



Pilkington Library

Author/Filing Title ASGHAR

Vol. No. Class Mark T

**Please note that fines are charged on ALL
overdue items.**

FOR REFERENCE ONLY

0402590767



**SOFTWARE AND HARDWARE IMPLEMENTATION TECHNIQUES FOR
DIGITAL COMMUNICATIONS RELATED ALGORITHMS**

BY

SAFDAR M ASGHAR

DOCTORAL THESIS


Submitted in partial fulfilment of the requirements

For the award of

Doctor of Philosophy of Loughborough University

18th June, 2001

© (Safdar M Asghar) (2001)

 Loughborough University Public Library
Date NW 02
Class
Acc No. 040259076

**Software and Hardware
Implementation Techniques
for
Digital Communications-Related Algorithms**

© 2000, 2001 Saf Asghar

All rights reserved. No part of this book shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without written permission from the publisher. No patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this book, the publisher and author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

Acknowledgements

To my wife Janet for her untiring support, encouragement and patience over the years.

Her love and sacrificies enabled me to fulfill my commitments and achieve my goals.

To Professor I. R. Smith for his advice and constructive comments in the formulation of this thesis.

To Marcus and Sharla Brewer for assisting me in bringing this thesis together.

Dedication

This book is dedicated to my mentor

PROFESSOR ADRIAN CLARK

without whose guidance I would not have achieved my goals in life.

Table of Contents

	Introduction	1
Chapter 1	Model of a Data Transmission System and Transmission Characteristics	5
Chapter 2	Digital Modems for Voiceband HF Radio Links	11
Chapter 3	The HF Channel	15
Section 3.1	Types of Distortion Occurring in HF Channels	15
Section 3.2	Characteristics of HF Channels	18
Section 3.3	Model of HF Channel	21
Chapter 4	Transmitter and Receiver Filters for the 2.4 K Bits/Sec 4-Point QAM H. F. Modem	37
Chapter 5	2400 Bits/Second Modem	47
Chapter 6	Detection of Multilevel QAM Signals	53
Chapter 7	Reduced-Complexity Detection Process for Multilevel Signals	65
Chapter 8	Adaptive Arrangement of Near-Maximum Likelihood Detection Process	73
Chapter 9	Detection Process for a Time-Varying Channel	79
Chapter 10	Digital Tone Detection and Generation	115
Chapter 11	Apparatus for Adaptively Tuning to a Received Periodic Signal	129
Chapter 12	Apparatus and Method for Synthesizing a Sinusoidal Signal	147
Chapter 13	Apparatus/Method for Analyzing Speech Signals to Find Speech Signal Parameter Characteristics	179
Chapter 14	Apparatus for Generating a Sinusoidal Output Signal	225
Chapter 15	Frequency Controlled Recursive Oscillator Having Sinusoidal Output	237
Chapter 16	Communications Processor	253
	Discussion and Conclusion	307
Appendix 1	Patents	309
Appendix 2	Computer Simulation Program	313

Introduction

There are essentially three areas addressed in the body of this thesis.

a. The first is a theoretical investigation into the design and development of a practically realizable implementation of a maximum-likelihood detection process to deal with digital data transmission over HF radio links. These links exhibit multipath properties with delay spreads that can easily extend over 12 to 15 milliseconds. The project was sponsored by the Ministry of Defense through the auspices of the Science and Engineering Research Council. The primary objective was to transmit voice band data at a minimum rate of 2.4 kb/s continuously for long periods of time during the day or night. Computer simulation models of HF propagation channels were created to simulate atmospheric and multipath effects of transmission from London to Washington DC, Ankara, and as far as Melbourne, Australia. Investigations into HF channel estimation are not the subject of this thesis. The detection process assumed accurate knowledge of the channel.

In the final simulation, an HF channel estimator was included in the receiver structure of the detector for demonstration purposes. Channel estimation and modeling was a separate research topic conducted by Frank McVerry. The author's contribution was entirely centered in the development of the detection process for known time-varying channels. The channel model was directly imported from McVerry's work to the computer simulation attached in Appendix 2. This project was commercialized, and the product was, and probably is, being manufactured by Marconi Communications, Essex, for commercial use, or otherwise, today.

b. The second area (Chapter 10 forward) addresses the development of a simple and cost-effective mathematical equation to generate sine/cosine waves backed by a U.S. patent. This equation progressed over time to fundamentally simplify various transmission-type functions in digital communications and speech signal processing. Six/seven patents that carry the basic theme and progression with systematic modifications to the equation are described to demonstrate the value, simplicity, and effectiveness of the

equation. The main applications of this equation extend from tone generation, tone detection, frequency estimation and correction. A simple form of a first order neural network for adaptive frequency estimation is also described. This equation was successfully applied to a very low bit rate speech coder, <2.0 kb/s for a cost-effective silicon implementation. Acknowledgements are extended to Messrs Bartkowiak, Linz, Ireton, Pyi, and Zhao. The patents described in this section are selected to portray a theme or an evolution of a tone generation scheme that even by today's standards is extraordinarily simple, elegant, and robust. The architecture can also be utilized for detection of tones without any change. The patents have been exploited in a number of successful products manufactured by Advanced Micro Devices, Inc.

c. A third area addressed in Chapter 16 highlights the aspect of efficient partitioning of communications-related signal processing algorithms in multiprocessing silicon architectures. It covers a methodology to implement a very complex speech compression algorithm in an architecture comprising an industry standard central processing unit (CPU) and a simple digital signal processor (DSP) for the North American Digital Cellular Standard (IS54). The objective here was low cost and low power consumption since mobile telephony requires it. There is no automated, computer-assisted tool available to efficiently partition such algorithms across architectures that are tailored for specific requirements. It is generally accepted that DSPs are not controllers, and CPUs are not DSPs. This methodology was compared with traditional implementations in standard general-purpose DSPs. The results from our implementation showed a significant reduction in power consumption, lower clock rate, and flexibility in the implementation. This type of architecture is resident in the majority of portable wireless terminals targeted for the wide area or local area networks.

Most digital communications and control system architectures have CPUs and DSPs integral to the system. Therefore, for low cost and optimum performance, it is very desirable to utilize or exploit efficiently the existing resources available on the system board. Chapter 16 describes one such scenario. More recently, CPUs and DSPs have been integrated on a single silicon substrate, further reducing the

overall system cost and improving the performance efficiency due to the proximity of the two components. This alleviates the need for power consuming interface drivers between the DSP and the CPU.

Chapter 1

Model of a Data Transmission System and Transmission Characteristics

The data transmission system considered in this thesis is based on the model shown in Figure 1-1 "Synchronous Serial Baseband Data Transmission System" on page 7.

This is a synchronous, serial baseband system, where the input to the transmitter filter is a stream of regularly spaced impulses. The value, or area, of each impulse, $s_i\delta(t - iT)$, represents the value of the corresponding signal element. These impulses may represent either binary or multilevel signals.

The transmission path could either be a lowpass channel, with a frequency limit no greater than about 10 KHz, or a typical voice frequency channel with a frequency band no wider than about 3 KHz, such as could be obtained over a telephone network or an HF radio link. In the latter case, the transmission path in Figure 1-1 is assumed to include a linear modulator (at the transmitter) and a linear demodulator (at the receiver), the whole forming a linear baseband channel. The HF radio transmission system is a modulated carrier system which essentially implies that the band pass channel introduces frequency translation effects comprising shifts of one to two Hz in the signal spectrum. This prevents correct detection of a baseband signal unless the frequency shift can be determined exactly at the receiver and suitably corrected for. In our analysis we assume the received carrier to be locked and timing recovery has been established. The system uses a simple approach of modulating the carrier with the original baseband signal, such that the spectrum of the carrier signal lies within the pass band of the transmission path.

Over practical or band pass channels, the characteristics of the channel are either not known prior to a transmission, or may vary considerably (but usually slowly) over time as in the case of HF radio links. It is necessary to estimate the sampled impulse response of the transmission channel as accurately as possible for the purposes of acceptable receiver performance. [1] Typical fading rates of an HF radio link are in the region of 4 to 15 fades a minute which for transmission rates of 1 or 2 k elements per sec. have relatively little impact on the overall transfer function of the channel. In recent years burst mode transmis-

sion which covers a few hundred symbols in a frame further improves receiver performance. Therefore, where there is "negligible" change in the sampled impulse response of the channel over the time interval of interest (~100 symbols) and the receiver has a good estimate of the channel detection processes such as equalization and or near maximum likelihood scheme become excellent candidates for HF radio receivers [2]. We have assumed throughout our study the sampled impulse response of the channel is known.

The transmitter filter limits the spectrum of the transmitted signal energy to the approximate available bandwidth of the transmission path. The transmission path is assumed to be a band pass channel with the transmitter to include low pass equivalents of all the filters involved in the linear modulator. The receiver includes all the low pass filter equivalents of all filters involved in the linear demodulation process. The impulse response of the equivalent baseband channel includes the transmission path and all the transmitter and receiver filters.

Transmission channels introduce noise which cause degradation in the received signal quality impacting receiver performance. For practical purposes, noise is assumed to be additive white Gaussian noise. It has been shown that if a data transmission system has a better tolerance to additive white Gaussian noise than another, it will also, in general, have a better tolerance to other types of additive noise. In our computer simulation of the recommended detection system at end of this chapter, extensive experiments using additive un-correlated and correlated Gaussian were conducted to satisfy practical and worst case scenarios that the algorithm had to operate in. Furthermore, additive Gaussian noise on a data transmission system may be readily analyzed theoretically and studied by computer simulation. Noise is introduced at the output of the transmission path. It has zero mean and a two sided power spectral density σ^2 , giving the zero mean Gaussian waveform $w(t)$ at the output of the receiver filter. Thus, the signal at the output of the receiver filter in Figur e1-1 is:

$$r(t) = \sum_i s_i y(t-iT) + w(t) \quad (1.1)$$

where the $\{s_i\}$ are the values of the transmitted data elements and the $y(t)$ is the impulse response of the equivalent baseband channel. The waveform $r(t)$ is sampled once per signal element at the time instants $\{iT\}$, where i takes on all positive integer values.

The average transmitted energy per signal element is unity for ensuring that any impairments that are generated during transmission are attributed to the transmission medium, not the equipment. Noise components $\{w_i\}$ at the output of the equivalent baseband channel are considered as sample values of statistically independent Gaussian random variables with zero mean and variance σ^2 [3]. At higher rates as mentioned earlier, correlated noise is also considered.

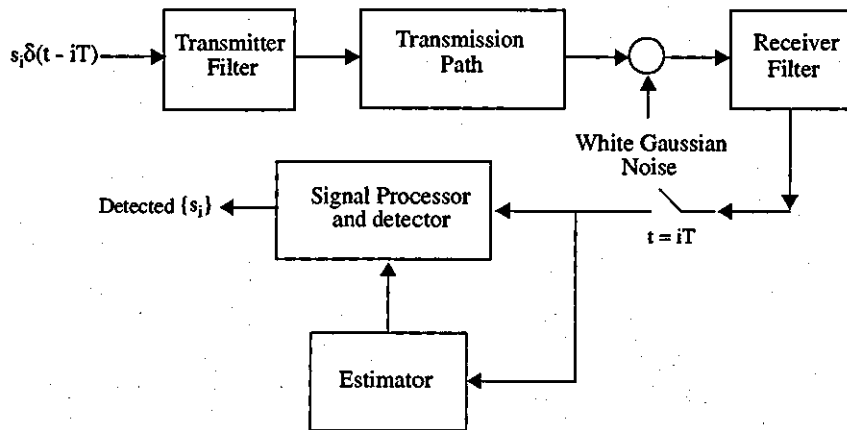


Figure 1-1 Synchronous Serial Baseband Data Transmission System

The sampled impulse response of the channel in Figure 1-1 is given by the $g+1$ component row vector:

$$\mathbf{V} = y_0 y_1 \dots y_g$$

where $y_i = y(iT)$.

The delay in transmission, other than that involved in the time dispersion of the transmitted signal, is neglected here, so that $y_0 \neq 0$ and $y_i = 0$ for $i < 0$ and $i > g$. y is not, in general, the largest component of \mathbf{V} .

Thus, the sample value of the received signal at the output of the baseband channel, at time $t = iT$ is

$$r_i = \sum_{i=0}^g s_{1-i} y_i + w_i \quad (1.2)$$

where $r_i = r(iT)$ and $w_i = w(iT)$.

In the signal processor and detector in Figure 1-1, the values of the $\{s_i\}$ are detected from the $\{r_i\}$ using any one of a number of different operations. The chosen detection process is based on the Viterbi algorithm [4]. The signal processor and detector have prior knowledge of both V and of the possible values of s_i . The average transmitted energy is unity, such that:

$$s_i^2 = 1 \quad (1.3)$$

where s_i^2 is the average value of s_i .

Hence, if $\{s\}$ represents a quaternary signal:

$$s_i = \pm 1/\sqrt{5} \text{ or } \pm 3/\sqrt{5} \quad (1.4)$$

References

1. A.P. Clark and F. McVerry, *Channel Estimation for an HF Radio Link*, IEE PROC. Vol. 128, Part F, No. 1, February 1981.
2. A.P. Clark and F. McVerry, *Performance of 2400 kb/s Serial and Parallel Modems Over an HF Channel Simulator*, IERE Conference Proceedings 49, Loughborough, England, April 1981, pp. 167-179.
3. A.P. Clark, *Advanced Data Transmission Systems*, Pentech Press, 1977.

Chapter 2

Digital Modems for Voiceband HF Radio Links

HF radio links can introduce a combination of multipath propagation and Rayleigh fading, which result in severe frequency selective fading. This makes it difficult to achieve an efficient use of bandwidth in a digital communication system. A conventional modem operates at 2400 bits/s, with the simultaneous transmission of several QPSK (quaternary phase shift keyed) signals. The signal carrier frequencies are regularly spaced over the voice frequency band, and differential coding is applied to each QPSK signal at the transmitter, together with differentially-coherent detection at the receiver [1]. The reason for using a parallel modem is that the signal distortion introduced by the multipath propagation can be removed from each of the parallel QPSK signals, leading to a realizable modem [5].

In principle, a more efficient use of bandwidth, together with a better tolerance to additive noise, should be achieved by a serial system, employing at the receiver both coherent demodulation and near-maximum likelihood detection.

The model of the serial data transmission system for an HF radio link is extended from Figure 1-1 to a more realizable structure as shown in Figure 2-1. The data symbols, $\{S_j\}$ are statistically independent and equally likely to have any of the m given values. Each transmitted signal element carries the corresponding data symbol. In a 2400 bits/s, $m = 4$, $s_j = \pm 1 \pm j$ ($j = \sqrt{-1}$), and the signal element/ baud rate is 1200. The baseband signal generator and linear modulator in Figure 2 convert the input data symbols $\{s\}$ into a serial data stream of QAM (quadrature amplitude modulated) signal elements, with a carrier frequency of 1800 Hz. Each signal element itself comprises the sum of two double side band suppressed carrier amplitude modulated with their carriers in phase quadrature, the "in-phase" and "quadrature" elements carrying the real and imaginary parts, respectively, of the corresponding data symbol s_j . The data symbols are differentially coded, before transmission, and the corresponding detected data symbols at the receiver are differentially decoded.

The HF radio link is assumed to have either two or three independent Rayleigh fading sky waves, with fixed transmission delays but the same average attenuation [2]. The only additive noise considered here is stationary white Gaussian noise with zero mean and a constant power spectral density, which is added to the data signal at the output of the HF radio link. The linear demodulator in Figure 2 filters and demodulates the received signal, using two linear coherent demodulators whose reference carriers have the same fixed frequency and are in phase quadrature. The demodulator removes any constant frequency offset in the received QAM signal, but does not track the variations in instantaneous frequency introduced by the HF radio link, the whole of which therefore appears in the demodulated waveform $r(t)$. The demodulated signals at the outputs of the "in-phase" and "quadrature" coherent demodulators are taken to be real and imaginary valued, respectively, so that the resultant demodulated baseband signal $r(t)$ is complex valued. The sample of the waveform $r(t)$, at a time $t = iT$, is;

$$r_I = \sum_{h=0}^g s_{i-h} y_{i,h} + w_I \quad (2.1)$$

where

$$Y_I = \{y_{I,0} y_{I,1} \dots \dots \dots y_{I,g}\} \quad (2.2)$$

is taken to be the sampled impulse of the linear baseband channel in Figure 2. The data symbols $\{s_i\}$ are here considered as the corresponding stream of impulses $\{s_i \delta(t-iT)\}$ and the baseband signal generator as an appropriate linear lowpass filter. The real and imaginary parts of the $\{w\}$ are Gaussian random variables with zero mean and fixed variance, which are either statistically independent or slightly correlated with the neighboring $\{w\}$ since the $r(t)$ are sampled fairly close to the Nyquist rate.

A wide range of different near maximum likelihood detectors have been studied and described in published literature [3, 4, 5]. A detailed description of the selected detector for 2400 bits/s transmission rate is presented in Chapter 8.

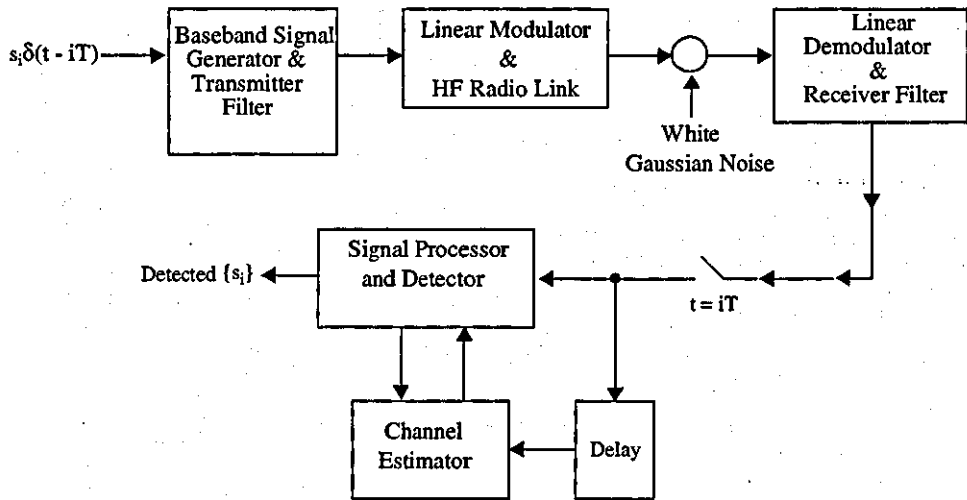


Figure 2-1 Model of a Serial Data Transmission System for an HF Radio Link

References

1. A.P. Clark and F. McVerry, *Performance of 2400 b/s Serial and Parallel Modems Over an HF Channel Simulator*, IERE Conference Proceedings 49, Loughborough, England, April 1981, pp. 167-179.
2. F. McVerry, Ph.D. Thesis, High Speed Data Over HF Radio Links, May 1982.
3. A.P. Clark, J.D. Harvey and J.P. Driscoll, *Near Maximum Likelihood Detection Processes for Distorted Digital Signals*, Radio and Electronic Engineering 1978, 48, pp. 301-309.
4. A.P. Clark, C.P. Kwong and J.D. Harvey, *Detection Processes for Severely Distorted Digital Signals*, IEE Journal of Electronic Circuits and Systems, 1979, 3, (1), pp. 27-37.
5. A.P. Clark and M. Fairfield, *Detection Processes for a 9600 b/s Modem*, Radio and Electronic Engineering, 1981, 51, pp. 455-465.

Chapter 3

The HF Channel

Section 3.1 Types of Distortion Occurring in HF Channels

Perhaps the most obvious form of distortion which occurs in HF radio transmission is reception via reflection from several different ionized layers or else via both one and two (and possibly more), when transmitting over very long distances. [1, 5, 6]. The example of a typical practical case is given in Figure 3-1. This example represents the situation on a 1000 km link on a winter's day. The critical frequency and height values quoted are typical for the layer at that time. From these parameters, the MUFs (maximum usable frequency) for each layer are calculated from the following equation:

$$\text{MUF} = \delta_c \text{ Sec } \theta_i. \quad (3.0)$$

Normal operating practice on such a link is to use transmission frequency, called the optimum working frequency (CWF) which is given by [2];

$$\text{CWF} = 0.85 * \text{MUF}. \quad (3.1)$$

The MUF in equation 3.1 is the highest transmissible frequency over the link and for the given example it is 21.36 MHz and uses the F2 layer reflection. The CWF for the example is therefore 18.16 MHz. If the CWF is used as the transmission frequency, this will result in single path reception because it exceeds the MUFs for the E and F1 layers. However, such flexibility in operating frequency is not always possible in practice, so it is interesting to consider the situation at other possible transmission frequencies. (see Table 1). The use of frequencies below 10.84 MHz for this link would be avoided in practice not only because of the severe multi-path but because signal attenuation is proportional to $\frac{1}{f^2}$, resulting in a badly distorted signal with low signal to noise ratio. One further characteristic of interest can be calculated for this example, this is the time spread or multi-path spread of the signal at the receiver [5, 7, 8]. If the transmitter sends a short pulse shown in Figure 3-6, the time spread of the received signal is the time between the reception of the first and the last pulses as shown in Figure e3-6, for the different cases given in Table

1. Other discrete paths may be presented in the received signal due to ray splitting. Ray splitting occurs in the F, F1 and F2 layers but is dependent on the orientation of the electron motion set up by a wave in the ionized layers and the Earth's magnetic field. It is also very dependent on operating frequency, the most noticeable splitting occurring at frequencies below the MUF of a layer. Merging of the two rays then takes place if operating frequency is further reduced. When transmission occurs via, say, both one and two hops, such larger time spreads than that shown in Figure 3-6 can be obtained, time spreads of over 10 ms have been measured over long distance HF radio links [6].

Another form of distortion which occurs on HF radio links is fading [1, 6-10]. This phenomenon can be divided into two types, long and short term fading. Long-term fading is the daily or seasonal variation in the received signal strength due to the night/day seasonal changes in the structure and ionization of the various layers. However, of much greater importance here is the short-term fading which can be subdivided into two distinct types, called selective (or multi-path) and Rayleigh fading [8, 10]. Selective fading is a function of the multi-path present in the received signal and its effect is dependent on the different frequency components which form the received signal. An example of the effect of the selective fading is shown in Figure 3-7. Here, the signal is shown to be a group of 21 sine wave carriers 100 Hz apart. Two equal strength paths are received and the path length difference is 300 km, corresponding to a differential delay of 1 ms. Constructive interference of the two paths occurs when the path length difference represents an even number of half wavelengths of the received signal. Destructive interference occurs for an odd number of half wavelengths. This results in a distorted signal being received (Figure 3-7), and the effective amplitude characteristic of the channel is given in Figure 3-7. The example shown in Figure 3-7 is a simplification of the practical channel, which has time varying characteristics. The time varying nature of the actual HF channel is due to the phenomenon of Rayleigh fading. There are several different effects which produce this type of fading [8], but, essentially, they are all functions of the short-term variations in the ionization of the reflecting layers and the variation in the position of these layers. Rayleigh

fading is that fading that occurs on the one or more paths that constitute the received signal. If one of these paths is considered in more detail at the point of reflection in the ionized layer, the situation at one point in time can be represented as shown in Figure 3-8. The signal received from this 'single' path is made up of the sum of several slightly different paths, all adding randomly at the receiver. Short term variations in the ionization of the layer alter the way these paths add at the receiver. If the transmitter is simply a sine wave carrier of fixed frequency and amplitude as shown in Figure 3-9 then the received signal will typically have the form shown in Figure 3-9. The amplitude distribution of the envelope of the received signal has a Rayleigh distribution from which the name of the fading is derived. The effect of this fading on the two equal paths of Figure 3-7 is to alter the positions and depth of the nulls (or selective fades) of the effective amplitude characteristic.

The occurrence of other fading characteristics has been found in practice, such as the Nakagami-Rice (or simply Rician) probability distribution [8, 11], which is a Rayleigh distribution with a specular (non-fading) component as a result of (say) direct ground wave reception as well as the sky wave paths. Variations in the positions of the reflecting layers also contribute to this fading, especially in the F layer variations as shown in Figure e3-10 for a typical June day [1]. Clearly from Figure e3-10, there are certain times of the day, notably from 5 a.m. until 8 a.m and from 4 p.m. to 7 p.m., when the reflecting layers are moving rapidly in one direction – some 50 km/hour for the F2 layer during the evening. This movement would produce a distinct Doppler shift on a received signal via this layer. The magnitude of the shift is, of course, frequency dependent but at a typical operating frequency of 15 MHz, the shift would be about 1 Hz. The Rayleigh fading characteristics on the different multi-paths present at a receiver tend to be uncorrelated, this being due to the fact that the fading is caused by localized variations in the ionized layers at the point of reflection.

In conclusion, this section has shown that a signal received via an HF channel is composed of one or several discrete multi-path components, each component being a faithful representation of the original

signal but distorted by fading and this fading is generally uncorrelated with the fading present on the other components. The amplitude/frequency and phase/frequency responses conventionally used to characterize communication channels have been shown to be an inappropriate measure for time varying HF channels where the characteristics are continuously varying. The next section will address the issue of HF channel characteristics.

Section 3.2 Characteristics of HF Channels

Since conventional methods of channel characteristics are inappropriate for HF channels, the quasi-static characteristics [6] of multi-path spread and fading are used instead. 'Quasi-static' characteristics are defined as characteristics which remain fixed or vary imperceptibly slowly when transmissions lasting up to several minutes are made over the channel. For a different transmission over the same channel, these characteristics may have changed significantly but would remain fixed or vary very imperceptibly slowly for this transmission.

The multi-path characteristic of an HF channel is the time difference between arrivals of the first and last paths at the receiver as explained in the previous section. This parameter is known variously as multi-path delay spread [7], time spread [10], differential time delay [10, 13], and multi-path spread [9] and hereafter will be known as the latter. Practical tests have shown that this parameter can vary from 0 ms (single path) to 2.5 ms [5] for long distance transmissions using optimum working frequencies (Figure 3-11). These tests were performed over a period of four years using some 4000 different transmissions as data, the transmission links being between London-Moscow, London-Melbourne and London-New York [5]. Where frequencies lower than optimum are used, multi-path effects become more severe as was shown in the previous section and experimental results obtained in this situation are shown in Figure 3-12. These results are based on 1600 transmissions for the period June-September 1961 [5].

The second parameter used for channel characterization is fading rate, but before discussing and quantifying rapidity of fading, it is appropriate here to discuss severity of fading [8] which at first sight

may appear to be a possible third characterizing parameter of HF channels. The assumption that the type of fading occurring on a single HF path has a Rayleigh distribution means, for example, that amplitude, u , of the envelope of a single unmodulated sine wave carrier frequency received via this path (Figure 3-13) has a probability density function:

$$\Gamma(v) = (2v/v_n^2) \exp(-v^2/v_n^2) \quad (3.2.1)$$

This is the Rayleigh probability distribution where v_n is the r.m.s. voltage of the fading signal. The proportion of time that the fading signal exceeds a value v_0 is given by the cumulative distribution function:

$$P(v_0) = \int_{v_0}^{\infty} \Gamma(v) dv \quad (3.2.2)$$

Equation (3.2.2) has been evaluated for some numerical cases in Table 1 which shows the percentage of time ($P(v_0)$) that a level relative to the median level of the fading signal is exceeded.

The severity of the fading on a single path is therefore fixed by the fact that the fading is characterized by a Rayleigh distribution. This is distinct from the severity of selective, which is a function of the multipath present in the received signal as shown previously. Clearly, the worst case selective fading occurs if two equal strength paths are present in the received signal, producing nulls of infinite attenuation at certain frequencies—the number and location of the nulls in the signal passband being a function of the multipath spread. If the Rayleigh fading is present on both paths and there is no correlation between the fading on each path, the depth and location of the nulls will vary with time. However, the selective fading is still at its most severe if the r.m.s. values of the fading amplitudes on the two paths are equal.

Rapidity of fading is determined by the channel auto-correlation function in time or its corresponding power density spectrum [8]. It has been shown [8] that this auto-correlation is generally in the form:

$$R(t) = R(0) \exp(-t^2/2t_0) \quad (3.2.3)$$

This Gaussian auto-correlation function means that the power density of the fading is also Gaussian with a standard deviation of $1/t_0$, known as the correlation (coherence) bandwidth. Fading rapidity can

also be expressed as the number of positive crossings per unit time through any specified level. If the fading has a Rayleigh distribution, the correlation bandwidth f_s is related to the fading rate, f_r , as follows:

$$f_r = 1.475 f_s \quad (3.2.4)$$

Finally, referring to Figure 3-9, the total frequency spread of the faded signal is twice the correlation bandwidth of the fading component which modulates the wanted signal, so that a frequency spread of 1 Hz corresponds to a fading rate of 44 per unit. Typical fading rates for HF channels are between 6 and 16 per minute.

A useful technique for displaying the time varying characteristics of an HF channel is the fadeogram [9]. This can serve as a useful aid to characterizing practical channels. An example of a fadeogram is given in Figure 3-13. It shows the variation of a channel's amplitude characteristics with frequency and time. Amplitude variations are shown as variations in intensity on the display, the lighter the display the greater is the attenuation at that point. The fadeogram of Figure 3-13 represents a two path signal (after demodulation to the voice band) with a Doppler shift of 0.1 Hz on one path and a multipath spread of 0.67 ms.

Several attempts have been made at HF channel characteristics [9, 10]. The USAEL classification [9] is given in Table 3-3. Here, fading rate is given as the speed of propagation of a selective fade through a 3 kHz band. Another classification made by CCIR [10] is given in Table 3-4. These are suggested parameter values for the testing of HF radio communication equipment on an HF channel simulator. The latter models HF channels as two independently fading paths with equal amplitude in the absence of fading, equal frequency spreads and no frequency shifts. The HF channels used for test purposes in this thesis are based on the CCIR conditions and are given in Table 3-5. The same two path model is assumed, but conditions for flutter fading have been altered because the CCIR conditions represent extremely disturbed conditions produced by ionospheric storms [1]. The flutter fading conditions given in Table 3-5 represent a fading rate of 88 per minute, which is five times worse than typical. The multipath spread has been

increased to 3 ms. These spreads are possible on very long distance links. This multi-path spread means that there will on average be ten selective fades in a 3 khz signal bandwidth.

Section 3.3 Model of HF Channel

There are two methods available for testing the performance of a transmission system for use on HF radio channels [10]. First, the constructed equipment can be used over actual HF channels and its performance evaluated by error rate measurements. This method of testing has two main disadvantages. First, it is difficult to ascertain the weaknesses of the equipment from its performance because poor performance may have one or several causes such as impulsive noise, fading rate, multipath or Doppler shift. Secondly, if one system is being compared with another, it is impossible to test both systems over exactly the same channel as it is continuously changing over time. The only way of comparing the system is to perform a large number of test transmissions and find the average performances, but, again, this gives no information about the relative strengths and weaknesses of the system in dealing with different forms of HF channel distortion.

The alternative to testing over real channels is to test over a channel simulator [10-14]. This is a device that models real channels and simulates the distortion found on such channels. The most important properties of simulators are control of distortion and repeatability. Control of distortion means that weakness of transmission systems can be isolated by introducing only one or two forms of distortion and reducing or omitting other forms of distortion. Repeatability means that a given channel with a certain time varying sequence can be obtained as often as required so that two or more different systems can be compared on identical time-varying channel conditions.

Many simulator designs exist and the more relevant ones have been included in the reference [11, 12]. These are baseband simulators, which means that the HF radio transmitter and receiver modulation and demodulation processes are assumed perfect and linear and therefore can be omitted, leaving only the signal from the equipment under test, available for processing in the simulator. The main advantage of base-

band simulators is that the input signal extends over a few kilohertz only; so modern digital techniques can be employed in their design, thus allowing very accurate and repeatable characteristics to be obtained. Hardware simulators would obviously require the existence of hardware equipment to carry out the tests. We, however, modeled the equipment and the HF channel in software to develop detection processes. This in turn allows considerable flexibility and scalability in the development and understanding of the entire system in a computer simulation environment and from an academic point of view the least expensive.

The channel model used in the computer simulation is given in the block diagram form in Figure 3-15. This is also the model used in the hardware simulators mentioned earlier. This model conforms to the CCIR requirement for a simulator which should have two independently Rayleigh fading multipath components of equal mean level [10]. The additive noise used in the model has a Gaussian probability density function although, on actual channels, this additive noise term is composed of contributions from several sources and can highly impulsive in nature [1-5]. For HF radio links naturally occurring, additive noise is mainly "atmospheric noise," which is near Gaussian. Man-made noise is generally impulsive noise. A good tolerance to additive Gaussian noise is a very good indication of good tolerance to atmospheric noise. Stationary, additive white Gaussian noise is assumed because it is the standard and generally the most reliable model of the actually occurring noise, and it can be easily measured. If a continuous sine wave signal, $V \sin w_c t$, is fed to the input of this channel model, its output is given by:

$$v_0(t) = v(B_1(t) \sin(w_c t + \theta_1(t)) + B_2(t) \sin(w(t+T) + \theta_2(t)) + v_n(t) \quad (3.3.1)$$

$B_1(t)$ = Randomly time-varying amplitude which has a Rayleigh probability Density function.

$\theta_1(t)$ = Randomly time-varying phase with a uniform probability density function in the range

0-2 π

$B_2(t)$, $\theta_2(t)$ = as $B_1(t)$, $\theta_1(t)$ but uncorrelated with them.

t = Multipath spread

$v_n(t)$ = Gaussian noise voltage

A digital implementation of the HF channel model in a computer program means that it is neither possible nor necessary to represent the fading signals $N_i(t)$ as continuous. These signals must be represented as discrete samples in time. Exploiting the Nyquist sampling theorem, the minimum sampling rate required to adequately represent $N_i(t)$ is twice the highest frequency contained in them. As the fading signals have Gaussian spectra, they theoretically contain all frequencies. For example, a 2 Hz frequency spread could be adequately represented by a sampling frequency of 10 Hz without any aliasing occurring. However, in computer simulations involving the HF model used in the computer simulations, the minimum sampling rate required to represent the $N_i(t)$ is determined by the sampling rate of the transmission/detection systems under test. Our systems were operating at 2.4 kbits/s data rate or 1200 samples per second for a QPSK constellation. Each sample has to be modified by the fading; therefore a sampling rate of 1.2 kHz for the $N_i(t)$ is adequate.

Table 3-1 Effect of Transmission Frequency on Multipath

Transmission Frequency MHz	Number of Paths Received
>21.36	0
14.4 – 21.36	1 (F2)
10.84 – 14.4	2 (E and F2)
<10.84	3 (E, F1 and F2)

Table 3-2 Theoretical evaluations of the percentage of time that a level relative to the median level of the fading signal is exceeded

Level relative to median level of fading signal (dB)	Percentage of time Level exceeds median level (%)
8.22	1
5.21	10
0	50
-8.18	90
-18.39	99

Table 3-3 U.S.A.E.L. Classification of HF Channels

Fade rate through 3 kHz band	Multipath Spread		
	Mild 0-0.4mS the selective fade	Medium: 0-4.1 mS. Two but not three selective fades	Severe: ≥ 1 mS Three or more selective fades
Slow: 10 secs or more	A - 1	B - 1	C - 1
Medium: 2 - 10 secs	A - 2	B - 2	C - 2
Fast: 2 secs or less	A - 3	B - 3	C - 3

Table 3-4 C.C.I.R Classification of HF Channels

(1) Good Conditions Multipath spread : 0.5 mS Frequency spread : 0.1 Hz
(2) Moderate Conditions Multipath spread : 1 mS Frequency spread : 0.5 Hz
(3) Poor Conditions Multipath spread : 2 mS Frequency spread : 1 Hz
(4) Flutter fading (if required) Multipath spread : 0.5 mS Frequency spread : 10 Hz

Table 3-5 Characteristics of the Three HF Channels Used in Subsequent Computer Simulations

Channel Number	Channel Description	Frequency Spread (Hz)	Multipath Spread (mS)
1	Moderate	0.5	1.0
2	Poor	1.0	2.0
3	Flutter Fading	2.0	3.0

Table 3-6 Filter Tap Values to Obtain the Required Frequency

Frequency Spread (Hz)	T ₁	T ₂	T ₃	T ₄	T ₅
0.5	-1.9	0.903135	-1.9276	0.9316561	-0.946
1	-1.8036	0.8155376	-1.8524	0.8680474	-0.895
2	-1.6218	0.6650064	-1.6954	0.753639	-0.801

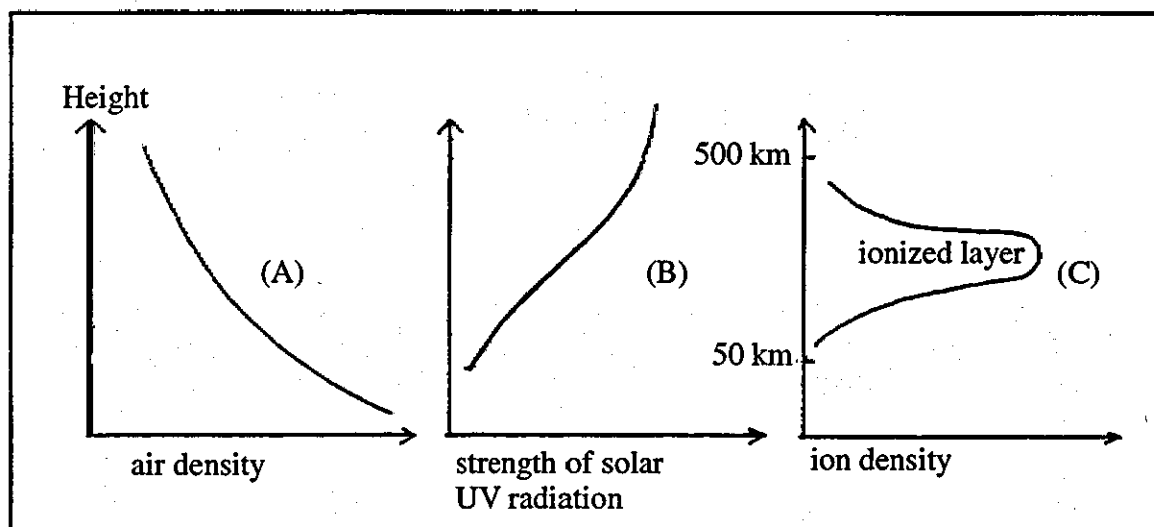


Figure 3-1 Formation of an Ionized Layer

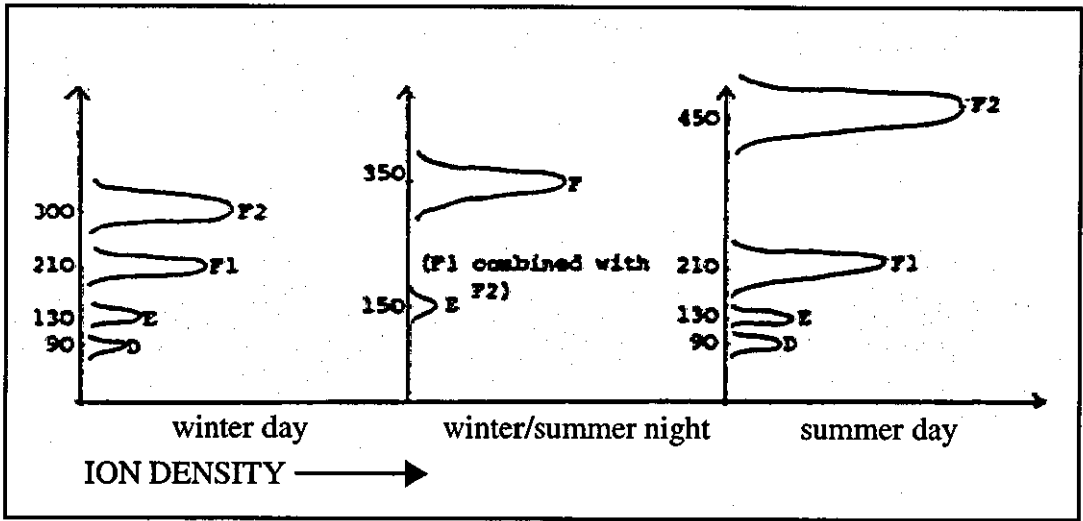


Figure 3-2 Typical Ionized Layer Profile

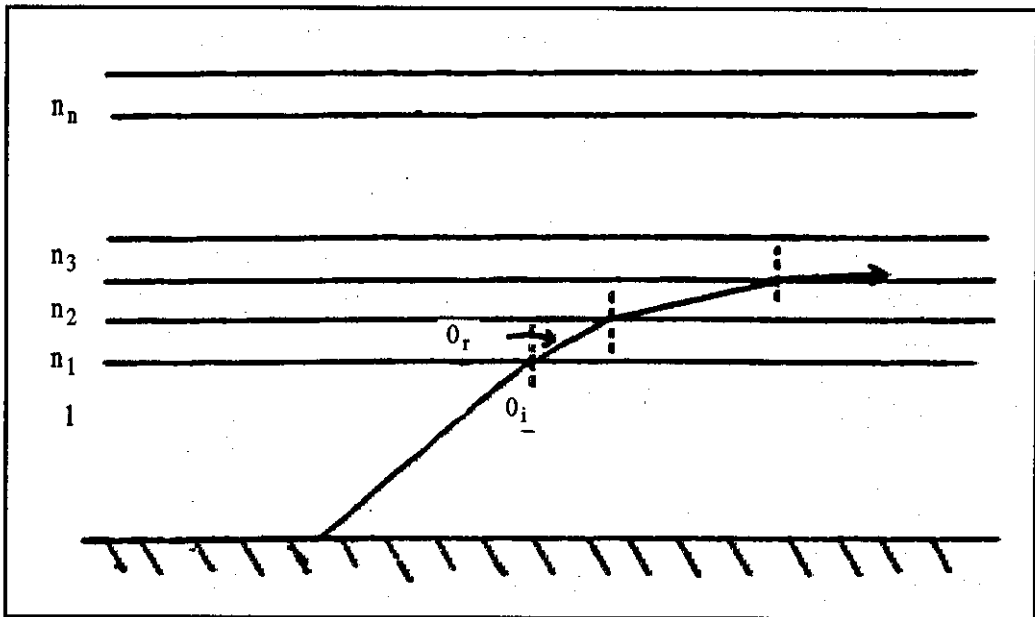


Figure 3-3 Refractive Bending in an Ionized Layer

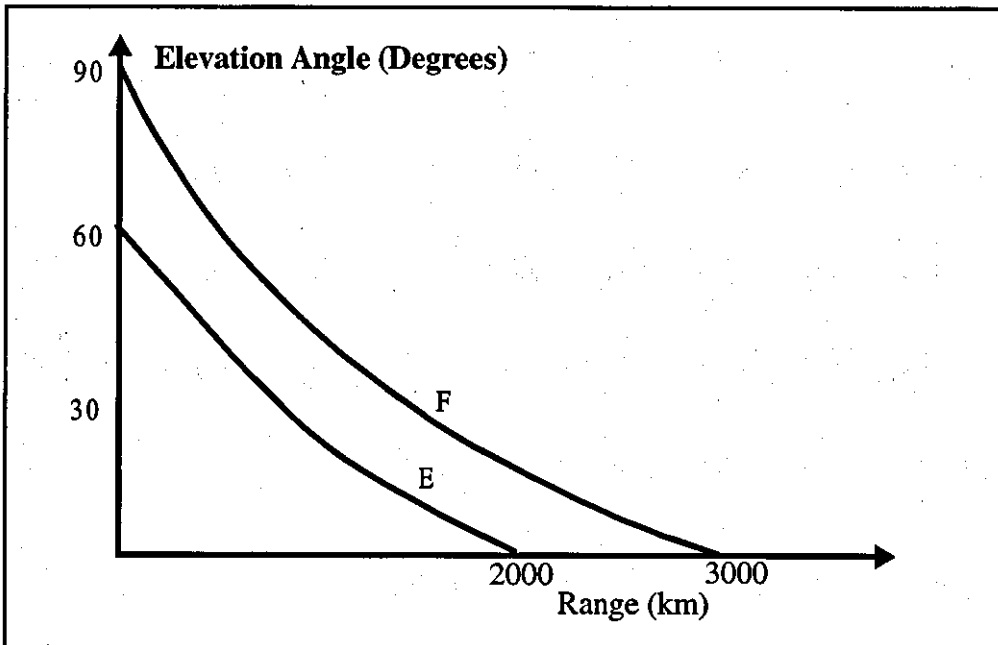


Figure 3-4 Effect of Elevation Angle on Range

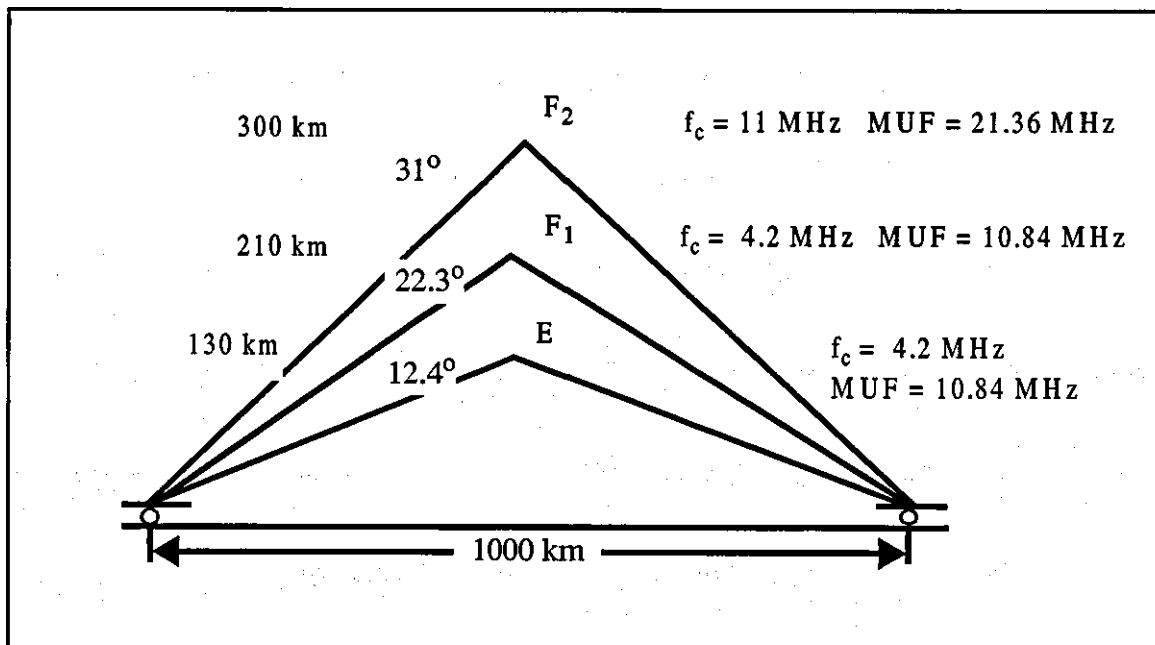


Figure 3-5 Example of Multipath Reception

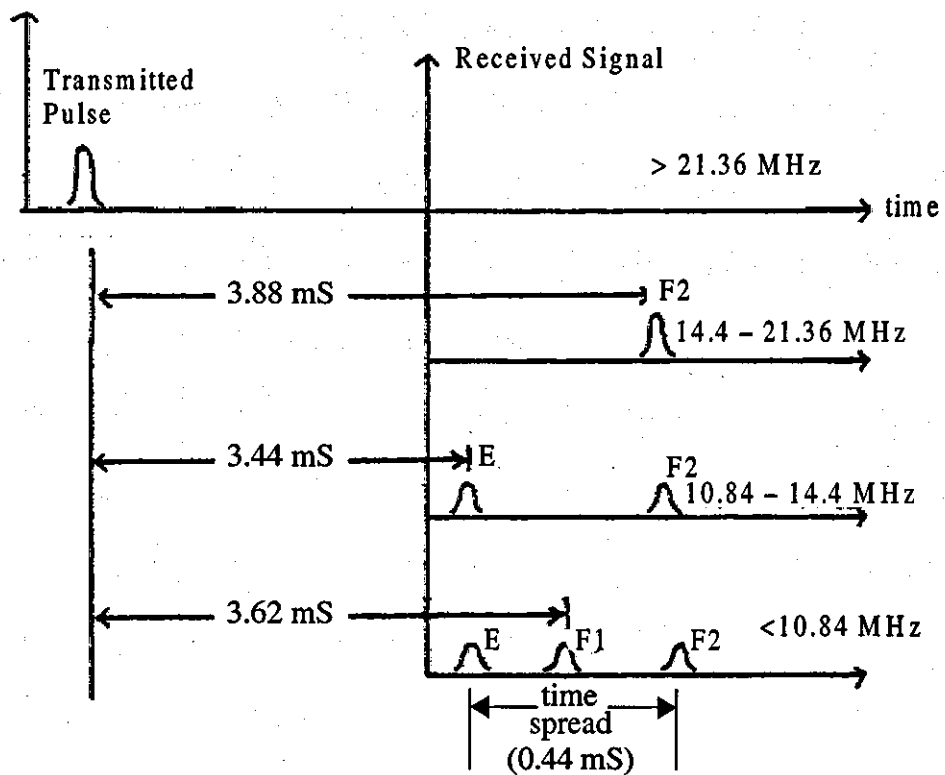


Figure 3-6 Impulse Response of Multipath Channel

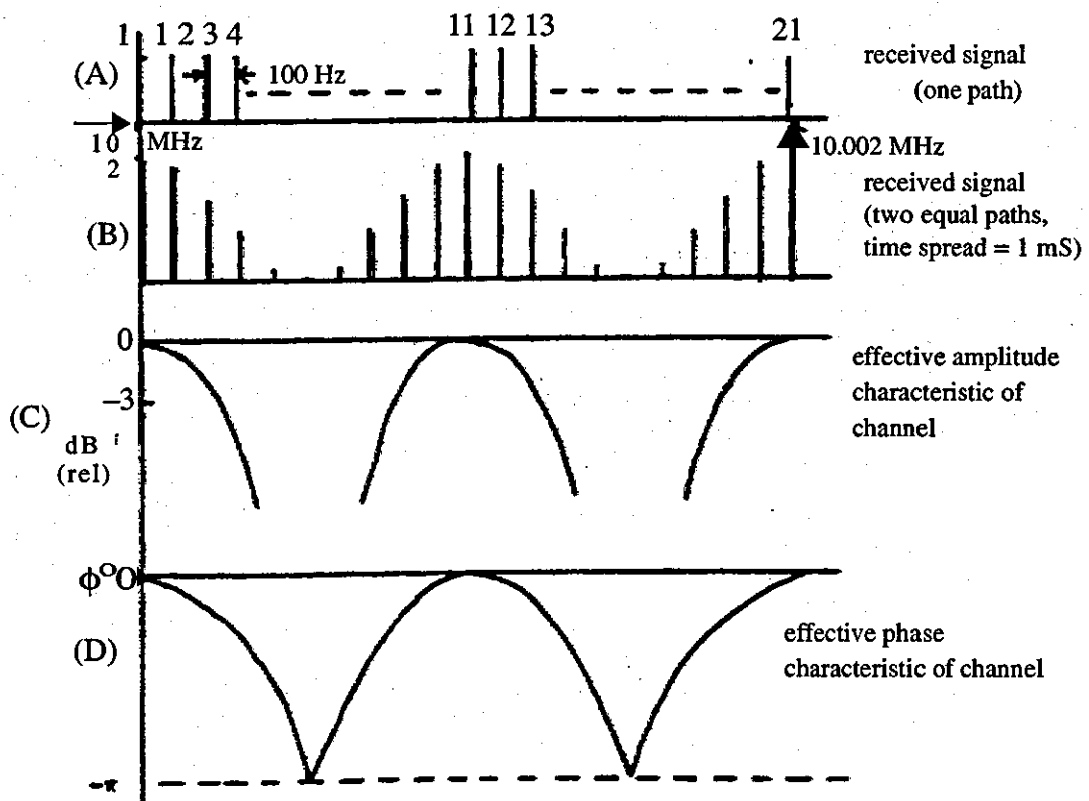


Figure 3-7 Frequency Characteristics of a Multipath Channel

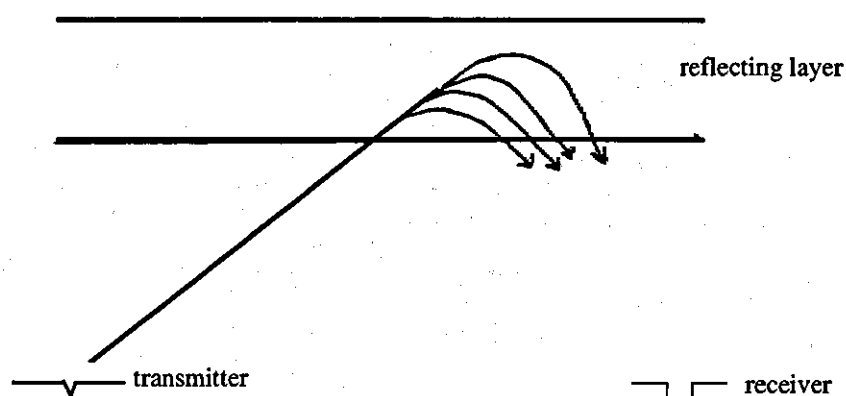


Figure 3-8 Rayleigh Fading

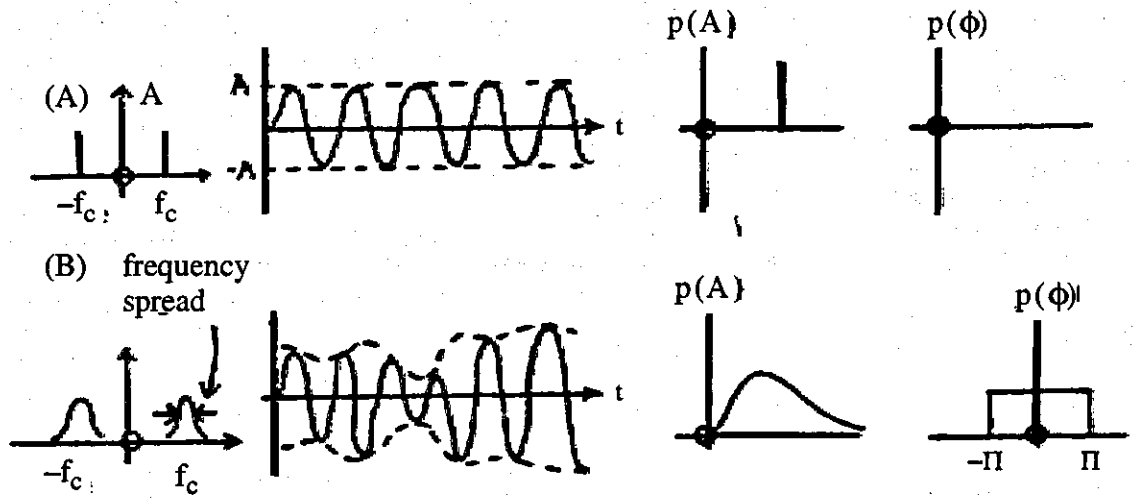


Figure 3-9 Characteristics of a Rayleigh-Faded Signal

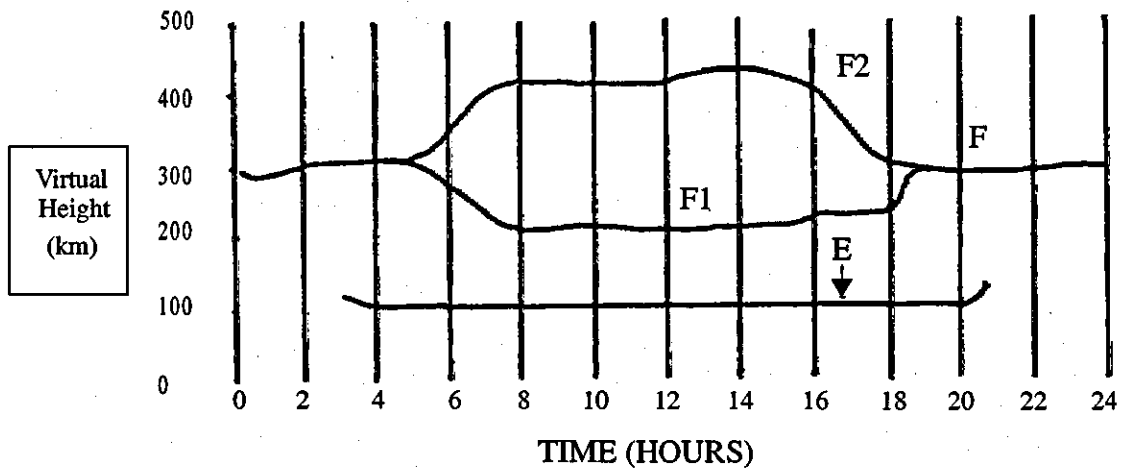


Figure 3-10 Diurnal Variation of Layer Heights

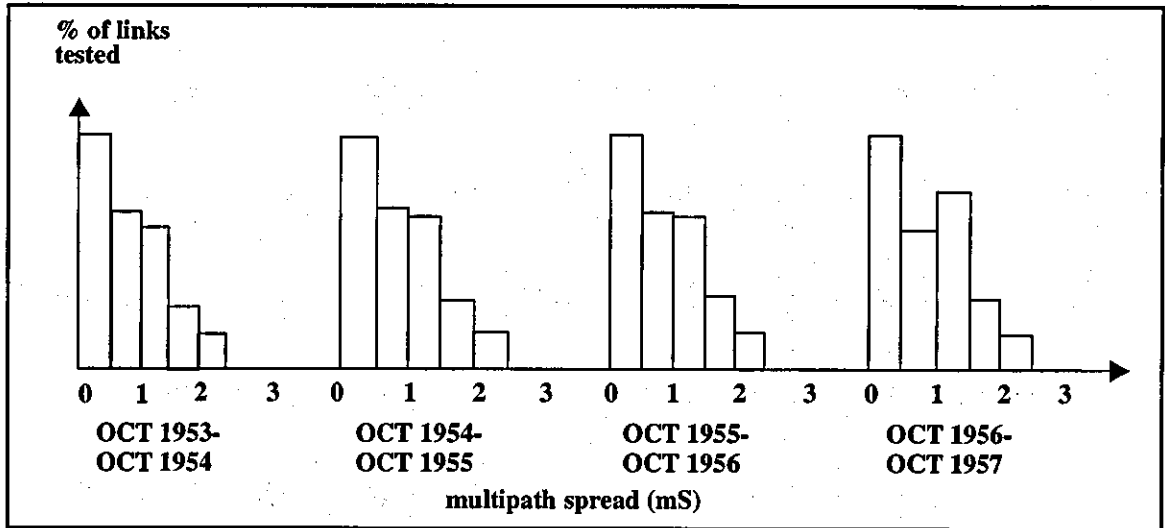


Figure 3-11 Multipath Spread Occurring on Actual Links Where the Transmission Frequency is Well Below MUF

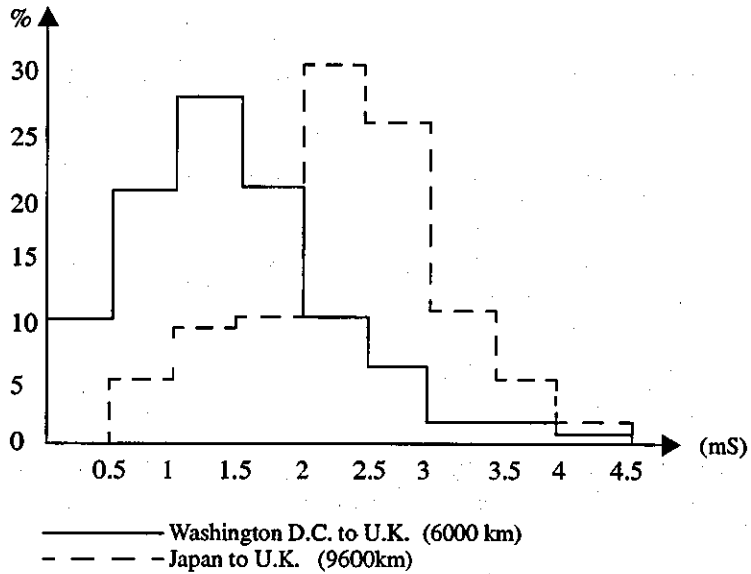


Figure 3-12 Multipath Spread Occurring on Actual Links Where the Transmission Frequency is Well Below MUF

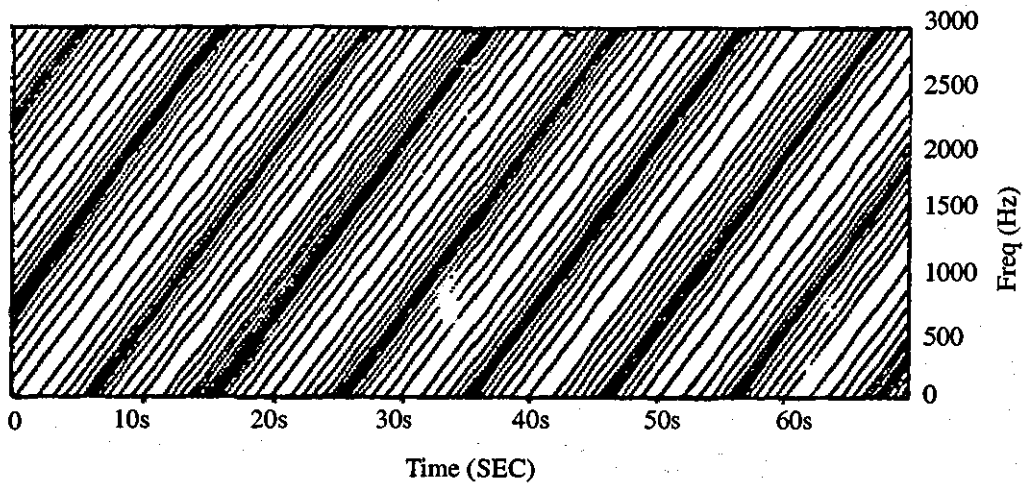


Figure 3-13 Fadeogram of a Typical HF Channel

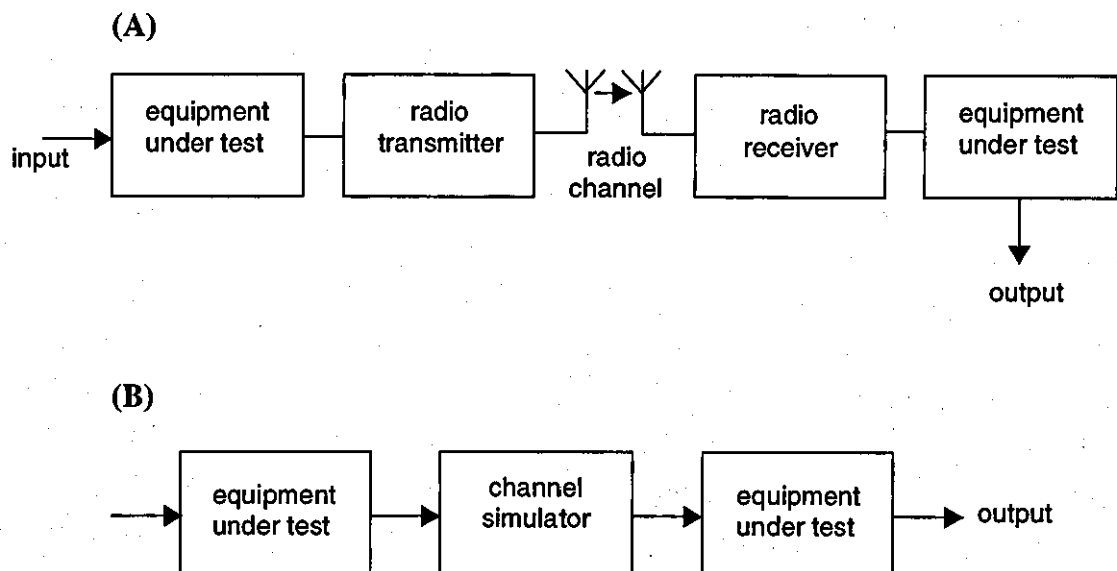


Figure 3-14 Baseband HF Simulator

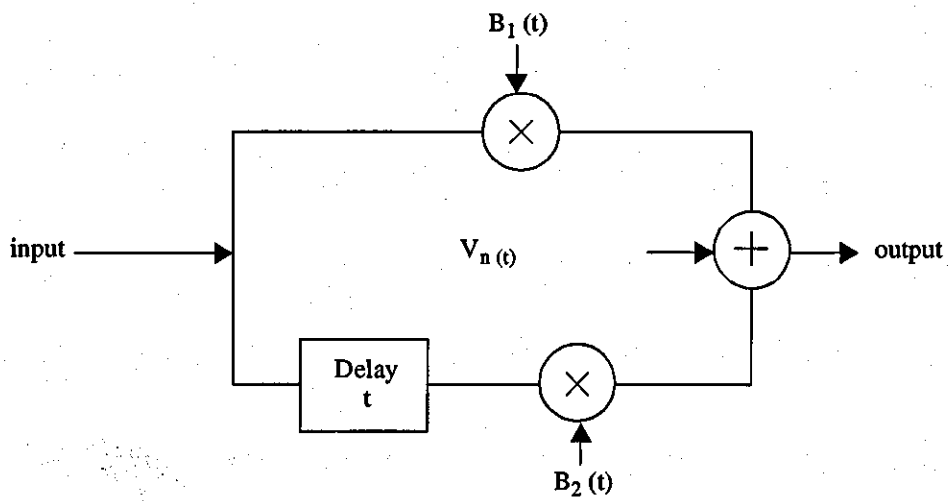


Figure 3-15 Block Diagram of the HF Channel Simulator Employed in the Computer Simulations

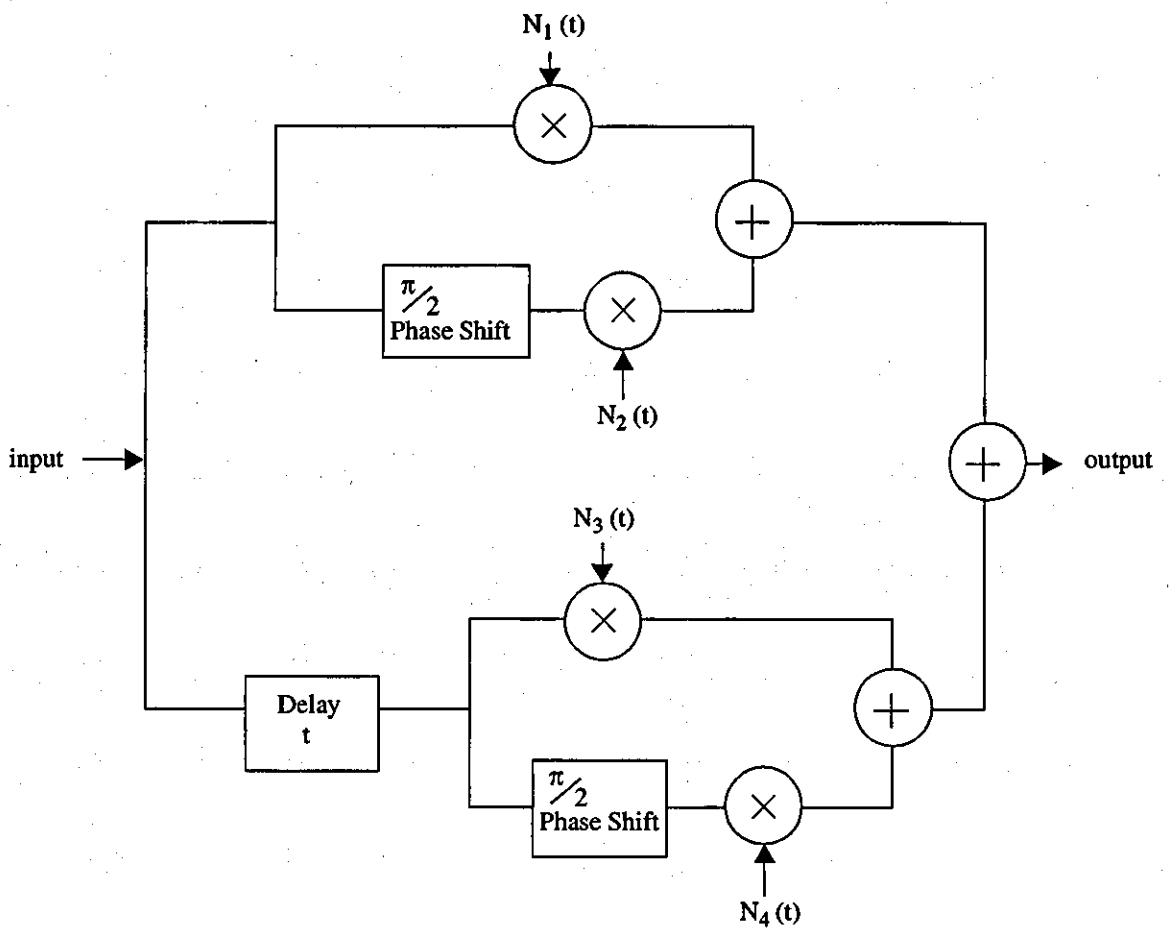


Figure 3-16 More Detailed Diagram of the HF Channel

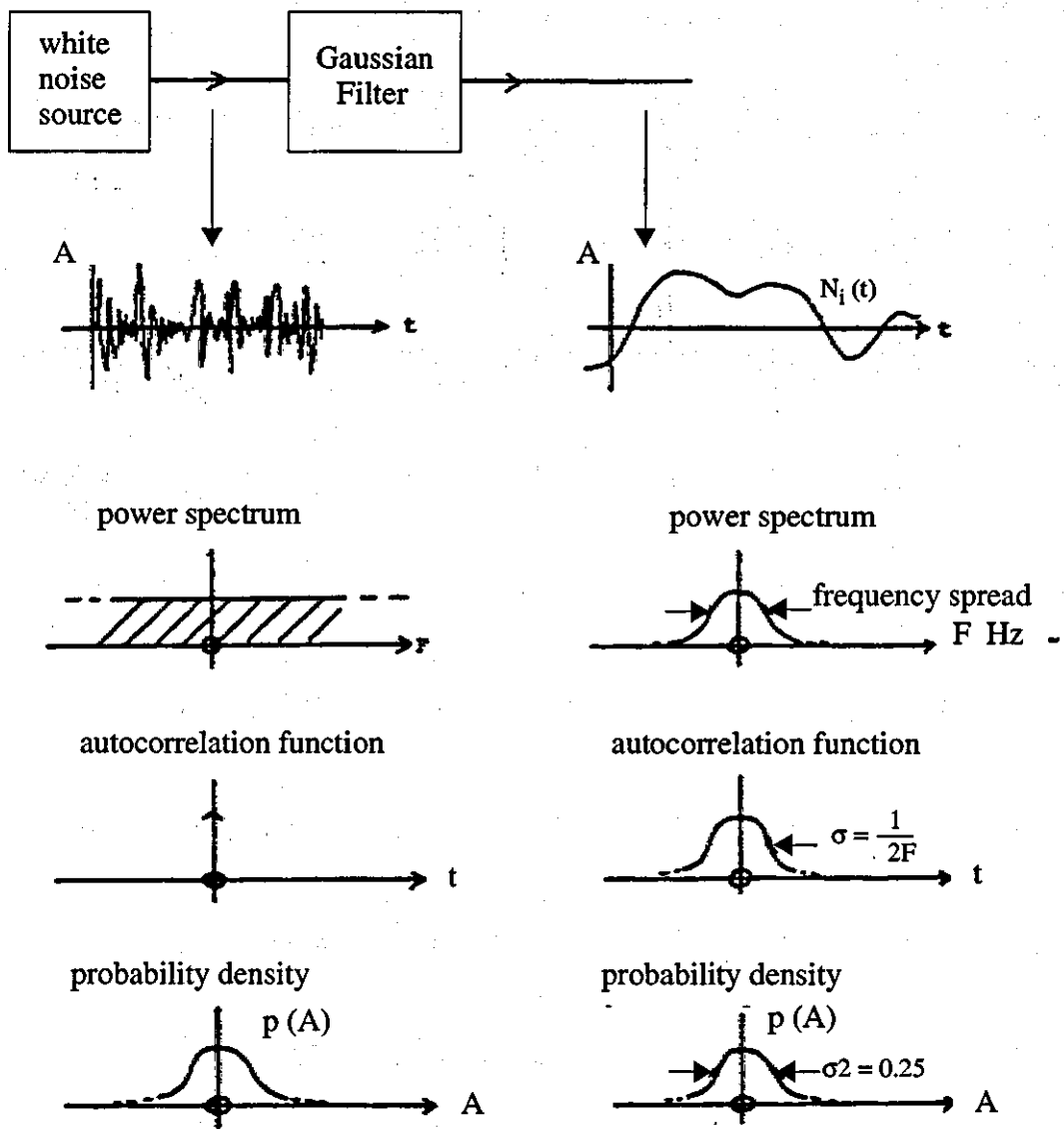


Figure 3-17 The Linear Interpolation Process

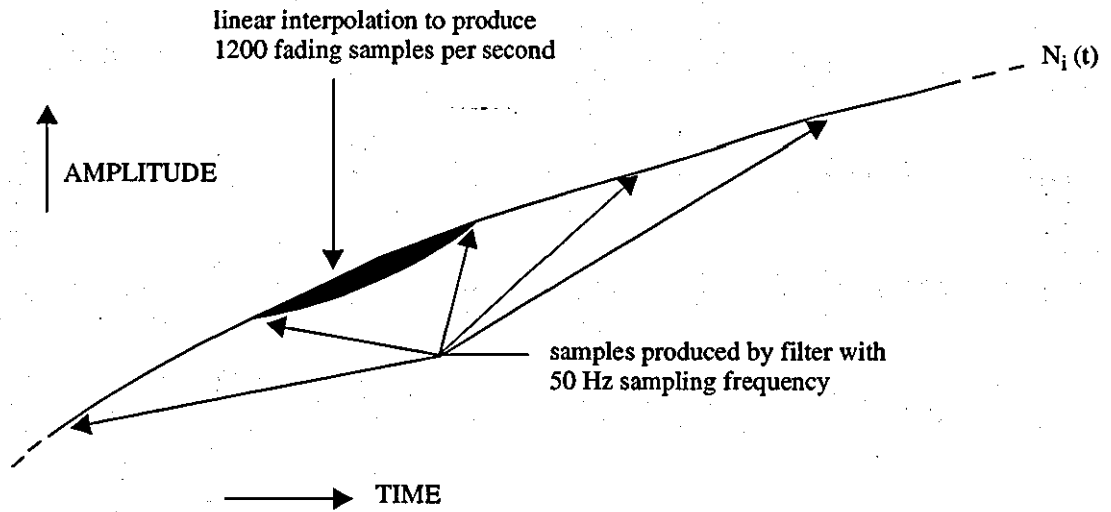


Figure 3-18 The Linear Interpolation Process

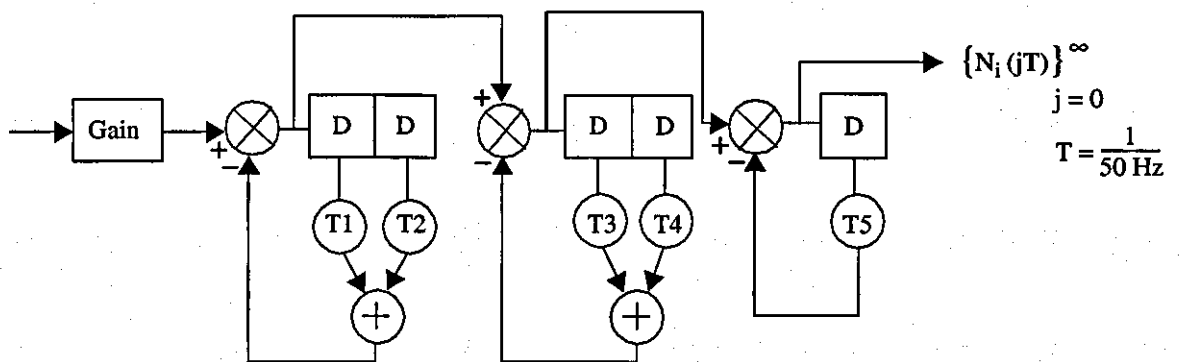


Figure 3-19 Block Diagram of the Filter Used for Generation of Fading Samples

References

1. Benning, T. W., *Short Wave Radio and the Ionosphere*, Iliffe and Sons Ltd., Second Edition, 1950.
2. Hills, M. T. and Evans, B. G., *Transmission Systems*, Allen and Unwin, 1962.
3. Davies, K., *Ionospheric Radio Propagation*, U. S. Department of Commerce, 1965.
4. *Radio Science*, Journal of the U. S. National Committee, International Union of Radio Science. Published by American Geophysical Union.
5. *Multipath Propagation of HF Radio Circuits*, CCIR, XIIIth Plenary Assembly, Vol. III, Report No. 203, pp 49-51, 1974.
6. Goldberg, B., *300 kHz - 300 MHz MF/HF*, IEEE Trans. on Communication Technology, Vol. COM-14, pp 767-784, December 1966.
7. Monsen, P., *Fading Channel Communications*, IEEE Communications Magazine, January 1980.
8. *Ionospheric Propagation Characteristics Pertinent to Radio Communication Design*, CCIR, XIII Plenary Assembly, Vol. VI, Report No. 266-3, pp 207-217, 1974.
9. Filter, J. H. J., Arazi, B. and Thompson, R. G. W., *The Fadeogram - a Sonogram-like Display of the Time-Varying Frequency Response of HF-SSB Radio Channels*, IEEE Trans. on Communication, Vol. COM-26, No. 6, pp 913-917, June 1978.
10. *HP Ionospheric Channel Simulators*, CCIR, XIII Plenary Assembly, Vol. III, Report No. 549, pp 66-75, 1974.
11. Shaver, H. N., Tupper, B. C. and Lomax, J. B., *Evaluation of Gaussian HF Channel Model*, IEEE Trans. on Communication Technology, Vol. COM-15, No. 1, pp 79-88, February 1967.
12. Matley, W. and Bywater, R. E. H., *A Digital High-Frequency Multipath Propagation Simulator*, The Radio and Electronic Engineer, Vol. 47, No. 7, pp 305-314, July 1977.
13. Ralphs, J. D. and Sladen, F. M. E., *An HF Channel Simulator Using a New Rayleigh Fading Method*, The Radio and Electronic Engineer, Vol. 46, No. 12, pp 579-587, December 1976.
14. Ream, N., *Simulation of Rayleigh Fading by Digital Generation of Amplitude and Phase*, The Radio and Electronic Engineer, Vol. 48, No. 11, pp 567-572, November 1978.
15. Kuo, F. F., *Network Analysis and Synthesis*, Wiley, 1966.

Chapter 4

Transmitter and Receiver Filters for the 2.4 K Bits/Sec 4-Point QAM H. F. Modem

(A) Requirements of Filtering

The combined transmitter and receiver filters should have a sampled impulse response which has as short a duration as possible.

The receiver filter should give adequate attenuation of unwanted components which may be present in the signal, e.g. unwanted mains frequency components at 50 Hz and odd harmonics to 250 Hz. There may also be tones present at the high frequency side of the signal, for example 4 kHz.

(B) Filter Design

The ideal equivalent baseband sampled impulse response of all the combined transmitter/receiver filters is a delta function (see Figure 4-1a).

In earlier computer simulations, a more practical raised cosine impulse response was assumed which gives the sampled values.

$$0 \ 1 \ 0 \qquad \qquad \qquad (4.1)$$

when sampling phase is correct (see Figure 4-1b). A perfect raised cosine pulse has a frequency spectrum which extends to infinity, but it can be closely approximated by a Gaussian impulse response which gives the sampled values

$$\dots\dots\dots x \ 1 \ x \ \dots\dots\dots \qquad \qquad \qquad (4.2)$$

when sampling phase is correct (see Figure 4-1c). When $x \leq 0.01$ this is a good approximation to the raised cosine pulse. All other samples $\ll x$.

Given a required sampled Gaussian impulse response, an expression can be obtained for the continuous impulse response in the form,

$$f(t) = \exp\left(\frac{-\pi t^2}{T^2}\right) \quad (4.3)$$

T = sampling period.

From this, the required filter characteristic which has this impulse response can be obtained from the Fourier transform pair,

$$e^{-\pi (t/T)^2} \Rightarrow |T| \exp(-\pi(fT)^2) \quad (4.4)$$

The larger the value of x in (2), the lower is the required bandwidth to generate that impulse response.

In this work, the largest acceptable value of x is 0.01, giving the response shown in Figure 4-2.

From the above data,

$$f(t) = e^{-\pi(1.1207 \frac{t}{T})^2}$$

which transforms to

$$F(f) = e^{-2.1432 (fT)^2} \quad (\text{normalized})$$

(C) Filter Tests

In order to check that this Gaussian impulse response can be obtained in the bandwidth obtainable in the radio equipment, the Gaussian filter was combined with the radio filter characteristics, and the resultant impulse response was obtained using a computer program. Further tests were run to ascertain the effect of low frequency attenuation which may be necessary if mains interference is a problem.

The tests were run as follows:

(1) Gaussian amplitude/frequency response

— this was run simply to ensure that the correct Gaussian impulse response was obtained. (see Figure 4-3).

(2) Gaussian + extra attenuation at low frequencies - the extra attenuation was obtained by adding a 3rd order high pass filter characteristic with break frequency at 500 Hz (see Figure 4-4).

(3) = (2) + radio filter amplitude and group delay response

— this shows the effect the radio filter has on the required Gaussian response

(Figure 4-5).

(4) = (2) + radio filter group delay response

— this shows the effect of the radio filter's group delay characteristic only on the Gaussian response

(Figure 4-6)

(5) = (2) + radio filter amplitude response

— this shows the effect of the radio filter's amplitude characteristic only on the Gaussian response

(Figure 4-7).

(D) Conclusions

The tests in (C) show that it is possible to obtain the required Gaussian response if passband equalization of the radio filter group delay characteristic is performed. This could be included in the design of the transmitter filter by pre-distorting the signal before it enters the radio. The filtering would be split equally between transmitter and receiver for optimum noise performance.

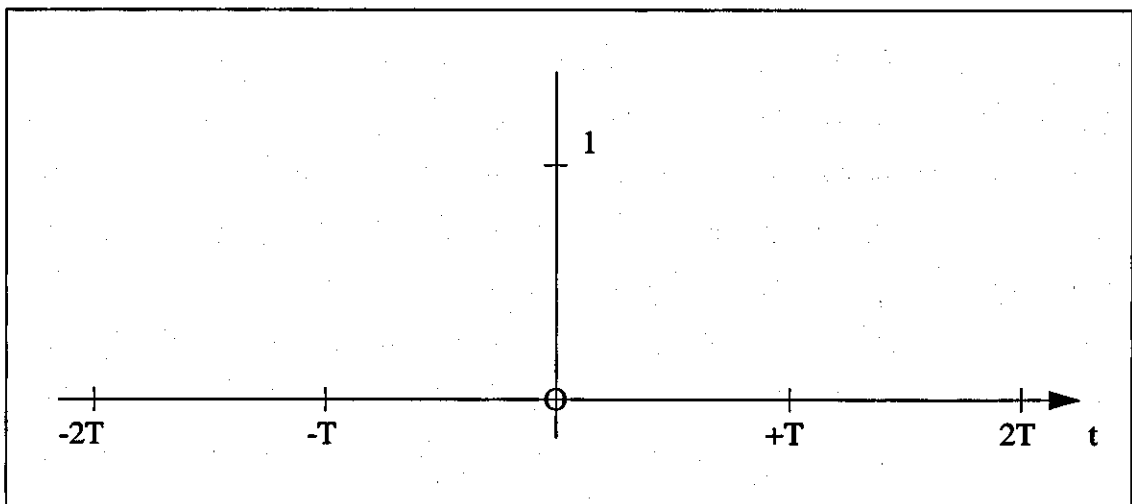


Figure 4-1a

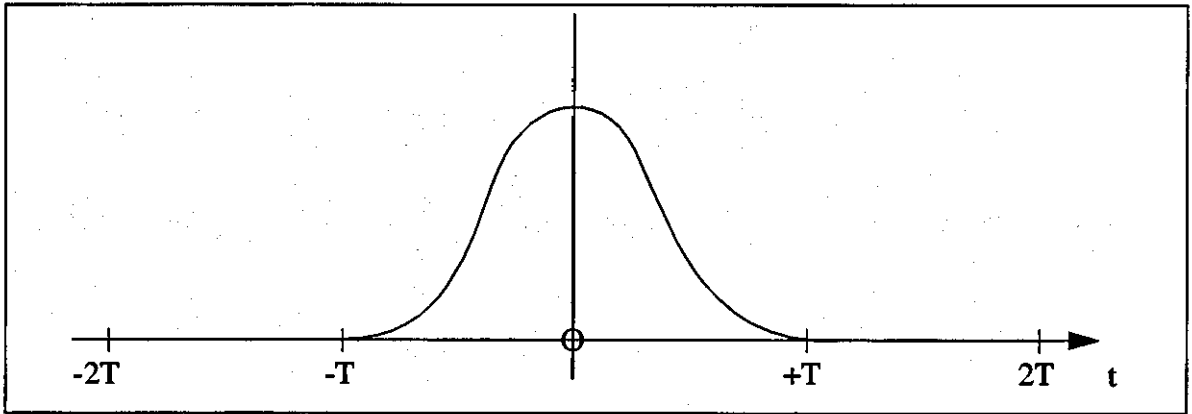


Figure 4-1b

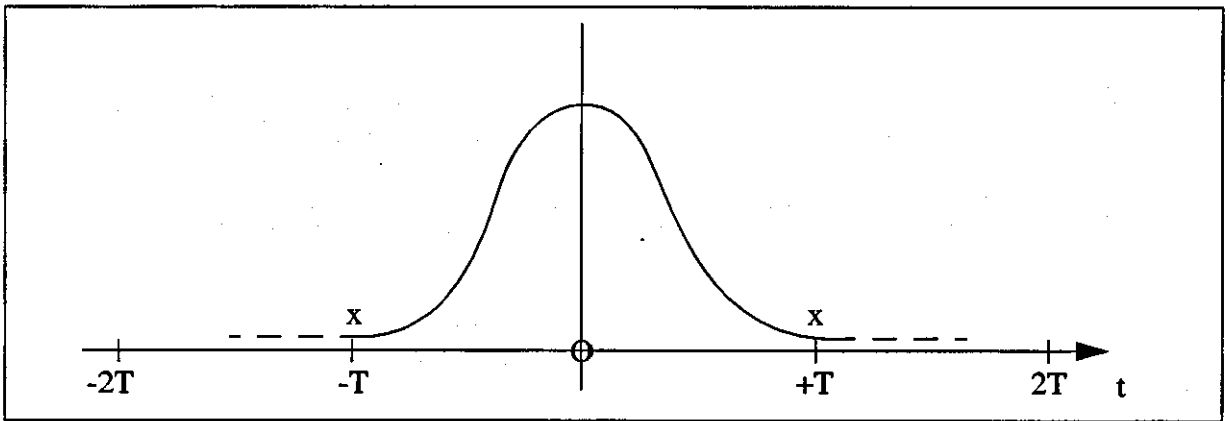


Figure 4-1c

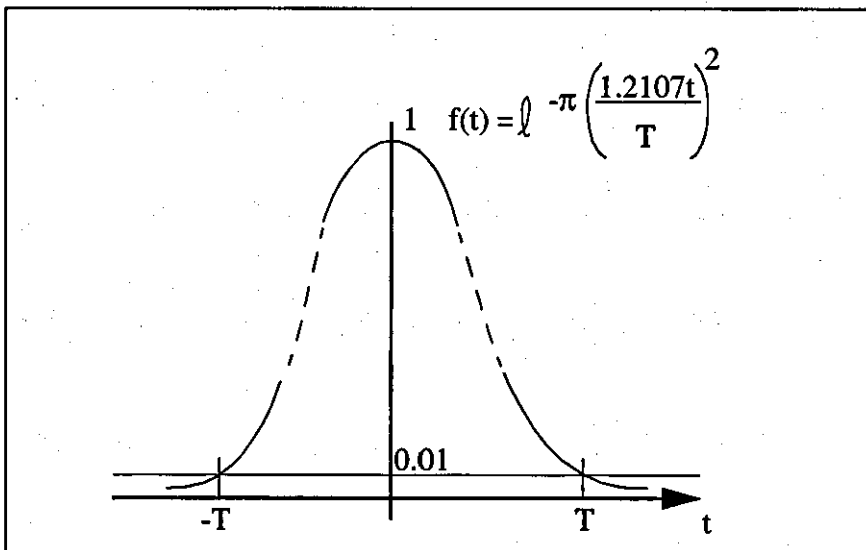


Figure 4-2

Filter 0 = Gaussian Filter
Channel 0 = Const Ampl/Zero Group Del

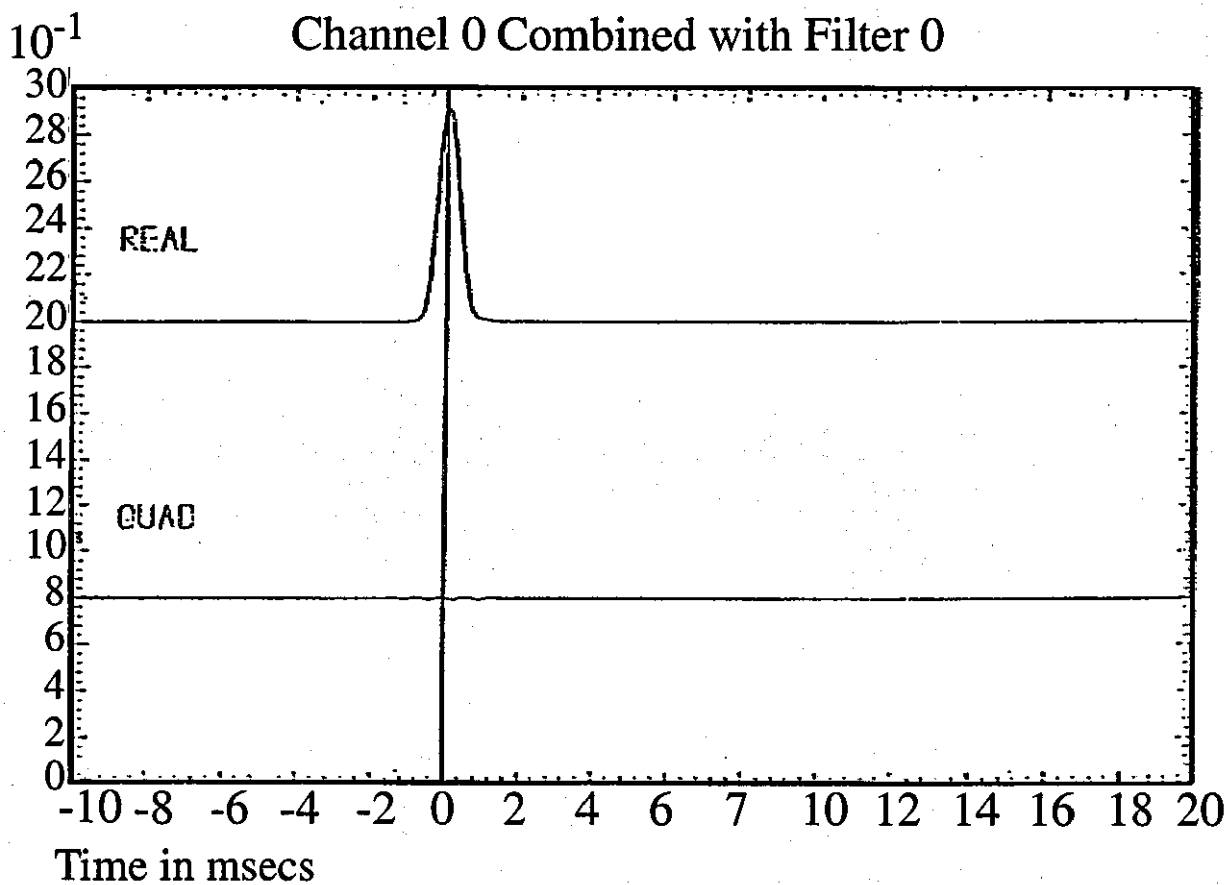
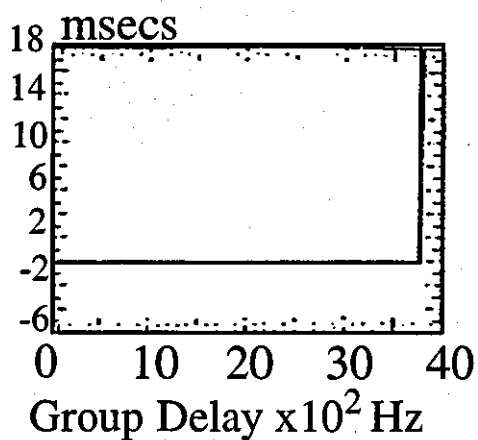
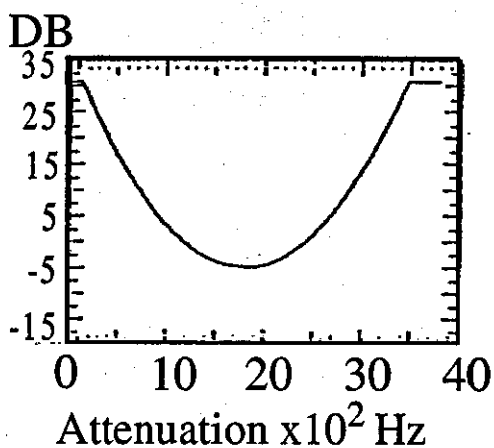


Figure 4-3

Filter 0 = Gaussian Filter
Channel 0 = 3rd Order H.P.F.

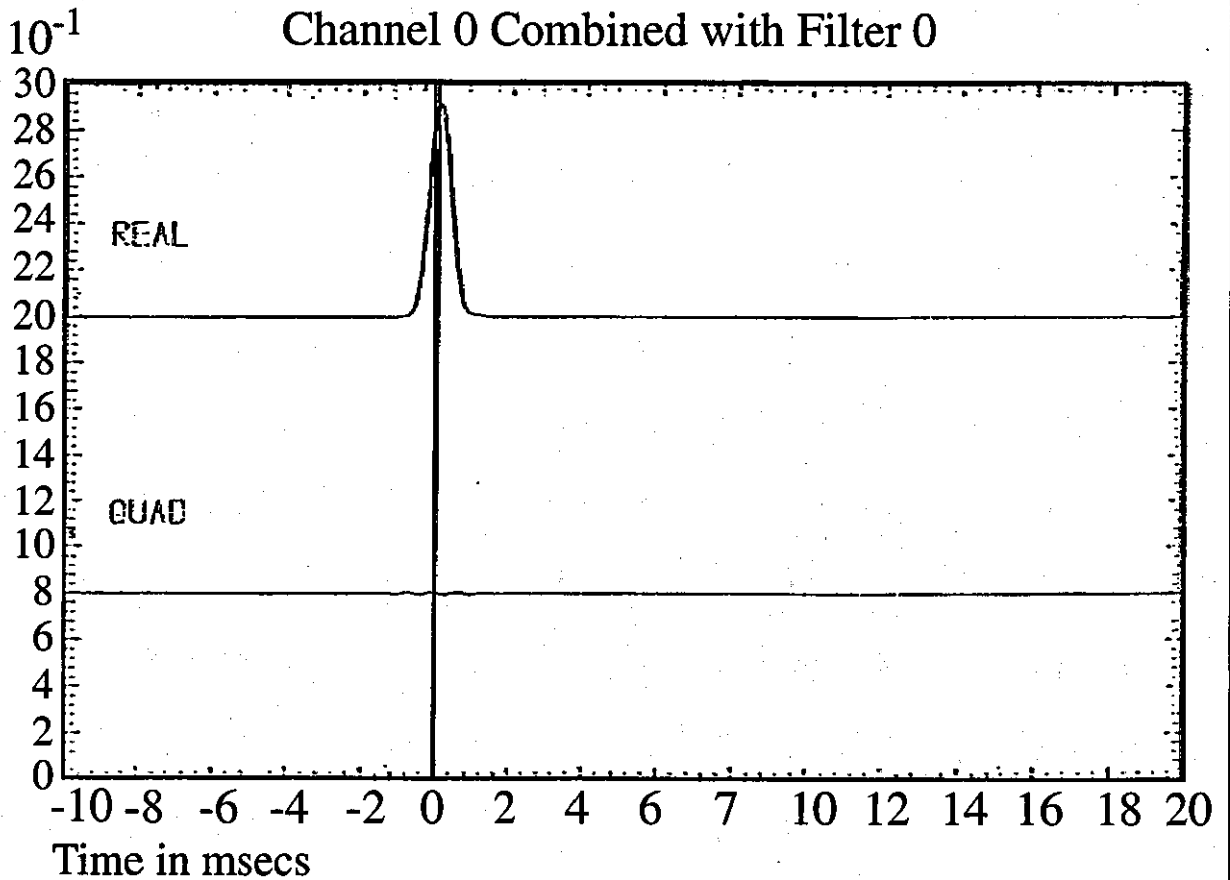
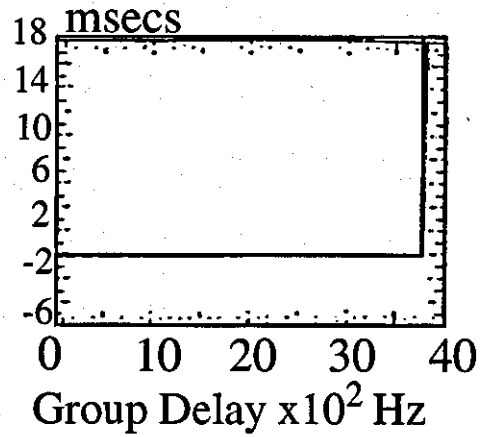
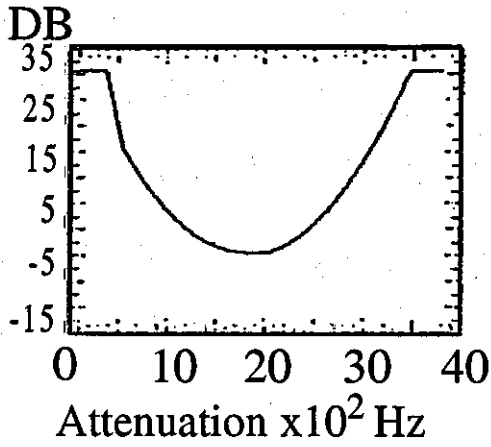


Figure 4-4

Filter 0 = Gaussian Filter + H.P.F.
Channel 0 = Radio Filter Amplitude and Group Delay

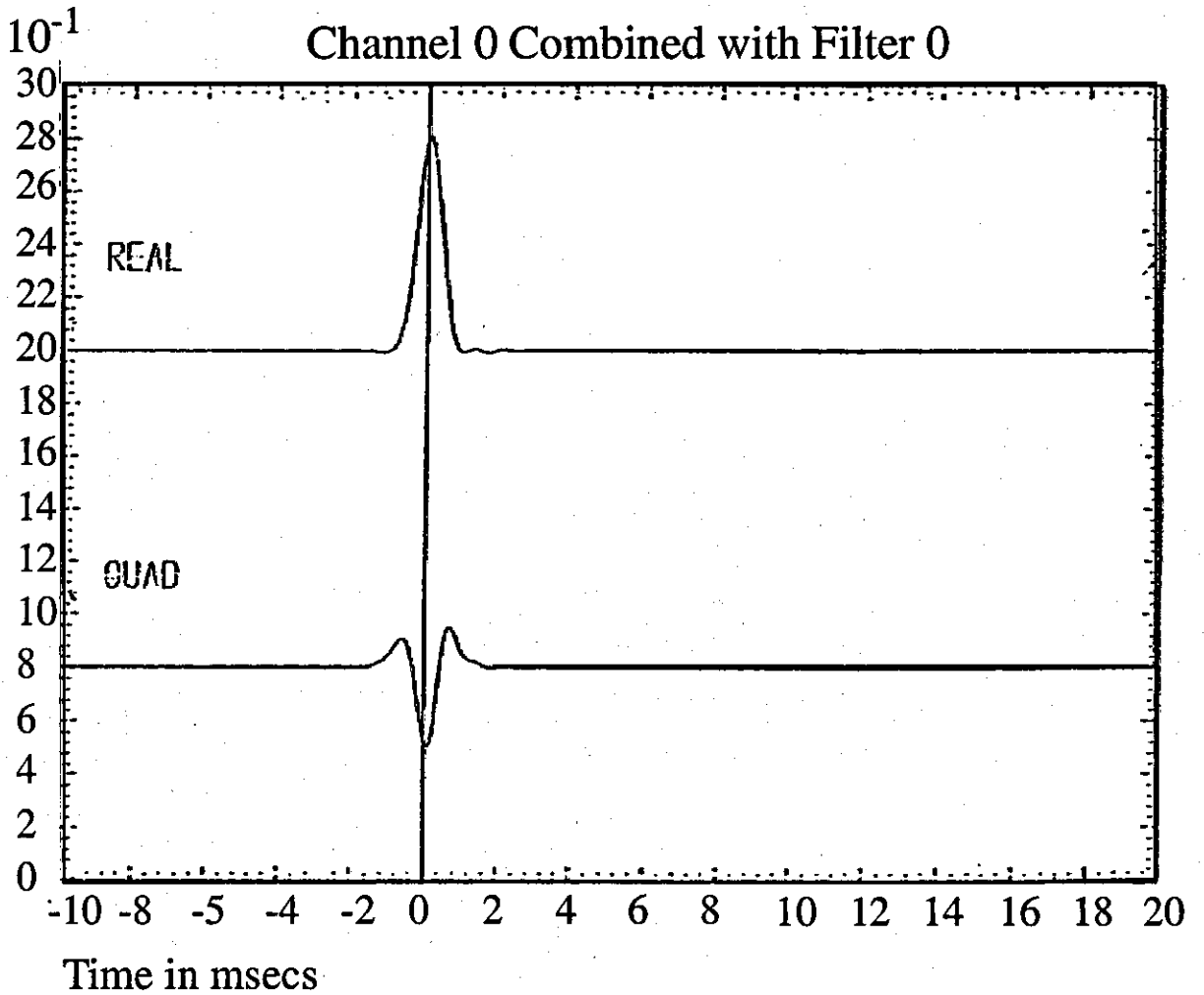
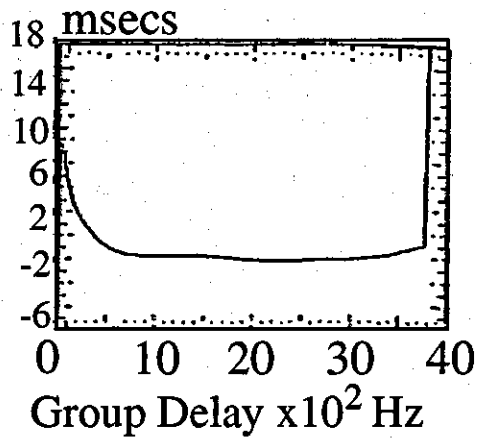
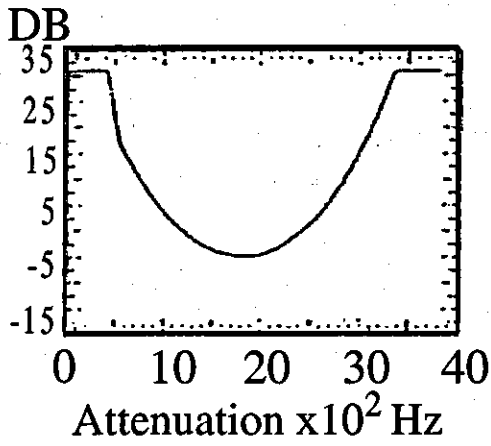


Figure 4-5

Filter 0 = Gaussian Filter + H.P.F.
Channel 1 = Radio Filter Group Delay

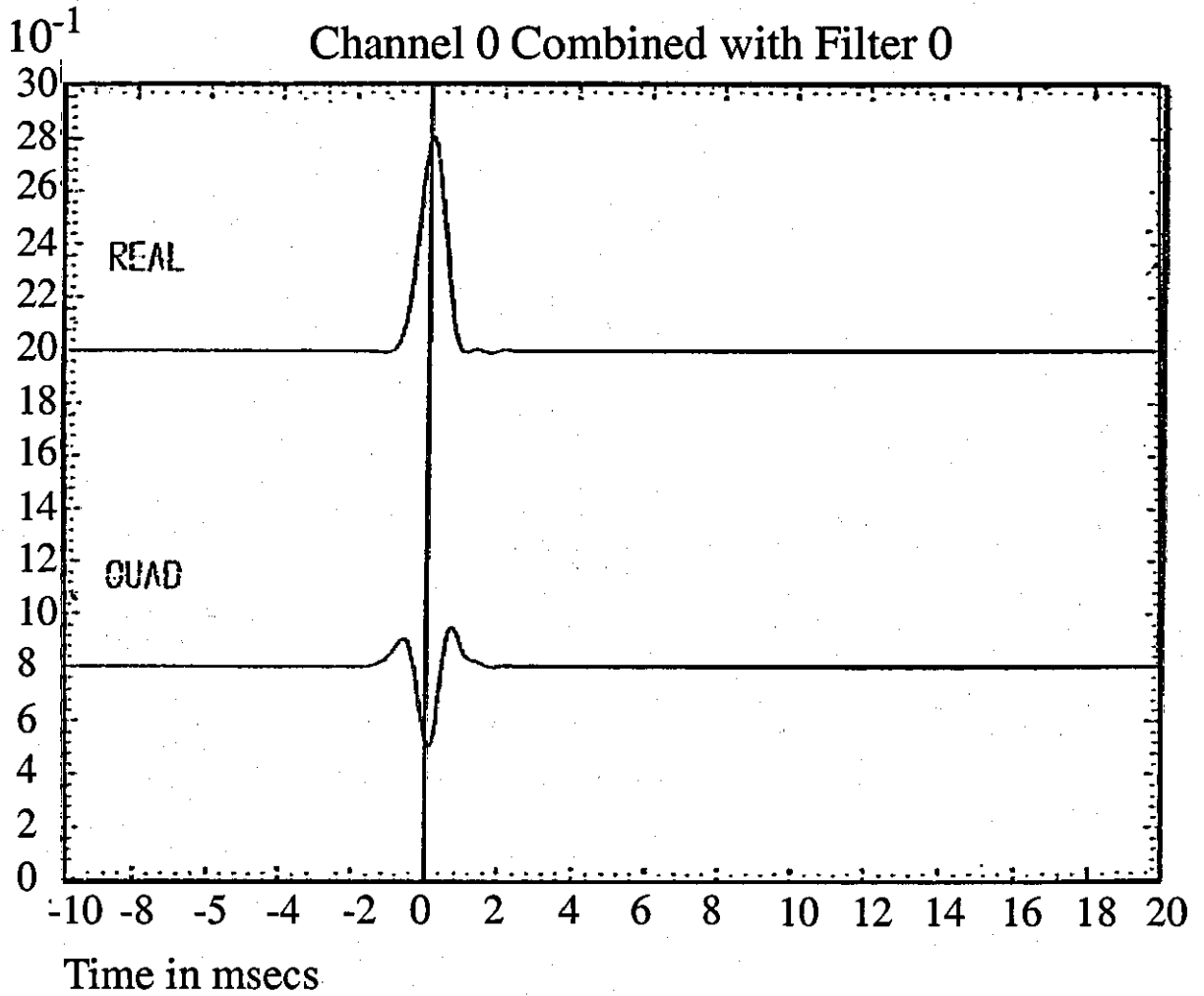
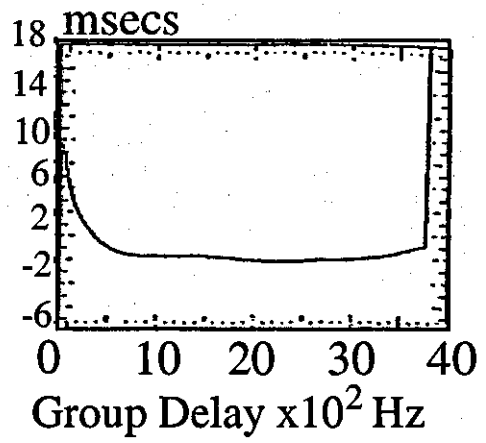
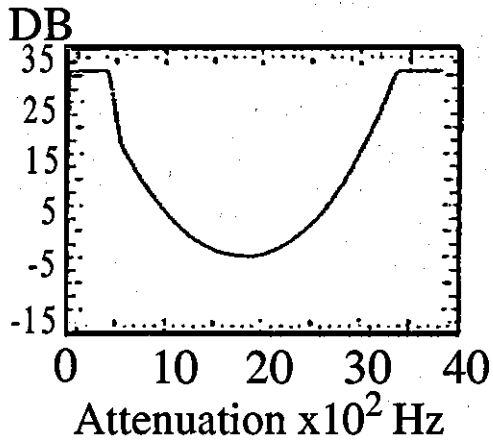


Figure 4-6

Filter 0 = Gaussian + H.P.F
Channel 2 = Radio Filter Amplitude

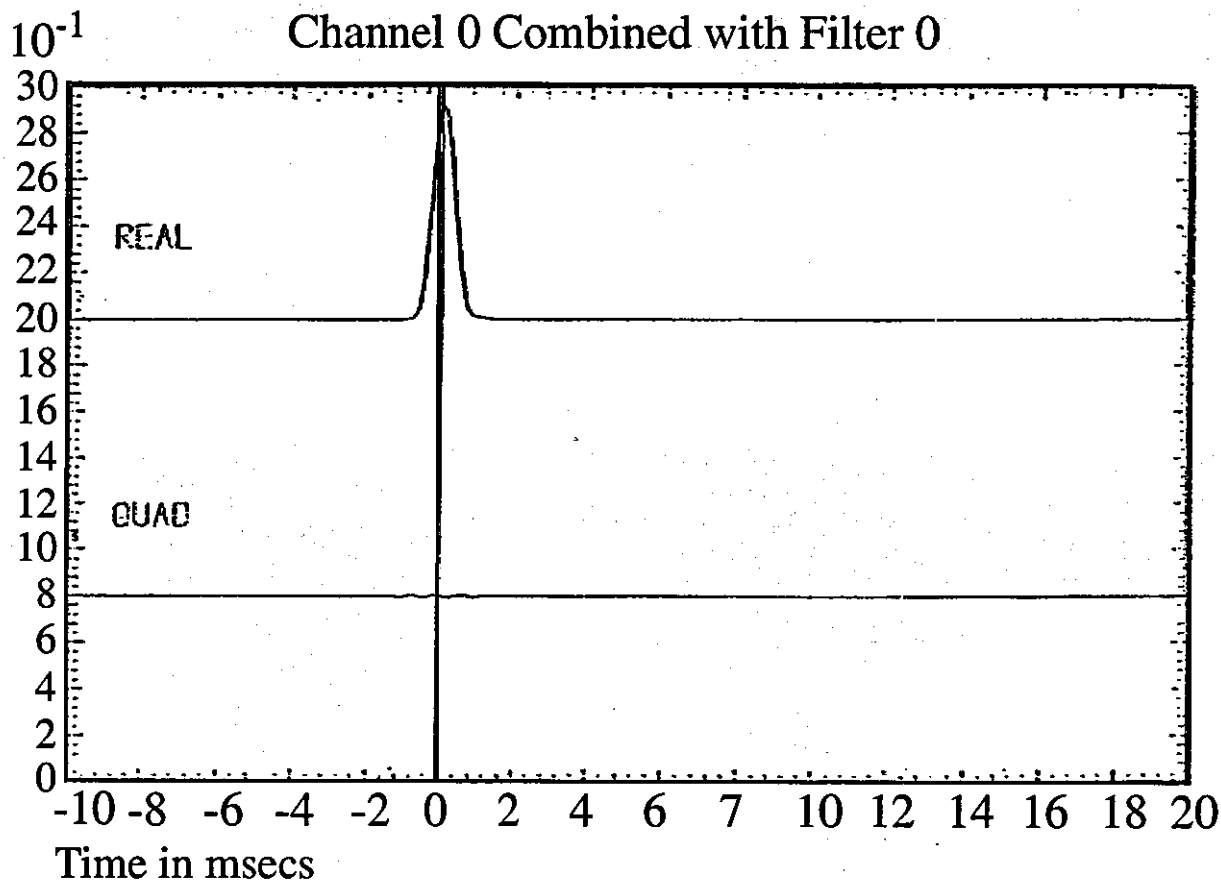
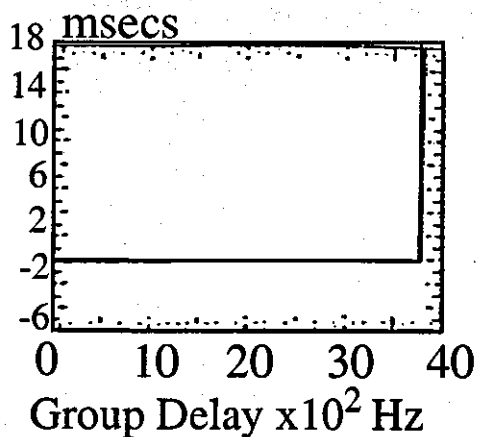
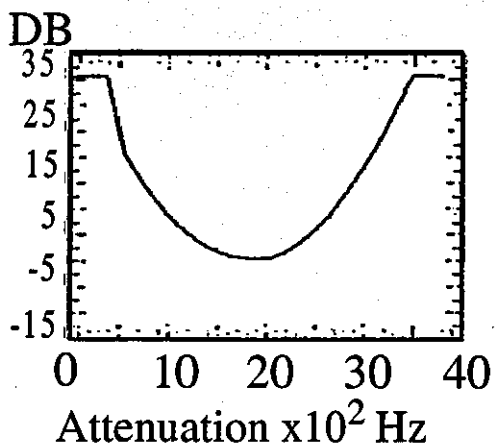


Figure 4-7

Chapter 5

2400 Bits/Second Modem

Transfer functions of transmitter and receiver filters in cascade, $H(f)$ is assumed to have the following shape:

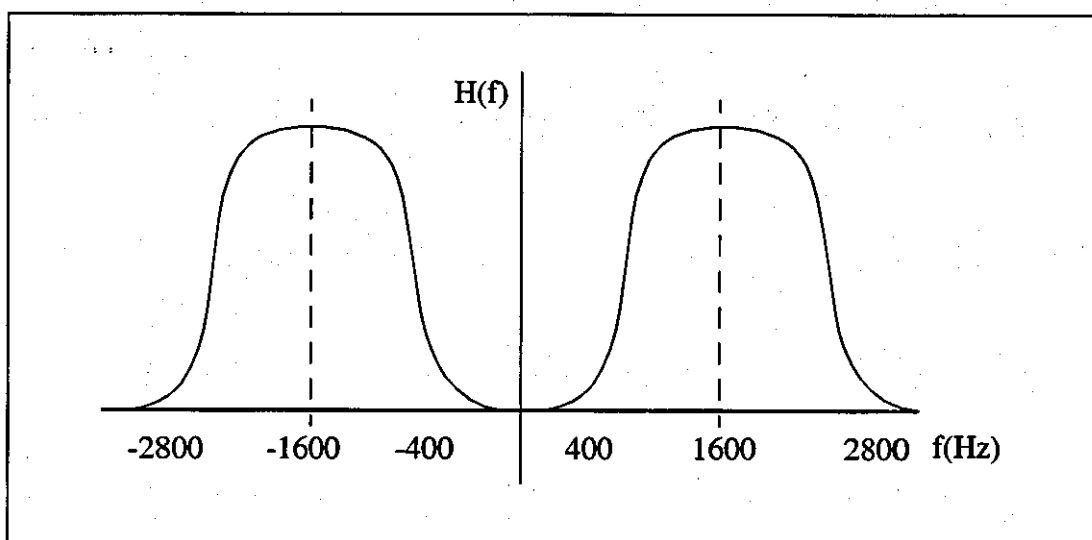
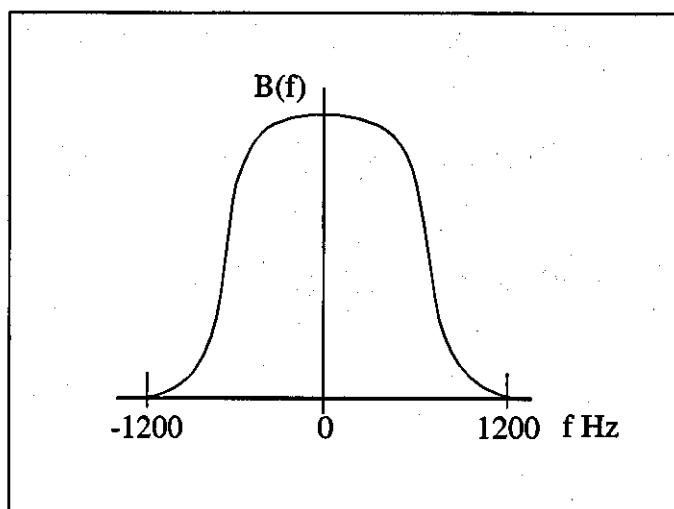


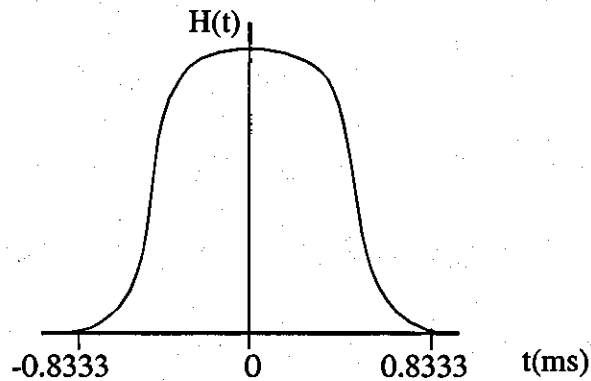
Figure 5-1

bearing in mind that the signal element rate is 1200 bauds.

Thus the transfer function $B(f)$ of the resultant baseband channel is



and the corresponding impulse response is



This means that when the HF radio link introduces no distortion, there is no intersymbol interference in the received sampled signal, the sampled impulse-response of the channel being

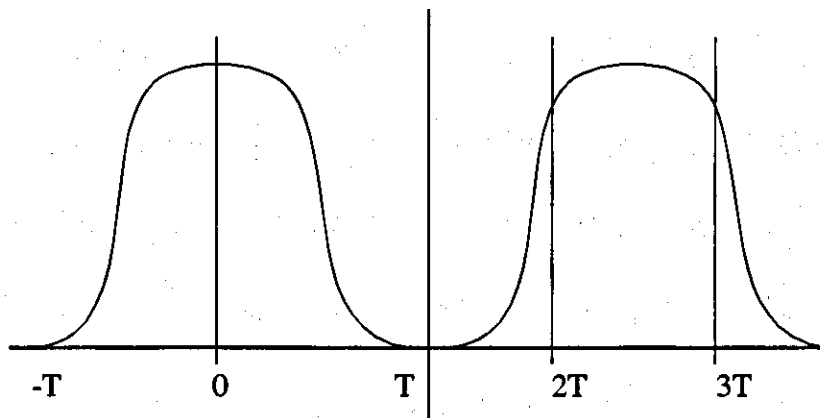
$$1 \ 0 \ 0 \ \dots$$

However, if now the phase of the receiver timing waveform is shifted by 50%, the sampled impulse response becomes approximately

$$0.5 \ 0.5 \ 0 \ 0 \ \dots$$

with a loss of 3dB in tolerance to additive white Gaussian noise.

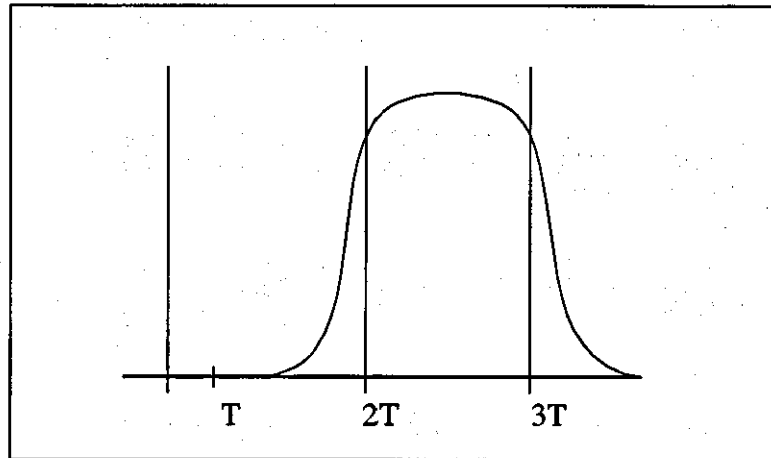
Again, when the multipath effects on the HF radio link give a resultant impulse response



with the corresponding sampled impulse response

$$1 \ 0 \ \frac{1}{2} \ \frac{1}{2} \ 0 \ 0 \ \dots$$

and if now the first component (pulse or echo) fades to zero leaving



with a sampled impulse response

$$0 \ 0 \ \frac{1}{2} \ \frac{1}{2} \ 0 \ 0 \ \dots$$

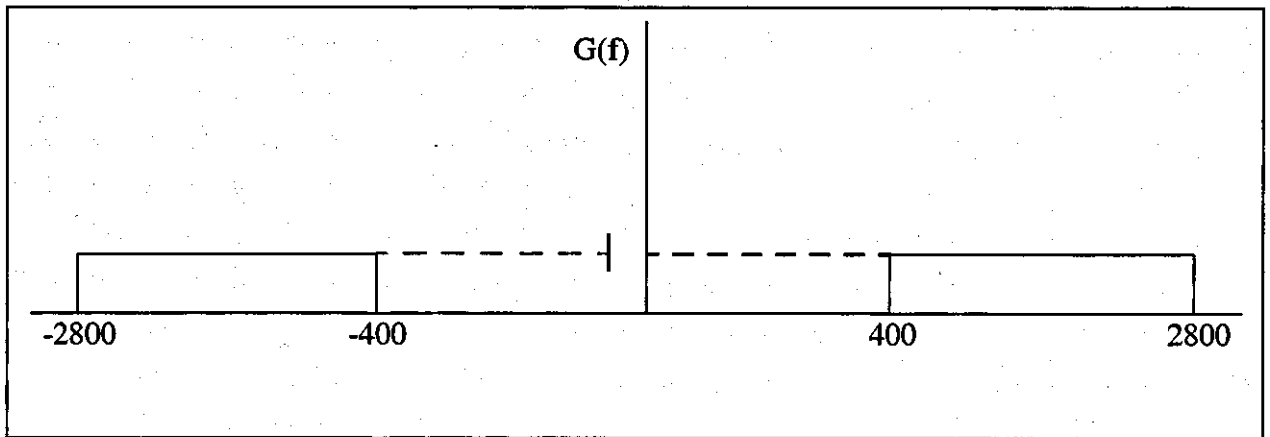
there is again a loss of 3dB in tolerance to additive white Gaussian noise relative to that with the optimum phase of the timing waveform for these conditions. Thus, the arrangement we are now using is subject to a maximum penalty of 3dB in tolerance to noise, and furthermore, it requires the timing waveform to be adjusted adaptively to a given phase.

A preferable arrangement seems to me to be the following:

A) Use the same transmitter and receiver filters to give a resultant transfer functions $H(f)$ and $B(f)$ as on page 1, but now sample at exactly twice the rate as before.

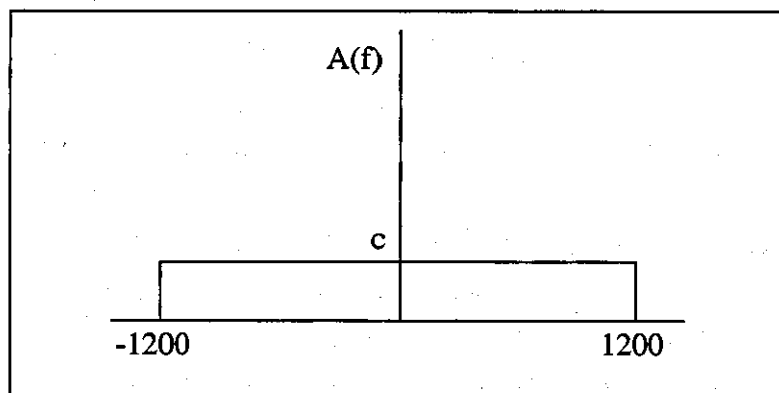
The Viterbi algorithm detector now accepts two new received samples for every detection (vector expansion and selection) process, but is otherwise basically unchanged. The disadvantage here is the high correlation between neighboring (adjacent) noise components.

B) Let the transmitter filters now have a transfer function $H(f)$ as shown in Figure 5-1 on page 47, and let the receiver filters have the following transfer function



so that the resultant transfer function is still $H(f)$ as before. The average transmitted energy per bit is now less than before so that larger magnitudes of s_i must be used for the same average transmitted energy per bit.

If the resultant transfer function of the receiver input filters and demodulator is



then the variance of the real or imaginary component of a resultant complex valued noise component of the detector input is now

$$2400 c^2 \cdot \frac{1}{2} N_0 = 1200 c^2 N_0$$

where $\frac{1}{2} N_0$ is the two-sided power spectral density of the white Gaussian noise at the input to the receiver input filter.

The advantage of this arrangement is that the noise components are uncorrelated so that the given received signal is detected in the optimum manner. Also, the bandwidth of the transmitted signal is reduced.

Each of the two arrangements (A) and (B) have a typical advantage of around $1\frac{1}{2}$ dB over the current system. The arrangements also avoid the need for adjusting the receiver timing waveform to any particular value. There is a strong case for trying the double-sampling systems, since they don't involve much increase in equipment complexity.

Chapter 6

Detection of Multilevel QAM Signals

The basic arrangement assumed here is that used in the systems 1B-5B in the paper *Detection process for severely distorted digital signals* (1), last modified according to the arrangement where only the first significant component of the latest received signal-element to be involved in the detection process is expanded. This arrangement will now be explained.

The transmitted data symbols $\{s_i\}$ are assumed to be 16-level and such that

$$s_i = s_{0,i} + js_{1,i} \quad (j = \sqrt{-1})$$

where

$$s_{1,i} = \pm 1 \text{ or } \pm 3$$

$$s_{0,i} = \pm 1 \text{ or } \pm 3$$

The sampled impulse-response of the channel is

$$Y = y_0 y_1 \dots y_g \quad (6.1)$$

where the $\{y_i\}$ are complex valued.

The sample received at time $t = (i + f) T$

is

$$r_{i+f} = \sum_{h=0}^g s_{i+f-h} y_h + w_{i+f} \quad (6.2)$$

where w_{i+f} is the Gaussian noise component.

Just prior to the reception of r_{i+f} , the receiver holds in store k vectors $\{x_{i-1}\}$

where

$$X_{i-1} = x_{i-n} x_{i-n+1} \dots x_{i-1} \quad (6.3)$$

and x_h , for $h = i-n, i-n+1, \dots, i-1$, has one of the 16 possible values of the data symbol s_h . On the reception of r_{i+f} , each of the vectors $\{X_{i-1}\}$ is expanded into 16 vectors

$$\{x_{i-n} \ x_{i-n+1} \ \dots \ x_{i-1} \ x_i\}$$

where $x_{i-n}, x_{i-n+1}, \dots, x_{i-1}$ have the given values in x_{i-1} , for each of the 16 expanded vectors, but x_i has its 16 different possible values in the 16 vectors.

There are therefore $16k$ expanded vectors.

Let

$$x_i = x_{0,i} + jx_{1,i} \quad (j = \sqrt{-1})$$

where

$$x_{1,i} = \pm 1 \text{ or } \pm 3$$

$$x_{0,i} = \pm 1 \text{ or } \pm 3$$

It is assumed that y_f is the first significant component of Y , and of course $0 \leq f \leq g$.

On the receipt of r_{i+f} , when the receiver forms the $16k$ expanded vectors

$$\{x_{i-n} \ x_{i-n+1} \ \dots \ x_{i-1} \ x_i\}$$

it needs to evaluate for each of these vectors the corresponding cost d_i , which is determined as follows.

Let

$$\|\alpha + jB\| = |\alpha| + |B| \quad (j = \sqrt{-1})$$

for any real-valued quantities α and B , where $|\alpha|$ and $|B|$ are the moduli or magnitudes of α and B , respectively. Also, let

$$v_0 = r_{i+f} - \sum_{h=0}^{g-f} x_{i-h} y_{f+h} \quad (6.4)$$

$$v_\ell = r_{i+f-\ell} - \sum_{h=0}^{g-f+\ell} x_{i-h} y_{f+h-\ell} \quad (6.5)$$

for $\ell = 1, 2, \dots, f$

$$\|U_{i-1}\| = \|U_{i-2}\| + \|r_{i-1} - \sum_{h=0}^g x_{i-1-h} y_h\| \quad (6.6)$$

$$\|U_i\| = \|U_{i-1}\| + \|r_i - \sum_{h=0}^g x_{i-h} y_h\| \quad (6.7)$$

Then

$$d_i = \|U_{i-1}\| + \sum_{\ell=0}^f \ell \quad (6.8)$$

Notice that v_0 is not given by setting $\ell = 0$ in equation (6.5). Having determined d_i from equation (6.8), $\|U_i\|$ is determined from equation (6.7), ready for the evaluation of d_{i+1} .

$$\begin{aligned} \text{Let } u &= u_0 + ju, & (j = \sqrt{-1}) \\ &= r_{i+f} - \sum_{h=1}^{g-f} i^{-h} y_{f+h} \end{aligned} \quad (6.9)$$

where u_0 and u , are real-valued quantities. Then, from equation (6.4),

$$\begin{aligned} v_0 &= u - x_i y_f \\ &= u_0 + ju - (x_{0,i} + jx_{1,i}) (y_{0,f} + jy_{1,f}) \\ &= (u_0 - x_{0,i} y_{0,f} + x_{1,i} y_{1,f}) + j(u_1 - x_{0,i} y_{1,f} - x_{1,i} y_{0,f}) \end{aligned}$$

$$\text{where } x_i = x_{0,i} + jx_{1,i}$$

$$y_f = y_{0,f} + jy_{1,f}$$

$x_{0,i}$, $x_{1,i}$, $y_{0,f}$ and $y_{1,f}$ being real-valued quantities. Thus

$$\|v_0\| = |u_0 - x_{0,i} y_{0,f} + x_{1,i} y_{1,f}| + |u_1 - x_{0,i} y_{1,f} - x_{1,i} y_{0,f}|$$

Associated with each original vector x_{i-1} that is expanded into m vectors $\{x_{i-n} \ x_{i-n+1} \ \dots \ x_i\}$, there is just one value of v_i for each value of i in the range 1 to f , since none of these $\{v_i\}$ are functions of x_i .

There are, however, 16 values of v_0 corresponding to the 16 possible values of x_i .

It can readily be shown that for some of the typical possible values of the parameters involved,

$$\|v_0\| = |u_0 - x_{0,i} (y_{0,f} + y_{1,f})| + |u_1 - x_{1,i} (y_{0,f} - y_{1,f})|$$

in which case, the four possible values of

$$|u_0 - x_{0,i} (y_{0,f} + y_{1,f})|$$

corresponding to the four possible values of $x_{0,i}$ can be evaluated as can the four possible values of

$$|u_1 - x_{1,i} (y_{0,f} - y_{1,f})|$$

corresponding to the four possible values of $x_{1,i}$, and now the 16 possible values of $\|v_0\|$ are given by the different possible combinations (sums) of the two sets of four values taken two at a time. This tech-

nique could reduce the amount of computation required to evaluate the 16 possible values of $\|v_0\|$. Unfortunately, it can also be shown that for certain typical values of the parameters involved

$$\|v_0\| \neq |u_0 - x_{0,i}(y_{0,f} + y_{1,f})| + |u_1 - x_{1,i}(y_{0,f} - y_{1,f})|$$

so that this technique obviously cannot be applied in the manner just described. However, all is not lost since it is possible to arrange matters so that the above basic technique can be applied.

Instead of using the cost d_i , as given by equation (6.8), the detector now uses the cost

$$c_i = \|U_{i-1}\| + \sum_{\ell=1}^f + \ell \left\| \frac{y_f^*}{|y_f|} v_0 \right\|$$

where y_f^* is the complex-conjugate of y_f and as before

$$\begin{aligned} v_0 &= r_{i+f} \sum_{h=0}^{g-f} x_{i+h} y_{f+h} \\ &= u - x_i y_f \text{ (from equation (6.9))} \end{aligned}$$

$$\text{so that } \left| \frac{y_f^*}{|y_f|} v_0 \right| = \frac{y_f^*}{|y_f|} u - \frac{y_f^*}{|y_f|} x_i y_f$$

$$= q - |y_f| x_i$$

$$\text{where } q = \frac{y_f^*}{|y_f|} u$$

$$= q_0 + jq_1$$

q_0 and q_1 being real-valued quantities.

$$\text{Thus } \left\| \frac{y_f^*}{|y_f|} v_0 \right\| = \|q_0 + jq_1 - |y_f|(x_{0,i} + jx_{1,i})\|$$

$$= \|(q_0 - |y_f| x_{0,i}) + j(q_1 - |y_f| x_{1,i})\|$$

$$= |q_0 - |y_f| x_{0,i}| + |q_1 - |y_f| x_{1,i}|$$

and now the 16 possible values of $\left\| \frac{y_f^*}{|y_f|} v_0 \right\|$

are given by the 4 possible values of

$$|q_0 - |y_f| x_{0,i}|$$

and the four possible values of

$$|q_1 - |y_f| x_{1,i}|$$

taking all combinations two at a time.

$\frac{y_f^*}{|y_f|}$ is a unit vector in the complex numbers plane so that $\left| \frac{y_f^*}{|y_f|} v_0 \right| = |v_0|$, but in general

$\left\| \frac{y_f^*}{|y_f|} v_0 \right\| \neq \|v_0\|$. The latter, however, does not matter very much since its effect is no worse than that of using the moduli-sum $\| \cdot \|$ in place of the Euclidean distance $| \cdot |$, which is already being done.

Having said this, it can be seen that a proper multiplication is now required and if there is, in addition, some small degradation in performance due to the fact that

$$\left\| \frac{y_f^*}{|y_f|} v_0 \right\| \neq \|v_0\|$$

(which, however, need not necessarily be the case,) there does not, on balance, appear to be any very great advantage in using this technique.

It depends critically on the additional complexity involved with the multiplication.

Where Euclidean distances are used, the technique just described can be applied (with $\| \cdot \|$ replaced by the appropriate function of $| \cdot |$) without the need for an additional multiplication, so that in this case a useful simplification should always be possible.

The most promising simplification to the 16-point QAM system under consideration here is the following modification. It has so far been assumed that

$$s_i = s_{0,i} + js_{1,i} \quad (j = \sqrt{-1})$$

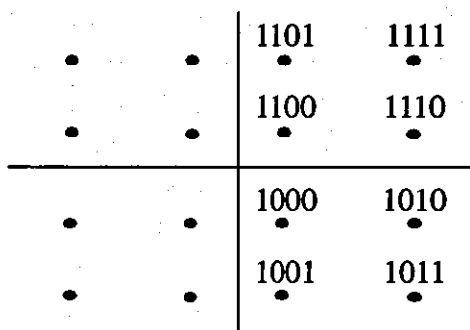
where $s_{0,i} = \pm 1$ or ± 3

$$s_{1,i} = \pm 1 \text{ or } \pm 3$$

Now let $s_i = a_i + b_i$

where $a_i = \pm 2 \pm 2j$
 $b_i = \pm 1 \pm j$

Notice that in using a 16-point QAM signal whose possible symbol values are shown as follows, in the complex number plane each point corresponds to a sequence of four binary digits, i.e. 0100 or 1110, etc.



It is essential that Gray coding is used, such that for any two immediately neighboring points of the 16-point QAM signal, the two corresponding binary sequences differ in only one binary digit. This minimizes the error rate in the decoded binary signal at the output of the receiver. Clearly with the new signal representation involving a_i and b_i , the appropriate coding must be used to generate the corresponding binary digits, and, of course, the computed error rate must be that in the decoded (and detected) binary digits and not that in the detected data symbols $\{s_i\}$.

The new arrangement operates essentially in the same way as the previous system except that each data symbol is now treated as the sum of two symbols a_i and b_i . The system operates as follows:

Just prior to the reception of $r_{i,f}$, the receiver holds in store k vectors $\{X_{a,i-1}\}$

where

$$X_{a,i-1} = x_{a,i-n} \ x_{a,i-n+1} \ \dots \ x_{a,i-1}$$

and $x_{a,h}$, for $h = i-n, i-n+1, \dots, i-1$, has one of the four possible values of the data symbol a_h , and the receiver also holds in store k vectors $\{X_{b,i-1}\}$ where

$$X_{b,i-1} = x_{b,i-n} \ x_{b,i-n+1} \ \dots \ x_{b,i-1}$$

and $x_{b,h}$, for $h = i-n, i-n+1, \dots, i-1$, has one of the four possible values of the data symbol b_h . Thus there are $2k$ stored vectors. The vectors $\{X_{a,i-1}\}$ are stored in groups of four, the four vectors in any one group all having the same value of $x_{a,h}$ for some given value of h . For instance, there could be eight groups of four vectors

$\{X_{a,i-1}\}$ corresponding to the four different possible values of $x_{a,i-1}$ and the four different possible values of $x_{a,i-2}$. This would give 32 stored vectors $\{X_{a,i-1}\}$ and 32 stored vectors $\{X_{b,i-1}\}$, and so 64 stored vectors in all. Each vector $X_{a,i-1}$ is associated with the corresponding vector $X_{b,i-1}$, the two vectors together defining the vector X_{i-1} in equation (6.3) for the original system. Each of the k pairs of vectors is associated with a cost, d_{i-1} , which is evaluated exactly as previously described, the vectors $X_{a,i-1}$ and $X_{b,i-1}$ being, for this purpose, replaced by the corresponding vector X_{i-1} .

On the receipt of y_{i+f} , each of the k vectors $\{X_{a,i-1}\}$ is expanded into four vectors

$$\{x_{a,i-n} \ x_{a,i-n+1} \ \dots \ x_{a,i-1} \ x_{a,i}\}$$

where $x_{a,i-n}, x_{a,i-n+1}, \dots, x_{a,i-1}$ have the given values in $X_{a,i-1}$, for each of the 4 expanded vectors, but $x_{a,i}$ has its four different possible values in the 4 vectors; there are therefore $4k$ expanded vectors. The receiver now evaluates v_0, v_1, \dots, v_f and hence d_i (equation (6.8)) for each expanded vector, remembering of course that the corresponding stored vector $X_{b,i-1}$ is involved in these computations and taking $x_{b,i}$ to have the value zero in every case. k of the $4k$ expanded vectors are now selected in the following manner.

The receiver selects k vectors according to the arrangement of system 5B, if possible, or system 3B if not (see the paper *Detection processes for severely distorted digital signals* (1)). The selection process must, however, be modified slightly to suit the new situation. Consider for example the selection of 32 stored vectors $\{X_{a,i}\}$ according to system 3B. For each of the 16 combinations of possible values of $x_{a,i-1}$ and $x_{b,i-1}$, the detector selects a vector $X_{a,i}$ together with the associated vector $X_{b,i-1}$ and stores the vector $X_{a,i}$, giving 16 stored vectors. Then, for each of the four possible values of $x_{a,i}$, the detector selects a vec-

tor $X_{a,i}$ together with the associated vector $X_{b,i-1}$ and stores the vector $X_{a,i}$. This is repeated another three times to give another 16 stored vectors. Notice that the detector must not select four vectors $\{X_{a,i}\}$, in turn, for each value of $x_{a,i}$, since this technique leads to the merging of the vectors, and hence to an effective reduction in the numbers of stored vectors.

Having selected k vectors $\{X_{a,i}\}$ and stored these, each one of the vectors being associated with the corresponding one of the stored vectors $\{X_{b,i-1}\}$, the two vectors of any pair completely defining the corresponding sequence of temporarily detected data symbols, the k vectors $\{X_{b,i-1}\}$ are now expanded, each giving 4 vectors

$$\{x_{b,i-n} \ x_{b,i-n+1} \ \dots \ x_{b,i-1} \ x_{b,i}\}$$

where $x_{b,i-1}, x_{b,i-n+1}, \dots, x_{b,i-1}$ have the given values in $X_{b,i-1}$, for each of the four expanded vectors, but $x_{b,i}$ has its four different possible vectors in the four vectors. The receiver now evaluates d_i for each expanded vector, remembering, of course, that the corresponding stored vector $X_{a,i}$ is involved in these computations. k of the $4k$ vectors are now selected in the following manner:

The receiver selects k vectors $\{X_{b,i}\}$ according to the arrangement of System 5B, if possible, or System 3B if not. Consider for example the selection of 32 stored vectors $\{X_{b,i}\}$ according to System 3B. For each of the 16 combinations of possible values of $x_{b,i-1}$ and $x_{a,i-1}$, the detector selects a vector $X_{b,i}$ together with the associated vector $X_{a,i}$ and stores the two vectors together with the associated value of d_i . This gives 16 stored pairs of vectors and associated costs. Then, for each of the 16 combinations of possible values of $x_{b,i}$ and $x_{a,i}$, the detector selects the appropriate vectors $X_{b,i}$ and $X_{a,i}$, and stores these together with the associated $\{d_i\}$. This gives a total of 32 stored pairs of vectors and associated costs $\{d_i\}$.

The arrangement just described has the advantage that the $k = 32$ stored vectors expand to just $4k = 128$ vectors, but it has the disadvantage that because the $\{x_{b,i}\}$ are temporarily ignored when the vectors $\{X_{a,i}\}$ are selected, there is bound to be some loss in performance which could be significant.

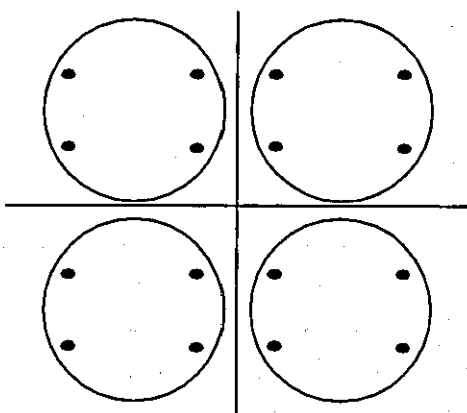
The arrangement with 32 stored vectors, just described, can be further simplified as follows: When

the detector selects a vector $X_{a,i}$ together with the associated $X_{b,i-1}$ for each of the four possible values of $x_{a,i-1}$. Instead of then repeating this operation another three times, the selection process now terminates to give a total of $16 + 4 = 20$ stored vectors. The process then continues as described. It is not clear as to what degradation in performance will result from this simplification or whether it is in fact a useful simplification.

An important property of the basic technique just described is that it can be applied to give a mean of more than the two columns that are used in the example described.

The important property of this selection process here is that whenever a vector $X_{a,i}$ is selected for a given value of say $x_{a,i-1}$ (or it could be $x_{a,i}$ or $x_{a,i-2}$, etc.) then associated with this selected value of

$x_{a,i-1}$, four pairs of vectors $\{X_{b,i}\}$ and $X_{a,i}\}$ must always be selected, having the given value of $x_{a,i-1}$ and also the 4 possible values of $x_{b,i-1}$. Thus the resultant vectors $\{X_i\}$ are always selected in groups of four, having the four possible values of some component x_h , that lie in one of the four quadrants:



Bearing this in mind, it can now be seen that the selection process for the stored vectors can be modified as follows:

When selecting the 32 stored vectors $\{X_{a,i}\}$, these are selected in eight groups of four vectors, the four vectors in any one group having a given value of $x_{a,h}$ and being associated with four vectors $\{X_{b,i}\}$ having the four different possible values of $x_{b,f}$. With this arrangement, the eight groups of four vectors $\{X_{a,i}\}$ can be selected to give either

a) the four possible values of $x_{a,i-1}$ and the four possible values of $x_{a,i}$, as in the example already considered, or

b) two possible values of $x_{a,i-2}$, two possible values of $x_{a,i-1}$, and the four possible values of $x_{a,i}$, or

c) one possible value of $x_{a,i-1}$, three possible values of $x_{a,i-1}$ and the four possible values of $x_{a,i}$

Note that in (b) above the two possible values of $x_{a,i-2}$ have been determined from the previous detection process, so that no selection is involved here in which of the values of $x_{a,i-1}$ to take. Of course, if the wrong value of $x_{a,i-2}$ is selected, it is most unlikely that a complete set of 4 pairs of vectors $\{X_{a,i}\}$ and $\{X_{b,i}\}$ will be available for selection. In the other cases there is a choice in which values of $x_{a,i-2}$ or $x_{a,i-1}$ to take, and these should be taken as the values which give the lowest value of d_i , when considering all four vectors in each group.

Obviously there are other arrangements which could be better or simpler, but the three selection processes listed on mentioned above are certainly worth considering.

A further study of the problem suggested that the multiplication of v_0 by $\frac{y_f^*}{|y_f|}$ led to an improved tolerance to noise since when the coefficient y_f of x_i in

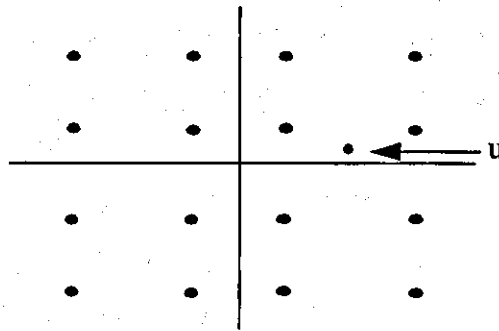
$$v_0 = u - y_f x_i$$

is real then the use of the moduli-sum as the distance measure, i.e. $\|v_i\|$, gives the same decision boundaries and hence tolerance to noise as does the Euclidean distance. Since the multiplication of the signal v_0 by $\frac{y_f^*}{|y_f|}$ would involve extensive equipment complexity, a simpler arrangement would be to change the sampled impulse-response of the channel itself by effectively multiplying the input signal after sampling by $\frac{y_f^*}{|y_f|}$. There should now be a further reduction in the complexity of the system.

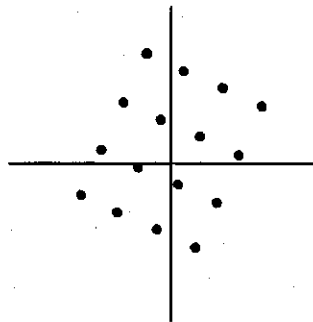
When selecting the value of x_i so as to minimize the error signal v_0 , where

$$v_0 = u - y_f x_i$$

and in general u , y_f , x_i and v_0 are complex valued, the Euclidean distance is minimized when $|v_0|$ is minimized or $|u - y_f x_i|$ is minimized so that x_i must be chosen to minimize this distance. When y_f is real, the possible values of $y_f x_i$ are as follow in the complex number plane:



It can be seen by simple arguments of symmetry that the possible value of $y_f x_i$ that is now at the minimum Euclidean distance from u is also at the minimum moduli-sum distance regardless of where u is. Thus both distance measures lead to the same decision boundaries and so to the same tolerance to noise. However, when y_f is complex valued, the possible values of $y_f x_i$ could be as follows:



The Euclidean and moduli-sum distance measures are not now the same. Since under the named conditions, the Euclidean distance measure is the correct (or optimum) measure, the moduli-sum distance measure is now sub-optimum.

Checks must however be made with different multiplying constants on the received signal to check the effect of these on the overall detector performance, since this might be adversely affected by the change.

In principle, it seems likely that the reason why the performance of the 16-point QAM system is infe-

rior to that of the 4-point, at least by 1/2 dB or so, is simply that an insufficient number of stored vectors are available for use in the 16-point system. Any arrangement which either leads to an increase in the number of stored vectors or else results in a more efficient use of these vectors, should give a performance at least as good as that obtainable with the 4-point QAM system. The techniques described here could help towards this end.

References

1. CLARK, A. P., KWONG, C. P. and HARVEY, J. D., *Detection processes for severely distorted digital signals*, Electronic Circuits and Systems, 1979, 3, pp. 27-37

Chapter 7

Reduced-Complexity Detection Process for Multilevel Signals

The arrangements described in Chapter 6 use the fact that

$$s_i = a_i + b_i$$

where

$$a_i = \pm 2 \pm 2j$$

$$(j = \sqrt{-1})$$

$$b_i = \pm 1 \pm j$$

The receiver here forms an estimate $x_{a,i}$ of a_i where $x_{a,i} = \pm 2 \pm 2j$, and an estimate $x_{b,i}$ of b_i , where $x_{b,i} = \pm 1 \pm j$.

On the receipt of r_{i+T} the receiver first forms the possible values of $x_{a,i}$ and then using these values, it forms the possible values of $x_{b,i}$, hence determining the values $\{d_i\}$ of the k stored pairs of vectors $\{X_{a,i}\}$ and $\{X_{b,i}\}$. an important feature of the techniques considered is that in selecting the vectors $\{X_{a,i}\}$ the receiver takes account of the previously selected vectors $\{X_{b,i}\}$ and in selecting the vectors $\{X_{b,i}\}$ it takes account of the previously selected vectors $\{X_{a,i}\}$, an efficient use being therefore made of the prior knowledge of the received signal in any vector selection process. The price that is paid for this is that the receiver must take account of combinations of the vectors $\{X_{a,i}\}$ and $\{X_{b,i}\}$ each vector $X_{a,i}$ being associated with four vectors $\{X_{b,i}\}$. Inevitably this reduces the simplification that can be achieved. For the greatest possible simplification of the basic system, the vectors $\{X_{a,i}\}$ and the vectors $\{X_{b,i}\}$ must be selected quite independently of each other. The basis arrangement now to be considered operates in this way and achieves a substantial reduction both in the number of operations and the amount of storage required. It may not however operate very well.

The first part of the receiver treats the received samples $\{Y_i\}$ where

$$Y_i = \sum_{h=0}^g i-h Y_h + w_i$$

as though Y_i was in fact equal to

$$\sum_{h=0}^g a_{i-h} y_h + w_i$$

and it implements the detection process of System 5B on this assumption. The detection process is therefore that for a 4-point QAM signal. On the receipt of Y_i the receiver now selects k vectors $\{X_{a,i}\}$ where

$$X_{a,i} = x_{a,i-n+1} x_{a,i-n+2} \dots x_{a,i}$$

It then detects a_{i-n+1} as the value of $x_{a,i-n+1}$ in the stored vector $X_{a,i}$ associated with the smallest d_i . d_i is here evaluated on the assumption that s_i is equal to a_i , so that no vectors $\{X_{b,i}\}$ are anywhere involved in the evaluation of the $\{d_i\}$ or the subsequent selection of the $\{X_{a,i}\}$. On the detection of a_{i-n+1} the receiver now discards any stored vector $X_{a,i}$ in which $x_{a,i-n+1}$ is not equal to the detected value of a_{i-n+1} . The detected value of a_{i-n+1} is designated a'_{i-n+1} and is added to the previous $n+g+f-1$ detected values

$$a'_{i-2n-g-f+2} a'_{i-2n-g-f+3} \dots a'_{i-n+1}$$

These are used to generate the sequence of $n+f$ symbols $\{q_l\}$, for

$$l = i-2n-f+2, i-2n-f+3, \dots, i-n+1$$

where

$$q_l = \sum_{h=0}^g a'_{l-h} y_h$$

and then the corresponding sequence of $n+f$ symbols $\{r'_l\}$, for

$$l = i-2n-f+2, i-2n-f+3, \dots, i-n+1$$

where

$$r'_l = r_l - q_l$$

This is of course an arrangement of decision directed cancellation of intersymbol interference such that with the correct detection of the $\{a_l\}$,

$$r'_l = r_l - q_l$$

$$\begin{aligned}
&= \sum_{h=0}^g (a \ell_{-h} + b \ell_{-h}) y_h + w \ell - \sum_{h=0}^g \ell_{-h} y_h \\
&= \sum_{h=0}^g b \ell_{-h} y_h + w \ell
\end{aligned}$$

Of course, only r'_{i-n+1} is generated as a result of the latest detection process, the other $\{r' \ell\}$ having been generated from the earlier detection processes and having been stored. No great equipment complexity is therefore involved in generating the $\{r' \ell\}$.

The second part of the receiver now operates the $\{r' \ell\}$ assuming the correct detection of the $\{a_h\}$ and so that the above equation holds. Just prior to the generation of r'_{i-n+1} the receiver holds in store k vectors $\{X_{b,i-n-f}\}$ where

$$X_{b,i-n-f} = x_{b,i-2n-f+1} x_{b,i-2n-f+2} \dots x_{b,i-n-f}$$

Just as in the first part of the receiver it is assumed that y_f is the first significant component of

$$Y = y_0 y_1 \dots y_g$$

so it is again assumed here that y_f is the first significant component of Y . Hence, $x_{b,i-n-f}$ is the last component of a stored vector just before the generation of r'_{i-n+1} .

Following the generation of r'_{i-n+1} , the k stored vectors $\{X_{b,i-n-f}\}$ are expanded, their costs $\{d_{i-n-f+1}\}$ are evaluated on the assumption that

$$r' \ell = \sum_{h=0}^g \ell_{-h} y_h + w \ell$$

and k vectors $\{X_{b,i-n-f+1}\}$ are selected, ready for the generation of r'_{i-n+2} . The value of $b_{i-2n-f+2}$ is now detected as the value of $x_{b,i-2n-f+2}$ in the stored vector $X_{b,i-n-f+1}$ that is associated with the smallest $d_{i-n-f+1}$.

The selection process for the vectors $\{X_{b,i-n-f+1}\}$ should be that of System 5B, the selection process being of course that for a 4-level signal in which $x_{b,i-n-f+1}$ is the latest received data symbol, this being taken as that whose first significant component $x_{b,i-n-f+1} b_f$ is present.

On the receipt of r_{i+f+1} , the $m \leq k$ stored vectors $\{X_{a,i}\}$ are expanded into $4m$ vectors, the value of m

depending upon the number of vectors that were discarded because $x_{a,i-n+1}$ was not equal to the detected value of a_{i-n+1} . The discarding of some of the stored vectors here means that the selection process must be that of System 5B and not System 3B, since the latter selection process will inevitably not always be able to select a full set of k vectors, whereas System 5B can and should now be arranged so that the numbers of vectors selected according to System 1B is always such as to make a total of k selected vectors.

The advantage of the arrangement just described is that a large number of column scans can now be achieved quite easily since the symbols $x_{a,h}$ and $x_{b,h}$ are processed quite separately and there is no question of selecting different combinations of values of $x_{a,h}$ and $x_{b,h}$. For instance, when there are 32 stored vectors $\{X_{a,i}\}$ and 32 stored vectors $\{X_{b,h}\}$, the selection process of System 3B would give 8 column scans for both the $\{X_{a,i}\}$ and the $\{X_{b,h}\}$. This compares with only 2 or 3 column scans obtainable with the arrangements previously described. Furthermore, the new arrangement is no more complex than those previously described.

The disadvantage of the new arrangement is that in the selection of the vectors $\{X_{a,i}\}$, the components $\{x_{b,h}\}$ are totally ignored and this could lead to a poor performance.

The weakness of the new arrangement may possibly be overcome by the following simple modification. When the m stored vectors $\{X_{a,i-1}\}$ are expanded, on the receipt of Y_{i+f} each expanded vector

$$x_{a,i-n} x_{a,i-n+1} \cdots x_{a,i-1} x_{a,i}$$

is associated with the corresponding vectors

$$x_{b,i-n} x_{b,i-n+1} \cdots x_{b,i-1} x_{b,i}$$

where $x_{b,i}$ is taken as the one of its four possible values that minimizes the cost d_i of the expanded vectors. Both the $\{x_{a,h}\}$ and $\{x_{b,h}\}$ are now involved in computing d_i , and the $\{x_{b,h}\}$ associated with the expanded vector

$$x_{a,i-n} x_{a,i-n+1} \cdots x_{a,i-1} x_{a,i}$$

were each determined in exactly the same way as was $x_{b,i}$, that is, when forming the last component of

the corresponding vector. Associated with each of the k selected vectors $\{X_{a,i}\}$ is the corresponding vector $X_{b,i}$.

Although the technique used here for determining the $\{x_{b,h}\}$ does not constitute a particularly good selection process, they involve simply the selection of one of the four possible values of $x_{b,i}$ after the selection of $x_{a,i}$ in any expanded vector, and so do not greatly increase the complexity of the system. This may be enough to give a useful improvement in the selection of the k stored vectors $\{X_{a,i}\}$. The $\{q\}$ and $\{r'\}$ are formed exactly as before, and the second part of the receiver again forms the k stored vectors $\{X_{b,i-n-f}\}$, exactly as described, the components $\{x_{b,l}\}$ here not necessarily being the same as any $\{x_{l,h}\}$ involved in the selection of the $\{X_{a,i}\}$.

Another disadvantage of the technique just described is that the delay in detection is approximately doubled, but this probably is not too serious. The essential virtue of the system is its simplicity coupled with the great increase in the numbers of column scans.

The reason for the discarding of the vectors $\{X_{a,i}\}$ is to insure that there is only one sequence of values

$$x_{a,i-2n-f-g+2} \ x_{a,i-2n-g-f+3} \ \cdots \ x_{a,i-n+1}$$

which are used to form the $\{q\}$ this having in fact already been done in the evaluation of the $\{d_i\}$, so that no additional computation is involved in forming the $\{r'\}$. If however, the discarding of the vectors is found to degrade the performance of the detector or to lead to undesirable complexity, the $\{a'\}$ could be used to form the $\{q\}$ in an additional operation, there being now no need to store any $\{x_{a,h}\}$ earlier than $x_{a,i-n+1}$ and no need to discard any vectors. This is a slightly more complex system, but more versatile and less likely to lead to awkward problems. It could therefore well be the better approach. Indeed the small increase in complexity involved is the use of the $\{a'_h\}$ to generate the $\{r'_h\}$ as a special operation should be more than affected by the complications introduced by the discarding of vectors.

A final point is that a slightly more complex and effective selection process could possibly be used

for the $\{x_{b,h}\}$ that are associated with the selection of the $\{X_{a,i}\}$, but of course this quickly leads back to the arrangements described in Chapter 6 so that there is not too much scope for change here. There is in any case no purpose in testing more complicated selection processes for the $\{x_{b,h}\}$ unless a satisfactory performance is not obtained with the other systems described here which should be tested first.

The basic system just described is about as far as one can go in simplifying the detection process for multilevel signals. The basic system can itself be expanded as follows: Just as the receiver uses the finally detected values of the $\{a_l\}$, for $l = i-2n-g-f+2, i-2n-g-f+3, \dots, i-n+1$, to generate the sequence of $n+f$ symbols $\{q_l\}$, for $l = i-2n-f+2, i-2n-f+3, \dots, i-n+1$, where

$$q_l = \sum_{h=0}^g l_{-h} y_h$$

and then the corresponding sequence of $n+f$ symbols $\{r'_l\}$ for $l = i-2n-f+2, i-2n-f+3, \dots, i-n+1$, where

$$q_l = \sum_{h=0}^g l_{-h} y_h$$

so the receiver uses the finally detected values of the $\{b_l\}$ for $l = i-3n-2f-g+3, i-3n-2f-g+4, \dots, i-2n-f+2$ to generate the sequence of $n+f$ symbols $\{q_l\}$, for $l = i-3n-2f+3, i-3n-2f+4, \dots, i-2n-f+2$, where

now

$$q_l = \sum_{h=0}^g l_{-h} y_h$$

and then the corresponding sequence of $n+f$ symbols $\{r'_l\}$, where $r'_l = r_l - q_l$ as before. This is an arrangement of decision directed cancellation of intersymbol interference such that with the correct detection of the $\{b_l\}$

$$\begin{aligned} r'_l &= r_l - q_l \\ &= \sum_{h=0}^g (l_{-h} + b_{l-h}) y_h + w_l - b_{l-h} y_h \\ &= \sum_{h=0}^g l_{-h} y_h + w_l \end{aligned}$$

The receiver now operates on these $\{r'_l\}$, assuming the correct detection of the $\{b_l\}$ and so that the above equation holds. Thus, just prior to the generation of $r'_{i-2n-f+2}$ (which is the

of the $n+f$ symbols $\{r' \ell\}$ the receiver holds in store k vectors $\{X_{a,i-2n-2f+1}\}$ where

$$X_{a,i-2n-2f+1} = x_{a,i-3n-2f+2} x_{a,i-3n-2f+3} \dots x_{a,i-2n-2f+1}$$

On the generation of $r'_{i-2n-f+2}$, the k stored vectors $\{X_{a,i-2n-2f+1}\}$ are expanded to give $4k$ vectors

$$\{x_{a,i-3n-2f+2} x_{a,i-3n-2f+3} \dots x_{a,i-2n-2f+2}\}$$

and their costs $\{d_i\}$ are evaluated on the assumption that

$$r' = \sum_{h=0}^g \ell_{-h} y_h + w \ell$$

Then k vectors $\{X_{a,i-2n-2f+2}\}$ are selected ready for the generation of $r'_{i-2n-f+3}$. The value of

$a_{i-3n-2f+3}$ is now finally detected as the value of $x_{a,i-3n-2f+3}$ in the stored vector $X_{a,i-2n-2f+2}$ that is associated with the smallest d_i . The selection process for the vectors $\{X_{a,i-2n-2f+2}\}$ should be that of System 5B as before (Notice that d_i here is of course $d_{i-2n-2f+2}$).

Following the final detection of $a_{i-3n-2f+3}$ the receiver has the sequence of $n+f+g$ finally detected values

$$a'_{i-4n-3f-g+4} a'_{i-4n-3f-g+5} \dots a'_{i-3n-2f+3}$$

and it uses these to form the sequence of $n+f$ symbols $\{q \ell\}$, where now

$$q = \sum_{h=0}^g \ell_{-h} y_h$$

and $\ell = i-4n-3f+4, i-4n-3f+5, \dots, i-3n-2f+3$ and then to form the corresponding sequence of $n+f$ symbols $\{r' \ell\}$ where

$$r' \ell = r \ell - q \ell$$

and $\ell = i-4n-3f+4, i-4n-3f+5, \dots, i-3n-2f+3$

The receiver now operates on these $\{r' \ell\}$ assuming the correct detection of the $\{a \ell\}$, so that

$$r' \ell = \sum_{h=0}^g \ell_{-h} y_h + w \ell$$

Thus, just prior to the generation of $r'_{i-3n-2f+3}$, the receiver holds in store k vectors $\{X_{b,i-3n-3f+2}\}$

where

$$X_{b,i-3n-3f+2} = x_{b,i-4n-3f+3} x_{b,i-4n-3f+4} \dots x_{b,i-3n-3f+2}$$

On the generation of $r'_{i-3n-2f+3}$, the k stored vectors $\{X_{b,i-3n-3f+3}\}$ are expanded to give $4k$ vectors

$$\{x_{b,i-4n=3f+3} \ x_{b,i-4n-3f+4} \ \dots \ x_{b,i-3n-3f+3}\}$$

and their costs are evaluated on the assumption that

$$r'_{\ell} = \sum_{h=0}^g \ell_{-h} y_h + w_{\ell}$$

k vectors $\{X_{b,i-3n-3f+3}\}$ are then selected, ready for $r'_{i-3n-2f+4}$, and so on.

The value of $b_{i-4n-3f+4}$ is now finally detected as the value of $x_{b,i-4n-3f+4}$ in the stored vector $X_{b,i-3n-3f+3}$ that is associated with the smallest $d_{i-3n-3f+3}$. As before, the selection process of System 5B should be used.

It can be seen that by expanding the detection process in the manner described, more reliable values are used for the $\{a'_{\ell}\}$ and $\{b'_{\ell}\}$ involved in the generation of $\{r'_{\ell}\}$ through decision directed cancellation of intersymbol interference, for the final detection of the $\{a_{\ell}\}$ and $\{b_{\ell}\}$. This should lead to a better detection process. Experience with similar techniques used in other applications suggests that no further useful improvement is likely to be achieved by any further expansion, but that a useful improvement should be achieved by the arrangement described. The weakness of this system is that it involves a doubling of the equipment complexity, since it doubles both the amount of storage required and the number of operations. It also doubles the delay in detection. Notice that the technique of discarding vectors has been abandoned here; it does not, on balance, appear to be a desirable thing to do.

Chapter 8

Adaptive Arrangement of Near-Maximum Likelihood Detection Process

Consider first one of the detection processes, Systems 1A-5A, described in the paper *Detection processes for severely distorted digital signals*. Just prior to the reception of the sample

$$r_j = \sum_{h=0}^g j-h y_h + w_j \quad (8.1)$$

where $Y = y_0 y_1 \dots y_g$ (8.2)

is the sampled impulse-response of the channel, the receiver holds in store k vectors $\{X_{j-1}\}$, where

$$X_{j-1} = x_{j-n} x_{j-n+1} \dots x_{j-1} \quad (8.3)$$

where x_i has one of the m possible values of the data symbol s_i . Associated with each stored vector X_{j-1} is the cost $|U_{j-1}|^2$, where

$$|U_{j-1}|^2 = u_1^2 + u_2^2 + \dots + u_{j-1}^2 \quad (8.4)$$

and $u_i = r_i - \sum_{h=0}^g x_{i-h} y_h$ (8.5)

Notice that the vector X_{j-1} only contains the last n components of the corresponding complete $(j-1)$ component vector

$$x_1 x_2 \dots x_{j-1}$$

which together with the $(j-1)$ component vector

$$r_1 r_2 \dots r_{j-1}$$

determines the associated $|U_{j-1}|^2$

Following the receipt of r_j each stored vector X_{j-1} is expanded into $m(n+1)$ - component vectors

$$x_{j-n} x_{j-n+1} \dots x_{j-1} x_j$$

where $x_{j-n}, x_{j-n+1}, \dots, x_{j-1}$ have the values in X_{j-1} , for each of the m expanded vectors, but x_j has its m different possible values in the m vectors. There are now mk expanded vectors. The cost

$$|U_j|^2 = |U_{j-1}|^2 + \left(r_j - \sum_{h=0}^g x_{j-h} y_h \right)^2 \quad (8.6)$$

of each of these is now evaluated using the appropriate value $|U_{j-1}|^2$ which has already been determined.

The detector then selects from the mk expanded vectors, k vectors according to one of the selection processes of the Systems 1A-5A. It removes the first component, x_{j-n} , from each of these vectors to give the k vectors $\{X_j\}$, which are stored together with their associated costs. The detected value of s_{j-n+1} is taken as the value of the component x_{j-n+1} in the stored vector

$$X_j = x_{j-n+1} x_{j-n+2} \dots x_j \quad (8.7)$$

which is associated with the smallest cost.

The arrangement described so far is the conventional arrangement of any one of the Systems 1A-5A, as described in the paper. The adaptive detection process is derived from this in the following changes:

The first change is to operate the detection process exactly as just described, but after a delay of p sampling intervals, where $1 \leq p \leq g$. The parameter p is in fact the greatest delay (in sampling intervals) in the location of the first "significant" component y_f of the sampled impulse response of the channel (relative to the first component y_0), likely to be experienced over the time-varying channel. It is therefore the greatest value of f for the System 1B-5B in the paper. Thus, instead of expanding the k stored vectors $\{X_{j-1}\}$ into mk vectors and then selecting k vectors $\{X_j\}$ immediately after the receipt of Y_j , this same operation is carried out immediately after the receipt of r_{j+p} .

The second change is that just prior to the reception of r_{j+p} , the detector holds in store the p quantities

$\{z_{j-1,i}\}$ for $i = 1, 2, \dots, p$

$$\text{where } z_{j-1,i} = \sum_{h=0}^{g-i} j-1-h y_{h+i} \quad (8.8)$$

Immediately after the receipt of r_{j+p} , but before any other operation, the detector evaluates

$$z_{j-1,p+1} = \sum_{h=0}^{g-p-1} j-1-h y_{h+p+1} \quad x \quad (8.9)$$

The appropriate $p+1$ quantities $\{z_{j-1,i}\}$ are now associated with each of the k stored vectors

$\{X_{j-1}\}$. $z_{j-1,i}$ is an estimated signal component in r_{j-1+i} .

The third change is that following the expansion of the k stored vectors $\{X_{j-1}\}$, after the reception of r_{j+p} , to give mk vectors

$$\{x_{j-n} \ x_{j-n+1} \ \dots \ x_j\}$$

the detector now decides that y_f is the first significant component of Y , where f is some positive integer in the range of 0 to g . This is done by examining Y and applying a suitable selection criterion for f (the criterion probably being determined by computer simulation).

The fourth change is that having made the above decision, the detector replaces $z_{j-1,f+1}$, as given by equation 8.8 by

$$z_{j,f} = \sum_{h=0}^{g-f} j-f y_{h+f} \quad x \quad (8.10)$$

This is achieved by adding $x_j y_f$ to $z_{j-1,f+1}$. The $\{z_{j-1,i}\}$ for $i = 1, 2, \dots, p+1$ and $i \neq f+1$ are left unchanged.

For the m expanded vectors $\{x_{j-n} \ x_{j-n+1} \ \dots \ x_j\}$

derived from any one X_{j-1} , x_j has its m possible values. It follows that for these expanded vectors, $z_{j,f}$ has m different values, but any $z_{j-1,i}$ has the same value (since it is independent of x_j).

The fifth change is that the detector, instead of associating with each of the mk expanded vectors the

corresponding cost $|U_j|^2$ as determined by equation 8.6, evaluates

$$v_i^2 = (r_{j-1+i} - z_{j-1,i})^2 \quad (8.11)$$

for $i = 1, 2, \dots, f$, and also

$$v_{f+1}^2 = (r_{j+f} - z_{j,f})^2 \quad (8.12)$$

and then sets

$$d_j = |U_{j-1}|^2 + \sum_{i=1}^{f+1} v_i^2 \quad (8.13)$$

Each expanded vector is now associated with the corresponding cost d_j . Using these costs, the k vectors $\{X_j\}$ are selected from the expanded vectors according to the selection process of one of the Systems 1A-5A. [The selected vectors are stored together with the associated $\{d_j\}$]. The detector also evaluates

$$|U_j|^2 = |U_{j-1}|^2 + \left(r_j - \sum_{h=0}^g x_{j-h} y_h \right)^2 \quad (8.14)$$

for each stored vector, using the given value of $|U_{j-1}|^2$, ready for the next detection process.

The sixth and final change is that for every stored vector X_j each

$$z_{j-1,i} = \sum_{h=0}^{g-i} x_{j-1-h} y_{h+i} \quad (8.15)$$

is now replaced by the corresponding

$$z_{j,i-1} = \sum_{h=0}^{g-i+1} x_{j-h} y_{h+i-1} \quad (8.16)$$

for $i = 2, 3, \dots, p+1$, $z_{j-1, f+1}$ having already been replaced by $z_{j,f}$. It can be seen from equations 8.15 and 8.16 that

$$z_{j,i-1} = z_{j-1,i} + x_j y_{i-1} \quad (8.17)$$

so that no great complexity is involved in this change.

The value of s_{j-n+1} is finally detected as the value of x_{j-n+1} in the stored vector X_j associated with the

smallest d_j .

At this stage, the receiver holds in store k vectors $\{X_j\}$ [together with the associated $\{d_j\}$]. It also holds the quantities

$$z_{j,1}, z_{j,2}, \dots, z_{j,p}$$

and $|U_j|^2$. It is now ready to receive r_{j+p+1} .

Since in any set of m expanded vectors that are derived from any one vector X_{j-1} , $z_{j-1,i}$ has the same value in all m expanded vectors for any given i , since $z_{j-1,i}$ is independent of x_j , it is clear that v_i^2 has the same value in all m expanded vectors, for any given i in the range 1 to f , as can be seen from equation 8.11. Thus, in the evaluation of d_j for any expanded vector,

$$c_j = \sum_{i=1}^f v_i^2 = \sum_{i=1}^f (r_{j-1+i} - z_{j-1,i})^2 \quad (8.18)$$

has the same value for all m expanded vectors that are derived from any one X_{j-1} . Thus

$$d_j = |U_{j-1}|^2 + c_j + (r_{j+f} - z_{j,f})^2 \quad (8.19)$$

and only the component $(r_{j+f} - z_{j,f})^2$ in d_j differs in any two of these expanded vectors.

It can be seen that in the detection process just described, no unduly great amount of computation is required.

The important property of this detection process is that it is fully adaptive, in the sense that the value of f chosen for any one detection process in no way affects the operation of the process for this or any other value of f in any other detection process, and in each case the operation performed by the process is that appropriate for the given value of f , so that there is no degradation in performance due simply to a change in the value of f .

The detection process can be simplified by using the "moduli-sum" as a distance measure (cost function) in place of the Euclidean distance (or sum of squares). Now

$$d_j = |U_{j-1}| + \sum_{i=1}^{f+1} |v_i| \quad (8.20)$$

where $|v_i| = |r_{j-1+i} - z_{j-1,i}|$ (8.21)

for $i = 1, 2, \dots, f$, and

$$|v_{f+1}| = |r_{j+f} - z_{j,f}| \quad (8.22)$$

and $|U_{j-1}| = |U_{j-2}| + |r_{j-1} - \sum_{h=0}^g x_{j-1-h} y_h|$ (8.23)

Both of these arrangements should be tested to determine the loss in tolerance to noise incurred in using the moduli-sum distance measure.

It can be seen that the moduli-sum distance measure avoids altogether the need for any multiplication or squaring operations and should therefore considerably reduce the amount of computation required.

Thus if the reduction in tolerance to noise that results from the use of the moduli-sum is not much more than 1/2 dB, this is probably the preferred system. It is assumed here that a 4-point QAM signal is used, in which case, the equipment can handle x_i as having one of the four values $\pm 1 \pm j$, which means that $x_i y_h$ is evaluated from y_h simply through sign changes and addition or subtraction.

Chapter 9

Detection Process for a Time-Varying Channel

A technique has recently been developed whereby it is possible, without undue complexity and without any adaptive linear prefiltering, to achieve near-maximum-likelihood detection of a sampled digital signal, where there is intersymbol interference extending over several samples of the signal. It is, however, important here that the channel impulse-response does not undergo large changes with time, since when it does, a considerable increase in the complexity of the system may be required to maintain correct operation. The paper describes a development of the detection process whereby correct operation is achieved with a relatively simple system, even when the channel introduces severe frequency-selective fading of the type sometimes experienced over HF radio links. Results of computer-simulation tests are presented, showing the tolerance of a synchronous serial data-transmission system to additive white Gaussian noise, when a 4-point quadrature amplitude modulated signal is transmitted at 2400 bits/second over a model of an HF radio link, with two independent Rayleigh fading sky waves and frequency spreads of $\frac{1}{2}$, 1 and 2 Hz, and when the novel detection process is used at the receiver. Correct estimation of the channel is assumed throughout.

List of Principal Symbols

1. a = parameter involved in determination of f
2. b_i = magnitude-sum distance between r_{k+i} and $z_{k-1,i+1}$, for $i=0, 1, \dots, f-1$
3. c_i = square of unitary distance between r_{k+i} and $z_{k-1,i+1}$, for $i=0, 1, \dots, f-1$
4. c'_i = square of unitary distance between r'_{k+i} and $z'_{k-1,i+1}$ for $i=0, 1, \dots, f'-1$
5. d_k = cost associated with either a stored vector Q_k or an expanded vector P_k
6. f = location of first significant component in Y_i
7. f' = location of first significant component in Y'_i
8. $g+1$ = number of components in Y_i of Y'_i
9. $j = \sqrt{-1}$
10. m = number of stored vectors held by detector
11. n = number of components in Q_k
12. Q_k = n -component row vector whose i th component is x_{k-n+i}
13. r_i = sample value of demodulated baseband signal at time $t=iT$
14. r'_i = sample value of demodulated baseband signal at time $t=(i-1/2)T$
15. R_k = k -component row vector whose i th component is r_i
16. s_i = 4-level data symbol
17. s'_i = detected value of s_i
18. s_k = k -component row vector whose i th component is s_i
19. T = reciprocal of signal element rate
20. u_i = possible value of w_i
21. U_k = k -component row vector whose i th component is u_i

22. w_i = Gaussian noise component in r_i
23. w'_i = Gaussian noise component r'_i
24. w_k = k-component row vector whose i th component is w_i
25. x_i = possible value of s_i
26. X_k = k-component row vector whose i th component is x_i
27. $y_{i,h} = (h+1)$ the component of Y_i
28. $y'_{i,h} = (h+1)$ the component of Y'_i
29. Y'_i = sampled impulse-response of resultant baseband channel at time $t=(i-1/2)T$
30. z_i = possible value of resultant (total) data-signal in r_i
31. $z_{k-1,i+1}$ = possible component of resultant data-signal in r_{k+i}
32. $z'_{k-1,i+1}$ = possible component of resultant data-signal in r'_{k+i}
33. z_k = k-component row vector whose i th component is z_i
34. Δ = relative transmission delay of two sky waves
35. ϕ = largest value of the integer f or f'
36. σ^2 = variance of w_i

1 Introduction

It is well known that when a transmitted digital signal has been subjected to severe amplitude distortion, which may vary considerably during any one transmission or from one transmission to another, a useful overall improvement in tolerance to additive white Gaussian noise can be achieved by using a maximum-likelihood detector in place of a nonlinear (decision-feedback) equalizer.¹⁻⁸ Unfortunately, in the implementation of a maximum-likelihood detector by means of the Viterbi-algorithm, an excessive amount of storage and an excessive number of operations per received data symbol are involved whenever the sampled impulse-response of the channel has more than a few non-zero components.¹⁻³ This can be avoided by inserting an adaptive linear feedforward transversal filter ahead of the Viterbi-algorithm detector, the adaptive filter being adjusted to restrict the number of components in the resultant sampled impulse-response of the channel and filter to an acceptable value.⁹⁻¹⁶ However, when the channel varies with time, there may now be difficulties in holding the linear filter correctly adjusted to achieve the required resultant sampled impulse-response, and furthermore there may from time to time be a excessive reduction in tolerance to noise relative to that obtained with a true maximum-likelihood detector, due to the partial linear equalization of the amplitude distortion introduced by the channel.¹

An alternative approach that has recently been studied is to modify the detection process itself so that this operates without undue complexity as a near-maximum-likelihood detector, when there are several components in the sampled impulse-response of the channel.^{1, 17-19} The adaptive linear filter that must in practice be used with a Viterbi-algorithm detector can now be discarded, thus avoiding the problems involved in the adaptive adjustment of the linear filter. For the correct operation of the near-maximum-likelihood detector, the sampled impulse-response of the channel must of course be estimated, but extensive computer-simulation tests (whose details are being published separately) have suggested that this can be achieved quite simply and to the required degree of accuracy, for the particular channels under consideration here, through the use of appropriate developments of known techniques.²⁰⁻²²

Unfortunately, when the sampled impulse-response of the channel undergoes large changes during any one transmission, alterations must be made, from time to time, in a critical parameter value of the near-maximum-likelihood detector (to be described later) and this can introduce appreciable complexity into the system. Some basic modifications have therefore been made to this detector to give a system that operates correctly and without complex equipment over a time-varying channel of the type just mentioned.

The paper describes first the near-maximum-likelihood detector, adapted for use with a channel having a time-varying complex-valued sampled impulse-response, and it then describes the various modifications that have been made to this detector. The paper then presents the results of computer-simulation tests on the novel detector operating on a signal that has been transmitted over a model of an HF radio link.

2 Model of System

The model of the data-transmission system is shown in Figure 9-1. The data-symbols $\{s_i\}$ are statistically independent and equally likely to have any of the four values $\pm 1 \pm j$, where $j = \sqrt{-1}$.

The baseband signal generator and linear modulator convert the input data-symbols $\{s_i\}$ into a serial stream of signal elements, with an 1800 Hz carrier and an element rate of 1200 bauds, each signal element itself comprising the sum of two binary double sideband suppressed carrier amplitude modulated elements with their carriers in phase quadrature, the binary values of the "in-phase" and "quadrature" elements being determined, respectively, by the real and imaginary parts of the corresponding data-symbol s_i . The resulting 4-point (4-level) QAM signal is fed to the HF radio link where its spectrum is shifted into the HF band, by a process of linear single sideband suppressed carrier amplitude modulation, the resulting signal being then transmitted via two independently Rayleigh fading sky waves to the receiver, where its spectrum is returned to the voice band by a process of linear single sideband suppressed carrier amplitude modulation.

The linear demodulator in Figure 9-1 includes at its input a band-pass filter that removes as much of the Gaussian noise from the received signal as is possible without unduly distorting the data signal. The filtered signal is fed to two linear coherent demodulators whose reference carriers are in phase quadrature and have the same frequency, which is constant and equal to the average instantaneous frequency of the received signal carrier, thus eliminating any constant frequency offset in the received QAM signal. The demodulated signals at the outputs of the "inphase" and "quadrature" coherent demodulators are taken to be real and imaginary valued, respectively, so that the resultant demodulated baseband signal $r(t)$ is complex valued.

Two different designs of the equipment filters involved in the baseband signal generator, linear modulator and linear demodulator (Figure 9-1) have been studied and are referred to here as the "ideal" and "actual" filters. When the ideal filters are used and when the HF radio link is omitted from Figure 9-1, the

demodulated baseband waveform in the receiver, that results from the transmission of the single data-symbol s_i in the absence of any noise, is shaped entirely by the equipment filters and is

$$r(t) = \frac{1}{2} s_i \left(1 + \cos \frac{\pi t}{T} \right) \quad (9.1)$$

over the time interval $(i-1)T < t < (i+1)T$, being zero elsewhere. The delay introduced by the filters is neglected here, and T ($+0.8333 \times 10^{-3}$) seconds is, of course, the sampling interval. The reason for considering the given short-duration impulse response of the channel is that a Viterbi-algorithm detector can now be implemented at the receiver (at least for the purposes of computer simulation), so that the performances of the near-maximum-likelihood detectors under investigation here can be compared with that of a true maximum-likelihood detector. The response, corresponding to $r(t)$ in equation (9.1), given by the actual filters, is shaped over its first part not very differently from $r(t)$ in equation (9.1), but this is followed by quite a long tail, which makes Viterbi-algorithm detection impractical.

The influence of each sky wave on the transmitted data signal is as shown in Figure 9-2, where $q_1(t)$ and $q_2(t)$ are sample functions of stationary zero-mean Gaussian random processes, each with a low-pass Gaussian spectral shaping and a root-mean-square bandwidth of $\frac{1}{4}$, $\frac{1}{2}$ or 1 Hz, giving a frequency spread of $\frac{1}{2}$, 1 or 2 Hz, respectively, into the output data signal. The four Gaussian waveforms involved in the two sky waves are statistically independent, with the same variance and the same bandwidth so that the two output signals have the same mean-square value and frequency spread. The relative delay in transmission of the two sky waves is taken to have a constant value of Δ seconds. The only additive noise introduced in the transmission path is stationary band-limited Gaussian noise with zero mean and a flat (frequency independent) power spectral density over the whole of the signal frequency band, which is added to that data signal at the output of the HF radio link. The important properties of the three different channels used in the tests are summarized in Table 1.

The demodulated waveform of any received signal-element (carrying that data-symbol s_i) is the sum

of two complex-valued pulses, whose peaks are separated by Δ seconds and generally have quite different values. The demodulated signal $r(t)$, comprising the stream of demodulated signal-elements to which is added a stationary zero-mean baseband Gaussian noise waveform, is sampled, once per received signal-element at the time instants $\{iT\}$. The phase of the sampling instants is assumed to be held adaptively adjusted such that a received signal-element is sampled at the peak of the first of the two pulses that make up its waveform. The delay in transmission over the channel involving the first of the two sky-waves is for convenience neglected here.

The complex-valued sample of the demodulated baseband signal $r(t)$, at time $t=iT$, is

$$r_i = \sum_{h=0}^g i-h Y_{i,h} + w_i \quad (9.2)$$

and the sequence of complex values given by the vector

$$Y_i = Y_{i,0} \quad Y_{i,1} \cdots Y_{i,g} \quad (9.3)$$

is taken to be the "sampled impulse-response" at time $t=iT$ of the linear baseband channel formed by the baseband signal generator, linear modulator, HF radio link, linear demodulator and sampler. It is assumed throughout that $y_{i,h} = 0$ for $h < 0$ and $h > g$. The real and imaginary parts of the complex-valued noise components $\{w_i\}$ are Gaussian random variables with zero mean and variance σ^2 . With the ideal equipment filters the real and imaginary parts of the $\{w_i\}$ are uncorrelated, whereas with the actual filters they are slightly correlated.

Since in this investigation we are not concerned with techniques for estimating the sampled impulse-response of a channel, such techniques having been studied elsewhere,²⁰⁻²² but rather with the performance of the near-maximum-likelihood detector relative to an ideal maximum-likelihood detector, it will be assumed throughout that the detector has exact knowledge of Y_i for all $\{i\}$. The results presented for any detector therefore represent the upper bound to the performance that can be achieved with the given detector, when an estimate of Y_i is used in place of its actual value. Tests with an adaptive Viterbi-algo-

rithm detector have however suggested that, for the time-varying channel used here and with a suitably designed channel estimator, no very serious loss in tolerance to noise need result when using an estimate of Y_i in place of its real value.

In a practical implementation of the system described here, differential coding would be used at the transmitter with the corresponding differential decoding of the detected signal at the receiver. Since the purpose of this investigation is to study the performance of the new detector relative to the Viterbi-algorithm detector, in which case differential coding has no significant effect, and since, in any case, the correct estimation of the channel is assumed, differential coding is not used.

3 Near-Maximum-Likelihood Detection Process

Let S_k , R_k and W_k be the k -component row-vectors (sequences) whose i th components are s_i , r_i and w_i , respectively, for $i=1, 2, \dots, k$. Also let X_k , Z_k , and U_k be the k -component row-vectors whose i th components are x_i , z_i and u_i , respectively, for $i=1, 2, \dots, k$, where x_i has one of the four possible values of s_i .

$$z_i = \sum_{h=0}^g i-h Y_{i,h} \quad (9.4)$$

and u_i is the possible value of w_i satisfying

$$r_i = z_i + u_i \quad (9.5)$$

In the k -dimensional complex vector-space containing the vectors R_k , Z_k and U_k , the square of the "unitary" distance between the vectors R_k and Z_k is

$$|U_k|^2 = |u_1|^2 + |u_2|^2 + \dots + |u_k|^2 \quad (9.6)$$

where $|u_i|$ is the absolute value (modulus) of u_i . When all real and imaginary parts of the $\{w_i\}$ are statistically independent, with zero mean and the same variance, the maximum-likelihood vector X_k is its possible value such that $|U_k|^2$ is minimized. Under the assumed conditions X_k is the possible value of S_k most likely to be correct.

Suppose now that in the sampled impulse-response of the channel (equation (9.3)), $Y_{i,f}$ is the first component of "significant" magnitude. f is here taken to be the smallest integer such that

$$\|y_{i,f}\| > a \|y_{i,\ell}\| \quad (9.7)$$

where $a = 0.7$, $\|y\|$ is the sum of the magnitudes of the real and imaginary parts of y , and

$$\|y_{i,\ell}\| > \|y_{i,h}\| \quad (9.8)$$

for all $\{h\}$ other than $h = \ell$.

The detector temporarily ignores the first f components $x_i Y_{i,0}$, $x_i Y_{i+1,1}$, \dots , $x_i Y_{i+f-1,f-1}$ of any

received signal-element until the $(f+1)$ th component $x_i y_{i+f}$ is received when all received components of the signal element are taken account of in the detection process.

Just prior to the receipt of the sample r_{k+f} the detector holds in store m different vectors $\{Q_{k-1}\}$,

where

$$Q_{k-1} = x_{k-n} x_{k-n+1} \dots x_{k-1} \quad (9.9)$$

each vector being associated with the corresponding cost (distance)

$$d_{k-1} = |U_{k-1}|^2 + \sum_{i=0}^{f-1} |r_{k+i} - \sum_{h=1}^{g-i} x_{k-h} y_{k+i, i+h}|^2 \quad (9.10)$$

which is also stored. m is a multiple of 8. In the evaluation of d_{k-1} , the detector here considers all $\{x_i\}$ for $i \leq k-1$ and ignores all $\{x_i\}$ for $i > k-1$.

On the receipt of r_{k+f} each of the stored vectors $\{Q_{k-1}\}$ is expanded into four $(n+1)$ - component vectors $\{p_k\}$, where

$$p_k = x_{k-n} x_{k-n+1} \dots x_k \quad (9.11)$$

The first n components of each of the four vectors $\{p_k\}$ derived from any one vector Q_{k-1} , are as in the original vector Q_{k-1} , and the last component x_k takes on its four different possible values in the four vectors. The cost

$$d_k = |U_{k-1}|^2 + \sum_{i=0}^f |r_{k+i} - \sum_{h=0}^{g-i} x_{k-h} y_{k+i, i+h}|^2 \quad (9.12)$$

of each of the $4m$ expanded vectors is now evaluated and the detector selects from these vectors a set of m vectors $\{Q_k\}$ together with their associated costs $\{d_k\}$ in one of the following three different ways referred to here as systems 1, 2 and 3.

System 1 starts by selecting for each of the four different possible values of $x_{k-\mu+2}$ where $\mu = m/4$, the vector Q_k with the given value of $x_{k-\mu+2}$ and associated with the smallest d_k . The process is then repeated in turn for $x_{k-\mu+3}, x_{k-\mu+4}, \dots, x_k$, to give a total of $m-4$ selected vectors. A vector once selected is not

available for selection a second time. A further four vectors $\{Q_k\}$ are finally selected from the remaining (non selected) vectors as those with the smallest $\{d_k\}$, regardless of the values of any of their $\{x_i\}$. This gives the required total of m selected vectors $\{Q_k\}$, which are stored together with their costs $\{d_k\}$. the detected value s'_{k-n+1} of the data-symbol s_{k-n+1} is now taken as the value of x_{k-n+1} in the stored vector Q_k associated with the smallest d_k .

System 2 is a simple modification of system 1. It starts by selecting for each of the four different possible values of $x_{k-\lambda+1}$ where $\lambda = m/8$, the vector Q_k with the given value of $x_{k-\lambda+1}$ and associated with the smallest d_k . The process is then repeated, in turn, for $x_{k-\lambda+2}, x_{k-\lambda+3}, \dots, x_k$, to give a total of $\frac{1}{2}m$ selected vectors. A further $\frac{1}{2}m$ vectors' $\{Q_k\}$ are finally selected from the remaining vectors as those with the smallest $\{d_k\}$ regardless of the values of any of their $\{x_i\}$. The system operates otherwise in exactly the same way as system 1.

System 3 starts by selecting the expanded vector p_k that is associated with the smallest d_k and taking the value of x_k in this vector as the detected value s'_k of the data-symbol s_k . The system then discards all vectors $\{p_k\}$ for which $x_k \neq s'_{k-n}$, and from the remaining vectors $\{p_k\}$ (including, of course, the one originally chosen) selects the m vectors $\{Q_k\}$ associated with the smallest $\{d_k\}$. It can be seen from equations (9.10) and (9.12) that if it is necessary to change to the value of f in order to maintain the correct operation of the detector, appreciable complexity may be involved in the next evaluation of the $\{d_k\}$. The detection process has therefore been substantially modified by introducing the following changes into its method of operation.

4 Detection Process for a Time-Varying Channel

Suppose that in the given application, the largest value of f is ϕ , where $\phi \geq g$. Just prior to the receipt of $r_{k+\phi}$ (not r_{k+f}) the detector here holds m stored vectors $\{Q_{k-1}\}$ together with their associated costs $\{d_{k-1}\}$. It also holds at this time associated with each vector Q_{k-1} , the ϕ quantities

$$\{z_{k-1, i+1}\} \quad i = 0, 1, \dots, \phi-1$$

where

$$z_{k-1, i+1} = \sum_{h=1}^{g-i} k-h y_{k+i, i+h} \quad (9.13)$$

and $z_{k-1, i+1}$ is an estimated (possible) component of the resultant data signal in r_{k+i} .

After the receipt of $r_{k+\phi}$, the detector evaluates $z_{k-1, \phi+1}$ (equation (9.13) with $i = \phi$) for every Q_{k-1} , this being the corresponding estimated (possible) component of the resultant data signal in $r_{k+\phi}$. Then following the expansion of the m stored vectors $\{Q_{k-1}\}$, the detector selects the value of f by applying equation (9.7) to the prediction of $y_{k+\phi}$, the latter being assumed here to be correct and in practice being evaluated in the channel estimator (not shown in Figure 9-1).

In the arrangement A of the detection process, the detector replaces every $z_{k-1, i+1}$ for $i = 0, 1, \dots, f$ and for each of the $4m$ expanded vectors, by

$$z_{k, i} = \sum_{h=0}^{g-i} k-h y_{k+i, i+h} \quad (9.14)$$

This is achieved by adding $x_k y_{k+i, i}$ to $z_{k-1, i+1}$. The $\{z_{k-1, i+1}\}$, for $i = f+1, f+2, \dots, \phi$ are left unchanged. In the arrangement B of the detection process, the detector replaces $z_{k-1, f+1}$ by $z_{k, f}$ for each of the $4m$ expanded vectors, leaving all the other $\{z_{k-1, i+1}\}$ unchanged.

With the arrangement A of the detection process, the detector evaluates for each expanded vector the corresponding cost

$$d_k = |U_{k-1}|^2 + \sum_{i=0}^f |r_{k+i} - z_{k,i}|^2 \quad (9.15)$$

which gives the same value for d_k as equation (9.12). With the arrangement B of the detection process, the detector evaluates for each expanded vector the quantities

$$c_i = |r_{k+i} - z_{k-1, i+1}|^2 \quad (9.16)$$

for $i = 0, 1, \dots, f-1$, and

$$c_f = |r_{k+f} - z_{k, f}|^2 \quad (9.17)$$

and then determines the corresponding cost as

$$d_k = |U_{k-1}|^2 + \sum_{i=0}^f c_i \quad (9.18)$$

With either arrangement of the detection process, m vectors $\{Q_k\}$ are selected from the $4m$ expanded vectors, according to one of the three selection processes (systems 1-3) previously described and using the appropriate set of costs $\{d_k\}$. The selected vectors are stored together with the associated costs. For each of these stored vectors, the detector then replaces every remaining $z_{k-1, i+1}$ (equation (9.13)) by the corresponding $z_{k,i}$ (equation (9.14)) and evaluates the quantity

$$|U_k|^2 = |U_{k-1}|^2 + |r_k - z_{k,0}|^2 \quad (9.19)$$

ready for the next detection process. The detection of the appropriate data symbol is carried out as described for the particular one of the three systems 1-3 in use. It can be seen here that arrangement B is not really suitable for use with system 3 since the cost evaluated by arrangement B for an expanded vector is not the full cost as given by equation (9.12) or (9.15), leading to an inferior detection process in system 3.

At this stage the receiver holds in store m vectors $\{Q_k\}$, and associated with each of these, it holds the costs d_k , the ϕ quantities $z_{k,1}, z_{k,2}, \dots, z_{k,\phi}$, and $|U_k|^2$. It is now ready to receive r_{k+f+1} , when the whole of the procedure just described is repeated and so on.

In the four expanded $(n+1)$ - component vectors $\{p_k\}$ derived from any one Q_{k-1} , x_k has its four possible values, so that when the arrangement B of the detection process is used, the corresponding four $\{z_{k,f}\}$ have four different values, but any $z_{k-1, i+1}$ has the same value in the four expanded vectors since $z_{k-1, i+1}$ is independent of x_k . It is clear therefore that c_i (equation (9.16)) has the same value in all four expanded vectors, for any given i in the range 0 to $f-1$ so that in equation (9.18)

$$\sum_{i=0}^{f-1} c_i = e_k \quad (9.20)$$

has the same value for all four expanded vectors that are derived from any one Q_{k-1} . Thus

$$d_k = |U_{k-1}|^2 + e_k + |r_{k+f} - z_{k,f}|^2 \quad (9.21)$$

and only the component $|r_{k+f} - z_{k,f}|^2$ has different values in the four expanded vectors. Evidently, when the arrangement B of the detection process is used in place of the arrangement A, a useful simplification is achieved.

The detection process can be further simplified by using the "magnitude-sum" distance in place of the "unitary distance in the evaluation of the cost associated with any stored vector. the magnitude-sum $\|F\|$ of the vector or scalar quantity F is the sum of the magnitude of all real and imaginary components of F . With the arrangement B of the detection process, equation (9.18) is now replaced by

$$d_k = \|U_{k-1}\| + \sum_{i=0}^f b_i \quad (9.22)$$

where

$$b_i = \|r_{k+i} - z_{k-1, i+1}\| \quad (9.23)$$

for $i = 0, 1, \dots, f-1$, and

$$b_f = \|r_{k+f} - z_{k,f}\| \quad (9.24)$$

Also

$$\|U_k\| = \|U_{k-1}\| + \|r_k - z_{k,0}\| \quad (9.25)$$

The advantage of this distance measure is that it avoids all the squaring operations that are involved in computing a unitary distance, thus significantly simplifying the system.

5 Double Sampling

With neither the ideal nor actual equipment filters is the received signal $r(t)$ being sampled at the Nyquist rate, so that the various detection processes described so far do not use all the available information in $r(t)$ and hence do not achieve the best available tolerance to noise. Ideally, the receiver input filter should be a "noise-whitened matched filter" but this requires to be held adaptively adjusted for the channel which is the very thing that the techniques studied here are designed to avoid. The bandwidth of both the transmitter and receiver filters could, of course, be reduced appropriately, but the transmitted signal now uses only a part of the available bandwidth of the transmission path and there is also a significant increase in the number of components of y_i . An alternative and preferable approach is to sample the received signal at twice the signal element rate, which brings the sampling rate close to the Nyquist rate. The samples of $r(t)$ now contain effectively all the information in $r(t)$, enabling an improved performance to be achieved by the detector.

When sampling twice per received signal-element, the sample of $r(t)$ at time $t=iT$ (T having its previous value and being therefore the reciprocal of the signal element rate) is again given by r_i in equation (9.2), whereas the sample of $r(t)$ at time $t=(i-1/2)T$ is

$$r'_i = \sum_{h=0}^g y'_{i,h} + w'_i \quad (9.26)$$

where $y'_i = y'_{i,0} y'_{i,1} \cdots y'_{i,g}$ (9.27)

and $y'_{i,h} = 0$ for $i < 0$ and $i > g$. The phase of the sampling instants is here assumed to be such that $y'_{i,0}$ is the first potentially nonzero component of the $\{y'_{i,h}\}$ and $\{y_{i,h}\}$, when these are arranged to form the sequence $\cdots y'_{i,0} y_{i,0} y'_{i,1} y_{i,1} \cdots y'_{i,g} y_{i,g} \cdots$

To be consistent with the previous definition of the sampled impulse-response of the channel and with equations (9.3) and (9.26), the sampled impulse-response of the channel is now taken to have the two different values y_i and y'_i , depending upon whether the sampling instant is iT or $(i-1/2)T$, respectively. The

real and imaginary parts of the complex-valued noise components $\{w'_i\}$ are Gaussian random variables with zero mean and variance σ^2 . With the ideal equipment filters the real and imaginary parts of the $\{w'_i\}$ are uncorrelated whereas with the actual filters they are slightly correlated. There is however, appreciable correlation between w_i and w'_i , the sampling interval here being $\frac{1}{2}T$. Any correlation between the noise samples tends, at least to some degree, to reduce the tolerance to noise of any of the detection processes described here, so that the receiver filters should be designed to minimize this correlation without either increasing the resultant effective bandwidth of the filters or else significantly increasing the duration of a received signal-element in $r(t)$.

Following any one detection process, two successive received samples r'_i and r_i are fed to the detector before the commencement of the next detection process. The operation of the detector is now exactly as described in Sections 3 and 4 for any combination of one of the systems 1-3 and arrangement A or B, except that two new received samples are processed instead of just one. Associated with the quantities $z_{k-1,i+1}$, $z_{k,i}$, c_i and c_f (in equations (9.13), (9.14), (9.16), and (9.17) respectively) all of which are determined as previously described, are the corresponding quantities

$$z'_{k-1,i+1} = \sum_{h=1}^{g-i} k-h y'_{k+i, i+h} \quad (9.28)$$

$$z'_{k,i} = \sum_{h=0}^{g-i} k-h y'_{k+i, i+h} \quad (9.29)$$

$$c_i = |r'_{k+i} - x'_{k-1, i+1}|^2 \quad (9.30)$$

for $i = 0, 1, \dots, f'-1$, where $y'_{k+\phi, f'}$ is the first significant component of $y'_{k+\phi}$, determined according to equation (9.7), and

$$c'_f = |r'_{k+f'} - z'_{k,f'}|^2 \quad (9.31)$$

so the equation (9.15) becomes

$$d_k = |U_{k-1}|^2 + \sum_{i=0}^f |r_{k+i} - z_{k,i}|^2 + \sum_{i=0}^{f'} |r'_{k+i} - z'_{k,i}|^2 \quad (9.32)$$

equation (9.18) becomes

$$d_k = |U_{k-1}|^2 + \sum_{i=0}^f c_i + \sum_{i=0}^{f'} c'_i \quad (9.33)$$

and equation (9.19) becomes

$$|U_k|^2 = |U_{k-1}|^2 + |r_k - z_{k,0}|^2 + |r'_k - z'_{k,0}|^2 \quad (9.34)$$

When the magnitude-sum distance is used in place of the unitary distance, the changes correspond to those previously described (equations (9.22) through (9.25)). It is evident that the detection process with double sampling is approximately twice as complex as that when sampling only once per received signal-element.

6 Comparison of Systems

Extensive computer-simulation tests have been carried out to compare the performances of the different detection processes described here. The arrangements tested and assumptions made are as described in Sections 2-5, and the results of the tests are shown in Figure 9-3–Figure 9-10. The signal/noise ratio is here taken to be ψ dB, where

$$\psi = 10 \log_{10} (1/\sigma^2) \quad (9.35)$$

the mean-square value of the data-symbol s_i , per bit of information carried, being unity, and the mean-square value of both the real and imaginary parts of the noise-component w_i (or w'_i) being σ^2 . In every case the appropriate correlation has been introduced between the different noise components. Every individual measurement used in plotting a graph in Figure 9-3–Figure 9-10 has involved the transmission of 26,400 data-symbols $\{s_i\}$ over a model of the appropriate channel. Since the measurement is to some degree affected, not only by the particular random number sequence used for the additive Gaussian noise signal but also, rather more seriously, by the random number sequence involved in generating the Rayleigh fading of each sky wave, care has been taken to use the same random number sequence (from which are generated the two independently fading sky waves) for each of the different systems tested and at all signal/noise ratios, together with different additive Gaussian noise signals at the different signal/noise ratios, the same noise signal being used for all systems at any one signal/noise ratio. Although this reduces the statistical errors in the relative performances of the different systems (probably to no more than around $\pm 1/2$ dB), some caution must still be used in interpreting the results.

Figure 9-3–Figure 9-5 compare the performance of system 1, using arrangement A and $m = 16$ or 32 , with that of a conventional Viterbi-algorithm detector for ideal equipment filters and for both single and double sampling. Figure 9-6–Figure 9-8 compare the performances of systems 1-3, again using arrangement A and $m = 16$ or 32 , but now for actual equipment filters and double sampling, and including also system 1 with arrangement B. All curves in Figure 9-3–Figure 9-8 apply for both $m = 16$ and $m = 32$. The

influence on the error rate of the parameter value a is shown in Figure 9-9, and the corresponding influence of the phase of the sampling waveform at the receiver (determining the time instants $\{iT\}$ and $\{(i-\frac{1}{2})T\}$) are shown in Figure 9-10. The two graphs shown here for channel 2 give the upper and lower limits of the tolerance to noise for 12 different phases of the timing waveform spaced at intervals of $1/12$ of the sampling interval $\frac{1}{2}T$. A very similar variation in performance is obtained over both channels 1 and 3. The unitary distance is used throughout Figure 9-3–Figure 9-10 and in every case $n = 16$. The tests in Figure 9-3–Figure 9-8 have been repeated with the magnitude-sum distance and it has been found that the degradation in tolerance to noise, resulting from the use of the latter distance measure, always lies in the range 0 - $\frac{1}{2}$ dB. The corresponding graphs are therefore not shown.

It can be seen from Figure 9-3–Figure 9-5 that, with the ideal filters and with either single or double sampling, when the Viterbi algorithm is replaced by system 1 (using arrangement A and $m = 16$ or 32), the degradation in tolerance to noise lies in the range 0-1 dB. However, with either detection process, an improvement of some 2 dB in tolerance to noise is achieved when single sampling is replaced by double sampling. When the value of m in system 1 is reduced from 16 to 8 (the latter curves not being shown in Figure 9-3–Figure 9-5), there is a reduction of some 0.2 dB in tolerance to noise. From Figure 9-6–Figure 9-8 there does not appear to be any very great difference between the performances of the system 1-3 so that the precise method used for selecting the stored vectors $\{Q_i\}$ does not seem to be critical, so long as a sufficient number of vectors is used. It is also evident from Figure 9-6–Figure 9-8 that the reduction in tolerance to noise, when arrangement A is replaced by arrangement B, is typically under $\frac{1}{2}$ dB, so that the former would only be used where the very best tolerance to noise is required. When the value of m is reduced from 16 to 8 (the latter curves not being shown in Figure 9-6–Figure 9-8) there is a reduction of some 0.1 dB in tolerance to noise. It should not therefore be necessary to use many more than 8 stored vectors, giving a detection process that can be implemented without undue complexity. Indeed, when now using arrangement B together with the magnitude-sum distance measure, the detector

appears likely to become relatively simple to implement. These conclusions, of course, require verification through the detailed design of the detector, which has yet to be done.

Figure 9-9 shows that, when single sampling is used, the optimum value of "a" (in equation 9.7) is 0.7, but there is no significant change in the error rate (for either arrangement A or B) as "a" varies from 0.5 to 0.9, so that the value of "a" does not appear to be critical. It is evident from Figure 9-10 that the phase of the timing waveform at the receiver does not seriously affect the tolerance of the system to noise.

7 Conclusions

When the sampled impulse-response of the channel is known and double sampling is used at the receiver, the more effective arrangements of the detection processes studied here give a tolerance to noise over the channels tested, which appears to be within about 1 dB of that of a Viterbi-algorithm detector. This suggests that in the given application, the adaptive linear filter, normally proposed for use ahead of a near-maximum-likelihood detector, may be discarded with no serious loss in tolerance to noise, thus avoiding the problems involved with the adaptive filtering of a time-varying channel. The techniques are clearly worthy of further investigation.

Table 9-1 Channels Used in the Tests

Channel	Frequency spread introduced into the data signal Hz	Relative transmission delay for the two sky waves
1	$\frac{1}{2}$	1×10^{-3}
2	1	2×10^{-3}
3	2	3×10^{-3}

Diagrams

Figure 9-1 "Model of the Data-Transmission Systems"

Figure 9-2 "Rayleigh Fading Introduced By a Sky Wave"

Figure 9-3 "Performance With Ideal Equipment Filters Over Channel 1"

Figure 9-4 "Performance With Ideal Equipment Filters Over Channel 2"

Figure 9-5 "Performance With Ideal Equipment Filters Over Channel 3"

Figure 9-6 "Performance With Actual Equipment Filters Over Channel 1"

Figure 9-7 "Performance With Actual Equipment Filters Over Channel 2"

Figure 9-8 "Performance With Actual Equipment Filters Over Channel 3"

Figure 9-9 "Influence of the parameter value a on the error rate for system 1 with ideal equipment filters and single sampling when operating over channel 3"

Figure 9-10 "Maximum range of the influence of the phase of the sampling waveform on the error rate for system 1 with actual equipment filters and double sampling when operating over channel 2"

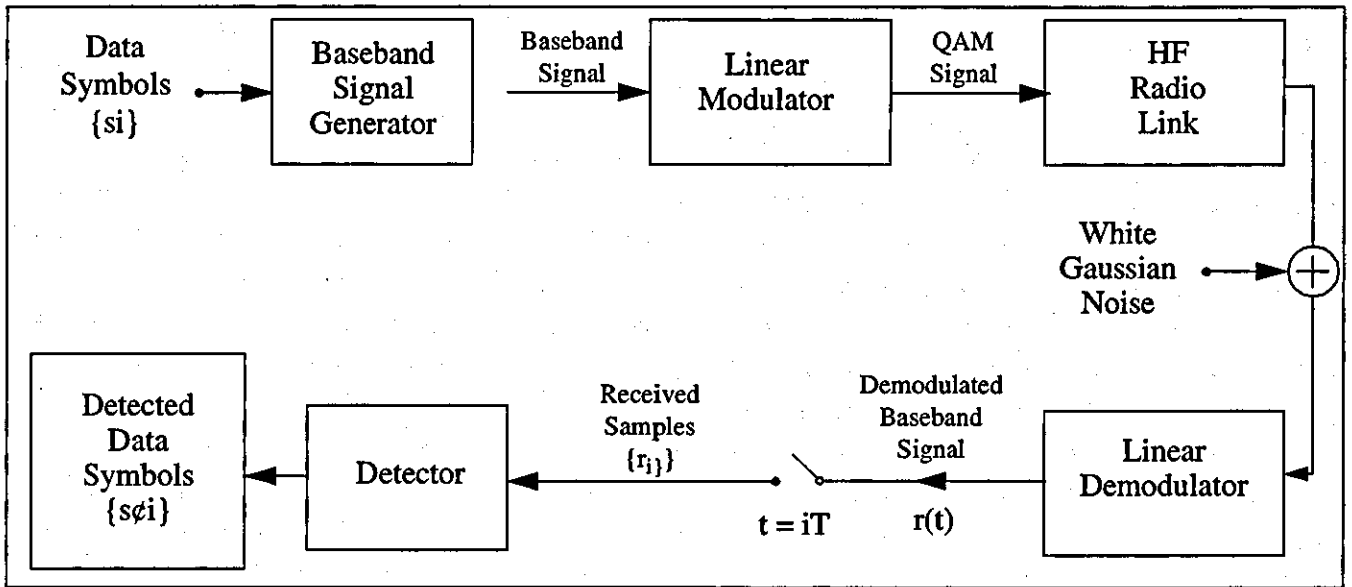


Figure 9-1 Model of the Data-Transmission Systems

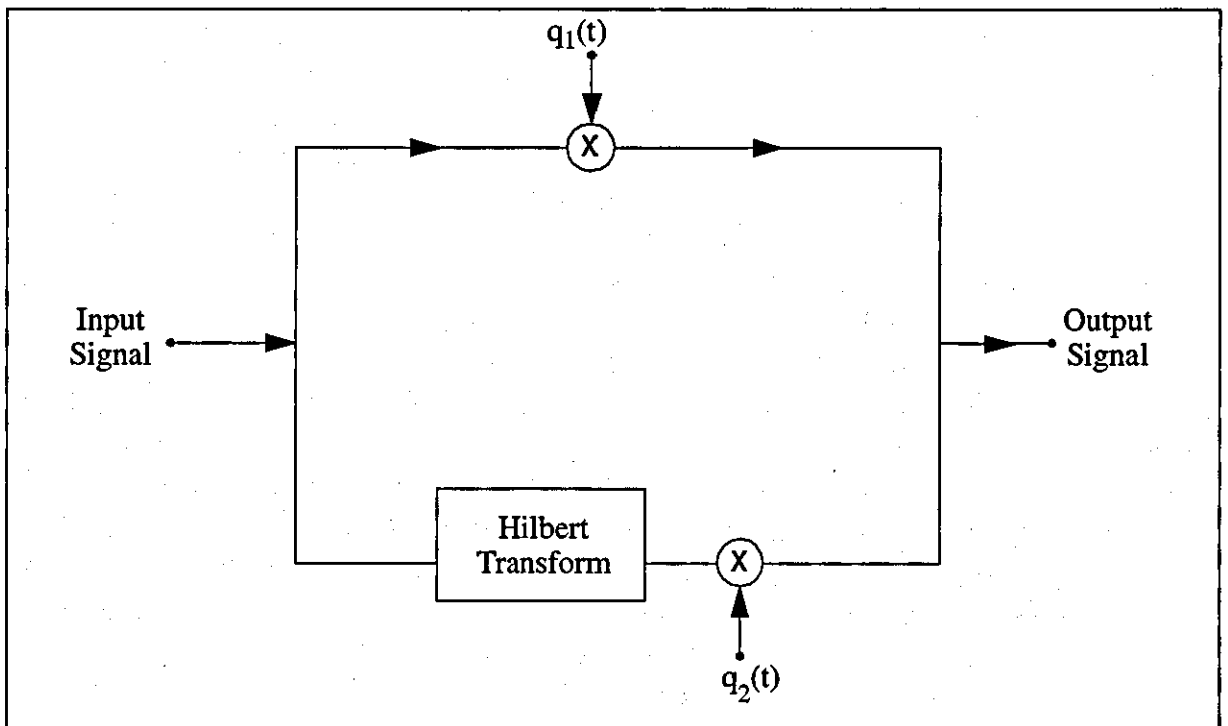


Figure 9-2 Rayleigh Fading Introduced By a Sky Wave

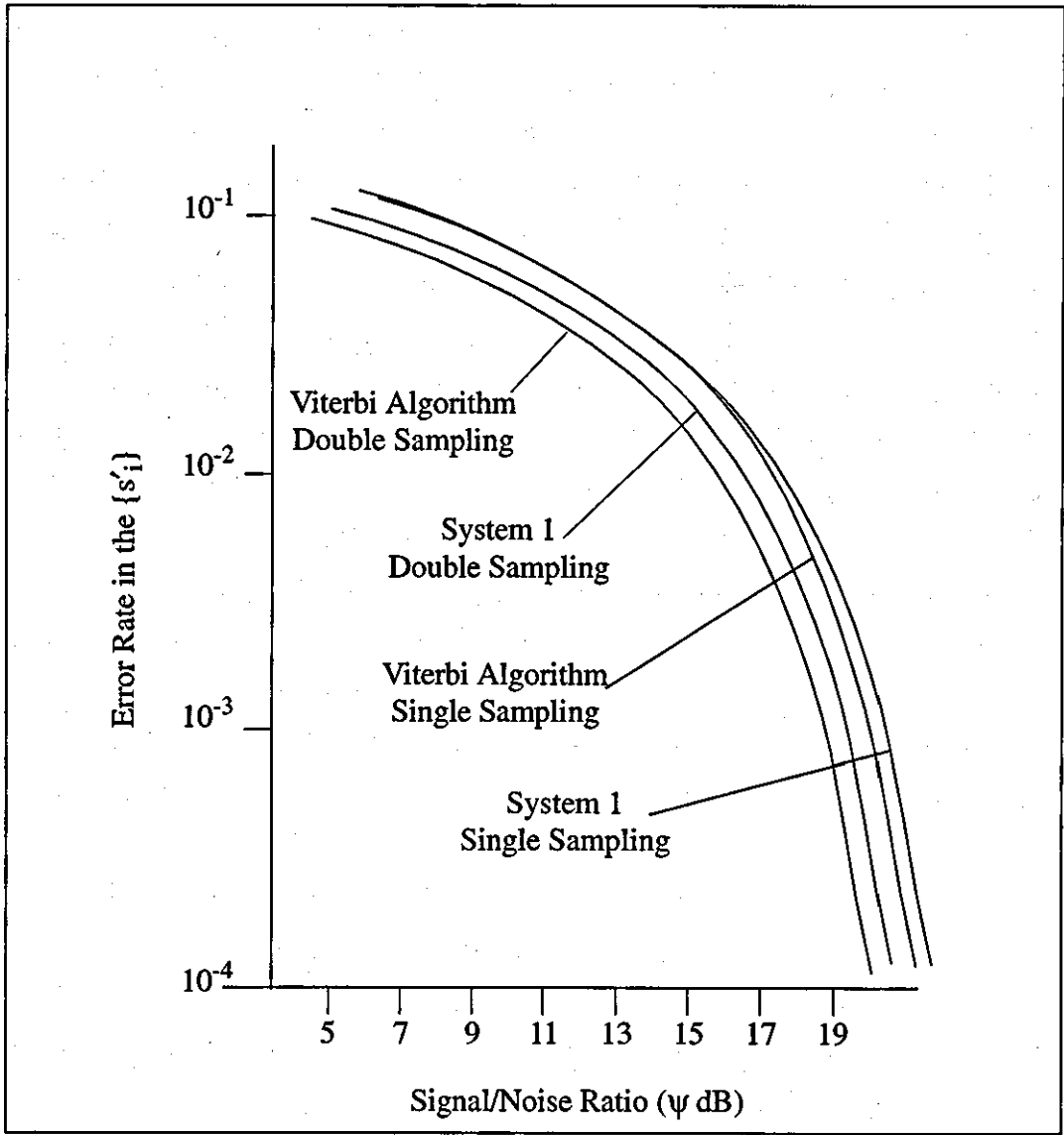


Figure 9-3 Performance With Ideal Equipment Filters Over Channel 1

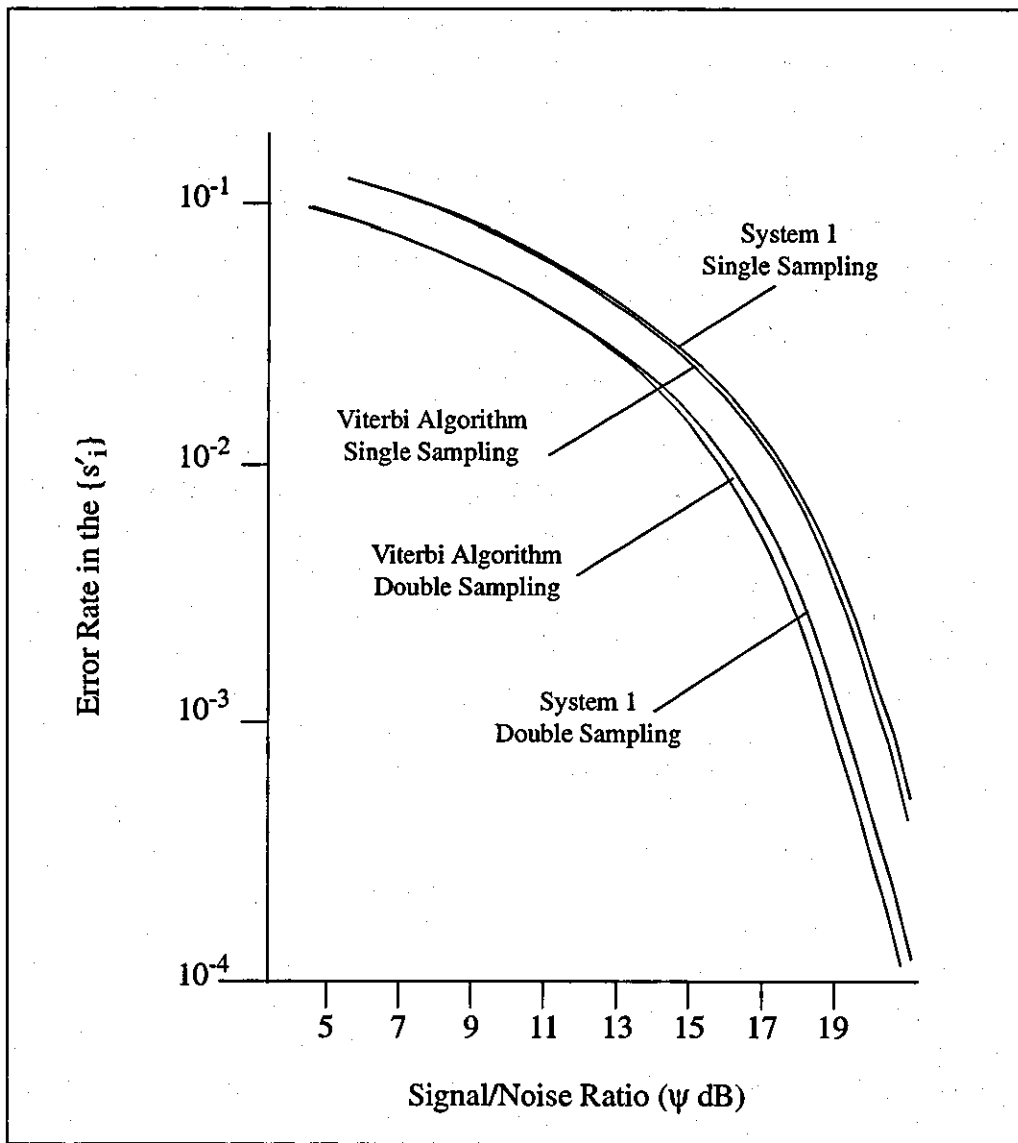


Figure 9-4 Performance With Ideal Equipment Filters Over Channel 2

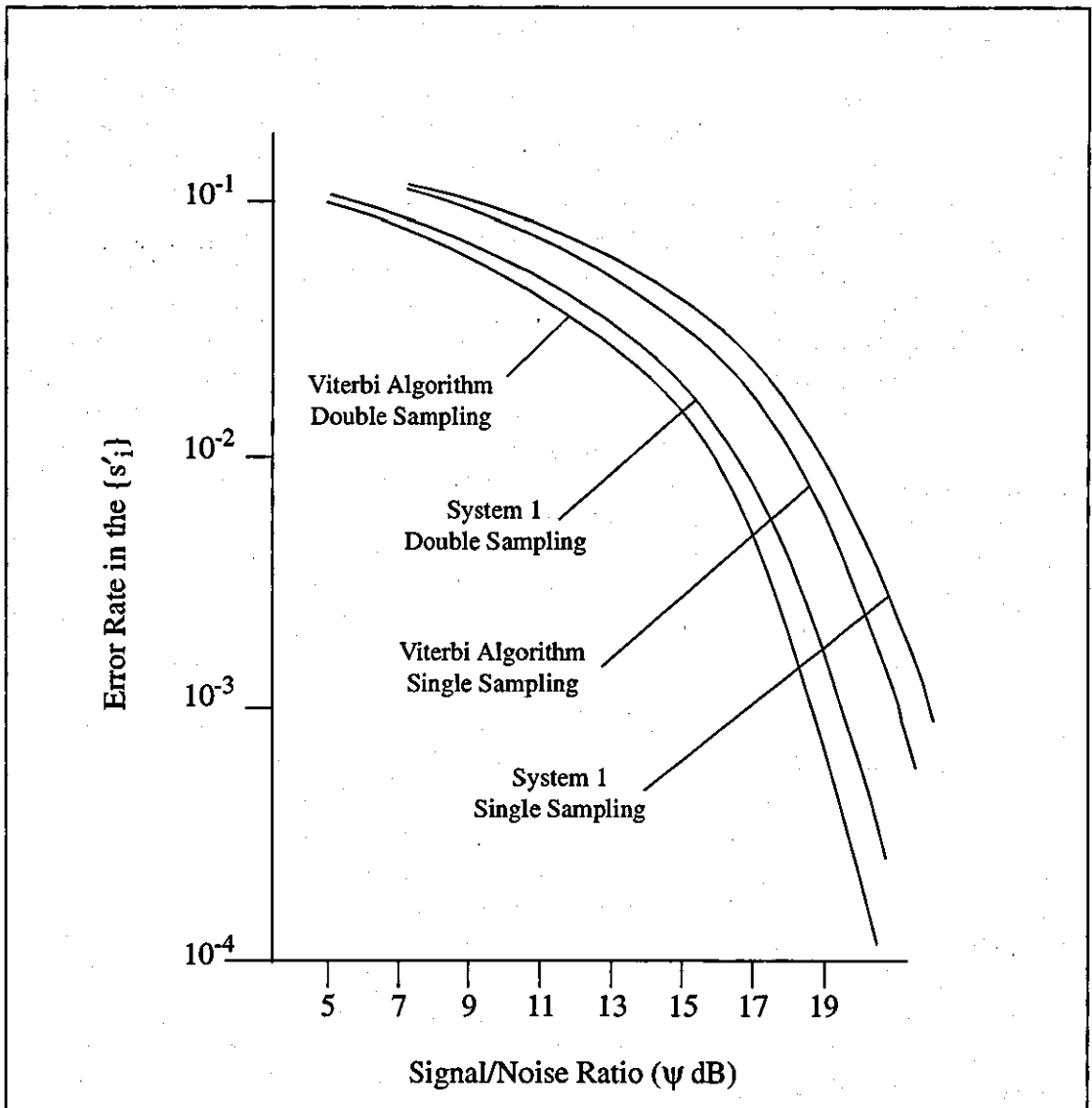


Figure 9-5 Performance With Ideal Equipment Filters Over Channel 3

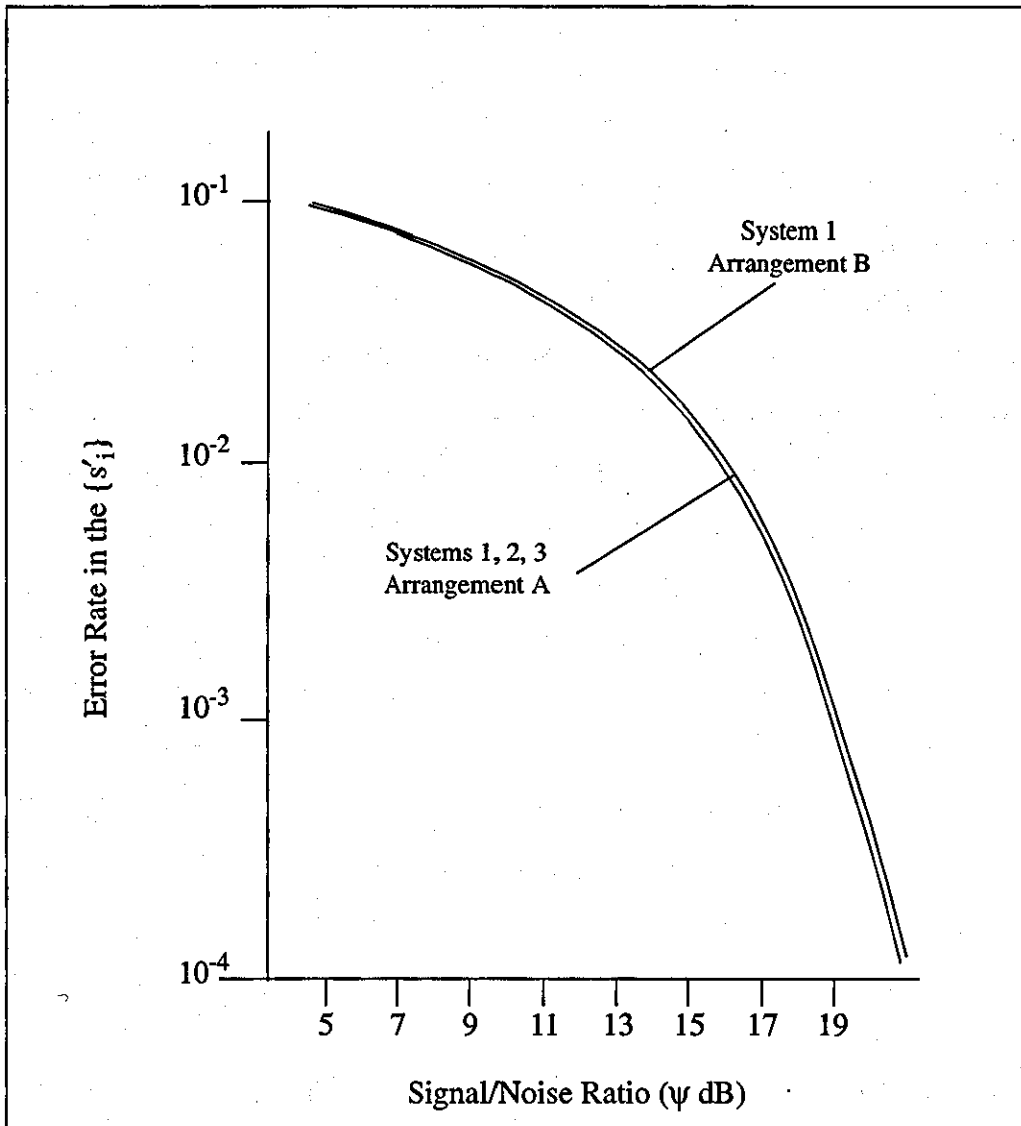


Figure 9-6 Performance With Actual Equipment Filters Over Channel 1

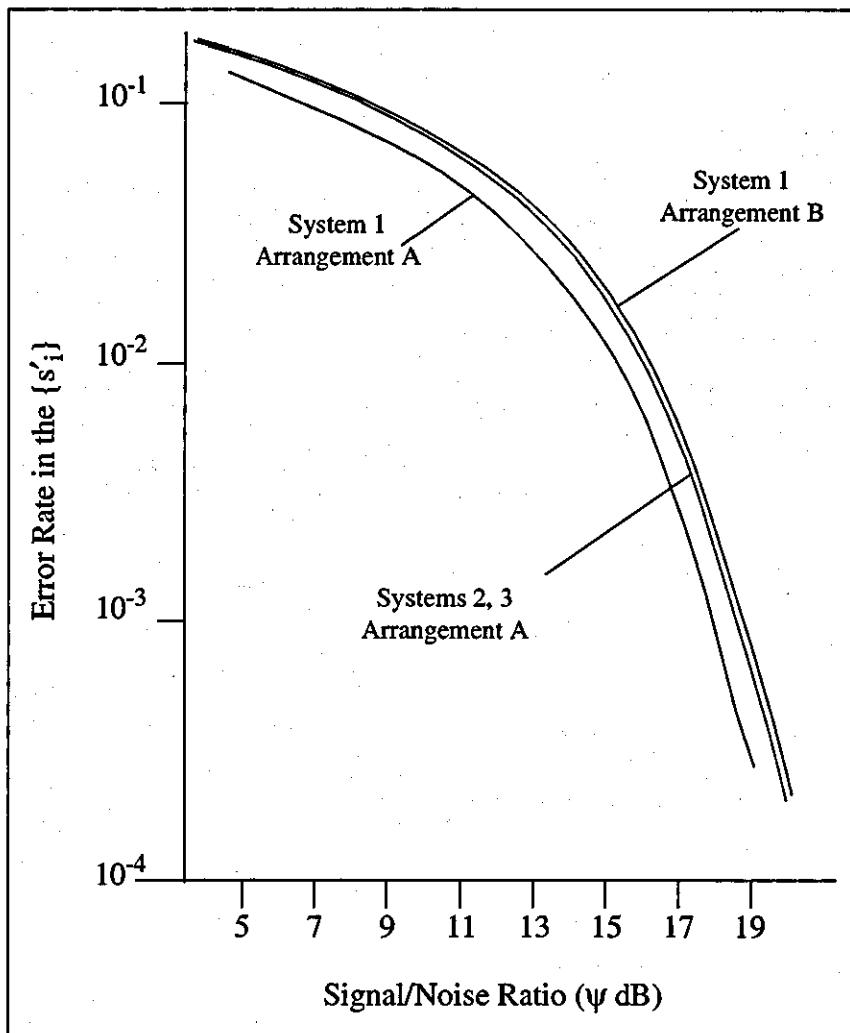


Figure 9-7 Performance With Actual Equipment Filters Over Channel 2

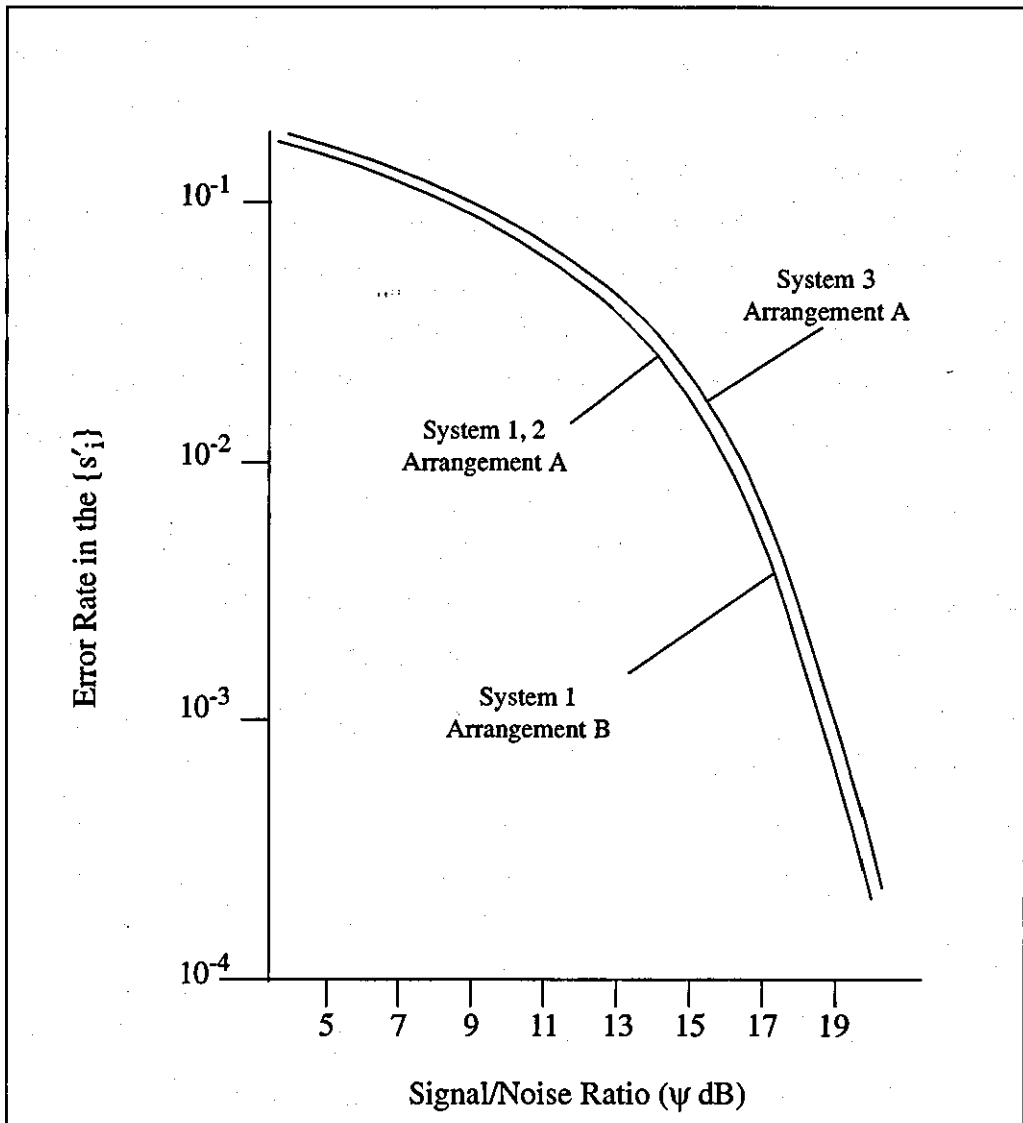


Figure 9-8 Performance With Actual Equipment Filters Over Channel 3

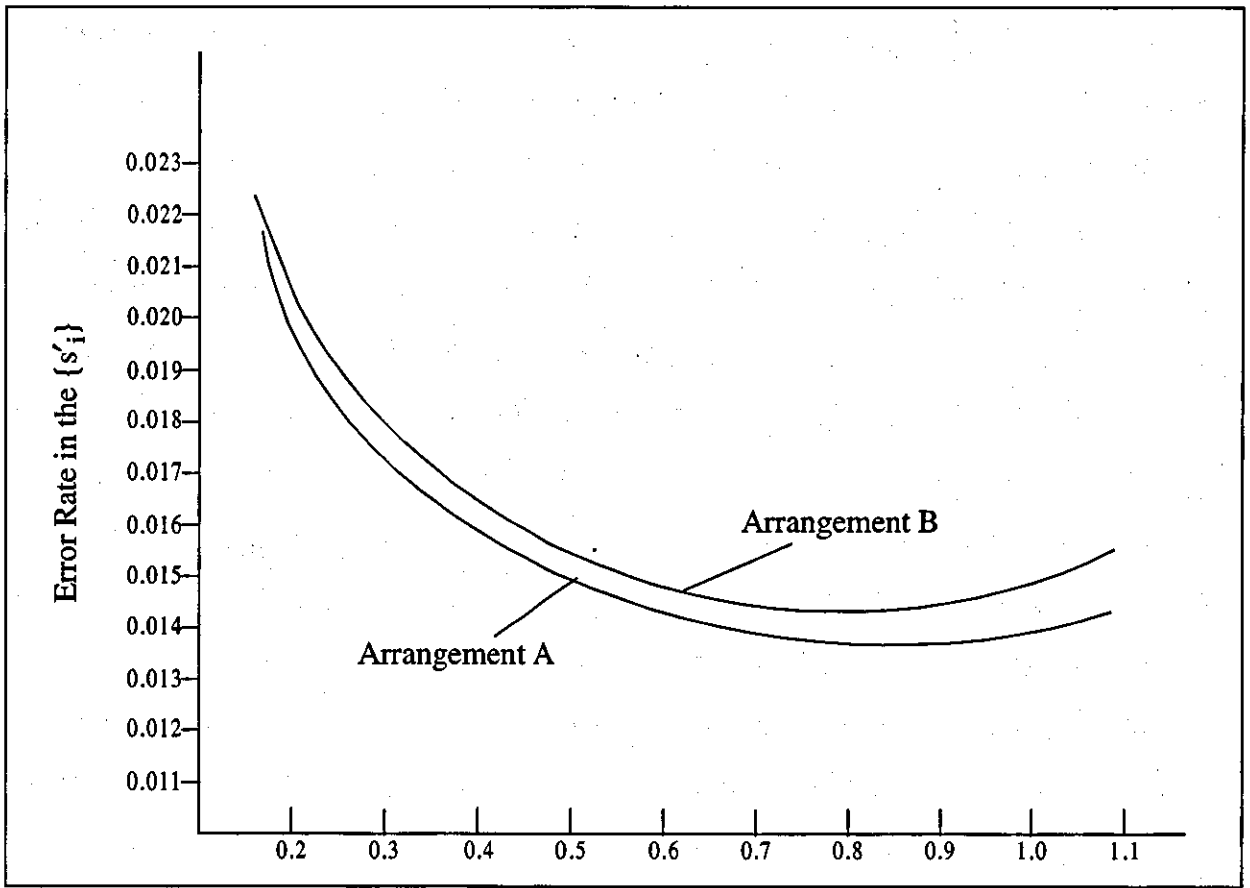


Figure 9-9 Influence of the parameter value a on the error rate for system 1 with ideal equipment filters and single sampling when operating over channel 3

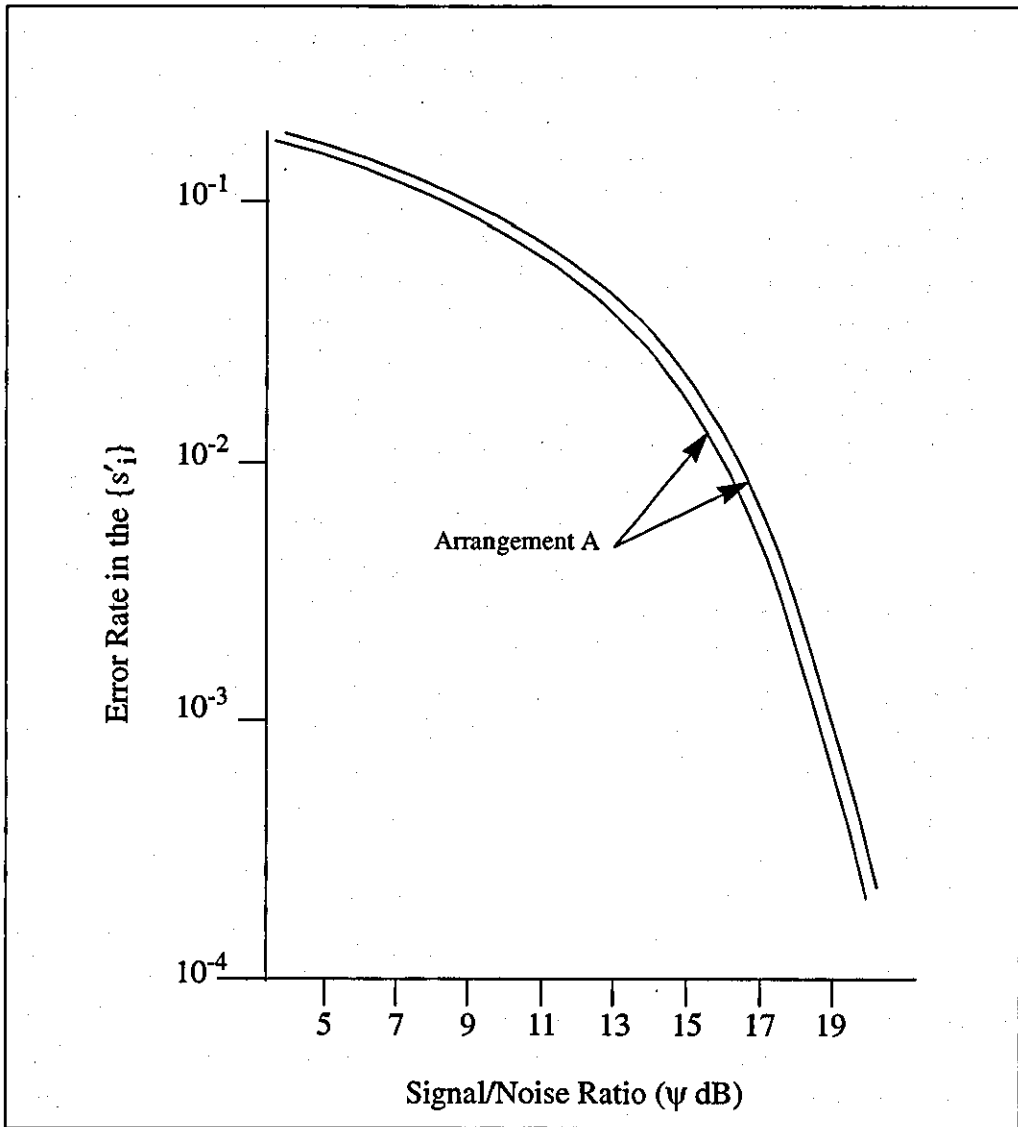


Figure 9-10 Maximum range of the influence of the phase of the sampling waveform on the error rate for system 1 with actual equipment filters and double sampling when operating over channel 2

References

1. CLARK, A. P., *Advanced data-transmission systems* (Pentech Press, 1977)
2. FORNEY, G. D., *Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference*, IEEE Trans., 1972, IT-18, pp. 363-378
3. FORNEY, G. D., *The Viterbi algorithm*, Proc. IEEE, 1973, 61, pp. 268-278
4. PULLEYBLANK, R. W., *A comparison of receivers designed on the basis of minimum mean-square error and probability of error for channels with intersymbol interference and noise*, IEEE Trans., 1973, COM-21, pp. 1434-1438
5. MESSERSCHMITT, D.G., *A geometric theory of intersymbol interference. Part II: performance of the maximum likelihood detector*, Bell Syst. Tech. J., 1973, 52, pp. 1521-1539
6. MAGEE, F. R., *A comparison of compromise Viterbi algorithm and standard equalization techniques over band-limited channels*, IEEE Trans., 1975, COM-23, pp. 361-367
7. FALCONER, D. D. and MAGEE, F. R., *Evaluation of decision feedback equalization and Viterbi algorithm detection for voiceband data transmission. Part I*, IEEE Trans., 1976, COM-24, pp. 1130-1139
8. FALCONER, D. D. and MAGEE, F. R., *Evaluation of decision feedback equalization and Viterbi algorithm detection for voiceband data transmission. Part II*, IEEE Trans., 1976, COM-24, pp. 1238-1245
9. QUERSHI, S. U. H and NEWHALL, E. E., *An adaptive receiver for data transmission over time dispersive channels*, IEEE Trans., 1973, IT-19, pp. 448-457
10. FALCONER, D. D. and MAGEE, F. R., *Adaptive channel memory truncation for maximum-likelihood sequence estimation*, Bell Syst. Tech. J., 1973, 52, pp. 1541-1562
11. CANTONI, A. and KWONG, K., *Further results on the Viterbi algorithm equalizer*, IEEE Trans., 1974, IT-20, pp. 764-767
12. FREDRICSSON, S. A., *Joint optimization of transmitter and receiver filters in digital PAM systems with a Viterbi detector*, IEEE Trans., 1976, IT-22, pp. 200-210
13. CANTONI, A. and BUTLER, P., *Linear minimum mean-square error estimators applied to channel equalization*, IEEE Trans., 1977, COM-25, pp. 441-446
14. DESBLACHE, A. E., *Optimal short desired impulse response for maximum-likelihood sequence estimation*, IEEE Trans., 1977, COM-25, pp. 735-738

15. BEARE, C. T., *The choice of the desired impulse response in combined linear-Viterbi algorithm equalizers*, IEEE Trans., 1978, COM-26, pp. 1301-1307
16. DOAN, H. B. and CANTONI, A., *Fast adaptation of equalizer and desired response for digital data receivers*, Electronic Circuits and Systems, 1978, 2, pp. 159-166
17. FOSCHINI, G. J., *A reduced state variant of maximum likelihood sequence detection attaining optimum performance for high signal-to-noise ratios*, IEEE Trans., 1977, IT-23, pp. 605-609
18. CLARK, A. P., HARVEY, J. D. and DRISCOLL, J. P., *Near-maximum likelihood detection processes for distorted digital signals*, Radio and Electronic Engineer, 1978, 48, pp. 301-309
19. CLARK, A. P., KWONG, C. P. and HARVEY, J. D., *Detection processes for severely distorted digital signals*, Electronic Circuits and Systems, 1979, 3, pp. 27-37
20. MAGEE, F. R. and PROAKIS, J. G., *Adaptive maximum likelihood sequence estimation for digital signalling in the presence of intersymbol interference*, IEEE Trans., 1973, IT-19, pp. 120-124
21. CLARK, A. P., KWONG, C. P. and McVERRY, F., *Estimation of the sampled impulse-response of a channel*, Signal Processing, 1980, 2, pp. 39-53
22. CLARK, A. P. and McVERRY, F., *Channel estimation for an HF radio link*, Communications, Radar and Signal Processing, 1981, 128, Part F.

Chapter 10

Digital Tone Detection and Generation

Abstract

This chapter is based on the invention patented under American patent 4,839,842 (S. Asghar, H. Pyi, June 1989).

Fast methods of tone detection and tone generation disclosed are particularly suitable for implementation in a digital signal processor. Chebyshev polynomials are employed to generate periodic waveforms and to detect such waveforms. In an alternative aspect of the invention, trigonometric formulae are employed to generate periodic waveforms which are representable as the sum of sine and cosine functions. Unlike analog techniques, the digital techniques do not entail long delays in the generation or detection of tones because a fast iterative recalculation is employed at each step. Accumulated error is avoided by restarting the procedure once a predetermined value is exceeded. Independent choice of a quality factor and a confidence level is provided by the digital tone detection technique.

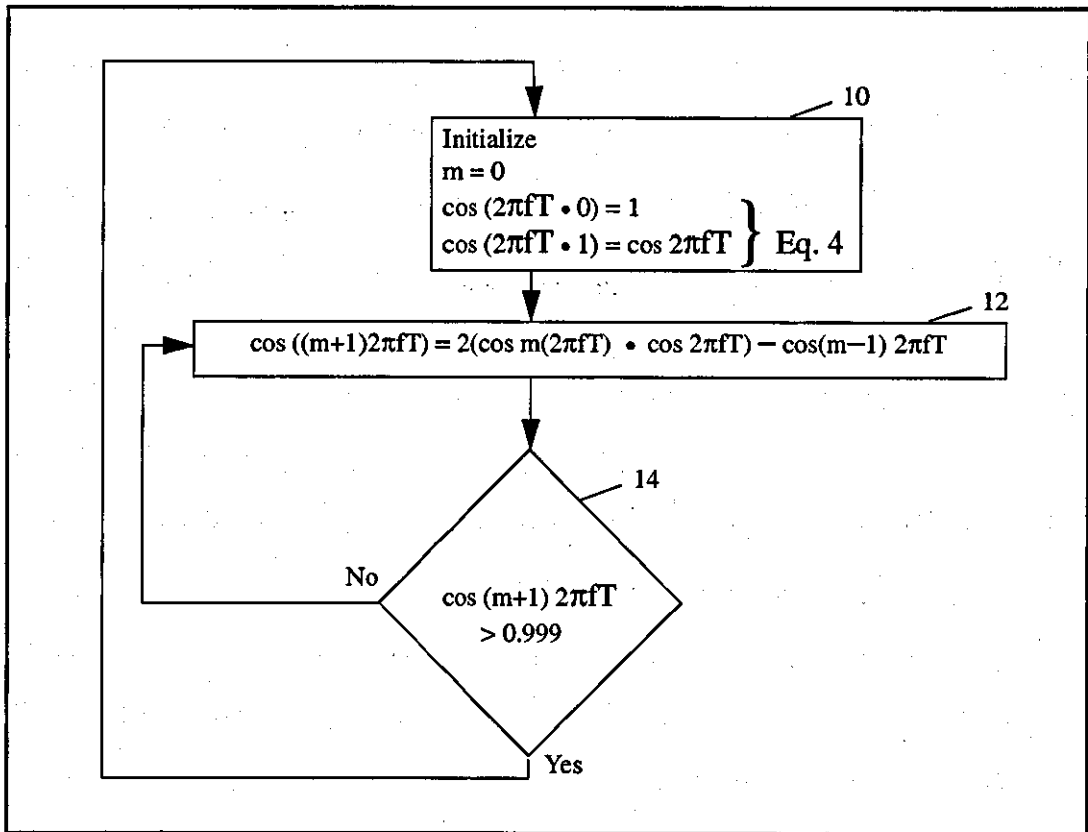


Figure 10-1

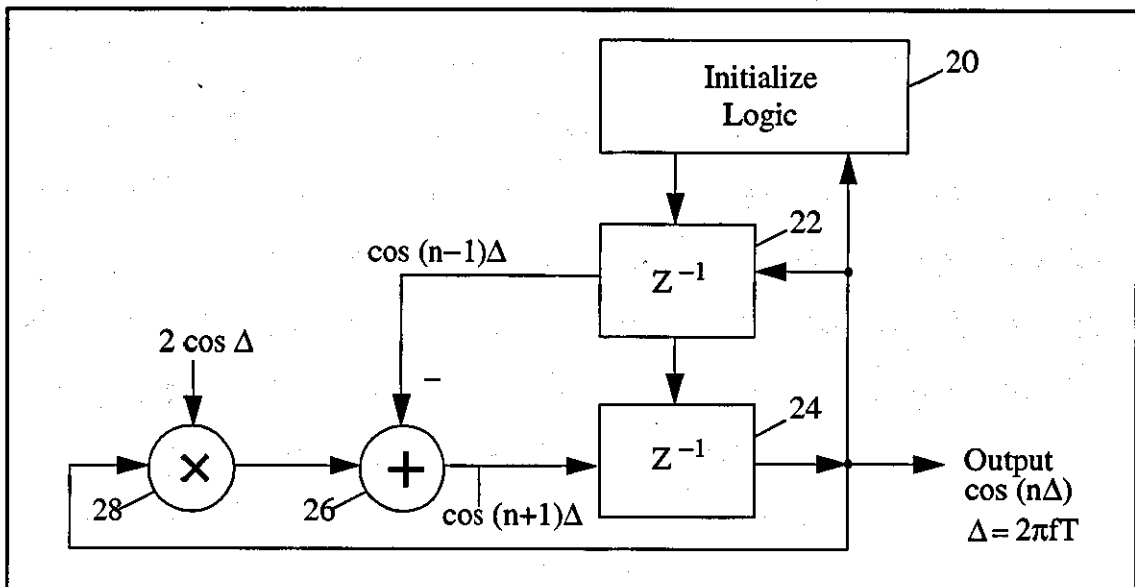


Figure 10-2

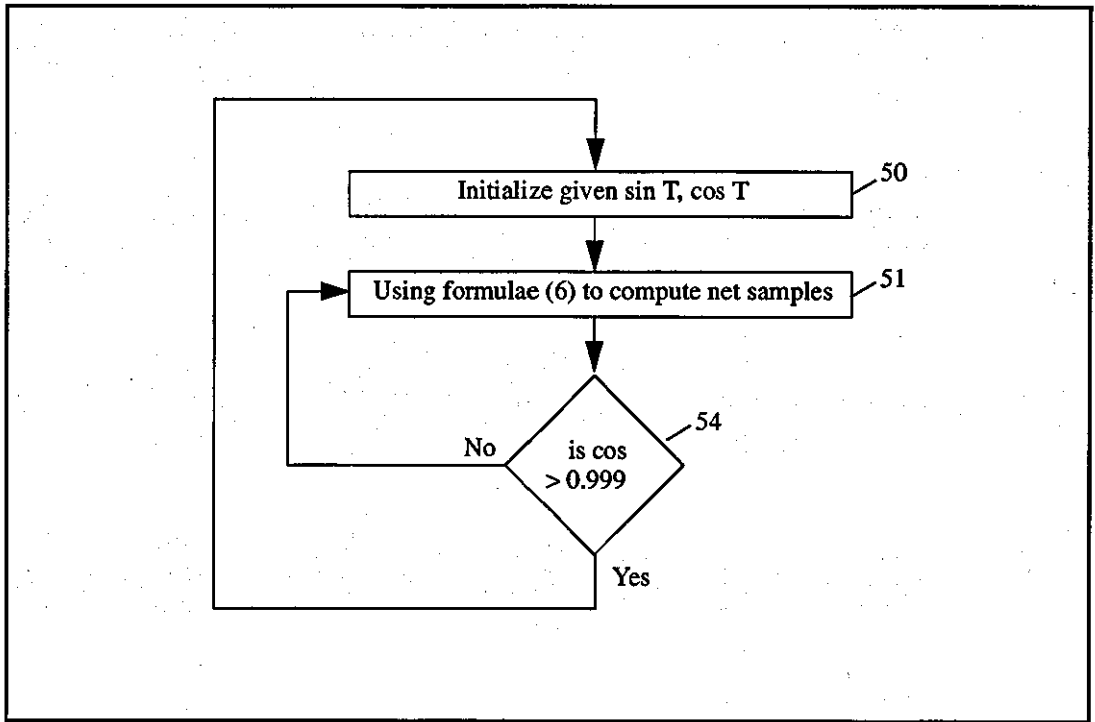


Figure 10-3

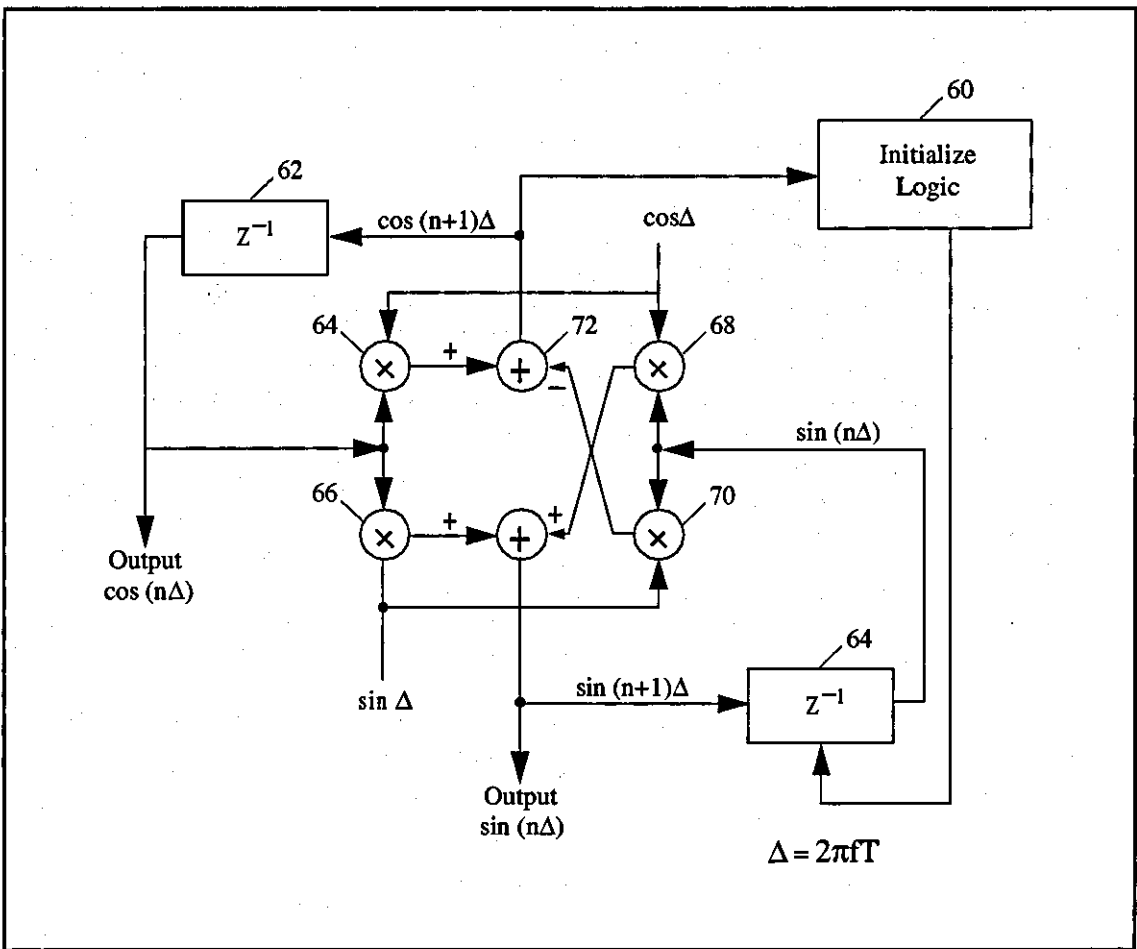


Figure 10-4

DESCRIPTION

The present invention relates to digital detection and generation of periodic signals and, more particularly, to a method using Chebyshev polynomials and trigonometric formulae to approximate periodic waveforms.

BACKGROUND OF THE INVENTION

Digital tone detection and generation is useful in telecommunications, such as in dual-tone multi-frequency (DTMF) used in "two tone" or "multi-tone" signaling. Known in the art are Sinerom, Taylor series and discrete fourier transform methods of generation. These methods are computationally complex and require a large storage memory. For tone detection, analog filtering of tones followed by analog energy detection is employed.

Conventional tone detection apparatus typically employs an analog tone filter which receives an input signal. The tone filter is typically a high-"Q" (quality-factor) providing extremely sharp rise and fall characteristics around the resonant frequency. Such analog circuitry is complex and costly to manufacture. More importantly, its response is sensitive to changes in temperature, voltage, component values and the like which result in variations in gain and band-pass characteristics.

The tone detection apparatus employs an energy detector following the tone filter responsive to the filtered signal which requires that a "confidence level" be selected. The energy detector produces a signal which is monitored by a decision device which determines whether the detected energy exceeds a "threshold" level and produces a binary-valued output therefrom indicating whether the received signal has the appropriate frequency at a satisfactory energy level.

The time required to detect a tone by such apparatus is determined by the Q factor and confidence level selected. This time can be substantial if a high level of quality and confidence is required.

SUMMARY OF THE INVENTION

The present invention provides a method for generation and detection of tone signals by digital means. In one embodiment of the invention, terms of a Chebyshev sequence are used recursively to digitally generate a sequence of normalized tone signals. Because roundoff and truncation errors will propagate and accumulate, errors will tend to grow if no corrective action is taken. The method of the instant invention provides that when a term in the sequence exceeds a predetermined value, the generation of tone signals is restarted.

The method of the instant invention is particularly suited to digital computation because it involves only one left shift, one multiplication and one subtraction to generate each term in the sequence of tone signals. Accordingly, little or no delay is experienced in the tone detection according to the invention.

In an alternative embodiment of the invention, terms of a trigonometric formulae are employed iteratively to generate both cosine and sine wave tone signals. As with the first embodiment, the method of the instant invention calls for restarting of the generation of tones whenever a term exceeds a predetermined value. The alternative method is equally well suited to digital computation.

In an application of the digital tone generation method of the instant invention, a method of digital tone detection is provided by the instant invention. The term-by-term difference between a sequence of measured tone signals and the signal predicted by the corresponding term of the Chebyshev sequence is determined. If the absolute value of this difference is less than a predetermined threshold value then it is concluded that the measured tone signals have the frequency of the Chebyshev sequence.

The digital tone detection method of the instant invention can be controlled to provide a high "Q" or a low "Q" filter. Independently, a selectable confidence level can be provided for the detection.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 10-1 is a flowchart describing the method of digital generation of tones according to the instant invention.

Figure 10-2 is a signal-flow diagram for the digital generation of tones according to the instant invention.

Figure 10-3 is a flowchart describing an alternative method of digital generation of tones according to the instant invention.

Figure 10-4 is a signal-flow diagram for the alternative digital generation of tones according to the instant invention.

DETAILED DESCRIPTION OF A PREFERRED EMBODIMENT

Any single tone signal can be appropriately scaled to be represented as a cosine or sine waveform using methods well-known to those of ordinary skill in the art. A cosine waveform having a frequency f can be sampled at periodic sampling intervals of T time units to produce the sequence $\{\cos(2\pi fTm) \ m=0, 1, 2, 3, \dots\}$.

Terms of the Chebyshev polynomial are given by the formula

$$C_m(x) = \cos(m \arccos(x)) \quad m=0, 1, 2, \dots \quad (10.1)$$

A general recursive relationship exists between the terms C_{m-1} , C_m and C_{m+1} :

$$C_{m+1}(x) = 2C_m(x)C_1(x) - C_{m-1}(x) \quad (10.2)$$

by expressing the terms C_{m-1} , C_m and C_{m+1} in equation (10.2) in terms of the cosine functions of equation (10.1), and using $T = \arccos(x)$ yields

$$\cos(m+1)2\pi fT = 2 \cos(m2\pi fT) \cdot \cos 2\pi fT - \cos 2\pi fT (m-1) \quad (10.3)$$

Equation (10.3) thus allows a sequence of normalized tone signals to be recursively generated for val-

ues of

$$m=1, 2, \dots, \text{ by starting with } \cos(0-2\pi fT)=1 \text{ and } \cos(1-2\pi fT)=\cos 2\pi fT. \quad (10.4)$$

The recursive generation of tone signals can thus be effected by repeated recalculation of equation (10.3). For a tone of frequency f and amplitude A , samples can be generated every T seconds resulting in the signal waveform

$$S=A \cos (m(2\pi fT)) \quad (10.5)$$

As will be appreciated by those skilled in the art, such recalculations are readily performed by a digital processor, in which the term $\cos (m(2\pi fT))$ can be iteratively calculated by equation (10.3). Roundoff and truncation errors will be propagated by such a procedure, however. The errors tend to accumulate and would produce a waveform of ever-increasing error if not corrected. By reinitializing the computation periodically, this accumulation problem can be avoided

A flowchart of the method of the instant invention is provided as Figure 10-1. A block 10 is initially entered which performs the initialization steps indicated by equation (10.4). A block 12 is next entered which performs the iterative calculation of the next sample term according to equation (10.3). Next a decision block 14 is entered which tests whether the term calculated in block 12 exceeds 0.999. If not, a "No" exit is taken from decision block 14 back to block 12 and the next sample term is calculated in block 12. If the test performed in block 14 indicates that the term exceeded 0.999, a "YES" exit is taken from block 14 which causes block 10 to be entered. Accordingly, any accumulated error is removed as the sequence of terms is reinitialized and restarted once again. In accordance with this method then, the calculations to generate one sample at block 12 requires one left shift, one multiplication and one subtraction.

As described hereinabove, the particular frequency and amplitude desired in the resulting waveform, S , is given by equation (10.5). The value used in block 12 of 0.999 is selected to be near the maximum value that the cosine function can take (1.0) and yet be readily reached within a few iterations of the loop

including the "No" exit.

Now with reference to Figure 10-2, a signal-flow diagram of an implementation of the method of the instant invention is provided. An initialization block 20 generates the first two terms of the sequence of a tone signal according to equation (10.4). These signals are received at a first input of a one time-unit delay block 22 and a first input of a one time-unit delay block 24. Delay block 22 generates a signal received by an adder node 26 at a subtractive input thereto. Adder 26 also receives at an additive input a signal generated by a multiplier node 28. Multiplier 28 receives at a first input the signal generated by second delay block 24 and at a second input the fixed signal $2 \cos T$. The signal generated by delay 24 is also conducted to a second input of delay 22 and the signal generated by adder 26 is conducted to a second input of delay 24. The signal generated by delay 24 is also conducted to the initialization block 20 and is the output signal corresponding to equation (10.3). Circuitry internal to block 20 implements the test performed in flowchart decision block 14 (Figure 10-1) and resets the signals generated so that the first two terms of the sequence of tone signals are conducted to delays 22 and 24.

Delay blocks 22 and 24 receive control signals (not shown) from initialization block 20 which cause the signals at their first inputs to be selected during the first time period after initialization and signals at their second inputs to be selected at all other time periods.

An alternative method of generation is required when both cosine and sine waveforms need be generated. Such is the case when a Quadrature Amplitude Modulation (QAM) scheme is used to transmit digital data over analog telecommunication lines.

In this case, the trigonometric identities

$$\cos(m+1)2\pi fT = \cos m2\pi fT \cdot \cos 2\pi fT - \sin m2\pi fT \cdot \sin 2\pi fT \quad (10.6)$$

$$\sin(m+1)T = \sin mT \cos T + \cos m2\pi fT \cdot \sin 2\pi fT$$

can be employed in an iterative manner to generate both cosine and sine wave samples of frequency f at intervals of T seconds.

With reference to Figure 10-3, a flowchart is provided employing the equations (10.6) to generate samples of periodic waveforms. A block 50 is initially executed which calls for initialization of the $\cos 2\pi fT$ and $\sin 2\pi fT$ terms. Such initialization values can be provided by numerical analysis methods known to those of ordinary skill in the art. Next a block 52 is entered which employs equations (10.6) to iteratively calculate the next sample values of cosine and sine. In order to avoid accumulation of truncation and roundoff error, a decision block 54 is next entered which tests whether $\cos 2\pi fT$ exceeds 0.999. This test results in either a return to the initialization block 50 or the recalculation block 52 as was described hereinabove in connection with the Chebyshev method of Figure 10-1.

With reference to Figure 10-4, a signal-flow diagram of an implementation of the method of the instant invention is provided. An initialization block 60 generates the first term of the sequence of tone signals. The signal is received at a first input of a one time-unit delay block 62 and a first input of a one time-unit delay block 64. Delay block 62 generates a signal received at a first input of a multiplier node 65 and a first input of a multiplier node 66. The signal generated by delay 64 is received at a first input of a multiplier node 68 and a first input of a multiplier node 70. A signal corresponding to $\cos T$ is received at a second input of multipliers 65 and 68. A signal corresponding to $\sin T$ is received at a second input of multipliers 66 and 70.

An adder node 72 receives the signal generated by multiplier 65 at an additive input and the signal generated by multiplier 70 at a subtractive input. An adder node 74 receives the signal generated by multiplier 66 at an additive input and the signal generated by multiplier 68 at a subtractive input. The signal generated by adder 72 is conducted to a second input of delay 62 and to the initialization block 60. The signal generated by adder 74 is conducted to a second input of delay 64 and is the output signal corresponding to the second of equations (10.6).

The signal generated by delay 62 is the output signal corresponding to the first of equations (10.6). Circuitry internal to block 60 implements the test performed in flowchart decision block 54 (Figure 10-3)

and resets the signal generated so that the first term of the sequence of tone signals are conducted to delays 62 and 64. Delay blocks 62 and 64 receive control signals (not shown) from initialization block 60 which cause the signals at their first inputs to be selected during the first time period following initialization and signals at their second inputs to be selected at all other time periods.

The method of digitally generating waveforms described in connection with Figure 10-1 and Figure 10-2 can be employed to provide a method of digital tone detection. If the tone to be detected is given by the sequence of samples

$$\{1, \cos 2\pi fT, \cos 2-2\pi fT, \cos 3-2\pi fT, \dots\} \quad (10.7)$$

and the tone signals received are represented by the sequence of samples

$$\{x_0, x_1, x_2, x_3, \dots\} \quad (10.8)$$

then, in the case of the method employing the Chebyshev sequence, an error term for each sample is given by the equation

$$\text{Error} = X_n - [(2 \cos 2\pi fT)X_{n-1} - X_{n-2}] \quad (10.9)$$

Equation (10.9) represents the difference between the measured tone signals X_n and the signal predicted by the corresponding term of the Chebyshev sequence C_n , shown in terms of X_{n-1} and X_{n-2} .

If the absolute value of the error is less than a selected threshold value, i.e.

$$|\text{error}| < \text{threshold} \quad (10.10)$$

then it is determined that the sequence of received samples (8) has the frequency of the sequence (7).

To control the "Q-factor" employed in conventional tone detectors the threshold value can be lowered, resulting in a "high-Q" filter, or raised, resulting in a "low-Q" filter.

Similarly, to increase the confidence level employed in conventional tone detectors, a voting-logic scheme can be used. For instance, several previous samples can be used in equation (10.9) to generate the predicted signal, rather than merely two, X_{n-1} and X_{n-2} . Then if a predetermined percentage of these samples, such as 90%, satisfy equation (10.10), it is determined that the sequence of received samples (8) has

the frequency of sequence (7). In this way, the two major parameters affecting the detection of tones can be chosen independently. This feature of the instant invention provides a significant advantage over the tone detectors of the prior art.

References

Related Patents

3649821	Mar., 1972	Gumacos	364/721.
<u>4249447</u>	Feb., 1981	Tomisawa	364/721.

1. Handscomb, *Methods of Numerical Approximation*, pp. 47-48, Pergamon Press Ltd, 1966.
2. Watson, *Approximation Theory and Numerical Methods*, pp. 102-110, copyright 1980 by John Wiley & Sons Ltd.

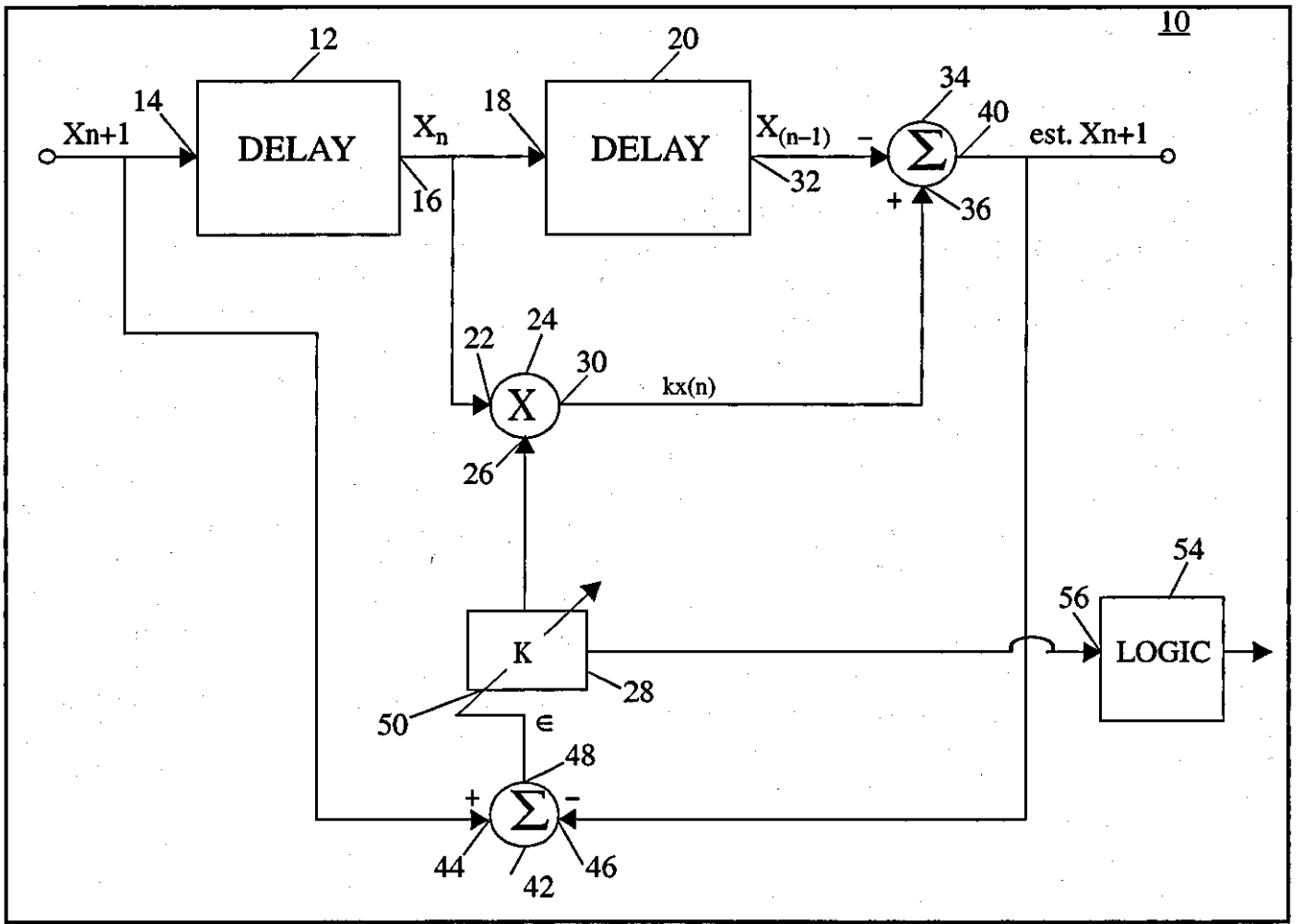
Chapter 11

Apparatus for Adaptively Tuning to a Received Periodic Signal

Abstract

This chapter is based on the invention patented under American patent 5,615,139 (S. Asghar, Y. Zhao, Aug. 1994).

An apparatus for adaptively tuning to a received periodic signal. The apparatus preferably employs digital signal processing techniques for algorithmic generation of a sinusoidal estimated output signal, determines the difference between the estimated output signal and the received signal, and generates an error signal based upon the difference between the estimated output signal and the received signal. The error signal is applied to vary at least one factor of the algorithmic determination of the estimated output signal appropriately to reduce the error between the estimated output signal and the received signal.



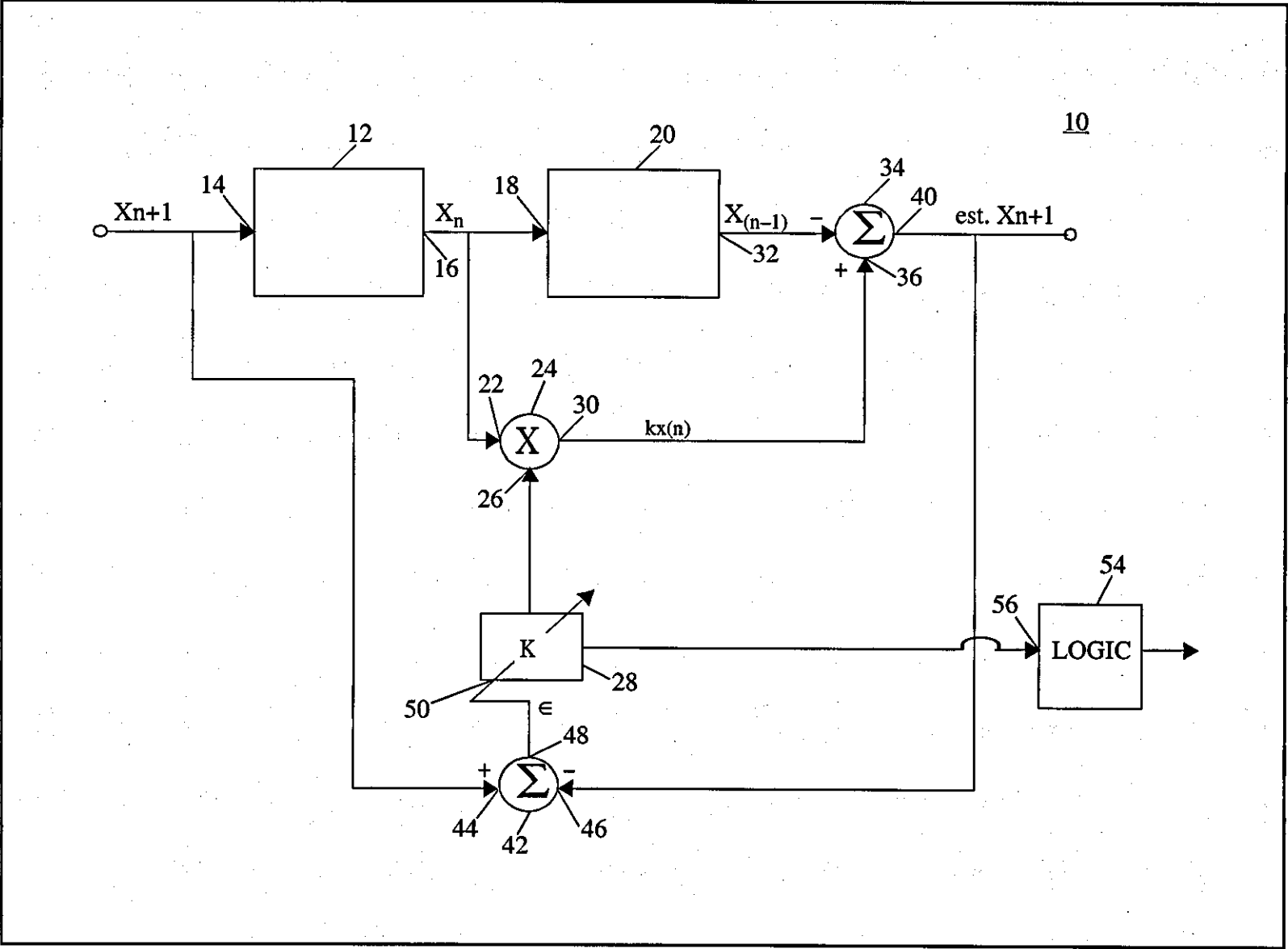
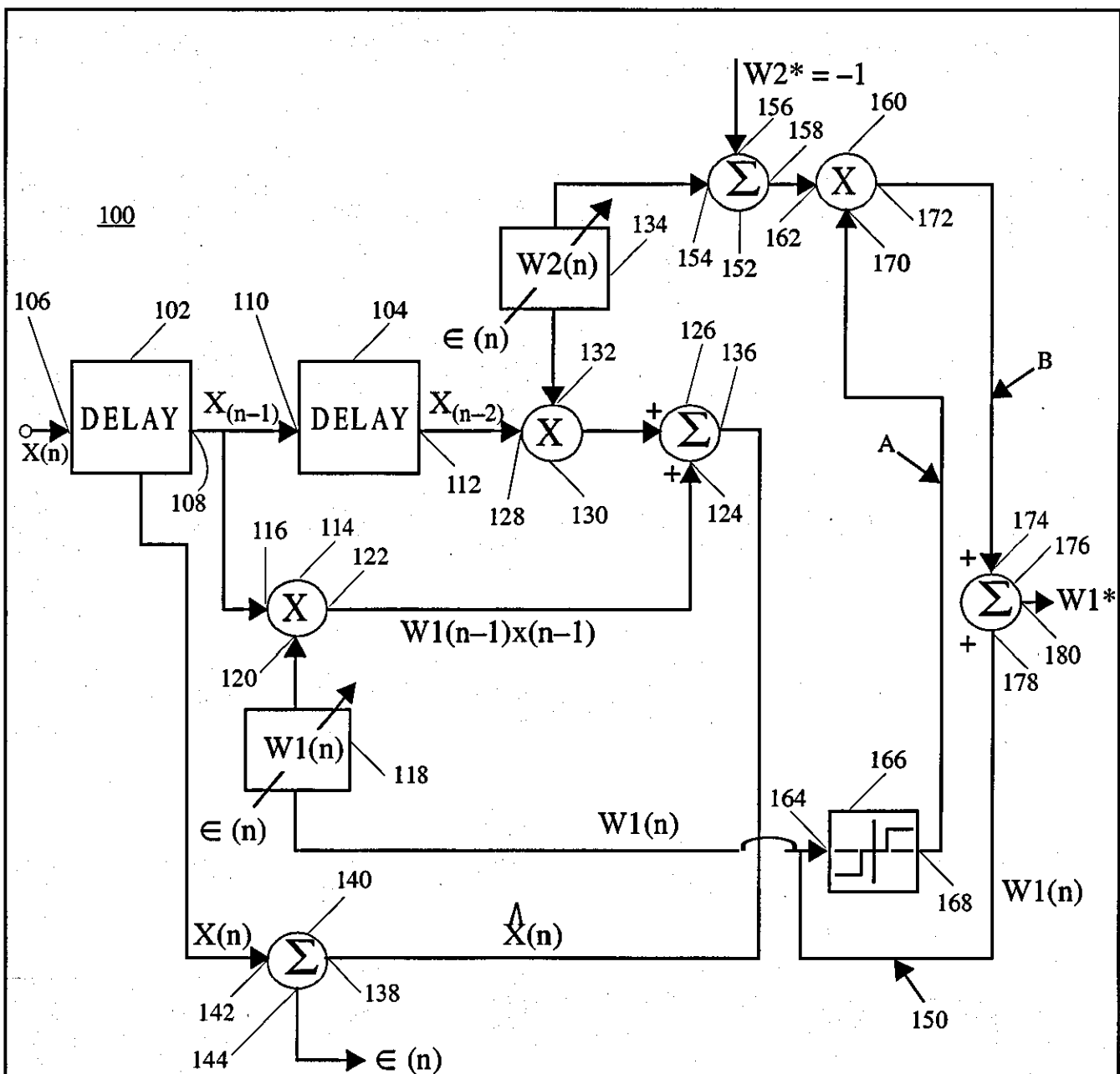


Figure 11-1



VALUE OF $W1(n)$	A	B
$-2 < W1(n) < -0.35$	-1	$-[1+W2(n)]$
$-0.35 < W1(n) < 0.28$	0	0
$0.28 < W1(n) < 2$	1	$[1+W2(n)]$

Figure 11-2

BACKGROUND OF THE INVENTION

The present invention is directed to an apparatus which will adaptively tune to a received periodic signal and, in one embodiment, will identify the frequency of the received signal to a user.

Adaptive tuning to a received periodic signal is useful in a variety of situations, such as tuning to a homing beacon or other similar navigational device, or, generally, tuning to or identifying any periodic signal of unknown frequency with speed and precision.

Prior art devices for identifying and tuning to an unidentified (i.e., the frequency is unknown) received signal involved filtering the received signal through an array of high Q filters, and establishing a frequency range within which the received frequency is propagated by noting which of the high Q filters in the array passes the received signal. In order to provide a high degree of accuracy in ascertaining the frequency of the received signal, such prior art circuits must necessarily have a large number of high Q filters in the filter array. A greater number of filters reduces the interval between detectable frequencies and, therefore, increased the accuracy by which the received frequency can be determined.

Thus, such prior art devices necessarily require that some estimation of the frequency of the received signal be known in advance in order that the frequency range detectable by the filter array can be assured to include the frequency expected to be received. In order to search a very wide frequency spectrum, it is sometimes required that a plurality of prior art devices be employed, with each device addressing a different range of frequencies.

The prior art devices thus described are slow to operate, require foreknowledge of the range within the anticipated frequency of the received signal will fall, and require the use of analog devices, such as high Q filters, which are known by those skilled in the art to be less stable than digital circuit elements, especially when subjected to variations of ambient temperature.

It is desirable that an apparatus be provided for detection of frequency of a received signal which is

stable in operation in a variety of environments, which is capable of accurate determination of the frequency of the received signal, and which is capable of ascertaining the frequency of the received signal speedily.

SUMMARY OF THE INVENTION

The invention is an apparatus for adaptively tuning to a received periodic signal. The apparatus preferably employs digital signal processing techniques for algorithmic generation of a sinusoidal estimated output signal, determines the difference between the estimated output signal and the received signal, and generates an error signal based upon the difference between the estimated output signal and the received signal. The error signal is applied to vary at least one factor of the algorithmic determination of the estimated output signal appropriately to reduce the error between the estimated output signal and the received signal.

In one embodiment of the present invention, when the error signal is reduced to zero, the algorithmic factor which was varied to effect reduction of the error signal is logically treated to ascertain the frequency of the received signal and such information is conveyed to a user.

An alternate embodiment of the invention accommodates white Gaussian noise in tuning to a received signal.

It is therefore an object of the present invention to provide an apparatus for adaptively tuning to a received periodic signal which employs digital signaling processing and is stable in a variety of operating environments.

A further object of the present invention is to provide an apparatus for adaptively tuning to a received periodic signal which is inexpensive to construct and ascertains the frequency of a received signal with a high degree of accuracy.

Yet a further object of the present invention is to provide an apparatus for adaptively tuning to a

received periodic signal which effects such adaptive tuning speedily.

Further objects and features of the present invention will be apparent from the following specification and claims when considered in connection with the accompanying drawings illustrating the preferred embodiment of the invention.

DETAILED DESCRIPTION OF THE DRAWINGS

Figure 11-1 is a schematic diagram of the preferred embodiment of the present invention.

Figure 11-2 is a schematic diagram of an alternate embodiment of the present invention configured to accommodate noise.

DETAILED DESCRIPTION OF THE INVENTION

As disclosed in U.S. Pat. No. 4,839,842 to Pyi and Asghar, which patent is assigned to the assignee of the present invention, a periodic waveform may be represented by the expression:

$$\cos (m+1) \delta = 2 \cos (m) \delta \cos \delta - \cos (m-1) \delta ; \quad (11.1)$$

where $\delta = 2 \cdot \pi \cdot f T$

f = signal frequency

T = sampling interval

Given that the sampling interval T is equal to one over f , (f , equals sampling frequency), the value of δ may be rewritten as:

$$\delta = \frac{2\pi f}{f_s}$$

For simplicity of expression, equation (11.1) can be rewritten as:

$$x_{n+1} = 2 \cos \delta x_n - x_{n-1} ; \quad (11.2)$$

where $x_n = \cos(n)\delta$

For any given signal frequency, f , the quantity $2 \cos \Delta$ is a constant, K . Thus, equation (11.2) can be expressed as:

$$x_{n+1} = Kx_n - x_{n-1}; \quad (11.3)$$

$$K = 2 \cos \delta \quad (11.4)$$

The relationship expressed in equation (11.3) may be advantageously employed in an apparatus for adaptively tuning to a received periodic signal as illustrated in Figure 11-1. In Figure 11-1, an adaptive tuning apparatus 10 is illustrated having a first delay circuit 12 receiving an incoming signal x_{n+1} at an input 14. First delay circuit 12 delays incoming signal x_{n+1} by one clock period and generates a once-delayed signal, x_n , at an output 16. Once-delayed signal, x_n , is provided to an input 18 of a second delay circuit 20 and to an input 22 of a multiplier circuit 24. Multiplier circuit 24 also receives a multiplier, K , at a multiplier input 26 from a multiplier source 28. Multiplier circuit 24 provides a multiplier output signal, Kx_n , at an output 30. Second delay circuit 20 delays once-delayed signal x_n one clock period and generates a twice-delayed signal, x_{n-1} , at an output 32. A first summing circuit 34 receives multiplier output signal Kx_n at an input 36, receives twice-delayed signal x_{n-1} at a negative input terminal 38, and generates an output signal at an output 40. The output signal generated at output 40 is an estimated signal, $\text{est. } x_{n+1}$, which comprises the difference between multiplier output signal Kx_n , less twice-delayed signal x_{n-1} .

A second summing circuit 42 receives estimated signal $\text{est. } x_{n+1}$ at a negative input terminal 46, receives received signal x_{n+1} at an input 44, and generates an output signal at an output 48. The output signal generated at output 48 is an error signal ϵ which comprises the difference between received signal x_{n+1} , less estimated signal $\text{est. } x_{n+1}$.

Error signal ϵ is applied to an input 50 of multiplier source 28. Multiplier source 28 responds to error signal ϵ to alter multiplier K provided to multiplier input 26 appropriately to reduce error signal ϵ .

When error signal ϵ equals zero, estimated signal $\text{est. } x_{n+1}$ equals received signal x_{n+1} , and adaptive tuning apparatus 10 has successfully tuned to received signal x_{n+1} .

When error signal ϵ equals zero, it is possible to ascertain the frequency of received signal x_{n+1} by recalling equation (11.4) above, so that for the frequency at which error signal ϵ equals zero,

$$K=2 \cos \delta \quad (11.5)$$

where θ identifies the target frequency, f_t (that is, the frequency of received signal x_{n+1}). Thus, since:

$$\delta = \frac{2\pi f_t}{f_s} \quad (11.6)$$

and since f_s , the sampling frequency, is known, then f_t can be readily determined by the expression:

$$f_t = \frac{f_s \arccos\left(\frac{K}{2}\right)}{2\pi} \quad (11.7)$$

Adaptive tuning apparatus 10 also may include a logic module 54 which receives an input signal from multiplier source 28 at an input 56. The input received from multiplier source 28 at input 56 comprises the value of K generated by multiplier source 28 when error signal ϵ equals zero. In the embodiment of the present invention illustrated in Figure 11-1, it is contemplated that logic module 54 will contain a stored arccosine table in appropriate detail to identify, with the accuracy desired, the arccosine of the quantity:

$$\left(\frac{K}{2}\right)$$

Further logic module 54 stores the value f_s of the sampling frequency and the value π to enable calculation of f_t , the target frequency, according to equation (11.7) above.

Figure 11-1 is a schematic diagram of an alternate embodiment of the present invention configured to accommodate noise. equation (11.3) discussed in connection with the preferred embodiment of the present invention illustrated in Figure 11-1, may be rewritten to account for white Gaussian noise in the

form:

$$x(n) = Wx(n-1) - x(n-2) \quad (11.8)$$

$$W = 2\cos \frac{2\pi f(n)}{f_s}$$

where $f(n)$ equals a time varying frequency of the received sinusoidal signal and f_s equals sampling frequency.

Referring to Figure 11-1, an apparatus 100 is illustrated having a first delay circuit 102 and a second delay circuit 104. An input signal, $x(n)$, is received by apparatus 100 at an input 106 to first delay circuit 102. First delay circuit 102 delays input signal $x(n)$ one clock period and generates at an output 108 a once-delayed signal $x(n-1)$; $x(n-1)$ is substantially input signal $x(n)$ delayed one clock period. Second delay circuit 104 receives once-delayed signal $x(n-1)$ at an input 110 and delays once-delayed signal $x(n-1)$ by one further clock period, generating at a twice-delayed signal $x(n-2)$ at an output 112. Once-delayed signal $x(n-1)$ is also applied to a multiplier 114 at an input 116.

A first weighted factor source 118 provides a first weighted factor, $W1(n)$, to an input 120 of multiplier 114. Multiplier 114 generates at an output 122 a weighted once-delayed signal which is substantially the product of once-delayed signal $x(n-1)$ and first weighted factor $W1(n)$. This weighted once-delayed signal is applied to an input 124 of a summing node 126.

Twice-delayed signal $x(n-2)$ is applied to an input 128 of a multiplier 130. Multiplier 130 also receives at an input 132 a second weighted factor, $W2(n)$, from a second weighted factor source 134. Summing node 126 produces at an output 136 an estimation of input signal $x(n)$, $\hat{x}(n)$, estimated input signal $\hat{x}(n)$ is applied to an input 138 of a summing node 140. Also received by summing node 140 at an input 142 is input signal $x(n)$. Summing node 140 produces at an output 144 an error signal ϵ . First weighted factor source 118 and second weighted factor source 134 are responsive to error signal ϵ to vary first weighted factor $W1(n)$ and second weighted factor $W2(n)$ appropriately to reduce error signal ϵ . That is, $W1(n)$ and $W2(n)$ are appropriately varied to reduce the difference between input signal $x(n)$

and estimated input signal $x(n)$. As may be seen from inspection of Figure 11-2:

$$\hat{x}(n) = W1(n-1)x(n-1) + W2(n-1)x(n-2) \quad (11.9)$$

Both weighted factors $W1(n)$, $W2(n)$ are adaptive and are preferably determined by a least mean square algorithm, so that:

$$W1(n) = W1(n-1) + \mu \varepsilon(n)x(n-1) \quad (11.10)$$

$$W2(n) = W2(n-1) + \mu \varepsilon(n)x(n-2) \quad (11.11)$$

Thus, from Figure 11-2 and equation (11.9):

$$\varepsilon(n) = x(n) - \hat{x}(n) = x(n) - W1(n-1)x(n-1) - W2(n-1)x(n-2) \quad (11.12)$$

If the input signal $x(n)$ is a pure sine wave, then, from equation (11.8), $W1(n)$ would converge to a value W defined by the expression:

$$W = 2 \cos \frac{2\pi f(n)}{f_s}$$

and $W2$ would have a value of -1 . However, one realistically should expect to receive a sine wave accompanied by a noise factor. Such a real world situation may be generally expressed by:

$$x(n) = x_s(n) + N(n) \quad (11.13)$$

$x_s(n)$ = signal term

$N(n)$ = noise term

In the presence of white Gaussian noise, the inventors have found that the first weighted factor $W1(n)$ does not provide an accurate frequency estimation. Thus, second weighted factor $W2(n)$ is employed to estimate noise energy. A correction is then made to first weighted factor $W1(n)$ to obtain an accurate frequency reading.

Further considering equation (11.10):

$$W1(n) = W1(n-1) + \mu \varepsilon(n)x(n-1) \quad (11.14a)$$

μ = gain factor

However,

$$W1(n-1) = W1(n-2) + \mu \epsilon(n-1)x(n-2) \quad (11.14b)$$

Thus,

$$W1(n) = W1(n-2) + \mu[\epsilon(n)x(n-1) + \epsilon(n-1)x(n-2)] \quad (11.14c)$$

Repetitive substitution yields an expansion of equation (11.14c):

$$W1(m) = W1(m) + \mu[\epsilon(n)x(n-1) + \epsilon(n-1)x(n-2) + \dots + \epsilon(m+1)x(m)] \quad (11.14d)$$

Therefore:

$$W1(n) = W1(m) + \mu \sum_{k=m+1}^n \epsilon(k)x(k-1) \quad (11.15)$$

When the system embodied in the apparatus 100 is in steady state, $\epsilon(n)$ is in the presence of noise, a small signal moving randomly about zero. In such a steady state, the weighting functions $W1(n)$ and $W2(n)$ may be defined as:

$$\begin{aligned} W1(n) &= W1^* + \delta W1(n) \\ W2(n) &= W2^* + \delta W2(n) \end{aligned} \quad (11.16)$$

Comparing equation (11.16) with equation (11.15), and noting a correspondence between the second terms of each of equations (11.15) and (11.16), one may note:

$$\begin{aligned} \text{If } \mu \rightarrow 0, \text{ then } |\delta W1(n)| &\ll 1 \\ \text{and } |\delta W2(n)| &\ll 1 \end{aligned} \quad (11.17)$$

Thus, substituting equation (11.12) into equation (11.15) yields:

$$W1(n) = W1(m) + \mu \sum_{k=m+1}^n [x(k) - W1(k-1)x(k-1) - W2(k-1)x(k-2)]x(k-1) \quad (11.18)$$

Substituting equation (11.13) and (11.16) into equation (11.18) yields:

$$\begin{aligned}
W1(n) = W1(m) + \mu \sum_{k=m+1}^n & \{x_s(k) + N(k)\} - \{W1^* + \delta W1(k-1)\} \\
& \{x_s(k-1) + N(k-1)\} - \{W2^* + \delta W2(k-1)\} \\
& \{x_s(k-2) + N(k-2)\}][x_s(k-1) + N(k-1)]
\end{aligned} \tag{11.19}$$

From equation (11.8), we know that (recall that $W2^* = -1$, in steady state; the case assumed in connection with equation (11.8)):

$$x_s(k) - W1^* x_s(k-1) - W2^* x_s(k-2) = 0 \tag{11.20}$$

Thus, equation (11.19) reduces to:

$$\begin{aligned}
W1(n) = W1(m) + \mu \sum_{k=m+1}^n & [N(k) - \delta W1(k-1)x_s(k-1) - \\
& W1^*N(k-1) - \delta W1(k-1)N(k-1) - \\
& \delta W2(k-1)x_s(k-2) - W2^*N(k-2) - \\
& \delta W2(k-1)N(k-2)][x_s(k-1) + N(k-1)]
\end{aligned} \tag{11.21}$$

Assuming that noise $N(k)$ is independent of signal $x_s(k)$, as is the case with white Gaussian noise (that is, $N(k)$ is an uncorrelated signal), then:

$$E\{N(k)\} = 0 \text{ (mean of } N(k) = 0)$$

$$E\{N(k)N(k+n)\} = 0, \text{ when } n \neq 0 \tag{11.22}$$

Therefore:

$$\begin{aligned}
\sum_{x=0}^{\infty} x_s(k)N(k) = 0 & \quad \text{and} \\
\sum_{x=0}^{\infty} x_s(k-1)N(k) = 0 &
\end{aligned} \tag{11.23}$$

That is, in general:

$$\sum_{k=0}^{\infty} x_s(k+L)N(k) = 0, \text{ for any } L \tag{11.24}$$

Expanding equation (11.21) yields:

$$\begin{aligned}
W1(n) = W1(m) + \mu \sum_{k=m+1}^n & \underline{[N(k)x_s(k-1) +} \\
& \underline{N(k)N(k-1) - \delta W1(k-1)x_s^2(k-1) -} \\
& \underline{\delta W1(k-1)x_s(k-1)N(k-1) - W1*N(k-1)x_s(k-1) -} \\
& \underline{W1N^2(k-1) - \delta W1(k-1)N(k-1)x_s(k-1) - \delta W1(k-1)N_2(k-1) -} \\
& \underline{\delta W2(k-1)x_s(k-1)x_2(k-2) - \delta W2(k-1)x_s(k-2)N(k-1) -} \\
& \underline{W2*N(k-2)x_s(k-1) - W2*N(k-2)N(k-1)} \\
& \underline{\delta W2(k-1)N(k-2)x_s(k-1) - \delta W2(k-1)N(k-2)N(k-1)]}
\end{aligned} \tag{11.25}$$

By equation (11.23) and equation (11.24), we know that all terms underlined in equation (11.25)

equal zero, so:

$$\begin{aligned}
W1(n) = W1(m) + \mu \sum_{k=m+1}^n & [-\delta W1(k-1)x_s^2(k-1) - \\
& W1*N^2(k-1) - \delta W1(k-1)N^2(k-1) - \delta W2(k-1)x_2(k-1)x_s(k-2)]
\end{aligned} \tag{11.26}$$

Recall that when $|\mu| \ll 1$, $|\delta W1(k)| \ll 1$ and $|\delta W2(k)| \ll 1$, so

$$W1(n) \cong W1(m) + \mu \sum_{k=m+1}^n [-W1*N^2(k-1)] \tag{11.27}$$

Therefore,

$$W1(n) \cong W1(m) - \mu W1*Rn(0) \tag{11.28}$$

Observing that $W1(n) \cong W1^*$ when $n \rightarrow \infty$, equation (11.28) may be expressed:

$$W1^* \cong W1(m) - \mu W1^* Rn(0) \tag{11.29}$$

Applying a similar derivation to equation (11.11) yields:

$$W2^* \cong W2(m) + \mu Rn(0) \tag{11.30}$$

$$\mu Rn(0) = W2^* - W2(m) \tag{11.31}$$

Substituting equation (11.31) in equation (11.29) yields;

$$W1^* \cong W1(m) - W1^*[W2^* - W2(m)] \quad (11.32)$$

$$W1^* \cong W1(m) - 2\cos \frac{2\pi f_0}{f_s} [W2^* - W2(m)] \quad (11.33)$$

$$W1^* \cong W1(m) + 2\cos \frac{2\pi f_0}{f_s} [1 + W2(m)] \quad (11.34)$$

Given the cosine function, generally, one can make the following approximations:

$$2\cos \frac{2\pi f_0}{f_s} = \begin{cases} -\frac{1}{2} & 1 < \frac{f_s}{f_0} < 3.6 \\ 0 & 3.6 < \frac{f_s}{f_0} < 4.4 \\ +\frac{1}{2} & 4.4 < \frac{f_s}{f_0} < \infty \end{cases} \quad (11.35)$$

Substituting equation (11.35) in equation (11.34) yields:

$$W1^* = \begin{cases} W1(m) - [1 + W2(m)] & 2 < \frac{f_s}{f_0} < 3.6 \\ W1(m) & 3.6 < \frac{f_s}{f_0} < 4.4 \\ W1(m) - [1 + W2(m)] & 4.4 < \frac{f_s}{f_0} < \infty \end{cases} \quad (11.36)$$

As previously mentioned, the quantity

$$\frac{f_s}{f_0}$$

Ps must be greater than two to satisfy the requirements of Nyquist sampling theory. Thus, if

$$W1(m) \cong 2\cos \frac{2\pi f_0}{f_s}$$

then equation (13.36) may be represented as:

$$W1^* = \begin{cases} W1(m) - [1 + W2(m)] - 2 W1(m) - 0.35 & \\ W1(m) & -0.35 < W1(m) < 0.28 \\ W1(m) - [1 + W2(m)] & 0.28 < W1(m) < 2 \end{cases} \quad (11.37)$$

Thus, given the theoretical background provided by equations (11.8) through (11.37), the operation of apparatus 100 illustrated in Figure 11-2 may be further explained in that apparatus 100 includes a frequency identification circuit 150. Frequency identification circuit 150 comprises a summing node 152 which receives second weighted factor W2 from second weighted factor source 134 at an input 154. Summing node 152 also receives at a second input 156 the steady state value of W2* (= -1). Since input 156 is an inverting input, summing node 152 produces at its output 158 a signal representing the quantity [1+W2(n)], which signal is received at an input 162 of a multiplier 160.

The quantity W1(n) is applied to an input 164 of a threshold circuit 166. Threshold circuit 166 is preferably designed to substantially implement the approximations represented by equation (11.37) above, so that the signal, "A", provided by threshold circuit 166 at its output 168 is variable, as indicated in the table associated with Figure 11-2, according to the value of signal W1(n) received at input 164. The output signal provided by threshold circuit 166 is applied to an input 170 of multiplier 160 so that multiplier 160 provides at its output 172 a signal "B". Signal "B" varies as indicated in the table associated with Figure 11-2, depending upon the value of W1(n) received at input 164 of threshold circuit 166. Signal "B" is applied to an input 174 of a summing node 176; the signal W1(n) is provided to a second input 178 of summing node 176 from first weighted factor source 118, and summing node 176 provides at its output 180 the value W1*. Thus signal W1* is calculated by frequency identification circuit 150 in accordance with equation (11.37) above.

The value provided at output 180 of summing node 176 uniquely identifies the frequency of the sinusoidal signal, $x_s(n)$, received at input 106 of delay circuit 102 through a look-up table or similar arrange-

ment in a logic circuit (not shown in Figur e11-2). In such manner, a second weighted factor, $W2(n)$, is employed to estimate noise energy in the received signal $x(n)$ and second weighted factor $W2(n)$ is employed to apply a correction to first weighted factor $W1(n)$ in order to accurately reflect an index uniquely identifying the sinusoidal wave form $x_s(n)$ contained within the received signal x_n , which is received in the form of equation (11.38), i.e.,

$$x_n = x_s(n) + N(n) \quad (11.38)$$

It is to be understood that, while the detailed drawings and specific example given describe a preferred embodiment of the invention, they are for the purpose of illustration only, that the apparatus of the invention is not limited to the precise details and conditions disclosed and that various changes may be made therein without departing from the spirit of the invention.

References

Related Patents

<u>4433422</u>	Feb., 1984	Kurth	364/724.
<u>4438504</u>	Mar., 1984	Favin	364/724.
<u>4791390</u>	Dec., 1988	Harris et al.	364/724.
<u>4839842</u>	Jun., 1989	Pyi et al.	364/721.
<u>5189381</u>	Feb., 1993	Asghar et al.	331/179.

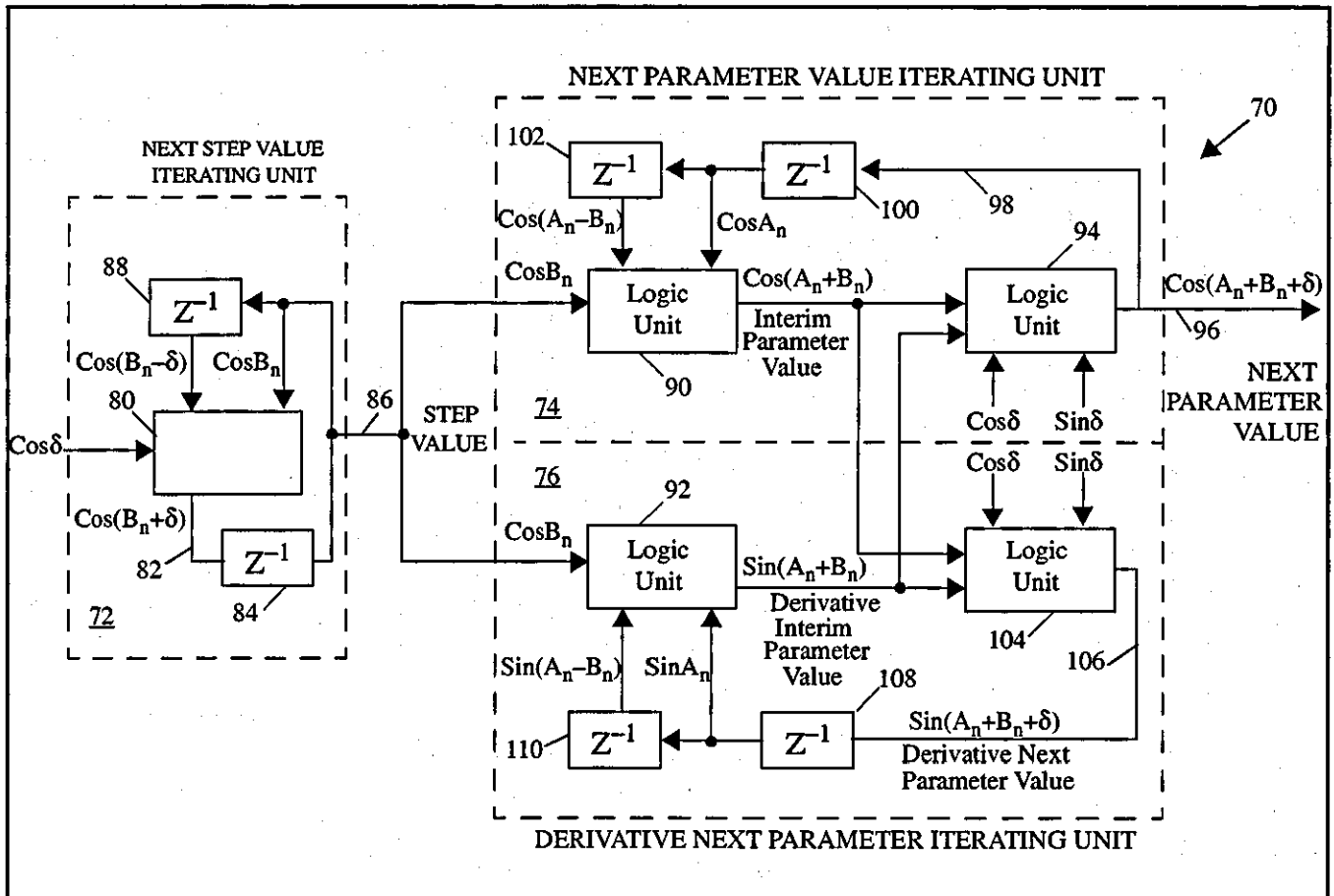
Chapter 12

Apparatus and Method for Synthesizing a Sinusoidal Signal

Abstract

This chapter is based on the invention patented under American patent 5,745,648 (S. Asghar, M. Iretton, May 1997).

An apparatus and method for synthesizing a sinusoidal signal generated from a plurality of sample values taken at sample times in succeeding sample intervals. The signal is defined by a parameter which varies by a step value during predetermined sample intervals when the parameter changes from a first value to a second value. The apparatus includes a first logic unit for iteratively treating an initial step value to generate succeeding samples of the step value and a next step value. The next step value is the step value in a next-succeeding sample interval. A second logic unit is included for iteratively generating a next parameter value which is the parameter value during the next-succeeding sample interval. The second logic unit receives succeeding samples of the step value and iteratively generates succeeding samples of an interim parameter value and succeeding samples of a derivative interim parameter value. The interim parameter value is the parameter value intermediate a current parameter value in a current sample interval and a next parameter value during a next-succeeding sample interval. The second logic unit iteratively treats the succeeding samples of the interim parameter value and the derivative interim parameter value to generate the next parameter value. The second logic unit continues iterative generating of the next parameter value until the end of a time interval or until the particular parameter value substantially equals the second parameter value.



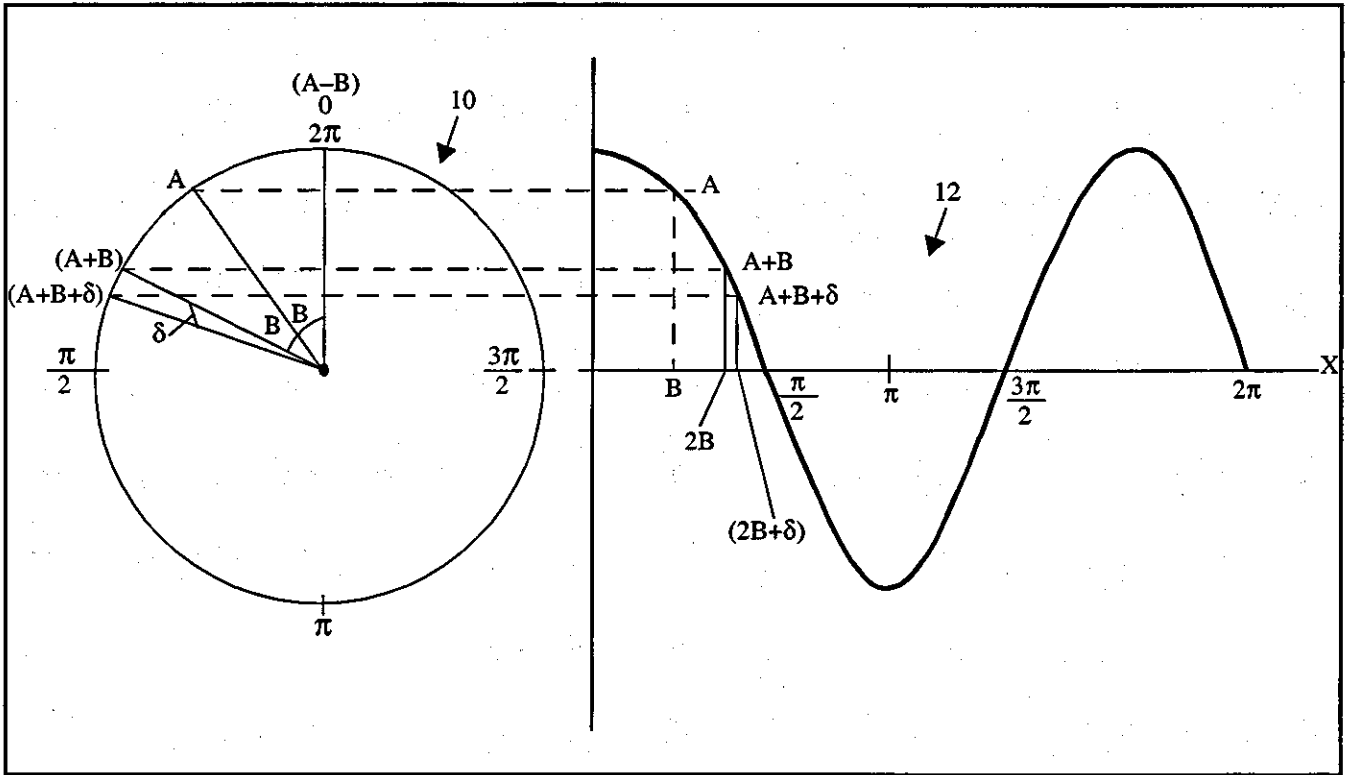


Figure 12-1

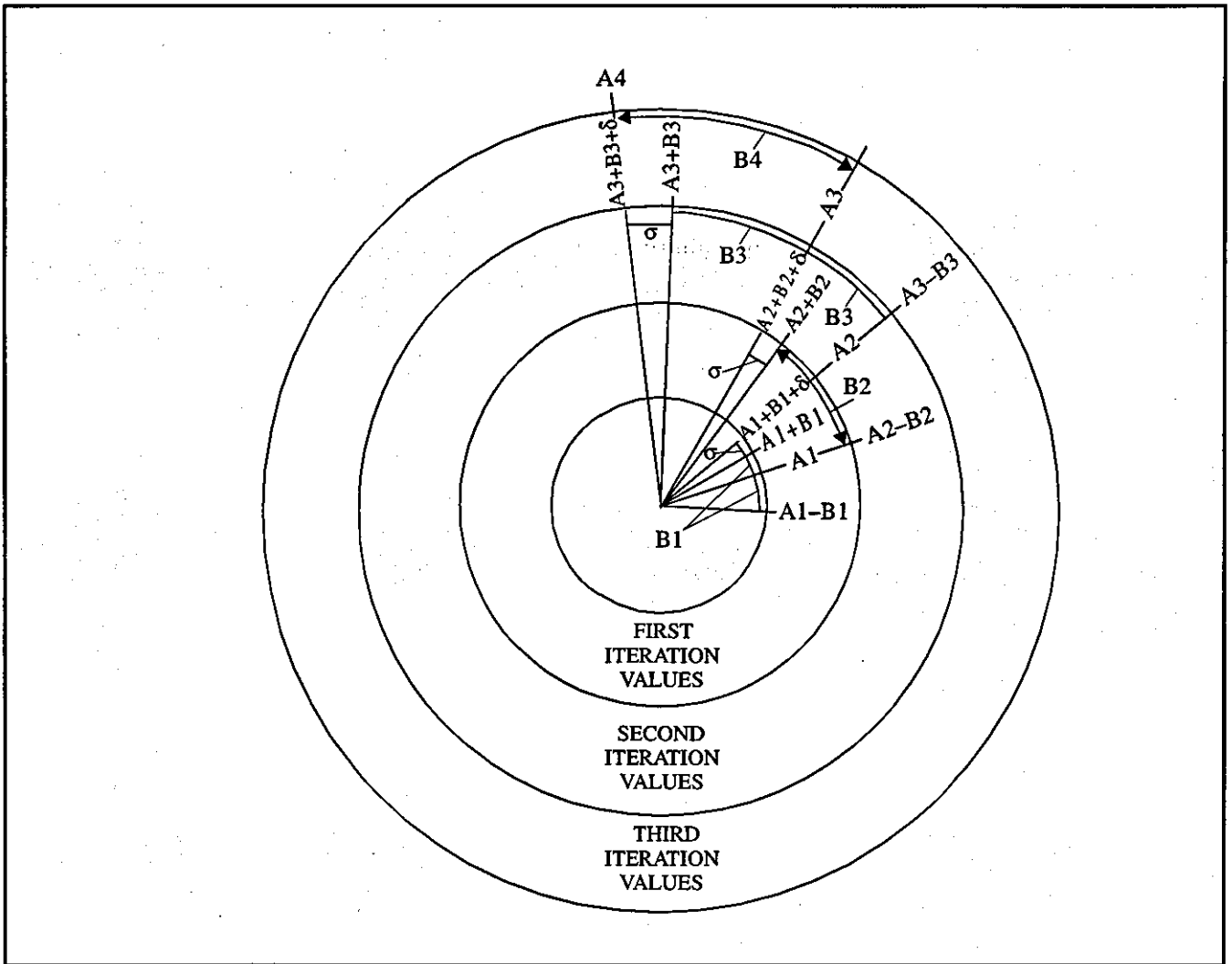


Figure 12-2

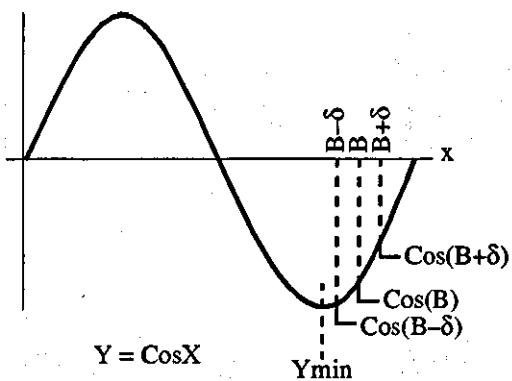


Figure 12-3a

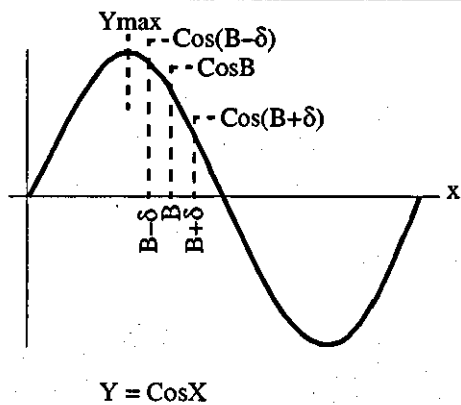


Figure 12-3d

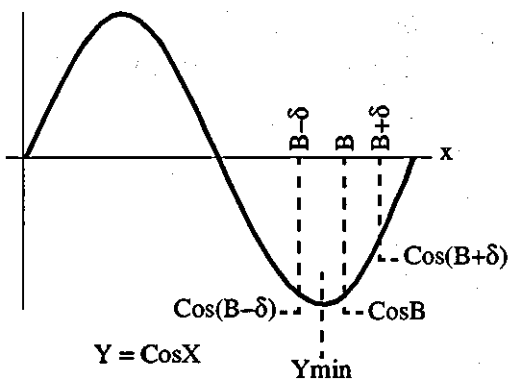


Figure 12-3b

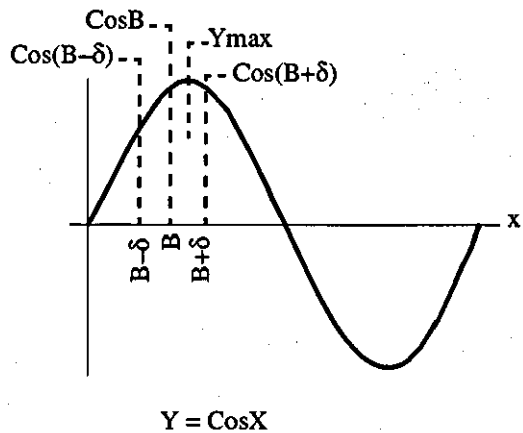


Figure 12-3e

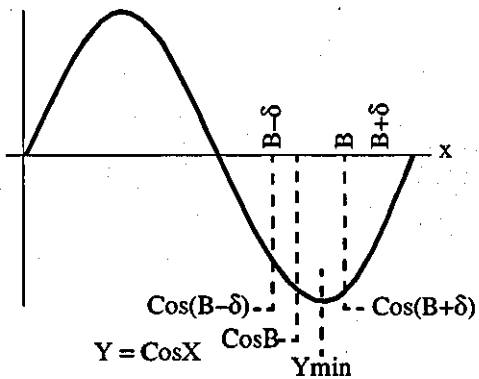


Figure 12-3c

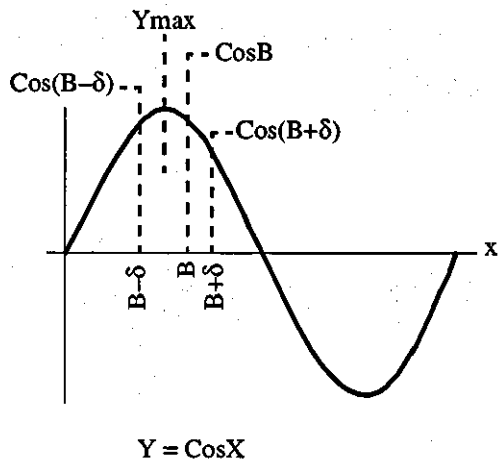


Figure 12-3f

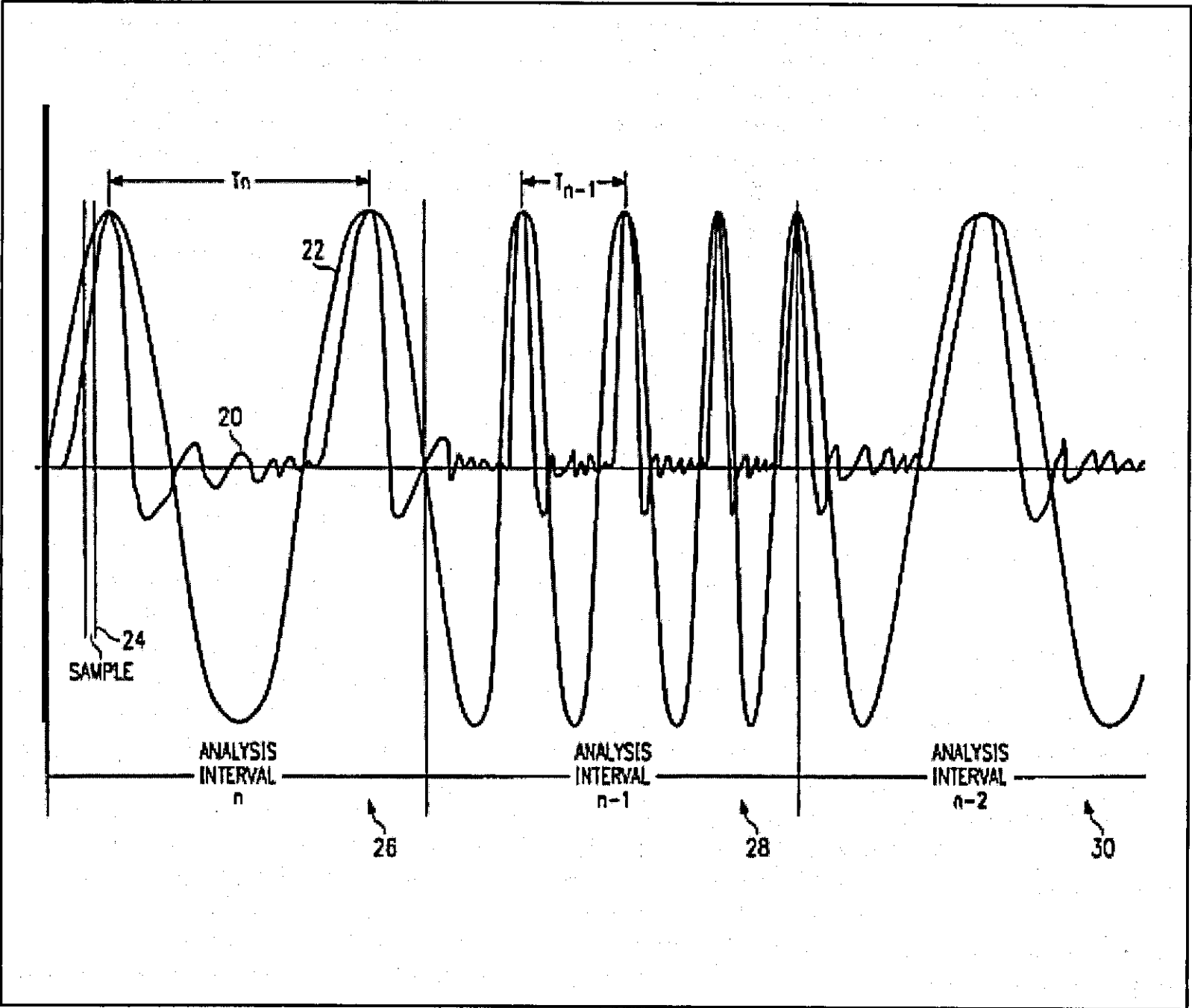


Figure 12-4

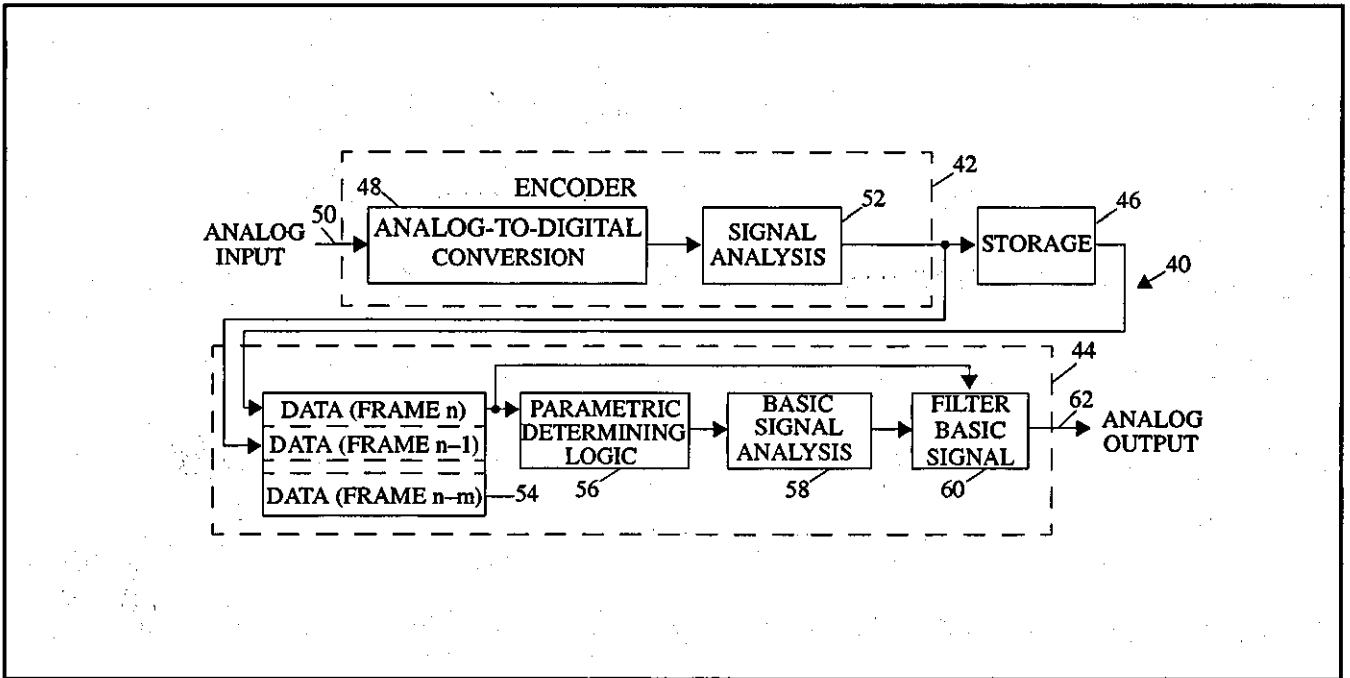


Figure 12-5

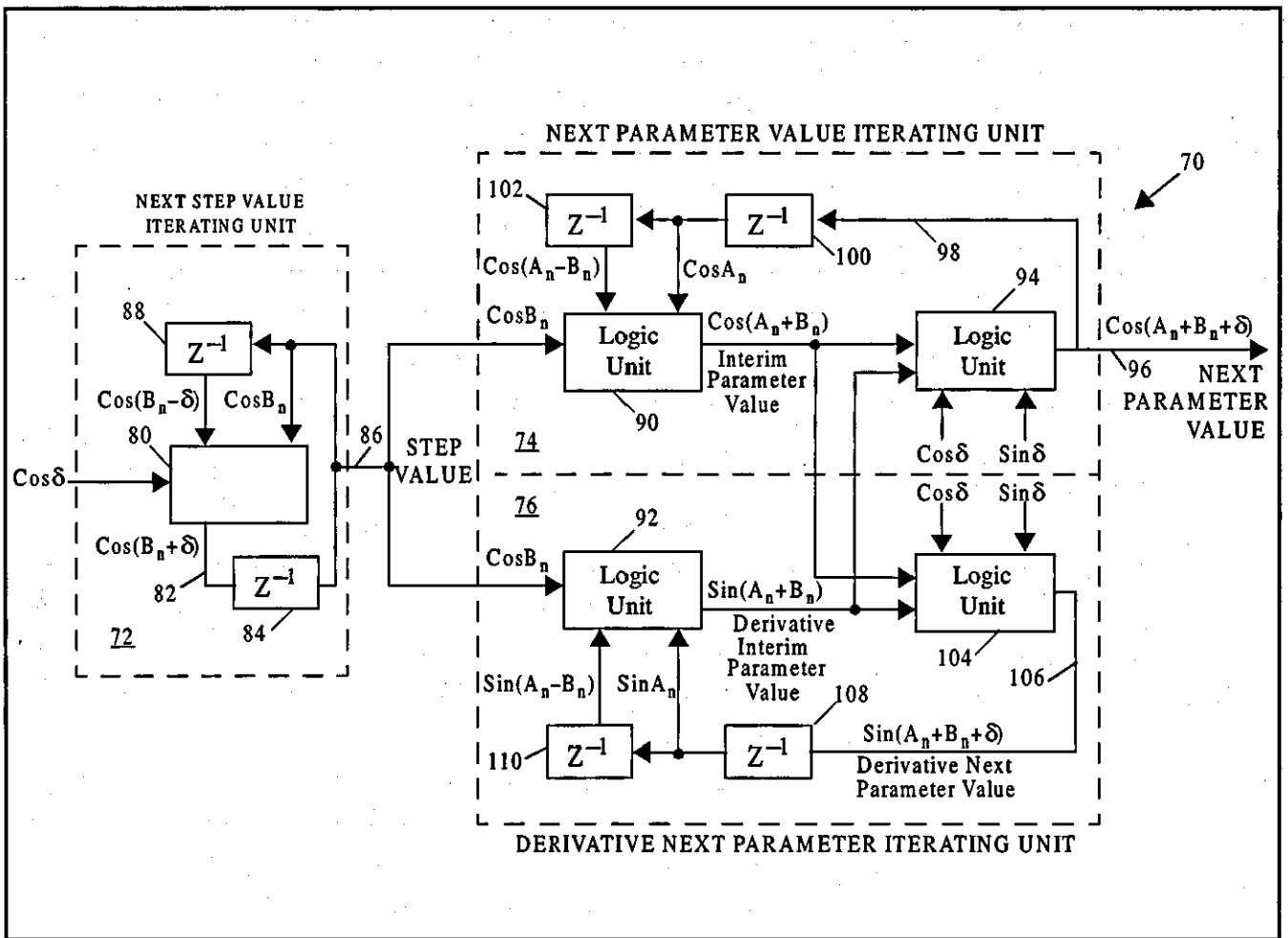


Figure 12-6

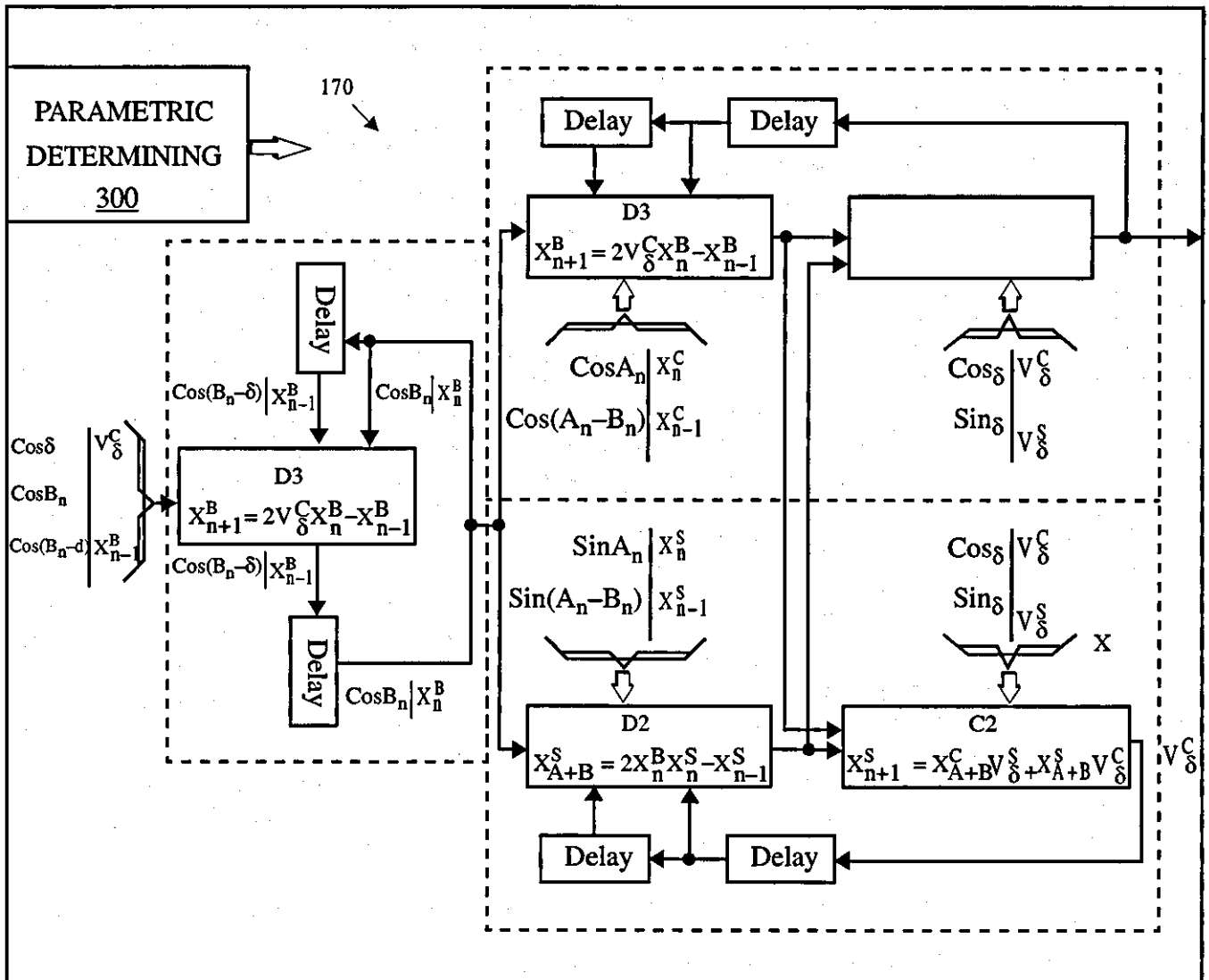


Figure 12-7

Figure 12-8

BACKGROUND OF THE INVENTION

The invention relates to an apparatus and method for generating sinusoidal signals that are modulated under control of a signal which is conditioned to perform a certain task.

Specifically, the invention relates to an apparatus and method for generating sinusoidal signals which are variable in a particular parameter, such as frequency, from one time frame to another time frame, or from one sampling interval to another sampling interval. A representative application of the present invention is generation of a sinusoidal signal which is appropriate as a pitch component upon vocal tract information which may be imposed to reproduce speech information.

Earlier sinusoidal signal generating apparatus and methods often have involved a pure pulse approach in order that the signal could be digitized. However, a problem with such a pure pulse approach is that the required data manipulations can occur only at specified points in time. Thus, adjacent samples may have sufficiently different frequencies that no smooth transition is achievable between samples.

The improved apparatus and method disclosed for generating sinusoidal signals enables generation of a mathematically continuous (as opposed to a discrete sample represented) signal. Such a continuous mathematical signal can be finessed from period to period (i.e., sample to sample) to provide smooth segments in the varying signal parameter (e.g., frequency).

Thus, the present invention provides an apparatus and method for producing a sinusoidal signal and calculating transitions of a parameter "on the fly" from a first value to a second value. The preferred parameter illustrated in this application is frequency, but the approach and model disclosed apply equally well for any sinusoidal parameter.

SUMMARY OF THE INVENTION

An apparatus and method are disclosed for synthesizing a sinusoidal signal generated from a plurality of sample values taken at a plurality of sample times. Each respective sample value is generated at a respective sample time, the plurality of sample times defining a plurality of succeeding sample intervals. The sinusoidal signal is defined by at least one parameter value, a particular parameter value of the at least one parameter value varying by a step value during a predetermined sample intervals of the plurality of sample interval when the particular parameter value changes from the first parameter value to a second parameter value. The apparatus comprises a first logic unit for iteratively treating an initial value of the step value to generate succeeding samples of the step value and a next step value. The next step value is the step value in a next-succeeding sample interval of the plurality of sample intervals.

The apparatus further comprises a second logic unit for iteratively generating a next parameter value which is the particular parameter value during the next-succeeding sample interval. The second logic unit is operatively connected with the first logic unit and receives succeeding samples of the step value from the first logic unit. The second logic unit iteratively generates succeeding samples of an interim parameter value and iteratively generates succeeding samples of a derivative interim parameter value. The interim parameter value is the particular parameter value intermediate a current particular parameter value in a current sample interval and a next particular parameter value during a next-succeeding sample interval following the current sample interval. The derivative interim parameter value is algorithmically related with the interim parameter value.

Further, the second logic unit iteratively treats the succeeding samples of the interim parameter value and succeeding samples of the derivative interim parameter value to iteratively generate the next parameter value. The second logic unit continues such iterative generating of the next parameter value until the

end of a time interval (or sample interval) or until the particular parameter value substantially equals the second parameter value.

In its preferred embodiment, the second logic unit includes a third logic unit for generating succeeding samples of the derivative interim parameter value. The third logic unit receives succeeding samples of the step value from the first logic unit and receives succeeding samples of the interim parameter value and iteratively generates the succeeding samples of the derivative interim parameter value.

The particular parameter value is preferably expressed as a particular parameter sinusoidal value; the particular parameter sinusoidal value being a sinusoidal value of an angular displacement representing the particular parameter value in a phasor representation.

The step value is also preferably expressed as a step value sinusoidal value being a sinusoidal value of an angular displacement representing the step value in a phasor representation.

Further, the interim parameter value is preferably expressed as an interim parameter value sinusoidal value which is a sinusoidal value of an angular displacement representing the interim parameter value in a phasor representation.

Still further, the derivative interim parameter value is preferably expressed as a derivative interim parameter value sinusoidal value which is a sinusoidal value of an angular displacement representing the derivative interim parameter value in a phasor representation; and the derivative interim parameter value sinusoidal value is preferably 90° displaced from the interim parameter value sinusoidal value in a phasor representation.

An advantage of the preferred embodiment of the present invention is that the first logic unit, the second logic unit, and the third logic unit are embodied as programmable logic units dealing with difference equations involving sinusoidal values, as opposed to dealing with equations which require deriving sinusoidal values from angular value inputs. Such treatment and calculation of the various values required by the apparatus and method of the present invention from directly entered values allows speedier calcula-

tion and therefore allows a higher sampling rate for providing samples to represent the desired sinusoidal signal being synthesized.

A further advantage of the preferred embodiment of the present invention is that such dealing in values, as opposed to deriving values from angular value inputs, also simplifies hardware requirements to support such logical activities.

It is, therefore, an advantage of the present invention to provide an apparatus and method for synthesizing a sinusoidal signal which directly employs sinusoidal values in effecting required logical calculations.

It is a further advantage of the present invention to provide an apparatus and method for synthesizing a sinusoidal signal which requires simple hardware.

It is yet a further advantage of the present invention to provide an apparatus and method for synthesizing a sinusoidal signal which can quickly calculate required values for accurately reproducing information regarding the sinusoidal signal.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 12-1 is a basic phasor representation of signal synthesis.

Figure 12-2 is a phasor representation of iterative signal synthesis.

Figure 12-3a-Figure 12-3f are graphic representations of various cases relating to the sample displacement value for a sinusoidal signal.

Figure 12-4 is a graphic representation of a voice signal with a fundamental sinusoidal signal indicated.

Figure 12-5 is a schematic diagram of a signal processing system using the present invention.

Figure 12-6 is a schematic diagram of an embodiment of the present invention.

Figure 12-7 is a schematic diagram of the preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Figure 12-1 is a basic phasor representation of signal synthesis. In Figure 12-1, a phasor representation 10, and its associated, more familiar, orthogonal axes representation 12 are illustrated. In phasor representation 10, since the function illustrated is $y = \cos x$, the zero radian angle of displacement is situated at the top (north) position of phasor representation 10. If the phasor is displaced counterclockwise about phasor representation 10 by a phasor angle B, the value A of $y = \cos x$ at displacement $X=B$ in two dimensional representation 12 corresponds to that phasor displaced by B radians in phasor representation 10. Thus, phasor angle B is analogous to angular velocity ω for a time-related function. Phasor angle B, therefore, as illustrated in Figure 12-1, may be expressed as $B = \frac{dA}{dx}$

If the phasor angle B is altered during the traversal of the phasor in phasor diagram 10 (for example, in a situation where the frequency of the signal varies) the variation of phasor angle B may be represented by a value δ . That value δ step is analogous to angular acceleration ($d\omega/dt$) in a time-based relationship. Thus, in Figure 12-1, δ may be characterized as $\delta = \frac{dB}{dx}$

Thus, in the orthogonal axes representation 12 of Figure 12-1, a function $y = f(x)$ is represented in (x,y) coordinates with displacement in radians (i.e., B) represented along the x -axis, and the function value $y = f(x) = \cos x$ is represented as function value A .

Figure 12-2 is a phasor representation of iterative signal synthesis. In Figure 12-2, a sinusoidal function (e.g., $y = \cos x$) is represented as being iteratively generated during three iterations with the phasor angle B being increased during each iteration by a value δ . Such a situation may exist, for example, when one wishes to vary a frequency of a sinusoidal signal from a first frequency to a second frequency in linear fashion by increasing phasor angle B . In such a situation, the predetermined value δ is preferably chosen so that, in altering the sinusoidal signal from a first frequency f_1 to a second frequency f_2 , phasor angle B may be altered from phasor angle B_1 to phasor angle B_2 such that $\frac{B_2 - B_1}{n} = \delta$

Where n = number of samples.

Thus, in Figure 12-2, a first iteration 14 provides that function value A vary from a first function value A_1 to a second function value A_2 , the phasor angle difference between the function values A_1, A_2 being phasor angle B_2 . However, phasor angle B_2 equals phasor angle B_1 plus δ .

In second iteration 16, function value A_2 is changed to function value A_3 . The difference between the function values A_2, A_3 is phasor angle B_3 ; phasor angle B_3 is equal to phasor angle B_2 plus δ .

In third iteration 18, function value A_3 is changed to function value A_4 . The difference between the function values A_3, A_4 is phasor angle B_4 ; phasor angle B_4 is equal to phasor angle B_3 plus δ .

Thus, in such an iteration where a parameter change is effected by a linearly changing phasor angle, it may be seen generally that:

$$B_{n+1} = B_n + \delta$$

This is a simplified example which presumes that δ is a constant to provide for linear change from a first parameter to a second parameter (in this example the parameter is frequency). Of course, δ could be varied according to any relationship (for example: a polynomial relationship, an exponential relationship,

or a sinusoidal relationship) without affecting the validity of the relationships illustrated in this description.

Thus, the signal represented by a function which is expressible by a phasor notation, as illustrated in Figure 12-1 and Figure 12-2, is modulated by variation of value δ . In order to facilitate understanding of the present invention and to keep the explanation straightforward, value δ will be presumed hereinafter to vary linearly, that is δ is presumed to be a constant locally within a given sample or analysis interval.

By inspection of Figure 12-2, observe:

$$\cos(A_{n+1}) = \cos(A_n + B_{n+1}) = \cos(A_n + B_n + \delta)$$

$$\cos(A_{n-1}) = \cos(A_n - B_n)$$

$$B_{n+1} = B_n + \delta$$

$$B_{n+1} = B_n - \delta$$

The following discussion will rely upon the four following relationships:

$$\cos(x+y) = \cos x \cdot \cos y - \sin x \cdot \sin y \quad (12.1)$$

$$\sin(x+y) = \sin x \cdot \cos y + \cos x \cdot \sin y \quad (12.2)$$

$$\cos(x-y) = \cos x \cdot \cos y + \sin x \cdot \sin y \quad (12.3)$$

$$\sin(x-y) = \sin x \cdot \cos y - \cos x \cdot \sin y \quad (12.4)$$

Thus, the iterative modulation of a signal (such as, for example, the cosine signal of Figure 12-2) may be algebraically represented as follows:

Stage 1: Determine $\cos(A_n + B_n)$ using equation (12.1) above with $x = A_n$, $y = B_n$:

$$\cos(A_n + B_n) = \cos A_n \cos B_n - \sin A_n \sin B_n \quad (12.5)$$

or

$$\cos(A_n + B_n) = \cos B_n \cos A_n - \sin B_n \sin A_n$$

This expression may be rewritten as:

$$X_{A+B}^C = 2 \cos B_n X_n^C - X_{n-1}^C \quad [D 1]$$

where:

$$X_{A+B}^C = \cos (A_n + B_n)$$

$$X_n^C = \cos A_n$$

$$X_{n-1}^C = \cos A_{n-1}$$

Stage 2: Determine $\sin (A_n + B_n)$ using equation (12.2) above with $x = A_n, y = B_n$:

$$\sin (A_n + B_n) = 2 \sin A_n \cos B_n - \sin (A_n - B_n) \quad (12.6)$$

This expression may be rewritten as:

$$X_{A+B}^S = 2 \cos B_n X_n^S - X_{n-1}^S \quad [D 2]$$

where:

$$X_{A+B}^S = \sin (A_n + B_n)$$

$$X_n^S = \sin A_n$$

$$X_{n-1}^S = \sin A_{n-1}$$

Stage 3: Determine $\cos (A_n + B_n + \delta)$ using equation (12.3) above, with $x = A_n + B_n, y = \delta$:

$$\cos (A_n + B_n + \delta) = \cos (A_n + B_n) \cos \delta - \sin (A_n + B_n) \sin \delta \quad (12.7)$$

This expression may be rewritten as:

$$\cos A_{n+1} = \cos (A_n + B_n) \cos \delta - \sin (A_n + B_n) \sin \delta$$

-or-

$$X_{n+1}^C = V_\delta^C X_{A+B}^C - V_\delta^C X_{A+B}^S \quad [C 1]$$

where:

$$X_{n+1}^C = \cos A_{n+1}$$

$$V_\delta^C = \cos \delta$$

$$V_\delta^S = \sin \delta$$

Stage 4: Determine $\sin (A_n + B_n + \delta)$ using equation (12.4) above, with $x = (A_n + B_n), y = \delta$.

$$\sin (A_n + B_n + \delta) = \sin (A_n + B_n) \cos \delta + \cos (A_n + B_n) \sin \delta \quad (12.8)$$

This expression may be rewritten as:

$$\sin A_{n+1} = \sin (A_n + B_n) \cos \delta + \cos (A_n + B_n) \sin \delta$$

-or-

$$X_{n+1}^S = V_{\delta}^C X_{A+B}^S + V_{\delta}^S X_{A+B}^C \quad [C 2]$$

where:

$$X_{n+1}^S = \sin A_{n+1}$$

Stage 5: Determine $\cos (B_n + \delta)$ using equation (12.1) above, with $x = B_n$, $y = \delta$.

$$\cos (B_n + \delta) = 2 \cos B_n \cos \delta - \cos (B_n - \delta) \quad (12.9)$$

This expression may be rewritten as:

$$\cos B_{n+1} = 2V_{\delta}^C \cos B_n - \cos B_{n-1}$$

or

$$X_{n+1}^B = 2 \cos \delta X_n^B - X_{n-1}^B \quad [D 3]$$

where:

$$X_n^B = \cos B_n$$

$$X_{n+1}^B = \cos B_{n+1}$$

$$X_{n-1}^B = \cos B_{n-1}$$

Stage 6: For the next iteration replace

Second Iteration Value	First Iteration Value
$\cos (A_2 - B_2)$	$\cos A_1$
$\cos A_2$	$\cos (A_1 + B_1 + \delta)$
$\cos B_2$	$\cos (B_1 + \delta)$
$\sin(A_2 - B_2)$	$\sin A_1$
$\sin A_2$	$\sin (A_1 + B_1 + \delta)$

Inspection of Figure 12-2 reveals that this is precisely the relationship depicted. In second iteration 16 the value $A_2 - B_2$ equals the value of A_1 of first iteration 14. Further, in second iteration 16, the value A_2 is equal to the value $(A_1 + B_1 + \delta)$. Still further, the value B_2 is equal to $B_1 + \delta$. One will observe that, at the start of first iteration 14, all angle differences are equal to phasor angle B_1 , and at the end of first iteration 14, all angle differences are the value $B_2 = (B_1 + \delta)$. The iteration process can be repeated through the six stages related above for as long as desired. However, a useful apparatus must be able to change the rate of modulation (value δ). The inventors have termed the points at which the rate of change of modulation is changed, "corner points".

Corner points may be inserted at any time and may, in the extreme, be inserted on a sample-by-sample (i.e., iteration-by-iteration) basis. Frequent insertion of corner points will enable quite close approximation of almost any sinusoidal function by a linearly modulated function (i.e., constant δ). A simple representation of corner points is a continuous piecewise linear function. That is, the start of a given linear element of the function occurs at the same point as the end of a preceding linear element. A non-continuous piecewise linear function would not have a common point shared by adjacent linear elements.

For simplicity, regarding the piecewise continuous linear function corner point case, assume that at a corner point δ changes from δ to δ_1 . When δ is changed to δ_1 , stages 1 through 4 of the iteration process above may proceed unchanged. At stage 5, however, one knows the values of $\cos B$ and $\cos (B - \delta)$ from the previous iteration. However, in order to apply equation (12.1) to perform stage 5 using $y = \delta_1$, one requires knowledge of $\cos B$ and $\cos (B - \delta_1)$. Thus, a corner point process must be inserted between stages 4 and 5.

It is worthy of note that we have assumed in this discussion that $\cos \delta$ and $\sin \delta$ are known. Because, in general, δ is updated infrequently, $\cos \delta$ and $\sin \delta$ can be determined by any suitable process (e.g., table look-up or sinusoidal generator) without having a great effect upon the overall system complexity.

Regarding the corner point process intermediate stages 4 and 5, $\cos (B - \delta_1)$ is determined using equa-

tion (12.3) with $x = (B-\delta)$, $y = (\delta-\delta_1)$. Thus, one must know the value of $\cos (\delta-\delta_1)$ and $\sin (\delta-\delta_1)$.

Since these values are required only at corner points, and therefore relatively infrequently, they can be determined using the same process used to determine $\cos \delta$ and $\sin \delta$. That is, they can be determined by a table look-up, or by a cosine generator, or by other known methods or processes. Further, one also requires knowledge of $\sin (B-\delta)$.

$\cos (B-\delta)$ is known because of its use in the iteration at stage 1.

A well-known equivalent relation in mathematics is that $\cos^2 x + \sin^2 x = 1$. This may be rewritten as $\sin^2 x = 1 - \cos^2 x$, or:

$$\sin x = \pm \sqrt{1 - \cos^2 x} \tag{12.10}$$

Thus, one can determine $\sin (B-\delta)$ given $\cos (B-\delta)$. The sign of equation (12.10) is always positive because the range of the value of B is preferably chosen to ensure such a result. It is possible, however, to determine the sign of equation (12.10) using the below-described procedure. Such a procedure may be required in certain circumstances. Further, the below-described procedure is required in later steps for computing the sign of the square root when computing $\sin (A_n - B_n)$ from $\cos (A_n - B_n)$.

In order to determine the correct sign, one must employ a well-known relationship:

$$\frac{d}{dx} (\cos x) = -\sin x$$

Thus, if

$$\frac{d}{dx} (\cos x)$$

is positive, then $\sin x$ is negative and if

$$\frac{d}{dx} (\cos x)$$

is negative, then $\sin x$ is positive.

$\cos B$ and $\cos (B-\delta)$ are known from the above-described process. However, this information is not sufficient to determine whether the slope of $\cos x$ at $x = B$ is positive or negative.

We can, however, use equation (12.1), with $x = B$, $y = \delta$ to determine $\cos (B+\delta)$. Such knowledge of $\cos (B-\delta)$, $\cos B$ and $\cos (B+\delta)$ suffices to determine the slope of $\cos x$ at $x = B$.

There are six possible cases; Figure 12-3a through Figure 12-3f are graphic representations of the six cases relating to sample displacement value for a sinusoidal signal. In Figure 12-3a, the function $y = \cos x$ passes a minimum point at y_{\min} . $\cos (B+\delta)$ is greater than $\cos B$, and $\cos B$ is greater than $\cos (B-\delta)$. Thus, the slope of $\cos x$ at $x = B$ is positive and $\cos (B+\delta) - \cos (B-\delta) > 0$.

In Figure 12-3b, $y = \cos x$ passes a minimum point at y_{\min} . $\cos (B+\delta)$ is greater than $\cos B$, $\cos (B+\delta)$ is greater than $\cos (B-\delta)$, and $\cos (B-\delta)$ is greater than $\cos B$. Thus, the slope of $\cos x$ at $x = B$ is positive and $\cos (B+\delta) - \cos (B-\delta) > 0$.

In Figure 12-3c, $y = \cos x$ passes a minimum point with y_{\min} . $\cos (B+\delta)$ is greater than $\cos B$; $\cos (B-\delta)$ is greater than $\cos B$; and $\cos (B+\delta)$ is less than $\cos (B-\delta)$. Thus the slope of $\cos x$ at $x = B$ is negative and $\cos (B+\delta) - \cos (B-\delta) < 0$.

In Figure 12-3d, $y = \cos x$ passes a maximum point at y_{\max} . $\cos (B+\delta)$ is less than $\cos B$ which, in turn, is less than $\cos (B-\delta)$. Thus, the slope of $\cos x$ at $x = B$ is negative and $\cos (B+\delta) - \cos (B-\delta) < 0$.

In Figure 12-3e, $y = \cos x$ passes a maximum point at y_{\max} . $\cos (B+\delta)$ is less than $\cos B$; $\cos (B+\delta)$ is greater than $\cos (B-\delta)$; and $\cos (B-\delta)$ is less than $\cos B$. Thus, the slope of $\cos x$ at $x = B$ is positive and $\cos (B+\delta) - \cos (B-\delta) > 0$.

In Figure 12-3f, $y = \cos x$ passes a maximum point at y_{\max} . $\cos (B+\delta)$ is less than $\cos B$ and $\cos (B+\delta) - \cos (B-\delta) < 0$.

The correlation of slope relationship to $\cos (B+\delta) - \cos (B-\delta)$ is summarized for Figure 12-3a through Figure 12-3f in the following table:

Figure 10-3	$\cos (B + \delta) - \cos (B - \delta)$	Slope
(a)	+	+
(b)	+	+
(c)	-	-
(d)	-	-
(e)	+	+
(f)	-	-

Thus, the sign of the slope of $\cos x$ at $x = B$, and consequently the inverse of the sign of $\sin x$ at $x = B$, is determined by the sign of the quantity $[\cos (x+\delta)-\cos (x-\delta)]$ at $x = B$. The above conclusion is based upon well-known properties of the function $f(x) = \cos x$ and, in particular, knowledge of the shape of that function and the fact that the function is continuous and symmetrical about points with a slope of $f(x) = 0$.

Given infinite mathematical precision and accuracy, equation (12.5) will accurately generate $\cos (A_n + B_n)$ as n approaches infinity. However, such is not the case for a finite accuracy mathematical system (such as a microprocessor-based system). Inaccuracies born of imprecision caused by less than infinite accuracy feed back through the calculations and eventually cause divergence from the desired true result. The rate of divergence and the speed of onset of significant errors increase as the number of significant digits in the arithmetic representation decreases. A correction mechanism is required in order to achieve satisfactory performance of a finite system employing the iterative process described for generating a sinusoidal signal.

Such a correction mechanism recognizes the limitations imposed upon a calculation of the iterative generation of a sinusoidal wave as described above by the finite capabilities of a system such as a microprocessor-based system. That is, the employment of difference equations employing actual values for

sinusoidal terms such as cosine and sine, as opposed to deriving the values of cosine and sine from angle values, inherently injects error through each iterative calculation.

There are two correction mechanisms which are useful for the present invention: modular retracking and resanitization.

Modular retracking may be performed periodically at any time, but is preferably performed at the end of a sample analysis frame or period. Modular retracking recognizes that iterative treatment of a sinusoidal relationship to accommodate a change of parameter (such as frequency) from a first parametric value to a second parametric value over a predetermined time period may not precisely result in the particular changed parameter having the desired second parametric value at the intended time. As previously mentioned, such inaccuracy can especially result in a finite mathematical system. The preferred method for accommodating (and correcting) such inaccuracies in the present invention is to determine the step value for effecting parametric value change over a second (succeeding) time period (i.e., analysis frame) using the actual parametric value at the end of the first time period as the initial parametric value for the second time period. That is, the actual initial parametric value is used rather than using the target (i.e., expected, or theoretical) parametric value in determining the appropriate step value for effecting the desired parametric value change during the second time period. Thus, the step value required for parametric transition in the second time period will reflect real-time conditions. "Blindly" using theoretical values will almost certainly result in unacceptable error in a relatively few analysis frames (time periods).

"Resanitization" is a term coined by the inventors to indicate the process of restoring the "sanity" of the presumption that sinusoidal values are accurate in the difference equations (D1, D2, D3) and combination equations (C1, C2) employed in the preferred embodiment of the present invention.

An example of a resanitization process in connection with the present invention is as follows:

Step 1. Assume the truth of $V_{\delta}^C = \cos \delta$; known $X_n^B = \cos B_n$; known

Step 2. Determine V_{δ}^S :

$$\sin \delta = \pm \sqrt{1 - \cos^2 \delta}$$

$$V_{\delta}^S = \pm \sqrt{1 - (V_{\delta}^C)^2}$$

Since δ is known, the above calculation may be made and the sign assigned by the known value of δ .

In the alternative, $\sin \delta$ and $\cos \delta$ may be supplied by other known means, such as by a look-up table or a sine/cosine generator.

Step 3. Since X_n^B is a term for a difference equation which is an actual value representing $\cos B_n$, the value of X_n^B can be >1 or <-1 . Of course, such a value can never exist for $\cos B_n$. Therefore, we must reset X_n^B as follows:

$$\text{If } X_n^B > 1 \text{ set } X_n^B = 1$$

$$\text{If } X_n^B < -1 \text{ set } X_n^B = 1$$

Step 4. Determine $\sin B_n$:

$$\sin B_n = \pm \sqrt{1 - \cos^2 B_n}$$

In general, the sign of the above expression should always be positive since B_n will preferably lie in the first or second quadrant. Values of B_n in the third or fourth quadrant imply either negative frequency or aliasing. Neither of those conditions is desirable in most applications. The sign may be determined by determining the sign of $[\cos(B_n + \delta) - \cos(B_n - \delta)]$; that is, $[X_{n+1}^B - X_{n-1}^B]$, as described above in connection with Figure 12-3a.

Step 5. Using equation (10.3) $[\cos(x+y) = \cos x \cdot \cos y - \sin x \cdot \sin y]$, with $x = B_n$, $y = -\delta$:

$$\cos(B - \delta) = X_{n-1}^B = \cos B_n \cos(-\delta) - \sin B_n \sin(-\delta)$$

It is well-known that:

$$\cos(-\delta) = \cos \delta$$

$$\sin(-\delta) = -\sin \delta$$

Thus, the above expression may be written as:

$$X_{n-1}^B = \cos B_n \cos \delta + \sin B_n \sin \delta$$

$\cos B_n$ is known (Step 1)

$\cos \delta$ is known (Step 1)

$\sin \delta$ is known (Step 2)

$\sin B_n$ is known (Step 3)

X_{n-1}^B is determined.

Step 6. Using equation (12.3), with $x = A_n$, $y = -B_n$:

$$\cos (A_n - B_n) = X_{n-1}^C$$

$$X_{n-1}^C = \cos A_n \cos (-B_n) - \sin A_n \sin (-B_n)$$

Which may be rewritten as:

$$X_{n-1}^C = \cos A_n \cos B_n + \sin A_n \sin B_n$$

$\cos B_n$ is known (Step 1)

$\sin B_n$ is known (Step 3)

$\cos A_n$ is assumed as part of the resetting which is effected by the resanitization process.

$\sin A_n$ may be determined:

$$\sin A_n = \pm \sqrt{1 - \cos^2 A_n}$$

The sign is determined by determining the sign of $(\cos (A_n + B_n) - \cos (A_n - B_n))$; that is, $(X_{A+B}^C - X_{A-B}^C)$. One may note that this expression is the same relationship as recited in connection with Step 4 above, with A_n substituted for B_n , and with B_n substituted for δ . Thus, X_{n-1}^C is determined.

Step 7. Using equation (12.4), with $x = A_n$, $y = -B_n$:

$$\cos (A_n - B_n) = X_{n-1}^S$$

$$X_{n-1}^S = \sin A_n \cos (-B_n) + \cos A_n \sin (-B_n)$$

which may be rewritten as:

$$X_{n-1}^S = \sin A_n \cos B_n - \cos A_n \sin B_n$$

$\cos B_n$ is known (Step 1)

$\sin B_n$ is known (Step 3)

$\cos A_n$ is assumed as part of the resetting which is effected by the resanitization process.

$\sin A_n$ is known (Step 6)

Thus, X_{n-1}^S is determined.

In summary, by resetting (i.e., assuming the truth, or sanity, of) $\cos \delta (V_{\delta}^C)$, $\cos B_n (X_n^B)$, and $\cos A_n (X_n^C)$, all other values in difference equations (D1, D2, D3) and combination equations (C1, C2) may be reset.

The above is an example of a resanitization process. Other assumptions and variations may be used to accomplish the same goal: to restore the "sanity" of presumptions in the difference equations (D1, D2, D3) and combination equations (C1, C2) to preclude divergence of the relationships because of inaccuracies born of finite mathematical calculations.

Clearly the calculations required in the resanitization process and in the modular retracking process are relatively time consuming compared with the time required to calculate the various difference equations (D1, D2, D3) and combination equations (C1, C2). Accordingly, it is preferable that the modular retracking and resanitization processes be effected relatively infrequently as compared with the frequency at which the difference equations (D1, D2, D3) and combination equations (C1, C2) are exercised.

Figure 12-4 is a graphic representation of a voice signal with its fundamental sinusoidal frequency. In Figure 12-4, a voice signal 20 is illustrated; the fundamental sinusoidal frequency 22 (pitch component) related with voice signal 20 is also illustrated. Voice signal 20 is sampled during a plurality of sample intervals, represented by sample interval 24. Preferably, analysis of signals 20, 22 occurs during succeeding analysis intervals, represented by analysis intervals 26, 28, 30. The period T of signals 20, 22 may

change, as indicated by the period of signals 20, 22 in analysis interval 26 being a period T_n , and the period of signal 20, 22 in analysis interval 28 being a period T_{n-1} .

A representative sample interval for a voice-band signal could be 125 microseconds. A representative analysis interval for such a voice-band signal may, for example, involve 160 samples per analysis frame for a duration of each analysis frame equal to 20 milliseconds. Of course, the present invention is compatible with any signal which may be represented by a fundamental sinusoidal signal or approximated by a sinusoidal signal. The present invention is not limited to voice-band signals. The voice-band signal illustrated in Figure 12-4 is included here for illustration purposes only.

Figure 12-4 is a schematic diagram of a signal processing system appropriate for processing a signal such as the voice-band signal illustrated in Figure 12-4, using the present invention.

In Figure 12-5, a signal processing system 40 includes an encoder 42, a decoder 44, and a storage unit 46. Encoder 42 includes an analog-to-digital conversion unit 48 which receives an analog signal (such as, for example, signals 20 and 22) at an analog input 50. Analog-to-digital conversion unit 48 converts the analog signal received at analog input 50 to a digital representation of that received signal and passes that digital representation to a signal analysis unit 52. Signal analysis unit 52 may be employed to determine particular aspects of the signal received at analog input 50, such as various parameters (e.g., frequencies) and parametric changes from analysis frame to analysis frame. The results of signal analysis performed by signal analysis unit 52 are stored in storage unit 46 and are passed to a register 54 in decoder unit 44. Data may be stored in storage unit 46 and in register 54, for example, to save the signal analysis output from signal analysis unit 52 relating to respective analysis frames. In the alternative, storage unit 46 may be eliminated, and appropriate data may be stored in register 54 alone. Information relating to the analyzed signal information from signal analysis unit 52 relating to earlier analysis frames than the currently extant analysis frame may be provided from storage unit 46 to register 54 in an appropriate portion of register 54 for each respective data frame.

A parametric determining logic unit 56 receives information relating to at least two succeeding data frames in order to determine appropriate parameters (such as δ , B_n , and the like; representative are parameters discussed above in connection with the present invention) for supply to a basic signal synthesis unit 58. In this representative apparatus 40 employing the present invention, the present invention resides in and generally comprises basic signal synthesis unit 58. Information from register 54 and basic signal synthesis unit 58 is provided to a filter unit 60. Filter unit 60 may take the output from basic signal synthesis unit 58 which will provide a sinusoidal signal, such as signal 22 in Figure 12-4 (constituting, for example, a pitch component for a voice reproduction signal). Filter unit 60 may impose voice tract or other information upon the pitch component received from basic signal synthesis unit 58 to produce an analog output signal at analog output 62 which substantially faithfully reproduces the analog signal received at analog input 50.

Figure 12-6 is a schematic diagram of an embodiment of the present invention. In Figure 12-6, an apparatus 70 includes a next step value iterating unit 72, a next parameter value iterating unit 74, and a derivative next parameter iterating unit 76.

A logic unit 80 in next step value iterating unit 72 receives an input value $\cos \delta$ and initial input values $\cos B_n$ and $\cos (B_n - \delta)$ to generate, according to equation (12.9), the value $\cos (B_n + \delta)$ at an output 82. A delay filter unit 84 imposes a delay on the signal appearing at output 82 so that a signal $\cos B_n$ is produced on line 86. Line 86 is fed back through a delay filter unit 88 to provide a feedback input to logic unit 80 having the values $\cos B_n$ and $\cos (B_n - \delta)$. Thus, next step value iterating unit 72 iteratively generates step values $\cos B_n$ on line 86.

The various delay filter units described in connection with Figure 12-6 are clocked at a predetermined time. Theoretically, they are all clocked at once, but as a practical matter they may be just-in-time cascade clocked. However they may be clocked, the result is that the input to a given delay filter unit is shifted to the output of that delay filter unit at the time the particular delay filter unit is clocked. It is by this clock-

ing arrangement that iteration is effected by apparatus 70.

Step value $\cos B_n$ on line 86 is applied to a logic unit 90 in next parameter value iterating unit 74. Logic unit 90 receives initial values representing $\cos (A_n - B_n)$ and $\cos A_n$ and generates an interim parameter value $\cos (A_n + B_n)$, preferably according to equation (12.5).

Step value $\cos B_n$ is also supplied from line 86 to a logic unit 92 in derivative next parameter iterating unit 76. Logic unit 92 receives initial values representing $\sin (A_n - B_n)$ and $\sin A_n$ and generates a derivative interim parameter value $\sin (A_n + B_n)$, preferably according to equation (12.6). Both derivative interim parameter value $\sin (A_n + B_n)$ and interim parameter value $\cos (A_n + B_n)$ are applied to a logic unit 94 in next parameter value iterating unit 74. Logic unit 94 receives initial values representing $\cos \delta$ and $\sin \delta$ and generates a next parameter value $\cos (A_n + B_n + \delta)$ at an output 96, preferably according to equation (12.7). Next parameter value $\cos (A_n + B_n + \delta)$ is fed back by a line 98 to a delay filter unit 100, and thence to a delay filter unit 102, from which iterated values of $\cos (A_n - B_n)$ and $\cos A_n$ are provided to logic unit 90. Thus, next parameter value iterating unit 74 iteratively generates interim parameter value $\cos (A_n + B_n)$ and next parameter value $\cos (A_n + B_n + \delta)$.

Similarly, both derivative interim parameter value $\sin (A_n + B_n)$ and interim parameter value $\cos (A_n + B_n)$ are applied to a logic unit 104 in derivative next parameter iterating unit 76. Logic unit 104 receives initial values of $\cos \delta$ and $\sin \delta$, and generates a derivative next parameter value $\sin (A_n + B_n + \delta)$, preferably according to equation (12.8). Derivative next parameter value $\sin (A_n + B_n + \delta)$ is fed back via a line 106 to a delay filter unit 108, and thence to a delay filter unit 100, from which iterated values of $\sin (A_n - B_n)$ and $\sin A_n$ are provided to logic unit 92. Thus, derivative next parameter value iterating unit 76 iteratively generates derivative interim parameter value $\sin (A_n + B_n)$ and derivative next parameter value $\sin (A_n + B_n + \delta)$.

Figure 12-8 is a schematic diagram of the preferred embodiment of the present invention. In Figure 12-8, an apparatus 170 includes a next step value iterating unit 172, a next parameter value iterat-

ing unit 174, and a derivative next parameter iterating unit 176. Next step value iterating unit 172 includes a logic unit 180 which periodically receives input values V^C_δ (representing $\cos \delta$), X^B_n (representing $\cos B_n$), and X^B_{n-1} (representing $\cos (B_n - \delta)$). Initially, logic unit 180 also receives an initial value X^B_{n-1} (representing $\cos (B_n - \delta)$) and an initial value X^B_n (representing $\cos B_n$). Logic unit 180 calculates a value X^B_{n+1} (representing $\cos (B_n + \delta)$) according to difference equation D3:

$$X^B_{n+1} = 2V^C_\delta X^B_n - X^B_{n-1}$$

The value X^B_{n+1} is applied by a line 182 to a delay unit 184. Delay unit 184 outputs a value (representing $\cos B_n$) on a line 186. Line 186 provides the value X^B_n (representing $\cos B_n$) to logic unit 180 and to a delay unit 188; delay unit 188 provides a value X^B_{n-1} (representing $\cos (B_n - \delta)$) as an input to logic unit 180 so that next step value iterating unit 172 may iteratively generate value X^B_n on line 186.

Value X^B_n is applied as an input to a logic unit 190 in next parameter value iterating unit 174. Logic unit 190 receives initial values X^C_{n-1} (representing $\cos (A_n - B_n)$), and X^C_n (representing $\cos A_n$). Logic unit 190 also periodically receives values X^C_n (representing $\cos A_n$) and X^C_{n-1} (representing $\cos (A_n - B_n)$). Logic unit 190 generates an interim parameter value output X^C_{A+B} (representing $\cos (A_n + B_n)$) according to difference equation D1:

$$X^C_{A+B} = 2 \cos B_n X^C_n - X^C_{n-1}$$

Value X^B_n is also applied as an input to a logic unit 192 in derivative next parameter iterating unit 176. Logic unit 192 receives initial values X^S_{n-1} (representing $\sin (A_n - B_n)$) and X^S_n (representing $\sin A_n$). Logic unit 192 also periodically receives values X^S_n (representing $\sin A_n$) and X^S_{n-1} (representing $\sin (A_n - B_n)$). Logic unit 192 generates a derivative interim parameter value X^S_{A+B} (representing $\sin (A_n + B_n)$) according to difference equation D2:

$$X^S_{A+B} = 2 \cos B_n X^S_n - X^S_{n-1}$$

Both the derivative interim parameter value X^S_{A+B} and the interim parameter value X^C_{A+B} are applied to a logic unit 194. Logic unit 194 receives inputs V^C_δ (representing $\cos \delta$) and V^S_δ (representing

sin δ). Logic unit 194 generates a next parameter value X_{n+1}^C (representing $\cos A_{n+1}$) at an output 196, preferably according to combination equation C1. The next parameter value X_{n+1}^C is fed back via a line 198 to a delay unit 200 and thence to a delay unit 202 to iteratively provide feedback values X_{n-1}^C (representing $\cos(A_n - B_n)$) and X_n^C (representing $\cos A_n$). Thus, next parameter value iterating unit iteratively generates interim parameter value X_{A+B}^C and next parameter value X_{n+1}^C .

Derivative interim parameter value X_{A+B}^S and interim parameter value X_{A+B}^C are also applied to a logic unit 204. Logic unit 204 periodically receives values V_δ^C (representing $\cos \delta$) and V_δ^S (representing $\sin \delta$). Logic unit 204 generates a derivative next parameter value X_{n+1}^S (representing $\sin A_{n+1}$) preferably according to combination equation C2. The derivative next parameter value X_{n+1}^S is fed back via a line 206 to a delay unit 208 and thence to a delay unit 210 to iteratively provide feedback values X_{n-1}^S (representing $\sin(A_n - B_n)$) and X_n^S (representing $\sin A_n$). Thus, derivative next parameter value iterating unit iteratively generates derivative interim parameter value X_{A+B}^S and derivative next parameter value X_{n+1}^S .

A parametric determining unit 300 provides periodic inputs at update inputs 302, 304, 306, 308, 310 involving periodic supply of finite values as previously described in connection with Figure 12-8.

References

Related Patents

<u>4905177</u>	Feb., 1990	Weave, Jr. et al.	364/721.
<u>5113361</u>	May., 1992	Damerow et al.	364/721.
<u>5187677</u>	Feb., 1993	Kovalick	364/721.
<u>5276633</u>	Jan., 1994	Fox et al.	364/721.
<u>5321642</u>	Jun., 1994	Goldberg	364/721.

Chapter 13

Apparatus/Method for Analyzing Speech Signals to Find Speech Signal Parameter Characteristics

Abstract

This chapter is based on the invention patented under American patent 5,339,251 (S. Asghar, M. Iretton, April 1998).

An apparatus and method for locating a plurality of roots of a line spectrum pair expression on a unit circle. The method comprises the steps of: (1) receiving an initial value for locating a first site on the unit circle; (2) receiving a step value for defining an arc-distance on the unit circle; (3) generating intervals on the unit circle, each having a lower limit and an upper limit; the lower limit of the initial interval is the initial value and the upper limit of the initial interval is displaced on the unit circle from the initial value by the arc-distance; each succeeding interval has its lower limit coincident with the upper limit of the next preceding interval and has its upper limit displaced on the unit circle from its lower limit by the arc-distance; (4) evaluating the expression for at least the upper limit and the lower limit of each respective interval; (5) recognizing a root of the expression when the expression changes sign within an interval; (6) designating each such interval as a solution interval; and (7) generating the lower and upper limits of each solution interval to identify where the root is located. The apparatus comprises a waveform generator which receives the initial value and the step value and defines the arc-distance, and generates the intervals; a zero detector for recognizing roots of the expression when the expression changes sign.

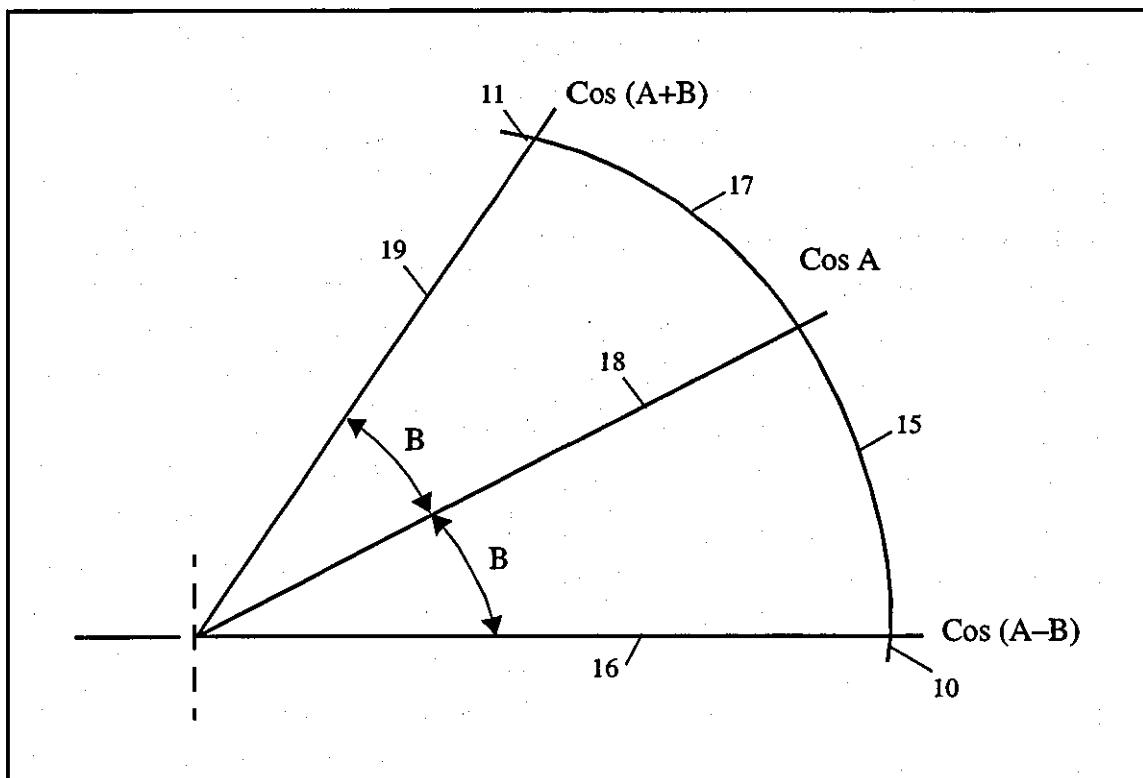


Figure 13-1

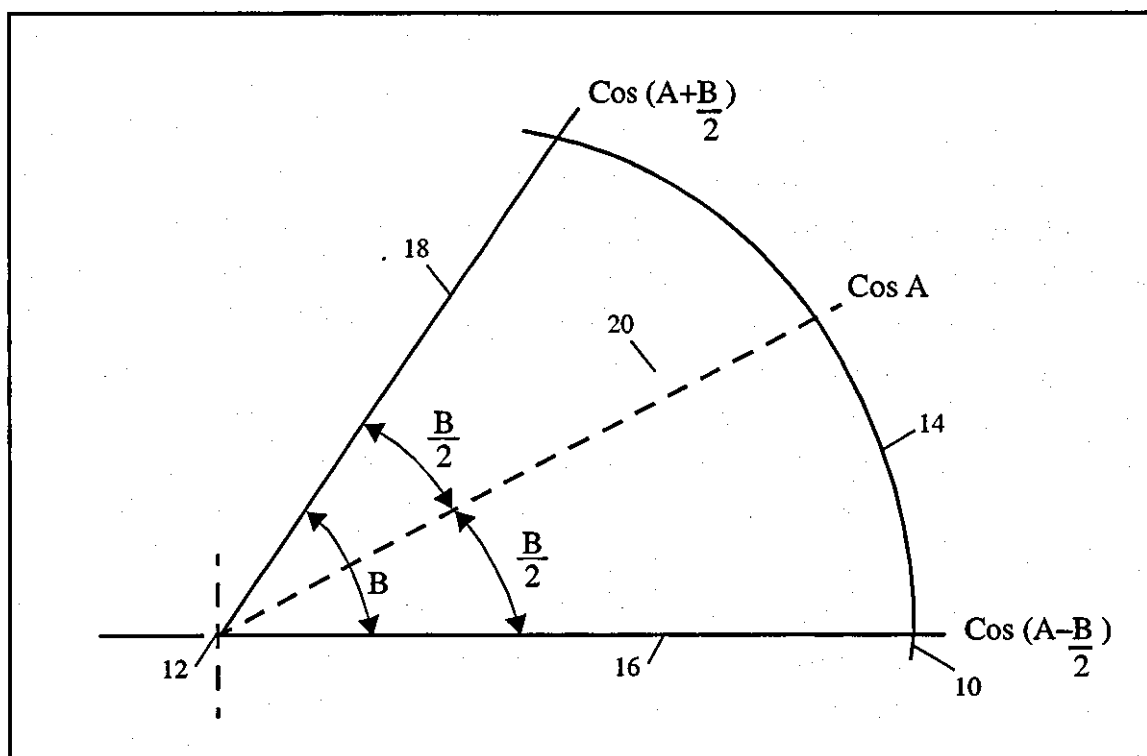
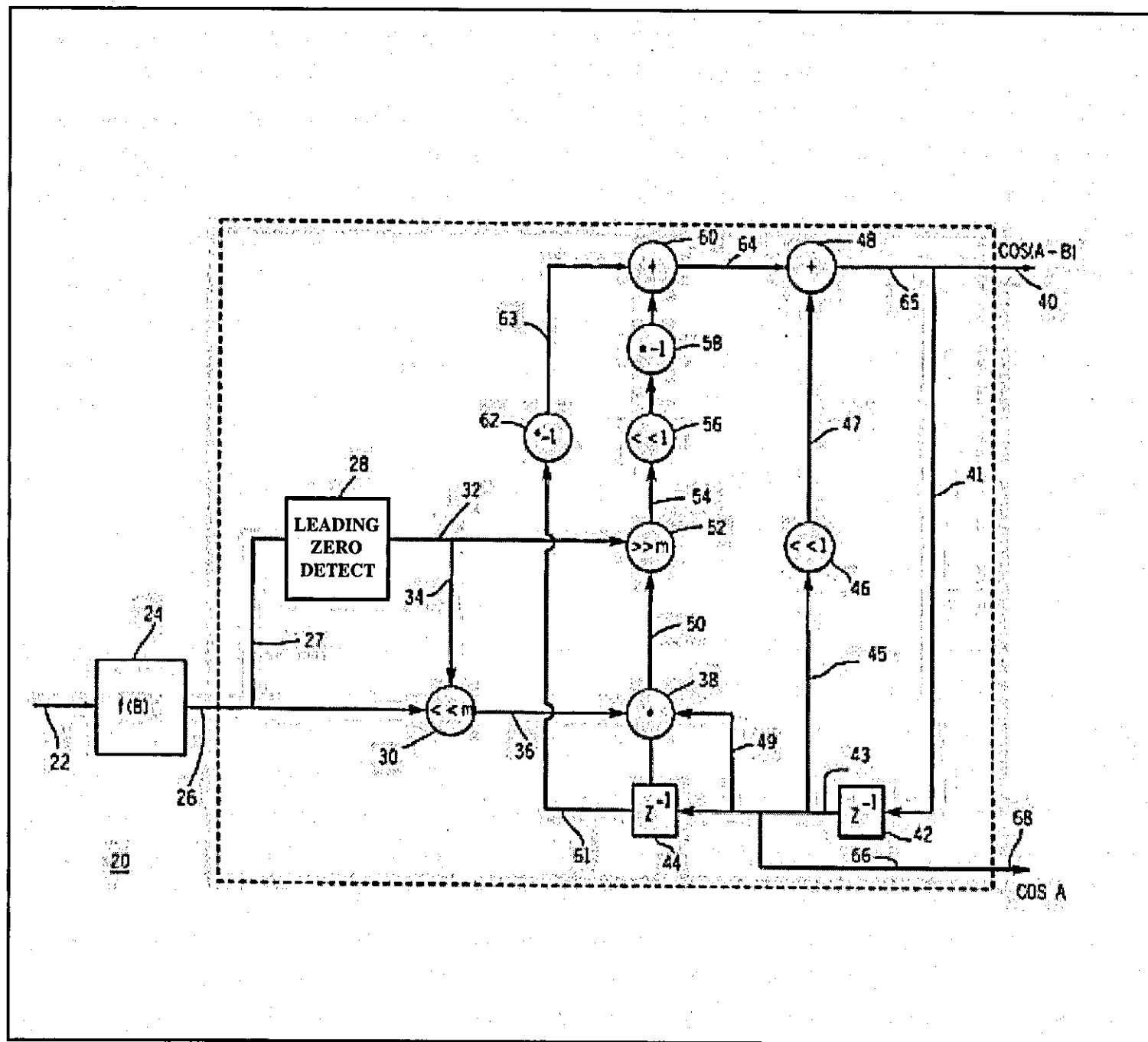


Figure 13-2

Figure 13-3



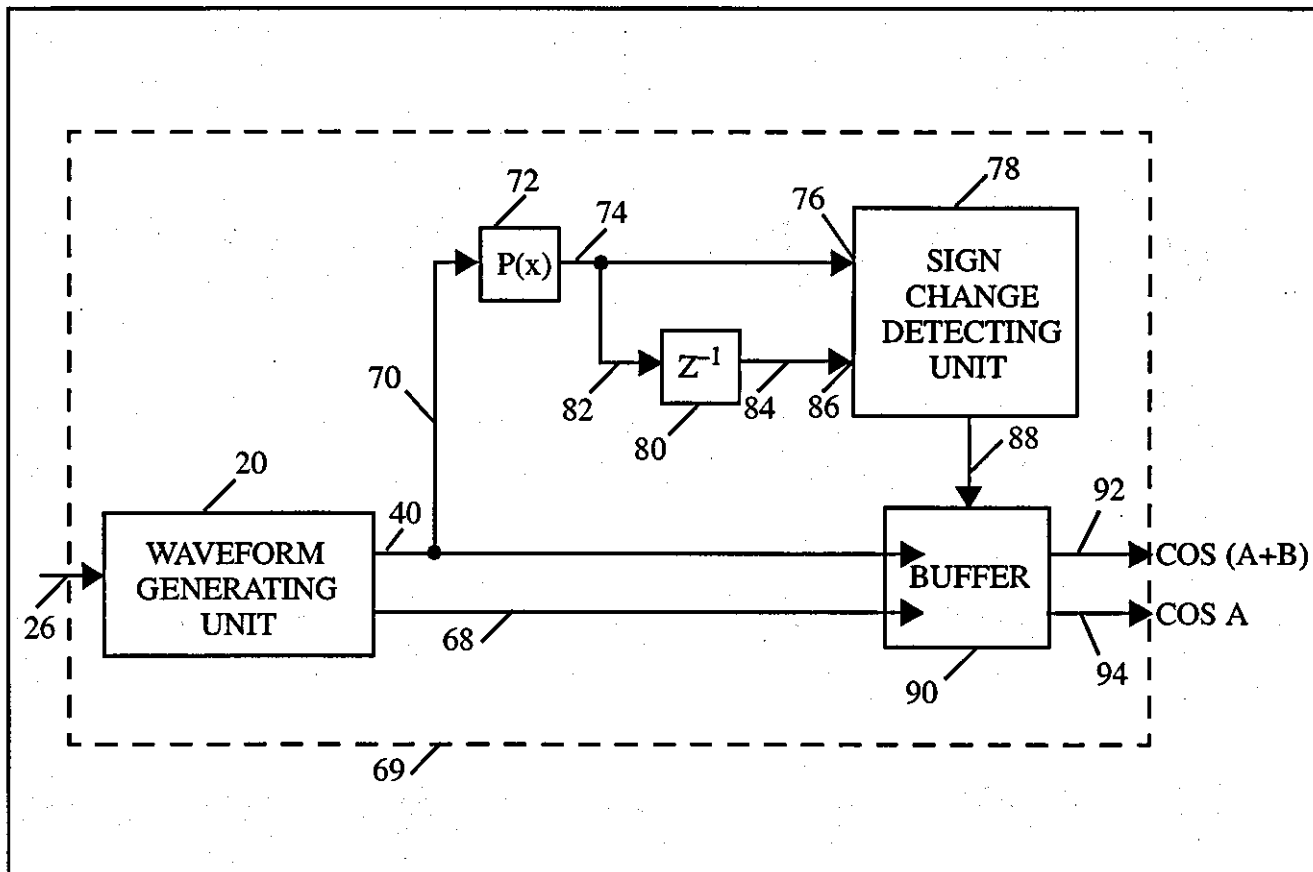


Figure 13-4

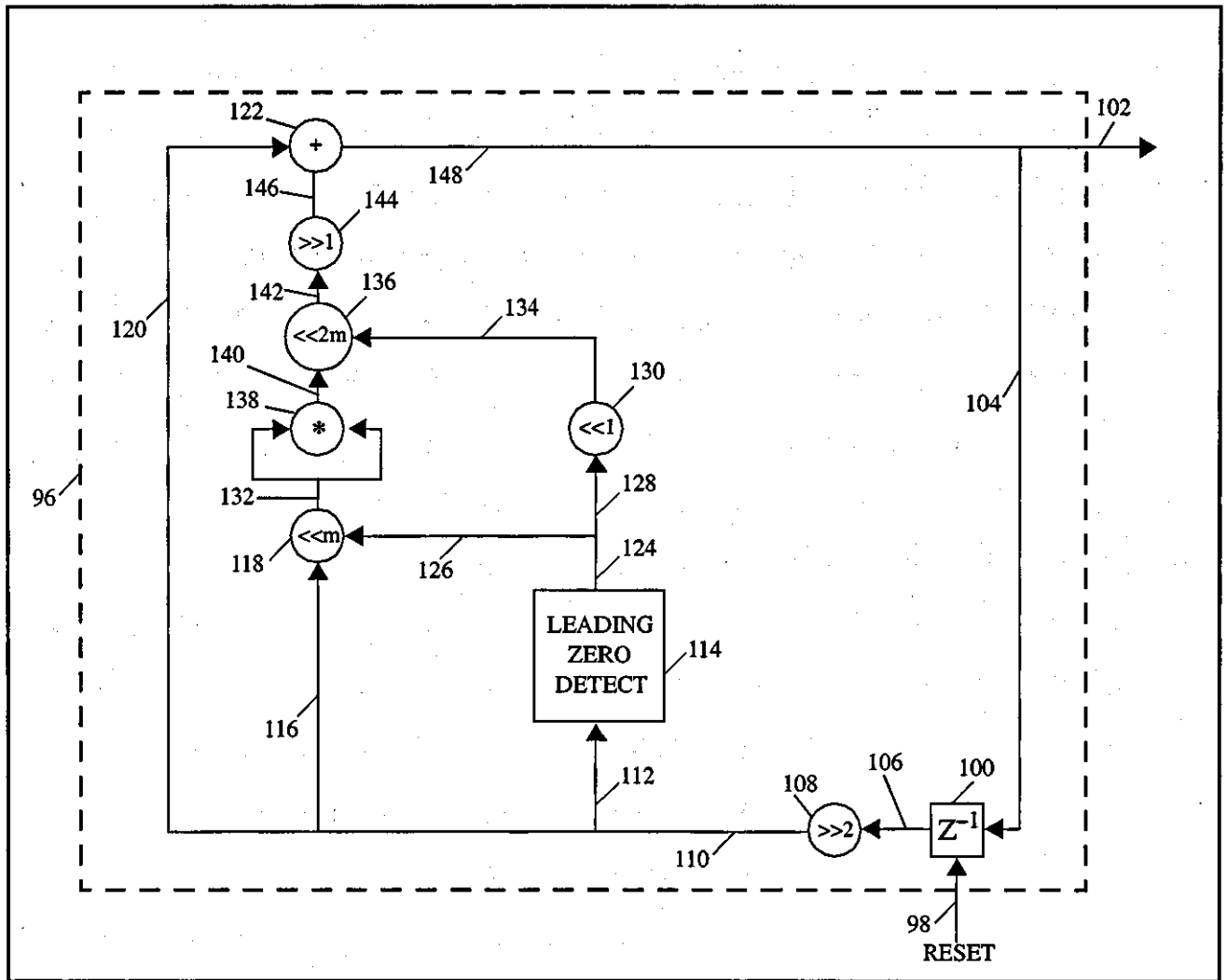


Figure 13-5

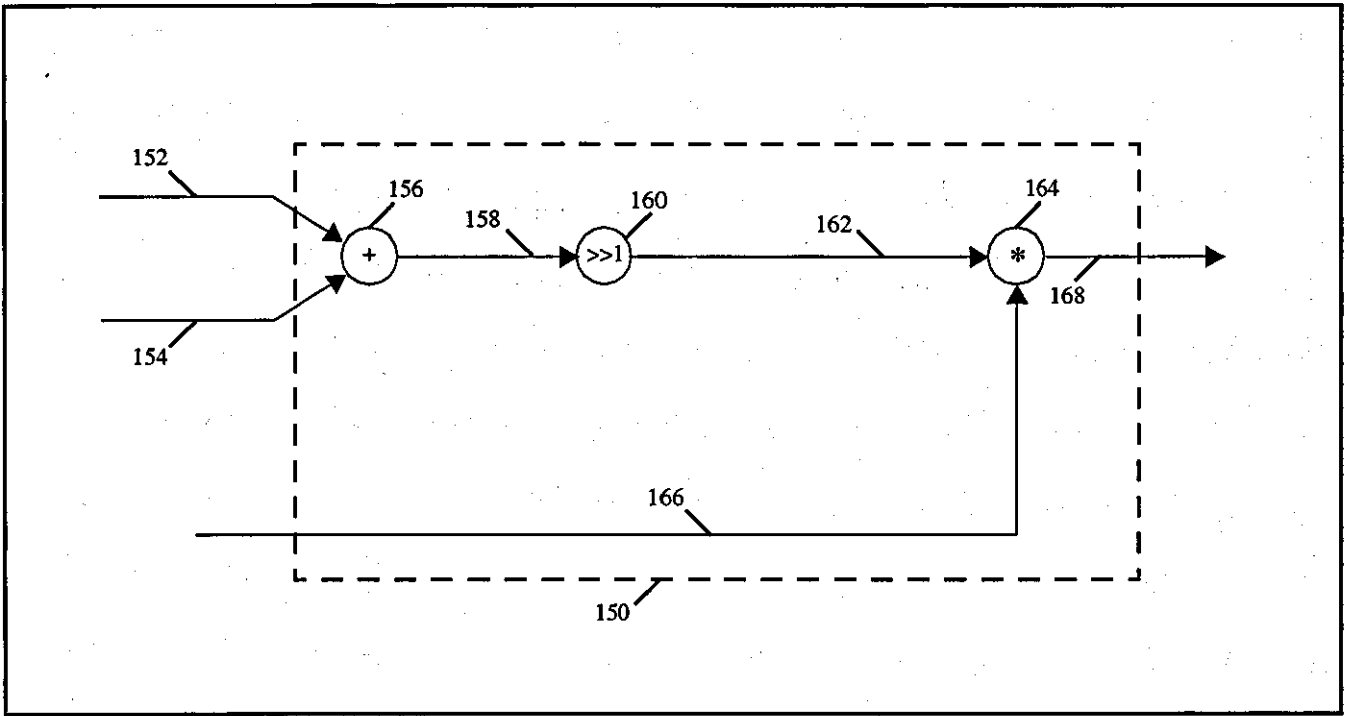


Figure 13-6

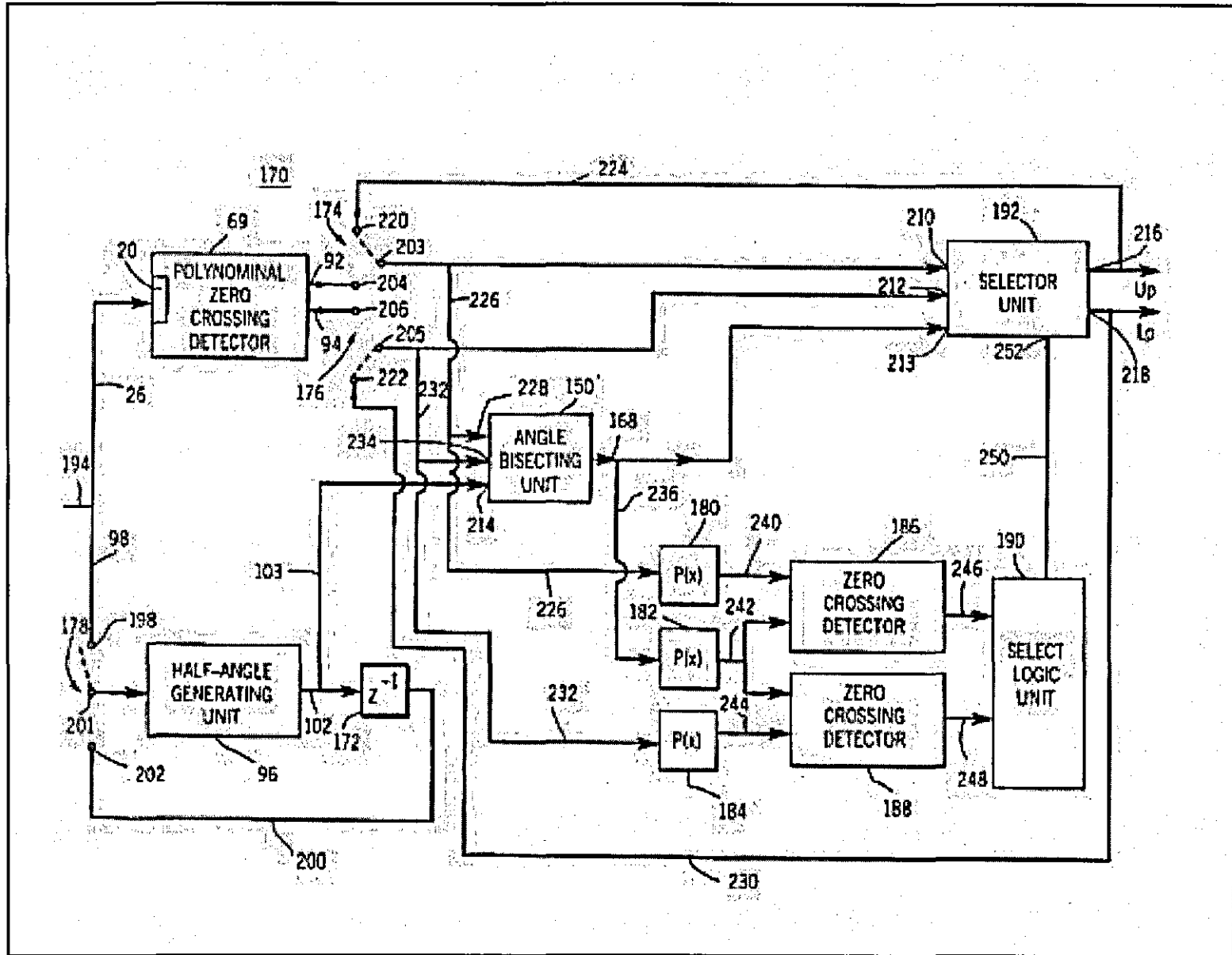
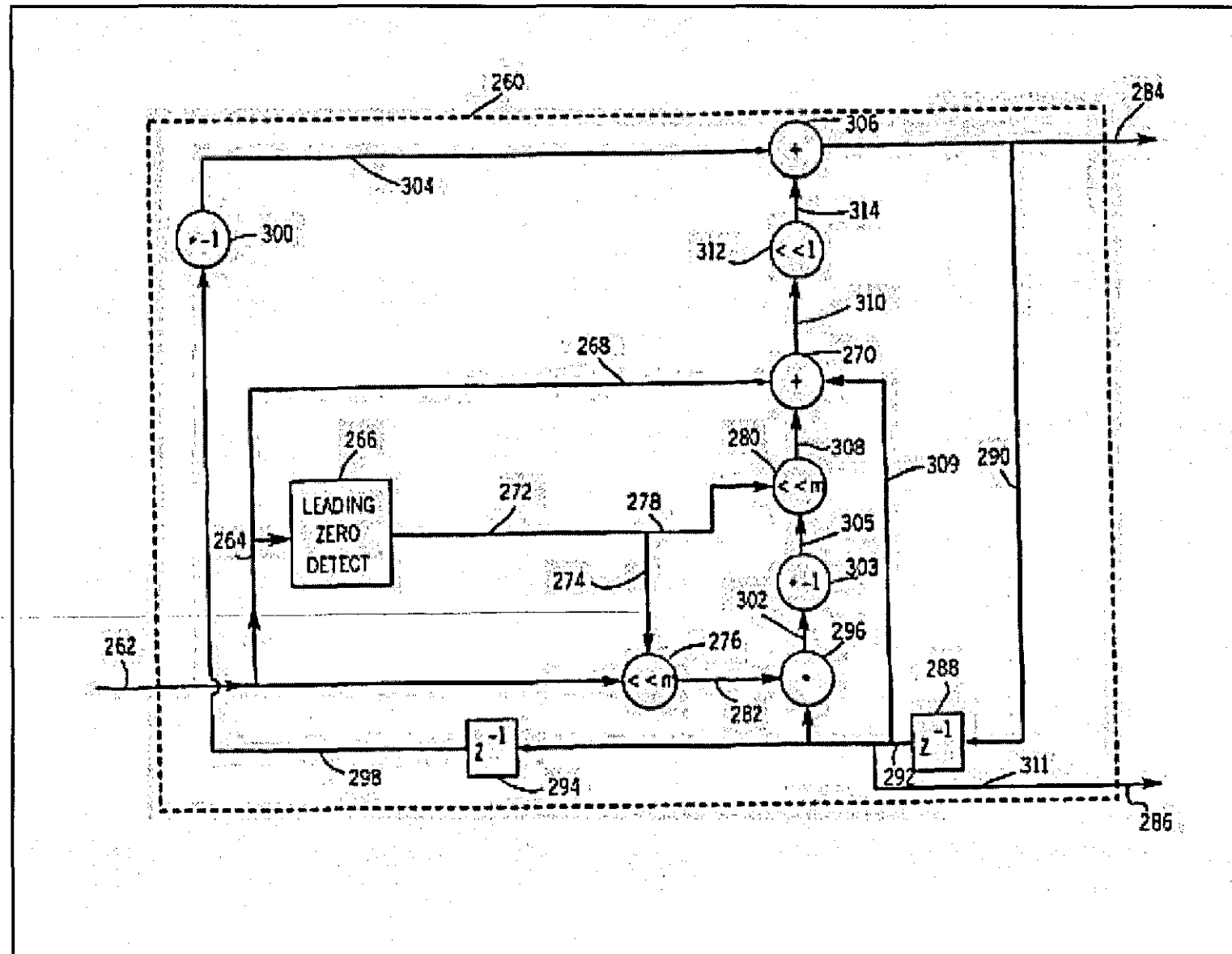


Figure 13-7

Figure 13-8



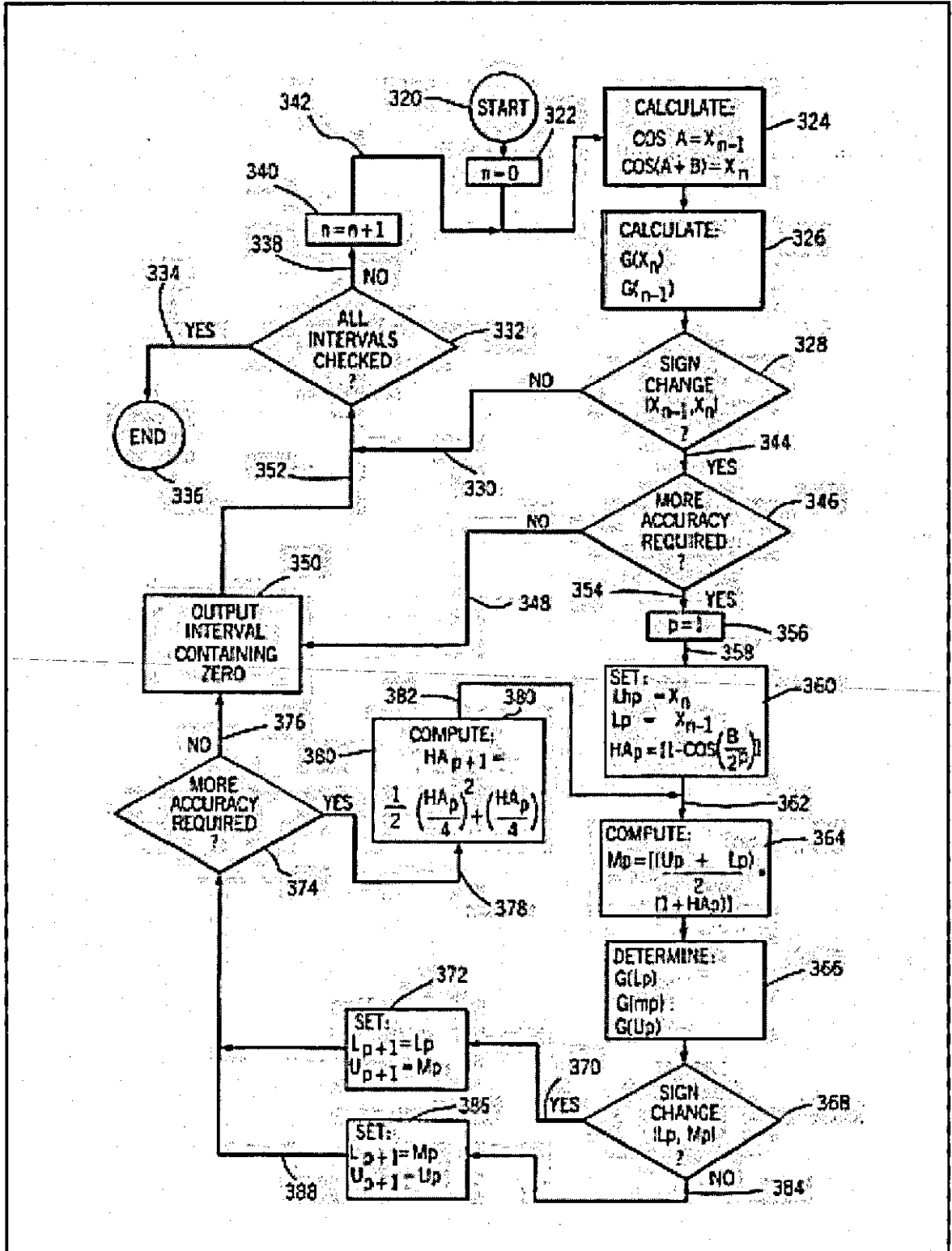


Figure 13-9

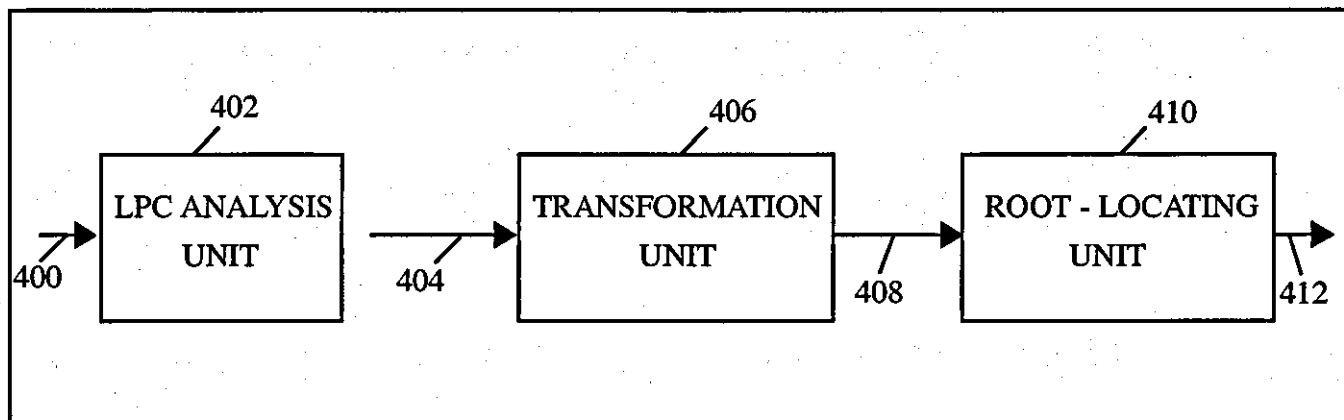


Figure 13-10

BACKGROUND OF THE INVENTION

Numerous methods have been developed for speech analysis and speech synthesis. The most successful methods thus far are concerned with speech spectrum conservation rather than speech waveform, such speech spectrum conservation is sufficient to insure adequate speech quality and intelligibility for many purposes. Generally, in a speech analysis and synthesis method, certain predetermined parameters corresponding to particular speech spectra are extracted at an analyzer end and, at the receiver end, the speech waveform is synthesized from those predetermined parameters. The difference among the various methods is the difference in the predetermined parameters.

Most research has been concentrated on finding predetermined parameters which express speech characteristics efficiently. One method, linear predictive coding (LPC), has been successful but has been found not as efficient as desired because of quantization characteristics associated with linear predictive coding.

Another idea for speech synthesis and analysis is based on an assumption that a speech signal can be approximately represented as an output signal from an all-pole filter. The PARCOR (PARTIAL autoCORre-

lation) lattice filter is employed to implement this assumption. The PARCOR speech analysis and synthesis method has been found to be efficient for narrow band speech coding. However, the PARCOR method has limitations with regard to data compression. In particular, PARCOR-synthesized speech quality rapidly deteriorates at bit rates lower than 4.8 kilobits per second. There are two main reasons for this: (1) in parameter quantization, between 4 and 8 bits are required for the PARCOR coefficients, and (2) the spectral distortion due to parameter interpolation increases rapidly as the parameter refreshing period is lengthened.

Another approach to speech synthesis and analysis is the line spectrum pair (LSP) method. The line spectrum pair method also employs an all-pole model of speech, and employs line spectrum pair parameters which are interpreted as one of the linear predictive coding parameters in the frequency domain.

Line spectrum pairs (LSP's) are a transformed representation of the canonical linear predictive coding (LPC) filter coefficients, which possess some useful characteristics. The LSP representation gives an accurate approximation to the short term spectrum using relatively few bits. The LSP representation also has the useful property of error localization. For example, an error in a particular coefficient will only introduce distortion in the frequency spectrum in frequencies close to the frequency represented by the coefficient. These parameters are also useful for vector quantization where good results can be obtained using a simple mean squared error measure on the coefficient vectors.

Any of the above-described methods (LPC, PARCOR, LSP) have heretofore been implemented in inefficient, cumbersome, memory-expensive ways. For example, LSP requires either a cosine computation algorithm or a large look-up table in which to store cosine values. The cosine computation is complex, and a look-up table requires a large amount of memory, especially if highly accurate results are required.

There is, therefore, a need for a method and apparatus for solving line spectrum pair expressions which is efficient in its storage requirements and efficient in effecting root determination.

SUMMARY OF THE INVENTION

The present invention is an apparatus and method for locating a plurality of roots of a line spectrum pair expression on a unit circle. In its preferred embodiment, the apparatus comprises a waveform generating unit which has a first input for receiving a representation of an initial value for locating a first site on the unit circle. The waveform generating unit also has a second input for receiving a representation of a step value for defining an arc-distance on the unit circle. The waveform generating unit generates a plurality of intervals on the unit circle, each interval having a lower limit and an upper limit. The plurality of intervals includes an initial interval and a plurality of succeeding intervals. The lower limit of the initial interval is the initial value, and the upper limit of the initial interval is displaced on the unit circle from the initial value by the arc-distance. Each respective succeeding interval has its respective lower limit coincident with the upper limit of the next-preceding interval and has its respective upper limit displaced on the unit circle from its respective lower limit by the arc-distance. The apparatus further includes a polynomial zero detecting unit coupled with the waveform generating unit for receiving a plurality of intervals and evaluating a line spectrum pair expression for at least the upper limit and the lower limit of each respective interval. The polynomial zero detecting unit recognizes the presence of a respective root of the line spectrum pair expression when the evaluated expression changes sign within a particular interval. The polynomial zero detecting unit designates each such particular interval as a solution interval and generates the lower limit and upper limit of each such solution interval.

For increased accuracy beyond the initial accuracy provided by the predetermined intervals, the apparatus may include a half-distance generating unit coupled with the waveform generating unit for generating a half-arc-distance which is substantially $1/2$ the arc-distance. Further included in this improved accuracy apparatus is an angle bisecting unit coupled with the half-distance generating unit and coupled with the polynomial zero detecting unit which receives the solution intervals from the polynomial zero

detecting unit and receives a half-arc-distance from the half-distance generating unit. The angle bisecting unit performs a bisecting operation defining a lower refined solution interval and an upper refined solution interval. The lower refined solution interval has a lower refined lower limit at the lower limit of the solution interval and has a lower refined upper limit spaced on the unit circle from the lower refined lower limit by the half-arc-distance. The upper refined solution interval has an upper refined upper limit at the upper limit of the solution interval and has an upper refined lower limit displaced on the unit circle from the upper refined upper limit by the half-arc-distance. The improved accuracy apparatus further includes a selection unit coupled with the angle bisecting unit and with the polynomial zero detecting unit. The selection unit generates a lower limit output and an upper limit output appropriately to indicate whether the respective root is located in the lower refined solution interval or in the upper refined solution interval. The apparatus may be employed to further refine accuracy of the location by employing the half-distance generating unit and the angle bisecting unit cooperatively to successfully perform the bisecting operation to effectively bisect each solution interval into a successive upper refined solution interval and successive lower refined solution interval, the respective root being located within one of the successive upper refined solution interval and the successive lower refined solution interval. The successive bisecting operation is iteratively effected until a predetermined desired accuracy of the arc-distance is included in a successive refined solution interval.

Preferably, the initial value is expressed as a sinusoidal value of an angular displacement on the unit circle and the step value is expressed as a sinusoidal value of a step-angular displacement on the unit circle.

The invention further includes a method for locating a plurality of roots of a line spectrum pair expression on a unit circle which comprises the steps of: (1) receiving a respective representation of an initial value for locating a first site on the unit circle; (2) receiving a representation of a step value for defining an arc-distance on the unit circle; (3) generating a plurality of intervals on the unit circle, each

interval having a lower limit and an upper limit, the intervals including an initial interval and a plurality of succeeding intervals (The lower limit of the initial interval is the initial value and the upper limit of the initial interval is displaced on the unit circle from the initial value by an arc-distance. Each respective succeeding interval has its respective lower limit coincident with the upper limit of the next preceding interval, and each respective succeeding interval has its respective upper limit displaced on the unit circle from its respective lower limit by the arc-distance); (4) evaluating the line spectrum pair expression for at least the upper limit and the lower limit of each respective interval; (5) recognizing the presence of a respective root of the line spectrum pair expression when the line spectrum pair expression changes sign within a particular interval; (6) designating each such particular interval as a solution interval; and (7) generating the lower limit and the upper limit of each such solution interval to identify the arc-distance within which the particular root is located.

It is therefore an object of the present invention to provide an apparatus and method of locating a plurality of roots of a line spectrum pair expression which employs sinusoidal values in its calculations to determine the roots without deriving the roots from an angular representation associated with the unit circle.

It is a further object of the present invention to provide an apparatus and method of locating a plurality of roots of a line spectrum pair expression on a unit circle which is efficient in its use of memory storage space and addressal of that memory storage space in determining the roots.

It is a still further object of the present invention to provide an apparatus and method for locating a plurality of roots of a line spectrum pair expression on the unit circle which may provide increased accuracy of location of a root by successively bisecting an interval within which a root is determined to be located.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 13-1 is a schematic drawing of the angles and values employed in "stepping around" the unit circle to locate roots of a line spectrum pair expression. Figure 13-2 is a schematic drawing of angles and values employed in approximating the cosine of a bisecting angle of a sector given the cosines of boundary angles of the sector.

Figure 13-3 is a schematic drawing of the preferred embodiment of a waveform generating unit for use in the present invention.

Figure 13-4 is a schematic drawing of the preferred embodiment of a polynomial zero crossing detector unit for use in the present invention.

Figure 13-5 is a schematic drawing of the preferred embodiment of a half-angle generating unit for use in the present invention.

Figure 13-6 is a schematic drawing of the preferred embodiment of an angle bisecting unit for use in the present invention.

Figure 13-7 is a schematic drawing of the preferred embodiment of the apparatus of the present invention.

Figure 13-8 is a schematic diagram of an alternate embodiment of a waveform generating unit for use in the present invention.

Figure 13-9 is a flow diagram illustrating the preferred embodiment of the method of the present invention.

Figure 13-10 is a block diagram showing an apparatus for analyzing speech signals.

DETAILED DESCRIPTION OF THE INVENTION

Before referring to the figures, a brief theoretical background discussion will be provided.

It is desirable in modeling vocal tract action to amplify and suppress certain spectral ranges (modifications of spectrum of "signals" from vocal cords). A basic known canonical feedback digital filter transfer function appropriate for modeling a vocal tract may be expressed as:

$$V(z) = \frac{1}{\sum_{i=0}^n a_i \cdot z^{-i}} ; a_0 = 1 \quad (13.1)$$

In models of vocal tract action, there is a need to transmit values of a_i to a decoder from an encoder, so one must be able to quantize the values of a_i . The most preferable quantization technique is the technique that uses the fewest bits in order to most efficiently employ the apparatus used for such transmission.

Generally, the coefficient values a_i have some undesirable qualities, such as: some coefficients a_i are more sensitive than others; sometimes an error in one coefficient a_i affects the entire spectrum sought to be represented.

It has been found that it is better to transform the coefficients a_i to a quantized transformation value which is, necessarily, capable of inversion to recover the original value from which the quantized value is transformed. There are many transformations; line spectrum pair (LSP) is one transformation.

The apparatus and method of the present invention are appropriate for determining roots of the expressions which result from the LSP transformation.

As is well known in the art of speech analysis and speech synthesis, a linear predictive coding (LPC) analysis results in an all-pole filter described by a transfer function involving parameters a_i known as LPC coefficients. It is also well known that LPC coefficients are inappropriate for quantization primarily

because of their wide dynamic range and variation of different LPC coefficients in their sensitivity to errors.

In summary, an apparatus for analyzing speech signals as described above is shown in Figure 13-10. An LPC analysis of speech signals 400 is performed by an LPC analysis unit 402 parameters or coefficients 404. A transformation unit 406 transforms LPC coefficients 404 to a corresponding line spectrum pair (LSP) expression or polynomial 408. Roots 412 of LSP expression 408 are then determined by a root-locating unit 410. As stated above, LPC analysis unit 402 and transformation unit 406 are well-known in the art. Thus, this detailed description focusses upon the details of root-locating unit 410.

Line spectrum pair parameters may be derived from a consideration of two extreme artificial boundary conditions applied to LPC coefficients. The resulting LSP parameters can be interpreted as the resonant frequencies of the vocal tract under the two extreme artificial boundary conditions at the glottis. The two polynomials which involve the line spectrum pair parameters possess some interesting properties summarized as follows:

- (1) all roots of the two polynomials lie on the unit circle in the complex plane; and
- (2) the roots of the two polynomials alternate each other on the unit circle so that the following relationship is always satisfied:

$$0 = \omega_0 < \omega_1 < \omega_2 < \dots < \omega_{p-1} < \omega_p < \omega_{p+1} \quad (13.2)$$

Even ω 's are derived from one polynomial, and odd ω 's are derived from the other polynomial. This is referred to as the ordering property of LSP parameters.

Further, all roots of each of the polynomials (each of the polynomials is a fifth order polynomial) lie on the unit circle in the Z-plane (the complex plane) so that when the model (or the system represented by the model) is in equilibrium, the roots lie on the unit circle. Outside the unit circle the signal amplitude increases, and inside the unit circle the signal damps.

Preferably, each polynomial is expressed in terms of $\cos \omega$. The inventors have found that intervals of

$$\left(\omega = \frac{\pi}{64} \right)$$

will not miss any roots. Further, bisecting the

$$\frac{\pi}{64}$$

angle twice to make

$$\pi = \frac{\pi}{256}$$

gives sufficient accuracy for purposes of decoding voice data.

A useful method for more accurately determining the roots, at least approximately, of the LSP polynomials is by approximating the cosine of a bisecting angle within an interval given the cosine of the boundary angles, and determining in which half-angle interval the root lies. Thus, one can double the precision by which one determines where on the unit circle a root lies simply by halving an interval and determining within which newly-defined half-interval the root lies.

It is known that

$$\cos(A+B) = 2 \cos A \cos B - \cos(A-B) \quad (13.3)$$

If we let

$$B = \frac{\pi}{64}$$

(the incremental value), we can be assured that the intervals are established appropriately that no roots will be missed on the unit circle. As a practical matter, however, since the value

$$B = \frac{\pi}{64}$$

is close to zero, the value of

$$\cos \frac{\pi}{64}$$

is too close to 1 to be efficiently handled by digitally configured logical elements. That is, the value of

$$\cos \frac{\pi}{64}$$

involves too many leading nines which must be carried and stored and manipulated throughout the various calculations required. Such inefficiency may be avoided by using the value

$$1 - \cos \frac{\pi}{64} = \frac{\pi}{64}$$

to ease storage and processing requirements.

Thus, equation (13.3) may be rewritten as:

$$\cos(A+B) = 2 \cos A \cos B - \cos(A-B) \quad (13.4)$$

Equation (13.4) is the equation used in the preferred embodiment of the present invention to "step around" the unit circle to locate the five roots of an LSP polynomial expression. However, for ease in understanding the invention, the remaining discussions explaining the invention will be based upon the expression in equation (13.3).

Figure 13-1a is a schematic drawing of the angles and values employed in "stepping around" the unit circle to locate roots of a line spectrum pair expression.

In Figure 13-1a, ω or the step angle used to "step around" a unit circle 11 to determine location of roots of a line spectrum pair expression is illustrated as being equal to B, so that the radii 16, 18 delineate a sector 15 of unit circle 11, and radii 18, 19 delineate a sector 17 of unit circle 11. We know that:

$$\cos(A+B) = 2 \cos B \cos A - \cos(A-B) \quad (13.5)$$

Thus, sector 15 has an upper limit equal to the value $\cos A$ and has a lower limit equal to the value $\cos(A-B)$. Further, sector 17 has an upper limit equal to the value $\cos(A+B)$ and has a lower limit equal to the value $\cos A$. If we let $A = \left(A - \frac{B}{2} \right)$, and let $B = \frac{B}{2}$, then

$$\left[\left(A - \frac{B}{2} \right) + \frac{B}{2} \right] = \quad (13.6)$$

$$2 \cos \frac{B}{2} \cos \left(A - \frac{B}{2} \right) - \cos \left[\left(A - \frac{B}{2} \right) - \frac{B}{2} \right]$$

Therefore:

$$\cos A = 2 \cos \frac{B}{2} \cos \left(A - \frac{B}{2} \right) - \cos (A - B) \quad (13.7)$$

Solving for $\cos \left(A - \frac{B}{2} \right)$:

$$\cos \left(A - \frac{B}{2} \right) = \frac{\cos A + \cos (A - B)}{2 \cos \left(\frac{B}{2} \right)} \quad (13.8)$$

$\frac{1}{2 \cos \left(\frac{B}{2^n} \right)}$ values of predetermined values of the expression

the bisection of step angle B can be easily performed using a computer apparatus. Typically, in the preferred embodiment of the present invention, only a very small number of stored values of the expression

$$\frac{1}{2 \cos \left(\frac{B}{2^n} \right)}$$

will be required. By such successive bisection, the technique for determining a root of an LSP expression may be very efficiently performed to a desired accuracy.

For the purposes of a coarse search for roots, the value of $\cos B$ is a constant stored in a computer memory. Recalling equation (13.5):

$$\cos (A+B) = 2 \cos B \cos A - \cos (A-B) \quad (13.9)$$

if one lets

$$A = \frac{B}{2} \quad \text{and} \quad B = \frac{B}{2}$$

$$\cos\left(\frac{B}{2} + \frac{B}{2}\right) = 2\cos\frac{B}{2} \cos\frac{B}{2} - \cos\left(\frac{B}{2} - \frac{B}{2}\right) \quad (13.10)$$

which reduces to:

$$\cos B = 2\cos\frac{B}{2} \cos\frac{B}{2} - 1 \quad (13.11)$$

Solving for $\cos\frac{B}{2}$:

$$\cos\frac{B}{2} = \sqrt{\frac{\cos B + 1}{2}} \quad (13.12)$$

More generally stated for values of n other than n = 1:

$$\cos\frac{B}{2^n} = \sqrt{\frac{\cos\frac{B}{2^{n-1}} + 1}{2}} \quad (13.13)$$

Hence, the cosines of the successively bisected step angles

$$B, \frac{B}{2}, \frac{B}{2^2}, \dots, \frac{B}{2^n}$$

may be calculated in sequence and stored as required.

One problem which has arisen is that the difference equation expressed as equation (13.5) is difficult to calculate accurately for step angles B when using finite precision arithmetic, a common problem with computers. This difficulty arises from the loss of precision induced by leading nines in the value of cos B. For example, $\cos 1^\circ$ equals 0.999847.

It is often necessary to work with such small angle or phase increments in determining roots of line spectrum pair expressions to a sufficiently accurate certainty. One solution is to use the expression (1-cos B) in the computation instead of the value of cos B.

Thus, if one substitutes $\cos B = 1 - (1 - \cos B)$ in equation (13.5):

$$\cos(A+B) = 2 \cos A (1 - (1 - \cos B)) - \cos(A-B) \quad (13.14)$$

which expands to:

$$\cos(A+B) = 2 \cos A - 2 \cos A (1 - \cos B) - \cos(A-B) \quad (13.15)$$

An advantage of this representation is that $(1 - \cos B)$ can be scaled so that all of the digits are significant. In terms of finite precision arithmetic implementation, the product $(\cos A (1 - \cos B))$ can be determined accurately and scaled before summing with the other components. Further, this formulation is ideal for application at locations in which the precision of the ALU (arithmetic logic unit) employed in a computer is greater than that of the multiplier. As a result, the values can be determined with minimal loss of precision.

By similar logic, equation (13.8) requires the use of values of

$$\cos\left(\frac{B}{2}\right)$$

and further bisection will require use of the values

$$\cos\left(\frac{B}{2^n}\right)$$

It is useful to rewrite equation (13.8) as:

$$\cos\left(A - \frac{B}{2}\right) = \frac{\cos A + \cos(A - B)}{2} \cdot \frac{1}{\cos\left(\frac{B}{2}\right)} \quad (13.16)$$

If we let $A = A+B$, equation (13.15) becomes:

$$\cos\left(A + \frac{B}{2}\right) = \frac{\cos A + \cos(A + B)}{2} \cdot \frac{1}{\cos\left(\frac{B}{2}\right)} \quad (13.17)$$

Recall the relationship (Taylor series):

$$(1-x)^{-1} = 1+x+x^2+x^3+\dots; |x|<1 \quad (13.18)$$

Therefore, if we let $(1-x)$ equal

$$\cos\left(\frac{B}{2}\right),$$

Equation (13.17) may be used to approximate $(1-x)^{-1} = 1+x$.

Thus if:

$$(1-x) = \cos\left(\frac{B}{2}\right) \quad (13.19)$$

then:

$$1+x = 1 + \left(1 - \cos\frac{B}{2}\right) \quad (13.20)$$

$$\frac{1}{\cos\left(\frac{B}{2}\right)} = (1-x)^{-1} \quad (13.21)$$

$$\frac{1}{\cos\left(\frac{B}{2}\right)} = 1+x = 1 + \left(1 - \cos\frac{B}{2}\right) \quad (13.22)$$

Thus, an approximation to

$$\left(\frac{1}{\cos\left(\frac{B}{2}\right)}\right)$$

and $\left(\frac{1}{\cos\left(\frac{B}{2^n}\right)}\right),$

is provided by the stored values of

$$1 - \cos \frac{B}{2^n} .$$

Since x in this situation is very small by definition of its requirement for this process, then the approximation by the Taylor series is accurate, especially when finite precision arithmetic is being used.

Thus, substituting equation (13.21) into equation (13.16) yields:

$$\cos \left(A + \frac{B}{2} \right) = \frac{\cos A + \cos(A+B)}{2} \cdot \left(1 + \left(1 - \cos \frac{B}{2} \right) \right) \quad (13.23)$$

As previously mentioned, for sufficiently small angles (10° is sufficiently small), the inventors have determined that the expression of equation (13.21) is sufficiently accurate.

Thus, given $(1 - \cos B)$, one can perform all of the operations required to determine the cosines of the angles of a line spectrum pair calculation using linear search bisection. Such an approach should be compatible with finite precision arithmetic generally used in computers. Thus, given equation (13.5):

$$\cos(A+B) = 2 \cos B \cos A - \cos(A-B)$$

let $\cos X = 1 - (1 - \cos X)$; so that equation (13.5) may be rewritten as:

$$1 - (1 - \cos(A+B)) = 2[1 - (1 - \cos A)(1 - (1 - \cos B))] - [1 - (1 - \cos(A-B))] \quad (13.24)$$

Thus:

$$1 - (1 - \cos(A+B)) = 2[1 - (1 - \cos B) - (1 - \cos A) + (1 - \cos A)(1 - \cos B)] - 1 + (1 - \cos(A+B)) \quad (13.25)$$

Further reducing:

$$1 - (1 - \cos(A+B)) = 2 + 2[-(1 - \cos B) - (1 - \cos A) + (1 - \cos A)(1 - \cos B)] - 1 + (1 - \cos(A+B)) \quad (13.26)$$

Thus:

$$-(1 - \cos(A+B)) = 2[-(1 - \cos B) - (1 - \cos A) + (1 - \cos A)(1 - \cos B)] + 1(1 - \cos(A+B)) \quad (13.27)$$

Thus:

$$(1 - \cos(A+B)) = -2[-(1 - \cos B) - (1 - \cos A) + (1 - \cos A)(1 - \cos B)] - 1(1 - \cos(A+B)) \quad (13.28)$$

$$(1 - \cos(A+B)) = 2[(1 - \cos A) - (1 - \cos A)(1 - \cos B) + (1 - \cos B)] - (1 - \cos(A+B)) \quad (13.29)$$

Equation (13.28) enables one to configure a waveform generator accurately for small angle increments for implementation of the present invention.

Figure 13-2 is a schematic drawing of angles and values employed in approximating the cosine of a bisecting angle of a sector given the cosines of boundary angles of the sector.

In Figure 13-2, a unit circle 10 is illustrated as centered on a center point 12. A sector 14 of unit circle 10 is delineated by a lower limit at a radius 16 and an upper limit at a radius 18. Radii 16, 18 are separated by a step angle B. Step angle B is bisected to two half-step angles

$$\frac{B}{2}$$

Thus, a central point in sector 14 is defined as the value $\cos A$ at a point where a radius 20 intersects unit circle 10. Radius 20 is defined as a radius displaced from radius 16 by half-step angle

$$\frac{B}{2}$$

Consequently, the lower limit of sector 14 as defined by the intersection of radius 16 with unit circle 10 is a point having a value

$$\cos\left(A - \frac{B}{2}\right);$$

the upper limit of sector 14 defined by the intersection of radius 18 with unit circle 10 has a value of

$$\cos\left(A + \frac{B}{2}\right)$$

Equation (13.3) may be rewritten to fit the scheme of Figure 13-1 as follows:

$$\cos\left(A + \frac{B}{2}\right) = 2\cos A \cos \frac{B}{2} - \cos\left(A - \frac{B}{2}\right) \quad (13.30)$$

Solving for $\cos A$:

$$\frac{\cos\left(A + \frac{B}{2}\right) + \cos\left(A - \frac{B}{2}\right)}{2\cos\frac{B}{2}} \quad (13.31) \quad \cos$$

In Figure 13-1b,

$$\cos\left(A + \frac{B}{2}\right) \text{ and } \cos\left(A - \frac{B}{2}\right)$$

are end points of sector 14. Thus, if one knows

$$\cos\frac{B}{2}$$

one can solve for $\cos A$. Clearly, one can more finely segment sector 14 by successively bisecting each half-step angle. Thus, the interval between successive radii in defining successively finer sectors may be expressed as

$$\cos\frac{B}{2^n}$$

Increasing the value of n will increase accuracy of the model for determining that a root of an equation lies in a sub-divided sector defined by a step angle of

$$\cos\frac{B}{2^n}$$

For simplicity of explaining the present invention, the relationships below will be discussed using $n = 1$.

If we let

$$x = 1 - \cos\frac{B}{2^n},$$

then the Taylor series (equation (13.16)) provides $(1-x)^{-1} \cong 1 + x$, for small x , so that:

$$\frac{1}{\cos \frac{B}{2^n}} = \frac{1}{1 - \left(1 - \cos \frac{B}{2^n}\right)} = 1 + \left(1 - \cos \frac{B}{2^n}\right) \quad (13.32)$$

$$1 - \left(1 - \cos \frac{B}{2}\right) = \frac{1}{2} \left(\frac{1 - \cos B}{4}\right)^2 + \left(\frac{1 - \cos B}{4}\right) \quad (13.33)$$

Thus, one can closely approximate $\cos A$ (the cosine of the angle bisecting sector 14 in Figure 13-1) by using equation (13.32), and one can approximate a sub-sector to any desired accuracy by further halving the sub-sector.

Figure 13-3 is a schematic drawing of the preferred embodiment of a waveform generating unit for use in the present invention.

In Figure 13-3, a waveform generating unit 20 is illustrated which implements the expression of equation (13.4). An input 22 provides an initial value for step angle B to a function generator 24 which provides an output $(1 - \cos B)$ on a line 26. Function generator 24 may be any known manner of device which takes an input such as a known step angle B and generates the value $(1 - \cos B)$ therefrom. That is, it may include a read only memory (ROM) or a random access memory (RAM), or a sinusoidal generating unit, or any other units known to those skilled in the art which can generate appropriate sinusoidal values. Line 26 provides the output from function generator 24 to a leading zero detect unit 28 via a line 27, and to a shift unit 30.

The embodiment illustrated in Figure 13-3 detects leading zeroes, contemplating storage and treatment of numbers within waveform generating unit 20 in decimal representation. Of course, other representation, storage and treatment formats will require somewhat different scaling techniques, such as detecting a sign bit and leading 1's or leading zeroes according to the value of the sign bit. Any scaling technique appropriate to a value representation approach is contemplated as applicable to and usable in the present invention. Leading zero detection is illustrated here in an exemplary role.

Leading zero detect unit 28 provides an output at line 32 indicating the number m of leading zeros detected in the output provided via line 26, 27 from function generator 24 to leading zero detect unit 28. The number of leading zeros m is provided via a line 34 to shift unit 30 and shift unit 30 shifts m places to the left to effect a binary multiplication of 2^m in order to scale the output received via line 26 from function generator 24. Thus, a scaled value $2^m (1-\cos B)$ of the output value $(1-\cos B)$ on line 26 is provided on line 36 to a multiplier 38. Considering Figure 13-1a and Figure 13-3 together, one may note that as the phasor, represented by radii 16, 18, 19 rotates counter-clockwise in Figure 13-1a, the radius is represented by radius 18 occurs later than radius 16, and the next-occurring phasor in stepping along unit circle 11 in Figure 13-1a is represented by radius 19. Radius 19 intersects unit circle 11 at a point represented by the value $\cos (A+B)$. Thus, when considering the output 40 of waveform generating unit 20 in Figure 13-3, that output representing the value $\cos (A+B)$, a delay unit 42 receiving via line 41 the value $\cos (A+B)$ will generate on a line 43 the value $\cos A$, since $\cos A$ is the once-earlier value of the intersection of the phasor with unit circle 11 as the phasor steps around unit circle 11 by increments measured by step angle B . The signal on line 43 representing the value $\cos A$ is provided as an input to delay unit 44, as an input to multiplier 38, and to a scaling unit 46. Scaling unit 46 shifts digits in the signal carried on line 43 to the left one space to effect a binary multiplication by two. Thus, the signal on a line 47 is equal to two times the signal present on input line 45. That is, line 47 carries the value $2 \cos A$. Line 47 provides that value $(2 \cos A)$ as an input to adder 48.

The signal carried on line 36 represents the value $2^m (1-\cos B)$ and is provided as a second input to multiplier 38. Multiplier 38 generates a signal on an output line 50 representing a value $2^m (\cos A (1-\cos B))$; the signal on line 50 is provided as an input to shift unit 52. Shift unit 52 shifts to the right m places (m being the number of leading zeroes detected by leading zero detect unit 28) to rescale the signal received from multiplier 38 via line 50. Shift unit 52 generates on an output line 54 a signal representing the value $\cos A (1-\cos B)$, which value is provided as an input to scaling unit 56. A multiplier 58 multi-

plies the output received from scaling unit 56 by the quantity -1 and provides a signal to adder 60 which represents the value $-2 \cos A (1 - \cos B)$.

Delay unit 44 generates a signal on a line 61 representing the value $\cos (A - B)$, which is the next-preceding phasor intersect on the unit circle 11 (see Figure e13-1a). The value $\cos (A - B)$ is provided to a multiplier 62 which multiplies that value by -1. Multiplier 62 provides a signal on a line 63 representing the value $-\cos (A - B)$ as a second input to adder 60.

Of course, for proper operation of waveform generating unit 20, delay unit 42 must be initialized to $\cos (-B)$ and delay unit 44 must be initialized to zero to properly initiate operations.

Adder 60 provides a signal on a line 64 representing the value $-2 \cos A (1 - \cos B) - \cos (A - B)$ to adder 48. Adder 48 provides a signal on line 65 representing the sum of inputs received via lines 47, 64 $> 2 \cos A - 2 \cos A (1 - \cos B) - \cos (A - B)$, which (by equation (13.4)) equals the value $\cos A + B$. The signal representing the value $\cos A$ is generated via a line 66 from line 43 to an output 68.

In order to facilitate understanding the present invention, like elements will be identified by like reference numerals in the various drawings.

Figure 13-4 is a schematic drawing of the preferred embodiment of a polynomial zero crossing detector unit for use in the present invention. In Figure e13-4, a polynomial zero crossing detector 69 is illustrated and includes a waveform generating unit 20 of the sort described in connection with Figure 13-3. Waveform generating unit 20 receives an input 26 bearing a signal representing the value $(1 - \cos B)$. Waveform generating unit 20 generates the value $\cos (A + B)$ at an output 40, and generates a value $\cos A$ at an output 68.

A line 70 delivers the value $\cos (A + B)$ from line 40 to a polynomial treating unit 72. Polynomial treating unit 72 receives the value $\cos (A + B)$ via line 70. Polynomial treating unit 72 evaluates the polynomial (preferably, in the present embodiment of the invention, a line spectrum pair polynomial) for the value received on line 70 and generates an evaluation value of that particular polynomial treated by poly-

nomial treating unit 72 on an output line 74. The evaluation value of the particular polynomial treated by polynomial treating unit 72 for the value provided via line 70 is provided at an input 76 of a sign change detecting unit 78. A delay unit 80 also receives the output from polynomial treating unit 72 via a line 82 and generates an evaluation value for evaluation of the polynomial treated by polynomial treating unit 72 during a once-previous time. That is, the output of delay unit 80 on line 84 is an evaluation value for the polynomial treated by the polynomial treating unit 72 at the value $\cos A$. This evaluation value for the polynomial at value $\cos A$ is provided to an input 86 of sign change detecting unit 78. Sign change detecting unit sends a sign change detected signal via a line 88 to a buffer 90.

Buffer 90 also receives via line 40 the value $\cos (A+B)$, and via line 68 the value $\cos A$. The sign change detected signal provided via line 88 to buffer 90 switches buffer 90 on when there is a sign change detected by sign change detecting unit 78. Output 92 of buffer 90 generates an output representing the value $\cos (A+B)$, and output 94 of buffer 90 generates an output representing the value $\cos A$. Outputs 92, 94 are updated when a sign change is detected by sign change detector 78 and a sign change detected signal is provided via line 88 to buffer 90. Stated another way, the values of $\cos (A+B)$ (on line 40) and $\cos A$ (on line 68) are gated through buffer 90 to outputs 92, 94 when sign change detecting unit 78 detects a sign change between its signals received at inputs 76 and 86 and sends a sign change detected signal via line 88 to turn on buffer 90. As a result, the signals provided at outputs 92, 94 from buffer 90 are such that a zero (i.e., a root) of the polynomial treated by polynomial treating unit 72 lies between the angles A and $(A+B)$, which are represented by the values $\cos A$ and $\cos (A+B)$.

Figure 13-5 is a schematic drawing of the preferred embodiment of a half-angle generating unit for use in the present invention. In Figure 13-5, a half-angle generating unit 96 implements the expression of equation (13.32). Thus, the output 102 of half-angle generating unit 96 carries a signal representing the value $1 - \cos \frac{B}{2^{n+1}}$.

This signal is also applied via a line 104 to a delay unit 100. Delay unit 100 presents at its output line

106 a signal representing the value

$$\cos \frac{B}{2^n}$$

Reset values are occasionally provided to delay unit 100 via a line 98 representing the value $(1 - \cos B)$. The output carried on line 106 is applied to a scaling unit 108 which shifts digits to the right two places to effect a division of the signal received via line 106 by four. Thus, the output of scaling unit 108 carried on line 110 is a signal which represents the value

$$\frac{\left(1 - \cos \frac{B}{2^n}\right)}{4}$$

That signal is applied via a line 112 to a leading zero detect unit 114, is applied via a line 116 to a shift unit 118, and is applied via a line 120 to a summing unit 122. Leading zero detect unit 114 generates at an output line 124 a signal representing m , the number of leading zeros detected in the value received via line 112. As described before in discussing leading zero detection, any scaling apparatus appropriate for scaling numbers represented in a given format may be employed in place of leading zero detect unit 114.

The value m is applied via a line 126 to shift unit 118, and via a line 128 to a scaling unit 130. Shift unit 118 shifts digits contained in the signal received via line 116 to the left m places to effect a multiplication by the quantity 2^m so that the output generated on line 132 from shift unit 118 represents the value

$$2^m \left(\frac{1 - \cos \frac{B}{2^n}}{4} \right)$$

Scaling unit 130 effects a shift to the left by one place to effect a multiplication by 2 to generate on a line 134 a signal representing the value $2m$. The value $2m$ is applied via line 134 to a shift unit 136. The signal carried on line 132 is applied to a squaring unit 138. Squaring unit 138 generates an output signal line 140 representing the value

$$2^{2m} \left(\frac{1 - \cos \frac{B}{2^n}}{4} \right)^2$$

The signal is applied via line 140 to shift unit 136; shift unit 136 generates an output signal on line 142 representing the value received via line 140 divided by

$$2^{2m} \left(\frac{1 - \cos \frac{B}{2^n}}{4} \right)^2$$

A scaling unit 144 shifts digits in signals received via line 142 to the right one place to effect a division by 2 so that signals carried on an output line 146 from scaling unit 144 to summing unit 122 represent the value

$$\frac{1}{2} \left(\frac{1 - \cos \frac{B}{2^n}}{4} \right)^2$$

Summing unit 122 generates on an output line 148 the value

$$\frac{1}{2} \left(\frac{1 - \cos \frac{B}{2^n}}{4} \right)^2 +$$

As we know from equation (13.31), that approximately equals $1 - \cos B/2^{n+1}$.

Figure 13-6 is a schematic drawing of the preferred embodiment of an angle bisecting unit for use in the present invention. In Figure 13-6, an angle bisecting unit 150 implements the expression of equation (13.22). Angle bisecting unit 150 receives two inputs 152, 154 which are appropriate for connection with outputs 92, 94 of the polynomial zero crossing detector unit 69 illustrated in Figure 13-4. Preferably, input 152 receives a signal representing the value $\cos(A+B)$, and input 154 receives a signal representing

the value $\cos A$. Both inputs 152, 154 are applied to a summing unit 156. Summing unit 156 generates a signal on a line 158 which represents the value

$\cos(A+B) + \cos A$. The signal is applied to a scaling unit 160. Scaling unit 160 shifts digits contained in the signal provided on line 158 to the right one space, thereby effecting a binary division by two so that scaling unit 160 generates on a line 162 a signal representing the value

$$\frac{\cos(A+B) + \cos A}{2}$$

Line 162 is connected to a multiplier 164. A second input to multiplier 164 is provided via a line 166. Line 166 carries a signal representing the value

$$1 + \left(1 - \cos \frac{B}{2}\right)$$

Thus, the output line 168 from multiplier 164 carries a signal representing the value

$$\frac{\cos(A+B) + \cos A}{2} \cdot \left(1 + \left(1 - \cos \frac{B}{2}\right)\right)$$

According to equation (13.22), that value equals

$$\cos\left(A + \frac{B}{2}\right)$$

Figure 13-7 is a schematic drawing of the preferred embodiment of the apparatus of the present invention.

In Figure 13-7 a root determining apparatus 170 is illustrated which includes a polynomial zero crossing detector 69 (Figure e13-4), which includes a waveform generating unit 20 (Figure 13-3); a half-angle generating unit 96 (Figure e13-5); a delay unit 172; and switches 174, 176, 178. Also included in root determining apparatus 170 are an angle bisecting unit 150 (Figure 13-6); polynomial treating units 180,

182, 184; zero crossing detectors 186, 188; a select logic unit 190; and a selector unit 192.

An input is provided to root determining apparatus 170 via a line 194 in the form of a signal representing the value $1 - \cos B$. This input signal is provided via a line 26 to waveform generator 20 in polynomial zero crossing detector 69, and is also provided via a line 98 and a pole 198 of switch 178 to half-angle generating unit 96.

Half-angle generating unit 96 generates a signal on a line 102 representing the value

$$1 - \cos \frac{B}{2^{n+1}}$$

and provides that signal to delay unit 172 as well as to an input 214 of angle bisecting unit 150. Delay unit 172 generates a signal on a line 200 representing the value

$$1 - \cos \frac{B}{2^n}$$

and applies that value to a pole 202 of switch 178.

Polynomial zero crossing detector 69 provides at its output 92 a signal representing the value $\cos(A+B)$, and applies that value to a pole 204 of switch 174. Polynomial zero crossing detector 69 also generates a signal on its output 94 representing the value $\cos A$, and applies that signal to a pole 206 of switch 176. Of course, recalling the explanation of polynomial zero crossing detector 69 in connection with Figure 13-4, polynomial crossing detector 69 ensures that the values represented by the signals generated on outputs 92, 94 bracket a zero solution of the line spectrum pair polynomial treated by the polynomial treating unit 72 (Figure 13-4) in polynomial zero crossing detector 69.

Switches 174, 176, 178 are arranged for accommodating initial setup of root determining apparatus 170 in a first orientation and for accommodating segmenting operations of root determining apparatus 170 in a second orientation. Thus, when switch 174 connects pole 203 with pole 204, and when switch 176 connects pole 205 with pole 206, and when switch 178 connects pole 201 with pole 198, then root

determining apparatus 170 is configured for initial/reset operation. In the initial/reset operation configuration, the signal on output 92 is provided to a selector unit 192 at a selector unit input 210, and the signal on output 94 is applied to a selector unit input 212 of selector unit 192. Similarly, in the initial/reset operation configuration, the input carried on line 98 representing the value $1 - \cos B$ is applied to half-angle generating unit 96.

The signal generated by half-angle generating unit 96 on line 102 is provided via a line 103 to an input 214 of angle bisecting unit 150. Angle bisecting unit 150 provides at its output 168 (Figure 13-6) a signal representing the value

$$\cos\left(A + \frac{B}{2}\right),$$

which is applied to a selector unit input 213 of selector unit 192.

In the initial/setup operation configuration wherein output 92 is connected via poles 203, 204 with selector unit input 210, and wherein output 94 is connected via poles 205, 206 with selector unit input 212, selector unit 192 selects input signals representing the quantity $\cos(A+B)$ for application to its output 216 as the upper limit U_p of a sector (such as sector 15 of Figure 13-1a) and selector unit 192 selects signals representing the quantity $\cos A$ from selector unit input 212 for application to its output 218 to represent the lower limit L_p of a sector such as sector 15 of Figure 13-1a. In such an initial/setup operation configuration, selector unit 192 does not consider signals appearing at its input 213 for application to its outputs 216, 218.

So long as the initial, or coarse, solution of the location of roots is sufficient without further bisection of a sector (such as sector 15 of Figure 13-1a) root determining apparatus 170 does not involve angle bisecting unit 150 in determining upper limit U_p and lower limit L_p outputs for its outputs 216, 218. Following the initial determination of the first sector 15 established by upper limit U_p and lower limit L_p by root determining apparatus 170, the next iteration of parameters is generated and dealt with, and root

determining apparatus 170 remains in its initial/reset operation configuration as root determining apparatus 170 "steps around" unit circle 11 in increments established as sectors 15, 17 (Figur e13-1). This is so in the case where the coarse (or first cut) estimation of location of roots on the unit circle suffices and there is no need to further segment sector 15 (Figur e13-1) to more finely determine location of roots on unit circle 11.

If a finer segmentation of sector 15 is required to more accurately locate roots on unit circle 11, then switch 174 is reoriented to connect pole 203 with pole 220, switch 176 is reoriented to connect pole 205 with pole 222, and switch 178 is reoriented to connect pole 201 with pole 202. In such a segmenting operations configuration, angle bisecting unit 150 is operationally included in root determining apparatus 170 so that upper limit U_p is provided via a feedback line 224 and a line 226 to an input 228 of angle bisecting unit 150. Similarly, lower limit L_p is provided via a feedback line 230 and a line 232 to an input 234 of angle bisecting unit 150. Still further, input from delay unit 172 is provided via line 200 and via switch 178 to the input of half-angle generating unit 96.

Still further, line 226 provides upper limit U_p to a polynomial treating unit 180, line 232 provides lower limit to polynomial treating unit 184, and output 168 (representing

$$\cos\left(A + \frac{B}{2}\right)$$

from bisecting unit 150 is provided via a line 236 to a polynomial treating unit 182. Polynomial treating units 180, 182, 184 preferably are similar to polynomial treating unit 72 (F igure13-4) in that they each provide an evaluation value to the polynomial for which roots are sought by root determining apparatus 170 for the values provided via their respective input lines 226, 236, 232. Thus, an evaluation value for the polynomial for which roots are sought by root determining apparatus 170 is provided for the value received via line 226 by polynomial treating unit 180 on an output line 240 to a zero crossing detector 186. An evaluation value for the polynomial for which roots are sought is provided for the value received

via input 236 by polynomial treating unit 182 on an output 242 to zero crossing detector 186 and to a zero crossing detector 188. Similarly, an evaluation value for the polynomial for which roots are sought is provided for the value provided via input 232 on an output line 244 to zero crossing detector 188.

Zero crossing detector 186 provides an output via a line 246 to a select logic unit 190, and zero crossing detector 188 provides an output via a line 248 to select logic unit 190. Thus, select logic unit 190 receives an indication via lines 246, 248 whether a zero crossing occurs between the values $\cos(A+B)$ and

$$\cos\left(A + \frac{B}{2}\right)$$

(via line 246) or whether a zero crossing occurs between the values $\cos A$ and

$$\cos\left(A + \frac{B}{2}\right)$$

(via line 248). That is, select logic unit 190 selects values to define the interval containing the zero crossing: an upper half-sector between lower limit

$$\cos\left(A + \frac{B}{2}\right)$$

and upper limit $\cos(A+B)$, or a lower half-sector between lower limit $\cos A$ and upper limit

$$\cos\left(A + \frac{B}{2}\right)$$

A signal is provided by select logic unit 190 via a line 250 to an input 252 to selector unit 192 indicating which half-sector contains the zero crossing.

Selector unit 192 contains logic which (1) defines a sector having an upper limit U_p (on output 216) set to the value received via input 213

$$\left[\cos\left(A + \frac{B}{2}\right) \right]$$

and having a lower limit L_p (on output 218) set to the value received via input 212 $\cos A$ when the input from select logic unit 192 via line 250 indicates that the zero crossing occurred in the lower half-sector bounded by $\cos A$ and

$$\cos\left(A + \frac{B}{2}\right)$$

or (2) defines a sector having an upper limit U_p set to the value received via input 210 $\cos(A+B)$, and having a lower limit L_p set to the value received via input 213

$$\left[\cos\left(A + \frac{B}{2}\right) \right]$$

when select logic unit 190 outputs a signal to selector unit 192 via line 250 that indicates the zero crossing occurred in the upper half-sector bounded by values

$$\cos\left(A + \frac{B}{2}\right)$$

and $\cos(A+B)$.

If further refinement of accuracy in determining the location of a root on unit circle 11 is required, delay unit 172 delivers via line 200 and switch 178 to the input of half-angle generating unit 96 a signal representing

$$1 - \cos \frac{B}{2^n}$$

Half-angle generating unit 96 provides a signal representing

$$1 - \cos \frac{B}{2^{n+1}}$$

to input 214 of angle bisecting unit 150 via lines 102, 103. Angle bisecting unit 150 uses inputs received at input 214 to further bisect sector 15 whereby each half-sector is now reestablished as a newly-defined sector having an upper limit U_p and a lower limit L_p which is then bisected. Zero crossing is determined to be either in the lower half-sector (line 248 to select logic 190) or in the upper half-sector (line 246 to select logic unit 190) of one of the newly-defined sectors. Selector unit 192 proceeds with the half-sector containing a zero crossing as a next-newly-defined sector for the next bisecting iteration.

Figure 13-8 is a schematic diagram of an alternate embodiment of a waveform generating unit for use in the present invention.

In particular, Figure 13-8 illustrates a waveform generating unit for use in the present invention when angles A and B are small enough that leading nines in the quantities $\cos A$ or $\cos B$ indicate it is useful to use quantities $(1 - \cos A)$ or $(1 - \cos B)$ in their place to ensure that all digits used to represent the required values in root determination operations are significant digits. Thus, Figure 13-8 implements equation (13.28) to enable one to configure a waveform generator accurately for small angle increments for implementation of the present invention.

In Figure 13-8, a waveform generating unit 260 receives an input on a line 262 representing the quantity $(1 - \cos B)$. The input received via line 262 is provided via a line 264 to a leading zero detect unit 266 and, via a line 268, to a summing unit 270. Leading zero detect unit 266 generates an output on a line 272 representing the number m of leading zeroes detected in the input received via line 264. As described before regarding leading zero detection, leading zero detect unit 266 may be replaced by any scaling apparatus appropriate to the number format used. The signal representing m is applied via a line 274 to a scaling unit 276, and is provided via a line 278 to a scaling unit 280.

The input signal applied at input line 262 is applied to scaling unit 276 so that scaling unit 276 provides a signal on line 282 representing the signal received via line 262 multiplied times 2^m .

The output signal provided at output 284 of waveform generating unit 260 is a signal representing the quantity $1-\cos(A+B)$, and the output signal provided at output 286 is a signal representing the quantity $1-\cos A$. Delay unit 288 receives the signal provided at output 284 via a feedback line 290 so that the output provided on line 292 from delay unit 288 is a signal representing the quantity $1-\cos A$. That signal is provided to a delay unit 294, to a multiplier 296, and to output 286 via a line 311. Of course, delay units 288, 294 must be appropriately initialized to ensure proper operation of waveform generating unit 260.

Delay unit 294 generates a signal on an output line 298 representing the quantity $1-\cos(A-B)$, which signal is applied to a multiplier 300. Multiplier 300 effects a multiplication by the quantity -1 . Multiplier 296 multiplies the value received via line 282 (i.e., $2^m(1-\cos B)$) times the value received via line 292 (i.e., $1-\cos A$) and generates a signal on a line 302 representing the quantity $2^m(1-\cos A)(1-\cos B)$. Multiplier 303 multiplies the value received via line 302 by the quantity -1 , and produces on line 305 a signal representing the quantity $-2^m(1-\cos A)(1-\cos B)$.

Multiplier 300 generates a signal on a line 304 representing the quantity $-(1-\cos(A-B))$. The signal carried on line 304 is applied to a summing unit 306. The signal carried on line 305 is applied to scaling unit 280. Scaling unit 280 provides a signal on line 308 representing the signal received via line 302 divided by 2^m , so that scaling unit 280 generates on line 308 a signal representing the value $-(1-\cos A)(1-\cos B)$, which signal is applied to summing unit 270. The signal carried on line 292 representing $1-\cos A$ is also applied via a line 309 to summing unit 270. Summing unit 270 generates a signal on a line 310 representing the value $-(1-\cos A)(1-\cos B)+(1-\cos A)+(1-\cos B)$.

The signal carried on line 310 is applied to a scaling unit 312 to multiply the signal received via line 310 times 2. Scaling unit 312 generates a signal on a line 314 representing the value $2((1-\cos A)-(1-\cos A)(1-\cos B)+(1-\cos B))$. The signal carried on line 314 is applied to summing unit 306 so that summing

unit 306 generates on output line 284 a signal representing the quantity $2((1-\cos A)-(1-\cos A)(1-\cos B)+(1-\cos B)-(1-\cos(A-B)))$.

We know from equation (13.28) that the quantity represented by the signal on line 284 equals the value $1-\cos(A+B)$.

Figure 13-9 is a flow diagram illustrating the preferred embodiment of the method of the present invention. In Figure 13-9, the method begins with "Start" at block 320 and a counter n is set to 0 at block 322. The values $\cos A$ and $\cos(A+B)$ are calculated (or retrieved from storage, or otherwise determined) in block 324 and are employed in block 326 to determine a solution for the polynomial $G(X)$ for each of the two values ($\cos A = X_{n-1}$, and $\cos(A+B) = X_n$). By block 328 an inquiry is made whether a sign change occurred in the values of the polynomial $G(X)$ in the interval between X_{n-1} and X_n .

If no sign change has occurred, then no root is present in the interval then being investigated and the "No" branch 330 is taken to decision block 332 where a check is made whether all intervals on unit circle 11 (Figure 13-1a) have been checked.

If all intervals have been checked, the "Yes" branch 334 is taken to "END" 336. If not all intervals have been checked, then "No" branch 338 is taken from decision block 332, counter n is incremented by block 340 and fed back via branch 342 to block 324 for recalculation of a new interval X_{n-1} , X_n . If the response in decision block 328 regarding presence of a sign change in the interval being considered is affirmative, the method proceeds via "Yes" branch 344 to decision block 346 where inquiry is made whether greater accuracy is required in determining location of the root found in decision block 328. If no further accuracy is required, the "No" branch 348 is taken to a function block 350 which effects outputting the interval containing the root (X_{n-1} , X_n) and then proceeds via branch 352 to decision block 332 for determination whether all intervals on unit circle 11 have been checked. Subsequent decisions and operations occur as described above in connection with answers to the query posed by decision block 332.

If it is decided at decision block 346 that more accuracy is required, then "Yes" branch 354 is taken. A

counter p is set at block 356 and provided via branch 358 to block 360. In block 360, an upper limit U_p is set equal to X_n , a lower limit L_p is set equal to X_{n-1} , and a half-angle increment HAP is set equal to

$$1 - \cos \frac{B}{2^p},$$

initially $p = 1$.

These values are provided via a branch 362 to a block 364. Block 364 implements equation (13.22), where the value of upper limit U_p is equal to the value $\cos(A+B)$ and the value of lower limit L_p is equal to the value $\cos A$. The value M_p indicated in block 364 is the midpoint of the particular segment being addressed by block 364 and equals the value

$$\cos\left(A + \frac{B}{2}\right)$$

Using the values computed in blocks 360, 364, block 366 evaluates the polynomial $G(X)$ for those values and provides evaluation values to decision block 368.

Decision block 368 determines whether there is a sign change in the interval between lower limit L_p and midpoint M_p . If there is such a sign change, then the root of the polynomial $G(X)$ is in the lower half-sector in the interval L_p, M_p of the sector determined in block 324 and bounded by X_{n-1}, X_n . "Yes" branch 370 is taken so that the next iterative sector limits for possible subsequent bisecting of the sector are established in block 372 for a newly-defined sector: a newly-defined lower limit L_{p+1} being set at the original lower limit L_p , and a newly-defined upper limit U_{p+1} being set at the midpoint M_p . Those newly-defined limit values L_{p+1}, U_{p+1} are provided to decision block 374 and inquiry is made whether further accuracy is required. If no further accuracy is required, "No" branch 376 is taken to block 350 where the interval containing the root (now defined by newly-defined lower limit L_{p+1} and newly-defined upper limit U_{p+1}) is output and the method proceeds via branch 352 to query whether all intervals are checked

(decision block 332). Subsequent decisions and operations occur as previously described in connection with answers to the query posed by decision block 332.

If further accuracy is required, "Yes" branch 378 is taken to block 380 in which the half-angle increment HA_{p+1} for the next iteration of accuracy (i.e., next bisection of the newly-defined sector established in block 372) is calculated implementing equation (13.32). The newly calculated half-angle increment HA_{p+1} is provided via branch 382 to block 364 for calculating of a new midpoint M_p . Determination of evaluation values of the polynomial $G(X)$ is effected in block 366 for a new lower limit L_p (as determined in block 372), a new midpoint M_p (as determined in block 364), and a new upper limit U_p (as determined in block 372). The method may continue in this loop (from decision block 368, to block 372, to decision block 374, to block 380, to block 364, and to block 366) until the response to the query posed by decision block 374 is negative, indicating that sufficient accuracy had been achieved in root location determination, whence the method proceeds via "No" branch 376 from decision block 374 as previously described.

If decision block 368 determines that no sign change has occurred in the interval L_p, M_p , then "No" branch 384 is taken from decision block 368 to block 386. In block 386 a newly-defined lower limit L_{p+1} is set at midpoint M_p and a newly-defined upper limit U_{p+1} is set at upper limit U_p . Those newly-defined limits L_{p+1}, U_{p+1} delimit a newly-defined sector established by block 386 which is the upper half-sector of the sector defined in block 360. Block 386 provides these newly-defined limit values via a branch 388 to decision block 374 and the method proceeds thereafter as previously described.

It is to be understood that, while the detailed drawings and specific examples given describe preferred embodiments of the invention, they are for the purpose of illustration, that the apparatus and method of the invention are not limited to the precise details and conditions disclosed and that various changes may be made therein without departing from the spirit of the invention.

References

Related Patents

<u>4393272</u>	Jul., 1983	Itakura et al.	395/2.
<u>4975956</u>	Dec., 1990	Liu et al.	381/36.
<u>5012518</u>	Apr., 1991	Liu et al.	381/42.
<u>5027404</u>	Jun., 1991	Taguchi	395/2.
<u>5305421</u>	Apr., 1994	Li	395/2.
<u>5414796</u>	May., 1995	Jacobs et al.	395/2.
<u>5448680</u>	Sep., 1995	Kang et al.	395/2.

References

1. COMSIG 88. Southern African Conference on Communications and Signal Processings; Smith et al, *Line-spectrum pairs -a review*, pp. 7-11, Jun. 1988.
2. ICASSP * *: 1988 International on Acoustics, Speech, and Signal processing; Paliwal, *A Study of Line Spectrum pair frequencies for speech recognition*, pp. 485-488, Apr. 1988.

Chapter 14

Apparatus for Generating a Sinusoidal Output Signal

Abstract

This chapter is based on the invention patented under American patent 5,189,381 (S. Asghar, A. Linz, Feb. 1993).

An apparatus is disclosed for generating an output signal in response to an input signal having a variable input level. The output signal has an output frequency which varies to represent variations in the input level. The apparatus comprises a processing circuit for implementing a digital signal processing algorithm to generate control signals in response to the input signal, which control signals are representative of the variable input level, and an oscillator circuit for generating the output signal in response to the control signals. The control signals include a first control signal representing a first output frequency, a second control signal representing a second output frequency, and at least one intermediate control signal representing output frequencies intermediate the first and second output frequencies. The processing circuit generates at least one intermediate control signal whenever the output signal varies between the first and the second output frequencies which serves to "soften" the transition between the first and second output frequencies and reduces noise which usually is generated by such sudden changes in output frequency.

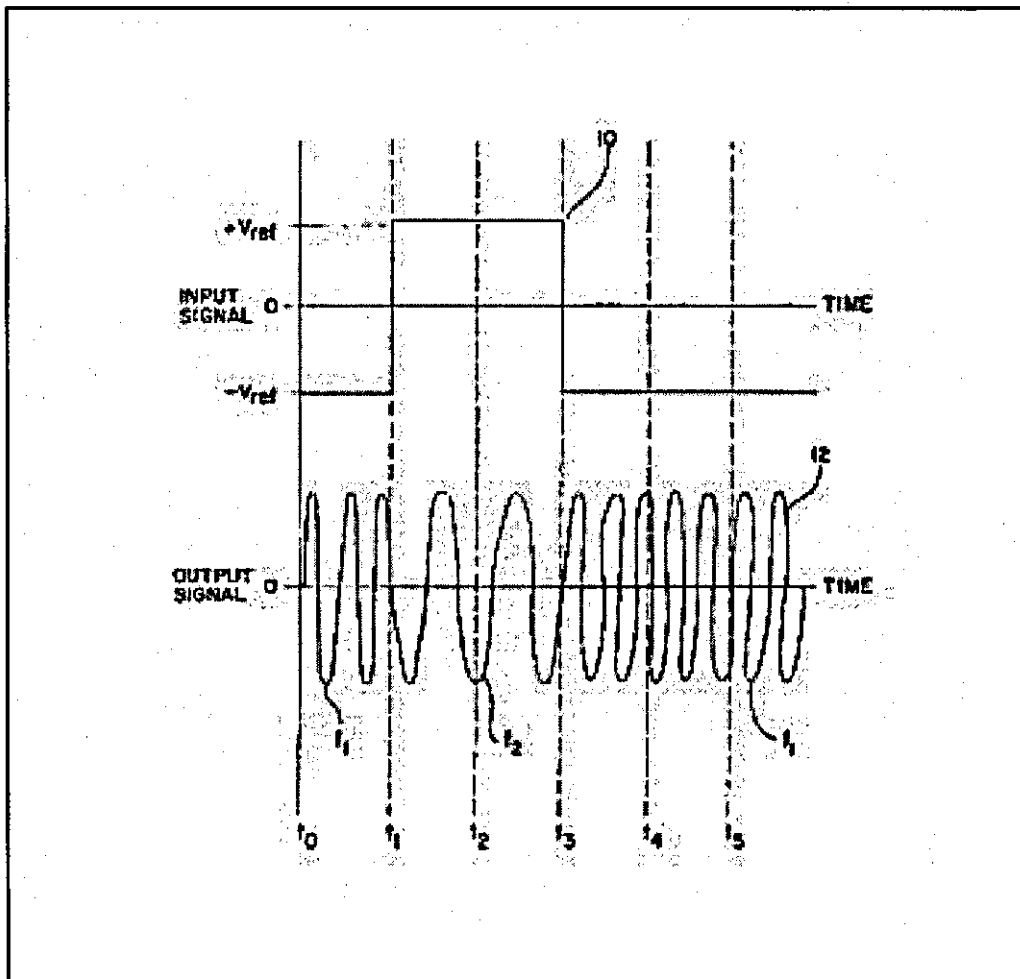
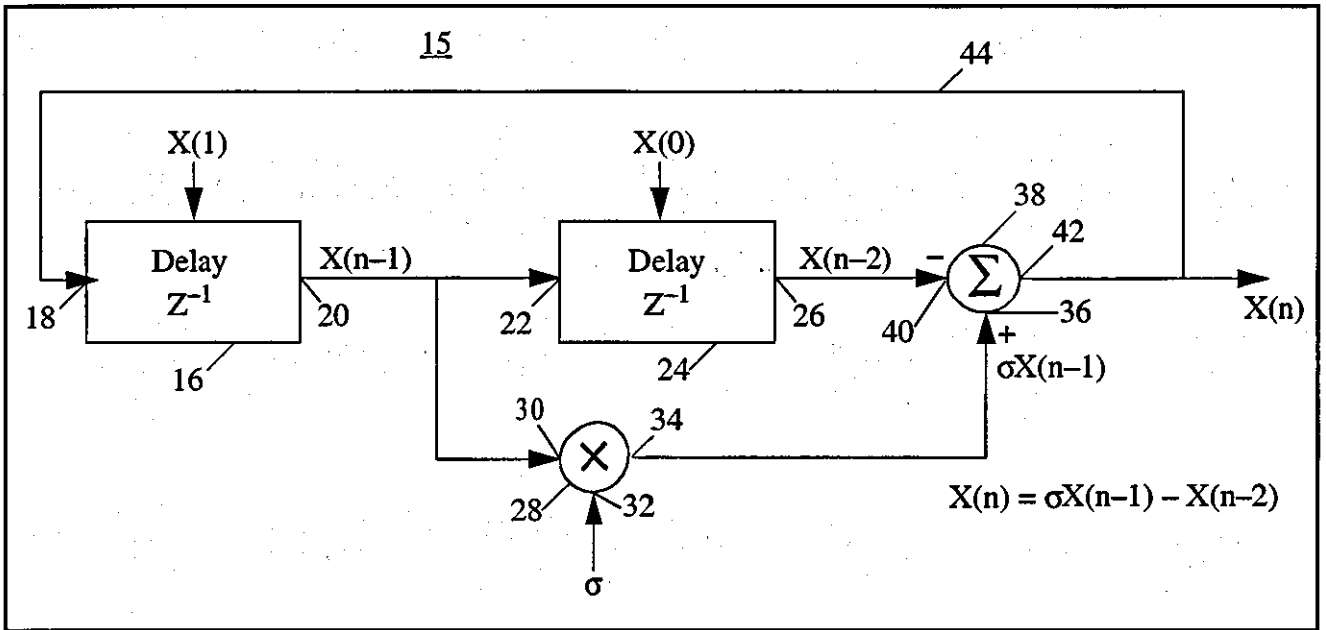


Figure 14-1

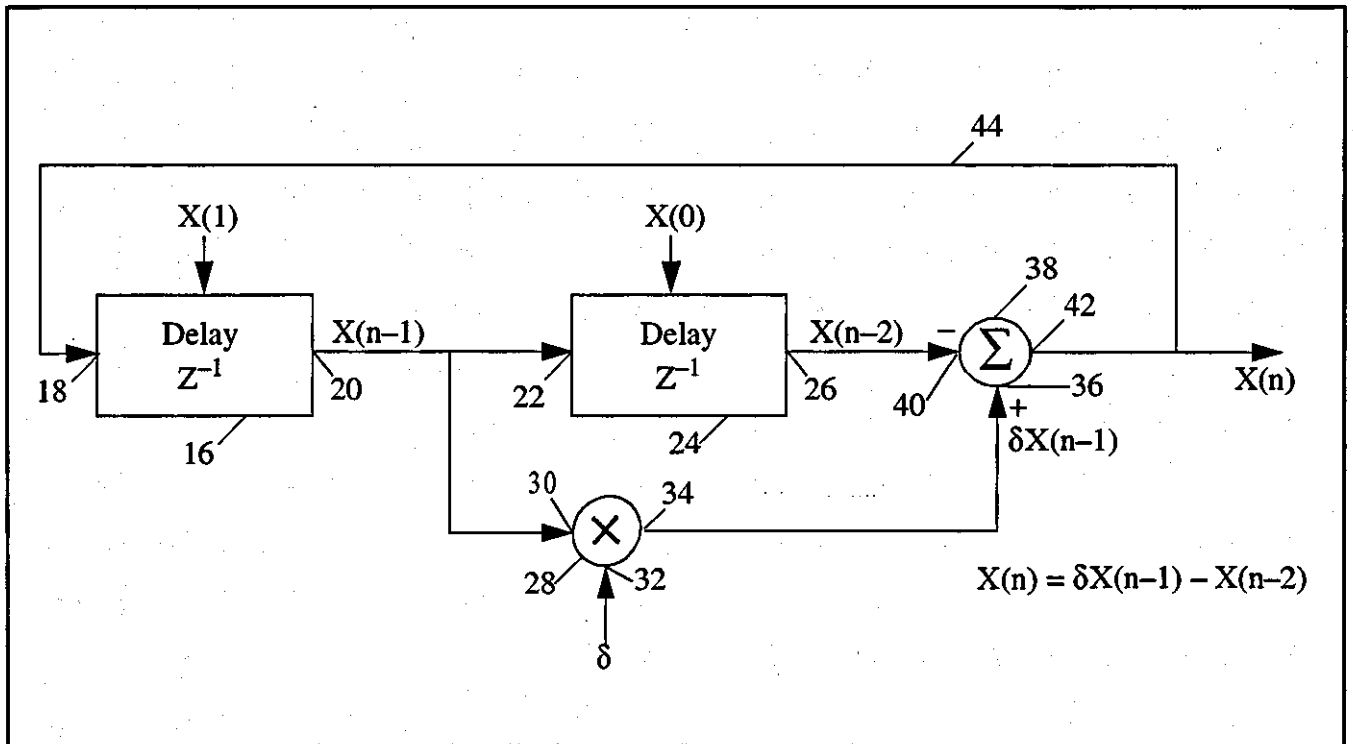


Figure 14-2

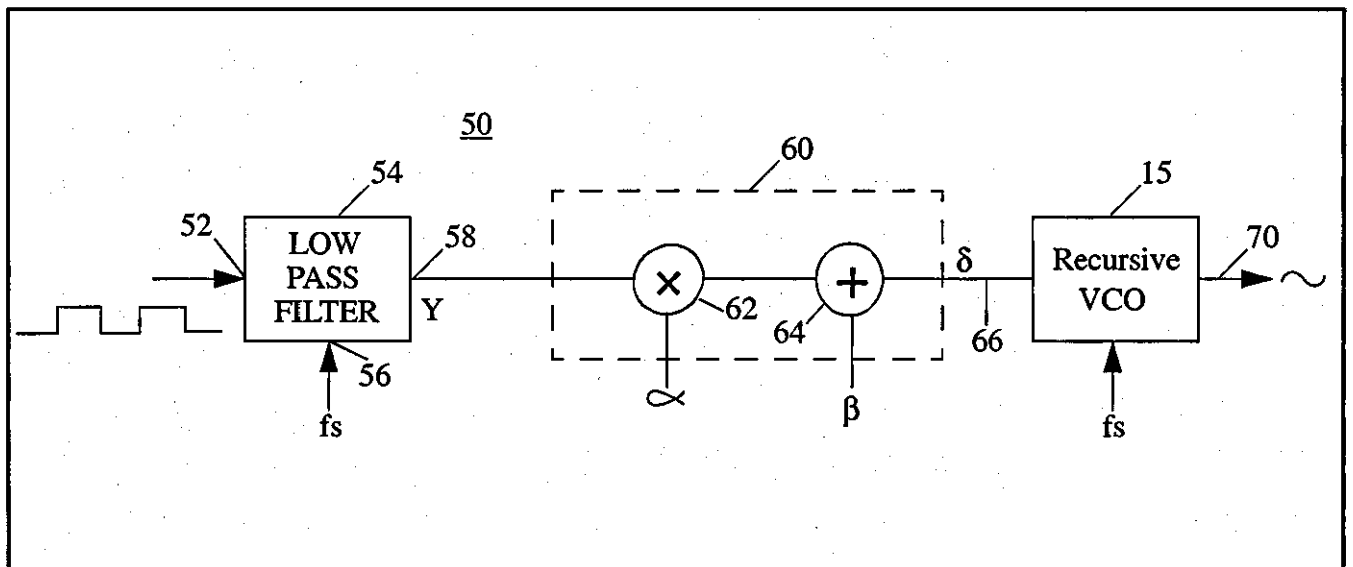


Figure 14-3

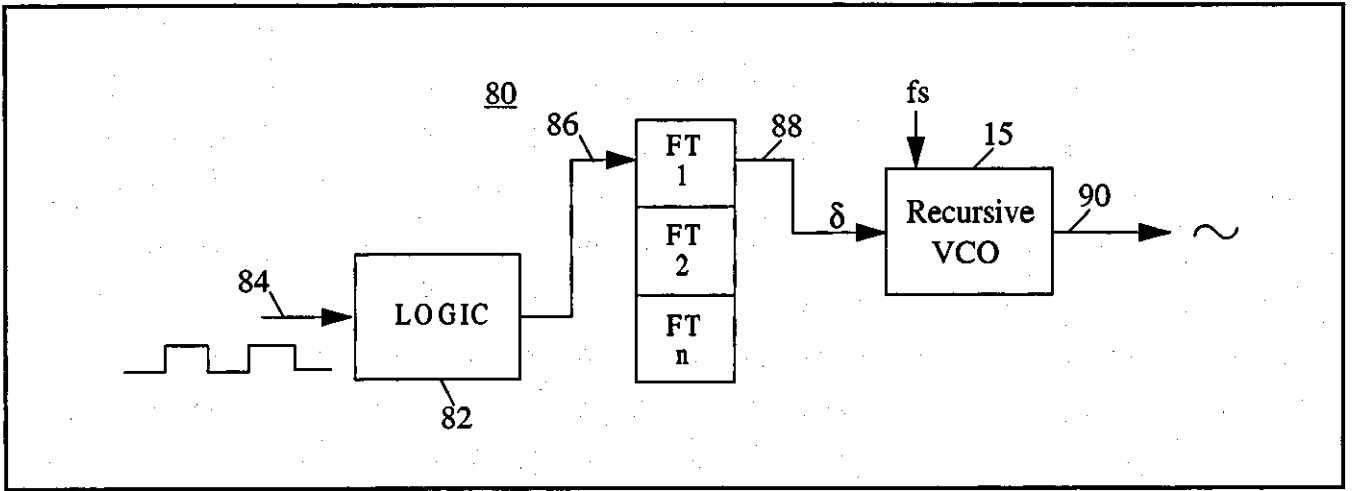


Figure 14-4

BACKGROUND OF THE INVENTION

Generation of sinusoidal output signals having a variable output frequency responsive to an input signal having a variable input level is a well-known method of effecting digital communications. Common applications of such a communications technique are frequency shift keying (FSK), and derivatives of FSK such as minimum shift keying (MSK) and Gaussian minimum shift keying (GMSK).

Generation of such FSK signals is sometimes effected by circuitry involving analog devices, and such analog devices are generally bulkier and less stable (especially in temperature-varying environments) than digital devices.

In apparatuses generating FSK signals employing digital devices, sudden frequency shifts in response to changes in signal levels of input signals generally involve undesirable noise during frequency transitions.

It is desirable that stable, generally compact digital devices be employed in digital communications effecting FSK operations without generating disruptive noise during frequency transitions.

SUMMARY OF THE INVENTION

The invention is an apparatus for generating an output signal in response to an input signal having a variable input level. The output signal has an output frequency which varies to represent variations in the input level. The apparatus comprises a processing circuit for implementing a digital signal processing algorithm to generate a plurality of control signals in response to the input signal, which plurality of control signals is representative of the variable input level, and an oscillator circuit for generating the output signal in response to the plurality of control signals. The plurality of control signals includes a first control signal representing a first output frequency of the output signal, a second control signal representing a second output frequency of the output signal, and at least one intermediate control signal representing

output frequencies of the output signal intermediate the first output frequency and the second output frequency. Preferably, each respective intermediate control signal is associated with a respective intermediate frequency.

The processing circuit generates at least one of the intermediate control signals whenever the output signal varies from the first output frequency to the second output frequency or whenever the output signal varies from the second output frequency to the first output frequency. Such intermediate frequency generation on every occasion of frequency shift by the oscillator circuit serves to "soften" the transition between the first and second output frequencies and reduces noise which usually is generated by such sudden changes in output frequency.

Preferably, the oscillator circuit comprises a first delay circuit which receives a first signal and generates a second signal, which second signal is delayed one clock period with respect to the first signal; and a second delay circuit which receives the second signal from the first delay circuit and generates a third signal, which third signal is delayed one clock period with respect to the second signal. Further included in the oscillator circuit is a multiplying circuit for multiplying a signal by a multiplier. The multiplying circuit receives the second signal from the first delay circuit, receives a multiplier at a multiplier input, and generates a multiplier output signal which comprises the second signal multiplied by the multiplier. The oscillator circuit further includes a differential circuit which receives the third signal from the second delay circuit and receives the multiplier output signal from the multiplying circuit, and generates a sinusoidal output which is the difference between the multiplier output signal less the third signal; the sinusoidal output signal is also provided in a feedback loop to the first delay circuit and is received by the first delay circuit as the first signal so that the oscillator circuit operates as a recursive oscillator.

In the preferred embodiment, the multiplier received by the multiplier circuit comprises a constant which is representative of the varying input level of the input signal. The multiplier thus is employed as a control signal and may be generated by a control signal generator. Such a control signal generator may, in

its preferred embodiment, include a scaling circuit for receiving the input signal and scaling the input signal according to at least one scaling factor to produce the control signal (i.e., the multiplier). In an alternate embodiment, such a control signal generator may comprise a logic circuit receiving the input signal and an array of digital filters. The logic circuit selects one digital filter of the array of digital filters in response to the input signal level. The selected digital filter receives the input signal from the logic circuit and filters the input signal to produce a selected control signal appropriate to cause the oscillator circuit to generate its sinusoidal output at the correct frequency to represent the respective input signal level at hand. Digital filters are provided for producing appropriate control signals to yield intermediate output frequencies to reduce noise during output signal frequency excursions.

It is, therefore, an object of the present invention to provide an apparatus for generating a sinusoidal output signal in response to an input signal having a variable input level which employs digital signal processing technology to effect such generation.

It is another object of the present invention to provide an apparatus for generating a sinusoidal output signal in response to an input signal having a variable input level which employs digital signal processing technology to effect such generation without generating disruptive noise during frequency transitions.

Further objects and features of the present invention will be apparent from the following specification and claims when considered in connection with the accompanying drawings illustrating the preferred embodiment of the invention.

DETAILED DESCRIPTION OF THE DRAWINGS

Figure 14-1 illustrates the relationship between an output signal and an input signal as contemplated by the operation of the present invention.

Figure 14-2 illustrates the preferred embodiment of an oscillator circuit employed in the present invention.

Figure 14-3 illustrates the preferred embodiment of the present invention.

Figure 14-4 illustrates an alternate embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

A typical frequency modulation scheme encountered in digital communications, and commonly encountered in frequency shift keyed (FSK) communication systems, is illustrated in Figure 14-1. In Figure 14-1, a digital input signal is illustrated varying over time between $-V_{\text{ref}}$ and $+V_{\text{ref}}$ so that input signal 10 is at a level $-V_{\text{ref}}$ from time t_0 to time t_1 ; shifts to a level $+V_{\text{ref}}$ at time t_1 , maintaining the level $+V_{\text{ref}}$ until time t_3 ; and shifts to level $-V_{\text{ref}}$ at time t_3 , maintaining level $-V_{\text{ref}}$ from time t_3 through time t_5 .

Output signal 12 is modulated according to the level of input signal 10. Thus, output signal 12 has a frequency f_1 between time t_0 and time t_1 when input signal 10 is at value $-V_{\text{ref}}$; output signal 12 has a frequency f_2 during the interval between time t_1 and time t_3 when input signal 10 has value $+V_{\text{ref}}$; and output signal 12 is transmitted at frequency f_1 during the time interval t_3 through t_5 when input signal 10 again has value $-V_{\text{ref}}$. In such manner, the frequency of output signal 12 represents the signal level of input signal 10.

The apparatus of the present invention is based upon the following trigonometric relations: A frequency function may be defined as:

$$\omega(f_1 f_s) = 2\pi \frac{f}{f_s} ;$$

where f_s is a sampling rate and f is a desired frequency.

If the expression $x(n) = A \sin(n\omega)$ represents a sinusoidal wave output of the apparatus, then the following delayed sequences of that sinusoidal output signal can be defined: Current t-time output:

$$x(n) = A \sin(n\omega) \quad [1]$$

Output, one time interval later:

$$x(n-1) = A \sin(n\omega - \omega)$$

$$x(n-1) = A \sin(n\omega) \cos(\omega) - A \cos(n\omega) \sin(\omega) \quad [2]$$

Output, two time intervals later:

$$x(n-2) = A \sin(n\omega - 2\omega)$$

$$x(n-2) = A \sin(n\omega) \cos(2\omega) - A \cos(n\omega) \sin(2\omega) \quad [3]$$

Multiplying Eq. [2] by $2\cos(\omega)$ yields:

$$[2\cos(\omega)][x(n-1)] = [A \sin(n\omega) \cos(\omega)][2\cos(\omega)] - [A \cos(n\omega) \sin(\omega)][2\cos(\omega)]$$

Using the trigonometric identity $\sin(2\omega) = 2\sin(\omega)\cos(\omega)$ yields:

$$[2\cos(\omega)][x(n-1)] = [2A \sin(n\omega) \cos^2(\omega)] - [A \cos(n\omega) \sin(2\omega)]$$

Subtracting Eq. [3] from Eq. [2] yields:

$$[2\cos(\omega)][x(n-1)] - x(n-2) = 2A \sin(n\omega) \cos^2(\omega) - A \cos(n\omega) \sin(2\omega) -$$

$$A \sin(n\omega) \cos(2\omega) + A \cos(n\omega) \sin(2\omega)$$

Reducing the above expression yields:

$$2\cos(\omega)x(n-1) - x(n-2) = A \sin(n\omega)[2\cos^2(\omega) - \cos(2\omega)]$$

Recalling that $x(n) = A \sin(n\omega)$, the expression may be rewritten:

$$2\cos(\omega)x(n-1) - x(n-2) = x(n)[2\cos^2(\omega) - \cos(2\omega)]$$

Using the trigonometric identity $\cos^2(\omega) = \frac{1 + \cos(2\omega)}{2}$ yields:

$$2\cos(\omega)x(n-1) - x(n-2) = x(n)$$

This expression may be written in the form:

$$x(n) = \delta x(n-1) - x(n-2)$$

where $\delta = 2\cos(\omega)$, and δ is a constant for a given frequency, ω .

Thus, the current output $x(n)$ is recursively related to outputs one time interval later ($x(n-1)$) and two time intervals later ($x(n-2)$), and the recursive nature of output signal $x(n)$ can be used to generate sine waves using a sampling frequency f_s that has no specific relationship to the desired frequency (f) to be generated, except that the sampling frequency must satisfy Nyquist's sampling theorem (i.e., $f_s \geq 2f$).

This recursive relationship facilitates simplified sinusoidal output signal generation. Distinct advantages are especially provided in the implementation of such recursive signal generation since prior art apparatuses employing look-up tables (read-only memories; ROM) to store wave-form samples, with the number of samples required to be stored being dependent upon the relationship between the desired frequency and the sampling rate. In contrast, the recursive capability of the present invention requires only two storage locations, regardless of what frequency is desired. Specifically, the recursive capability of the present invention facilitates generation of a sinusoidal output signal using two successive signal samples [$x(n-1)$, and $x(n-2)$]. Thus, two storage locations are all that are required for employing the recursive relationship: one storage location for $x(n-1)$ and one storage location for $x(n-2)$.

Figure 14-2 is an illustration of the preferred embodiment of a recursive oscillating circuit apparatus employing the above-described relationships to advantage. In Figure 14-2, a first delay circuit 16 receives a first signal at an input 18 and generates a once-delayed output signal at an output 20. The once-delayed output signal is delayed one time period from the first signal received at input 18. The once-delayed output signal from output 20 of first delay circuit 16 is received as an input signal by a second delay circuit

24 at an input 22. Second delay circuit 24 produces a twice-delayed output signal at its output 26, which twice-delayed output signal is delayed one clock period from the once-delayed output signal received at its input 22.

The once-delayed output signal from first delay circuit 16 is also provided as an input to a multiplier circuit 28 at an input 30. Multiplier circuit 28 multiplies the once-delayed output signal received from first delay circuit 16 by a multiplier δ received at a multiplier input 32, and provides a multiplier output signal at its output 34. The multiplier output signal is provided to a positive node 36 of a summing unit 38. Summing unit 38 also receives at its negative node 40 the twice-delayed output signal from output 26 of second delay circuit 24. Thus, summing unit 38 provides at its output 42 a difference signal representing the difference between the multiplier output signal received at its input node 36, less the twice-delayed output signal received at its negative node 40. This difference signal is the sinusoidal output signal $x(n)$.

Sinusoidal output signal $x(n)$ is provided through a feedback line 44 to first delay circuit 16 and comprises the first signal received by first delay circuit 16 at its input 18. Thus, there is provided a recursive oscillator circuit 15 which provides an output $x(n)$ determined by a multiplier δ received at multiplier input 32. Initial conditions are established for oscillator circuit 15 by an input $x(1)$ to first delay circuit 16 and by an input $x(0)$ to second delay circuit 24. Preferably, $x(1)=A\sin\omega_0$ and $x(0)=0$, where A signifies amplitude of the signal $x(1)$ and ω_0 signifies an initial frequency

$$\left(\omega_0 = \frac{2\pi f_0}{f_s} \right)$$

The employment of recursive oscillator circuit 15 as illustrated in Figure 14-2 in a communications system is illustrated, in its preferred embodiment, in Figure 14-3. In Figure 14-3, a communications system 50 is illustrated receiving a digital signal input at an input 52 to a low pass filter (or signal synthe-

sizer) 54. Low pass filter 54 employs a sampling frequency f_s received at an input 56. A signal "y" is generated from low pass filter 54 at an output 58 and passed to a scaling circuit 60. Scaling circuit 60 comprises a multiplier unit 62 and an adder unit 64 to which output signal "y" is successively applied. Scaling circuit 60 applies scaling factors α and β to signal y to produce an output δ on line 66 such that $\delta = \alpha y + \beta$. The scaled output δ is applied to recursive oscillator 15 and recursive oscillator 15 generates an output signal at line 70 in a manner as described in connection with Figure 14-2. Thus, the output signal at line 70 is a sinusoidal output signal which is defined by control signal δ which is received by recursive oscillator 15 via line 66 from scaling circuit 60.

An alternate embodiment of a communications system employing a recursive oscillator 15 of the type illustrated in Figure 14-2 is illustrated in Figure 14-4. In Figure 14-4, a communications system 80 is illustrated comprising a logic circuit 82 which receives a digital input signal at an input 84. Depending upon the signal level of the input signal received at input 84, logic circuit 82 selects an appropriate filter among an array of filters FT1, FT2, . . . , FTn for receiving a representation of the input signal on line 86. The selected appropriate filter, for example filter FT1 in Figure 14-4, filters the representative input signal received on line 86 to produce a control signal δ on a line 88 for provision to a recursive oscillator 15. Selection of the appropriate filter FT1, FT2, . . . , FTn by logic circuit 82 is preprogrammed in logic circuit 82 for effecting generation of a control signal δ to facilitate effecting noise-free transition of the frequency of an output signal produced at output 90 to accurately reflect deviations in the signal level of the input signal received at input 84 of logic circuit 82.

References

Related Patents

4573025

Feb., 1986

McKinzie, III

332/102.

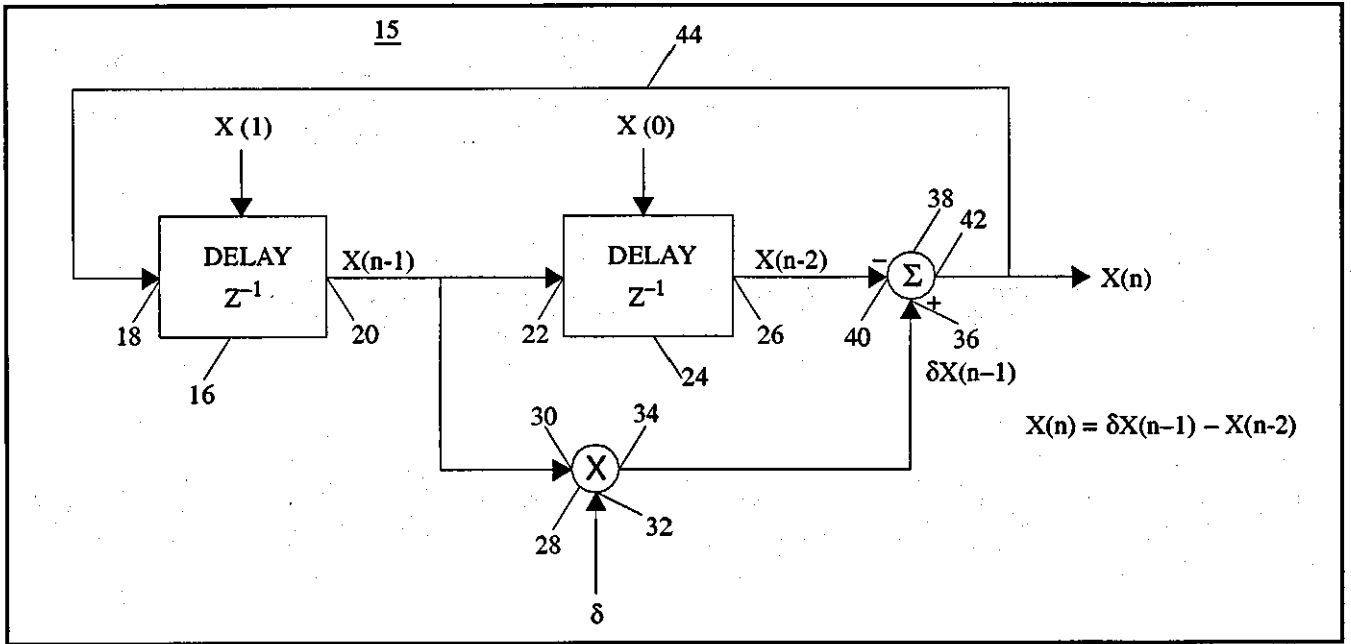
Chapter 15

Frequency Controlled Recursive Oscillator Having Sinusoidal Output

Abstract

This chapter is based on the invention patented under American patent 5,204,642 (S. Asghar, A. Linz, April 1993).

An apparatus is disclosed for generating an output signal in response to an input signal having a variable input level. The output signal has an output frequency which varies to represent variations in the input level. The apparatus comprises a processing circuit for implementing a digital signal processing algorithm to generate control signals in response to the input signal, which control signals are representative of the variable input level, and an oscillator circuit for generating the output signal in response to the control signals. The control signals include a first control signal representing a first output frequency, a second control signal representing a second output frequency, and at least one intermediate control signal representing output frequencies intermediate the first and second output frequencies. The processing circuit generates at least one intermediate control signal whenever the output signal varies between the first and the second output frequencies which serves to "soften" the transition between the first and second output frequencies and reduces noise which usually is generated by such sudden changes in output frequency.



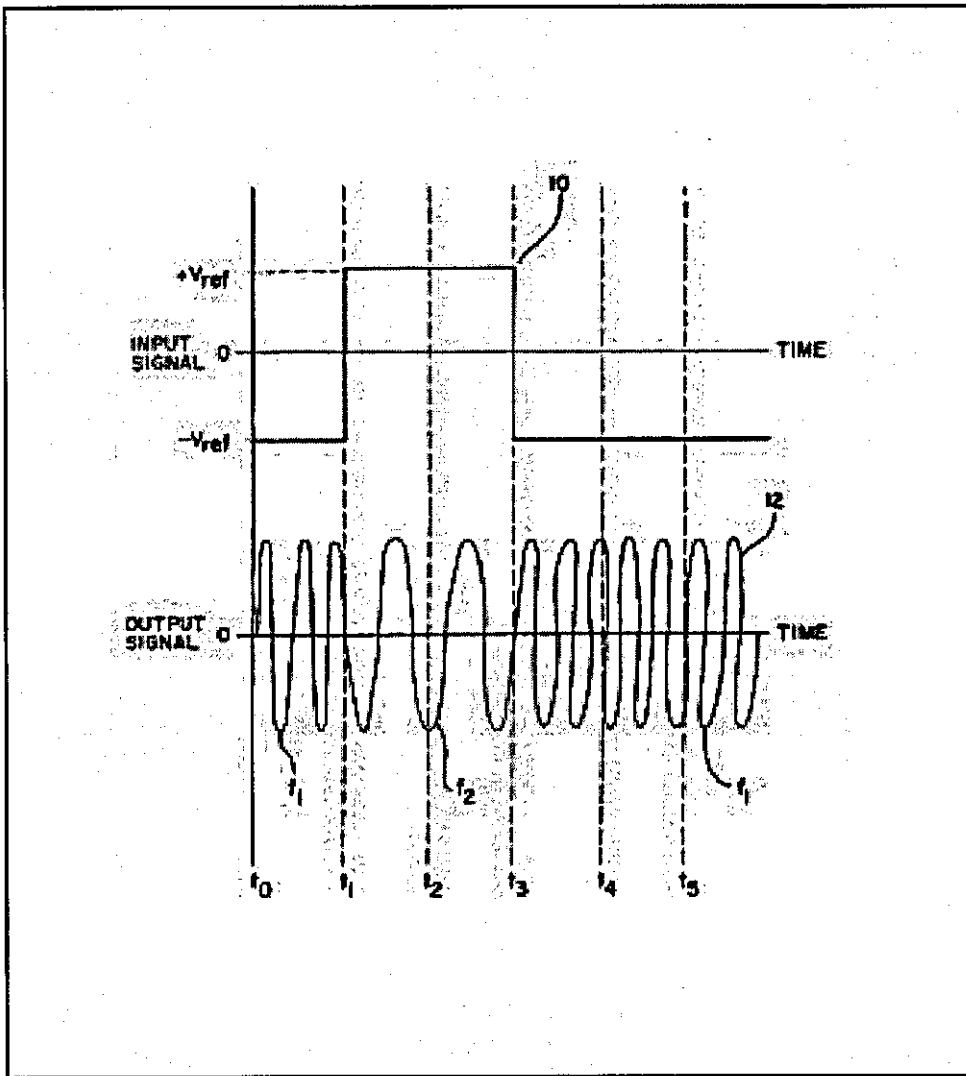


Figure 15-1

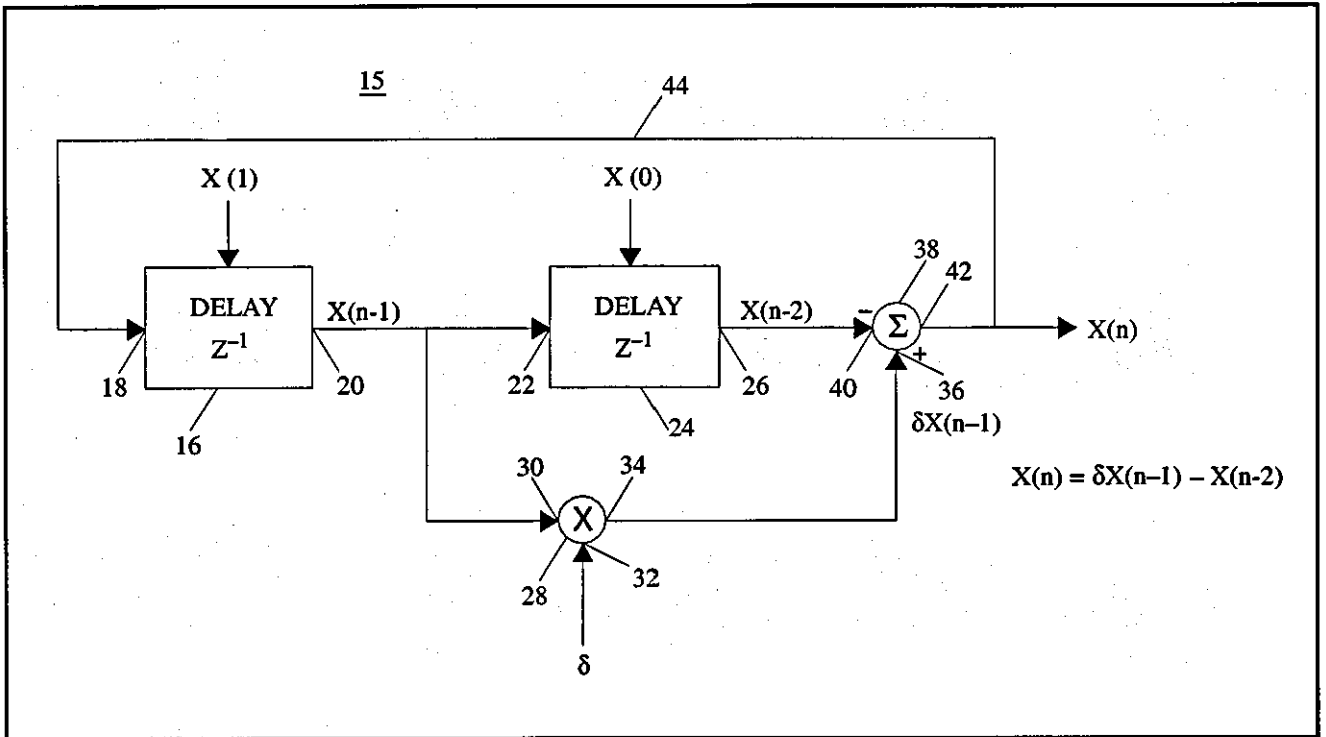


Figure 15-2

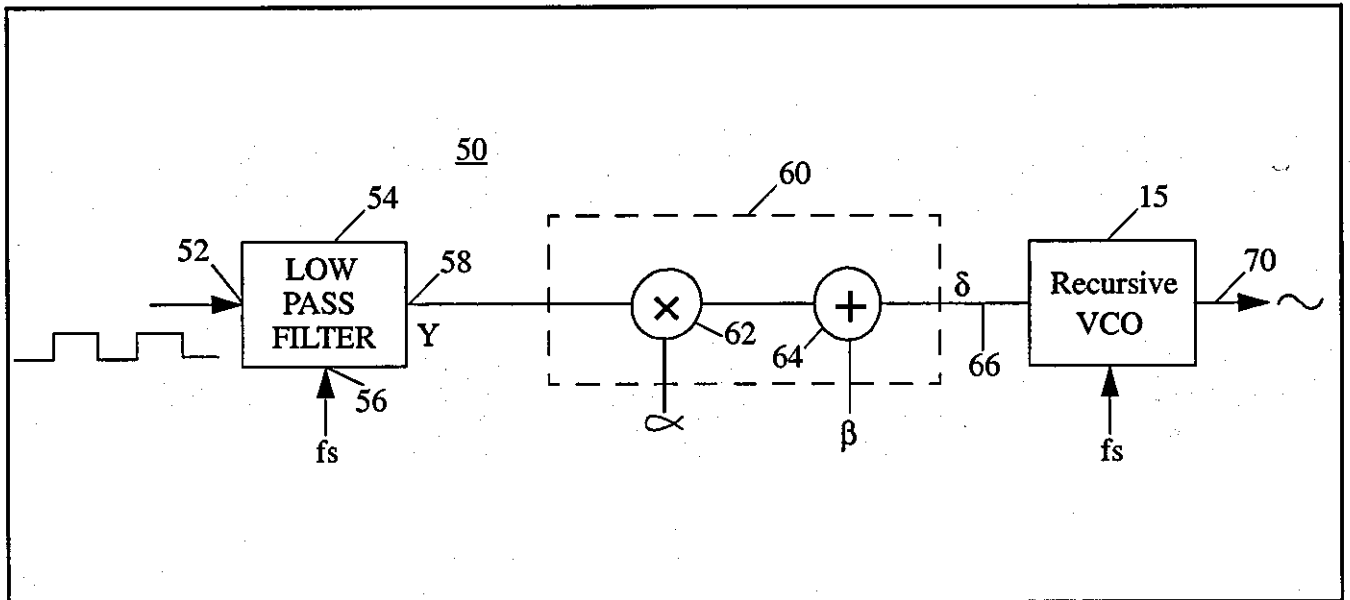


Figure 15-3

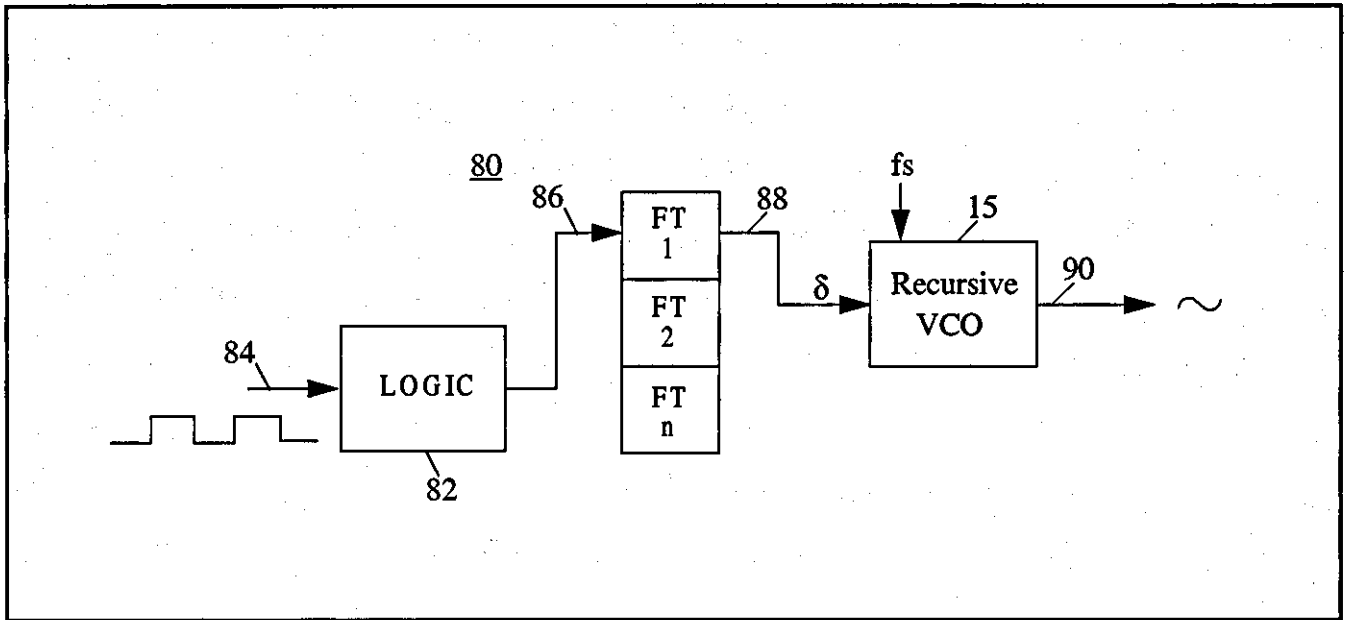


Figure 15-4

BACKGROUND OF THE INVENTION

Generation of sinusoidal output signals having a variable output frequency responsive to an input signal having a variable input level is a well-known method of effecting digital communications. Common applications of such a communications technique are frequency shift keying (FSK), and derivatives of FSK such as minimum shift keying (MSK) and Gaussian minimum shift keying (GMSK).

Generation of such FSK signals is sometimes effected by circuitry involving analog devices, and such analog devices are generally bulkier and less stable (especially in temperature-varying environments) than digital devices.

In apparatuses generating FSK signals employing digital devices, sudden frequency shifts in response to changes in signal levels of input signals generally involve undesirable noise during frequency transitions.

It is desirable that stable, generally compact digital devices be employed in digital communications effecting FSK operations without generating disruptive noise during frequency transitions.

SUMMARY OF THE INVENTION

The invention is an apparatus for generating an output signal in response to an input signal having a variable input level. The output signal has an output frequency which varies to represent variations in the input level. The apparatus comprises a processing circuit for implementing a digital signal processing algorithm to generate a plurality of control signals in response to the input signal, which plurality of control signals is representative of the variable input level, and an oscillator circuit for generating the output signal in response to the plurality of control signals. The plurality of control signals includes a first control signal representing a first output frequency of the output signal, a second control signal representing a second output frequency of the output signal, and at least one intermediate control signal representing output frequencies of the output signal intermediate the first output frequency and the second output frequency. Preferably, each respective intermediate control signal is associated with a respective intermediate frequency.

The processing circuit generates at least one of the intermediate control signals whenever the output signal varies from the first output frequency to the second output frequency or whenever the output signal varies from the second output frequency to the first output frequency. Such intermediate frequency generation on every occasion of frequency shift by the oscillator circuit serves to "soften" the transition between the first and second output frequencies and reduces noise which usually is generated by such sudden changes in output frequency.

Preferably, the oscillator circuit comprises a first delay circuit which receives a first signal and generates a second signal, which second signal is delayed one clock period with respect to the first signal; and a second delay circuit which receives the second signal from the first delay circuit and generates a third signal, which third signal is delayed one clock period with respect to the second signal. Further included in the oscillator circuit is a multiplying circuit for multiplying a signal by a multiplier. The multiplying

circuit receives the second signal from the first delay circuit, receives a multiplier at a multiplier input, and generates a multiplier output signal which comprises the second signal multiplied by the multiplier. The oscillator circuit further includes a differential circuit which receives the third signal from the second delay circuit and receives the multiplier output signal from the multiplying circuit, and generates a sinusoidal output which is the difference between the multiplier output signal less the third signal; the sinusoidal output signal is also provided in a feedback loop to the first delay circuit and is received by the first delay circuit as the first signal so that the oscillator circuit operates as a recursive oscillator.

In the preferred embodiment, the multiplier received by the multiplier circuit comprises a constant which is representative of the varying input level of the input signal. The multiplier thus is employed as a control signal and may be generated by a control signal generator. Such a control signal generator may, in its preferred embodiment, include a scaling circuit for receiving the input signal and scaling the input signal according to at least one scaling factor to produce the control signal (i.e., the multiplier). In an alternate embodiment, such a control signal generator may comprise a logic circuit receiving the input signal and an array of digital filters. The logic circuit selects one digital filter of the array of digital filters in response to the input signal level. The selected digital filter receives the input signal from the logic circuit and filters the input signal to produce a selected control signal appropriate to cause the oscillator circuit to generate its sinusoidal output at the correct frequency to represent the respective input signal level at hand. Digital filters are provided for producing appropriate control signals to yield intermediate output frequencies to reduce noise during output signal frequency excursions.

It is, therefore, an object of the present invention to provide an apparatus for generating a sinusoidal output signal in response to an input signal having a variable input level which employs digital signal processing technology to effect such generation.

It is another object of the present invention to provide an apparatus for generating a sinusoidal output signal in response to an input signal having a variable input level which employs digital signal processing

technology to effect such generation without generating disruptive noise during frequency transitions.

Further objects and features of the present invention will be apparent from the following specification and claims when considered in connection with the accompanying drawings illustrating the preferred embodiment of the invention.

DETAILED DESCRIPTION OF THE DRAWINGS

Figure 15-1 illustrates the relationship between an output signal and an input signal as contemplated by the operation of the present invention.

Figure 15-2 illustrates the preferred embodiment of an oscillator circuit employed in the present invention.

Figure 15-3 illustrates the preferred embodiment of the present invention.

Figure 15-4 illustrates an alternate embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

A typical frequency modulation scheme encountered in digital communications, and commonly encountered in frequency shift keyed (FSK) communication systems, is illustrated in Figure 15-1. In Figure 15-1, a digital input signal is illustrated varying over time between $-V_{\text{ref}}$ and $+V_{\text{ref}}$ so that input signal 10 is at a level $-V_{\text{ref}}$ from time t_0 to time t_1 ; shifts to level $+V_{\text{ref}}$ at time t_1 , maintaining the level $+V_{\text{ref}}$ until time t_3 ; and shifts to level $-V_{\text{ref}}$ at time t_3 , maintaining level $-V_{\text{ref}}$ from time t_3 through time t_5 .

Output signal 12 is modulated according to the level of input signal 10. Thus, output signal 12 has a frequency f_1 between time t_0 and time t_1 when input signal 10 is at value $-V_{\text{ref}}$; output signal 12 has a frequency f_2 during the interval between time t_1 and time t_3 when input signal 10 has value $+V_{\text{ref}}$; and output signal 12 is transmitted at frequency f_1 during the time interval t_3 through t_5 when input signal 10

again has value $-V_{ref}$. In such manner, the frequency of output signal 12 represents the signal level of input signal 10.

The apparatus of the present invention is based upon the following trigonometric relations: A frequency function may be defined as:

$$\omega(f f_s) = 2\pi \frac{f}{f_s}$$

where f_s is a sampling rate and f is a desired frequency.

If the expression $x(n) = A \sin(n\omega)$ represents a sinusoidal wave output of the apparatus, then the following delayed sequences of that sinusoidal output signal can be defined: Current t-time output:

$$x(n) = A \sin(n\omega) \tag{1}$$

Output, one time interval later:

$$\begin{aligned} x(n-1) &= A \sin(n\omega - \omega) \\ x(n-1) &= A \sin(n\omega) \cos(\omega) - A \cos(n\omega) \sin(\omega) \end{aligned} \tag{2}$$

Output, two time intervals later:

$$\begin{aligned} x(n-2) &= A \sin(n\omega - 2\omega) \\ x(n-2) &= A \sin(n\omega) \cos(2\omega) - A \cos(n\omega) \sin(2\omega) \end{aligned} \tag{3}$$

Multiplying Eq. [2] by $2\cos(\omega)$ yields:

$$[2\cos(\omega)][x(n-1)] = [A \sin(n\omega) \cos(\omega)][2\cos(\omega)] - [A \cos(n\omega) \sin(\omega)][2\cos(\omega)]$$

Using the trigonometric identity $\sin(2\omega) = 2\sin(\omega)\cos(\omega)$ yields:

$$[2\cos(\omega)][x(n-1)] = [2A \sin(n\omega) \cos^2(\omega)] - [A \cos(n\omega) \sin(2\omega)]$$

Subtracting Eq. [3] from Eq. [2] yields:

$$\begin{aligned} [2\cos(\omega)][x(n-1)] - x(n-2) &= 2A \sin(n\omega) \cos^2(\omega) - A \cos(n\omega) \sin(2\omega) - A \sin(n\omega) \cos(2\omega) + \\ &A \cos(n\omega) \sin(2\omega) \end{aligned}$$

Reducing the above expression yields:

$$2\cos(\omega)x(n-1) - x(n-2) = A\sin(n\omega)[2\cos^2(\omega) - \cos(2\omega)]$$

Recalling that $x(n) = A\sin(n\omega)$, the expression may be written:

$$2\cos(\omega)x(n-1) - x(n-2) = x(n)[2\cos^2(\omega) - \cos(2\omega)]$$

Using the trigonometric identity $\cos^2(\omega) = \frac{1 + \cos(2\omega)}{2}$

$$2\cos(\omega)x(n-1) - x(n-2) = x(n)$$

This expression may be written in the form:

$$x(n) = \delta x(n-1) - x(n-2)$$

where $\delta = 2\cos(\omega)$, and δ is a constant for a given frequency, ω .

Thus, the current output $x(n)$ is recursively related to outputs one time interval later ($x(n-1)$) and two time intervals later ($x(n-2)$), and the recursive nature of output signal $x(n)$ can be used to generate sine waves using a sampling frequency f_s that has no specific relationship to the desired frequency (f) to be generated, except that the sampling frequency must satisfy Nyquist's sampling theorem (i.e., $f_s \geq 2f$).

This recursive relationship facilitates simplified sinusoidal output signal generation. Distinct advantages are especially provided in the implementation of such recursive signal generation since prior art apparatuses employing lookup tables (read-only memories; ROM) to store wave-form samples, with the number of samples required to be stored being dependent upon the relationship between the desired frequency and the sampling rate. In contrast, the recursive capability of the present invention requires only two storage locations, regardless of what frequency is desired. Specifically, the recursive capability of the present invention facilitates generation of a sinusoidal output signal using two successive signal samples [$x(n-1)$, and $x(n-2)$]. Thus, two storage locations are all that are required for employing the recursive relationship: one storage location for $x(n-1)$ and one storage location for $x(n-2)$.

Figure 15-2 is an illustration of the preferred embodiment of a recursive oscillating circuit apparatus employing the above described relationships to advantage. In Figure 15-2, a first delay circuit 16 receives a first signal at an input 18 and generates a once-delayed output signal at an output 20. The once-delayed

output signal is delayed one time period from the first signal received at input 18. The once-delayed output signal from output 20 of first delay circuit 16 is received as an input signal by a second delay circuit 24 at an input 22. Second delay circuit 24 produces a twice-delayed output signal at its output 26, which twice-delayed output signal is delayed one clock period from the once-delayed output signal received at its input 22.

The once-delayed output signal from first delay circuit 16 is also provided as an input to a multiplier circuit 28 at an input 30. Multiplier circuit 28 multiplies the once-delayed output signal received from first delay circuit 16 by a multiplier δ received at a multiplier input 32, and provides a multiplier output signal at its output 34. The multiplier output signal is provided to a positive node 36 of a summing unit 38. Summing unit 38 also receives at its negative node 40 the twice-delayed output signal from output 26 of second delay circuit 24. Thus, summing unit 38 provides at its output 42 a difference signal representing the difference between the multiplier output signal received at its input node 36, less the twice-delayed output signal received at its negative node 40. This difference signal is the sinusoidal output signal $x(n)$.

Sinusoidal output signal $x(n)$ is provided through a feedback line 44 to first delay circuit 16 and comprises the first signal received by first delay circuit 16 at its input 18. Thus, there is provided a recursive oscillator circuit 15 which provides an output $x(n)$ determined by a multiplier δ received at multiplier input 32. Initial conditions are established for oscillator circuit 15 by an input $x(1)$ to first delay circuit 16 and by an input $x(0)$ to second delay circuit 24. Preferably, $x(1)=A\sin\omega_0$ and $x(0)=0$, where A signifies amplitude of the signal $x(1)$ and ω_0 signifies an initial frequency

$$\left(\omega_0 = \frac{2\pi f_0}{f_s} \right)$$

The employment of recursive oscillator circuit 15 as illustrated in Figure 15-2 in a communications system is illustrated, in its preferred embodiment, in Figure 15-3. In Figure 15-3, a communications sys-

tem 50 is illustrated receiving a digital signal input at an input 52 to a low pass filter (or signal synthesizer) 54. Low pass filter 54 employs a sampling frequency f_s received at an input 56. A signal "y" is generated from low pass filter 54 at an output 58 and passed to a scaling circuit 60. Scaling circuit 60 comprises a multiplier unit 62 and an adder unit 64 to which output signal "y" is successively applied. Scaling circuit 60 applies scaling factors α and β to signal y to produce an output δ on line 66 such that $\delta = \alpha y + \beta$. The scaled output δ is applied to recursive oscillator 15 and recursive oscillator 15 generates an output signal at line 70 in a manner as described in connection with Figure 15-2. Thus, the output signal at line 70 is a sinusoidal output signal which is defined by control signal δ which is received by recursive oscillator 15 via line 66 from scaling circuit 60.

An alternate embodiment of a communications system employing a recursive oscillator 15 of the type illustrated in Figure 15-2 is illustrated in Figure 15-4. In Figure 15-4, a communications system 80 is illustrated comprising a logic circuit 82 which receives a digital input signal at an input 84. Depending upon the signal level of the input signal received at input 84, logic circuit 82 selects an appropriate filter among an array of filters FT1, FT2, . . . , FTn for receiving a representation of the input signal on line 86. The selected appropriate filter, for example filter FT1 in Figure 15-4, filters the representative input signal received on line 86 to produce a control signal δ on a line 88 for provision to a recursive oscillator 15. Selection of the appropriate filter FT1, FT2, . . . , FTn by logic circuit 82 is preprogrammed in logic circuit 82 for effecting generation of a control signal δ to facilitate effecting noise-free transition of the frequency of an output signal produced at output 90 to accurately reflect deviations in the signal level of the input signal received at input 84 of logic circuit 82.

US Patents 5,189,381 and 5,204,642 are essentially based on the same architecture. The differences lie in the claims. Patent 5,189,381 addresses a process for generating an output signal in response to an input signal having a variable input level. The output frequency is representative of variations in the input level. The claims cover functionality and the process of generating output signals. Patent 5,204,642

addresses the actual apparatus for generating a sinusoidal signal recursively, and the claims accordingly cover the hardware elements of the architecture.

References

Related Patents

<u>4573025</u>	Feb., 1986	McKinzie	332/102.
<u>4686489</u>	Aug., 1987	Caspell	331/DIG.
<u>5105444</u>	Apr., 1992	Gard	332/100.

Chapter 16

Communications Processor

Abstract

This chapter is based on the invention patented in Europe under patent EP 0 465 054 B1 (S. Asghar, J. Bartkowiak, May 1995).

This invention relates generally to processors and more particularly to digital which have the capability of processing code excited linear predictive algorithms.

Recently, cellular telephone systems have become more practical and popular. Their popularity has increased to the point where service availability will soon be unable to meet demand. It has therefore been proposed to implement a digital cellular network which incorporates predictive speech signal coding for reducing the bandwidth of transmitted speech thereby enhancing system performance and enabling time division multiplexing which significantly increases user availability.

The Telecommunication Industries Association has adopted a standard, Digital Cellular Standard IS-54, which implements a vector sum excited linear predictive (VSELP) vocoder algorithm for use in the United States. This algorithm is computationally intensive, requiring on the order of 6.45×10^6 arithmetic operations per second and 15.6 million instructions per second (MIPS).

Other areas of the world are also moving toward digital cellular systems. Japan has recently adopted a VSELP algorithm similar to the United States, and Europe is moving toward a CELP algorithm referred to as Group Special Mobile (GSM).

Currently, digital signal processors (DSPs) such as the SDP65000 family manufactured by Motorola, Inc. are available for implementing digital algorithms such as digital filters, fast fourier transforms, correlation functions, etc. Such processors have special features such as modulo addressing, hardware DO loops, 24-bit hardware multipliers, etc. to facilitate the large number of high precision arithmetic opera-

tions required in digital signal processing. DSPs are ideally suited for executing the VSELP and similar algorithms except that current models have too low a clock rate. The DSP65000 family, for example has a nominal clock rate of 20.48 MHz giving an execution rate of 10.24 MIPS. This can be increased to 27 MHz for an execution rate of 13.5 MIPS. To accommodate the 15.6 MIPS needed for IS-54, the clock rate would have to be increased to about 32 MHz. Such an increase would require redesign of the DSP.

DE-A-1-3 314600 discloses a communications speech processor comprising a central processing unit (CPU), a fast computing circuit and an Input/Output unit, all connected by data and address lines. The CPU controls the system, while the fast computing circuit executes the speech processing algorithms.

In the present specification we describe a communications processor implementing IS-54 and other computationally intensive digital signal processing algorithms using conventionally available components operating at convention clock rates.

There is also described a communications processor which contains a user programmable functionality permitting the execution of instructions in addition to the computationally intensive digital signal processing algorithm without adversely affecting execution of the algorithm.

There is also described a communications processor implemented on a single integrated circuit chip.

In accordance with the invention there is provided an apparatus comprising in a single integrated circuit chip, the combination of a central processing unit (CPU) having an execution unit with an arithmetic logic unit and accumulators, a program counter, memory, a clock generator, a timer, a bus interface, a chip select unit, and an interrupt processor, a digital signal processor (DSP) having an instruction set to carry out a digital signal processing algorithm, an execution unit for carrying out multiply and accumulate operations and an external interface, an address bus connected between the CPU and the DSP, a data bus connected between the CPU and the DSP, and scheduling means for statically scheduling execution of the signal processing algorithm between the digital signal processor and the CPU.

In accordance with other aspects, the digital signal processing algorithm may be a digital speech pro-

cessing algorithm and the scheduler may be contained in the DSP and may include an interrupt generator for generating interrupts to the CPU to command execution by the CPU of portions of the speech processing algorithm. The CPU interrupt processor processes the interrupts for the DSP as nonmaskable interrupt signals to ensure that execution of the algorithm takes priority over other programs.

To exploit the advantages of the DSP and CPU, the static scheduler partitions execution of the signal processing algorithm to cause all multiply and multiply-accumulate operations to be executed in the DSP and all other operations such as add, subtract, divide, compare, etc. to be executed in the CPU.

In accordance with other aspects of the invention, the speech processing algorithm may be a code excited linear predictive coding algorithm and, in particular, may be a vector sum excited linear predictive coding algorithm.

The CPU includes operating system support for permitting user programming of the CPU. The static scheduler controls the operating system support to give priority to execution of the signal processing algorithm.

The static scheduler effects simultaneous operation of the CPU and DSP during portions of the execution of said signal processing algorithm.

Brief Description of the Drawings

The above and other objects of the invention will become more readily apparent as the invention is more fully understood based on the detailed description below, wherein like reference numerals represent like parts throughout and wherein:

Figure 16-1 is a block diagram showing the main components of the communications processor of the present invention;

Figure 16-2 is a more detailed block diagram of the communications processor of the present invention;

Figure 16-3 is a block diagram showing the audio interface of Figure 16-2;

Figure 16-4 is a block diagram showing more detail of the Multiplier-Accumulator engine of Figure 16-2;

Figure 16-5 is a block diagram showing the receiver section of the control channel processor of Figure 16-2;

Figure 16-6 is a block diagram showing the transmitter section of the central channel processor of Figure 16-2;

Figure 16-7 is a block diagram showing the frame formatter of Figure 16-2;

Figure 16-8 is a block diagram showing the organization of the watch dog timer of Figure 16-2;

Figure 16-9 is a block diagram showing the keypad scanner of the keypad interface of Figure 16-2;

Figure 16-10 is a block diagram showing the parallel I/O port of Figure 16-2;

Figure 16-11 is a functional block diagram showing the communications processor used to implement a digital mode cellular telephone application;

Figure 16-12 is a functional block diagram showing the digital transmit signal flow of the processor configured as in Figure 16-10;

Figure 16-13 is a functional block diagram showing the digital receive signal flow of the processor configured as in Figure 16-10;

Figure 16-14 is a flow diagram showing a basic code excited linear prediction (CELP) process;

Figure 16-15 is a flow diagram showing a codebook search procedure used in a CELP process;

Figure 16-16 is a flow diagram showing a revised codebook search procedure;

Figure 16-17 is a flow diagram showing an implementation of a vector sum excited linear predictive (VSELP) encoder;

Figure 16-18 is a flow diagram showing an implementation of a VSELP decoder;

Figure 16-19a is timing diagram useful with table 21 to show the partitioning of arithmetic operations between the digital signal processor and the central processing unit of Figure 16-2 when implementing a VSELP algorithm;

Figure 16-20 is a block diagram showing the communications processor implementing an analog mode operation;

Figure 16-21 is a flow diagram showing the analog transmit flow of Figure 16-20; and

Figure 16-22 is a flow diagram showing the analog receive flow of Figure 16-20.

Detailed Description of the Preferred Embodiments

Figure 16-1 shows the communications processor 100 of the present invention. Processor 100 is a highly integrated IC for implementing IS-54 of GSM compliant cellular telephones. This handset-resident device combines with RAM and ROM to provide a complete set of functions between audio transducers and an RF modem, including speech processing, user interface and system control.

One of the key elements of the architecture is the unique combination of a standard microprocessor shown in Figure 16-1 as CPU 200 with a dedicated custom DSP engine shown in Figure 16-1 as multiplier-accumulator (MAC) engine 300 and a multi-port static RAM (SCRAM) 400 to process the VSELP and GSM speech compression algorithms. This combination reduces die size while performing both voice processing and system control.

Figure 16-1 shows the organizational relationship of the major components of the communications processor 100 of the present invention. CPU 200 may be a convention 80186 type processor such as the model 80C186 microprocessor sold by Advanced Micro Devices of Sunnyvale California. This processor includes a clock generator 202, a bus interface 204, an execution unit 206, a direct memory access unit 208, a chip select unit 210, timers 212, and an interrupt processor 214. In accordance with the present invention, CPU 200 can be programmed by a user to implement user designated functions and operates in conjunction with DSP 300 to carry out VSELP in accordance with IS-54 and other signal processing algorithms, as will be explained in greater detail below.

CPU 200 communicates with SRAM 400 and DSP 300 through a bus system which includes address bus 102 and data bus 104. These busses also provide communication to a transmit/receive buffer 500 and an audio interface section 600.

As will be understood by reference to Figure 16-1, communications to a base station or the like, if a cellular phone system implementation is adopted, is made through buffer 500 which can connect to a

transmission system including a modem, a radio or the like. Communication to the local user is made through interface 600 which connects to a microphone and a speaker, as will be discussed in greater detail below.

Figure 16-2 shows more details of the communications processor 100. As shown in Figure 16-2, the on-chip circuitry also includes an operating system support unit 700, a watchdog timer unit 800, a system clock 900, a system test logic unit 1000, and a reset and power modes unit 1200.

User programming of the CPU is made available through a keypad interface 1300, a serial display and I/O bus unit 1400 and one or more parallel output ports 1500.

A brief description of the function of each component of Figure 16-2 will now be set forth.

The audio interface 600 is shown in more detail in Figure 16-3. Interface 600 provides an ear piece and loudspeaker 604 interface, handset microphone 601 and hands-free microphone 602 interface 620; A/D converter 606, D/A converter 608, hardware decimator 610, hardware interpolator 612 and tone ringer unit 614. Programmable gain units 616 are provided for all analog inputs and outputs.

The transmit path of the interface 600 has two inputs which are multiplexed into a programmable gain stage (0 to 24 dB; 3 dB steps) connected to an over-sampling A/D converter 806. The output of the A/D is decimated in hardware unit 610 to 16 kHz. This signal is loaded into the on-chip SRAM 400 for use by the speech processing engine. With 6 dB of programmed gain, a 625 mV input produces a full-scale digital output.

On the receive side, the 16 kHz output of the speech processor is interpolated in hardware unit 612 and fed into D/A converter 606. The output of the D/A feeds either the ear piece, loudspeaker, or both. A programmable analog gain stage is located in each path (0 to 36 dB, 3 dB steps). The ear piece driver can drive a 540 Ohm load at 5V P-P. The loudspeaker driver can drive a 40 Ohm load at 5V P-P. With 0 dB of gain programmed into either path, a full-scale digital code produces the full-scale output voltage.

The tone ringer 614 produces a nominal 50% duty cycle rectangular wave form having a programma-

ble frequency range for 12 kHz to 200 Hz. The amplitude of the output is programmable, starting with a 5V P-P maximum level, attenuated by up to -36 dB in 3 dB steps. The tone ringer is summed into the loudspeaker output. The ringer and loudspeaker output buffer can be operated with the rest of the audio interface disabled.

Analog loop back and input mute function is provided in interface 620. An analog sidetone path, with programmable gain (range and resolution) is also provided.

A reference voltage output, approximately 2.4 Volts, is provided for biasing external circuitry (including electret microphones). A related pin is provided for filtering the internal bias reference from which the reference voltage output is derived.

A memory interface 622 moves digitized audio between the internal SRAM 400 and the audio interface by a dedicated DMA mechanism. Transmit and receive buffer spaces are set aside in the SCRAM. These buffers function as FIFOs, and are sufficiently deep to compensate for the block nature of the speech processing algorithm. The buffers have fixed base addresses. The memory interface moves data to and from the buffers at a rate of 16k words per second in each direction.

The CPU 200 is a complete 80C186 microprocessor, capable of running at up to 20 MHz. The 80C186 provides two 16-bit timers, programmable chip select outputs, and an interrupt controller. Some of these resources are used internally, and are not available to the user.

The clock speed of the CPU is dynamically controlled by the system clock generator block 900, with the user having programmable control over the clock speed.

The clock generator block 900 serves as the master distribution point for clock signals used by the various block. This centralized scheme is used to reduce power consumption within the communications processor 100. The block contains the crystal oscillators and various dividers. One of these dividers is associated with the CPU clock input. This divider is programmable, allowing the clock speed, and thus the power consumption, of the processor to be reduced whenever the workload on the processor is low.

The processor clock speed can also be changed automatically in response to hardware interrupts. If this options is selected, any hardware interrupt to the CPU will force the clock speed to maximum. This reduces interrupt response latency when operating at NMI, and automatically returned to its previous speed upon a return from NMI. After power on reset, the maximum clock speed is applied to the CPU as a default. The maximum dividing factor that can be applied to the system clock generator is 64. The CPU clock can also be programmed to stop.

These are three classes of interrupts supported by the CPU: 1) NMI from the MAC engine 300; 2) direct interrupts to the CPU; and 3) indirect interrupts. The NMI from the MAC is the basic task/scheduler interrupt controlling the switch between speech processing and user CPU time. The NMI is generated by the MAC engine 300, and fed directly to the CPU's NMI input. The direct interrupts are those connected directly to four maskable interrupts inputs on the CPU 200. The keypad scanner 1300, parallel I/O 1500, and control channel processor 502 generate direct interrupts. The fourth direct interrupt is generated by an indirect interrupt controller, which gathers all remaining interrupts into a single interrupt. Any interrupts from external user circuitry feed into the parallel I/O port.

The 80C186 processor used as CPU 200 has three timers, two have input and output pins, (Timers 0 and 1), one (timer 2) has no I/O connections. Timer 0 is pinned-out for user access, timer 1 is not used either internally or externally, and timer 2 is dedicated to internal functions.

A19/S6	INT2-INTA0/+
A18/S5	INT3-INTA1/+
A17/S4	LCS/
A16/S3	LOCK/*
AD15	MCS0/-PEREQ
AD14	MCS1/ERROR
AD13	/

AD12	MCS2/*
AD11	MCS3-NPS/*
AD10	NMI+
AD9	PCS5-A1*
AD8	PCS6-A2*
AD7	PSC4/*
AD6	PSC3/*
AD5	PSC2/*
AD4	PSC1/
AD3	PSCO/
AD2	RD/-QSMD/
AD1	RES/*
AD0	RESET*
ALE/QSO	S2/
ARDY	S1
BHE/	S0
CLKOUT*	SRDY
DEN	TEST/-BUSY*
DRQ0*	TMR IN 0
DRQ1*	TMR IN 1*
DT-R/	TMR OUT 0
HOLD*	TMR OUT 1*
HLDA*	UCS/
INTO+	WR/-QS1

INT1+

X1

X2*

The Multiply/Accumulate (MAC) engine 300 is a special purpose DSP engine designed to work with the CPU 200 to perform complex speech processing algorithms. The MAC engine is shown in greater detail in Figure 16-4 to include an execution unit 302, a sequencer 304, various loop counters 306, 308 and address generators 310, 312, an instruction ROM 314, and an external interface to the SRAM 400. The MAC may also include conventional DSP features such as a 24 bit X 24 bit hardware multiplier and hardware do loops to help carry out its specialized function.

The MAC engine 300 functions as the master in a master/slave relationship with the CPU 200. Synchronization is maintained by non-maskable interrupts (NMIs) generated by the MAC engine 200 and transmitted to the CPU 200 NMI input on NMI line 326. A sequence counter is maintained by the MAC engine 300. This counter is incremented by the MAC engine 300 every time an NMI is generated, and cleared at the end of each frame. The CPU 300 reads the counter as an interrupt pointer.

The speech algorithms or other signal processing programs to be implemented by MAC 300 are microcoded in internal ROM 314 and are not accessible by the user. These algorithms process complex speech processing algorithms on sampled speech data stored as blocks in RAM 316.

Instruction Set -- The MAC Instruction set supports digital speech processing algorithms which include such functions as convolution, correlation, interpolation, decimation, vector orthogonalization, and recursive filtering these functions are performed on data stored as blocks and the architecture of the engine reflects this application. The operations necessary to implement these functions include multiply, multiply-accumulate, add, subtract, data overflow management, sequential block addressing, data movement, and data scaling among others. They are summarized below:

Accumulator Memory Reference Instructions

Absolute value of Accumulator

Add to Accumulator with shift

Add to Accumulator with carry

Add to high Accumulator

Add to Low Accumulator with sign extension suppressed

Load accumulator with shift

Store High Accumulator with shift

Store Low Accumulator with shift

Subtract from Accumulator with shift

Subtract from Accumulator with borrow

Subtract from High Accumulator

Subtract from Low Accumulator with sign extension suppressed

Zero Accumulator

Memory Address Generator Instructions

Load address generator

Modify address generator

Store address generator

Multiply Instructions

Add product register (Preg) to Accumulator

Load temporary register (Temp)

Load Temp and accumulate previous product

Load Temp and store Preg in Accumulator

Load Temp and subtract previous product

Multiply and accumulate
Multiply (with Temp, store product in Preg)
Multiply and Subtract previous product
Multiply (with Temp) and accumulate previous product
Multiply unsigned
Load Accumulator with Preg
Store High Preg
Store Low Preg
Set Preg output shift mode

I/O and Data Memory Operation

Input data from mailbox
Output data to mailbox Set External Interrupt flag and load Action Register
Reset External Interrupt flag
Enable Loop Counter
Disable Loop Counter

As discussed above, the MAC engine 300 operates as the master to slave CPU 200. Control of the CPU is achieved by setting an external flag on line 326 which acts as a non-maskable Interrupt (NMI) to the CPU. An action counter is updated by the MAC engine which is readable by the CPU on interrupt request acknowledge and thereby task synchronization is maintained. Communication of data and control parameter to and from the CPU is made via the dedicated dual-ported RAM 316 which employs a mailbox handshaking mechanism. Static scheduling of the speech processing algorithm between the MAC engine and CPU allows this simple and efficient interface to communicate parameters in order to compute the algorithm functions.

The internal SRAM unit 400 is a single port structure with a multi-port access arbitrator. The SRAM unit 400 is accessed by the MAC engine 300, the CPU 200, the audio interface 600, and the control channel processor 502, which is part of the transmit/receive buffer 500. The RAM is organized on byte, word (16), and long word (24) boundaries depending on which module is accessing it. The total RAM size is consistent with the requirements of the program to be run. The SRAM may be partitioned into separate blocks to simplify the access control mechanism. For example, the RAM associated with the FACCH, SACCH, CDVCC functions may be located in a separate partition from RAM used for speech data.

The operating support logic unit 700 is hardware specific to provide task scheduling for the CPU 200 between a speech processing operation and "user code" received and located in this block. The partitioning between the speech functions and the user functions is rigidly enforced by a static scheduler.

In order to support the static partitioning of the CPU between speech processing and non-speech tasks, all real-time events, such as interrupts not associated with the speech algorithm, must be presented to the CPU 200 only during non-speech windows. The function of the OS support logic block 700 is to perform this isolation function. In general, interrupts will automatically be blocked by the fact that the speech functions occur during the NMI service routine which cannot be interrupted by other interrupts (the NMI is used as the mechanism by which the speech function gains control of 80C186, with control being returned to non-speech tasks when a return from NMI instruction is executed). Therefore, no specific logic is required to hold off non-speech interrupts.

As mentioned above, the CPU operates both as the system CPU and as part of the speech processor. What the user actually sees is a portion of the CPU processing capacity. The partitioning between system functions and speech processing is enforced by the OS support logic block and a firmware shell. The scheduling mechanism is static in nature, giving the CPU resource to the speech operation unconditionally at very specific times. When the CPU is not performing speech related tasks, it is available for system functions. As a consequence of the static scheduling, the user sees the CPU for a portion of the time

with rigidly enforced gaps when the processor is unavailable. These gaps are transparent to the user, with the exception that the CPU is not working on non-speech tasks during the gaps. This means that all systems tasks including interrupts are blocked (held off) during speech processing windows.

Some handshaking may be required between the OS support logic and the power mode control functions to insure that the shut-down and idle modes are entered and exited in an orderly fashion.

The MAC engine 300 provides the indication as to whether the 80C186 is in speech or non-speech mode at any given time.

The control channel processor 502 functions differently depending on the algorithm being implemented, such as whether it is in GSM or IS-54 mode. The basic function of the control channel processor 502 is to perform forward error control and cyclic redundancy check (CRC) functions on the data channel, and bit-level processing on the control channel.

In IS-54 mode, the control channel processor 502 is ideal during analog and call set-up operation, to be discussed below. The 80C186 CPU 200 handles all control processing during analog and call set-up modes. During digital operation, there are four channels to be processed: SACCH, speech CDVCC and FACCH. The presence of Fast Associated Control Channel (FACCH) messages in the data channel must be detected and the data processed. Figure 16-5 and Figure 16-6 show block diagrams of the receive and transmit sections of the control channel processor.

ACCH, Receive - as shown in Figure 16-5 receive SACCH data is fed to the control channel processor from the frame formatter on lines 510. Hardware within the control channel processor block 512 DMA's the data from a shift register to a buffer in internal SRAM. The SACCH data is in encoded form (1/2 convolution code) and interleaved with SACCH data from 23 slot times. The data is not processed in hardware, rather, it is moved to an internal RAM buffer. From there it is moved by the 80C186 CPU 200. From there it is moved by the 80C186 CPU 200 to a buffer in internal RAM. As shown in Figure 16-6, hardware within the control channel processor unit 20 DMA's the data to a shift register 522 inside the

control channel processor.

Speech, Receive - The data channel can contain either Speech or FACCH data. Since the communications processor 100 does not know in advance which type of data is contained in any given received frame, the data must be processed as both Speech and FACCH data.

In the receive direction, as shown in Figure 16-5, speech data is split into class 1 and class 2 portions in a sequenced 526. The class 1 portion is processed through a 1/2 convolution code processor 528 and a CRC checker 532 before being placed into internal SRAM by controller 534. Class 2 data is placed directly into the SRAM by controller 534. **FACCH, Receive** - The received data channel is also passed through the FACCH processor wherein it is fed a 1/4 convolution decoder 530 and a CRC checker 536. If the CRC check is valid, it is an indication that the data was in fact FACCH data, otherwise it is considered Speech data. Valid FACCH data is DMAed into an internal RAM buffer.

Speech, Transmit - On the transmit side (Figure 16-6), the system knows in advance whether the data is Speech or FACCH. Speech data to be transmitted is placed into SRAM by the MAC engine in class 1 and class 2 fields. The class 1 field is fed through a 1/2 convolution encoder 540 and a CRC generator 542 by DMA unit 546, before being passed to the frame formatter by multiplexer 544. Class 2 data is passed directly to the frame formatter by multiplexer 544.

FACCH, Transmit - FACCH data to be transmitted is placed into a buffer in internal RAM by the 80C186 CPU 200. From there it is DMAed into the control channel processor and fed through a 1/4 convolution coder 550 and a CRC generator 552 before being passed to the frame formatter by multiplexer 544.

CDVCC, Receive - The received CDVCC data from the frame formatter is processed through a 12, 8 hamming code decoder 560 and DMAed to a buffer in internal RAM. From there it is processed by the 80C186.

CCVCC, Transmit - The 80C186 CPU 200 writes the CDVCC byte to a buffer within the control

channel processor. From there it is processed through a 12, 8 hamming code encoder 554 and shipped to the frame formatter.

GSM - In GSM mode, the following functions are performed on the Speech channel: In the receive direction, the HDLC (LAPM) bit level processing is handled in the control channel processor, and the data is placed in the internal SRAM. In the transmit direction, packets are formatted by the 80C186 and placed into the internal SRAM. From there the control channel processor executes the bit-level HDLC protocol and passes the data to the frame formatter.

Interrupts - The control channel processor generates a direct interrupt to the CPU 200 in response to certain status conditions. These conditions include: FACCH, SACCH and CDVCC received data service requests; FACCH and SACCH transmission complete indications; speech error mute requests and an FACCH error indication.

The frame formatter 504, which is also part of the transmit/receive buffer 500, functions differently depending on the algorithm being implemented, such as on whether it is in GSM or IS-54 mode. In IS-54 mode, the frame formatter assembles and disassembles 20 millisecond transmit and receive slots (two per frame). The data is broken down into coded digital verification color codes (CDVCC), slow associated control channel (SACCH), and voice/fast associated control channel (FACCH) data streams. The frame formatter sits between the serial radio interface 506 and the control channel processor.

In the transmit direction, speech/FACCH, SACCH and CDVCC data is combined with sync, guard, and ramp bits from units 580, 582 and 584 in multiplex sequencer 586 and sent to the serial radio interface 506. The speech or FACCH data is assembled in a buffer 560 by a sequencer 562 that performs a slot interleaving operation over two slot times. Because of the nature of the interleaving algorithm, the buffer is two blocks deep. Data is transmitted out of one block while the next block is being assembled.

Since the serial radio interface carries both control and data information, its bit rate is faster than the bit rate for the radio. For this reason, a radio bit rate clock is generated for speech operations and the

serial radio interface (SRI) bit clock on line 590 is used for transmission and reception of non-speech functions. The radio bit rate clock is phase-locked to the recovered slot synchronization.

The receiver demultiplexes that data received from the serial radio interface into speech/FACCH, SACCH, and CDVCC bit streams in a duplexer/sequencer 570. The speech data is deinterleaved and stored in a two-slot deep buffer 592. The SACCH and CDVCC data streams are fed directly to the control channel processor A sync detector 594 identifies both the slot boundaries as well as the slot ID>

The serial radio interface 506 (Figure 16-2) combines the bit stream to be transmitted (or received) with control commands to and from the CPU and the radio into a single serial bit stream. This is a three wire interface having transmit data, receive data and bit clock signals. A framing pattern is embedded into the data stream. The definition of this block is totally dependent upon the user radio to which it is attached.

A sanity timer is included in the communications processor 100. The timer is permanently enabled, generating a 2 ms pulse on the RESET/(active low) pin if it times out. Hardware internal to the communications processor is also reset by a watchdog timer 600 (Figure 16-8) time-out; the result is identical to activating the RESET/pin. A hardware reset is generated to the communications processor 100 if a counter 802 is not cleared every 2 seconds. If counter 802 is not cleared, an output is generated to reset pulse generator 802. A specific key sequence must be written to a Watchdog Key Register 804 by the CPU 200 within this 2 seconds after reset and within 2 seconds of the previous update to prevent time-out.

The key sequence is a two stage function starting from reset. The watchdog timer starts in stage one, waiting for a write to the Watchdog Key register. If the value written to the Watchdog Key Register is anything buy an A5H, a system reset is generated (just as if the terminal count had been reached). If the write was an A5H, stage two is entered. In stage two, the watchdog timer waits for another write. If the write is an 5AH, the timer is reset and stage one is re-entered. If the write is something other than a 5AH,

a system reset is generated.

The watchdog timer has a hardware pinstrapable disable mode, allowing the watchdog to be disabled for software development. When the WDTDISABLE pin is tied low, the watchdog timer is enabled.

When tied high, the watchdog timer is disabled.

The keypad interface 1300 (Figure 16-2) includes a keypad scanner 1302, shown in Figure 16-9. The scanner 1302 supports keypads with up to 25 keys. The scanner works autonomously from the CPU 200. Keypad activity is detected and reported via a status register 1304 mapped into the CPU's address space. Interrupts are generated when a key is depressed and when it is released. Activity is debounced on both depression and release. A wake-up signal is generated if keypad activity is detected when MSC is in shut-down or idle modes. An error code is generated if multiple keys are depressed.

The following functions are performed by the keypad scanner 1302; detection of keypad activity, determination of the identity of the key closure, debouncing of the key down and a key up transitions, generations of a direct interrupt to the CPU 200. The status of the keypad is reported via a register mapped into CPU 200 user address space. This status includes a not key down code (00000000), a multiple keys down code (XXXXXXXX1), and codes for each of the keys (RRKKKKK0); R = reserved, K = key code and X = don't care. A maskable interrupt is generated when the visible register changes. The keypad scanner is designed to operate in a static state when no activity is present, waking up automatically upon the detection of activity.

The serial I/O, display interface 1400 (Figure 16-2) is a three-wire bus allowing the CPU 200 to talk to external serial devices such as LCDs. The three-wire serial bus (clock, data in and data out) is provided for communicating with the display module and any other serially controlled device that the user requires, such as a serial EEPROM. The serial bus is a peripheral to the CPU 200 and is solely under the control of user software. The protocol used by the serial bus is to be specified by the user. If multiple devices are connected to the bus the parallel I/O port can be used to provide the required chip select func-

tion. The serial bus can be disabled via software to reduce power consumption. When the communications processor is in shut-down mode, activity on the data input will not wake up the communications processor 100.

The system clock generator 900 produces all clocks internal to the communications processor 100. Dynamic control of clock speeds for reduced power consumption is handled in this block, in concert with the reset/mode control block.

The reset and power mode controller 1200 provided power-on-reset and low voltage detection. Separate reset input and output pins are provided. In addition to the system reset function, control over the entering and exiting the various operation modes of the MSC is located in this block.

Parallel I/O ports 1500 are provided as general purpose I/O pins which can be programmed as either an input or an output. The logic state of each output is programmable by the CPU 200 by setting input/output select register 1502 (Figure 16-10).

Pins programmed as Inputs report both the current status at the pin as well as an indication that the pin has changed state from a high to a low since last read. A single maskable interrupt is generated to the CPU 200 when any input changes state from a high to a low. This interrupt is maskable on a bit by bit basis. At reset, all pins are inputs and the interrupt is disabled.

The state of pins programmed as outputs directly reflects the logic level programmed into register 1504 by the CPU 200.

Control/Status Registers --Four types of registers are utilized, all of which are mapped into CPU 200 address space. The first type of register 1502 has a bit for each pin, controlling whether that pin is an input or an output. The second type 1504 contains a bit for each pin. These bits present the current state of input pins, and set the state of output pins. The third type 1506 contains one bit per pin, reporting whether each input has changed state since the register was last read. The last type of register 1508 provides an interrupt enable bit for each pin. The parallel port interrupt is a direct interrupt, fed straight to the

CPU 200.

The system test logic block 100 provides hardware for performing system tests on the telephone. These tests are determined based on the algorithm being supported.

The role of the communications processor 100 can best be shown with reference to drawing of a typical IS-54 cellular telephone. Figure 16-11 shows the functions required for an IS-54 telephone operating in digital mode (GSM operation is similar to this drawing, with the exception that the speech and signaling algorithms are different). The communications processor 100 performs the functions within the dashed box.

Figure 16-20 shows the telephone functions for analog operating mode. In this mode, the communications processor performs all of the required speech and control functions. It is interesting to note that communications processor 100 processes the speech and control signals using the same DSP resources used for the digital mode. The A/D and D/A converters present in the digital mode DQPSK modem are used in the analog mode to convert the analog signal present at the radio to and from a digital format. This reduces the duplication of hardware that would otherwise be required for dual-mode operation.

Figure 16-12 shows the transmit path for digital speech processing using the present invention. This figure shows: Analog Gain block 616, A/D 606, Decimator 610 -- These blocks amplify the input from the microphone(s), digitize it and decimate it.

Hands-Free Attenuator (HF Attenuator) -- This is a programmable attenuator under the control of the hands-free unit which may be provided, if desired.

High-Pass Filter (HPF) -- This is a high-pass filter with a 120 Hz cut-off.

DTMF -- DTMF tones to be transmitted are injected into the signal path here by DTMF block. These tones can also be generated by the base station.

VSELP Encoder -- The speech compression algorithm is performed by this block.

CRC - A CRC is calculated for the 12 most perceptually significant bits of each encoded block of

speech.

Convolutional Encoding -- A 1/2 convolution code is used to process the 77 class 1 bits of each block of speech.

FACCH Convolutional Encoding -- When FACCH data is transmitted instead of speech, it is processed through a 1/4 convolution coder.

SACCH Convolutional Encoding -- SACCH data is processed through a 1/2 convolution coder.

CDVCC -- The 12-bit CDVCC is constructed using a 12.8 hamming code and passed to the frame formatter as a serial bit stream.

Frame Formatter -- The CDVCC, SACCH and speech/FACCH bit streams are arranged into frames for transmission along with guard time, ramp time and sync bits.

DQPSK Modem -- The bit stream from the frame formatter is converted to an analog format via the DQPSK modem.

RF/FM Modulator -- This block is the "radio" providing FM modulation and RF functions.

The receive path for digital speech processing is shown in Figure 16-13. In this figure:

RF/FM Demodulator -- This block is the radio receiver providing RF and FM Demodulation functions.

Differential Quadrature Phase Shift Keying (DQPSK) Modem -- The output of the radio is converted into a digital bit stream by a DQPSK modem.

Frame de-Formatter -- The bit stream from the modem consists of frames. The frame de-formatter separates each frame into SACCH, CDVCC and speech/FACCH data. Additionally, slot synchronization is detected in this block.

Convolutional Decoder -- The 178 bits of encoded class speech data are processed through a 1/2 convolution decoder.

FACCH Convolutional Decoder -- If present, FACCH data is processed through a 1/4 convolution

decoder.

SACCH Convolutional Decoder -- SACCH data is processed through a 1/2 convolution decoder.

CDVCC -- Hamming encoded (12,8 code) CDVCC is decoded in this block.

CRC -- A 7-bit CRC was appended to the 12 most perceptually significant class 1 speech bits before they were convolutionally encoded by the base station transmitter. This CRC field is checked by this block.

VSELP Decoder -- The VSELP speech expansion algorithm is processed by this block.

Mute -- Under certain conditions the voice signal must be muted. Any transients created either entering or leaving mute must meet specified amplitude and duration requirements.

Call Progress Tones -- Tones such as dial tone and DTMF are generated to provide user feedback. These tones are injected into the receive path at this point. Note that the tone generator can also be used to produce the transmit side DTMF signals.

Hands-Free Attenuator -- This is a programmable attenuator under the control of the hands-free unit.

Interpolator, D/A, LPF, Analog Gain Stage -- The interpolator, D/A and low-pass filter recreate the analog audio signal which is amplified by the programmable gain output driver(s).

Speech Processing (IS-54)

Introduction -- The speech coding algorithm belongs to a family of speech coders known as Code Excited Linear Predictive (CELP) Coders. Such methods employ codebooks to vector quantize the excitation (residual) signal and use an analysis-by-synthesis approach to determine which code to use to represent the excitation. The algorithm used is a variation on CELP called Vector-Sum Excited Linear Predictive (VSELP) Coding which employs a codebook structure which is predetermined. Such techniques reduce significantly the overall codebook search process. The algorithm requires both encode and decode functions which employ traditional LPC synthesis methods shown in Figure 16-7.

CELP Synthesizer

Figure 16-14 shows the basic speech CELP synthesis routine. The CELP synthesizer in the decoder uses a received code to determine which excitation vector from the codebook to use. The codebook contains 128 vectors each of length 40 samples, these vectors being typically random white gaussian variables. The vector chosen is scaled by a gain term gamma and applied to a set of linear filters to obtain 40 samples of reconstructed speech. The filters include a "long-term" or "pitch" filter which inserts periodicity into the excitation. The output of the "long-term" filter is applied to the "short-term" or "formant" filter which adds the spectral envelope to the signal.

"Long-term" Filter -- The long-term filter incorporates a single-tap predictor which attempts to predict the next output sample from one past sample. The transfer function is given by

$$B(z) = \frac{1}{1 - bz^{-1}}$$

where $B(z)$ is characterized by the quantities b and L . L is called the "lag" and for voiced speech would typically be the pitch period or a multiple of it. The parameter b is the "long term" predictor coefficient.

"Short-term" Filter -- The short-term filter incorporates a short-term predictor which attempts to predict the next output sample from the previous 10 output samples. This filter is equivalent to the traditional LPC synthesis filter and has a transfer function given by

$$A(z) = \frac{1}{1 - \sum (a[i]z^{-i})}$$

for $i = 1$ to 10

The "short-term" filter is characterized by the $a[i]$ parameters, which are the direct form filter coefficients for the all pole "synthesis" filter.

Parameter Update -- The various parameters (code, gain, filter coefficients) are not all transmitted at the same rate to the synthesizer. The "short-term" parameters are updated at the "frame" rate while the "long-term" parameters are updated at the "subframe" rate. A frame is composed of 4 subframes where a subframe is composed of 40 samples and the sampling rate is 8 KHz.

Short-term Predictor Parameters -- The short-term predictor parameters are the $a[i]$'s of the short-term filters. These are standard LPC direct form filter coefficients and are generated by an LPC analysis technique called fast fixed-point covariance lattice algorithm known as FLAT. It has the advantage of lattice algorithms including guaranteed filter stability, non-windowed analysis and the ability to quantize the reflection coefficients within the recursion.

Analysis-by-synthesis -- The codebook search procedure consists of trying each codevector as a possible excitation for the CELP synthesizer which is shown in the Figure 16-8. The synthesized speech $s'(n)$ is compared against the input speech and a difference signal is generated. This difference signal is then filtered by a weighting filter, $W(z)$, to generate a weighted error signal. The power in the error signal, $e(n)$, is computed and the codevector which produces the minimum weighted error power is chosen as the codevector for the subframe.

Figure 16-15 shows the routing for carrying out the CELP codebook search. In Figure 16-15 the following are depicted:

Weighting Filter $W(z)$ -- The weighting filter serves to weight the error spectrum based on perceptual considerations. It is a function of the speech spectrum and is expressed in terms of $a[i]$ parameters of the "short-term" (spectral) filter.

$$W(z) = \frac{1 - \sum(a[i]z^{-1})}{1 - \sum(a[i]fz^{-i})}$$

The parameter which defines the amount of error weighting is f , which controls the amount of error to be allowed in the formant regions of the speech signal. In this implementation, $f=0.8$.

A revised CELP codebook search is shown in Figure 16-16. The weighting filter is moved from both input paths to the subtracter giving the equivalent configuration shown in Figure 16-16. Here $H(z)$ is a combination of $A(z)$, the short-term (spectral) filter and $W(z)$ the weighting filter. These filters are combined since the denominator of $A(z)$ is canceled by the numerator of $W(z)$.

$$H(z) = \frac{1}{1 - \sum(a[i]fz^{-i})}$$

Gain Optimization -- Instead of determining the gain parameter prior to the codebook search based on residual energy, a method which would fix the gain for the entire search, an approach is chosen which optimizes the gain for each codevector and yields better results. The codevector which yields the minimum weighted error is chosen and its corresponding optimal gain would be used for. This implies that the gain term must be updated at the subframe rate.

Codevector Construction -- The VSELP coder uses an excitation codebook of 2^7 codevectors which is constructed from 7 basis vectors. Defining $v_m(n)$ as the m 'th vector and $u_i(n)$ as the i 'th codevector in the codebook, then

$$u_i(n) = \text{SUM}\{g_{im} \hat{P} v_m(n)\}$$

$$u_i(n) = \hat{A}\{g_{im} X V_m(n)\}$$

for $m = 1$ to 7 ; $i = 0$ to 127 ; $n = 0-39$

and $g_{im} = +1$ if bit m of codeword $i = 1$
 $= -1$ if bit m of codeword $i = 0$

In other words, each codevector in the codebook is constructed as a linear combination of the 7 basis vectors. The advantages of the VSELP codebook over random codebooks are

Extremely efficient codebook search procedure

Low codebook storage requirement

More robust to channel errors

Efficient joint optimization of the codeword and "lag" coefficient b

Bit Allocation -- The basic data rate of the speech coder is 7950 bits per second. This is coded up to 13 kilobits per second using forward error correction/detection techniques. There are 159 bits per frame (20ms) for the speech coder (before error control) which are allocated as follows:

Short-term filter coefficients	38 bits/frame	
Frame energy	5 bits/frame	
Lag, L bits/frame	7 bits/subframe	28
Codewords bits/frame	7+7 bits/subframe	56
b, γ_1 , γ_2 vector quantizer 8 bits/subframe 32 bits/frame		

Figure 16-17 shows a flow diagram of the steps to be carried out in executing the VSELP encoding algorithm according to IS-54. Figure 16-18 shows the decoding VSELP algorithm of IS-54. A detailed explanation of each of the steps of Figure 16-17 and Figure 16-18 can be found in the EIA/TIA Project Number 2215 publication "Dual-Mode Mobile Station-Base Station Compatibility Standard IS-54" published in December 1989 by the Electronic Industries Association, which is incorporated herein by reference.

Figure 16-19a demonstrates the timing of partitioning of the arithmetic steps required to carry out the steps of Figure 16-17 and Figure 16-18. Table 1 below:

Table 16-1

No	Function	No. of Arithmetic Operations		Frame/ Subframe	Comments
		DSP	CPU	F/SF	
	Encoder E				
	Decoder D				
E①	High Pass Filter	1440		F	160 samples
E②	Autocorrelation & Windowing	1870+ 55		F	
E③	Covariance Lattice Algorithm	1475	10	F	
E④	Convert r_i to a_i	45		SF	
E⑤	Interpolate a_i	20*		SF	*subframes 1, 2, 3 only
E⑥	Recover to r_i - stability	90*	9	SF	*subframes 1, 2, 3 only
E⑦	Calculate Long-term Predictor Lag	11556	126	SF	
E⑧	Zero-state Response of each codevector to $H(Z)$	2450		F	Codebook 1 search
E⑨	Interpolate zero-state response	280*		SF	*subframes 1, 2, 3 only
E⑩	Orthogonalise codevectors	600	1	SF	
E⑪	Calculate R_m, D_{mj}	280 + 1120		SF	
E⑫	Calculate C_i, G_i, C_u, G_u		616	SF	
E⑬	Calculate $C_i^2 G_{best}$ & $C_{best}^2 G_i$	192		SF	
E⑭	Compare values		64	SF	End of codebook 1 search
E⑧'	as ⑧ for codebook 2	450		F	Codebook 2 search
E⑨'	as ⑨ for codebook 2	280		SF	*subframes 1, 2, 3 only
E⑩'	Orthogonalise codebooks 1 and 2	6320	8	SF	
E⑪'	as ⑪ for codebook 2	280 + 1120		SF	
E⑫'	as ⑫ for codebook 2		61	SF	
E⑬'	as ⑬ for codebook 2		192	SF	
E⑭'	as ⑭ for codebook 2		64	SF	End of codebook 2 search
E⑮	Calculate Rec, $R_x(i)$	240 + 40		SF	Gain Quantisation

No	Function	No. of Arithmetic Operations		Frame/ Subframe	Comments
		DSP	CPU	F/SF	
	Encoder E				
	Decoder D				
E(16)	Calculate RS, R _{pc}	22 + 120		SF	
E(17)	Calculate a, b, c, d, e, f, g, h, i	21	3	SF	
E(18)					
E(19)	Gain error minimisation	2304		SF	
E(20)	Compare gain error values		255	SF	
E(21)	Calculate B ₉ Y ₁₄ Y ₂₄	3		SF	
E(22)	Generate selected codevectors		560	SF	
E(23)	Generate combined excitation	120		SF	
E(24)	Update weighted synthesis Filter	400		SF	
E(25)	Update long-term Filter states			SF	Address pointer update
D(1)	Convert r _i to a _i	45		SF	
D(2)	Autocorrelate Postfilter impulse resp.	55+55+10		F	Subframe 4 only, item interpolate
D(3)	Binominal Windowing of Autocorr.	10		F	
D(4)	Levinson Recursion (n)	110	10	F	
D(5)	Interpolate a _i	20*		SF	*subframes 1, 2, 3 only
D(6)	Convert to r _i - stability	90*	9	SF	*subframes 1, 2, 3 only
D(7)	Calculate R _{q(o)} Frame energy			SF	
D(8)	Interpolate Frame energy		1	SF	
D(9)	Calculate RS	22		SF	
D(10)	Transform GS, PO, P _q to P _q Y ₁₄ Y ₂₄	40+9	3	SF	
D(11)	Generate combined excitation	120	560	SF	
D(12)	Update synthesis filter A(z)	400		SF	
D(13)	Update long-term filter states			SF	Address pointer update

No	Function	No. of Arithmetic Operations		Frame/ Subframe	Comments
		DSP	CPU	F/SF	
		Decoder D			
D ¹⁴	Update spectral post filter	840		SF	
D ¹⁵	Signal scaling	122	2	SF	

shows the number of arithmetic operations needed to execute these various steps of Figure 16-17 and Figure 16-18. The circled numerals in Table 1 correspond to the circled numerals in Figure 16-19a and in Figure 16-17 and Figure 16-18. In each case a numeral in conjunction with the letter E indicates an encoding operation while a number in conjunction with a D indicates a decode operation. Also, some processing steps are performed at frame rates and others are performed at subframe rates. The digital speech data according to IS-54 is divided into 160 sample frames 20 ms long. It can therefore be seen that Figure 16-19a represents one frame period of 20 ms with each line in Figure 16-19a representing one subframe period of 5 ms. Operations such as the long term prediction computations are performed once for each frame. Short term analyses, on the other hand, are carried out once for each subframe. Table 1 makes this distinction by showing a "F" for frame rate operations and an "SF" for subframe operations. Figure 16-19a indicates this difference by placing a numeral next to the D or E for subframe rate operations. For example, "E1" means the operation is being performed for the first subframe, "E2" means the operation is being performed for the second subframe, etc.

Figure 16-19a is a time graph divided into four time lines of 5 ms each. Points or lines above the main time line indicate operation of the CPU and points or lines below the main time line indicate operation of the DSP or MAC engine. Cross hatching indicates simultaneous operations of the CPU and DSP. The operations are partitioned here so that multiply or multiple and accumulate operations are performed by the DSP which is adapted for such calculations due to its hardware multipliers, hardware do loops, etc. Other operations such as additions, subtractions, divides, square roots, exclusive ORs, look-up tables and

comparisons are performed by the CPU. The execution of these partitioned operations is scheduled to take advantage of as much simultaneous operation of the CPU and DSP as is feasible in order to reduce the total time required to execute the entire procedure. It will be understood by reference to the figures that this means some of the calculations are performed before they are actually required.

The partitioning and scheduling demonstrated by Figure 16-19a permit the VSELP algorithm to be performed effectively and efficiently using conventional clock rates and yet leaves ample time for the CPU to process user programmed routines.

The communications processor 100 can also be used to handle conventional analog cellular communications. The IS-54 is a dual mode standard providing both digital operation and backward compatibility with the existing analog IS-3 standard. As a result, two complete control and voice mechanisms must be supported.

The approach chosen takes advantage of the high bandwidth A/D and D/A resources required in the digital modem (digital voice operation) to convert the analog mode signal into digital form. Once the analog signals have been digitized, the DSP resources required in digital mode can be used to process the speech (emphasis/de-emphasis, compression/expansion, etc.) and control (SAT, FSK modem, etc). The major open architectural decision remaining is the partitioning of these tasks between the DSP engine in the MSC and the DSP engine in the digital modem. While the MSC has the horsepower to perform the required functions using the digital modem's DSP reduces the data rate required between the communication processor 100 and the digital modem -- this is important since this serial channel is carried in the cable between the handset and the radio box in applications where the radio is not built into the handset.

Call Set-Up/Analog Conversation

Since the call set-up functions are essentially the same in IS-54 as those used in IS-3 (same signalling channel format), we can use the same basic architecture for both call set-up and analog conversation.

The transmit signal flow is shown in Figure 16-20 and the receive signal flow is shown in Figure 16-21 for analog operation.

Figure 16-21 shows the following:

Analog Gain Block, A/D, Decimators -- These blocks amplify the input from the microphone(s), digitize it and decimate it.

Hands-Free Attenuator -- This is a programmable attenuator under the control of the hands-free unit.

Compressor -- This block performs a square-root function on the input data. Specific attack and decay times must be met when responding to steps in input amplitude.

Preemphasis -- The preemphasis block provides a +6 dB/octave (20 dB/decade) slope between 300 and 3000 Hz.

Bandpass Filter -- This is a standard 300-3400 Hz bandpass filter.

Limiter -- The limiter performs a clipping function to limit the maximum amplitude into the modulator, thus limiting the maximum frequency deviation.

Low-Pass Filter/Interpolator -- This filter guarantees that any high frequency components generated by the clipping action of the limiter do not exceed the specified limits.

SAT Tone Level Adjustment -- The level of the received SAT tone is adjusted to produce the desired frequency deviation. The tone must be phase-locked to the incoming SAT tone. The SAT tone is added to the voice signal.

DTMF -- DTMF tones to be transmitted are injected into the signal path here.

Signaling Tone (ST) -- The required 10 kHz (nominal) signalling tone is injected here.

Manchester Encoder and Scaler -- The logic high and low signals are scaled to produce the required ± 8 kHz frequency deviation of the carrier. This scaling is digitally with the resulting signal injected prior to the D/A converter.

D/A and Low-Pass Filter -- D/A converter and reconstruction filter. Includes interpolation.

RF/FM Modulator -- This block is the "radio" providing FM modulation and RF functions.

Figure 16-22 shows the following:

RF/FM Demodulator -- This block is the radio receiver providing RF and FM Demodulation functions.

A/D Converter -- Analog to digital converter includes decimation.

10kHz Bandpass Filter -- This filter detects the presence of a 10 kbps Manchester encoded binary signal (wideband signalling data). This filter is followed by a block that performs detection of bit and word synchronization. Presence of the signal precludes voice or SAT and vice versa.

6kHz Bandpass Filter -- Bandpass filter centered at 600 Hz to separate out and detect the presence of SAT tones.

SAT Detector -- This block discriminates between three SAT tones generating an indication of the presence, or the absence of, any SAT tone within 250 ms. This detection must be performed at least once every 250 ms. The recovered SAT is sent to the transmitter where it is phase-locked and retransmitted.

4kHz Low-Pass Filter/Decimator -- Standard voice band limiting filter.

Deemphasis -- This block is a network having a -6 dB/octave (-20 dB/decade) slope between 300 and 3000 Hz.

Expander -- This block computes the square of the incoming data. Specific attack and decay times must be met when responding to steps in input amplitude.

Mute -- Under certain conditions the voice signal must be muted. any transients created either entering or leaving mute must meet specific amplitude and duration requirements.

Call Progress Tones -- Tones such as dial tone and DTMF are generated to provide user feedback. These tones are injected into the receive path at this point. Note that the tone generator can also be used to produce the transmit side DTMF signals.

Hands-Free Attenuator -- This is a programmable attenuator under the control of the hands-free unit.

Interpolator, D/A, Analog Gain Stage -- The interpolator and D/A recreate the analog audio signal which is amplified by the programmable gain output driver(s). These functions are described in more detail in the Audio Interface hardware description section.

Note: The received voicepath is inactive when wideband signalling data is being received.

Once a call has been established, the mobile station can be directed to use a digital channel.

The foregoing description is intended to describe the present invention but not to limit it. Clearly numerous additions, substitutions and other modifications can be made to the invention without departing from the scope thereof as set forth in the appended claims.

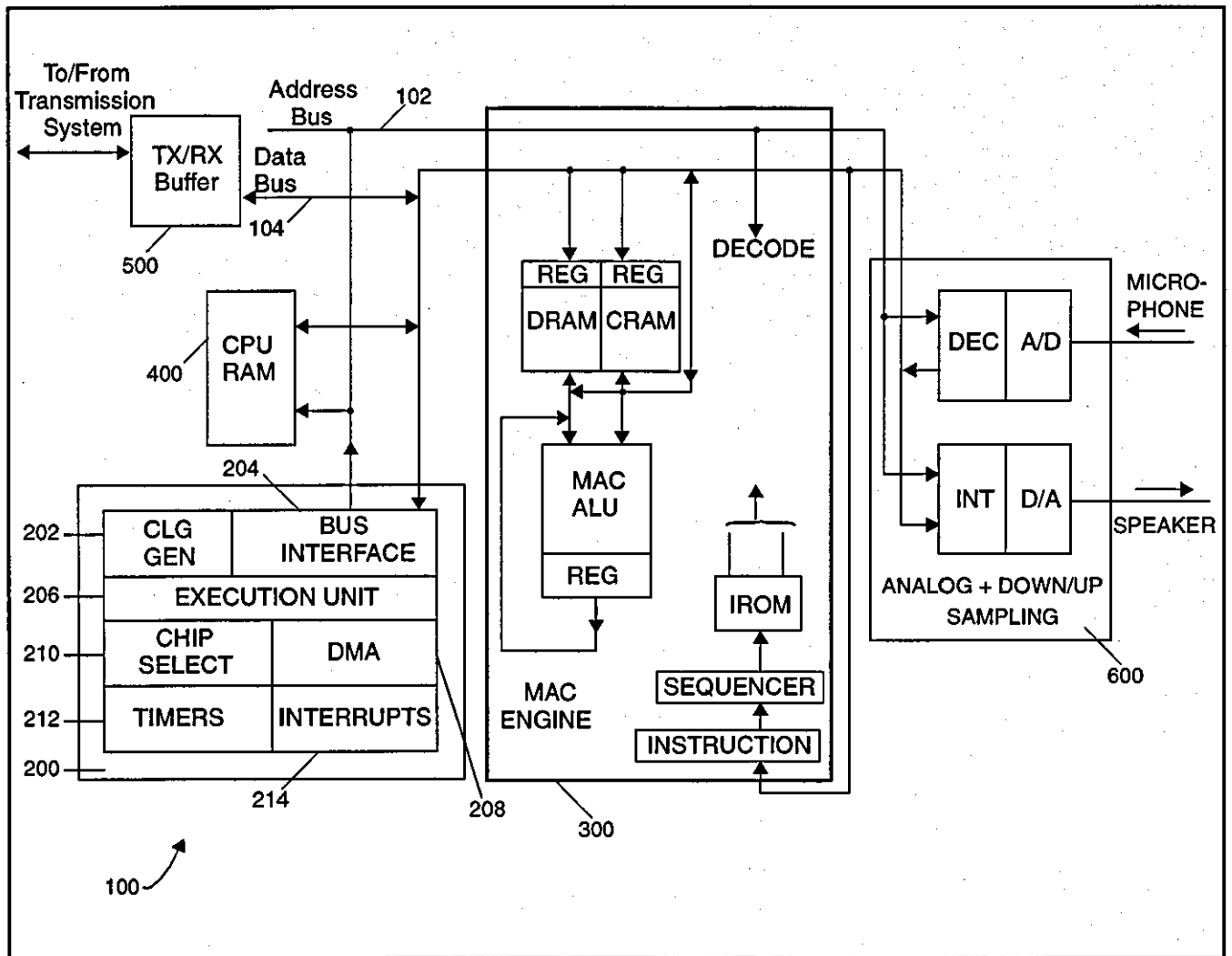


Figure 16-1

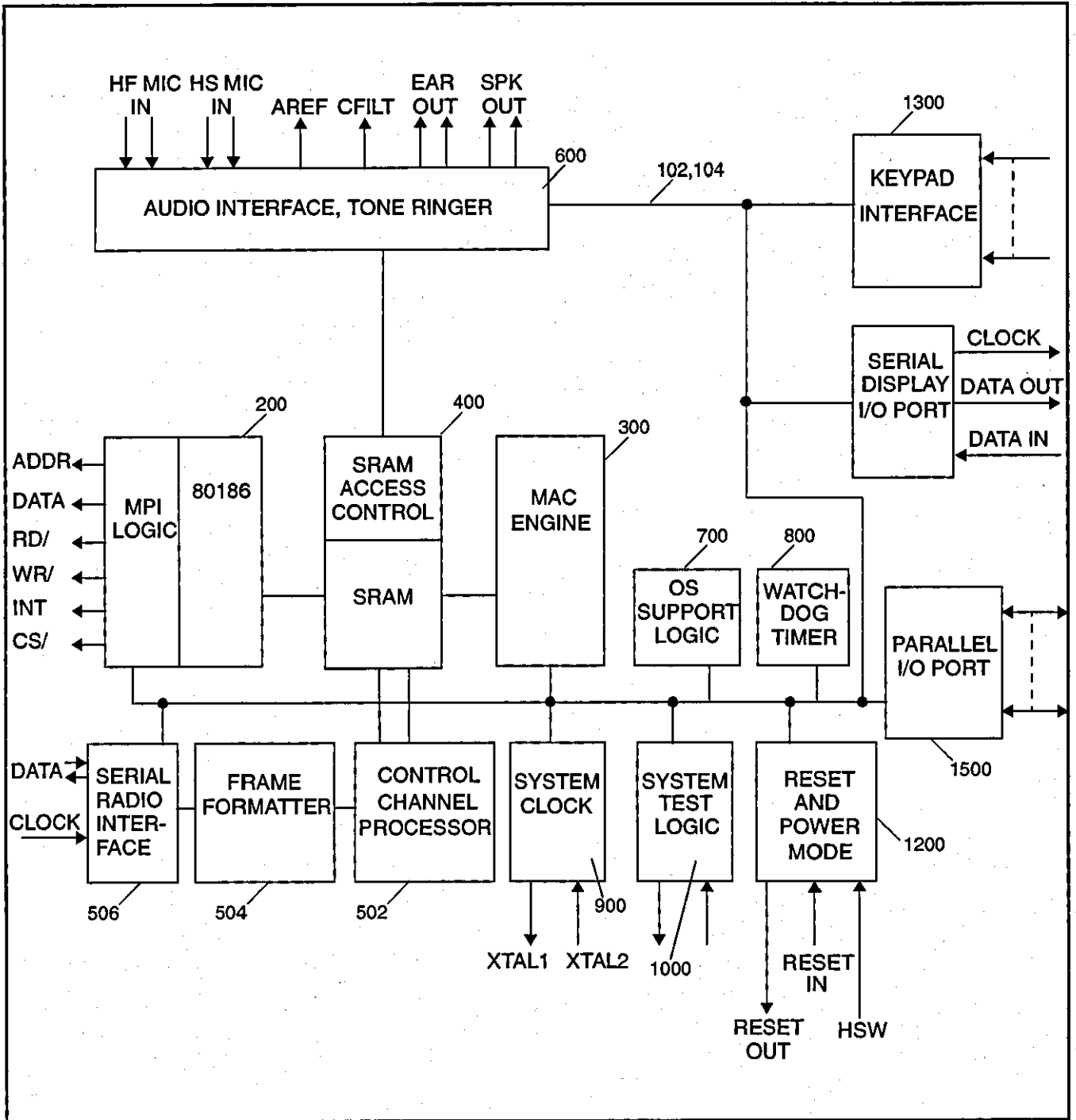


Figure 16-2

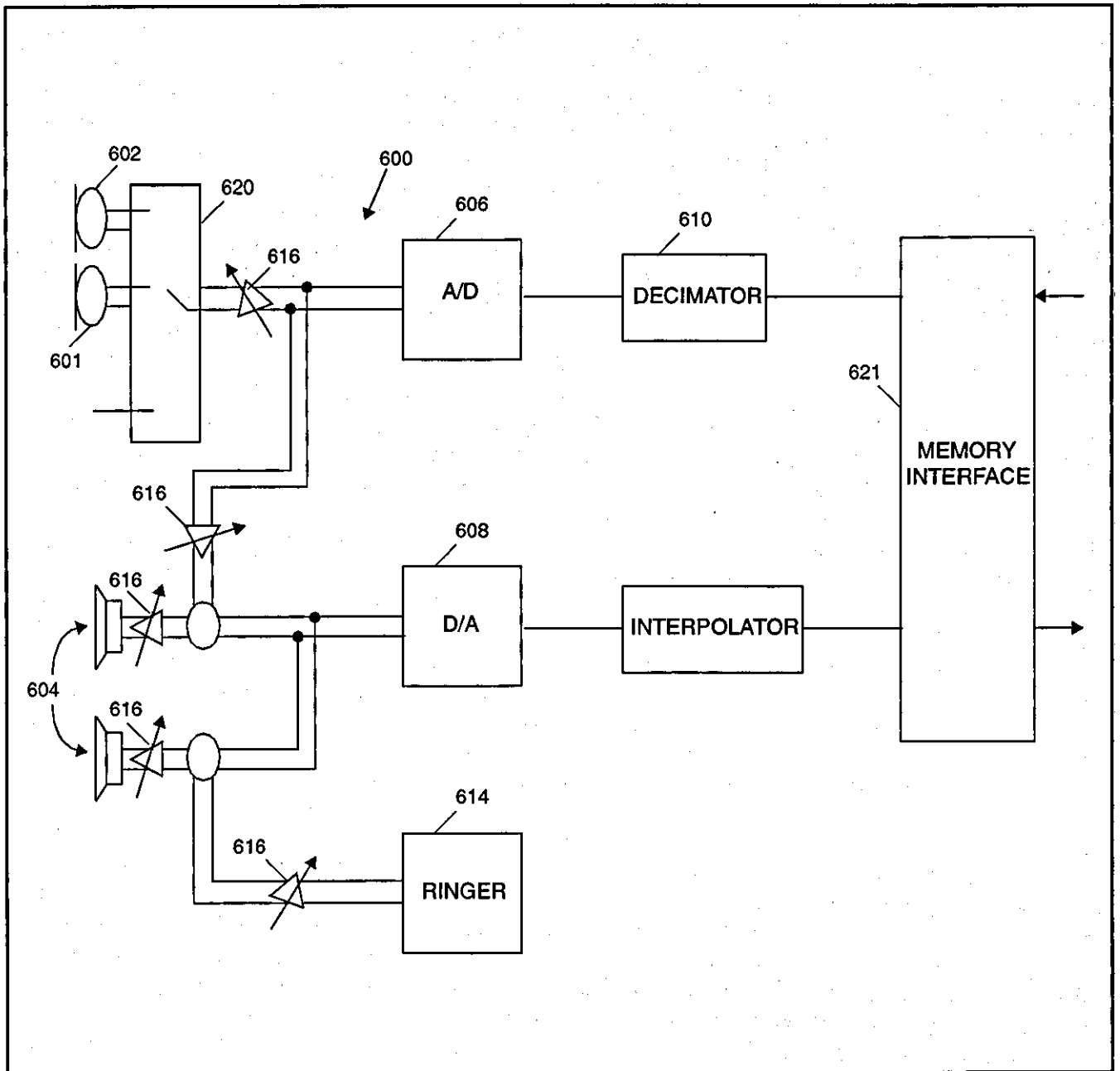


Figure 16-3

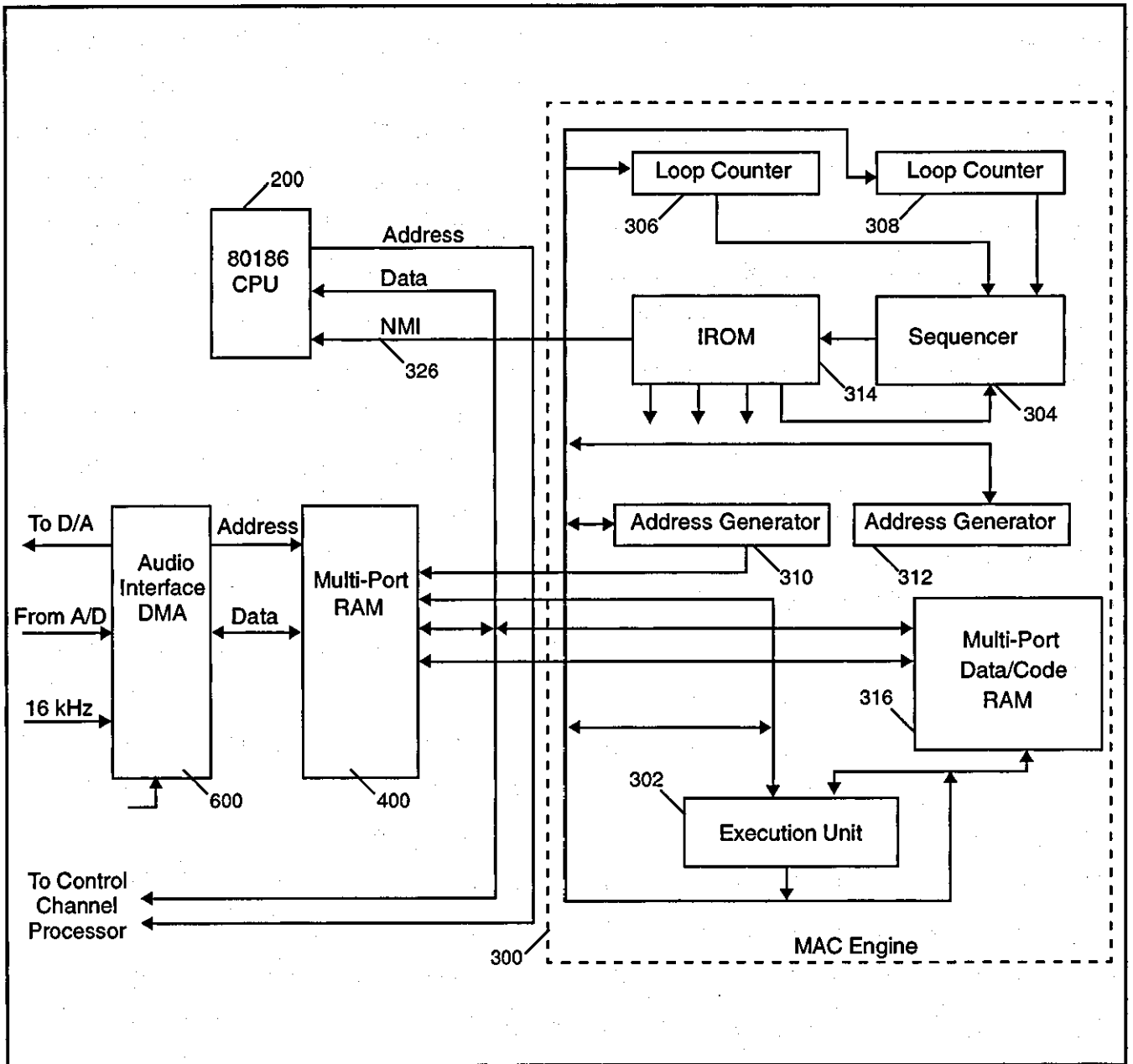


Figure 16-4

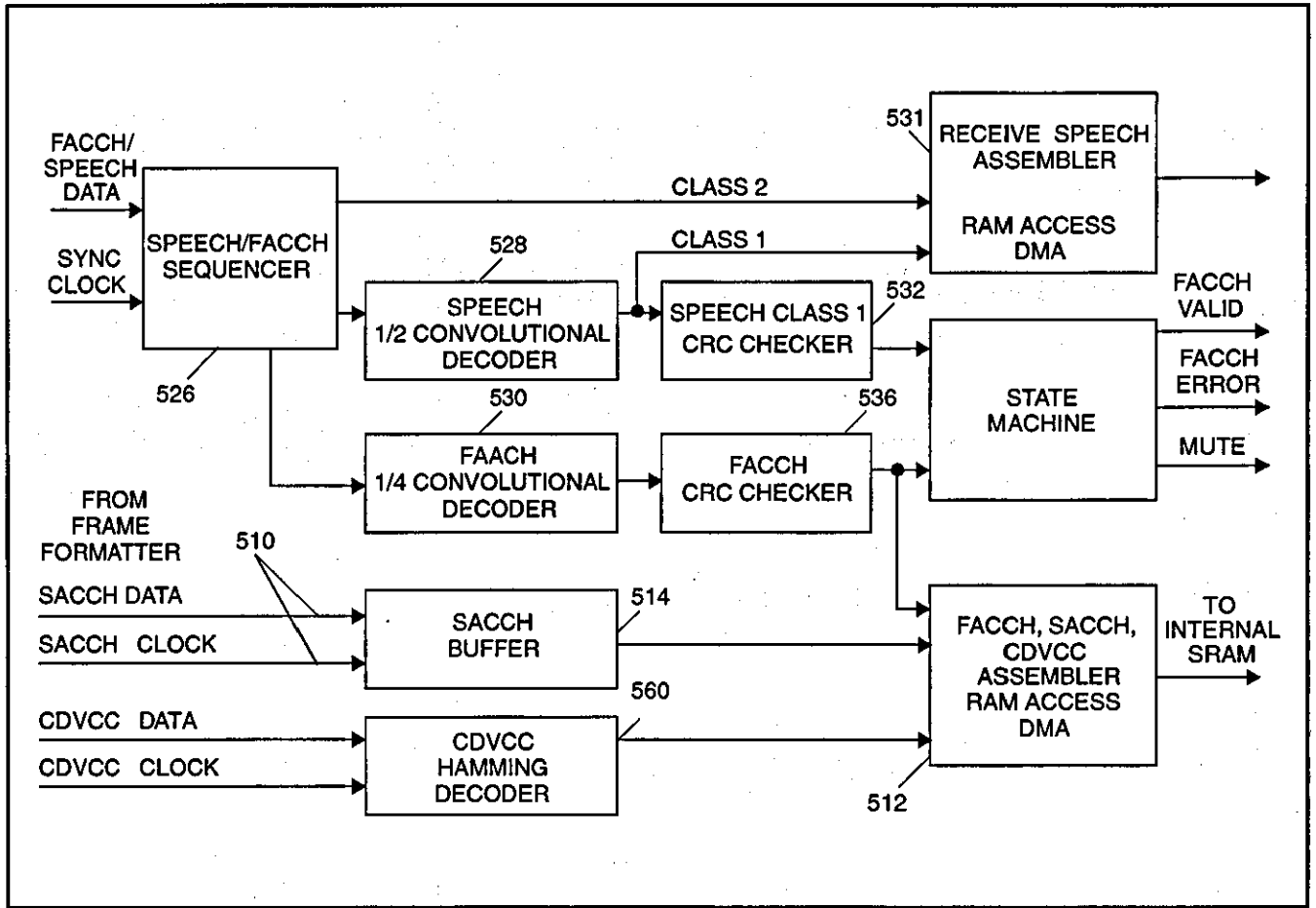


Figure 16-5

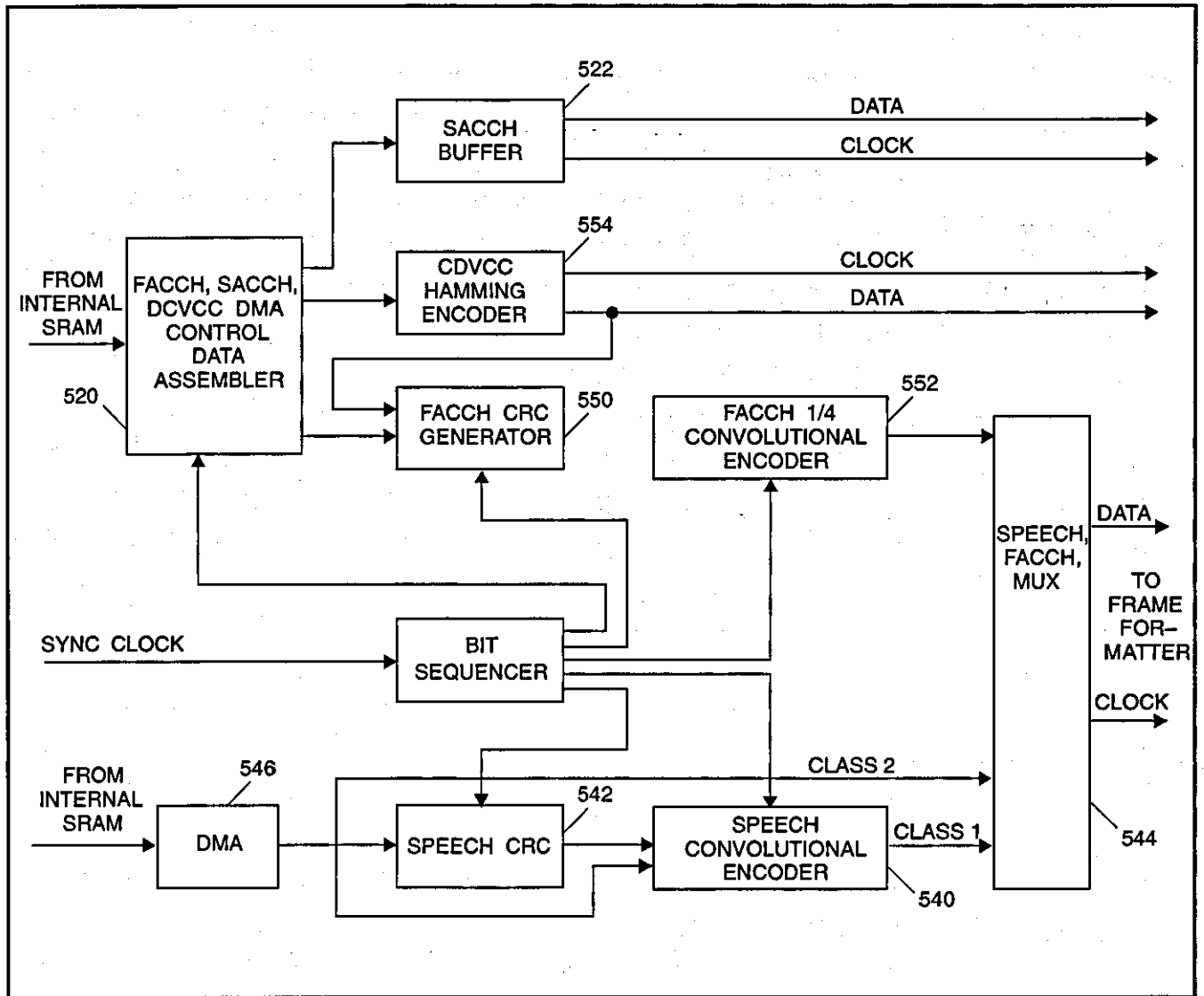


Figure 16-6

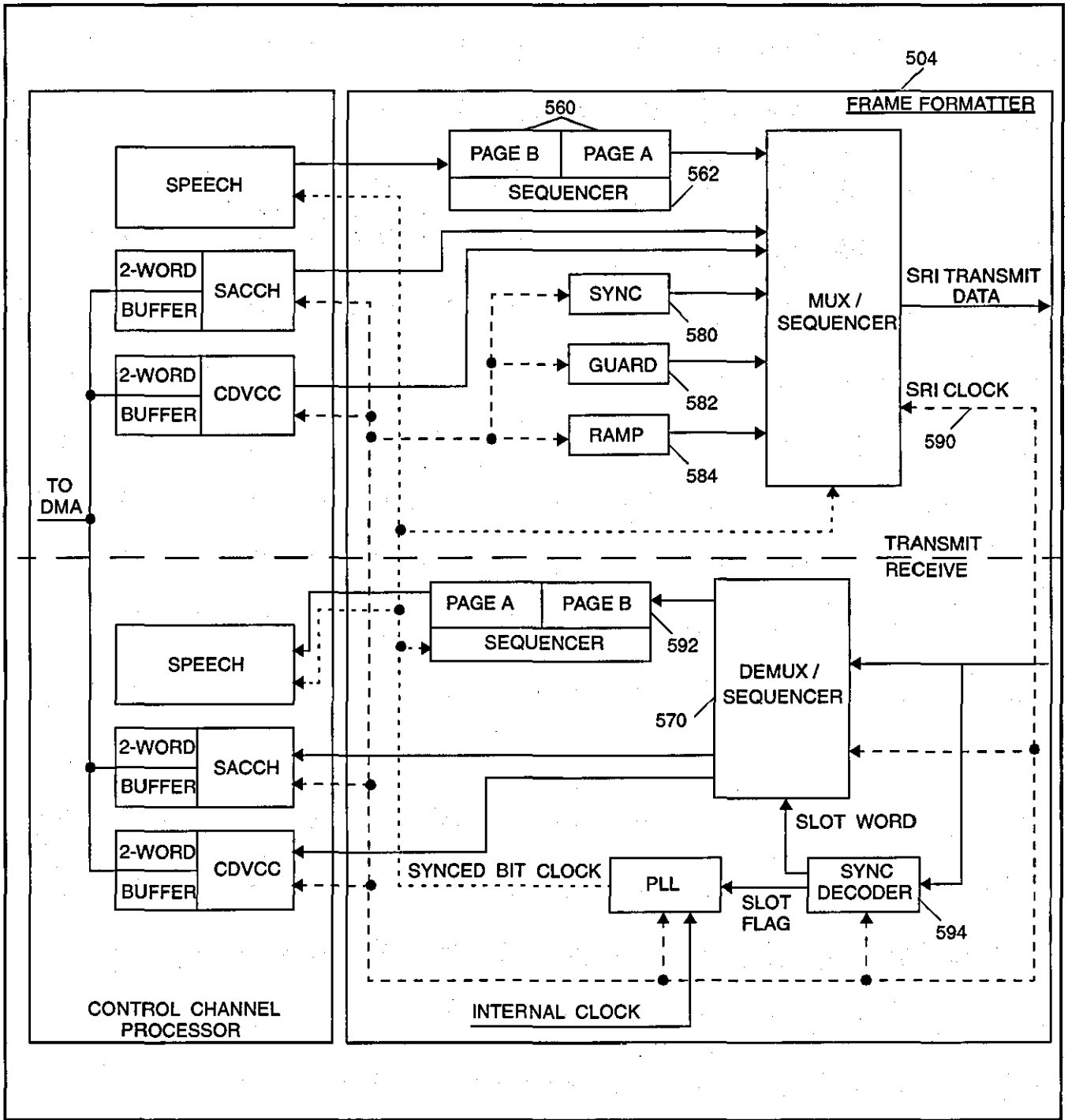


Figure 16-7

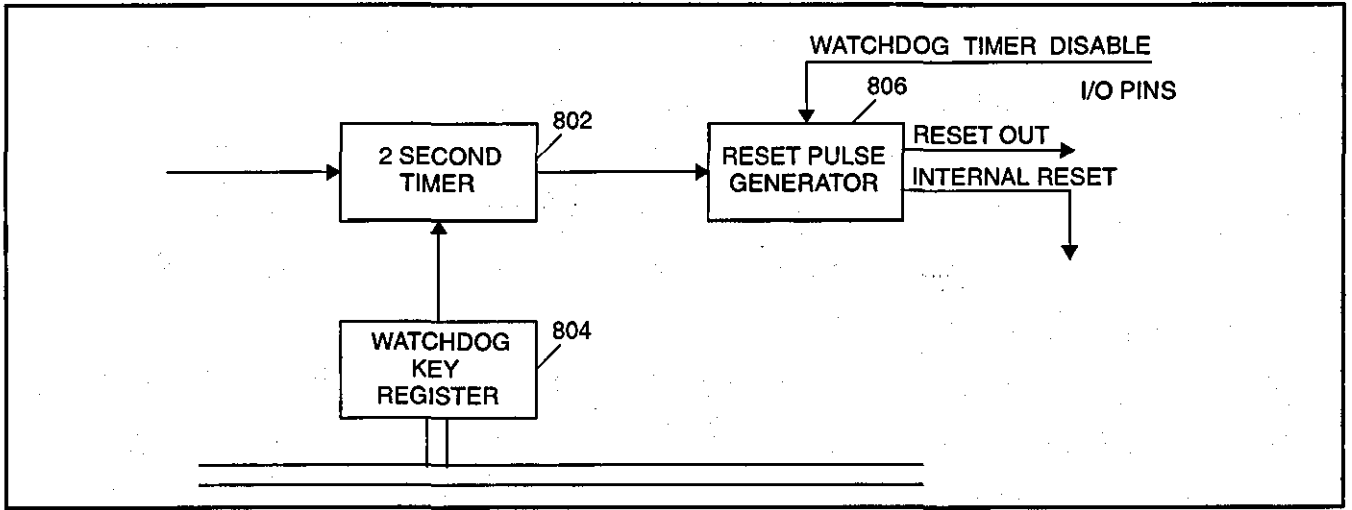


Figure 16-8

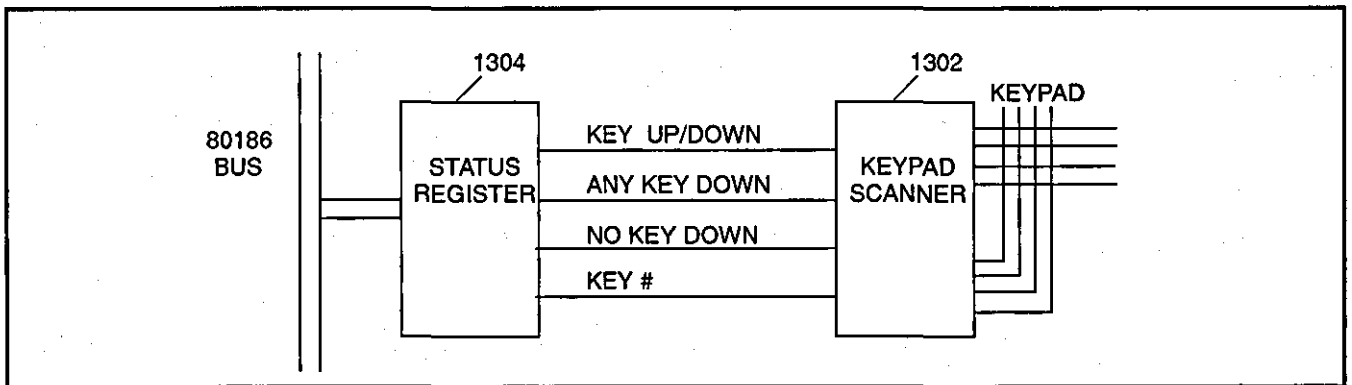


Figure 16-9

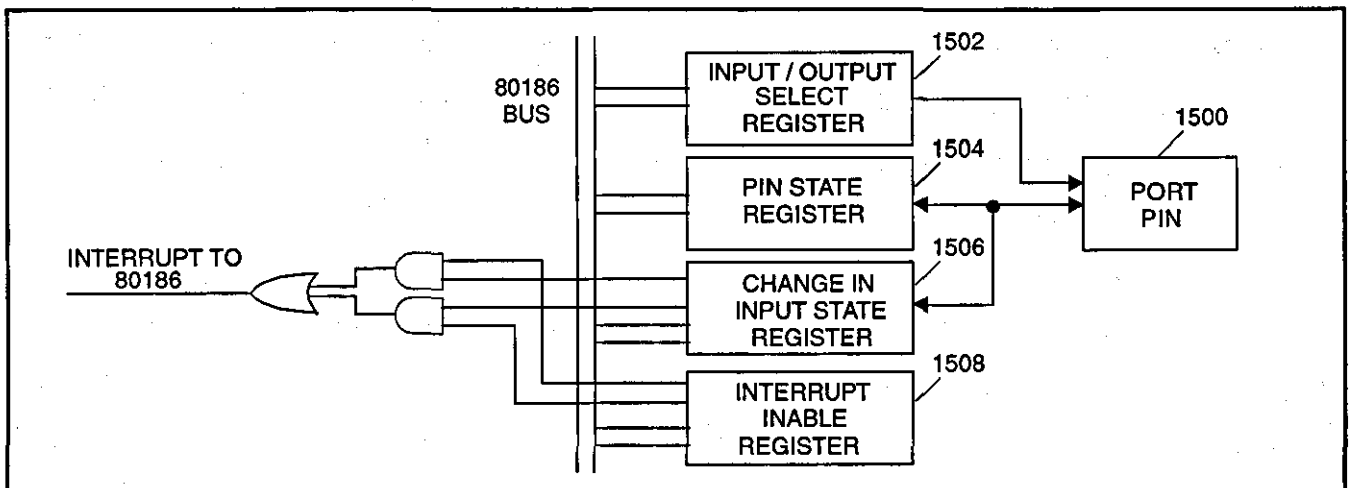


Figure 16-10

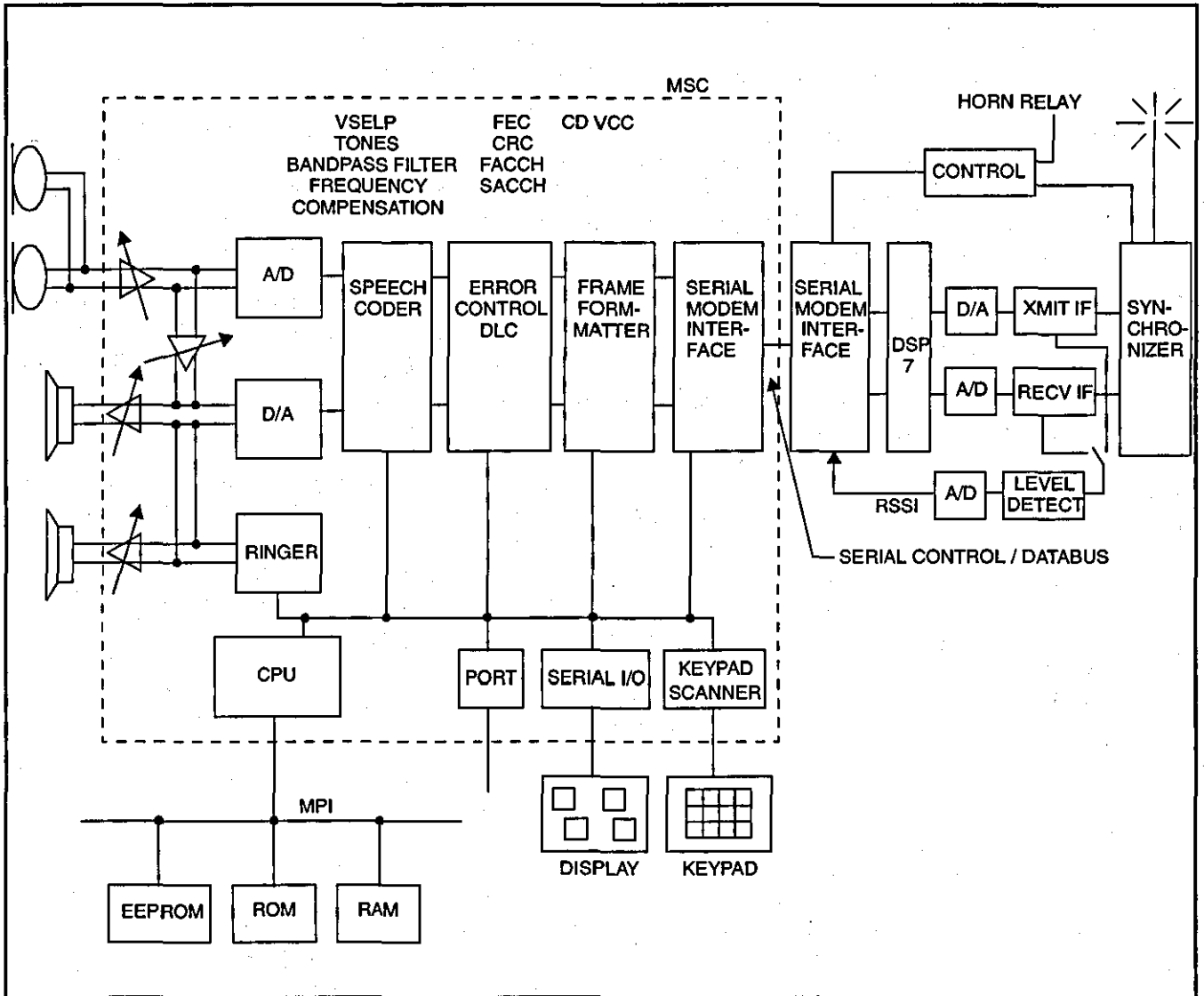


Figure 16-11

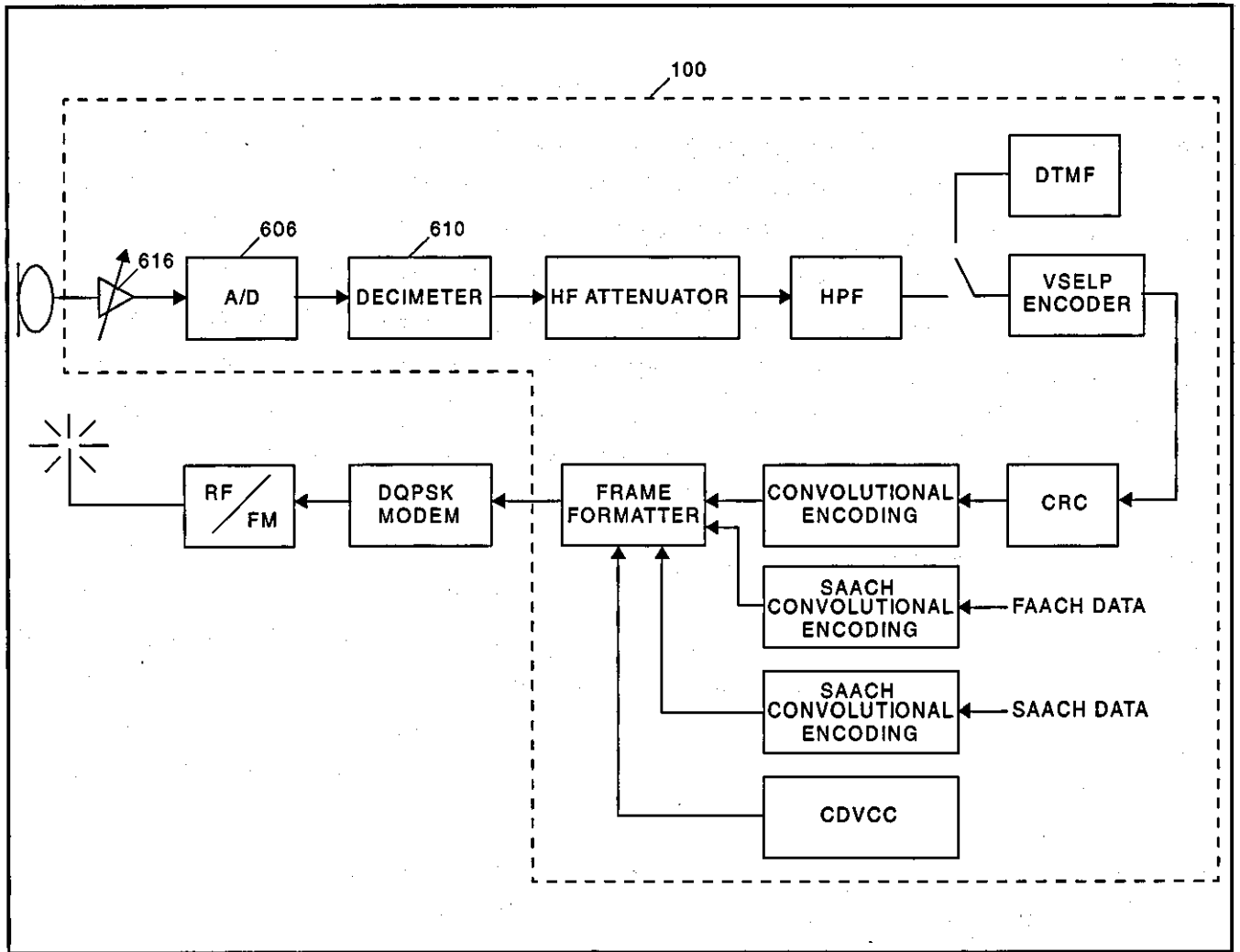


Figure 16-12

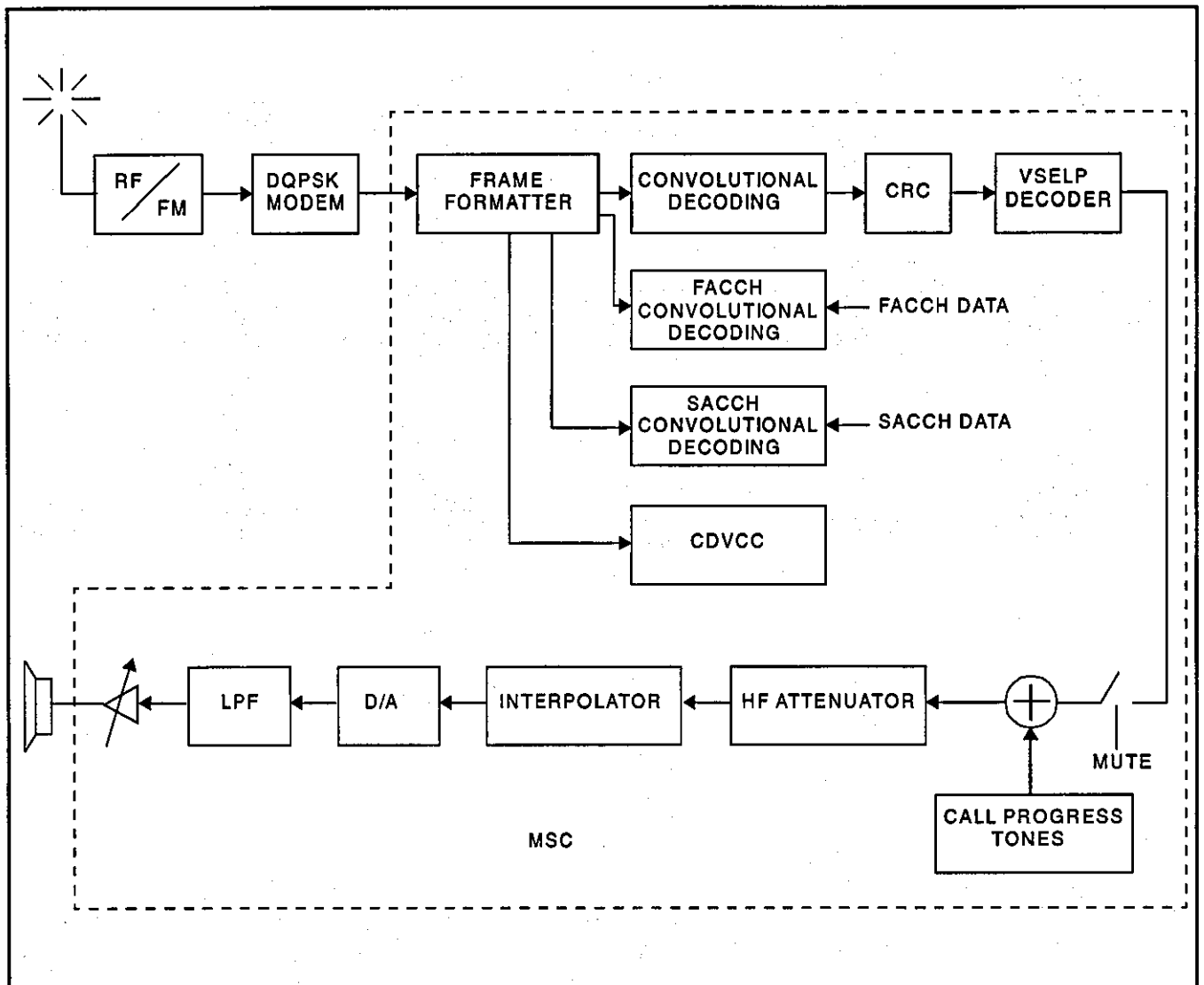


Figure 16-13

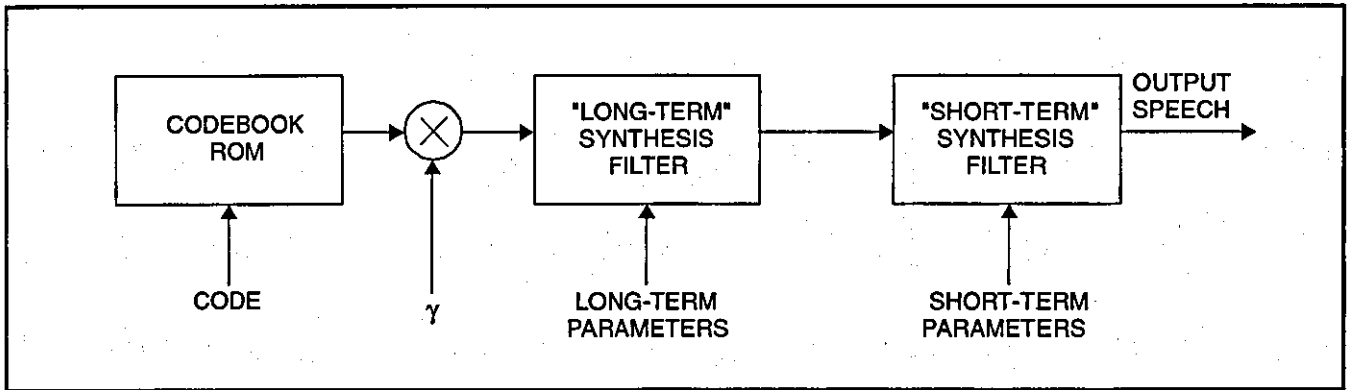


Figure 16-14

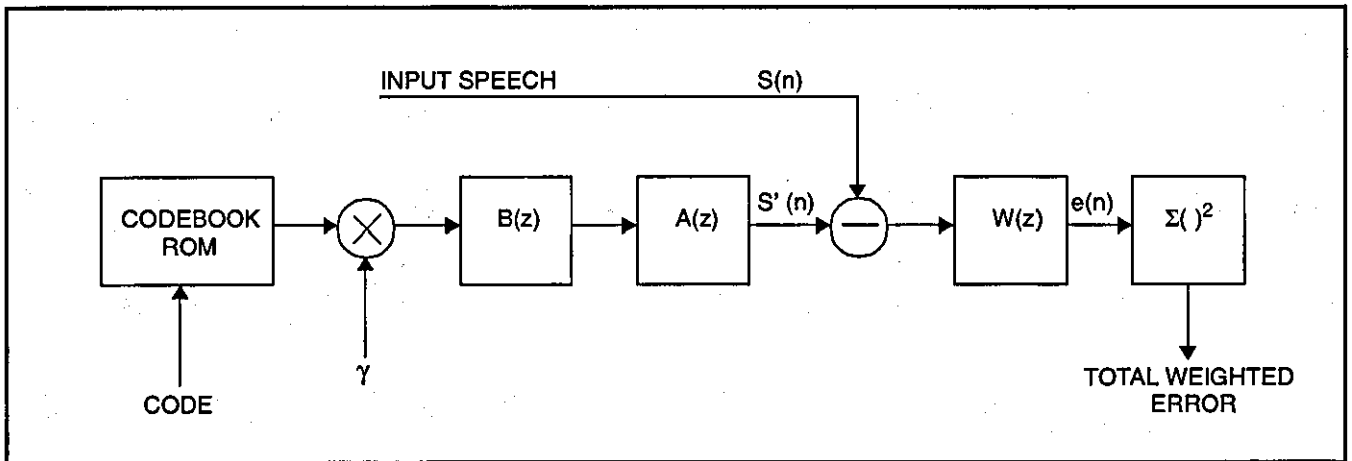


Figure 16-15

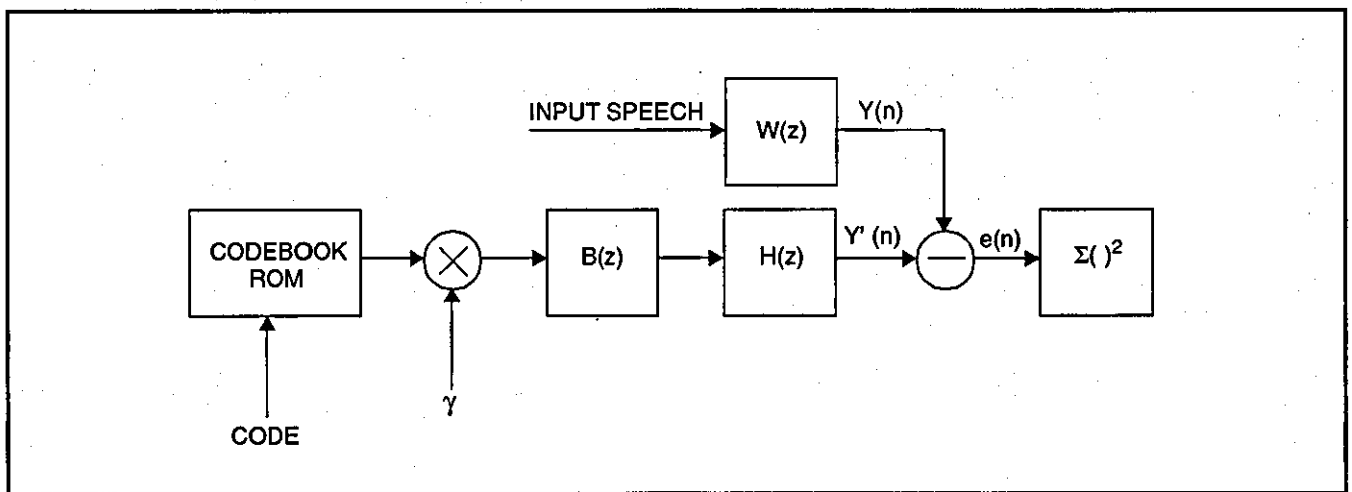


Figure 16-16

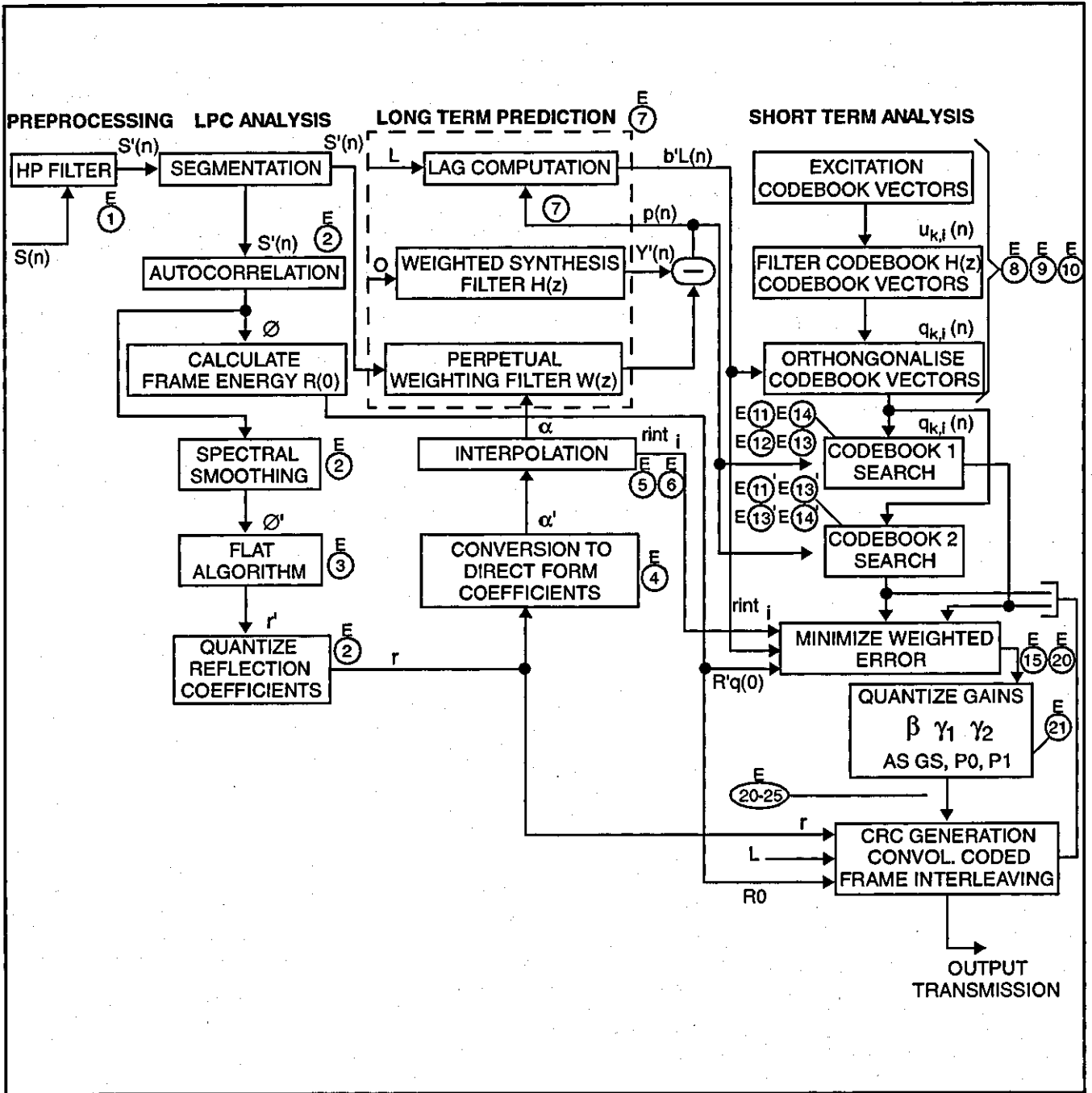


Figure 16-17

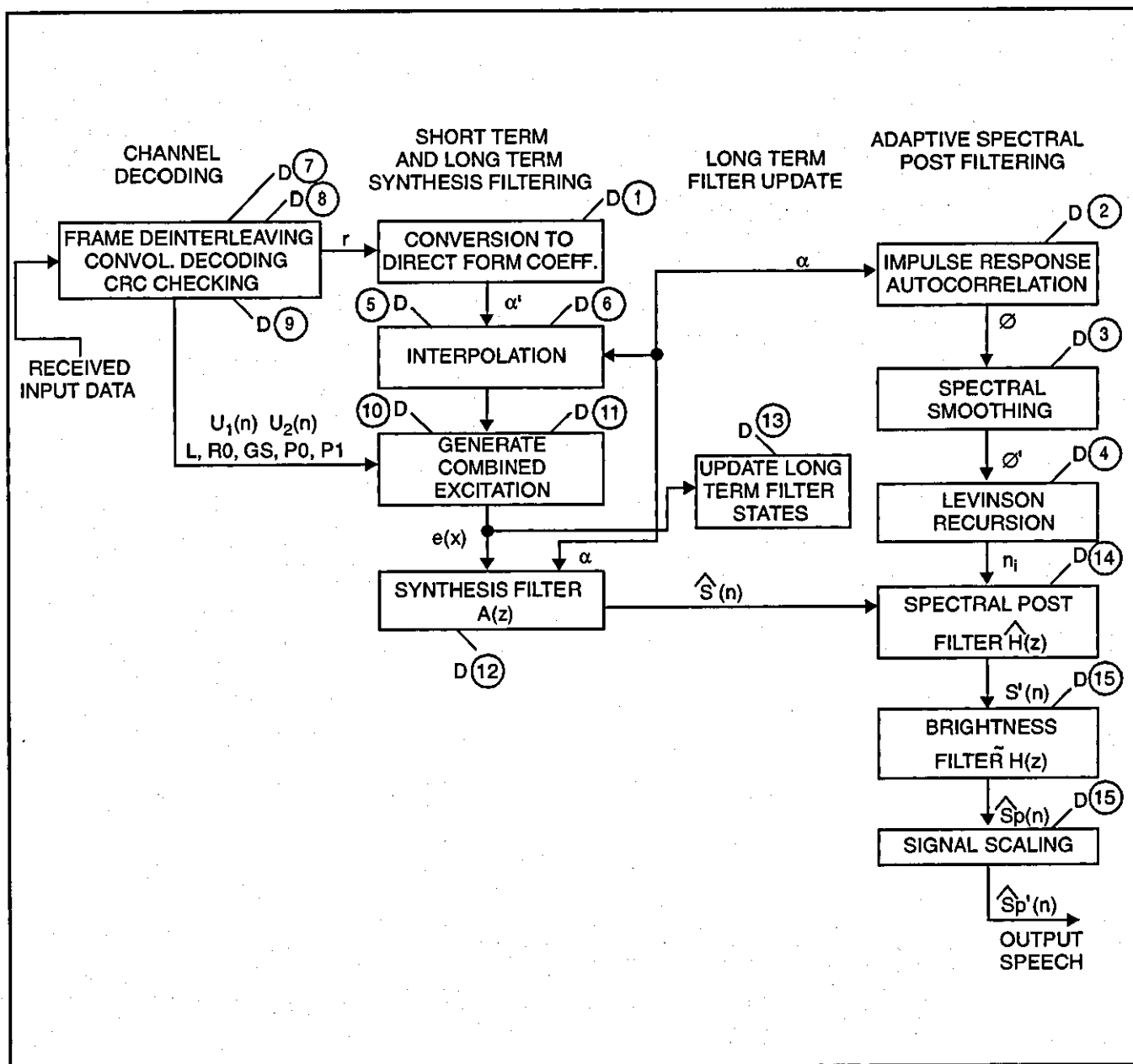


Figure 16-18

Figure 16-19a

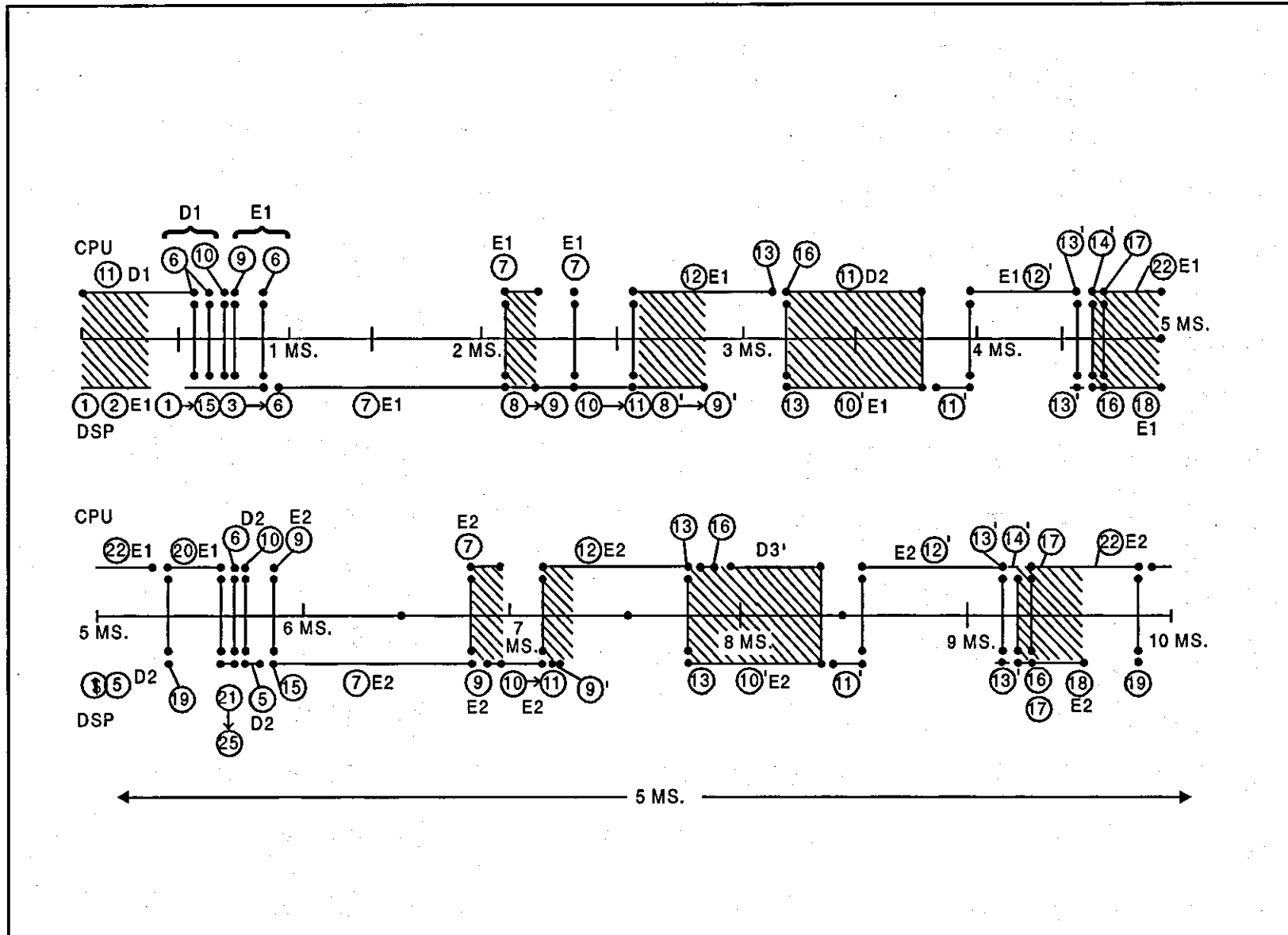
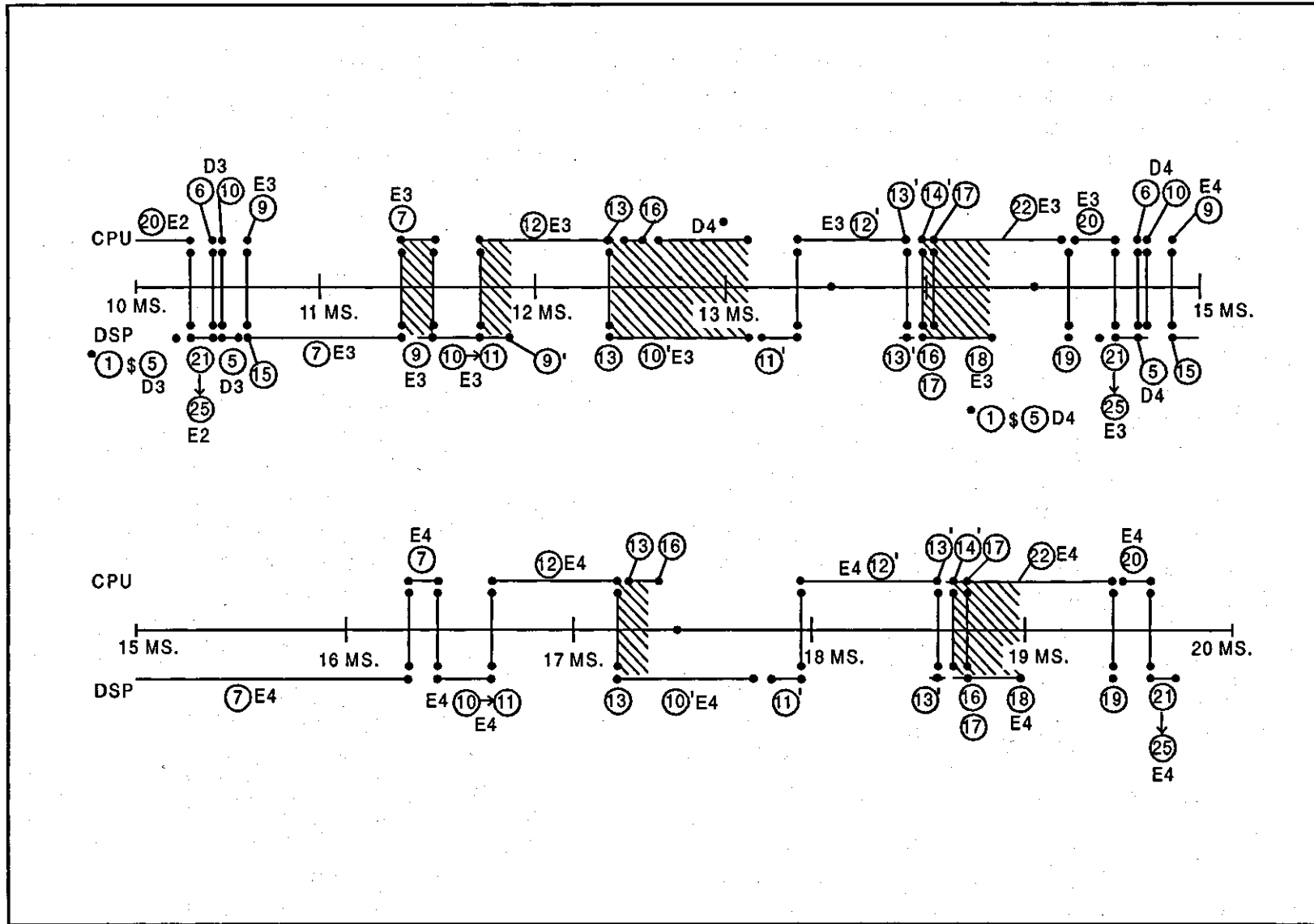


Figure 16-19b



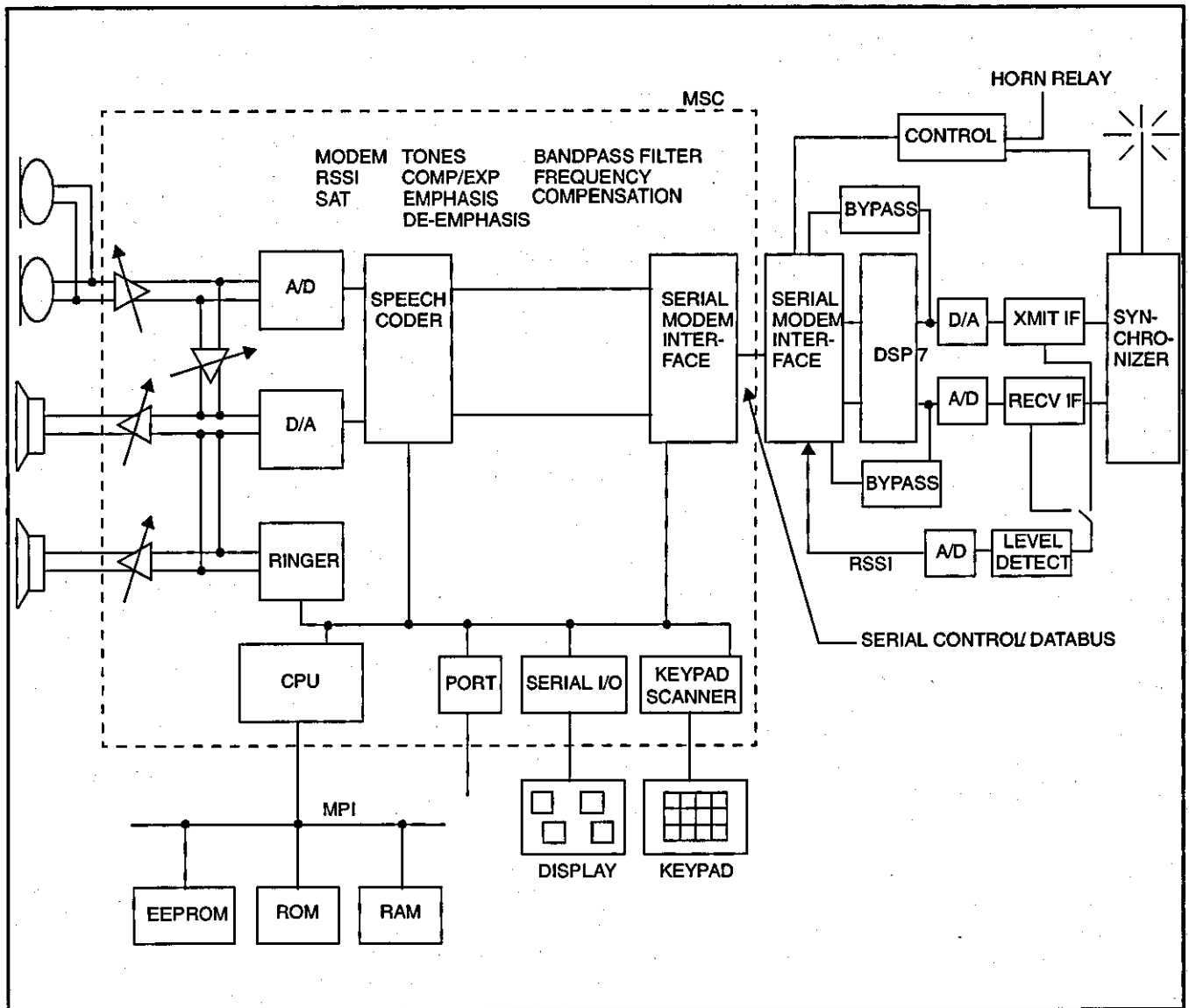


Figure 16-20

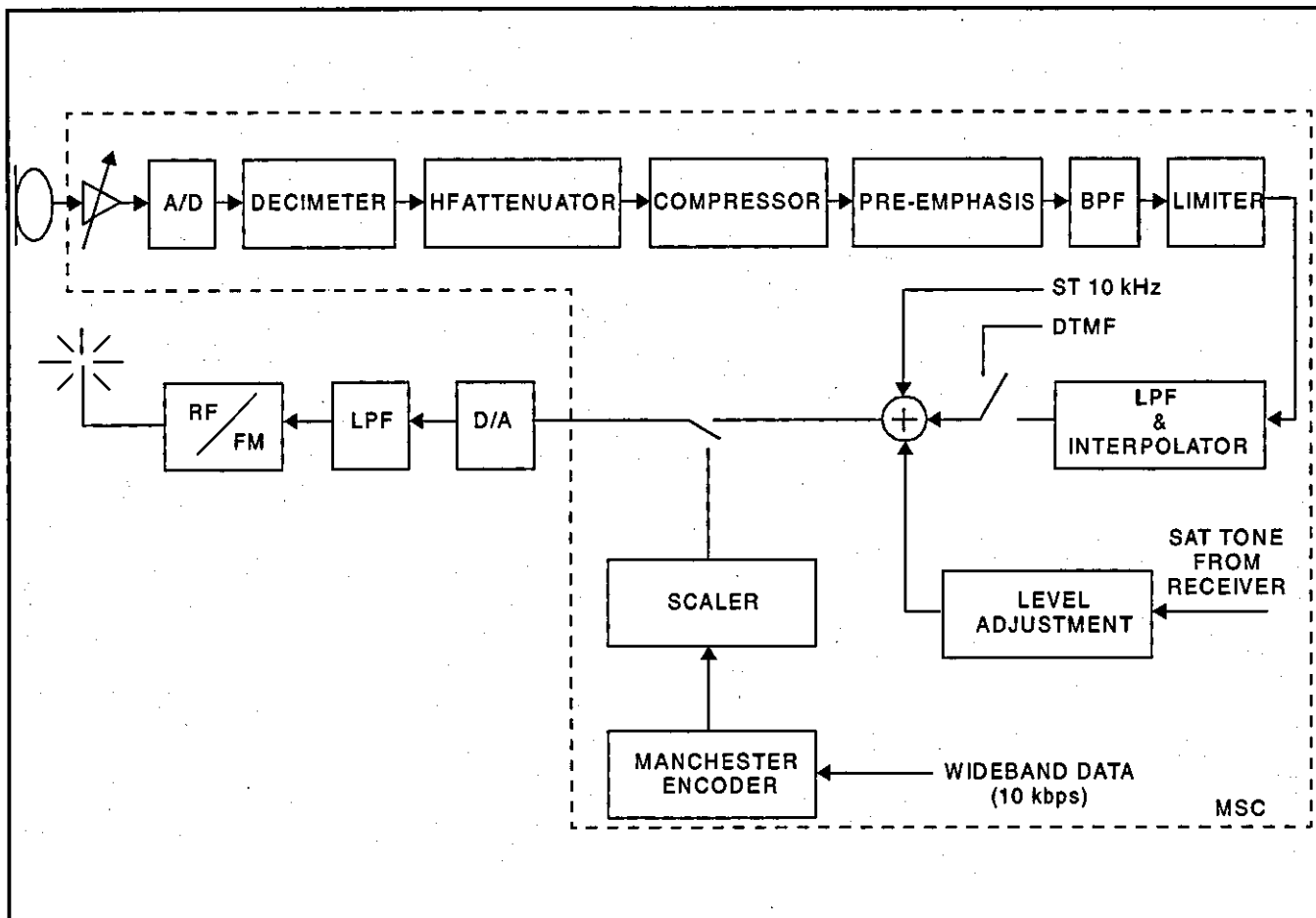


Figure 16-21

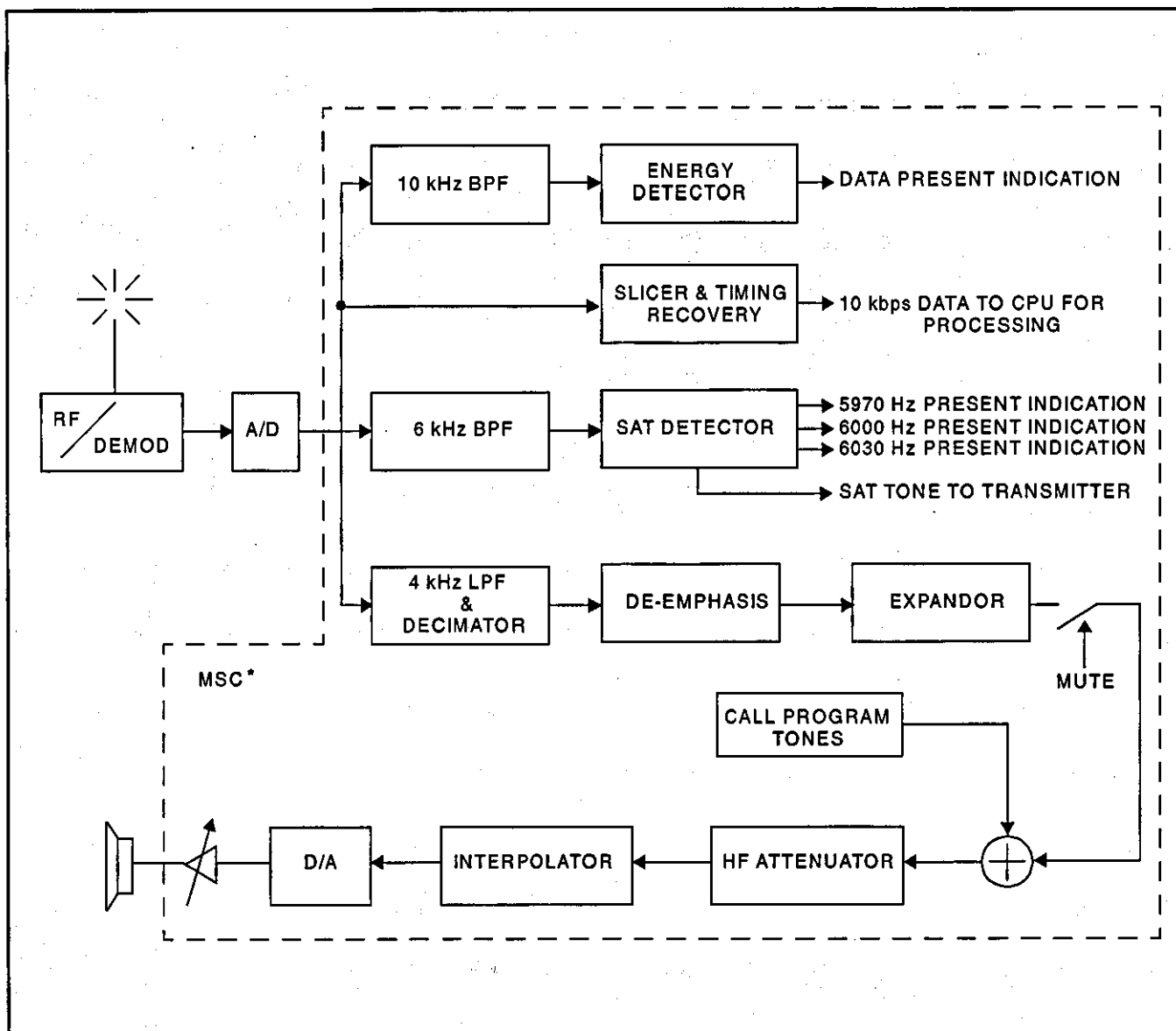


Figure 16-22

Discussion and Conclusion

This thesis is essentially a representation of the author's experiences in academia and industry. The author was very fortunate to have had the opportunity and guidance from his supervisor who instilled a certain discipline in his students to pay constant attention to detail and to the accuracy of data gathered from computer simulations. The validity of the data was verified against the anticipated outcome. Basically, the model and the method for developing the model was always under scrutiny. This discipline was very successfully adopted in the author's industrial environment and is still being followed presently in various projects undertaken.

The work at Loughborough was directed toward developing an effective and manufacturable detection process for distorted digital data transmitted over time varying channels or channels that exhibit multipath characteristics. In order to achieve the desired model of the system and implementation, a considerable number of different detection processes were developed and analyzed for complexity, memory utilization, clock speed, and power consumption. The thesis includes three illustrations of detection processes and the final model for the reader. A Fortran-based computer simulation model is also attached as an appendix.

The computer simulation modelling process goes through a number of phases. The most critical phase is phase one, where one has to create accurate mathematical representations of the various elements of the transmission chain, for instance signal encoding, shaping, propagation channel, matched filtering, and signal detection. An accurate representation of additive noise is essential for realistic bit-error-rate analysis. Transmitted data must be uniformly distributed and have the attributes of equal likelihood to span over all the possible signal values required for transmission.

The next phase entails conversion of the mathematical representation to a computer model in a high-level language such as Fortran. Initially, infinite precision is used to ensure the accuracy of the computer model in relation to the mathematical representation. This is very much an ideal-world scenario and

allows one to create a "true" resemblance of the theoretical model. We were on a mission to develop a system which could be implemented cheaply using off-the-shelf, fixed-point digital signal processors and central processing units. Therefore, the model was further modified to run in a fixed-point mode. The results generated were compared with the theoretical model, and over a number of iterations the difference was minimized. Obviously factors such as the dynamic range of the signal input at the receiver plays a significant part in determining the bit width. DSPs and CPUs generally do not have onboard functions such as SIN, COS, TAN, etc. These functions are compute intensive; therefore, creative methods were developed to emulate such functions efficiently in the available DSPs or CPUs.

Appendix 1

Patents

Patent Number	Title
4,994,993	System for detecting and correcting errors generated by arithmetic image moved logic units
4,994,801	Apparatus adaptable for use in effecting communications between an image moved between an analog device and a digital device
4,931,973	Method of generating updated transversal filter coefficients
4,839,842	Digital tone detection and generation
4,800,517	Word-sliced signal processor
16,219,642	Quantization using frequency and mean compensated frequency input data for robust image moved speech recognition
6,218,931	Home-appliance network with nodes identified by direct-sequence spreading codes
6,212,551	Digitized audio data attachment to text message for electronic mail
6,115,762	PC wireless communications utilizing an embedded antenna comprising a plurality of image moved radiating and receiving elements responsive to steering circuitry to form a direct antenna beam
6,108,390	Method of and apparatus for encoding of output symbol size
6,085,314	Central processing unit including APX and DSP cores and including selectable APX and DSP execution modes
6,070,136 recognition	Matrix quantization with vector quantization error compensation for robust speech
6,067,515	Split matrix quantization with split vector quantization error compensation and selective enhanced processing for robust speech recognition
6,044,343	Adaptive speech recognition with selective input data to a speech classifier

- 6,032,247 Central processing unit including APX and DSP cores which receives and processes APX and DSP instructions
- 6,032,116 Distance measure in a speech recognition system for speech recognition using frequency shifting factors to compensate for input signal frequency shifts
- 6,026,130 System and method for estimating a set of parameters for a transmission channel in a communication system
- 6,021,133 Communication processing method using a buffer array as a virtually circular buffer
- 6,014,719 Modulated bus computer system having filters with different frequency coverages for devices on the bus
- 6,009,391 Line spectral frequencies and energy features in a robust signal recognition system
- 6,008,856 Method of using an audio transmission signal to transmit video data image moved
- 6,003,003 Speech recognition system having a quantizer using a single robust codebook designed at multiple signal to noise ratios
- 5,991,725 System and method for enhanced speech quality in voice storage and retrieval systems
- 5,943,493 Retargetable VLIW computer architecture and method of executing a program corresponding to the architecture
- 5,890,187 Storage device utilizing a motion control circuit having an integrated digital image moved signal processing and central processing unit
- 5,794,068 CPU with DSP having function preprocessor that converts instruction sequences intended to perform DSP function into DSP function identifier
- 5,790,824 Central processing unit including a DSP function preprocessor which scans instruction sequences for DSP functions
- 5,784,640 CPU with DSP function preprocessor having look-up table for translating instruction sequences intended to perform DSP function into DSP macros
- 5,781,792 CPU with DSP having decoder that detects and converts instruction sequences intended to perform DSP function into DSP function identifier

- 5,771,394 Apparatus having signal processors for providing respective signals to master processor to notify that newly written data can be obtained from one or more memories
- 5,771,393 Servo loop control apparatus having master processor to control the apparatus and second processor dedicated to specific preprogrammed servo loop control tasks
- 5,768,613 Computing apparatus configured for partitioned processing
- 5,754,878 CPU with DSP function preprocessor having pattern recognition detector that uses table for translating instruction sequences intended to perform DSP function into DSP macros
- 5,745,648 Apparatus and method for analyzing speech signals to determine parameters expressive of characteristics of the speech signals
- 5,630,165 Servo system controlled by master and second processors through memory being accessed for read and write by processors in separate portions respectively
- 5,615,139 Apparatus and method for synthesizing a sinusoidal signal
- 5,507,037 Apparatus and method for discriminating signal noise from saturated signals and from high amplitude signals
- 5,459,750 Apparatus and method for discriminating and suppressing noise within an incoming signal
- 5,452,242 Method and apparatus for multiplying a plurality of numbers
- 5,424,970 Method and apparatus for multiplying a plurality of N numbers
- 5,379,388 Digital signal processing apparatus with sequencer designating program routines
- 5,369,791 Apparatus and method for discriminating and suppressing noise within an incoming signal
- 5,347,480 Digital signal processing apparatus
- 5,343,414 Apparatus for adaptively tuning to a received periodic signal
- 5,339,251 (Apparatus for adaptively tuning to a received periodic signal
- 5,299,233 Apparatus and method for attenuating a received signal in response to presence of noise
- 5,291,430 Method and apparatus for multiplying a plurality of numbers

- 5,204,642 Frequency controlled recursive oscillator having sinusoidal output
- 5,200,912 Apparatus for providing power to selected portions of a multiplying device
- 5,189,381 Apparatus for generating a sinusoidal output signal
- 5,136,537 Method and apparatus for determining the product of two numbers
- 5,043,932 Apparatus having modular interpolation architecture
- 5,003,309 Apparatus having shared architecture for analog-to-digital and for digital-to-analog signal conversion
- 4,999,626 Apparatus having a modular decimation architecture
- 4,996,528 Apparatus having shared modular architecture for decimation and interpolation

Appendix 2

Computer Simulation Program

#LISTING OF TNUXX.223191PBR(47LU) PRODUCED ON 22SEP80 AT 22.38.30

#OUTPUT BY LISTFILE IN TNUXX.XEM ON 23SEP80 AT 11.53.53 USING I122

DOCUMENT 223191PBR(47LU)

~~#LISTING RECEIVED FROM REMOTE HOST BY JANUS NETWORK~~

~~#OUTPUT BY UNIVERSITY OF NOTTINGHAM 1906A~~

#LISTING OF TNUXX.23191PB13483(1/B176) PRODUCED ON 21SEP80 AT 13.48.32

#OUTPUT ON UMRC 45 BY TNUXX.23191PB ON 22SEP80 AT 22.31.24 USING I511

DOCUMENT 23191PB13483(1/B176)

UNLABELED OLDFE
AGE 1

IDENT C159

UPDATE 1.3-508.

21/09780 12.39.23.

P *

***** IDENT C159
***** *DELETE C106-1.12;C106-1.12
NRUN=5
***** *DELETE C106-1.16;C106-1.16
IR=8

MODIFICATIONS / CONTROL CARDS

OPT2	NRUN=1	C106-1	12	0
OPT2	NRUN=5	C159	1	1
OPT2	IR=16	C106-1	16	0
OPT2	IR=8	C159	2	1

CORRECTION IDENTS ARE LISTED IN CHRONOLOGICAL ORDER OF INSERTION

OPT2 C106-1 C159

DECKS ARE LISTED IN THE ORDER OF THEIR OCCURRENCE ON A NEW PROGRAM LIBRARY IF ONE IS CREATED BY THIS UPDATE

YANKS\$ OPT2

DECKS WRITTEN TO COMPIL6 FILE

OPT2

THIS UPDATE REQUIRED 246008 WORDS OF SCM
AND 070008 WORDS OF LCM.


```

PROGRAM DST      76/76  OPT=1 TRACE      FTN 4,8+508      21/09780 12.41.49      P -
AGE      2
7          FRACC=0.7          C106*1      10
8          SNR=0.0          C106*1      11
9          NRUN=5          C159       1
10         DBS=4.0          C106*1     13
11         DO 5 LL=1,NRIIN  C106*1     14
12         SNR=SNR*DBR      C106*1     15
13         NEC=0          OPT2      50
14         NECR=0          OPT2      51
15         NECJ=0          OPT2      52
16         NECT=0          OPT2      53
17         IFA=1          OPT2      54
18         IFB=1          OPT2      55
19         IFC=1          OPT2      56
20         IFD=1          OPT2      57
21         NFA=1          OPT2      58
22         NFB=2          OPT2      59
23         NFD=3          OPT2      60
24         NED=6          OPT2      61
25         ISTEP=4          OPT2      62
26         LC=9          OPT2      63
27         IEXP=0          OPT2      64
28         IEXPD=0          OPT2      65
29         SV=8          OPT2      66
30         IDELAY=Z          OPT2      67
31         ICOL=8          OPT2      68
32         ICA=23          OPT2      69
33         M=SV          OPT2      70
34         SV1=SV-1          OPT2      71
35         JCOL=ICOL-1          OPT2      72
36         N=M+1          OPT2      73
37         IR=8          C159       2
38         JOEL=IDELAY+1  OPT2      75
39         IC=17          OPT2      76
40         ICY=IC-IR/ISTEP+2  OPT2      77
41         IC1=IC-1          OPT2      78
42         IC3=IC-3          OPT2      79
43         IC7=IC-7          OPT2      80
44         ICR=ICA+1          OPT2      81
45         IR2=2*IR          OPT2      82
46         IR4=IR*4          OPT2      83
47         C SET UP RECEIVER STONE OPT2      84
48         C          OPT2      85
49         READ(I,510) (COEFF(I),I=1,5),DCGAIN OPT2      88
50         DCGAIN=1.0/DCGAIN OPT2      95
51         CALL GDSRAP(XXX) OPT2      96
52         CALL GDSRAP(XXX) OPT2      97
53         DO 3 I=1,20          OPT2      98
54         ANI(I)=0.0          OPT2      99
55         ANQ(I)=0.0          OPT2     100
56         3 CONTINUE          OPT2     101
57         DO 4 I=1,16          OPT2     102
58         HI(I,1)=0.0          OPT2     103
59         HQ(I,1)=0.0          OPT2     104
60         4 CONTINUE          OPT2     105
61         IDEL=INT(DELT*2*SR) OPT2     106
62         IMPR=INRL*IDEL      OPT2     107
63         STIP=1/24.0          OPT2     108

```

Copyrighted material from the JSTOR Database

AGE

115

DO 10 I=1,5

OPT2 109

RAYL1(1)=0.0

OPT2 110

RAYLQ1(1)=0.0

OPT2 111

RAYL12(1)=0.0

OPT2 112

10 RAYLQ2(1)=0.0

OPT2 113

120

DO 40 J=1,50

OPT2 114

OP11(1)=G05ADF(0.5)=(RAYL1(1)*COEFF(1)+RAYL1(2)*COEFF(2))

OPT2 115

OPQ1(1)=G05ADF(0.5)=(RAYLQ1(1)*COEFF(1)+RAYLQ1(2)*COEFF(2))

OPT2 116

OP12(1)=G05ADF(0.5)=(RAYL12(1)*COEFF(1)+RAYL12(2)*COEFF(2))

OPT2 117

OPQ2(1)=G05ADF(0.5)=(RAYLQ2(1)*COEFF(1)+RAYLQ2(2)*COEFF(2))

OPT2 118

125

OP11(2)=OP11(1)*(RAYL1(3)*COEFF(3)+RAYL1(4)*COEFF(4))

OPT2 119

OPQ1(2)=OPQ1(1)*(RAYLQ1(3)*COEFF(3)+RAYLQ1(4)*COEFF(4))

OPT2 120

OP12(2)=OP12(1)*(RAYL12(3)*COEFF(3)+RAYL12(4)*COEFF(4))

OPT2 121

OPQ2(2)=OPQ2(1)*(RAYLQ2(3)*COEFF(3)+RAYLQ2(4)*COEFF(4))

OPT2 122

130

X1(2)=OP11(2)-RAYL1(5)*COEFF(5)

OPT2 125

Y1(2)=OPQ1(2)-RAYLQ1(5)*COEFF(5)

OPT2 126

X2(2)=OP12(2)-RAYL12(5)*COEFF(5)

OPT2 125

Y2(2)=OPQ2(2)-RAYLQ2(5)*COEFF(5)

OPT2 126

RAYL1(3)=X1(2)

OPT2 127

RAYLQ1(3)=Y1(2)

OPT2 128

RAYL12(3)=X2(2)

OPT2 129

RAYLQ2(3)=Y2(2)

OPT2 130

X1(2)=X1(2)+DCGAIN

OPT2 131

Y1(2)=Y1(2)+DCGAIN

OPT2 132

X2(2)=X2(2)+DCGAIN

OPT2 133

Y2(2)=Y2(2)+DCGAIN

OPT2 134

140

RAYL1(4)=RAYL1(3)

OPT2 135

RAYL1(3)=OP11(2)

OPT2 136

RAYL1(2)=RAYL1(1)

OPT2 137

RAYL1(1)=OP11(1)

OPT2 138

RAYLQ1(4)=RAYLQ1(3)

OPT2 139

RAYLQ1(3)=OPQ1(2)

OPT2 140

RAYLQ1(2)=RAYLQ1(1)

OPT2 141

RAYLQ1(1)=OPQ1(1)

OPT2 142

RAYL12(4)=RAYL12(3)

OPT2 143

150

RAYL12(3)=OP12(2)

OPT2 144

RAYL12(2)=RAYL12(1)

OPT2 145

RAYL12(1)=OP12(1)

OPT2 146

RAYLQ2(4)=RAYLQ2(3)

OPT2 147

RAYLQ2(3)=OPQ2(2)

OPT2 148

RAYLQ2(2)=RAYLQ2(1)

OPT2 149

155

40 RAYLQ2(1)=OPQ2(1)

OPT2 150

110P=100+1100P

OPT2 151

DO 45 K=1,110P

OPT2 152

OP11(1)=G05ADF(0.5)=(RAYL1(1)*COEFF(1)+RAYL1(2)*COEFF(2))

OPT2 153

OPQ1(1)=G05ADF(0.5)=(RAYLQ1(1)*COEFF(1)+RAYLQ1(2)*COEFF(2))

OPT2 154

OP12(1)=G05ADF(0.5)=(RAYL12(1)*COEFF(1)+RAYL12(2)*COEFF(2))

OPT2 155

OPQ2(1)=G05ADF(0.5)=(RAYLQ2(1)*COEFF(1)+RAYLQ2(2)*COEFF(2))

OPT2 156

OP11(2)=OP11(1)*(RAYL1(3)*COEFF(3)+RAYL1(4)*COEFF(4))

OPT2 157

OPQ1(2)=OPQ1(1)*(RAYLQ1(3)*COEFF(3)+RAYLQ1(4)*COEFF(4))

OPT2 158

165

OP12(2)=OP12(1)*(RAYL12(3)*COEFF(3)+RAYL12(4)*COEFF(4))

OPT2 159

OPQ2(2)=OPQ2(1)*(RAYLQ2(3)*COEFF(3)+RAYLQ2(4)*COEFF(4))

OPT2 160

X1(1)=OP11(2)-RAYL1(5)*COEFF(5)

OPT2 161

Y1(1)=OPQ1(2)-RAYLQ1(5)*COEFF(5)

OPT2 162

X2(1)=OP12(2)-RAYL12(5)*COEFF(5)

OPT2 163

Y2(1)=OPQ2(2)-RAYLQ2(5)*COEFF(5)

OPT2 164

170

RAYL1(5)=X1(1)

OPT2 165

ASF

PROGRAM DS1

76776

OPT=1 TRACE

FTN 4,8+508

21709780 12.41.49

P

175

```

RAYLQ1(5)=Y1(1)
RAYL12(5)=X2(1)
RAYLQ2(5)=Y2(1)
X1(1)=X1(1)*DCGAIN
Y1(1)=Y1(1)*DCGAIN
X2(1)=X2(1)*DCGAIN
Y2(1)=Y2(1)*DCGAIN

```

```

OPT2 166
OPT2 167
OPT2 168
OPT2 169
OPT2 170
OPT2 171
OPT2 172

```

180

```

RAYL11(4)=RAYL11(3)
RAYL11(3)=OPT1(2)
RAYL11(2)=RAYL11(1)
RAYL11(1)=OPT1(1)
RAYLQ1(4)=RAYLQ1(3)
RAYLQ1(3)=OPQ1(2)
RAYLQ1(2)=RAYLQ1(1)
RAYLQ1(1)=OPQ1(1)
RAYL12(4)=RAYL12(3)
RAYL12(3)=OPT2(2)
RAYL12(2)=RAYL12(1)
RAYL12(1)=OPT2(1)
RAYLQ2(4)=RAYLQ2(3)
RAYLQ2(3)=OPQ2(2)
RAYLQ2(2)=RAYLQ2(1)
RAYLQ2(1)=OPQ2(1)

```

```

OPT2 173
OPT2 174
OPT2 175
OPT2 176
OPT2 177
OPT2 178
OPT2 179
OPT2 180
OPT2 181
OPT2 182
OPT2 183
OPT2 184
OPT2 185
OPT2 186
OPT2 187
OPT2 188

```

185

190

195

200

205

210

215

220

225

```

RAY1(K)=X1(1)
RAY2(K)=Y1(1)
RAY3(K)=X2(1)
45 RAY4(K)=Y2(1)
DO 664 J=1,IMPR
DO 664 J=1,4
R(I,J)=0.0
64 Q(I,J)=0.0
DO 666 J=1,SV
DO 677 J=1,SV
DVR(I,J)=0.0
DVJ(I,J)=0.0
677 CONTINUE
686 CONTINUE
DO 910 IA=1,IR
ISTORAX(IA,IC)=1
DO 920 IB=1,IC1
ISTOR(IA,IB)=1
ISTORA(IA,IB)=1
920 CONTINUE
910 CONTINUE
DO 921 FI=1,7H
DO 919 IJ=1,ISTEP
VR(IJ,11)=0.0
VJ(IJ,11)=0.0
919 CONTINUE
BTA(IJ)=0.0
BSTB(IJ)=0.0
JFB(IJ)=0
JPC(IJ)=1
DA(IJ)=1
DB(IJ)=1
921 CONTINUE
DO 922 II=1,ICB

```

```

OPT2 189
OPT2 190
OPT2 191
OPT2 192
OPT2 193
OPT2 194
OPT2 195
OPT2 196
OPT2 197
OPT2 198
OPT2 199
OPT2 200
OPT2 201
OPT2 202
OPT2 203
OPT2 204
OPT2 205
OPT2 206
OPT2 207
OPT2 208
OPT2 209
OPT2 210
OPT2 211
OPT2 212
OPT2 213
OPT2 214
OPT2 215
OPT2 216
OPT2 217
OPT2 218
OPT2 219
OPT2 220
OPT2 221
OPT2 222

```

Copyrighted material from John G. ...

AGE	PROGRAM DS1	76/76 OPTET TRACE	FTN 4,8+508	21/09/00 12.01.97
3				
4				
5				
6				
7				
8	230	922 CONTINUE		OPT2 223
9		DO 930 I=1,IR4		OPT2 226
10		IPPOINT(I)=1		OPT2 225
11		930 CONTINUE		OPT2 226
12		IERR=0		OPT2 227
13	235	DO 950 I=1,IR		OPT2 228
14		DO 950 I=1,IR		OPT2 229
15		DO 950 I=1,IR		OPT2 230
16		IFF=IFF+1		OPT2 231
17		NSEL(I)=IFF		OPT2 232
18		NEST(I)=IFF		OPT2 233
19	240	950 CONTINUE		OPT2 234
20		950 CONTINUE		OPT2 235
21				OPT2 236
22		DO 923 I3=1,IR		OPT2 237
23		DO 924 I4=1,JDEL		OPT2 238
24	245	TSA(I3,I4)=0.0		OPT2 239
25		TSD(I3,I4)=0.0		OPT2 240
26		924 CONTINUE		OPT2 241
27		923 CONTINUE		OPT2 242
28		DO 925 I5=1,IR4		OPT2 243
29	250	STC(I5)=0.0		OPT2 244
30		STD(I5)=0.0		OPT2 245
31		TSX(I5)=0.0		OPT2 246
32		TSY(I5)=0.0		OPT2 247
33		TSZ(I5)=0.0		OPT2 248
34	255	DO 926 I6=1,JDEL		OPT2 249
35		TSE(I5,I6)=0.0		OPT2 250
36		TSD(I5,I6)=0.0		OPT2 251
37		926 CONTINUE		OPT2 252
38		925 CONTINUE		OPT2 253
39	260	C INTRODUCE NEW COST FUNCTION		OPT2 254
40		U(1)=0.0		OPT2 255
41		DO 110 I=1,IR		OPT2 256
42		U(I)=1E+6		OPT2 257
43		110 CONTINUE		OPT2 258
44	265	SIGMA=10*(1-SNR/20.0)		OPT2 259
45		DO 400 IRUN=1,ILOP		OPT2 260
46		IERR=0		OPT2 261
47		X1(1)=RAY1(IRUN)		OPT2 262
48		Y1(1)=RAY2(IRUN)		OPT2 263
49	270	X2(1)=RAY3(IRUN)		OPT2 264
50		Y2(1)=RAY4(IRUN)		OPT2 265
51		CONST1=(X1(1)-X1(2))*STIP		OPT2 266
52		CONST2=(Y1(1)-Y1(2))*STIP		OPT2 267
53		CONST3=(X2(1)-X2(2))*STIP		OPT2 268
54	275	CONST4=(Y2(1)-Y2(2))*STIP		OPT2 269
55		68 DO 200 ISYM=1,24		OPT2 270
56		DO 67 J=1,IMPR		OPT2 271
57		DO 67 I=1,4		OPT2 272
58		R(I,J)=0.0		OPT2 273
59	280	67 R(I,J)=0.0		OPT2 274
60		KKK=ISYM-1		OPT2 275
61		A1NCR=KKK*CONST1		OPT2 276
62		B1NCR=KKK*CONST2		OPT2 277
63		DO 80 I=1,IMPI		OPT2 278
64	285	R(I,1)=C1(I)*(X1(2)+A1NCR-Y1(2)+B1NCR)-C0(I)*(Y1(2)+B1NCR+X1(2)+A1		OPT2 279

Copyright © 1997 by IBM Corporation

```

      INCR)
80 Q(1,1)=Q(1,1)*X1(2)+AINCR*Y1(2)+BINCR)+GD(1)*X1(2)+AINCR*Y1(2)=BI OPT2 280
      INCR)
      DO 81 I=1,IMPI
      R(2,1)=R(1,1) OPT2 281
      Q(2,1)=R(1,1) OPT2 282
      R(3,1)=R(1,1) OPT2 283
      Q(3,1)=R(1,1) OPT2 284
      R(4,1)=R(1,1) OPT2 285
      295 81 Q(4,1)=R(1,1) OPT2 286
      AINCR=KKK*CONST5 OPT2 287
      BINCR=KKK*CONST4 OPT2 288
      DO 350 I=1,IMPL OPT2 289
      R(1,1+IDEL)=R(1,1+IDEL)+CID(1)*X2(2)+AINCR*Y2(2)+BINCR)+GD(1)*Y OPT2 290
      300 12(2)+BINCR*X2(2)+AINCR) OPT2 291
      330 Q(1,1+IDEL)=Q(1,1+IDEL)+CID(1)*Y2(2)+BINCR*X2(2)+AINCR)+GD(1)*X OPT2 292
      12(2)+AINCR*Y2(2)+BINCR) OPT2 293
      DO 340 I=1,IMPL OPT2 294
      R(2,1+IDEL)=R(1,1+IDEL) OPT2 295
      305 Q(2,1+IDEL)=R(1,1+IDEL) OPT2 296
      R(3,1+IDEL)=R(1,1+IDEL) OPT2 297
      Q(3,1+IDEL)=R(1,1+IDEL) OPT2 298
      R(4,1+IDEL)=R(1,1+IDEL) OPT2 299
      340 Q(4,1+IDEL)=R(1,1+IDEL) OPT2 300
      590 DO 600 I=1,IMPR OPT2 301
      HI(1,1)=Q(1,1)+R(1,1)*0.5 OPT2 302
      HO(1,1)=Q(1,1)+R(1,1)*0.5 OPT2 303
      YMIN=0.0 OPT2 304
      YMINN=0.0 OPT2 305
      315 DO 911 ITX=1,SV OPT2 306
      SVR(ITX)=HI(2+ITX-1,1) OPT2 307
      SVJ(ITX)=HO(2+ITX-1,1) OPT2 308
      SVRR(ITX)=HI(2+ITX,1) OPT2 309
      SVJJ(ITX)=HO(2+ITX,1) OPT2 310
      320 SVXX(ITX)=ABS(SVR(ITX))+ABS(SVJ(ITX)) OPT2 311
      SVXX(ITX)=ABS(SVRR(ITX))+ABS(SVJJ(ITX)) OPT2 312
      IFSVX(ITX)=YMIN)911,911,913 OPT2 313
      913 YMIN=SVX(ITX) OPT2 314
      JTX=ITX OPT2 315
      325 911 CONTINUE OPT2 316
      IFSVXX(ITX)=YMINN)911,911,9133 OPT2 317
      9133 YMINN=SVXX(ITX) OPT2 318
      JTX=ITX OPT2 319
      911 CONTINUE OPT2 320
      330 IEXP=JTX-1 OPT2 321
      IEXP=JTX-1 OPT2 322
      DO 604 NA=1,SV1 OPT2 323
      NB=SV+NA+1 OPT2 324
      DO 605 NC=1,SV OPT2 325
      SVR(NB,NC)=DVR(NB-1,NC) OPT2 326
      DVJ(NB,NC)=DVJ(NB-1,NC) OPT2 327
      DVRR(NB,NC)=DVRR(NB-1,NC) OPT2 328
      DVJJ(NB,NC)=DVJJ(NB-1,NC) OPT2 329
      605 CONTINUE OPT2 330
      340 604 CONTINUE OPT2 331
      DO 606 NG=1,SV OPT2 332
      DVR(1,NG)=SVR(NG) OPT2 333
      606 CONTINUE OPT2 334
      604 CONTINUE OPT2 335
      DO 606 NG=1,SV OPT2 336
      DVR(1,NG)=SVR(NG) OPT2 336
  
```

Copyrighted material from John Callaghan

AGE

7

5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64

343

```

DVJ(1,NG)=SVJ(NG)
SVRR(1,NG)=SVRR(NG)
DVJJ(1,NG)=SVJJ(NG)
404 CONTINUE
SVZ=YMIND*FRAC
SVZ2=YMIND*FRAC
DO 914 I=1,JYX

```

```

OPT2 337
OPT2 338
OPT2 339
OPT2 340
OPT2 341
OPT2 342
OPT2 343
OPT2 344
OPT2 345
OPT2 346
OPT2 347
OPT2 348
OPT2 349
OPT2 350
OPT2 351
OPT2 352
OPT2 353
OPT2 354
OPT2 355
OPT2 356
OPT2 357
OPT2 358
OPT2 359
OPT2 360
OPT2 361
OPT2 362
OPT2 363
OPT2 364
OPT2 365
OPT2 366
OPT2 367
OPT2 368
OPT2 369
OPT2 370
OPT2 371
OPT2 372
OPT2 373
OPT2 374
OPT2 375
OPT2 376
OPT2 377
OPT2 378
OPT2 379
OPT2 380
OPT2 381
OPT2 382
OPT2 383
OPT2 384
OPT2 385
OPT2 386
OPT2 387
OPT2 388
OPT2 389
OPT2 390
OPT2 391
OPT2 392
OPT2 393

```

350

```

914 CONTINUE
GO TO 916
915 LTX=1
IEXR=LTX-1
355 916 CONTINUE
DO 917 I=1,JYXX
IF(SVXX(I)*SVZ2)917,918,918

```

360

```

917 CONTINUE
GO TO 9919
918 LTX=1
IEXPP=LTX-1
9919 CONTINUE
B=IPIX(1,0)+0*GD$AAF(0,5)
IF(B.EQ.3)B=4

```

365

```

C SET DELAY LINE
DO 120 JA=1,ICA
JB=ICB-JA+1
XST(JB)=XST(JB-1)

```

370

```

120 CONTINUE
XST(1)=B
BB=XST(1CB)
C

```

375

```

DO 202 I=1,SV
VR(I,I)=SVR(I)+SVJ(I)
YRR(1,I)=SVRR(I)-SVJJ(I)
YJ(1,I)=SVR(I)+SVJ(I)
YJJ(1,I)=SVRR(I)+SVJJ(I)
YR(2,I)=-YJ(1,I)
YRR(2,I)=-YJJ(1,I)
YJ(2,I)=VR(1,I)
YJJ(2,I)=YRR(1,I)
YR(3,I)=YR(1,I)
YRR(3,I)=YRR(1,I)
YJ(3,I)=YJ(1,I)
YJJ(3,I)=YJJ(1,I)
YR(4,I)=YJ(1,I)
YRR(4,I)=YJJ(1,I)
YJ(4,I)=YR(1,I)
YJJ(4,I)=YRR(1,I)

```

380

385

390

```

202 CONTINUE
C FEED DATA ELEMENT TO A FEEDFORWARD FILTER
C
IP(1)=IDAT(B)
AA=JDAT(N,1)
AB=JDAT(B,2)
XA=0.0
XB=0.0
XAA=0.0

```

395

56
57
58
59
60
61
62
63
64

```

PROGRAM DS1      76776  OPT=1 TRACE      FTN 4.8+508      21/09780  12.41.49      P =
AGE      0
400      XBB=0.0      OPT2      394
        DA(1)=AA      OPT2      395
        DB(1)=AB      OPT2      396
        DO 130 J=1,M      OPT2      397
        JJ=M+1-J      OPT2      398
405      XA=XA+YR(JP(JJ+1),JJ+1)      OPT2      399
        XB=XB+YJ(JP(JJ+1),JJ+1)      OPT2      400
        XAA=XAA+YRR(JP(JJ+1),JJ+1)      OPT2      401
        XBB=XBB+YJJ(JP(JJ+1),JJ+1)      OPT2      402
        JP(JJ+1)=JP(JJ)      OPT2      403
410      130 CONTINUE      OPT2      404
        XA=XA+YR(B,1)      OPT2      405
        XB=XB+YJ(B,1)      OPT2      406
        XAA=XAA+YRR(B,1)      OPT2      407
        XBB=XBB+YJJ(B,1)      OPT2      408
415      C      OPT2      409
        C      INTRODUCE WHITE GAUSSIAN NOISE      OPT2      410
        C      OPT2      411
        XN=0.0      OPT2      412
        XNN=0.0      OPT2      413
420      IW=8      OPT2      414
        DO 1300 I=1,IW      OPT2      415
        JF=9-I+1      OPT2      416
        ANI(JF)=ANI(JF-1)      OPT2      417
        ANQ(JF)=ANQ(JF-1)      OPT2      418
        XN=XN+ANI(JF)*WSI(JF)+ANQ(JF)*WSQ(JF)      OPT2      419
        XNN=XNN+ANI(JF)*WSQ(JF)+ANQ(JF)*WSI(JF)      OPT2      420
        1300 CONTINUE      OPT2      421
        ANI(1)=G05AEP(0.0,SIGMA)      OPT2      422
        ANQ(1)=G05AEP(0.0,SIGMA)      OPT2      423
430      XN=XN+ANI(1)*WSI(1)+ANQ(1)*WSQ(1)      OPT2      424
        XNN=XNN+ANI(1)*WSQ(1)+ANQ(1)*WSI(1)      OPT2      425
        XA=XA+XN      OPT2      426
        XB=XB+XNN      OPT2      427
        XN=0.0      OPT2      428
435      XNN=0.0      OPT2      429
        DO 1301 I=1,IW      OPT2      430
        JF=9-I+1      OPT2      431
        ANI(JF)=ANI(JF-1)      OPT2      432
        ANQ(JF)=ANQ(JF-1)      OPT2      433
        XN=XN+ANI(JF)*WSI(JF)+ANQ(JF)*WSQ(JF)      OPT2      434
        XNN=XNN+ANI(JF)*WSQ(JF)+ANQ(JF)*WSI(JF)      OPT2      435
        1301 CONTINUE      OPT2      436
        ANI(1)=G05AEP(0.0,SIGMA)      OPT2      437
        ANQ(1)=G05AEP(0.0,SIGMA)      OPT2      438
445      XN=XN+ANI(1)*WSI(1)+ANQ(1)*WSQ(1)      OPT2      439
        XNN=XNN+ANI(1)*WSQ(1)+ANQ(1)*WSI(1)      OPT2      440
        XAA=XAA+XN      OPT2      441
        XBB=XBB+XNN      OPT2      442
        DO 131 JF=1,JDELAY      OPT2      443
        JF=JDEL+JF-1      OPT2      444
        BSTAA(JF)=BSTAA(JF-1)      OPT2      445
        BSTBB(JF)=BSTBB(JF-1)      OPT2      446
        BSTA(JF)=BSTA(JF-1)      OPT2      447
        131 BSTB(JF)=BSTB(JF-1)      OPT2      448
455      BSTA(1)=XA      OPT2      449
        BSTB(1)=XB      OPT2      450

```

Copyright Material - Review Only - Not for Redistribution

AGE	PROGRAM DS1	76/76 OPT=1 TRACE	FTN 4,8+508	21/04/00 12.41.47
6				
7		BSTAA(1)=XAA		OPT2 451
8		BSTBB(1)=XBB		OPT2 452
9		YA=BSTAA(JDEL)		OPT2 453
10	460	YB=BSTBB(JDEL)		OPT2 454
11		YAA=BSTAA(JDEL)		OPT2 455
12		YBB=BSTBB(JDEL)		OPT2 456
13		IF(IIRUN.GT.1)GO TO 401		OPT2 457
14		IF(ICSYM=TDREAY)200,200,132		OPT2 458
15	465	132 IF(ISVM=JDEL)912,912,401		OPT2 459
16				OPT2 460
17		C CALCULATE NEW COST FUNCTION		OPT2 461
18		012 ENR=IC-1		OPT2 462
19		DO 300 KA=1,IR		OPT2 463
20	470	INC=0		OPT2 464
21		HUM=SV		OPT2 465
22		DO 305 KB=2,LD		OPT2 466
23		LA=KB-1		OPT2 467
24	475	CA=0.0		OPT2 468
25		CB=0.0		OPT2 469
26		CAA=0.0		OPT2 470
27		CBB=0.0		OPT2 471
28		DO 301 LR=KB,SV		OPT2 472
29		ICON=IC1-LB+2+INC		OPT2 473
30	480	IAD=ISTORA(KA,ICON)		OPT2 474
31		REAL=JDAT(IAD,1)*DVR(NUM,LB)+JDAT(IAD,2)*DVJ(NUM,LB)		OPT2 475
32		QUAD=JDAT(IAD,2)*DVR(NUM,LB)+JDAT(IAD,1)*DVJ(NUM,LB)		OPT2 476
33		CA=CA+REAL		OPT2 477
34		CB=CB+QUAD		OPT2 478
35	485	REALL=JDAT(IAD,1)*DVR(NUM,LB)+JDAT(IAD,2)*DVJ(NUM,LB)		OPT2 479
36		QUADD=JDAT(IAD,2)*DVR(NUM,LB)+JDAT(IAD,1)*DVJ(NUM,LB)		OPT2 480
37		CAA=CAA+REALL		OPT2 481
38		CBB=CBB+QUADD		OPT2 482
39		301 CONTINUE		OPT2 483
40	490	INC=INC+1		OPT2 484
41		TS1(KA,LA)=CA		OPT2 485
42		TS2(KA,LA)=CB		OPT2 486
43		TS11(KA,LA)=CAA		OPT2 487
44		TS22(KA,LA)=CBB		OPT2 488
45	495	NUM=NUM+1		OPT2 489
46		305 CONTINUE		OPT2 490
47		300 CONTINUE		OPT2 491
48		401 KB=0		OPT2 492
49		LE=LD+1		OPT2 493
50	500	DO 302 KC=1,IR		OPT2 494
51		CINTA=0.0		OPT2 495
52		CINTB=0.0		OPT2 496
53		CINTAA=0.0		OPT2 497
54		CINTBB=0.0		OPT2 498
55	505	DO 303 J=LE,SV		OPT2 499
56		ICON=IC1-J+LE		OPT2 500
57		IAD=ISTOR(KC,ICON)		OPT2 501
58		CINTA=CINTA+YR(IAD,J)		OPT2 502
59		CINTB=CINTB+YJ(IAD,J)		OPT2 503
60	510	CINTAA=CINTAA+YRB(IAD,J)		OPT2 504
61		CINTBB=CINTBB+YJJ(IAD,J)		OPT2 505
62		303 CONTINUE		OPT2 506
63		NUM=SV		OPT2 507

Copyright Material - Review Only - Not for Redistribution

515	DO 304 I=1,ISTEP	OPT2	508
	KD#KD=1	OPT2	509
	IAD=NEST(KD)	OPT2	510
	ZX(KD)=CINTA	OPT2	511
	ZY(KD)=CINTB	OPT2	512
	ZXX(KD)=CINTAA	OPT2	513
520	ZYY(KD)=CINTBB	OPT2	514
	REAL=JDAT(IAD,1)*DVR(NUM,1)+JDAT(IAD,2)*DVJ(NUM,1)	OPT2	515
	QUAD=JDAT(IAD,2)*DVR(NUM,1)+JDAT(IAD,1)*DVJ(NUM,1)	OPT2	516
	REAL=JDAT(IAD,1)*DVR(NUM,1)+JDAT(IAD,2)*DVJ(NUM,1)	OPT2	517
	QUAD=JDAT(IAD,2)*DVR(NUM,1)+JDAT(IAD,1)*DVJ(NUM,1)	OPT2	518
525	TSX(KD)=TS1(KC,1)*REAL	OPT2	519
	TSY(KD)=TS2(KC,1)*QUAD	OPT2	520
	CX=BSTA(JDEL)=TSX(KD)	OPT2	521
	CY=BSTB(JDEL)=TSY(KD)	OPT2	522
	STX(KD)=CX*CX	OPT2	523
530	STY(KD)=CY*CY	OPT2	524
	TSXX(KD)=TS11(KC,1)*REAL	OPT2	525
	TSYY(KD)=TS22(KC,1)*QUAD	OPT2	526
	CXX=BSTAA(JDEL)=TSXX(KD)	OPT2	527
	CYY=BSTBB(JDEL)=TSYY(KD)	OPT2	528
535	STXX(KD)=CXX*CXX	OPT2	529
	STYY(KD)=CYY*CYY	OPT2	530
	UZ(KD)=U(KC)+STX(KD)+STY(KD)+STXX(KD)+STYY(KD)	OPT2	531
	D1(KD)=UZ(KD)	OPT2	532
	UA(KD)=U(KC)	OPT2	533
540	304 CONTINUE	OPT2	534
	302 CONTINUE	OPT2	535
	IF(IEXP,EG,0)GO TO 99	OPT2	536
	JEXP=JEXP+1	OPT2	537
	KD=0	OPT2	538
545	KJ=JDEL+JEXP+1	C106=1	17
	DO 402 KH=1,IR	C106=1	18
	DO 403 KA=1,ISTEP	C106=1	19
	KD#KD=1	C106=1	20
	JAD=NEST(KD)	C106=1	21
550	REAL=JDAT(JAD,1)*DVR(KJ,JEXP)+JDAT(JAD,2)*DVJ(KJ,JEXP)	C106=1	22
	QUAD=JDAT(JAD,2)*DVR(KJ,JEXP)+JDAT(JAD,1)*DVJ(KJ,JEXP)	C106=1	23
	IF(JEXP,HE,SU)GO TO 308	C106=1	24
	UX(KD)=ZX(KD)+REAL	C106=1	25
	UY(KD)=ZY(KD)+QUAD	C106=1	26
555	GO TO 410	C106=1	27
	308 UX(KD)=TS1(KH,JEXP)+REAL	C106=1	28
	UY(KD)=TS2(KH,JEXP)+QUAD	C106=1	29
	410 CONTINUE	C106=1	30
	QX=BSTA(KJ)=UX(KD)	C106=1	31
560	QY=BSTB(KJ)=UY(KD)	C106=1	32
	STC(KD)=QX*QX	C106=1	33
	STD(KD)=QY*QY	C106=1	34
	D1(KD)=0.0	C106=1	35
	DO 404 I=1,IEXP	C106=1	36
565	KKJ=JDEL+I+1	C106=1	37
	PX=BSTA(KKJ)=TS1(KH,I)	C106=1	38
	PY=BSTB(KKJ)=TS2(KH,I)	C106=1	39
	STC(KD)=STC(KD)+PX*PX	C106=1	40
	STD(KD)=STD(KD)+PY*PY	C106=1	41
570	404 CONTINUE	C106=1	42

Copyright IBM Corp. 1970. All rights reserved.

Line	Code	Statement	Address
7		D1(KD)=STC(KD)+STD(KD)	OPT2 561
8		STC(KD)=0.0	OPT2 562
9		STD(KD)=0.0	OPT2 563
10		403 CONTINUE	OPT2 564
11	575	402 CONTINUE	OPT2 565
12		99 IF(1EXP.EQ.0)GO TO 999	OPT2 566
13		KD=0	OPT2 567
14		JEXPP=1EXPP+1	OPT2 568
15	580	KJ=JDEL-1EXPP+1	OPT2 569
16		DO 4022 KH=1,IR	C106+1 43
17		DO 4053 KA=1,STEP	C106+1 44
18		KD=KD+1	C106+1 45
19		JAD=NEST(KD)	C106+1 46
20	585	REALL=JDAT(JAD,1)+OVRR(KJ,JEXPP)+JDAT(JAD,2)+DVJJ(KJ,JEXPP)	C106+1 47
21		QUADD=JDAT(JAD,2)+OVRR(KJ,JEXPP)+JDAT(JAD,1)+DVJJ(KJ,JEXPP)	C106+1 48
22		IF(JEXPP.NE.SV)GO TO 6078	C106+1 49
23		HXX(KD)=ZXX(KD)+REALL	C106+1 50
24		UYX(KD)=ZYX(KD)+QUADD	C106+1 51
25		GO TO 6079	C106+1 52
26	590	6078 UXX(KD)=YS11(KH,JEXPP)+REALL	C106+1 53
27		UYX(KD)=YS22(KH,JEXPP)+QUADD	C106+1 54
28		6079 CONTINUE	C106+1 55
29		QXX=BSTAA(KJ)+UXX(KD)	C106+1 56
30	595	QYY=BSTBR(KJ)+UYX(KD)	C106+1 57
31		STCC(KD)=QXX+QXX	C106+1 58
32		STDD(KD)=QYY+QYY	C106+1 59
33		D11(KD)=0.0	C106+1 60
34		DO 4044 I=1,1EXPP	C106+1 61
35	600	KX=JDEL-I+1	C106+1 62
36		PXX=BSTAA(KX)+YS11(KH,1)	C106+1 63
37		PYY=BSTBR(KX)+YS22(KH,1)	C106+1 64
38		STCC(KD)=STCC(KD)+PXX+PXX	C106+1 65
39		STDD(KD)=STDD(KD)+PYY+PYY	C106+1 66
40	605	4044 CONTINUE	C106+1 67
41		D11(KD)=STCC(KD)+STDD(KD)	C106+1 68
42		STCC(KD)=0.0	OPT2 592
43		STDD(KD)=0.0	OPT2 593
44		4053 CONTINUE	OPT2 594
45	610	4022 CONTINUE	OPT2 595
46		999 IF(1EXP.EQ.0.AND.1EXPP.EQ.0)GO TO 888	OPT2 596
47		IF(1EXP.NE.0)GO TO 887	OPT2 597
48		DO 3023 I=1,IR4	OPT2 598
49		3023 D1(I)=0.0	OPT2 599
50	615	887 DO 3022 I=1,IR4	OPT2 600
51		3022 D1(I)=D1(I)+D1(I)+UA(I)	OPT2 601
52		888 K=1	OPT2 602
53		C COMPARISON PROCESS	OPT2 603
54		C WORK OUT VECTORS WITH LEAST NEW COST FUNCTIONS	OPT2 604
55	620	KK=0	OPT2 605
56		DO 100 J=1CV,IC	OPT2 606
57		TA=10000000.0	OPT2 607
58		TB=10000000.0	OPT2 608
59		TC=10000000.0	OPT2 609
60	625	TD=10000000.0	OPT2 610
61		DO 251 I=1,IR4	OPT2 611
62		IF(J.EQ.IC)1STORAT(SEL(I),J)=NEST(I)	OPT2 612
63			OPT2 613

		NSTOR=ISTORA(NSEL(I1),J)	OPT2	614
		IF(IPOINT(I1)=1)251,260,251	OPT2	615
630	260	IF(NSTOR=1)252,261,252	OPT2	616
	261	IF(D1(I1)=1)253,251,251	OPT2	617
	253	IA=D1(I1)	OPT2	618
		NFA=I1	OPT2	619
		IFA=NSEL(I1)	OPT2	620
635		GO TO 251	OPT2	621
	252	IF(NSTOR=2)254,256,254	OPT2	622
	256	IF(D1(I1)=1)257,251,251	OPT2	623
	257	IB=D1(I1)	OPT2	624
		NFB=I1	OPT2	625
640		IFB=NSEL(I1)	OPT2	626
		GO TO 251	OPT2	627
	254	IF(NSTOR=3)258,259,258	OPT2	628
	259	IF(D1(I1)=1)310,251,251	OPT2	629
	310	IC=D1(I1)	OPT2	630
645		NFC=I1	OPT2	631
		IFC=NSEL(I1)	OPT2	632
		GO TO 251	OPT2	633
	258	IF(D1(I1)=1)311,251,251	OPT2	634
	311	ID=D1(I1)	OPT2	635
650		NFD=I1	OPT2	636
		ID=NSEL(I1)	OPT2	637
	251	CONTINUE	OPT2	638
		IPOINT(NFA)=+1	OPT2	639
		IPOINT(NFB)=+1	OPT2	640
655		IPOINT(NFC)=+1	OPT2	641
		IPOINT(NFD)=+1	OPT2	642
		JFB(1)=NFA	OPT2	643
		JFB(2)=NFB	OPT2	644
		JFB(3)=NFC	OPT2	645
660		JFB(4)=NFD	OPT2	646
		IF(J=BT+1)GO TO 154	OPT2	647
		CFIX=1000000.0	OPT2	648
		DO 268 I=1,4	OPT2	649
		IF(D1(JFB(I))-CFIX)269,268,268	OPT2	650
665	269	CFIX=D1(JFB(I))	OPT2	651
		JFC=JFB(I)	OPT2	652
		JFD=NSEL(JFC)	OPT2	653
	268	CONTINUE	OPT2	654
		DEF=ISTORA(IFD,1)	OPT2	655
670	154	DO 160 LB=1,IC1	OPT2	656
		LS=LB+1	OPT2	657
		ISTOR(K,LS)=ISTORA(IFA,LS)	OPT2	658
		ISTOR(K+1,LS)=ISTORA(IFB,LS)	OPT2	659
		ISTOR(K+2,LS)=ISTORA(IFC,LS)	OPT2	660
675		ISTOR(K+3,LS)=ISTORA(IFD,LS)	OPT2	661
	160	CONTINUE	OPT2	662
		ISTOR(K,IC1)=NEST(NFA)	OPT2	663
		ISTOR(K+1,IC1)=NEST(NFB)	OPT2	664
		ISTOR(K+2,IC1)=NEST(NFC)	OPT2	665
680		ISTOR(K+3,IC1)=NEST(NFD)	OPT2	666
		DO 161 LT=1,ICOL	OPT2	667
		TSA(K,LT)=TS1(IFA,LT)	OPT2	668
		TSA(K+1,LT)=TS1(IFB,LT)	OPT2	669
		TSA(K+2,LT)=TS1(IFC,LT)	OPT2	670

Quality Assurance by John D. ...

AGE	PKUURAM US1	10/10	UPI=1 TRACE	FTN 4,8+508	21/07/80	12.41.47	P -
685	TSB(K*3,LT)=TS2(IFD,LT)					OPT2	671
	TSB(K*1,LT)=TS2(IFB,LT)					OPT2	672
	TSB(K*2,LT)=TS2(IFC,LT)					OPT2	673
	TSB(K*3,LT)=TS2(IFD,LT)					OPT2	674
	TSB(K*4,LT)=TS2(IFE,LT)					OPT2	675
690	TSAA(K*1,LT)=TS11(IFB,LT)					OPT2	676
	TSAA(K*2,LT)=TS11(IFC,LT)					OPT2	677
	TSAA(K*3,LT)=TS11(IFD,LT)					OPT2	678
	TSBB(K*1,LT)=TS22(IFB,LT)					OPT2	679
	TSBB(K*2,LT)=TS22(IFC,LT)					OPT2	680
	TSBB(K*3,LT)=TS22(IFD,LT)					OPT2	681
	TSBB(K*4,LT)=TS22(IFE,LT)					OPT2	682
	TSBB(K*5,LT)=TS22(IFG,LT)					OPT2	683
	161 CONTINUE					OPT2	684
	DO 2700 I1=1,ISTEP					OPT2	685
700	KK=K+1					OPT2	686
	YSA(KK,ICOL)=ZX(JFB(I1))					OPT2	687
	YSB(KK,ICOL)=ZY(JFB(I1))					OPT2	688
	YSA(KK,ICOL)=ZX(JFB(I1))					OPT2	689
	YSB(KK,ICOL)=ZY(JFB(I1))					OPT2	690
705	U(KK)=U2(JFB(I1))					OPT2	691
	2700 CONTINUE					OPT2	692
	K=K+6					OPT2	693
	100 CONTINUE					OPT2	694
	DO 282 J1=K,IR					OPT2	695
710	CMIN=10000000.0					OPT2	696
	DO 283 J2=1,IRA					OPT2	697
	IF(IPOINT(J2)=(-1))263,264,265					OPT2	698
	264 IF(D1(J2)-CMIN)265,263,265					OPT2	699
	265 CMIN=D1(J2)					OPT2	700
	JFA=NSL(J2)					OPT2	701
715	JFC=J2					OPT2	702
	263 CONTINUE					OPT2	703
	IPOINT(JFC)=+1					OPT2	704
	IY(J1)=U2(JFC)					OPT2	705
720	DO 266 LB=1,IC1					OPT2	706
	LS=LB+1					OPT2	707
	266 ISTOR(J1,LS)=ISTORA(JFA,LS)					OPT2	708
	ISTOR(J1,IC1)=NEST(JFC)					OPT2	709
	DO 267 LB=1,ICOL					OPT2	710
725	YSA(J1,LS)=YS1(JFA,LS)					OPT2	711
	YSB(J1,LS)=YS2(JFA,LS)					OPT2	712
	YSA(J1,LS)=YS1(JFA,LS)					OPT2	713
	YSB(J1,LS)=YS2(JFA,LS)					OPT2	714
	267 CONTINUE					OPT2	715
730	YSA(J1,ICOL)=ZX(JFC)					OPT2	716
	YSB(J1,ICOL)=ZY(JFC)					OPT2	717
	YSA(J1,ICOL)=ZX(JFC)					OPT2	718
	YSB(J1,ICOL)=ZY(JFC)					OPT2	719
	262 CONTINUE					OPT2	720
735	C					OPT2	721
	IF(JDAT(RDET,1),NE,JDAT(BB,1))NECR=NECR+1					OPT2	722
	IF(JDAT(RDET,2),NE,JDAT(BB,2))NECJ=NECJ+1					OPT2	723
	171 IF(RDET=BB)172,170,172					OPT2	724
	172 NEC=NEC+1					OPT2	725
740	IERR=IERR+1					OPT2	726
	C RESET MATRIX=ISTORA					OPT2	727

AGE 14

PROGRAM DS1

76776 OPT=1 TRACE

FTN 4,8+308

21709780 12.41.49

P

```

170 DO 190 IS=1,IR
DO 210 IT=1,IC1
745 210 ISTOREA(1S,IT)=ISTOR(1S,IT)
DO 192 IV=1,ICOL
TS1(1S,IV)=TSA(1S,IV)
TS2(1S,IV)=TSB(1S,IV)
TS11(1S,IV)=TSAAT(1S,IV)
750 TS22(1S,IV)=TSBBT(1S,IV)
192 CONTINUE
190 CONTINUE
DO 405 KJ=1,IR4
NV=NV1
755 ICX=2
IAD=ISTORA(KJ,IC1)
DO 406 KL=1,ICOL
KLL=KJ+1
REAL=JDAT(IAD,1)*DVR(NV,ICX)+JDAT(IAD,2)*DVJ(NV,ICX)
760 QUAD=JDAT(IAD,2)*DVR(NV,ICX)+JDAT(IAD,1)*DVJ(NV,ICX)
REAL1=JDAT(IAD,1)*DVR(NV,ICX)+JDAT(IAD,2)*DVJ(NV,ICX)
QUADD=JDAT(IAD,2)*DVR(NV,ICX)+JDAT(IAD,1)*DVJ(NV,ICX)
TS1(KJ,KL)=TS1(KJ,KLL)+REAL
TS2(KJ,KL)=TS2(KJ,KLL)+QUAD
765 TS11(KJ,KL)=TS11(KJ,KLL)+REAL1
TS22(KJ,KL)=TS22(KJ,KLL)+QUADD
ICX=ICX+1
NV=NV+1
406 CONTINUE
770 405 CONTINUE
DO 211 NM=1,IR4
D1(NM)=0.0
D11(NM)=0.0
211 IPOINT(NM)=1
775 200 CONTINUE
NERR(IRUN)=IERR
X1(2)=X1(1)
Y1(2)=Y1(1)
X2(2)=X2(1)
780 Y2(2)=Y2(1)
400 CONTINUE
NECT=NECR+NECJ
NECAV=NECT/2.0
510 FORMAT(5F10.7,F10.1)
785 576 FORMAT(140,8)
WRITE(2,4000)
6000 FORMAT(//)
WRITE(2,603)NECA,NECJ,NECT
WRITE(2,601)NECAV,NEC
790 603 FORMAT(1H0,1X,'INPHASE COMP ERRORS= ',15,1X,'QUAD COMP ERRORS= ',1,1
15,1X,'TOTAL NO. OF BIT ERRORS= ',15)
601 FORMAT(1H0,1X,'BIT ERRORS AVERAGE= ',15,1X,'SYMBOL ERRORS= ',15)
WRITE(2,602)SIGMA,SNR,FRAC,FRACC
602 FORMAT(1H0,1X,'SIGMA= ',F10.5,2X,'SNR= ',F10.5,2X,'SIG COMP= ',F8.5)
795 15,3X,'SIG COMP DS= ',F8.5)
WRITE(2,609)NERR(1K),IK=1,1100)
609 FORMAT(1H ,2X,'ERROR DISTRIBUTION: ',1,1X,22(13,2X))
WRITE(2,8888)

```

OPT2 728
OPT2 729
OPT2 730
OPT2 731
OPT2 732
OPT2 733
OPT2 734
OPT2 735
OPT2 736
OPT2 737
OPT2 738
OPT2 739
OPT2 740
OPT2 741
OPT2 742
OPT2 743
OPT2 744
OPT2 745
OPT2 746
OPT2 747
OPT2 748
OPT2 749
OPT2 750
OPT2 751
OPT2 752
OPT2 753
OPT2 754
OPT2 755
OPT2 756
OPT2 757
OPT2 758
OPT2 759
OPT2 760
OPT2 761
OPT2 762
OPT2 763
OPT2 764
OPT2 765
OPT2 766
OPT2 767
OPT2 768
OPT2 769
OPT2 770
OPT2 771
OPT2 772
OPT2 773
OPT2 774
OPT2 775
OPT2 776
OPT2 777
OPT2 778
OPT2 779
OPT2 780
OPT2 781
OPT2 782
OPT2 783
OPT2 784

Copyright 1980 by John D. ...

```

8888 FORMAT(///)
800 DO 3333 MC=1,18 OPT2 785
WRITE(2,2000)(1$TOR(MC,MD),MD=1,101),U(MC) OPT2 786
2000 FORMAT(C1H:15X:16Y12,3X):5X:10-53) OPT2 787
3333 CONTINUE OPT2 788
805 END OPT2 789
OPT2 790

```

Copyright Clearance Center, Inc. All rights reserved.

3	0	0	2	0	1	0	0	0	0	0	0	0	3	0	0	0	2	0	0	0	0	
4	1	0	0	1	3	0	3	0	1	4	1	1	1	4	6	2	4	6	6	5	4	5
5	0	5	4	4	11	7	7	10	6	10	12	11	11	7	13	10	12	16	12	11	10	10
6	10	11	6	8	13	8	13	7	8	12	10	11	10	8	7	4	5	4	7	6	5	7
7	7	9	8	6	8	6	6	2	3	5	5	3	7	4	9	4	1	5	4	5	4	1
8	4	4	1	4	4	8	6	4	4	5	2	8	1	5	3	7	3	7	5	4	7	3
9	7	6	6	3	1	5	4	5	6	6	7	8	11	7	6	4	3	2	6	6	3	7
10	7	3	2	11	4	13	8	13	6	6	7	12	9	5	6	9	11	10	8	6	10	
11	8	8	14	10	12	3	6	8	7	7	11	11	14	8	8	14	8	5	11	8	6	13
12	8	7	7	10	0	12	7	6	2	9	6	9	13	4	12	11	17	9	12	10	10	6
13	6	15	9	4	7	6	8	11	12	10	12	11	8	10	8	8	7	13	13	9	8	12
14	13	17	12	15	12	13	14	12	14	13	13	16	11	13	16	15	12	12	13	15	12	11
15	10	10	7	6	8	3	7	4	5	12	3	5	3	3	6	3	5	1	2	3	2	2
16	0	2	2	3	1	2	4	1	0	2	1	0	2	0	2	3	4	5	2	9	6	3
17	5	9	6	7	10	7	4	5	10	12	10	12	6	6	9	10	11	4	4	5	4	3
18	4	5	3	2	2	1	8	6	4	4	4	2	2	4	3	1	4	3	4	1	3	4
19	2	0	2	0	2	4	1	2	1	4	1	1	0	3	3	2	1	2	3	2	3	2
20	0	3	3	2	1	1	6	1	1	0	0	0	0	2	0	2	2	0	5	0	9	2
21	1	4	2	5	1	6	5	3	1	3	2	5	1	3	7	5	2	10	5	3	4	7
22	4	7	3	5	4	7	4	4	8	13	9	4	12	9	7	9	13	7	10	11	6	7
23	9	14	10	10	12	10	9	12	8	11	13	5	11	9	0	16	12	9	9	11	11	8
24	7	7	14	8	7	12	5	6	4	4	5	4	2	3	1	1	6	4	3	2	2	0
25	2	5	2	2	0	2	4	2	2	1	1	1	1	1	1	1	2	2	1	1	4	4
26	4	1	4	4	5	3	5	4	1	4	6	6	5	6	7	11	8	4	11	10	11	5
27	11	8	9	10	5	7	10	5	4	6	7	3	5	2	5	4	6	5	2	1	1	0
28	1	3	2	4	4	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	0	0	4	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	3
31	0	1	0	1	1	0	2	0	0	3	2	1	5	9	1	2	2	2	1	2	2	7
32	6	4	2	3	1	4	4	6	2	2	3	2	5	2	2	2	2	4	2	1	4	6
33	6	8	9	2	7	3	7	3	10	9	7	6	12	3	9	7	7	10	6	11	8	9
34	4	10	6	9	9	7	10	5	6	8	10	8	8	11	4	3	5	8	5	6	3	1
35	3	1	0	5	1	3	1	0	1	1	2	0	1	0	1	0	0	0	1	0	0	0
36	0	2	1	0	0	1	1	1	0	1	1	0	0	0	0	0	0	1	0	1	0	0
37	1	1	1	2	3	1	1	0	1	2	3	2	3	2	0	5	0	1	2	4	3	2
38	5	3	3	3	8	3	3	2	3	3	4	1	4	1	2	4	2	2	3	4	4	1
39	4	3	4	2	1	3	3	1	6	3	1	8	2	2	0	1	2	2	2	3	3	4
40	4	4	2	0	0	0	3	2	1	4	2	4	3	2	4	5	0	2	2	2	2	1
41	0	0	2	0	1	2	1	4	3	5	3	1	3	5	3	2	5	2	1	3	8	5
42	5	6	6	8	6	3	14	6	8	9	14	7	8	9	9	7	11	11	9	13	8	13
43	7	13	11	10	10	11	16	13	11	9	12	8	8	11	13	9	7	6	9	8	3	7
44	4	4	7	7	5	4	7	4	5	0	6	2	3	0	2	3	5	2	3	3	3	3
45	3	1	4	2	4	1	2	3	2	4	5	1	2	1	1	4	1	1	3	1	4	2
46	1	2	1	1	1	0	1	1	1	3	1	2	1	1	2	2	1	3	1	0	1	2
47	3	3	7	2	3	3	2	4	3	0	3	1	3	4	0	5	5	4	7	4	4	11
48	3	8	6	9	7	7	4	6	8	8	8	6	7	5	7	6	7	8	4	6	4	8
49																						
50																						
51																						
52																						
53																						
54																						
55																						
56																						
57																						
58																						
59																						
60																						
61																						
62																						
63																						
64																						

INPHASE COMP ERRORS= 1013 QUAD COMP ERRORS= 977 TOTAL NO OF BIT ERRORS= 1990

2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
6 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

11
12
13
14 4 4 1 1 2 1 3 4 3 1 1 4 3 2 3 1 1056-61875
15 4 4 1 1 2 1 3 4 3 1 1 4 3 2 3 2 1056-63962
16 4 4 1 1 2 1 3 4 3 1 1 4 3 2 3 3 1056-67971
17 4 4 1 1 2 1 3 4 3 1 1 4 3 2 3 4 1056-63884
18 4 4 1 1 2 1 3 4 3 1 1 4 3 2 3 2 1056-66386
19 4 4 1 1 2 1 3 4 3 1 1 4 3 2 3 2 1057-60258
20 4 4 1 1 2 1 3 4 3 1 1 4 3 2 3 3 1056-68393
21 4 4 1 1 2 1 3 4 3 1 1 4 3 2 3 1 1056-62297
22

23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64

Country of Origin: USA, Denmark

```

EEEEEE UU  UU NN  NN X  X X  X
EEEEEE UU  UU NNN NN XX XX XX XX
EE  UU  UU NNNN NN XX XX XX XX
EE  UU  UU NN NN NN XXXX XXXX
EEEE UU  UU NN NNN  XX  XX
EE  UU  UU NN NN XXXX XXXX
EE  UU  UU NN NN XX XX XX XX
EEEEEE UUUUUUU NN  NN XX  XX XX XX
EEEEEE UUUUUU NN  NN X  X X  X

```

```

ZZZZZZ 55555 111 9999 111 DDDDDP BBBBBD
ZZZZZZ 3333333 111 99999 111 PPPPPPP BBBBBD
ZZ 3 33 1 1 99 99 1 1 PP PP BB BB
ZZ 33 11 99 99 11 PP PP BB BB
ZZZZZZ 333 11 9999999 11 DDDDDP BBBBBD
ZZ 333 11 9999999 11 PPPPPP BBBBBD
ZZ 3 33 11 99 99 11 PP PP BB BB
ZZ 3 33 11 9 99 11 PP BB BB
ZZZZZZ 3333333 1111111 99999 1111111 PP BBBBBD
ZZZZZZ 333333 1111111 9999 1111111 PP BBBBBD

```

1904S/7600 BULLETIN BOX NO. 174

SCOPE 2.1.5 LEVEL 512C

THERE WILL BE A SECOND FIELD TRIAL OF SCOPE 2.1.5
 LEVEL 512C ON TUESDAY, 23RD SEPTEMBER, FROM 10:00 TO
 17:30 HOURS.
 THERE ARE NO CHANGES IN THIS VERSION THAT DIRECTLY AFFECT
 USERS.

PLEASE INFORM UMRCC ADVISORY OF ANY PROBLEMS.

THE MICROFILM SERVICE

THE MICROFILM SERVICE IS NOW BACK IN ACTION. WE APOLOGISE
 FOR ANY INCONVENIENCE THAT THE BREAK IN SERVICE HAS
 CAUSED.

18/9/80

