NEAR EAST UNIVERSITY

Faculty of Engineering

Department of Electrical and Electronic Engineering

APPLICATION IN ADAPTIVE FILTERING

Graduation Project
EE-400

Student: Wassim Ahmed (971031)

Supervisor: Mr.Jamal Abo Hasna

Lefkosa - 2001

TABLE OF CONTENTS

ACKNOLEDGMENT

First of all I would like thank Mr.Jamal Abu Hasna being my advisor, because of this he gave me all what i need in my project. I successfully overcome many difficulties and learn a lot adaptive filtering.In each discussion he was my guide, was an open book. And I felt my quick progress from his advises. really he was so kind. Thankfulness to all Electrical and Electronic Engineering department stuff. So so the Palestinian Embassy by giving me financial supporting.

Special thank to Omar and Houssam (my home mates), they helped me in my project, in any problem I faced related to computer they were helping me. Thank to Faculty of Engineering for having such a good computational environment.

I also want to thank my friends in NEU:Hisham,Emad, Marwan, Omran, Fadi, Haydar Ramiz, Obaid,Wissam,Riyad Alaa, Alaa Kabha,Waled, Asim, Yousef, Awny, Khaled, Hamada Ihsan,Belma,Gozda,Gonul,Ozlem,Abed Pella, Didam. Mateyya, Jihad, Aref, Ahmad Katanany, AhmadTaha,Ayman, Jabra , Zia, Taysar, abu Taha, Ziad, Ebru Fadi Elladaa Ammar, Atallah, Husien, Osama, adnan,Burak Ramadan and to all if I had forgotten some one.

Finally, special thank to my family, specially my parents. Without their endless support and love for me,I would never achieved my current position. Also special thank to my big brother Abdallah who was supporting me during my whole studying period. I wish my mother lives happily always, and my father in the heaven be proud of me.

ABSTRACT

While the important of analog filters is being continuously reduced by their digital counterparts, they remain an important study, if for no other reason than they provide a gateway to the study of digital filter.

The design of a wiener filter requires a priori information about the statistics of the data to be processed. The filter requires a priori information about the statistics of data to be processed. The filter is optimum only when the statistical characteristics of the input data match the priori information on which the design of the filter is based. When this information is not known completely, however it may not possible to design the Wiener filter or else the design may no longer be optimum. An adaptive filter having self organizing structure based on recursive algorithm make it possible to perform satisfactory filtering in an environment where complete knowledge of the relevant signal characteristics are not available.

Comparison between LMS algorithm and other algorithm show that LMS algorithm is simple for realization and computation, and it does not require off-line gradient estimations of the data. But instead knowledge of signal statistics, it uses instantaneous estimation. The performance limits the adaptive echo cancellation techniques are investigated.

Network echo cancellers are located at a convenient point in the four-wire part of the telephone circuit. Their purpose is to cancel the echoes of speech on the go path, which appear on the return. A circuit requires two echo cancellers, which will probably not be co-located, as the nearer the ends of the four-wire circuit they are, the shorter the delay of the echoes to be canceled.

INTRODUCTION

The term filter is often used to describe a device in the form of piece of physical hardware or computer software that applied to aset of noise data in order to extract information about prescribed quantity of interest. Filtering is the extraction of information about quantity of interest at time t by using data measure up to including time t. A more efficient method is to use an adaptive filter. By such a device we mean one that is self-designing in that the adaptive filter relies for its operation on a recursive algorithm, which makes it possible for filter.

The first chapter is talking about the essential and principal property of the adaptive system is its time-varying, self-adjusting performance. Realizing that if a designer develops a system of fixed design, the implications are that the designer has foreseen all possible input conditions, at least statistically, may readily see the need for such performance. Finally, the designer has chosen the system that appears best according to the performance criterion selected.

Chapter two is dealed with the use of the telephony network for data and speech communication. Telecommunication is growing and changing industry which has proved to be fertile ground for the application of adaptive filters

Chapter three: in this chapter a class of adaptive filter algorithm is examined that transforms the input signal into the frequency domain before adaptive filtering. The transformations considered here are fixed nature. Also there are principal advantages to frequency-domain implementations of adaptive filters.

Chapter four analyzed the effects of signal characteristics such as auto and cross correlation on the achievable echo suppression. Techniques to enhance signal characteristics such as to improve both the learning ability and the steady state echo suppression quality are identified.

CHAPTER 1

ADAPTIVE FILTERS

1.1 Overview

The design of a Wiener filter requires a priori information about the statistics of the data to be processed. The filter is optimum only when the statistical characteristics of the input data match the a priori information on which the design of the filter is based. When this information is not known completely, however, it may not be possible to design the Wiener filter or else the design may no longer be optimum.

A straight forward approach that we may use in such situations is the "estimate and plug" procedure. This is a two-stage process whereby the filter first "estimates" the statistical parameters of the relevant signals and then "plugs" the results so obtained into a non-recursive formula for computing the filter parameters. For real-time operation, this procedure has the disadvantage of requiring excessively elaborate and costly hardware. A more efficient method is to use an adaptive filter. By such a device we mean one that is self-designing in that the adaptive filter relies for its operation on a recursive algorithm, which makes it possible for the filter to perform satisfactorily in an environment where complete knowledge of the relevant signal characteristics is not available. The algorithm starts from some predetermined set of initial conditions, representing whatever we know about the environment. Yet, in a stationary environment, we find that after successive iterations of the algorithm it converges to the optimum Wiener solution in some statistical sense.

In a non-stationary environment, the algorithm offers a tracking capability, in that it can track time variations in the statistics of the input data, provided that the variations are sufficiently slow.

As a direct consequence of the application of a recursive algorithm whereby the parameters of an adaptive filter are updated from one iteration to the next, the parameters become data dependent. This, therefore, means that an adaptive filter is in reality a non-linear device, in the sense that it does not obey the principle of superposition. Notwithstanding the property, adaptive filters are commonly classified as linear or non-linear. An adaptive filter is said to be Linear if the estimate of a quantity of interest is computed adaptively (at the output of the filter) as a Linear combination of the available set of observations applied to the filter input. Otherwise, the adaptive filter is said to be non-linear.

A wide variety of recursive algorithms have been developed in the literature for the operation of linear adaptive filters. In the final analysis, the choice of one algorithm over another is determined by one or more of the following factors:

- Rate of convergence. This is defined as the number of iterations required for the algorithm, in response to stationary inputs, to converge "close enough" to the optimum Wiener solution in the mean-square sense. A fast rate of convergence allows the algorithm to adapt rapidly to a stationary environment of unknown statistics.
- Misadjustment. For an algorithm of interest, this parameter provides a quantitative measure of the amount by which the final value of the mean-squared error, aver- aged over an ensemble of adaptive filters, deviates from the minimum mean- squared error that is produced by the Wiener filter.
- Tracking. When an adaptive filtering algorithm operates in a non-stationary environment, the algorithm is required to track statistical variations in the environment. The tracking performance of the algorithm, however, is influenced by two contradictory features: (1) rate of convergence, and (b) steady-state fluctuation due to algorithm noise.
- Robustness. For an adaptive filter to be *robust,* small disturbances (i.e., disturbances with small energy) can only result in small estimation errors. The disturbances may arise from a variety of factors, internal or external to the filter.
- Computational requirements. Here the issues of concern include (a) the number of operations (i.e., multiplications, divisions, and additions/subtractions) required to make one complete iteration of the algorithm, (b) the size of memory locations required to store the data and the program, and (c) the investment required to program the algorithm on a computer.
- Structure. This refers to the structure of information flow in the algorithm, determining the manner in which it is implemented in hardware form. For example, an algorithm whose structure exhibits high modularity, parallelism, or concurrency is well suited for implementation using very large-scale integration (VLSI).

• Numerical properties. When an algorithm is implemented numerically, inaccuracies are produced due to quantization errors. The quantization errors are due to analog-to-digital conversion of the input data and digital representation of internal calculations. Ordinarily, it is the latter source of quantization errors that poses a serious design problem. In particular, there are two basic issues of concern: numerical stability and numerical accuracy. Numerical stability is an inherent characteristic of an adaptive filtering algorithm. Numerical accuracy, on the other hand, is determined by the number of bits (i.e., binary digits) used in the numerical representation of data samples and filter coefficients. An adaptive filtering algorithm is said to be numerically robust when it is insensitive to variations in the word-length used in its digital implementation. These factors, in their own ways, also enter into the design of non-linear adaptive filters, except for the fact that we now no longer have a well-defined frame of reference in the form of a Wiener filter. Rather, we speak of a non-linear filtering algorithm that may converge to a local minimum or, hopefully, a global minimum on the error-performance surface.

In recent years, agrowing field of research in "adaptive systems" has resulted in a variety of adaptive automatos whose characterestics in limited ways resemble certain characterestics of living systems and biological adaptive processes.

Some meanings of "adaptation" are:

1. the act of adapting. 2. the state of being adapted; adjustment. 3. Biol. a. any alternation in the structure or function of an organism or any of its parts that results from natural selection and by which the organism becomes better fitted to survive and multiply in its environment. b. a form or structure modified to fit changed environment. 4. Physiol. the decrease in response of sensory receptor organs, as those of vision, touch, tempreture, olfaction, audition, and pain, to changed, constantly applied, environmental conditions. 5. Social. A slow, usually unconscious modification of individual and social acrivity in adjustment to cultural surroundings.It will be noted that the definition above is expressed primarily in terms of biological adaptation to environment. The same definitions serve at least to some extent for "artificial" or human-made adaptive systems, which are the central concern of this chapter.

An adaptive automation is asystem whose structure is alterable or adjustable in such a way that its behavior or performance (according to some desired criterion) improves throug contact with its environment. A simple example of an automaton or automatic adaptive system is the automatic gain control (AGC) used in radio and television receivers. The function of this circuit is to adjust the sensitivity of the receiver inversely as the average incoming signal strength. The receiver is thus able to adapt a wide range of input levels and to produce a much narrower range of output intensities.

The purpose of this book is to present certain basic principles of adaptation; to explain the design, operating characteristics, and applications of the simpler forms of adaptive systems; and to describe means for their physical realization. The types of systems discussed include those designed primarily for the purposes of adaptive control and adaptive signal processing. Such systems usually have some or all of the following characteristics:

- They can automatically adapt (self-optimize) in the face of changing (nonstationary) environments and changing system requirements.
- They can be trained to perform specific filtering and decision-making tasks. Synthesis of systems having these capabilities can be accoplished automatically through training. In a sense, adaptive systems can be "programmed" by atrain process.
- Because of the above, adaptive systems do not require the elaborate synthesis procedures usually needed for nonadaptive systems. Instead, they tend to be "self-designing."
- They can be extrapolate a model of behavior to deal with new situations after having been trained on a finite and often small number of training signals or patterns.
- To alimited extent, they can repair themselves; that is, they can adapt around certain kinds of internal defects.
- They can usually be described as nonlinear systems with time-varying parameters.
- Usually, they are more complex and difficult to analyze than nonadaptive systems, but they offer the possibility of substantially increased system performance when input signal characteristics are unknown or time varying.

1.2 General Properties

The essential and principal property of the adaptive system is its time-varying, self-adjusting performance. The need for such performance may readily be seen by realizing that if a designer develops a system of fixed design which he or she considers optimal, the implications are that the designer has foreseen all possible input conditions, at least statistically, and knows what he or she would like the system to do under each of these conditions. The designer has then chosen a specific criterion whereby performance is to be judged, such as the amount of error between the output of the actual system and that of some selected model or "ideal"system.

Finally, the designer has chosen the system that appears best according to the performance criterion selected, generally choosing this system from an a priorirestricted class of designs (such as linear systems).

In many instances, however, the complete range of input conditions may not be known exactly, or even statistically; or the conditions may change from time to time. In such circumstances,an adaptive system that continually seeks the optimum within an allowed class of possibilities, using an ordinarly search process, would give superior performance compared with a system of fixed design.

By their very nature, adaptive systems must be time varying and nonlinear. Their characteristics depend, among other things, on their input signals. If an input signals x_1 is applied, an adaptive system will adapt to it and produce an output y_1. If another input signal, x_2, is applied, the system will adapt to this second signal and will again produce an output y_2.

Generally, the form or the structure or the adjustments of the adaptive system will be different for the two different inputs. If the sum of the two inputs is applied to the adaptive system, the latter will adapt to this new input-but it will produce an output that will generally not be the same as y_1+y_2, the sum of the outputs that would have corresponded to inputs x_1 and x_2. In such a case, as illustrated in Figure 1.1, the principle of superposition does not work as it does with linear systems. If a signal is applied to the input of an adaptive system to test its response characteristics, the systems adapts to this specific input and thereby changes its own form. Thus the adaptive system is inherently difficult to characterize in conventional terms.

Whithin the realm of nonlinear systems, adaptive systems cannot be distinguished as belonging to an absolutely clear subset. However, they have two features that generally distinguish them from other forms of nonlinear systems.

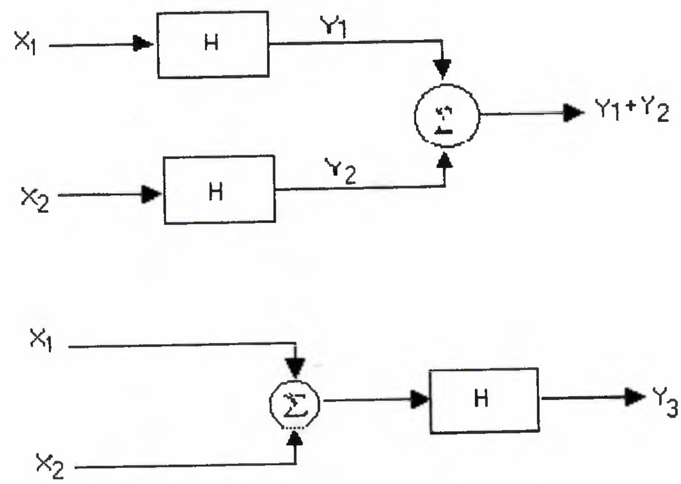


Figure 1.1 The Lower Output Y3 if H is A Linear System, if H is Adaptive Y3 is Generally Different From Y1+Y2

First, adaptive systems are adjustable, and their adjustments usually depend on finite-term average signal characteristics rather than on instantaneous values of signals or instantaneous values of the internal system state. Second, the adjustments of the adaptive systems are changed purposefully in order to optimize specified performance measures.

Certain forms of adaptive systems become linear systems when their adjustments are held constant after adaptation. These may be called "linear adaptive systems." They are very useful; they tend to be mathematically tractable; and they are generally easier to design than other forms of adaptive systems.

1.3 Open-And Closed-Loop Adaptation

Several ways to classify adaptive schemes have been proposed in the literature. It is most convenient here to begin by thinking in terms of open-loop and closed-loop adaptation. The open-loop adaptive process involves making measurements of input or environmental characteristics, applying this information to a formula or to a computational algorithm, and using the results to set the adjustments of the adaptive system. Closed-loop adaptation, on the other hand, involves automatic experimentation with these adjustments and knowledge of their outcome in order to optimize a measured system performance. The latter process called adaptation by "performance feedback."

The principles of open- and closed-loop adaptation are illustrated in figures 1.2 and 1.3. The "other data" in these figures may be data about the environment of the adaptive system, or in the closed-loop case, it may be a desired version of the output signal.

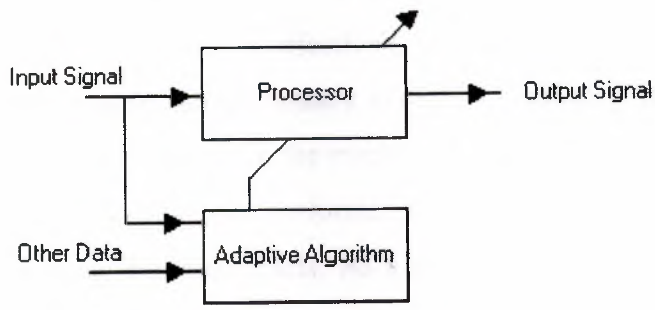


Figure 1.2 Open Loop Adaptation

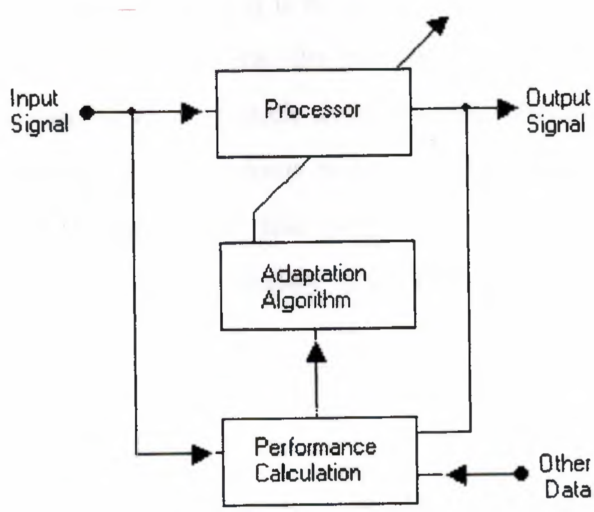


Figure 1.3 Closed Loop Adaptation

When designing an adaptive process, many factors determine the chpice of clsed-loop versus open-loop adaptation. The availability of input signals and performance-indicating signals is a major consideration. Also, the amount of computing capacity and the type of computer required to implement the open-loop and closed-loop adaptation algorithms will generally differ. Certain algorithms require the use of a general-purpose digital computer, whereas other algorithms could be implemented far more economically with special-purpose chips or other apparatus.

It is difficult to develop general principles to guide all choices, but several advantages and a few disadvatntages of closed-loop adaptation, which is the main subject can be pointed out here.

Closed-loop adaptation has the advantages of being workable in many applications where no analytic synthesis procedure either exists or is known, for example, where error criteria other than mean-square are used, where systems are nonlinear or time variable, where signals are nonsattionary, and so on.

Closed-loop can also be used effectively in situations where physical system component values are variable or inaccurately known. Closed-loop adaptation will find the best choice of component values. In the event of partial system failure, an adaptation mechanism that continually monitors performance will optimize this performance by adjusting and reoptimizing the intact parts. As a result, system reliability can often be improved by the use of performance feedback.

The closed-loop adaptation process is not always free of difficulties, however. In certain situations, performance functions do not have unique optima. Automatic optimization is an uncertain process in such situations. In othersituations, the closed-loop adaptation process, like a closed-loop control system, could be unstable. The adaptation process could diverge rather than converge. In spiteof these possibilities, performance feedback is a powerful, widely applicable technique for implementing adaptation.

1.4 Applications

The ability of an adaptive filter to operate satisfactorily in an unknown environment and track time variations of input statistics make the adaptive filter a powerful device for signal-processing and control applications. Indeed, adaptive filters have been successfully applied in such diverse fields as communications, radar, sonar, seismology, and biomedical engineering. Although these applications are indeed quite different in nature, nevertheless, they have one basic common feature: an input vector and a desired response are used to compute an estimation error, which is in turn used to control the values of a set of adjustable filter coefficients. The adjustable coefficients may take the form of tap weights, reflection coefficients, rotation parameters, or synaptic weights, depending on the filter structure employed. However, the essential difference between the various applications of adaptive filtering arises in the manner in which the desired response is extracted. In this context, we may distinguish four basic classes of adaptive filtering applications, as depicted in Fig.1.4. For convenience of presentation, the following notations are used in this figure:

u = input applied to the adaptive filter

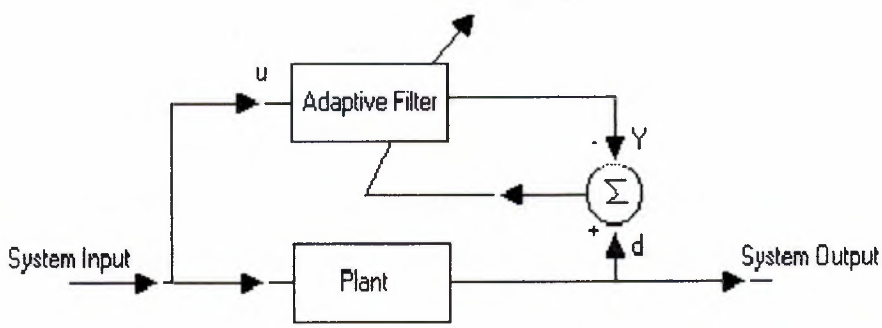
Y = output of the adaptive filter

d = desired response

e = d -y = estimation error .

The functions of the four basic classes of adaptive filtering applications depicted herein are as follows:

I. Identification Fig. 1.4(a). The notion of a mathematical model is fundamental to sciences and engineering. In the class of applications dealing with identification, an adaptive filter is used to provide a linear model that represents the best fit (in some sense) to an unknown plant. The plant and the adaptive filter are driven by the same input. The plant output supplies the desired response of the adaptive filter. If the plant is dynamic in nature, the model will be time varying.



(a)

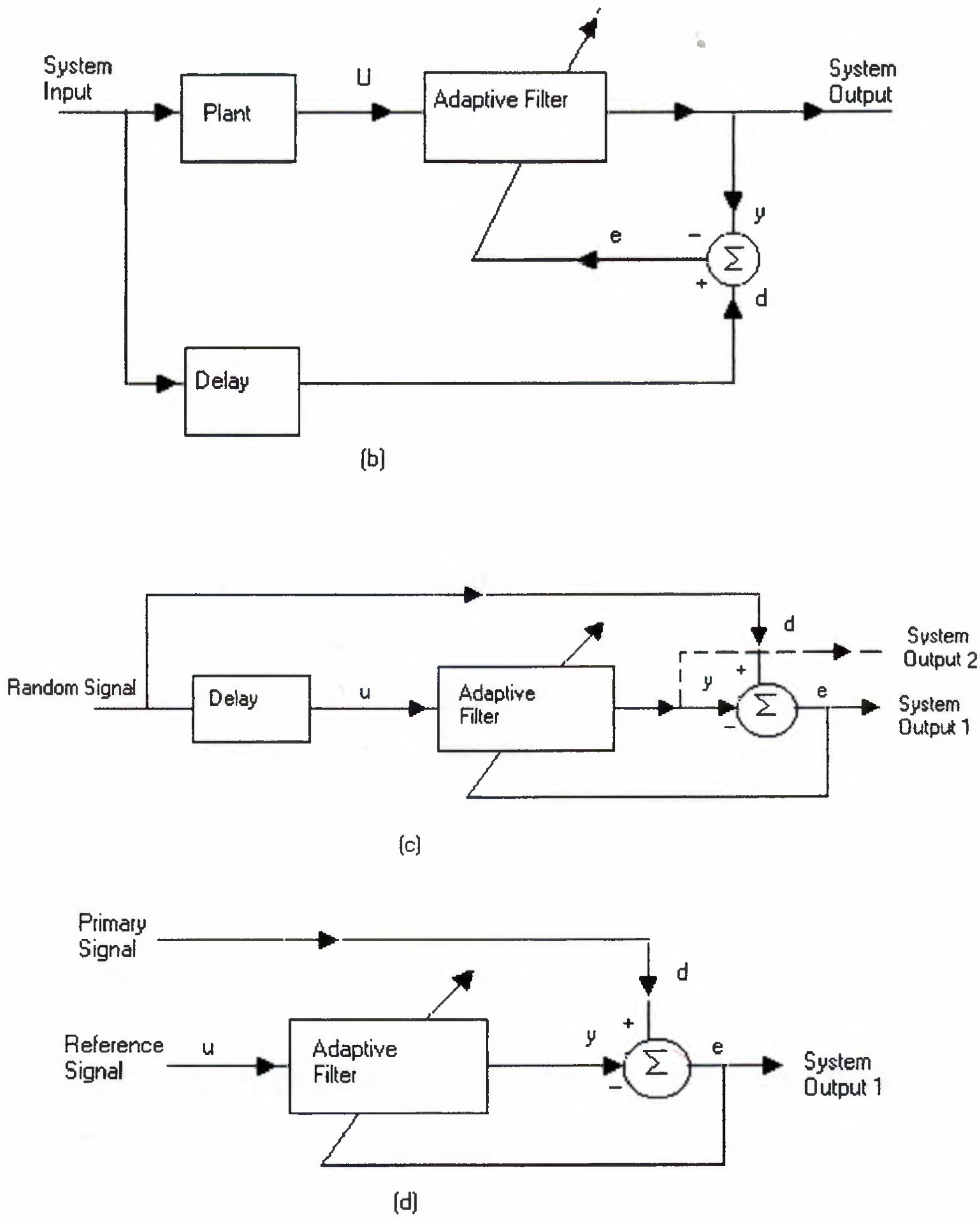


Fig. 1.4 Four Basic Classes of Adaptive Filtering Applications
(a) Identification (b) Inverse Modelling (c) Prediction (d) Interference Cancelling

II. Inverse modeling Fig. 1.4(b). In this second class of applications, the function of the adaptive filter is to provide an inverse model that represents the best fit (in some sense) to an unknown noisy plant. Ideally, in the case of a linear sys- tem, the inverse model has a transfer function equal to the reciprocal (inverse) of the plant' s transfer function, such that the combination of the two constitutes an ideal transmission medium. A delayed version of the plant (system) input constitutes the desired response for the adaptive filter. In some applications, the plant input is used without delay as the desired response.

III. Prediction Fig.1.4(c). Here the function of the adaptive filter is to provide the best prediction (in some sense) of the present value of a random signal. The present value of the signal thus serves the purpose of a desired response for the adaptive filter. Past values of the signal supply the input applied to the adaptive filter. Depending on the application of interest, the adaptive filter output or the estimation (prediction) error may serve as the system output. In the first case, the system operates as a predictor; in the latter case, it operates as a prediction- error filter.

IV. Interference cancelling Fig.1.4 (d). In this final class of applications, the adaptive filter is used to cancel unknown interference contained (alongside an information-bearing signal component) in a primary signal, with the cancellation being optimized in some sense. The primary signal serves as the desired response for the adaptive filter. A reference (auxiliary) signal is employed as the input to the adaptive filter. The reference signal is derived from a sensor or set of sensors located in relation to the sensor(s) supplying the primary signal in such a way that the information-bearing signal component is weak or essentially undetectable.

1.5 When To Use Adaptive Filters And Where They Have Been Used

The contamination of a signal of interest by other unwanted, often larger, signals or noise is a problem often encountered in many applications. Where the signal and noise occupy fixed and separate frequency bands, conventional linear filters with fixed coefficients are normally used to extract the signal. However, there are many instances when it is necessary for the filter characteristics to be variable, adapted to changing signal characteristics, or to be altered intelligently. In such cases, the coefficients of the filter must vary and cannot be specified in advance. Such is the case where there is a spectral overlap between the signal and noise, see Figure 1.5. or if the band occupied by the noise is unknown or varies with time.

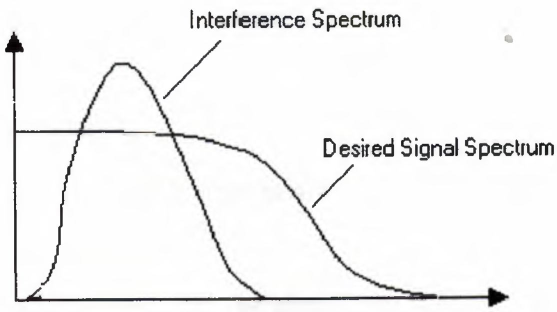


Figure 1.5. An Illustration of Spectral Overlap Between a Signal and a Strong Interference

Typical applications where fixed coefficient filters are inappropriate are the following.

(1) Electroencephalography (EEG), where artefacts or signal contamination produced by eye movements or blinks is much larger than the genuine electrical activity of the brain and shares the same frequency band with signals of clinical interest. It is not possible to use conventional linear filters to remove the artefacts while preserving the signals of clinical interest.

(2) Digital communication using a spread spectrum, where a large jamming signal, possibly intended to disrupt communication, could interfere with the desired signal. The interference often occupies a narrow but unknown band within the wideband spectrum, and can only be effectively dealt with adaptively.

(3) In digital data communication over the telephone channel at a high rate. Signal distortions caused by the poor amplitude and phase response characteristics of the channel lead to pulses representing different digital codes to interfere with each other (intersymbol interference), making it difficult to detect the codes reliably at the receiving end. To compensate for the channel distortions, which may be varying with time or of unknown characteristics at the receiving end, adaptive equalization is used.

An adaptive filter has the property that its frequency response is adjustable or modifiable automatically to improve its performance in accordance with some criterion, allowing the filter to adapt to changes in the input signal characteristics. Because of their self-adjusting performance and in-built flexibility, adaptive filters have found use in many diverse applications such as telephone echo cancelling, radar signal processing, navigational systems, equalization of communication channels, and biomedical signal enhancement.

In summary we use adaptive filters :

When it is necessary for the filter characteristics to be variable, adapted to changing conditions,

When there is spectral overlap between the signal and noise, or if the band occupied by the noise is unknown or varies with time.

1.6 Main Components of the Adaptive Filter

In most adaptive systems, the digital filter in figure 1.6. is realized using a transversal or finite impulse response (FIR) structure figure 1.6.1. Other forms are sometimes used, for example the infinite impulse response (IIR) or the lattice structures, but the FIR structure is the most widely used because of its simplicity and guaranteed stability. For the N-point filter depicted in figure 1.6.1, the output is given by

$$\hat{n}_k = \sum_{i=0}^{N-1} w_k(i) x_{k-i} \tag{1.1}$$

where $w_k(i)$, i=0,1,..., are the adjustable filter coefficients (or weights) and $x_k(i)$ and $\hat{x}_k$ are the input and output of the filter. Figure 1.6.1. Illustrates the single-input, single-output system. In a multiple-input single-output system, the x_k may be simultaneous inputs from N different signal sources.

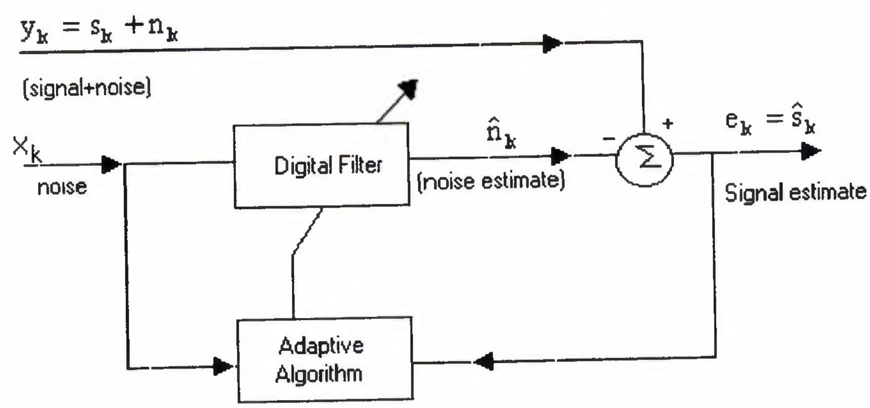


Figure 1.6. Block Diagram of an Adaptive Filter as a Noise Canceller

1.7 Other Applications

1.7.1 Loud Speaking Telephones

- The hybrid network is used to separate the transmit and receive paths (that is, the loudspeaker from the microphone), but there is a significant acoustic coupling between the loudspeaker and the microphone because of their proximity as well as a leakage across the imperfectly matched hybrid network.
- The difficulty then is how to provide adequate gain for the receive and transmit directions without causing instability.
- The conventional solution to the problems is to use a voice-activated switch to select the transmit and receive paths, but this is not satisfactory because it does not allow full duplex communication.
- A better solution is to use adaptive filtering techniques to estimate and control the acoustic and hybrid echoes Figure 1.7(b). The number of filter coefficients here can be quite large, for example 512, making the use of a fast algorithm attractive.
- In teleconferencing networks (or public address systems) acoustic feedback leads to problems similar to those described above. Adaptive filters used for these may require large numbers of coefficients (250 to 1000), especially in rooms with long reverberation times, and must converge rapidly.

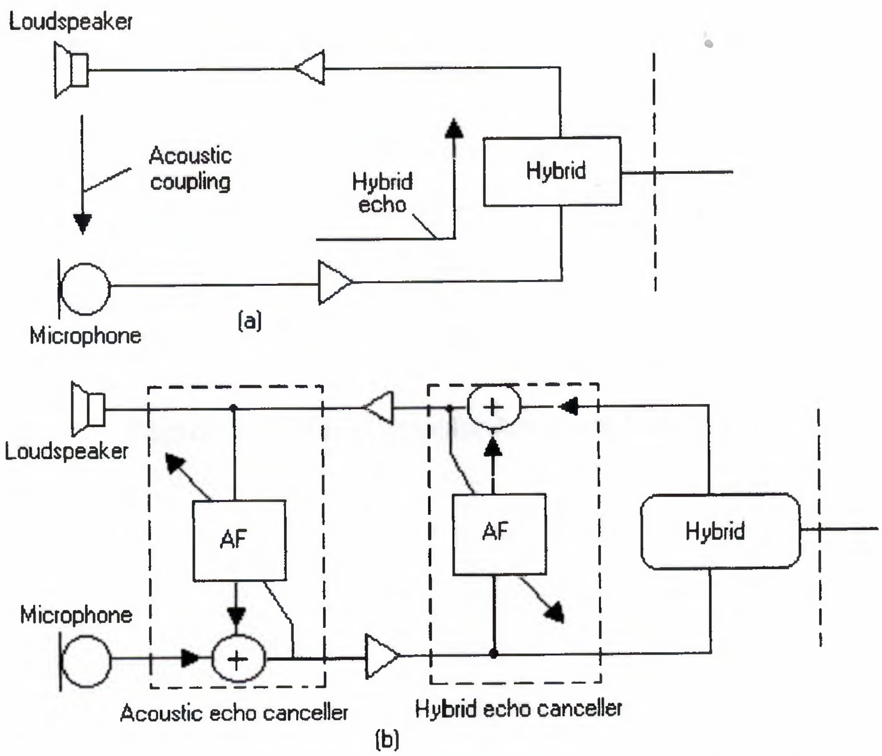


Figure 1.7. (a) Loud Speaking Telephone (b) Acoustic and Hybrid Echo Cancellation in Loud speaking Telephone

1.7.2 Radar Signal Processing

Adaptive signal processing techniques are widely used to solve a number of problems associated with radar. For example, adaptive filters are used in monostotic radar systems to remove or cancel clutter components from the desired target signals. In HF ground wave radar, adaptive filters are used to reduce co-channel interference, which is a major problem in the HF band.

1.7.3 Separation of Speech Signals From Background Noise

Acoustic background noise is a serious problem in speech processing. An adaptive filter may be used to enhance the performance of speech systems in noisy environments (for example in fighter aircrafts, tanks, cars) to improve both intelligibility and recognition of speech.

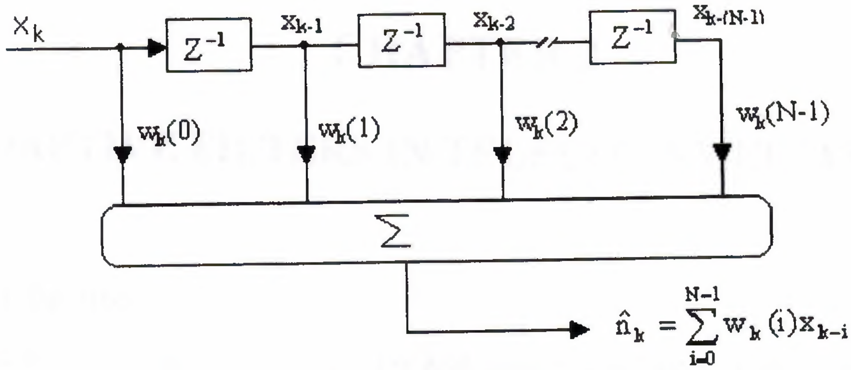


Figure 1.7.3. Finite Impulse Response Filter Structure

CHAPTER 2

ADAPTIVE FILTERS IN TELECOMNMUICATIONS

2.1 Introduction

Telecommunications is a growing and changing industry, which has proved to be fertile ground for the application of adaptive filters. The reasons for this are threefold: rapid advances in silicon technology, especially the advent of large-scale integrated (LSI) and very large scale integrated (VLSI) circuits, have made possible the implementation of adaptive filters at commercially acceptable costs; second, a rapid growth in data communications has created a need for adaptive filtering to overcome impairments inherent in existing telephony networks; and third, a desire to provide improved speech communications where echoes cause subjective impairment or instability. The two broad areas of data transmission and speech communications provide a natural division for the material in this chapter. However, it is instructive to remember first the two different roles that adaptive filters play, namely as equalizers and cancellers.

For equalization the adaptive filter is cascaded with an unknown linear channel $C(f)$ and its purpose is to approximate the inverse of $C(f)$. In the cancellation role the adaptive filter is in parallel with the unknown linear channel and is required to approximate $C(f)$. Adaptive filters are used in both these roles in telecommunications applications.

This distinction between roles is important because it results in different constraints on the operation of the adaptive filter: for example, interfering signals or noise at the output of the unknown channel have a different effect on the adaptive filter in each case; also for equalization, but not for cancellation, the channel characteristics can affect the rate of convergence of the adaptive filter.

In other chapters of this book various adaptive filter structures and adoption algorithms are described. Here we shall be concerned almost exclusively with transversal filters adjusted using the stochastic gradient least-mean-squares (LMS) algorithm and its variants. Although other structures and algorithms have been investigated for telecommunications applications, they are of less practical importance. This is a testimony to the simplicity of the transversal structure and the robustness of the stochastic gradient LMS algorithm. Other structures and algorithms are mentioned where appropriate.

The bulk of the material that follows is concerned with digital data transmission over telephony channels and metallic pair cables. This is a reflection of the vast amount of research and development that has been expended in this field and its importance in providing the means of digital communication over a network dominated by the needs of speech communication.

The remainder of the chapter is concerned with applications where adaptive filters are required to suppress echoes in speech communications. Alternative methods of achieving the same results are already used but adaptive filters provide a subjectively more acceptable performance.

2.2 Data Transmission

Although data transmission in the form of telegraphy predates telephony, speech communication came to dominate the evolution of telecommunications networks. Developed countries, therefore, have telephony networks that are unrivaled in their ubiquity and offer worldwide communication. When the growth in computer usage created a need for data communications it was not surprising that telephony networks initially offered the best medium for this communication. Unfortunately, transmission systems in telephony networks were optimized for analog speech waveforms and introduce various impairments that impede data communications. The most serious of these impairments are linear distortions, and linear filters could be used to equalize or cancel the distortion. However, such distortions vary widely between different network connections, so it became necessary to use adaptive filters.

Today, adaptive filters are widely used to provide equalization in data modems which transmit data at rates of 2400 bits/s up to 16,000 bits/s over speech-band channels (nominally, 300 to 3400 Hz). Although it is theoretically possible to achieve even higher rates, it is practically difficult to obtain a satisfactory error-rate performance without recourse to wider bandwidths. Higher-speed data modems (48,000 to 72,000 bits/s) are commercially available for operation over wider-bandwidth (60 to 108 kHz) channels, and some of these use adaptive equalization.

Recently, there has been a growing interest in duplex data transmission over speech-band circuits, which has resulted in adaptive filters being investigated for use as echo cancellers. As yet, very few modems using echo cancellers are commercially available, but that situation may well change in the next few years. Both these applications are described in this section, but first an outline of the types of linear distortion encountered in telephony channels is necessary.

2.2.1 Linear Distortions in Telephony Networks

Linear distortions arise in many different ways in telephony networks, but three distinct types can be identified: amplitude distortion, group-delay distortion, and echoes. Figure 2.1 illustrates how these arise in a typical telephony network connection. A subscriber is usually connected to his or her local switch by metallic pair cable; within the speech band this introduces amplitude slope, as shown in Figure 2.2(a). Between the local switch and other switches there may be loaded junction cable, which introduces group-delay distortion at the top end of the speech band, as shown in Figure 2.2(b). Between switches four-wire circuits are used to enable signal amplification and multichannel transmission systems to be employed.

Multichannel transmission systems use band-limiting filters, which introduce both group-delay, and amplitude distortion, as shown in Figure 2.2(c) for frequency-division multiplex (FDM) carrier system filters. Hybrid transformers are used to separate the go and return paths of the four-wire circuit and should ideally introduce infinite attenuation between the two paths. In practice the attenuation is finite, allowing signals to circulate around the four-wire loop, creating echoes. Those appearing back at the transmitter are referred to as talker echoes, while those arriving at the receiver are called listener echoes. Impedance mismatches in the network are a further source of echoes. Listener echoes give rise to ripples in the frequency response of the channel, the amplitude of the ripples being proportional to the echo-to-signal ratio and the frequency of the ripple being proportional to the echo delay.

Real network connections are often more complicated than this simple model and are becoming more so as modern pulse-code modulation (PCM) transmission systems and digital switches are introduced. However, the three basic impairments remain and identifying them separately helps us to understand what the adaptive filters used to combat linear distortion are required to do and how they behave.

Effects of modulation and demodulation. Because the telephony channel is band pass and generally passes through multichannel transmission equipment, which introduces small frequency offsets, data transmission systems use modulation to place the signal spectrum in the usable bandwidth and demodulation to recover the data and remove offsets. A simple system is shown in Figure 2.3 a stream of binary data (symbols) at rate 1/T is band-limited to $\frac{1}{2}(1+\alpha)/T$ Hz where α is the roll-off factor ($0< a\leq 1$), and modulated onto a carrier of frequency fc; if fc is chosen to be near the center of the speech band, the full signal spectrum can be received at the far end provided that $fc-\frac{1}{2}(1+\alpha)/T > 300$ and $fc+(1+\alpha)/T < 3400$. Demodulation by a carrier of the correct phase recovers the baseband signal, which is sampled every T seconds at the appropriate instant to detect the data with a low probability of error. If the channel is nondistorting and the two band-limiting filters are correctly designed, then, at the sampling instant, data symbols do not interfere with each other. If the channel introduces linear distortion, intersymbol interference (ISI) is caused, which degrades the performance of the system. If a single impulse (data symbol) is applied to such a system, then at the output of the receiver low-pass filter an impulse response is obtained which is the equivalent baseband impulse response of the channel. From knowledge of the three basic linear channel impairments, fc, the demodulating carrier phase φ, and the band-limiting filter responses, the equivalent baseband impulse response may be calculated. The 151 caused by the linear distortion is governed by the impulse response sampled at T spaced intervals.

Simple double-sideband amplitude modulation (DSBAM) as shown in Figure 2.3 uses twice the bandwidth required for the baseband signal; practical data transmission systems use more efficient modulation methods, of which one is of particular interest: quadrature amplitude modulation (QAM).

A QAM signal is formed by summing two DSBAM signals where the two carriers are of identical frequency but are 90° out of phase, as shown in Figure 2.4. At the output of a nondistorting channel, the two DSBAM signals are separated by demodulating with two carriers again 90° out of phase. Distortion, as well as introducing ISI into each of the two equivalent baseband channels, causes interference between the two channels. The equivalent baseband channel can be drawn as the cross-connected networks shown in Figure 2.5. A very convenient way of representing this is to regard the two data inputs (and outputs) as real and imaginary and then the equivalent baseband response may be represented by a complex impulse response.

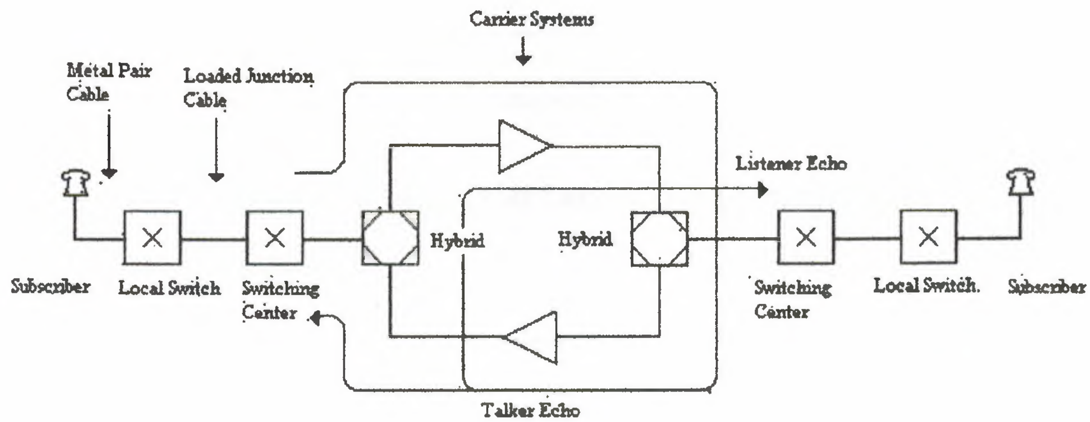
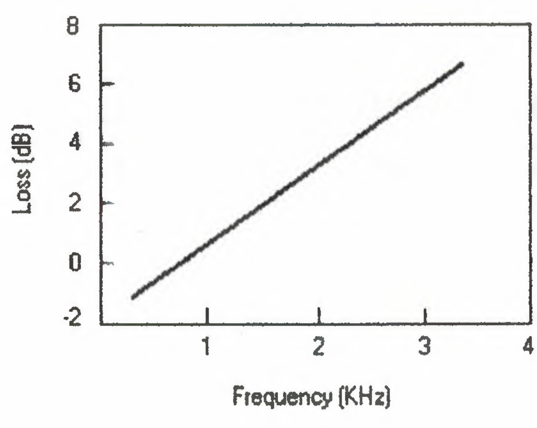
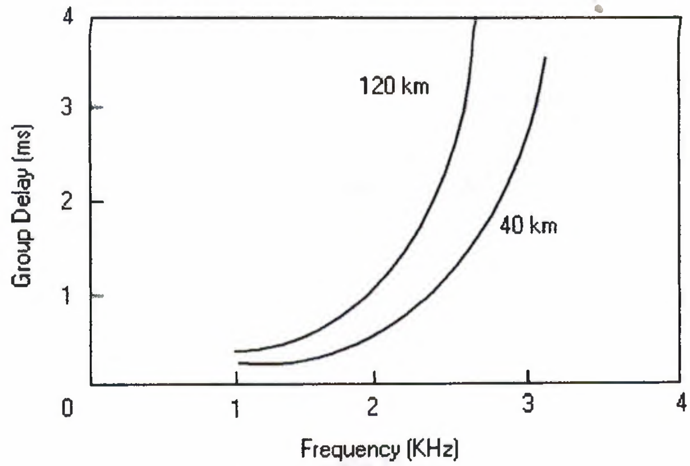


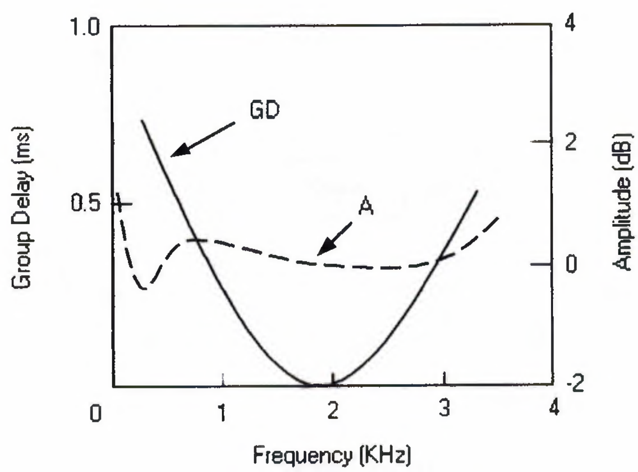
Figure 2.1 typical telephony network connection



(a) Amplitude Slope Of 4 Km Of Local Cable.



(b) Group Delay Delay Distortion of Loaded Cable.



(c) Linear Distortion of FDM Carrier Filters.

Figure 2.2 Circuit Characteristics.

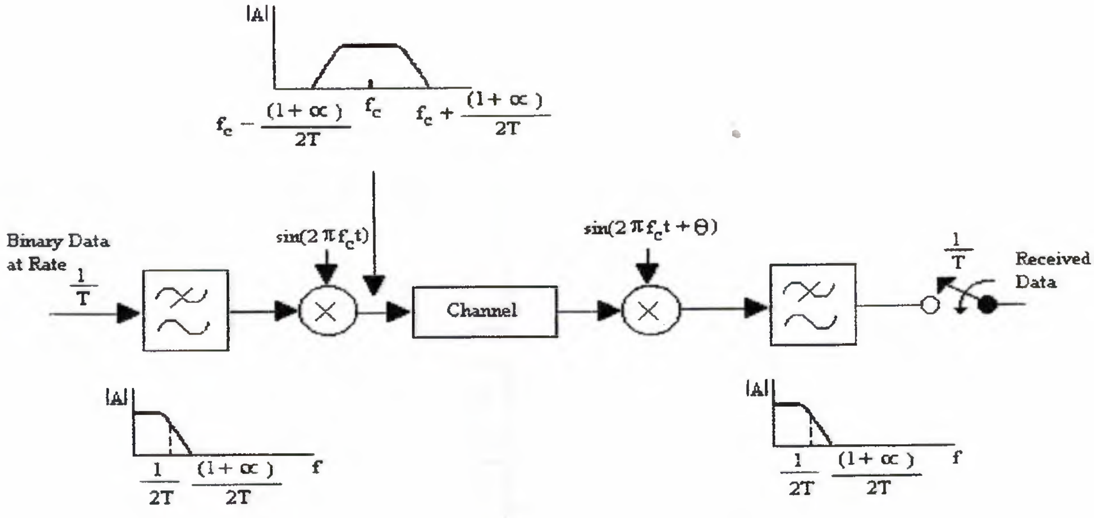


Figure 2.3 Simple Data Transmission System.

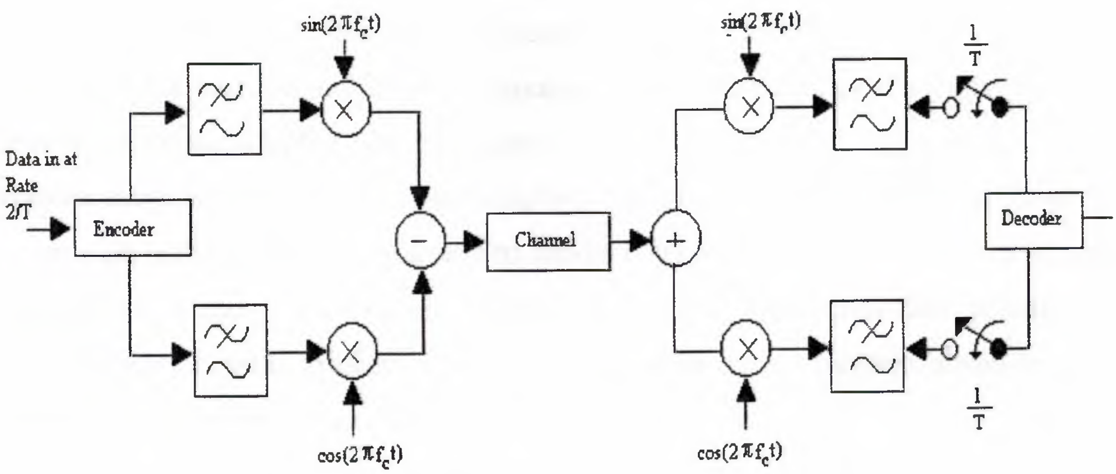


Figure 2.4 Quadrature Amplitude Modulation (QAM) Data Transmission System

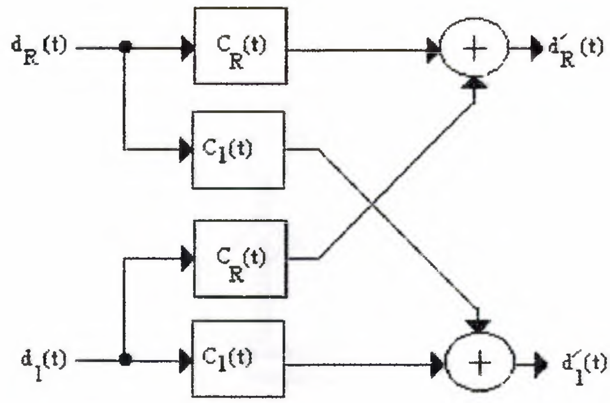


Figure 2.5 Equivalent Baseband Response Of QAM System

The concept of a complex channel response is very useful for the design of QAM modems. The imaginary part of the channel impulse response is simply the Hubert transform of the real channel impulse response. To equalize or cancel a complex channel response, a complex adaptive filter is required.

Complex adaptive filters. In earlier chapters the adaptive filters considered have been operating on real signals; the extension to handle complex signals is straightforward and is presented here without proof Figure 2.6 shows a complex transversal filter updated using the complex stochastic gradient LMS algorithm. Indicating complex quantities by an asterisk superscript, the output of the filter is given by

$$\acute{y}^*(n) = S^T{}^*(n)H^*(n) \qquad (2.1)$$

Adaption by the stochastic gradient algorithm with a fixed-gain constant μ is by the recursion

$$H^*(n+1) = H^*(n) + \mu\,[S^*(n)]'[y^*(n) - \acute{y}^*(n)] \qquad (2.2)$$

where the prime indicates conjugation of the complex quantities in the vector.

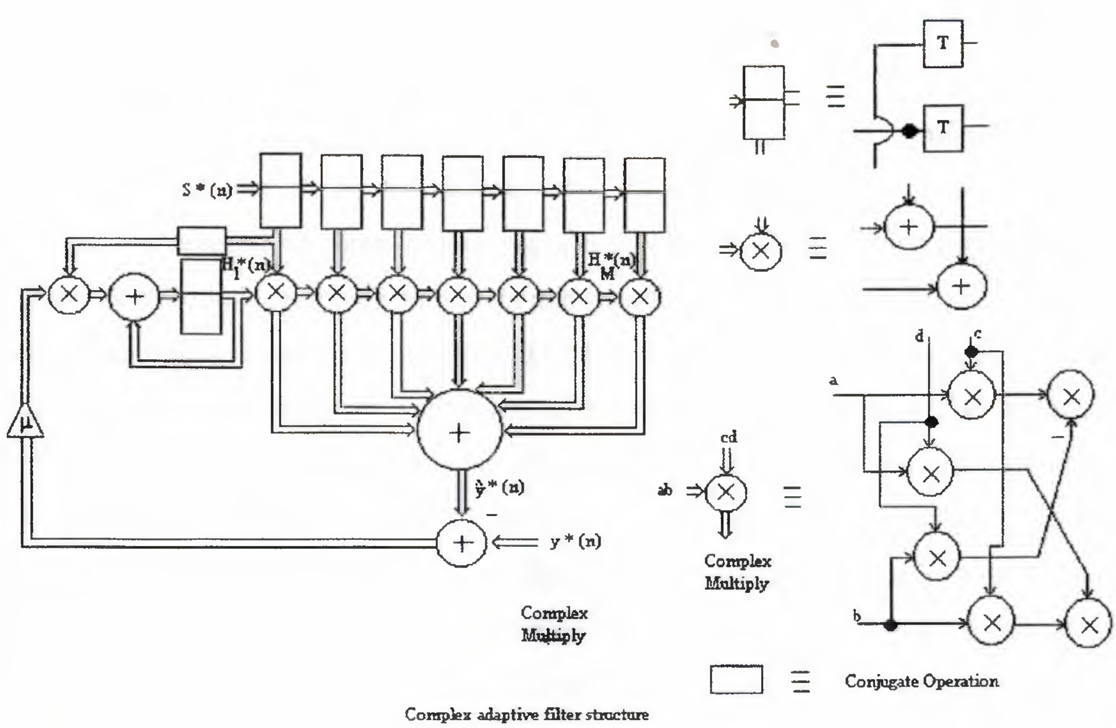


Figure 2.6 Complex adaptive filter structure

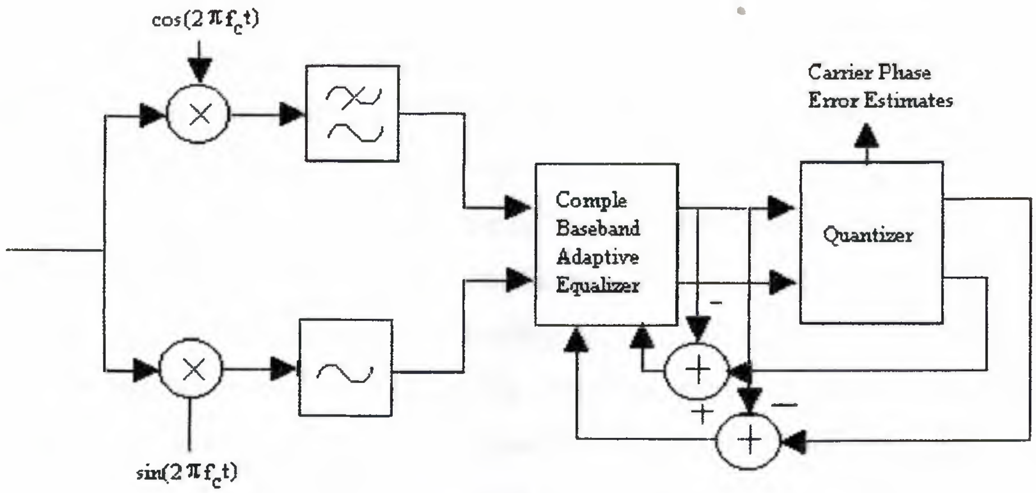
2.2.2 Speech-Band Equalizers

Most speech-band modems conform to CCITT-recommended modulation formats, which, for the high-speed modems needing equalizers, involve either pure phase modulation or combined phase and amplitude modulation. Both types of modulation can be viewed, as forms of QAM and so require the use of complex adaptive equalizers.

As well as linear distortion, speech-band channels generally introduce frequency offset and phase jitter onto the data signal. The modem receiver therefore has to use some form of carrier-phase tracking circuitry to remove frequency offset and reduce phase jitter. There are two common modem structures for combining complex adaptive equalizers with carrier phase tracking; Figure 2.7 shows them in block diagram form. In the first, equalization is performed on the complex base-band signal after demodulation using quadrature carriers obtained via a digital phase-locked-loop (DPLL) from carrier-phase error estimates generated at the quantizer. The quantizer is the device that decides which of the two-dimensional signal states is being received at the time of sampling and outputs the complex number corresponding to the signal state.

The error signal for adapting the equalizer is the difference between the input and output of the quantizer. In the second structurethe order of demodulation and equalization is reversed. As the equalization is performed on the modulated data signal the error signal for the equalizer has to be modulated using the recovered carrier phase information; although, involving more signal processing, the second structure is often preferred because it does not introduce delay via the filters and equalizer in the phase tracking loop, so that rapid phase jitter is more easily tracked. The DPLL and the equalizer both derive tracking information from the quantizer, and as the equalizer taps are a function of the demodulation carrier phase (a change of phase of + ϕ will rotate each complex tap by - ϕ), careful design of a modem is required to ensure that the two loops do not interact adversely. One way of ensuring this is to use a joint gradient algorithm or, alternatively, make one loop much slower to respond to changes than the other. In telephony networks the linear distortion characteristics do not usually vary significantly rapidly for this to be a disadvantage. The convergence behavior and residual error of the two-equalizer arrangements, known as baseband and pass band equalization, respectively, are equivalent.

Types of equalizer. : As well as the way in which the equalizer is combined with carrier-phase tracking, there are further variations on the way the adaptive transversal filter is used as an equalizer. For simplicity we will describe these variations in terms of their action on a real equivalent baseband channel; they can be applied to complex baseband channels and to the pass band structure as well. There are three important types of equalizer used in speech-band data communications:



(a) Baseband Equalization

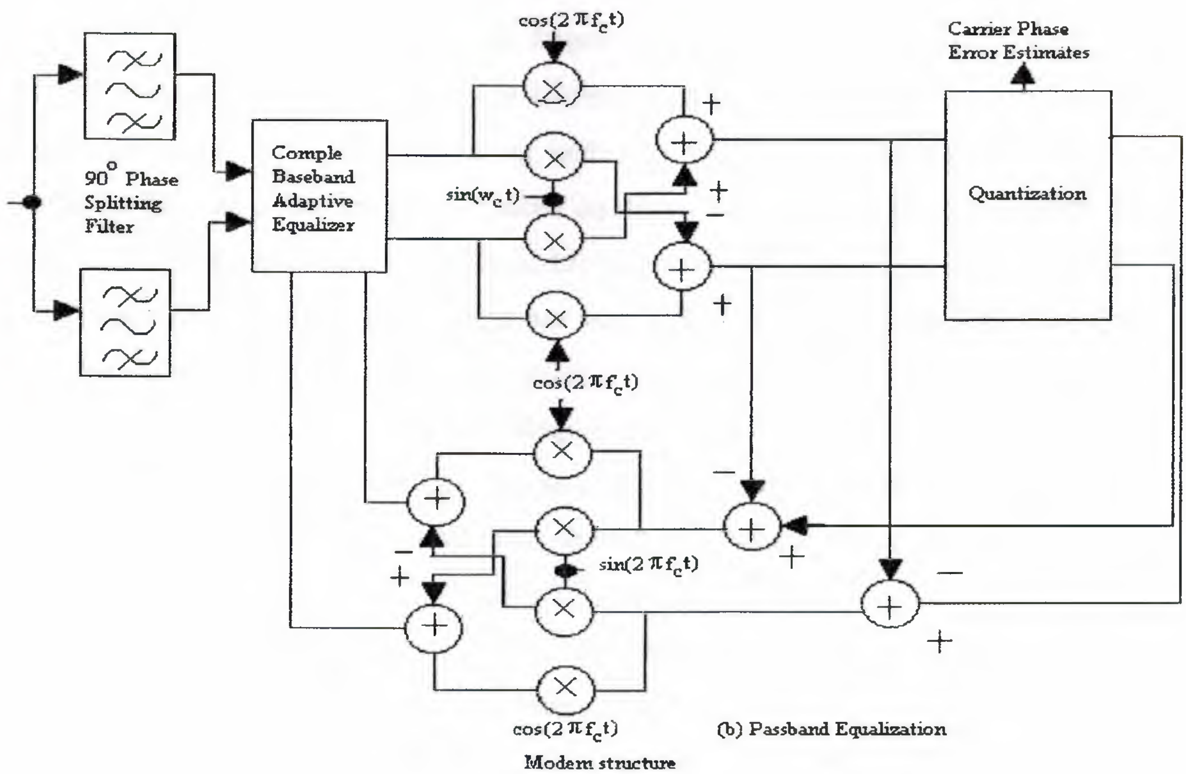


Figure 2.7 Modem structures

1. The T-spaced equalizer.
2. The fractional tap (FT) equalizer.
3. The decision feedback (DFB) equalizer.

To understand the reasons for using these different types of equalizer we must first elaborate on the concept of equalization presented in the introduction. The impulse response of the equivalent baseband channel sampled at rate l/T may be represented by the z-transform C(z). The frequency response of C(z) is defined completely by the response in the bandwidth 0 to O.5/T Hz. The coefficients of C(z), and therefore the frequency response C (f) are a function of the sampling phase. The unsampled baseband data signal occupies a bandwidth of ½ $(1+\alpha)$ and the sampling process causes spectral components above 0.5/T to fold over and add to components below O.5/T. For a distortionless channel, properly designed band-limiting filters, and a correct choice of sampling phase the fold-over process results in the sampled channel frequency response having flat amplitude and linear-phase from 0 to 0.5 /T Hz. Distortion in the baseband channel in the region up to½ $(1+\alpha)/T$ will cause the sampled channel frequency response to deviate from this ideal; so $C(f) \neq 1$ and the job of the equalizer is to restore as far as possible a flat amplitude and linear-phase response. A transversal filter with taps spaced at T intervals and sample rate l/T can do this and is known as a T-spaced equalizer. In situations where the group-delay distortion is changing relatively slowly in the region $½(1 - \alpha)/T$ to $½(1 + \alpha)/T$, the T-spaced equalizer operates very well and has been widely used. However, when the group-delay distortion is more severe, the summation of components about 0.5/T can lead to deep nulls in the sampled channel amplitude/frequency response, especially if the timing phase chosen for the sampling is inaccurate. As the equalizer attempts to remove these nulls it can amplify channel noise by an unacceptable amount. This disadvantage may be avoided by using a fractional tap (FT) spaced equalizer.

An FT equalizer is shown in Figure 2.8. The equalizer is now an adaptive filter with taps spaced at nT/rn, where n and m are integers ($n < m$), and sample rate m/T. Sampling at rate 1/T takes place after the equalizer. Such an adaptive filter can correct the channel response up to0.5m/(nT) hertz. If $½(1 + \alpha)/T < 0.5m/(nT)$, any adverse fold-over problems can be avoided because the channel is corrected before T-spaced sampling. It should be noted that it is the T-spaced sampled channel spectrum that is equalized; the frequency response before T-spaced sampling of the combined equalizer and channel $L(f) = H(f)$ $C(f)$ is, ideally, flat amplitude and linear phase, that is, $\exp(-2\pi j f\, t_0)$, where t_0 is an arbitrary delay, from 0 to ½(1 — α)T Hz, but from ½ (1 — a)/T to ½(1 + α)/T it is such $L[(1/2T) - f] + L[(1/2T) + f] = \exp(-2\pi j f t_0)$.

FT equalization is so effective that the residual error performance of the equalizer is virtually independent of timing phase. In addition, the FT equalizer can provide more optimum filtering of the received data signal, giving a better signal-to-noise ratio at the data detector. The penalty paid for this improved performance is that the number of taps for a given equalizer time span is increased by the factor m/n and the number of delay elements by m. However, for representative telephony channels with only amplitude and group-delay distortion for a fixed number of taps, the FT equalizer gives a better performance than does the T-spaced equalizer. With listener echo present, however, the T-spaced equalizer with its greater time span may be preferable.

For equivalent baseband channels with severe amplitude distortion both the T-spaced and FT equalizers enhance channel noise because they introduce gain to combat the amplitude losses. Another alternative in this case is the DEB equalizer, shown in Figure 2.9. A pure DFB equalizer is shown on the right-hand side of the illustration: the detected data symbols are used as the input to a transversal filter whose output is subtracted from the received signal. If the main (largest) sample of the channel impulse response is the first, the ISI samples that follow are removed by the transversal filter, whose taps are equal to the ISI samples. Thus the pure DFB equalizer is, by the definition given in the introduction, operating as an ISI canceller, not as an equalizer. However, common usage has sanctioned the term "DEB equalizer."

Because the filter operates on noiseless data (post decision) the channel noise is not enhanced and equivalent baseband channels with severe amplitude distortion can be equalized more effectively. However, for pure amplitude distortion the impulse response is symmetrical about a peak and the DFB equalizer cannot cancel the prepeak ISI. Therefore, the DFB equalizer is usually preceded by a T-spaced equalizer, which has the job of equalizing the prepeak ISI. Comparisons of T-spaced and DFB equalizers suggest that there is a performance advantage to be gained from the use of the DFB equalizer especially when the data transmission system bandwidth is such that severe distortion is being experienced at the edges of the data signal spectrum. The DFB equalizer is also very good for removing listener echo with no noise enhancement (echoes cause pronounced ripples in the amplitude frequency response). Linear equalization and cancellation can also be combined in other ways to give improved performance. It has been shown, for example, that using tentative decisions obtained after linear equalization as inputs to an ISI canceller can give improved performance.

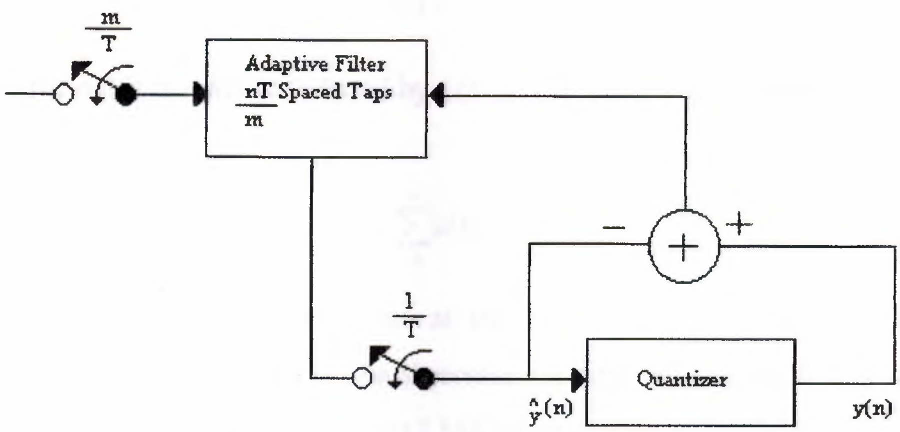


Figure 2.8 Fractional Tap Equalizer

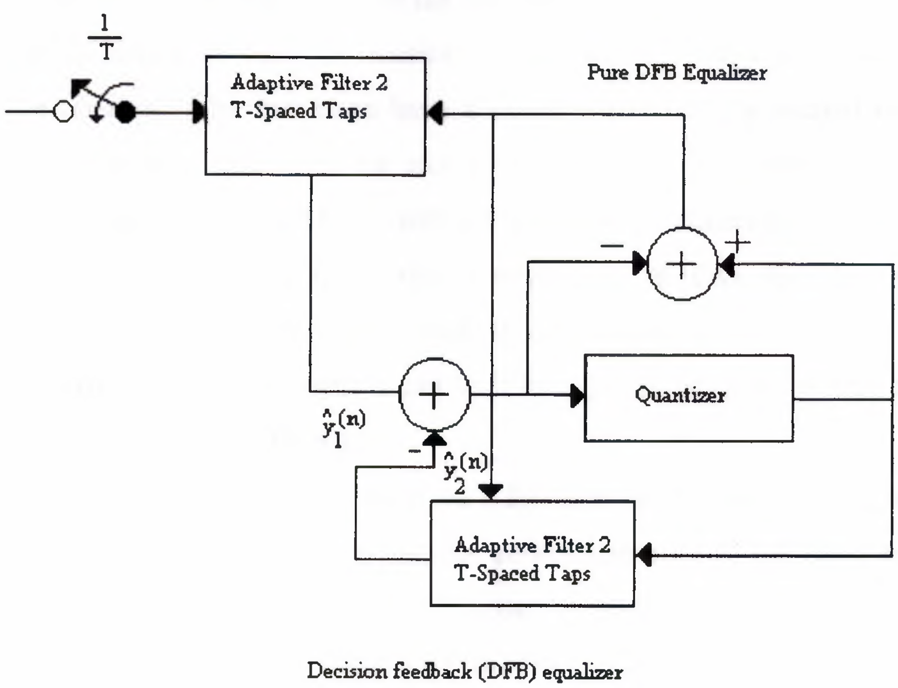


Figure 2.9 Decision Feedback (DFB) Equalizer.

Equalizer adaption. The job of an equalizer is to approximate the inverse channel response with a finite number of taps. There are various ways of doing this, but the most robust is to adjust the equalizer taps so that the sum of the mean square residual ISI and noise is minimized, that is, if the combined channel and equalizer response is

$$L(z) = z^{-M} + \sum_{-\infty}^{\infty} P_i z^{-i} \tag{2.3}$$

where M is the delay ($i \neq M$) introduced by $L(z)$ to the main sample, then the LMS equalizer minimizes

$$E\left\{ \sum_{-\infty}^{\infty} \left| d_i p_i \right|^2 + \sigma^2 \right\} \tag{2.4}$$

where σ^2 is the variance of the noise at the output of the equalizer and di are the transmitted complex data elements. The algorithm for achieving adaption to this state is the complex version of the stochastic gradient LMS algorithm given by (2.2).

An immediate problem that arises is that of the reference signal y*(n) required to form the error signal for the equalizer. By the very nature of data communications the input to the channel is separated from the receiver. There are two ways in which a reference signal may be obtained. The first is to have a stored reference; the second is to use the output from the decision circuit in the modem, as indicated in Figure 2.7. The stored reference, which has to be synchronized with the transmitted sequence, is used to train the equalizer initially; but for tracking during transmission of data the decision-directed technique must be used. Decision-directed training without any stored reference is possible, but high error rates before convergence can lead to false convergence of the equalizer to nonglobal quasi-stable minima [Mazo].

The rate of convergence is shown to be a function of the number of equalizer taps, the gain constant μ of the update loop and the power spectrum of the input signal to the filter. Generally, the number of taps will be chosen to meet the equalization requirement and the value of g is governed by stability constraints and the amount of tap jitter that can be tolerated.

The power spectrum of the input to the filter is determined by the equivalent sampled baseband channel amplitude/frequency characteristic and the power spectrum of the transmitted data sequence. It is usual to ensure that the power spectrum of the data sequence is white by employing data scramblers and descramblers at the transmitter and receiver, respectively. The convergence of the filter is then a function of the equivalent baseband channel amplitude/frequency characteristics only.

Fortunately, it has been shown that the channel characteristics only weakly affect the convergence rate and as, for most applications, the convergence rate of the equalizer is not particularly critical, the standard complex stochastic gradient LMS algorithm is adequate. There is one application, however, where the rate of convergence is crucial. In some data communication networks a central modem polls each of a number of out-station modems all connected to a multipoint circuit. To receive a reply from each of the modems, the central modem has to train its equalizer in turn for the channel between each out-station and itself Often, the messages returned from the out-stations are short, so that the train-up time must also be short if it is not to be a significant proportion of the transmission time.
Various schemes have been proposed for achieving fast equalizer convergence, including frequency-domain equalization, matrix inversion algorithms, Kalman filter techniques, orthogonalization techniques, and cyclic equalization.

It is desirable from an implementation point of view to try to minimize the-complexity of the adaptive filter. To maintain linearity there is not much that can be done to reduce the accuracy requirements of the filter itself however; the variables in the stochastic gradient algorithm can be modified *drastically* without destroying *its* ability to converge, albeit at a slower rate. Digital signal processing can be much simplified if for multiplications the multiplier and/or the multiplicand are reduced in accuracy. The gradient algorithm variables that can be treated in this way are μ, the error signal, and the signal inputs to the correlation multipliers. As μ is a fixed quantity, setting it to 2^{-i}, where i is an integer, results in a simple shift in the complex error signal words.

Operations on the error signal also slow convergence. We shall return to this topic in Section 2.3, where simplifications in processing are even more important and drastic simplification of the error signal with a significant slowing of convergence can be made acceptable.

Equalizer complexity: The number of taps required in the complex transversal equalizer is governed by the severity and type of the distortions the modem is expected to work over, the carrier frequency fc, the roll-off factor α, the signaling rate $1/T$, and the required performance. The distortions encountered by the data transmission system depend on its application. Many high-speed modems are required to work over dedicated conditioned circuits. A conditioned circuit is one, which, within the network, has been equalized using fixed filters to within a recommended (e.g., CCITT Recommendation M1020) maximum group-delay and amplitude variation. The adaptive equalizer then has the job of removing any residual distortion and so needs very few taps. On the other hand, modems required to function over switched network channels encounter far more severe distortions and require much longer equalizers, especially if they must deal with long-delayed listener echoes.

The carrier frequency affects the amount of significant ISI by virtue of where it places the spectrum of the signal. As the carrier frequency is increased, for example, the upper frequencies of the signal spectrum will experience more and more distortion as the edge of the speech band is approached. Similarly, increasing the signaling rate or α will widen the signal bandwidth and thus affect how much band-edge distortion the data signal encounters.

The performance requirement of the equalizer can be expressed in a number of ways, but the most useful one from an equalizer design point of view is the mean-square error at its output (i.e., the mean-square residual ISI plus the variance of any noise). The target value of this quantity depends on the number of signal states in the modem line signal (e.g., signal phases for P5K) and the tolerable error rate.

Implementations. The first consideration in devising an implementation of an adaptive equalizer is the number of taps required; typical choices range from about 8 taps for a 4800-bit/s polling modem up to 64 or more for a 9600-bit/s modem intended for switched network applications. The second consideration is the sampling rate of the equalizer. Typically, these range from 600 samples per second up to 4800 samples per second or more. Although analog realizations of speech-band equalizers are possible, commercially competitive modems now almost exclusively use digital signal processing (DSP) realizations, which are both cheaper and give better performance.

An important consideration in a DSP realization concerns the word lengths required for each part of the equalizer structure. These will vary with the performance requirements of the equalizer, the number of taps, and the gain constant.

Two types of DSP realization can be used, each of which affects the word-length consideration in a different way.

The DSP hardware can be realized either as a, dedicated, or semidedicated, custom-designed LSI or VLSI circuit, or with a more general microprocessor architecture. The former is usually more efficient, but the latter allows for much greater flexibility in design so that one design of IC can, by reprogramming, be used for different modems, or even different signal processing applications entirely. Both approaches are used, although now that more complex VLSI implementations are possible, efficiency is not so important and there is a tendency to opt for the more flexible approach, allowing the development costs to be amortized over a greater number of products.

In the custom-designed approach the word lengths of each part of the equalizer can usually be specified independently and so are minimized to reduce the circuit complexity. In the microprocessor architecture there is usually a global word length, which must obviously be greater than the maximum word length required by the adaptive filter (and any other DSP functions required in the modem if these are also implemented on the same device). The precision requirements for adaptive equalizers have been studied theoretically [Gitlin and Weinstein 1979], but usually simulation studies are used to determine the necessary word lengths.

Generally, the tap coefficients require greater word lengths than the signal samples. Typical values range from 12 to 20 bits for the coefficients and 6 to 10 bits for the signal samples, depending on the number of tap.

An example of custom-designed LSI circuits implementing a complex adaptive filter for speech-band modem applications is shown in Figure 2.10. It consists of three different LSI circuits: an adaptive filter processing IC, a shift register IC, and an IC that performs the data detection and error signal generation as well as various other modem functions. Two of each of the processing and storage chips are combined with one of the data detection chips to form a 2400-sample per second 72-tap complex adaptive equalizer mounted on a hybrid circuit substrate. The circuits are realized in *5-μm* NMOS technology working at about 2 MHz clock rate and so by modern standards not particularly complex or fast. Other LSI implementations are described by [Guidoux and Le Riche, Murano et al.].

Speech-band adaptive equalizers have also been realized in standard bit-slice microprocessors, but competitive products are now usually based on LSI and VLSI circuits.

2.2.3 Echo Cancellation for Speech-Band Data Transmission

There is a growing need in data communications for duplex transmission over two-wire switched circuits. Where the data rate required is 2400 bits/s or less, frequency-division techniques are employed so that fixed filters can be used to separate a received signal from talker echoes. Above 4800 bits/s the limited bandwidth available in the speech channel precludes the use of frequency division because the data signals would require too many states for reliable detection without sophisticated and expensive processing. Cancellation of the talker echo using an adaptive filter is the only way of achieving two-wire duplex data transmission at the higher rates. In fact, echo cancellation has also been applied at 2400 bits/s in a commercially available modem [Stein] as an alternative to the frequency-division approach and higher rate designs are emerging. Figure 2.11 shows how an adaptive filter can be used to cancel talker echoes. As illustrated, the adaptive filter has a single input and output and is a wholly real filter. To cancel all the echo frequency components in the bandwidth of the received data signal, the sample rate of the adaptive filter must be at least twice the highest frequency present in the data signal spectrum. Therefore, if the sample rate *is* fs, *then*

$$fs > \frac{1+a}{T} + 2fc \tag{2.5}$$

The direct application of a real adaptive filter has two disadvantages

1. The adaptive filter is driven by analog samples or, if a digital implementation is used, digitally encoded analog samples. As we shall see later, the dynamic range of the filter is usually required to be large (e.g., >60 dB), so the filter delay line is required to store samples very accurately. Also, each of the multiplications in the filter is between two accurately represented quantities.

2. The input signal has sample-to-sample correlation imposed on it by the filters in the modem transmitter, which tends to slow the convergence of the filter.

Both these disadvantages are circumvented by using data-driven echo-canceller structures.

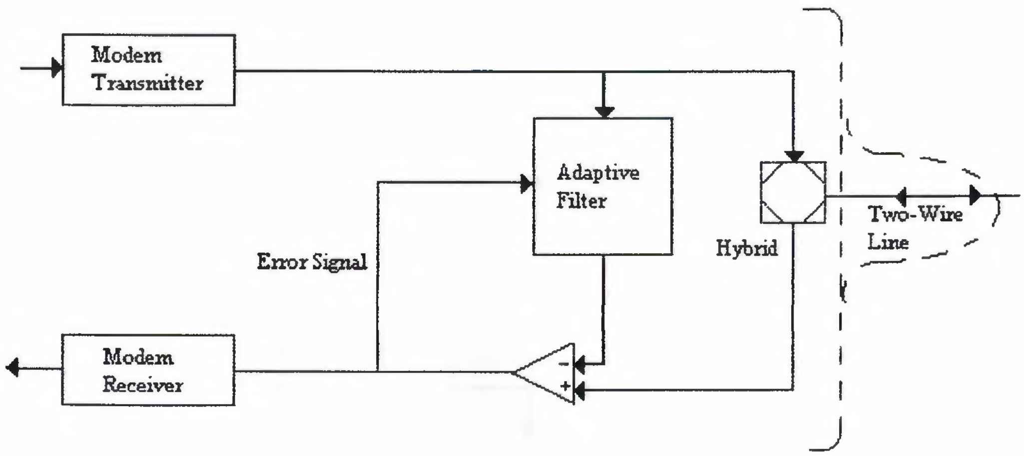


Figure 2.11 Echo Cancellation For Data Transmission

Data –driven echo cancellers.A data-driven echo canceller is shown in figure 2.12. A complex adaptive filter is driven by the transmit data after it has been encoded into its complex form prior to modulation. Because the canceller is canceling the line signal, a modulator operating at the transmitter carrier frequency follows the adaptive filter. Note that the sample rate of the filter is an integer multiple m of the modem signaling rate 1/T and still obeys (2.5). To generate the complex error signal for the filter, a complex line signal is formed with a Hibert transformer and then demodulated to adapt the baseband filter. The adaptive filter is required to model the equivalent baseband echo response convolved with the response of the spectrum shaping filters in the modem transmitter. The interpolation filter restores the real line signal from the T/m spaced samples to a continuous waveform ready for resampling by the modem receiver. This is necessary because the sample timing in the receiver is not necessarily of exactly the same frequency as the transmit timing. At first sight this structure looks far more complicated than the use of a real adaptive filter.

However, under certain conditions often pertaining in data transmission systems, it allows the amount of signal processing to be significantly reduced. There are also modifications to the basic structure, giving further savings in processing.

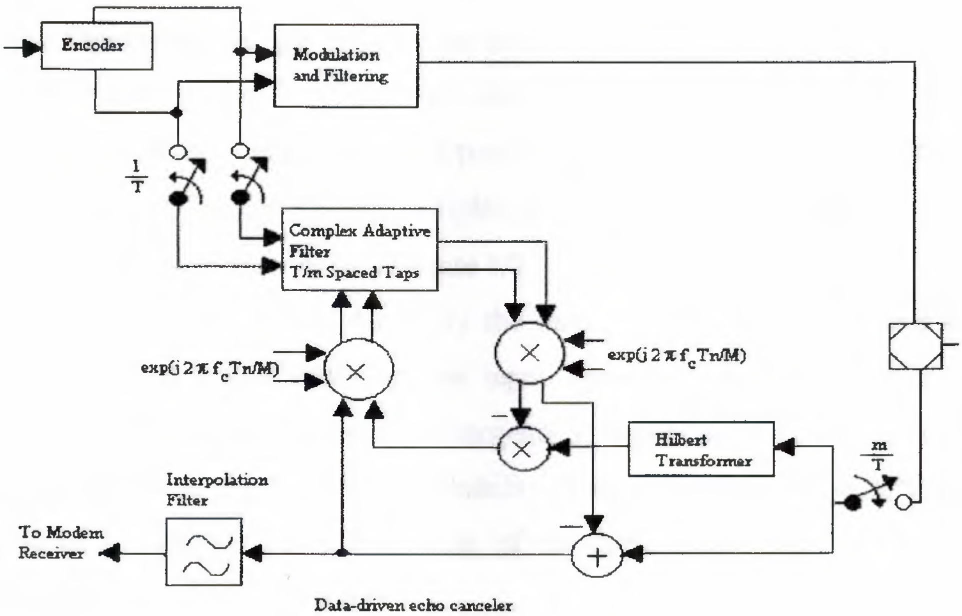


Figure 2.12 Data – Driven Echo Canceller

For many QAM signal formats the data elements after encoding consist of a few discrete levels. If a digital delay line is used in the adaptive filter, very few bits of storage are required for each delay element. This also means that in a digital realization one input to the tap and correlation multipliers has very few bits; the multipliers are, therefore, very simple to implement. The multiplications in the modulator remain as complicated operations. However, if the data system is such that its carrier frequency fc and signaling rate 1/T are related so that $2\pi f c\mathrm{T/m}$ is a multiple of $\pi/2$, the multiplications by sin $(2\pi f c\mathrm{Tn/m})$ and cos $(2\pi f c\mathrm{Tn/m})$ become multiplications by 0 or $\pm$ 1 or, by scaling by $\sqrt{2}$ and shifting by $\pi/4$, just $\pm$ 1. This condition is met in a number of modulation formats.

Another useful structure is obtained by reversing the order of modulation and adaptive filtering as shown in Figure 2.13. Provided that in this case the carrier frequency and signaling rate are such that $2\pi f c\mathrm{T}$ *is* a multiple of $\pi/2$, the data entering the adaptive filter are again very simple. In addition, the error signal does not need demodulating.

This structure has an additional advantage when it is required to cancel the real line signal only as shown in Figure 2.14. As the adaptive filter is required to produce only the real output, half the processing (that which produces the imaginary output) disappears. The error signal is now purely real, so the tap updating is simpler, but the penalty for this is that the mean convergence rate of the filter is approximately halved.

Adaptive Operation. In the data-driven structures the adaptive filters are driven by a succession of data symbols at T intervals with $m-1$ zero values between them. This means that the adaptive filter operates as m independent adaptive filters, each producing an output every T seconds, the outputs being multiplexed in time. We can, therefore, examine the convergence of a single filter of sample rate 1/T

As the echo canceller is driven by the data (or modulated data) and the data are normally scrambled before encoding, the input signal to the adaptive filter is spectrally white. Therefore, the echo canceller convergence is dependent only on the number of taps of the adaptive filter, the amplitude probability density function of the data symbols, and the value of μ. Analysis of the evolution of the mean-square tap maladjustment power (MSTMP) gives a recursion formula:

$$E\{|\xi(n+1)|^2\} = [1 - 4x\mu A + x\mu^2(B + 4(N-1)A^2)]\, E\{|\xi(n)|^2\} + 2x\mu^2 NAE\{|w|^2\} \tag{2.6}$$

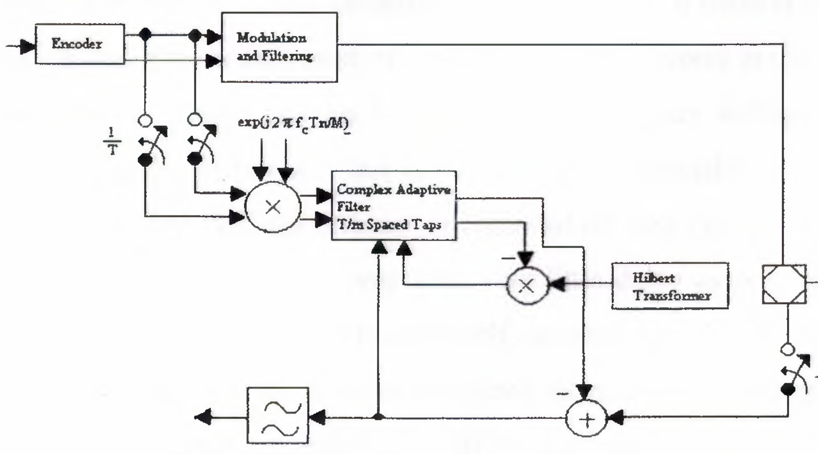
where N is the number of T spaced taps, $2A$ is the average value of the square of the modulus of the complex data elements, B is the average value of the fourth power of the modulus of the complex data elements, x is 1 for complex error signals and ½ for real error signals, and $E\{|w|^2\}$ is the expectation of the uncancelable component of the received signal. This includes echo components outside the span of the echo canceller. noise, and most important, the wanted data signal from the far end. The analysis assumes that $w(n)$ is uncorrelated with the input to the adaptive filter. To ensure that the wanted data signal is uncorrelated with the transmit signal, different scramblers and descramblers are usually employed for each direction of transmission. The residual MSTMP after convergence is obtained by iterating (2.6) to give

$$E\{|\xi(\infty)|^2\} = \frac{2\mu NAE\{|w|^2\}}{4A - \mu\left(B + 4(N-1)A^2\right)} \tag{2.7}$$

The fastest convergence is obtained by using the optimum gain constant

$$\mu_{opt} = \frac{2A}{B + 4(N-1)A^2} \tag{2.8}$$

Unfortunately, for μ(opt) and large enough N the residual MSTMP is, from (2.7), approximately $\frac{1}{2}E\{|w|^2\}$. The residual tap misadjustment gives rise to an uncanceled residual echo of power $2AE\{|\xi(\infty)|^2\} = E\{|w|^2\}$. As in a well-designed system the desired data signal from the distant transmitter is the dominant component of the received signal, then for μ(opt) the residual echo is as large as the wanted data signal! Therefore, the gain constant must be reduced to μt, which is the value of μ giving an acceptably small ratio $E\{|\xi(\infty)|^2\}/E\{|w|^2\}$. The trade-off between convergence rate and residual echo is illustrated in Figure 2.15. As μt is small, the adaptive filter can track only very slowly time-varying echo responses.



Modulated data-driven echo canceler

Figure 2.13 Modulated Data – Driven Echo Canceller

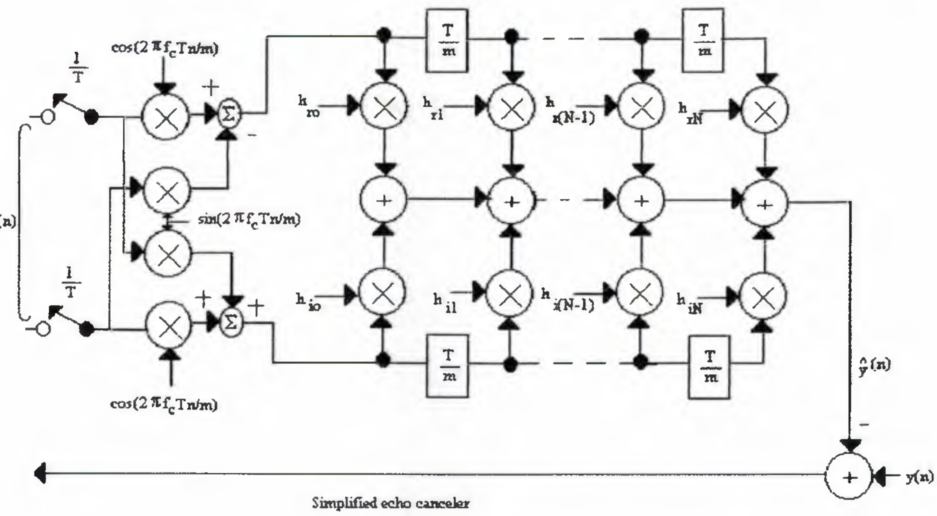


Figure 2.14 Simplified echo canceller

Echo characteristics of speech-band circuits. Echo cancellers in general have to model two types of echo signal. One is the result of leakage across the hybrid in the modem itself. Although, theoretically a hybrid can be balanced to prevent any leakage between the transmitter and the receiver, in practice this is very difficult, especially when a modem is required to work on any line, and the amount of trans-hybrid toss can be as low as 8 dB. Thus there is a large talker echo component with very little delay associated with it. The second type of echo results from the network itself, as shown in Figure 2.1. Generally, more than one discrete echo can occur. These echoes are usually smaller than the trans-hybrid echo (e.g.. 20 dB down) and can have a delay varying from a few milliseconds to hundreds of milliseconds if a satellite circuit is involved. Also, because the four-wire portion of the telephony circuit is nominally zero loss, there is only a weak correlation between echo amplitude and delay. A further problem is that the network echoes may be offset in frequency due to the modulation and demodulation taking place in four-wire carrier systems. Although the offsets are usually quite small (e.g., <1 Hz), the resulting time variation of the echo response is difficult for an echo canceller to track.

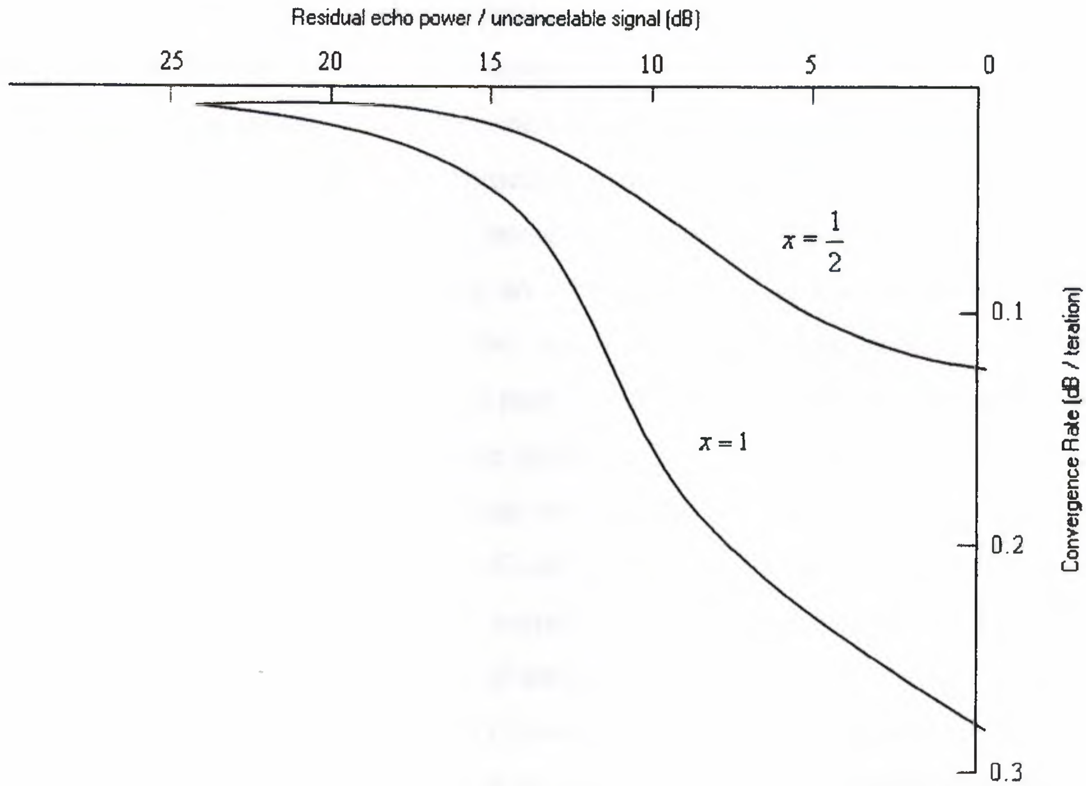


Figure 2.15 Performance Of An Echo Canceller Shown In Terms Of Residual Echo Versus Convergence Rate ($N = 16$, $A = 1/2$, $B = 1$)

Implementation considerations. Four fundamental parameters control the complexity of the echo canceller: the maximum echo signal level, the minimum received signal level, the required signal-to-uncanceled echo ratio, and the number of echo canceller taps. Earlier it was stated that the trans-hybrid loss could be as little as 8 dB; as the hybrid echo is the dominant one, this gives a maximum echo level of –8dB relative to the transmit level. The maximum loss over a full-switched network connection is ≈48 dB. Typically, a received signal-to-uncanceled echo ratio of better than 20 dB is required. This means the echo canceller must suppress echoes by more than 60 dB. Together with the fact that the echo canceller may have hundreds of taps, achieving this level of performance requires a digital implementation as far as possible; an analog implementation would introduce too much spurious noise. As the line signal is analog, the echo canceller must therefore contain an analog-to-digital converter (ADC) and a digital-to-analog converter (DAC).

Three structures are possible depending on whether the subtraction of the echo-canceller output is done digitally, by an analog sampled-data subtraction, or by subtraction of continuous analog signals. These three methods are illustrated in Figure 2.16 for the case of cancellation of the real line signal. Structure 1 requires an ADC and a DAC of such an accuracy that the quantization noise introduced onto the received wanted signal is insignificant. Structure 2 avoids having an ADC and DAC in the signal path, but the interpolation filter creates delay in the tap update loop, and because out-of-band components appear as noise in the signal path, the interpolation filter requires a high stop-band attenuation. Structure 3 avoids these problems: The loop delay is minimized, there is no ADC or DAC in the signal path, and the interpolation filter has a much less severe out-of-band attenuation requirement. The DAC still needs to be of sufficient accuracy that the quantization noise of the echo-canceller output is small compared to the received signal. However, the ADC does not need to be of such accuracy. In fact, it is possible to reduce it down to a single (sign) bit with the penalty of very much slower convergence.

It is important in all these structures to maintain good linearity in the transmitter, the hybrid, the ADC, and the DAC when they are in the signal and/or echo paths. The reason for this is that nonlinear distortion cannot be modeled using a linear adaptive filter. If the echo-cancellation requirement is for 60-dB suppression, any nonlinear distortion components must be more than 60 dB below the transmit signal level, as must any extraneous noise in the analog circuitry. These requirements place severe, although not impractical constraints on the circuit design.

Depending on the exact application, ADCs and DACs in the signal or echo paths have word lengths of 10 to 12 bits or more. The tap coefficient word lengths depend additionally on the number of taps and the value of g; typically 20 to 32 bits is required. These very long word lengths can be reduced by using averaged gradients for the updating, but the number of bits of storage per tap is still large because of the need to store the tap update during averaging. Clearly, speech-band echo cancellers require a great deal more storage for implementation than do equalizers, although using the data-driven structures;

the processing per tap is usually simpler.

If the echo canceller is required to deal with network echoes that are offset in frequency, the complexity is even greater. One technique for dealing with this case is to use a second adaptive transversal filter combined with a phase-tracking circuit to remove the frequency offset Clearly, this is a much more complicated implementation.

Design example: A 9600-bit/s echo-canceling modem. With the exception of the modem mentioned earlier, echo-canceling modems have yet to make a major commercial impact, although this situation is likely to change over the next few years. The following was a typical experimental design constructed for tests in the U.K.

The design used an experimental 4 x 4 QAM modem with a carrier frequency of 1800 Hz and signaling rate of 2400 baud, and was aimed at providing a large coverage of the U.K. national network. The echo canceller was of the type shown in Figure 2.14 and structure 1 of Figure 2.16. Its time span was 26 ms *(256, T/4* spaced complex taps) and it suppressed echoes to 60 dB below the transmit signal level. An efficient parallel processing hardware structure allowed the large amount of storage (approximately 15 kilobits) and processing (approximately 15 million add/subtract operations per second) to be implemented by standard MSI TTL and MOS integrated circuits on a single board of approximately 12 in. x *5* in. A very much more compact version using VLSI circuits should be possible in 3-μm technology.

Experimental echo-canceling modems were used to establish the feasibility of the echo-canceling technique for 9600-bit/s duplex transmissions over the U.K. switched network. Tests on a wide variety of network connections showed that the technique did work and in particular that there is normally no frequency offset on U.K network echoes. Shown in Figure 2.17 are oscillograms of typical echo-canceller tap values obtained by reading out the digital tap values (real and imaginary interleaved) through a DAC. These clearly illustrate the observations made earlier about the characteristics of echoes.

The experimental evidence is, therefore, that echo cancellation is a feasible technique for duplex data transmission over switched networks that contain no frequency offset echoes, have sufficient linearity, and do not have very long delayed echoes. Although echo cancellers can be devised to cope with all these problems.

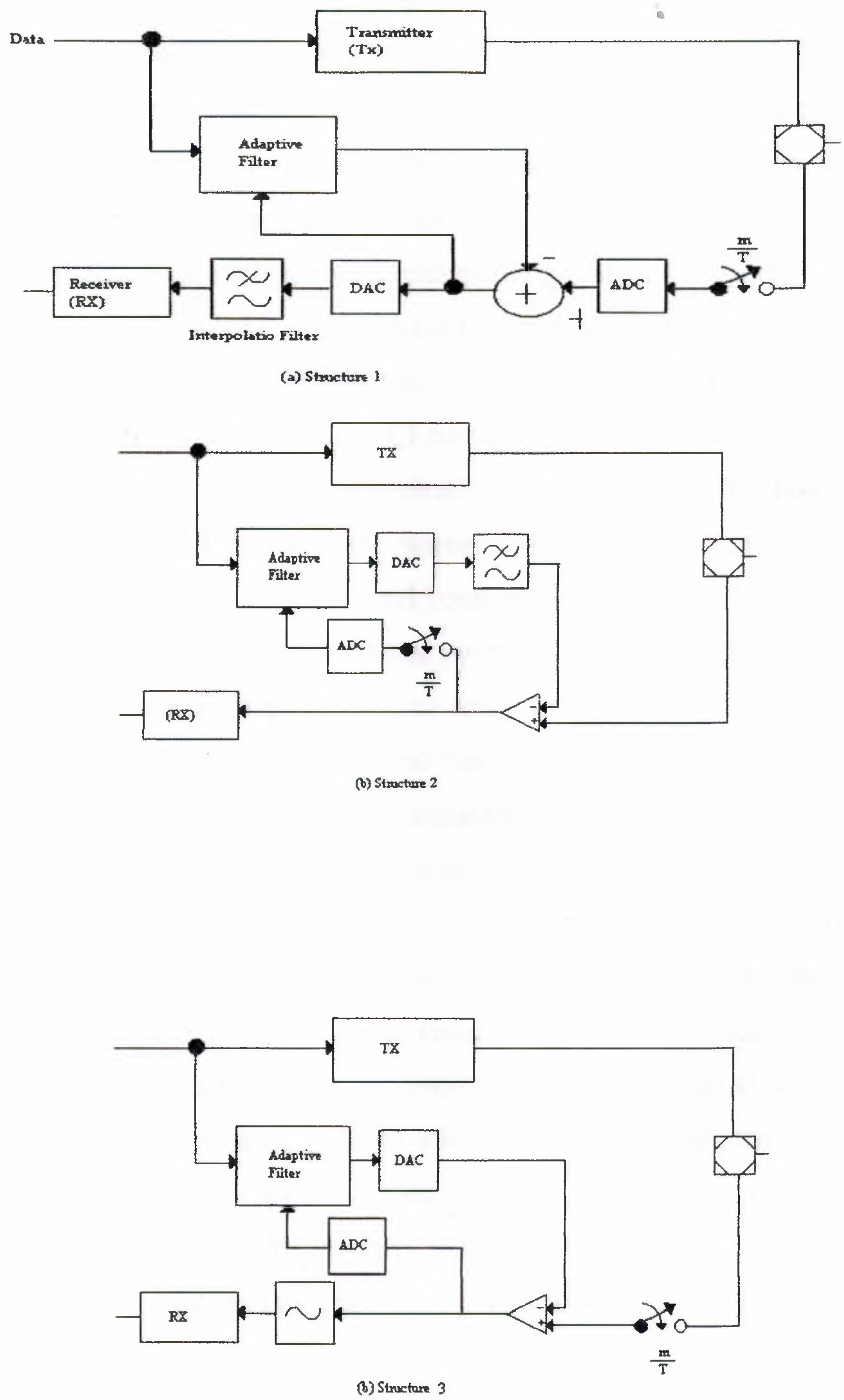


Figure 2.16 Implementation Of Various Echo Canceller Structures

2.3 Digital Transmission Over Local Networks

Telephony networks are making increasing use of digital transmission and switching techniques to provide considerable cost savings in network implementation and maintenance compared to the currently dominant analog equipment. The scale and pace of the introduction of digital equipment varies from country to country, but the time can be foreseen when most local switches are interconnected by digital channels.

These channels would still be aimed primarily at telephony but would also be capable of providing data communication an order of magnitude faster than speech-band modems (e.g., in Europe the standard CCI commended bit rate of 64 kilobits/s is used). The next step in the spread of digital communication is to connect each subscriber to his or her local switch by a digital transmission system. Such systems must be very cheap, must operate over existing local line plant, and because most subscribers can have only a single two-wire circuit, provide two-wire duplex operation. The last requirement has stimulated much interest in the use of adaptive filters as echo cancelers.

Subscribers' loops consist of twisted-pair metallic cables with a variety of diameters, several of which may occur on any one connection, and lengths varying from a few meters to several kilometers; consequently, the transmission characteristics of the connections vary widely. Figure 2.18 shows the insertion loss and phase characteristics for a length of a typical local network cable. Also shown is a plot of crosstalk attenuation. These characteristics illustrate two conflicting requirements. For a system to work over the longest possible line lengths, the power spectrum of the digital signal should be confined to as low a frequency band as possible to avoid crosstalk interference. However, by using modulation or line codes, which have their spectrum at the higher frequencies (where the linear distortion is less), no equalization is necessary. The reach of systems using such techniques is limited by crosstalk interference. This is not too significant a problem, as the majority of subscribers, especially in cities, are close to their local switch. Generally, therefore, two types of subscriber loop transmission system may be identified:

Those that are essentially self-equalizing (without an adaptive equalizer) but of limited reach, and those that to obtain a longer reach use baseband transmission with adaptive equalization to counter the linear distortion at low frequencies. Both types have been the subject of research and development and systems are now becoming commercially available.

The precise system bit rates used vary depending on the application but are generally in the region of 100 kilobits/s. The adaptive filters are thus required to work up to sample rates of several hundred samples per second an order of magnitude or more up on the adaptive filters used in the speech-band modem applications. Essentially all the techniques described previously for equalization and cancellation may be applied to subscriber loop transmission.. However, the higher sample rate for the adaptive filters makes different demands on implementation technology.

Fortunately, the adaptive filters often need fewer taps, which, combined with the absence of the complication of carrier phase tracking, allows some diverse adaptive filter implementations to be used. The following examples are chosen to illustrate this diversity.

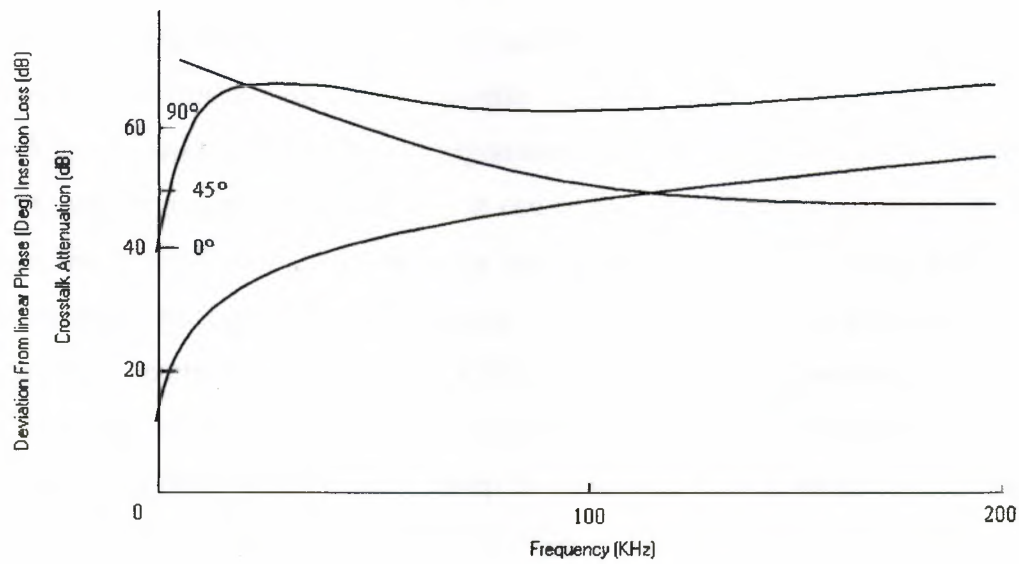


Figure 2.18 Characteristics Of Local Network Cable

2.3.1 Echo Cancellation for WAL2 Transmission

A good example of a self-equalizing line code is the WAL2, so called because its signaling waveforms shown in Figure 2.19(a) resemble the second Walsh function. It has the power spectrum shown in Figure 2.19(b). The smaller spectral lobes above $2/T$ hertz are usually suppressed by a line filter.

To design an echo canceller for this system we need to cancel all the significant energy in the echo signal, which, because it occupies the frequency band 0 to *2/T,* requires an adaptive filter working at a sample rate of *4/T.* The length of echo canceller depends on the degree of echo suppression required, which in turn is a function of the loss experienced by the signal received. Generally, however, the number of taps required is fairly small because the WAL2 code also tends partially to equalize the echo response. In the two examples that follow the number of *T/4* taps are 12 and 24, respectively, for systems with worst-case received signal losses of 30 and 40 dB, respectively. The examples are chosen because they illustrate the interplay between technology and adaptive filter implementation and show how currently available components can be used to make compact and inexpensive adaptive filters.

Analog realization. One particular application of WAL2 transmission called for a system with a bit rate of 80 kilobits/s to operate over cables with up to 30-dB loss at 80 kHz. Experiments showed that a 12-tap echo canceller was sufficient to give adequate echo suppression. To realize the echo canceller cheaply without recourse to LSI circuit technology, the analog circuit shown diagrammatically in Figure 2.20 was developed which was both very compact and inexpensive. It can be seen that the implementation corresponds to structure 2 of Figure 2.16, but with the adaptive filter, apart from the delay line, implemented by analog circuits. The reason for choosing this particular structure was that the use of an interpolation filter, where shown, helps to isolate switching transients and noise from the received signal path. The circuit implementation used extremely simple switching multipliers, which exploit cheap bi-directional analog switches. The automatic gain control (AGC) for the system was included in the tap-update hop to heap overcome direct-current offset problems in the filter implementation and was decoupled from the loop by virtue of its fast adaption time compared to the filter. The filter suppressed echoes sufficiently to give very good performance over lines with a 30-dB loss at 80 kHz and converged on the worst-case line in <70 ms. Figure 2.21(a) shows a photograph of the adaptive filter (excluding the AGC and filter) which consisted of only 12 ICs and consumed *<150* mW. Experiments showed that the limitations of this design were due to nonideal performance of the analog circuit elements, not to the limited number of taps. Therefore, an analog design of this type has a limited field of application.

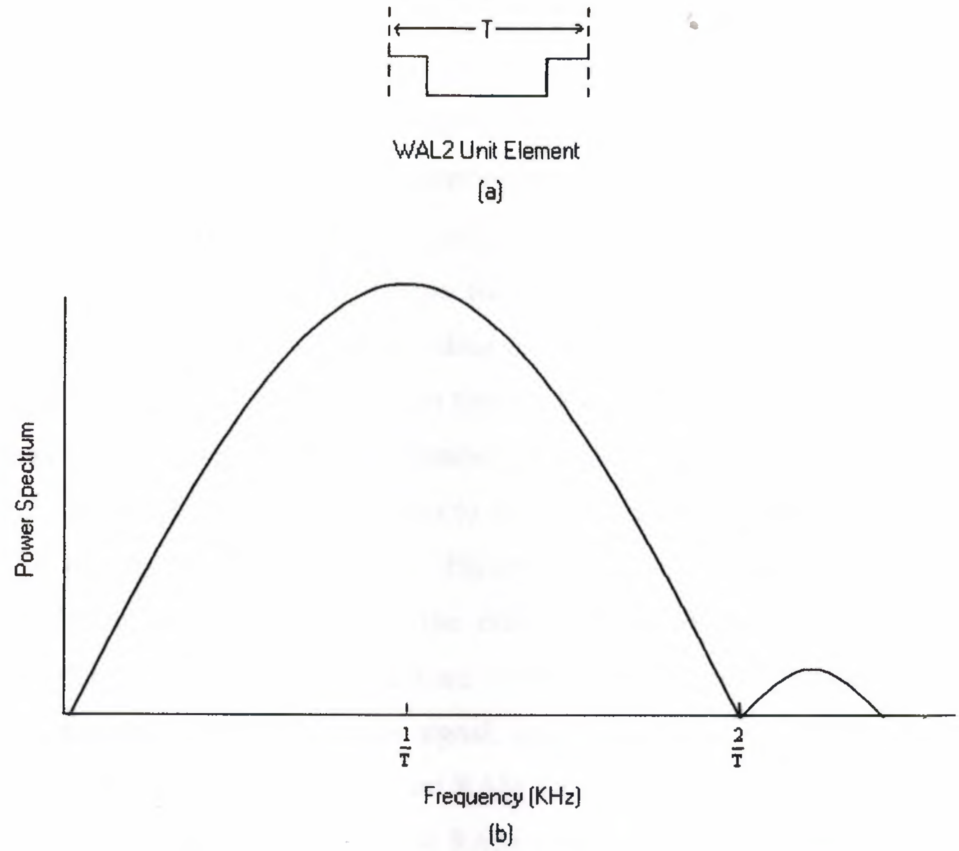


Figure 2.19 WAL2 Line Code And Power Spectrum

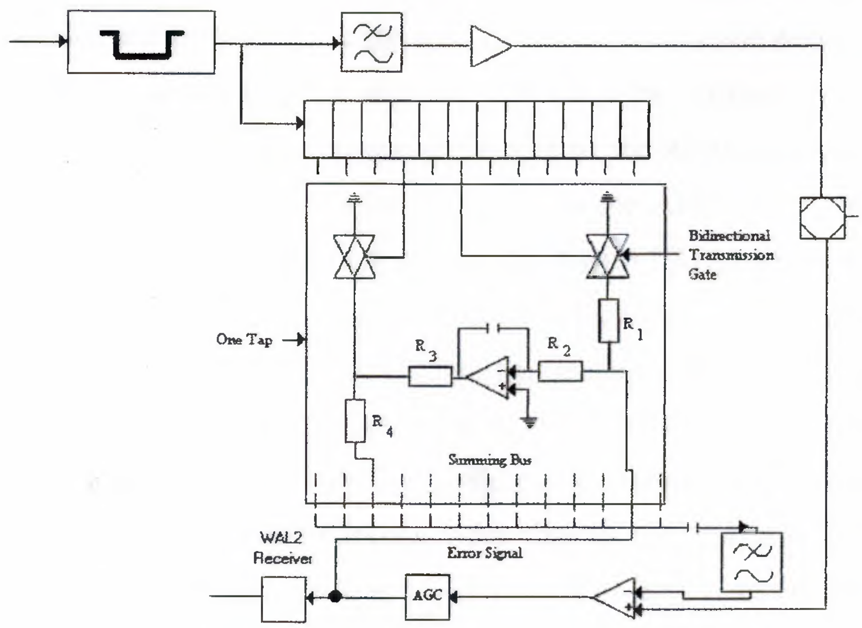
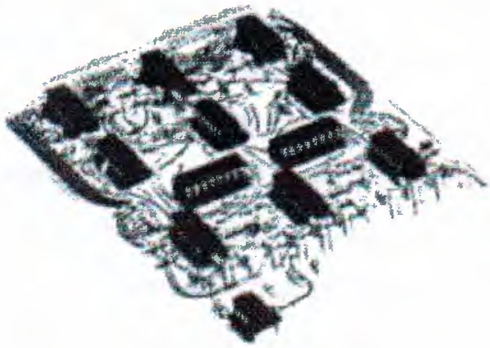


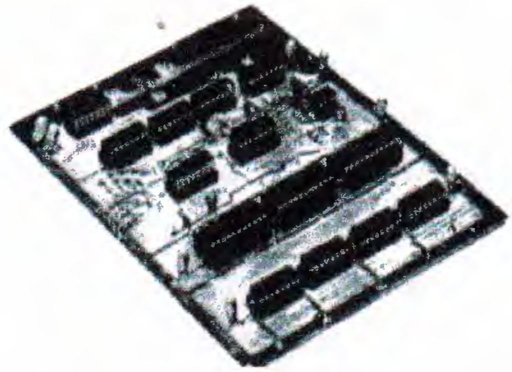
Figure 2.20 Analog Echo Canceller

Digital realization. An 80-kilobit/s WAL2 system required to operate over a 40-dB loss at 80 kHz needed a digital echo canceller with 24 taps. A direct digital realization of the preferred structure 3 in Figure 2.16 is rather complex unless LSI circuit implementation is used. A realization that allows easy implementation is the lookup-table approach. The idea is to look up the required cancellation signal stored in a random access memory (RAM) by using the transmitted data as the address for the RAM. However, it is not necessary to address the RAM with 24 bits corresponding to the 24 taps. As stated in Section 2.2.3, a data-driven echo canceller operating at m times the data signaling rate can be viewed as m independent echo cancelers. Thus the number of address lines for this example is required to be six to give a 6T time span, plus two to divide the RAM into four parts corresponding to the four independent echo cancelers. Figure 2.22 shows a block diagram of the echo canceller. The adaptive algorithm for the echo canceller is very simple: As each echo location is addressed, the RAM contents are converted to an analog signal by the DAC; the signal is subtracted from the received signal, and the error signal, after analog-to-digital conversion, is simply added to the current RAM location word to form a better estimate of the echo, which is then read back into the RAM. At each sampling instant only one location in the RAM is updated, so the convergence rate of this realization is much slower than the normal one where every tap is updated. In local network applications, however, this is not a problem; the stability and invariability of the echo on a given connection means that power-down storage may be used to hold the echo samples when the system is not being used. When it is required to be used., any minor adjustment to the RAM contents would take place very rapidly. A further simplification is to replace the ADC with a sign detector; convergence can still be ensured, although it is slower. Figure 2.21(b) shows a photograph of an experimental version of such an echo canceller, which used CMOS technology and consumed <300 mW. The worst-case convergence time using just a sign update error signal was found to be <500 ms from the condition when the RAM contents all start at zero.

Another property of this adaptive filter realization is that the table lookup places no constraints on the type of echo path response, other than its time duration. Therefore, any nonlinearity in the transmitter, line interface, or echo-canceller DAC is also modeled.



(a)



(b)

Figure 2.21 Echo-Canceller Hardware: (A) Analog Design And (B) Digital Lookup-Table Design

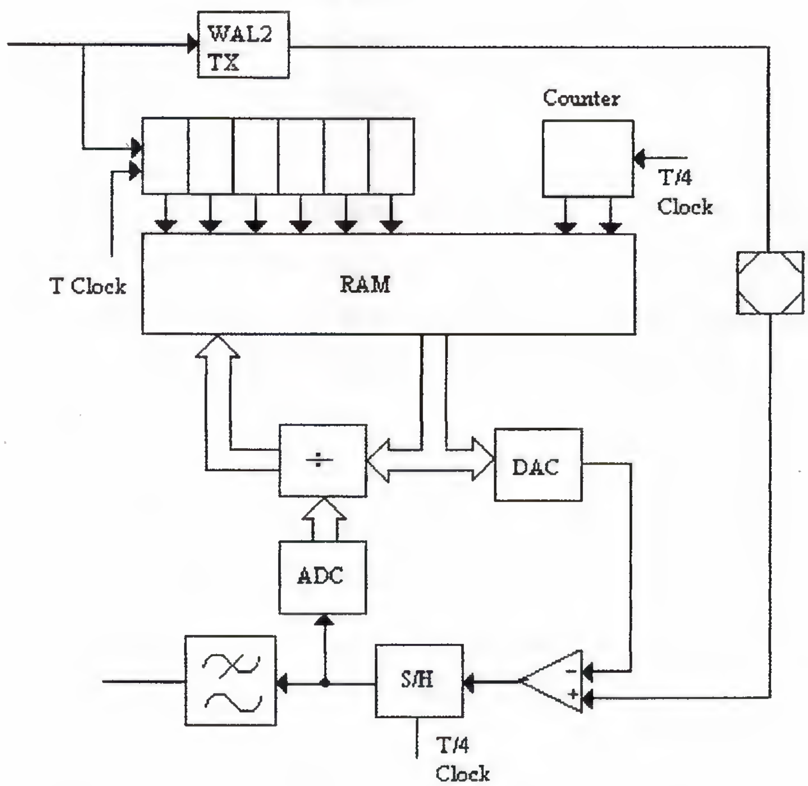


Figure 2.22 Digital Lookup-Table Echo-Canceller

2.3.2 Baseband Transmission

To obtain longer reach, baseband transmission systems are essential. However, these experience much longer echo responses and severe ISI, as shown in Figure 2.23. The straightforward way of dealing with this is to use a receiver with a T/2 echo canceller and an equalizer-the baseband equivalent of the echo-canceling modems of Section 2.2.3. In many local network transmission systems the timing clock used for the customer to local switch direction is locked to the clock derived from the received signal from the local switch. This synchronization of the clocks introduces some problems for the echo canceller but also allows another structure to be used.

The pulse responses of local network cables, as exemplified in Figure 2.23. exhibit two common features: the leading edge rises rapidly to a maximum and the response eventually decays away. On the shorter lengths the maximum of the pulse response is only slightly more than one T interval after the start of the pulse; on longer cables the pulse has risen to over half its maximum after one T. Any of the equalizers described in Section 2.2.2 may be used, but with a suitable choice of sampling phase, the DFB equalizer stands out as being eminently suitable.

The decaying tail of the response suggests that some simple fixed linear equalization may be used to reduce the length of the tail and the size of the DFB equalizer taps an important point for the reduction in error extension effects, that is, the tendency of the DEB equalizer to generate additional errors from a single decision error because an error results in uncanceled ISI.

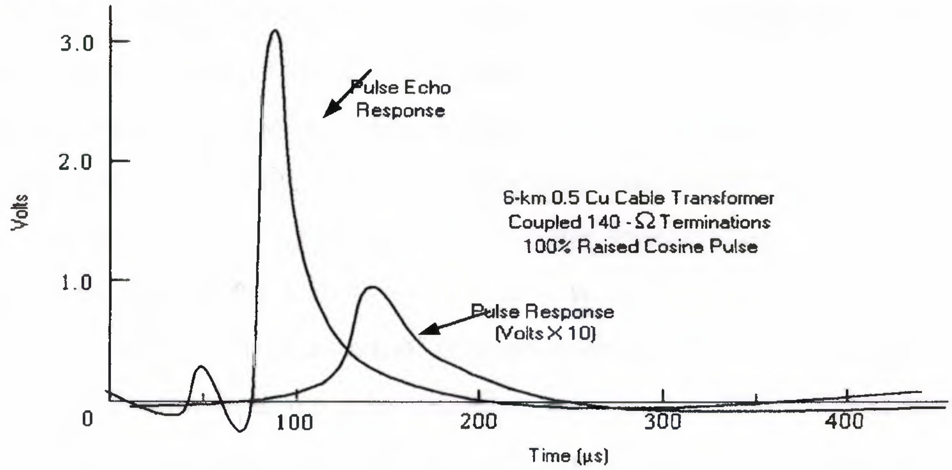


Figure 2.23 Pulse And Echo Responses Of Local Network Cables

Simplified echo canceller for baseband transmission. Although the echo canceller is required to have only T12 tap spacing compared to T/4 in the wider-bandwidth WAL2 system, the longer echo pulse responses require the number of taps to be substantially more (e.g., as great as a 12T or larger time span). The analog realization is not capable of expansion to this number of taps because of implementation problems. For a binary baseband system the lookup4able approach would require a memory of 2 words or more of at least 16 bits accuracy. Although RAM storage is becoming very cheap and compact, this is still a not inconsiderable amount and it does not take many more taps before it becomes prohibitive. In such cases a digitally implemented data-driven linear adaptive filter is the more acceptable alternative. If mild nonlinear distortion of the echo is a problem, there are techniques available for modifying the linear filter to account for them. Structure 3 of Figure 2.16 is the best one, for the reasons given earlier, but one of the main items of expense in the structure is the ADC in the error signal loop. As mentioned in Section 2.2.2, shortening the word length of the error signal can simplify the correlation multiplications; it can also simplify the ADC.

The greatest simplification comes when the error signal is reduced to a sign-only representation. In the absence of a receive signal the echo canceller will converge. With a received signal present, convergence is assured only if the received signal is sometimes smaller than the error signal from the canceller. If it is not, the sign of the received signal controls the updating and the filter will not converge. In a system with synchronous timing for the two directions of transmission there is no guarantee that this will be true. This problem is usually avoided by adding a dither signal to the echo-canceller update loop. The same problem is found with the lookup-table echo canceller of Section 2.3.1, and it can be dealt with in the same way. An echo canceller implemented in this fashion has been integrated for a digital 1 + 1 carrier system using WAL2 transmission.

T-s paced echo cancellation and DFB equalization. DFB equalization may be combined with any of the fractional-tap echo cancellers described in the preceding sections, but the synchronous clocking of many local network transmission systems suggests that provided that a means can be found to recover timing-phase information from T-spaced samples of the line signal, the echo canceller need only have T-spaced taps, with a consequent saving in processing. Fortunately, the shape of the leading edge of the pulse response of local network cables enables such a timing recovery scheme to be implemented. Therefore, the combined T-spaced echo canceller and DFB equalizer shown in Figure 2.24 may be used. To generate an error signal the output of the decision device is fed to an automatic reference control (ARC) to take account of the loss in the cable. Such a configuration is preferred to an AGC because it gives a better joint convergence characteristic for the echo canceller and DEB equalizer. It can be shown that in the absence of decision errors the convergence of the two adaptive filters is identical to the convergence of a single adaptive filter with $N + M + 1$ taps, where N is number of echo canceller taps and M the number of DFB equalizer taps. In this structure the received signal is no longer present on the error signal, so the gain constant can be near its optimum value, giving the fastest possible convergence (and an increase of approximately 3 dB in residual noise). A computer simulation of the structure's convergence showed that for binary signals $N = 20$ and $M = 13$ and, assuming that the decision device generates no errors, convergence to -70 dB takes <6 ms for a 100-kilobaud system (Section 2.3.1). In practice, errors due to the high initial interference from uncanceled echo and ISI slow down convergence, but the use of a suitable training routine can overcome this.

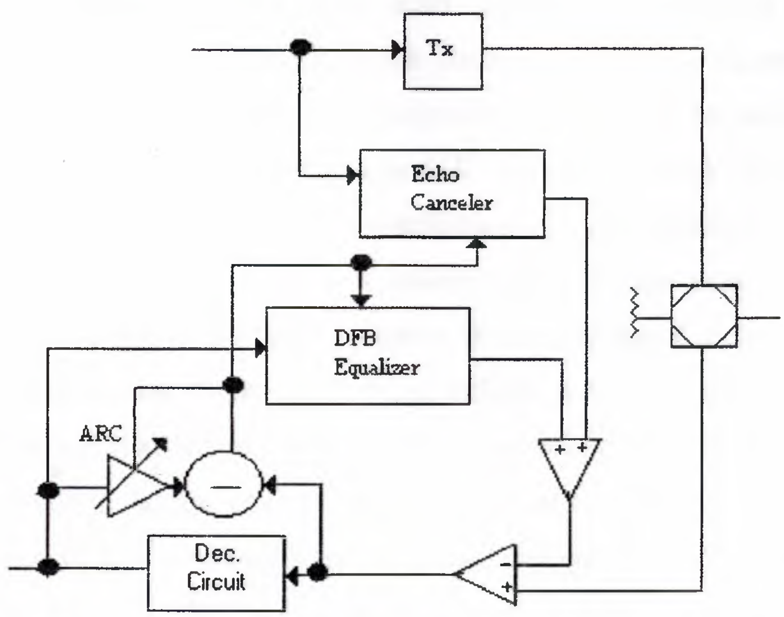


Figure 2.24 Combine Decision Feedback (DFB) Equalizer And Echo – Canceller

2.4 Echo Cancellation For Telephony

Echoes in speech-band circuits affect speech communications in two ways: subjective degradation and circuit stability. Subjective degradation of speech arises in long-haul circuits where echoes delayed by more than about 30 ms can upset the normal conversational process. Stability problems can arise in applications such as loud-speaking telephones and audio teleconferencing, where high-gain amplification can cause circuits to "howl." The traditional solution to both these problems is to use voice-activated attenuation of the return channel in the four-wire part of the circuit (Figure 2.1). However, under certain conditions, voice-activated attenuation can cause subjective degradations such as speech clipping and circuit deadness. Echo cancellation provides a better solution. Two applications are now described to highlight the design features of adaptive filters for speech cancellation.

2.4.1 Network Echo Cancellers

Network echo cancellers are located at a convenient point in the four-wire part of the telephone circuit. Their purpose is to cancel the echoes of speech on the go path, which appear on the return. A circuit requires two echo cancelers, which will probably not be co-located, as the nearer the ends of the four-wire circuit they are, the shorter the delay of the echoes to be canceled. To design the echo canceller knowledge of the echo delay and duration is required with reference to the position of the echo canceller in the network. Figure *2.25* shows some typical echo responses, measured at an international gateway in the United Kingdom, obtained from the U.K. national network. Such echoes can have delays as long as 20 ms and very long oscillatory tails with significant energy up to 30 ms beyond the start of the echo. This is in contrast to the echoes experienced in speech-band data transmission, where the band limitation of the data signal causes the echo tail to die away more quickly (see Figure 2.17). Most echoes, however, have shorter delays and the amount of the tail that has to be canceled is dependent on the degree of echo suppression required. Published designs of echo canceller's aim at canceling echo components up to 16 ms in duration to give a balance return loss enhancement of 25 to 30 dB. Further reduction of the residual echo is achieved by using techniques such as center clipping. With the usual 8000 samples per second used to encode a speech channel, the adaptive filter is required to have 128 taps. If realized *digitally,* the input samples for a filter would need to have an accuracy of 12 bits to account for the wide dynamic range of telephony speech. With the usual long word lengths required for the tap values, a speech echo canceller has to perform a large number of complicated multiply operations and to store long word-length values. Clearly, speech echo cancelers even in this minimal form are more difficult to implement than are their data-driven counterparts. If such problems as time-varying echoes are to be overcome, the degree of complexity can become very high.

Problems of adaption. : In data transmission the input to the echo canceller is well behaved in terms of signal power level and spectrum. The speech input for network echo cancelers is not so obliging; in particular, the following properties create additional serious problems:

1. The speech varies dramatically in power from talker to talker and between different syllables for a single talker. Consequently, if a fixed ıu is used, it must be set very low, to avoid instability. This is often overcome by dynamically normalizing μ to the instantaneous power of the input samples in the filter delay line.
2. Some speech syllables (e.g., voiced sounds) are highly correlated, which can also result in instability if μ is set too high.
3. Conversations are essentially half-duplex in nature, and therefore the input to the adaptive filter disappears from time to time, making tracking of any time-varying echoes difficult.
4. If a conversation were truly half-duplex, the appearance of speech from the far end would not matter because the lack of an input signal would prevent the tap jitter due to a noisy error signal. However, conversations contain periods of double-talking during which speech is present in both directions at once. To prevent misadjustment of the echo canceller, most designs incorporate double-talk detectors, which inhibit the tap updating

Implementations A number of experimental network echo cancelers have been built and as the costs of implementation fall, the commercial exploitation of such designs is beginning to take place. The simplest designs ignore frequency-offset problems, use double-talk detectors to prevent misadjustment, and in some cases use a center clipper to improve the overall performance.

A typical block diagram of such a design is shown in Figure 2.26. To reduce the hardware complexity, pseudobgarithmic coding of the speech samples and the tap values is employed. Designs of this type have been integrated onto a single VLSI $5\text{-}\mu m$ NMOS circuit and, exploiting the fact that a number of echo cancelers may well be co-located, realized in a 12-channel multiplexed MSI TTL hardware structure.

Experimental designs have also been investigated which attempt to deal with the time-varying echoes resulting from frequency offset These are generally much more complicated, using such techniques as delaying the return path while fast tap estimation is performed, and using phase-adaptive structures

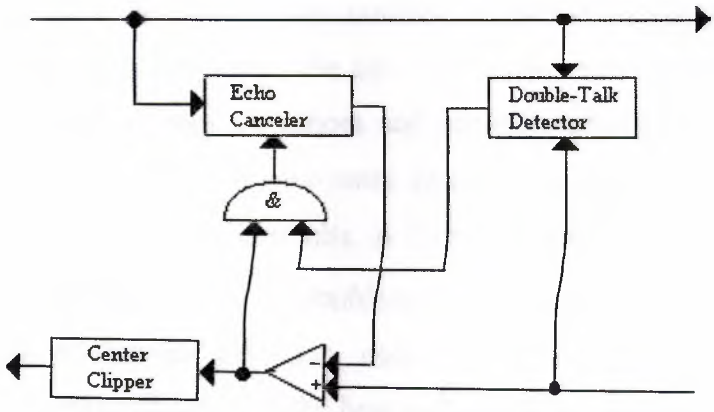


Figure 2.26 Typical Speech Echo – Canceller

2.4.2 Terminal Echo Cancellers

Speech terminals such as loud-speaking telephones (LSTs) and audio tele-conferencing facilities involve the use of amplifiers to drive loudspeakers. Acoustic isolation of the microphone and loudspeaker is difficult and instability can result Echo cancelers can be used to reduce the effect of acoustic coupling, and proposals have been made to apply echo cancelers to both audio teleconferencing, and to LSTs. The echo canceller requirements in both these applications are different from those of network echo cancelers, due partly to their position in the overall connection and partly to the fact that their function is different. The LST application is described next as an illustrative example of a terminal speech echo canceller.

Loud-speaking telephone: The use of adaptive filters in LSTs is illustrated in Figure

2.27. Acoustic coupling between the loudspeaker and the microphone and leakage across the hybrid circuit combine to create feedback resulting in instability. By using adaptive filters to model the acoustic coupling and the hybrid leakage, the amount of feedback can be reduced sufficiently to render the telephone stable when combined with some shallow voice switching.

The acoustic path can be considered as a multireflection medium with impulse response duration of several hundreds of milliseconds. The hybrid leak- age, as in the data transmission application, has two components: the direct leakage across the hybrid and the echoes, which will have the characteristic outlined in Section 2.2 (but with longer decays), obtained from the network. However, the aim of the echo cancelers is to produce stability, so that the longer delayed network echoes and acoustic reflections, which are of smaller magnitude, can be ignored. An experimental design using adaptive filters with a limited time span was found to give good results. A foreground/background method of adaption was used to overcome the problem of double-talking. The method, which is claimed to be more reliable than double-talk detection, uses two echo-path models, foreground and background. The foreground model is the best estimate of the echo path so far and is used to calculate the echo replica for cancellation. The stochastic gradient LMS algorithm until continuously updates the background model; by comparison of the error signals from the foreground and background models, it is deemed to be a better model.

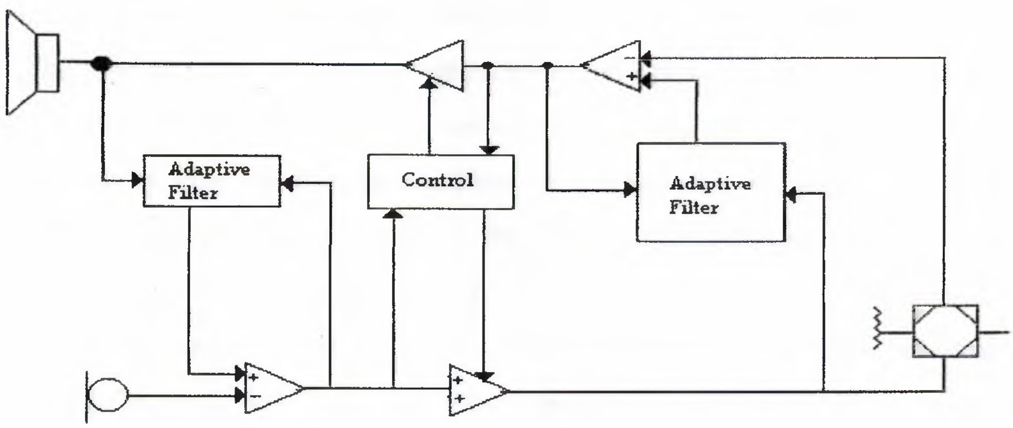


Figure 2.27 Loud – Speaking Telephone (LST) Using Adaptive Filters.

CHAPTER 3

ADAPTIVE FILTERING

3.1 Introduction

In this chapter a class of adaptive filter algorithms is examined that transforms the input signal into the frequency domain before adaptive filtering. The transformations considered here are of a fixed nature in contrast to the data-dependent orthogonalizing transforms. The adaptive algorithms considered here use the gradient descent algorithms.

There are two principal advantages to frequency-domain implementations of adaptive filters. First, the amount of computation required to process a fixed amount of data can be greatly reduced compared with time-domain approaches. This reduction is accomplished by replacing convolution with a multiplication of transforms, as is done in "fast" convolution. Second, the convergence properties of the adaptive process can be improved over simple gradient descent.

In gradient descent algorithms, the weights converge to their optimal solution as a sum of exponentials, each exponential associated with a natural mode of the adaptive process. The time constants of these modes are inversely proportional to the eigenvalues of the input autocorrelation matrix. The mean-square error also decreases as a sum of exponentials whose time constants depend on the eigenvalues. For sufficiently long time-domain FIR filters, the eigenvalues of the input autocorrelation matrix are given approximately by uniformly spaced samples of the input power spectrum. A heuristic interpretation of this result is that modes associated with areas of the spectrum having little power converge more slowly than those modes associated with frequencies having greater power. A large variation in the input power spectrum with frequency leads to highly disparate eigenvalues and therefore highly disparate time constants, some of which may be very long. Frequency-domain techniques, although based on steepest descent, can easily be modified to allow more uniform convergence of the modes of the adaptive process, thus improving convergence rate of the slower modes.

Use of the frequency domain results in block processing, in which a block of input data is processed simultaneously, producing a block of output. The nature of block processing requires that the filter coefficients be held fixed during the block. This process is in contrast to usual methods of time-domain adaptive processing, in which the filter coefficients may change at the input sampling rate. Although filter coefficients are updated less often-using frequency-domain approaches, they can be adjusted at each update with greater precision, since the gradient can be estimated using an entire block of data. The result is that adaptation can proceed just as rapidly and accurately in the frequency domain as it can in the time domain, with one exception. When the input autocorrelation matrix has highly disparate eigenvalues, stability considerations set an upper limit on the rate of adaptation that can be much slower than the corresponding limit for time-domain processing. Modified gradient techniques that effectively reduce the eigenvalue disparity can alleviate this problem.

In the following, uppercase symbols will denote frequency-domain variables, lowercase symbols stand for time-domain variables, and boldface will denote vectors or matrices. An asterisk will denote complex conjugate transpose. Define F to be a symmetric $N \times N$ matrix whose ith, kth element is $F_{ik} = \exp(-j2\pi ik/N)$, where j is the square root of -1. When F operates on a column vector of length N the result is a column vector containing the DFT of the original vector. Similarly F^{-1}, is the inverse DFT operator. It can be shown that $F^* = NF^{-1}$ and $F/\sqrt{N}$ is a unitary transformation. If c is a circulant matrix then FcF^{-1} is a diagonal matrix whose elements are the DFT of the first column of the circulant matrix.

3.2 Frequency-Domain Adaptive Filter Based On Circular Convolution

One of the simplest frequency-domain adaptive filters is that shown in Figure 3.1. The input signal *x(n)* and desired response *d(n)* are accumulated in buffer memories to form N-point data blocks. They are then transformed by N-point FFTs. Each of the FFT outputs comprises a set of N complex numbers. The desired response transform values are subtracted from the input transform values at corresponding frequencies to form N complex error signals There are N complex weights, one corresponding to each spectral bin. Each weight is independently updated once for each data block. The weighted outputs are fed to an inverse FFT operator to produce the output signal *y(n)*.

The complex LMS algorithm is used to update each weight. The *i*th complex weight for the ith frequency bin is updated according to

$$Hi(k+1) = H(k) + \mu Ei(k)Xi^*(k) \tag{3.1}$$

where μ is a constant that determines rate of convergence and stability of the adaptive process. For statistically stationary inputs, the weight-update equation (3.1) eventually minimizes the mean-square error at the *i*th frequency bin, provided that μ is chosen sufficiently small.

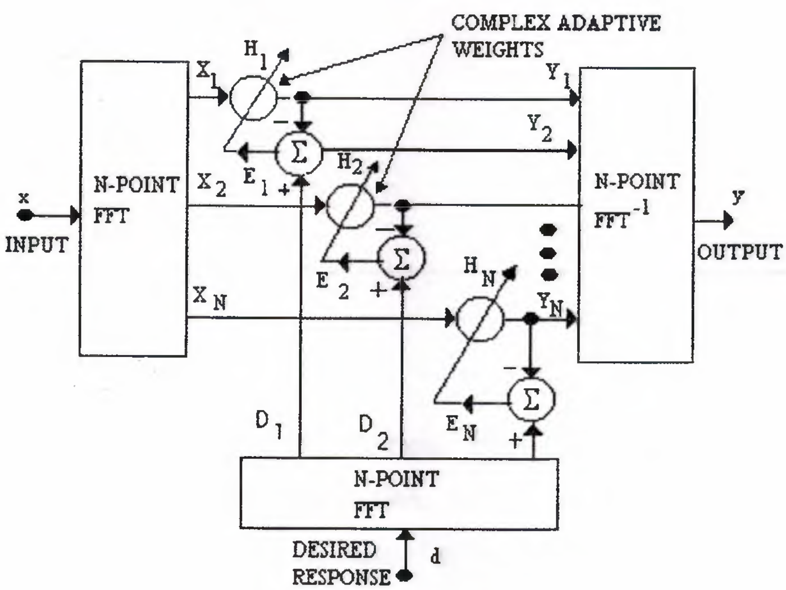


Figure 3.1 frequency-domain adaptive filter performing circular convolution.

A substantial reduction in computation is obtained with this frequency-domain adaptive filter as compared with conventional time-domain adaptive filtering. This fact can be demonstrated by examining the number of multiply operations required to process a fixed amount of data. To produce *N* output data points with an N-tap time-domain LMS adaptive filter requires $2N^2$ real multiplies. To produce the same amount of output with this frequency-domain filter requires three N-point FFTs and 2N complex multiplies for the complex weighting and updating. For real input data, however, all transforms are symmetric, so that approximately half the weights can be discarded. Furthermore, for real data, an N-point FFT can be realized with an N/2-point FFT and N/2 complex multiplies

An N/2-point FFT takes approximately $(N/4)\log_2(N/2) - N/2$ complex multiplies for a radix-2 transform. Assuming four real multiplies per complex multiply yields $3N\log_2(N/2) + 4N$ real multiplies for the frequency-domain filter, compared with $2N^2$ multiplies for the time-domain filter. For large filters, the computational savings produced by the frequency-domain filter is substantial, as shown in the following listing.

N	$\dfrac{\text{Frequency} - \text{Domain Real Multiplies}}{\text{LMS Real Multiplies}}$
16	0.41
32	0.25
64	0.15
256	0.049
1024	0.015

Unfortunately, the frequency-domain filter of Figure 3.1 produces circular convolution, rather than linear convolution, of the input signal with the adaptive filter impulse response. The use of circular instead of linear convolution transforms a linear time-invariant filter into a periodic time-varying filter, whose output is periodically non-stationary for a stationary input. While the method of Figure 3.1 has found application in frequency-domain signal detection, its circular convolution property makes it less useful for general filtering applications. The use of circular rather than linear convolution generally tends to shorten the effective impulse response length in order to reduce the effect of "wraparound" error due to circular convolution. The effect of circular convolution can be made small if the filter length (FFT size) can be chosen much larger than the effective nonzero length of the optimal impulse response for linear convolution. Although this substantially reduces the computational efficiency of the approach, its simplicity, coupled with the fact that the various weights are adapted independently, makes it attractive to consider.

To analyze this algorithm, define the frequency-domain weight vector for the *k*th block by

$$\mathrm{H}^{T}(K)=[H_1(K)H_2(K)...H_N(K)] \tag{3.2}$$

and the diagonal matrix of input FFT coefficients by

$$X(k)=\begin{bmatrix} X_1(k) & & 0 \\ & X_2(k) & \\ & \ddots & \\ 0 & & X_N(k) \end{bmatrix} \tag{3.3}$$

Similarly, let Y(k), D(k), and E(k) be vectors containing the frequency-domain output, desired response, and error for the *k*th block. Note that

$$Y(k)=X(k)H(k) \tag{3.4}$$

$$E(k)=D(k)-Y(k) \tag{3.5}$$

The frequency-domain weight update equation can be expressed as:

$$\begin{aligned} H(k+1) &= H(K)+\mu \mathrm{X}^{*}(k)E(k) \\ &= H(k)+\mu\{\mathrm{X}^{*}(k)D(k)-X^{*}(k)X(k)H(k)\} \end{aligned} \tag{3.6}$$

It is useful to consider the equivalent time-domain operations implied by (3.6). Equation (3.6) can be transformed into the time domain to yield

$$h(k+1) = h(k) + \mu\left\{x^T(k)d(k) - x^T(k)x(k)h(k)\right\} \tag{3.7}$$

where

$$h(k) = F^{-1}H(k) \tag{3.8}$$

$$d(k) = F^{-1}D(k) \tag{3.9}$$

and x(k) is a circulant matrix given by

$$x(k) = F^{-1}X(k)F \tag{3.10}$$

The first column of x(k) is the input vector x(k), since it is the inverse DFT of the diagonal elements of X(k). Therefore, the circulant matrix x(k) is given by

$$x(k) = \begin{bmatrix} x(k) & x(k+N-1) & . & . & . & x(k+1) \\ x(k+1) & x(k) & . & . & . & x(k+2) \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ x(k+N-1) & x(k+N-2) & . & . & . & x(k) \end{bmatrix} \tag{3.11}$$

Denoting the *i*th row of x(k) by $x_i^T(k)$ (3.7) becomes:

$$h(k+1)=h(k)+\mu\sum_{i=1}^{N}\{d_i(k)x_i(k)-y_i(k)x_i(k)\} \tag{3.12}$$

where $y_i(k)$ is the ith element of the output vector

$$y(k)=x(k)h(k) \tag{3.13}$$

The vector *y(k)* contains the elements of the *k*th output block of the filter. The elements of *y(k)* are obtained by circularly convolving the impulse response h(k) with rotated versions of the input vector *x(k)*.

The equivalent weight-update equation in the time domain can be written

$$h(k+1)=h(k)+\mu\sum_{i=1}^{N}e_i(k)x_i(k) \tag{3.14}$$

where $e_i(k) = d_i(k)-y_i(k)$. Note that (3.14) differs from the usual LMS algorithm in that, although adaptation is performed only once per block, gradient estimates are summed over an entire block of data before being used to update the weights.

Optimum weight vector.

The optimum solution for H, which minimizes the mean-square error between y(k) and d(k), can be determined (assuming that *d* and x are stationary) in the following manner. It is sufficient to minimize

$$NE[(\mathrm{d}(k)-\mathrm{y}(k))^*(\mathrm{d}(k)-\mathrm{y}(k))]=E[(\mathrm{D}(k)-\mathrm{y}(k))^*(\mathrm{D}(k)-\mathrm{Y}(k))] \tag{3.15}$$

$$=E[(\mathrm{D}^*(k)\mathrm{D}(k)]-\mathrm{R}^*{}_{xd}\mathrm{H}-\mathrm{H}^*\mathrm{R}_{xd}+\mathrm{H}^*\mathrm{R}_{xx}\mathrm{H}$$

where $$\mathrm{R}_{xd}=E[\mathrm{X}^*(k)\mathrm{D}(k)] \tag{3.16}$$

and $$\mathrm{R}_{xx}=E[\mathrm{X}^*(k)\mathrm{X}(k)] \tag{3.17}$$

Note that R_{xx} is diagonal, with its ith diagonal element given by $E[X^*_i(k)X_i(k)]$. The ith element of R_{xd} is $E[X_i^*(k)D_i(k)]$. Taking the gradient of (3.15) with respect to H and setting it to zero yields the optimum frequency-domain weights:

$$H_{opt} = R^{-1}_{xx} R_{xd} \tag{3.18}$$

Taking the inverse DFT of (3.18) yields the optimum time-domain weights for a circularly convolving filter:

$$h_{opt} = r^{-1}_{xx} r_{xd} \tag{3.19}$$

where

$$r_{xx} = F^{-1} R_{xx} F \tag{3.20}$$

and

$$r_{xd} = F^{-1} R_{xd} F \tag{3.21}$$

The matrix r_{xx} is circulant, since R_{xx} is diagonal. The first row of r_{xx} is given by lags zero through $N - 1$ of the circular autocorrelation function of the input x. The circular autocorrelation function at lag $i, \Phi_c(i)$, can be expressed in terms of the usual linear autoeorrelatiofl function $\Phi(i)$ by

$$\Phi_c(i) = \frac{N-i}{N}\Phi(i) + \frac{i}{N}\Phi(i-N) \tag{3.22}$$

A similar expression is obtained for the circular cross-correlation between d and x, which make up the elements of the vector r_{xd}.

Convergence properties. Taking expected values of (3.6) and assuming stationary inputs, we have:

$$E[H(k+1)] = E[H(k)] + \mu\{R_{xd} - R_{xd}E[H(k)]\} \tag{3.23}$$

where the usual assumption of uncorrelatedness between X(k) and H(k) has been made. Although this assumption is generally not strictly true, it is commonly made in adaptive filter analysis because of its simplicity, and has led to a number of useful results. We will make this simplifying assumption here in order to gain insight into the adaptive process. Equation (3.23) implements the method of steepest descent and has been well studied. The input autocorrelation matrix R_{xx} whose eigenvalues determine the stability and convergence rate of the adaptive process, is diagonal in this case. The eigenvalues of this matrix are therefore given by its diagonal elements, which are the powers of the DFT bins. If μ is chosen small enough, the expected value of the weight vector will converge to

$$\lim_{k\to\infty} E[\mathrm{H}(k)] = \mathrm{R}^{-1}{}_{xx}\mathrm{R}_{xd} = \mathrm{H}_{\mathrm{opt}} \tag{3.24}$$

Thus the mean weight vector converges to the optimum weight vector. The condition for stability of the algorithm is

$$\mu \langle \frac{2}{\lambda_{\max}} \tag{3.25}$$

where $\lambda_{\max}$ is the maximum eigenvalue of R_{xx}. The weights converge independently of each other since R_{xx} is diagonal. The time constant for the convergence of the *p*th weight is given by

$$\tau_p = \frac{1}{\mu\lambda_p} \quad \text{blocks} \tag{3.26 a}$$

$$= \frac{\mathrm{N}}{\mu\lambda_p} \quad \text{samples} \tag{3.26 b}$$

where λ_p is the *p*th eigenvalue of R_{xx} Note that since is diagonal, its eigenvalues are simply its diagonal elements, and these are the powers in the FFT bins.

Misadjustment Since the gradient is estimated using a finite amount of data, it has some error, resulting in random fluctuations of the filter coefficients about their optimal value. The result is a mean-square error greater than the minimum mean-square error. This excess mean-square error, normalized by the minimum mean-square error, is defined as the "misadjustment".

If it is assumed that the $\mathrm{x}_i(k)$ are uncorrelated and that $\mathrm{e}_i(k)$ and $\mathrm{x}_i(k)$ are zero-mean Gaussian, an expression for the misadjustment of the adaptive process can be derived. Although these assumptions may not be strictly true, they are commonly made in order to simplify the analysis and will allow a misadjustment formula to be obtained that can be compared with that of conventional time-domain filters.

3.3 Algorithms For General Adaptive Filtering

Adaptive filters that allow linear convolution of the filter input and impulse response are more generally useful in filtering applications than are those that perform only circular convolution. Two frequency-domain adaptive filters are described in this section that allow linear convolution. One adaptive filter, denoted in the literature as block LMS or fast LMS, performs strictly linear convolution. It permits an efficient frequency-domain implementation while maintaining performance equivalent to that of the widely used LMS adaptive filter. The other adaptive filter considered in this section, the unconstrained frequency-domain LMS adaptive filter, allows either linear or circular convolution, whichever best minimizes the mean-square error.

3.3.1 Fast LMS Adaptive Filter

The block LMS adaptive filter and the fast LMS adaptive filter are essentially identical frequency-domain implementations of the time-domain block LMS algorithm. In this algorithm, the data are grouped into N-point blocks, with the filter weights held constant over each block. During the *k*th block, the adaptive filter equations are:

$$\mathrm{h}(k+1) = \mathrm{h}(k) + \mu\sum_{i=o}^{N-1} e(kN+i)x(kN+i) \qquad (3.27)$$

$$= \mathrm{h}(k) + \mu\nabla(k)$$

and

$$y(kN+i) = \mathrm{h}^{\mathrm{T}}(k)x(kN+i) \qquad (3.28)$$

where $I = 0, 1, 2, \ldots$

and h(k) is a vector containing the filter weights during the *k*th block:

$$h^T(k) = [h_0(k) h_1(k) .. h_{N-1}(k)] \tag{3.29}$$

and x(n) contains the *N* most recent filter input samples at time *n:*

$$x^T(n) = [x(n)x(n-1)...x(n-N+1)] \tag{3.30}$$

The elements of x(n) can be viewed as the outputs of an N-tap tapped delay line. The error *e(n)* is the difference between the desired response *d(n)* and the filter output,

$$e(n) = d(n) - y(n) \tag{3.31}$$

The block LMS algorithm has properties that are identical, except for stability, to the conventional LMS adaptive filter, in which the filter weights are updated at the sampling rate. These properties will be discussed later.

The block LMS algorithm can be implemented in the frequency domain using the "overlap-save" method, resulting in a substantial reduction in computation over time-domain processing. A frequency-domain implementation employing the "overlap-add" method is also possible but results in more computations than are needed in the overlap-save method. Although it is possible to implement the filter with any amount of overlap, the case of 500/o overlap (block size equal to number of weights) is the most efficient and will be presented here. The filter output equation (3.28) is a convolution between the filter input and impulse response, and can be computed efficiently using the overlap-save method. According to this method, the weights must be padded with *N* zeros, and 2N-point FFTs must be used. Let H(k) be a vector of length 2N whose elements are the FFT coefficients of the zero-padded, time-domain weight vector:

$$H^T(k) = FFT[h^T(k) \quad 0 \ ... \ 0] \tag{3.32}$$

H(k) is the frequency-domain weight vector. Let X(*k*) be a diagonal matrix whose elements are the 2N-point transform of the (*k* — 1)th and *k*th input blocks:

$$x(k) = \mathrm{diag}\{FFT[\underbrace{x(kN-N)...x(kN-1)}_{(k-1)\text{th block}}\underbrace{x(kN)..x(kN+N-1)}_{\text{kth block}}]\} \tag{3.33}$$

The convolution in (3.28) is realized by

$$y(k) = [y(kN)...y(kN+N-1)]^T \tag{3.34}$$

$$= \text{last } N \text{ terms of } FFT^{-1}\{X(k)H(k)\}$$

Equation (3.34) gives the filter output values for the *k*th block. Note that an N-weight transversal filter in the time domain requires a 2N-weight filter in the frequency domain.

To implement the weight-vector-update equation (3.27) in the frequency domain, notice that the jth element of $\nabla(k)$ can be rewritten as:

$$\nabla_j(k) = \sum_{i=0}^{N-1} \mathrm{e}(kN+i)x(kN+i-j), \qquad \text{where j = 0, 1, ..., N-1}$$

so that the elements of $\nabla(k)$ are given by the cross-correlation of the error sequence with the filter input. $\nabla(k)$ can be computed using FFTs if we first compute the transform E(k) of the error sequence preceded by N zeros:

$$E(k) = \mathrm{FFT}[\underbrace{0 \ldots 0}_{\text{N zeroes}}\ \underbrace{\mathrm{d}(kN) \ldots \mathrm{d}(kN+N-1) - \mathrm{y}(kN+N-1)}_{k\text{th error block}}]^{\mathrm{T}} \tag{3.35}$$

and then compute

$$\nabla(k) = \text{first N terms of } \mathrm{FFT}^{-1}\{\mathrm{X}^*(k)\mathrm{E}(k)\} \tag{3.36}$$

Finally, the frequency-domain weight-vector-update equation is:

$$\mathrm{H}(k+1) = \mathrm{H}(k) + \mu\mathrm{FFT}\begin{bmatrix} \nabla(k) \\ 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \end{bmatrix} \text{N zeros} \tag{3.37}$$

If the last N values of the inverse transform of the initial weight vector H(0) are forced to zero, (3.37) is an exact implementation of (3.27) in the frequency domain.

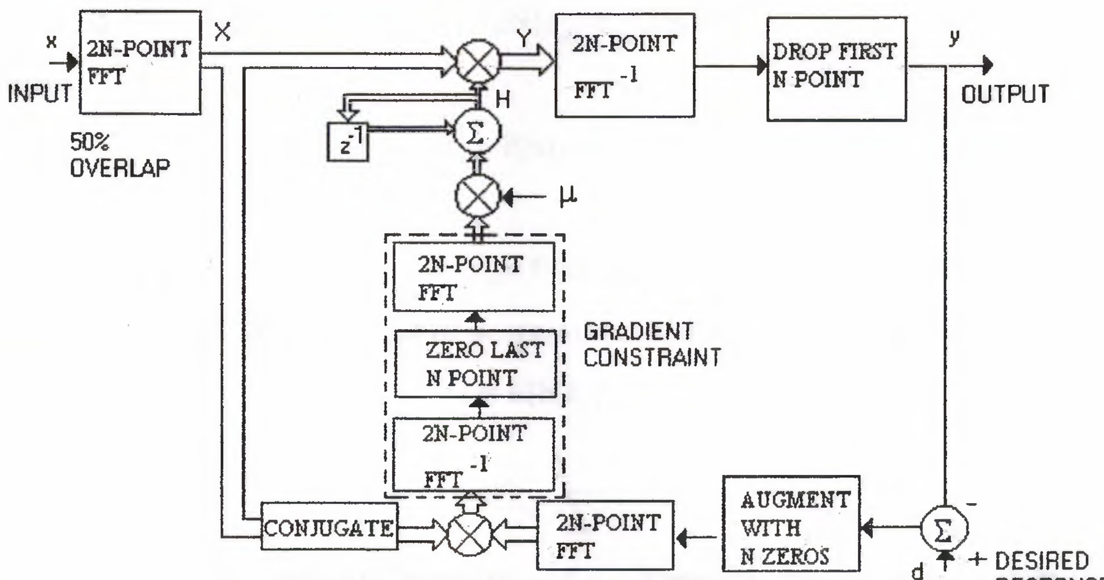
Equations (6.32) to (3.37) define the fast LMS (FLMS) adaptive filter. A block diagram of the filter is shown in Figure 3.2. Double lines in Figure 3.2 denote parallel flow of frequency-domain data.

For each N-point block, the FLMS filter requires five 2N-point FFTs and two 2N-point complex multiplies. For real input data, all transforms are symmetric and require computation of only the firist N + 1 terms. Furthermore, for real data, a 2N-point FFT can be realized with an N-point FFT and N complex multiplies. An N-point radix-2 FFT requires approximately $(N/2)\log_2(N) - N$ complex multiplies [Singleton] (a radix-4 FFT requires somewhat less computation). Therefore, the number of complex multiplies per block is $(5N/2)\log_2(N) - N$ for the five FFTs and approximately 2N for the complex weighting and updating. To produce N output points with the conventional LMS adaptive filter requires $2N^2$ real multiplies. Assuming that one complex multiply is equivalent to four real multiplies yields the following ratio:

$$\frac{\text{FLMS real multiplies}}{\text{LMS real multiplies}} = \frac{5(\log_2 N)+4}{N} \tag{3.38}$$

This ratio is computed for several values of N in the following tabulation:

N	$\frac{\text{FLMS Real Multiplies}}{\text{LMS Real multiplies}}$
16	1.5
32	0.91
64	0.53
256	0.17
1024	0.053



Fast least-mean-square (FLMS) adaptive filter

Figure 3.2 Fast Least Mean Square (FLMS)

For larger filters the computational savings gained by using the ELMS algorithm is substantial, even though five FFTs are required.

The ELMS algorithm can be written in the following matrix format, which will be useful in the sequel:

$$\mathrm{H}(k+1)=\mathrm{H}(k)+\mu\mathrm{F}\begin{bmatrix}\mathrm{I}_{\mathrm{N}} & 0\\ 0 & 0\end{bmatrix}\mathrm{F}^{-1}\mathrm{X}^{*}(k)\mathrm{E}(k) \tag{3.39}$$

$$\mathrm{E}(k)=\mathrm{F}\begin{bmatrix}0\\ \mathrm{I}_{\mathrm{N}}\end{bmatrix}(\mathrm{d}(k)-\mathrm{y}(k)) \tag{3.40}$$

$$\mathrm{y}(k)=[0 \quad \mathrm{I}_{\mathrm{N}}]\mathrm{F}^{-1}\mathrm{X}(k)\mathrm{H}(k) \tag{3.41}$$

where F is a 2N x 2N DFT matrix, and I_{N} is an N x N identity matrix.

Convergence properties. Since the ELMS algorithm is an exact implementation of the block LMS algorithm, it is sufficient to study the convergence properties of the latter. Using (3.27) and (3.28), a recursion for the expected value of the weight vector can be obtained under the assumption that d(n) and x(n) are stationary and that the x(n) are uncorrelated in time:

$$\mathrm{E}[\mathrm{h}(k+1)]=\mathrm{E}[\mathrm{h}(k)]+\mu\mathrm{N}\{\mathrm{r}_{xd}-\mathrm{r}_{xx}\mathrm{E}[\mathrm{h}(k)]\} \tag{3.42}$$

where

$$\mathrm{r}_{xx}=\mathrm{E}[\mathrm{x}(n)\mathrm{x}^{\mathrm{T}}(n)] \tag{3.43}$$

and

$$\mathrm{r}_{xd}=\mathrm{E}[\mathrm{d}(n)\mathrm{x}(n)] \tag{3.44}$$

Applying the results for the method of steepest descent it can be shown that provided that

$$\lim_{k\to\infty}\mathrm{E}[\mathrm{h}(k)]=\mathrm{r}^{-1}{}_{xx}\mathrm{r}_{xd} \tag{3.45}$$

$$\mu\langle\frac{2}{\mathrm{N}\,\lambda_{\max}} \tag{3.46}$$

where $\lambda_{\max}$ max is the maximum eigenvalue of r_{xx} Therefore, the converged weight vector is identical to that obtained with the conventional LMS algorithm. However, the conditional for stability of the conventional LMS algorithm does not have the N in the denominator of (3.46), allowing faster adaptation. Therefore, the stability condition for the block LMS algorithm is more restrictive than that for the conventional LMS algorithm. This may be a problem when the eigenvalues of r_{xx} are highly disparate.

The time constants for the convergence of the N modes of the adaptive process can be shown to be

$$\tau_i = \frac{1}{\mu\lambda_i N} \quad \text{blocks} \qquad (3.47\text{ a})$$

$$= \frac{1}{\mu\lambda_i} \quad \text{samples} \qquad (3.47\text{ b})$$

These time constants are identical to that of the conventional LMS algorithm. Assuming further that e(n) and x(n) are zero-mean Gaussian, an expression for the misadjustment
the adaptive process can be found to be:

$$M = \frac{\mu}{2}\operatorname{tr}\{r_{xx}\} \qquad (3.38\text{ a})$$

$$= \frac{\mu NP}{2} \qquad (3.38\text{ b})$$

which is identical to that for the conventional LMS algorithm. Therefore, to achieve the same steady-state error in either algorithm μ should be chosen the same, resulting in identical convergence rates for either algorithm. This may not always be possible because of the more restrictive stability conditions for the block LMS algorithm. The stability requirement limits the misadjustment of the block LMS algorithm to satisfy:

$$M \langle \frac{\lambda_{avg}}{\lambda_{max}} \qquad (3.39)$$

Since the desired misadjustment would normally be under 0.1, the restriction (3.39) is only a problem for the case of highly disparate eigenvalues. The restriction becomes less of a problem as the data overlap between successive FFTs is increased, although these results in less efficient processing since fewer output samples are obtained from each iteration.

3.3.2 Unconstrained Frequency-Domain LMS Adaptive Filter

The ELMS algorithm of the preceding section required five FFTs per processed block; two of them were needed to impose a time-domain constraint in which the last half of the time-domain weights were forced to zero. This was necessary in order to implement strictly linear convolution between the filter input and impulse response. In the unconstrained frequency-domain LMS (UFLMS) adaptive filter, this constraint is removed, which produces a simpler adaptive filter that can implement either linear or circular convolution. which ever best minimizes the mean-square error. Allowing the filter the freedom to implement circular convolution would be acceptable in a number of applications-those for which we do not care how the filter input is used to minimize the mean-square error or multichannel signal enhancing. In fact, it is even possible that allowing some circular convolution may reduce the mean-square error. However, a problem arises if we attempt to use UFLMS for adaptive prediction or line enhancement. In these applications, past values of an input signal are used by an adaptive filter to predict the signal's current value. Obviously, the adaptive filter should not be able to use the signal's current value to predict it. However, when block processing is used, as in the UFLMS algorithm, the adaptive filter has access, not only to past input values, but also to many current and future values. Unless a constraint is placed on the weights, the adaptive filter can use the current and future values to minimize the mean-square error. This is disastrous for line enhancing since it produces a filter output that is nearly identical to the filter input, with little enhancement of spectral lines. The foregoing comments apply only when predicting ahead some number of samples that is less than the FFT size; for longer prediction intervals there is no problem, since the data in corresponding desired response and filter input blocks are disjoint.

The block diagram of the UFLMS adaptive filter is identical to that in Figure 3.2 except that the constraint shown in the dashed line is removed, which eliminates two of the FFTs used in the FLMS adaptive filter. The ratio of UFLMS real multiplies to conventional LMS real multiplies is therefore:

$$\frac{\text{UFLMS real multiplies}}{\text{LMS real multiplies}} = \frac{3(\log_2 N) + 4}{N} \tag{3.50}$$

This ratio is tabulated below for several values of N:

N	$\dfrac{\text{UFLMS Real multiplies}}{\text{LMS Real Multiplies}}$
16	1.0
32	0.59
64	0.34
256	0.11
1024	0.033

By removing the zero-forcing constraint in the FLMS algorithm [(3.39) to (6.41)] the UFLMS algorithm can be expressed as:

$$\mathrm{H}(k+1) = \mathrm{H}(k) + \mu \mathrm{X}^*(k)\mathrm{E}(k) \tag{3.51}$$

$$\mathrm{E}(k) = \mathrm{F}\begin{bmatrix} 0 \\ \mathrm{I}_\mathrm{N} \end{bmatrix}(\mathrm{d}(k) - \mathrm{y}(k)) \tag{3.52}$$

$$\mathrm{y}(k) = [0 \quad \mathrm{I}_\mathrm{N}]\mathrm{F}^{-1}\mathrm{X}(k)\mathrm{H}(k) \tag{3.53}$$

Optimum weight vector. The optimum weights that minimize the mean-square error between the filter output and desired response are now computed. The mean-square error ζ expressed as a function of the weight vector H is:

$$\zeta = \mathrm{E}[(\mathrm{d}(k) - \mathrm{y}(k))^*(\mathrm{d}(k) - \mathrm{y}(k))] \tag{3.54}$$

$$= \frac{1}{2N}\mathrm{E}\{(\mathrm{d}(k) - \mathrm{y}(k))^*[0 \quad \mathrm{I}_\mathrm{N}]\mathrm{F}^*\mathrm{F}\begin{bmatrix} 0 \\ \mathrm{I}_\mathrm{N} \end{bmatrix}(\mathrm{d}(k) - \mathrm{y}(k))\}$$

where we have used the fact that $\mathrm{F}^*\mathrm{F} = 2\mathrm{NI}_{2\mathrm{N}}$. Therefore,

$$2\mathrm{N}\zeta = \mathrm{E}\{[\mathrm{D}(k) - \mathrm{FwF}^{-1}\mathrm{X}(k)\mathrm{H}]^*[\mathrm{D}(k) - \mathrm{FwF}^{-1}\mathrm{X}(k)\mathrm{H}]\} \tag{3.55}$$

where D*(k)* is the 2N-point DFT of d*(k)* preceded by *N* zeros:

$$\mathrm{D}(k) = \mathrm{F}\begin{bmatrix} 0 \\ \mathrm{d}(k) \end{bmatrix} \tag{3.56}$$

and w is a 2N x 2N windowing matrix

$$\mathrm{w} = \begin{bmatrix} 0 & 0 \\ 0 & \mathrm{I}_\mathrm{N} \end{bmatrix}$$

Expanding (3.55) results in:

$$2N\zeta = E[D^*(k)D(k)] - E[D^*(k)WX(k)]H - H^*E[X^*(k)W^*D(k)] + H^*E[X^*(k)WX(k)]H$$

(3.57)

where

$$W = FwF^{-1} \quad (3.58)$$

Setting the gradient of (3.57) with respect to H equal to zero yields the optimum weight vector,

$$H_{opt} = R^{-1}{}_{xx} R_{xd} \quad (3.59)$$

where

$$R_{xx} = E[X^*(k)WX(k)] \quad (3.60)$$

and

$$R_{xd} = E[D(k)W^*X^*(k)] \quad (3.61)$$

The equivalent time-domain weight vector can be found to be

$$h_{opt} = r_{xx}{}^{-1} r_{xd} \quad (3.62)$$

where

$$r_{xx} = E[x^*(k)wx(k)] \quad (3.63)$$

$$r_{xd} = E\left\{ x^*(k) \begin{bmatrix} 0 \\ d(k) \end{bmatrix} \right\} \quad (3.64)$$

and x(k)is a circulant matrix defined by:

$$x(k) = F^{-1}x(k)F \quad (3.65)$$

Convergence properties. Using an argument similar to that in Section 3.2, the UFLMS algorithm satisfies

$$\lim_{k \to \infty} E[H(k)] = H_{opt} \quad (3.66)$$

provided that

$$\mu \langle \frac{2}{\lambda_{max}} \quad (3.67)$$

where λ_{max} is the maximum eigenvalue of λ_{max} or r_{xx} The time constants of the 2N modes of the adaptive process are given by:

$$\tau_p = \frac{1}{\mu\lambda_p} \quad \text{blocks} \tag{3.68 a}$$

$$= \frac{N}{\mu\lambda_p} \quad \text{samples} \tag{3.68 b}$$

where λ_p is the pth eigenvalue of R_{xx} or r_{xx}. Under a Gaussian assumption stated in Section 3.1, the misadjustment can be found to be

$$M = \frac{\mu}{2N} tr\{R_{xx}\} \tag{3.69 a}$$

$$= \mu NP \tag{3.69 b}$$

where P is the power of the filter input.

To ensure stability, the misadjustment is bounded by

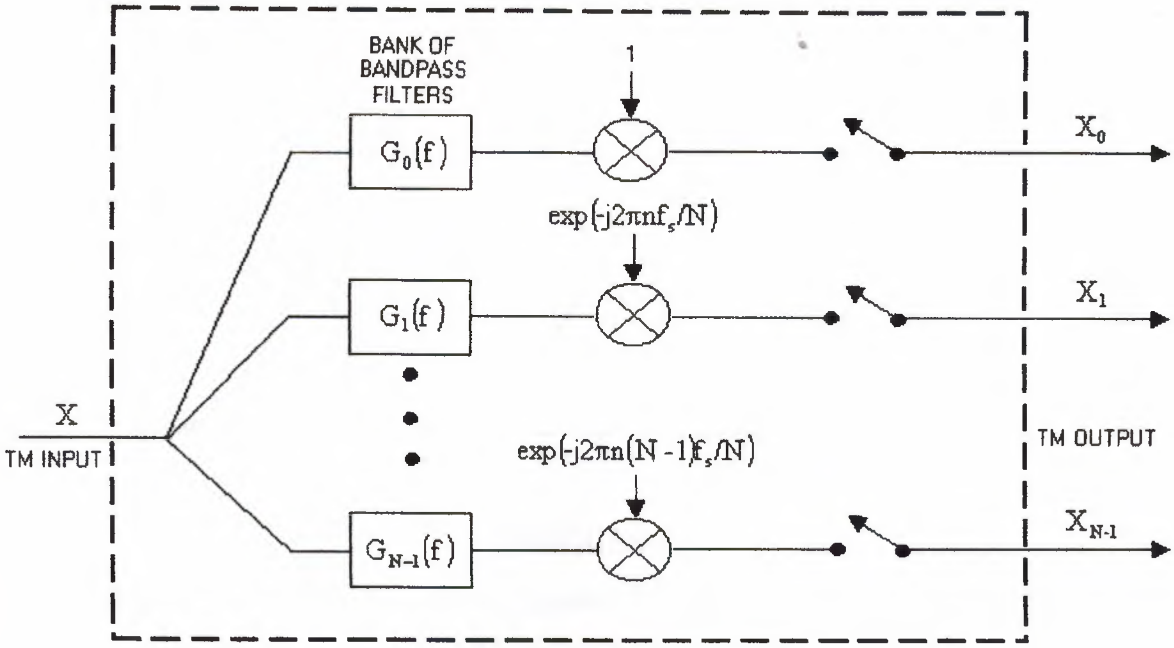
$$M \langle \frac{\lambda_{avg}}{\lambda_{max}} \tag{6.70}$$

where λ_{avg} is the average eigenvalue of R_{xx}.

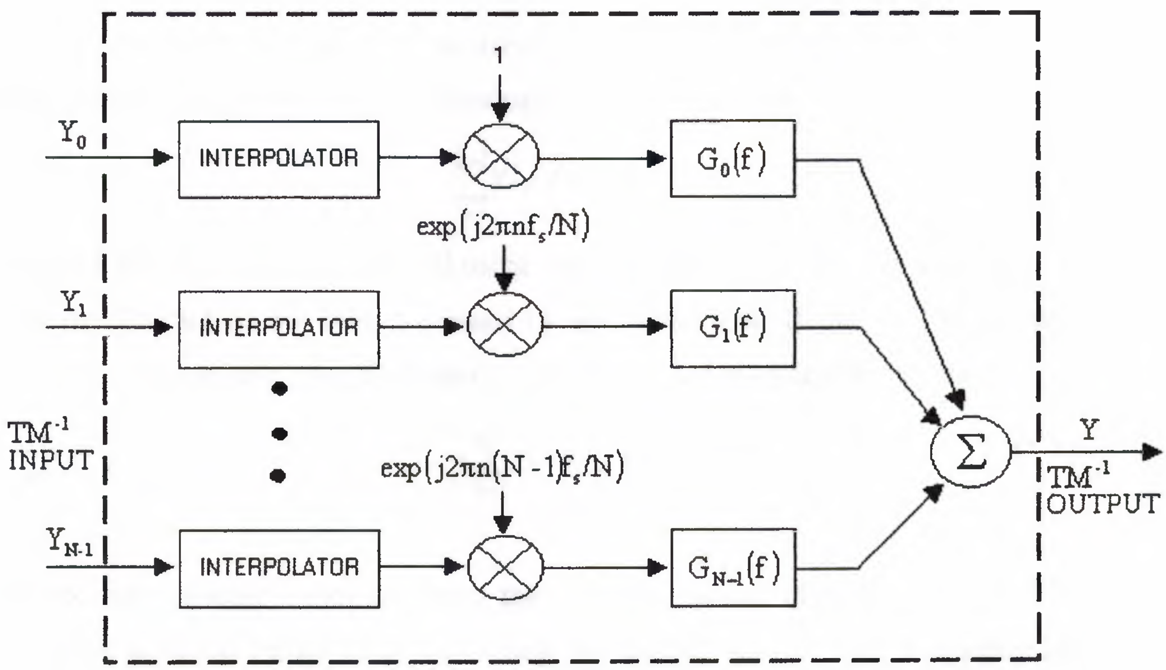
3.4 Transmultiplexer Adaptive Filter

Transmultiplexers are used in telecommunication networks to translate efficiently between time-division multiplexing (TDM) and frequency-division multiplexing (FDM). Some of the techniques used in transmultiplexing can be used to implement an adaptive filter-in an efficient manner. For our purposes, the effect of a transmultiplexer is shown in Figure 3.3(a) (actual implementation will be discussed later). The input to the transmultiplexer (TM) is filtered by a bank of N complex bandpass filters, each having essentially zero response outside a bandwidth of $2fs/N$ where fs is the sampling frequency of the input data. Adjacent bandpass filters may overlap in frequency, as shown in Figure 3.4. The filter outputs are shifted down in frequency to baseband and can be decimated by a factor of N/2 without loss of information. This process is similar to converting FDM to TDM.

The effect of the inverse transmultiplexer (similar to TDM-to-FDM conversion) is shown in Figure 3.3(b). Each input channel is interpolated to increase the sampling rate by a factor of N/2 and then processed by another bank of bandpass filters.



(a)



(b)

Figure 3.3 (a) Model of Transmultiplexer Operation; (b) Model Of Inverse transmultiplexer

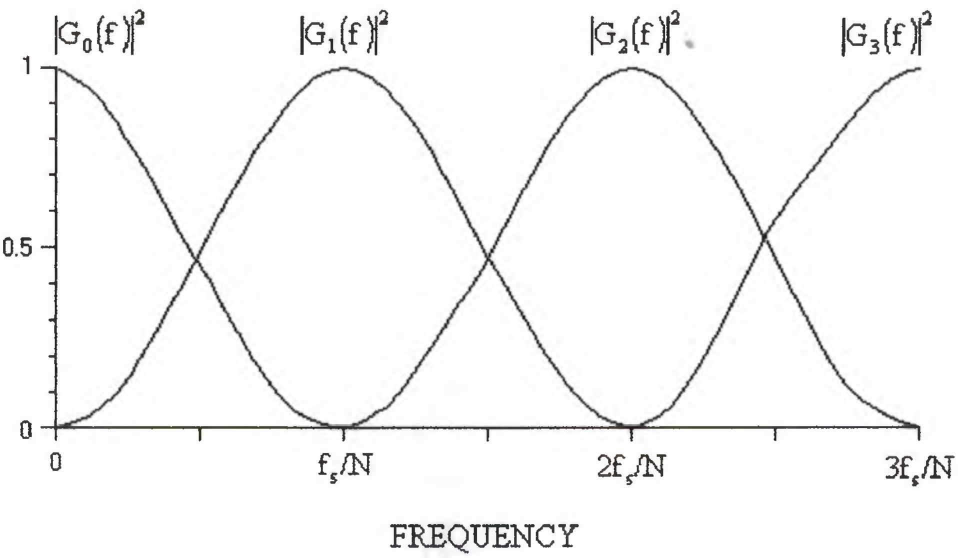


Figure 3.4 possible Magnitude-Squared Transfer Function For The bandpass Filter

The bandpass filter outputs are then summed. For the processes described in Figure 3.3 to truly be inverses of each other, the bandpass filters must satisfy

$$\sum_{i=o}^{N-1} |G_i(f)|^2 = 1 \tag{3.71}$$

General filtering can be performed using transmultiplexers by suitably weighting the outputs of the TM, which are then processed by an inverse TM. If the ith TM output is weighted by H_1, the resulting overall transfer function obtained is (except for a delay)

$$H(f) = \sum_{i=0}^{N-1} H_i |G_i(f)|^2 \tag{3.72}$$

Since only adjacent bandpass filters may overlap, design of many types of fixed filters could be achieved rather easily by setting the desired gain of each frequency bin. However, the primary advantage of the approach is in adaptive filtering.

Figure 3.5 shows a block diagram of an adaptive filter using transmultiplexers. The filter weight for the ith frequency bin is adapted to minimize the mean-square error between the output for that bin, $Y_i(k)$ and the desired response for that bin, $D_1(k)$, using the LMS algorithm:

$$H_i(k+1) = H_i(k) + \mu E_i(k) X_i^*(k) \tag{3.73}$$

where

$$E_i(k) = D_i(k) - Y_i(k) \tag{3.74}$$

and

$$Y_i(k) = H_i(k) X_i(k) \tag{3.75}$$

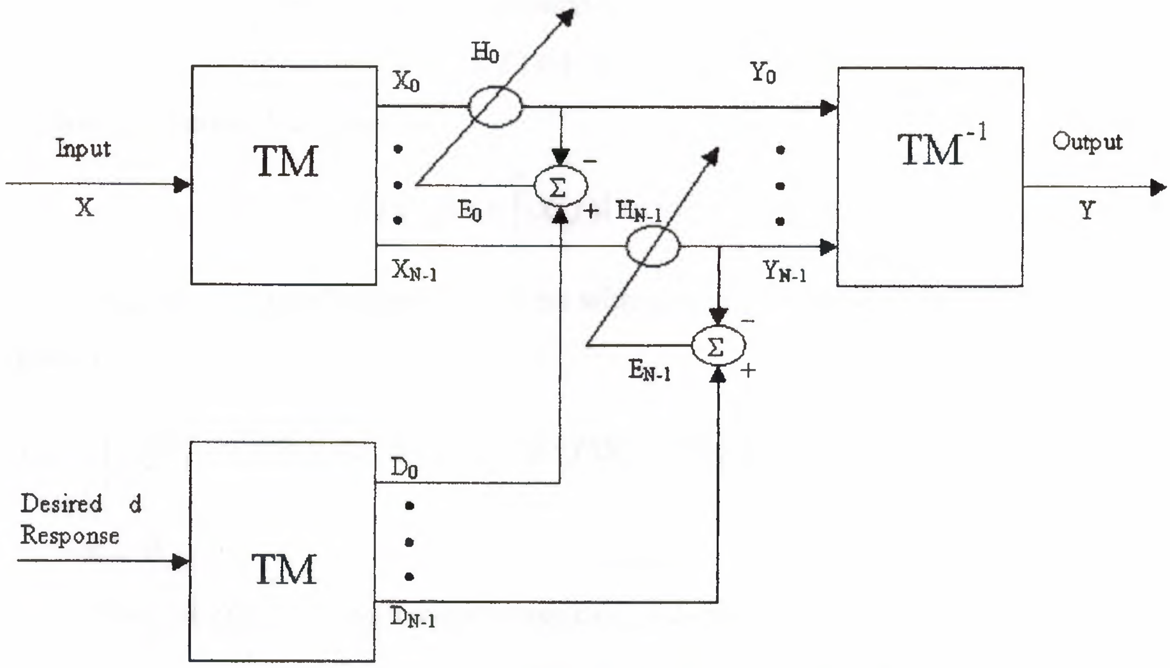


Figure 3.5 Transmultiplexer (TM) Adaptive Filter

The convergence properties of each channel are the same as that for a one-tap LMS filter. For statistically stationary inputs and μ sufficiently small, the mean value of the ith weight will converge turn to:

$$\lim_{k\to\infty} E[H_i(k)] = \frac{E[D_i(k)X_i(k)]}{E[X_i(k)X_i(k)]} \tag{3.76}$$

This weight is not, in general, the optimum weight setting to minimize the mean-square error between d(k) *and* y(k) unless the bandpass filters have no frequency overlap and have unity gain in their passband. Even so, quite reasonable results can be obtained for many scenarios, since overlap is confined to adjacent bins. Several experiments have shown that the transfer function obtained is practically indistinguishable from optimum when the optimum varies little over a bin width. Even when the optimum transfer function contained narrow notches, near-optimum performance was obtained with the adaptive filter. Optimum performance can be approached as closely as desired by reducing the amount of overlap between adjacent bandpass filters, although the complexity of the transmultiplexers is increased.

The optimum weights can be determined in the following manner. Let H be a vector containing the weights and let $\Gamma(f)$ be a vector whose ith element is $|G_i(f)|^2$. Then, according to (3.72), the filter transfer function is

$$H(f) = \Gamma^T(f)H \tag{3.77}$$

We wish to choose H to minimize

$$E\int_0^{fs}|Y(f)-D(f)|^2 df = E\int_0^{fs}|X(f)\Gamma^T(f)H-D(f)|^2 df \tag{3.78}$$

Expanding out (3.78) and setting its gradient with respect to H equal to zero results in

$$H_{opt} = \left[\int_0^{fs} E[\Gamma(f)X(f)X^*(f)\Gamma^T(f)]\right]^{-1}\int_0^{fs} E[\Gamma(f)X^*(f)D(f)]df \tag{3.79 a}$$

$$\equiv R_{xx}^{-1}R_{xd} \tag{3.79 b}$$

The ith element of the second integral above can be shown to be

$$\int_0^{fs} E[G^*_i(f)X^*(f)D(f)G_i(f)]df = E[X^*_i(k)D_i(k)] \tag{3.80}$$

If the bandpass filters are disjoint in frequency and have unity gain in their passband (zero elsewhere), R_{xx} is diagonal and its ith element is given by $E[X_i(k)X_i^*(k)]$ Therefore, under these assumptions, the proposed adaptive filter converges to the optimum solution. However, these assumptions will not be strictly satisfied in a practical system and some loss in performance from optimum will generally be obtained.

A transmultiplexer producing the effect of Figure 3.3(a) can be implemented in the following manner. Let g(n) be the M-point impulse response of a lowpass filter whose transfer function G(f) satisfies:

$$G(f) \approx 0 \quad \text{for } |f| \rangle \frac{fs}{N} \tag{3.81}$$

and

$$\sum_{i=0}^{N-1}\left|G\left(\frac{f-ifs}{N}\right)\right|^2 \approx 1 \tag{3.82}$$

The bandpass filter G,(*f*) of Figure 3.3(a) will be $G(f - ifs/N)$,the frequency-shifted low-pass filter. It will be convenient to let N be a power of 2 and M = LN for some small positive integer L $\geq$ 2. The bigger the value of L, the less the overlap can be between adjacent bandpass filters. The bandpass filter shapes of Figure 3.3 were obtained with L = 3. Figure 3.6 shows the transmultiplexer implementation for L = 3. The input is commutated into two banks of three-tap filters whose filter weights are obtained from the original low-pass filter impulse response. When the upper commutator reaches the top of the filter bank, an N-point FFT^{-1} of the filter bank outputs is computed. When the lower commutator reaches the middle of its filter bank, another FFT^{-1} is computed. The FFT^{-1} outputs are interleaved to produce a sampling rate for each output channel that is 2/N times the input sampling rate.

The inverse transmultiplexer can be implemented as shown in Figure 3.7 for L = 3. The input channels are processed by a FFT^{-1} whose outputs go into a bank of N 2L-tap filters whose filter weights are again determined by the original low-pass impulse response. The output commutator sweeps down N/2 filters from the top. At this time, a new FFT^{-1} is computed and updates all N filters. The commutator continues sweeping down until it is ready to return to the top filter, at which time another FFT^{-1} must be computed. Thus two FFT^{-1} are computed for every N points of output obtained. Because of this, each filter need only compute every other output.

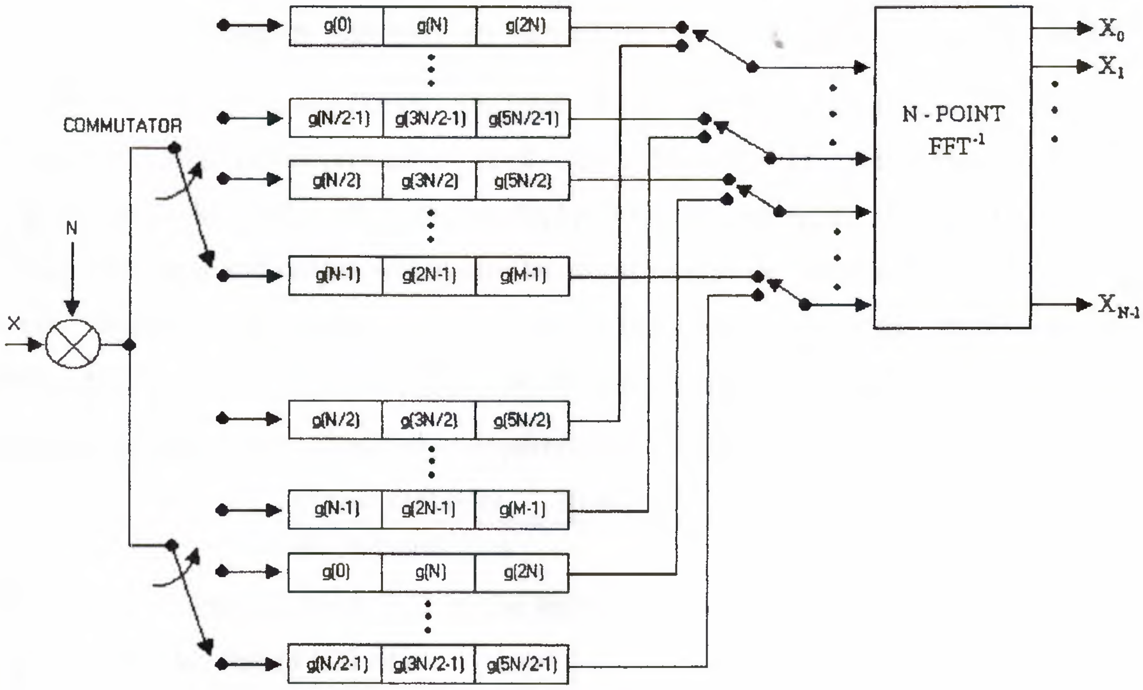


Figure 3.6 Transmultiplexer Implementation

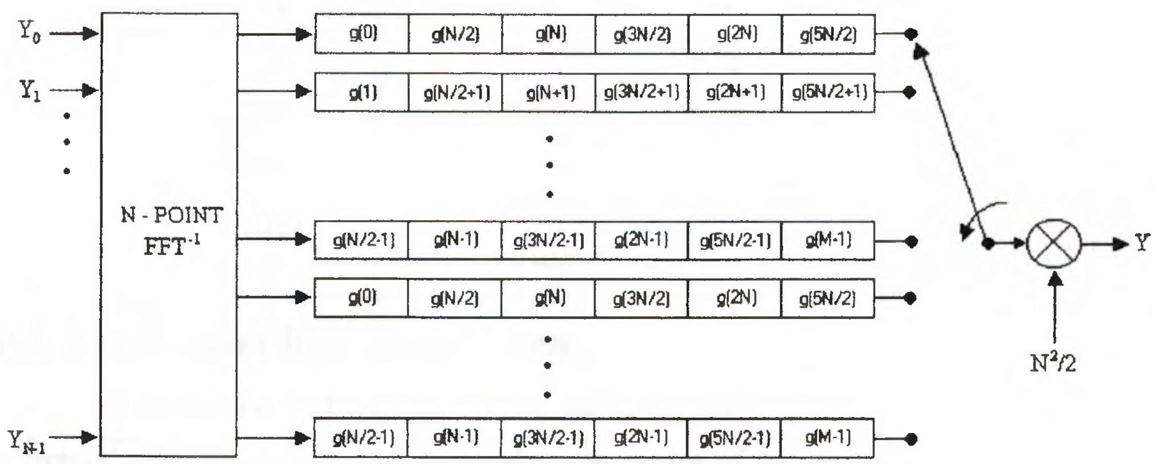


Figure 3.7 Inverse Transmultiplexer Implementation

The number of multiplies needed to compute N real output points for the transmultiplexer adaptive filter can now be computed. The input, desired response, and output transmultiplexers require two N-point FFTs each. As stated earlier, an N-point FFT can be computed using an N/2-point FFT and N/2 complex multiplies. Therefore, a total of

$\left(3N/2\right)\log_2\left(N/2\right)$ complex multiplies are needed for the FFTs. Since the weights should be symmetric, only half of them need to be used. Weighting and weight updating then take 2N complex multiplies per N processed points. Computation for the transmultiplexer input and output filters takes 2M = 2LN real multiplies each for the input, desired response, and output. The input and output scaling can be absorbed into the transmultiplexer filter coefficients. Assuming that one complex multiply equals four real multiplies yields $6N\log_2 N + 2N + 6LN$ real multiplies per N processed data points. This figure can be compared to the number of multiplies required for an N-tap LMS adaptive filter:

$$\frac{\text{TM real multiplies}}{\text{LMS real multiplies}} = \frac{3(\log_2) + 1 + 3L}{N} \tag{3.83}$$

This ratio is computed for L = 3 in the following table. The amount of computation is slightly more than that of UFLMS.

N	$\frac{\text{LM Real Multiplies}}{\text{LMS real multiplies}}(L=3)$
16	1.4
32	0.78
64	0.44
256	0.13
1024	0.04

3.5 Convergence Rate Improvement

In this section methods for altering the convergence rate of the various modes of an adaptive process are discussed. Generally, the modes of an adaptive process converge at different rates, with the rate of each mode determined by the associated eigenvalue of the input autocorrelation matrix. The eigenvalues, and thus the convergence rates, can be vastly different when the input power spectrum varies greatly with frequency. To keep the fastest converging modes from becoming unstable, the convergence rates of some of the other modes may be unacceptably slow, resulting in an inability to track nonstationarities in the input data. It would generally be desirable to make the convergence rates of the modes more equal.

The convergence rate problem is particularly bothersome for the case of block processing employed by the frequency-domain adaptive algorithms discussed in this chapter. In this case the filter weights are updated only once per block and, even though a very good estimate of the gradient may be obtained by averaging data over the entire block, the updates must be in small enough increments to ensure stability of the adaptive process. For example, when using the FLMS algorithm with N-point blocks, 2μ must be chosen less than $1/N\lambda_{max}$ to attain stability; with sample-by-sample updates, it need only be less than $1/\lambda_{max}$ Choosing the adaptation rate to satisfy the block processing stability limit is not a problem when the eigenvalues are nearly equal, since the maximum, μ would be determined by reasonable misadjustment rather than stability. However, when the eigenvalues are highly disparate, stability requires that adaptation proceed much more slowly than is necessary to achieve reasonable performance.

In either the adaptive transmultiplexer or the circular adaptive filter of Section 3.2, the weights are adapted independently from each other so that each weight is associated with one mode of the adaptive process. Since the modes are easily accessible, it is easy to alter their convergence rates. Since each weight corresponds to a one-tap LMS adaptive filter, the convergence time for the ith weight, assuming stationary inputs, is inversely proportional to $\mu\lambda_i$, where λ_i is the input power seen by that weight (λ_i is also the ith eigenvalue of the input autocorrelation matrix). In order to make all the modes converge at the same rate, we could make μ be different for each weight according to:

$$\mu_i = \frac{\alpha}{p_i} \tag{3.84}$$

where p_i is an estimate of the input power seen by the ith weight. If the p_i are good estimates for the powers, the weights all converge at the same rate with a time constant of

$$\tau = \frac{N}{\alpha} \text{ samples} \tag{3.85}$$

For the circular adaptive filter the misadjustment for *N* weights would then be:

$$M = \frac{\alpha}{2} \tag{3.86}$$

These convergence properties would be obtained provided-that the environment is stationary and good estimates of the powers are available at the start of adaptation. If the environment is nonstationary, or if the power seen by each weight is unknown, a recursive update of the power estimates can be used. A simple recursion is

$$p_i(k) = \gamma p_i(k-1) + (1-\gamma)|X_i(k)|^2 \tag{3.87}$$

where $X_i(k)$ is the input to the *i*th weight at time *k* and γ is chosen between 0 and 1. This corresponds to using an exponentially weighted average of the magnitude squared of the input values:

$$p_i(k) = (1-\gamma)\sum_{m=0}^{\infty} \gamma^m |X_i(k-m)|^2 \tag{3.88}$$

The *i*th weight is then adapted according to:

$$H_i(k+1) = H_i(k) + \frac{\alpha}{p_i(k)} E_i(k) X_i(k) X_i^*(k) \tag{3.89}$$

A particularly good choice for a is 1 —*y,* for then the weights are chosen at each point in time to minimize an exponentially weighted average of the square error given by

$$\sum_{m=o}^{\infty} \gamma^m |E_i(k-m)|^2 \tag{3.90}$$

It should be noted that when p_i changes in time, either because of nonstationarity or transients in the power estimates following startup of the adaptive process, convergence is not exponential as in gradient descent algorithms because the gradient estimate in the weight-update equation has been modified by the multiplication of the various time-varying μ_i values. For stationary inputs, the weights converge in the mean to the same final solution obtained with identical μ_i values for each weight (except when the weights are constrained, as discussed below), provided that the modes are all stable. Choosing ensures stability $\alpha = 1 - \gamma$ for then the weights always minimize a weighted least-squares criterion.

To see the effect of recursively updating the power estimates on adaptive filter convergence, it is useful to consider a step change in input power to one of the weights. For a step increase in input power of the *i*th mode, the effective $\mu_i\lambda_i$ product is initially big, since the power estimate is close to the input power before the increase. Then $\mu_i\lambda_i$ decreases to its steady-state value as the power estimate converges to the input power after the step increase. Thus the response of the adaptive filter is fastest right after the step. For a step decrease in power, just the opposite is true $\mu_i\lambda_i$, starts out small following the step, then gradually increases.

In the UFLMS adaptive filter, adaptation of the weights is not uncoupled, since the error used to adapt each weight depends on all the weights. However, we shall see that the weight adaptation becomes approximately uncoupled when there are a sufficiently large number of weights, provided that the input is stationary with no periodic components. This allows the time constants of the adaptive process to be set approximately to desired values by choosing a different μ_i for each weight as discussed previously.

The DFT coefficients of a stationary random process are essentially uncorrelated provided that the power spectrum of the process changes slowly over the bandwidth of a single DFT bin. If there are no periodic components present (including DC), this condition can be satisfied by choosing a large enough DFT size, which, for the UFLMS algorithm, is equal to the number of weights.. This is so since the *i*th, *j*th element of R_{xx} is:

$$\left(R_{xx}\right)_{ij} = W_{ij} E[X_i^*(k) X_j(k)] \tag{3.91}$$

where $X_i(k)$ is the *i*th DFT coefficient of the *k*th input block. A diagonal R_{xx} matrix implies that the expected values of the frequency-domain weights are uncoupled during adaptation, the eigenvalues of the R_{xx} matrix are given by its diagonal elements, and the *i*th time constant is $\tau_i = 1/\mu_i\lambda_i$ Computer simulation has shown that the UFLMS algorithm, with a different μ_i for each weight determined by (3.84) and (3.87), converges faster than LMS for the case of highly disparate eigenvalues.

CHAPTER 4

ADAPTIVE ECHO CANCELLATION

4.1 Overview

The performance limits of adaptive echo cancellation techniques are investigated. In particular we analyze the effects of signal characteristics such as auto and cross correlation on the achievable echo suppression. Techniques to enhance signal characteristics such as to improve both the learning ability and the steady state echo suppression quality are identified. A nice feature of our work is that it links in a natural way the complexity of the learning task (via the dimension of the adapted parameter vector), the available information (via the signal characteristics) to the achievable echo suppression quality. A number of papers are currently in a review process for journal publication

4.2 Definition

Wireless phones are increasingly being regarded as essential communications tools, dramatically impacting how people approach day-to-day personal and business communications. As new network infrastructures are implemented and competition between wireless carriers increases, digital wireless subscribers are becoming ever more critical of the service and voice quality they receive from network providers. A key technology to provide near-wire line voice quality across a wireless carrier's network is echo cancellation.

Subscribers use speech quality as the benchmark for assessing the overall quality of a network. Regardless of whether or not this is a subjective judgment, it is the key to maintaining subscriber loyalty. For this reason, the effective removal of hybrid and acoustic echo inherent within the digital cellular infrastructure is the key to maintaining and improving perceived voice quality on a call. This has led to intensive research into the area of echo cancellation, with the aim of providing solutions that can reduce background noise and remove hybrid and acoustic echo before any transcoder processing. By employing this technology, the overall efficiency of the coding can be enhanced, significantly improving the quality of speech. This tutorial discusses the nature of echo and how echo cancellation is helpful in making mobile calls meet acceptable quality standards.

4.3 History of Echo Cancellation

The late 1950s marked the birth of echo control in the telecommunications industry with the development of the first echo-suppression devices. These systems, first employed to manage echo generated primarily in satellite circuits, were essentially voice-activated switches that transmitted a voice path and then turned off to block any echo signal. Although echo suppressers reduced echo caused by transmission problems in the network, they also resulted in choppy first syllables and artificial volume adjustment. In addition, they eliminated double-talk capabilities, greatly reducing the ability to achieve natural conversations.

Echo-cancellation theory was developed in the early 1960s by AT&T Bell Labs, followed by the introduction of the first echo-cancellation system in the late 1960s by COMSAT TeleSystems (previously a division of COMSAT Laboratories). COMSAT designed the first analog echo canceller systems to demonstrate the feasibility and performance of satellite communications networks. Based on analog processes, these early echo-cancellation systems were implemented across satellite communications networks to demonstrate the network's performance for long-distance, cross-continental telephony. These systems were not commercially viable, however, because of their size and manufacturing costs.

In the late 1970s, COMSAT TeleSystems developed and sold the first commercial analog echo cancellers, which were mainly digital devices with an analog interface to the network. The semiconductor revolution of the early 1980s marked the switch from analog to digital telecommunications networks. More sophisticated digital interface, multichannel echo-canceller systems were also developed to address new echo problems associated with long-distance digital telephony systems. Based on application-specific integrated circuit (ASIC) technology, these new echo cancellers utilized high-speed digital signal-processing techniques to model and subtract the echo from the echo return path. The result was a new digital echo-cancellation technique that outperformed existing suppression-based techniques, creating improved network performance.

The 1990s have witnessed explosive growth in the wireless telecommunications industry, resulting from deregulation that has brought to market new analog and digital wireless handsets, numerous network carriers, and new digital network infrastructures such as TDMA, CDMA, and GSM. According to the Cellular Telecommunications Industry Association (CTIA), new subscribers are driving the growth of the wireless market at an annual rate of 40 percent. With wireless telephony being widely implemented and competition increasing as new wireless carriers enter the market, superior voice transmission quality and customer service have now become key determining factors for subscribers evaluating a carrier's network. Understanding and overcoming the inherent echo problems associated with digital cellular networks will enable network operators and telcos to offer subscribers the network performance and voice quality they are demanding today.

4.4 Types of Echo

4.4.1 Acoustic Echo

Acoustic echo is generated with analog and digital handsets, with the degree of echo related to the type and quality of equipment used. This form of echo is produced by poor voice coupling between the earpiece and microphone in handsets and hands-free devices. Further voice degradation is caused as voice-compressing encoding/decoding devices (vocoders) process the voice paths within the handsets and in wireless networks. This results in returned echo signals with highly variable properties. When compounded with inherent digital transmission delays, call quality is greatly diminished for the wireline caller. Acoustic echo was first encountered with the early video audio conferencing studios and as also occurs in typical mobile situations, such as when people are driving their cars. In this situation, sound from a loudspeaker is heard by a listener, as intended. However, this same sound also is picked up by the microphone, both directly and indirectly, after bouncing off the roof, windows, and seats of the car. The result of this reflection is the creation of multipath echo and multiple harmonics of echo, which, unless eliminated, are transmitted back to the distant end and are heard by the talker as echo. Predominant use of hands-free telephones in the office has exacerbated the acoustic echo problem.

Acoustic echo cancellation is required in order to provide full duplex, fully interruptible speech. The acoustic echo canceller functions by modeling the speech being passed to the loudspeaker and removing any echoes picked up by the microphone. This type of operation necessitates a much more complex unit than is used in telephony in order to remove the many acoustic (multipath) echoes generated with each syllable of speech. The tail circuit requirement, or the amount of time the canceller has to hold the power.

4.4.2 Hybrid Echo

Hybrid echo is the primary source of echo generated from the public-switched telephone network (PSTN). This electrically generated echo is created as voice signals are transmitted across the network via the hybrid connection at the two-wire/four-wire PSTN conversion points, reflecting electrical energy back to the speaker from the four-wire circuit.

Hybrid echo has been around almost since the advent of the telephone itself. The signal path between two telephones, involving a call other than a local one, requires amplification using a four-wire circuit. Although not a factor in itself on digital cellular networks, hybrid echo becomes a problem in PSTN–originated calls. The cost and cabling

required rules out the idea of running a four-wire circuit out to the subscriber's premise from the local exchange. For this reason, an alternative solution had to be found. Hence, the four-wire trunk circuits were converted to two-wire local cabling, using a device called a "hybrid" (see Figure 4.1).

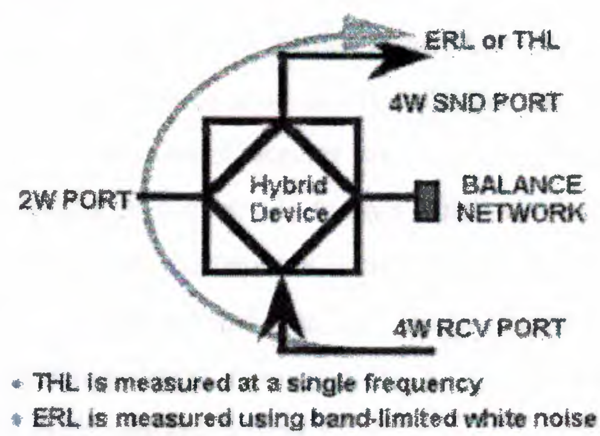


Figure 4.1. Hybrid Echo

Unfortunately, the hybrid is by nature a leaky device. As voice signals pass from the four-wire to the two-wire portion of the network, the energy in the four-wire section is reflected back on itself, creating the echoed speech. Provided that the total round-trip delay occurs within just a few milliseconds (i.e., within 28 ms), it generates a sense that the call is live by adding side tone, which makes a positive contribution to the quality of the call.

In cases where the total network delay exceeds 36 ms, however, the positive benefits disappear, and intrusive echo results. The actual amount of signal that is reflected back depends on how well the balance circuit of the hybrid matches the two-wire line. In the vast majority of cases, the match is poor, resulting in a considerable level of signal reflecting back. This is measured as echo return loss (ERL). The higher the ERL, the lower the reflected signal back to the talker, and vice versa.

4.5 Causes of Echo

Acoustic echo apart, background noise is generated through the network when analog and digital phones are operated in hands-free mode. As additional sounds are directly and indirectly picked up by the microphone, multipath audio is created and transmitted back to the talker. The surrounding noise, whether in an automobile or in a crowded, public environment, passes through the digital cellular vocoder, causing distorted speech for the wireline caller. Digital processing delays and speech-compression techniques further contribute to echo generation and degraded voice quality in wireless networks. Delays are encountered as signals are processed through various routes within the networks, including copper wire, fiber optic lines, microwave connections, international gateways, and satellite transmission. This is especially true with mixed technology digital networks, where calls are processed across numerous network infrastructures. Echo-control systems are required in all networks that produce one-way time delays greater than 16 ms. In today's digital wireless networks, voice paths are processed at two points in the network within the mobile handset and at the radio frequency (RF) interface of the network. As calls are processed through vocoders in the network, speech processing delays ranging from 80 ms to 100 ms are introduced, resulting in an unacceptable total end-to-end delay of 160 ms to 200 ms. As a result, echo cancellation devices are required within the wireless network to eliminate the hybrid and acoustic echoes in a digital wireless call

.

4.6 The Combined Problem On Digital Cellular Networks

To deal with hybrid echo created by vocoder processing delays, it is mandatory for digital cellular mobile calls to have a group echo canceller installed—even for local calls. As a result, all calls on to the PSTN must pass through an echo canceller to remove what would otherwise be a noticeable and annoying echo, as shown in Figure 4.2.

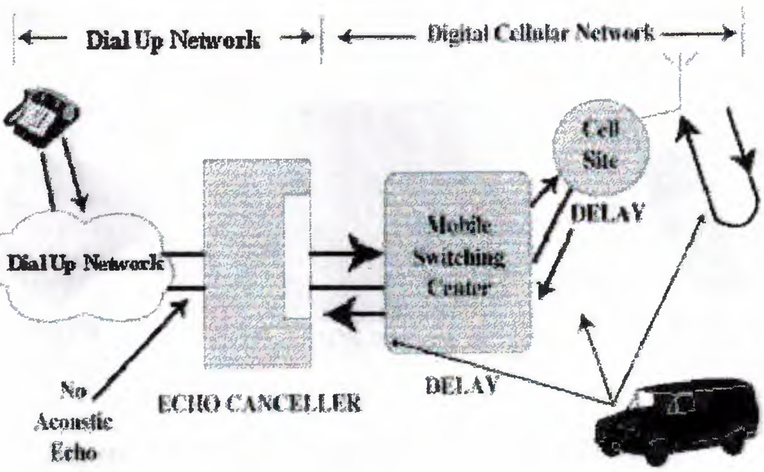


Figure 4.2. Digital Cellular Network

For example, consider a digital cellular mobile user who makes a call to the PSTN without an echo canceller in place. The user would hear his or her own speech being echoed back 180 ms or more later, even if the called person is in the same locality. The mobile user will either be using a hands-free system installed in his or her vehicle or a hand portable. In either case, these units will involve the occurrence of direct and indirect coupling between the microphone and the speaker, creating acoustic echo. In this situation, however, it is the PSTN user who suffers by experiencing poor speech quality. Hence, the echo canceller installed in the digital cellular network must be capable of handling both sources of echoes.

4.7 Process Of Echo Cancellation

In modern telephone networks, echo cancellers are typically positioned in the digital circuit, as shown in Figure 4.3. The process of canceling echo involves two steps. First, as the call is set up, the echo canceller employs a digital adaptive filter to set up a model or characterization of the voice signal and echo passing through the echo canceller. As a voice path passes back through the cancellation system, the echo canceller compares the signal and the model to cancel existing echo dynamically. This process removes more than 80 to 90 percent of the echo across the network. The second process utilizes a non-linear processor (NLP) to eliminate the remaining residual echo by attenuating the signal below the noise floor.

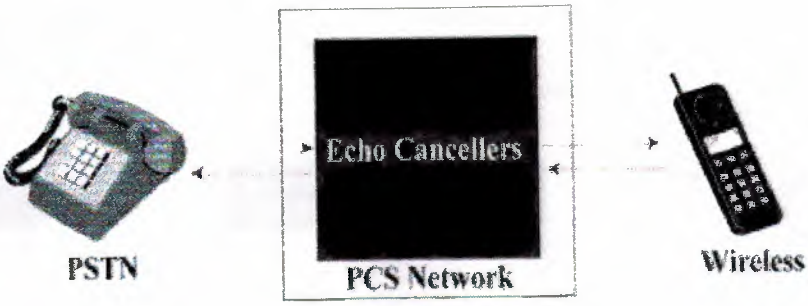


Figure 4.3. Typical Location of Echo Cancellers

Today's digital cellular network technologies, namely TDMA, CDMA, and GSM, require significantly more processing power to transmit signal paths through the channels. As these technologies become even more sophisticated, echo control will be more complex. Echo cancellers designed with standard digital signal processors (DSPs), which share processing time in a circuit within a channel or across channels, provide a maximum of only 128 ms of cancellation and are unable to cancel acoustic echo. With network delays occurring in excess of 160 ms in today's mixed-signal network infrastructures, a more powerful, application-specific echo-cancellation technology is required to control echo across wireless networks effectively

4.8 Controlling Acoustic Echo

In echo cancellation, complex algorithmic procedures are used to compute speech models. This involves generating the sum from reflected echoes of the original speech, then subtracting this from any signal the microphone picks up. The result is the purified speech of the person talking. The format of this echo prediction must be learned by the echo canceller in a process known as adaptation. It might be said that the parameters learned from the adaptation process generate the prediction of the echo signal, which then forms an audio picture of the room in which the microphone is located. Figure 4.4. shows the basic operation of an echo canceller in a conference room type of situation.

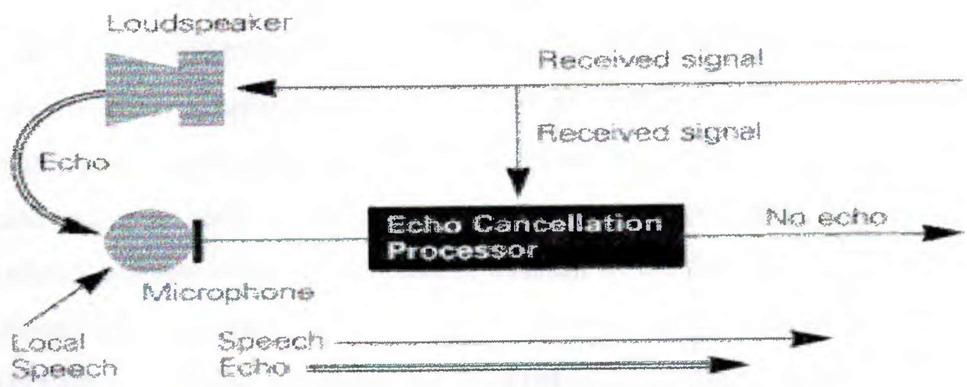


Figure 4.4. Operation of an Acoustic Echo Canceller

During the conversation period, this audio picture constantly alters, and, in turn, the canceller must adapt continually. The time required for the echo canceller to fully learn the acoustic picture of the room is called the convergence time. The best convergence time recorded is 50 ms, and any increase in this number results in syllables of echo being detected.

Other important performance criteria involve the acoustic echo canceller's ability to handle acoustic tail circuit delay. This is the time span of the acoustic picture and roughly represents the delay in time for the last significant echo to arrive at the microphone. The optimum requirement for this is currently set at 270 ms—any time below this could result in echoes being received by the microphone outside the ability of the echo canceller to remove them, and hence in participants hearing the echoes.

Another important factor is acoustic echo return loss enhancement (AERLE). This is the amount of attenuation which is applied to the echo signal in the process of echo cancellation-i.e., if no attenuation is applied, full echo will be heard. A value of 65 dB is the minimum requirement with the non-linear processor enabled, based on an input level of -10 dBm white noise electrical and 6 dB of echo return loss (ERL).

The canceller's performance also relies heavily on the efficiency of a device called the center clipper, or non-linear processor. This needs to be adaptive and has a direct bearing on the level of AERLE that can be achieved.

4.9 Controlling Complex Echo In A Wireless Digital Network

Although acoustic echo is present in every hands-free mobile call, the amount of echo depends on the particular handset design and model that the mobile user has. On the market are a few excellent handsets that limit the echo present, but, due to strong price pressures, most handsets do not control the echo very well at all, in fact some phones on the market have been determined to have a terminal compiling loss of 24 dB. Echo becomes a problem when the processing inherent to the digital wireless network adds an additional delay (typically in excess of 180 ms round-trip). This combination makes for totally unacceptable call quality for the fixed network customer, as shown in Figure 4.5

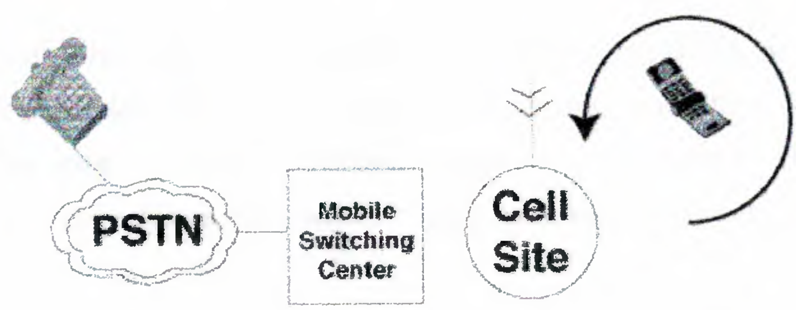


Figure 4.5. Acoustic Echo in a Mobile Environment

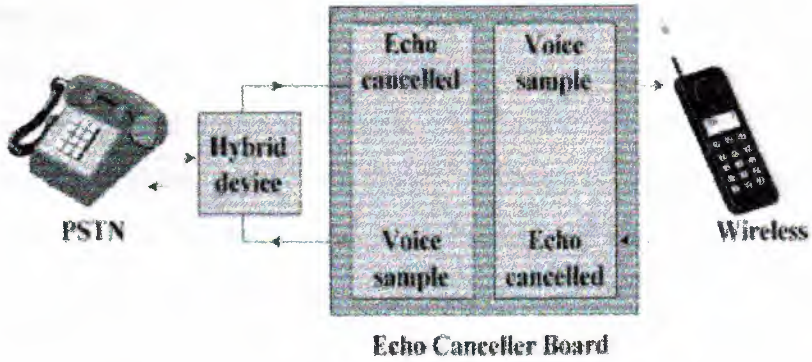


Figure 4.6. Bi-directional Echo Cancellation

This back-to-back configuration ensures a high audio quality for both PSTN and mobile customers. In addition, the echo canceller's software configuration is designed to provide a detailed analysis of background noises, including acoustic echo from the mobile user's end. Some echo cancellers incorporate a user-settable network delay, which enables network operators to fine-tune the echo control to suit their parameters via a menu option on the canceller's hand-held terminal or on the network management system (NMS).

4.10 Room For Improvement In The Handset

Applying effective echo control via the echo-cancellation platform is one way of improving the overall call clarity on digital cellular networks. Another derives from improvements that must be made within the handset or terminal itself. There also is considerable room to enhance the network itself, focusing principally on vocoder development.

Recent headlines have charted the ongoing commercial battles regarding which digital technologies will eventually emerge as the winners, as equipment manufacturers fight it out. However, this public battle will soon be overshadowed by another battle concerning handsets. At present, there are four major players in the digital cordless market. Europe has cordless telephony (CT2) and digital European cordless telephony (DECT), while Japan has the personal handyphone system (PHS) and the United States has personal communications services (PCS).

Connecting directly into the plain old telephone system, CT2 was one of the first digital technologies to provide low-cost mobile phones. Although the technology worked well, it had a fundamental problem: it could not handle cell handovers. DECT and GSM have overcome this problem and will eventually dominate European cellular services.
During the development of early cordless telephony, attention was paid to basic and enhanced functions and interworking with different network architectures. While the early generation of handsets looked very elegant and aesthetically pleasing, very little attention was paid to designing the handset with echo suppression/cancellation in mind. The result was that they looked good but were extremely poor at reducing acoustic echo.

In the setting of standards for GSM and PCS, handset design and the impact of different design approaches on call quality was researched. As a result, recommendations stated a range of parameters, including sidetone tolerance and echo return loss performance. With the resultant advent of new recommendations with much tighter requirements for handsets, there is a call for greatly improved designs to be implemented. This, complemented by ongoing improvements in network technology and echo cancellation techniques, will bring digital wireless telephony much closer to matching wireline quality.

4.11 Echo Cancellation System For Radio Telephony

The customer has developed technology for major manufacturers in the telecommunications industry. This has included a host of popular electronic devices, from cellular and cordless telephones, to computers and digital television products.
Microsystems Engineering has worked extensively with the customer over a number of years developing echo cancellation systems for radiotelephony projects. The project described here is an example of a recent echo cancellation system designed and implemented by Microsystems Engineering.

Many radiotelephony devices require additional echo control to be incorporated into the system. The main reason for this is the group delay usually imposed by the radio link protocol. Sources of echo that would not normally be noticeable to the user become annoying due to the 10 - 20ms round trip delay that often exists between the portable part and the fixed part of the system.

The diagram below illustrates the elements of a typical radiotelephony system that relate to echo and its control.

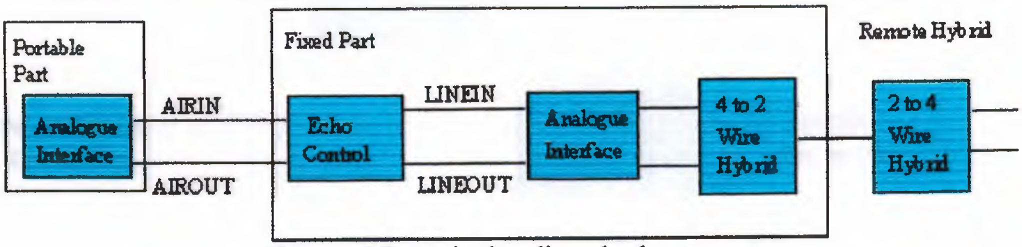


Figure 4.7. typical radio telephony system

The echo control part of the system would generally reside at the base station (in the Fixed Part of the above diagram) and would be responsible for three kinds of echo:

- Coupling at the portable part resulting in echo of the AIROUT signals back into the AIRIN signal. This is generally of the order of 20-21ms. For the PSTN equipment to suppress this an artificial echo signal needs to be added by the system.
- Reflection at the fixed part 4-wire to 2-wire hybrid resulting in short delay echo of LINEOUT signals in the LINEIN signal (0- 4ms). This is cancelled using an adaptive FIR algorithm.
- Reflection at the exchange 2-wire to 4-wire hybrid resulting in long delay echo of LINEOUT signals in the LINEIN signal. This can be between 0 and 70ms. This is reduced to acceptable levels using a soft suppressor algorithm.

Five main operations were implemented:

- Signal energy analysis (for both LINEIN and AIRIN inputs)
- Local hybrid echo estimation (FIR implementation) - delays in the range 0-4ms
- Local hybrid echo estimator adaptation (FIR tap weight adaption)
- Artificial handset echo generation
- Network echo suppression

These functions were implemented using using a Texas Instruments TMS320C50 DSP.

Testing An echo control simulation environment was designed for testing. This involved the digital simulation of the DECT echo environment using internal DSPs. It was PC based, using plug-in DSP hardware and a Windows 95 controller application.

The diagram below shows an example screenshot from the controller application

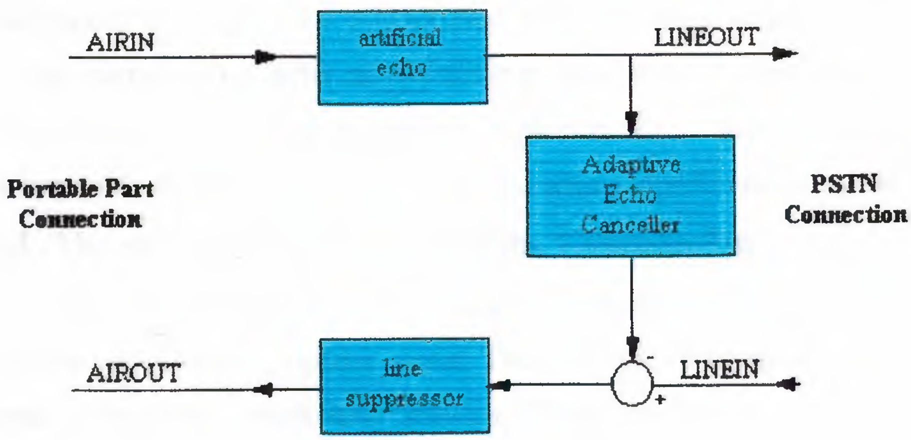


Figure 4.8 screenshot of controller application

The controller application allows full control over the air and line echo characteristics. Real time subjective testing was possible using external telephone handsets.

The system underwent approvals testing to ensure it met the Digital European Cordless Telephony standards.

4.12 Adaptive Sub-Band Cancellation of Acoustic Echo In Loud-Speaking Telephone

Adaptive cancellation of the acoustic echo is essential to the perceived quality of any loud-speaking telephone system including video telephone and tele-conferencing equipment. In general the speech quality is optimized by combining the adaptive echo canceller with a voice control and careful acoustic design of the speech terminal. By splitting the speech signal into a suitable number of subbands, the computational complexity is reduced roughly by a factor equal to the number of subbands. When using conventional transform (DFT) or critically sampled filterbanks, the subband signals contain aliased frequency components disturbing the adaptive algorithm for updating the cancellation filters. This problem is overcome by introducing oversampling, that is by decimating the subband signals with a number less than the number of subband channels. This causes a penalty in complexity quadratically proportional to the oversampling factor.

SINTEF DELAB has cooperated with the Norwegian Telecoms Research lab (TF) in the development of a rationally oversampled filterbank with (near) perfect reconstruction. Rational oversampling factor allows for implementation of oversampling factors close to 1. This filterbank algorithm is used to implement a real time echo cancellation system, and is suitable for other applications such as audio and video compression.

4.13 The principle Of Acoustic Echo Cancellation

The far-end signal is fed through an adaptive filter forming an output to be subtracted from the microphone signal. The microphone signal is in general consisting of signals from three contributing sources. These are, the near end speaker, near end background noise and echo of the far-end audio signal originating from the terminal loudspeaker. The latter is an undesired component. Thus the objective of the adaptive echo canceller is to form a replica of the acoustic echo signal as picked up by the terminal microphone. This is achieved by inserting an adaptive filter parallel to the signal path through loudspeaker, room and microphone. The objective of the adaptive filter is then to provide a response as equal as possible to that of the acoustic signal path. Then the far end speech signal is fed through the adaptive filter to resemble the echo part of the microphone signal. The echo is then subtracted from the input audio signal to enhance the speech quality.

4.14 Practical Concerns

As described so far, the echo cancellation system appears as a straight-forward realization of an adaptive filtering algorithm as the gradient-based LMS-algorithm (Widrow-Hoff) or a recursive least squares (RLS) approach. However, as many researchers have experienced, there is a lot more to it. One concern is the computational complexity for a full band implementation. As an example cancellation of 1/2 s echo response at 16 kHz audio sampling frequency involves filtering through an FIR-filter consisting of 8 000 taps working at 16 kHz, not mentioning the adaptive updating of 8 000 filtertaps. In addition the speech signal itself possess properties one has to account for in the adaptive filtering algorithms. The dynamic character of speech including intervals of complete silence is proven to be a problem in adaptive filtering. In addition the far from white spectral character slows down the adaptation speed causing long convergence time and making the system sensitive to changes of the acoustic room response. Finally the near-end speech and background noise if present also put demands on the system design. As viewed from the adaptive algorithm the near-end speech signal appears as a disturbance. As this signal in general is and should be the strongest component it may cause large misadjustment to the filter taps. To overcome this problem an adaptation control algorithm is needed. In general such an algorithm is based on an analysis of the microphone signal.

4.14.1 The Sub-Band Approach

The idea of sub-band realization addresses two of the concerns as stated above. First the complexity is reduced by dividing the signal into subbands and applying adaptive filters to a decimated signal in each subband. Also the spectral variability within a subband is reduced as compared to the full band signal. To maintain transparency of the near end speech signal one will require the cascade of the analysis and synthesis filterbanks to provide perfect reconstruction. This simply means that the a signal fed through the analysis and synthesis filterbank system shall be an exact but delayed copy of the input signal. The intuitive first step to a subband implementation is to use a critically sampled filterbank. A finite length, discrete Fourier transform (DFT) can be viewed as an example of such a scheme. This approach will always imply spectral folding (aliasing) of the subband signal. The aliased signal components are disturbing the adaptive updating algorithm working in each subband and thus reduce the signal quality. One way to circumvent this problem is to introduce cross-cancellation filters between adjacent subbands. This approach is shown to give only minor improvement of the signal quality. An alternative is to introduce oversampling in the subbands. This way the complete passband and transient band of the subband signal can be constructed to fall within the aliasing-free band of the subband sampling frequency. The principles of critical sampling and oversampling are illustrated

Of course some aliasing noise will still be present but in this case the aliased signal is due to signal leakage through filter sidelobes only. By proper filter design this leakage can be made much smaller than for the critically sampled case where parts the transient band will dominate the aliased signal components. A filter design based on oversampling by a factor of two is carried out and demonstrated in a real time environment.

However the price to pay for introducing oversampling is an increase in adaptive filter complexity by a factor equal to the square of the oversampling factor. This motivates for development of schemes with oversampling factor less than two which implies a non-integer oversampling factor. This can be implemented with a rational oversampling factor, for example by using 4 complex subbands (8 channels) and a decimation factor of 7 or less.

4.15 Design Of A Filterbank With Rational Oversampling And Near Perfect Reconstruction

Motivated by the items as elaborated through the above discussion, we started a study of rationally over sampled filter banks with perfect or near perfect reconstruction. We found very soon that a perfect reconstruction filter bank is always possible to implement with finite length analysis and synthesis filters but the synthesis filters turn out to contain a much larger number of taps than the analysis filters.

For equally length synthesis and analysis filters we have not been able to show the existence of an exact perfect reconstruction filter banks. However by developing an optimization routine we were able to design a filterbank with better than -60 dB reconstruction error and less than -50 dB aliasing noise within reasonable complexity and with an oversampling ratio of 4/3.

4.16 Status And Further Work

The project ended up with a real time demonstration of the system including a filterbank with rational oversampling. Possible extensions include industrialization of the system, alternative applications of the rationally oversampled filter bank, and further theoretical analysis of the design of rationally oversampled filter banks.

18el Digital Echo Canceller. TRANSMIT OPTICAL AMPLIFIER. DART is a product development strategy that rides the technology curve rather than chasing it. Erbium Doped Fiber Amplifiers. Optical Amplifiers, optical networking products. The smallest echo canceller available, providing the highest packing density in the industry. Erbium Doped Fiber Amplifiers. Digital echo canceller products. Optical Amplifiers, Networking

Business. Echo Cancellers, Optical Amplifiers. The TRANSMIT OPTICAL AMPLIFIER provides an optimal solution to support the fiber networks of the future. Echo Canceller Product Series. DART is a product development strategy that rides the technology curve rather than chasing it. Ditech's product advantages are built on a fundamentally more efficient means of solving the echo cancellation problem. Because of DART, a single Ditech DSP can monitor multiple channels and dynamically assign the full processor resources to each Ditech's DART approach is more efficient because it only applies processing power when and where it is needed. Echo control is becoming an increasingly important key to ensuring call quality and customer retention. Solving the echo cancellation problem. Ditech's product advantages are built on a fundamentally more efficient means of solving the echo cancellation problem called Optical Amplifiers, Networking Business. TRANSMIT OPTICAL AMPLIFIER. Optical Amplifiers, optical networking products. Erbium Doped Fiber Amplifiers. Echo Cancellers, Optical Amplifiers. Erbium Doped Fiber Amplifiers. Digital echo canceller products. The smallest echo canceller available, providing the highest packing density in the industry. The TRANSMIT OPTICAL AMPLIFIER provides an optimal solution to support the fiber networks of the future.

Because of DART, a single Ditech DSP can monitor multiple channels and dynamically assign the full processor resources to each Ditech's product advantages are built on a fundamentally more efficient means of solving the echo cancellation problem. Ditech's product advantages are built on a fundamentally more efficient means of solving the echo cancellation problem called Ditech's DART approach is more efficient because it only applies processing power when and where it is needed. Echo control is becoming an increasingly important key to ensuring call quality and customer retention. Solving the echo cancellation problem. Echo Canceller Product Series. 18e1 Digital Echo Canceller. DART is a product development strategy that rides the technology curve rather than chasing it. Erbium Doped Fiber Amplifiers. Optical Amplifiers, optical networking products.

The TRANSMIT OPTICAL AMPLIFIER provides an optimal solution to support the fiber networks of the future. The smallest echo canceller available, providing the highest packing density in the industry. Erbium Doped Fiber Amplifiers. Echo

Echo Cancellers, Optical Amplifiers. Erbium Doped Fiber Amplifiers. Optical Amplifiers, optical networking products. Digital echo canceller products. The TRANSMIT OPTICAL AMPLIFIER provides an optimal solution to support the fiber networks of the future. TRANSMIT OPTICAL AMPLIFIER. Ditech's product advantages are built on a fundamentally more efficient means of solving the echo cancellation problem. Ditech's product advantages are built on a fundamentally more efficient means of solving the echo cancellation problem called 18e1 Digital Echo Canceller. Echo control is becoming an increasingly important key to ensuring call quality and customer retention. Solving the echo cancellation problem.

Because of DART, a single Ditech DSP can monitor multiple channels and dynamically assign the full processor resources to each DART is a product development strategy that rides the technology curve rather than chasing it. Echo Canceller Product Series. Ditech's DART approach is more efficient because it only applies processing power when and where it is needed. Optical Amplifiers, optical networking products. The TRANSMIT OPTICAL AMPLIFIER provides an optimal solution to support the fiber networks of the future. Echo Cancellers, Optical Amplifiers. Digital echo canceller products.

CONCLUSION

While the importance of the analogue filters is continuously being continuously reduced by their digital counterparts, they remain an important study, if for no other reason than they provide a gateway to study of digital filters. An anti-aliasing filter introduced to the signal processing system is based on analogue filtering. Analogue and digital conventional filters provide filtering if there is no overlap between spectrum signals noise.

An adaptive filter having self organizing structure based on recursive algorithm make it possible to perform satisfactory filtering in an environment where complex knowledge the relevant signal characteristics are not available.

Comparison between LMS algorithm with others shows that LMS algorithm is simple for realization and computation, and it does not require off-line gradient estimation of data .

The performance limits the adaptive echo cancellation techniques are investigated. In particular was analyzed the effects of signals characteristics such as auto-and cross-correlation on the achievable echo suppression. Techniques to enhance signal characteristics such as linearity and signal to noise ratio are identified.

Acoustics echoes generated by telephone hand- set line echo, results obtained based on the analysis and interpretation theory an design of adaptive filters were recommended for echo cancellation in the digital telephone communication system. Simulation echo cancellator by MATLAB, shows that it satisfy the requirements for real time noise cancellation.

REFERENCES

[1] Waren ,H. Telecommunication, 3^{rd} edition .prentice- hall inc.Englewood Cliff NJ.1998

[2]Sindhu,P 'retransmission Error Control with Memory ' IEE Transaction on Communications, vol.Com-25 , May 1977, PP.1-10.

[3] John, E.M "cellular phone and adaptive filtering.", http://www.mcw.edu/gere. Retrieved Mart 25.2000.

[4] Tel-599 . Thesis handbook http://www.usd/edu/history/deptguied.htm retrieved in Feb. 20.2000.

[5] Warren Hioky. Telecommunication.2^{nd} edition. Printice hall International Edition 1995

[6] Daniel Minoli. Telecommunication (adaptive filtering) 1993.

[7] Rhee Man Young . Cellular Mobile Communication and Networks security. Printice Hall Inernational Edition 1999.

[8] Bedya, Wiliamsj. Comparison between LMS algorithms .2^{nd} edition. Printice Hall.1995.