Data transmission at 9600 bit/sec over an HF radio link

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Additional Information:

- A Doctoral Thesis. Submitted in partial fulfillment of the requirements for the award of Doctor of Philosophy of Loughborough University.

Metadata Record: https://dspace.lboro.ac.uk/2134/7432

Publisher: © S.N. Abdullah

Please cite the published version.
This item is held in Loughborough University’s Institutional Repository (https://dspace.lboro.ac.uk/) and was harvested from the British Library’s EThOS service (http://www.ethos.bl.uk/). It is made available under the following Creative Commons Licence conditions.

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/
DATA TRANSMISSION AT 9600 BIT/SEC
OVER AN HF RADIO LINK

by

S.N. ABDULLAH, BSc, MSc

A Doctoral Thesis
submitted in partial fulfilment of the
requirements for the award of the
degree of Doctor of Philosophy
of the Loughborough University of Technology

September 1986

Supervisor: Professor A.P. Clark

Department of Electronic and Electrical Engineering

© by S.N. Abdullah, 1986
# LIST OF CONTENTS

<table>
<thead>
<tr>
<th></th>
<th>Page No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>v</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>vii</td>
</tr>
<tr>
<td>Glossary of Symbols and Terms</td>
<td>viii</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>1. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Outline of the Investigation</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>2. THE HF CHANNEL</td>
<td>6</td>
</tr>
<tr>
<td>2.1 The Ionosphere</td>
<td>6</td>
</tr>
<tr>
<td>2.2 Ionospheric Radio Propagation</td>
<td>7</td>
</tr>
<tr>
<td>2.3 Types of Distortion on HF Channels</td>
<td>8</td>
</tr>
<tr>
<td>2.3.1 Multipath Propagation and Time Dispersion</td>
<td>8</td>
</tr>
<tr>
<td>2.3.2 Flat Fading</td>
<td>10</td>
</tr>
<tr>
<td>2.3.3 Doppler Shifts</td>
<td>12</td>
</tr>
<tr>
<td>2.4 Model and Simulation of an HF Channel</td>
<td>12</td>
</tr>
<tr>
<td>2.4.1 Model of the HF Channel</td>
<td>12</td>
</tr>
<tr>
<td>2.4.2 Generation of the Random Process $q_1(t)$ in the Computer Simulation</td>
<td>17</td>
</tr>
<tr>
<td>2.4.3 Model of the HF Radio Link with $n$ Skywaves</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>3. MODEL OF THE DATA TRANSMISSION SYSTEM</td>
<td>24</td>
</tr>
<tr>
<td>3.1 Transmission of Data Using Quadrature Amplitude Modulation (QAM)</td>
<td>24</td>
</tr>
<tr>
<td>3.2 Data Transmission Over a Model of an HF Channel Using QAM</td>
<td>25</td>
</tr>
<tr>
<td>3.3 Model of the HF Data Transmission Used in the Tests</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>4. DETECTION PROCESSES FOR DISTORTED SIGNALS</td>
<td>51</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>51</td>
</tr>
<tr>
<td>4.2 Detection Processes for Time Invariant Channels</td>
<td>56</td>
</tr>
<tr>
<td>4.2.1 Model of the System</td>
<td>56</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>4.2.2</td>
<td>Conventional Non-linear Equalizer</td>
</tr>
<tr>
<td>4.2.3</td>
<td>The Viterbi-Algorithm Detector with a Whitened Matched Filter</td>
</tr>
<tr>
<td>4.2.4</td>
<td>Near-Maximum-Likelihood Detection</td>
</tr>
<tr>
<td>4.3</td>
<td>Detection Processes for Time Varying Channels (HF Links)</td>
</tr>
<tr>
<td>4.4</td>
<td>Results of Computer Simulation Tests</td>
</tr>
<tr>
<td>4.4.1</td>
<td>'Back-to-Back' Tests</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Tests for the Detection Systems Operating Over an HF Radio Channel</td>
</tr>
<tr>
<td>5.</td>
<td>HF CHANNEL ESTIMATION</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>5.2</td>
<td>Model of the Data Transmission System</td>
</tr>
<tr>
<td>5.3</td>
<td>Methods of HF Channel Estimation</td>
</tr>
<tr>
<td>5.3.1</td>
<td>Feedforward Transversal-Filter Estimator</td>
</tr>
<tr>
<td>5.3.2</td>
<td>Feedforward Estimator with Prediction</td>
</tr>
<tr>
<td>5.3.3</td>
<td>The Kalman Estimator</td>
</tr>
<tr>
<td>5.3.4</td>
<td>Improved Channel Estimator</td>
</tr>
<tr>
<td>5.4</td>
<td>Improved Channel Estimator</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Improved Channel Estimator with a Two-Skywave HF Channel</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Improved Channel Estimator with One or Three Skywaves HF Channels</td>
</tr>
<tr>
<td>5.5</td>
<td>Fast Channel Estimation</td>
</tr>
<tr>
<td>5.5.1</td>
<td>Fast Channel Estimation Method</td>
</tr>
<tr>
<td>5.5.2</td>
<td>Computer Simulation Tests</td>
</tr>
<tr>
<td>5.6</td>
<td>Starting-Up Procedure for the Improved Estimator</td>
</tr>
<tr>
<td>5.6.1</td>
<td>Starting-Up with Number of Skywaves Known at the Receiver</td>
</tr>
<tr>
<td>5.6.2</td>
<td>Starting-Up with an Unknown Number of Skywaves</td>
</tr>
<tr>
<td>5.7</td>
<td>Adaptive Improved Estimator</td>
</tr>
<tr>
<td>5.7.1</td>
<td>Introduction</td>
</tr>
<tr>
<td>5.7.2</td>
<td>Adaptive Estimation</td>
</tr>
<tr>
<td>5.7.3</td>
<td>Results and Analysis of Simulation Tests</td>
</tr>
</tbody>
</table>
6. COMBINED ESTIMATION, ADAPTIVE LINEAR FILTERING AND DETECTION FOR A SERIAL 9600 BIT/SEC MODEM .... 166
   6.1 Introduction .... 166
   6.2 Model of the Data Transmission System .... 166
   6.3 Results and Analysis of Simulation Tests .... 170
      6.3.1 Tolerance of the Modem to Additive White Gaussian Noise .... 171
      6.3.2 Performance of the Modem with Retraining 175
      6.3.3 Tolerance of the Modem to Impulsive Noise 183
      6.3.4 Tolerance of the Modem to Sudden Level or Phase Changes in the Received Signal Carrier 184
      6.3.5 Tolerance of the Modem to Interference from Other Radio Signals .... 185
   6.4 Signal Quality Measures .... 187
   6.5 Non-Minimum Phase Filters .... 188

7. MODEM SYNCHRONIZATION .... 191
   7.1 Introduction .... 191
   7.2 Time Synchronization of an HF Modem .... 191
      7.2.1 Basic Algorithm .... 191
      7.2.2 Results and Analysis of Computer Simulation Tests .... 192
      7.2.3 Arithmetic Accuracy in Element Synchronization Algorithm .... 201
   7.3 Carrier Phase Tracking .... 217
      7.3.1 Carrier Phase Tracking at the Start of Transmission .... 217
      7.3.2 Carrier Phase Tracking During Data Transmission .... 218

8. COMMENTS ON THE RESEARCH PROJECT .... 228
   8.1 Conclusions .... 228
   8.2 Possible Further Investigations .... 229
### APPENDICES:

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Title</th>
<th>Page No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appendix A</td>
<td>Derivation of Raleigh Fading Filters</td>
<td>231</td>
</tr>
<tr>
<td>Appendix B</td>
<td>Differential Encoding and Decoding</td>
<td>238</td>
</tr>
<tr>
<td>Appendix C</td>
<td>Signal Distortion Introduced by a Channel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Sampled Impulse Response</td>
<td>242</td>
</tr>
<tr>
<td>Appendix D</td>
<td>Kalman Estimator Algorithm</td>
<td>246</td>
</tr>
<tr>
<td>Appendix E</td>
<td>Expanded-Memory Polynomial Filter</td>
<td>248</td>
</tr>
<tr>
<td>Appendix F</td>
<td>Adaptive Adjustment of the Linear Filter</td>
<td>251</td>
</tr>
<tr>
<td>Appendix G</td>
<td>Computer Programs</td>
<td>262</td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>327</td>
</tr>
</tbody>
</table>
ABSTRACT

The thesis is concerned with serial data transmission at 9600 bit/sec over a voiceband channel, where the main impairments are additive noise and intersymbol interference, and the latter varies slowly with time.

The thesis includes a brief description of the ionospheric propagation medium and presents an equivalent baseband model of the HF channel, suitable for computer simulation of quadrature amplitude modulation (QAM) systems. A study of 16-point QAM signals transmitted over voiceband HF channels is then carried out using the given channel model.

Several cost effective near-maximum-likelihood detection processes have been developed for HF modems. Each detector is here preceded by an adaptive linear filter that is adjusted to make the sampled impulse response of the channel and filter minimum phase. These detectors require an accurate knowledge of the sampled impulse response of the channel, if their full potential is to be achieved.

The results of computer-simulation tests on the near-maximum-likelihood detectors are given, where these tests assume that other receiver operations such as channel estimation and adaptive linear filtering, together with element timing synchronisation and Doppler shift correction, are carried out perfectly.

A recently developed HF channel estimator employing a simple feedforward transversal-filter and requiring knowledge of the number of skywaves is next investigated and a starting up procedure is developed for such an estimator. The technique is then made fully adaptive in the sense that it continues to operate correctly when the number of skywaves changes.

Results of computer simulation tests are then presented showing the performance of the above detectors when operating with a channel estimator and adaptive linear filtering.
Finally modem synchronisation is studied and various techniques of element timing and carrier frequency synchronisation are proposed.
The author would like to express his gratitude to his supervisor, Professor A P Clark, for his considerable help and guidance.

The financial support of the Government of Iraq is gratefully acknowledged. The author would also like to thank his wife for her patience during the project and Janet Smith for her excellent typing.
GLOSSARY OF SYMBOLS AND TERMS

a(t)  Impulse response of a filter
A(f)  Frequency response of a filter
|A(f)|  Absolute value of A(f)
a(t)*b(t)  Convolution between a(t) and b(t)
e_i  Error in the estimated value of r_i
E[.]  Expectation operator
g+1  Number of samples in the sampled impulse response of linear baseband channel
Im[.]  Imaginary part of a complex number
j  When used as a subscript is an integer, otherwise it is \(\sqrt{-1}\)
n(t)  White Gaussian noise with zero mean and two-sided power spectral density \(\frac{1}{2}N_0\)
\(\frac{1}{2}N_0\)  Power spectral density of n(t)
\(\{q_h(t)\}\)  Statistically independent random processes
\(\{q_{h,i}\}\)  Sequence obtained by sampling \(q_h(t)\)
Re[.]  Real part of a complex number
Rm[.]  Real or imaginary part of a complex number
r(t)  Received signal
\(\{r_i\}\)  Sequence of received signal samples
s_i  Data symbol
Superscript *  Complex conjugate
Superscript T  Matrix transpose
Superscript (*T)  Complex conjugate transpose
T  Sampling interval
y(t)  Impulse response of linear baseband channel
\(Y_i\)  Sampled impulse-response of linear baseband channel at time \(t = iT\)
\(Y_i\)  Estimate of \(Y_i\) at time \(t = iT\)
\( Y_{i+n,i}^{'} \)  
Prediction of \( Y_{i+n} \) at time \( t = iT \) obtained from the \( \{Y_{h}^{'}\} \) for \( h = i, i-1, \ldots \)

\( \theta \)  
Small positive quantity

\( \xi \)  
Mean-square error in the estimate (prediction of \( Y_{i} \))

\( \xi_{i} \)  
Square of the error in the estimate (prediction of \( Y_{i} \))

\( \psi \)  
Signal-to-noise ratio (or S/N)

\( w(t) \)  
Gaussian random process with zero mean

\( \sigma^2 \)  
Variance of \( w(t) \) or \( \{w_{i}\} \)

\( ||x|| \)  
Absolute value of the real part of \( x \) + absolute value of the imaginary part of \( x \)
CHAPTER 1

INTRODUCTION

1.1 BACKGROUND

The radio frequency band in the region of 3 to 30 MHz is traditionally known as the high-frequency (HF) band [1,2]. At these frequencies, propagation of radio signals is achieved by ionospheric reflection from one or more layers of the ionosphere [3,4].

The problem of reliable transmission of digital information over the HF (ionospheric) channel has always presented the communicator with a significant challenge. Compared to a satellite link, a conventional "narrow band" HF radio link can be described as inexpensive, robust, low capacity, and unreliable. The first two properties stem from the fact that the ionosphere is "free and indestructable" and emphasize the significance of HF in the military environment. However, the disadvantage, stemming largely from the unpredictable nature of an ionospheric channel, have encouraged the use of reliable (though expensive and vulnerable) satellite systems [5,1,6,7]. More recently, however, a new wave of interest is in evidence and efficient signal processing techniques combined with high speed processing capabilities promise to provide the means to high reliability HF data communication [8,7,9-15].

The earliest attempts at HF radio data transmission made use of serial a synchronous techniques such as manually transmitted and received morse code (= 10 bits/s) and radio-teletype, RTTY (= 50 bit/s). The success of these low data rate transmission, and their high information densities of 1-2 bit/s/Hz of available bandwidth, suggests that a simple extension of the channel bandwidth to 3 kHz voiceband channels and similar increase in data rates would allow HF data transmission at several kilobits per second. Unfortunately, such a simple extrapolation gives very poor results because of the nature of the HF transmission medium.
Distortion of the high data-rate signal is produced by dispersion in the HF channel, which is produced by the reception of several discrete paths with different transmission delays, each of the paths fading independently of each other with time. The multipath delay spread can commonly be several milliseconds, resulting in the simultaneous reception of several separate data bits at the receiver. This pulse "smearing" is known as intersymbol interference and is the principal reason for the poor performance of simple high data rate transmission/reception methods over HF radio channels [3,6,7].

The first means of high speed data transmission over voiceband HF channels is by using multi-channel ("parallel") differential phase-shift-keying (DPSK) transmission. Essentially this parallel technique splits the high speed data to be sent, between a large number typically 24, low speed data subchannels equally spaced through the available bandwidth. An information density of 1.5 bit/s/Hz can be easily achieved with such a technique at data rates of 2.4 kbit/s using 4 phase DPSK in the subchannels. Unfortunately, this transmission technique suffers from the disadvantage that in the presence of channel fading, and even in the complete absence of additive noise, there exists a finite error rate in the detected data symbols at the receiver which is proportional to the fading rate [7].

An alternative approach to the above parallel transmission of data is to use serial transmission and to employ some form of adaptive signal processing at the receiver. Comparison of the two transmission techniques at a speed of 2400 bit/s have suggested that the serial modem offers a better overall performance [8,10]. With an increase in the processing speed of digital hardware and also improvements in signal processing techniques, serial modems are now challenging the dominance of parallel modems for high speed applications.

In serial transmission (which is considered in this thesis) the data being extracted from the received signal using a sophisticated detection process which overcomes the intersymbol interference problem
mentioned earlier. In the detection of serially transmitted digital data, the detector may adopt one of several strategies which are broadly classified into two groups. In the first group, the detector employs a device known as an equalizer to remove the intersymbol interference from the received signal [16-33]. The process of removing the intersymbol interference by the equalizer often results in using only part of the transmitted signal energy in the detection of a data symbol, with a consequent reduction in tolerance to additive noise. Some of the recently proposed serial HF modems use equalization techniques in the detection process [9,10,12,14]. The other group of detection techniques, instead of removing the intersymbol interference, take account of it, thus using the entire transmitted energy in the detection process. These techniques are known as the maximum or near-maximum-likelihood detectors [21,29,34-54]. Unlike the equalizer, the maximum-likelihood detector is optimum in the sense that, under the appropriate conditions, it minimises the probability of error in the detection of the whole message, and in many cases, its performance is as good as if there were no intersymbol interference [34]. The major drawback of the maximum-likelihood detector, compared with the equalizer, is its high complexity, which in most cases prevents its use in practice. As a compromise, the near-maximum likelihood detector combines the simplicity of the equalizer with the optimality of the maximum-likelihood detector to give a system which is closer to the former in its simplicity and to the latter in its performance.

Recent developments in the detection of data, using near-maximum-likelihood detectors, have led to the design of a 9600 bit/s serial modem. This modem is capable of operating successfully over a model of a voiceband HF radio link with two Raleigh fading skywaves [52,11]. The technique has however two serious weaknesses. Firstly, a complex detection process is required, and secondly the estimate of the sampled impulse response of the channel (which is needed by the near-maximum-likelihood detector) involves prediction over many sampling intervals, which places a strict upper limit on both the time dispersion and frequency spread of the received signals that are tolerated by the system. Both of these disadvantages could be overcome if the near-
maximum-likelihood detector were preceded by an adaptive linear filter that is adjusted to make the sampled impulse response of the channel and filter minimum phase, and this is the approach considered here.

1.2 OUTLINE OF THE INVESTIGATION

Essentially, the investigation is concerned with the serial transmission of 9.6 kbit/sec data over an HF link and the performance of a novel serial modem measured in terms of its error rate. The research has been carried out using computer simulation of the modem and HF radio link. In this situation, computer simulation is a valid means of evaluating the system because the modem is a digital processor performing computer-like operations on sets of numbers.

Chapter 2 contains a detailed description of the HF radio channels. Firstly, the physical characteristics of the earth's atmosphere are considered and these lead to an understanding of the types of distortion occurring on such channels. Finally a model of the HF channel is presented suitable for computer simulation.

In Chapter 3 a model of a synchronous serial QAM digital data transmission system, which includes the model of the HF link is presented.

In Chapter 4 a brief description of known techniques for digital data detection is given. Then some newly developed near-maximum-likelihood detectors are described. Finally results are presented of tests on these near-maximum-likelihood detectors over the HF channel, using the model derived in Chapter 3 and assuming perfect adaptive linear filtering and estimation.

Chapter 5 is concerned with HF channel estimation, which is a necessary function involved with a maximum or near-maximum-likelihood detector. A description of the improved estimator is given first, followed by a starting up procedure for this estimator and then the results of tests over the HF link. A description is next presented of a method to make
the estimator fully adaptive. This adaptive estimator continues giving accurate estimates of the channel even if the number of skywaves changes during the data transmission. Finally, results of the computer simulation are given for the adaptive estimator.

In Chapter 6 the near-maximum-likelihood detector, adaptive linear filter and estimator are connected together and the combined system performance is tested and analysed.

In Chapter 7 the remaining serial modem functions of carrier phase tracking and symbol timing are considered. Several techniques are proposed here and results of computer simulation tests on the suggested techniques are then presented.
CHAPTER 2

THE HF CHANNEL

2.1 THE IONOSPHERE [55,3]

The ionosphere is a region of ionized particles, which includes free electrons, and it extends from some 50 km to over 700 km above the earth's surface. The ionization is caused by the action of the sun's radiation on the upper atmosphere of the earth. The free electrons act as reflectors for HF radio waves, and their density (and hence their ability to reflect radio signals) varies with height, the greatest density being normally at a height in the neighbourhood of 300 km. The density does not, however, usually increase steadily with height until the greatest density is reached, but generally has two or three regions in which the density reaches a local maximum. These regions are known as ionized layers. During the daytime the ionosphere has three main ionized layers, each of which could, under the appropriate conditions reflect an HF radio wave. The layers are known as the E, F₁, and F₂ layers whose heights are, respectively, around 100, 200, and 250-350 km. The F₂ layer is the most useful part of the ionosphere for HF radio communication. It is the most reliable reflecting media during both day and night. The density of free electrons in the F₂ layer is greater than that in the F₁ layer, which in turn, is greater than that in the E layer. At night the F₁ layer tends to fade out such that the F₁ and F₂ layers effectively coalesce into a single F layer which has a height of around 300 km. The E layer again has a lower density of free electrons than the F layer and occasionally almost disappears. During the daytime there is a fourth ionized layer of relatively small ionization density known as the D-layer, whose height is typically 50-90 km. This layer cannot be used to reflect an HF radio wave and tends to attenuate skywaves that are reflected from any of the E, F₁ and F₂ layers.
2.2 IONOSPHERIC RADIO PROPAGATION [56,2,7]

Refractive bending is the actual process by which HF radio waves are returned to earth. The refractive index $\varepsilon$, of the layer changes continuously with its height. This is due to the dependence of $\varepsilon$ on the electron density of the ionized medium. The refractive index of the medium is given by:

$$\varepsilon = (1 - \frac{81 N_e}{f^2})^{\frac{1}{2}}$$  \hspace{1cm} (2.2.1)

where $f$ is the frequency of the radio waves and $N_e$ is the number of free electrons per cubic metre (m$^3$).

A finely stratified region of the ionosphere, with a constant refractive index of $\varepsilon$ in the $i$th layer, is shown in Figure 2.2.1 to demonstrate more clearly the bending process. For a given angle of incidence of a radio wave meeting a reflecting layer, total internal reflection occurs and skywave is formed when

$$\sin \theta_i = (1 - \frac{81 N_e}{f^2})^{\frac{1}{2}}$$  \hspace{1cm} (2.2.2)

$\theta_i$ is the angle of incidence of the wave measured from the normal.

The critical frequency of a layer is obtained when $\theta_i = 0$ (vertical incidence) and represents the highest reflectable frequency of the layer at this incidence. It is given by

$$f_{cr} = 9\sqrt{N_{em}}$$  \hspace{1cm} (2.2.3)

where $N_{em}$ is the maximum number of free electrons per m$^3$ in the layer.
Higher frequencies can be reflected from this layer at other angles of incidence but for any given angle, there is a maximum frequency at which reflection takes place. This is called maximum usable frequency (MUF) and is given by:

\[
\text{MUF} = f_{cr} \sec \theta_i
\]  \hspace{1cm} (2.2.4)

Furthermore, the effect of the earth's magnetic field has been shown to split an incident wave on entering the ionized medium into two circularly polarized waves the ordinary and extraordinary waves. The effect is known as magneto-ionic splitting. The two rays travel along different paths but they can sometimes recombine on leaving the ionized medium to give an elliptically polarized wave. Under extreme conditions only one of the two rays is reflected. Also magneto-ionic splitting is dependent on the operating frequency, the most noticeable splitting occurring at a frequency just below the MUF of a layer. Merging of the two rays takes place if the operating frequency is further reduced.

2.3 TYPES OF DISTORTION ON HF CHANNELS

2.3.1 Multipath Propagation and Time Dispersion

The transmitted radio wave may be propagated to the receiver along one or more different paths of unequal lengths, that is by multipath propagation [3]. Many of these propagation paths or modes are possible, especially for long distance propagation; however, the number of 'effective' modes are small. For example, the transmitted radio signal may travel from the transmitter to the receiver via two skywaves which are reflected at two different layers, in the ionosphere. Alternatively the transmitted radio signal may travel from the transmitter to the receiver via both one and two hops, that is being reflected either once or twice from the ionosphere (Figure 2.3.1).
Fig. 2.2.1: Refractive bending in an ionized layer

(a)

Fig. 2.3.1: Multipath propagation:
(a) Reflection from different layers of the ionosphere
(b) Different number of hops

(b)
In general when a short pulse of RF energy is transmitted, the received signal will have a profile such as that of Figure 2.3.2 the time between the reception of the first and last pulse is known as the time spread or time dispersion of the received signal [57-60]. The time dispersion of a transmitted signal-element is usually less than 3 ms.

2.3.2 Flat Fading

Flat fading is the variation with time of the received signal level. It occurs when a radio signal (skywave) is reflected from just one layer of the ionosphere and reaches the receiver via one hop. The ionosphere really acts as a large number of different reflectors at different (but not widely different) heights and introducing different attenuations, to give a large number of reflected waves with different levels and carrier phases as shown in Figure 2.3.3. The relative heights of the reflectors and their attenuations may now vary slowly and in a random manner. Variations in the relative heights of the reflectors cause the corresponding variations in path lengths and hence in the relative delays of the reflected waves, leading to slow and random variations in the relative carrier-phases of the reflected waves [3]. Any two of these therefore sometimes tend to add and other times tend to cancel, leading to a random variation in the level of the resulting wave. The variation in the level and phase of the many different reflected waves causes Raleigh fading of the overall resultant reflected wave. The envelope of the wave varies according to a Raleigh distribution and its phase varies according to a uniform distribution. If the transmitted signal is simply a sine wave carrier of fixed frequency and amplitude as in Figure 2.3.4a, then the received signal will have typically the form shown in Figure 2.3.4b [7]. It is evident from Figure 2.3.4 that the received signal here is modulated in amplitude, the modulating waveform having only very low frequency components. Thus the bandwidth of the received signal is very slightly increased. The increase in bandwidth is known as "frequency spread", the value of frequency spread being around 0.1 Hz under mild conditions and around 0.5 Hz under poor conditions. Frequency spreads of up to 1 or 2 Hz are, however, likely to be experienced sufficiently often for it to be important that the modem has
Fig. 2.3.2: Typical response of multipath channel

Fig. 2.3.3: Flat fading (Raleigh fading)
an adequate tolerance to these [3].

2.3.3 Doppler Shifts

A movement of one of the ionospheric layers would produce a distinct Doppler shift on a signal received via that layer. It can be seen clearly from Figure 2.3.5 that there are certain times of the day, notably from 5 am until 8 am and from 5 pm to 7 pm when the reflecting layers are moving rapidly in one direction - some 50 km/hour for the F$_2$ layer during the evening [55]. The magnitude of the Doppler shift is, of course, frequency dependent but at a typical operating frequency of 15 MHz, the shift would be around 1 Hz [7].

2.4 MODEL AND SIMULATION OF AN HF CHANNEL

2.4.1 Model of the HF Channel

There are two methods available for testing the performance of a transmission system for use on HF radio channels [60]. Firstly, the constructed equipment can be used over an actual HF channel and its performance evaluated by error rate measurements. This method may be costly to implement and very time consuming, since any change required for the adjustments and/or improvements of the equipment could well involve alterations to the hardware. Also, when several systems are to be compared, they have to be tested simultaneously because the same propagation and channel conditions are difficult to obtain at different times due to the random variations in time and frequency of the HF channel. The other disadvantage of this method, is that it is difficult to ascertain the weaknesses of the equipment from its performance, because a poor performance may have one or several causes such as impulsive noise, fading rate, multipath or Doppler shifts. Finally the equipment should be tested for a sufficiently long time to cover most of the likely channel conditions.

The alternative to testing over real channels is to test over a channel simulator [61,60,62-64]. HF channel simulators, whether in hardware or
Fig. 2.3.4: Characteristics of a Raleigh-faded signal

Fig. 2.3.5: Diurnal variation of layer heights
software, are versatile in that a variety of channel conditions can be produced quite simply, and, if desired, these can be repeated any number of times with consistent results. Also the type and amount of distortion can be controlled so that any particular weakness of the system can be identified and studied in isolation. So, a channel simulator, when it is valid, has the advantage of accuracy, regularity of performance, repeatability, availability of a large range of channel conditions, and lower cost, when used in place of an actual HF channel to compare different systems.

Many simulator designs exist and the more relevant have been included in the references [61,63]. These are baseband simulators which means that the modulation and demodulation processes in the radio transmitter and receiver are assumed perfect and linear, and so can be replaced by the corresponding baseband model. The main advantage of a baseband simulator is that the frequency band of the input signal extends over only a few kilohertz, so that modern digital techniques can be employed in their design. This allows very accurate and repeatable characteristics to be obtained.

The hardware simulators require the existence of transmission equipment in hardware form before testing can be carried out. The work which is described in this thesis is based on the software simulation of the equipment and channel simulator, thus simplifying the development and design process of the equipment to changes in the computer simulation program.

Most of the simulator designs given in literature are based on the tapped delay line to represent the HF channel which has been proposed by Watterson et al in reference 65. This model has been adopted unanimously by the International Radio Consultative Committee (CCIR) of the International Telecommunications Union (ITU). This investigation uses the tapped delay-line model as shown in Figure 2.4.1, but operating here on a complex-valued baseband signal. The input signal is fed to an ideal delay line and is delivered at several taps, one for each ionospheric propagation path. Raleigh fading is then imposed on the
delayed signal by multiplying each signal by a suitable tap gain function \( Q_h(t) \). The resulting delayed and modulated signals from the different taps are added to form an output of the tapped delay-line. The received signal is the sum of the output of the delay line and an additional noise term \( V_N(t) \) which represents the noise and/or interference introduced by the HF channel. Although various types of noise, such as atmospheric, man-made and thermal, are present in HF channels, it is a common practice to represent these by white Gaussian noise [62,3].

Now, if we consider only one propagation path the Raleigh fading introduced by the skywave is modelled as in Figure 2.4.2 [66] where \( q_1(t) \) and \( q_2(t) \) are two random processes which must have the following properties [62]:

1. Each random process must be Gaussian with zero mean and the same variance.

2. The random processes \( q_1(t) \) and \( q_2(t) \) must be statistically independent.

3. The power spectrum of each random process must be Gaussian in shape and with the same rms frequency \( f_{\text{rms}} \).

The power spectra of \( q_1(t) \) and \( q_2(t) \) are given by:

\[
|Q_1(f)|^2 = |Q_2(f)|^2 = \exp(-\frac{f^2}{2f_{\text{rms}}^2})
\]  \hspace{1cm} (2.4.1)

and are as shown in Figure 2.4.3.

The frequency spread \( f_{\text{sp}} \), introduced by \( q_1(t) \) and \( q_2(t) \) into an unmodulated carrier is defined [62] as the width of the power spectrum and this is given by:
Fig. 2.4.1: General model of an HF channel

Fig. 2.4.2: Raleigh fading introduced by a skywave
\[ f_{sp} = 2f_{rms} \]  \hspace{1cm} (2.4.2)

The rms frequency is related to the fading rate \( f_e \), which is defined (for a single carrier) as the average number of downward crossings per second of the envelope through the median value, according to the equation [57]:

\[ f_{rms} = \frac{f_e}{1.475} \]  \hspace{1cm} (2.4.3)

From equations 2.4.2 and 2.4.3 \( f_{sp} \) is related to \( f_e \) by:

\[ f_{sp} = 1.356 f_e \]  \hspace{1cm} (2.4.4)

So, for example, a 1 Hz frequency spread is equivalent to a fading rate of 44 fades/min.

2.4.2 Generation of the Random Process \( q_h(t) \) in the Computer Simulation

The random process \( q_1(t) \) is generated by filtering a zero mean white Gaussian noise waveform \( V_1(t) \) as shown in Figure 2.4.4. \( q_2(t) \) is similarly generated but using different Gaussian noise waveforms \( V_2(t) \), which is independent of \( V_1(t) \). The frequency response of the linear filter is also Gaussian so that it matches the required power spectra of a \( q_1(t) \) and \( q_2(t) \).

In a digital implementation of the HF model it is neither possible nor necessary to represent the fading signal \( q_h(t) \) as continuous, such as shown above. Each of these signals must be represented as the corresponding sequence of discrete samples in time. From Nyquist's sampling theorem, the minimum sampling rate required in order to represent \( q_h(t) \) is twice the highest frequency component of \( q_h(t) \).
\[ |Q_1(f)|^2 = |Q_2(f)|^2 \]

Fig. 2.4.3: Power spectrum of \( q_1(t) \) and \( q_2(t) \)

**White noise source** → **Gaussian filter** → **\( q_h(t) \)**

Power spectrum

Probability density

Fig. 2.4.4: Characteristics of the \( q_h(t) \)
As the fading signals have Gaussian spectra, they contain all frequencies, but, for practical purposes, a fading signal of (say) 2 Hz frequency spread could be represented adequately at a sampling frequency of 10 samples/second without any significant aliasing occurring. For reasons connected with the processing of the transmitted data signal, the actual sampling rate used is 4800 samples/second. Unfortunately at this rate the filter poles are very close to the unit circle in the z-plane, and to obtain the desired filter characteristics, these poles must be specified to a very high degree of accuracy, otherwise instability of the filter can occur [68-70]. This problem was solved by the use of a much lower sampling frequency, 50 samples/second, in the digital filters and employing a process of linear interpolation between the samples produced by the filters, to obtain the remaining samples, as shown in Figure 2.4.5 [7]. The 50 Hz sampling frequency was chosen so that it is clearly high enough to satisfy the Nyquist sampling criterion for $q_h(t)$, yet not so low as to introduce inaccuracies in the fading samples when linear interpolation is applied. Sinx/x interpolation has also been used, instead of linear interpolation, and it has been found that this does not affect the model, in the sense that, the error rate of the modem is the same for linear and sinx/x interpolations. The reason for this is the considerable over sampling of $q_h(t)$ (above the Nyquist rate) as illustrated in Figure 2.4.5.

The type of filter used in the computer simulation tests to realise the required Gaussian spectral shaping of $q_h(t)$ is a 5th order Bessel (also known as a Gaussian or linear phase) filter [71]. This filter is implemented using the arrangement in Figure 2.4.6 and the tap values required to obtain the two frequency spreads of interest, namely 0.5 Hz and 2 Hz, are given in Table 2.4.1. The derivation of these tap values, using a 5th order Bessel polynomial as the starting point, is given in Appendix A. The input to the filter $\{V_{h,i}\}$ is a sequence of statistically independent Gaussian random variables with zero mean and fixed variance.
Linear interpolation to produce the remaining fading samples.

Fig. 2.4.5: The linear interpolation process

Input sequence of statistically independent Gaussian random variables with zero mean.

Fig. 2.4.6: Block diagram of the filter used for generation of fading samples $T = (1/50)$ sec.
TABLE 2.4.1: Filter tap values to obtain the required frequency spreads

<table>
<thead>
<tr>
<th>Frequency Spread (Hz)</th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>-1.9</td>
<td>0.903135</td>
<td>-1.9276</td>
<td>0.9316561</td>
<td>-0.946</td>
</tr>
<tr>
<td>2</td>
<td>-1.6218</td>
<td>0.6650064</td>
<td>-1.6954</td>
<td>0.753639</td>
<td>-0.801</td>
</tr>
</tbody>
</table>
2.4.3 Model of the HF Radio Link with n Skywaves

As can be seen from Figure 2.4.2, two signals \( q_1(t) \) and \( q_2(t) \) are required for the fading of one skywave. For the second skywave, another two signals \( q_3(t) \) and \( q_4(t) \) are required, and so on. These signals are generated by the method described previously, but using different input Gaussian noise waveforms \( \{V_{h,i}\} \). The variance of the \( q_{h,i} \) are equal and the actual values depend on the number of skywaves present. For example, if there are two skywaves, the number of signals \( \{q_h(t)\} \) required is 4 and the variance of each \( q_{h,i} \) is 0.25. When there are three skywaves, 6 of the signals \( \{q_h(t)\} \) are required, the variance of each \( q_{h,i} \) becomes 1/6. Thus, by arranging the variance of each \( q_{h,i} \) to be equal to the reciprocal of the number of \( \{q_h(t)\} \) required, the total variance of the \( \{q_{h,i}\} \) equals 1. The reason for this will become clear later. The value of the gain at the input of the filter is adjusted such that the \( \{q_{h,i}\} \) have the required variance.

A complete model of the HF radio link with n skywaves is shown in Figure 2.4.7. In this figure and throughout this thesis \( \tau_1 \) is the relative delay in transmission between the first and second skywaves and \( \tau_2 \) is that between the first and the third skywaves.
Fig. 2.4.7: Model of HF radio link
CHAPTER 3

MODEL OF THE DATA TRANSMISSION SYSTEM

3.1 TRANSMISSION OF DATA USING QUADRATURE AMPLITUDE MODULATION (QAM)

The information to be transmitted over a voice-frequency channel is originally in the form of a baseband signal. The baseband signal cannot itself be transmitted satisfactorily over this type of channel because a significant fraction of the signal power will usually be lost in transmission. Also, the received signal will normally be so severely distorted as to make satisfactory detection impossible [21]. A good number of the voice-frequency channels introduce a small frequency-offset into the transmitted signal. Where this occurs, the whole of the spectrum of the baseband data signal is shifted by a few Hz so as to make it impossible to detect the transmitted signal correctly, unless the frequency shift is determined exactly and suitably corrected for at the receiver [21]. A much simpler approach is to use the original baseband signal to modulate a sinusoidal carrier. The frequency of the carrier is such that the spectrum of the transmitted data signal is placed in the centre of the available frequency band [3]. Among the various possible modulation formats, QAM is frequently used for high-speed data transmission, e.g. 4800 bit/sec and above [21,40]. A QAM signal is the sum of two double sideband suppressed carrier AM signals whose carriers are of the same frequency but in phase quadrature (i.e. at phase angle of 90°). The two AM signals are in element synchronism [67]. Clearly, a QAM system requires two modulators at the transmitter and two coherent demodulators at the receiver, which inevitably complicates the system. However, there are several advantages of using QAM and some of these are summarized as follows [21,43,59]:

1. It is a linear modulation method which greatly simplifies the implementation of the detector.
2. Efficient bandwidth utilization.
3. By using a suitable detection process, no particular phase relationship need be maintained between the reference carriers used for demodulation of the two double sideband AM signals and the suppressed carriers of the two AM signals themselves, just as long as the rate of change of the relative phase angles remains fairly small. The two reference carriers must, of course, themselves be in phase quadrature.

4. High immunity to Gaussian noise.

It is due to these and also other advantages that a QAM system is used in the investigation.

3.2 DATA TRANSMISSION OVER A MODEL OF AN HF CHANNEL USING QAM

In this section, we will consider the transmission of the QAM signals over a model of an HF radio channel that has been described in Chapter 2. Figure 3.2.1 shows the model of the data-transmission system which employs the HF radio channel as the transmission path. Here the input filter A' has the real-valued impulse response a'(t) and its function is to shape the transmitted signal spectrum so that it approximately fits the voice-frequency band of the HF channel. The transfer function A'(f) of filter A' is assumed to be such that

\[ A'(f) = 0 \text{ for } f < -f_C + kf_{sp} \text{ and } f_C > f_C - kf_{sp} \quad (3.2.1) \]

where f_{sp} is the largest value of the frequency spread expected to be introduced by the HF channel into the transmitted QAM signal, and k is an integer the value of which will be discussed later. The radio transmitter (Figure 3.2.1) uses single sideband modulation (SSB) to shift the voiceband spectrum to the HF band, whereas the radio receiver linearly demodulates the received signal to return its spectrum back to the voiceband. The two processes of linear modulation and demodulation ideally do not introduce any distortion into the signal apart from that introduced by the radio equipment
Fig. 3.2.1: Model of the data transmission system using QAM
filters and the HF transmission path itself. It is assumed that white Gaussian noise with zero mean and a two-sided power spectral density of \((1/2)N_0\) is added to the data signal at the output of the HF radio link, as shown in Figure 3.2.1.

At the receiver, the bandpass filter \(C\), after the SSB demodulator, removes the noise frequencies outside the data signal band without excessively distorting the signal. The impulse response of this filter is \(c(t)\). The noisy, Raleigh-fading and distorted QAM signal at the output of the filter \(C\) is fed to two coherent demodulators whose reference carriers are in phase quadrature and have the frequency \(f_c\), which is equal to the average instantaneous carrier frequency of the received signal, thus eliminating any constant frequency offset which may be present in the received QAM signal. In Figure 3.2.1 the frequency response of the receiver filter \(B'\) is such that

\[
B'(f) = 0 \text{ for } |f| > f_c
\]

Now in Figure 3.2.1 if:

1. The baseband signals transmitted over the in phase channel (that associated with \(\cos(2\pi f_c t)\)) are represented by real valued quantities, and the baseband signals over the quadrature channel (that associated with \(\sin(2\pi f_c t)\)) by imaginary-valued quantities. (The bandpass signals are all real valued).

2. The single sideband modulators and demodulators are replaced by only their corresponding filters \(d(t)\) and \(g(t)\) respectively. Then the equivalent model of the data transmission system will be as shown in Figure 3.2.2. In Figure 3.2.2 the input to the system is the stream of data elements \(\sum_i s_i \delta(t - iT)\) where

\[
s_i = s_{r,i} + js_{q,i} \quad j = \sqrt{-1}
\]
Fig. 3.2.2: Equivalent model of the data transmission system
Here, \( s_i \) is a scalar and may have one of finite number of complex values and \( 1/T \) is the signal-element rate in bauds. The QAM signal \( x_2(t) \) is the real part of the complex signal \( x(t) \) at the output of the modulator.

From Figure 3.2.1

\[
x_2(t) = \sqrt{2} \left[ \sum_{i} s_{r,i} a'(t-iT) \cos \omega_c t - \sum_{i} s_{q,i} a'(t-iT) \sin \omega_c t \right] \tag{3.2.4}
\]

or equivalently (Figure 3.2.2):

\[
x_2(t) = \sqrt{2} \text{Re} \left[ \sum_{i} s_{i} a'(t-iT) e^{j2\pi f_c t} \right] \tag{3.2.5}
\]

where

\[
e^{j2\pi f_c t} = \cos 2\pi f_c t + j\sin 2\pi f_c t \tag{3.2.6}
\]

This signal \( x_2(t) \) is fed to the transmitter filter \( G \) of the radio equipment. The impulse response of this filter is \( g(t) \) and has the voiceband transfer function \( G(f) \). The real-valued QAM signal at the output of the radio transmitter filter \( G \)

\[
x(t) = \text{Re} \left[ \sqrt{2} \sum_{i} s_{i} a'(t-iT) e^{j2\pi f_c t} \right] \ast g(t) \tag{3.2.7}
\]

or equivalently

\[
x(t) = \frac{1}{\sqrt{2}} \left[ \sum_{i} s_{i} a'(t-iT) e^{j2\pi f_c t} + s_{i}^* (a'(t-iT))^* e^{-j2\pi f_c t} \right] \ast g(t) \tag{3.2.8}
\]

where

\( s_{i}^* \) and \((a'(t))^*\) are the complex conjugates of \( s_{i} \) and \( a'(t) \) respectively.
But [72]

\[ [f_1(t) * f_2(t)] e^{-j2\pi f c t} = (f_1(t)e^{-j2\pi f c t}) * (f_2(t)e^{-j2\pi f c t}) \]

(3.2.9)

Hence

\[ x(t) = \frac{1}{\sqrt{2}} \left[ \sum s_1 a(t-iT)e^{j2\pi f c t} + s_1^* (a(t-iT))^* e^{-j2\pi f c t} \right] \]

(3.2.10)

where

\[ a(t-iT) = a'(t-iT)^* (g(t)e^{-j2\pi f c t}) \]

(3.2.11)

which represents the overall filtering carried out at the transmitting side of the baseband signal.

The Hilbert transform of \( x(t) \) is given by:

\[ \hat{x}(t) = x(t) * f(t) \]

(3.2.12)

where \( f(t) \) is the impulse response of the Hilbert transform whose Fourier transform is:

\[ F(f) = \begin{cases} 
   j & f < 0 \\
   0 & f = 0 \\
   -j & f > 0
\end{cases} \]

(3.2.13)

Substituting for \( x(t) \) from equation 3.2.10 into equation 3.2.12 we have:
\[ \hat{x}(t) = \frac{1}{\sqrt{2}} \left\{ \sum s_i [a(t-iT) e^{j2\pi f_c t}]*f(t) \right\} e^{-j2\pi f_c t} + s_i^* [(a(t-iT))^* e^{j2\pi f_c t}]*f(t) \} \] (3.2.14)

which may be written as (equation 3.2.9):

\[ \hat{x}(t) = \frac{1}{\sqrt{2}} \left\{ \sum s_i [a(t-iT)^* f(t) e^{j2\pi f_c t}] e^{-j2\pi f_c t} + s_i^* [(a(t-iT))^* f(t) e^{j2\pi f_c t}] e^{-j2\pi f_c t} \right\} \] (3.2.15)

From equation 3.2.11 the frequency response of \( a(t) \) is bandlimited to that of \( A'(f) \) (equation 3.2.1). The Fourier transform of \( f(t)e^{j2\pi f_c t} \) (i.e. \( F(f+f_C) \)), has the value \(-j\) over the frequency band \(-f_C\) to \(f_C\), whereas the Fourier transform of \( f(t)e^{-j2\pi f_c t} \) (i.e. \( F(f-f_C) \)) has the value \(j\) over the same frequency band, as shown in Figure 3.2.3. Consequently, by taking the Fourier transform of \( \hat{x}(t) \), replacing \( F(f+f_C) \) and \( F(f-f_C) \) by their corresponding values over the frequency band of \( A(f) \), \((-f_C\) to \(f_C\)), and taking the inverse Fourier transform, equation 3.2.15 becomes

\[ \hat{x}'(t) = \frac{1}{\sqrt{2}} \left[ \sum \left( -j s_i a(t-iT) e^{j2\pi f_c t} + j s_i^* [(a(t-iT))^* e^{j2\pi f_c t}] \right) \right] \] (3.2.16)

In order to simplify the description of the system, it is assumed that the HF channel has only two independent Raleigh fading skywaves. The relative delay in transmission between the skywaves is \( \tau \) seconds. Hence the signal at the output of the model of the HF radio channel (see Section 2.4) is given by

\[ z(t) = x(t)q_1(t) + \hat{x}'(t)q_2(t) + x(t-\tau)q_3(t) + \hat{x}(t-\tau)q_4(t) \] (3.2.17)
Fig. 3.2.3: Frequency responses involved in equation 3.2.15
From equation 3.2.16 and 3.2.10 we have

\[ z(t) = \frac{1}{\sqrt{2}} \left\{ \sum s_i a(t-iT) [q_1(t) - jq_2(t)] e^{\frac{j2\pi f_c t}{T}} + s_i^* (a(t-iT))^\ast [q_1(t) + jq_2(t)] e^{-\frac{j2\pi f_c t}{T}} 
+ s_i a(t-iT) [q_3(t) - jq_4(t)] e^{\frac{j2\pi f_c (t-\tau)}{T}} - s_i^* (a(t-iT))^\ast [q_3(t) + jq_4(t)] e^{-\frac{j2\pi f_c (t-\tau)}{T}} \right\} \] (3.2.18)

If we let

\[ h_i(t-iT) = a(t-iT) [q_1(t) - jq_2(t)] e^{-\frac{j2\pi f_c T}{T}} + a(t-\tau-iT)[q_3(t) - jq_4(t)] \] (3.2.19)

then equation 3.2.18 can be written as

\[ z(t) = \frac{1}{\sqrt{2}} \left\{ \sum s_i h_i(t-iT) e^{\frac{j2\pi f_c t}{T}} + s_i^* (h_i(t-iT))^\ast e^{-\frac{j2\pi f_c t}{T}} \right\} \] (3.2.20)

The delay is assumed to be a constant. Therefore, the factor \( e^{-\frac{j2\pi f_c T}{T}} \) in equation 3.2.19 is a complex-valued scalar with absolute value of 1, and since \( q_3(t) \) and \( q_4(t) \) are statistically independent with zero mean, the delay has no effect on the statistical properties of \( [q_3(t)-jq_4(t)] e^{-\frac{j2\pi f_c T}{T}} \) nor does it affect the power spectrum of this signal. Thus we may rewrite equation 3.2.19 as:

\[ h_i(t-iT) = a(t-iT)[q_1(t) - jq_2(t)] + a(t-\tau-iT)[q_3(t)-jq_4(t)] \] (3.2.21)
The signal at the output of the linear demodulator in Figure 3.2.2 is now

\[ r(t) = \sqrt{2} \left\{ [z(t) \cdot d(t) \cdot c(t)] e^{-j2\pi f_{c}t} \right\} \cdot b(t) \]

+ \sqrt{2} \left\{ [n(t) \cdot d(t) \cdot c(t)] e^{-j2\pi f_{c}t} \right\} \cdot b'(t) \]  

(3.2.22)

If we let

\[ b(t) = \left\{ [d(t) \cdot c(t)] e^{-j2\pi f_{c}t} \right\} \cdot b'(t) \]  

(3.2.23)

and

\[ w(t) = \sqrt{2} \left\{ [n(t) \cdot d(t) \cdot c(t)] e^{-j2\pi f_{c}t} \right\} \cdot b'(t) \]  

(3.2.24)

equation 3.2.22 may be written as (equation 3.2.9)

\[ r(t) = \sqrt{2} \left[ z(t) e^{-j2\pi f_{c}t} \right] \cdot b(t) + w(t) \]  

(3.2.25)

b(t) represents the overall filtering carried out on the signal at the receiver end of the system considered as operating on the demodulated baseband signal and w(t) is the Gaussian noise component in r(t). Taking into account that \( f_{c}' = f_{c} \) equations 3.2.20 and 3.2.25 give

\[ r(t) = \sqrt{2} \left[ z_{1}(t-iT) + s_{1}(h_{1}(t-iT)) e^{-j4\pi f_{c}t} \right] \cdot b(t) + w(t) \]  

(3.2.26)

Now, \( h_{1}(t-iT) \) consist of the time invariant impulse response a(t) and the random components \( q_{1}(t) \) to \( q_{4}(t) \). As mentioned before, each of
these random components has a Gaussian-shaped power spectral density, the rms frequency of which is of the order of a few Hz, and which extends over all frequencies. However since this power density decreases very sharply when f increases, the frequency response of h_i(t-iT) may be considered to be strictly bandlimited and given by the Fourier transform of a(t) dispersed both upwards and downwards by less than 10 f_{rms}. According to equation 3.2.11 the Fourier transform of a(t) is bandlimited to the frequency band of A(f)', the transfer function of a(t)', which is defined by equation 3.2.1. Thus, if k in equation 3.2.1 is equal to 5 (f_{sp} = 2f_{rms}) then the frequency response of h_i(t-iT) will be strictly bandlimited such that

$$|H(f)| = 0 \text{ for } |f| > f_c$$

(3.2.27)

Hence, the Fourier transform of (h_i(t-iT))^{*}e^{-j4\pi f_c t} in equation 3.2.26 will be outside the passband of the lowpass filter whose impulse response is b(t) and so equation 3.2.26 reduces to

$$r(t) = \sum_i s_i h_i(t-iT)^{*}b(t) + w(t)$$

(3.2.28)

Let

$$y_i(t-iT) = h_i(t-iT)^{*}b(t)$$

(3.2.29)

then equation 3.2.28 becomes:

$$r(t) = \sum_i s_i y_i(t-iT) + w(t)$$

(3.2.30)

The impulse response y_i(t-iT) in equation 3.2.29 may also be written as:

$$y_i(t-iT) = \{a(t-iT) [q_1(t)-jq_2(t)] + a(t-iT) [q_3(t)-jq_4(t)]\}^*b(t)$$

(3.2.31)
\(y_i(t-iT)\) is the impulse response of the linear baseband channel, which is obviously time varying.

Equation 3.2.30 represents the baseband model of the QAM system over the HF radio link and this is shown in Figure 3.2.4 (for two skywaves). In this model, the impulse response \(a(t)\) represents the overall filtering at the transmitter side of the data transmission system, and \(b(t)\) is the overall filtering at the receiver side of the system, each impulse response being, of course, a baseband waveform. The characteristics of these filters are given in Figures 3.2.5-3.2.7 [72] where the filters are for convenience, shown as the equivalent bandpass filter operating on the voiceband signal since this enables them to be compared with corresponding radio filters.

The average transmitted energy per signal element at the output of the transmitter filter in Figure 3.2.4 is given by:

\[
E_t = E \left[ \int_{-\infty}^{\infty} |s_i a(t-iT)|^2 \, dt \right] \quad (3.2.32)
\]

where \(E[.]\) is the expected value of \([.].\).

If we denote

\[
\overline{s_i^2} = E[|s_i|^2] \quad (3.2.33)
\]

then, using Parseval's theorem, equation 3.2.32 becomes

\[
E_t = \overline{s_i^2} \int_{-\infty}^{\infty} |A(f)|^2 \, df \quad (3.2.34)
\]

the average energy per signal-element at the input to the receiver filter in Figure 3.2.4 is given by:
Fig. 3.2.4: Baseband model of the QAM system over the HF radio link
Fig. 3.2.5: Characteristics corresponding to the impulse response
\[ j2\pi f_t \left[ a(t) \ast b(t) \right] e^{-ct} \]
Fig. 3.2.6: The attenuation and group delay characteristics, corresponding to the transfer function of radio filters G and D in cascade over the positive frequencies.
Fig. 3.2.7: Characteristics corresponding to the impulse response
\[ a'(t) \ast \{ c(t) e^{-j2\pi f_c t} \} \ast b'(t) \} e^{j2\pi f_c t} \]
\[
E_r = E \left[ \int_{-\infty}^{\infty} \left| \sum_{i=1}^{4} a(t-iT)(q_1(t)-jq_2(t) + a(t-\tau-iT)(q_3(t)-jq_4(t)) \right|^2 \, dt \right]
\]

\[
= \bar{s}_1^2 \left( \bar{q}_1^2(t) + \bar{q}_2^2(t) + \bar{q}_3^2(t) + \bar{q}_4^2(t) \right) \int_{-\infty}^{\infty} |A(f)|^2 \, df
\]

(3.2.35)

where \( \bar{q}_1^2(t), \bar{q}_2^2(t), \bar{q}_3^2(t) \) and \( \bar{q}_4^2(t) \) are the variances of \( q_1(t), q_2(t), q_3(t) \) and \( q_4(t) \) respectively. Clearly, when these four variances are equal and their sum equals 1, the energy \( E_r \) is the same as \( E_t \). This means that the skywaves do not introduce, on average, any gain or attenuation into the transmitted data signal, which greatly simplifies the calculation of the signal to noise ratio in the computer simulation tests.

The above derivation is a development of the one given in reference 72. The latter has been modified here to make it more rigorous.

It can be shown using similar derivations, that if there are \( n \) skywaves with delays \( \tau_1, \tau_2, \ldots, \tau_{n-1} \) then

\[
y_i(t-iT) = (a(t-iT)[q_1(t)-jq_2(t)] + a(t-\tau_1-iT)[q_3(t)-jq_4(t)] + \ldots
\]

\[
+ a(t-\tau_{n-1}-iT) [q_{2n-1}(t)-jq_{2n}(t)] \right) \ast b(t)
\]

(3.2.36)

Also it can be shown that the model for a QAM data transmission system over a time invariant channel (transmission path) will be as given in Figure 3.2.8 [43]. Now, the difference with Figure 3.2.4 is that \( m(t) \) is a fixed (time invariant) impulse response. This means that the impulse response of the linear baseband channel \( y(t) \), is obviously time invariant and
Fig. 3.2.8: Model of a QAM data transmission system (time invariant channel)
\[ r(t) = \sum_{i} s_i y(t-iT) + w(t) \]  

(3.2.37)

Here \[ y(t) = a(t) * m(t) * b(t) \]  

(3.2.38)

### 3.3 MODEL OF THE HF DATA TRANSMISSION SYSTEM USED IN THE TESTS

The data transmission system is a synchronous serial system, with a 16-level QAM signal and an adaptive detection process. The information to be transmitted is a sequence of binary digits of \( a_K \) where \( a_K = 0 \) or \( 1 \). The \( \{a_K\} \) being statistically independent and equally likely to have either binary value. The \( \{a_K\} \) are fed, at a rate of 9600 bit/sec, to the encoder which gives at its output the corresponding sequence of data-symbols \( s_i \) at a rate of 2400 symbols/sec where

\[ s_i = s_{r,i} + js_{q,i} \]  

(3.3.1)

and \( s_{r,i} = \pm 1 \) or \( \pm 3 \) and \( s_{q,i} = \pm 1 \) or \( \pm 3 \). Thus, the \( s_{r,i} \) and \( s_{q,i} \), which are real and imaginary parts of \( s_i \), are 4-level data symbols whereas \( s_i \) is a 16-level data symbol (\( s_i \) constellation is shown in Figure 3.3.1). The \( \{s_i\} \) are statistically independent and equally likely to have any of their 16 possible values. The data symbols \( \{s_i\} \) are fed to a model of the QAM system over the HF radio link as described in Section 3.2 and given in Figure 3.2.4. The impulse response of the linear baseband channel is time varying and for two skywaves at time \( t=iT \) is given by:

\[ y_i(t-iT) = [a(t-iT)[q_1(t)-jq_2(t)] + a(t-iT)[q_3(t)-jq_4(t)]]*b(t) \]  

(3.3.2)

where \[ a(t) = a(t)^* (g(t)e^{-j2\pi f_c t}) \]  

(3.3.3)

and \[ b(t) = ([d(t)*c(t)]e^{-j2\pi f_c t})*b(t)^* \]  

(3.3.4)
The impulse response of the transmitter and receiver filters used in the tests, are those given by the above equations and shown in Figures 3.2.5-3.2.7 after converting them to minimum phase to get $a''(t)$, $b''(t)$. The reasons and methods used for converting the impulse response to minimum phase are discussed in Reference 59. Hence, for the minimum phase filters equation 3.3.2 can be written as:

$$y_i(t-iT) = \{a''(t-iT)[q_1(t)-jq_2(t)] + a''(t-T-iT)[q_3(t)-jq_4(t)]\}*b(t)$$

(3.3.5)

The carrier frequency of the QAM signal has the value of $f_c = 1800$ Hz.

The demodulated baseband signal $r(t)$ at the output of the QAM system model comprises the stream of data elements $\{s_iY_i(t-iT)\}$ to which is added the stationary zero mean complex-valued baseband Gaussian noise waveform $w(t)$. The waveform $r(t)$ is sampled, once per data-symbol $s_i$, at the time instants $iT$. The sampling rate is assumed to be correct. The delay in transmission is, for convenience, taken to be such that the first potentially non-zero sample of a received signal element arrives with no delay. The complex valued sample of $r(t)$ at time $iT$ is now

$$r_i = \sum_{h=0}^{g} s_{i-h} Y_{i,h} + w_i$$

(3.3.6)

where $Y_{i,h} = Y_{i-h}(iT)$

(3.3.7)

and $Y_{i,h} = 0$ for $h<0$ and $h>g$ for practical purposes. The sequence of complex values given by the vector

$$Y_i = Y_{i,0}, Y_{i,1} \cdots Y_{i,g}$$

(3.3.8)
is taken to be the sampled impulse response of the linear baseband channel at time $t=iT$.

In the computer simulation tests, the vector $Y_i$ is obtained by sampling the $\{y_i(t-iT)\}$ at a rate of 2400 samples/sec. Since the process is performed in the discrete time domain, and to avoid any aliasing likely to occur when any of the $q_1(t)$, $q_2(t)$, $q_3(t)$ or $q_4(t)$ is changing rapidly, the convolution in equation 3.3.5 is carried out with a sampling rate of 4800 samples/sec which is well above the Nyquist rate for filters A and B. The functions $q_1(t)$, $q_2(t)$, $q_3(t)$ and $q_4(t)$ are generated as described in Chapter 2. Let the three sequences $A_1$, $A_2$ and $B_1$ represent the three impulse responses $a''(t)$, $a''(t-\tau)$ and $b(t)$ respectively sampled at 4800 samples/sec.

\[
\begin{align*}
A_1 &= a''_{1,0} \ a''_{1,1} \ \ldots \ a''_{1,p} \\
A_2 &= a''_{2,0} \ a''_{2,1} \ \ldots \ a''_{2,p} \\
B_1 &= b'_{0} \ b''_{1} \ \ldots \ b''_{p}
\end{align*}
\] (3.3.9)

\[
\begin{align*}
a''_{1,k} &= a''(kT/2) \\
a''_{2,k} &= a''(kT/2-\tau) \\
b''_{k} &= b''(kT/2)
\end{align*}
\] (3.3.10)

$p$ is an integer the value of which will become clear later on and $1/T$ is the data-symbol rate of 2400 symbols/sec.

For practical purposes (see Figure 3.3.2)

\[
a''(t) = b''(t) = 0 \text{ for } t < 0 \text{ and } t > (p-p')T/2
\] (3.3.11)

$p'$ being an integer such that
Fig. 3.3.1: Data symbol $s_1$ constellation

Amplitude

$Rm[a''(t)]$ $Rm[a''(t-\tau)]$

0 $p'T/2$ $\tau$ $p'T/2$

Fig. 3.3.2: The timing relationship between the transmitter filter impulse response (real or imaginary) and its delayed version
\[ \tau = p' \frac{T}{2} + \tau' \]  

(3.3.12)

and \( \tau' < \frac{T}{2} \). This implies that

\[ a_{1,k}^{''} = b_{k}^{''} = 0 \text{ for } k<0 \text{ and } k>p-p' \]  

(3.3.13)

\[ a_{2,k}^{''} = 0 \text{ for } k<p' \text{ and } k>p \]  

(3.3.14)

Also note that \( p-p' \) is a fixed quantity, since \( a(t) = b(t) \) effectively has a fixed length. Hence

\[ (p-p') \frac{T}{2} = P_f \frac{T}{2} \]  

(3.3.15)

and \( P_f \) is a constant.

The sampled impulse responses of \( A_1 \) and \( B_1 \), used in the tests, are given in Table 3.3.1. Also, the four components \( q_1(t) \), \( q_2(t) \), \( q_3(t) \) and \( q_4(t) \) are sampled at 4800 samples/sec, and the corresponding resultant samples up to time instant \( t=iT \) are represented by the four sequences:

\[ Q_{1,i} = q_{1,1} \quad q_{1,2} \quad \ldots \quad q_{1,2i} \]  

(3.3.16)

\[ Q_{2,i} = q_{2,1} \quad q_{2,2} \quad \ldots \quad q_{2,2i} \]

\[ \vdots \]

\[ Q_{4,i} = q_{4,1} \quad q_{4,2} \quad \ldots \quad q_{4,2i} \]
TABLE 3.3.1: The sampled impulse response of the minimum-phase transmitter and receiver filters sampled at 4800 sample/second

<table>
<thead>
<tr>
<th>Sampled impulse response of the transmitter filter A1</th>
<th>Sampled impulse response of the receiver filter B1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real part</td>
<td>Imaginary part</td>
</tr>
<tr>
<td>-0.1795896</td>
<td>+2.3539405</td>
</tr>
<tr>
<td>-3.0773453</td>
<td>+20.7590237</td>
</tr>
<tr>
<td>-9.9409021</td>
<td>+45.5584592</td>
</tr>
<tr>
<td>-11.7869473</td>
<td>+41.4909978</td>
</tr>
<tr>
<td>-3.4618271</td>
<td>+8.7045826</td>
</tr>
<tr>
<td>4.4438154</td>
<td>-11.7869820</td>
</tr>
<tr>
<td>3.0642536</td>
<td>-5.5819054</td>
</tr>
<tr>
<td>-1.3596576</td>
<td>+3.1582131</td>
</tr>
<tr>
<td>-1.4973528</td>
<td>+1.7365460</td>
</tr>
<tr>
<td>0.2925598</td>
<td>-0.7776891</td>
</tr>
<tr>
<td>0.5180829</td>
<td>-0.1292556</td>
</tr>
<tr>
<td>-0.1842786</td>
<td>+0.2880296</td>
</tr>
<tr>
<td>-0.3167778</td>
<td>-0.2324818</td>
</tr>
<tr>
<td>0.0021899</td>
<td>-0.2107548</td>
</tr>
<tr>
<td>-0.0443806</td>
<td>+0.0392056</td>
</tr>
<tr>
<td>0.0515533</td>
<td>+0.0098505</td>
</tr>
</tbody>
</table>
Now, from equation 3.3.5 it can be seen that $y_i(t-iT)$ is the result of a convolution between $a''(t-iT)$, modified by the time varying HF link with the receiver impulse response $b''(t)$, and similarly for $a''(t-\tau-iT)$.

Hence equation 3.3.7 can be written:

$$Y_{i,h} = (T/2) \sum_{k=0}^{2h} [a''_{1,k} (q_1, 2(i-h)+k - jq_2, 2(i-h)+k)]$$

$$+ a''_{2,k} (q_3, 2(i-h)+k - jq_4, 2(i-h)+k)]b''_{2h-k}$$

(3.3.17)

for $h = 0, 1, \ldots, g$.

From equations 3.3.13-3.3.14 it can be shown that [21]

$$g = \frac{2p-p'+1}{2} = \frac{2p_1 + p' + 1}{2}$$

(3.3.18)

Hence $g$ is a function of $p'$ and is therefore a function of $\tau$ also (equation 3.3.12). For example for a delay $\tau$ of 3 msec and $p_1$ of 15 (Table 3.3.1) from equation 3.3.12 $p' = 14$, $(\tau' = 0.4T/2)$ hence $g = 22$ (equation 3.3.18). Similarly for a delay $(\tau)$ of 1 msec $g = 17$.

The detector in Figure 3.2.4 operates on the received samples $\{r_i\}$ to produce the detected data symbols $\{s_i\}$, which should be the same as the sequence $\{s_i\}$ if the detection is correct.

Suppose that, on the receipt of the signal sample $r_i$, the detector determines the value of $s_i$ from $r_i$. Now, if $Y_{i,h}$ have a non-zero value for $h = 0, 1, \ldots, g$ then the channel is said to introduce signal distortion and equation 3.3.6 may be written as:

$$r_i = s_i Y_{i,0} + \sum_{h=1}^{g} s_{i-h} Y_{i,h} + w_i$$

(3.3.19)
the term \( \sum_{h=1}^{q} s_i - h y_i, h \) represents intersymbol interference.

If the intersymbol interference is sufficiently small, it can be tolerated by a simple threshold detector, which detects \( s_i \) as its nearest possible value to \( r_i / y_i, o \). However, under typical conditions, correct decisions may not be obtained, even in the absence of noise. Several techniques are available to handle the detection of digital data signals in the presence of intersymbol interference. These techniques are the subject of the next chapter.

When the detected data symbols are determined they are fed into the decoder which gives at its output the corresponding binary digits \( a_k' \) where \( a_k' = 0 \) or \( 1 \). The \( \{a_k'\} \) being the detected value of the transmitted binary digits \( \{a_k\} \) (see beginning of Section 3.3). The details of the encoder and decoder are given in Appendix B.
CHAPTER 4

DETECTION PROCESSES FOR DISTORTED SIGNALS

4.1 INTRODUCTION

Since the 1960s the efforts of many communication engineers has been directed to designing fast data transmission systems over telephone and radio channels. Increasing the data transmission rate over band-limited channels results in an increasing amount of intersymbol interference, which leads to an increase in error rate. Detection processes for distorted digital signals may be classified into two separate groups.

In the first group an equalizer is employed to remove the intersymbol interference from the received sample \( \{ r_i \} \) (equation 3.3.19), such that the signal at the detector input becomes:

\[
\delta_i = s_i + u_i
\]

and \( u_i \) is a function of the \( \{ w_i \} \) (see equation 3.3.19) according to a relationship determined by the equalizer. The resultant received signal at the output of the equalizer \( \{ \delta_i \} \), is then detected in the conventional manner, as normally applied to a serial digital signal in the absence of intersymbol interference, by comparing the corresponding sample value with the appropriate threshold level (or levels). This group of detection processes includes linear equalization [21,29]. Basically linear equalization is a process of linear filtering of the distorted signal by a transversal filter (Figure 4.1.1).

Two criteria have found widespread use in optimizing the transversal filter tap gains. One criterion is to minimize the peak distortion. In this case the filter is called a zero-forcing filter, and the equalizer acts as the inverse of the channel, so that the channel and equalizer
together introduce negligible distortion. The other criteria is to minimise the mean square error due to both noise and intersymbol interference in the output signal. This generally gives a more useful or effective degree of equalization for a given number of taps, than an equalizer that minimizes the peak distortion. When the signal distortion varies with time (as in HF links), not only must the receiver detect the data-symbol values of the received data signal, but it must also continuously keep adjusting the equalizer so that this is held correctly adjusted in relation to the channel.

The first group of detection processes also includes Decision Feedback equalization (DFE) [16,18,20,21,29,32,33]. A decision feedback equalizer is shown in Figure 4.1.2 and consists of two sections: a feedforward section and a feedback section. The input to the feedforward section is the sequence of received signal samples. The feedback section has as its input a sequence of detected data symbols. In many practical applications, the DFE equalizer is adjusted adaptively (using gradient [21,19], or Kalman [17,19,22], or Lattice algorithms [31]) to minimize the mean square error in the equalized signal at its output.

These DFEs, in adjusting the tap gains to minimize the mean square error, assume that the channel is modelled as a first order Markov process, which is not a valid model for HF channels [66]. All the available evidence [59] shows that this assumption leads to an adjustment of the tap gains that does not minimize the mean square error for an HF channel. The performance of the DFE over such channels is consequently degraded.

Another arrangement of the DFE that has been extensively studied, minimizes the mean square error in the equalized signal, subject to the perfect equalization of the channel [21]. This DFE is also referred to here as the conventional nonlinear equalizer. At high signal to noise ratios the conventional nonlinear equalizer has about the same performance as the DFE that minimizes the mean square error [32]. These DFE equalizers can sometimes achieve a much better tolerance to additive noise than linear equalizers [32].
Fig. 4.1.1: Linear equalizer

Fig. 4.1.2: Decision feedback equalizer
In the second group of detection processes, the decision process itself is modified to take account of the signal distortion that has been introduced by the channel. This group includes maximum-likelihood detectors [21]. In maximum-likelihood detection, instead of removing the intersymbol interference (as is done in the previous methods) the detector takes full account of the intersymbol interference, thus using the entire transmitted energy in the detection process. With the appropriate design of the demodulator and associated filter in the receiver, white Gaussian noise at the receiver input gives statistically independent Gaussian noise samples at the maximum-likelihood detector input. Under these conditions a maximum-likelihood detector selects as the detected message the possible sequence of transmitted data symbols for which there is the minimum mean square difference between the samples of the corresponding received data signal, for the given signal distortion but in the absence of noise, and the samples of the signal actually received. When the transmitted data symbols are statistically independent and equally likely to have any of their possible values, the maximum-likelihood detector minimizes the probability of error in the detection of the received message, and in this sense offers the best performance possible [21,29]. However, it is not feasible to implement the maximum-likelihood detection process in practice because of the excessive equipment complexity involved.

It has also been shown that, with a sufficient delay in detection, a detection process employing the Viterbi-algorithm can (for practical purposes) achieve the same tolerance to noise as that of a maximum-likelihood detection process [34,73,29]. Although the Viterbi-algorithm requires fewer operations and much less storage than the ideal maximum-likelihood detection process, its equipment complexity grows exponentially with the length of the channel sampled impulse response. Unfortunately, in the transmission of data at 9600 bit/sec over HF link, the sampled impulse response of the channel contains a large number of components, and even the Viterbi-algorithm now involves both an excessive amount of storage and an excessive number of operations per data symbol. The precise reason for this will be explained in more detail later on, when describing this algorithm.
The second group of detectors includes also the Viterbi-algorithm operating with a desired impulse response. One approach for overcoming the problem of high complexity in the Viterbi-algorithm is to use a linear feedforward transversal filter at the detector input to reduce the number of components in the channel sampled impulse response [35-38, 41,44]. The disadvantage of this arrangement is that, the linear filter may equalize some of the amplitude distortion (Appendix C) introduced by the channel and under these conditions maximum-likelihood detection is no longer achieved [21], leading to an inferior performance. The linear filter should ideally perform the function of a "whitened matched filter" [34] which ensures that true maximum-likelihood detection is achieved by the Viterbi-algorithm detector.

The remaining detectors in the second group are a reduced state Viterbi detector and a near-maximum-likelihood detector. A promising approach that has recently been considered is to modify the Viterbi-algorithm detector itself in such a way as to reduce both the amount of storage and the number of operations per received data-symbol, but without reducing unduly the tolerance of the detector to noise. Some of the earlier of these systems are referred to as reduced-state Viterbi algorithm detectors [37] but others differ in so many details from the Viterbi algorithm that they are best referred to simply as near-maximum-likelihood detectors [42]. These near-maximum-likelihood detectors do not require the number of components in the sampled impulse response of the channel to be reduced to some small value and can therefore be used with a noise whitened match filter (or its equivalent) ahead of the detector.

In a further development of the latter techniques, the noise whitened matched filter is omitted altogether [45,8,46]. The weakness of this arrangement is that without the noise whitened matched filter at the input to the near-maximum-likelihood detector, all phase distortion introduced by the transmission path appears in the data signal at the detector input. Computer simulation tests have shown that in the presence of very severe phase distortion, such as can occur over the HF
link, considerable equipment complexity is now required for the correct operation of the detector [52,11]. In the preferred system (which is used in this thesis) the receiver employs an adaptive linear feedforward transversal filter ahead of the near-maximum-likelihood detector, the filter being the effective equivalent of (but not the same as) a noise-whitened matched filter. This filter is adjusted adaptively to remove all phase distortion introduced by the channel, and furthermore to shape the received signal elements to be "minimum phase", that is, with the energy of each signal element concentrated towards its start. The latter condition enables near-maximum-likelihood detection to be achieved by means of a simple detector [45]. The adaptive adjustment of the linear filter can be carried out in a relatively simple manner by a recently developed method [74].

In this chapter, the conventional nonlinear equalizer, the Viterbi-algorithm, and the proposed near-maximum-likelihood detectors are described in detail. This is done firstly for time invariant channels. Then the necessary modifications required by these detectors, to work with an HF channel, are described. Finally the results are given of computer simulation tests on the detection processes, when operating with an HF link.

4.2 DETECTION PROCESSES FOR TIME INVARIANT CHANNELS

4.2.1 Model of System

The model of the synchronous serial data-transmission system is shown in Figure 4.2.1. This is a synchronous serial 16-point QAM system. The information to be transmitted is carried by the data symbol \( s_i \)

\[
\text{where } s_i = \pm(1 \text{ or } 3) \pm j(1 \text{ or } 3) \text{ and } j = \sqrt{-1} \quad (4.2.1)
\]

The \( s_i \) are statistically independent and equally likely to have any of their 16 possible values. A finite sequence of \( b+1 \) signal elements in the form of the impulses \( \{ s_i \delta(t-iT) \}_{i=0,1,\ldots,b} \), is now fed to the linear baseband channel Figure 4.2.1, and it is assumed that \( s_i = 0 \) for \( i<0 \) and \( i>b \). The linear baseband channel includes the lowpass filter A
Fig. 4.2.1: Model of data transmission system
(transmitter output filter), a linear baseband transmission path and the lowpass filter B (receiver input filter). The receiver filter is a lowpass filter having a flat amplitude response and linear phase characteristics and a cut off frequency at half the signal element rate. The resultant channel has an impulse response \( y(t) \), which has, for practical purposes, a finite duration. White Gaussian noise is added to the data signal at the output of the transmission path to give the bandlimited Gaussian noise waveform \( w(t) \), with zero mean and fixed variance at the output of the lowpass filter B. The output waveform from the linear baseband channel is

\[
r(t) = \sum_i s_i y(t-iT) + w(t) \quad (4.2.2)
\]

where \( r(t) \), \( y(t-iT) \) and \( w(t) \) are complex valued. The received waveform \( r(t) \) is sampled at the time instants \( \{iT\} \) to give the samples \( \{r_i\} \) which are fed to the linear feedforward transversal filter D, whose corresponding output samples are the \( \{v_i\} \). The real and imaginary parts of the noise components in the \( \{r_i\} \) are statistically independent Gaussian random variables with zero mean and fixed variance. The resultant sampled impulse response of the linear baseband channel and filter D is given by the vector (sequence)

\[
E = [1 \ e_1 \ e_2 \ldots \ e_g] \quad (4.2.3)
\]

with z-transform

\[
E(z) = 1 + e_1 z^{-1} + \ldots + e_g z^{-g} \quad (4.2.4)
\]

The delay in transmission over the channel and filter D, other than that involved in the time dispersion of the signal, is neglected here, so that the first component of \( E \) (with value unity) is taken to have no
delay. Also \( e_i = 0 \) for \( i < 0 \) and \( i > g \).

All zeros of \( E(z) \) lie inside or on the unit circle in the \( z \)-plane. Furthermore, the zeros of \( E(z) \) are derived from the zeros of the \( z \)-transform of the sampled impulse response of the linear baseband channel, Figure 4.2.1., by replacing all zeros of the latter that lie outside the unit circle by the complex conjugates of their reciprocals, all remaining zeros being left unchanged \([47, 32]\). Thus, the linear filter \( D \) is an all pass network that adjusts the sampled impulse response of the channel and filter to be minimum phase, without changing any amplitude distortion in the received signal \([32]\). Furthermore, the filter does not change the signal/noise ratio or any of the noise statistics, other than the noise level. The most important single function performed by the filter is that it minimizes the noise variance at its output subject to the constraint that equation 4.2.3 is satisfied. It therefore maximizes the ratio of the magnitude of the first non-zero component of \( E \) (which is fixed at unity) to the output noise variance \([32]\). The various properties just described for the filter \( D \), of course, only hold exactly if the filter has an infinite number of taps. However, by using an appropriately large number of taps (which is usually not excessive) the properties of the filter can be made to approach as closely as required to those described \([32]\). Furthermore, the filter may be adjusted adaptively without undue complexity \([74]\). The ideal adjustment of the filter is assumed throughout this section. The signal at the output of the filter \( D \), at time \( t = iT \), is

\[
\begin{align*}
v_i &= \sum_{h=0}^{g} s_{i-h} e_h + u_i \\
\end{align*}
\]

(4.2.5)

where the real and imaginary parts of the \( \{u_i\} \) are statistically independent Gaussian random variables with zero mean and fixed variance. The detector has exact knowledge of \( E \) and of the possible values of the \( \{s_i\} \). It uses this prior knowledge to determine the detected values \( \{s_i'\} \) of the data symbol \( \{s_i\} \) from its input signal \( \{v_i\} \).
4.2.2 Conventional Nonlinear Equalizer [21,32]

Figure 4.2.2 shows the details of a conventional nonlinear equalizer. The characteristics of filter D are as described previously, and the samples \( \{v_i\} \) are given by

\[
v_i = \sum_{h=0}^{g} s_{i-h} e_h + u_i \tag{4.2.6}\]

A square marked T is here a store that effectively introduces a delay of one sampling interval (T seconds). Thus, as can be seen from Figure 4.2.2, the detected data-symbols \( \{s_i'\} \) are fed to a linear transversal filter FB. Associated with each store in this filter is a multiplier that multiplies the output signal \( s_{i-h} \) from the store by \( e_h \), and the resulting products are then added so that the output signal from the transversal filter, at time \( t=iT \), is:

\[
\Omega_i = \sum_{h=1}^{g} s_{i-h} e_h \tag{4.2.7}\]

This is now subtracted from \( v_i \) to give the equalized signal \( \delta_i = v_i - \Omega_i \) at the detector input. With correct detection of the data symbols

\[
s_{i-1}, s_{i-2}, \ldots, s_{i-g}
\]

\[
\delta_i = v_i - \Omega_i
\]

\[
= \sum_{h=0}^{g} s_{i-h} e_h + u_i - \sum_{h=1}^{g} s_{i-h} e_h
\]

\[
= s_i + \sum_{h=1}^{g} s_{i-h} e_h + u_i - \sum_{h=1}^{g} s_{i-h} e_h
\]

\[
= s_i + u_i \tag{4.2.8}\]
Fig. 4.2.2: Conventional nonlinear equalizer
Regardless of the detected values $s_{i-1}, s_{i-2}, \ldots s_{i-g}$ the threshold detector takes as the detected data symbol $s_i$ the possible value of $s_i$ closest to $\delta_i$.

The arrangement just described is the conventional nonlinear equalizer, which minimizes the mean square error in the equalizer signal, subject to the exact equalization of the channel and the correct detection of $s_{i-1}, s_{i-2}, \ldots s_{i-g}$.

4.2.3 The Viterbi-Algorithm Detector with a Whitened Matched Filter

4.2.3.1 Whitened matched filter

A Viterbi-algorithm detector achieves the best tolerance to additive white Gaussian noise when a "whitened matched filter" is connected immediately ahead of the detector [34]. This whitened matched filter is shown in Figure 4.2.3 and is a combination of:

1. A linear filter matched to the channel.
2. Sampler that samples the output signal from this filter at the signal element rate.
3. A linear noise whitening network implemented as a feedforward transversal filter that causes the noise components at its output to be uncorrelated and therefore statistically independent. In its preferred arrangement, this filter actually removes all zeros of the z-transform of the channel and matched filter that lie outside the unit circle leaving the remaining zeros unchanged.

When the data signal is transmitted over the channel at (or a little over) the Nyquist rate, which is ideally the case where the highest possible transmission is required over a strictly bandlimited channel, instead of using a whitened matched filter at the input to the detector, the received signal may first be filtered through a sharp cut off lowpass filter with a constant amplitude response and linear phase characteristics over passband and cut off frequency at half the signal element rate. The signal is then sampled at regular intervals, once per signal element, so that with additive white Gaussian noise at the input
to the lowpass filter, the noise samples at the output of the sampler are statistically independent Gaussian random variables [47]. The samples of the received (filtered) signal contain all the information in this signal which, in turn, contains the same data signal as received at the input to the lowpass filter. The samples therefore form a set of sufficient statistics for the optimum detection of the received data symbols. This arrangement becomes the equivalent to that with a whitened matched filter, when the arrangement is modified by inserting a linear feedforward transversal filter at the output of the sampler, the filter being adjusted to replace all zeros (roots) of the z-transform of the sampled impulse response of the channel, that lie outside the unit circle in the z-plane, by the complex conjugates of their reciprocals [47]. This filter is identically the same as the filter D in Figure 4.2.1 and is assumed in the model, as can be seen from Section 4.2.1 and Figures 4.2.1 and 4.2.4.

4.2.3.2 The Viterbi-algorithm [73,21,75]

The data transmission system is as shown in Figure 4.2.1. As before, \( v_i \) is given by equation 4.2.5. Let \( V_i \), \( S_i \) and \( U_i \) be the i component row vectors whose jth components are \( v_j \), \( s_j \) and \( u_j \) respectively for \( j = 1,2 \ldots i \). Also let \( X_i \), \( Z_i \) and \( C_i \) be the i component row vector whose jth components are \( x_j \), \( z_j \) and \( c_j \) respectively, for \( j = 1,2 \ldots i \), where \( x_j \) has one of the 16 possible values of \( s_j \) and

\[
V_i = \sum_{h=0}^{q} S_{i-h} U_{h} + V_{i-h}
\]

\[
z_j = \sum_{h=0}^{q} x_{j-h} c_{h}
\]

(4.2.9)

and \( c_j \) is the possible value of \( u_j \) satisfying

\[
v_j = z_j + c_j
\]

(4.2.10)

In the i-dimensional complex vector space containing the vectors \( V_i \), \( Z_i \) and \( C_i \) the square of the unitary distance between the vectors \( V_i \) and \( Z_i \) is:
Fig. 4.2.3: Whitened matched filter

Fig. 4.2.4: Equivalent whitened matched filter
where \(|c_j'|^2\) is the absolute value of \(c_j'\). When all real and imaginary parts of the \(\{u_j\}\) are statistically independent Gaussian random variables with zero mean and fixed variance, the maximum-likelihood vector \(X_i\) is its possible value such that \(|c_i'|^2\) is minimized. When the data-symbols \(\{s_j\}\) are statistically independent and equally likely to have any of their 16 possible values (as is also the case here), \(S_i\) is equally likely to have any of its \(16^g\) different possible values, and the maximum-likelihood vector \(X_i\) is now the possible value of \(S_i\) most likely to be correct, given the received vector \(V_i\). Obviously the smaller the value of \(|c_i'|^2\) associated with any vector \(X_i\), the better is the estimate of \(S_i\) that is given by \(X_i\). \(|c_i'|^2\) is said to be the cost of \(X_i\).

In the Viterbi-algorithm detector the receiver holds in store \(16^g\) vectors \(\{Q_i\}\), where

\[
Q_i = [x_{i-n}, x_{i-n+1}, \ldots, x_i]
\]  

and \(n \geq g\). These \(16^g\) vectors \(\{Q_i\}\) have the \(16^g\) different possible combinations of the last \(g\) components:

\[
x_{i-g+1}, x_{i-g+2}, \ldots, x_i
\]

and each vector is the one with lowest cost for the particular combination of values of its last \(g\) components. Associated with each vector \(Q_i\), is stored the corresponding cost \(|c_i'|^2\). The vector \(Q_i\) with the lowest cost forms the last \(n+1\) component of the maximum-likelihood vector \(X_i\), and the value of the first component \(x_{i-n}\) of this vector \(Q_i\)
is taken as the detected value $s'_i$ of the data symbol $s_{i-n}$. On the receipt of the signal $v_{i+1}$ at the output of the adaptive filter, at time $t = (i+1)T$, each of the $16^g$ stored vectors $\{Q_i\}$ is used to form 16 vectors $\{Q_{i+1}\}$ in which the first $n$ components are given by the last $n$ components of the original vector $Q_i$ and the last component $x_{i+1}$ takes on its 16 different possible values. Each of the resulting $16^g+1$ vectors $\{Q_{i+1}\}$ has the cost

$$|C'_{i+1}|^2 = |C'_i|^2 + |v_{i+1} - \sum_{h=0}^{g} x_{i+1-h} e_h|^2 \quad (4.2.13)$$

which is determined using the appropriate stored value of $C'_i$. For each of $16^g$ possible combinations of values of $x_{i-g+2}, x_{i-g+3}, \ldots, x_{i+1}$ the detector now selects the vector $Q_{i+1}$ with the smallest value of $|C'_{i+1}|^2$. The resulting $16^g$ vectors $\{Q_{i+1}\}$ are then stored together with the associated $|C'_{i+1}|^2$. The vector $Q_{i+1}$ with the lowest cost forms the last $n+1$ component of the maximum-likelihood vector $X_{i+1}$, and the value of the first component of $x_{i-n+1}$ of this vector $Q_{i+1}$ is taken as the detected value $s'_{i-n+1}$ of the data symbol $s_{i-n+1}$. The process continues in this way.

4.2.4 Near-Maximum-Likelihood Detection

The Viterbi-algorithm detector becomes impractical for channels with long sampled impulse responses. The reason for this is the $16^g$ data symbols vectors must be stored and processed over every symbol period. Any practical derivatives must therefore employ a much smaller number of vectors, say $k$, where $k\ll16^g$. A considerable amount of work has been done on such derivatives [42,47,50,49,51,75,53]. In order to clarify the methods of operation of the new detectors, the original detector (system A) [47] is first described and then it is shown how this is modified to give the new detectors (systems B and C). All these near-maximum-likelihood detectors, that will be discussed, are preceded by the equivalent noise whitening matched filter, as previously described. The impulse response of the channel and filter is $E$, and $v_1$ is given by
4.2.4.1 System A [47]

The data transmission system is as shown in Figure 4.2.1. Just prior to the receipt of signal $v_i$ from the transversal filter $D$, the detector holds in store $k$ $n$-component vectors (sequences) $\{Q_{i-1}\}$, where

$$Q_{i-1} = [x_{i-n}, x_{i-n+1}, \ldots, x_{i-1}] \quad (4.2.14)$$

and $x_{i-h}$ takes on a possible value of $s_{i-h}$ for $h=1,2,\ldots,n$, the $k$ vectors $\{Q_{i-1}\}$ being all different. Each vector $Q_{i-1}$ represents a possible sequence of values of the data symbols $s_{i-n}, s_{i-n+1}, \ldots, s_{i-1}$. On the receipt of the signal $v_i$, at the output of the filter $D$ at time $t=iT$, each stored vector $Q_{i-1}$ is expanded into 16 $(n+1)$-component vectors $\{P_i\}$, where 16 is, of course, the number of possible values of data symbol $s_i$, and

$$P_i = [x_{i-n}, \ldots, x_i] \quad (4.2.15)$$

In each group of 16 vectors $\{P_i\}$, derived from any one vector $Q_{i-1}$, the first $n$ components are as in the original $Q_{i-1}$ and the last component $x_i$ takes on its 16 different possible values. Each of the resulting 16 vectors $\{P_i\}$ has the cost

$$|C_i^2| = |C_{i-1}^2| + |v_i - \sum_{h=0}^{q} x_{i-h} e_h|^2 \quad (4.2.16)$$

where $|C_{i-1}^2|$ is the cost of the vector $Q_{i-1}$ from which $P_i$ was derived, such that
$$|c_{i-1}|^2 = \sum_{j=0}^{i-1} |v_j - \sum_{h=0}^{q} x_{j-h} e_h|^2$$ (4.2.17)

and $x_i = 0$ for $i < 0$ and $i > b$. $|v|$ is the absolute value (modulus) of $v$. The detected value $s_{i-n}'$ of the data-symbol $s_{i-n}$ is now taken as the value of $x_{i-n}$ in the vector $P_i$ with smallest cost. Any vector $P_i$ whose first component $x_{i-n}$ differs from $s_{i-n}'$ is then discarded, and from the remaining vectors $\{P_i\}$ (including that from which $s_{i-n}$ was detected) are selected the $k$ vectors having the smallest costs $\{|C_i'|^2\}$. The first component of each of the $k$ selected vectors $\{P_i\}$ is now omitted (without changing its cost) to give the corresponding vectors $Q_i$ which are then stored together with the associated costs $\{|C_i'|^2\}$, ready for the next detection process. The discarding of the vectors $\{P_i\}$ just mentioned, is a convenient method of ensuring that the $k$ stored vectors $Q_i$, are always different, provided only that they were different at the first detection process, which can easily be arranged [42]. So long as the amplitude distortion in the received signal is not too severe, and so long as the number of stored vectors, $k$, is not much less than 16, the detection process just described has a tolerance to additive white Gaussian noise that is within about 1 dB of that of the corresponding Viterbi-algorithm detector [42]. The main weakness of the detection process is that it involves $k$ searches through $16k$ costs. When $16k$ is large, this requires an excessive number of operations. A promising technique for reducing the number of operations under these conditions is that employed in pseudobinary and pseudoquaternary systems [47,50,51,75,53]. The detector here considers only the two or four of the 16 expanded vectors $\{P_i\}$, originating from any one $Q_{i-1}$ that have the smallest costs, the remaining vectors not being used. This is achieved by means of a simple threshold-level comparison technique that does not itself involve the evaluation of any costs [47,75]. The complexity of the detection process is therefore reduced effectively to that for the corresponding binary or quaternary signal.
4.2.4.2 System B [53]

This system is a development of system A. Two different versions of system B, the pseudobinary and the pseudouaternary arrangements, have been studied and the pseudobinary version will be considered first. Just prior to the receipt of the signal $v_i$, in the pseudobinary version of the system, the detector holds in store $k$ vectors $\{Q_{i-1}\}$ together with their costs $\{|C_{i-1}'|^2\}$, as before. However, $k$ must now be even and the vectors $\{Q_{i-1}\}$ are arranged in pairs, where the two vectors $\{Q_{i-1}\}$ in any one pair differ only in the values of the last component $x_{i-1}$. For the given value of $x_{i-n}$, $x_{i-n+1}$, ..., $x_{i-2}$ in a pair of vectors $\{Q_{i-1}\}$, the value of $x_{i-1}$ in the first of the pair is the one of its possible values giving the smallest cost $\{|C_{i-1}'|^2\}$, and the value of $x_{i-1}$ in the second of the pair is the one of its possible values giving the second smallest cost. On the receipt of $v_i$, each vector $Q_{i-1}$ is expanded into the corresponding vector $P_i$ having the smallest cost. The selection of $P_i$ is achieved through the use of simple threshold-level comparisons and does not involve the computation of any costs [50,75]. The detector next evaluates the costs $\{|C_i'|^2\}$ of the $k$ vectors $\{P_i\}$ and selects the vector with the smallest cost, taking the value of its first component $x_{i-n}$ as the detected value $s_{i-n}'$ of the data-symbol $s_{i-n}'$. All vectors $\{P_i\}$ for which $x_{i-n} \neq s_{i-n}'$ are now discarded, and the first component of all remaining vectors $\{P_i\}$ are omitted (without changing their costs), to give the corresponding $n$-component vectors $\{Q_i\}$.

In addition to the vector $Q_i$ with the smallest cost, which has already been selected, the detector now selects, from the remaining vectors $\{Q_i\}$, the 0.5$k$-1 vectors with smallest costs, to give a total of 0.5$k$ selected vectors $\{Q_i\}$. To each of these vectors is then added an additional vector $Q_i$, whose first $n-1$ components are as in the original vector $Q_i$, and whose last component $x_i$ takes on its possible value for which the cost of $Q_i$ has its second smallest value. The value of $x_i$ giving the smallest cost is, of course, that in the original vector $Q_i$. The simple algorithm previously mentioned [50,75] is employed here, so that no actual costs need be evaluated. The detector now holds in store $k$ vectors $\{Q_i\}$, in the form of 0.5$k$ pairs of vectors, the two vectors of
any pair having the same values of $x_{i-n+1}$, $x_{i-n+2}$, ..., $x_{i-1}$. The costs $|C_i|^2$ of the 0.5$k$ vectors $Q_i$ not yet determined are next evaluated, and associated with each vector $Q_i$ is then stored the corresponding cost.

The pseudoquaternary version of system B is basically similar to the pseudobinary version. Just prior to the receipt of $v_i$, the detector holds in store $k$ vectors $Q_{i-1}$ together with their costs $|C_{i-1}|^2$, as before. However $k$ is now a multiple of four and the vectors $Q_{i-1}$ are arranged in groups of four, where the four vectors in any one group have the same values of $x_{i-n}$, $x_{i-n+1}$ ..., $x_{i-2}$ and four different values of $x_{i-1}$. The latter four values are determined by simple threshold-level comparisons and without requiring the evaluation of any costs, and they give the four of the 16 possible vectors with the smallest costs [47,53].

On the receipt of $v_i$, the detection process proceeds exactly as for the pseudobinary version of system B, until the detector has $k$ vectors $Q_i$ together with the associated $|C_i|^2$, every vector $Q_i$ has been derived from the corresponding vector $P_i$ by omitting the first component. In addition to the vector $Q_i$ with the smallest cost, which has already been selected, the detector now selects from the remaining vectors $Q_i$ the 0.25$k$-1 vectors with the smallest costs, to give a total of 0.25$k$ selected vectors $Q_i$. Each of these vectors is then expanded into four vectors $Q_i$, through the addition of three vectors to the original. The four vectors in any one group have the same values of $x_{i-n}$, $x_{i-n+1}$ ..., $x_{i-1}$ as the original $Q_i$ and four different values of $x_{i}$. The latter four values generally give the four smallest costs for the corresponding $Q_i$ and are determined in the same way as the four values of $x_{i-1}$ in any one group of four vectors $Q_i$ at the start of the detection process, as before. The selection process are simple threshold-level comparisons and do not involve the evaluation of any costs [47,53]. The costs $|C_i|^2$ of the 0.25$k$ vectors $Q_i$ not yet determined are now evaluated, and the detector then stores $k$ vectors $Q_i$ together with the associated costs, ready for the next detection process.
The particular virtue of system B is that, for a given number of stored vectors, it involves fewer operations per detection process than does the corresponding arrangement of system A. Thus, with \( k \) stored vectors, the pseudobinary version of system B evaluates \( 1.5k \) costs and carries out \( 0.5k \) searches through \( k \) costs, per detection process, and the pseudoquaternary version evaluates \( 1.75k \) costs and carries out \( 0.25k \) searches through \( k \) costs. On the other hand, the pseudobinary version of system A evaluates \( 2k \) costs and carries out \( k \) searches through \( 2k \) costs, per detection process, and the pseudoquaternary version evaluates \( 4k \) costs and carries out \( k \) searches through \( 4k \) costs. The important property of system B is that, for the pseudobinary or pseudoquaternary version and for a given number of stored vectors, it often achieves a performance almost as good as that of the corresponding version of system A, but with many fewer operations per detection process.

4.2.4.3 System C [76]

System B may be modified to give now the following system, known here as system C. The number of vectors \( \{Q_i\} \) derived from any one vector \( Q_{i-1} \) is reduced in moving from the vector \( Q_{i-1} \) with the smallest cost to that with the largest cost. In the example to be studied here, there are 16 stored vectors \( \{Q_{i-1}\} \). Each of these is first expanded to give the corresponding vector \( P_i \), and hence the corresponding vector \( Q_i \), with the smallest cost (ignoring for convenience, any discarded vectors). To each of the two vectors \( \{Q_i\} \), with the smallest and second smallest costs of the set of 16, are added three vectors \( \{Q_{i}\} \) differing only in the last component and with the smallest costs. Then to each of the two vectors \( \{Q_i\} \) with the third and fourth smallest costs of the original set of 16, is added the vector \( Q_{i} \) differing only in the last component and with the smallest cost. No vectors are added to the four vectors \( \{Q_{i}\} \) having the fifth to eighth smallest costs of the original set of 16, and the remaining eight of this set of vectors are discarded. If any vectors \( \{P_i\} \) are discarded, following the detection of \( s_{i-n'} \), this correspondingly reduces the number of the remaining vectors \( \{Q_{i}\} \) that are subsequently discarded, but does not otherwise change the detection.
process. There are now 16 vectors \( Q_i \), together with their costs, ready for the next detection process. An appropriate arrangement of the technique just described with, perhaps, different numbers of vectors expanded four ways, two ways and one ways, may often lead to a more cost-effective detector than the best pseudobinary or pseudoquaternary version of system B. When \( k=1 \), so that there is only one stored vector, each of the near-maximum-likelihood detectors described previously degenerates into the conventional nonlinear (decision feedback) equalizer that minimizes the mean square error in the equalized signal, subject to the accurate equalization of the channel.

4.3 DETECTORS FOR TIME VARYING CHANNELS (HF LINKS)

As is evident from Chapter 3, the same model of data transmission system shown in Figure 4.2.1 can be used here. However, now the transmission path is the HF link, which is time varying.

Also, the sampled impulse response of the channel \( Y_i \) (equation 3.3.8) together with sampled impulse response \( E_i \) of the channel and adaptive filter \( D \) now vary with time. As before, the receiver is assumed to have an accurate estimate of \( E_i \), for each received sample.

The adaptive filter \( D \) has a very large number of taps but now it is constrained to operate only on those roots (zeros) of the z-transform of the sampled impulse response of the linear baseband channel (Figure 4.2.1) that have absolute values greater than \( d \) [74]. When \( d=1 \), the filter is adjusted to its ideal setting, as previously assumed. For any greater value of \( d \), such roots as are processed by the filter are handled as before, the remaining roots being left unchanged. The filter is still an all pass network, but \( E_i \) (equation 4.2.3) is no longer minimum phase [76].

The other difference from the time invariant model is that, from Chapter 3, the receiver filter \( B \) now has not got the ideal characteristics described in Section 4.2.1, so the noise samples at the output of this filter are slightly correlated, but tests have shown that the
correlation is not sufficient to affect noticeably the performance of the systems studied [89]. The same detectors described in Sections 4.2.2, 4.2.3 and 4.2.4 may be used here, with $E$ replaced by $E_i$, because it is a time varying channel. At time $t=IT$

$$E_i = [e_{i,1}, e_{i,2} \ldots e_{i,g}]$$

(4.3.1)

$E_i$ is the sampled impulse response of the channel and filter, and $v_i$ will be

$$v_i = \sum_{h=0}^{g} s_{i-h} e_{i,h} + u_i$$

(4.3.2)

4.4 RESULTS OF COMPUTER SIMULATION TESTS

4.4.1 Back-to-Back Tests

These tests were run in order to discover the performance of the serial modem when no distortion is introduced by the channel. In other words, the channel has no fading or multi-path and the transmitter and receiver are connected 'back-to-back' as shown in Figure 4.4.1.

White Gaussian noise is added at the receiver input and a large number of symbols (or bits) are then transmitted and received and the transmitted data stream ('data in' of Figure 4.4.1) is compared with the received data stream ('data out') to check for errors.

The sampled impulse response of the transmitter and receiver filters used in the tests are those given in Table 3.3.1. In these tests system C with 16 stored vectors is used as the near-maximum-likelihood detector. The delay in detection, $n$, is arranged to be 32. The reason for using system C will become clear later on. Furthermore, it is assumed that the adjustment of the adaptive filter and channel estimation are carried out perfectly. The back-to-back test is carried
Fig. 4.4.1: Block diagram of the arrangement for "back-to-back" error rate tests
out for different levels of additive Gaussian noise so that a plot of error rate against signal to noise ratio can be made. The error rate is defined simply as:

\[
\text{error rate} = \frac{\text{total number of symbols (or bits) received in error}}{\text{symbol (or bit)}} \times \frac{\text{total number of symbols (or bits) transmitted}}{\text{error rate}}
\] (4.4.1)

The signal to noise ratio is defined as:

\[
\psi = 10 \log_{10} \left( \frac{E_{av}}{\frac{1}{2}N_0} \right)
\] (4.4.2)

where \(E_{av}\) is the average energy per bit of information at the input of the receiver. \(\frac{1}{2}N_0\) is the two-sided power spectral density of the additive Gaussian noise at the input of the receiver. This signal to noise ratio definition has the advantage that the performance of different data transmission systems can be directly compared if this signal to noise ratio measure is employed.

Finally, the results of these tests are plotted in Figure 4.4.2. Note that around 1.5 x 10^6 data symbols are being used in plotting this curve.

4.4.2 Tests for the Detection Systems Operating Over an HF Radio Channel

Computer simulation tests have been carried out on the detection processes described in previous sections in the 9600 bit/s, HF serial data transmission system. The data transmission system is exactly the same as that given in Figure 4.4.1, except that the signal at the output of the transmitter filter now passes through the HF link before it reaches the receiver. The model of the HF links used in the tests has three independent Raleigh fading skywaves with different delays in transmission as shown in Figure 4.4.3. The six waveforms \(\{q_i(t)\}\) here...
Fig. 4.4.2: Error rate versus S/N for the simulated modem over a perfect channel
Fig. 4.4.3: Model of skywaves in HF radio link
are sample functions of stationary zero-mean real-valued Gaussian random processes, each with a low pass spectral shaping that is approximately Gaussian, and each with a root-mean-square bandwidth of 1 Hz, giving a frequency spread of 2 Hz in the output data signal. The six random processes involved in the three skywaves are statistically independent and each has the same variance, so that the three output signals given by the three skywaves have the same mean-square value and the same frequency spread of 2 Hz.

In order to obtain the most accurate comparison possible between the various detectors studied, the same fading sequence, that is the same sequence of values of the sampled impulse-response of the linear baseband channel \( \{Y_i\} \), was used over the given number of 40,000 received samples \( \{r_i\} \) that were employed in each test. The results of the computer simulation tests are shown in Figures 4.4.4-4.4.7. The signal to noise ratio and error rate are as defined by equations 4.4.1 and 4.4.2 respectively.

The near-maximum-likelihood detectors introduce a delay in detection of \( n \) sampling intervals. To avoid any noticeable degradation in performance due to an inadequate delay in detection \( n \) has been set equal to 32 (\( n > g \)). The letter A, B or C against any curve in Figures 4.4.4-4.4.7 represent the system tested. The numeral 2 or 4 preceding the letter indicates a pseudobinary or pseudoquaternary system, respectively, and the numeral following the letter indicates the number of stored vectors \( \{Q_i\} \). Thus, for example, 4B16 is the pseudoquaternary system B with 16 stored vectors. System C is, of course, neither pseudobinary nor pseudoquaternary. The curves marked "upper bound" on such of the four diagrams is a measured upper bound to the performance of a true maximum-likelihood detector, such as a Viterbi-algorithm detector, for the given channel. It shows the performance obtained when the signal elements are transmitted at intervals of \( (g+1)T \) seconds, so that the received elements do not overlap in time. Ideal matched filter detection is now carried out separately on each received signal element, at the output of the filter D, assuming exact prior knowledge of \( E_i \) and the appropriate adjustment of the filter D. Since there is no intersymbol interference
in any detection process here, the latter minimizes the probability of error in the detection of each received signal element [3]. The information rate achieved with the arrangement just described is only \(1/(g+1)\) of that continuous transmission, previously described, and its tolerance to noise is likely to be somewhat better than that obtainable with an ideal Viterbi-algorithm detector, operating on a continuous signal [76]. The latter is, however, far too complex to be tested by computer simulations.

Since the receiver has prior knowledge of \(E_i\), it is sufficient to measure the error rate in the \(s_{i}'\), thus avoiding the need for specifying the bit-to-symbol coding at the transmitter and the symbol-to-bit decoding at the receiver. An average of around \(1.5 \times 10^6\) data-symbol being used in plotting each curve.

Before comparing the performance of the different systems, it is helpful first to consider the relative complexity of the near-maximum-likelihood detectors tested. An approximate assessment of these is given in Table 4.4.1. The evaluation of a cost is here a considerably more complex process than is one of the separate operations involved in a search through the costs. Thus the systems, listed in order of complexity and starting with the most complex, are: 4A16, C16, 4B16 and 2B8. System 4A16 is very much more complex than system C16, the latter being only moderately more complex than system 4B16.

It can be seen from Figures 4.4.4-4.4.7 that, at an error rate of 1 in \(10^3\) in the \(\{s_{i}'\}\) the better of the near-maximum-likelihood detectors achieve an advantage of about 3-4 dB in tolerance to additive white Gaussian noise, over the conventional nonlinear equalizer. Furthermore, system C16 has a performance very close to that of system 4A16 and is the most cost-effective of the systems tested, for applications where a good performance is important. However, where a loss in tolerance to noise of about 1 dB can be tolerated, system 2B8 may be preferred to system C16, because of its greatly reduced complexity.
The reason for testing the various detectors with different values of $d$ (Section 4.3), is not only to determine their tolerance to departures in $E_i$ from its ideal minimum-phase condition, but also to obtain a feel for the best compromise between the complexity of the adaptive filter $D$ and that of the detectors. The greater the value of $d$, the fewer taps are needed by the adaptive filter and the fewer operations, per received sample $r_i$, are needed to hold it correctly adjusted [74]. Thus increasing $d$ reduces the complexity of the adaptive filter. However, it is clear from Figures 4.4.4-4.4.7 that the detectors are seriously affected, even by a small increase in the value of $d$. Thus, the smallest practically realisable value of $d$, which is about 1.05, should be used where possible. In practice, $d$ cannot be reduced to 1.0, since an infinite number of taps are now required for the accurate adjustment of the filter $D$. 
### TABLE 4.4.1: Relative complexities of the different detectors tested

<table>
<thead>
<tr>
<th></th>
<th>4A16</th>
<th>4B16</th>
<th>2B8</th>
<th>C16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of costs evaluated per data-symbol</td>
<td>64</td>
<td>28</td>
<td>12</td>
<td>24</td>
</tr>
<tr>
<td>Number of searches through the costs, per data-symbols</td>
<td>16</td>
<td>4</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>Number of costs checked per search</td>
<td>64</td>
<td>16</td>
<td>8</td>
<td>16</td>
</tr>
<tr>
<td>Number of separate operations involved in the searches</td>
<td>1024</td>
<td>64</td>
<td>32</td>
<td>128</td>
</tr>
</tbody>
</table>
Fig. 4.4.4: Performance of systems over HF radio link when $d = 1.0$
Fig. 4.4.5: Performance of systems over HF radio link when $d = 1.05$
Fig. 4.4.6: Performance of systems over HF radio link when \( d = 1.1 \)
Fig. 4.4.7: Performance of systems over HF radio link when $d = 1.2$
CHAPTER 5

HF CHANNEL ESTIMATION

5.1 INTRODUCTION

The results of simulation tests of near-maximum-likelihood detectors in the previous chapter represent upper bound performances because they assume perfect adaptive filtering ahead of the detector and perfect estimation of the channel sampled impulse response. HF channel estimation is considered in this chapter. A model of the data transmission system under consideration is described briefly in Section 5.2. Section 5.3 describes the methods available for HF channel estimation. The more recently developed HF channel estimator (known as the improved channel estimator) is discussed in detail in Section 5.4. This estimator is based on the feedforward estimator with prediction, but now the knowledge of the basic structure of the HF channel is used which consequently gives it an improved performance. A fast channel estimator is described in Section 5.5. This estimator uses a self orthogonal data sequence to estimate the channel. These estimates are then used for the starting up procedure of the improved estimator at the beginning of data transmission. Section 5.6 describes the starting up methods in the improved estimator. Finally, in Section 5.7, the improved estimator is made adaptive in the sense that it operates correctly, irrespective of changes in the structure of the HF link.

5.2 MODEL OF THE DATA TRANSMISSION SYSTEM

The model of the QAM data transmission system that is used in the tests is basically the same as that described in Chapter 3 and is shown in Figure 5.2.1.

It is a synchronous serial system involving a model of an HF radio link as the transmission path. As shown in Figure 5.2.1, at the receiver the channel estimator operates on two inputs, namely the received signal
Fig. 5.2.1: Model of data transmission system
samples \(\{r_i\}\) and the detected data symbols \(\{s_i'\}\). The received samples \(\{r_i\}\) are complex valued and such that, at time \(t=iT\),

\[
 r_i = \sum_{h=0}^{g} s_{i-h} Y_i, h + w_i
\]  

(5.2.1)

The received samples \(\{r_i\}\) are complex valued and such that, at time \(t=iT\),

\[
 r_i = \sum_{h=0}^{g} s_{i-h} Y_i, h + w_i
\]

The sequence of complex values given by the \((g+1)\) component row vector

\[
 Y_i = [Y_i,0 \ Y_i,1 \ \cdots \ Y_i,g]
\]  

(5.2.2)

is taken to be the "sampled impulse response" at time \(t=iT\) of the linear baseband channel. The detector in Figure 5.2.1 is a near-maximum-likelihood detector and the delay in detection is \((n-1)\) sampling intervals. Thus, following the receipt of \(r_{i+n-1}\) at time \(t=(i+n-1)T\), the detected data symbol \(s_i'\) is determined. Since we are here concerned with the operation of the channel estimator and not the detector, the correct detection of all data symbols is assumed, so that \(s_i' = s_i\) for all \(i\). Tests have indicated that the performance of the estimator is only seriously affected by errors in the \(\{s_i'\}\) at the higher error rates [66,8,77].

The channel estimator here is a data aided device, and the best estimate that it can make, on receipt of \(r_{i+n'\nu}\) is an estimate of the impulse response at time \(t=iT\), that is \(Y_i\). This is because just prior to receipt of \(r_{i+n'\nu}\) the most recently detected symbol is \(s_i'\) and from this and earlier detected symbols, the channel estimator produces the estimate \(Y_i'\). The data detector requires knowledge of the impulse response \(Y_{i+n}\) on receipt of sample \(r_{i+n'\nu}\) so the channel estimator must somehow form a prediction \(Y_{i+n,1}'\) of \(Y_{i+n}\) from its more recent estimates \(Y_i', Y_i'-1', \ldots\) of the channel impulse response. Clearly, for a time invariant channel \(Y_i'\) will be a very good estimate of \(Y_{i+n}\) but when the channel impulse response is varying with time then \(Y_{i+n}\) can be quite different from \(Y_i'\).
5.3 METHODS OF HF CHANNEL ESTIMATION
5.3.1 Feedforward Transversal-Filter Estimator

This estimator has been developed by Magee and Proakis [78]. The estimator has \( g+1 \) taps which is equal to the number of components in the sampled impulse response of the channel and these tap gains are adjusted in such a way to minimize the mean-square error between the actual received sample \( r_i \) and its estimate \( r_i' \) at the output of the estimator for time invariant channels. Under ideal conditions, the resulting values of the tap gains are the components of the sampled impulse response of the channel.

A block diagram of the complex feedforward estimator [66] is given in Figure 5.3.1. The feedforward estimator operates as follows:

Each box labelled \( T \) in Figure 5.3.1 is a store that holds the corresponding detected data-symbols \( s_{i-h} \). Each time the stores are triggered, the stored values are shifted one place to the right. On the receipt of \( r_{i+n} \) and before the detection of \( s_{i+1} \), the estimator is fed with the received sample \( r_i \) and the detected data symbol \( s_i \). If \( Y_{i-1} \) is the previous stored estimate of \( Y_i \), then an estimate \( r_i' \) of \( r_i \) at the output of the estimator is given by

\[
r_i' = \sum_{h=0}^{g} s_{i-h}Y_{i-1,h} \quad (5.3.1)
\]

the error in this estimate which is

\[
e_i' = r_i - r_i' \quad (5.3.2)
\]

is then scaled by a small positive quantity \( \Delta \) resulting in the signal \( \Delta e_i' \). Each signal \( (s_{i-h})^* \) for \( h=0,1,...,g \) is multiplied by \( \Delta e_i' \) and the products are added to the corresponding components of the previous estimate \( Y_{i-1}' \), giving the new stored estimate \( Y_i' \), where the \((h+1)\)th component of \( Y_i' \) is given by
Fig. 5.3.1: Feedforward transversal-filter estimator
Equation 5.3.3 is usually known as the stochastic gradient algorithm. The factor $\Delta$ in equation 5.3.3 is usually known as the step size of the estimator and it need not necessarily be a constant. It is desirable to make $\Delta$ as small as possible so that the additive noise will have a small effect on $Y_i$ [77]. However this results in the estimator having a slower rate of response to change in $Y$. Clearly, the feedforward estimator can be implemented easily and it is also able to track slow variations in the channel response. In this method there is no prediction, so it uses $Y_i$ as an estimate of $Y_{i+n}$ in the detector.

Hence

$$Y_{i+n,i} = Y_i$$ (5.3.4)

5.3.2 Feedforward Estimator with Prediction

In this method, instead of using the components $\{y_{i-1,h}\}$ of the estimate $Y_{i-1}$ to form $Y_i$ (in equation 5.3.3), the corresponding components of the one-step prediction $Y_{i,i-1}$ are used [66] where

$$Y_{i,i-1} = [y_{i,i-1,0}, y_{i,i-1,1}, \ldots, y_{i,i-1,g}]$$ (5.3.5)

$Y_{i,i-1}$ is derived from the past estimates $Y_{i-1}', Y_{i-2} \ldots$ using degree 1 least squares fading memory prediction [66]. Thus, the estimate of $r_i$ is now:

$$r_i' = \sum_{h=0}^{g} s_{i-h} Y_{i,i-1,h}$$ (5.3.6)

and the $(h+1)$th components of the updated estimate, $Y_i'$ becomes

$$Y_{i,h} = Y_{i-1,h} + \Delta e_i' s_{i-h}^*$$ (5.3.3)
where $e'_i$ is given by equations 5.3.2 and 5.3.6. The process otherwise proceeds as described for the feedforward estimator, the difference between the two estimators being in equations 5.3.1, 5.3.3, 5.3.6 and 5.3.7. With a time varying channel, $Y_i, i-1$ should be closer to $Y_i$ than $Y_{i-1}$ leading to a more accurate value of the updated estimate $Y'_i$. Furthermore, this estimator, after finding the estimate $Y'_i$ (equation 5.3.7), forms two predictions, first the one-step prediction $Y_{i+1, i}$ of $Y_{i+1}$ for use in the estimator in place of $Y'_i$, when estimating $Y_{i+1}$, and secondly the n-step prediction $Y_{i+n, i}$ of $Y_{i+n}$ for use in the detector, when detecting $s_{i+j}$ [66]. This feedforward estimator with prediction has been shown [66] to be capable of achieving a much better performance than the feedforward estimator (Section 5.3.1) over HF links.

5.3.3 The Kalman Estimator

The development of the Kalman filter as an HF channel estimator is studied in reference 59. It has been found that the essential weakness of the conventional Kalman estimator, for applications over HF radio links, stems from the fact that the conventional Kalman estimator is based on an incorrect model of the channel. Any attempt at modifying the channel model (on which the estimator is based) with a view to improving the performance of the Kalman estimator, appears likely to result in a considerable increase in the complexity of an already complex system. Furthermore, it has been shown that a performance, similar to that of a conventional Kalman estimator can be achieved by means of a very much simpler system [59,79].

The Kalman filter does not therefore appear to be very suitable as a channel estimator for HF radio links [59,79].

These results have been confirmed by carrying out an independent study of the Kalman estimator (see Table 5.3.1). Clearly, from this table the best performance is given by the feedforward estimator with prediction,
TABLE 5.3.1: Comparison of the mean square error in the estimates of the channel sampled impulse response (ε'')* given by the feedforward, Kalman and feedforward with prediction estimators for a 3 skywave HF channel 
τ₁ = 1 msec, τ₂ = 3 msec, 2 Hz frequency spread

<table>
<thead>
<tr>
<th>S/N** dB</th>
<th>ε'' dB Feedforward Estimator</th>
<th>ε'' dB Kalman Estimator</th>
<th>ε'' dB Feedforward with Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>30</td>
<td>-23.87</td>
<td>-25.95</td>
<td>-28.99</td>
</tr>
<tr>
<td>60</td>
<td>-25.25</td>
<td>-28.17</td>
<td>-37.34</td>
</tr>
</tbody>
</table>

* ε'' = 10 Log₁₀(\(\frac{1}{48000}\sum_{i=4000}^{52000}|y_i - y_i'|^2\))

** S/N as defined by equation 4.4.2.
then by the Kalman estimator, followed by the feedforward estimator. The developed Kalman estimator algorithm used in the study is given in Appendix D.

5.3.4 Improved Channel Estimator

This estimator is based on the feedforward estimator (Section 5.3.1) but has been modified to include a predictor and also utilizes some prior knowledge of the basic structure of the channel in the estimation process [80]. Comparison of the results of this estimator with that of the feedforward estimator with prediction, shows a considerable superiority for the improved estimator [80].

5.4 IMPROVED CHANNEL ESTIMATOR
5.4.1 Improved Channel Estimator with a Two-Skywave HF Channel [80]

Let us assume that the HF radio signal is transmitted via two independent skywaves. In this case the sampled impulse response of the linear baseband channel at time \( t = iT \) can be taken to be

\[
Y_i = \mu_i L + \lambda_i M
\]  

where \( L, M \) are fixed \((g+1)\)-component vectors and the complex-valued scalar parameters \( \mu_i \) and \( \lambda_i \) vary with time \( i \). \( \mu_i L \) and \( \lambda_i M \) are in fact the sampled impulse response corresponding to the first and second skywaves.

Clearly, from equation 5.4.1, if the receiver can determine the time invariant vectors \( L \) and \( M \), then an estimate of \( Y_i \) is obtained by estimating only the variables \( \mu_i \) and \( \lambda_i \). However, the vectors \( L \) and \( M \) are not easily determined and they are not normally orthogonal or even related in any very simple manner. But it is evident from equation 5.4.1 that the vector \( Y_i \) must lie in the subspace (plane) spanned by \( L \) and \( M \), in the \((g+1)\)-dimensional unitary vector space containing all \((g+1)\)-
component vectors over the complex field. Since L and M are fixed, the subspace spanned by these vectors is also fixed, so that the receiver need only estimate the subspace. This is achieved by estimating two orthonormal vectors A and B, which also span the subspace such that

\[ Y_i = a_iA + b_iB \quad (5.4.2) \]

\( a_i \) and \( b_i \) are complex-valued scalar parameters.

Now, just prior to the receipt of \( r_{i+n} \) at time \( t=(i+n)T \), the receiver has formed the prediction

\[ Y'_{i,i-1} = [Y'_{i,i-1,0} \ Y'_{i,i-1,1} \ldots \ Y'_{i,i-1,g}] \quad (5.4.3) \]

of the vector \( Y_i \), in a manner to be described later. The receiver also holds the estimates of A and B, which are the vectors

\[ A_i = [a_{i,0} \ a_{i,1} \ldots \ldots \ a_{i,g}] \quad (5.4.4) \]

and

\[ B_i = [b_{i,0} \ b_{i,1} \ldots \ldots \ b_{i,g}] \quad (5.4.5) \]

respectively. \( A_i \) and \( B_i \) are orthonormal vectors that lie close to the two dimensional subspace containing \( Y_i \).

On the receipt of \( r_{i+n} \) the receiver forms an estimate of \( r_i \) which is

\[ r'_i = \sum_{h=0}^{g} s_{i-h} Y'_{i,i-1,h} \quad (5.4.6) \]
is then subtracted from $r_i$ to give the error signal

$$e_i' = r_i - r_i'$$

Thus, using equation 5.3.7, the $(h+1)^{th}$ component of $Y_i'$ is:

$$Y_i', h = Y_{i, i-1}, h + \Delta e_i' (s_i-h)^*$$

for $h = 0, 1, \ldots, g$ where $\Delta$ is a small positive real quantity.

$Y_i'$ will not usually lie in the two dimensional subspace (plane) spanned by $A_i$ and $B_i$ (because of the noise). The estimator therefore forms the $(g+1)$ component vector $F_i$ that lies in the subspace and is at the minimum unitary distance from $Y_i'$. $F_i$ is usually a better estimate of $Y_i'$ than $Y_i'$. Hence, using the projection theorem, it can be shown that [80]:

$$Y_i = \alpha_i A_i + \beta_i B_i$$

where

$$\alpha_i = Y_i' A_i^T$$

$$\beta_i = Y_i' B_i^T$$

since $L$ and $M$ may in fact vary slowly with time, so also may the subspace. This means that, for satisfactory operation, the subspace spanned by $A_i$ and $B_i$ should be adjusted to track the received signal.

$Y_i'$ must therefore be used to adjust the subspace and hence determine the vectors $A_{i+1}$ and $B_{i+1}$ that span a subspace slightly closer to $Y_i'$. To achieve this the estimator forms the vector
\[ H_i = Y_i' - F_i \quad (5.4.12) \]

to give

\[ A_{i+1} = A_i + \eta \alpha_i^* H_i \quad (5.4.13) \]

\[ B_{i+1} = B_i + \eta \beta_i^* H_i \quad (5.4.14) \]

where \( \eta \) is a small positive real constant. The \( A_{i+1} \) and \( B_{i+1} \) vectors are then orthonormalized [81,82]. The receiver next determines the predictions \( \alpha_{i+1,i} \) and \( \alpha_{i+n,i} \) of \( \alpha_{i+1} \) and \( \alpha_{i+n} \) respectively using degree-1 least square fading memory prediction [66] as follows

\[ \varepsilon_{\alpha,i} = \alpha_i - \alpha_{i,i-1} \quad (5.4.15) \]

\[ \alpha_{i+1,i} = \alpha_{i,i-1} + (1-\theta)^2 \varepsilon_{\alpha,i} \quad (5.4.16) \]

\[ \alpha_{i+1,i} = \alpha_{i,i-1} + (1-\theta)^2 \varepsilon_{\alpha,i} \quad (5.4.17) \]

and

\[ \alpha_{i+n,i} = \alpha_{i+1,i} + (n-1) \alpha_{i+1,i} \quad (5.4.18) \]

\( \theta \) is a real constant between 0 and 1. By a similar method, \( \beta_{i+1,i} \) and \( \beta_{i+n,i} \) can be found. The receiver now forms

\[ Y_{i+1,i} = \alpha_{i+1,i} A_{i+1} + \beta_{i+1,i} B_{i+1} \quad (5.4.19) \]

which is the one-step prediction used in the feedforward estimator to determine \( Y'_{i+1} \). The n-step prediction of \( Y_{i+n} \) used in the detector is given by
\[ Y_{i+n,i} = \alpha_{i+n,i}A_{i+1} + \beta_{i+n,i}B_{i+1} \quad (5.4.20) \]

### 5.4.2 Improved Channel Estimator with One or Three Skywaves HF Channel

For a three-skywave HF channel the sampled impulse response of the linear baseband channel (equation 5.4.1) becomes [80]

\[ Y_i = v_iL + \lambda_iM + \rho_iN \quad (5.4.21) \]

where \( L, M \) and \( N \) are fixed \((g+1)\)-component vectors, and \( v_i, \lambda_i \) and \( \rho_i \) are complex-valued scalars. Each of the terms in equation 5.4.21 corresponds to the sampled impulse response of the corresponding skywave, so that, \( Y_i \) has the value \( v_iL, \lambda_iM \) or \( \rho_iN \) if only the first, second or third skywave is present, respectively.

It is evident from equation 5.4.21 that the vector \( Y_i \) must lie in a three dimensional subspace spanned by \( L, M \) and \( N \), in the \( g+1 \)-dimensional unitary vector space.

Since \( L, M \) and \( N \) are fixed, the subspace spanned by these vectors is also fixed, so that the receiver need only estimate the subspace. This is achieved by estimating the three orthonormal vectors \( A, B \) and \( C \) which also span the three dimensional subspace such that:

\[ Y_i = a_iA + b_iB + c_iC \quad (5.4.22) \]

The scalars \( a_i, b_i \) and \( c_i \) and the components of \( A, B \) and \( C \) are all complex-valued.
For one skywave, the sampled impulse response of the linear baseband channel can be written as [80]:

\[ Y_i = a_i A \]  \hspace{1cm} (5.4.23)

where \( A \) is a given \((g+1)\) component vector of unit length and \( a_i \) is a complex-valued scalar component.

It can be seen from equation 5.4.23 that, for one skywave, the vector \( Y_i \) must lie in the one dimensional subspace spanned by the vector \( A \).

The improved channel estimator, described previously for two skywaves, may be developed for operation with one or three skywaves which have fixed transmission delays, the basic structure and method of operation being essentially unchanged. The complete algorithm for one skywave HF channel estimation is as follows:

Step 1: \[ r_i' = \sum_{h=0}^{g} s_{i-h} Y_{i,i-1} \]

Step 2: \[ e_i' = r_i - r_i' \]

Step 3: \[ Y_i' = Y_{i,i-1} + \triangle e_i' S_i^* \]

\((S_i^*')^\dagger\) is the \((g+1)\)-component vector given by

\[
(S_i^*')^\dagger = [(S_i^1)^* (S_{i-1}^1)^* \ldots (S_{i-g}^1)^*]
\]

Step 4: \[ a_i = Y_i' A_i^* T \]

Step 5: \[ F_i = a_i A_i \]

Step 6: \[ H_i = Y_i' - F_i \]

Step 7: \[ A_{i+1}' = A_i + \eta a_i^* H_i \]
Step 8: \[ A_{i+1} = |A_{i+1}'|^{-1} A_{i+1}' \]

Step 9: \[ \varepsilon_{a,i} = a_i - a_{i,i-1} \]
\[ a_{i+1,i} = a_{i,i-1} + (1-\theta)^2 \varepsilon_{a,i} \]
\[ a_{i+1,i} = a_i + (1-\theta) \varepsilon_{a,i} \]

Step 10: \[ Y_{i+1,i} = a_{i+1,i} A_{i+1} \]

Step 11: \[ a_{i+n,i} = a_{i+1,i} + (n-1) a_{i+1,i} \]

Step 12: \[ Y_{i+n,i} = a_{i+n,i} A_{i+1} \]

For three skywaves HF channels the steps of the algorithm of estimation are as follows:

Steps 1, 2 and 3 here are the same as steps 1, 2 and 3 for one skywave

Step 4: \[ \alpha_i = Y_i^T A_i x^T \]
\[ \beta_i = Y_i^T B_i x^T \]
\[ \gamma_i = Y_i^T C_i x^T \]

Step 5: \[ F_i = \alpha_i A_i + \beta_i B_i + \gamma_i C_i \]

Step 6: \[ H_i = Y_i - F_i \]

Step 7: \[ A_{i+1}' = A_i + \eta \alpha_i H_i \]
\[ B_{i+1}' = B_i + \eta \beta_i H_i \]
\[ C_{i+1}' = C_i + \eta \gamma_i H_i \]
Step 8: \[ A_{i+1} = \frac{1}{|A_{i+1}|} A_{i+1} \]
\[ B_{i+1}^{''} = B_{i+1}^{'} - B_{i+1}^{'} A_{i+1}^{\dagger T} A_{i+1} \]
\[ B_{i+1} = \frac{1}{|B_{i+1}|} B_{i+1}^{''} \]
\[ \phi_{1i} = C_{i+1} A_{i+1}^{\dagger T} \]
\[ \phi_{2i} = C_{i+1} B_{i+1}^{\dagger T} \]
\[ C_{i+1}^{''} = \phi_{1i} A_{i+1} + \phi_{2i} B_{i+1} \]
\[ C_{i+1} = C_{i+1} - C_{i+1}^{''} \]
\[ C_{i+1} = \frac{1}{|C_{i+1}|} C_{i+1}^{''} \]

Step 9:
\[ \epsilon_{\alpha, i} = \alpha_{i} - \alpha_{i-1} \]
\[ \epsilon_{\beta, i} = \beta_{i} - \beta_{i-1} \]
\[ \epsilon_{\gamma, i} = \gamma_{i} - \gamma_{i-1} \]
\[ \alpha^{'}_{i+1, i} = \alpha^{'}_{i, i-1} + (1-\theta)^2 \epsilon_{\alpha, i} \]
\[ \beta^{'}_{i+1, i} = \beta^{'}_{i, i-1} + (1-\theta)^2 \epsilon_{\beta, i} \]
\[ \gamma^{'}_{i+1, i} = \gamma^{'}_{i, i-1} + (1-\theta)^2 \epsilon_{\gamma, i} \]
\[ \alpha_{i+1, i} = \alpha_{i, i-1} + \alpha^{'}_{i+1, i} + (1-\theta^2) \epsilon_{\alpha, i} \]
\[ \beta_{i+1, i} = \beta_{i, i-1} + \beta^{'}_{i+1, i} + (1-\theta^2) \epsilon_{\beta, i} \]
\[ \gamma_{i+1, i} = \gamma_{i, i-1} + \gamma^{'}_{i+1, i} + (1-\theta^2) \epsilon_{\gamma, i} \]

Step 10:
\[ \gamma_{i+1, i} = \alpha_{i+1, i} A_{i+1} + \beta_{i+1, i} B_{i+1} + \gamma_{i+1, i} C_{i+1} \]
Step 11: 
\[ \alpha_{i+n,i} = \alpha_{i+1,i} + (n-1) \beta_{i+1,i} \]
\[ \beta_{i+n,i} = \beta_{i+1,i} + (n-1) \beta_{i+1,i} \]
\[ \gamma_{i+n,i} = \gamma_{i+1,i} + (n-1) \gamma_{i+1,i} \]

Step 12: 
\[ Y_{i+n,i} = \alpha_{i+n,i} \beta_{i+1,i} + \beta_{i+n,i} \gamma_{i+1,i} + \gamma_{i+n,i} \gamma_{i+1,i} \]

5.5 FAST CHANNEL ESTIMATOR

Before the improved estimator can start to operate, the initial values of the vectors \( A_i, B_i \) and \( C_i \) must be available. These vectors \( A_0, B_0 \) and \( C_0 \) may be determined from the estimates of the channel, as will be described later on. Thus initial estimates of the unknown channel should be provided before the data can be transmitted. Furthermore, this estimator should provide the estimates in a short time in order to speed up the starting up of the improved estimator. The fast channel estimator that will be considered here yields the optimum estimate of the channel at the fastest possible rate, and by a relatively simple method. This estimator requires some special inputs and is ideal for use during the training period at the start of a transmission.

5.5.1 Fast Channel Estimation Method

Consider first a QAM data transmission system with a time invariant transmission path, which has a complex valued impulse response:

\[ Y = [y_0 y_1 \ldots y_g 0 0 0] \]  \hspace{1cm} (5.5.1)

\( Y \) is here an \( n_S \)-component row vector with the appropriate number of 0's added at the end of the original \( g+1 \) component vector. Assume a fixed sequence of data symbols \( s_i \) of length \( n_S \) is transmitted over the channel repetitively. Where each sequence is given by the row vector:

\[ S_0 = [s_0 s_1 \ldots s_{n_S-1}] \]
It is assumed, for convenience that, at time \( t=0 \), the data symbol \( s_0 \) is transmitted. Consider now the positive integer \( i \), where \( i=mn_s+k \) and where \( m \) and \( k \) are positive integers such that \( 0<k<n_s \). Then at time \( t=iT \), the data symbol \( s_k \) is transmitted

\[
k \equiv i \mod n_s
\]  

(5.5.3)

and \( k \) is the integer \( i \) expressed modulo-\( n_s \) for example, if \( i=10 \) and \( n_s=4 \), then \( k=2 \) thus \( s_i=s_k \).

Here the received sample \( r_i \) is given by:

\[
r_i = \sum_{h=0}^{n_s-1} s_{i-h}a_h + w_i
\]  

(5.5.4)

from equation 5.5.3

for \( 0 \leq h \leq k \) \( s_{i-h} = s_{k-h} \)  

(5.5.5)

and for \( n_s > h > k \) \( s_{i-h} = s_{k-h+n_s} \)  

(5.5.6)

\( w_i \) is a complex-valued Gaussian random variable. The real and imaginary parts of \( w_i \) are statistically independent with zero mean and variance \( \sigma^2 \).

Define

\[
R_i = [r_i \ r_{i+1} \ldots \ r_{i+n_s-1}]
\]  

(5.5.7)

\[
W_i = [w_i \ w_{i+1} \ldots \ w_{i+n_s-1}]
\]  

(5.5.8)

then, for \( i=n_s, 2n_s, 3n_s \ldots \) equation 5.5.4 may be expressed in a matrix form as follows
\[ R_i = YG + W_i \]  \hspace{1cm} (5.5.9)

where \( G \) is defined as

\[
G = \begin{bmatrix}
  s_0 & s_1 & s_2 & \cdots & s_{n_S-1} \\
  s_{n_S-1} & s_0 & s_1 & \cdots & s_{n_S-2} \\
  \vdots & \vdots & \vdots & \ddots & \vdots \\
  s_2 & s_3 & s_4 & \cdots & s_1 \\
  s_1 & s_2 & s_3 & \cdots & s_0
\end{bmatrix}
\]  \hspace{1cm} (5.5.10)

The \( G \) matrix is known at the receiver.

Assume a self orthogonal sequence is used as the repetitive data sequence \( S_0 \). Mathematically a self orthogonal sequence is defined by the following property:

\[
\sum_{i=0}^{n_S-1} s_i^* s_{i+(i+k) \mod n_S} = \begin{cases} 
\lambda & \text{if } k = 0 \\
0 & \text{if } k \neq 0
\end{cases}
\]  \hspace{1cm} (5.5.11)

where

\[
\lambda = \sum_{i=0}^{n_S-1} |s_i|^2
\]

Then it can be shown that, under the assumed conditions, the maximum-likelihood estimate of \( Y \) (which also minimizes the mean square error) can be obtained from \( R_i \), as follows [84]:

\[
Y^* = R_i G^{-1} = \lambda^{-1} R_i (G^*)^T
\]  \hspace{1cm} (5.5.12)
where

$$(G^*)^T = \begin{bmatrix}
  s_0^* & s_{n-1}^* & \cdots & s_1^* \\
  \vdots & \ddots & \vdots & \vdots \\
  s_{n-1}^* & \cdots & s_0^*
\end{bmatrix} \quad (5.5.13)$$

$s_k^*$ is the complex conjugate of $s_k$.

Also it can be shown [84] that the mean square error in the estimate $Y''$ is

$$E \left( |Y'' - Y|^2 \right) = 2\sigma^2 n_s / \lambda \quad (5.5.14)$$

and this estimate is the best estimate of the channel under the assumed conditions.

A further important property of this method of estimating $Y$ is that the error in the estimates is independent of $Y$ itself (equation 5.5.14).

Note that the above method can be used to estimate a slowly varying impulse response $Y_i$ (as in the case of HF links), by assuming that $Y_i$ remains constant during the time of $r_i, r_{i+1} \cdots r_{i+n_s-1}$, but now the estimator is not optimum but suboptimum due to the above assumption.

5.5.2 Computer Simulation Tests

A self orthogonal sequence is used as a training sequence $S_0$ at the start of transmission. A small DC value ($= 0.15$) is added to the components of a pseudorandom binary sequence of length $n_s = 31$ [85] to convert it to a self orthogonal sequence [84]. The pseudorandom binary
sequence used in the tests is given in Table 5.5.1. This sequence length is chosen so as to be able to estimate impulse responses even when the delay between the skywaves reaches 5msec. The number of components in the impulse response increases as the delay between the skywaves increases (Chapter 3).

Computer simulation results for the mean square error in the channel estimate vs signal/noise ratio (s/n) for the channel using this self orthogonal sequence are as shown in Figure 5.5.1. The time variant channel here is a two skywave HF link with a relative transmission delay of 0.8 msec. The signal received over each skywave has the same frequency spread ($f_{sp}$). In this figure the mean square error in the estimate is defined as:

$$\frac{1}{L} \sum_{i=0, 31, 62, \ldots, 31(L-1)} |Y_i^" - Y_{i+15}|^2$$

where

$$L = 1700$$

$Y_i^"$ is the estimated impulse response obtained from the received samples $r_i, r_{i+1} \ldots r_{i+30}$; $Y_{i+15}$ is the actual sampled impulse response at time $t=(i+15)T$. As it can be seen from equation 5.5.15 each estimate is compared with $Y_{i+15}$, which is at the centre of the interval $t=(iT)$ to $(i+30)T$. The s/n is as defined by equation 4.4.2. Note that $Y_i$ is not the true sampled impulse response, but is formed by the components of this sequence at time $t=iT$.

The maximum-likelihood estimates given by equation 5.5.12 assume that the impulse response remains constant during the time interval $r_i$ to $r_{i+nS-1}$. For a time variant channel this assumption is not exactly true, and this explains why a difference exists between the time variant and time invariant channels in the Figure 5.5.1. The more rapid the rate of variation of the channel (or the wider the frequency spread), the greater is the error in the estimate and hence the greater is the deviation from the time invariant channel. This difference in the error of the estimate is more prominent at high s/n, since here most of the error in the estimate is caused by
TABLE 5.5.1: Pseudo random binary sequence of length 31

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>1</th>
<th>-1</th>
<th>1</th>
<th>1</th>
<th>-1</th>
<th>1</th>
<th>1</th>
<th>1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
</tbody>
</table>

1
Fig. 5.5.1: Fast channel estimator performance
the fading or channel variation and there is only a small error due to the noise.

5.6 STARTING UP PROCEDURE FOR IMPROVED ESTIMATOR
5.6.1 Starting Up with the Number of Skywaves Known at the Receiver
5.6.1.1 Two skywaves starting up:

As can be seen from Section 5.4 the estimates of the channel are found by a recursive algorithm and so it must be initialised. The following initial values are needed for a two-skywave HF channel estimator:

i) $A_0$ and $B_0$

ii) $a_0, -1, a_0', \beta_0, -1, \beta_0', a_0, -1, \beta_0', -1$ and $Y_0', -1$

i) The Initial Subspace

The initial subspace, spanned by $A_0$ and $B_0$, in the improved estimator may be formed from the initial estimates of the channel by any of the following methods.

METHOD A

Using a maximum-likelihood estimation method (Section 5.5), estimates of the sampled impulse response of the channel are obtained at two well-spaced time instants $t=-kT$ and $t=0$. Let these estimates be $Y_{-k}$ and $Y_0'$ respectively. The constant $k$ is a reasonably large positive integer, say 100. It is assumed that these estimates are not co-linear. The estimates $Y_{-k}'$ and $Y_0''$ are then orthonormalized to give the orthonormal vectors [80]

$$A_0 = |Y_{-k}'|^{-1} Y_{-k}'$$  \hspace{1cm} (5.6.1)

$$B_0' = Y_0'' - Y_0'' A_0 A_0^T A_0$$  \hspace{1cm} (5.6.2)

$$B_0 = |B_0'|^{-1} B_0'$$  \hspace{1cm} (5.6.3)
The vectors $A_0$ and $B_0$ therefore forms an orthonormal basis of the two-dimensional subspace spanned by $Y_{-k}$ and $Y_0$.

**METHOD B**

Consider the estimates of the channel $Y_1^\prime, Y_2^\prime, \ldots, Y_m^\prime$, found from the maximum likelihood estimation method described in Section 5.5, where:

\[
Y_1^\prime = [Y_1,0 \ Y_1,1 \ \cdots \ Y_1,g']
\]
\[
Y_2^\prime = [Y_2,0 \ Y_2,1 \ \cdots \ Y_2,g']
\]
\[
Y_m^\prime = [Y_m,0 \ Y_m,1 \ \cdots \ Y_m,g']
\]

(5.6.4)

$g' = 30$ (Section 5.5) and $m$ is an integer the value of which will be determined later on. $g'$ is used (not $g$) since we are assuming the delay between the skywaves to be unknown at the receiver (Section 5.5.2).

Now, the $i$th component of the estimate of the sampled impulse-response $Y_i^\prime$ is $Y_{i,h}^\prime$ where $h=0,1, \ldots, g'$.

If no noise is added during transmission, the amplitude variation with $i$ of the real part of the estimate of a component $Y_{i,h}^\prime$ are as shown in Figures 5.6.1-5.6.4. In these figures, the points on the lines give the real part of a component of the sampled impulse-response at different times. Notice the very slowly changing and therefore highly predictable variation of the components, when there is no noise. The variation appears in each case to be linear with time and a similar relationship seems to hold also for all other real and imaginary components (i.e. for all $h$). This is the typical situation in high speed HF data transmission, where the data rate is much greater than the maximum channel-fading bandwidth [7].
Fig. 5.6.1: Amplitude variation of the real part of the fourth component of the estimate of sampled impulse response (y''_i,3) with time for two skywaves, \( \tau_1 = 0.8 \) msec, 0.5 Hz frequency spread

---

- **without noise**
- **with noise**
- **S/N = 30 dB**
Fig. 5.6.2: Amplitude variation of the real part of the fifth component of the estimate of sampled impulse response (y1, 4) with time for two skywaves, t1 = 0.8 msec, 0.5 Hz frequency spread.
Fig. 5.6.3: Amplitude variation of the real part of the ninth component of the estimate of the sampled impulse response \( y''_{i,8} \) with time, for two skywaves \( \tau_1 = 0.8 \text{ msec}, 0.5 \text{ hZ frequency spread} \).
Fig. 5.6.4: Amplitude variation of the real part of the tenth component of the estimate of the sampled impulse response \( y_{1,9} \) with time, for two skywaves, \( \tau_1 = 0.8 \text{ msec}, 0.5 \text{ Hz frequency spread} \)
In practice noise is always present. Thus the components of the estimate $Y_i$, obtained from a process of maximum-likelihood estimation, will have noise added to them (Section 5.5), and the components now differ from those without noise, as depicted in Figures 5.6.1-5.6.4. It can be seen clearly from these figures that if a degree-1 least squares fit is applied to the noisy components, the resulting straight line approximates quite closely to that for no noise. The points (components) given by this line have considerably less noise than the original noisy estimates, and so are more accurate estimates. These more accurate estimates, in turn, give a more accurate subspace. Clearly, from equation 5.4.2, the best subspace (plane) could be found from the two vectors $Y_{-K}$, $Y_0$, which are the exact impulse responses of the channel without any error and provided that they are not co-linear (see Section 5.4.1).

In method B, $m$ estimates ($Y_1^\prime Y_2^\prime \ldots Y_m^\prime$) of the channel are found, using maximum-likelihood estimation as described in Section 5.5. Then an estimate of the channel, for time $t=t_1$, is obtained from the sequence of vectors $Y_1^\prime Y_2^\prime \ldots Y_{m/2}^\prime$ by determining the set of $g+1$ polynomials of degree 1, each of which gives the least-squares fit to the corresponding component of the vectors $Y_1^\prime Y_2^\prime \ldots Y_{m/2}^\prime$. Then using the values of the polynomials at time $t=t_1$, as shown in Figure 5.6.5A, the $g+1$ components of the $Y_{t_1}^\prime$ can be determined. Actually this line fitting could be carried out by a simple recursive algorithm, called expanded memory (Appendix E). Thus, using the above technique $Y_{t_1}^\prime$ will have less error due to noise. Now another vector is required to form the initial plane. This may be found by exactly the same method, that is, by applying a degree-1 fit to $Y_{m/2+1}^\prime \ldots Y_m^\prime$ to get $Y_{t_2}^\prime$. From $Y_{t_1}^\prime$ and $Y_{t_2}^\prime$ the two orthonormal vectors $A_\alpha$, $B_\alpha$ can be obtained using equations 5.6.1-5.6.3. If the mid points of the lines are taken, then $t_1$ and $t_2$ are as shown in Figure 5.6.5A. However, if end points are taken on the line fit then $t_1$, $t_2$ are as shown in Figure 5.6.5B. Two line fittings are used here because if the period ($\Delta t$) is too large there is a possibility that the variation will not remain linear when there is a large frequency spread. It is important here that the two vectors $Y_{t_1}^\prime$ and $Y_{t_2}^\prime$ are obtained at well-spaced time instants $t_1$, $t_2$ to avoid the possibility of them being
Fig. 5.6.5: Degree 1 fitting to the component amplitude variation with time:

A. $t_1$, $t_2$ mid-points of the line fits
B. $t_1$, $t_2$ end-points of the line fits

$R_m[y''_{i,h}]$ real or imaginary part of $y''_{i,h}$
colinear because of the slowly varying HF channel. This could be achieved by making \( m \) large.

**METHOD C:**

This method is similar to method B, but rather than using degree-1 fittings (employing two line fittings), a single degree-2 recursive fitting (Appendix E) is used for the components as shown in Figure 5.6.6. Similarly two vectors \( Y_{t_1}', Y_{t_2}' \) are taken to form the plane.

**METHOD D:**

This method is a degree-0 fitting, to give

\[
Y_{t_1}' = \frac{1}{m} \sum_{i=1}^{\frac{m}{2}} Y_{i}''
\]

(5.6.5)

and

\[
Y_{t_2}' = \frac{1}{m} \sum_{i=\frac{m}{2}+1}^{m} Y_{i}''
\]

(5.6.6)

from \( Y_{t_1}' \) and \( Y_{t_2}' \) the two orthonormal vectors \( A_o \) and \( B_o \) can be obtained using equations 5.6.1-5.6.3. Finally, in this method, \( Y_{t_1}' \) and \( Y_{t_2}' \) can be found recursively without the need to store all the \( (m) \) impulse responses as suggested by equations 5.6.5-5.6.6 (Appendix E).

**METHOD E:**

In this method \( Y_1'' \) and \( Y_{(m/2)+1}'' \) are taken, and \( A_1 \) and \( B_1 \) are found from them (using equations 5.6.1-5.6.3). Then, from \( Y_2'' \) and \( Y_{(m/2)+2}'' \), \( A_2 \) and \( B_2 \) are found. In general, from \( Y_i'' \) and \( Y_{(m/2)+i}'' \), the \( A_i \) and \( B_i \) are found, for \( i=1,2, \ldots \frac{m}{2} \). Then the resultant vectors are averaged to give

\[
A_o' = \frac{1}{m} \sum_{i=1}^{\frac{m}{2}} A_i
\]

(5.6.7)

and

\[
B_o' = \frac{1}{m} \sum_{i=1}^{\frac{m}{2}} B_i
\]

(5.6.8)
Fig. 5.6.6: Degree 2 fitting to the component amplitude variation with time
Finally equations 5.6.1-5.6.3 are used to find the two orthonormal vectors \( A_0 \) and \( B_0 \) from \( A_0', B_0' \).

**METHOD F:**

\( A_0'' \) and \( B_0'' \) are found from \( Y_1'' \) and \( Y_m'' \) using method A. Then the subspace spanned by \( A_0'', B_0'' \) is moved to contain \( Y_2'', Y_3'' \ldots Y_i'' \ldots Y_m'' \). It can be shown that this could be carried out as follows [80]:

\[
A_i'' = A_0'' + \eta \alpha_i H_i
\]

(5.6.9)

and

\[
B_i'' = B_0'' + \eta \beta_i H_i
\]

(5.6.10)

for \( i = 2, 3, \ldots, m-1 \)

where

\[
\eta = (|\alpha_i|^2 + |\beta_i|^2)^{-1}
\]

(5.6.11)

The resultant subspaces are averaged, to give

\[
A_0' = \frac{1}{m} (2 A_0'' + A_2'' + \ldots + A_{m-1}'')
\]

(5.6.12)

and

\[
B_0' = \frac{1}{m} (2 B_0'' + B_2'' + \ldots + B_{m-1}'')
\]

(5.6.13)

Next, the two orthonormal vectors \( A_0, B_0 \) are obtained from \( A_0', B_0' \) using equations 5.6.1-5.6.3.

**ii) Initial Values in the Improved Estimator**

In the improved estimator, for two skywaves, the initial values of the scalars \( a_{i,i-1}', \beta_{i,i-1}', a_i', \beta_i', a_{i,i-1}, \beta_{i,i-1} \) are set to:
\[ a_{o,-1} = y_o^* y_o \cdot y_o^* = \alpha \cdot y_o^* y_o \cdot y_o^* T \quad (5.6.14) \]

\[ b_{o,-1} = y_o^* b_o \cdot b_o^* = \beta \cdot y_o^* b_o \cdot b_o^* T \quad (5.6.15) \]

\[ a_{o,-1}' = 0.0 \quad (5.6.16) \]

\[ b_{o,-1}' = 0.0 \quad (5.6.17) \]

and the initial vector \( y_{i,1} \) is given by \( y_{o,-1}' = y_{m}'' \) (5.6.18)

### 5.6.1.2 Starting Up with One or Three Skywaves

Basically the same previously described methods (A-F) can be used to form the initial subspace, if the number of skywaves is known to be three. However, now in method A three well-spaced estimates of the channel \( y_{-2k}'' \), \( y_{-k}'' \) and \( y_{o}'' \) are required to form the initial three dimensional subspace. \( k \) is a reasonably large positive integer. Here, firstly a two dimensional subspace (plane) is formed from the vectors \( y_{-2k}'' \) and \( y_{o}'' \) using equations 5.6.1-5.6.3

\[ a_o = \left| y_{-2k}'' \right|^{-1} y_{-2k}'' \quad (5.6.19) \]

\[ b_o' = y_o'' - y_o'' a_o \cdot a_o^* b_o \quad (5.6.20) \]

\[ b_o = \left| b_o' \right|^{-1} b_o' \quad (5.6.21) \]

Then the component of \( y_{-k}'' \) orthogonal to the \( a_o-b_o \) plane is found using the following equations [80]:
\[ \phi_{11} = Y_{-k} A_o * T \]  
\[ \phi_{12} = Y_{-k} B_o * T \]  
\[ C''_o = \phi_{11} A_o + \phi_{12} B_o \]  
\[ C''_o = Y_{-k} - C''_o \]  

Finally \( C''_o \) is normalized to give:

\[ C_o = \left| C''_o \right|^{-1} C''_o \]  

The vectors \( A_o, B_o \) and \( C_o \) form an orthonormal basis of the three dimensional subspace scanned by \( Y_{-2k}, Y_{-k}, Y_o \). As before, degree-0 and degree-1 fittings can be used to obtain three estimates \( Y_{t_1}', Y_{t_2}', Y_{t_3}' \) employing three straight-line fittings (methods B and D). Also, three estimates can be obtained using a single degree-2 curve fitting (method C). The points \( t_1, t_2, t_3 \) are as shown in Figures 5.6.7, 5.6.8 for degree-1 and degree-2 fittings. From \( Y_{t_1}', Y_{t_2}', Y_{t_3}' \), the initial three dimensional subspace is formed using the above equations, now with \( Y_{-2k}', Y_{-k}', Y_o' \) replaced by \( Y_{t_1}', Y_{t_2}' \) and \( Y_{t_3}' \) respectively.

Methods E and F are the same as that for the two skywaves, but a three dimensional subspace is used here instead of a two dimensional subspace. If there is one skywave in the HF link, the initial one dimensional subspace can be obtained from one accurate estimate \( Y_{t_1}' \) by simply normalizing this vector. Hence, methods B, C and D (one line fittings) can be used here to get one required accurate estimate \( Y_{t_1}' \).
Fig. 5.6.7:  
A. $t_1$, $t_2$, and $t_3$ mid-points of the line fits  
B. $t_1$, $t_3$ end points of the line fits
Fig. 5.6.8: Degree 2 fitting to the component amplitude variation with time
Finally, the initial values of other parameters in the improved estimator, for one and three skywaves, are set similarly to that for two-skywave starting up (equations 5.6.14-5.6.18).

5.6.1.3 Computer Simulation Tests

In these tests, the subspace is formed from initial estimates of the channel using any of the different methods (A-F). The other values are initialized as described before. The values of the constants $\Delta$ and $\theta$ in the improved estimator (Section 5.4) are optimized by the method described in Ref. 59 (see Table 5.6.1 for the optimum values of $\Delta$ and $\theta$). Furthermore, in the improved estimator, $n$ is made large at the start and then changed to a smaller value, the reason for this will become clear later on. With all the initial values now specified, the improved estimator operates exactly as previously described in Section 5.4, and the behaviour during start up is shown in Figures 5.6.9-5.6.13. In these tests, $\tau_1$ is the relative delay in transmission between the first and the second skywaves, and $\tau_2$ is that between the first and the third skywaves.

The s/n is as defined by equation 4.4.2 and it is 30 dB in these tests. The parameter $\xi_1$ is here the square of the error in $Y_{i+n,i}'$ measured in dB relative to unity and is

$$\xi_1 = 10 \log_{10} (|Y_{i+n} - Y_{i+n,i}'|^2)$$

(5.6.27)

$|Y_{i+n} - Y_{i+n,i}'|$ is the unitary length of the vector $Y_{i+n} - Y_{i+n,i}'$ and so is the unitary distance between the vector $Y_{i+n}$ and $Y_{i+n,i}'$. In these tests $n=17$, where $n$ sampling interval is the delay in estimation.

In Figures 5.6.9-5.6.13 the estimator does not start operating at time $t=0$, there being a certain delay. This is because a time delay is required to estimate the subspace, as described before. The delay is greater for two skywaves than for one skywave, because two well-spaced estimates in time are required for two skywaves. For three skywaves, the delay is even longer than that for two skywaves, since here three
TABLE 5.6.1: Optimum $\Delta$ and $\theta$ for the improved channel estimator

<table>
<thead>
<tr>
<th>Frequency Spread</th>
<th>$\Delta$</th>
<th>$\theta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5 Hz</td>
<td>0.03</td>
<td>0.97</td>
</tr>
<tr>
<td>2 Hz</td>
<td>0.05</td>
<td>0.94</td>
</tr>
</tbody>
</table>
Fig. 5.6.10: Error in the estimate vs time for two skywaves, $\tau_1 = 0.8$ msec, methods A and F of forming the initial subspace, 0.5 Hz frequency spread
Fig. 5.6.11: Error in the estimate vs time for one skywave. Methods B, C and D of forming the initial subspace, 0.5 Hz frequency spread
Fig. 5.6.12: Error in the estimate vs time for three skywaves, $T_1 = 0.8$ msec, $T_2 = 1.8$ msec methods B, C, D and E of forming the initial subspace.

$E_{n}^{m}$ dB
Fig. 5.6.13: Error in the estimate vs time for three skywaves $\tau_1 = 0.8$ msec, $\tau_2 = 1.8$ msec. Method A and F of forming initial subspace, 0.5 Hz frequency spread
well-spaced estimates are now required. Extensive computer simulation tests have been carried out to find approximately the optimum values of the above delays. These optimum delay values are shown in Figures 5.6.9-5.6.13.

Tables 5.6.2-5.6.4 give the mean-square error $\xi$ in $y'_{i+n,i}$, for each method of forming the subspace, measured in dB relative to unity, to give

$$\xi = 10 \log_{10} \left( \frac{1}{48000} \sum_{i=4801}^{52800} |y_{i+n} - y'_{i+n,i}|^2 \right) \quad (5.6.28)$$

It can be seen from Table 5.6.2 and Figures 5.6.9-5.6.10 that, for two skywaves, methods B-E give the best planes (estimate of the subspace), since for these planes, the estimator converges to the steady state condition rapidly. The reason that degree-0 and degree-1 (middle points) give about the same plane may be clarified by Figure 5.6.14. It can be seen that both give almost the same point "aa", particularly at high s/n.

Also it has been found that making $\eta$ large at the start and then switching it to a smaller value, speeds up the initial convergence, as shown in Figure 5.6.15.

Furthermore, tests have shown that setting the values of $a'_{o,-1}$ and $b'_{o,-1}$ as suggested by reference 80, does not speed up the convergence at start up over that obtained by setting them to zero.

For one skywave (Figure 5.6.11), methods C, B and D give about the same performance. Furthermore, for three skywaves (Figures 5.6.12-5.6.13), methods B-E of forming the subspace are much better than methods A and F, as is also the case for two skywaves.

Since method D is the simplest of the methods B-E, it is the most promising technique for forming the initial subspace. Firstly this
TABLE 5.6.2: Two skywaves mean square error in the estimates ($\xi$) for different methods of forming the initial subspace

S/N = 30 dB, $\tau_1 = 0.8$ msec, 0.5 Hz frequency spread

<table>
<thead>
<tr>
<th>Method</th>
<th>$\xi$ (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>B (mid points)</td>
<td>-42.78</td>
</tr>
<tr>
<td>B (end points)</td>
<td>-42.79</td>
</tr>
<tr>
<td>C</td>
<td>-42.8</td>
</tr>
<tr>
<td>D</td>
<td>-42.76</td>
</tr>
<tr>
<td>E</td>
<td>-42.75</td>
</tr>
<tr>
<td>A</td>
<td>-41.22</td>
</tr>
<tr>
<td>F</td>
<td>-41.59</td>
</tr>
</tbody>
</table>
### TABLE 5.6.3: One skywave mean square error in the estimates for different methods of forming the initial subspace
(S/N = 30 dB, 0.5 Hz, frequency spread)

<table>
<thead>
<tr>
<th>Method</th>
<th>ξ (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>-45.46</td>
</tr>
<tr>
<td>B</td>
<td>-45.46</td>
</tr>
</tbody>
</table>

### TABLE 5.6.4: Three skywaves mean square error in the estimates for different methods of forming the initial subspace
(S/N = 30 dB, τ₁ = 0.8 msec, τ₂ = 1.8 msec, 0.5 Hz frequency spread)

<table>
<thead>
<tr>
<th>Method</th>
<th>ξ (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>B (mid points)</td>
<td>-40.19</td>
</tr>
<tr>
<td>B (end points)</td>
<td>-40.41</td>
</tr>
<tr>
<td>C</td>
<td>-40.67</td>
</tr>
<tr>
<td>D</td>
<td>-40.12</td>
</tr>
<tr>
<td>E</td>
<td>-40.07</td>
</tr>
<tr>
<td>A</td>
<td>-34.78</td>
</tr>
<tr>
<td>F</td>
<td>-35</td>
</tr>
</tbody>
</table>
Fig. 5.6.14: Relationship between degree 0 and degree 1 fitting to the components $y_{i,h}^{''}$.
Fig. 5.6.15: Different $\eta$ values at the start of estimation for two skywaves, $\tau_1 = 0.8$ msec, method D of forming the initial subspace, 0.5 Hz frequency spread.
method is simple and does not need many computations. Secondly, the operations in this method may be carried out recursively (Appendix E) and hence less storage is required. The other conclusion that can be derived from the results (Tables 5.6.2-5.6.4) is that the smaller the number of skywaves (or the smaller the dimension of the subspace) the more accurate are the estimates. For example the mean square error for one skywave is better than that for two skywaves. This is because the number of parameters estimated is reduced, which reduces the error. For one skywave only $a_i$ is estimated while for two skywaves $a_i$ and $b_i$ are estimated, so that greater errors are expected.

Finally, from Table 5.6.2 and Figure 5.5.1 it can be seen clearly that the improved channel estimator is better than the maximum-likelihood estimator for $s/n = 30$ dB. This is because of the use of the available prior knowledge of the basic structure of the channel by the improved estimator.

5.6.2 Starting Up with an Unknown Number of Skywaves

5.6.2.1 Starting up procedure

In a situation where the number of skywaves is unknown at the receiver, the estimator must first estimate the number of skywaves, form the appropriate initial subspace and then use the correct estimation algorithm. Hence, start up may be carried out here in the following steps:

Step 1: Three vectors are found using the following equations:

\[
\hat{A} = \frac{1}{m/3} \sum_{i=1}^{m/3} Y_i^" \\
\hat{C} = \frac{1}{m/3} \sum_{i=(m/3)+1}^{2m/3} Y_i^" \\
\hat{B} = \frac{1}{m^3} \sum_{i=(2m/3)+1}^{m} Y_i^" 
\] (5.6.29, 5.6.30, 5.6.31)
\( y_i \) is the estimate of the channel found from maximum-likelihood estimation (Section 5.5), \( m = 100 \).

The degree-0 method is used to find the vectors \( \hat{A}, \hat{B} \) and \( \hat{C} \).

Step 2: The angle \( \phi \) between the two vectors \( \hat{A} \) and \( \hat{B} \) is for our purposes taken to be given by the equation

\[
\phi = \cos^{-1} \frac{|\hat{A} \hat{B}^*|}{|\hat{A}| |\hat{B}|}
\]  

(5.6.32)

If the angle between \( \hat{A} \) and \( \hat{B} \) is small (less than 180°) or the two vectors are approximately colinear, a one dimensional subspace is formed from the vector \( \hat{B} \) by normalizing it. The channel estimation is then carried out using a one skywave estimation algorithm (Section 5.4), starting from the initial subspace formed from the \( \hat{B} \) vector. The other initial values are set as in Section 5.6.1. The two vectors are usually approximately colinear when there is one skywave.

However, if the angle is large, (greater than 180°), then a two dimensional subspace is formed from these vectors. The components of the vector \( \hat{C} \), orthogonal to the \( \hat{A}-\hat{B} \) plane is then calculated. If this orthogonal component (\( C_{\text{Orth}} \)) is small (less than a threshold value), that means that there are only two skywaves. Then the two dimensional subspace formed from the \( \hat{A}-\hat{B} \) vectors is used as the initial subspace for starting up a two skywave estimation algorithm (Section 5.4).

Finally, if the orthogonal component of \( \hat{C} \) is large, (\(|C_{\text{Orth}}|^2 > 0.006\)) a three dimensional subspace is formed from the \( \hat{A}, \hat{B} \) and \( \hat{C} \) vectors and this subspace is used to start up a three-skywave channel estimator algorithm (Section 5.4). A flow diagram of the above starting up algorithm is given in Figure 5.6.16.
For the appropriate subspace setting other initial values \((\alpha_0, \ldots)\) and switching to the correct algorithm of estimation.

Finding estimates of the sampled impulse response (Section 5.5)

Finding \(\hat{A}, \hat{B}, \hat{C}\) vectors from the estimates of the channel

Estimate the number of skywaves

Forming the appropriate subspace setting other initial values \((\alpha_0, \ldots)\) and switching to the correct algorithm of estimation

Switch to 1. IF \(\hat{A}-\hat{B}\) angle is small
2. IF \(\hat{A}-\hat{B}\) angle is large and the orthogonal component of \(\hat{C}\) to \(\hat{A}-\hat{B}\) plane is small
3. IF \(\hat{A}-\hat{B}\) angle is large and the orthogonal component of \(\hat{C}\) to \(\hat{A}-\hat{B}\) plane is large

Fig. 5.6.16: The algorithm for starting up when the number of skywaves is unknown
5.6.2.2 Computer Simulation Tests

Using the above technique the starting up behaviour for a one, two and three skywave HF channel are obtained as shown in Figures 5.6.17-5.6.19. Also Figures 5.6.20-5.6.22 show the starting up behaviour for different HF channel conditions. Here $\xi_1$ and $s/n$ are as defined in Section 5.6.1.

In these tests, the number of skywaves is unknown at the receiver. The receiver here finds the number of skywaves by the method described above and switches to the appropriate estimation algorithm.

As before, in these tests $n$ is made large at the start and then switched to a smaller value.

As can be seen from Figures 5.6.17-5.6.22, by this method of starting up, the improved estimator usually reaches the steady state condition within about 1.25 seconds (around 3000 samples). This is so provided that the skywaves are all present, with significant magnitudes, during the starting up period (i.e. none of the skywaves is going through a deep fade).

The delay between the different skywaves is usually small (typically less than 2 msec) [9]. For this reason the last few components of the sampled impulse response are usually small compared to the first few components (Figure 5.6.23). Hence by putting a threshold, we can reduce the last few components of the vectors $\hat{A}$, $\hat{B}$ and $\hat{C}$ (in equations 5.6.29-5.6.31) to zero. In this way, even more accurate estimates and so better subspaces can be obtained.

5.7 Adaptive Improved Estimator

5.7.1 Introduction

If the HF link is formed from one skywave, the estimator commences operation by using the previously described starting up procedure. The improved estimator performance will be at its best, as long as the number of skywaves remains the same as it started with (i.e. one
Fig. 5.6.19: Error in the estimate vs time for three skywaves, $\tau_1 = 0.8$ msec, $\tau_2 = 1.8$ msec, 0.5 Hz frequency spread, S/N 30 dB
Fig. 5.6.21: Error in the estimate vs time for two skywaves, \( \tau_1 = 0.8 \) msec, 2 Hz frequency spread, S/N = 30 dB
Fig. 5.6.22: Error in the estimate vs. time for three skywaves, $\tau_1 = 0.8$ msec, $\tau_2 = 2.8$ msec, 2 Hz frequency spread, $S/N = 30$ dB
Fig. 5.6.23: The typical relative amplitudes of the components of the estimate of the sampled impulse response
skywave). However if another skywave appears later on, and this could happen in practice, the estimator performance will be degraded considerably. The estimator performance will now be even worse than a simple estimator (feedforward estimator with prediction) as can be seen from Figure 5.7.1. That is because it assumes and uses a one-skywave estimation algorithm (Section 5.4.2) for the channel which has now two skywaves, so there is a serious error in the assumed subspace. The estimation method (described in Section 5.4) assumes the use of correct subspace. The same thing applies when the estimator commences operation with two skywaves and then a third skywave appears later on (Figure 5.7.2). To overcome this problem, the estimator should be adaptive. An adaptive estimator is explained in Section 5.7.2 and the results of computer simulation tests on this estimator are given in Section 5.7.3.

5.7.2 Adaptive Estimation

To make the estimator adaptive, that is to make it change its subspace according to the structure of the HF link, the estimator must be able to do two things. Firstly, it should be able to detect the presence of additional skywaves quickly, before the estimator performance becomes too poor (see Figures 5.7.1-5.7.2). Secondly, it should then find the new number of skywaves and the new subspace, and then switch to the correct estimation model.

To achieve these objectives, it has been proposed that the estimator should be backed (supported) by a simpler separate estimator which is the feedforward estimator with prediction (Section 5.3.2). The simple estimator, as can be seen from Section 5.3.2, makes no particular assumptions concerning the structure of the channel model such as, for example, the number of skywaves and their relative transmission delays. When the improved estimator is operating correctly, its performance is better than that of the simple estimator (Section 5.3.4).

In the proposed adaptive estimator (Figure 5.7.3), a continuous comparison is made between the performance of the two estimators as determined by the magnitudes of the error signals used to update the
Fig. 5.7.1: Error in the estimate vs time for one skywave, then two skywaves.
S/N = 30 dB, 0.5 Hz frequency spread
Fig. 5.7.2: Error in the estimate vs time for two skywaves then three skywaves.
S/N = 30 dB, 0.5 Hz frequency spread
Simple estimator
(feedforward estimator with prediction)

Fig. 5.7.3: Adaptive estimator
channel estimate. Whenever the estimate from the improved estimator becomes worse than that of the simple estimator, the simple estimator is used temporarily in place of the other.

The comparison of the performances of the two estimators is done here as follows:

In the improved estimator, from equation 5.4.7:

\[ e_i' = r_i - r_i' \]  \hspace{1cm} (5.7.1)

where from equation 5.4.6

\[ r_i' = \sum_{h=0}^{g'} s_{i-h} Y_{i-1,h} \]  \hspace{1cm} (5.7.2)

\( Y_{i-1,0}, Y_{i-1,1}, \ldots, Y_{i-1,g'} \) are the one-step prediction of the estimates of the channel from the improved estimator and \( s_i \) is the detected data symbols.

Let

\[ (e_{av})^i = \frac{1}{N_a} \sum_{j=i}^{i+95} \| e_j' \| \]  \hspace{1cm} (5.7.3)

\( e_{av} \) is the short term average of the \( \| e_j' \| \) and \( N_a = 96 \).

At low error rates, when the estimates \( Y_{i-1} \) are accurate, then \( r_i' \) is close to \( r_i \).

Hence, \( e_i' \) (or \( e_{av} \)) will be small particularly at high s/n. Therefore \( e_{av} \) is a measure of how good is the performance of this estimator. When it is small the estimates are accurate and vice versa. \( e_{av} \) is taken here (and not \( e_i' \)) because the accuracy of the estimates oscillates over a range of values, as can be seen from the results of Section 5.6.

Similarly for the simple estimator
\[ e''_i = r_i - r''_i \]  \hspace{1cm} (5.7.4)

where \[ r''_i = \sum_{h=0}^{q'} s_{i-h} y_{i-1,h} \]  \hspace{1cm} (5.7.5)

\[ y_{i-1,0} \ldots y_{i-1,q'} \] is the one-step prediction of the estimates of the channel from the simple estimator.

Also let

\[ (e''_{av})^j = \frac{1}{N_a} \sum_{j=1}^{i+95} ||e''_j|| \]  \hspace{1cm} (5.7.6)

As before, \( e''_{av} \) can be taken as a measure of the performance of the simple estimator. Now, if \( e_{av} \) is less than the corresponding \( e''_{av} \), the improved estimator has a better performance and is operating correctly. However, if \( e_{av} \) becomes greater than \( e''_{av} \) the simple estimator has a better performance and should be used in place of the other. The condition \( (e_{av} > e''_{av}) \) occurs when an additional skywave appears. In this way, the errors in the estimates will always be bounded by those of the simple estimator, being always as good as or better.

The switch over to the simple estimator, where estimates of the channel are taken from the latter, continues until the improved estimator finds out the new number of skywaves and correct subspace. In the adaptive estimator this is carried out by means of a new starting-up procedure after detection of the error in the subspace. This starting up procedure assumes that the number of skywaves is unknown at the receiver (Section 5.6.2) and uses the estimates of the channel from the simple estimator rather than from maximum-likelihood estimation, as has been done at the beginning of data transmission. After determining the number of skywaves and forming the subspace, the adaptive estimator switches back to the improved estimator. A flow diagram of the adaptive estimator algorithm is shown in Figure 5.7.4. In this algorithm, the previous (old) subspace could be used in forming the initial new subspace. For example, when there is a change from one to two skywaves
Starting up procedure (Section 5.6.2)

Set $J = 0$, $K = 0.0$

$J = J + 1$

Simple estimator algorithm (Section 5.3.2)

Improved estimator algorithm (Section 5.4) which gives estimates of the channel

Average the error of both estimators ($e'_i$, $e''_i$) recursively

Is $J = 96$

Yes

$e_{av} \leq e''_{av}$

$K = K + 1$

Simple estimator algorithm which gives the estimates of the channel now

Find $\hat{A}$, $\hat{B}$, $\hat{C}$ vectors (eqns. 5.6.29-5.6.31) recursively

Is $K = 3000$

Yes

No

Set $J = K = 0.0$

Find number of skywaves (from $A$, $B$ and $C$ vectors) form initial subspace and set other initial values to start the improved estimator again (Sect. 5.6.2)

Fig. 5.7.4: Adaptive estimator algorithm
or one to three skywaves, the previous one dimensional \( (A_i) \) could be stored. Then this subspace \( (A_i) \) may be used instead of the \( \hat{A} \) vector (equation 5.6.29) in the new starting-up procedure for the adaptive estimator. This is because \( A_i \) will be more accurate than \( \hat{A} \). A similar process could be carried out for the change from two to three skywaves.

### 5.7.3 Results and Analysis of Simulation Tests

The same channel model is used as that described in Section 5.2. The only change in the model here is that, for certain tests, after 13200 data symbols has been sent, one or two skywaves may appear or disappear gradually. The appearance of a skywave is simulated in the model by multiplying the second or third skywave by a linearly increasing function as shown in Figure 5.7.5A. Similarly a skywave may disappear by multiplying the skywave by a linearly decreasing function (as shown in Figure 5.7.5B). As before the errors in the estimates \( \{ \xi_i \} \) are as defined by equation 5.6.27. The delay in detection is 17 and the correct detection of all data symbols is assumed.

The s/n is as defined by equation 4.4.2. In these tests the s/n, before the appearance or disappearance of the skywaves, is 30 dB.

The following possible HF structures have been studied:
1. one skywave all the time
2. one skywave then two skywaves
3. one skywave then three skywaves
4. two skywaves then one skywave
5. two skywaves all the time
6. two skywaves then three skywaves
7. three skywaves then one skywave
8. three skywaves then two skywaves
9. three skywaves all the time

When the number of skywaves is one, two or three, all the time (possibilities 1, 5, 9), a comparison is made between the estimators but, since all the time the improved estimator has a better performance,
Fig. 5.7.5: (A) Appearance of a skywave  
(B) Disappearance of a skywave
no switching is made to the simple estimator, as shown in Figures 5.7.6-5.7.8. Note that in all the tests here the initial starting-up procedure for the estimator uses the method described in Section 5.6.2. Again when a skywave disappears, no switch over is made to the simple estimator (see Figures 5.7.9-5.7.11). Here the improved estimator is always better than the simple estimator. The reason why a disappearance of a skywave does not affect the performance of the improved estimator may be explained as follows: if there are, say, two skywaves and one skywave disappears, the previous subspace is a valid subspace for the remaining skywave although with an extra dimension. However if an additional skywave appears (possibilities 2, 3, 6), a switch over is made to the simple estimator for a predefined interval of 3000 samples. During this time, the estimates of the simple estimator are used by the improved estimator to find out the new number of skywaves and the corresponding subspace, or the adaptive estimator performs a new starting up procedure (Section 5.6.2). The adaptive estimator then switches back to the improved estimator (see Figures 5.7.12-5.7.14). The optimum values of $\Delta$ and $\theta$ for the simple estimator are found to be 0.01 and 0.97 respectively.
Fig. 5.7.6: Error in the estimate of the adaptive estimator vs time for one skywave HF channel, 0.5 Hz frequency spread
Fig. 5.7.7: Error in the estimate of the adaptive estimator vs time for two skywaves, $\tau_1 = 0.8$ msec, 0.5 Hz frequency spread
Fig. 5.7.8: Error in the estimate of the adaptive estimator vs time for three skywaves, $\tau_1 = 0.8$ msec, $\tau_2 = 1.8$ msec, 0.5 Hz frequency spread
Fig. 5.7.9: Error in the estimate of the adaptive estimator vs time, for two skywaves ($r_1 = 0.8$ msec) then one skywave, 0.5 Hz frequency spread
Fig. 5.7.10: Error in the estimate of the adaptive estimator vs time for three skywaves, \( \tau_1 : 0.8 \text{ msec}, \)
\( \tau_2 : 1.8 \text{ msec} \) then two skywaves (\( \tau_1 : 0.8 \text{ msec} \)), 0.5 Hz frequency spread
Fig. 5.7.11: Error in the estimate of the adaptive estimator vs time for three skywaves, ($\tau_1 : 0.8$ msec, $\tau_2 : 1.8$ msec), then one skywave, 0.5 Hz frequency spread
Fig. 5.7.12: Error in the estimate of the adaptive estimator vs time for two skywaves ($\tau_1 : 0.8$ msec) then three skywaves ($\tau_1 : 0.8$ msec), ($\tau_2 : 1.8$ msec), 0.5 Hz frequency spread.
Fig. 5.7.13: Error in the estimate of the adaptive estimator vs time for one skywave, then three skywaves ($\tau_1 : 0.8$ msec, $\tau_2 : 1.8$ msec), 0.5 Hz frequency spread
Fig. 5.7.14: Error in the estimate of the adaptive estimator vs time for one skywave, then two skywaves ($\tau_1 = 0.8$ msec), 0.5 Hz frequency spread
CHAPTER 6

COMBINED, ESTIMATION, ADAPTIVE LINEAR FILTERING AND DETECTION FOR A SERIAL 9.6 KBIT/SEC MODEM

6.1 INTRODUCTION

In Chapter 4 various near-maximum-likelihood detectors have been considered assuming perfect channel estimation and adjustment of the linear filter. In Chapter 5 the estimator has been studied separately assuming correct detection. The adjustment of the adaptive linear filter can be carried out by the iterative algorithm described in reference 86.

Now in this chapter the three basic parts of the detection process, the near-maximum-likelihood detector, the estimator and the adaptive filter are connected together to give the combined system. Computer simulation tests on this combined system are then carried out to examine its tolerance to various effects such as Gaussian noise, impulsive noise, interference from radio signals, and so on.

6.2 MODEL OF THE DATA TRANSMISSION SYSTEM

The model of the data-transmission system (Figure 6.2.1) that is used in the computer simulation tests, the results of which are presented in Section 6.3, is basically the same as that previously described in Chapter 3, but with the following exceptions:

1. It is assumed that the HF radio signal is transmitted via three independent skywaves. The relative transmission delay between the second and first skywave ($\tau_1$) is 0.9 msec and that between the third and first skywave ($\tau_2$) is 2.8 msec. The model of the HF link used in the tests is shown in Figure 6.2.2. Also it is assumed in this model that the signal received over each skywave has the same frequency spread which is 2 Hz.
Fig. 6.2.1: Model of the data transmission system
Fig. 6.2.2: Model of the HF radio link
2. At the receiver, the detection process includes the near-maximum-likelihood detector, the estimator and the adaptive filter.

The near-maximum-likelihood detector is any of the near-maximum-likelihood detectors 2B8, 4B16 or C16 described in Chapter 4. These detectors use the estimates \( \{E_i^1\} \) of the sampled impulse response of the channel and adaptive filter, together with the signal (samples) at the output of the linear adaptive filter \( (v_i) \) and a prior knowledge of the possible values of \( s_i \) to form the detected data symbol \( s_{i-n}^1 \) where \( n \) is the delay in detection. The detector also forms \( s_1^1 \), the early detected data symbol [8,7] of zero delay in detection, which is used in the estimator (Figure 6.2.1).

The channel estimator is the improved channel estimator described in Chapter 5. This estimator uses the received sample \( r_i \) and the detected data symbols \( (s_1^1, s_{i-1}^1, \ldots) \) to form an estimate of the channel sampled impulse response \( Y_{i+1,i} \) needed by the adaptive filter.

The adaptive linear filter uses the estimates of the channel, given by the channel estimator, both to adjust the adaptive filter tap gains and to determine \( E_i^1 \). When this filter is correctly adjusted, the zeros of the \( z \)-transform of the estimate of the sampled impulse response of the channel and filter are derived from the zeros of the \( z \)-transform of estimate of the sampled impulse response of the channel, by replacing all zeros of the latter that lie outside the unit circle in the \( z \)-plane by the complex conjugate of their reciprocals, all remaining zeros being left unchanged. The technique employed here in the adaptive adjustment of the filter [74,86] is essentially a root-finding algorithm that determines in sequence, the roots (zeros) of the \( z \)-transform of the estimate of the sampled impulse response of the channel, that lie outside the unit circle in the \( z \)-plane, using an iterative process. It then uses the knowledge of these roots to determine the tap gains of the linear feedforward transversal filter, and forms an estimate of the sampled impulse response of the channel and filter. The linear filter at all times introduces an orthogonal transformation into the received signal (Appendix C), being constrained
to be an all-pass network. This avoids both noise enhancement and magnification of the error in the estimate of the channel. More detailed explanation of the algorithm for adjusting the linear filter is given in Appendix F.

6.3 RESULTS AND ANALYSIS OF SIMULATION TESTS

Computer simulation tests have been carried out on the combined system, when operating in the 9600 bit/sec serial data transmission system shown in Figure 6.2.1. A complete computer simulator of the system is given in Appendix G.

The s/n is as defined by equation 4.4.2. In all the tests the adaptive filter ahead of the detector has 50 taps and tries to replace the roots of the sampled impulse response of the channel that lie outside the circle of radius 1.05 in the z-plane by their complex conjugate reciprocals [74,86]. The radius is chosen to be 1.05, because it has been found (see Chapter 4) that when the radius is greater than that value (1.1 or 1.2) the performance of the near-maximum-likelihood detectors is significantly degraded. In the channel estimator the input \( r \) is obtained at time \( t = (i+50)T \) by delaying the received sample at the output of the sampler by 50T, as shown in Figure 6.2.1. Furthermore the delay \( n \) in the near-maximum-likelihood detector is arranged to be 32.

Separate computer simulation tests have been carried out to examine the tolerance of the modem to:

1. Gaussian noise
2. Impulsive noise
3. Sudden level and phase changes in the received signal carrier
4. Interference from other radio signals.
6.3.1 Tolerance of the Modem to Additive White Gaussian Noise

In these tests the level of the noise added during data transmission is varied and its effect on the system has been studied by measuring the bit error rate. We are assuming here that initial adjustment of the estimator has been carried out as described in Section 5.6. Also, each test here involves the transmission of 50,000 data symbols over the HF channel. These tests are carried out using different fading sequences, which give different sequences of values of the sampled impulse-response \( \{Y_i\} \). For many fading sequences the combined system performance using any of the near-maximum-likelihood detectors 2B8, 4B16 or C16 has been quite satisfactory. The system gives no errors in the detection of \( s_i \) for a \( s/n \) of 30 dB. However, for a fading sequence where there is a deep fade (i.e. \( Y_i \) becomes very small in certain parts of the sequence of \( \{Y_i\} \)), the system collapses after the fade and gives a high error rate in the following \( \{s_i\} \). This is due to the errors in the \( \{s_i''\} \) over a fairly long period of time (due to the low \( s/n \) over the deep fade) reducing the accuracy of the estimates of the channel, since the estimates of the channel depend on the \( \{s_i''\} \) and the previous estimates of the channel, as can be seen from Figure 6.2.1 and equation 5.4.8. As a result of this inaccuracy in the estimates, the adaptive filter (as described in Section 6.2) will not be properly adjusted and the estimates of the channel and filter, \( \{E'_i\} \) which are used by the detector, will be inaccurate. Consequently, there will be more errors in the \( \{s_i'\} \) even when \( |Y_i| \) becomes high again and so on. The feedback process continues and the system collapses. In fact, after a certain period of time, the estimates from the channel estimator become quite different from the sampled impulse response of the channel. Thus, the system loses the accurate information on the channel, which is important for the correct operation of the detection process.

6.3.1.1 Modem with Retraining

One technique for making the combined system recover from such a collapse may be by providing it again with an accurate estimate of the channel. To achieve this it has been proposed to send a retraining
data sequence approximately every 0.5 seconds, as shown in Figure 6.3.2, and to derive the estimates of the channel from these known retraining sequences at the receiver, using maximum-likelihood estimation. Sending the retraining sequence regularly, as shown in Figure 6.3.2, means that around 10% of the symbol rate (or bit rate) will be used for retraining. The retraining data sequences that are used are the self orthogonal data sequence employed in the starting up procedure. Since the algorithm for the maximum-likelihood estimator, as described in Section 5.5, is not recursive or the present estimate does not depend on the previous estimates, the estimates of the channel from such an estimator during the retraining process will have the same accuracy whether or not there has been a collapse during the previously sent data symbols. When the error rate is low or there is no collapse in the system, these maximum-likelihood estimates are not as accurate as those given by the improved estimator (Chapter 5). Thus, when the estimates given by the maximum-likelihood estimator during retraining are much better than the estimates from the improved estimator, then that can be taken as a sign of modem collapse during the last block of data symbols.

In the combined system whether there has been a collapse or not, the following operations are carried out during the retraining period:

1. Correct detection is assumed for the improved estimator since $s_i^n$ (or the retraining sequence) is now known at the receiver.
2. Orthonormalization of the $A_i$, $B_i$ and $C_i$ vectors of the subspace of the channel estimator is carried out once at the end of the retraining period, but not during the reception of data. It has been found that it is not necessary to do the orthonormalization every sample [59]. This reduces considerably the amount of computation required in the three-skywave estimation algorithm (Section 5.4).
3. At the end of a retraining period, the vectors of the near-maximum-likelihood detector are set to their correct values and the cost of one of these vectors is set to zero while the costs of the others are set to a high value, say 1000 [49].
4. Maximum-likelihood estimates of the channel are found from the retraining sequence and when the first maximum likelihood estimate of the channel is obtained (say at \( i = i_c \)) a comparison is made between the maximum likelihood estimate and the corresponding estimate from the improved estimator, as follows.

In the improved estimator we have, from equation 5.4.6

\[
\frac{r_i}{c} = \frac{g}{r} \sum_{h=0}^{\infty} s_{i_c-h} Y_{i_c, i_c-1, h}
\]

(6.3.1)

where \( Y_{i_c, i_c-1, 0}, Y_{i_c, i_c-1, 1}, \ldots, Y_{i_c, i_c-1, g'} \)

are given by the improved estimator.

\( \{s_{i_c-h}\} \) are here the known retraining data sequence.

Also, from equation 5.4.7

\[
e_{i_c} = \frac{r_i}{c} - \frac{r_i}{c}
\]

(6.3.2)

It can be seen from equation 3.3.6 that if the estimates are accurate, then \( r_i \) will be close to \( r_i \) or \( e_i \) will be small, especially at high \( s/n \). Hence, the magnitude of \( e_i \) here can be taken as a measure of the accuracy of the estimate.

Also let

\[
\frac{r_i}{c} = \frac{g}{r} \sum_{h=0}^{\infty} s_{i_c-h} Y_{i_c, h}
\]

(6.3.3)

where \( Y_{i_c, 0}, Y_{i_c, 1}, \ldots, Y_{i_c, g'} \)

are the estimates from the maximum-likelihood estimator.
and let 
\[ e_i^{''} = r_i^i - r_i^{''} \]  
(6.3.4)

Clearly, when \[ |e_i^{''}| < |e_i^i| \] then the estimate given by the improved estimator is better than the estimate from the maximum-likelihood estimator. However, when \[ |e_i^{''}| \gg |e_i^{''}| \] then the estimates of the improved estimator are worse than that of the maximum-likelihood estimator. This condition \((|e_i^{''}| \gg |e_i^i|))\) can only happen if there has been a collapse in the last block of data symbols. Since, as mentioned before, a prolonged burst of errors in the \(s_i^{''}\) during a collapse makes the estimate from the improved estimator quite inaccurate. Consequently, as can be seen from equations 6.3.1-6.3.2, \(r_i^{'}\) is different from \(r_i\) or \(e_i^{'}\) becomes large. When the combined system in the above test is not collapsing, no more operations are carried out during retraining. However, if there is a collapse, a starting-up procedure, which includes the following operations is carried out.

a) When a collapse is detected at \(i = i_c\) then the parameters \(\alpha_{i_c,i_c-1}^{'}\), \(\beta_{i_c,i_c-1}^{'}\), \(\gamma_{i_c,i_c-1}^{'}\), \(\alpha_{i_c-1,i_c-2}^{'}\), \(\beta_{i_c-1,i_c-2}^{'}\), and \(\gamma_{i_c-1,i_c-2}^{'}\) in the improved estimator are set as follows:

\[
\begin{align*}
\alpha_{i_c,i_c-1}^{'} &= 0.0 \quad (6.3.5) \\
\beta_{i_c,i_c-1}^{'} &= 0.0 \quad (6.3.6) \\
\gamma_{i_c,i_c-1}^{'} &= 0.0 \quad (6.3.7) \\
\alpha_{i_c,i_c-1} &= \alpha_{i_c} \quad (6.3.8) \\
\beta_{i_c,i_c-1} &= \beta_{i_c} \quad (6.3.9) \\
\gamma_{i_c,i_c-1} &= \gamma_{i_c} \quad (6.3.10)
\end{align*}
\]

and \(Y_{i_c,i_c-1}^{'} = Y_{i_c}^{''}\)  
(6.3.11)
This is similar to what is done in the starting up procedure of Section 5.6. The values are set as shown, since due to the collapse they become quite different from their correct values. The subspace used in the improved estimator is taken as that used before the collapse. The subspace will not have been significantly affected by the collapse because of the small value of \( \eta \).

b) The initial adjustment of the adaptive linear filter is given in Table F 1.1 and not from the previous starting points (Appendix F).

Computer simulation tests on the combined system using one of the bad fading sequences, with s/n = 30 and using detector 2B8 or C16 shows that:

1. The combined system without retraining after the deep fade, the system collapses and never recovers as illustrated in Figure 6.3.3A.

2. The combined system with retraining after the deep fade, the system collapses but then it recovers after the next retraining process and there are no further errors in the \( \{s_i\} \) for the remainder of the test, as shown in Figure 6.3.3B.

In all the following tests, it is assumed that a retraining is used to the modem.

6.3.2 Performance of the Modem with Retraining

In order to obtain the graph of bit error rate vs s/n (the performance of the modem), the tests may be carried out either with a fixed fading sequence and different s/n or different fading sequences for different s/n. It is best to carry out the tests with the worst fading sequence, which has the deepest fade, where \( |Y_i| \) becomes very small, and with different s/n. In addition to the worst sequence, some typical fading sequences are needed to check the performance of the system under more typical conditions.
FIGURE 6.3.2: Retraining of the modem

(A) deep fade

(B) deep fade

FIGURE 6.3.3: A. modem without retraining
B. modem with retraining
A search has been carried out using computer simulation to find such fading sequences. After testing 100 fading sequences each with 50,000 \( \{Y_i\} \) (to give a total of 5 million \( \{Y_i\} \)) a possible worst sequence has been found. The graph of \( |Y_i|^2 \) vs \( i \) for that sequence is shown in Figure 6.3.4. To find a typical sequence, the above 100 sequences have been arranged in order of increasing values of the minimum \( |Y_i|^2 \), and the sequences in the middle of the resulting list are taken to be typical sequences. Two of such typical sequences are shown in Figures 6.3.5-6.3.6 and are called fade 1 and fade 2. Note the minimum values of \( |Y_i|^2 \) for the typical sequences are greater than that for the worst sequence.

The bit error rate vs s/n curves for the combined system with retraining are given in Figures 6.3.7-6.3.8. In plotting each of these graphs around 2 million data symbols have been sent. In any two measurements, different random number sequences are used in the generation of the data symbols \( \{s_i\} \) and the Gaussian noise. This minimizes the influence of the choice of the random number sequence on the results. The adaptive adjustment of the linear filter in the combined system is carried out once every 4 samples. It is found that this does not degrade the system bit error rate performance but considerably reduces the amount of computation required. This is because the channel is varying quite slowly with time, that \( Y_i \) does not change significantly over 4 samples.

The conclusions that can be derived from these results are as follows:

1. System C16 has the best performance over all fading sequences, and this results agrees with that of Chapter 4.
2. System 4B16 gains no advantage in performance over system 2B8 and it is more complex to implement (Table 4.4.1).
3. System 2B8 has a performance as good as that of system C16 except for the worst fading sequence, where there is a loss of around 0.5 dB in performance. However, system 2B8 can be very simply implemented (Table 4.4.1), so it appears that system 2B8 is the most cost effective near-maximum-likelihood detector.
Fig. 6.3.5: $|y_i|^2$ vs $i$ for a typical fading sequence (fade 1)
Fig. 6.3.7: Bit error rate vs S/N for the combined system (worst fading sequence)
Fig. 6.3.8: Bit error rate vs S/N for the combined system (typical fading sequences)
4. For the worst fading sequence, as can be seen from Figure 6.3.7, there are errors in all systems even at high s/n. This is because any small noise during the deep fade will result in the system collapse due to the low signal level here. Therefore a very high s/n is required to reduce the error rate to a low value.

5. For the typical fading sequences, fade 1 and fade 2, which have approximately the same deep fade (minimum $|y_i|^2$), the performance with sequence fade 2 is worse than that with fade 1 (Figure 6.3.8). This is because, for fade 2, during the deep fade there is also a rapid channel variation (Figure 6.3.6), which degrades the performance of the estimator in that region. Hence, there is an increased chance of collapse here.

Finally, the results with the typical fading sequences apply, of course, for three skywaves with rapid channel variations (2 Hz frequency spread), as described in Section 6.2, and do not therefore apply to a typical HF channel condition.

6.3.3 Tolerance of the Modem to Impulsive Noise

Impulsive noise in an HF link is usually man-made interference and therefore only becomes really important in built up areas or where the radio receiving equipment is close to a source of electrical interference [3]. The tolerance to impulsive noise of the combined system with retraining is studied in this section. Tests have been carried out with a typical fading sequence (fade 1) and a s/n of 30 dB. After a period of correct detection by the combined system using detector C16 it is assumed that there has been impulsive noise in the form of a burst of high level additive noise. The impulsive noise has been simulated by replacing the received samples $\{r_i\}$ by $k_i$ for the duration of the burst where

$$k_i = k_{r,i} + j k_{m,i} \quad (6.3.12)$$
\( K_{r,i} \) is equally likely to be +7.5 or -7.5
\( K_{m,i} \) is equally likely to be +7.5 or -7.5
and \( |K_i| > \text{largest value of } |r_i| \)
\( |K_{r,i}| > \text{largest } |\text{real } r_i| \)
\( |K_{m,i}| > \text{largest } |\text{imaginary } r_i| \)

\( K_{r,i} \) and \( K_{m,i} \) are found by means of a computer-simulation test. It
has been found that the system collapses after the burst and then
recovers after retraining. To improve the system performance, as far
as impulsive noise is concerned, the estimator has been modified in
such a way that when it detects the presence of impulsive noise, the
estimator stops updating the estimates (using equation 5.4.8) and uses
only predictions from the previous estimates of the channel, and this
continues until the end of the impulsive noise. The estimator can
detect the presence of the impulsive noise from the saturation of the
analogue to digital converter, that gives the received sample \( r_i \) at
the receiver. The idea behind using prediction only, without
updating, is that by that means we are preventing the estimates from
being affected by the errors in the \( \{s_i^n\} \) due to the impulsive noise.
It has been found that by implementing the above technique the system
can withstand impulsive noise of duration 5-20 msec (10-50 data
symbols) without any collapse. If the bursts are of longer durations
then the system collapses and recovers after the next retraining
process. The typical duration of impulsive noise for the 9600 bit/sec
system using telephone channels is of the order of 20 msec [3].

For slower channel variations (less than the 2 Hz which is assumed
here), the system is expected to tolerate longer duration bursts
without collapse, because the prediction of \( Y_i \) is now better.

6.3.4 Tolerance of the Modem to Sudden Level and Phase Changes in the
Received Signal Carrier

Now for the same channel and combined system conditions studied
previously in Section 6.3.3 and after correct start-up (no errors in
the \( \{s_i\} \)), the level of the signal carrier is changed suddenly. This
is simulated by scaling the transmitted carrier (or equivalently the data symbols) by a constant. It has been found that for a sudden level change of 3 dB or less no collapse occurs due to such level change but for higher level changes the system collapses then recovers after the retraining, as shown in Figure 6.3.9. For a sudden phase change in the received carrier, it has been found that if the phase change is less than or equal to 20° the system does not collapse (no errors in the \( s'_i \)) but for greater phase changes it collapses and then recovers after retraining, as shown in Figure 6.3.10.

6.3.5 Tolerance of the Modem to Interference from Other Radio Signals

One of the problems in the transmission of digital signals over HF radio links is interference from other radio signals whose spectra overlap that of the wanted signal [3]. The tolerance of the modem to an interfering radio signal can conveniently be assessed by measuring its tolerance to an interfering tone whose frequency is adjusted, in turn, to different points within the signal frequency band [3].

Thus, for the same conditions as assumed in Section 6.3.3, a sampled sinusoidal wave of frequency 1000 Hz or 2000 Hz is added to the received samples \( \{ r_i \} \). These frequencies are chosen because they will be within the band of the output lowpass filter at the receiver.

\[
\text{Hence } r_i'' = r_i + a''e^{j(2\pi f_1 iT)} \tag{6.3.13}
\]

where \( a'' \) is a constant, \( f_1 \) is 1000 or 2000 Hz.

It has been found that when the amplitude \( a'' \) is small \( (a'' < 0.1) \), the interference has no effect on the error rate of the system. However when the amplitude is greater \( (a'' = 0.2) \) it collapses at certain intervals of time and then recovers later. In fact, it collapses when \( |Y_i| \) becomes small, because of the added interference. Finally it has
Fig. 6.3.9: Sudden level change in the carrier level (> 3 dB)

\[ r_i = |r_i| e^{j\psi_i} \]

\[ r_{in} = |r_i| e^{j(\psi + \theta_i)} \]

Fig. 6.3.10: Sudden phase change in the carrier phase (> 20°)
been found that when \( a'' = 1.0 \) it collapses all the time for obvious reasons.

6.4 SIGNAL QUALITY MEASURES

It will be useful for the receiver to know whether there has been a collapse or not in the previously sent block of data symbols. Two techniques of detecting the collapse are suggested here:

METHOD A

By comparing the estimate of the channel given by maximum-likelihood estimation (found during retraining) with the estimate of the channel from the improved estimator used in the system, a collapse can be detected as was explained before in Section 6.3.1. The method has been tested using computer simulation and showed a satisfactory performance.

METHOD B

In the improved estimator, from equation 5.4.7:

\[
e_i' = r_i - r_i
\]  \hspace{1cm} (6.4.1)

where, from equation 3.3.6

\[
r_i = \sum_{h=0}^{g} s_{i-h} Y_{i,h} + w_i
\]  \hspace{1cm} (6.4.2)

\( r_i \) may be reconstructed by convolving the detected data-symbols with the estimated impulse response \( Y_{i,i-1} \)

\[
r_i' = \sum_{h=0}^{g'} s_{i-h} Y_{i,i-1,h}
\]  \hspace{1cm} (6.4.3)
When there is a collapse there are errors in the $\{s'_i\}$ and $Y'_{i,i-1}$ is inaccurate, so here $r_i'$ is quite different from $r_i$, which makes the average $\|e'_i\|$, over a short period of time, quite large (greater than a certain threshold). However, during correct detection $r_i'$ is close to $r_i$, so $\|e'_i\|$ is small.

A large number of tests have been carried out, on method B. Almost all the time when there is a collapse, the average $\|e'_i\|$, for the last 200 data symbols of each block of data symbols, has been large ($> 0.5$). This may be used as a measure of signal quality, that is, as an indication of collapse in the combined system.

6.5 NON-MINIMUM PHASE FILTERS

In all the previous work, it has been assumed that the impulse response of the linear baseband channel has been adjusted to be minimum phase in the absence of multipath propagation [59] and the transmitter and receiver filters sampled impulse response are as given in Table 3.3.1. We now consider the combined system with retraining, where this adjustment has not been carried out and the sampled impulse response of the filters are as given in Table 6.5.1 [7]. Assume the fading sequence to be the typical fading sequence (fade 1) and detector 2B8 is used. The performances of the combined system with minimum and non-minimum phase filters are now as shown in Figure 6.3.11. As it can be seen from this figure, there is a loss of around 1 dB in tolerance to noise if the filters are not adjusted to be minimum phase.
TABLE 6.5.1: THE NON-MINIMUM PHASE SAMPLED IMPULSE-RESPONSES OF THE TRANSMITTER AND RECEIVER FILTERS SAMPLED AT 4800 SAMPLES/SECOND

<table>
<thead>
<tr>
<th>Sampled impulse-response of the transmitter filter</th>
<th>Sampled impulse-response of the receiver filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real Part</td>
<td>Imaginary Part</td>
</tr>
<tr>
<td>----------</td>
<td>----------------</td>
</tr>
<tr>
<td>-0.0382</td>
<td>0.5007</td>
</tr>
<tr>
<td>0.2457</td>
<td>-1.3823</td>
</tr>
<tr>
<td>0.0557</td>
<td>-2.3229</td>
</tr>
<tr>
<td>4.3052</td>
<td>-1.6233</td>
</tr>
<tr>
<td>22.3184</td>
<td>2.3123</td>
</tr>
<tr>
<td>44.8006</td>
<td>7.0695</td>
</tr>
<tr>
<td>42.0069</td>
<td>5.2753</td>
</tr>
<tr>
<td>9.6632</td>
<td>-2.7801</td>
</tr>
<tr>
<td>-13.7143</td>
<td>-5.0775</td>
</tr>
<tr>
<td>-7.1940</td>
<td>-0.3151</td>
</tr>
<tr>
<td>4.7637</td>
<td>0.6568</td>
</tr>
<tr>
<td>2.6054</td>
<td>-1.3191</td>
</tr>
<tr>
<td>-1.9526</td>
<td>0.0767</td>
</tr>
<tr>
<td>-0.2429</td>
<td>1.0017</td>
</tr>
<tr>
<td>0.7289</td>
<td>-0.5064</td>
</tr>
<tr>
<td>-0.4394</td>
<td>-0.3751</td>
</tr>
</tbody>
</table>
Fig. 6.3.11: Bit error rate vs S/N for the combined system using minimum and non-minimum phase filters.
CHAPTER 7

MODEM SYNCHRONIZATION

7.1 INTRODUCTION

In earlier chapters, the modem functions of detection and channel estimation have been studied. However in order to complete a practical modem, other modem functions must be performed. These are carrier phase tracking and symbol timing tracking. The problems and solutions associated with each of these are considered in Sections 7.2 and 7.3, together with the results of simulation tests. In this Chapter it is assumed that the model of the data transmission system is the same as that described in Section 3.3.

7.2 TIME SYNCHRONIZATION OF AN HF MODEM

7.2.1 Basic Algorithm [87]

Previously it has been assumed that the receiver sampling (timing) of the received signal has been performed at precisely the baud rate of the transmitted signal. However, this situation does not normally exist in practice because of the slight offset from the nominal baud rate which will exist in both transmitter and receiver timing clocks.

The existence of timing offset in the receiver must be avoided and the receiver must somehow 'learn' the baud rate of the received signal or adjust the sampling rate to its correct value. A simple algorithm, suitable for HF channels, for adjusting the sampling rate at the receiver is now described. Suppose that, at time t=iT, the estimate of the sampled impulse response of the HF channel is

\[ y_i' = [y_{i,0}^' \ y_{i,1}^' \ ... \ y_{i,g}^'] \]  

(7.2.1)
and the corresponding estimate at time $t=(i-1)T$ is

$$Y_{i-1} = [Y_{i-1,0} \ Y_{i-1,1} \ \cdots \ Y_{i-1,g}]$$  \hspace{1cm} (7.2.2)

and let

$$\phi_i = \text{Re} \left[ \sum_{h=0}^{g} (y_i^h - y_{i-1}^h)(y_i^{h+1} - y_{i,h-1}^*) \right]$$  \hspace{1cm} (7.2.3)

where $y_i^h = y_{i-1}^h = 0.0$ for $h < 0$ and $h > g$.

Then the sampling interval $T_{i+1}$ is

$$T_{i+1} = T_i - b \phi_i$$  \hspace{1cm} (7.2.4)

where $b$ is a constant, and hence the sampling instant $\theta_{i+1}$ is

$$\theta_{i+1} = \theta_i + T_i - b \phi_i$$  \hspace{1cm} (7.2.5)

(see Figure 7.2.1).

7.2.2 Results and Analysis of Computer Simulation Tests

Tests have been carried out on the above algorithm, using the correct channel impulse response (or assuming perfect channel estimation), since inaccuracies in the sampled impulse response, $Y_i$, produced by the channel estimator do not significantly affect the performance of the algorithm [87]. All the relevant parameters for each channel used in the tests are summarised in Table 7.2.1. $\tau_1$ is here the relative delay in transmission between the first and second skywave and $\tau_2$ is that between the first and third skywave. In this table, the different fading sequences 1, 2, 3 and 4 give different sequences of the values...
Fig. 7.2.1: Sampling instants in equation 7.2.5
TABLE 7.2.1: Parameters of the channels used in the test

<table>
<thead>
<tr>
<th>Channel</th>
<th>Number of Skywaves</th>
<th>τ₁ (ms)</th>
<th>τ₂ (ms)</th>
<th>Frequency Spread</th>
<th>Fading Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>2.8</td>
<td>-</td>
<td>2 Hz</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>2.8</td>
<td>-</td>
<td>2 Hz</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>0.9</td>
<td>2.8</td>
<td>2 Hz</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>T</td>
<td>2.8</td>
<td>2 Hz</td>
<td>4</td>
</tr>
</tbody>
</table>
of the sampled impulse response $Y_i$. In these tests (Figures 7.2.2-7.2.5), an error in the timing frequency is introduced at the receiver and the ability of the timing loop to track the transmitter timing is determined. Each computer simulation test involves a total of 52800 sampling instants. For $i=1, 2 \ldots 2400$ the sampling instants are steadily advanced relative to their correct instants $iT$ by setting $T_i = T + 10^{-4}T$, and during this time the timing loop is 'open' (inoperative) and so the timing phase error $(\theta_i - iT)$, increases steadily at a rate determined by the timing frequency offset (which is $10^{-4}T$). The parameter $t_s$ in these figures is the fractional error in the sampling instant given by

$$t_s = \frac{\theta_i - iT}{T} \quad (7.2.6)$$

$T$ is the correct data-symbol interval (i.e. $T = 1/2400$).

Ideally here, $t_s$ should remain constant as $i$ increases from 2400 to 52800, the value of $t_s$ here being that obtained at $i = 2400$, when the algorithm is switched into operation. The actual value at which $t_s$ remains constant is unimportant. In practice, the value of $t_s$ should, if possible, remain within the range $-1$ to $+1$ relative to its value at $i = 2400$ and should vary only slowly with $i$ [87].

Tests show that the above algorithm performs well under many channel conditions (Figures 7.2.2-7.2.4 and the results in Ref 87). However for certain channel conditions the algorithm breaks down, as shown in Figure 7.2.5. To overcome this problem it has been proposed to limit the sampling interval $T_{i+1}$ using prior knowledge that the sampling interval will be within a given range of $T$. The operation of the limiter is shown in Figure 7.2.6. $T_{i+1}$ is the sampling interval found from equation 7.2.4, $T_{i+1}'$ is the sampling interval after applying the limiter. It is assumed in Figure 7.2.6 that a clock stability of $1$ in $10^4$ can be achieved at the transmitter [88].
Fig. 7.2.6: Limiter operation

\[
\begin{align*}
T + 10^{-4}T & \\
T & \\
T - 10^{-4}T & \\
T_{i+1} & \\
T_{i+1} & \\
\end{align*}
\]
The method worked very well and it prevented the algorithm from becoming unstable for the given channel conditions (as can be seen from Figure 7.2.7).

7.2.3 Arithmetic Accuracy in Element Synchronization Algorithm

An important aspect of the use of a digital computer to simulate a system, is the accuracy to which a digital computer operates. Usually, digital computers represent variables to a high degree of precision. However, if the system were to be built, it would operate, in practice, with a much lower precision than is assumed by the computer simulation model, possibly giving rise to falsely optimistic results. Thus, if a system were to be built in practice, the appropriate numerical precision used in the system must be determined.

Research carried out by Hau [86], Harvey [43] and Harun [59] showed that 16 bits accuracy is required for the proper operation of the adaptive filter, near-maximum-likelihood detector, and the improved channel estimator in the modem.

The method that has been used in this section to find the numerical precision required in the timing algorithm, is the one developed by Hau [86], which takes each number and quantizes it, and represents it by 16 bits (or any other appropriate number of bits) and then all the operations (additions, subtractions, divisions and multiplications) are carried out with 16 bits arithmetic. This is exactly the situation in a practical modem. The algorithm for timing as described in Section 7.2.1, has essentially two parts. Firstly $\Phi_i$ is found using equation 7.2.3, secondly this $\Phi_i$ is used to update the sampling clock (equation 7.2.4). Using the above method to study the effects of limited arithmetic accuracy, it has been shown to be quite satisfactory to use 16 bits accuracy in the first step (which contains most of the computations), but 32 bits are needed in the second step (because of the small value of the constant $b$). For these tests the element rate $1/T$ in the algorithm is scaled to be 1.0.
Fig. 7.2.7: Performance of the timing algorithm with a limiter, channel 4, $b = 10^{-3}T$
As can be seen from Figures 7.2.8 and 7.2.9 the performance with limited arithmetic accuracy under the assumed conditions is quite good (|t_s| < 1), but it is not as good as the performance with full computer accuracy under the same conditions (Figures 7.2.2-7.2.3). This is because the value of \( y'_i, h \) is very close to \( y_i-1, h \), so that \( (y'_i, h - y_i-1, h) \) in equation 7.2.3 is too small to be represented accurately enough by 16 bits. To improve the results the following methods have been proposed.

**METHOD A:**

In this method \( y'_i, h - y_i-1, h \) in equation 7.2.3 is replaced by \( y'_i, h - y_i-m, h \) where \( m > 1 \). Thus, here equation 7.2.3 becomes:

\[
\phi'_i = \text{Re} \left[ \sum_{h=0}^{0} (y'_i, h - y_{i-m, h})(y'_i, h+1 - y_{i, h-1}) \right] \tag{7.2.7}
\]

Also equation 7.2.4 is changed to:

\[
T_{i+1} = T_i - \frac{b}{m} \phi'_i \tag{7.2.8}
\]

By taking \( y'_i, h \) and \( y_{i-m, h} \) the difference between \( y'_i, h \) and \( y_{i-m, h} \) will become greater because of the greater separation in time between them. The reason that we can replace \( y'_i, h - y_i-1, h \) by \( y'_i, h - y_i-m, h \) is that the slope of the line \( y'_i, h \) vs \( i \) is required in the algorithm. Clearly, both \( y'_i, h - y_i-1, h \) and \( (y'_i, h - y_i-m, h)/m \) can give that slope provided that \( m \) is not too large. Notice the division by the constant \( m \) here is carried out in equation 7.2.8. Figures 7.2.10-7.2.15 show the results when this modified algorithm is used. In all these tests, as before, the exact impulse response has been used. Furthermore 16 bits are used in equation 7.2.7, 32 bits are used in equation 7.2.8 and \( T \) is scaled to 1.0. As can be seen from Figures 7.2.10-7.2.15, the results are now very near to those with computer accuracy (Figures 7.2.2-7.2.3).
Fig. 7.2.8: Performance of the timing algorithm with limited arithmetic accuracy, channel 1, $b = 10^{-3} \tau$.
Fig. 7.2.9: Performance of the timing algorithm with limited arithmetic accuracy, channel 2, $b = 10^{-3} T$
Fig. 7.2.10: Performance of the timing algorithm with limited arithmetic accuracy, method $A$, $m = 8$, channel 1, $b = 10^{-3} T$
Fig. 7.2.11: Performance of the timing algorithm with limited arithmetic accuracy, method A, m = 16, channel 1, \( b = 10^{-3} T \)
Fig. 7.2.12: Performance of the timing algorithm with limited arithmetic accuracy, method A, $m = 32$, channel 1, $b = 10^{-3}T$. 
Fig. 7.2.13: Performance of the timing algorithm with limited arithmetic accuracy, method A, $m = B$, channel 2, $b = 10^{-3}T$. 
Fig. 7.2.14: Performance of the timing algorithm with limited arithmetic accuracy, method A, \( m = 16 \), channel 2, \( b = 10^{-3}T \).
Fig. 7.2.15: Performance of the timing algorithm with limited arithmetic accuracy, method A, m = 32,
Also it can be seen from the results the optimum value of \( m \) is 16. Increasing \( m \) (>16) does not improve the results, but it increases the amount of storage required. Decreasing \( m \) degrades the results.

**METHOD B**

Figure 7.2.16A shows a typical variation with time of a component of the sampled impulse response (the real part of the component) for a short period of time using all computer accuracy. However, if limited arithmetic accuracy is used (say 16 bits), then this component variation may appear as shown in Figure 7.2.16B, where each component is quantized. \( \text{Re}(y_{i,h}) - \text{Re}(y_{i-1,h}) \) can be obtained more accurately if a least-squares straight-line fit is applied to the quantized values and the 16 bits accuracy is assigned to the range of values \( y_{\text{max}} \) to \( y_{\text{min}} \) (Figure 7.2.16B). Clearly, in Figure 7.2.16B (\( x_2-x_1 \)) or the slope of the line in 16 bits will be very close to \( \text{Re}(y_{i,h}) - \text{Re}(y_{i-1,h}) \) for the full computer accuracy. The same method as above can be used to find \( \text{Im}(y_{i,h}) - \text{Im}(y_{i-1,h}) \) more accurately using 16 bits accuracy. This method has been tested and it gives the results shown in Figure 7.2.17, which is close to the corresponding result with the full computer accuracy (Figure 7.2.2). The disadvantage of this method is that it involves a considerable amount of computation. Furthermore it does not give any advantage over method A in performance. Thus, the simple method A is the preferred one.

Finally, Figure 7.2.18 shows the results of a test using method A but without using the limiting of \( T_{i+1} \). As before, for such conditions the algorithm breaks down. However, if the above test is repeated with the limiting of \( T_{i+1} \) (as described in Section 7.2.2), the algorithm operates correctly, as shown in Figure 7.2.19.

The above results suggest that, with limited arithmetic accuracy, method A with limiting of \( T_{i+1} \) should be used.
Fig. 7.2.16: Component variation of the impulse response with time (real part)
A - all computer accuracy
B - limited arithmetic accuracy
Fig. 7.2.18: Performance of the timing algorithm with limited arithmetic accuracy, method A, $m = 16$, channel 4
Fig. 7.2.19: Performance of the timing algorithm with limited arithmetic accuracy, method A, \( m = 16 \), limiting \( T_{i+1} \), channel 4
7.3 CARRIER PHASE TRACKING

Until now it has been assumed that transmitter and receiver modulation and demodulation frequencies are exactly equal. However, in practice, some discrepancy will always exist between the transmitter modulating carrier and receiver demodulating carrier, and frequency offsets of several Hz may be possible [89,7]. Furthermore, the HF link can introduce a Doppler shift into the signal giving a corresponding frequency offset (Section 2.3.3). Carrier phase tracking may be divided into:

a) Carrier phase tracking at the start of transmission
b) Carrier phase tracking during data transmission

A different technique is used at the start from that during data transmission, and the reason for using different techniques will become clear later on.

7.3.1 Carrier phase Tracking at the Start of Transmission

The carrier frequency of the received signal can be obtained by transmitting a burst of the carrier frequency at the start of transmission. At the receiver, the received signal \( \hat{r}(t) \) is sampled at 4800 samples/sec to give the received samples \( \{ \hat{r}_i \} \). These samples \( \{ \hat{r}_i \} \) are then applied to a digital phase locked loop (DPLL) as shown in Figure 7.3.1 [90,89]. Finally, at the end of this burst, the output of the digital voltage controlled oscillator (Figure 7.3.1) is taken to have the correct frequency and is used for the subsequent demodulation process. Here the centre frequency of the digital voltage controlled oscillator is adjusted to be the nominal value of the carrier frequency (i.e. 1800 Hz) at the start.

7.3.1.1 Results of computer simulation tests

In these tests the HF channel model is assumed to have 3 skywaves with 2 Hz frequency spread. The model of the HF link is as shown in Figure
6.2.2. The digital phase locked loop in Figure 7.3.1 is used to generate the reference carrier used for demodulating the received bandpass signal. The digital lowpass filter in the digital phase locked loop is arranged to have a wide bandwidth at the start to track the frequency of the incoming signal rapidly, and a narrow bandwidth at the end of the tracking process to reduce noise. This is achieved by making $\beta_i^0$ in Figure 7.3.2 variable [89] and

$$\beta_i^0 = \beta_{i-1}^0 - \Delta (\beta_{i-1}^0 - \beta_{a_i}^0) \tag{7.3.1}$$

$\beta_{o_i}$, $\beta_{a_i}$ are the respective start and end points of $\beta_i^0$ with $\Delta$ controlling the rate of decay of $\beta_i^0$. Also, the value of $\alpha_i$ is made equal to $(\beta_{1/2}^0 / 2.0)$.

It has been found by computer simulation that the digital phase locked loop can lock to the frequency of the transmitter oscillator or converge to that frequency within around half a second (2400 samples). Hence it is necessary at the transmitter to send the carrier burst for the duration of 0.5 sec. Furthermore, it has been found that the system can tolerate a frequency offset of $\pm 10$ Hz around the 1800 Hz value.

7.3.2 Carrier Phase Tracking During Data Transmission

During data transmission, in the case of an HF link, the presence of any frequency offset adds to the rate of change caused by fading in the components of $Y_i$ [7]. Therefore, if no means of removing the frequency offset is employed in the receiver, then it is necessary for the channel estimator (in the combined system of Chapter 6), to be able to track much faster channel variations. This would be an acceptable solution but for the fact that the performance of the channel estimator degrades rapidly with an increasing rate of channel variation. This degradation in performance is due to the wider bandwidth required by the estimator to track the fast channel
Fig. 7.3.1: Digital phase locked loop

Fig. 7.3.2: Linear digital filter
variations, which results in it having poor rejection of additive noise on the received signal.

The proper solution to the problem is to employ a phase tracking loop at the receiver which removes the frequency offset from the received signal \( r_i \). Obviously a digital phase locked loop cannot be employed here as has been used at the start, because there is simply no carrier in the received QAM signal now [89].

Systems which estimate and track the received carrier phase from such received signals are known as suppressed carrier tracking loops. These loops can be broadly categorised into [91,92,93]:

a) Non data aided (e.g. squaring loops, Costas loops)
b) Data aided.

It has been found that for a QAM signal, when there is intersymbol interference, the data aided system gives a very substantial improvement over the non data aided [89].

A block diagram for a data aided carrier phase tracking loop is shown in Figure 7.3.3. A more detailed diagram of the carrier tracking loop is given in Figure 7.3.4. This system is taken from Ref. 89 with some modifications, to make it operate correctly with a 16 point QAM signal.

The changes that have been made are:

1. After taking the imaginary part of the signal \( e^{-j\theta_{i-1}} r_{i-1} r_{i-1}^* \) in Figure 7.3.4. It is now divided by \( |r_{i-1}| |r_{i-1}^*| \) (both \( r_{i-1} \) and \( r_{i-1}^* \) are known at the receiver). Now, the input to the digital filter is only proportional to the phase error as is required in such carrier tracking loops [89].
2. No limiter is used before the digital filter here because it has been found from computer simulation tests that it degrades the performance of the phase tracking loop.
Fig. 7.3.3: Block diagram of carrier phase tracking loop within the receiver
Fig. 7.3.4: Carrier phase tracking loop.
The above data aided loop cannot be used to synchronise the receiver at the start of transmission because the estimator is not yet operational.

### 7.3.2.1 Operation of the loop

The operation of the loop can be explained by means of some simple examples. Firstly, assume that the received signal has a very high s/n and no frequency offset exists, so that no phase correction is required by the phase tracking loop. This means that the signal $r_{i+n}$ at the input to the phase correcting second demodulator must equal the signal $r_{i+n}$ at the output of this demodulator. Thus the phase tracking loop must produce no correction. This means $\hat{\theta}_{i-1}$ in the loop must be zero. Now, if these conclusions are true, the received signal $r_{i-1}$ must equal its estimate obtained from the channel estimator $r'_{i-1}$. Therefore, the updating error signal, given by the imaginary part of $(e^{-j\hat{\theta}_{i-1}} r_{i-1} r^*_{i-1})$ is zero, because $\hat{\theta}_{i-1}$ is zero, and the product $r_{i-1} r^*_{i-1}$ is real valued. This is true for all $i$, that is no phase correction is applied to the received samples $\{r_{i+n}\}$, so that the loop is stable and performing correctly. Secondly, assume the conditions of the first example except that a frequency offset of $\Delta f$ exists, such that the received signal samples are being rotated by an amount of $\Delta \phi$ per symbol period ($\Delta \phi = 2\pi \Delta f T$) due to this offset. Again, assuming the phase tracking loop is operating correctly, then the two outputs of the loop appear as shown in Figure 7.3.5. If the frequency offset was initiated at time zero, the relation

$$r_{i-1} = r'_{i-1} e^{j((i-1)\Delta \phi)}$$

holds. It is interesting now to consider the updating error signal to the filter

$$\text{Im}(e^{-j\hat{\theta}_{i-1}} r_{i-1} r^*_{i-1}) = \text{Im}(e^{-j\Delta \phi(i-1)} R'_{i-1} e^{j\Delta \phi(i-1)}) = \text{Im}(R'_{i-1}) = 0$$
Fig. 7.3.5: Variation with time of the two outputs of the phase tracking loop
Again as was shown in the first example, the error signal in the phase tracking loop is zero. This therefore means that the output of the integrator $x$ must be the value $\Delta \phi$ in order for the two outputs of the loop to appear as in Figure 7.3.5 that is both increasing steadily at a rate of $\Delta \phi$ per symbol period to match the frequency offset present on the received signal.

### 7.3.2.2 Computer Simulation Tests

In these tests, the combined system of adaptive linear filtering, channel estimation and near-maximum-likelihood detection, as described in Chapter 6, together with the carrier tracking loop has been simulated. The near-maximum-likelihood detector used here is system 2B8. The system has been tested here over an HF channel with 3 skywaves, a 2 Hz frequency spread, $s/n = 30$ dB and a typical fading sequence (fade 1). The model of the HF link is shown in Figure 6.2.2. Furthermore, these tests involve the transmission of 50,000 data symbols. For no frequency offset between transmitter and receiver carriers, the above conditions give no errors in the detected data symbols (Chapter 6). It has next been assumed that, after a correct start-up (no frequency offset), there has been a Doppler shift (frequency offset) of $f_{os}$, as shown in Figure 7.3.6. In all the following tests the change $f_{os}$ occurs within 1 sec, or the rate of change is $f_{os}$/sec. The following rates of change have been tested:

1) $1$ Hz/sec (or $f_{os} = 1$ Hz in Figure 7.3.6): This rate of change ($1$ Hz/sec) is the typical rate of change in practice [88]. Computer tests show that during and following this change of frequency, that is, over the duration of test, there are no errors in the detected data symbols.

2) $2$ Hz/sec (or $f_{os} = 2$ Hz in Figure 7.3.6): For this rate of change, the combined system collapses once, immediately after the frequency offset has been introduced but then it recovers after the next retraining process and gives no further errors in
the \( s_i \) (Figure 7.3.7). Also, it has been observed from this test that the phase tracking loop is approximately removing the above frequency offset from the received samples after it recovers from the collapse and for the remainder of the duration of the test. The collapse is due to the rapid channel variation after the introduction of the frequency offset and before the convergence of the tracking loop is completed (to remove the frequency offset).

### iii) 3 Hz/sec - 10 Hz/sec

Tests that have been carried out show that for these frequency offsets the system collapses a number of times, then eventually it recovers and gives no further errors in the detected data symbols. Also, it has been found that the number of collapses increases as the frequency offset \( f_{os} \) increases. As before, the loop is finally removing the frequency offset introduced and effectively the system locks on to the new carrier frequency. 10 Hz/sec is described by Ref. 88 as the worst condition possible in practice.

Finally, it has been found that if the phase tracking loop has not been used at all, the system collapses all the time for a 3-10 Hz frequency offset \( f_{os} \). This is because the estimator cannot work alone with such rapid channel variations, without the aid of a phase tracking loop.

The conclusion that can be drawn from the above series of tests is that the results show the necessity of a phase tracking loop in the receiver, if a frequency offset greater than 2 Hz is introduced into the received signal, during data transmission.
Fig. 7.3.6: Frequency offset in the system

Fig. 7.3.7: Errors in $s_i$ when there is a frequency offset of 2 Hz ($f_{os} = 2$ Hz)
CHAPTER 8

CONCLUSIONS

In this thesis various near-maximum-likelihood detectors have been studied for use with a synchronous serial 9600 bit/sec data transmission system. Computer simulation tests have shown that the developed near-maximum-likelihood systems B and C (Chapter 4) achieve a much better compromise between performance and complexity than the previously available near-maximum-likelihood detectors.

Simulation results have also shown that the method D (explained in Section 5.6) is the most cost effective method of forming the initial subspace required for the starting up procedure of the improved estimator. Furthermore, from the results of Section 5.7, it can be seen that when the number of skywaves changes, then for the improved estimator to continue giving accurate estimates of the channel, it should be backed (supported) by a simple feedforward estimator with prediction.

It can be seen from the results of Section 6.3 that if the combined system of channel estimation, adaptive linear filtering and detection is required to operate correctly under severe conditions such as prolonged high level bursts of noise, or prolonged periods of severe flat fading (deep fades), then there should be regular retraining for the modem to recommence tracking the channel and decoding the data. It is also found that it is adequate to use around 10% of the transmitted information for retraining processes. Furthermore, tests have shown that the receiver can detect the presence of long bursts of errors in the \( s_i \) occurring due to the above severe channel conditions, by using the simple methods described in Section 6.4. Also it is clear from the results of Section 6.5 that the impulse response of the linear baseband channel in the absence of multipath propagation should
be adjusted to be minimum phase, otherwise there is a loss of around 1 dB in the tolerance of the modem to additive white Gaussian noise.

Finally there should be a limiter in the element timing algorithm described in reference 87 in order that it continues to operate correctly without any breakdown under the unfavourable conditions described in Section 7.2. Simulation results have shown that most of the computations required in this timing algorithm may be carried out with 16 bits accuracy. Also tests in Section 7.3 show the necessity of a carrier frequency correction loop in the modem for Doppler shifts of greater than 2 Hz.

8.2 POSSIBLE FURTHER INVESTIGATIONS

Useful future investigations might be made into the following areas:

1. In Chapter 4, all the detectors that have been tested are seriously affected when the sampled impulse response of the HF channel and all-pass filter ceases to be minimum phase. There is clearly a need for a near-maximum-likelihood detector that has a better tolerance to the type of signal distortion that occurs here. Sequential detection techniques could be considered for such conditions. Hence, the new approach might be to simplify the adaptive filter (d = 1.1 or 1.2) at the cost of increasing the complexity of the detector.

2. Simple differential coding is used in the HF data transmission system considered this thesis. Further investigations into applying more sophisticated coding would be very useful in improving the system error rate performance.

3. The detailed hardware designs of the more promising systems developed in this thesis could be produced to assess the cost effectiveness of these systems for use in a synchronous serial data transmission system over HF links.
4. Further investigations might be carried out to study the performance of a combined system similar to the one considered in this thesis (i.e. adaptive linear filtering, channel estimation and near-maximum-likelihood detection), but now with a reduced bit rate such as 4.8 kbits/sec or 2.4 bits/sec. In these modems, simpler estimators can be used such as a feedforward estimator with prediction.

5. Further work may be carried out on the problem of synchronization of an HF modem to find more cost effective methods.
APPENDIX A

DERIVATION OF RALEIGH FADING FILTERS

The frequency response of the filter from equation 2.4.1 is given by:

\[ F(f) = e^{-f^2/4} f_{\text{rms}}^2 \]  

(Al.1)

From equation Al.1 the 3 dB cut off frequency of the filter is:

\[ f_c = 1.17741 f_{\text{rms}} \]  

(Al.2)

An alternative expression for \( f_c \) using equation 2.4.2 is:

\[ f_c = 0.588705 f_{\text{sp}} \]  

(Al.3)

It is well known [94] that as the order of a Bessel filter increases, the frequency and impulse response of filter tend toward Gaussian. Therefore a Bessel filter is used. The s-plane transfer function of the Bessel filter is [94]:

\[ H(s) = \frac{d_0}{L} \sum_{k=0}^{L} d_k s^k \]  

(Al.4)

where \( L \) is the order of the filter and

\[ d_k = \frac{(2L-K)!}{2^{L-K} K! (L-K)!} \]  

(Al.5)
As a practical choice, L has been chosen to be 5 so that equation A1.4 becomes:

\[
H(s) = \frac{945}{s^5 + 15s^4 + 105s^3 + 420s^2 + 945s + 945}
\]

(A1.6)

Factorizing the denominator in equation A1.6 yields:

\[
H(s) = \frac{945}{\prod_{i=1}^{5} (s-P_i)}
\]

(A1.7)

where the \(P_i\) are known as the poles of \(H(s)\) and are given by [71]:

\[
P_1 = -3.64674
\]

\[
P_2, P_3 = -3.35196 \pm j1.74266
\]

(A1.8)

\[
P_4, P_5 = -2.32467 \pm j3.57102
\]

where the frequency response of the filter is given by equation A1.7 by substituting \(s\) by \(j\Omega\) i.e.

\[
H(j\Omega) = \frac{945}{\prod_{i=1}^{5} (j\Omega-P_i)}
\]

(A1.9)

where \(\Omega\) is the angular frequency, \(j = \sqrt{-1}\) and the poles are given by equation A1.8. The 3 dB cut off angular frequency is [72]:

\[
\Omega_c = 2.4274 \text{ rad/sec}
\]

(A1.10)
Since two channels with different frequency spreads are required to be simulated, the cut off frequency of the Bessel filter must be changeable to correspond to the different frequency spreads. This is achieved by first introducing a new angular frequency variable $\omega$, such that:

$$\omega = c_o \Omega \quad \text{(A1.11)}$$

where

$$c_o = \frac{\omega c}{\Omega c} = \frac{2\pi f_c}{\pi c} \quad \text{(A1.12)}$$

where $f_c$ is the desired cut off frequency. From equations A1.10 and A1.12:

$$c_o = 2.58844 f_c \quad \text{(A1.13)}$$

Using equation A1.13 to replace $\Omega$ in equation A1.9, we have

$$H(j\omega) = \frac{945}{\pi} \sum_{i=1}^{5} \frac{\omega}{c_o (j\omega - P_i)} \quad \text{(A1.14)}$$

or

$$H(j\omega) = \frac{945 c_o^5}{\pi} \sum_{i=1}^{5} \frac{1}{(j\omega - P_i)} \quad \text{(A1.15)}$$

and substituting the value of $c_o$ from equation A1.13 gives:

$$H(j\omega) = \frac{109805.05 f_c^5}{\pi} \sum_{i=1}^{5} \frac{d_o}{(j\omega - P_i)} = \frac{d_o}{\pi} \sum_{i=1}^{5} \frac{1}{(j\omega - P_i)} \quad \text{(A1.16)}$$
where the \( p_i' \) are the poles of the Bessel filter and is given by

\[
P_i' = c_0 P_i
\]

\[
P_i' = 2.58844 f_c P_i \quad i = 1, 2, \ldots, 5
\] (Al.17)

substituting \( j\omega = s \) equation Al.16 becomes:

\[
H(s) = \sum_{i=1}^{5} \frac{d_o}{(s - P_i')}
\] (Al.18)

The two different values of the frequency spreads are 0.5 and 2 Hz. Using equation Al.13 the corresponding cut off frequencies of the Bessel filter are 0.2943 and 1.1774 Hz, respectively. Thus all the parameters of the Bessel filter have been specified and they are summarized in Table Al.1.

The Bessel filters derived above are all in analogue form. Since they are to be simulated on a digital computer, the digital form of the filters is required. Various methods, such as impulse invariance, bilinear and matched z-transform may be applied [94,70] to digitize the analogue transfer function of equation Al.18. Here the impulse invariance technique is chosen so that the impulse response of the resulting digital filter is a sampled version of the impulse response of the analogue filter [94]. Using this technique the poles \( \{P_i'\} \) in the s-plane are transformed to poles at \( (e^{P_i'T}) \) in the z-plane [94] where \( T \) is the sampling interval. Thus the transfer function given by equation Al.18 becomes:

\[
H(z) = \frac{K}{\sum_{i=1}^{5} \frac{d_o}{(1 - e^{P_i'T} z^{-1})}}
\] (Al.19)

or

\[
H(z) = \frac{K}{\sum_{i=1}^{5} \frac{d_o}{(1 - q_i z^{-1})}}
\] (Al.20)
so

\[ q_i = e^{p'_iT} \]  \hspace{1cm} (A1.21)

\[ q_i - \text{z-plane poles.} \]

\[ p'_i - \text{s-plane poles} \]

\[ T - \text{sampling interval} \]

\[ T = 20 \text{ msec (50 Hz sampling rate)} \]

The z-plane poles have been calculated using the above transform and are given in Table A1.2.

Finally the tap gains of the recursive filter shown in Figure 2.4.6 are calculated as follows using the data in Table A1.2.

The digital filter in Figure 2.4.6 is a cascade of two 2-pole filters and one 1-pole filter. If each 2-pole section is allocated a complex conjugate pair of poles \( R \pm jI \), then the transfer function of that section is given by \[68,95,70\]

\[ F(z) = \frac{K}{(1-(R+jI)z^{-1})(1-(R-jI)z^{-1})} \]

\[ = \frac{K}{1 - 2Rz^{-1} + G^2z^{-2}} \]  \hspace{1cm} (A1.22)

where \( G^2 = R^2 + I^2 \)

\( K = \text{DC gain} \)

The one pole section is used to realise the real z-plane pole and its transfer function is

\[ \frac{K}{1 - Rz^{-1}} \]  \hspace{1cm} (A1.23)
The overall transfer function of the three cascade sections is the product of their individual transfer functions and will have two pairs of complex conjugate poles and one real pole. The values of tap gains are therefore defined as follows:

\[ c_1 = -2R_2 \]
\[ c_2 = R_2^2 + I_2^2 \]
\[ c_3 = -2R_3 \]
\[ c_4 = R_3^2 + I_3^2 \]
\[ c_5 = -R_1 \]

These tap gains have been evaluated using the pole location given in Table A1.2 and the results are given in Table 2.4.1.
TABLE A1.1: Fifth order analogue Bessel filter for different frequency spreads

<table>
<thead>
<tr>
<th>Frequency spread $f_{sp}$</th>
<th>0.5 Hz</th>
<th>2 Hz</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut off frequency $f_c$</td>
<td>0.2943</td>
<td>1.1774</td>
</tr>
<tr>
<td>Filter poles in the s-plane</td>
<td>$-2.7785 + j0$</td>
<td>$-11.114 + j0$</td>
</tr>
<tr>
<td></td>
<td>$-2.554 \pm j1.3277$</td>
<td>$-10.2156 \pm j5.311$</td>
</tr>
<tr>
<td></td>
<td>$-1.7712 \pm j2.7208$</td>
<td>$-7.0848 \pm j0.8830$</td>
</tr>
</tbody>
</table>

TABLE A1.2:

<table>
<thead>
<tr>
<th>Frequency spread (Hz)</th>
<th>0.5</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter poles in the z-plane</td>
<td>$R_1$</td>
<td>0.946</td>
</tr>
<tr>
<td>$R_2 \pm j I_2$</td>
<td>0.95$\pm j$ 0.0252</td>
<td>0.8109$\pm j$0.0863</td>
</tr>
<tr>
<td>$R_3 \pm j I_3$</td>
<td>0.9638$\pm j$ 0.0524</td>
<td>0.8477$\pm j$0.1872</td>
</tr>
</tbody>
</table>
APPENDIX B

DIFFERENTIAL ENCODING AND DECODING

In converting the \( \{a_K\} \) into the \( \{s_i\} \) by the encoder at the transmitter (Figure 3.2.4 and Figure 6.2.1), and the \( \{s_i\} \) into the \( \{a'_K\} \) by the decoder at the receiver, differential coding is employed [72]. To reduce the error rate in the \( \{a'_K\} \) for a given error rate in the \( \{s_i\} \), the coding must, in addition, be as near as possible to Gray coding, the exact realization of this being unfortunately not attainable with the given signal. The coding and decoding processes assumed here are described in references 72 and 51. The stream of binary digits \( \{a_K\} \) to be transmitted is divided into adjacent groups of four digits, such that the \( i \)th group \( \bar{a}_4(i-1)+1', \bar{a}_4(i-1)+2', \bar{a}_4(i-1)+3', \bar{a}_4(i-1)+4 \) is converted by the encoder into the corresponding data symbol \( s_i \). The first two binary digits \( \bar{a}_4(i-1)+1' \) and \( \bar{a}_4(i-1)+2' \) in the \( i \)th group, are recoded in the encoder to give the corresponding two binary digits \( \bar{a}_4(i-1)+1 \) and \( \bar{a}_4(i-1)+2 \) (with possible values 0 or 1) according to Table B1.1. The resulting group of four binary digits \( \bar{a}_4(i-1)+1', \bar{a}_4(i-1)+2', \bar{a}_4(i-1)+3', \bar{a}_4(i-1)+4 \) considered here as the corresponding binary-coded number is encoded into the appropriate data symbol \( s_i \), according to Figure B1.1. Thus the first two binary digits in the binary coded number determines the quadrant containing \( s_i \), and the remaining two digits determine the position of \( s_i \) in the quadrant. The latter two digits in any quadrant are the same as those in the all positive quadrant, if this is rotated to coincide with the given quadrant.

Following the detection of \( s_i \), at the receiver, the corresponding sequence of four detected binary digits \( \bar{a'}_4(i-1)+1', \bar{a'}_4(i-1)+2', \bar{a'}_4(i-1)+3', \bar{a'}_4(i-1)+4 \) is determined from Figure B1.1, using of course, the detected values of \( s_{r,i} \) and \( s_{q,i} \) (equation 3.2.1) in place of their actual values.
The detected values of $\alpha_{4(i-1)+1}$ and $\alpha_{4(i-1)+2}$ are then determined from Table B1.1, using the detected values of $\alpha_{4(i-2)+1}$, $\alpha_{4(i-2)+2}$, $\alpha_{4(i-1)+1}$ and $\alpha_{4(i-1)+2}$. It can be seen from Figure B1.1 and Table B1.1 that a shift of a multiple of 90° in the phase relationship between the reference carrier in the coherent demodulator (Figure 3.2.2) and the received signal carrier, such as can occur following a deep fade, does not change the values of $\alpha'_{4(i-1)+3}$ and $\alpha'_{4(i-1)+4}$ corresponding to any given value of $s_1'$. To reduce further the error rate in the $\{a_k\}$ the coding in Figure B1.1 is as near as possible to Gray coding. Actually Gray coding is achieved over each quadrant in Figure B1.1.
Fig. B1.1: Encoding of the \( \{s_i\} \)
TABLE B1.1: Differential coding of binary digits in encoder

<table>
<thead>
<tr>
<th>$\alpha_4(i-1)+1$</th>
<th>$\alpha_4(i-1)+2$</th>
<th>$\bar{\alpha}_4(i-2)+1$</th>
<th>$\bar{\alpha}_4(i-2)+2$</th>
<th>$\alpha_4(i-1)+1$</th>
<th>$\alpha_4(i-1)+2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
<td>00</td>
</tr>
<tr>
<td>01</td>
<td>00</td>
<td>01</td>
<td>11</td>
<td>11</td>
<td>01</td>
</tr>
<tr>
<td>11</td>
<td>00</td>
<td>01</td>
<td>11</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>00</td>
<td>01</td>
<td>11</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>00</td>
<td>01</td>
<td>01</td>
<td>11</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>01</td>
<td>01</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>00</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>00</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>01</td>
</tr>
<tr>
<td>00</td>
<td>10</td>
<td>10</td>
<td>11</td>
<td>11</td>
<td>01</td>
</tr>
<tr>
<td>01</td>
<td>10</td>
<td>10</td>
<td>11</td>
<td>11</td>
<td>00</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>10</td>
<td>11</td>
<td>11</td>
<td>01</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
</tbody>
</table>
APPENDIX C

SIGNAL DISTORTION INTRODUCED BY A CHANNEL SAMPLED IMPULSE RESPONSE

The signal distortion introduced by a channel sampled impulse response has been described in detail in reference 21, and only the more important results are quoted here.

The channel is said to introduce signal distortion whenever there are two or more non-zero components \( y_h \) in the sampled impulse response \( Y \). Two types of distortion can be introduced by the sampled impulse response of a channel and these are the amplitude and phase distortion which are defined in terms of the discrete Fourier transform components of the corresponding sampled impulse response.

C.1 PHASE DISTORTION

The signal distortion introduced by a channel is defined to be pure phase distortion (that is no amplitude distortion or attenuation) when all the discrete Fourier components of the sampled impulse of this channel have magnitude unity. To see the time domain equivalence of this definition, let \( V(z) \) be the z-transform of the channel sampled impulse response so that

\[
V(z) = y_0 + y_1 z^{-1} + \ldots + y_g z^{-g}
\]

Furthermore, let \( U(z) \) be the z-transform of the sequence of values obtained by reversing the order of the complex conjugate of the sequence with z-transform \( V(z) \), the reversal being pivoted about the components at time \( t = 0 \). Thus,

\[
U(z) = y_0^* + y_1^* z + \ldots + y_g^* z^g
\]
It is shown in reference 21 that a channel with z-transform $V(z)$ introduces pure phase distortion when the sequence with z-transform $U(z)$ is the same as the sequence with z-transform $V^{-1}(z)$. That is, if the complex conjugate of the reversed sequence of a channel sampled impulse response is also the inverse of the sequence, then signal distortion introduced by this channel is a pure phase distortion.

Also it has been shown that the pure phase distortion represents an orthogonal transformation. Consequently, there is no inevitable loss in tolerance to noise when the received sequence is processed by the appropriate detection process. Hence phase distortion is not usually considered when assessing the distortion present in a sampled waveform, provided that a suitable detection process is used in the receiver which can reverse the orthogonal transformation.

Finally in the case of pure phase distortion each pole or zero of the z-transform of the sampled impulse response ($V(z)$) is accompanied by a zero or pole, respectively, at the complex conjugate of the reciprocal value of $z$. Thus, when a filter introducing pure phase distortion is connected in cascade with a channel, and when the poles of the z-transform of the filter coincide with zeros of the z-transform of the channel, then the action of the filter is to replace these zeros by the corresponding set of zeros at the complex conjugate reciprocal values of $z$. Thus the replacement of one or more zeros in a z-transform, by zeros at the complex conjugate of the reciprocal values of $z$, is pure phase distortion and is therefore an orthogonal transformation (Figure Cl.1).

C.2 AMPLITUDE DISTORTION

The signal distortion by a channel is defined to be pure amplitude distortion (that is no phase distortion or delay) when all the discrete Fourier transform components of the sampled impulse of this channel are real-valued quantities. The time domain equivalence of this definition is that the components of the channel sampled impulse response $y_0, y_1, y_g$ are Hermitian about its central component $y_{g/2}$ where $g$ is an even
value. Thus,

\[ Y_i = Y_{g-1}^* \quad \text{for} \quad i = 0, 1 \ldots \ldots \quad (g/2)-1 \]

It can be shown that a channel introducing pure amplitude distortion does not introduce an orthogonal transformation onto the transmitted signal and it normally reduces the best tolerance to additive white Gaussian noise regardless of the type of signal processor used at the receiver. Consequently, amplitude distortion is a much more important factor (than phase distortion) to be considered when assessing the severity of the signal distortion introduced by a channel.

Finally, all zeros of the z-transform of the sampled impulse response of the channel occur in complex conjugate reciprocal pairs when the channel introduces pure amplitude distortion as shown in Figure C1.2.
Figure C1.1: (a) zeros (or roots) of a channel sampled impulse response in the z-plane
(b) zeros and poles of a filter introducing pure phase distortion
(c) zeros of the channel and filter in cascade in the z-plane

Figure C1.2: Zeros of the sampled impulse response of a channel introducing pure amplitude distortion
Consider a QAM data transmission system with data-symbols \( \{s_i\} \) transmitted at a rate of \( 1/T \) symbols per second. Let the complex sampled impulse response of the channel at time \( t = iT \) be

\[
Y_i = [y_{i,0} \ y_{i,1} \ \ldots \ y_{i,g}]
\]  \hspace{1cm} (D1.1)

Hence, the received signal sample at time \( t = iT \) is (equation 3.3.6)

\[
r_i = \sum_{h=0}^{g} s_{i-h} Y_{i,h} + w_i
\]  \hspace{1cm} (D1.2)

Let

\[
S_i = [s_i \ s_{i-1} \ \ldots \ s_{i-g}]
\]  \hspace{1cm} (D1.3)

Also let \( Y'_{i-1} \) be an estimate of \( Y_{i-1} \) based on the \( i-1 \) received samples \( r_1 \ r_2 \ \ldots \ r_{i-1} \).

Then the steps of the Kalman estimator algorithm will be as follows [9, 25, 29].

Step 1: Compute \( r_i' \)

\[
r_i' = Y_{i-1}' S_i'^T
\]

Step 2: Compute the error \( e_i' \)

\[
e_i' = r_i - r_i'
\]
Step 3: Compute Kalman gain vector $K_i$

$$K_i = \frac{P_{i-1}S_i^T}{\omega + S_i P_{i-1} S_i^T}$$

Step 4: Update the matrix $P_i$

$$P_i = \frac{1}{\omega} [P_{i-1} - K_i S_i P_{i-1}]$$

Step 5: Update the estimates of the sampled impulse response of the channel

$$Y_i' = Y_{i-1}' + K_i e_i'$$

Note that $(\omega)$ in the algorithm is a positive constant less than unity selected to provide short term averaging appropriate for the time variations of the channel.

Finally, to start the above algorithm values of $Y_o'$ and $P_o$ are required. $Y_o'$ may be set to a null vector and $P_o$ (which is a $(g+1) \times (g+1)$ matrix) to the identity matrix.
APPENDIX E

EXPANDED-MEMORY POLYNOMIAL FILTER

Assume that scalar observations are made \( x_0, x_1, \ldots, x_n \) at equally spaced time instants as shown in Figure E1.1.

![Figure E1.1](image)

Let the prediction of \( x_{n+1} \) (the one-step prediction) which is the value given by a polynomial of degree (m) that fits most closely with the values of \( x_0, x_1, \ldots, x_n \) according to the least-squares criterion, be \( Z_{n+1,n} \) (Figure E1.1). Notice that \( Z_{n+1,n} \) lies on the polynomial fit curve whereas in general, the \( x_i \) for \( i = 0, 1, \ldots, n \) do not.

\( Z_{n+1,n} \) can be found by:

1. Determining the polynomial of the least-squares fit by using all the stored values \( x_0, x_1, \ldots, x_n \). For example, let \( n = 4 \) and \( m = 1 \) then \( Z_{n+1,n} \) by this method is \( Z_{5,4} = \frac{1}{10} [8x_4 + 5x_3 + 2x_2 - x_1 - 4x_0] \).

This method is called the fixed memory polynomial filter [83].
2. Also the prediction $Z_{n+1,n}$ can be obtained recursively without the need to store all the values $x_0, x_1, ..., x_n$. Here the prediction $Z_{n+1,n}$ will be a linear combination of its predecessor and the latest observation $x_n$. As a result only one datum need be retained, and when the recursion is cycled, that datum can be discarded and room made for its successor. Only one memory location is thus needed for the data. This method is called the expanded-memory polynomial filter [83].

The algorithms for degrees 0, 1 and 2 expanded-memory polynomial filters are given in Table E1.1.

The terms $Z_{i+1,i}$ and $Z_{i+1,i}$ in the table are functions of the first and second differentials of $Z_{i+1,i}$ with respect to time.

Also, for any given degree (m) of the polynomial, the quantities in Table E1.1 are evaluated in the order shown, so that the terms on the right hand side of any equation are known at the time at which they are required. To start these predictors, set

$$Z_{0,-1}^{(1)} = Z_{0,-1}^{(2)} = Z_{0,-1} = 0.0$$

Finally, any value on the polynomial fit curve (say $\hat{Z}$) in Figure E1.2 can be found as follows:

$$\hat{Z} = Z_{n+1,n} - (\Delta t) Z_{n+1,n}^{(1)} + (\Delta t)^2 Z_{n+1,n}^{(2)}$$

![FIGURE E1.2](image-url)
### TABLE El.1: The Expanded-Memory Polynomial Filters

<table>
<thead>
<tr>
<th>Degree of Polynomial</th>
<th>One-step prediction at time $t=iT$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$z_{i+1,i} = z_{i,i-1} + \frac{1}{i+1} \epsilon_i$</td>
</tr>
<tr>
<td>1</td>
<td>$z_{i+1,i}^{(1)} = z_{i,i-1}^{(1)} + \frac{6}{(i+2)(i+1)} \epsilon_i$</td>
</tr>
<tr>
<td></td>
<td>$z_{i+1,i} = z_{i,i-1}^{(1)} + z_{i+1,i}^{(1)} + \frac{2(2i+1)}{(i+2)(i+1)} \epsilon_i$</td>
</tr>
<tr>
<td>2</td>
<td>$z_{i+1,i}^{(2)} = z_{i,i-1}^{(2)} + \frac{30}{(i+3)(i+2)(i+1)} \epsilon_i$</td>
</tr>
<tr>
<td></td>
<td>$z_{i+1,i} = z_{i,i-1}^{(1)} + 2z_{i+1,i}^{(2)} + \frac{18(2i+1)}{(i+3)(i+2)(i+1)} \epsilon_i$</td>
</tr>
<tr>
<td></td>
<td>$z_{i+1,i} = z_{i,i-1}^{(1)} + z_{i+1,i}^{(1)} + \frac{3(2i^2+3i+2)}{(i+3)(i+2)(i+1)} \epsilon_i$</td>
</tr>
</tbody>
</table>

* $\epsilon_i = x_i - z_{i,i-1}$
APPENDIX F

ADAPTIVE ADJUSTMENT OF THE LINEAR FILTER

F.1 TIME INvariant CHANNEL [74]

Let the sampled impulse response of the channel be the \((g+1)\) complex components row vector

\[
Y = [Y_0, Y_1, \ldots, Y_g]
\]

whose z-transform is

\[
Y(z) = Y_0 + Y_1 z^{-1} + \ldots + Y_g z^{-g} \quad (F1.1)
\]

suppose now that:

\[
Y(z) = Y_1(z) \cdot Y_2(z) \quad (F1.2)
\]

where

\[
Y_1(z) = \eta (1 + \alpha_1 z^{-1})(1 + \alpha_2 z^{-1}) \ldots (1 + \alpha_m z^{-1}) \quad (F1.3)
\]

and

\[
Y_2(z) = z^{-m}(1 + \beta_1 z)(1 + \beta_2 z) \ldots (1 + \beta_m z) \quad (F1.4)
\]

It is assumed here that no roots (zeros) of \(Y(z)\) lie exactly on the unit circle in the \(z\)-plane, also \(|a_i| < 1\) and \(|b_i| < 1\) where \(a_i\) is the negative of a root of \(Y(z)\) and \(b_i\) is the negative of the reciprocal of a root of \(Y(z)\). The quantity \(\eta\) is the appropriate complex value needed to satisfy equations F1.2-F1.4. \(|a_i|\) and \(|b_i|\) are the absolute values of \(a_i\) and \(b_i\) respectively. The adaptive linear transversal filter has \((n+1)\) taps where \(n\) is typically between 20 and 60. It is for
convenience assumed here that the filter is adjusted to its ideal form (as required in the conventional nonlinear equalizer). The z-transform of the sampled impulse of the filter is now approximately:

\[ D(z) = z^{-n} Y_2^{-1}(z) Y_3(z) \quad \text{(Fl.5)} \]

where:

\[ Y_3(z) = (1+\beta_1^*z^{-1})(1+\beta_2^*z^{-1}) \cdots (1+\beta_m^*z^{-1}) \quad \text{(Fl.6)} \]

and \( \beta_1^* \) is the complex conjugate of \( \beta_1 \). Thus the z-transform of the channel and linear filter is approximately:

\[ F(z) = Y(z)D(z) \quad \text{(Fl.7)} \]

\[ = z^{-n} Y_1(z)Y_3(z) \]

Equations Fl.5 and Fl.7 only apply to the given system when \( n \rightarrow \infty \), but a very good approximation is usually obtained here without using an unduly large value of \( n \). Clearly, all the roots of \( F(z) \) lie inside the unit circle in the z-plane, which means that the channel and linear filter together have a response that is minimum phase. The roots of \( Y_3(z) \) are the complex conjugates of the reciprocals of the roots of \( Y_2(z) \), so that the linear filter replaces all roots of \( Y(z) \) that lie outside the unit circle by the complex conjugates of their reciprocals, leaving the remaining roots (those of \( Y_1(z) \)) unchanged. Since the receiver is assumed to know \( Y(z) \), it can, in principle, evaluate \( D(z) \) and hence also \( F(z) \). This involves the determination of the roots of \( Y(z) \) that lie outside the unit circle, with no restrictions in time or complexity, could be achieved by any conventional root-finding algorithm. However the technique that will be described briefly now determines the wanted roots of \( Y(z) \) and at the same time evaluates \( D(z) \) and \( F(z) \) by a simple and accurate way.
F.2 THE ALGORITHM

The basic principle behind the adjustment of the adaptive linear filter is as follows. The receiver first forms a filter with z-transform:

\[ A_i(z) = (1+\lambda_i z)^{-1} \]  

(Fl. 8)

for \( i=0,1, \ldots, k \) in turn, using an iterative process to adjust \( \lambda_i \) so that as \( i \) increases \( \lambda_i \to \beta_1 \), \( \beta_1 \) is the negative of the reciprocal of the first roots of \( Y(z) \) to be processed by the system and, of course, \( |\beta_1| < 1 \). Since the filter with z-transform \( A_i(z) \) does not operate on the received signal in real time, its z-transform is not limited to zero and negative powers of \( z \). At the end of the iterative process, when \( i=k \), the z-transform of the filter is

\[ A_k(z) = (1+\beta_1 z)^{-1} \]  

(Fl. 9)

The receiver now forms a filter with the z-transform:

\[ C_i(z) = (1+\lambda_k^* z^{-1})^{-1}(1+\lambda_k^* k z^{-1}) \]

\[ = (1+\beta_1 z)^{-1}(1+\beta_1^* z^{-1}) \]  

(Fl. 10)

The whole of this process is carried out for each \( \beta_h \) \((h=2, \ldots, m)\) to give a total of \( m \) filters with z-transform \( (C_h(z)) \), which are combined together (connected in cascade) and a delay of \( n-m \) sampling intervals added. The \( m \) filters and associated delay are in fact implemented as a single filter whose z-transform now approximates to \( D(z) \).
The algorithm used to find $D(z)$ and $F(z)$ are as follows:

1. First the receiver holds in store the sequence $Y$ and an estimate $\lambda_1$ of the quantity $\beta_1$. The first estimate of $\beta_1$ at the start of the process is one of a number of different starting points. In the case of 16-point QAM and the telephone line as the transmission path, the starting points are $0.0$, $0.5$, $-0.5j$, $0.5j$ and $0.5$.

2. Having determined $\lambda_1$, the receiver appropriately adjusts the one-tap feedback transversal filter shown in Figure F1.1. The stored sequence $Y$ is now reversed in order, so that it starts with the component $y_g$, and it is fed through the feedback transversal filter. The sequence $Y$, passing through the filter in reverse order, is taken to be moving backwards in time starting with the component $y_0$ at time $t=0$. The delay of one sampling interval $T$ in the feedback filter now becomes an advance of $T$ with $z$-transform $z$. Thus the effective $z$-transform of the feedback filter becomes $A_i(z)$ and the output from the filter is the sequence of the $\{e_i, h\}$, only the $g+1$ components $e_{i,0} e_{i,1} ... e_{i,g}$ of this sequence are, in fact, generated. An improved estimate of $\beta_1$ is now given by (74)

$$\lambda_{i+1} = \lambda_i + c e_i, c / e_i \tag{Fl.11}$$

where $c$ here is a constant in the range $(0-1)$

and $e_i = e_{i,1} - e_{i,2} \lambda_i + e_{i,3} \lambda_i^2 - \ldots e_{i,g}(-\lambda_i)^{g-1} \tag{Fl.12}$

This gives a new one tap feedback transversal filter, with $\lambda_i$ replaced by $\lambda_{i+1}$. The effective $z$-transform of this filter, when operating on the sequence $Y$ in reverse order is

$$A_{i+1}(z) = (1 + \lambda_{i+1}z)^{-1} \tag{Fl.13}$$
and the coefficients of $z^{-h}$ in $Y(z)A_{i+1}(z)$ is $e'_{i+1,h}$. The iterative process continues in the manner described until the term $e'_{i,0} / \varepsilon_i$ in equation F1.11 satisfies

$$|e'_{i,0} / \varepsilon_i| < d$$  \hspace{1cm} (F1.14)

where $d$ is an appropriate small positive real constant or else until either $i=40$ or $|\lambda_1| > 1$ and in each case the process is terminated (the action taken when $i=40$ or $|\lambda_1| > 1$ will be considered later). When equation F1.14 is satisfied the iterative process is taken to have converged. Let the value of $i$ at convergence be $k$ so that

$$\lambda_k = \beta_1$$  \hspace{1cm} (F1.15)

3. The receiver next appropriately adjusts the two tap feedforward transversal filter shown in Figure F1.2 which has the z-transform

$$B_k(z) = 1 + \lambda_k^* z^{-1}$$  \hspace{1cm} (F1.16)

The sequence of the $\{e'_{k,h}\}$ for $h=0,1,\ldots,g$ is now fed through this filter in the correct order. This gives the output sequence $g+2$ components with z-transform

$$f_{1,-1} + f_{1,0} z^{-1} + \ldots + f_{1,g} z^{-g-1}$$  \hspace{1cm} (F1.17)
which is approximately equal to \( Y(z)A_K(z)B_K(z) \) and where \( f_1,-1 = 0 \). The resultant effect on the sequence \( Y \) of the two filters (Figures F1.1 and F1.2) giving the sequence of the \( \{f_1, h\} \) approximately to that of a single filter with z-transform

\[
C_1(z) = A_K(z)B_K(z) \quad \text{(F1.18)}
\]

\[
= (1+\beta_1 z)^{-1} (1+\beta_1^* z^{-1})
\]

as in equation F1.10.

This filter is an all-pass network having the same basic properties as the ideal adaptive linear transversal filter with z-transform \( D(z) \). Finally the output sequence of the \( \{f_1, h\} \) is advanced by one place (sampling interval) and the first component \( f_1,-1 \) discarded to give the sequence \( F_1 \) with z-transform.

\[
F_1(z) = f_1,0 + f_1,1 z^{-1} \ldots + f_1,g z^{-g} \quad \text{(F1.19)}
\]

\[
= z C_1(z) Y(z)
\]

For practical purposes the linear factor \((1+\beta_1 z)\) in equation F1.4 is replaced in \( F(z) \) by the linear factor \((1+\beta_1^* z^{-1})\). Thus the root \(-1/\beta_1\) of \( Y(z) \) is replaced by the root \(-\beta_1^*\) which is the complex conjugate of its reciprocal and lies inside the unit circle. \( F_1(z) \) contains, in addition, an advance of one sampling interval. The sequence \( F_1 \) (with z-transform \( F_1(z) \)) is an estimate of the sampled impulse response of the channel and adaptive linear transversal filter, when the z-transform of the latter is \( zC_1(z) \). The iterative process is repeated with \( Y \) replaced by \( F_1 \) for the tracking of more roots.
To adjust the tap gains of the (n+1) tap adaptive linear feedforward transversal filter, whose ideal z-transform is \( D(z) \), all tap gains of the filter are initially set to zero except for the last tap whose gain is set to unity. Thus the initial z-transform of the filter is

\[
D_O(z) = z^{-n}
\]

(F1.20)

and the initial z-transform of the channel (and filter) is \( z^{-n}Y(z) \). When convergence has been obtained in the iterative process previously described such that \( \lambda_K = \beta_1 \) the sequence \( D_O \) is fed through the two-tap feedforward transversal filter with z-transform \( B_K(z) \) starting with the first component of \( D_O \), to give an output sequence with \( (n+2) \) components and the z-transform \( D_O(z)B_K(z) \). The latter sequence is fed in reverse order, and starting with the last component, through the one tap feedback transversal filter (Figure F1.1) whose effective z-transform is now \( A_K(z) \) to give the output sequence with z-transform that is approximately

\[
D_O(z)C_1(z) = D_O(z)A_K(z)B_K(z)
\]

(F1.21)

When \( n+1 \) components of the output sequence have been obtained, the process is halted. These \( n+1 \) components, in the order in which they are received, are the coefficients of \( z^{-n-1}, z^{-n}, \ldots, z^{-1} \) in \( D_O(z)C_1(z) \). The tap gains of the hth tap of the adaptive filter is now set to the coefficient of \( z^{-h} \) for \( h=1,2,\ldots,n+1 \) to give the required tap gains. Thus the z-transform of the adaptive filter is approximately

\[
D(z) = zC_1(z)D_O(z)
\]

(F1.22)
The whole of the above procedure just described (steps 1, 2 and 3) is now repeated, but using $F_1(z)$ (in equation Fl.11) in place of $Y(z)$, and $D_1(z)$ in place of $D_0(z)$. At the end of the iterative process $\lambda_K = \beta_2$ and the values of $\lambda_K F_1(z)$ and $D_1(z)$ determines $F_2(z)$ and $D_2(z)$ which are then used in place of $F_1(z)$ and $D_1(z)$ for processing $\delta_3$.

The system continues in this way until, on the $h$th repetition of the whole procedure, no roots of $F_h(z)$ outside the unit circle are found, starting from any of the five possible values of $\lambda_0$. In the case where, from each starting point, $|\lambda_i| > 1$, for some value of $i$, or $i$ reaches 40, it is assumed that all $m$ roots of $Y(z)$ outside the unit circle have been located, such that $h=m$, and, this being so, all roots of $Y(z)$ that lie outside the unit circle are replaced by the complex conjugate of their reciprocals, in the $z$-transform of the channel and adaptive filter. The $z$-transform of the adaptive filter is now approximately

$$D_m(z) = D(z)$$  \hspace{1cm} (Fl.23)

so that the $z$-transform of the channel and adaptive filter is

$$a_0 + a_1 z^{-1} + \ldots + a_{n+g} z^{-n-g} = Y(z) D_m(z)$$

$$= F(z)$$  \hspace{1cm} (Fl.24)

where $a_h = 0$ for $h=0,1, \ldots, n-1$.

The estimates of the sampled impulse response of the channel and adaptive filter, that are employed by the detector, is the sequence $F_m$, with $z$-transform

$$F_m(z) = f_{m,0} + f_{m,1} z^{-1} + \ldots + f_{m,g} z^{-g} = z^D F(z)$$

$$= Y_1(z) Y_3(z)$$  \hspace{1cm} (Fl.25)
(equations F1.3 and F1.6). The delay of n sampling intervals introduced by the adaptive filter is, for convenience, ignored here but must obviously be taken into account when comparing $Y(z)D_m(z)$ and $F_m(z)$.

F.3 TIME VARYING HF CHANNEL [86]

The impulse response of the channel is now time varying. Here the algorithm will try to find the roots for each impulse response, and also the corresponding tap gains of the filter and the impulse response of the channel and filter $F_m$. Nine starting points are used for the HF link and they are given in Table F1.1.

The other modification of the algorithm is that during the first run, when the system has just started, the algorithm always starts with $\lambda_0$ set to starting point number 1 (Table F1.1) and it uses all the 9 possible starting points for tracking any new roots. This procedure is used to ensure that all or as many as possible of the roots outside the unit circle are found at the beginning. It is now assumed that all of the m roots outside the unit circle are found. For an HF link the corresponding set of $\beta_1$ are added to the original 9 starting points to give a total of $(m+9)$ starting points as shown in Figure F1.3. For the subsequent runs the algorithm is started with $\lambda_0$ set to $\beta_1$ ($\beta_1$ is the first root tracked from the previous run). Whenever a root is found or a new starting point is required because the algorithm has diverged, the next of the $(m+9)$ starting points is used. This process is repeated until all $(m+9)$ starting points have been used and this is taken to mean that there are no more roots (zeros) of $Y(z)$ lying outside the unit circle. The root finding algorithm process is now terminated until required again. Once a set of roots is found, the corresponding set of $m$ is added to the original 9 and the new set of starting points is used for the subsequent run.

The last modification in the algorithm is that now, when $i$ reaches 100 (not 40) or $|\lambda_1| > 1$, from a starting point, the iterative process, is terminated.
Fig. F1.1: One-tap feedback transversal filter

Fig. F1.2: Two-tap feedforward transversal filter

![Diagram of a one-tap feedback transversal filter](image1)

![Diagram of a two-tap feedforward transversal filter](image2)

Fig. F1.3

<table>
<thead>
<tr>
<th>$\beta_1$</th>
<th>$\beta_2$</th>
<th>$\cdots$</th>
<th>$\beta_m$</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>9</td>
</tr>
</tbody>
</table>
TABLE F1.1: Starting points for the adaptive filter for time varying channel

<table>
<thead>
<tr>
<th>Starting point number</th>
<th>Starting point</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0+j0.0</td>
</tr>
<tr>
<td>2</td>
<td>0.90909</td>
</tr>
<tr>
<td>3</td>
<td>-j0.90909</td>
</tr>
<tr>
<td>4</td>
<td>j0.90909</td>
</tr>
<tr>
<td>5</td>
<td>-0.90909</td>
</tr>
<tr>
<td>6</td>
<td>0.64282-j0.64282</td>
</tr>
<tr>
<td>7</td>
<td>0.64282+j0.64282</td>
</tr>
<tr>
<td>8</td>
<td>-0.64282+j0.64282</td>
</tr>
<tr>
<td>9</td>
<td>-0.64282-j0.64282</td>
</tr>
</tbody>
</table>
APPENDIX G

COMPUTER PROGRAMS

G1  Computer simulation for the generation of the sampled impulse responses of the HF channel \( \{Y_i\} \) (Section 3.3) and then converting them to minimum phase (i.e. perfect adaptive linear filtering)

G2  Computer simulation program for the adaptive estimator (Section 5.7)

G3  Computer simulation program for the combined system of adaptive linear filtering, channel estimation and near maximum likelihood detection (2B8) together with carrier tracking loop
PROGRAM TEST
C
YTR & YTI ARE THE Exact IMPULSE RESPONSE OF THE HF CHANNEL
C
TXR, TXI ARE THE TRANSMITTER FILTER IMPULSE RESPONSE REAL &
C
AND IMAGINARY PARTS
C
RXR & RXI ARE RECEIVER FILTER IMPULSE RESPONSES
C
THIS IS A PROGRAM FOR 3 SKYWAVES (1,3 MSECOND)DELAYS, 2HZ
C
FREQUENCY SPREAD
DIMENSION SSR(100), SSQ(100),RECR(100),RECI(100),SSRER(100)
C, SSREI(100), YR(31), YO(31), Y1R(31), Y1Q(31), AI(31),
CBR(31), BI(31), FIR(31), FII(31), EIR(31), EII(31), YRP(31), YQP(31)
DIMENSION CR(100), CJ(100)
DIMENSION OR(22), OI(22)
DIMENSION XR(35), XI(35)
DIMENSION AC(35), REZ(35)
REAL IMZ(35)
DIMENSION IS1(30), IS2(30), C(30), Y1(30), Y2(30)
DIMENSION IX1(30,30), IX2(30,30), CCC(30), IXXI(5)
DIMENSION IXXX1(5), IXXX2(5)
DIMENSION DF1(4), DF2(4)
C, IXX2(5), DK1(5), DK2(5), IXT1(30), IXT2(30), IXT1I(30),
CIXTT2I(30)
DIMENSION CR1(31, 102), CI1(31, 102)
DIMENSION YTR1(31, 101), YTI1(31, 101)
DIMENSION Y2R(31), Y2Q(31), ETOTR(31), EOTI(31)
DIMENSION EPAR(31), EPAI(31), Y2NR(31), Y2NQ(31), YDR(31), YDQ(31)
DIMENSION TXRDDI(16), TXIDD(16), RAY5(860), RAY6(860)
C
C
C
DATA COEFF/-1.9, 0.903135, -1.9276, 0.9316561, -0.946/
DATA COEFF/-1.6218, 0.6650064, -1.6954, 0.753639, -0.801/
DATA TXRDD/-0.7630237, -5.6487240, -11.9216180, -9.3588701
DATA TXR/-0.1795896, -3.0773455, -9.9409021, -11.8694731
C
C
C
C
DATA TDWR(5), A4I(1100), TXID(16), TXRD(16)
C
DIMENSION RXR(860), RXI(860), YTR(50), YTI(50), TXR(16)
C, TR(40, 40), WSI(30), WS(30), YTR(50), YTI(50), TXR(16)
C, TXI(16), COEFF(5), A4I(1100), TXID(16), TXRD(16)
DIMENSION RAY1(860), RAY2(860), RAY3(860), RAY4(860)
DIMENSION SB1(31, 32), YTRX(32), YTX(32)
DIMENSION SB2(31, 31), R3I(31), R4I(31)
DIMENSION AR1(31, 102), AI1(31, 102), BR1(31, 102), BI1(31, 102)
DIMENSION HNOSI(50), HNOSR(50), RXI(30), RXR(30), TI(40, 40)
DIMENSION FIX(860), FIX(860), FIX(860), FIX(860)
C
DATA TWR(5), A4I(1100), TXID(16), TXRD(16)
DIMENSION RAY1(860), RAY2(860), RAY3(860), RAY4(860)
DIMENSION SB1(32, 32), YTRX(32), YTX(32)
DIMENSION SB2(31, 31), R3I(31), R4I(31)
DIMENSION AR1(31, 102), AI1(31, 102), BR1(31, 102), BI1(31, 102)
DIMENSION HNOSI(50), HNOSR(50), RXI(30), RXR(30), TI(40, 40)
DATA TDWR(5), A4I(1100), TXID(16), TXRD(16)
DIMENSION RAY1(860), RAY2(860), RAY3(860), RAY4(860)
DIMENSION SB1(32, 32), YTRX(32), YTX(32)
DIMENSION SB2(31, 31), R3I(31), R4I(31)
DIMENSION AR1(31, 102), AI1(31, 102), BR1(31, 102), BI1(31, 102)
DIMENSION HNOSI(50), HNOSR(50), RXI(30), RXR(30), TI(40, 40)
DATA TDWR(5), A4I(1100), TXID(16), TXRD(16)
C8.7045826,-11.7869820,-5.5819054,3.1582131,1.7365460,0.0098505/  
DATA TXRD/-0.3990578,-4.2813263,-11.0911373,4.9023211,2.0666985, -0.0236212/  
DATA TXID/-2.0820312,-8.5761452,-12.3102829,3.5476141,3.9398962, -0.7258700/  
DATA TXDD/-1.3136537,-7.1104051,-12.3146972,2.2353854,4.5938614, -0.0931639/  
DATA RXI/1.3625952,11.5941040,27.3342937,7.2714615,-9.2602472, -0.0954462/  
P=SQRT(2.0/3.0)  
IQ=500  
POS=-1.0  
IMPL=16  
ILOOP=1100  
CCCCCG2=1.000  
OPEN(2,FILE='OUTPUT')  
OPEN(7,FILE='AH33',FORM='UNFORMATTED')  
OPEN(7,FILE='INFL',FORM='UNFORMATTED')  
READ(7,917)(YTR(I),I=1,22)  
FORMAT(1H6,F16.12)  
READ(7,917)(YTI(I),I=1,22)
CALL G05CBF(IQ)
DCGAIN=693.06
DCGAIN=1.0/DCGAIN
C
CALL G05CBF(IQ)
DO 7 1=1,5
RAYL11(I)=0.0
RAYL12(I)=0.0
RAYL13(I)=0.0
RAYL14(I)=0.0
RAYL15(I)=0.0
RAYL16(I)=0.0
DI1(I)=0.0
DI2(I)=0.0
DI3(I)=0.0
DI4(I)=0.0
DI5(I)=0.0
DI6(I)=0.0
7 CONTINUE
DO 31 1=1,50
HNOSI(I)=0.0
31 HNOSR(I)=0.0
C
YTR(I)=0.0
C31 YTI(I)=0.0
DO 35 J=1,102
DO 35 I=1,31
AR1(I,J)=0.0
A11(I,J)=0.0
BR1(I,J)=0.0
35 B11(I,J)=0.0
DO 8 J=1,1LOOP
RAY1(J)=0.0
RAY2(J)=0.0
RAY3(J)=0.0
RAY4(J)=0.0
RAY5(J)=0.0
8 RAY6(J)=0.0
DO 8999 J=1,5
A1(J)=0.0
A2(J)=0.0
A3(J)=0.0
A4(J)=0.0
A5(J)=0.0
A6(J)=0.0
8999 CONTINUE
SR=2.4
C
DEI=1.0
DEI1=1.0
DEI2=1.5
DEI3=3.0
C
DEI2=2.0
IDEL1=INT(DEI1*2*SR)
IDEL2=INT(DEI2*SR*2)
IMPI=IMPL+IDEL2
DO 32 1=1,IMPI
DO 32 J=1,IMPI
TR(I,J)=0.0
32 TI(I,J)=0.0
IMPS=(2*IMPL+IDEL2-1)/2
ISTEP=96
STEP=1.0/ISTEP
DO 63 I=1,100
SSR(I)=1.0
SSQ(I)=1.0
SSSER(I)=1.0
SSREI(I)=1.0
RECR(I)=0.0
RECI(I)=0.0
KL=2*IMPES
KLP=KL+2
KLM=KL+1
C
WRITE(2,772)IQ, ILOOP
772 FORMAT(1H 'IQ=', I4,2X, ILOOPL=I, I4)
WRITE(2,773)DEL, SR, ILOOP, DEL1, DEL2
C 'DEL1=', F4.2,3X, 'DEL2=', F4.2)
DO 64 I=1,31
YDR(I)=0.0
YDQ(I)=0.0
Y2R(I)=0.0
Y2Q(I)=0.0
ETOTR(I)=0.0
ETOTI(I)=0.0
YR(I)=0.0
YQ(I)=0.0
Y1R(I)=0.0
Y1Q(I)=0.0
AR(I)=0.0
AI(I)=0.0
BR(I)=0.0
BI(I)=0.0
FIR(I)=0.0
FII(I)=0.0
EIR(I)=0.0
EII(I)=0.0
YRP(I)=0.0
YQP(I)=0.0
64 CONTINUE
DO 5555 I=1,16
TTTR(I)=TXRDD(I)
TTTI(I)=TXIDD(I)
TTR(I)=TXRD(I)
5555 TTI(I)=TXID(I)
C
DO 1606 I=1,1MPL
C
TXRD(I)=0.0
C
TXID(I)=0.0
C
TXRDD(I)=0.0
C1606 TXIDD(I)=0.0
NSOK=13
WRITE(2,210)NSOK
210 FORMAT(1H 'NO. OF SKYWAVES=', I2)
DO 9 J=1,50
DI1(1)=G05DDF(0.0,P)-(RAYLI1(1)*COEFF(1))
C+RAYLI1(2)*COEFF(2))
DI1(2)=DI1(1)-(RAYLI1(3)*COEFF(3)+RAYLI1(4)*COEFF(4))
A1(2)=DI1(2)-(RAYLI1(5)*COEFF(5))
DQ1(1)=G05DDF(0.0,P)-(RAYL12(1)*COEFF(1))
C+RAYL12(2)*COEFF(2))
DQ1(2)=DQ1(1)-(RAYL12(3)*COEFF(3)+RAYL12(4)*COEFF(4))
A2(2)=DQ1(2)-(RAYL12(5)*COEFF(5))
DH1(1) = G05DDF(0.0, P) - (RAYLI3(1) * COEFF(1) + RAYLI3(2) * COEFF(2))
C * COEFF(1) + RAYLI3(2) * COEFF(2))
   DH1(2) = DH1(1) - (RAYLI3(3) * COEFF(3) + RAYLI3(4) * COEFF(4))
   A3(1) = DH1(2) - (RAYLI3(5) * COEFF(5))
   DX1(1) = G05DDF(0.0, P) - (RAYLI4(1) * COEFF(1))
C * COEFF(1) + RAYLI4(2) * COEFF(2))
   DX1(2) = DX1(1) - (RAYLI4(3) * COEFF(3) + RAYLI4(4) * COEFF(4))
   A4(1) = DX1(2) - (RAYLI4(5) * COEFF(5))
   DY1(1) = G05DDF(0.0, P) - (RAYLI5(1) * COEFF(1))
C * COEFF(1) + RAYLI5(2) * COEFF(2))
   DY1(2) = DY1(1) - (RAYLI5(3) * COEFF(3) + RAYLI5(4) * COEFF(4))
   A5(1) = DY1(2) - (RAYLI5(5) * COEFF(5))
   DZ1(1) = G05DDF(0.0, P) - (RAYLI6(1) * COEFF(1))
C * COEFF(1) + RAYLI6(2) * COEFF(2))
   DZ1(2) = DZ1(1) - (RAYLI6(3) * COEFF(3) + RAYLI6(4) * COEFF(4))
   A6(1) = DZ1(2) - (RAYLI6(5) * COEFF(5))

CONTINUE
DO 10 K = 1, ILOOP
   DI1(1) = G05DDF(0.0, P) - (RAYLI1(1) * COEFF(1))
   C * COEFF(1) + RAYLI1(2) * COEFF(2))
   DI1(2) = DI1(1) - (RAYLI1(3) * COEFF(3) + RAYLI1(4) * COEFF(4))
   A1(1) = DI1(2) - (RAYLI1(5) * COEFF(5))
CONTINUE
DQ1(1) = G05DDF(0.0, P) - (RAYLI2(1) * COEFF(1)) + RAYLI2(2) * COEFF(2)
DQ1(2) = DQ1(1) - (RAYLI2(3) * COEFF(3) + RAYLI2(4) * COEFF(4))
A2(1) = DQ1(2) - (RAYLI2(5) * COEFF(5))
DH1(1) = G05DDF(0.0, P) - (RAYLI3(1) * COEFF(1)) + RAYLI3(2) * COEFF(2)
DH1(2) = DH1(1) - (RAYLI3(3) * COEFF(3) + RAYLI3(4) * COEFF(4))
A3(1) = DH1(2) - (RAYLI3(5) * COEFF(5))
DX1(1) = G05DDF(0.0, P) - (RAYLI4(1) * COEFF(1)) + RAYLI4(2) * COEFF(2)
DX1(2) = DX1(1) - (RAYLI4(3) * COEFF(3) + RAYLI4(4) * COEFF(4))
A4(1) = DX1(2) - (RAYLI4(5) * COEFF(5))
DY1(1) = G05DDF(0.0, P) - (RAYLI5(1) * COEFF(1)) + RAYLI5(2) * COEFF(2)
DY1(2) = DY1(1) - (RAYLI5(3) * COEFF(3) + RAYLI5(4) * COEFF(4))
A5(1) = DY1(2) - (RAYLI5(5) * COEFF(5))
DZ1(1) = G05DDF(0.0, P) - (RAYLI6(1) * COEFF(1)) + RAYLI6(2) * COEFF(2)
DZ1(2) = DZ1(1) - (RAYLI6(3) * COEFF(3) + RAYLI6(4) * COEFF(4))
A6(1) = DZ1(2) - (RAYLI6(5) * COEFF(5))
RAYLI1(5) = A1(1)
RAYLI1(4) = RAYLI1(3)
RAYLI1(3) = DI1(2)
RAYLI1(2) = RAYLI1(1)
RAYLI1(1) = DI1(1)
RAYLI2(5) = A2(1)
RAYLI2(4) = RAYLI2(3)
RAYLI2(3) = DQ1(2)
RAYLI2(2) = RAYLI2(1)
RAYLI2(1) = DQ1(1)
RAYLI3(5) = A3(1)
RAYLI3(4) = RAYLI3(3)
RAYLI3(3) = DI1(2)
RAYLI3(2) = RAYLI3(1)
RAYLI3(1) = DI1(1)
RAYLI4(5) = A4(1)
RAYLI4(4) = RAYLI4(3)
RAYLI4(3) = DX1(2)
RAYLI4(2) = RAYLI4(1)
RAYLI4(1) = DX1(1)
RAYLI5(5) = A5(1)
RAYLI5(4) = RAYLI5(3)
RAYLI5(3) = DY1(2)
RAYLI5(2) = RAYLI5(1)
RAYLI5(1) = DY1(1)
RAYLI6(5) = A6(1)
RAYLI6(4) = RAYLI6(3)
RAYLI6(3) = DZ1(2)
RAYLI6(2) = RAYLI6(1)
RAYLI6(1) = DZ1(1)
A5(1) = A5(1) * DCGAIN
A6(1) = A6(1) * DCGAIN
A1(1) = A1(1) * DCGAIN
A2(1) = A2(1) * DCGAIN
A3(1) = A3(1) * DCGAIN
A4(1) = A4(1) * DCGAIN
RAY1(K) = A1(1)
RAY2(K) = A2(1)
RAY3(K) = A3(1)
RAY4(K) = A4(1)
RAY5(K) = A5(1)
RAY6(K) = A6(1)

10 CONTINUE
DO 400 IRUN = 1, ILOOP
  IF (IRUN .GT. 200) GO TO 401
A1(1) = RAY1(IRUN)
A2(1) = RAY2(IRUN)
A3(1) = RAY3(IRUN)
A4(1) = RAY4(IRUN)
A5(1) = RAY5(IRUN)
A6(1) = RAY6(IRUN)
SLOP1 = (A1(1) - A1(2)) * STEP
SLOP2 = (A2(1) - A2(2)) * STEP
SLOP3 = (A3(1) - A3(2)) * STEP
SLOP4 = (A4(1) - A4(2)) * STEP
SLOP5 = (A5(1) - A5(2)) * STEP
SLOP6 = (A6(1) - A6(2)) * STEP
DO 300 ISLM = 1, ISTEP

C IF (POS .LT. 0.0) GO TO 300
CCC CCCREAD(7)(Y1(I), I = 1, IMPES)
CCC CCCREAD(7)(Y2(I), I = 1, IMPES)
CCC C IF (IRUN .NE. 2) GO TO 9161
CCC CWRITE(2, 917)(Y1(I), I = 1, IMPES)
C917 FORMAT(1H 6F10.5)
CCC CWRITE(2, 917)(Y2(I), I = 1, IMPES)
C9161 GO TO 300
  IF (IRUN .LT. 300) GO TO 300
K = ISLM - 1
DELT1 = K * SLOP1
DELT2 = K * SLOP2
DO 200 I = 1, 50
  R(I) = 0.0
Q(I) = 0.0
DO 500 I = 1, IMPES
  R(I) = TXR(I) - TXI(I)
C500 Q(I) = TXR(I) + TXI(I)
R(I) = TXR(I) * (A3(2) + DELT3 - A4(2) - DELT4) - TXI(I) * (A3(2) + DELT3 + A4(2) + DELT4)
Q(I) = TXR(I) * (A3(2) + DELT3 + A4(2) + DELT4) + TXI(I) * (A3(2) + DELT3 - A4(2) - DELT4)
DELT3 = K * SLOP3
DELT4 = K * SLOP4
C IF (IRUN .LT. 275) GO TO 7777
C DO 9986 I = 1, IMPES
C TXRD(I) = TTR(I) * FF2
C9986 TXID(I) = TTI(I) * FF2
C IF (FF2 .GE. 1.0) GO TO 7777
C FF2 = XX2
C IF (FF2 .LE. 0.0) GO TO 7777
C FF2 = 1.0 - XX2
C XX2 = XX2 + 0.00002
DO 600 I = 1, IMPES
  R(I + IDEL1) = R(I + IDEL1) + TXRD(I) * (A3(2) + DELT3 - A4(2) - DELT4) - TXI(I) * (A3(2) + DELT3 + A4(2) + DELT4)
C-TXID(I) * (A3(2) + DELT3 + A4(2) + DELT4)
Q(I + IDEL1) = Q(I + IDEL1) + TXRD(I) * (A3(2) + DELT3 + A4(2) + DELT4) + TXI(I) * (A3(2) + DELT3 - A4(2) - DELT4)
DELT5 = K * SLOP5
DELT6 = K * SLOP6

C IF (IRUN . LT. 275) GO TO 8888
C DO 9987 I = 1, IMPL
C TXRDD(I) = TTTR(I) * FF3
C9987 TXIDD(I) = TTTI(I) * FF3
C IF (FF3 . GE. 1.0) GO TO 8888
C FF3 = XX3
C IF (FF3 . LE. 0.0) GO TO 8888
C XX3 = XX3 + 0.00002
DO 601 I = 1, IMPL
R(I + DEL2) = R(I + DEL2) + TXRDD(I) * (A5(2) + DELT5 - A6(2) - DELT6)
C - TXIDD(I) * (A5(2) + DELT5 + A6(2) + DELT6)
601 Q(I + DEL2) = Q(I + DEL2) + TXRDD(I) * (A5(2) + DELT5 - A6(2) - DELT6)
C + TXIDD(I) * (A5(2) + DELT5 + A6(2) + DELT6)
IM1 = IMPL - 1
DO 596 I = 1, IMP1
DO 596 J = 1, IM1
TR(I, IMP1 + 1 - J) = TR(I, IMP1 - J)
596 TI(I, IMP1 + 1 - J) = TI(I, IMP1 - J)
DO 700 I = 1, IMP1
TR(I, 1) = (R(I) + Q(I)) * 0.5
700 TI(I, 1) = (Q(I) - R(I)) * 0.5
IMPP = IMPES + 2
IMPRP = IMPES + 1
P = 0.05
C P = 0.00158
DO 800 I = 1, IMPES
HNOSR(IMPP - I) = HNOSR(IMPRP - I)
800 HNOSI(IMPP - I) = HNOSI(IMPRP - I)
HNOSR(1) = G05DDF(0.0, P)
HNOSI(1) = G05DDF(0.0, P)
WR = 0.0
WI = 0.0
DO 900 I = 1, IMPES
WR = WR + HNOSR(I) * WSI(I) - HNOSI(I) * WSQ(I)
900 WI = WI + HNOSR(I) * WSQ(I) + HNOSI(I) * WSI(I)
POS = POS
I0 = 0
IF (POS . LT. 0.0) GO TO 300
DO 2010 I = 1, IMP1, 2
I0 = I0 + 1
YTR(I0) = 0.0
YTI(I0) = 0.0
DO 2000 J = 1, I
YTR(I0) = YTR(I0) + TR(J, I + 1 - J) * RXR(I + 1 - J) - CTR(J, I + 1 - J) * RXI(I + 1 - J)
2000 YTI(I0) = YTI(I0) + TI(J, I + 1 - J) * RXR(I + 1 - J) + CTR(J, I + 1 - J) * RXI(I + 1 - J)
YTR(I0) = YTR(I0) * 2.08333333E-4
2010 YTI(I0) = YTI(I0) * 2.08333333E-4
DO 2030 I = 1, IMP1, 2
I0 = I0 + 1
YTR(I0) = 0.0
YTI(I0) = 0.0
K = I + 1
DO 2020 J = K, IMP1
YTR(I0) = YTR(I0) + TR(J, IMP1 + 1 - J) * RXR(IMP1 + 1 - J) - CTR(J, IMP1 + 1 - J) * RXI(IMP1 + 1 - J)
2020 YTI(I0) = YTI(I0) + TI(J, IMP1 + 1 - J) * RXR(IMP1 + 1 - J)
C+TR(J, IMP1+1+I-J)*RXI(IMP1+1+I-J)

YTR(IO)=YTR(IO)*2.0E333333E-4

LC3=LC3+1

IF (LC3.NE.200) GO TO 4031

WRITE(2,2251)(YTR(I), I=1, IMPES)

WRITE(2,2251)(YTI(I), I=1, IMPES)

C THIS PART IS WRITTEN TO WORK OUT THE ROOTS OF A POLYNOMIAL
C OF DEGREE N WITH COMPLEX COEFFICIENTS USING NAG LIBRARY
C CO2ADF.
C
N=IMPES

IFAIL=0

P1=0.1

TOL=X02AAF(P1)

CALL CO2ADF(AR, AC, N, REZ, IMZ, TOL, IFAIL)

IF (IFAIL.NE.0) GO TO 401

WRITE(4,910)(REZ(I), I=1, 19)

WRITE(4,910)(IMZ(I), I=1, 19)

CALL SRCON(REZ, IMZ, OR, OI, N1, D1, NM)

DO 56 I=1, NM

CCR=YYR*OR(I)-YYQ*OI(I)

CCI=YYR*OI(I)+YYQ*OR(I)

YYR=CCR

YYQ=CCI

CALL EXPAD(REZ, IMZ, XR, XI, N)

CALL MULT(XR, XI, YYR, YYQ, N)

WRITE(4,910)(XR(I), I=1, 22)

WRITE(4,910)(XI(I), I=1, 22)

CONTINUE

DO 350 IP=1, 10

IQ=IQ+6

P=P+0.008

ER=0.0

IE=0

P=0.05
DO 999 I=1,IMPES
  Y1(I)=XR(I)
  Y2(I)=XI(I)
  WRITE(7),(Y1(I),I=1,IMPES)
  WRITE(7),(Y2(I),I=1,IMPES)
  IF(LC3.NE.200)GO TO 300
  WRITE(2,2251),(Y1(I),I=1,IMPES)
  WRITE(2,2251),(Y2(I),I=1,IMPES)
  CONTINUE
C IF(IRUN.LT.25)GO TO 400
C CCCCCWRITE(2,917)(YTR(I),I=1,IMPES)
C 917 FORMAT(1H ,6F16.12)
C CCCCCWRITE(2,917)(YTI(I),I=1,IMPES)
C CCCCCWRITE(2,910)IRUN,NM,SUM1
C 910 FORMAT(1H,'IRUN=',13.2X,'NO. OF ROOTS'='S=',13,2X,
C C 'SUM1=',F10.5)
C CCCCCDOC411140CI=1,NM
C 4CC4C40 OR(I)=-OR(I)
C CCCCCWRITE(2,911)(OR(I),I=1,NM)
C 911 FORMAT(1H,6F10.5)
C CCCCCWRITE(2,911)OI(I),I=1,NM)
C AMSE=0.0
DO 911 I=1,IMPES
  AMSE=AMSE+YTR(I)**2+YTI(I)**2
  WRITE(2,917)AMSE
  CONTINUE
C WRITE(2,4400)(YTR(I),I=1,IMPES)
C 44C400 FORMAT(1H,6F10.5)
C WRITE(2,4400)(YTI(I),I=1,IMPES)
C 401 STOP
END
SUBROUTINE MULT(XR, XI, YYR, YYQ, N)
THIS SUBROUTINE IS USED TO MULTIPLY A COMPLEX SEQUENCE BY A COMPLEX NUMBER YYR+JYYQ
DIMENSION XR(N), XI(N)
DO 70 I=1,N
  ARR=XR(I)*YYR-XI(I)*YYQ
  ACC=XR(I)*YYQ+XI(I)*YYR
  XR(I)=ARR
  XI(I)=ACC
RETURN
END
SUBROUTINE EXPAD(REZ, IMZ, XR, XI, N)
C THIS SUBROUTINE IS USED TO EXPAND THE POLYNOMIAL (1+AX)(1+BX)...
WHERE A,B,... ARE STORED IN REZ,IMZ
DIMENSION REZ(N), XR(N), XI(N), CR(100), CJ(100)
REAL IMZ(N)
N1=N-1
DO 10 I=1,N
XR(I) = 0.0
XI(I) = 0.0
XR(1) = 1.0
DO 40 J = 1, N1
DO 20 I = 1, N1
CR(I) = XR(I) * REZ(J) - XI(I) * IMZ(J)
CJ(I) = XR(I) * IMZ(J) + XI(I) * REZ(J)
DO 30 I = 1, N1
XR(I+1) = XR(I+1) + CR(I)
XI(I+1) = XI(I+1) + CJ(I)
30 CONTINUE
RETURN
END

SUBROUTINE SRCON(REZ, IMZ, OR, OI, N1, D1, NM)

C
C THIS SUBROUTINE IS WRITTEN TO SORT OUT ALL THE ROOTS
C WHICH LIE OUTSIDE THE UNIT CIRCLE, AND CONVERT THEM TO THEIR
C COMPLEX CONJUGATE RECIPROCAL
C
DIMENSION REZ(N1), OR(22), OI(22)
REAL IMZ(N1)
IRR = 0
DO 940 I = 1, N1
OR(I) = 0.0
OI(I) = 0.0
DO 80 I = 1, N1
S1 = (REZ(I)**2 + IMZ(I)**2)
S2 = SQRT(S1)
IF (S2 LT D1) GO TO 15
IRR = IRR + 1
OR(IRR) = -REZ(I)
OI(IRR) = IMZ(I)
REZ(I) = REZ(I)/S1
IMZ(I) = IMZ(I)/S1
80 CONTINUE
CONTINUE
NM = IRR
WRITE(4, 910) (OR(I), I = 1, NM)
WRITE(4, 910) (OI(I), I = 1, NM)
C910 FORMAT(1H, 6F10.5)
DO 930 I = 1, N1
REZ(I) = -REZ(I)
930 IMZ(I) = -IMZ(I)
RETURN
END
PROGRAM TAST3

DIMENSION SSR(100), SSQ(100), RECR(100), RECI(100), SSRER(100)
C, SSRE1(100), YR(31), YQ(31), Y1R(31), Y1Q(31), AR(31), AI(31),
C, BRI(31), F1R(31), F1I(31), EIR(31), E1R(31), YRP(31), YQP(31)
DIMENSION CR(31), CI(31)
DIMENSION YTR1(31,105), YT11(31,105), YTR(31), YQR(31)
DIMENSION Y2R(31), Y2Q(31), ETOTR(31), ETOTI(31)
DIMENSION EPAR(31), EPAI(31), Y2NR(31), Y2NQ(31)
DIMENSION TXRDD(16), TXIDD(16), RAY5(1200), RAY6(1200)
C, RAYL5(5), RAYL6(5), D1(5), A5(5), A6(5)
DIMENSION TTR(16), TTI(16), TTRR(16), TTRR(16)
DIMENSION SB2(31,31), R3(31), R4(31)
DIMENSION AR1(31,105), A11(31,105)
DIMENSION HNOS1(50), HNOSR(50), RXI(30), RXR(30), TI(40,40)
C, TR(40,40), WSQ(30), YTR(50), YTI(50), TXR(16)
C, TXI(16), COEFFF, A4(100), TXID(16), TXRD(16)
DIMENSION RAY1(1200), RAY2(1200), RAY3(1200), RAY4(1200)
DIMENSION SB1(32,32), R1(32), R2(32), YTRX(32), YTPX(32)
DATA COEFF/1.9, 0.903135, -1.9276, 0.9316561, -0.946/
C, DATA COEFF/1.6218, 0.6650064, -1.6954, 0.753639, -0.801/
C, DATA TXRDD/-1.3136537, -7.1104051, -12.3469721, -7.5848703,
C, C2.2353854, 4.5938614, 0.0931639, -1.9704176, -0.3233694,
C, C0.6312328, -0.1035718, -0.3865939, -0.0734526, -0.0386471,
C, C0.0608046, -0.0713496/
C, DATA TXIDD/1.1104051, -1.23469721, -7.5848703,
C, C2.353854, 4.5938614, 0.6312238, 0.0931639, -1.9704176, -0.3233694,
C, C0.6312328, -0.1035718, -0.3865939, -0.0734526, -0.0386471,
C, C0.0608046, -0.0713496/
C, DATA TXRDD/-1.3136537, -7.1104051, -12.3469721, -7.5848703,
C, C2.2353854, 4.5938614, 0.0931639, -1.9704176, -0.3233694,
C, C0.6312328, -0.1035718, -0.3865939, -0.0734526, -0.0386471,
C, C0.0608046, -0.0713496/
C, DATA TXIDD/1.1104051, -1.23469721, -7.5848703,
C, C2.2353854, 4.5938614, 0.0931639, -1.9704176, -0.3233694,
C, C0.6312328, -0.1035718, -0.3865939, -0.0734526, -0.0386471,
C, C0.0608046, -0.0713496/
C, DATA TXRDD/1.1104051, -1.23469721, -7.5848703,
C, C2.2353854, 4.5938614, 0.0931639, -1.9704176, -0.3233694,
C, C0.6312328, -0.1035718, -0.3865939, -0.0734526, -0.0386471,
C, C0.0608046, -0.0713496/
C, DATA TXIDD/1.1104051, -1.23469721, -7.5848703,
C, C2.2353854, 4.5938614, 0.0931639, -1.9704176, -0.3233694,
C, C0.6312328, -0.1035718, -0.3865939, -0.0734526, -0.0386471,
C, C0.0608046, -0.0713496/
C, DATA TXRDD/1.1104051, -1.23469721, -7.5848703,
C, C2.2353854, 4.5938614, 0.0931639, -1.9704176, -0.3233694,
C, C0.6312328, -0.1035718, -0.3865939, -0.0734526, -0.0386471,
C, C0.0608046, -0.0713496/
C, DATA TXIDD/1.1104051, -1.23469721, -7.5848703,
C, C2.2353854, 4.5938614, 0.0931639, -1.9704176, -0.3233694,
C, C0.6312328, -0.1035718, -0.3865939, -0.0734526, -0.0386471,
C, C0.0608046, -0.0713496/
C, DATA TXIDD/1.1104051, -1.23469721, -7.5848703,
275

c
c
c
c
c
c
c
c
c
c
c
c
c
c
c
c
c
c
c
c
c
c
c
c

c
c
c
c
c

CO. 0392435,0.0164020/
DATA TXRD/-1.6694374,
-7.8492148,
-12.3887079,
-6.6023157,
C2.9408554,4.3005084,
-0.3368383,
-1.9014342,
-0.1433592,
CO. 6242601,0.0278577,
-0.3820071,
-0.0416905,
-0.0439705,
CO. 0749333,
-0.0594132/
DATA TXID/13.2372707,39.6493461,46.9272219,19.2346609,
C-8.8804125,
-9.0256163,1.6284281,2.8139013,
-0.4311352,
C-0.4537174,0.3081762,
-0.0772327,
-0.3043271,0.0085057,
CO. 0093809,0.0094992/
DATA TXRD/-2.0820312,
-8.5764574,
-12.3102829,
-5.5766967,
C3.5478141,3.9398962,
-0.7258700,
-1.7957047,0.0207712,
CO. 6018517,
-0.0464514,
-0.3680978,
-0.0182370,
-0.0471151,
CO. 0789564,
-0.0402623/
DATA TXID/15.5920086,41.8806123,45.4936549,15.6491184,
C-10.1779415,
-7.9404850,2.2650217,2.4780490,
C-0.5908297,
-0.3459233,0.3143350,
-0.1350979,
-0.2790169,
CO. 0231422,0.0071662,0.0054678/
DATA TXRDD/-1.3136537,
-7.1104051,
-12.3469721,
-7.5848703,
C2.2353854,4.5938614,0.0931639,
-1.9704176,
-0.3233694,
CO. 6313238,0.1035718,
-0.3865939,
-0.0734526,
-0.0386471,
CO. 0608046,
-0.0713496/
DATA TXIDD/11.0688962,37.2136597,47.9575159,22.8262482,
C-7.2498590,
-10.0026703,0.8695437,3.1072800,
-0.2261096,
C-0.5552906,0.2882096,
Ol07706,
-0.0156703,
-0.3215770,
-O.
CO. 0140909,0.0135711/
DATA RXR/-1.9417691,
-15.9797864,
-35.1417733,
-34.4788717,
C-11.2301982,7.8155160,7.51214057,
-0.5057505,
-3.3707125,
C-0.6759166,1.0482656,0.3621876,
-0.3105902,0.0438410,
CO. 0738947,
-0.0646936,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0-0,
o. O/
CO. 0,0.0,0.0,0.0,0.0,
DATA RXI/1.3625952,11.5941040,27.3342937,28.0870086,
C7.27JL4615,
-9.2602472,
-5.0954462,3.2326498,1.8975352,
C-1.2813604,
-0.1532672,
-0.4830313,0.7614804,0.1979014,
CO. 0940330,
-0.0312132,0.0,0.0,0.0,0.0,0-0,0-0,0-0,0-0,
o. O/
CO, 0,0.0,0.0,0,0,0.0,
DATA WSI/-0.0280463,
-0.1622055
-0.4980021
-0.5075768,
-0.23080-/1
C, 0.1128849,0.1085069,
-0.0097627,
-0.0466855,
-0.0073049,
CO. 0151408,0.0052313,
-0.0044861,0.0006332,0.0010673,
C-0.0009344,0.0,0.0,0.0,0.0,0.0,0-0,0-0,0-0,0-0,0-0,
CO. 010.010.010.0/
DATA WSQ/0.0196809,0.1674616,0.3948080,0.4056800,0.1050267,
C-0.1337521,
-0.0185076,
-0.0735970,0.0466914,0.0274074,
C-0.0069'768,0.0109986,0.0028584,
-0.0004508
-0.0022137,0,0013582,
C, 0.0,0.0,0.0,0.0,0.0,
o. o, o. o, o. o, o. olo. olo. Olo. 0,0.0,0.0/
ONE SKYWAVE STANDARD DEVIATION
P=SQRT(2-0)
TWO SKYWAVES STANDARD DEVIATION
P=1.0
THREE SKYWAVES STANDARD DEVIATION
P=SQRT(2.0/3.0)
IQ=899
QPEN(6, FILE=IOUTPUT')
RRR=0.0
WN=0.0
INTF=O
IPH=O
ISL=O
ALT=1.0
IBR=O
ILS=O


IAGC=0
KLV=7.5
SSL=SQRT(4.0)
AMP=0.2
TT=0.0
LLL=0.0
SOS=-1.0
NNN=3
CST=1.25
CED=1.0
C=CST
NUS=0
MMM=0
MEA=0
LC2=0
LC3=0
MEM=0
LC1=0
JD=0
ETR=0.0
ETI=0.0
ETR1=0.0
ET11=0.0
LPD=0.0
IPD=0.0
LPDD=0
NN=1
LIS=57
THR=0.95
L1=88
NLS=0.0
NDET=0.0
LP0=0.0
JTO=0
NUMS=0.0
L2=89
BMW=0.0
L3=90
LX=90
MDEL=22
POS=-1.0
TOTMS=0.0
AMSE=0.0
IMPL=16
ILOOP=1100
TAMSE=0.0
DCGAIN=325623.4
DCGAIN=893.06
DCGAIN=1.0/DCGAIN
CALL G05CBF(IQ)
DO 7 I=1,5
RAYLI1(I)=0.0
RAYLI2(I)=0.0
RAYLI3(I)=0.0
RAYLI4(I)=0.0
RAYLI5(I)=0.0
RAYLI6(I)=0.0
DI1(I)=0.0
DY1(I)=0.0
DZ1(I) = 0.0
DH1(I) = 0.0
DQ1(I) = 0.0
DX1(I) = 0.0

CONTINUE
DO 31 I = 1, 50
HNOS1(I) = 0.0
HNOSR(I) = 0.0
YTR(I) = 0.0
YTI(I) = 0.0
DO 35 J = 1, 2
DO 35 I = 1, 31
AR1(I, J) = 0.0
A11(I, J) = 0.0

CONTINUE
DO 8 J = 1, ILOOP
RAY1(J) = 0.0
RAY2(J) = 0.0
RAY3(J) = 0.0
RAY4(J) = 0.0
RAY5(J) = 0.0
RAY6(J) = 0.0
DO 8999 J = 1, 5
A1(J) = 0.0
A2(J) = 0.0
A3(J) = 0.0
A4(J) = 0.0
A5(J) = 0.0
A6(J) = 0.0

CONTINUE
SR = 2.4
C DEL = 1.0
DEL1 = 1.0
C DEL1 = 1.5
DEL2 = 3.0
DEL1 = 1.0
DEL2 = 2.0
IDEL1 = INT(DEL1 * 2 * SR)
IDEL2 = INT(DEL2 * SR * 2)
IMP1 = IMPL + IDEL2
DO 32 I = 1, IMP1
DO 32 J = 1, IMP1
TR(I, J) = 0.0

CONTINUE
IMPES = (2 * IMPL + IDEL2 - 1) / 2
INPES = 31
INPES = 31
SOS = -1.0
ISTEP = 96
STEP = 1.0 / ISTEP
CEE = 0.03
C THETA = 0.85
ETA = 0.04
LS2 = 0
CE1 = 0.01
SUMT1 = 0.0
THTF = 0.97
THHO = (1 - THTF)**2
THH1 = (1 - THTF**2)
THETA = 0.97
THO = (1 - THETA)**2
TH1 = (1 - THETA**2)
DO 63 I = 1, 100
SSR(I) = 1.0
SSQ(I) = 1.0
SSRER(I) = 1.0
SSREI(I) = 1.0
RECR(I) = 0.0
RECI(I) = 0.0
KL = 2 * IMPES
KLP = KL + 2
KLM = KL + 1
N = 17
ISBR = 0
NK = 0.0
DO 811 I = 1, 31
NK = NK + 1
GO TO (1, 1, 2, 1, 1, 2, 1, 1, 2, 2, 1, 2, 2, 1, 2, 1, 1, 1, 2, 2, C2, 2, 1, 2, 2, 1, 2, 2, 1), NK
1 SSR(NK) = 1.0
GO TO 811
2 SSR(NK) = -1.0
811 CONTINUE
DC = ((SQRT(32.0)) - 1.0) / 31.0
DO 11 I = 1, 31
11 SSR(I) = SSR(I) + DC
DO 30 I = 1, 31
DO 25 J = 1, 31
N1 = I + J - 1
IF (N1 - 31) 23, 23, 21
N1 = N1 - 31
23 SB1(N1, I) = SSR(J)
25 CONTINUE
30 CONTINUE
DO 40 I = 1, 31
DO 44 J = 1, 31
M1 = J + 15
N1 = I + J - 1
IF (N1 - 31) 42, 42, 41
N1 = N1 - 31
41 N1 = N1 - 31
42 IF (M1 - 31) 43, 43, 39
39 M1 = M1 - 31
43 SB2(N1, I) = SSR(M1)
44 CONTINUE
40 CONTINUE
MNL = 1.0
SUMB = 0.0
DO 33 I = 1, 31
33 SUMB = SUMB + SSR(I) * SB1(I, 1)
WRITE (6, 948) IMP1
948 FORMAT (1H, 'IMP1=', 15)
NO = 0.0
L = 76
MIS = 85
HAH = 0.025
BMAX = 0.0
N7 = 30
N8 = 0.0
WRITE (6, 772) CEE, ETA, THETA, N, IQ, HAH
C'N=',I3,3X,'IQ=',I5,3X,'HAH=',F10.5)
WRITE(6,773)DEL,SR,LOOP,DEL1,DEL2
773 FORMAT(1H , 'DEL=',F6.3,3X,'SR=',F6.3,3X,'LOOP=',I3/
C'DEL1=',F4.2,3X,'DEL2=',F4.2)
DO 64 I=1,31
YDRI(I)=0.0
YDQI(I)=0.0
Y2RI(I)=0.0
Y2QI(I)=0.0
ETOTRI(I)=0.0
ETOTII(I)=0.0
YRR(I)=0.0
YQR(I)=0.0
Y1RI(I)=0.0
Y1QI(I)=0.0
CRI(I)=0.0
CRI(I)=0.0
AIR(I)=0.0
AIR(I)=0.0
BR(I)=0.0
BR(I)=0.0
FIRI(I)=0.0
FIRI(I)=0.0
FII(I)=0.0
FII(I)=0.0
EIRI(I)=0.0
EIRI(I)=0.0
YRP(I)=0.0
YQP(I)=0.0
CONTINUE
DO 5555 I=1,16
TTTR(I)=TXRDD(I)
TTTI(I)=TXIDD(I)
TTR(I)=TXRD(I)
TTTI(I)=TXID(I)
5555 TTTI(I)=TXID(I)
NSOK=3
WRITE(6,210)NSOK
210 FORMAT(1H 'NO. OF SKYWAVES=',I2)
FF2=0.0
FF2=1.0
FF3=0.0
FF3=1.0
X2=0.0
X3=0.0
DO 9 J=1,50
DI1(1)=G05DDF(0.0,P)-(RAYLI1(1)*COEFF(1)
C+RAYLI1(2)*COEFF(2))
DI1(2)=DI1(1)-(RAYLI1(3)*COEFF(3)+RAYLI1(4)*COEFF(4))
A1(2)=DI1(2)-(RAYLI1(5)*COEFF(5))
DQ1(1)=G05DDF(0.0,P)-(RAYLI2(1)*COEFF(1)
C+RAYLI2(2)*COEFF(2))
DQ1(2)=DQ1(1)-(RAYLI2(3)*COEFF(3)+RAYLI2(4)*COEFF(4))
A2(2)=DQ1(2)-(RAYLI2(5)*COEFF(5))
DH1(1)=G05DDF(0.0,P)-(RAYLI3(1)
C+RAYLI3(2)*COEFF(2))
DH1(2)=DH1(1)-(RAYLI3(3)*COEFF(3)+RAYLI3(4)*COEFF(4))
A3(2)=DH1(2)-(RAYLI3(5)*COEFF(5))
DX1(1)=G05DDF(0.0,P)-(RAYLI4(1)*COEFF(1)
C+RAYLI4(2)*COEFF(2))
DX1(2)=DX1(1)-(RAYLI4(3)*COEFF(3)+RAYLI4(4)*COEFF(4))
A4(2)=DX1(2)-(RAYLI4(5)*COEFF(5))
DY1(1)=G05DDF(0.0,P)-(RAYLI5(1)*COEFF(1)
C*RAYL15(2)*COEFF(2))
  DY1(2)=DY1(1)-(RAYL15(3)*COEFF(3)+RAYL15(4)*COEFF(4))
  A5(2)=DY1(2)-(RAYL15(5)*COEFF(5))
  DZ1(1)=G05DDF(0.0,P)-(RAYL16(1)*COEFF(1))
C*RAYL16(2)*COEFF(2))
  DZ1(2)=DZ1(1)-(RAYL16(3)*COEFF(3)+CRAYL16(4)*COEFF(4))
  A6(2)=DZ1(2)-(RAYL16(5)*COEFF(5))
RAYL11(5)=A1(2)
RAYL11(4)=RAYL11(3)
RAYL11(3)=DI1(2)
RAYL11(2)=RAYL11(1)
RAYL12(5)=A2(2)
RAYL12(4)=RAYL12(3)
RAYL12(3)=DQ1(2)
RAYL12(2)=RAYL12(1)
RAYL12(1)=DQ1(1)
RAYL13(5)=A3(2)
RAYL13(4)=RAYL13(3)
RAYL13(3)=DQ1(2)
RAYL13(2)=RAYL13(1)
RAYL13(1)=DQ1(1)
RAYL14(5)=A4(2)
RAYL14(4)=RAYL14(3)
RAYL14(3)=DX1(2)
RAYL14(2)=RAYL14(1)
RAYL14(1)=DX1(1)
RAYL15(5)=A5(2)
RAYL15(4)=RAYL15(3)
RAYL15(3)=DY1(2)
RAYL15(2)=RAYL15(1)
RAYL15(1)=DI1(1)
RAYL16(5)=A6(2)
RAYL16(4)=RAYL16(3)
RAYL16(3)=DZ1(2)
RAYL16(2)=RAYL16(1)
RAYL16(1)=DZ1(1)
A5(2)=A5(2)*DCGAIN
A6(2)=A6(2)*DCGAIN
A1(2)=A1(2)*DCGAIN
A2(2)=A2(2)*DCGAIN
A3(2)=A3(2)*DCGAIN
A4(2)=A4(2)*DCGAIN
CONTINUE
DO 10 K=1, ILOOP
C IF(K,L.T.275)GO TO 9999
C P=SQRT(2.0/3.0)
  DI1(1)=G05DDF(0.0,P)-(RAYL11(1)*COEFF(1))
C*RAYL11(2)*COEFF(2))
  DI1(2)=DI1(1)-(RAYL11(3)*COEFF(3)+RAYL11(4)*COEFF(4))
  A1(1)=DI1(2)-(RAYL11(5)*COEFF(5))
  DQ1(1)=G05DDF(0.0,P)-(RAYL12(1)*COEFF(1))
C*RAYL12(2)*COEFF(2))
  DQ1(2)=DQ1(1)-(RAYL12(3)*COEFF(3)+RAYL12(4)*COEFF(4))
  A2(1)=DQ1(2)-(RAYL12(5)*COEFF(5))
  DH1(1)=G05DDF(0.0,P)-(RAYL13(1)*COEFF(1))
C*RAYL13(2)*COEFF(2))
  DH1(2)=DH1(1)-(RAYL13(3)*COEFF(3)+RAYL13(4)*COEFF(4))
A3(1) = DH1(2) - (RAYL13(5) * COEFF(5))
DX1(1) = G05DDF(0.0, P) - (RAYL14(1) * COEFF(1))
C * RAYL14(2) * COEFF(2))
DX1(2) = DX1(1) - (RAYL14(3) * COEFF(3) + RAYL14(4) * COEFF(4))
A4(1) = DX1(2) - (RAYL14(5) * COEFF(5))
DY1(1) = G05DDF(0.0, P) - (RAYL15(1) * COEFF(1))
C * RAYL15(2) * COEFF(2))
DY1(2) = DY1(1) - (RAYL15(3) * COEFF(3) + RAYL15(4) * COEFF(4))
RAYL16(4) * COEFF(4))
A5(1) = DY1(2) - (RAYL15(5) * COEFF(5))
DY1(1) = G05DDF(0.0, P) - (RAYL16(1) * COEFF(1))
C * RAYL16(2) * COEFF(2))
DY1(2) = DY1(1) - (RAYL16(3) * COEFF(3) +
C * RAYL16(4) * COEFF(4))
A6(1) = DY1(2) - (RAYL16(5) * COEFF(5))
RAYL17(5) = A1(1)
RAYL17(4) = RAYL17(3)
RAYL17(3) = DI1(2)
RAYL17(2) = RAYL17(1)
RAYL17(1) = DI1(1)
RAYL18(5) = A2(1)
RAYL18(4) = RAYL18(3)
RAYL18(3) = DQ1(2)
RAYL18(2) = RAYL18(1)
RAYL18(1) = DQ1(1)
RAYL19(5) = A3(1)
RAYL19(4) = RAYL19(3)
RAYL19(3) = DH1(2)
RAYL19(2) = RAYL19(1)
RAYL19(1) = DH1(1)
RAYL20(5) = A4(1)
RAYL20(4) = RAYL20(3)
RAYL20(3) = DX1(2)
RAYL20(2) = RAYL20(1)
RAYL20(1) = DX1(1)
RAYL21(5) = A5(1)
RAYL21(4) = RAYL21(3)
RAYL21(3) = DY1(2)
RAYL21(2) = RAYL21(1)
RAYL21(1) = DY1(1)
RAYL22(5) = A6(1)
RAYL22(4) = RAYL22(3)
RAYL22(3) = DZ1(2)
RAYL22(2) = RAYL22(1)
RAYL22(1) = DZ1(1)
A5(1) = A5(1) * DCGAIN
A6(1) = A6(1) * DCGAIN
A1(1) = A1(1) * DCGAIN
A2(1) = A2(1) * DCGAIN
A3(1) = A3(1) * DCGAIN
A4(1) = A4(1) * DCGAIN
RAY1(K) = A1(1)
RAY2(K) = A2(1)
RAY3(K) = A3(1)
RAY4(K) = A4(1)
RAY5(K) = A5(1)
RAY6(K) = A6(1)
CONTINUE
C IQ=249
C CALL G05CBF(IQ)
SNO=30
P1=10**(-SNO/20.0)
P2=P1*(SQRT(2.5))
WRITE(6,6900)IQ,SNO
6900 FORMAT(1H ,'IQ=',I5,2X,'SNO=',F10.5)
DO 400 IRUN=1,ILOOP
C IF(IRUN.GT.300)GO TO 401
A1(1)=RAY1(IRUN)
A2(1)=RAY2(IRUN)
A3(1)=RAY3(IRUN)
A4(1)=RAY4(IRUN)
A5(1)=RAY5(IRUN)
A6(1)=RAY6(IRUN)
SLOP1=(A1(1)-A1(2))*STEP
SLOP2=(A2(1)-A2(2))*STEP
SLOP3=(A3(1)-A3(2))*STEP
SLOP4=(A4(1)-A4(2))*STEP
SLOP5=(A5(1)-A5(2))*STEP
SLOP6=(A6(1)-A6(2))*STEP
DO 300 ISLM=1,STEP
K=ISLM-1
DELT1=K*SLOP1
DELT2=K*SLOP2
DO 200 I=1,50
R(I)=0.0
Q(I)=0.0
DO 500 I=1,IMPL
R(I)=TXR(I)*(A1(2)+DELT1-A2(2)-DELT2)-TXI(I)*
C(A1(2)+DELT1+A2(2)+DELT2)
Q(I)=TXR(I)*(A1(2)+DELT1-A2(2)-DELT2)+TXI(I)*
C(A1(2)+DELT1-A2(2)-DELT2)
DELT3=K*SLOP3
DELT4=K*SLOP4
C IF(IRUN.LT.275)GO TO 7777
C DO 9986 I=1,IMPL
C TXRD(I)=TTR(I)*FF2
C9986 TXID(I)=TTI(I)*FF2
C IF(FF2.GE.1.0)GO TO 7777
C FF2=XX2
C IF(FF2.LE.0.0)GO TO 7777
C XX2=XX2+0.00002
7777 DO 600 I=1,IMPL
R(I+IDEL1)=R(I+IDEL1)+TXRD(I)*(A3(2)+DELT3-A4(2)-DELT4)
C-TXID(I)*(A3(2)+DELT3+A4(2)+DELT4)
600 Q(I+IDEL1)=Q(I+IDEL1)+TXRD(I)*(A3(2)+DELT3+A4(2)+DELT4)
C+TXID(I)*(A3(2)+DELT3-A4(2)-DELT4)
DELT5=K*SLOP5
DELT6=K*SLOP6
C IF(IRUN.LT.275)GO TO 8888
C DO 9987 I=1,IMPL
C TXRDD(I)=TTRR(I)*FF3
C9987 TXIDD(I)=TTTI(I)*FF3
C IF(FF3.GE.1.0)GO TO 8888
C FF3=XX3
C IF(FF3.LE.0.0)GO TO 8888
C XX3=XX3+0.00002
8888 DO 601 I=1,IMPL
R(I+IDEL2)=R(I+IDEL2)+TXRDD(I)*(A5(2)+DELT5-A6(2)-DELT6
(I+IDEL2) = Q(I + IDEL2) + TXRDD(I) * (A5(2) + DELT5 + A6(2) + DELT6)
C) - TXIDD(I) * (A5(2) + DELT5 + A6(2) + DELT6)

IM1 = IMP1 - 1
DO 596 I = 1, IMP1
DO 596 J = 1, IMP1
TR(I, IMP1 + 1 - J) = TR(I, IMP1 - J)
596
TI(I, IMP1 + 1 - J) = TI(I, IMP1 - J)
DO 700 I = 1, IMP1
TR(I,1) = (R(I) + Q(I)) * 0.5
700
TI(I, 1) = (Q(I) - R(I)) * 0.5
IMPP = IMPES + 2
IMPRP = IMPES + 1
C
P = P1
P = 0.032
IF(IRUN.LT.1)GO TO 55
IF(IRUN.EQ.1.AND.ISLM.LT.MIS)GO TO 55
P = P2
C
P = 0.00158
P = 0.05
55 DO 800 I = 1, IMPES
HNOSR(IMPP - I) = HNOSR(IMPRP - I)
800
HNOSI(IMPP - I) = HNOSI(IMPRP - I)
HNOSR(I) = GDPDF(0.0, P)
HNOSI(I) = GDPDF(0.0, P)
WR = 0.0
WI = 0.0
DO 900 I = 1, IMPES
WR = WR + HNOSR(I) * WSI(I) - HNOSI(I) * WSQ(I)
900
WI = WI + HNOSR(I) * WSQ(I) + HNOSI(I) * WSI(I)
POS = -POS
10 = 0
IF(POS.LT.0.0)GOTO 300
DO 2010 I = 1, IMP1, 2
10 = 10 + 1
YTR(IO) = 0.0
YTI(IO) = 0.0
DO 2000 J = 1, I
YTR(IO) = YTR(IO) + TR(J, I + 1 - J) * RXR(I + 1 - J) -
CTR(J, I + 1 - J) * RXI(I + 1 - J)
2000
YTI(IO) = YTI(IO) + TI(J, I + 1 - J) * RXR(I + 1 - J) +
CTR(J, I + 1 - J) * RXI(I + 1 - J)
YTR(IO) = YTR(IO) * 2.08333333D-4
2010
YTI(IO) = YTI(IO) * 2.08333333D-4
IM1 = IM1 - 1
DO 2030 I = 1, IM11, 2
10 = 10 + 1
YTR(IO) = 0.0
YTI(IO) = 0.0
K = I + 2
DO 2020 J = K, IMP1
YTR(IO) = YTR(IO) + TR(J, IMP1 + 2 + I - J) * RXR(IMP1 + 2 + I - J) -
CTR(J, IMP1 + 2 + I - J) * RXI(IMP1 + 2 + I - J)
2020
YTI(IO) = YTI(IO) + TI(J, IMP1 + 2 + I - J) * RXR(IMP1 + 2 + I - J) +
CTR(J, IMP1 + 2 + I - J) * RXI(IMP1 + 2 + I - J)
YTR(IO) = YTR(IO) * 2.08333333D-4
2030
YTI(IO) = YTI(IO) * 2.08333333D-4
IF(IAGC.EQ.1)THEN
ANSE = 0.0
DO 51010 I = 1, IMPES
ANSE = ANSE + YTR(I)**2 + YTI(I)**2
ANSE = 1.0 / SQRT(ANSE)
DO 51016 I = 1, IMPES
   YTR(I) = YTR(I) * ANSE
   YTI(I) = YTI(I) * ANSE
ENDIF
IF (IRUN .NE. 10 .OR. ISLM .NE. 1) GO TO 3111
DO 3112 I = 1, NPES
   AR1(I, 1) = YTR(I)
   AL1(I, 1) = YTI(I)
3112 IF (IRUN .GT. 149) GO TO 401
C IF (IRUN .LT. 147) GO TO 1212
C WRITE(2, 1214) (YTR(I), I = 1, IMPES)
C FORMAT(1H 6F10.5)
C WRITE(2, 1214) (YTI(I), I = 1, IMPES)
3113 DO 301 I = 1, 86
   SSR(88-I) = SSR(87-I)
301 SSQ(88-I) = SSQ(87-I)
C SL = 2 / SQRT(10)
C SSR(1) = 0.0
C SSQ(1) = 0.0
IF (IRUN .LE. 10) GO TO 300
IF (IRUN .LT. L) GO TO 1450
IF (IRUN .EQ. L AND. ISLM .LT. MIS) GO TO 1450
C IF (IRUN .LT. L3 .AND. IRUN .GE. L1) GO TO 1450
C IF (IRUN .EQ. L3 .AND. ISLM .LT. LIS) GO TO 1450
W = GO5DAF(0.0, 4.0)
IF (W .LT. 3 ) 61, 61, 71
71 SSR(1) = 3.0
   GO TO 1400
61 IF (W .LT. 2 ) 81, 81, 91
91 SSR(1) = -3.0
   GO TO 1400
81 IF (W .LT. 1 ) 101, 101, 111
111 SSR(1) = 1.0
   GO TO 1400
101 SSR(1) = -1.0
1400 W = GO5DAF(0.0, 4.0)
   IF (W .LT. 3 ) 62, 62, 72
72 SSQ(1) = 3.0
   GO TO 1500
62 IF (W .LT. 2 ) 82, 82, 92
92 SSQ(1) = -3.0
   GO TO 1500
82 IF (W .LT. 1 ) 102, 102, 112
112 SSQ(1) = 1.0
   GO TO 1500
102 SSQ(1) = -1.0
   GO TO 1500
1450 NO = NO + 1
   IF (NO .LT. 31 ) 110, 110, 117
117 NO = NO - 31
110 GO TO (3, 3, 4, 3, 4, 3, 3, 3, 4, 4, 4, 3, 4, 3, 4, 3, 4, 4, 4, 3, 4, 3, 4, 4, 3, 4, 3, 4, 4, 3), NO
SSR(1)=-1.0+DC

DO 766 I=1,99
RECI(101-I)=RECI(100-I)
RECR(101-I)=RECR(100-I)
S1=SSR(1)
S2=SSQ(1)
RECR(1)=0.0
IF(ISL.EQ.1.AND.IRUN.GT.105)THEN
SSR(1)=SSR(1)*SSL
SSQ(1)=SSQ(1)*SSL
ENDIF
RECI(1)=0.0
DO 1600 I=11,IMPES
RECR(1)=RECR(1)+(SSR(I)*YTR(I)-SSQ(I)*YTI(I))
RECI(1)=RECI(1)+(SSR(I)*YTI(I)+SSQ(I)*YTR(I))
RRR=RRR+RECR(1)**2+RECI(I)**2
WN=WN+WR**2+WI**2
ENDIF
RECR(1)=RECR(1)+WR
RECI(1)=RECI(I)+WI
IF(IBR.EQ.0)GO TO 6902
IF(Irun.LT.101)GO TO 6902
IF(LLL.EQ.1216)ALT=-ALT
IF(ALT.GT.0.0)GO TO 6902
C
IF(LLL.EQ.900)ILS=1
C
IF(IIS.EQ.1)GO TO 6902
IF(LLL.GT.700.AND.LLL.LT.755)THEN
X11=G05DAF(0.0,2.0)
IF(X11.GT.1.0)THEN
RECR(1)=KLV
ELSE
RECR(1)=-KLV
ENDIF
X22=G05DAF(0.0,2.0)
IF(X22.GT.1.0)THEN
RECI(1)=KLV
ELSE
RECI(1)=-KLV
ENDIF
ENDIF
6902 IF(INTF.EQ.1.AND.IRUN.GT.105)THEN
RECR(1)=RECR(1)+AMP*COS(2*(22/7.0)*1000*TT)
RECI(1)=RECI(I)+AMP*SIN(2*(22/7.0)*1000*TT)
TT=TT+(1.0/2400)
ENDIF
IF(IPH.EQ.1.AND.IRUN.GT.105)THEN
RR=RECR(1)**2+RECI(I)**2
RR=SQRT(RR)
ANG=ATAN(RECI(1)/RECR(1))
ANG=ABS(ANG)
IF(RECR(1).GT.0.0.AND.RECI(1).LT.0.0)THEN
ANG=-ANG
ENDIF
IF(RECR(1).LT.0.0.AND.RECI(1).GT.0.0)THEN
ANG=(22/7.0)-ANG
ENDIF
IF(RECR(1).LT.0.0.AND.RECI(1).LT.0.0)THEN
ANG=(22/7.0)+ANG
ENDIF
ANGN=ANG+(22/7.0)*(1/9.0)
RECR(1)=RR*COS(ANGN)
RECI(1)=RR*SIN(ANGN)
ENDIF
C IF(IRUN.LT.L3.AND.IRUN.GE.L1)GO TO 76
C IF(IRUN.EQ.L3.AND.ISLM.LT.LIS)GO TO 76
C IF(IRUN.EQ.L3.AND.ISLM.EQ.LIS)GO TO 718
IF(IRUN.EQ.11)GO TO 300
IF(IRUN.EQ.12.AND.ISLM.LT.27)GO TO 300
IF(IRUN.LT.L)GO TO 76
IF(IRUN.EQ.L.AND.ISLM.LT.MIS)GO TO 76
IF(IRUN.EQ.L.AND.ISLM.EQ.MIS)GO TO 719
IF(IRUN.GE.L)GO TO 719
76 N7=N7+1
IF(N7-31)6120,73,6120
73 N2=32
DO 712 I=1,31
N2=N2-1
R1(N2)=RECR(I)
712 R2(N2)=RECI(I)
N7=N7-31
GO TO 810
IF(MNL.EQ.1)GO TO 70
DO 114 I=1,15
M=I+16
R3(M)=R1(I)
114 R4(M)=R2(I)
DO 709 I=1,31
YTRX(I)=0.0
709 YTIX(I)=0.0
DO 809 J=1,31
DO 809 I=1,31
YTRX(J)=YTRX(J)+R3(I)*SB2(I,J)
809 YTIX(J)=YTIX(J)+R4(1)*SB2(I,J)
DO 910 I=1,31
YTRX(I)=YTRX(I)/SUMB
910 YTIX(I)=YTIX(I)/SUMB
N8=N8+1
DO 911 I=1,31
YTRX(I,N8)=YTRX(I)
911 YTIX(I,N8)=YTIX(I)
70 DO 113 I=1,16
M=I+15
R3(I)=R1(M)
113 R4(I)=R2(M)
MNL=MNL+1
810 DO 708 I=1,31
YTRX(I)=0.0
708 YTIX(I)=0.0
DO 808 J=1,31
DO 808 I=1,31
YTRX(J)=YTRX(J)+R1(I)*SB1(I,J)
707 YTIX(J)=YTIX(J)+R2(1)*SB1(I,J)
808 CONTINUE
DO 909 I=1,31
YTRX(I)=YTRX(I)/SUMB
909 YTIX(I)=YTIX(I)/SUMB
N8=N8+1
IF(IRUN .NE. 89) GO TO 365

N8 = N8 - 3

365 DO 916 I = 1, INPES
YTR1(I, N8) = YTRX(I)

916 YTII(I, N8) = YTIX(I)

C WRITE(2, 251) ISLM, IRUN
C251 FORMAT(IH, 'ISLM=', I3, 'IRUN=', I3)

6120 IF(NUMS .EQ. 1) GO TO 1001
IF(NUMS .EQ. 2) GO TO 1001
IF(NUMS .EQ. 3) GO TO 1003

719 IF(IRUN .NE. L OR ISLM .NE. MIS) GO TO 1003

718 LD = NB - 1
LF = N8 - 2
DO 1102 I = 1, INPES
A = (YTR1(I, N8) + YTR1(I, LD) + YTR1(I, LF)) / 3.0
B = (YTII(I, N8) + YTII(I, LD) + YTII(I, LF)) / 3.0
N8 = N8 - 3
GO TO 1003

1011 ISUB = 3
DO 50 J = 1, N8
ALFR = YTR1(I, J)
ALFI = YTII(I, J)
IF(J .NE. 1) GO TO 48
ALFDR = 0.0
ALFDI = 0.0
NX = 0.0
ALFPR = ALFR
ALFPI = ALFI

48 HO = 6.0 / ((NX + 2.0) * (NX + 1.0))
H1 = (2 * (2 * NX + 1.0)) / ((NX + 2.0) * (NX + 1.0))
EPALR = ALFR - ALFPR
EPALI = ALFI - ALFPI
ALFDR = ALFDR + HO * EPALR
ALFDI = ALFDI + HO * EPALI
ALFPR = ALFPR + ALFDR * H1 * EPALR
ALFPI = ALFPI + ALFDI * H1 * EPALI
NX = NX + 1.0
CONTINUE

49 CONTINUE
C AR(I) = ALFPR
C AI(I) = ALFPI
C BR(I) = ALFPR - ALFDR * N8
C BI(I) = ALFPI - ALFDI * N8
C CR(I) = ALFPR - ALFDR * N9
C CI(I) = ALFPI - ALFDI * N9
50 CONTINUE

NMIS = N8 / 3.0
NHR = NMIS
NFH = 1 + NHR
NHRR = 2 * NMIS
NHHR = 1 + NHRR
NHRRR = 3 * NMIS
DO 8990 I = 1, INPES
DO 8990 J = 1, NHR
AR(I) = AR(I) + YTR1(I, J)

8990 AI(I) = AI(I) + YTII(I, J)
DO 8991 I = 1, INPES
DO 8991 J = NFH, NHRR
CR(I) = CR(I) + YTR1(I,J)
CI(I) = CI(I) + YTI1(I,J)
DO 8992 I = 1, INPES
DO 8992 J = NFHH, NHRRR
BR(I) = BR(I) + YTR1(I,J)
8992 BI(I) = BI(I) + YTI1(I,J)
DO 8993 0 = 1, INPES
AR(I) = AR(I) / NMIS
AI(I) = AI(I) / NMIS
BR(I) = BR(I) / NMIS
BI(I) = BI(I) / NMIS
CR(I) = CR(I) / NMIS
8993 CI(I) = CI(I) / NMIS
WRITE(6,78)ISUB
78 FORMAT(1H , 'THE SUBSPACE IS=', 12)
GO TO 15
DO 245 I = 1, 31
IF(ABS(AR(I)) - HAH) 232, 232, 233
232 AR(I) = 0.0
233 AI(I) = 0.0
236 AI(I) = 0.0
237 BR(I) = 0.0
239 BI(I) = 0.0
240 CR(I) = 0.0
241 CI(I) = 0.0
244 CI(I) = 0.0
245 CONTINUE
JAH = 0.0
INPS = INPES
397 IF(AR(INPS).NE.0.0)GO TO 396
IF(AI(INPS).NE.0.0)GO TO 396
IF(BR(INPS).NE.0.0)GO TO 396
IF(BI(INPS).NE.0.0)GO TO 396
IF(CR(INPS).NE.0.0)GO TO 396
IF(CI(INPS).NE.0.0)GO TO 396
JAH = JAH + 1.0
INPS = INPS - 1.0
GO TO 397
396 INPES = INPES - JAH
LNAE = INPES + 1.0
DO 20 I = LNAE, 31
YRP(I) = 0.0
20 YQP(I) = 0.0
15 DOTR = 0.0
IF(MEA .NE. 1)GO TO 325
C GO TO 281
IF(NUS.EQ.1)GO TO 281
DOTR = 0.0
DOTI = 0.0
ABYR = 0.0
ABY1R = 0.0
DO 277 I = 1, INPES
DOTR = DOTR + YTR1(I,5)*YTR1(I,6) + YTI1(I,5)*YTI1(I,6)
DOTI = DOTI + YTR1(I,5)*YTI1(I,6) - YTI1(I,5)*YTR1(I,6)
ABYR = ABYR + YTR1(I,5)**2 + YTI1(I,6)**2
ABY1R = ABY1R + YTR1(I,6)**2 + YTI1(I,6)**2
ANGL = ((DOTR**2 + DOTI**2) / (ABYR*ABY1R))**0.5
IF(ANGL. LT. THR) THEN
C IF(ANGL. GT. THR) GO TO 190
   DO 278 I=1, INPES
      AR(I) = YTR1(I, 14)
      AI(I) = YTI1(I, 14)
      BR(I) = YTR1(I, 15)
   278 BI(I) = YTI1(I, 15)
ELSE
   C DO 279 I=1, INPES
      AR(I) = YTR1(I, 6)
   279 AI(I) = YTI1(I, 6)
ENDIF
   GO TO 325
   DO 281 I=1, INPES
      AR(I) = YTR1(I, 15)
   281 AI(I) = YTI1(I, 15)
   IPD = 0
   DOTI = 0.0
   DOTR = 0.0
   ABYR = 0.0
   ABY1R = 0.0
   DO 999 I=1, INPES
      DOTR = DOTR + AR(I) * BR(I) + AI(I) * BI(I)
      DOTI = DOTI + AR(I) * BI(I) - AI(I) * BR(I)
      ABYR = ABYR + AR(I)**2 + AI(I)**2
      999 ABY1R = ABY1R + BR(I)**2 + BI(I)**2
   ANGL = ((DOTR**2 + DOTI**2) / (ABYR * ABY1R))**0.5
   WRITE(6, 966) ANGL, ABYR, ABY1R
  966 FORMAT(1H 'ANGL, =' F10.5, 1X, 'I A I=' F10.5, 1X, 'I B I=' F10.5)
C IF(ABY1R. GT. ABYR) THEN
   C DO 166 I=1, INPES
      YTR1(I, 20) = AR(I)
   C YTI1(I, 20) = AI(I)
   C AR(I) = BR(I)
   C AI(I) = BI(I)
   C BR(I) = YTR1(I, 20)
   C166 BI(I) = YTI1(I, 20)
C ENDIF
C IF(ANGL. GE. 0.95) GO TO 401
C DO 8888 I = 1, INPES
C YTR1(I, 4) = AR(I)
C C8888 YTI1(I, 4) = AI(I)
C DO 888 I = 1, INPES
C AR(I) = YTR1(I)
C C888 AI(I) = YTI1(I)
BMOD = 0.0
LS2 = 0
ETA = 0.04
JD = 1
   DO 322 I = 1, INPES
   BMOD = BMOD + AR(I)**2 + AI(I)**2
   BMOD = 1.0 / SQRT(BMOD)
   DO 344 I = 1, INPES
   AR(I) = AR(I) * BMOD
   344 AI(I) = AI(I) * BMOD
   IF(ANGL. LT. THR) GO TO 793
C IF(NUMS. EQ. 2. AND. ANGL. GE. THR) GO TO 792
C GO TO 401
C IF(ANGL. GE. THR) GO TO 1001
CNI=0.0
CNR=0.0
C    IF(NUMS.NE.2)GO TO 7678
C    DO 7679 I=1,INPES
C    AR(I)=YTR1(I,2)
C    AI(I)=YTI1(I,2)
C    BR(I)=YTR1(I,3)
C7679   BI(I)=YTI1(I,3)
C    GO TO 1001
C    IF(NUMS.EQ.2)GO TO 362
C    IF(NUMS.NE.2)GO TO 7678
C    DO 7679 I=1,INPES
C    L7678AR(I)=YTR1(I,4)
C    AI(I)=YTI1(I,4)
C    BR(I)=YTR(I)
C7678   BI(I)=YTI(I)
    BMOD=0.0
    DO 8886 I=1,INPES
8886   BMOD=BMOD+AR(I)**2+AI(I)**2
    BMOD=1.0/SQRT(BMOD)
    DO 8885 I=1,INPES
8885   AR(I)=AR(I)*BMOD
    AI(I)=AI(I)*BMOD
    DO 2222 I=1,INPES
2222   CNR=CNR+BR(I)*AR(I)+BI(I)*AI(I)
2222   CNI=CNI+BI(I)*AR(I)-BR(I)*AI(I)
    DO 2255 I=1,INPES
2255   BR(I)=BR(I)-CNR*AR(I)+CNI*AI(I)
2255   BI(I)=BI(I)-CNR*AI(I)-CNI*AR(I)
    BMOD=0.0
    DO 2333 I=1,INPES
2333   BMOD=BMOD+(BR(I)**2+BI(I)**2)
    BMOD=1.0/SQRT(BMOD)
    DO 2424 I=1,INPES
2424   BR(I)=BR(I)*BMOD
    BI(I)=BI(I)*BMOD
    DO 189 I=1,INPES
189    YTR1(I,8)=AR(I)
189    YTI1(I,8)=AI(I)
189    YTR1(I,9)=BR(I)
189    YTI1(I,9)=BI(I)
    ALFR=0.0
    ALFI=0.0
    BETR=0.0
    BETI=0.0
    DO 8030 I=1,INPES
8030   ALFR=ALFR+YTR1(I,8)*CR(I)+CI(I)*YTI1(I,8)
8030   ALFI=ALFI+YTR1(I,8)*CI(I)-CR(I)*YTI1(I,8)
8030   BETR=BETR+YTR1(I,9)*CR(I)+CI(I)*YTI1(I,9)
8030   BETI=BETI+YTR1(I,9)*CI(I)-CR(I)*YTI1(I,9)
DO 8031 I = 1, INPES
FIR(I) = ALFR * YTR1(I, 8) - ALFI * YTR1(I, 8) + BETR * YTR1(I, 9) - BETI * YTR1(I, 9)
FII(I) = ALFR * YTR1(I, 8) + ALFI * YTR1(I, 8) + BETR * YTR1(I, 9) + BETI * YTR1(I, 9)
CR(I) = CR(I) - FIR(I)
CI(I) = CI(I) - FII(I)
8031 WRITE(2, 8992)(CR(I), I = 1, INPES)
8992 FORMAT(1H6F10.5)
WRITE(2, 8992)(CI(I), I = 1, INPES)
BMW = 0.0
DO 1993 I = 1, INPES
BMW = BMW + CR(I)**2 + CI(I)**2
1993 WRITE(2, 1994)BMW
1994 FORMAT(1H6F10.5)
WRITE(6, 1985)BMW
1985 FORMAT(1H6F10.5)
WRITE(2, 1986)(AR1(I, 1), I = 1, INPES)
1986 FORMAT(1H6F10.5)
WRITE(2, 1986)(AI1(I, 1), I = 1, INPES)
WRITE(2, 1986)(YTR1(I, 8), I = 1, INPES)
WRITE(2, 1986)(YT11(I, 8), I = 1, INPES)
WRITE(2, 1986)(YTR1(I, 9), I = 1, INPES)
WRITE(2, 1986)(YT11(I, 9), I = 1, INPES)
BTADR = 0.0
BTADI = 0.0
ALFDR = 0.0
ALFDI = 0.0
NUMS = 2
LS1 = 0
IF(BMW.LT.0.00600) GO TO 1001
BMW = 0.0
DO 8032 I = 1, INPES
BMW = BMW + CR(I)**2 + CI(I)**2
8032 BMW = (1.0/SGRT(BM0D))BMW
DO 8033 I = 1, INPES
CR(I) = CR(I) * BMW
8033 CI(I) = CI(I) * BMW
WRITE(2, 9001)(AR1(I), I = 1, INPES)
9001 FORMAT(1H6F10.5)
WRITE(2, 9001)(AI1(I), I = 1, INPES)
WRITE(2, 9001)(BR1(I), I = 1, INPES)
WRITE(2, 9001)(BI1(I), I = 1, INPES)
ALFDR = 0.0
ALFDI = 0.0
DO 2012 I = 1, INPES
YR(I) = YTR1(I, N8)
2012 YQ(I) = YT11(I, N8)
BTADR = 0.0
BTADI = 0.0
ETA = 0.04
LS3 = 0
CTDR = 0.0
CTDI = 0.0
NUMS = 3
GO TO 1003
1001 RIDR = 0.0
IF(JTO.EQ.1) GO TO 298
RIDR1 = 0.0
RID1 = 0.0
DO 295 I = 1, INPES
RID1 = RIDR1 + (SSRE1(I) * Y2R(I) - SSRE1(I) * Y2Q(I))
295 \[ \text{RIDI1} = \text{RIDI1} + (\text{SSRER(I)} \times \text{Y2Q(I)} + \text{SSREI(I)} \times \text{Y2R(I)}) \]

\[ \text{ERIR1} = \text{REC1(1+N)} - \text{RIDR1} \]

\[ \text{ERII1} = \text{REC1(1+N)} - \text{RIDI1} \]

DO 296 I = 1, INPES

\[ \text{YYR(I)} = \text{Y2R(I)} + \text{CE1} \times (\text{ERIR1} \times \text{SSRER(I)} + \text{ERII1} \times \text{SSREI(I)}) \]

\[ \text{YYQ(I)} = \text{Y2Q(I)} + \text{CE1} \times (\text{ERII1} \times \text{SSRER(I)} - \text{ERIR1} \times \text{SSREI(I)}) \]

DO 93 I = 1, INPES

\[ \text{YYR(I)} = \text{YYR(I)} - \text{Y2R(I)} \]

\[ \text{YYQ(I)} = \text{YYQ(I)} - \text{Y2Q(I)} \]

\[ \text{YDR(I)} = \text{YDR(I)} + \text{THHO} \times \text{EPAR(I)} \]

\[ \text{YDQ(I)} = \text{YDQ(I)} + \text{THHO} \times \text{EPAI(I)} \]

\[ \text{Y2R(I)} = \text{Y2R(I)} + \text{YDR(I)} + \text{THH1} \times \text{EPAR(I)} \]

\[ \text{Y2Q(I)} = \text{Y2Q(I)} + \text{YDQ(I)} + \text{THH1} \times \text{EPAI(I)} \]

\[ \text{Y2NR(I)} = \text{Y2R(I)} + (N-1) \times \text{YDR(I)} \]

\[ \text{Y2NQ(I)} = \text{Y2Q(I)} + (N-1) \times \text{YDQ(I)} \]

IF(IPD.NE.1.O)GO TO 298

\[ \text{LC1} = \text{LC1} + 1 \]

IF(MEA.EQ.1)GO TO 118

IF(LC1.GT.1500)GO TO 118

DO 149 I = 1, INPES

\[ \text{AR(I)} = (1 - (1.0/LC1)) \times \text{AR(I)} + (1.0/LC1) \times \text{YYR(I)} \]

\[ \text{AI(I)} = (1 - (1.0/LC1)) \times \text{AI(I)} + (1.0/LC1) \times \text{YYQ(I)} \]

GO TO 152

IF(LC1.GT.1500)GO TO 119

\[ \text{LC2} = \text{LC2} + 1 \]

DO 150 I = 1, INPES

\[ \text{CR(I)} = (1 - (1.0/LC2)) \times \text{CR(I)} + (1.0/LC2) \times \text{YYR(I)} \]

\[ \text{CI(I)} = (1 - (1.0/LC2)) \times \text{CI(I)} + (1.0/LC2) \times \text{YYQ(I)} \]

GO TO 152

IF(LC1.GT.3000)GO TO 120

\[ \text{LC3} = \text{LC3} + 1 \]

DO 151 I = 1, INPES

\[ \text{BR(I)} = (1 - (1.0/LC3)) \times \text{BR(I)} + (1.0/LC3) \times \text{YYR(I)} \]

\[ \text{BI(I)} = (1 - (1.0/LC3)) \times \text{BI(I)} + (1.0/LC3) \times \text{YYQ(I)} \]

GO TO 152

DO 153 I = 1, INPES

\[ \text{YR(I)} = \text{YYR(I)} \]

\[ \text{YQ(I)} = \text{YYQ(I)} \]

JTO = 1

\[ \text{LC1} = 0 \]

\[ \text{MMM} = 0 \]

\[ \text{C} = \text{CST} \]

\[ \text{ISBR} = 1 \]

\[ \text{LC2} = 0 \]

\[ \text{LC3} = 0 \]

\[ \text{LPDD} = 0 \]

GO TO 15

DO 299 I = 1, INPES

\[ \text{YPQ(I)} = \text{Y2NR(I)} \]

GO TO 632

298 JTO = 0

IF(NUMS.EQ.2)GO TO 1002

IF(NUMS.EQ.3)GO TO 1003

NUMS = 1.0

\[ \text{RIDI} = 0.0 \]

IF(MMM.EQ.0)C = 1.25

DO 4000 I = 1, INPES

\[ \text{RIDR} = \text{RIDR} + (\text{SSRER(I)} \times \text{YR(I)} - \text{SSREI(I)} \times \text{YQ(I)}) \]

299 YQP(I) = Y2NQ(I)

GO TO 632

298 JTO = 0

IF(NUMS.EQ.2)GO TO 1002

IF(NUMS.EQ.3)GO TO 1003

NUMS = 1.0

\[ \text{RIDI} = 0.0 \]

IF(MMM.EQ.0)C = 1.25

DO 4000 I = 1, INPES

\[ \text{RIDR} = \text{RIDR} + (\text{SSRER(I)} \times \text{YR(I)} - \text{SSREI(I)} \times \text{YQ(I)}) \]
4000  RIDI = RIDI + (SSRER(I) * YQ(I) + SSREI(I) * YR(I))
ERIR = RECR(1+N) - RIDR
ERII = RECI(1+N) - RIDI
DO 4010 I = 1, INPES
YR(I) = YR(I) + CEE * (ERIR * SSRER(I) + ERII * SSREI(I))
YQ(I) = YQ(I) + CEE * (ERII * SSRER(I) - ERIR * SSREI(I))
ALFR = 0.0
ALFI = 0.0
BETR = 0.0
BETI = 0.0
DO 4020 I = 1, INPES
ALFR = ALFR + AR(I) * YR(I) + YQ(I) * AI(I)
ALFI = ALFI + AI(I) * YR(I) - YQ(I) * AR(I)
BETR = BETR + BR(I) * YR(I) + YQ(I) * BI(I)
BETI = BETI + BI(I) * YQ(I) - YR(I) * BR(I)
DO 4030 I = 1, INPES
FIR(I) = ALFR * AR(I) - ALFI * AI(I)
FII(I) = ALFR * AI(I) + ALFI * AR(I)
EIR(I) = YR(I) - FIR(I)
EII(I) = YQ(I) - FII(I)
C661 ETOTR(I) = (1 - (1/NN)) * ETOTR(I) + (1/NN) * EIR(I)
ETOTI(I) = (1 - (1/NN)) * ETOTI(I) + (1/NN) * EII(I)
ETR1 = (1 - (1.0/NN)) * ETR1 + (1.0/NN) * (ABS(ERIRI))
ETI1 = (1 - (1.0/NN)) * ETI1 + (1.0/NN) * (ABS(ERIII))
ETR = (1 - (1.0/NN)) * ETR + (1.0/NN) * (ABS(ERIR))
ETI = (1 - (1.0/NN)) * ETI + (1.0/NN) * (ABS(ERII))
NN = NN + 1
LS2 = LS2 + 1
IMA = LS2
IF (LS2 .LT. 700) GO TO 4195
ETA = 0.01
4195 IF (LS2 .LT. 16000) GO TO 4196
ETA = 0.005
C ETA = 0.005
4196 DO 4040 I = 1, INPES
AR(I) = AR(I) + ETA * (ALFR * EIR(I) + ALFI * EII(I))
AII(I) = AII(I) + ETA * (ALFR * EII(I) - ALFI * EIR(I))
C IF (IMA .GT. 7500) GO TO 168
BMOD = 0.0
DO 4070 I = 1, INPES
BMOD = BMOD + AR(I)**2 + AI(I)**2
BMOD = 1.0 / SQRT(BMOD)
DO 4080 I = 1, INPES
AR(I) = AR(I) * BMOD
AII(I) = AII(I) * BMOD
IF (LS2 .NE. 6000) GO TO 204
C = CED
MMM = 1
MEA = 1
NUS = 1
DO 205 I = 1, INPES
YTR1(1, 5) = FIR(I)
YII(1, 5) = FII(I)
204 IF (IRUN .NE. L OR. ISLM .NE. MIS) GO TO 7778
ALFPR = ALFR
ALFDR = 0.0
ALFDI = 0.0
JD = 0
ALFPI = ALFI
294

C7778 IF(IRUN.NE.L.OR.ISLM.NE.M1S)GO TO 57
7778 IF(JD.NE.1)GO TO 57

ALFPR=ALFR
JD=0
ALFDR=0.0
ALFDI=0.0
ALFPI=ALFI
57

EPALR=ALFR-ALFPR
EPALI=ALFI-ALFPI
ALFDR=ALFDR+(THO)*EPALR
ALFDI=ALFDI+(THO)*EPALI
ALFPR=ALFPR+ALFDR+TH1*EPALR
ALFPI=ALFPI+ALFDI+TH1*EPALI
DO 4060 I=1,INPES
YR(I)=ALFPR*AR(I)-ALFPI*AI(I)
4060 YQ(I)=ALFPR*AI(I)+ALFPI*AR(I)

ALFNR=ALFPR+(N-1)*ALFDR
ALFNI=ALFPI+(N-1)*ALFDI
IF(ISBR.EQ.0)GO TO 288
IF(IMA.GT.580)GO TO 288
C=1.0
DO 289 I=1,INPES
YRP(I)=Y2NR(I)
289 YQP(I)=Y2NQ(I)
GO TO 632
288 DO 133 I=1,INPES
YRP(I)=ALFNR*AR(I)-ALFNI*AI(I)
133 YQP(I)=ALFNR*AI(I)+ALFNI*AR(I)
GO TO 632
1002 RIDR=0.0
IF(MMM.EQ.0)C=1.25
RIDI=0.0
DO 45 I=1,INPES
RIDR=RIDR+(SSRER(I)*YR(I)-SSREI(I)*YQ(I))
45 RIDI=RIDI+(SSRER(I)*YQ(I)+SSREI(I)*YR(I))
ERIR=RECRI(1+N)-RIDR
ERII=RECII(1+N)-RIDI
NUMS=2.0
C
C IF(IRUN.LT.40)GO TO 80
C
Z=ABS(ERIR)
C F=ABS(ERII)
C IF(Z.LE.0.1)GO TO 77
C IF(ERIR).LT.0.025)GO TO 77
C78 ERIR=-0.1
C GO TO 77
C79 ERIR=0.1
C77 IF(F.LE.0.1)GO TO 80
C IF(ERII).LT.0.1)GO TO 80
C83 ERII=-0.1
C GO TO 80
C84 ERII=0.1
C IF(IRUN.LE.30)GO TO 59
C IF(CEE.LE.0.025)GO TO 59
DO 46 I=1,INPES
YR(I)=YR(I)+CEE*(ERIR*SSRER(I)+ERII*SSREI(I))
46 YQ(I)=YQ(I)+CEE*(ERII*SSRER(I)-ERIR*SSREI(I))
ALFR=0.0
ALFI=0.0
BETR=0.0
BETI=0.0
DO 47 I=1,INPES
ALFR=ALFR+AR(I)*YR(I)+YQ(I)*AI(I)
ALFI=ALFI+AR(I)*YQ(I)-YR(I)*AI(I)
BETR=BETR+BR(I)*YR(I)+YQ(I)*BI(I)
BETI=BETI+BR(I)*YQ(I)-YR(I)*BI(I)
DO 51 I=1,INPES
FIR(I)=ALFR*AR(I)-ALFI*AI(I)+BETR*BR(I)-BETI*BI(I)
FII(I)=ALFR*AI(I)+ALFI*AR(I)+BETR*BI(I)+BETI*BR(I)
EIR(I)=YR(I)-FIR(I)
Ell(I)=YQ(I)-FII(I)
C DO 85 I=1,INPES
C ETOTR(I)=(1-(1/NN))*ETOTR(I)+(1/NN)*EIR(I)
C ETOTI(I)=(1-(1/NN))*ETOTI(I)+(1/NN)*EII(I)
ETR1=(1-(1.0/NN))*ETR1+(1.0/NN)*(ABS(ERIR1))
ETI1=(1-(1.0/NN))*ETI1+(1.0/NN)*(ABS(ERII1))
EIR=(1-(1.0/NN))*EIR+(1.0/NN)*(ABS(ERIR))
ETI=(1-(1.0/NN))*ETI+(1.0/NN)*(ABS(ERII))
NN=NN+1
LS1=LS1+1
IMA=LS1
IF(LS1.LT.700)GO TO 52
C ETOTR(I)=(1-(1/NN))*ETOTR(I)+(1/NN)*EIR(I)
C ETOTI(I)=(1-(1/NN))*ETOTI(I)+(1/NN)*EII(I)
ETR1=(1-(1.0/NN))*ETR1+(1.0/NN)*(ABS(ERIR1))
ETI1=(1-(1.0/NN))*ETI1+(1.0/NN)*(ABS(ERII1))
ETR=(1-(1.0/NN))*ETR+(1.0/NN)*(ABS(ERIR))
ETI=(1-(1.0/NN))*ETI+(1.0/NN)*(ABS(ERII))
NN=NN+1
LS1=LS1+1
IMA=LS1
IF(LS1.LT.700)GO TO 52
DO 60 I=1, INPES
BR(I)=BR(I)-BAR*AR(I)+BAI*AI(I)
60 BI(I)=BI(I)-BAI*AR(I)-BAR*AI(I)
BMOD=0.0
DO 65 I=1, INPES
BMOD=BMOD+(BR(I)**2+BI(I)**2)
BMOD=1.0/SQRT(BMOD)
DO 66 I=1, INPES
BR(I)=BR(I)*BMOD
66 BI(I)=BI(I)*BMOD
139 IF(IRUN.NE.L.OR.ISLM.NE.MIS)GO TO 67
ALFPR=ALFR
ALFPI=ALFI
JD=0
BTAPR=BETR
BTAPI=BETI
67 IF(JD.NE.1)GO TO 159
ALFPR=ALFR
ALFPI=ALFI
JD=0
BTAPR=BETR
BTAPI=BETI
159 EPALR=ALFR-ALFPR
EPAI=ALFI-ALFPI
EPBTR=BETR-BTAPR
EPBTI=BETI-BTAPI
ALFDR=ALFDR+(THO)*EPALR
ALFDI=ALFDI+(THO)*EPAI
ALFPR=ALFPR+ALFDR+TH1*EPALR
ALFPI=ALFPI+ALFDI+TH1*EPAI
BTADR=BTADR+(THO)*EPBTR
BTADI=BTADI+(THO)*EPBTI
BTAPR=BTAPR+BTADR+TH1*EPBTR
BTAPI=BTAPI+BTADI+TH1*EPBTI
DO 68 I=1, INPES
YR(I)=(ALFPR*AR(I)-ALFPI*AI(I))+(BTAPR*BR(I)-BTAPI*BI(I))
68 YQ(I)=ALFPR*AI(I)+ALFPI*AR(I)+BTAPR*BI(I)+BTAPI*BR(I)
ALFNR=ALFPR+(N-1)*ALFDR
ALFNI=ALFPI+(N-1)*ALFDI
BTANR=BTAPR+(N-1)*BTADR
BTANI=BTAPI+(N-1)*BTADI
IF(ISBR.EQ.0)GO TO 286
IF(IMA.GT.580)GO TO 286
C=1.0
DO 287 I=1, INPES
YRP(I)=Y2NR(I)
287 YQP(I)=Y2NQ(I)
GO TO 632
286 DO 69 I=1, INPES
YRP(I)=ALFNR*AR(I)-ALFNI*AI(I)+BTANR*BR(I)-BTANI*BI(I)
69 YQP(I)=ALFNR*AI(I)+ALFNI*AR(I)+BTANR*BI(I)+BTANI*BR(I)
GO TO 632
1003 RIDR=0.0
IF(MMM.EQ.0)C=1.25
NUMS=3.0
RIDI=0.0
DO 747 I=1, INPES
RIDR=RIDR+(SSRER(I)*YR(I)-SSREI(I)*YQ(I))
747 RIDI=RIDI+(SSRER(I)*YQ(I)+SSREI(I)*YR(I))
ERIR = RECR(1+N) - RIDR
ERII = RECI(1+N) - RIDI

C IF(IRUN.GT.L3.OR.IRUN.LT.L1) GO TO 8720
C IF(IRUN.EQ.L3.AND.ISLM.GT.L1) GO TO 8720

DO 748 I=1,INPES
    YR(I) = YR(I) + CEE*(ERIR*SSRER(I) + ERII*SSREI(I))

748    YQ(I) = YQ(I) + CEE*(ERII*SSRER(I) - ERIR*SSREI(I))
    ALFR = 0.0
    ALFI = 0.0
    BETR = 0.0
    CTR = 0.0
    CTI = 0.0
    BETI = 0.0

DO 2202 I=1,INPES
    ALFR = ALFR + AR(I)*YR(I) + YQ(I)*AI(I)
    ALFI = ALFI + AR(I)*YQ(I) - YR(I)*AI(I)
    CTR = CTR + CR(I)*YR(I) + YQ(I)*CI(I)
    CTI = CTI + CR(I)*YQ(I) - YR(I)*CI(I)
    BETR = BETR + BR(I)*YR(I) + YQ(I)*BI(I)
    BETI = BETI + BR(I)*YQ(I) - YR(I)*BI(I)

2202   BET = BET + BR(I)*YQ(I) - YR(I)*BI(I)

DO 2251 I=1,INPES
    FIR(I) = ALFR*AR(I) - ALFI*AI(I) + BETR*BR(I) - BETI*BI(I) +
            CCT*CR(I) - CTI*CI(I)
    FII(I) = ALFR*AI(I) + ALFI*AR(I) + BETR*BI(I) + BETI*BR(I) +
            CCT*CI(I) - CTI*CR(I)
    EIR(I) = YR(I) - FIR(I)
    EII(I) = YQ(I) - FII(I)

2251   ETR = (1 - (1.0/NN))*ETR + (1.0/NN)*ABS(ERIR)
    ETI = (1 - (1.0/NN))*ETI + (1.0/NN)*ABS(ERII)

NN = NN + 1
LS3 = LS3 + 1
IMA = LS3
IF(LS3.LT.700) GO TO 2301
ETA = 0.01

2301   IF(LS3.LT.16000) GO TO 2401
C = CED
MMM = 1
ETA = 0.005

2401   IF(LS3.LT.32000) GO TO 1988
ETA = 0.005

1988   DO 4021 I=1,INPES
    BR(I) = BR(I) + ETA*(BETR*ER(I) + BETI*EI(I))
    AR(I) = AR(I) + ETA*(ALFR*ER(I) + ALFI*EI(I))
    CR(I) = CR(I) + ETA*(CTR*ER(I) + CTI*EI(I))
    CI(I) = CI(I) + ETA*(CTR*ER(I) - CTI*ER(I))
    AI(I) = AI(I) + ETA*(ALFR*EI(I) - ALFI*ER(I))

4021   BI(I) = BI(I) + ETA*(BETR*EI(I) - BETI*ER(I))

C IF(IMA.GT.7500) GO TO 140
BMOD = 0.0

4031   DO 4031 I=1,INPES
    BMOD = BMOD + AR(I)**2 + AI(I)**2
    BMOD = 1.0/SQRT(BMOD)

4041   DO 4041 I=1,INPES
    AR(I) = AR(I) * BMOD
    AI(I) = AI(I) * BMOD

4041   DO 12 I=1,INPES
BAR = BAR + BR(I) * AR(I) + BI(I) * AI(I)

BAI = BAI + BI(I) * AR(I) - BR(I) * AI(I)

DO 36 I = 1, INPES
BR(I) = BR(I) - BAR * AR(I) + BAI * AI(I)
BI(I) = BI(I) - BAI * AR(I) - BAR * AI(I)
BMOD = 0.0

DO 37 I = 1, INPES
BMOD = BMOD + (BR(I) ** 2 + BI(I) ** 2)
BMOD = 1.0 / SQRT(BMOD)

DO 38 I = 1, INPES
BR(I) = BR(I) * BMOD
BI(I) = BI(I) * BMOD

CCR = 0.0
CCI = 0.0
CFR = 0.0
CFI = 0.0

DO 8034 I = 1, INPES
CCR = CCR + AR(I) * CR(I) + CI(I) * AI(I)
CCI = CCI - AR(I) * CI(I) - CR(I) * AI(I)
CFR = CFR + BR(I) * CR(I) + CI(I) * BI(I)
CFI = CFI + BR(I) * CI(I) - CR(I) * BI(I)

DO 8035 I = 1, INPES
FIR(I) = CCR * AR(I) - CCI * AI(I) + CFR * BR(I) - CFI * BI(I)
FII(I) = CCR * AI(I) + CCI * AR(I) + CFR * BI(I) + CFI * BR(I)

CR(I) = CR(I) - FIR(I)
CI(I) = CI(I) - FII(I)
BMOD = 0.0

DO 8036 I = 1, INPES
BMOD = BMOD + CR(I) ** 2 + CI(I) ** 2
BMOD = 1.0 / SQRT(BMOD)

DO 8037 I = 1, INPES
CR(I) = CR(I) * BMOD
CI(I) = CI(I) * BMOD

140 IF (IRUN . NE. L . OR. ISLM . NE. MIS) GO TO 777
ALFPR = ALFR
JD = 0
ALFPI = ALFI
CTPR = CTR
CTPI = CTI
BTAPR = BETR
BTAPI = BETI

C777 IF (IRUN . NE. LX . OR. ISLM . NE. LIS) GO TO 749
777 IF (JD . NE. 1) GO TO 749
ALFPR = ALFR
ALFPI = ALFI
CTPR = CTR
CTPI = CTI
BTAPR = BETR
JD = 0
BTAPI = BETI

749 EPALR = ALFR - ALFPR
EPALI = ALFI - ALFPI
EPBTR = BETR - BTAPR
EPBTI = BETI - BTAPI
EPCTR = CTR - CTPR
EPCTI = CTI - CTPI
ALFDR = ALFDR + (THO) * EPALR
ALFDI = ALFDI + (THO) * EPALI
ALFPR = ALFPR + ALFDR + TH1 * EPALR
ALFPI = ALFPI + ALFDI + TH1 * EPALI
BTADR = BTADR + (THO) * EPBTR
BTADI = BTADI + (THO) * EPBTI
BTAPR = BTAPR + BTADR + TH1 * EPBTR
BTAPI = BTAPI + BTADI + TH1 * EPBTI
CTDR = CTDR + (THO) * EPCTR
CTDI = CTDI + (THO) * EPCTI
CTPR = CTPR + CTDR + TH1 * EPCTR
CTPI = CTPI + CTDI + TH1 * EPCTI
DO 750 I = 1, INPES
   YR(I) = (ALFPR * AR(I) - ALFP I * AI(I)) + (BTAPR * BR(I) - BTAPI * BI(I)) +
            CCTPR * CR(I) - CTPI * CI(I)
750   YQ(I) = ALFPR * AI(I) + ALFP I * AR(I) + BTAPR * BI(I) + BTAPI * BR(I) +
            CCTPR * CI(I) + CTPR * CR(I)
ALFNR = ALFPR * (N-1) * ALFDR
ALFNI = ALFPI * (N-1) * ALFDI
BTANR = BTAPR * (N-1) * BTADR
BTANI = BTAPI * (N-1) * BTADI
CTNR = CTPR * (N-1) * CTDR
CTNI = CTPI * (N-1) * CTDI
IF(ISBR.EQ.0) GO TO 103
IF(IMA.GT.580) GO TO 103
C = 1.0
DO 188 I = 1, INPES
   YRP(I) = Y2NR(I)
188   YQP(I) = Y2NQ(I)
   GO TO 632
103 DO 134 I = 1, INPES
   YRP(I) = ALFNR * AR(I) - ALFNI * AI(I) + BTANR * BR(I) - BTANI * BI(I) +
            CCTNR * CR(I) - CTNI * CI(I)
134   YQP(I) = ALFNR * AI(I) + ALFNI * AR(I) + BTANR * BI(I) + BTANI * BR(I) +
            CCTNR * CI(I) + CTNI * CR(I)
632 DO 302 I = 1, KL
   SSRER(KLP-1) = SSRER(KLM-1)
302   SSRER(I) = SSR(N)
   SSREI(KLP-1) = SSREI(KLM-1)
   SSREI(I) = SSQ(N)
   AMSE = 0.0
   DO 303 I = 1, 31
503   AMSE = AMSE + (YTR(I) - YRP(I))**2 + (YTI(I) - YQP(I))**2
   IF(IRUN.LT.85 .OR. IRUN.GE.100) GO TO 666
   TAMSE = TAMSE + AMSE
666 IF(IRUN.LE.100) GO TO 665
   TOTMS = TOTMS + AMSE
665 GO TO 300
611 DO 615 I = 1, INPES
   Y1R(I) = YTR(I)
   Y1Q(I) = YTI(I)
615   YRP(I) = YTR(I)
612 DO 616 I = 1, INPES
   Y2R(I) = YTRX(I)
   Y2Q(I) = YTIX(I)
616   YRP(I) = YTR(I)
616   YQP(I) = YTI(I)
   GO TO 632
300 CONTINUE
   IMNU = 48 * IRUN
IF(LPDD.EQ.1) GO TO 8993
LPD = LPD + 1.0
IF(LP.DE. NNN) GO TO 8993
LPD = 1.0
NN = 1
C SUMT1 = 0.0
C DO 135 I = 1, INPES
C SUMT1 = SUMT1 + ABS(ETOTR(I)) + ABS(ETOTI(I))
C DO 86 I = 1, INPES
C ETOTR(I) = 0.0
C ETOTI(I) = 0.0
C IF((MA.LT.530) GO TO 297
C IF(NUMS.EQ.3) GO TO 297
ECT1 = ETR1 + ET11
ECT = ETR + ETI
C GO TO 297
C IF(ECT.LT.C*ECT1) GO TO 297
LPDD = 1
IPD = 1
C GO TO 401
C MEM = 0
C IF(MEM.EQ.1) GO TO 297
DO 148 I = 1, INPES
C MEM = 1
AR(I) = 0.0
AI(I) = 0.0
BR(I) = 0.0
BI(I) = 0.0
CR(I) = 0.0
CI(I) = 0.0
WRITE(6,169) ECT, ECT1
169 FORMAT(1H 'ECT=', F10.5, 2X, 'ECT1=', F10.5)
297 ETR1 = 0.0
ETI1 = 0.0
ETR = 0.0
ETI = 0.0
8993 SER = 10.0*Aalog10(1.0E-10+AMSE)
WRITE(6,250) IMNU, SER
250 FORMAXT(1H, IM, F10.5)
C WRITE(2,250) IMNU, SER, ETR1, ETI1, ETR, ETI
C250 FORMAT(1H, IM, F10.5, 2X, 'SUMT1=', F10.4)
C250 FORMAT(1H, 5F10-4)
C IF(IRUN.NE.30) GO TO 333
WRITE(6,963) P
963 FORMAT(1H, 'STANDARD DEVIATION OF NOISE ADDED=', F7.5)
333 A1(2) = A1(1)
C IF(SUMT1.LT.1.1) GO TO 297
C GO TO 297
C ECT1 = ETR1 + ETI1
C ECT = ETR + ETI
C IF(ECT.LT.0.1) GO TO 297
C IF(ECT.LT.2*ECT1) GO TO 297
IPD = 1.0
C MEM = 0
C IF(MEM.EQ.1) GO TO 297
C DO 148 I = 1, INPES
C AR(I) = 0.0
C MEM = 1
C AI(I) = 0.0
C BR(I) = 0.0
C  BI(I)=0.0
C  CR(I)=0.0
C148  CI(I)=0.0
C  GO TO 401
     A2(2)=A2(1)
     A3(2)=A3(1)
     A4(2)=A4(1)
     A5(2)=A5(1)
     A6(2)=A6(1)
C 400  CONTINUE
C 401  TAMSE=10.0*LOG10(TAMSE/720.0)
     SN=10.0*LOG10(RRR/WN)
     TOTMS=10.0*LOG10(TOTMS/48000.0)
     WRITE(6,253)TOTMS,SN
C 253  FORMAT(1H 'TOTMS=',F10.5,3X,'SN=',F10.5)
     STOP
     END
###S
###S
PROGRAM TEST6
DIMENSION SSR(100), SSQ(100), RECR(100), RECI(100), SSRE(100)
C, SSREI(100), YR(31), YQ(31), AR(31), AI(31),
CBR(31), BI(31), YRP(31), YQP(31)
DIMENSION CR(31), CI(31), C(64), RR(54), RRJ(54)
DIMENSION PR(54), PI(54)
DIMENSION YB1(32,32), SR(32)
INTEGER A1, A2, B1, B2, A3, B3, F5, F6
INTEGER RH1(40), RH2(40), DFR1(40), DFR2(40), DCOD(16, 2)
C, BIN(4, 4, 4)
DIMENSION IH1(64), IH2(40)
DIMENSION YIRX(31), YITX(31)
DIMENSION YFI(31), YFJ(31), YI(32, 28), YJ(32, 28), FI(32), FJ(32)
REAL IX1(16, 35), IX2(16, 35), IS1(88), IS2(88)
DIMENSION SB1(128), SB2(128), XB1(33), XB2(33)
DIMENSION SPI(30), SPJ(30)
DIMENSION YR1(31), YQ1(31)
DIMENSION DI(50), DJ(50), RI(50), RJ(50)
DIMENSION FTI(32), FTJ(32)
REAL IB1, IB2
COMMON /CM3 /DI, DJ
LC01=4
THEI=0.0
ESIGP1=0.0
ESIGP2=0.0
BCAR=0.06
ECOMP=0.0
C
DATA IH1/0,0,1,1,0,0,1,0,1,0,1,0,1,1,1,
C00,0,1,0,0,0,0,0,1,0,0,0,1,1,0,
C10,0,1,0,1,0,0,0,1,0,0,0,1,1,1,
C10,0,1,1,0,0,0,1,0,1,1,0,1,1,1/;
C
THREE SKYWAVES STANDARD DEVIATION
OPEN(5, FILE='INPUT')
OPEN(2, FILE='OUTPUT')
R1=0.0
LC02=0
R2=0.0
IA1=0
IA2=0
READ(5,700)(AR(I), I=1,31)
READ(5,700)(AI(I), I=1,31)
READ(5,700)(BR(I), I=1,31)
READ(5,700)(BI(I), I=1,31)
READ(5,700)(CR(I), I=1,31)
READ(5,700)(CI(I), I=1,31)
READ(5,700)(YR(I), I=1,31)
READ(5,700)(YQ(I), I=1,31)
READ(5,700)(SSRER(I), I=1,44)
READ(5,700)(SSREI(I), I=1,44)
READ(5,701)ALFPR, ALFPI, BTRP, BTPR
READ(5,701) ALFDR, ALFDI, BTADR, BTADI
READ(5,702) CTDR, CTDI
READ(5,700) (RECR(I), I=1,54)
READ(5,700) (RECI(I), I=1,54)
READ(5,700) (SSR(I), I=1,86)
READ(5,700) (SSQ(I), I=1,86)
700 FORMAT(6F10.5)
701 FORMAT(4F10.5)
702 FORMAT(2F10.5)
READ(5,700) (SB1(I), I=1,128)
DO 505 I=1,128
SB2(I)=0.0
DO 500 I=1,33
JJ=129-I
II=34-I
XB1(II)=SB1(JJ)
DO 500 I=1,33
JJ=129-I
II=34-I
XB2(II)=SB2(JJ)
LCO=1104
EXR=0.0
NN=1
III=17
L=52500
C
L=10500
C
L=46500
DO 9900 I=1,4
II=(I-1)*16
DO 9900 K=1,4
K1=(K-1)*4+II
DO 9900 KK=1,4
BIN(I,K, KK)=IH1(K1+KK)
9900 CONTINUE
DO 9990 J=1,2
DO 9990 I=1,16
DCOD(I,J)=0.0
9990 CONTINUE
DCOD(2,2)=1
DCOD(3,1)=1
DCOD(4,1)=1
DCOD(4,2)=1
DCOD(5,1)=1
DCOD(7,1)=1
DCOD(7,2)=1
DCOD(8,2)=1
DCOD(9,2)=1
DCOD(10,1)=1
DCOD(10,2)=1
DCOD(12,1)=1
DCOD(13,1)=1
DCOD(13,2)=1
DCOD(14,1)=1
DCOD(15,2)=1
A1=0
ISOS=0
A2=0
B1=0
B2=0
A3=0
B3=0
F5=0
F6=0
IER11 = 0
IER21 = 0
IER12 = 0
IER22 = 0
IERD11 = 0
IERD21 = 0
DO 45 J = 1, 43
  IJ = J + 1
  SSRER(J) = SSRER(IJ)
  SSREI(J) = SSREI(IJ)
45 CONTINUE
DO 30 J = 1, 54
  RRI(J) = RECR(J)
  RRJ(J) = RECI(J)
30 CONTINUE
DO 35 J = 1, 86
  JJ = 87 - J
  IS1(JJ) = SSR(J)
  IS2(JJ) = SSQ(J)
35 CONTINUE
DO 20 J = 1, 54
  JJ = 55 - J
  RECR(JJ) = RRI(J)
  RECI(JJ) = RRJ(J)
20 CONTINUE
NK = 0.0
DO 811 I = 1, 31
  NK = NK + 1
  GO TO (4, 4, 5, 4, 4, 5, 4, 4, 4, 4, 5, 5, 4, 4, 4, 5, 5, 4, 4, 5, 5, 4, 5, 5, 4, 5, 5, 4, 5, 5, 4), NK
4 SR(NK) = 1.0
5 SR(NK) = -1.0
811 CONTINUE
DC = ((SQRT(32.0)) - 1)/31.0
DO 11 I = 1, 31
  SR(I) = SR(I) + DC
11 CONTINUE
DO 39 I = 1, 31
  DO 25 J = 1, 31
    N1 = I + J - 1
    IF(N1 > 31) GO TO 23
    N1 = N1 - 31
    YB1(N1, I) = SR(J)
 25 CONTINUE
  SUMB = 0.0
  DO 33 I = 1, 31
    SUMB = SUMB + SR(I) * YB1(I, 1)
33 CONTINUE
OPEN(7, FILE = 'ID26', FORM = 'UNFORMATTED')
DO 400 LLL = 1, L
  IF(LLL .GT. 8000) GO TO 401
  IF(IESOS .EQ. 1) THEN
    WRITE(2, 254) LLL
    254 FORMAT(1H 'RESTART AT LLL=', I6)
    LCO1 = 4
    ENDIF
  C IF(IE .EQ. 1.OR. IE .EQ. 2 .OR. IE .EQ. 100) THEN
    WRITE(2, 257) IE, LCO, LLL
    257 FORMAT(1H 'IE=', I3, 2X, 'LCO=', I6, 1X, 'LLL=', I6)
    ENDIF
    LCO = LCO + 1
    IF(LCO .EQ. 1217) THEN
C LCO= 1
C III=1
ENDIF
C IF(LLL.EQ.1)GO TO 50
READ(7)R1, R2, SA1, SA2
C IA1=SA1
C IA2=SA2
50 LCO1=LCO1+1
C IF(LCO1.EQ.5)THEN
CALL EQLZR(ISOS, LCO, LLL, YR, YQ, YRP, YQP, R1, R2, FI, FJ, XI, XJ, YI
C, YJ, RRI, RRJ, SPI, SPJ, RI, RJ, ECOMP)
DO 340 I=1, 28
IF(I.EQ.1)FTJ(I)=FJ(I)
FTJ(1)=FJ(I)
LC01=1
ELSE
DO 120 I=1, 53
RRI(55-I)=RRI(54-I)
120 RRJ(55-I)=RRJ(54-I)
RRI(1)=R1
RRJ(1)=R2
DO 130 I=1, 50
RI(I)=RRI(I+4)
130 RJ(I)=RRJ(I+4)
CS=COS(ECOMP)
SN=SIN(ECOMP)
RI1=RI(1)*CS+RJ(1)*SN
RJ1=RJ(1)*CS-RI(1)*SN
RI(1)=RI1
RJ(1)=RJ1
RRI(5)=RI(1)
RRJ(5)=RJ(1)
DO 110 J=1, 27
DO 100 I=1, 28
YI(I,J)=YI(I,J+1)
YJ(I,J)=YJ(I,J+1)
100 CONTINUE
110 CONTINUE
DO 115 I=1, 28
YI(I,28)=YR(I)
115 YJ(I,28)=YQ(I)
DO 300 IK=1, 32
FI(IK)=0.0
300 FJ(IK)=0.0
DO 330 I=1, 28
FI(I)=FTI(I)
330 FJ(I)=FTJ(I)
C DO 340 I=1, 28
C J1=28+1-I
C DO 330 J=1, J1
C II1=28-J+1
C IY=J+1-I
C ID=50-I+1
C FI(J)=FI(J)+YI(IY,II1)*DI(ID)-YJ(IY,II1)*DJ(ID)
C FJ(J)=FJ(J)+YI(IY,II1)*DJ(ID)+YJ(IY,II1)*DI(ID)
C330 CONTINUE
C340 CONTINUE
XI=0.0
XJ=0.0
DO 350 I=1, 50
XI = XI + RI(I) * DI(I) - RJ(I) * DJ(I)
XJ = XJ + RI(I) * DJ(I) + RJ(I) * DI(I)
ENDIF
C GO TO 100
C IF (LCO. NE. 1097) GO TO 100
IF (LCO. EQ. 1097. OR. LCO. EQ. 1128. OR. LCO. EQ. 1159) THEN
CALL ESTM(SUMB, YB1, RRI, RRJ, YTRX, YTIX)
ENDIF
CALL DETECT(IE1, FI, FJ, XI, XJ, SA1, SA2, LLL, IE, IX1, IX2, IS1, IS2,
C, XB1, XB2, LCO, IB1, IB2, C, SB1, SB2, III)
IF (LCO. GT. 1088) THEN
IB1 = SB1(III)
IB2 = SB2(III)
III = III + 1
ENDIF
CALL ENCOD(LLL, BIN, DCOD, RH1, IH1, RH2, IH2, DFR1, DFR2,
C, SB1(III), IS2(33))
CALL DECOD(LLL, DCOD, BIN, IX1(1,1), IX2(1,1), A1, A2, B1, B2, A3,
CB3, F5, F6)
IF (LCO. EQ. 1120. AND. LCO. GT. 32) THEN
CALL ERCONT(LLL, A1, A2, B1, B2, A3, B3, RH1, RH2, IH1, IH2, DFR1,
CDFR2, IER11, IER12, IER21, IER22, IERD11, IERD21)
ENDIF
IF (LLL. LT. 500) GO TO 510
C IF (LCO. NE. 1135) GO TO 510
IF (LCO. EQ. 1135. OR. LCO. EQ. 1166. OR. LCO. EQ. 1197) THEN
DO 600 I = 1, 31
YR(I) = YTRX(I)
600 YQ(I) = YTIX(I)
ENDIF
510 IF (LLL. EQ. 1) THEN
DO 131 I = 1, 54
PR(I) = RRI(I)
131 PI(I) = RRJ(I)
ELSE
DO 132 I = 1, 53
PR(55-I) = PR(54-I)
132 PI(55-I) = PI(54-I)
PR(1) = R1
PI(1) = R2
ENDIF
CALL ESTM(ISOS, LCO, AR, AI, BR, BI, CR, CI, YR, YQ, RI, RJ,
C, SSREI, ALFPR, ALFPI, BTAPR, BTAPI, CTPR, CTPI,
CALFDR, ALFD1, BTADR, BTADI, CTDI, IB1, IB2, YRP,
CYQP, R1, R2, LLL, LS3, YR1, YQ1, LC02, EXR, NN, THEI, ESI0P1, ESI0P2,
C, ECOMP, PR, PI, BCAR)
CALL ESI0P(ISOS, LCO, AR, AI, BR, BI, CR, CI, YR, YQ, RI, RJ,
C, SSREI, ALFPR, ALFPI, BTAPR, BTAPI, CTPR, CTPI,
CALFDR, ALFD1, BTADR, BTADI, CTDI, IB1, IB2, YRP,
CYQP, R1, R2, LLL, LS3, YR1, YQ1, LC02, EXR, NN, THEI, ESI0P1, ESI0P2,
C, ECOMP, PR, PI, BCAR)
FORMAT (1H15X, 'OVER ALL AV. BIT ERROR RATE='
C, F10.5, 20X, 'OVERALL DIF. AV BIT ERROR RATE=', F10.5)
FORMAT (1H149)
FORMAT (1H219)
WRITE (2, 2357) ER, IE1
2357 FORMAT (1H 'ERROR=', 'E12.5, 'IE1=', '17)
SUBROUTINE ESTM(SUMB, YB1, RRI, RRJ, YTRX, YTIX)
DIMENSION R1(32), R2(32)
DIMENSION YB1(32,32), YTRX(31), YTIX(31), RRI(54), RRJ(54)
N2=32
DO 712 I=1,31
N2=N2-1
R1(N2)=RRI(I)
R2(N2)=RRJ(I)
DO 708 I=1,31
YTRX(I)=0.0
DO 808 J=1,31
DO 707 I=1,31
YTRX(J)=YTRX(J)+R1(I)*YB1(I,J)
YTIX(J)=YTIX(J)+R2(I)*YB1(I,J)
CONTINUE
DO 909 I=1,31
YTRX(I)=YTRX(I)/SUMB
YTIX(I)=YTIX(I)/SUMB
RETURN
END

SUBROUTINE ENCOD(LLL, BIN, DCOD, RH1, IH1, RH2, IH2, CDFR1, DFR2, FA11, FA22)
INTEGER BIN(4,4,4), DCOD(16,2)
INTEGER RH1(40), RH2(40), DFR1(40), DFR2(40)
DIMENSION IH1(64), IH2(40)
IF (LLL.EQ.1 )THEN
DO 3 I=1,40
RH1(I)=0
RH2(I)=0
IH1(I)=0
IH2(I)=0
DFR1(I)=0
DFR2(I)=0
RH1(33)=0
IH1(33)=1
ENDIF
DO 2 J=1,32
IJ=J+1
RH1(J)=RH1(IJ)
IH1(J)=IH1(IJ)
RH2(J)=RH2(IJ)
IH2(J)=IH2(IJ)
DFR1(J)=DFR1(IJ)
DFR2(J)=DFR2(IJ)
D1=FA11
D2=FA22
K1=(D1+5)/2
K2=(D2+5)/2
RH1(33)=BIN(K2,K1,1)
IH1(33)=BIN(K2,K1,2)
RH2(33)=BIN(K2,K1,3)
IH2(33)=BIN(K2,K1,4)
IADD1=RH1(32)*8+IH1(32)*4+RH1(33)*2+IH1(33)+1
DFR1(33)=DCOD(IADD1,1)
DFR2(33)=DCOD(IADD1,2)
RETURN
SUBROUTINE DECOD(LLL, DCOD, BIN, EF1, EF2, A1, A2, B1, B2, A3, B3, F5, F6)
    INTEGER DCOD(16,2), BIN(4,4,4)
    INTEGER A1, A2, B1, B2, A3, B3, F5, F6
    IF(LLL.LT.33)GO TO 5
    DR=EF1
    DI=EF2
    C DR, DI ARE DETECTED DATA SYMBOLS
    K1=(DR+5)/2
    K2=(DI+5)/2
    A1=BIN(K2,K1,1)
    A2=BIN(K2,K1,2)
    B1=BIN(K2,K1,3)
    B2=BIN(K2,K1,4)
    IADD1=F5*8+F6*4+A1*2+A2+1
    A3=DCOD(IADD1,1)
    B3=DCOD(IADD1,2)
    F5=A1
    F6=A2
    5 RETURN
END

SUBROUTINE ERCONT(LLL, A1, A2, B1, B2, A3, B3, RH1, RH2, IH1, CIH2, DFR1, DFR2, IER11, IER12, IER21, IER22, IERD11, IERD21)
    INTEGER A1, A2, B1, B2, A3, B3
    DIMENSION IH1(64), IH2(40)
    INTEGER RH1(40), RH2(40), DFR1(40), DFR2(40)
    IF(LLL.LT.500)GO TO 6
    IF(A1.NE.RH1(1))IER11=IER11+1
    IF(B1.NE.RH2(1))IER21=IER21+1
    IF(A2.NE.IH1(1))IER12=IER12+1
    IF(B2.NE.IH2(1))IER22=IER22+1
    IF(A3.NE.DFR1(1))IERD11=IERD11+1
    IF(B3.NE.DFR2(1))IERD21=IERD21+1
    6 RETURN
END

SUBROUTINE EST(ISOS, LCO, AR, AI, BR, BI, CR, CI, YR, YQ, RECR, RECI, CSSRER, SSREI, ALFPR, ALFPI, BTAPR, BTAPI, CTPR, CTPI, CALFDR, ALFDI, BTADR, BTADI, CTDI, CTDI, IB1, IB2, C, YRP, YQP, R1, R2, LLL, LS3, YR1, YQ1, LCO2, EXR, NN, THEI, ESIGP1, ESIGP2, C, ECOMP, PR, PI, BCAR)
    DIMENSION RECR(50), RECI(50), SSRER(100)
    C, SSRE1(100), YR(31), YQ(31), AR(31), AI(31), CBR(31), B1(31), F1(31), F1(31), EIR(31), EI(31), EII(31), YRP(31), YQP(31)
    DIMENSION CR(31), CI(31)
    DIMENSION YR1(31), YQ1(31)
    DIMENSION PR(54), PI(54)
    REAL IB1, IB2
    IF(LLL.NE.1)GO TO 520
    LS3=0.0
    520 NUMS=0.0
    INPES=31
    DELT=0.005
    BFI=0.03
    BCAR=BCAR-DELT*(BCAR-BFI)
    ACAR=BCAR/2.0
    THR=1000
    C CEE=0.05
    CEE=0.01
    C IF(LCO.EQ.1098.AND.EXR.GT.0.5)THEN
DO 119 I=1,INPES
YR(I)=0.0
YQ(I)=0.0
ISOS=1
CEE=0.2
ENDIF

IF(LCO.GT.1098)CEE=0.2
ETA=0.0001

THETA=0.94
THETA=0.90
TH0=(1-THETA)**2
TH1=(1-THETA**2)
KL=43
KLP=KL+2
KLM=KL+1

N=6
N=3

RIDR=0.0
RIDI=0.0
DO 302 I=1, KL
SSRER(KLP-I)=SSRER(KLM-I)
SSREI(KLP-I)=SSREI(KLM-I)

SSRER(1)=IB1
SSREI(1)=IB2
XT1=ABS(RECR(50))
XT2=ABS(RECI(50))
LNC=0

IF(XT1.EQ.7.AND.XT2.EQ.7)LNC=1
IF(LCO2.GT.0.0.AND.LCO2.LT.25)LNC=1
IF(LNC.EQ.1)GO TO 7070
IF(LCO.EQ.1135)THEN
RIDL1=0.0
RIDI1=0.0
DO 200 I=1, INPES
RIDL=RIDL1+(SSRER(I)*YR1(I)-SSREI(I)*YQ1(I))
RIDI=RIDI1+(SSRER(I)*YQ1(I)+SSREI(I)*YR1(I))
ER1=RECR(50)-RIDL
EI1=RECI(50)-RIDI1
ENDIF
DO 747 I=1,INPES
RIDL=RIDL+...(SSRER(I)*YR(I)-SSREI(I)*YQ(I))
RIDI=RIDI+...(SSRER(I)*YQ(I)+SSREI(I)*YR(I))

XA=PR(54)**2+PI(54)**2
XB=SQRT(XA)
ECARI=PR(54)*RIDL+PI(54)*RIDI
ECARQ=PI(54)*RIDL-PR(54)*RIDI
ESIG=ECARQ*COS(THEI)-ECARI*SIN(THEI)
ESIG=ESIG/(XA*XB)

IF(ESIG.GT.THRH)ESIG=THRH
IF(ESIG.LT.-THRH)ESIG=-THRH
ESIG1=BCAR*ESIG
ESIG2=ACAR*ESIG1
ESIG2=ESIG2+ESIGP1
ESIG3=ESIG1+ESIGP1
ESIG3=ESIG3+ESIGP2
ESIGP1=ESIG2
ESIGP2=ESIG3
THEI = ESIGP2  
ECOMP = ESIGP2 + (49.0) * ESIGP1  
ERIR = RECR(50) - RIDR  
ERRI = RECI(50) - RIDI  
IF (LCO . GT. 888 . AND. LCO . LE. 1088) THEN  
ER = ABS (ERIR) + ABS (ERRI)  
NN = NN + 1  
ENDIF  
DO 748 I = 1, INPES  
YR(I) = YR(I) + CEE * (ERIR * SSRER(I) + EERRI * SSRERI(I))  
748  
DO 250 I = 1, INPES  
YR1(I) = YR(I)  
250  
IF (LCO . EQ. 1134) THEN  
WRI = ABS (ERI) + ABS (ERRI)  
WRITE (2, 251) WRI, T2  
251  
WRITE (2, 7777) EXR, ESIG  
7777  
WRITE (2, 7222) ESIGP1, THEI, ECOMP  
7222  
NN = 1  
EXR = 0.0  
ENDIF  
ALFR = 0.0  
ALFI = 0.0  
BETR = 0.0  
CTR = 0.0  
CTI = 0.0  
BETI = 0.0  
DO 2202 I = 1, INPES  
ALFR = ALFR + AR(I) * YR(I) + YQ(I) * AI(I)  
ALFI = ALFI + AR(I) * YQ(I) - YR(I) * AI(I)  
CTR = CTR + CR(I) * YR(I) + YQ(I) * CI(I)  
CTI = CTI + CR(I) * YQ(I) - YR(I) * CI(I)  
BETR = BETR + BR(I) * YR(I) + YQ(I) * BI(I)  
BETI = BETI + BR(I) * YQ(I) - YR(I) * BI(I)  
2202  
IF (LLL . LT. 500) GO TO 2500  
IF (LCO . NE. 1135) GO TO 2500  
IF (ISOS . EQ. 1) THEN  
C  
ECOMP = 0.0  
C  
THEI = 0.0  
C  
ESIGP1 = 0.0  
C  
ESIGP2 = 0.0  
BCAR = 0.06  
ALFPR = ALFR  
C  
ISOS = 0  
ALFPI = ALFI  
BTAPR = BETR  
BTAPI = BETI  
CTPR = CTR  
CTPI = CTI
ALFDR=0.0
ALFDI=0.0
BTADR=0.0
BTADI=0.0
CTDR=0.0
CTDI=0.0
ENDIF

2500 DO 2251 1=1, INPES
  FIR(I)=ALFR*AR(I)-ALFI*AI(I)+BETR*BR(I)-BETI*BI(I)+
  CCTR*CR(I)-CTI*CI(I)
  FII(I)=ALFR*AI(I)+ALFI*AR(I)+BETR*BR(I)+BETI*BR(I)+
  CCTR*CI(I)+CTI*CR(I)
  EIR(I)=YR(I)-FIR(I)
  EII(I)=YQ(I)-FII(I)
  LS3=LS3+1
  IMA=LS3
  IF(LS3.LT.700)GO TO 2301
  ETA=0.0001
  2301 IF(LS3.LT.16000)GO TO 2401
  ETA=0.0001
  2401 IF(LS3.LT.32000)GO TO 1985
  ETA=0.0001
  1988 DO 4021 I=1, INPES
    BR(I)=BR(I)+ETA*(BETR*EIR(I)+BETI*EII(I))
    AR(I)=AR(I)+ETA*(ALFR*EIR(I)+ALFI*EII(I))
    CR(I)=CR(I)+ETA*(CTR*EIR(I)+CTI*EII(I))
    CI(I)=CI(I)+ETA*(C'TR*EII(I)-CTI*EIR(I))
    AI(I)=AI(I)+ETA*(ALFR*EII(I)-ALFI*EIR(I))
    BI(I)=BI(I)+ETA*(BETR*EII(I)-BETI*EIR(I))
  4021 DO 4031 1=1, INPES
    BMOD=BMOD+AR(I)**2+AI(I)**2
    BMOD=1.0/SQRT(BMOD)
    DO 4041 I=1, INPES
      AR(I)=AR(I)*BMOD
      4041 ALI(I)=AI(I)*BMOD
      BAR=0.0
      BAI=0.0
      DO 12 I=1, INPES
        BAR=BAR+BR(I)*AR(I)+BI(I)*AI(I)
      12 BAI=BAI+BI(I)*AR(I)-BR(I)*AI(I)
      DO 36 I=1, INPES
        BR(I)=BR(I)-BAR*AR(I)+BAI*AI(I)
      36 BI(I)=BI(I)-BAI*AR(I)-BAR*AI(I)
      BMOD=0.0
      DO 37 I=1, INPES
        BMOD=BMOD+(BR(I)**2+BI(I)**2)
        BMOD=1.0/SQRT(BMOD)
      37 BMOD=BMOD+(BR(I)**2+BI(I)**2)
      DO 38 I=1, INPES
        BR(I)=BR(I)*BMOD
      38 BI(I)=BI(I)*BMOD
      CCR=0.0
      CCI=0.0
      CFR=0.0
      CFI=0.0
      DO 8034 I=1, INPES
        CCR=CCR+AR(I)*CR(I)+CI(I)*AI(I)
        CCI=CCI+AR(I)*CI(I)-CR(I)*AI(I)
CFR = CFR + BR(I) * CR(I) + CI(I) * BI(I)
CFI = CFI + BR(I) * CI(I) - CR(I) * BI(I)

DO 8035 I = 1, INPES
FIR(I) = CCR * AR(I) - CCI * AI(I) + CFR * BR(I) - CFI * BI(I)
FII(I) = CCR * AI(I) + CCI * AR(I) + CFR * BI(I) + CFI * BR(I)
CR(I) = CR(I) - FIR(I)
8035 CI(I) = CI(I) - FII(I)

BMOD = 0.0
DO 8036 I = 1, INPES
BMOD = BMOD + CR(I)**2 + CI(I)**2
BMOD = 1.0 / SQRT(BMOD)
DO 8037 I = 1, INPES
CR(I) = CR(I) * BMOD
8037 CI(I) = CI(I) * BMOD

EPALR = ALFR - ALFPR
EPALI = ALFI - ALFPI
EPBTR = BTTR - BTAPR
EPBTI = BETI - BTAPI
EPCTR = CTR - CTPI
EPCTI = CTI - CTPI
ALFDR = ALFDR + (THO) * EPALR
ALFDI = ALFDI + (THO) * EPALI
ALFPR = ALFPR + ALFDR + TH1 * EPALR
ALFPI = ALFPI + ALFDI + TH1 * EPALI
BTADR = BTADR + (THO) * EPBTR
BTADI = BTADI + (THO) * EPBTI
BTAPR = BTAPR + BTADR + TH1 * EPBTR
BTAPI = BTAPI + BTADI + TH1 * EPBTI
CTDR = CTDR + (THO) * EPCTR
CTDI = CTDI + (THO) * EPCTI
CTPR = CTPR + CTDR + TH1 * EPCTR
CTPI = CTPI + CTDI + TH1 * EPCTI

IF (LNC.EQ.1) THEN
ALFPR = ALFPR + ALFDR
ALFPI = ALFPI + ALFDI
BTAPR = BTAPR + BTADR
BTAPI = BTAPI + BTADI
CTPR = CTPR + CTDR
CTPI = CTPI + CTDI
LCO2 = LCO2 + 1
ENDIF
DO 750 I = 1, INPES
YR(I) = (ALFPR*AR(I) - ALFPI*AI(I) + BTAPR*BR(I) - BTAPI*BI(I) + CCTPR*CR(I) - CTPI*CI(I))
750 YQ(I) = ALFPR*AI(I) + ALFPI*AR(I) + BTAPR*BI(I) + BTAPI*BR(I) + CCTPR*CI(I) + CTPI*CR(I)
ALFNR = ALFPR + (N-1) * ALFDR
ALFNI = ALFPI + (N-1) * ALFDI
BTANR = BTAPR + (N-1) * BTADR
BTANI = BTAPI + (N-1) * BTADI
CTNR = CTPR + (N-1) * CTDR
CTNI = CTPI + (N-1) * CTDI
DO 134 I = 1, INPES
YRP(I) = ALFNR*AR(I) - ALFNI*AI(I) + BTANR*BR(I) - BTANI*BI(I) + CCTNR*CR(I) - CTNI*CI(I)
134 YQP(I) = ALFNR*AI(I) + ALFNI*AR(I) + BTANR*BI(I) + BTANI*BR(I) + CCTNR*CI(I) + CTNI*CR(I)
RETURN
END
SUBROUTINE DETECT(IE1, Y1, Y2, R1, R2, SA1, SA2, LL, IE, IX1, IX2, IS1, IS2
C, XB1, XB2, LCO, IB1, IB2, C, SB1, SB2, III)
DIMENSION C(64), Y1(30), Y2(30)
DIMENSION SB1(128), SB2(128)
REAL CCC(64), IXX1(35)
REAL IX1(16,35), IX2(16,35), IS1(88), IS2(88)
DIMENSION F1(64), F2(64), INN(16), B1(64), B2(64), C1(64)
REAL IXXX1(64,35), IXXX2(64,35)
DIMENSION ISS1(30), ISS2(30)
DIMENSION XB1(33), XB2(33)
DIMENSION Z1(64), Z2(64)
REAL DF1(4), DF2(4)
C, IXXX2(35), DK1(5), DK2(5), IXT1(30), IXT2(30), IXTT1(30), IXTT2(30)
REAL IB1, IB2
IMPES=33
JSHT=IMPES-1
N=33
NI=N-1
C
NN=N-1
IF(LL.EQ.1)GO TO 50
DO 45 J=1,85
   IJ=J+1
   IS1(J)=IS1(IJ)
45   IS2(J)=IS2(IJ)
   I31(86)=SA1
   I32(86)=SA2
50   M=8
   M2=M/2
   MM=2*M
   C(1)=0.0
   POS=-1.0
   IF(LL.NE.1)GO TO 5142
   IE=0.0
   IE1=0.0
   DO 1 J=2, M
      C(J)=1000.0
   DO 2 I=1, M
   DO 2 J=1, IMPES-1
         Ixx1(I,J)=IS1(J)
   2     IXX2(I,J)=IS2(J)
   DO 505 I=1, 16
         ISS1(I)=1
   505    ISS2(I)=1
   JSHT=IMPES-1
5142   IF(LL.EQ.1)GO TO 422
   DO 6 I=1, M
   DO 6 J=1, JSHT
         IJ=J+1
         Ixx1(I,J)=IX1(I,IJ)
   6     IXX2(I,J)=IX2(I,IJ)
422    SR=Y1(1)
   IF(LCO.GT.1088)THEN
         DO 7770 I=1, M
            IX1(I,N)=SB1(III)
            IX2(I,N)=SB2(III)
      7770        CONTINUE
   ENDIF
   IF(LCO.GT.1088)GO TO 4010
   SI=Y2(1)
\[ S_1 = (Y_1(1)^2 + Y_2(1)^2) \]
\[ \text{DO 2050 I=1,28} \]
\[ \text{XRE} = (SR\cdot Y_1(I) + SI\cdot Y_2(I)) / S_1 \]
\[ \text{XIM} = (SR\cdot Y_2(I) - SI\cdot Y_1(I)) / S_1 \]
\[ Y_1(I) = \text{XRE} \]
\[ 2050 \quad Y_2(I) = \text{XIM} \]

\[ \text{CALCULATE THE INTERSYMBOL INTERFERENCE} \]

\[ \text{IF (LCO.EQ.1) THEN} \]
\[ \text{DO 707 I=1,M} \]
\[ \text{DO 707 J=1,32} \]
\[ IJ = J+1 \]
\[ IX1(I,J) = XB1(IJ) \]
\[ IX2(I,J) = XB2(IJ) \]
\[ 707 \quad \text{CONTINUE} \]
\[ \text{DO 909 I=1,M} \]
\[ C(I) = 1000 \]
\[ C(1) = 0.0 \]
\[ \text{ENDIF} \]
\[ \text{DO 4000 I=1,M} \]
\[ ZZ1 = 0.0 \]
\[ ZZ2 = 0.0 \]
\[ \text{DO 3500 J=2,28} \]
\[ IJ = 34-J \]
\[ ZZ1 = ZZ1 + IX1(I, IJ) \cdot Y_1(J) - IX2(I, IJ) \cdot Y_2(J) \]
\[ ZZ2 = ZZ2 + IX1(I, IJ) \cdot Y_2(J) + IX2(I, IJ) \cdot Y_1(J) \]
\[ 3500 \quad \text{CONTINUE} \]
\[ Z1(I) = ZZ1 \]
\[ Z2(I) = ZZ2 \]
\[ 4000 \quad \text{CONTINUE} \]
\[ \text{IF (LCO.EQ.60) THEN} \]
\[ \text{WRITE (2,253) IE,LL} \]
\[ 253 \quad \text{FORMAT(1H,'IE=',I6,2X,'LLL=',I6)} \]
\[ \text{ENDIF} \]
\[ R11 = (SR\cdot R1 + SI\cdot R2) / S_1 \]
\[ R22 = (SR\cdot R2 - SI\cdot R1) / S_1 \]
\[ R1 = R11 \]
\[ R2 = R22 \]
\[ \text{DO 250 I=1,M} \]
\[ E1 = R1 - Z1(I) \]
\[ E2 = R2 - Z2(I) \]
\[ F1(I) = E1 \]
\[ F2(I) = E2 \]

\[ \text{COMPARE THE VALUE OF E TO A DECISION THRESHOLD} \]
\[ \text{AND OBTAIN THE BEST VALUE OF X} \]

\[ \text{IF (E1) 120,120,135} \]
\[ 120 \quad \text{IF (E1+2.0) 125,125,130} \]
\[ 125 \quad \text{IX1(I,N)} = -3 \]
\[ \text{GO TO 150} \]
\[ 130 \quad \text{IX1(I,N)} = -1 \]
\[ \text{GO TO 150} \]
\[ 135 \quad \text{IF (E1-2.0) 140,140,145} \]
\[ 140 \quad \text{IX1(I,N)} = 1 \]
\[ \text{GO TO 150} \]
\[ 145 \quad \text{IX1(I,N)} = 3 \]
\[ 150 \quad \text{CONTINUE} \]
C IF(E2) 155, 155, 170
155 IF(E2+2.0) 160, 160, 165
160 IX2(I,N)= -3
GO TO 185
165 IX2(I,N)= -1
GO TO 185
170 IF(E2-2.0) 175, 175, 180
175 IX2(I,N)= 1
GO TO 185
180 IX2(I,N)= 3
185 CONTINUE
C CALCULATE D
D1=E1-IX1(I,N)
D2=E2-IX2(I,N)
C1(I)=C(I)+D1*D1+D2*D2
B1(I)=D1
B2(I)=D2
250 CONTINUE
C FIND THE BEST VECTOR FROM THE FIRST K VECTORS
CC=10000.0
DO 915 I=1,M
IF(C1(I)-CC) 255,915,915
255 CC=C1(I)
II=I
915 CONTINUE
CCC(I)=CC
C1(II)=10000.0
INN(II)=II
DO 920 I=1,N
IXX1(I,I)=IX1(II,I)
IXX2(I,I)=IX2(II,I)
920 CONTINUE
C TAKE THE FIRST COMPONENT OF THE BEST VECTOR TO BE THE
C DETECTED VALUE
C IF(LCO.EQ.1088)THEN
DO 351 I=2,33
145=IS1(I)-IX1(II,I)
146=IS2(I)-IX2(II,I)
IF(145) 311,312,311
312 IF(146) 311,351,311
311 IE=IE+1
IE1=IE+1
351 CONTINUE
ENDIF
145=IS1(I)-IX1(II,1)
146=IS2(I)-IX2(II,1)
C IF(LL.LT.500)GO TO 50212
IF(LCO.GT.1088.AND.LCO.LE.1216)GO TO 50212
IF(145) 350,345,350
345 IF(146) 350,401,350
350 IE=IE+1
401 CONTINUE
DISCARD SOME VECTORS

50212 IB1 = IX1(I1, N)
IB2 = IX2(I1, N)

C50212 IB1 = IS1(33)
IB2 = IS2(33)

C IF (LL.LT.5000) THEN
IB1 = IS1(33)
IB2 = IS2(33)
ELSE
IB1 = IX1(I1, N)
IB2 = IX2(I1, N)
ENDIF

DO 295 I = 1, M
149 = IX1(1, NN) - IXXX1(1, NN)
150 = IX2(1, NN) - IXXX2(1, NN)
IF (149) 925, 285, 925
IF (150) 925, 295, 925
925 C1(I) = 1000.0
295 CONTINUE

SELECT 1/2M-1 MORE VECTORS WITH LOW COSTS

IK = 3
DO 330 I = 2, M2
CC = 10000.0
DO 305 J = 1, M
IF (C1(J) - CC) 302, 305, 305
302 CC = C1(J)
JJ = J
305 CONTINUE
CCC(IK) = CC
C1(JJ) = 100000.0
INN(I) = JJ
IK = IK + 2
330 CONTINUE

EXPAND M/2 SELECTED VECTORS TO GIVE THE VECTORS WITH SECOND LOWEST COST

IM = 2
DO 700 I = 1, M2
J = INN(I)
RD1 = SIGN(1.1, B1(J))
RD2 = SIGN(1.1, B2(J))
ID1 = INT(RD1)
ID2 = INT(RD2)
IF (ABS(B2(J)) - ABS(B1(J))) 695, 695, 205
695 IXX1(I) = IX1(J, N) + ID1 + ID1
IXX2(I) = IX2(J, N)
IF (ABS(IXX1(I)) - 4) 6200, 195, 195
195 IXX2(I) = IX2(J, N) + ID2 + ID2
IXX1(I) = IX1(J, N)
IF (ABS(IXX2(I)) - 4) 6200, 600, 600
600 IXX2(I) = IX2(J, N) - ID2 - ID2
GO TO 6200
205 IXX2(I) = IX2(J, N) + ID2 + ID2
IXX1(I) = IX1(J, N)
IF (ABS(IXX2(I)) - 4) 6200, 6100, 6100

GO TO 6200
**6100**

\[ IXX1(I) = IXX1(J, N) + ID1 + ID1 \]

\[ IXX2(I) = IXX2(J, N) \]

\[ \text{IF (ABS(IXX1(I)) - 4)}6200, 215, 211 \]

\[ IXX1(I) = IXX1(J, N) - ID1 - ID1 \]

**6200 CONTINUE**

**C**

**CALCULATE THE COST**

\[ DD1 = F1(J) - IXX1(I) \]

\[ DD2 = F2(J) - IXX2(I) \]

\[ CCC(IM) = C(J) + DD1 * DD1 + DD2 * DD2 \]

\[ IM = IM + 2 \]

**700 CONTINUE**

**C**

**SUBTRACT THE SMALLEST COST FROM ALL THE COSTS**

**C**

**DO 910 I = 2, M**

\[ CCC(I) = CCC(I) - CCC(1) \]

**910**

\[ C(1) = CCC(1) \]

\[ C(1) = 0.0 \]

**C**

**TRANSFER THE M/2 SELECTED VECTORS (EACH EXPANDED 2 WAYS) TO AN ARRAY IXXX TO GIVE A TOTAL OF M VECTORS**

**C**

**JJ = 1**

**DO 800 I = 1, M, 2**

\[ IL = INN(JJ) \]

**DO 7750 J = 1, N**

\[ IXXX1(I, J) = IXXX(I, IL, J) \]

\[ IXXX2(I, J) = IXXX2(I, IL, J) \]

**7750 CONTINUE**

\[ JJ = JJ + 1 \]

**800 CONTINUE**

**JJ = 1**

**DO 850 I = 2, M, 2**

\[ II = INN(JM) \]

**DO 8250 J = 1, N**

\[ IXXX1(I, J) = IXXX1(II, J) \]

\[ IXXX2(I, J) = IXXX2(II, J) \]

**8250 CONTINUE**

\[ JM = JM + 1 \]

**850 CONTINUE**

**DO 8500 I = 1, M**

\[ DO 360 J = 1, N \]

\[ SIX = IXXX1(I, J) \]

\[ IXX1(I, J) = SIX \]

\[ SIX = IXXX2(I, J) \]

\[ IXX2(I, J) = SIX \]

**360 CONTINUE**

**8500 CONTINUE**

**4010 RETURN**

**H3005**

**SUBROUTINE EQLZR(ISOS, LCO, III, Y1I, Y1J, YFI, YFJ, R1, R2, FI, FJ, XI, XJ**

**C, Y1, YJ, RRI, RRIJ, SP1, SP1J, R1, R2, ECOMP)**

**C**

**THIS SUBROUTINE IS WRITTEN TO SIMULATE THE ADAPTIVE LINEAR FILTER**

**C**

**OF THE HF MODEM. AT TIME T = (I + 49) T, IT REQUIRE THE S.I.R. OF**

**C**

**CHANNEL FROM T = (I - 77) T TO T = (I - 50) T AND ITS PREDICTED VALUE**

**C**

**AT T = (I + 5) T. FURTHERMORE, IT REQUIRE THE 50 MOST RECENT RECEIVED**
SAMPLES, {RI}.

IT GIVES THE S.I.R. OF CHANNEL + FILTER WHICH IS REQUIRED BY DETECTOR
AND THE SAMPLE AT OUTPUT OF FILTER.

THE NUMBER OF TAPS IS N+1 = 50

YI = YI,0 YI,1 YI,2 ... YI,G (G+1 = 28)

III = FLAG TO INDICATE IF IT IS THE FIRST RUN.

YI & YJ = YI+1,1 (MUST BE PROVIDED AFTER 1ST RUN)

YFI & YFJ = YI+5,1 (MUST BE PROVIDED AFTER 1ST RUN)

R1 & R2 = RECEIVED SAMPLE (MUST BE PROVIDED AFTER 1ST RUN)

FI & FJ = FI AS REQUIRED BY DETECTOR

XI & XJ = OUTPUT SAMPLE FROM FILTER

YI & YJ = CONTAINS YI-77,1-1 TO YI-1 AND YI+5,1 INPUT FROM FILE

RRI & RRJ = THE 54 {RI} FROM (1+53) TO IT) INPUT FROM FILE

(I, J CORRESPOND TO THE REAL AND IMAGINARY PARTS RESPECTIVELY)

DIMENSION HIJ (22,40), SPI(30), SPJ(30)
DIMENSION YI(32,28), YJ(32,28), RI(50), RJ(50)
DIMENSION YFI(31), YFJ(31), YFI(31), YFJ(31), RRI(54), RRJ(54)
DIMENSION FI(32), FJ(32)

NH=28
NY=28
NX=77
NTAP=50
NH1=NH-1

IF (III .EQ. 1) THEN

DO 40 J=1, NY, 1
READ (5,2333)(YI(I,J),I=1,31,1)
READ (5,2333)(YJ(I,J),I=1,31,1)

2333 FORMAT (6F10.5)

40 CONTINUE

DO 60 I=29, 32, 1
DO 50 J=1, 28, 1
YI(I,J)=0.0EO
YJ(I,J)=0.0EO

50 CONTINUE

60 CONTINUE

READ(5,2333)(YFI(I),I=1,31)
READ(5,2333)(YFJ(I),I=1,31)

ELSE

DO 110 J=1, NY-1, 1
DO 100 I=1, NH, 1
YI(I,J)=YI(I,J+1)
YJ(I,J)=YJ(I,J+1)

100 CONTINUE

110 CONTINUE

DO 115 I=1, NH, 1
YI(I,NH)=YI(I)
YJ(I,NH)=YJ(I)

115 CONTINUE

DO 120 I=1, 53, 1
RRI(55-I)=RRI(54-I)
RRJ(55-I)=RRJ(54-I)

120 CONTINUE

RRI(1)=R1
RRJ(1)=R2

ENDIF

DO 130 I=1, NTAP, 1
R1(I)=RRI(I+4)
RJ(I)=RRJ(I+4)

130 CONTINUE
CS = COS(ECOMP)
SN = SIN(ECOMP)
RI1 = RI (1) * CS + RJ (1) * SN
RJ1 = RJ (1) * CS - RI (1) * SN
RI (1) = RI1
RJ (1) = RJ1
RRI (5) = RI (1)
RRJ (5) = RJ (1)
DO 140 I = 1, 40, 1
HIJ (1, I) = 0.0E0
HIJ (2, I) = 0.0E0
140 CONTINUE
DO 150 I = 1, NH, 1
HIJ (1, I) = YFI (I)
HIJ (2, I) = YFJ (I)
150 CONTINUE
CALL RFAI (ISOS, LCO, III, HIJ, NH, SPI, SPJ)
DO 160 I = 1, NH
FI (I) = HIJ (1, I)
FJ (I) = HIJ (2, I)
160 CONTINUE
CALL FIL! (YI, YJ, Ni, RI, RJ, NX, NTAP, XI, XJ)
RETURN
END

SUBROUTINE RFAI (ISOS, LCO, III, HIJ, NH, SPI, SPJ)
THIS IS THE ROOT-FINDING ALGORITHM H1104
DIMENSION HIJ (2, 40), ALPRI (40), ALPRJ (40), USTI (40), USTJ (40)
DIMENSION REZ (40)
DIMENSION ALPYI (20), ALPYJ (20)
DIMENSION AII (10), AJJ (10), SPI (30), SPJ (30)
REAL IMZ (40)
COMMON /CM1/ NRZ
COMMON /CM2/ ALPYI, ALPYJ
DATA AII / 0.0E0, 0.0E0, 0.0E0, 0.0E0, 0.0E0, 0.0E0, 0.0E0, 0.0E0, 0.0E0, 0.0E0, 0.0E0,
1 0.642824346E0, 0.642824346E0, -0.642824346E0, -0.642824346E0, -0.642824346E0, 0.0E0/
DATA AJJ / 0.0E0, 0.0E0, 0.0E0, 0.0E0, 0.0E0, 0.0E0, 0.0E0, 0.0E0, 0.0E0, 0.0E0,
1 -0.642824346E0, 0.642824346E0, 0.642824346E0, 0.642824346E0, 0.642824346E0, 0.0E0/
C
NH1 = NH - 1
NIT = 100
AC = 1.0E0
THD = 1.0E-10
AD = 1.05E0
ALTHD = 1.0E0 / AD
IF (III .EQ. 1.0 OR. ISOS .EQ. 1) THEN
C
IF (III .EQ. 1.0) THEN
DO 12 I = 1, 9, 1
SPI (I) = AII (I)
SPJ (I) = AJJ (I)
12 CONTINUE
ENDIF
DO 50 I = 1, 40, 1
ALPRI (I) = 0.0E0
ALPRJ (I) = 0.0E0
50 CONTINUE
IRR = 0
IREF = 1

DO 130 I = 1, 40, 1
USTI(I) = 0.0E0
USTJ(I) = 0.0E0
CONTINUE
ALPHAI = 0.0E0
ALPHAJ = 0.0E0
ALPI = 0.0E0
ALPJ = 0.0E0
ALPM = 0.0E0

IF (III.EQ.1 .OR. ISOS.EQ.1) IREF = 1
ISOS = 0.0
NSTOP = 0
CALL NEWST(SPI, SPJ, ALPHAI, ALPHAJ, IREF)
STR = ALPHAI
STJ = ALPHAJ

DO 400 I = 1, NIT, 1
CALL EQR (1, NH, HIJ, ALPHAI, ALPHAJ, USTI, USTJ)
ALPM = SQRT(ALPHAI**2 + ALPHAJ**2)
CALL ALG (ALPHAI, ALPHAJ, DELTAI, DELTAJ, USTI, USTJ, NH)
DEL = DELTAI**2 + DELTAJ**2

IF (ALPM < ALTHD) THEN
  IF (DEL >= THD) THEN
    ALPHAI = ALPHAI + AC*DELTAI
    ALPHAJ = ALPHAJ + AC*DELTAJ
    ALPM = SQRT(ALPHAI**2 + ALPHAJ**2)
    IF (ALPM < ALTHD) THEN
      IF (STR .NE. AI(9) .OR. STJ .NE. AJJ(9)) THEN
        IREF = IREF + 1
        CALL NEWST(SPI, SPJ, ALPHAI, ALPHAJ, IREF)
        STR = ALPHAI
        STJ = ALPHAJ
      ELSE
        NSTOP = 1
        GO TO 500
      ENDIF
    ENDIF
  ELSE
    NSTOP = 1
    GO TO 500
  ENDIF
ELSE
  THRESHOLD NOT YET REACHED.
  ALPHAI = ALPHAI + AC*DELTAI
  ALPHAJ = ALPHAJ + AC*DELTAJ
  ALPM = SQRT(ALPHAI**2 + ALPHAJ**2)
  IF (ALPM > ALTHD) THEN
    IF (STR .NE. AI(9) .OR. STJ .NE. AJJ(9)) THEN
      IREF = IREF + 1
      CALL NEWST(SPI, SPJ, ALPHAI, ALPHAJ, IREF)
      STR = ALPHAI
      STJ = ALPHAJ
    ELSE
      NSTOP = 1
      GO TO 500
    ENDIF
  ELSE
    THRESHOLD REACHED.
    IREF = IREF + 1
    IF (STR .EQ. AI(9) .AND. STJ .EQ. AJJ(9)) NSTOP = 1
    GO TO 500
  ENDIF
ELSE
  C ALPHA IS GREATER THAN 1.0
  IF (STR .NE. AI(9) .OR. STJ .NE. AJJ(9)) THEN
    IREF = IREF + 1
    CALL NEWST(SPI, SPJ, ALPHAI, ALPHAJ, IREF)
    STR = ALPHAI
    STJ = ALPHAJ
  ELSE
    NSTOP = 1
    GO TO 500
  ENDIF
ENDIF
ENDIF
CALL EQR1(NH,ALPHAI,ALPHA J,USTI,USTJ)
DO 580 J=1,NH,1
J1=J+1
USTI(J)=USTI(J1)
USTJ(J)=USTJ(J1)
580 CONTINUE
USTI(NH+1)=0.0E0
USTJ(NH+1)=0.0E0

IRR=IRR+1
ALPRI(IRR)=ALPHAI
ALPRJ(IRR)=ALPHA J
DO 600 J=1,40,1
HIJ(1,J)=0.0E0
HIJ(2,J)=0.0E0
600 CONTINUE
SET THE REDUCED CHANNEL S.I.R. TO THE Q-SEQUENCE.
DO 610 J=1,NH,1
HIJ(1,J)=USTI(J)
HIJ(2,J)=USTJ(J)
610 CONTINUE
GO TO 700
DO 21 I=1,30,1
SPI(I)=0.0E0
SPJ(I)=0.0E0
21 CONTINUE
DO 30 I=1,IRR,1
SPI(I)=ALPRI(I)
SPJ(I)=ALPRJ(I)
30 CONTINUE
DO 35 I=1,9,1
SPI(IRR+I)=AI I(I)
SPJ(IRR+I)=AJ J(I)
35 CONTINUE
NRZ=IRR
DO 811 I=1,20,1
ALPYI(I)=0.0E0
ALPYJ(I)=0.0E0
811 CONTINUE
DO 812 I=1,IRR,1
ALPYI(I)=ALPRI(I)
ALPYJ(I)=ALPRJ(I)
812 CONTINUE
RETURN
END

SUBROUTINE FIL1(YI,YJ,NH,RI,RJ,IST,NTAP,XI,XJ)
THIS SUBROUTINE IS WRITTEN TO WORK OUT THE LINEAR FILTER
NTAP = THE LINEAR OF LINEAR FILTER REQUIRED (MAX=50).
DIMENSION YYIJ(2,51)
DIMENSION QI(51), QJ(51), USTI(51), USTJ(51)
DIMENSION ALZI(20), ALZJ(20)
DIMENSION FI(32), FJ(32), DI(50), DJ(50)
DIMENSION DI(50), DJ(50)
DIMENSION YI(32,28), YJ(32,28), RI(50), RJ(50)
COMMON /CM1/NRZ
COMMON /CM2/ALZI, ALZJ
COMMON /CM3/DI, DJ

IOU=0
NH1 = NH-1
NTAP1=NTAP-1
NNPP=NH+NTAP-1
DO 80 I=1,51,1
YYIJ(1,I)=0.0E0
YYIJ(2,I)=0.0E0
80 CONTINUE
IP=NTAP
YYIJ(1,IP)=1.0E0
IRR=NRZ
DO 200 I=1,IRR,1
ALPHAI=ALZI(I)
ALPHAJ=ALZJ(I)

C****** EQUALISER INPUT
CCALPI= ALPHAI
CCALPJ=-ALPHAJ

C WORK OUT Q.
QI(1)=YYIJ(1,1)
QJ(1)=YYIJ(2,1)
DO 90 J=1,50,1
J1=J+1
QI(J1)=YYIJ(1,J1)+(YYIJ(1,J)*CCALPI-YYIJ(2,J)*CCALPJ)
QJ(J1)=YYIJ(2,J1)+(YYIJ(1,J)*CCALPJ+YYIJ(2,J)*CCALPI)
90 CONTINUE

C WORK OUT P
PI=0.0
PJ=0.0
DO 100 J=1,51,1
J1=52-J
PPI=QI(J1)-(ALPHAI*PI-ALPHAJ*PJ)
PPJ=QJ(J1)-(ALPHAI*PJ+ALPHAJ*PI)
PI=PPI
PJ=PPJ
USTI(J1)=PPI
USTJ(J1)=PPJ
100 CONTINUE

C****** EQUALISER OUTPUT
DO 130 J=1,NTAP,1
J1=J+1
YYIJ(1,J)=USTI(J1)
YYIJ(2,J)=USTJ(J1)
130 CONTINUE
DO 150 J=1,51,1
QI(J)=0.0E0
QJ(J)=0.0E0
150 CONTINUE
DO 200 J=1,NTAP,1
DI(J) = YYIJ(1,J)
DJ(J) = YYIJ(2,J)

210 CONTINUE
C DO 300 IK = 1, 32
C FI(IK) = 0.0E0
C FJ(IK) = 0.0E0
C300 CONTINUE
C IST1 = IST - NTAP1
C DO 340 I = 1, NH, 1
C J1 = NH + 1 - 1
C DO 330 J = 1, J1, 1
C II1 = IST1 - J + 1
C IY = J1 + I - 1
C ID = NTAP1 - 1
C FI(J) = FI(J) + YI(IY, II1) * DI(ID) - YJ(IY, II1) * DJ(ID)
C FJ(J) = FJ(J) + YI(IY, II1) * DJ(ID) + YJ(IY, II1) * DI(ID)
C330 CONTINUE
C340 CONTINUE
XI = 0.0E0
XJ = 0.0E0
DO 350 I = 1, NTAP, 1
XI = XI + RI(I) * DI(I) - RJ(I) * DJ(I)
XJ = XJ + RI(I) * DJ(I) + RJ(I) * DI(I)
350 CONTINUE
RETURN
END

SUBROUTINE NEWST(AII, AJJ, ALPHAI, ALPHAJ, IREF)
C THIS SUBROUTINE ASSIGN Alpha TO THE NEXT POSSIBLE
C STARTING POINTS.
DIMENSION AII(30), AJJ(30)
ALPHAI = AI(1)
ALPHAJ = AJ(1)
RETURN
END

SUBROUTINE EQR(IEQR, NH, HIJ, ALPHAI, ALPHAJ, USTI, USTJ)
C THIS SUBROUTINE IS USED TO PASS A SEQUENCE THROUGH THE
C COMBINATION OF EQUALISER SECTION 1 (EQRII) AND EQUALISER
C SECTION 2 (EQRII). THE CONFIGURATION IS SET UP FOR THE ROOT
C -1/(ALPHA) WHICH LIES OUTSIDE THE UNIT CIRCLE.
C IEQR = 1 ONLY EQRII IS USED.
C = 2 BOTH EQRI+EQRII ARE USED.
C NH = NO. OF COMPONENTS IN S.I.R. OF CHANNEL.
C HIJ = 2-D ARRAY CONTAINING THE S.I.R. OF CHANNEL.
C ALPHAI = REAL PART OF ALPHA.
C ALPHAJ = IMAG. PART OF ALPHA.
C USTI = 1-D ARRAY HOLDING REAL PARTS OF OUTPUT AT EQRII.
DIMENSION HIJ(2,40), USTI(40), USTJ(40)
DIMENSION QI(40), QJ(40)
PI = 0.0E0
PJ = 0.0E0
PPI = 0.0E0
PPJ = 0.0E0
DO 5 I = 1, 40, 1
QI(I) = 0.0E0
QJ(I) = 0.0E0
CONTINUE
IF (IEQR .EQ. 2) GOTO 10
CCALPI = 0.0E0
CCALPJ = 0.0E0
GOTO 20
CCALPI = ALPHAI
CCALPJ = ALPHAJ
WORK OUR Q.
QI(1) = HIJ(1, 1)
QJ(1) = HIJ(2, 1)
DO 30 J = 1, NH, 1
J1 = J + 1
QI(J1) = HIJ(1, J1) + (HIJ(1, J) * CCALPI - HIJ(2, J) * CCALPJ)
QJ(J1) = HIJ(2, J1) + (HIJ(1, J) * CCALPJ + HIJ(2, J) * CCALPI)
CONTINUE
WORK OUT P
PI = 0.0
PJ = 0.0
N1 = NH + 1
N2 = NH + 2
DO 40 J = 1, N1, 1
J1 = N2 - J
PP1 = QI(J1) - (ALPHAI * PI - ALPHAJ * PJ)
PPJ = QJ(J1) - (ALPHAI * PJ + ALPHAJ * PI)
PI = PPI
PJ = PPJ
USTI(J1) = PPI
USTJ(J1) = PPJ
CONTINUE
RETURN
END

SUBROUTINE ALG(ALPHAI, ALPHAJ, DELTAI, DELTAJ, USTI, USTJ, NH)
THIS SUBROUTINE WORKS OUT SIGNAL DELTA FOR THE UPDATING OF
THE NEXT ALPHA.
THIS IS THE SECOND VERSION OF THE ALGORITHM IN WHICH THE
MORE COMPLICATED EXPRESSION IS USED.
ALPHAI & ALPHAJ = -VE OF THE RECIPROCAL OF THE ROOT.
DELTAI & DELTAJ = SIGNAL FOR THE UPDATE OF THE NEXT ALPHA.
USTI & USTJ = CONTAINS SIGNALS U(M-1), U(M), U(M+1), U(M+2),
U(M-3), U(M-4), ....... ETC.
NH = NO. OF COMPONENTS IN CHANNEL S.I.R.
DIMENSION USTI(40), USTJ(40)
NH1 = NH - 1
AI = -ALPHAI

10-6-82
SUBROUTINE EQRI(NH, A1, P1, ALPHA, A1, AJ)

THIS SUBROUTINE IS USED TO PASS A SEQUENCE THROUGH THE
EQUALISER SECTION 1 (EQRI). THE EFFECT WOULD BE TO INSERT
A ROOT AT -(ALPHA)*. NOTE THAT -1/(ALPHA) IS THE ROOT OUTSIDE
THE UNIT CIRCLE.

NH = NO. OF COMPONENTS IN S.I.R. OF CHANNEL.
ALPHAI = REAL PART OF ALPHA.
ALPHAJ = IMAG. PART OF ALPHA.
AI = 1-D ARRAY HOLDING REAL PARTS OF OUTPUT AT EQRI.
AJ = 1-D ARRAY HOLDING IMAG. PARTS OF OUTPUT AT EQRI.

DIMENSION AI(40), AJ(40), BI(40), BJ(40)
CCALPI = ALPHAI
CCALPJ = -ALPHAJ
BI(1) = AI(1)
BJ(1) = AJ(1)
DO 10 J = 1, NH, 1
J = J+1
BI(J1) = AI(J1) + (AI(J) * CCALPI - AJ(J) * CCALPJ)
BJ(J1) = AJ(J1) + (AI(J) * CCALPJ + AJ(J) * CCALPI)
10 CONTINUE
N = NH+1
DO 20 I = 1, N, 1
AI(I) = BI(I)
AJ(I) = BJ(I)
20 CONTINUE
RETURN
END
PAGE MISSING IN ORIGINAL
REFERENCES


76. CLARK, A.P. and ABDULLAH, S.N: "Near-maximum-likelihood detectors for voice band channels". To be published.


82. THRALL, R.M. and TORNHEIM, L: "Vector space and matrices", (John Wiley and Sons 1957).


88. Data from Marconi company.


