Algorithms for Minimizing a Function of Several Variables," J. SIAM, 12, pp. 74-92, 1964.

54. Fletcher, R. and Reeves, C.M., "Function Minimization by Conjugate Gradients," Computer J., 7, 1964, pp. 149-154.

55. Hestenes, M.R. and Stiefel, E., "Methods of Conjugate Gradients for Solving Linear Systems," Journal of Research of the National Bureau of Standards, Vol. No. 6, December 1952, pp. 409-436.

56. Powell, M.J.D., "Some Convergence Properties of the Conjugate Gradient Method," Mathematical Programming, Vol. 11, pp. 42-49, 1976.

57. Davidson, W.C., "Variable Metric Method for Minimization," A.E.C. Res. Dev. Rep. ANL-5990 (rev.), 1959.

58. Fletcher, R.,"Optimization Symposium of the Institute of Mathematics and Its Application." University of Keele, England, 1960.

59. Adby and Dempster, Introduction to Optimization Methods, Wiley and Sons.

60. Daniel, James W., "Convergence of the Conjugate Gradient Method with Computationally Convenient Modifications," Numer. Math., 10, 1967, pp. 125-131.

61. Dorny, A Vector Space Approach to Models and Optimization, Wiley Interscience.

62. Heine, V., "Models for Two-Dimensional Stationary Stochastic Processes," Biometrika, Vol. 42, 1955, pp. 170-178.

63. Householder, A.S., "The Approximate Solution of Matrix Problems," J. Assoc. Comp. Mach., 5, 1958, pp. 205-243.

64. Beckman, F.S., "The Solution of Linear Equations by the Conjugate Gradient Method," Mathematical Methods for Digital Computers, A. Ralston and H.S. Wilf (Eds.); Wiley, New York, 1960.

65. Powell, M.J.D., "An Iterative Method for Finding Stationary Values of a Function of Several Variables," Computer Journal, Vol. 5, No. 2, 1962.

66. Swerling, P., "Statistical Properties of the Contours of Random Surfaces," IRE Trans. Inf. Th., July 1962,

67. Whittle, P., "Stochastic Processes in Several Dimensions," Bull. Int. Stat. Inst., Vol. 40, 1963, pp. 974-994.

68. Karnopp, D.C., "Random Search Techniques for Optimization Problems," Automatica, Vol. 1, 1963, pp. 111-121.

69. Fletcher, R. and Powell, M.J.D., "A Rapidly Converging Descent Method for Minimization," Computer J., 6, 1963, pp. 163-168.

70. Kantorovich, L.V. and Akilov, G.P., Functional Analysis on Normed Spaces, Macmillan, New York, 1964.

71. Box, J.J., "A Comparison of Several Current Optimization Methods, and the Use of Transformations in Constrained Problems," Computer J., 9, No. 1, 1966.

72. McMurtry, G.J. and Fu, K.S., "A Variable Structure Automation Used as a Multimodel Searching Technique, IEEE Trans. on Automatic Control, July 1966.

73. Fletcher, R., "Function Minimization Without Evaluating Derivatives, a Review, Computer J., Vol. 8, No. 1, 1965.

74. Fletcher, R., Optimization, Academic Press.

75. Ralston, A First Course in Numerical Analysis, McGraw-Hill.

76. Schwartz and Shaw, Signal Processing, McGraw-Hill.

77. Yakwitz, Mathematics of Adaptive Control Processes, American Elsevier Publishing Company, Inc.

78. Applied Optimal Estimation, Gelb, Ed., The M.I.T. Press.

79. Henrici, Peter, The Siam Series in Applied Mathematics, Wiley.

80. Van Trees, Detection, Estimation and Modulation Theory, Wiley.

81. Shubert, Bruno O. and Larson, Harold J., Probabilistic Models in Engineering Sciences, Vol. I, II, John Wiley and Sons, 1979.

82. Rall, Louis B., Error in Digital Computation, Vol. 2, John Wiley and Sons, Inc.

83. Papoulis, Signal Analysis, McGraw-Hill.

84. Householder, The Theory of Matrices in Numerical Analysis, Blaisdell Publishing Co.

85. Goldfield, S.M.; Quandt, R.E.; and Trotter, H.F., "Maximization by Quadratic Hill-Climbing," Econometrica, 34, 1966, pp. 541-551.

86. Franks, L.E., "A Model for the Random Video Process," Bell Sys. Tech. J., Vol. 45, No. 4, April 1966, pp. 609-630.

87. Wong, E., "Two-Dimensional Random Fields and Representation of Images," SIAM J. Appl. Math, Vol. 16, No. 4, July 1968, pp. 756-770.

88. Rushforth, C.K. and Harris, R.W., "Restoration, Resolution, and Noise," J. Opt. Soc. Amer., Vol. 58, No. 4, April 1968, pp. 539-545.

89. Myers, G.E., "Properties of the Conjugate-Gradient and Davidon Methods," Journal of Optimization Theory and Applications, Vol. 2, No. 4, 1968.

90. Luenberger, D.G., Optimization by Vector Space Methods, Wiley, New York, 1969.

91. Nahi, N.E., Estimation Theory and Applications, John Wiley & Sons, Inc., New York, 1969.

92. Owsley, N.L.,"A Constrained Gradient Search Method With Application to Adaptive Sensor Arrays," Navy Underwater Sound Lab., Fort Trumbull, New London, Conn., USL Technical Memorandum No. 2242-207-69, 3 September 1969, AD B 026993.

93. Richardson, W.H., "Bayesian-Based Iterative Method of Image Restoration," J. Opt. Soc. Amer., Vol. 62, No. 1, January 1972, pp. 55-59.

94. Hunt, B.R., "The Application of Constrained Least Squares Estimation to Image Restoration by Digital Computer," IEEE Trans. Comp., Vol. C-22, No. 9, September 1973, pp. 805-812.

95. Nahi, N.E. and Franco, C.A., "Recursive Image Enhancement--Vector Processing," IEEE Trans. Comm., April 1973, pp. 305-311.

96. Rastrigin, L.A., "Random Search in Problems of Optimization, Identification and Training of Control Systems, Journal of Cybernetics, 1974, 3, pp. 93-103.

97. Polak, E.; Sargent, R.W.H.; and Sebastian, D.J., "On the Convergence of Sequential Minimization Algorithms," J. Opt. Theory and Applications, Vol. 14, No. 4, 1974.

98. McCormick, G.P. and Ritter, K., "Alternative Proofs of the Convergence Properties of the Conjugate-Gradient Method," J. Opt. Theory and Applications, Vol. 13, No. 5, 1974.

99. Larichev, O.I. and Gorvits, G.G., "New Approach to Comparison of Search Methods Used in Nonlinear Programming Problems," J. Opt. Theory and Applications, Vol. 13, No. 6, 1974.

100. Powell, S.R. and Silverman, L.M., "Modeling of Two-Dimensional Covariance Functions with Applications to Image Restoration," IEEE Trans. Aut. Contr., Vol. AC-19, No. 1, February 1974, pp. 8-13.

101. Habibi, A. and Robinson, G.S., "A Survey of Digital Picture Coding," Computer, Vol. 7, No. 5, May 1974, pp. 22-34.

102. Andrews, H.C., "Digital Image Restoration: A Survey," Computer, Vol. 7, No. 5, May 1974, pp. 36-45.

103. Anderson, B.D.O. and Jury, E.I., "Stability of Multi-Dimensional Digital Filters," IEEE Trans. Circ. and Sys., Vol. CAS-21, No. 2, March 1974, pp. 300-304.

104. Mersereau, R.M. and Dudgeon, D.E., "Two-Dimensional Digital Filtering," Proc. IEEE, Vol. 63, No. 4, April 1975, pp. 610-623.

105. Hunt, B.R., "Digital Image Processing," Proc. IEEE, Vol. 63, No. 4, April 1975, pp. 610-623.

106. Attasi, S., "A Generalization of 'Kalman' Statistical Techniques to Image Processing," IRIA Report, Domaine de Voluceau, Rodquencourt, France, January 1975.

107. Asher, R.B., "Recursive Estimation in Image Enahncement: A Tutorial Review," Report from Frank J. Seiler Research Lab, U.S. Air Force Academy, Colorado Springs, to appear.

108. Mascarenhas, N.D.A., and Pratt, W.K., "Digital Image Restoration under a Regression Model," IEEE Trans. Circ. and Sys., Vol. CAS-22, No. 3, March 1975, pp. 252-266.

109. Aboutalib, A.O. and Silverman, L.M., "Restoration of Motion Degraded Images," IEEE Trans. Circ. and Sys., Vol. CAS-22, No. 3, March 1975, pp. 278-286.

110. Nahi, N.E. and Habibi, A., "Decision-Directed Recursive Image Enhancement," IEEE Trans. Circ. and Sys., Vol. CAS-22, No. 3, March 1975, pp. 286-293.

111. Lucky, R.W., "Automatic Equalization for Digital Communication," BSTJ, April 1975.

112. Price, W.L., "A Controlled Random Search Procedure for Global Optimization," The Computer Journal, Vol. 20, No. 4, 1976.

113. Oren, S.S., "On Quaisi-Newton and Pseudo-Newton Algorithms," J. Opt. Theory and Applications, Vol. 20, No. 2, Oct 1976.

114. Allwright, J.C., "Conjugate Gradient Versus Steepest Descent," J. Opt. Theory and Applications, Vol. 20, No. 1, September 1976.

115. Specicato, E., "A Variable-Metric Method for Function Minimization Derived from Invariancy to Nonlinear Scaling," J. Opt. Theory and Applications, Vol. 20, No. 3, November 1976.

116. Tits, A., "Some Investigations about a Unified Approach to Quadratically Convergent Algorithms for Function Minimization," J. Opt. Theory and Applications, Vol. 20, No.4, December 1976.

117. Johnson, C.R.; Larimore, M.G., "Comments on and Additions to 'An Adaptive Recursive LMS Filter'" Proc. IEEE, September 1977.

118. Tsypkin, Ya Z., "Some Properties of Random Search," Automation and Remote Control, Vol. 38, November 1977.

119. Tapia, R.A., "Diagonalized Multiplier Methods and Quasi-Newton Methods for Constrained Optimization," J. Opt. Theory and Applications, Vol. 22, No. 2, June 1977.

120. Han, S.P., "A Globally Convergent Method for Nonlinear Programming," J. Opt. Theory and Applications, Vol. 22, No. 3, July 1977.

121. Mangasarian, O.L., "Solution of Symmetric Linear Complementarity Problems by Iterative Methods," J. Opt. Theory and Applications, Vol. 22, No. 4, August 1977.

122. Ting, A.C., "On the Convergence of a Class of Derivative-Free Minimization Algorithms," J. Opt. Theory and Applications, Vol. 22, No. 4, August 1977.

123. Devroye, L.P., "The Uniform Convergence of Nearest Neighbor Regression Function Estimators and Their Application in Optimization," IEEE Trans. on Information Theory, March 1978.

124. Chang, Peter Y. and Meadows, henry E., "Gradient Methods Applied in Model Reference Adaptive Algorithms," Proc. of Asilomar Conf. on Circuit Systems and Computers, Pacific Grove, CA, 1978.

125. Pereria, Filho, J.L., Interframe Image Processing with Application to Target Detection and Tracking, Ph.D. Thesis, Naval Postgraduate School, Monterey, CA, 1979.

126. Koray, A. Muhtar, Focal Plane Processing Techniques to Suppress IR Background Clutter, EE Thesis, Naval Postgraduate School, Monterey, CA, 1979.

127. Celik, K., Focal Plane Signal Processing for Clutter Suppression and Target Detection, EE Thesis, Naval Postgraduate School, Monterey, CA, 1979.

128. Titus, H.A. and Pereira, J., "Interframe Image Processing with Application to Target Detection and Tracking," 13th Annual Asilomar Conf. on Circuit Systems and Computers, November 1979.

129. Otazo, J.J., "Digital Filters for the Detection of Resolved and Unresolved Targets Embedded in Infrared Scenes," SPIE, Vol. 178, Smart Sensors, 1979.

130. Boland, W.R. and Kowalik, J.S., "Extended Conjugate-Gradient Methods with Restarts," J. Opt. Theory and Applications, Vol. 28, No. 1, May 1979.

131. Benson, H.P., "Vector Maximization with Two Objective Functions," J. Opt. Theory and Applications, Vol. 28, No. 2, June 1979.

132. Gopalakrishnan Nair, G., "On the Convergence of the Louis-Jaakola Search Method," J. Opt. Theory and Applications, Vol. 28, No. 3, July 1979.

133. Dennis, J.E., Jr. and Met, H.H.W., "Two New Unconstrained Optimization Algorithms which Use Function and Gradient Values," J. Opt. Theory and Applications, Vol. 28, No. 4, August 1979.

134. Eveleigh, Adaptive Control and Optimization Techniques, McGraw-Hill.

135. Rustagi, Optimizing Methods in Statistics, Academic Press.

136. Huang, H.Y., "Unified Approach to Quadratically Convergent Algorithms for Function Minimization," J. Opt. Theory and Applications, Vol. 5, No. 6, 1970.

137. Powell, M.J.D., "A Survey of Numerical Methods for Unconstrained Optimization," SIAM Rev., 12, 1, 1970, pp. 79-97.

138. Tokumaru, H.; Adachi, N.; and Goto, K., "Davidon's Method for Minimization Problems in Hilbert Space with an Application to Control Problems," SIAM J. Control, 8, May 1970, pp. 163-178.

139. Adachi, N., "On Variable-Metric Algorithms," J. Opt. Theory and Applications, Vol. 7, No. 6, 1971.

140. Powell, M.J.D., "Problems Related to Unconstrained Optimization," Chapter 3 of Numerical Methods for Non-Linear Optimization. F.A. Lootsma (Ed.), Academic Press, New York, 1972.

141. Nahi, N.E. and Assefi, T., "Bayesian Recursive Image Estimation," IEEE Trans. Comp., Vol. C-21, No. 7, July 1972, pp. 734-738.

142. Pratt, W.K., "Generalized Wiener Filtering Computation Techniques," IEEE Trans. Comp., Vol. C-21, No. 7, July 1972, pp. 636-641.

143. Woods, J.W., "Two-Dimensional Discrete Markovian Fields," IEEE Inf. Th., Vol. IT-18, No. 2, March 1972, pp. 232-240.

144. Huang, T.S., "Stability of Two-Dimensional Recursive Filters," IEEE Trans. Aud. Electr., Vol. AU-20, No. 2, June 1972, pp. 158-163.

145. Nahi, N.E., "Role of Recursive Estimation in Statistical Image Enhancement," Proc. IEEE, Vol. 60, No. 7, July 1972, pp. 872-877.

146. Habibi, A., "Two-Dimensional Bayesian Estimate of Images," Proc. IEEE, Vol. 60, No. 7, July 1972, pp. 878-883.

147. Amir, Haim and Kodres, Uno R., "The TRW Multiplier Chip Used to Enhance the Intel 8086 Arithmetic Capability." Mini-Micro Computer J., January 1980.

148. "Real Time Signal Processing" Ed. T. Tao, Proceedings SPIE, Vol. 154, 1978.

149. "Real Time Signal Processing II" Ed. T. Tao, Proceedings SPIE, Vol. 180, 1979.

150. "Real Time Signal Processing III" Ed. T. Tao, Proceedings SPIE, Vol. 241, 1980.

151. "High Speed Computer and Algorithm Organization" Ed. D, Kuck, D. Lawrie, A. Sameh. Academic Press, 1977.

152. Thurber, K., Large Scale Computer Architecture. Hayden, 1976.

153. Parallel Processor Systems, Technologies and Applications, Ed. L. Hobbs. Spartan Book, 1970.

154. Parallel Processing Springer Verlag, 1975.

155. "Proceedings of Annual International Conference on Parallel Processing."

156. Hintz, R. and Tate, D., "Control Data STAR-100 Processor Design," Proceeding COMPCON, 1972.

157. Sites, R., "An Analysis of the CRAY-1 Computer," Proc. Symposium Computer Architecture, 6, 1978.

158. Texas Instruments, Inc. "A Description of the Advanced Scientific Computer System," 1973.

159. Datawest Corp., "Real Time Series of Microprogrammable Array Transform Processor."

160. Control Data Corporation, "Advanced Flexible Processors," 1979.

161. Thurber, K. and Masson, G., Distributed Processor Communication Architecture. Lexington Books, 1979.

162. Thurber, K., "An Assessment of the Status of Network Architectures," Proc. COMPCON, Fall '80, pp. 87-94, 1980.

163. "Local Area Networking," Ed. I. Cotten, U.S. Dept. of Commerce, National Bureau of Standards Publication 500-31, 1978.

164. Clark, D. et al., "An Introduction to Local Area Networks," Proc. IEEE, 66, pp. 1497-1517, 1978.

165. Kimbleton, S. and Schaneider, G., "Computer Communication Networks: Approaches, Objectives and Performance Consideration," ACM Computing Survey, 1975.

166. "Proceedings of Annual Fault Tolerant Computing Systems ", 198

167. Computer, Special Issue on "Fault Tolerant Computing," March 1980.

168. Stiffler, J. and VanDoren, A., "FTSC-Fault Tolerant Spaceborne Computer," Proceedings Fault Tolerant Computing Systems, 9, p. 143, 1979.

169. Wensley, J.; Lamport, L.; Goldberg, J., et al., "SIFT: Design and Analysis of a Fault Tolerant Computer for Aircraft Control," Proc. IEEE, 66, pp. 1240-1255, 1978.

170. August Systems, "A Family of Ultra-Reliable Computers," 1979.

171. Berg, R. et al, "PEPE-An Overview of Architecture, Operation and Implementation," Proc. National Elect. Conf., pp. 312-317, 1972.

172. Wulfand, W. and Harbison, S., "Reflections in a Pool of Processors--An Experience Report on Cmmp/Hydra," Proc. National Computer Conf., pp. 939-951, 1978.

173. Swan, R; Bechtolsheim, A.; Lai, K.; and Ousterhout, J., "The Implementation of the Cm* Multi-microprocessor," Proc. National Computer Conf., pp. 645-655, 1977.

174. Despain, A., "X-Tree: A Multiple Microcomputer System," Proc. COMPCON, Spring '80, pp. 324-327, 1980.

175. Sullivan, H.; Bashkow, T.; and Klappholz, D., "A Large Scale Homogeneous, Fully Distributed Parallel Machine," Proc. Symp. on Computer Architecture, pp. 105-117, 1977.

176. Harris, J. and Smith, D., "Hierarchical Multiprocessor Organizations," Proc. Symp. on Computer Architecture, pp. 41-48, 1977.

177. Widdoes, L., "Thes-1 Project: Developing High Performance Digital Computers," Proceedings COMPCON, Spring '80, pp. 282-291, 1980.

178. Mazare, C., "A Few Examples of How to Use a Symmetrical Multimicroprocessor," Proc. Symp. on Computer Architecture, pp. 57-62, 1977.

179. Marsan, M.; Conte, G.; Corosoand, D.; and Gregoretti, F., "Architecture, Communication Procedures and Performance Evaluation of the μ* Multi-Microprocessor System," Proc. 1st Inter. Distributed Computer System Conf., pp. 106-115, 1979.

180. Takenouchi, H.; Eatada, M.; Miyama, K.; Inamori, H.; and Toda, A., "Parallel Processing Simulator for Network Systems Using Multi-Microcomputers," Proc. COMPCON, Fall '80, pp. 55-62, 1980.

181. Tripathi, A. and Lipovski, G., "Packet Switching in Banyan Networks," Proc. Symp. on Computer Architecture, pp. 160-167, 1979.

182. Flanagan, K. and Woo, L., "Role of Simulation in Top Down Distributed Signal Processor Design," Real Time Signal Processing III, Ed. T. Tao, Proceedings SPIE, Vol. 241, 1980.

183. Session Thursday II-4, "Application of Distributed Computing to Modular Missile-Borne Computers," Proceedings, 1st Int. Conf. Distributed Computing Systems, pp. 707-756, 1979.

184. Farber, D. and Larson, K., "System Architecture of Distributed Computer System--Communication System," Proc. Computer Communication Networks and Tele-Traffic, pp. 21-27, 1972.

185. Liu, M.; Prado, R.; Tsay, D.; Wolf, J.; Weide, B.; and Chou, C., "System Design of the Distributed Double Loop Computer Network (DDLCN)", Proc. 1st Int. Distributed Computer Conf., pp. 95-105, 1979.

186. Metcalf, R. and Boggs, D., "Ethernet: Distributed Packet Switching for Local Computer Networks," CACM, 18, pp. 395-404, 1976.

187. Thurber, K. and Freeman, H., "Architecture Considerations for Local Computer Networks," Proc. 1st Int. Conf. Distributed Computing Systems, pp. 131-141, 1979.

188. Thurber, K. and Freeman, H., "Local Computer Network Architecture," Proceedings COMPCON, Spring 79, pp. 258-261, 1979.

189. Organick, Elliott I., The Multics System, M.I.T. Press, 1974.

190. Schell, R.; Kodres, O.; Amir, H.; Wasson, J.; Tao, T.F., "Processing of Infrared Images by Multiple Microcomputer System, Proceedings SPIE, Real Time Signal Processing IV, Vol. 241, 1980.

191. Siewcork, Dan, Analysis Synthesis and Design of Reliable Fault Tolerant Distributed Data Processing Systems. Carnegie-Mellon University, 1979.

192. Madnick, Stuart E. and Donovan, John J., Operating Systems, McGraw-Hill Computer Science Series, 1974.

193. Shaw, Alan, The Logical Design of Operating Systems, Prentice-Hall, 1974.

194. Shubert, B.O., "Finite Memory Classification of Bernoulli Sequences Using Reference Samples," IEEE Trans. Info. Theory, Vol. IT-20, No. 3, pp. 384-387, 1974.

195. Joel, A., "Circuit Switching: Unique Architecture and Applications," Computer, pp. 10-22, June 1979.

196. The 8086 Family Users Manual, October 1979. Intel, Santa Clara, California.

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center 2
Cameron Station
Alexandria, Virginia 22314

2. Library, Code 0142 2
Naval Postgraduate School
Monterey, California 93940

3. Electrical Engineering Department
Chairman, Code 62 2
Department of Electrical Engineering
Naval Postgraduate School
Monterey, California 93940

4. Professor T. F. Tao, Code 62Tv 10
Department of Electrical Engineering
Naval Postgraduate School
Monterey, California 93940

5. Professor G. L. Sackman, Code 62Sa 1
Department of Electrical Engineering
Naval Postgraduate School
Monterey, California 93940

6. Professor U. R. Kodres, Code 52Kr 1
Department of Computer Science
Naval Postgraduate School
Monterey, California 93940

7. Professor J. P. Powers, Code 62Po 1
Department of Electrical Engineering
Naval Postgraduate School
Monterey, California 93940

8. Professor B. O. Shubert, Code 55Sy 1
Department of Operations Research
Naval Postgraduate School
Monterey, California 93940

9. Professor H. A. Titus, Code 62Ts 1
Department of Electrical Engineering
Naval Postgraduate School
Monterey, California 93940

10. Lieutenant Amir Haim (KUBA) 10
86/A
Hatishbi Carmel
Haifa, Israel

11. Lieutenant Rapantzikos Demosthenis 1
Karaoli 7, Salamis
Nisos Salamis, Greece

12. Lieutenant E. Giotis 1
Kountouriotov 65
Glyfada
Athens, Greece

13. Mr. Kurt Holmquist, Code 62Tv 1
Solidstate Lab
Naval Postgraduate School
Monterey, California 93940

14. Mrs. Atara Rozic 2
6040 Glenarms Drive
Oakland, California 94611

15. Mr. Richard Wooten 1
6860 Worth Way
Camarillo, California 93010

16. Commander R. A. McGonigal, Code 0305 1
Naval Postgraduate School
Monterey, California 93940

17. Lieutenant Commander J. Schnieder 2
SMC #1977
Naval Postgraduate School
Monterey, California 93940

18. Lieutenant Colonel Adam Feit 1
SMC #1079
Naval Postgraduate School
Monterey, California 93940

Thesis
A4348 Amir 190668
c.1 Adaptive image processing and implementation by multiple microcomputer system.