1

A

52762

National Library
of CanadaBibliothèque nationale
du CanadaCANADIAN THESES
ON MICROFICHETHÈSES CANADIENNES
SUR MICROFICHE

NAME OF AUTHOR/NOM DE L'AUTEUR SUN-SIU DAVID CHAN

TITLE OF THESIS/TITRE DE LA THÈSE INVESTIGATION OF A MIXED DIGITAL AND ANALOG
ANALOG ADAPTIVE FILTER

UNIVERSITY/UNIVERSITÉ CARLETON UNIVERSITY

DEGREE FOR WHICH THESIS WAS PRESENTED/
GRADE POUR LEQUEL CETTE THÈSE FUT PRÉSENTÉE MASTER OF ENGINEERING

YEAR THIS DEGREE CONFERRED/ANNÉE D'OBTENTION DE CE DEGRÉ 1981

NAME OF SUPERVISOR/NOM DU DIRECTEUR DE THÈSE PROF. C. H. CHAN

Permission is hereby granted to the NATIONAL LIBRARY OF
CANADA to microfilm this thesis and to lend or sell copies
of the film.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈ
QUE NATIONALE DU CANADA de microfilmer cette thèse et
de prêter ou de vendre des exemplaires du film.

The author reserves other publication rights, and neither the
thesis nor extensive extracts from it may be printed or other-
wise reproduced without the author's written permission.

L'auteur se réserve les autres droits de publication; ni l
thèse ni de longs extraits de celle-ci ne doivent être imprimés
ou autrement reproduits sans l'autorisation écrite de l'auteur

DATED/DATE JAN 16, 1980

SIGNED/SIGNÉ

PERMANENT ADDRESS/RÉSIDENCE FIXE 44-46 BOUNDARY ST. E/F FLAT "C"

FUK LUK BLDG. KOWLOON, HONG KONG

National Library of Canada
Collections Development Branch

Canadian Theses on
Microfiche Service

Bibliothèque nationale du Canada
Direction de développement des collections

Service des thèses canadiennes
sur microfiche

NOTICE

The quality of this microfiche is heavily dependent upon the quality of the original thesis submitted for microfilming. Every effort has been made to ensure the highest quality of reproduction possible.

If pages are missing, contact the university which granted the degree.

Some pages may have indistinct print especially if the original pages were typed with a poor typewriter ribbon or if the university sent us a poor photocopy.

Previously copyrighted materials (journal articles, published tests, etc.) are not filmed.

Reproduction in full or in part of this film is governed by the Canadian Copyright Act, R.S.C. 1970, c. C-30. Please read the authorization forms which accompany this thesis.

THIS DISSERTATION
HAS BEEN MICROFILMED
EXACTLY AS RECEIVED

AVIS

La qualité de cette microfiche dépend grandement de la qualité de la thèse soumise au microfilmage. Nous avons tout fait pour assurer une qualité supérieure de reproduction.

S'il manque des pages, veuillez communiquer avec l'université qui a conféré le grade.

La qualité d'impression de certaines pages peut laisser à désirer, surtout si les pages originales ont été dactylographiées à l'aide d'un ruban usé ou si l'université nous a fait parvenir une photocopie de mauvaise qualité.

Les documents qui font déjà l'objet d'un droit d'auteur (articles de revue, examens publiés, etc.) ne sont pas microfilmés.

La reproduction, même partielle, de ce microfilm est soumise à la Loi canadienne sur le droit d'auteur, SRC 1970, c. C-30. Veuillez prendre connaissance des formules d'autorisation qui accompagnent cette thèse.

LA THÈSE A ÉTÉ
MICROFILMÉE TELLE QUE
NOUS L'AVONS REÇUE

INVESTIGATION OF A MIXED
DIGITAL AND SAMPLED ANALOG ADAPTIVE FILTER

by

S. David Chan, B. Eng.

A thesis submitted to the Faculty of Graduate
Studies and Research in partial fulfillment
of the requirements for the degree of

Master of Engineering

Department of Electronics
Faculty of Engineering
Carleton University
Ottawa, Canada
November 1980

The undersigned recommend to the Faculty of Graduate Studies
the acceptance of the thesis:

"Investigation of A Mixed Digital
and Sampled Analog Adaptive Filter"

submitted by **D. David Chan** in partial fulfillment of the
requirements for the degree of Master of Engineering.

Prof. C.H. Chan
Thesis Supervisor
Prof. A.R. Boothroyd
Chairman
Department of Electronics

November, 1980

ABSTRACT

A novel mixed digital and sampled analog adaptive filter is presented in this thesis. The basic computational elements, a multiplying digital to analog converter and a switched capacitor integrator, are time-shared so that the hardware complexity is highly reduced. The approach combines the accuracy and flexibility of digital signal processing with the simplicity of the analog signal processing. No analog to digital (A/D) or digital to analog (D/A) conversion is required for the signal to be processed.

A 32 tap mixed digital and sampled analog programmable transversal filter (PTF) was designed and built to be used as the main processor of the adaptive filter. The least mean square (LMS) algorithm was hardware implemented using digital approximation. The PTF and the adaptive filter were characterized as a frequency selective filter and a parameter estimator, respectively. The experimental results were analysed and discussed.

ACKNOWLEDGEMENTS

The author wishes to express his indebtedness to his thesis supervisor, Prof. C.H. Chan, for his support, excellent guidance and thoughtful suggestions during the course of this research, as well as in the production of this thesis.

The author also wishes to thank Dr. D. Roy of Interactive Circuits and Systems, Ottawa, for his interesting suggestions while the work was in progress, to Mr. Eric Samson, for his careful and efficient scrutiny of the thesis' draft, and to Miss Ellen Ip, for her untiring and speedy typing of this thesis.

Financial support from the Natural Sciences and Engineering Research Council of Canada is gratefully acknowledged. .

TABLE OF CONTENTS

	<u>PAGE</u>
List of Symbols and Abbreviations	iii
List of Figures	vi
 CHAPTER 1. INTRODUCTION	
1.1 Introduction	1
1.2 The Need For Adaptive Filter	2
1.3 Thesis Objective	6
1.4 Thesis Format	9
 CHAPTER 2. TRANSVERSAL AND ADAPTIVE FILTER	
2.1 Introduction	11
2.2 Transversal Filter	11
2.3 Programmable Transversal Filter	14
2.4 Adaptive Filter	16
2.5 Error Analysis	18
2.6 The Steepest Descent Approach	24
2.6.1 Convergence Properties of The LMS Algorithm	26
2.6.2 Some Nonlinearity For Faster Convergence	29
2.7 Conclusion	33
 CHAPTER 3. IMPLEMENTATION	
3.1 Introduction	34
3.2 The Programmable Transversal Filter	34
3.2.1 All Digital Approach	35
3.2.2 All Analog Approach	36
3.2.3 Mixed Digital and Sampled Analog Approach	39
3.3 System Organization of The Proposed Adaptive Filter	41
3.3.1 Programmable Transversal Filter Realization	44
3.3.2 LMS Algorithm Realization	44
3.4 Hardware Implementation of The Adaptive Filter	51
3.4.1 Four Quadrant Multiplying Digital to Analog Converter	54
3.4.2 Sequential Analog Summer	54
3.4.3 Digital Realization of The Tap weight Modification	58
3.4.4 Analog Extractor and Error Generator	61
3.5 Summary	61

CHAPTER 4. EXPERIMENTAL RESULTS

4.1	Introduction	63
4.2	Characterization of The Programmable Transversal Filter	63
4.3	Adaptive Filter Performance	68
4.3.1	Operation of The Parameter Estimator	68
4.3.2	Adapting to FIR System	70
4.3.3	Adapting to An IIR System	75
4.4	System Error Analysis	82
4.4.1	Quantization Effects on The Adaptive Filter	82
4.4.2	Possible Error in the PTF	89
4.5	Comparison and Conclusion	93

CHAPTER 5. CONCLUSIONS.

5.1	Summary	97
5.2	Recommendation For a Modification to The Present System	98
5.3	Future Work	100

References	103
Appendix A	106
Appendix B	117

LIST OF SYMBOLS AND ABBREVIATIONS

$[\alpha]$	Correlation vector
$[A]^T$	Transpose of matrix A
A/D	Analog to digital conversion
ADC	Analog to digital converter
BBD	Bucket brigade device
CCD	Charge coupled device
CMOS	Complementary metal-oxide semiconductor
CTD	Charge transfer device
D/A	Digital to analog conversion
DAC	Digital to analog converter
$d(n)$	Desire signal at n^{th} filtering operation
$\epsilon(n)$	Error signal after n^{th} filtering operation
$\Delta\epsilon_1, \Delta\epsilon_2$	Error deadzone values
MSE,	Mean square error
EPROM	Erasable programmable read only memory
f_c	Sampling frequency
$\phi(\epsilon)$	Nonlinearity function of error
ϕ_n, ϕ_x , etc.	Latch clocks
$\psi(x)$	Nonlinearity function of input signal
FIR	Finite impulse response
h_k	Tap weight for the k^{th} tap
$h_k(n)$	Tap weight for the k^{th} tap at n^{th} filtering operation
h_k^f	Tap weight for the k^{th} tap after convergence

$H(f)$	Frequency response
Hz	Hertz
IIR	Infinite impulse response
LMS	Least mean square algorithm
LSI	Large scale integration
$\{\lambda\}$	Eigenvalues of covariance
M_1, M_2	Bits for modification
MDAC	Multiplying digital to analog converter
msec	Millisecond
mV	Millivolt
MOS	Metal-oxide-semiconductor
N	Number of stages of the transversal filter
N-MOS	N channel metal-oxide-semiconductor
PTF	Programmable transversal filter
$R(S)$	Reference signal in frequency domain
r	Geometric ratio of
σ_x	Input signal power
∇_h	Gradient vector
τ	Time constant
T	Sampling period
TDL	Tapped delay line
TF	Transversal filter
TTL	Transistor-transistor logic
μ	Convergence factor
VLSI	Very large scale integration

$x(n)$	Input sample at n^{th} filtering operation
$x(n)$	Decoded input sample
$x(n-k)$	Input sample at k^{th} tap during the n^{th} filtering operation
Δx	Signal deadzone
$X(S)$	Input signal in frequency domain
$y(n)$	Programmable transversal filter output after n^{th} filtering operation
$Y(S)$	Programmable transversal filter output in frequency domain
$H(Z)$	Z-transform of $h(t)$

LIST OF FIGURES

<u>FIGURE NO.</u>		<u>PAGE</u>
1.1	A simplified adaptive system	3
1.2	A simplified structure of a transversal filter	3
1.3	Block diagram of a simplified adaptive filter	5
1.4	A simplified predictive coding system	5
1.5	Block diagram of a PTF with digital tap weights	8
1.6	Block diagram showing the time sharing of the MDAC	8
2.1	An N-tap transversal filter	12
2.2	The block diagram of a PTF	15
2.3	A parameter estimator	17
2.4	The Mean Square Error surface in tap weight vector space	22
2.5	Schematic of an adaptive filter with LMS algorithm	27
2.6a]	An infinite clipper	32
2.6b]	A binary staircase nonlinearity	32
2.6c]	An analog nonlinearity	32
3.1	A possible digital programmable transversal filter	37
3.2	PTF with time-sharing one MDAC	42
3.3	Simplified timing requirement of the proposed adaptive filter	43
3.4	Block diagram of the proposed adaptive filter	45
3.5	Block diagram of the programmable transversal filter	46

<u>FIGURE NO.</u>		<u>PAGE</u>
3.6	Timing diagram of the transversal filter	48
3.7	Signals at different taps for two consecutive sampling period	50
3.8	Block diagram of the proposed adaptive filter	52
3.9	Timing diagram of the proposed adaptive filter	53
3.10	Schematic diagram of a multiplying digital to analog converter	55
3.11a]	MDAC two quadrant multiplier	56
3.11b]	MDAC four quadrant multiplier	56
3.12a]	Schematic of the switched capacitor integrator	57
3.12b]	Switching sequence for the integrator	57
3.13	Schematic of the digital tap weight modification block	60
4.1	Rectangular window	65
4.2	Frequency translated bandpass filter response	67
4.3	Block diagram of the adaptive parameter estimation	69
4.4	Adaptive filter frequency response in parameter mode	71
4.5a]	Unknown system tap weights	73
4.5b]	Copied tap weights in the adaptive filter	73
4.6	Adaptation to a FIR Lowpass filter	74
4.7a]	Unknown system's tap weights	76
4.7b]	Adaptive filter's tap weights after convergence	76

FIGURE NO.

4.8	Measurement of convergence time for the adaptive filter when adapting to a FIR	77
4.9	Frequency response of the IIR filter and the adaptive filter	79
4.10	Frequency response of the adaptive filter and the residue error power	80
4.11	Display of adaptation to an IIR lowpass filter in time domain	81
4.12	Measurement of convergence time for the adaptive filter when adapting to a FIR bandpass filter	85
4.13	Measurement of convergence time for the adaptive filter when adapting to a FIR bandpass filter	86
4.14	RMS residue error as a function of $\Delta\epsilon_1$	88
4.15	Adaptivity as function of $\Delta\epsilon_1 / \sigma_x$	90
4.16	Tap weights which correct the error introduced by the tapped delay line buffers	94
5.1	Timing requirement for the adaptive filter	99
5.2	Simplified timing requirement for the adaptive filter with burst modifications	101
A.1	Block diagram of the PTF	109
A.2	Timing diagram for PTF	111
A.3	Schematic of the logic control generation	112
A.4	Clock driver for TAD-32 TDL	113
A.5	Analog Path of the PTF	115
A.6	Four quadrant multiplier configuration	116
B.1	Block diagram of the LMS algorithm circuit	120

<u>FIGURE NO.</u>		<u>PAGE</u>
B.2	Timing requirement for the adaptive filter	121
B.3	Control logic circuit diagram	122
B.4	Quantization and digital multiplication circuit	124
B.5	Schematic of the adder/subtractor with overflow clamp	125

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Signal processing is very important in various fields such as speech and data communication, biomedical engineering, acoustics, radar and nuclear science, etc. Prior to 1950, signal processing was not widely applicable. Techniques employed for signal processing were primitive, using discrete passive and active elements. In the late 50's and early 60's, the evolution of the digital computer made signal processing applicable in almost every field of interest. However, real time signal processing was not always possible because of the hardware limitation, especially in analog form. Since digital computers were used as the main processors, analog to digital (A/D) converters and digital to analog (D/A) converters were required to interface between the digital processors and the outside world.

Ever since the mid 60's, the rapid progress in semiconductor technology has made real time signal processing practically realizable. The large scale integrated circuit (LSI) technology introduced in early 70's, which can implement a complex system on a single silicon chip, makes the application of signal processing even more attractive. The introduction of the charge transfer device (including both the bucket brigade device, BBD, and the charge coupled device,

CCD) introduces the sampled analog approach of signal processing. The introduction of the multiplying digital to analog converter (MDAC), which can perform multiplication between a digital number and an analog reference voltage, makes possible the exploitation of a mixed digital and analog approach. This new approach offers some advantages over its all digital or all analog counterparts.

This thesis describes a novel mixed digital and sampled analog adaptive filter by using a charge transfer delay line based programmable transversal filter (PTF) as the main processor. The design and experimental study of the proposed adaptive filter will be presented in this thesis.

1.2 THE NEED FOR ADAPTIVE FILTER

An adaptive system has the ability to vary its response according to its previous output and some reference inputs. A self-learning system has many applications in various fields. Applications of an adaptive filter such as automatic equalizer, echo canceller, noise canceller and parameter estimator have shown the importance of the adaptive filter, especially in telephone communication [1-5].

Figure (1.1) shows a simple adaptive system. The system response varies according to the previous output, $Y(S)$, and a reference input, $R(S)$. A decision block decides how the system response is modified. Simple systems with

Fig. (1.1) A simplified adaptive system

Fig. (1.2) A simplified structure of a transversal filter

4.

this feature have been widely used in feedback control system [6]. However, sophisticated adaptive filters in signal processing were difficult to implement due to hardware limitations. The researches on signal processing were done by computer simulations. Thus, practical applications were limited. The introduction of the hardware implemented programmable transversal filter (PTF) in the 70's started a new era in signal processing. The significant breakthrough of the PTF is its ability to implement virtually any finite impulse response (FIR) functions. The response of a PTF is defined by a set of coefficients which are usually referred to as tap weights. Figure (1.2) shows the basic structure of a FIR transversal filter (TF). The filter's output is given by:

$$\begin{aligned}
 y(n) &= x(n-1)h_1 + x(n-2)h_2 + \dots + x(n-N)h_N \\
 &= \sum_{k=1}^N x(n-k)h_k
 \end{aligned}
 \tag{1.1}$$

where N is the number of delay stages of the TF.

$x(n-k)$ is the delayed signal sample at k^{th} tap position and h_k is the tap weight for the k^{th} tap.

The importance of the PTF is its ability to vary responses which makes it a prime candidate in adaptive system applications. Figure (1.3) shows a simplified adaptive system using a PTF. This system is known as the decision directed adaptive filter. An important application in predictive coding can be achieved by using an adaptive

Fig. (1.3) Block diagram of a simplified adaptive filter

Fig. (1.4) A simplified predictive coding system

filter as a linear predictor [7-9]. The block diagram of a predictive coding system is shown in Fig.(1.4). By comparing the desired signal, $x(n)$, (the input signal in this case) with the filter output, $y(n)$, (predicted signal), an error, $e(n)$, is generated which is used to predict the next input sample, $x(n+1)$. Provided the input signal is not very fast varying, the predicted signal will match the input signal with a very small error. If the initial tap weights are identical in both the transmitting and receiving end, the input signal can be recovered at the receiving end by transmitting only the error signal. This application is particularly important because the bandwidth required for transmission is highly reduced. This concept has been studied in the past and a number of papers were published [10-12]. To make the aforementioned applications practical an adaptive filter with good performance is essential.

1.3 THESIS OBJECTIVE

Research and development work on adaptive filters has been carried out in the past two decades. However, most of the works have focused at the development of new adaptive algorithms [13-14]. The development on hardware implementation was limited by the tremendous hardware complexity and as most of the research works were based on computer simulations. Before the late 70's, the digital approach was about the only method of implementing. Not

until the end of the decade did PTF's become realizable, thanks to the introduction of sampled analog integrated circuit technology. The sampled analog approach was reported in late 70's [15-18].

In general, signals requiring processing are analog in nature. Nevertheless, pure analog processors have poor tap weight programmability. The all digital processors are excellent in tap weight programmability but require A/D and D/A conversions. By using an MDAC and an CTD tapped delay line (TDL), the mixed digital and sampled analog approach is realizable. All these approaches will be discussed in Chapter Three. However, the mixed digital and analog approach is employed in this thesis as it keeps the signal in sampled analog form throughout the course of processing eliminating the need for A/D and D/A conversions while the excellent tap weight programmability is fully exploited to perform the needed scaling or multiplying function. Figure (1.5) shows a PTF with digital tap weights which was reported by Haque and Copeland [15,16], and later also by Ahuja [14]. Roy and Menard [19] suggested that only one MDAC is required for a PTF using a time sharing technique. Figure (1.6) shows the block diagram of a PTF which time shares an MDAC. The simplicity of the one MDAC PTF approach has a high potential for monolithic integration of the complex adaptive filter.

Fig. (1.5) Block diagram of a PTF with digital tap weights

Fig. (1.6) Block diagram showing the time sharing of the MDAC

The goal of this thesis is to design and characterize a mixed digital and sampled analog adaptive filter which employs the one-MDAC PTF. The thesis is focused at the simplicity of hardware implementation so that monolithic integration of the entire system is practical. The difficulty of implementing the well known Least Mean Square (LMS) algorithm in a pipeline system is discussed and a solution is suggested.

1.4 THESIS FORMAT

Chapter One discusses the general signal processing and the motivation of the present work. The direction which leads to the proposed approach, the mixed digital and sampled analog approach with multiplexing MDAC and the use of the LMS algorithm, are presented.

Chapter Two reviews the underlying theories for transversal filtering and adaptive filtering. The LMS algorithm and the steepest descent approach are well explored. Various performance indices are derived and discussed such that the experimental results can be better appreciated.

Chapter Three compares the different approaches and techniques. The present approach with system implementation is presented in detail. Implementation difficulties are discussed. Approximations in the implementation are presented.

Chapter Four presents the experimental results and their significance. The performances of the PTF and the adaptive filter are discussed. The effects on the approximation in the implementation are discussed in detail. The difficulties with the sampled analog delay line are also discussed.

Chapter Five concludes the thesis and recommends the direction for future studies and development in this area.

CHAPTER 2

TRANSVERSAL AND ADAPTIVE FILTER

2.1 INTRODUCTION

In this chapter, theoretical background of transversal and adaptive filtering is reviewed. As the PTF plays an important role in adaptive systems, it is worthwhile to briefly review the theoretical background of the programmable transversal filtering as well. The hardware realization of a general transversal filter is briefly described. An adaptive filter, as an application of the PTF is described. The least mean square (LMS) algorithm and the steepest descent approach are presented later in this chapter. Performance indices such as convergence time and adaptivity are explored at the end of this chapter.

2.2 TRANSVERSAL FILTER

Figure (2.1) shows a simple structure for an N-tap transversal filter. The input signal, $x(t)$, is sampled and held in a tapped delay line. The input signal samples are shifted by one delay unit every T second. A set of coefficients known as tap weights (usually provided externally), when multiplied by the corresponding delayed samples generates a set of weighted products. The output, $y(nT)$, is obtained by summing all the weighted products every T second, which is given by:

Fig. (2.1) An N-tap transversal filter

$$y(nT) = h_0 x(nT) + h_1 x[(n-1)T] + \dots \\ \dots + h_{N-1} x[(n-N+1)T] \quad (2.1)$$

Normalizing time t with sampling time T , Eqn. (2.1) can be written as:

$$y(n) = \sum_{k=0}^{N-1} h_k x(n-k) \quad (2.2)$$

Note that if the input, $x(n)$, is an impulse, $\delta(n)$, then the output of the system is the impulse response, i.e.

$$y(n) = h_n \quad (2.3)$$

Thus the tap weights represent the impulse response of the transversal filter. By choosing different sets of tap weights, different impulse response can be achieved. An ideal transversal filter has infinite number of delays. In practice, however, the number of delay units, N , is finite. Therefore the impulse response exists only for N sampling periods. The transfer function of such a filter in Z domain is given by:

$$H(Z) = \sum_{k=0}^{N-1} h_k Z^{-k} \quad (2.4)$$

Note that $H(Z)$ has N poles at $Z = 0$ and N zeros spread over the Z -domain, hence implying an inherently stable system. Linear phase characteristic can be achieved by choosing symmetrical tap weights, i.e. $h_k = h_{N-k+1}$.

Hence Eqn. (2.2) can be rewritten as:

$$y(n) = \sum_{k=n-k+1}^n x(k)h_{n-k} \quad (2.5)$$

According to Eqn. (2.2), the tap weights can be thought of as stationary and the input samples can be thought of as moving across the tap weights, shifting at a rate of $1/T$ samples per second. This realization has been adopted by the author. Detailed implementation will be presented in Chapter Three.

2.3 PROGRAMMABLE TRANSVERSAL FILTER

A programmable transversal filter is a transversal filter which can perform different filter response at different time according to a specific algorithm. That is, the filter response is a function of time. Figure (2.2) shows the block diagram of a PTF. The filter response is controlled by the tap weight control algorithm. The tap weight control algorithm can be implemented in various ways for different applications. Generally, the algorithm is either implemented to select different preset tap weights at different time or to modify the existing tap weights. Hence the PTF has the real time programmability feature. Since the tap weights are time varying. Eqn. (2.1) can be written as:

Fig. (2.2) The block diagram of a PTF

$$y(nT) = h_0(nT)x(nT) + h_1(nT)x[(n-1)T] + \dots \\ \dots h_{N-1}(nT)x[(n-N+1)T] \quad (2.6)$$

Normalizing with sampling period, T , Eqn. (2.6) can be rewritten as:

$$y(n) = \sum_{k=0}^{N-1} h_k(n)x(n-k) \quad (2.7)$$

The impulse response of the programmable transversal filter is $h_k(n)$ at the n^{th} filtering operation. The filter response can be quite different for different filtering operation. This feature is extremely important in adaptive system. An adaptive filter based on this real time programmability is presented in the next section along with the review of its background theory.

2.4 ADAPTIVE FILTER

Adaptive filter has many applications such as parameter estimator, echo canceller, noise canceller and linear predictor, etc. Parameter estimator is chosen to review the basic theory behind the adaptive filter. Figure (2.3) shows the block diagram of an adaptive filter in parameter estimator mode. Note that the PTF plays an important role in the adaptive filter. The tap weight controller modifies the existing tap weights so that the filter response will match the response of the unknown system.

Fig. (2.3) A parameter estimator

In Fig. (2.2), input samples are fed to both the unknown system and the PTF. By comparing the unknown system output, $d(n)$, and the filter output, $y(n)$, an error signal, $e(n)$, is generated. The error signal decides how the tap weights are modified through an algorithm. There are many different adaptive algorithms. Among them, the well known LMS algorithm is employed and will be reviewed in the next section. Generally, the adaptive algorithm tries to minimize the error signal by adjusting the tap weights. When the error is minimized, the filter output will match very closely to the output of the unknown system. Under this condition, the adaptive filter is said to adapt to the unknown system, and the transversal filter response is a copy of the unknown system response.

A quantitative analysis of the adaptive filter with the LMS algorithm will now be presented. The actual implementation will be described in Chapter Three.

2.5 ERROR ANALYSIS

The analysis of the adaptive filter is developed based on the PTF shown in Fig. (2.2) [14]. With an input signal $x(n)$ and a set of N tap weights $[h_0(n), h_1(n), \dots, h_{N-1}(n)]$, the output $y(n)$ is given by:

$$y(n) = \sum_{k=0}^{N-1} h_k(n)x(n-k)$$

Consider the tap weights as a column vector,

$$[h(n)] = \begin{bmatrix} h_0(n) \\ h_1(n) \\ \vdots \\ h_{N-1}(n) \end{bmatrix} \quad (2.8)$$

and the input samples present at N delay units at time sequence can be designated as another column vector,

$$[X_n] = \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-N+1) \end{bmatrix} \quad (2.9)$$

Hence the output of the PTF can be written as:

$$\begin{aligned} y(n) &= [h(n)]^T [X_n] \\ &= [X_n]^T [h(n)] \end{aligned} \quad (2.10)$$

Note that the tap weights are time varying.

In Fig. (2.2), $d(n)$ is the output of the unknown system after n^{th} filtering operation as the desired response. Therefore the error signal, $\epsilon(n)$, after n^{th} filtering operation is given by:

$$\epsilon(n) = d(n) - y(n) \quad (2.11)$$

The square of this error signal is:

$$\epsilon^2(n) = d^2(n) - 2d(n)y(n) + y^2(n) \quad (2.12)$$

Substituting Eqn. (2.10) to Eqn. (2.12),

$$\begin{aligned} \epsilon^2(n) = & d^2(n) - 2d(n)[X_n]^T[h(n)] + \\ & [h(n)]^T[X_n][X_n]^T[h(n)] \end{aligned} \quad (2.13)$$

The mean square error, ξ , as the expected value of $\epsilon(n)$ is:

$$\begin{aligned} \xi \triangleq \overline{\epsilon^2(n)} = & \overline{d^2(n) - 2d(n)[X_n]^T[h(n)] +} \\ & \overline{[h(n)]^T[X_n][X_n]^T[h(n)]} \\ = & \overline{d^2(n)} - 2[\alpha]^T[h(n)] + \\ & [h(n)]^T[A][h(n)] \end{aligned} \quad (2.14)$$

where the correlation vector $[\alpha]$ between the input signals present at different delay units and the desired response is defined as:

$$[\alpha] \triangleq \overline{d(n)[X_n]^T} = \begin{bmatrix} d(n)x(n) \\ d(n)x(n-1) \\ \vdots \\ d(n)x(n-N+1) \end{bmatrix} \quad (2.15)$$

and A is defined as:

$$\begin{aligned} A \triangleq & \overline{[X_n][X_n]^T} \\ = & \begin{bmatrix} x(n) \\ x(n-1) \\ \vdots \\ x(n-N+1) \end{bmatrix} [x(n) \ x(n-1) \dots x(n-N+1)] \end{aligned}$$

$$= E \begin{bmatrix} x(n)x(n) & x(n)x(n-1) \dots x(n)x(n-N+1) \\ x(n-1)x(n) & x(n-1)x(n-1) \dots x(n-1)x(n-N+1) \\ \vdots & \vdots \vdots \vdots \\ x(n-N+1)x(n) & x(n-N+1)x(n-1) \dots x(n-N+1)x(n-N+1) \end{bmatrix} \quad (2.16)$$

where E denotes the expected value and matrix $[A]$ is a positive definite input correlation matrix.

For a filter having only 2 taps, ξ is given by:

$$\begin{aligned} \xi = & h_0^2(n) \overline{x^2(n)} + h_1^2(n) \overline{x(n-1)} + \\ & 2h_1(n)h_0(n) \overline{x(n)x(n-1)} - \\ & 2h_0(n) \overline{d(n)x(n)} - 2h_1(n) \overline{d(n)x(n-1)} + \overline{d^2(n)} \end{aligned} \quad (2.17)$$

This is a bowl shape surface in orthogonal vector space defined by the tap weights, $[h(n)]$. This is shown in Fig. (2.4). The adaptive process will try to minimize the mean square error, ξ , by modifying the tap weights for the transversal filter. The minimum ξ is at the bottom of the bowl surface such that the gradient with respect to the tap weights is zero. Note that the bottom of the bowl surface would be moving if the input signal and/or the unknown system's response are nonstationary nonergodic processes. This is the case in slowly varying channels. Thus the adaptive process must track the bottom of the

Fig. (2.4) The Mean Square Error surface in tap weight vector space

bowlshape surface. In this thesis, only the stationary and ergodic process for input signals and the desired response will be considered.

In order to find the minima of ξ , the gradient vector has to be computed. From Eqn. (2.14), the gradient vector is given by:

$$\nabla_h \xi = -2[\alpha] + 2[A][h(n)] \quad (2.18)^*$$

By equating Eqn. (2.18) to zero, the optimum tap weight can be obtained as:

$$[h^*] = [A]^{-1}[\alpha] \quad (2.19)$$

which is the Weiner-Hopf equation in matrix form [24].

Substituting $[h^*(n)]$ into Eqn. (2.14), $\xi_{\min}$ is given by:

$$\xi_{\min} = d^2(n) - [\alpha]^T[h^*(n)] \quad (2.20)$$

Substituting Eqns. (2.19) and (2.20) into (2.14), the mean square error, ξ , in terms of the minimum mean square error, $\xi_{\min}$, is given by:

$$\xi = \xi_{\min} - [h^*]^T[A][h(n)] + [h(n)]^T[A][h(n)] \quad (2.21)$$

Having found the mean square error, ξ , and its gradient vector, $\nabla_h \xi$, an algorithm which can modify the tap weights to reach their optimum values has to be derived. The next section presents the derivation of the LMS algorithm

via the steepest descent approach.

2.6 THE STEEPEST DESCENT APPROACH

This approach is based on the theory that if a particle descends on a curved surface without leaving the surface, the quickest way it can do so is by adjusting its velocity proportional to the slope of the curved surface. The same principle can be employed for the adaptive process to reach the bottom of the mean square error (MSE) bowl surface. That is, the tap weight vector is changed proportional to the negative of the gradient vector after each filtering operation [21]. This can be expressed as;

$$[h(n+1)] = [h(n)] + \frac{\mu}{2}(-\nabla_h \xi_n) \quad (2.22)$$

where (n) represents the present filtering operation, $(n+1)$ represents the next filtering operation and μ is a constant of proportionality which is called the convergence factor. Hence the recursive relation for the k^{th} tap weights is given by;

$$h_k(n+1) = h_k(n) + \frac{\mu}{2} \left(- \frac{\partial \xi}{\partial h_k(n)} \right) \quad (2.23)$$

From Eqns. (2.13) and (2.14), Eqn. (2.23) can be rewritten as:

$$h_k(n+1) = h_k(n) + \frac{\mu}{2} \left[- \frac{\partial}{\partial h_k(n)} (d^2(n) + y^2(n) - 2d(n)y(n)) \right]$$

$$\begin{aligned}
&= h_k(n) + \frac{\mu}{2} \left[-2y(n) \frac{\partial y(n)}{\partial h_k(n)} + 2d(n) \frac{\partial y(n)}{\partial h_k(n)} \right] \\
&= h_k(n) + \frac{\mu}{2} \left[2(d(n) - y(n)) \frac{\partial y(n)}{\partial h_k(n)} \right] \\
&= h_k(n) + \mu \left[\epsilon(n) \frac{\partial y(n)}{\partial h_k(n)} \right]
\end{aligned}$$

$$h_k(n+1) = h_k(n) + \mu [\epsilon(n)x(n-k)] \quad (2.24)$$

where $\epsilon(n)$ is the error signal after n^{th} filtering operation, and $x(n-k)$ is the delayed input sample at k^{th} tap position. If the minimum mean square error, $\epsilon_{\min}$, is reached after m operations, then:

$$\overline{\epsilon(n)x(n-k)} = 0 \quad \text{for } n \geq m \text{ and all } k.$$

Therefore the tap weights modification will stop as the optimal tap weights have been achieved.

Since the knowledge of cross correlation of error and the delayed signal at different tap position is very limited, a less accurate estimation of the cross correlation can be taken. Widrow et al [21] suggested that instead of performing the cross correlation, a simple multiplication can be used as follows:

$$h_k(n+1) = h_k(n) + \mu \epsilon(n)x(n-k), \quad k = 0, 1, \dots, N-1 \quad (2.25)$$

The k^{th} tap weight for the next filtering operation is obtained by simply adding the product of the present error

signal and the present delayed sample at k^{th} tap to the present k^{th} tap weight. The approach is known as the LMS algorithm.

The operation of the adaptive filter proceeds as follows. An initial set of tap weights starts the filtering operation. After the filtering operation is completed, an error signal is generated by comparing the output sample and the desired response. Then all the tap weights are modified according to Eqn. (2.25). Now the adaptive filter is ready for the next filtering operation. This recursive operation continues until the tap weights settle down to their optimum values. Figure (2.5) shows the schematic of an adaptive filter with the LMS algorithm. Some important properties of the LMS algorithm are summarized in the following subsections.

2.6.1 CONVERGENCE PROPERTIES OF THE LMS ALGORITHM

The convergence of an adaptive filter depends on the input samples, the error signal and the convergence factor, μ . The only true variable is the convergence factor, which is discussed in this subsection.

By using Eqn. (2.18), Eqn. (2.22) can be rewritten as:

$$\begin{aligned} [h(n+1)] &= [h(n)] - \mu[A][h(n)] + \mu[\alpha(n)] \\ &= [I - \mu[A]][h(n)] + \mu[\alpha(n)] \end{aligned} \quad (2.26)$$

where I is the identity matrix.

Eqn. (2.26) is an iterative process. Forysthe and

Fig. (2.5) Schematic of an adaptive filter with LMS algorithm

Moler [22] have shown that the above iterative process will converge to the optimum tap weights, $[h^*]$, provided the eigenvalues of the matrix $[I - \mu[A]]$ are less than unity in absolute value, i.e.

$$|1 - \mu\lambda_k| < 1, \quad k = 0, 1, 2, \dots, N-1 \quad (2.27)$$

where $\{\lambda_k\}$ is a set of N eigenvalues of covariance matrix $[A]$. Since $[A]$ is a positive definite matrix, it has all real and positive eigenvalues. Thus the tap weights will converge to their values provided μ satisfies the following inequality suggested by Widrow [21]:

$$0 < \mu < 2/\lambda_{\max} \quad (2.28)$$

where $\lambda_{\max}$ is the largest eigenvalue of matrix $[A]$.

Equation (2.26) is a first order difference equation which gives an exponential envelope to the tap weights as they change in each iteration. A rough approximation for the time constant of the envelope, τ , can be determined as follows:

$$\begin{aligned} \text{The geometric ratio } r &= (1 - \mu\lambda_{\max}) \\ &= e^{-T/\tau} \end{aligned}$$

where T is the sampling period. For a slow adaptation, $\tau \gg T$. Then approximation of $e^{-T/\tau}$ can be written as:

$$e^{-T/\tau} \approx 1 - T/\tau$$

$$= (1 - \mu \lambda_{\max})$$

$$\text{or } \tau = \frac{T}{\mu \lambda_{\max}} \quad (2.29)$$

For positive definite matrix [A] [ref. 23]

$$\begin{aligned} N \cdot \lambda_{\max} &< \sum_{k=0}^{N-1} \lambda_k = \text{trace [A]} \\ &= N \overline{x^2(n)} \end{aligned}$$

Hence, the adaptive time constant is:

$$\tau > \frac{T}{\mu \sigma_x^2} \quad (2.30)$$

Equation (2.30) states that the adaptive time constant, τ , is inversely proportional to the convergence factor, μ , the average input signal power, $\overline{x^2(n)}$ and the sampling frequency, $f_c (= 1/T)$. This is quite significant for the adaptive process. For a filter working at 8 KHz sampling rate with 1 V_{rms} input signal and $\mu = 0.01$, the value of τ is approximately 12.8 msec.

Note that Eqn. (2.30) is valid only if Eqn. (2.22) is used. If Eqn. (2.25) is used, the value of τ will be larger by one order as given in Eqn. (2.30) [14].

2.6.2 SOME NONLINEARITY FOR FASTER CONVERGENCE

The LMS algorithm described in Eqn. (2.20) modifies

the tap weights in the tap weight vector space directly proportional to the gradient of the mean square error, ξ . The convergence time may be short enough for some adaptive processes but a faster convergence can be obtained by modulating μ according to the value of the gradient. The idea is to implement some different increments in the modification scheme according to the size of the error. If the error is larger, hence the gradient is larger, then the modification made in tap weights should be consequently larger or vice versa. This can be achieved by introducing a nonlinearity function, ϕ , to operate on the error signal, $\epsilon(n)$, before its gradient is computed. Replacing $\overline{\epsilon^2(n)}$ by $\phi^2(\epsilon)$, Eqn. (2.22) can be rewritten as:

$$[h(n+1)] = [h(n)] + \frac{\mu}{2} (-\nabla_h \phi^2(\epsilon))$$

where $\phi(\epsilon)$ is a positive even function of error $\epsilon(n)$ such that:

$$\frac{d\phi(\epsilon)}{d\epsilon} = \phi_o(\epsilon) \text{ is a monotonically nondecreasing odd}$$

function of ϵ . Then, Eqn. (2.24) is modified as:

$$\begin{aligned} h_k(n+1) &= h_k(n) + \frac{\mu}{2} \left(\frac{-\partial}{\partial h_k(n)} [\phi^2(\epsilon(n))] \right) \\ &= h_k(n) + \frac{\mu}{2} \left(-2 \frac{\partial \phi(\epsilon(n))}{\partial \epsilon(n)} \cdot \frac{\partial \epsilon(n)}{\partial h_k(n)} \right) \\ &= h_k(n) + \mu \phi_o(\epsilon(n)) x(n-k) \end{aligned}$$

If $\phi_0(\epsilon(n)) = \epsilon(n)$ is used in the above equation, the steepest decent LMS algorithm of Eqn. (2.24) results.

Similarly, a nonlinearity function, ψ , can be implemented to the sampled signal $x(n-k)$. This nonlinearity function has the same properties as $\phi_0(\epsilon(n))$, i.e. $x(n-k)$ is replaced by $\psi(x)$ where ψ is a non-decreasing odd function of x .

Figure (2.6) shows three different kinds of nonlinearity functions. Note that a dead zone around zero is included in all three functions. The purpose of the dead zone is to stop the tap weights from wandering around the bottom of the bowl shape surface once the minima is reached closely enough.

The infinite clipper as shown in Fig. (2.6a) can be obtained simply by looking at the sign of error signal, $\epsilon(n)$ and the delayed sample, $x(n-k)$, after a comparator with designated dead zone. A fixed increment or decrement, depending on the two signs, can be made in the corresponding tap weight irrespective of the size of error or signal values. This particular scheme does not provide fast convergence but is very easy to implement. This scheme was first used by Lucky [4] in a 3-tap digital equalizer.

The binary staircase nonlinearity shown in Fig. (2.6b) is quite strong for convergence. If the error is small, the increment or decrement made to the tap weight is small. When

Fig. (2.6a) An Infinite clipper

Fig. (2.6b) A binary staircase nonlinearity

Fig. (2.6c) An analog nonlinearity

the error is large, the increment or decrement made to the tap weights are relatively larger. Instead of implementing two nonlinearity functions, one can use the same idea to implement nonlinearity on the product, $e(n)x(n-k)$, rather than $e(n)$ only.

The third type of nonlinearity shown in Fig. (2.6c) is even stronger. The continuous nonlinearity is very difficult to implement. This nonlinearity can be approximated by using the staircase technique with very fine steps. However, it is not practical in implementation.

2.7 CONCLUSION

The basic theories of transversal filters and adaptive filters have been discussed in this chapter. The mean square error has been taken as the performance index and the convergence properties of the LMS algorithm have been presented.

The reviewed theoretical background forms the foundation of this thesis. Implementation and experimental work will be presented in the following chapters.

CHAPTER 3

IMPLEMENTATION

3.1 INTRODUCTION

This chapter describes the implementation of the proposed adaptive filter. As the PTF is the main processor in the adaptive filter, different approaches of the PTF implementation are reviewed. A compromise is made among the approaches which results in a mixed digital and sampled analog approach. The present approach keeps the signal to be processed in analog form and the tap weights in digital form. The reasons for this approach become clear as this chapter proceeds. Design considerations for the main processor and the peripheral circuits are presented.

Other than the PTF, the adaptive algorithm is equally important for an adaptive filter. The implementation of the LMS algorithm with a nonlinearity function is presented. As the tap weights are in digital form, the LMS algorithm is operating in the digital domain.

A multiplexing technique is employed in the proposed adaptive filter implementation to reduce the complexity of the circuit although at the expense of a more complex timing circuitry and reduced system bandwidth.

3.2 THE PROGRAMMABLE TRANSVERSAL FILTER

Since an electrically PTF is the core of an adaptive filter, different worthy approaches are reviewed in this

section. There are mainly three approaches to be discussed in terms of performance and implementation difficulties. Each approach can perform the function as stated in Eqn. (2.7):

$$y(n) = \sum_{k=0}^{N-1} h_k(n) x(n-k)$$

Advantages and disadvantages of these approaches are discussed in the following subsections.

3.2.1 ALL DIGITAL APPROACH

Classically, the digital approach is used to realize a programmable transversal filter, especially when a computer is used as the main processor. Since all operations are performed digitally, both the input and the tap weights must be digitized. Hence an analog to digital (A/D) converter and a digital to analog (D/A) converter are required to interface with the real world. Real time processing is not always possible especially when the input signal is fast varying. A digital multiplier is needed to perform the multiplication between $h_k(n)$ and $x(n-k)$ for hardware realization. However, a digital multiplier is expensive, especially when accurate multiplication is required. Even though a time multiplexing technique can be used to share the main operators such as adders and multipliers, hardware implementation is still complicated. As the number of taps (N) is increased for improved performance, the addressing circuitry required to perform multiplexing becomes complex and speed is reduced.

However, the tap weight programmability for this approach is excellent and the noise margin is good as most operations are performed digitally. The A/D converter and D/A converter usually introduce inaccuracy but with today's sophisticated converters, a high degree of accuracy can be achieved. The tapped delay line can be realized by using memories. Figure (3.1) shows a possible architecture of a digital PTF. In this configuration, the input signal is quantized into m bit words and then stored in an N -tap delay line (Nxm bit memory). After each filtering operation, the oldest signal is dumped and the forthcoming signal is placed to the tapped delay line. Delay action is achieved by selecting the signals stored in the memory. The tap weights are quantized into n -bit words and stored in another memory location. Multiplication between the delayed signal and its corresponding tap weight is performed sequentially by a digital adder. After a filtering operation is completed, the digital summed product is converted to analog form through a digital to analog converter. Cost and speed are the main limitation of this approach. Also the overall size of the whole system would be too large for monolithic integration.

3.2.2 ALL ANALOG APPROACH

An all analog approach was made possible by the development of the charge transfer device (CTD, which includes both Bucket-Brigade Device (BBD) and Charge Coupled Device

Fig. (3.1) A possible digital programmable transversal filter

(CCD)). The CTD simplifies the implementation of the analog delay line. Mavor et al [18] reported a programmable CCD transversal filter in 1977. In their approach, the analog input is delayed and stored in a CCD delay line while the analog tap weights are held stationary on fixed sample and hold sites. The multiplication of tap weights with the analog signals, which are tapped from the CCD delay line, is obtained by using an NMOS transistor as a voltage controlled conductance multiplier (while the transistor is operating at the triode region) for each tap position. Note that the variations of the linearity, offset, gain and threshold voltage of each multiplier pose practical limitations in performance. Also, the analog tap weights need to be refreshed periodically because of the leakage current associated with MOS sample and hold circuit.

When applying such filter in an adaptive system, it is difficult to implement the LMS algorithm. The tap weights modification as stated in Eqn. (2.25):

$$h_k(n+1) = h_k(n) + \mu e(n)x(n-k), \quad k = 0, 1, 2, \dots, N-1$$

requires one analog adder and one analog multiplier for each tap. Note that each delayed signal can only modify one specific tap weight. Conceivably, duplicated circuits can be reduced by time-sharing one analog adder and one analog multiplier for all N taps. However, timing control becomes more complex and the system bandwidth is reduced.

Also a properly delayed signal must be extracted from the delay line to modify a specific tap weight. Poor accuracy in tap weight modification can be expected due to the variations associated with the analog operators.

Poor tap weight programmability and poor system performance are the main limitations of the analog PTF.

3.2.3 MIXED DIGITAL AND SAMPLED ANALOG APPROACH

This hybrid approach makes use of the MDAC and a charge transfer device. The MDAC is capable of performing multiplication between an analog voltage and a binary coded number. The resulting product is in analog form. Hence the signal to be processed remains in sampled analog form while the tap weights are digital. Filter response can be changed simply by changing the binary coded tap weights. This implies excellent tap weight programmability. As the signal to be processed is kept in sampled analog form, no A/D converter and/or D/A converter is required. Haque and Copeland [15] reported a rotating tap weight PTF using this approach in 1977. In their approach, the roles of the input signal and tap weights are exchanged. The input signal is sampled and held on fixed sites. The delayed signals are tapped through buffers to perform a convolution with their corresponding tap weights which move across the sample sites by shift registers. The gain and offset voltage variance among the buffers affect the filter performance. A fixed

pattern noise of considerable magnitude was reported [15].

In adaptive filter applications, tap weights can be modified digitally as the tap weights are binary coded. The LMS algorithm requires the existing tap weights and the delayed signal to update the tap weights. The digital tap weights can be extracted easily as they are stored in the digital memory. The delayed sampled signal at each tap must be extracted through an analog extractor. However, only one extractor is needed for all taps if the multiplexing technique is used.

A mixed digital and sampled analog 32 tap programmable transversal/adaptive filter was reported by Ahuja in 1978 [14]. Ahuja fabricated the PTF similar to the one reported by Haque and Copeland. In his version, the digital tap weights are provided outside the chip and the LMS algorithm for adaptive filter application was implemented separately. The tap weight modification block was time-shared by all tap weights. In the PTF realization, one MDAC is used for multiplication between a delayed input sample and its corresponding tap weight. The complete adaptive filter could not be integrated as the MDACs consume a considerably large fraction of the total chip area. The performance of this adaptive filter was claimed to be good.

In 1978, another mixed digital and sampled analog PTF was reported by Roy and Menard [19]. Their approach employed the multiplexing technique to reduce the number of

MDACs to one. Figure (3.2) shows the schematic of the PTF which time-multiplexes one MDAC for all taps. System bandwidth reduction can be expected. However, hardware duplication is eliminated. A commercial product, the PTF-1 programmable transversal filter board, was later developed based on the block diagram shown in Fig. (3.2). The sampling frequency of such a filter was claimed to be as high as 16 KHz. The trade-off between hardware complexity and system bandwidth is obvious. However, with less hardware complexity and VLSI technology, the monolithic integration of a complex adaptive system is possible.

3.3 SYSTEM ORGANIZATION OF THE PROPOSED ADAPTIVE FILTER

In the proposed adaptive filter, a mixed sampled analog and digital PTF is used as the main processor. The LMS algorithm is implemented to monitor the filter response. The multiplexing technique is used both in the PTF and the LMS algorithm implementations.

In the PTF implementation, the computational element (MDAC) is time shared by all taps. This implies the multiplications of the tapped signals and their tap weights are performed sequentially. Hence the tap weights can be updated just before each multiplication is performed. Figure (3.3) shows the simplified timing requirement of the proposed adaptive filter.

The block diagram of the proposed adaptive filter is

Fig. (3.2) PTF with time-sharing one MDAC

ONE COMPLETE FILTERING OPERATION

- * Each tap operation consists of a multiplication between the tapped sample and the updated tap weight and a summation of the weighted product and the previous sum.

Fig. (3.3) Simplified timing requirement of the proposed adaptive filter

shown in Fig. (3.4). The implementations of the PTF and the LMS algorithm are presented in the following subsections.

3.3.1 PROGRAMMABLE TRANSVERSAL FILTER REALIZATION

A 32 tap mixed digital and sampled analog PTF has been developed to be used as the main processor of the proposed adaptive filter. In the PTF realization, the computational element (MDAC) is time shared by all 32 taps. Figure (3.5) shows the block diagram of the PTF. The input signal is sampled and delayed in the tapped delay line at the sampling rate of f_c . Hence the contents of the tapped delay line are refreshed at the rate of f_c . The act of multiplexing the MDAC is performed by scanning the tap positions through an analog multiplexer between sample times. The rate of scanning is $32 f_c$ for a 32 tap filter. Thus the delayed samples are brought to the MDAC for multiplication with their corresponding tap weights in a sequential manner. The weighted products are summed over the sampling period to provide the output.

Figure (3.6) shows the timing diagram of the PTF. Note that ϕ_{MOD} clock is the designated time slot for tap weight modifications.

3.3.2 LMS ALGORITHM REALIZATION

For adaptive filter application, an algorithm which modifies tap weights according to certain criteria is essential. The well known LMS algorithm described in Section (2.4)

Fig. (3.4) Block diagram of the proposed adaptive filter

Fig. (3.5) Block diagram of the programmable transversal filter

is adapted to this project. Usually, the LMS algorithm modifies the present tap weights for the next filtering operation. The LMS algorithm consists of two major operations according to Eqn. (2.25). The first operation is to calculate the increment for modification according to the error signal and the delayed signal. The second operation is to modify the existing tap weight by addition. Presumably, these operations are performed after each filtering operation but before the following filtering operation as the error signal must be ready before modification can be performed. Modification of all tap weights can be performed either simultaneously or sequentially. The sequential approach time shares one tap weight modification block. Ahuja suggested a burst modification [14] which updates the tap weights sequentially while the filtering operation is stopped. This approach requires a complicated timing control circuit.

In this thesis, a different sequential approach has been developed to make use of the existing timing circuitry for the PTF. As in any pipeline processor, the timing required in performing each individual operation is to avoid the excessive loop delay such that it does not exceed one clock period. In the PTF filter implementation, a time slot has been reserved for tap weight modification (refer to Fig. (3.6)). In this approach, a tap weight is modified before the multiplication between the delayed sample, $x(n-k)$, and its corresponding tap weight, $h_k(n)$. The equation for the LMS algorithm can be algebraically arranged to demonstrate the validity of the aforementioned method. From Eqn. (2.25), the equation

* ϕ_{MOD} is the designated time slot for tap weight modifications.

Fig. (3.6) Timing Diagram of the transversal filter

for the LMS algorithm is given by:

$$h_k(n+1) = h_k(n) + \mu \epsilon(n)x(n-k)$$

By replacing n by $n-1$, the above equation can be written as:

$$h_k(n) = h_k(n-1) + \mu \epsilon(n-1)x((n-1)-k) \quad (3.1)$$

Eqn. (3.1) can be arranged as:

$$h_k(n) = h_k(n-1) + \mu \epsilon(n-1)x(n-(k+1)) \quad (3.2)$$

Hence, the valid tap weight for the immediate multiplication is obtained by using the previous error signal, $\epsilon(n-1)$, the previous tap weight, $h_k(n-1)$, and the previous delayed signal at k^{th} tap. Obviously, $\epsilon(n-1)$ and $h_k(n-1)$ can be obtained as the new error signal will not be ready before the n^{th} filtering operation is completed. Also, the previous k^{th} tap weight is available as it has not yet been updated. However, $x(n-(k+1))$ is the delayed signal appearing at $(k+1)^{\text{th}}$ tap in the n^{th} filtering operation. Figure (3.7) shows the delayed signals appearing at the taps in two consecutive sampling periods. During $t = (n-1)T$, the signals at all taps are held stationary. At $t = nT$, the oldest sample, $x(n-1-N)$ is dumped from the delay line, all the other samples are moved to the following taps, and the newest sample is placed in the first tap position. From Fig. (3.7), it can be seen that:

$$x((n-1)-k) = x(n-(k+1)) \quad \text{for } 1 \leq k \leq N$$

Fig. (3.7) Signals at different taps for two consecutive sampling periods

The above equality does not hold for $k = N$ because of the finite number (N) of delay stages. Hence the N^{th} tap weight must be modified differently. Assuming $h_N(n-1)$ is valid for the n^{th} filtering operation, then no modification is required prior to the multiplication. As soon as the filtering operation is completed, a new error signal can be generated. Then tap weight modification takes place so that the tap weight is valid for the next filtering operation. This special routine can be expressed mathematically as:

$$h_N(n+1) = h_N(n) + \mu e(n)x(n-N) \quad (3.3)$$

where $h_N(n)$ is the tap weight corresponding to the last delayed signal, $x(n-N)$, and $h_N(n+1)$ is the tap weight valid for the last delayed signal in the next filtering operation, $x((n+1)-N)$. Thus the last tap weight modification is processed by bypassing the regular tap weight modification routine. The block diagram of the proposed adaptive filter is shown in Fig.(3.8). Such implementation is essential in maintaining the timing stability. The corresponding timing diagram is shown in Fig. (3.9). The hardware implementation of the adaptive filter is presented in the next section.

3.4 HARDWARE IMPLEMENTATION OF THE ADAPTIVE FILTER

The block diagram shown in Fig. (3.8) uses some simplified functional blocks to describe the proposed adaptive filter. The hardware realization of various blocks is presented in the following subsections.

*: Normally closed, opens for the last tap operation, closed again after the new error signal is obtained

Fig. (3.8) Block diagram of the proposed adaptive filter

Fig. (3.9) Timing Diagram of the proposed adaptive filter

3.4.1 FOUR QUADRANT MULTIPLYING DIGITAL TO ANALOG CONVERTER

The MDAC is the computational element in the PTF. An MDAC does high speed multiplications between an analog voltage and a digital number to provide an analog current output. Figure (3.10) shows the schematic of an "inverted R-2R" type MDAC. Binary weighted currents flow continuously in the shunt arms of the network irrespective of the switch positions if the outputs are maintained at ground level potential (true or virtual). The CMOS switches steer the current to appropriate output lines in response to the individually applied logic levels of the digital input. The output currents can be converted to a proportional voltage by using an operational amplifier as a current to voltage converter shown in Figure (11a). For offset binary coded digital inputs, four quadrant multiplication can be obtained by using the configuration shown in Figure (11b). In this project, an 8-bit MDAC (AD 7523) is used as a four quadrant multiplier to produce weighted products sequentially.

3.4.2 SEQUENTIAL ANALOG SUMMER

The use of an MDAC as a multiplier provides an analog output, which must be summed sequentially. A switched capacitor integrator together with its timing requirement shown in Fig. (3.12) is used to serve as a sequential summer. S_1 , S_2 , S_3 and S_4 are analog switches controlled by digital signals. Serial summation of the multiplier output voltages

Fig. (3.10) Schematic diagram of a multiplying digital to analog converter

(a) MDAC two quadrant multiplier

(b) MDAC four quadrant multiplier

Fig. (3.11) MDAC multiplier configurations

(a) Schematic of the switched capacitor integrator

$t_a - t_b$: S_1 closed, S_2 open, S_3 open, S_4 open.
 C_1 charges.

$t_b - t_c$: S_1 open, S_2 closed, S_3 and S_4 open.
 Charge in C_1 is transferred to C_2
 Total charge in C_2 is proportional to output
 C_{ho} charged to V_{c2} through S_4

$t_c - t_d$: S_1 closed, S_2 open, S_3 closed, S_4 open
 C_2 discharges, beginning of new accumulation

(b) Switching sequence for the integrator

Fig. (3.12) Operation of a switched capacitor integrator

is achieved by first charging capacitor C_1 to the input voltage by closing S_1 and opening S_2 , and then transferring the charge on C_1 to C_2 by opening S_1 and closing S_2 . The output can be kept within the dynamic range of the operational amplifier by properly scaling the capacitors C_1 and C_2 . After integrating N samples, the output of the integrator, which is proportional to the total charge in C_2 , is sampled by the switch S_4 and held in capacitor C_0 . The capacitor C_2 is then discharged by closing S_3 before beginning the accumulation of a new output sample.

3.4.3 DIGITAL REALIZATION OF THE TAP WEIGHT MODIFICATION

The LMS algorithm expressed in Eqn. (3.2) requires an analog multiplier to perform the multiplication between two analog signals, $e(n-1)$ and $x((n-1)-k)$, for each tap weight modification. As the tap weights are binary coded, the product from the analog multiplier must be digitized before it can be added to the previous tap weight. The analog multipliers are either too expensive or too inaccurate for the purpose of tap weight modification. A digital method suggested by Ahuja [14] can be employed to eliminate the use of analog multipliers. This digital method is based on the fact that if the tap weights are represented by eight bit offset binary numbers, then the tap weights cannot be modified by more than two least significant bits in one modification. Otherwise, the convergence factor becomes too large, which

leads to poor performance and potential instability. By quantizing both analog signals ($\epsilon(n-1)$ and $x(n-1)-k$), a digital multiplication can be performed. The digital result can then be added to the existing tap weight directly. As the maximum number of bits required for modification is only three (one for sign and two for magnitude), the digital multiplication is rather simple to implement. By quantizing the error signal, $\epsilon(n-1)$, and the extracted delayed signal, $x((n-1)-k)$ into a 3-bit word and a 2-bit word respectively, the digital multiplication can be performed by a simple digital network. Figure (3.13) shows the schematic of the digital multiplier for two analog signals. Digitization of the analog signals is accomplished by using voltage comparators. The quantization levels will define the non-linearity function of the delayed signal and the error signal. The output of the multiplier is added to the appropriate tap weight digitally.

Since the tap weights are offset binary coded and modifications are made by addition or subtraction, overflow may occur and the system becomes unstable. To avoid overflow, the digital adder is equipped with a overflow clamp feature. Thus the digital adder will be clamped to a maximum or a minimum value, depending on the type of overflow. This is achieved by using simple logic gates. The schematic of the digital tap weight modification is shown in Fig. (3.13).

Fig. (3.13) Schematic of the digital tap weight modification block

3.4.4 ANALOG EXTRACTOR AND ERROR GENERATOR

In the LMS algorithm implementation, tap weights are updated before being multiplied by the delayed signal. This implies that the signal appearing at the MDAC is different from the signal appearing at the tap weight modification block. To ensure proper synchronization, the analog extractor delays and buffers the delayed signal. This is achieved by using a sample and hold unit.

The error signal can be generated simply by using a differential amplifier. The multiplying factor, μ , is incorporated by using an operational amplifier with variable gain.

3.5 SUMMARY

A mixed sampled analog and digital approach is employed in the proposed PTF implementation, as suggested by Roy and Menard [19]. Time multiplexing is used in the programmable transversal filter and the LMS algorithm realizations. The present approach has excellent tap weight programmability as the tap weights are in digital form. The signal to be processed is kept in sampled analog form to avoid analog to digital and digital to analog conversions. As a result of the reduced hardware complexity due to multiplexing, monolithic integration is believed to be possible. System performance and other related experimental results will be presented in Chapter 4.

Thus, the system implementation has been presented along with the hardware implementation. Detailed circuit diagrams are presented in Appendix A.

CHAPTER 4

EXPERIMENTAL RESULTS

4.1 INTRODUCTION

This chapter presents the experimental results of the adaptive filter described in Chapter Three. The adaptive filter is tested as a parameter estimator; a variety of unknown system characteristics are developed to characterize the parameter estimator performance in terms of the residue error and the convergence time. The experimental results are analysed and compared with the results obtained by Ahuja whose system is comparable to the present one. The performance of the PTF is also presented.

The implementation of the LMS algorithm described in Chapter Three has a non-linearity function and its quantization effects are presented. The tapped delay line (TDL) used for this thesis has uneven gains, and offset voltages at the buffers. The effects of these variance on the adaptive system are also discussed.

4.2 CHARACTERIZATION OF THE PROGRAMMABLE TRANSVERSAL FILTER

The 32 tap PTF described in Section (3.3) was built on a wire-wrap board for testing purposes. The detailed design and circuit diagrams are given in Appendix A. The PTF accepts a maximum input voltage of 2 volts peak to peak without clipping, and is provided with output amplifier so that any output amplitude can be obtained. The maximum

output without saturation was set at 5 volts peak to peak. The digital tap weights for the PTF were stored in an ultra-violet erasable programmable read only memory (EPROM, Intel 2708).

A sinc function and a frequency translated bandpass filter function were developed and stored in an EPROM for the evaluation of the PTF as a frequency selective filter. The frequency responses of these two filters were measured using an HP 3580 spectrum analyser. Transfer functions were obtained by using the tracking oscillator output of the spectrum analyser as the input to the PTF.

A sinc function was used to characterize the system. This function has a sampled impulse response for a 32 tap filter of:

$$\begin{aligned} h_r(k) &= 1.0 & 0 \leq k \leq 31 \\ &= 0 & \text{elsewhere} \end{aligned}$$

The frequency response is given by:

$$\begin{aligned} H_r(f) &= \sum_{k=0}^{31} h_r(k) e^{(-2 k f / f_c)} \\ &= e^{(-j32 f / f_c)} \frac{\sin(32 f / f_c)}{\sin(f / f_c)} \end{aligned}$$

where f_c is the sampling frequency.

Theoretically, the frequency response has a main lobe width of $f_c/16$. Figure (4.1) shows the filter response for the 32 point rectangular window function. The main lobe has

Scales: Vertical - 10 dB/div.
Horizontal - 500 Hz/div.
Resolution bandwidth - 30 Hz.

Upper trace: frequency response of a 32 point rectangular window function.

Lower trace: noise spectrum when the input is grounded.

Fig. (4.1) Rectangular window

a width of 900 Hz. ($.063f_c$) with the first null occurring at 450 Hz ($.031f_c$). The response closely matched with the theoretical response. The first side lobe amplitude was about -14.5 dB whereas the theoretical value is -13.5 dB. The lower trace was obtained by applying a d.c. voltage to the input. The noise level was about -80 dB except at the lower frequencies. Note that there was no sign of any fixed pattern noise as previously reported [1,14]. The noise between d.c. and few hundred hertz is due to the finite bandwidth to the measuring instrument.

A frequency translation of a Hamming window results, in a bandpass filter characteristic which is shown in Fig. (4.2). The sampled impulse response of the translated window is given by:

$$h_b(k) = (0.54 - 0.46 \cos \frac{2k\pi}{31}) \cos \frac{k\pi}{4}, \quad 0 \leq k \leq 31$$

This bandpass filter should have its centre frequency at $f_c/4$ and a main lobe width of $f_c/4$, where f_c is the Nyquist frequency ($= 1/2T$ Hz). It can be seen that the stopband attenuation was about .35 dB. The irregular amplitudes and the uneven side-lobe widths can be attributed to the tap weight quantization errors.

The experimental results demonstrate the feasibility

Top trace: frequency response

Lower trace: noise spectrum

Spectrum Analyser setting: Same as in
Fig. (4.1)

Sampling frequency = 10 KHz

Fig. (4.2) Frequency translated bandpass filter
response

of the PTF. An adaptive filter was built by using this PTF as the main processor. The experimental results of the adaptive filter are presented in the next section.

4.3 ADAPTIVE FILTER PERFORMANCE

The adaptive filter was characterized by using the adaptive filter as a parameter estimator. The detailed circuit diagram of the adaptive filter is given in Appendix B. Various parameters of the adaptive filter were obtained experimentally. The operation of the parameter estimator is presented and followed by the experimental results. The results are analysed in the subsequent section.

4.3.1. OPERATION OF THE PARAMETER ESTIMATOR

A parameter estimator is capable of copying the impulse response of an unknown system. Figure (4.3) shows the block diagram of the adaptive filter in its parameter estimator mode. A reference input, $x(n)$, is applied to both the unknown system (whose parameters, i.e. the impulse response, have to be estimated) and the adaptive filter. The output of the unknown system, $d(n)$, is the desired response which acts as the reference input to the adaptive filter for tap weight modifications. The output of the adaptive filter, $y(n)$, is compared with $d(n)$ to generate the error signal, $e(n)$. The tap weights are then updated according to the LMS algorithm for the next filtering operation. These procedures are repeated until the adaptive

Fig. (4.3) Block diagram of the adaptive parameter estimation

filter converges to give a minimum mean square error. Hence the parameters of the unknown system are duplicated in the adaptive filter.

The commercially available PTF-1 filter board* was used to generate various filter characteristics having finite impulse response (FIR). The experimental results with the FIR systems are presented in the following subsection.

4.3.2. ADAPTING TO FIR SYSTEMS

A linear phase bandpass filter characteristic having a 23 dB attenuation in the stopband and a centre frequency of 0.8 KHz ($0.4f_c$) was achieved with the PTF-1 filter board. The sampling frequency was 4.5 KHz. The output of the tracking oscillator from the spectrum analyser (HP 3580) was applied to the inputs of both the PTF-1 filter and the adaptive filter. The input signal was bandlimited to 2.25 KHz. and had an r.m.s. value of 0.707 volt. The nonlinearity function for the LMS algorithm was set at $\Delta x = 0.44$ volt, $\Delta \epsilon_1 = 70$ mV and $\Delta \epsilon_2 = 99$ mV. Adaptation results for the aforementioned bandpass filter characteristics are shown in Fig. (4.4). After the filter converges, the tap weights were frozen due to the dead zone $\Delta \epsilon_1$. In Fig. (4.4a), the adaptive filter response is compared with the response of the PTF-1. It can be seen that the adaptive filter response closely

* Supplied by Interactive Circuits and Systems Corp, Ottawa

- (a) Top trace: adaptive filter response
 Bottom trace: unknown system response
 Horizontal: 200 Hz/div.
 Vertical: 10 dB/div.
 Resolution bandwidth: 30 Hz
 Sampling frequency 4.5 KHz

- (b) Top trace: adaptive filter response
 Bottom trace: error spectrum
 Setting: same as (a)

Fig. (4.4) Adaptive filter frequency response in parameter mode

matches the frequency response of the unknown system over the entire Nyquist range. The residue error spectrum is shown in Fig. (4.4b). It can be seen that the residue error is at least 20 dB below the passband response. The residue error power in the stopband is relatively larger. However, the stopband response has very little effect on the overall filter performance.

The impulse responses of the unknown system and the adaptive filter after convergence, which are represented by their tap weights, are shown in Fig. (4.5). The desired impulse response is shown in Fig. (4.5a) and the adaptive filter response after convergence is shown in Fig. (4.5b). It can be seen that the two systems' tap weights correspond very closely. The slight difference in magnitude is due to the different gains incorporated with the two systems.

Another FIR system tested was a linear phase lowpass filter achieved by using the PTF-1 filter board. This filter (unknown system) had a 23 dB attenuation in the stopband and corner frequency of 0.8 KHz ($0.4 f_c$). The calibrations of the adaptive filter for the FIR bandpass filter were kept unchanged. Results of adaptation to the lowpass filter are shown in Fig. (4.6) and Fig. (4.7). In Fig. (4.6a), the frequency response of the adaptive filter after convergence compared with the frequency response of the unknown system. It can be seen that the adaptive filter

(a) Unknown system tap weights

(b) Copied tap weights in the adaptive filter

Fig. (4.5) Impulse response (tap weights) of the bandpass filter ($0.4 f_c$)

(a) Top: Adaptive filter response
Bottom: Desired response
Horizontal: 200Hz/div.
Vertical: 10 dB/div.
Sampling frequency 4.5 KHz

(b) Top: Adaptive filter response
Bottom: Error spectrum

Fig. (4.6) Adaptation to a FIR Lowpass filter

response closely matched the desired response, especially in the passband. It can be seen that the adaptive filter did not closely match the desired response in the stopband. The minimum attenuation at the stopband was roughly 15 dB. In Fig. (4.6b), the error power spectrum is compared with the converged filter response. The error spectrum is about 15 dB below the peak response. The tap weights of the unknown system and the adaptive filter after convergence are shown in Fig. (4.7). Again, the tap weights for the two systems are quite closely matched.

The measurement of the convergence time is shown in Fig. (4.8). The convergence time of the adaptive filter was found to be 920 msec, which corresponds to 4140 samples. The r.m.s. value of the error after convergence was slightly more than 20 mV.

The performance of the adaptive filter is highly dependent on the performance of the adaptive algorithm. The digital approach of implementing the LMS algorithm introduces some quantization noise to the adaptive filter. The quantization effects on the adaptive filter will be presented later in this chapter.

4.3.3 ADAPTING TO AN IIR SYSTEM

Generally, analog systems have infinite impulse response. Thus copying an IIR system response with an FIR PTF inherently implies truncation of the impulse response. An

(a) Unknown system's tap weights

(b) Adaptive filter's tap weights after convergence

Fig. (4.7) Tap weights of the unknown system and the adaptive filter

Fig. (4.8) Measurement of convergence time for the adaptive filter when adapting to a FIR bandpass filter

experiment was conducted to investigate the ability of the present adaptive filter in copying the system response of an IIR lowpass filter.

A Krohn-Hite 4th order Butterworth adjustable linear filter was used as the unknown system to provide a lowpass characteristic. The corner frequency was 200 Hz. A HP 3582A FFT spectrum analyser was used to measure the frequency responses of the experiment: the periodic noise source provided by the spectrum analyser was used as the input to both systems. The results are presented in Figures (4.9), (4.10) and (4.11). Figure (4.9) shows the two system responses. The linear filter had about 60 dB attenuation in the stopband while the adaptive filter had only 20.4 dB. Figure (4.10) shows the adaptive filter response and the residue error spectrum. The residue error in the passband is at least -9.3 dB. However, the residue error power is as high as the adaptive filter power in the stopband. Figure (4.11) shows the output signals of the two systems. The top trace is the IIR filter output and the lower trace is the adaptive filter output. It is seen that the waveform is copied identical but for some high frequency noise which is due to the low attenuation (-20.4 dB) at the stopband.

The experimental results with the IIR lowpass filter demonstrate the capability of copying an IIR system response. The degraded results are mainly because of the finite number

Top trace: Adaptive filter response
Bottom trace: IIR lowpass filter response

Fig. (4.9) Frequency response of the IIR lowpass filter and the adaptive filter

Top trace: Adaptive filter response
Bottom trace: Error power spectrum

Fig. (4.10) Frequency response of the adaptive filter and the residue error power

Top trace: IIR lowpass filter output
Bottom trace: Adaptive filter output
Horizontal scale: 2 msec/div
Vertical scale: 1V/div

Fig. (4.11) Display of adaptation to an IIR
lowpass filter in time domain

of taps of the PTF. In practical applications, the number of taps can be increased to achieve better performance. An alternative solution is to limit the energy of the unknown impulse response within the length of the PTF [14].

4.4 SYSTEM ERROR ANALYSIS

This section presents a quantitative analysis of the adaptive filter. The physical limitations of the present system are discussed. The effects of the approximations in the system implementation are also presented.

4.4.1 QUANTIZATION EFFECTS ON THE ADAPTIVE FILTER

The digital approach of implementing the LMS algorithm described in Section (3.4) reduces its hardware complexity. However, this approach is an approximation which deviates from the ideal. It was mentioned in Section (3.4) that the multiplication between $x(n-(k+1))$ and $e(n-1)$ are performed digitally. The quantization of these two signals affects the convergence time, adaptivity and residue error of the adaptive filter. The tap weight modification, Δh , due to the quantization effects can be written as:

$$\Delta h(n,k) = \begin{cases} 0, & \text{if } |e(n)| < \Delta e \text{ or } |x(n-k+1)| < \Delta x \\ 2^{-7}, & \text{if } \Delta e_1 \leq |e(n)| \leq e_2 \text{ and} \\ & |x(n-k+1)| \geq \Delta x \\ 2^{-6}, & \text{if } |e(n)| \geq \Delta e_2 \text{ and } |x(n-k+1)| \geq \Delta x \end{cases}$$

while the ideal value of the modification is $\mu e(n) \cdot x(n-k-1)$.

2 2
OF / DE

4

Thus the effective value of the convergence factor, $\mu(n,k)$, varies inversely with the product, $\epsilon(n) \cdot x(n-k-1)$ and is given by:

$$\mu(n,k) = \begin{cases} 0 & , \text{ if } |\epsilon(n)| < \Delta\epsilon \text{ or } |x(n-k-1)| < \Delta x \\ \frac{2^{-7}}{|(n) \cdot x(n-k-1)|} & , \text{ if } \Delta\epsilon_1 \leq |\epsilon(n)| \leq \Delta\epsilon_2 \text{ and } |x(n-k-1)| \geq \Delta x \\ \frac{2^{-6}}{|(n) \cdot x(n-k-1)|} & , \text{ if } |\epsilon(n)| \geq \Delta\epsilon_2 \text{ and } |x(n-k-1)| \geq \Delta x. \end{cases}$$

According to the above relations, the maximum convergence factor, $\mu_{\max}$ occurs only at the boundary $|\epsilon(n)| \geq \Delta\epsilon_2$ and $|x(n-k-1)| \geq \Delta x$. The convergence factor decreases as $\epsilon(n)$ and $x(n-k-1)$ depart from these boundaries. Thus, provided the analog signal, $x(n-k-1)$, has a magnitude greater than Δx , the adaptive process starts moving toward the minimum of mean square error, MSE. As the error signal decreases, the correction step size increases since the value of μ increases. This process is opposite to the ideal adaptive process where larger corrections are made initially due to large values of μ followed by smaller corrections near the steady state due to smaller values of μ . However, with the dead zone effect (when $|\epsilon(n)| < \Delta\epsilon_1$ or $|x(n-k-1)| < \Delta x$) which stops the adaptive process, the present implementation works satisfactorily. Ahuja demonstrated that the tap weights

could reach their optimal values by using a constant convergence factor, $\hat{\mu}$ rather than using a quantized μ [14]. Extensive computer simulations were carried out by Ahuja and Campanella et al [14,20] to evaluate the quantization effects on the LMS algorithm. They pointed out that the performance of the adaptive filter depends on the dead zone values, i.e. $\Delta x/\sigma_x$ and ϵ_2/ϵ_1 . In this thesis, $\Delta x/\sigma_x$ was set to be 0.6 as suggested by Campanella et al and $\Delta\epsilon_1/\sigma_x$ was set to be 1.4 as suggested by Ahuja. However, experiments were conducted to evaluate the performance of the present adaptive filter by using $\Delta\epsilon_1/\sigma_x$ as a variable. The effects of $\Delta\epsilon_1/\sigma_x$ on convergence time, residue error and adaptivity are discussed below:

The convergence time of the adaptive filter depends on the values of the error deadzone, $\Delta\epsilon_1$. Faster convergence time can be expected with larger $\Delta\epsilon_1$ because the adaptive process stops much earlier. However, the residue error will be much larger after convergence. An experiment was conducted to demonstrate this property. Figures (4.12) and (4.13) show the different convergence times of the adaptive filter caused by two different values of $\Delta\epsilon_1$. The ratio of $\Delta\epsilon_2/\Delta\epsilon_1$ was kept unchanged. With smaller $\Delta\epsilon_1$, the convergence time is longer with relatively smaller residue error as shown in the figures. The best combination of the convergence time (920 msec) and the residue error (22 mV) was obtained by

Fig. (4.12) Measurement of convergence time for the adaptive filter when adapting to a FIR bandpass filter

Fig. (4.13) Measurement of convergence time for the adaptive filter when adapting to a FIR bandpass filter

setting $\Delta\epsilon_1 = 70$, $\Delta\epsilon_2/\Delta\epsilon_1 = 1.4$, $x/\sigma_x = 0.6$ and $\sigma_x = 0.71 \text{ V}_{\text{rms}}$.

The error deadzone value also affects the residue error power. The relationship between the error after convergence and the value of the error deadzone is shown in Fig. (4.14). It is seen that the minimum residue error occurs when $\Delta\epsilon_1$ is about 70 mV. The residue error increases when $\Delta\epsilon_1$ is increased, as is to be expected. However, it is seen that when $\Delta\epsilon_1$ decreases below 70 mV, the residue error also increases. This can be explained by the fact that when $\Delta\epsilon_1$ is too small, then any noise greater than the deadzone value will bring the adaptive filter out of the converge state. Thus the adaptive process must readjust the tap weights to reconverge. This repeated process will increase the residue error considerably [14,20].

An equation suggested by Ahuja [14] which represents how well the adaptive system can estimate the parameters of the unknown system is given by:

$$\text{Adaptivity} = 10 \log_{10} \frac{\overline{\epsilon^2}/\sigma_x^2}{\sum_{k=0}^{N-1} (h_k^f)^2} \text{ dB} \quad (4.1)$$

where $\overline{\epsilon^2}$ is the variance of the residue error,

σ_x is the RMS value of the input and

h_k^f is the tap weights after convergence.

It has been pointed out that the performance of the

Fig. (4.14) RMS residue error as a function of $\Delta\epsilon_1$

adaptive filter depends on the ratios $\Delta\epsilon_1/\sigma_x$, $\Delta\epsilon_2/\Delta\epsilon_1$ and $\Delta x/\sigma_x$. In the present work, $\Delta\epsilon_2/\Delta\epsilon_1$ and $\Delta x/\sigma_x$ are kept unchanged. By varying the ratio $\Delta\epsilon_1/\sigma_x$, a plot of adaptivity was obtained and is shown in Fig. (4.15). It can be seen that the adaptivity has a relatively flat portion at approximately -17 dB for $\Delta\epsilon_1/\sigma_x$ lying in the range of 0.07 to 0.11.

Thus the present system gives best performance results, in terms of convergence time, residue error and adaptivity, for a range of $\Delta\epsilon_1/\sigma_x$ between 0.07 to 0.11. This result shows the validity of implementing the LMS algorithm by the digital approximation method.

4.4.2 POSSIBLE ERRORS IN THE PTF

The PTF implementation described in Section (3.3.1) employs a CTD tapped delay line to perform the delay action. The tapped output buffers have uneven gains and offset voltages which affect the performance of the adaptive filter. The error appearing at each tapped output buffer consists of gain error and offset voltage error. Thus the error introduced at each tapped output buffer is also a function of the input signal strength and the tap weight. Detailed investigation of these parameters is difficult. Therefore a simple model will be presented, which assumes that the error at each tapped output buffer can be represented by a noise signal added to the input signal. This simple model,

Fig. (4.15) Adaptivity as function of $\Delta \epsilon_1 / \sigma_x$

although not completely valid, demonstrates the importance of the error introduced at each tap position. A qualitative analysis of the simple model is presented here.

Let Δx_k represents the error introduced at the k^{th} tap of the delay line, so that the output of the transversal filter can be written as:

$$y(n) = \sum_{k=0}^{N-1} h_k(n)[x(n-k) + \Delta x_k] \quad (5.2)$$

If the input is grounded, the above equation becomes:

$$y_g(n) = \sum_{k=0}^{N-1} h_k(n) \cdot \Delta x_k \quad (5.3)$$

In transversal filter applications where h_k does not vary, the variation Δx_k will only introduce a d.c. offset at the output [19]. Thus the performance of the transversal filter is not affected. However, if the PTF is used in an adaptive system, this variation will affect the overall system performance in terms of the convergence time and the residue error. The error signal, $\epsilon(n)$, for the adaptive filter is given by:

$$\epsilon(n) = d(n) - y(n) \quad (5.4)$$

Substituting Eqn. (5.2) into Eqn. (5.4), then:

$$\epsilon(n) = d(n) - \sum_{k=0}^{N-1} h_k(n)[x(n-k) + \Delta x_k] \quad (5.5)$$

If both inputs of the adaptive filter and the unknown system are grounded, then the output of the unknown system, $d(n)$, and the delayed signal, $x(n-k)$, become zero. Equation (5.5) can be rewritten as:

$$\epsilon_g(n) = - \sum_{k=0}^{N-1} h_k(n) \cdot \Delta x_k \quad (5.6)$$

If Δx_k is non-zero and both Δx_k and $\epsilon_g(n)$ are larger than the designated deadzone values, then the LMS algorithm will start updating the tap weights until the error signal is smaller than the deadzone value. The tap weight modification can be expressed as:

$$h_k(n+1) = h_k(n) + \mu \epsilon_g(n) [x(n-k) + \Delta x_k] \quad (5.7)$$

Since $x(n-k) = 0$, by substituting Eqn. (5.6) into Eqn. (5.7)

$$h_k(n+1) = h_k(n) - \mu \sum_{k=0}^{N-1} h_k(n) \cdot \Delta x_k^2 \quad (5.8)$$

Hence the tap weights of the adaptive filter will be updated to correct the error in the tapped delay line, provided the error is large enough to trigger the adaptive process.

An experiment was carried out to verify the above argument. By grounding the input of the adaptive filter and its reference input, the tap weights, which were initially set to their maximum value, were updated to correct the

error introduced by the tapped delay line. Figure (4.16) shows the tap weights after convergence under the conditions mentioned. Note that Δx_k is a function of the gain, and offset voltage variances of the TDL buffers. The detailed modelling of these parameters is beyond the scope of this thesis. However, the experimental result shown in Fig. (4.16) demonstrates that the adaptive filter always has to correct the errors introduced by the TDL. Consequently, the performance of the adaptive filter is affected in terms of convergence time and residue error. To overcome this problem, a TDL with better gain and offset control to its buffers should be used.

4.5 COMPARISON AND CONCLUSION

The experimental results and the analysis of the adaptive filter have been presented. It has been shown that the PTF can achieve a satisfactory performance. When the PTF is used as a frequency selective filter, the stop-band attenuation is about 35 dB, only 5 dB less than that achieved by Ahuja [14] who used 32 MDAC's in his PTF realization. The 64 stages PTF, suggested by Nair [25] achieved 60 dB. However, in Nair's approach, A/D and D/A conversions are required for operating a digital delay line. Moreover, a two clock period delay at the PTF output was reported by Nair. This delay does not affect the PTF performance but is not desirable in adaptive systems. Due to

Vertical scale: 0.5 V/div
Horizontal scale: 30 μ sec/div (calibrated)

Fig. (4.16) Tap weights which correct the error introduced by the tapped delay line buffers.

the effect of time division multiplexing of an MDAC, the maximum sampling rate achieved by the present PTF is 14.5 KHz while most of the other PTF can sample at up to hundreds of KHz.

In Section (4.3), the performance of the proposed adaptive filter was discussed. The only adaptive filter which can be used for comparison was reported by Ahuja [14], as most of the other published adaptive systems were not hardware implemented. The present adaptive filter achieved -17 dB adaptivity compared to -20 dB achieved by Ahuja. However, the convergence time achieved by the present system is 920 msec (4140 samples) as opposed to 320 msec (2500 samples) achieved by Ahuja. Proakis [26] reported a computer simulation of an adaptive filter which required 4000 samples for convergence. This is approximately the same as the present experimental result. The residue error for the present system was found to be 20 mV_{rms} as opposed to less than 10 mV_{rms} reported by Ahuja. Ahuja's low residue error is believed to be the result of his use of a double correlation circuit for fixed pattern noise suppression.

In Section (4.4), the parameters which affect the present system were discussed. It should be pointed out that the digital approach of implementing the LMS algorithm works satisfactorily even though it is not ideal.

Considering the simplicity of the present system,

which needs only one MDAC for a 32-stages PTF, the experimental results are encouraging. Even though the sampling frequency of the PTF is low, its performance is comparable to Ahuja's. Nevertheless, the hardware simplicity for logic control can be traded for speed, as will be discussed in Chapter Five.

CHAPTER 5

CONCLUSION

5.1 SUMMARY

A novel, mixed digital and sampled analog adaptive filter with reduced hardware complexity has been described in this thesis. A mixed digital and sampled analog PTF, which used only one MDAC, was developed for use as the main processor of the adaptive filter. The LMS algorithm was hardware implemented to perform an adaptive loop function. As a result, substantial reduction in hardware complexity was achieved.

The theoretical background of both transversal filtering and adaptive filtering were studied to determine the validity of the proposed approach. Different approaches were reviewed to determine the advantages of the present method. A 32 tap PTF and the hardware implementing the LMS algorithm were built on separate wire-wrap boards for testing. The PTF was characterized as a frequency selective filter to verify its ability to perform the adaptive filtering function. The adaptive filter was characterized as a parameter estimator to obtain results for analysis and comparison with the one adaptive filter reported in the literature [14]. The effects of quantization on the approximate digital implementation of the LMS algorithm were discussed in detail and the hardware limitations were

presented.

The experiment results of the proposed adaptive filter were encouraging. The performance was, to a great extent, comparable to that of the previously reported adaptive filter [14]. However, the useful bandwidth and the convergence time were found to be inferior because of the multiplexing effect. It was pointed out in Chapter Four that simplicity of the digital control could be traded for both increased bandwidth and decreased convergence time. The most important achievement of this thesis is that it represents a step towards the realization of a monolithic integrated adaptive system.

From the result of this work, it is able to integrate the tap weight modification logic circuit together with a time-multiplexing single MDAC PTF. As evidence in the circuit shown in Appendix B, most of the hardware required for the LMS algorithm are either small scale or medium scale integrated circuits. The saving of the area from multiplexing one MDAC is ample to provide the space for the LMS algorithm implementation.

5.2 RECOMMENDATION FOR A MODIFICATION TO THE PRESENT SYSTEM

The most important limitation of the present system is the limited bandwidth. The maximum sampling frequency for the PTF was found to be 14.5 KHz whereas, the maximum sampling frequency for the adaptive filter was only 4.5 KHz.

Fig. (5.1) Timing requirement for the adaptive filter.

Most voiceband communication channels require 8 KHz sampling rate. Thus the present system is limited to sub-voiceband applications. The easiest way to increase the bandwidth is to redesign the logic control circuit for the tap weight modification.

The present logic control for tap weight modification was derived from the clock signal of the PTF. Figure (5.1) shows the timing as reproduced from Fig. (3.9). The advantage of this method is that only simple hardware is required for the logic control function. Note that there are some time slots not used during each tap operation; this is the result of the nonuniform tap weight updating scheme adopted in this project.

The proposed system alteration performs a burst modification as suggested by Ahuja [14] so that all tap weights are updated after filtering operation. In this way, the "empty" time slots can be eliminated and the bandwidth increased.

The timing requirement for a burst modification is shown in Fig. (5.2). Sampling frequency could be increased by over 90% as the digital modification would require less time to perform. Although the operation of the overall adaptive filter would be quite different from the system developed in this thesis, in that it would require the PTF to stop while the tap weights are updated. The proposed modification can be achieved simply by generating another

* All 32 tap operations are performed during "ON" time.

** All 32 tap weights are updated for next filtering operation during "ON" time.

Fig. (5.2) Simplified timing requirement for the adaptive filter with burst modifications

set of control logic for the tap weight modification. In view of the advantage, this modification would be highly worthwhile.

5.3 FUTURE WORK

The present work has shown a method for simplifying the architecture of the mixed digital and sampled analog adaptive filter. However, further research is recommended.

In Chapter Four, it was pointed out that the variances, such as the uneven gains and different offset voltages, of the TDL output buffers affect the performance of the overall adaptive system. Thus, it is recommended that an improved TDL should be developed. The objective of the new TDL is, obviously, that of minimizing the error introduced by the tap output buffers. It is also recommended that detailed modelling of the effects on the adaptive filter of the TDL errors be carried out.

The architecture of the proposed adaptive filter would be very easy to integrate using MOS technology. The successful integration of the PTF reported by Ahuja [14] - which had 32 MDACs is proof of the feasibility of integrating the present system. The switched capacitor integrator, logic circuits and the 32x8 bit memory required for the adaptive filter are already minimal by comparison, and could be readily integrated. Finally, by using a 12 bit MDAC, the accuracy should improve.

REFERENCES

1. Haque, Y.A., and Copeland, M.A., "Design and characterization of a Real-Time Correlator", IEEE Journal of Solid State Circuits, Vol. SC-12, pp. 642-650, Dec. 1977.
2. Widrow, B., et al, "Adaptive noise cancelling principles and applications", Proc. IEEE, Vol. 63, pp. 1692-1716, Dec. 1976.
3. Lucky, R.W., "Techniques for adaptive equalization of digital communication", Bell System Technical Journal, Vol. 45, pp. 255-286, Feb. 1966.
4. Lucky, R.W., "Automatic equalization for digital communication", Bell System Technical Journal, Vol. 44, pp. 547-588, April 1965.
5. Sondhi, M.M., "An adaptive echo canceller", Bell System Technical Journal, pp. 497-511, Mar. 1967.
6. Kuo, B.C., "Automatic control systems", Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1975.
7. Morgan, D.R., and Craig, S.E., "Real time adaptive linear prediction using the least square gradient algorithm", IEEE Transaction on acoustics, Speech, and signal processing, Vol. ASSP-24, No. 6, pp. 494-507, Dec. 1976.
8. Atal, B.S., and Schroedn, M.R., "Adaptive predictive coding of speech signal", Bell System Technical Journal, pp. 1973-1986, Oct. 1970.
9. Morgan, D.R., "A note on real-time linear prediction of speech waveforms", IEEE Transaction on Acoustics, Speech and Signal Processing, pp. 386-387, 1975.
10. Goodman, D.J., Scagliola, C., Crochiere, R.E., Rabiner, L.R. and Goodman, J., "Objective and subjective performance of tandem connections of waveform coders with an LPC vocoder", Bell System Technical Journal, Vol. 58, No. 3, pp. 601-629, March 1979.
11. Tribolet, J.M., Noll, P., McDermott, B.J., and Crochiere, R.E., "A comparison of the performance of four low-bit-rate speech waveform coders", Bell System Technical Journal, Vol. 58, No. 3, pp. 699-712, March 1979.

12. Fjallbrant, T., "Data reduction of sampled speech signals: method and results", Int. J. Electronics, Vol. 46, No. 4, pp.401-412, 1979.
13. Satorius, E.H., and Alexander, S.T., "Channel equalization using adaptive lattice algorithm", IEEE Transactions on Communications, Vol. COM-27, NO. 6, pp. 899-906, June 1979.
14. Ahuja, B.K., "A sampled analog MOS LSI adaptive/programmable transversal filter", Technical report No. CERL-79-02, Department of Electronics, Carleton University, Ottawa, 1979.
15. Haque, Y.A., "Design and characterization of a programmable transversal filter". a Ph.D. thesis submitted to Carleton University, Ottawa, May 1977.
16. Roberts, J.B.G., "The roles for CCD and SAW in signal processing", North Atlantic Treaty Organization, Advisory Group for Aerospace Research and Development, AGARD-CPP-230, Oct. 1977.
17. Tiemann, J.J., et al, "A surface charge correlator", IEEE Journal of Solid State Circuits, Vol. SC-9, pp.403-410, Dec. 1974.
18. Mavor, J. and Denyer, P.B., "The design and development of CCD programmable transversal filter and correlator", AGARD-CPP-230, Oct. 1977.
19. Roy, D., and Menard, P., "High performance digitally programmable analog transversal filter", Electronics letters, Vol. 14, No. 20, pp. 671-672, Sept. 1977.
20. Campanella, S.J., et al, "Analysis of an adaptive impulse response echo canceller", COMSAT Technical Review, Vol. 2, No. #, pp. 1-36, Spring 1972.
21. Widrow, B., et al, "Stationary and nonstationary learning characteristics of the LMS adaptive filter", Proceeding IEEE, Vol. 64, No. 8, Aug. 1976.
22. Forsythe, G.E., and Moler, C.B., "Computer Solution of linear algebraic systems", Prentice Hall, Anglewood Cliffs, N.J. pp. 120-131, 1967.
23. Papoulis, A., "Probability, random variables, and stochastic process", Mc-Graw Hill, N.Y., 1965.

24. Chan, Y.T., Hattin, R.V., and Plant, J.B., "The least square estimation of time delay and its use in signal detection", IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. ASSP-26, No. 3, pp. 217-222, June 1978.
25. Nair, N.G., "A mixed digital/sampled analog processing architecture", Technical report No. CERL-80-03, Department of Electronics, Carleton University, Ottawa, 1980.
26. Proakis, J.G., "Adaptive digital filters for equalization of telephone channels", IEEE Transactions on Audio and Electroacoustics, Vol. AU-18, No. 2, June 1970.

APPENDIX A

DESIGN DETAILS OF THE PTF

A.1 INTRODUCTION

The block diagram of the PTF is given in Fig. (A.1) which is reproduced from Fig. (3.5). The detailed circuit design, for the 32 tap mixed digital and sampled analog PTF is presented in this appendix. The adaptive algorithm is presented in Appendix B.

A.2 CHOICE OF THE TAP WEIGHT WORD LENGTH

The word length of the tap weight is chosen to be eight. The decision was made based on the availability of the other components such as EPROM and MDAC, etc. The word length can be increased to achieve better performance. However, eight bits word length is sufficient for experimental purpose.

A.3 CHOICE OF COMPONENTSA.3.1 TAPPED DELAY LINE

A recticon TAD-32 BBD tapped analog delay line is selected. The success in implementing PTFs by using the TAD-32 encourages this choice. It has a 60 dB dynamic range and maximum 5 MHz sampling frequency. The number of taps is 32 which can provide a reasonable impulse response

characteristics. The power supply require is 15 volts with respect to ground which is compatible with most CMOS ICs.

A.3.2 MULTIPLYING DIGITAL TO ANALOG CONVERTER

The CMOS multiplying digital to analog converter (AD 7523) is chosen to perform multiplication between an analog voltage and an 8-bit digital word. The AD 7523 has a maximum settling time of 200 nanoseconds, and so an inexpensive operational amplifier can be used with it to provide a total settling time of better than one microsecond. The reference voltage input of the AD 7523 can accept voltage varying from -10 volts to +10 volts. Thus different reference voltages can be used without biasing.

A.3.3 OPERATIONAL AMPLIFIER

The general purpose FET input operational amplifier, LF 356, is chosen for general use. The speed of the LF 356 is, however, inadequate for the amplifier with the MDAC. Thus, a high slew rate operational amplifier, LM 318, is chosen for the MDAC. All the operational amplifier are biased at ± 15 V.

A.3.4 ANALOG TRANSMISSION SWITCH

A CMOS analog switch, CD 4066, is found to be the most suitable with respect to the speed and cost. Its switching time is better than 100 nanoseconds, at a supply voltage of 10 V. Its switch-off time is always less than

the switch-on time, which helps in the implementation of break-before-make SPDT switches. One difficulty in using this chip when compared to other analog switches is that the analog signals have to be biased by a positive voltage before switching.

A.3.5 DIGITAL CONTROL CIRCUITS

Standard CMOS integrated circuits are chosen for low power consumption and compatability with other ICs such as the tapped delay line and MDAC. The digital control circuits are operated with a V_{DD} of +15 volts and V_{SS} of 0 volt. A monostable/astable multivibrator, CD 4047, is used to generate the master clock. A synchronous counter, CD 4520, is used as the frequency dividers and counters. A decade counter/divider with 10 decoded outputs, CD 4017, is used for part of the control signal generation.

A.4 CIRCUIT DESIGN

The circuit built is capable of performing the PTF function. The circuit is designed such that the LMS adaptive algorithm can be used. Timing control is designed so that the LMS algorithm can update each tap weight before multiplication. Figure (A.1) shows the block diagram of the PTF and Fig. A2 shows the corresponding timing diagram. The detailed descriptions for the block is presented in the following subsections.

Fig. (A.1) Block diagram of the PTF

A.4.1 DIGITAL CONTROL LOGIC

The digital control logic generates a master clock and all the necessary clocks for the PTF. All the circuits are built by using standard CMOS ICs. The master clock is generated by a CD 4047 Monostable/Astable Multivibrator as an astable Multivibrator. The master clock frequency can be selected by changing the resistance value. The master clock is fed into a CD 4017 Decade Counter/Divider with 10 Decoded Outputs for control signals generation. The master clock is also fed into a pair of CD 4520 Dual Synchronous Up Counter for address (5-bit) generation. The 5-bit address generated is latched out for synchronization by a CD 4076 Quad D-Flip-Flop and a CD 4027 J-K Flip-Flop. Thus the address for the multiplexer (CD 4067) and the tap weight address is generated. The 5-bit address is reset by the counter or an external reset. This is necessary for the adaptive filter evaluation. An OR gate is used for this purpose. Figure (A.2) shows the required timing signal. Figure (A.3) shows the schematic diagram as built on the wire-wrap board. The clock driver for the TAD-32 TDL is shown as in Fig. (A.4).

A.4.2 ANALOG CIRCUIT DESIGN

The analog circuit consists of a TDL, an analog extraction circuit (which will be described in Appendix B).

* ϕ_{MOD} is the designated time slot for tap weight modifications.

Fig. (A.2) Timing of the PTF

Fig. (A.3) Schematic of the logic control generation

Note: two such circuits provide the complementary clock required by the TDL.

Fig. (A.4) Clock driver for TAD-32 TDL

a multiplication block, a switched capacitor integrator. The sample and hold circuit and biasing circuits are inserted where appropriate. The input, analog signal is sampled and held at the sampling frequency before fed into the TDL. The outputs of the TDL is multiplexed by 2 CD 4067 analog multiplexers. The output of the analog multiplexer is buffered for the adaptive filter purpose. This is necessary as the output of the analog multiplexer is a high impedance point. The buffered signal is sampled and held for synchronization purpose. Then the bias added to the signal is subtracted by using a LF 356 operational amplifier. The signal is then ready for multiplication. The MDAC is arranged for a four quadrant multiplication. The output of the MDAC is then summed by a switched capacitor integrator. A dc bias voltage is added to the signal because the analog switches (CD 4066) can only operate between 0 to 15 volts. The operations of the switched capacitor integrator and the MDAC were described in Chapter Three. After all 32 samples are summed, the summed signal is sampled for output. The dc bias is then removed by using an LF 356 operational amplifier as a differential amplifier. All switches used are CD 4066 and the dc bias addition/subtraction are precisely controlled by using trim-pots. Fig. (A.5) shows the circuit diagram of the analog path of the PTF.

Fig. (A.5) Analog Path of the PTF

Fig. (A.6) Four quadrant multiplier configuration

APPENDIX B

CIRCUIT DESIGN OF THE LMS ADAPTIVE ALGORITHM

B.1 INTRODUCTION

This appendix presents the detailed circuit design of the LMS algorithm. The logic control circuit is derived from the timing developed for the PTF. The major design includes the digital multiplication between the delayed signal and the error signal. The tap weight updating block is also presented in detail.

B.2 CHOICE OF COMPONENTS

B.2.1 MEMORY

An Intel 2111A 256X4 N-channel MOS RAM is selected to be used as the tap weight memory bank. Two such chips together are used to form the eight-bit memory for the tap weights. This chip operates on a single +5 volt power supply. The access time is 350 nanoseconds. This chip is used because of its fast access ability. However, this chip could be replaced by a smaller size one.

B.2.2 OPERATIONAL AMPLIFIERS AND VOLTAGE COMPARATORS

The general purpose FET input operational amplifier, LF 356, is used for the sample and hold, differential amplifier (for error generation).

The LM 311 voltage comparator is chosen to perform quantization. The LM 311 has a fast switching time and a low cost which make it a prime candidate. The comparator can compare either positive or negative voltage depending on the connection.

B.2.3 ANALOG TRANSMISSION SWITCHES

The CMOS analog switch, CD 4066, is used for input sample and hold so that proper synchronization can be achieved.

B.2.4 DIGITAL CONTROL LOGIC AND INTERFACE

The digital control logic is built using standard TTL integrated circuits. Since the tap weight updating requires a fast operation, the TTL is suitable. Moreover, most of the other integrated circuits are operated on +5 volt power supply, thus using TTL integrated circuits is reasonable. For testing purpose, the TTL integrated circuit is sufficient. However, when the adaptive system is monolithically integrated, CMOS logic control should be used.

The interface circuit is necessary because the PTF is mostly built using CMOS integrated circuits. Thus the CD 4049 and CD 4050 CMOS integrated circuits are used to convert the CMOS level to TTL level. Also 7406 and 7407 TTL integrated circuits are used to convert the TTL level to CMOS level.

B.3 CIRCUIT DESIGN

The circuit built is capable of performing the LMS algorithm. The error is generated for tap weight updating. The detailed block diagram of the circuit is shown as in Fig. (B.1). The blocks are described in detail in the following subsections.

B.3.1 DIGITAL LOGIC CONTROL

The digital control of the LMS algorithm is derived from the existing timing developed for the PTF. The timing diagram for the adaptive filter is shown in Fig. (B.2) which is reproduced from Fig. (3.9). The clocks necessary for the adaptive algorithm are mainly for synchronization. These clocks are derived using standard TTL logic gates. All the digital data are latched by using edge-trigger latches. The schematic of the digital control logic is shown in Fig. (B.3). The digital signals from the PTF are converted to TTL using CD 4049 or 4050 integrated circuits. The required signals are:

- (i) ϕ_x : latch the digitized delayed input sample.
- (ii) ϕ_{MOD} : latch the product of delayed signal and error signal to the adder/subtractor.
- (iii) ϕ_h : latch the updated tap weight to the RAM and the MDAC.
- (iv) LR: skip the tap weight modification.
- (v) R/W, OD: control signal for the RAM.

From analog multiplier
in the PTF

Fig. (B.1) Block diagram of the LMS algorithm circuit

Fig. (B.2) Timing requirement for the adaptive filter

Fig. (B.3) Control logic circuit diagram

B.3.3 ADDER/SUBTRACTOR WITH OVERFLOW-CLAMP

The tap weights are updated by means of adding the quantized product (of delayed input signal and error signal), overflow may occur as the number of bits is only 8. To ensure the modified tap weight does not overflow, an overclamp feature is equipped with the adder subtractor.. The signs of the delayed input signal and the error can define the sign of the updated tap weight. Thus overflow can be detected. If overflow occurs, the updated tap weight will be damp to all "ONES" or all "ZEROS". This is done by using two 74H87 true/complement integrated circuits.

The binary operation of this circuit is either addition or subtraction. The sign bits of the delayed input signal and the error signal are sufficient to define which operation is appropriate. If addition is required, both the delayed input signal and the error signal are left as is for addition. If subtraction is required, then the error signal is complemented so that the addition will appear as subtraction.

B.3.4 ERROR GENERATION

The error generation circuit is capable of performing a subtraction between two analog signals. The error generated is used for tap weight modification. This is realized by using a LF operational amplifier as a differential amplifier. The reference signal (desired signal) is sampled and held

Fig. (B.4) Quantization and digital multiplication circuit

Fig. (B.5) Schematic of the adder/subtractor with overflow clamp

for proper synchronization.

END

30H08H821

FIN