Signal design for satellite links

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

Additional Information:

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

Metadata Record: https://dspace.lboro.ac.uk/2134/12435

Publisher: © Zhi Chun Zhu

Please cite the published version.
This item was submitted to Loughborough University as a PhD thesis by the author and is made available in the Institutional Repository (https://dspace.lboro.ac.uk/) 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/
SIGNAL DESIGN FOR SATELLITE LINKS

BY

ZHI CHUN ZHU, B.Sc.

A doctoral thesis submitted in partial fulfilment of the requirements for the award of Doctor of Philosophy of the Loughborough University of Technology.

November, 1986

Supervisor: A.P. Clark,
Department of Electronic and Electrical Engineering

© by Zhi Chun Zhu, 1986
# CONTENTS

| ABSTRACT                                      | 1  |
| ACKNOWLEDGEMENTS                              | iii |
| GLOSSARY OF IMPORTANT SYMBOLS AND TERMS       | vi  |
| Chapter 1: **INTRODUCTION**                   | 1  |
| 1.1 Background                                | 2  |
| 1.2 Outline of the investigation              | 18 |
| Chapter 2: **MODEL OF THE DATA TRANSMISSION SYSTEM** | 22 |
| 2.1 General system model                      | 23 |
| 2.2 Satellite links                           | 33 |
| 2.3 Bandpass model for MPSK signals           | 40 |
| 2.4 Uncoded QPSK and 8PSK system models       | 52 |
| Chapter 3: **CODING AND DECODING TECHNIQUES** | 61 |
| 3.1 Convolutional codes                       | 62 |
| 3.2 Convolutional coding with multi-phase signals | 75 |
| 3.3 Viterbi algorithm decoder                 | 100|
| 3.4 A near-maximum likelihood decoding scheme (system A) | 117|
| Chapter 4: **MODULO-M CORRELATIVE-LEVEL CODES FOR MPSK SIGNALS** | 133|
| 4.1 Correlative coded MPSK signals            | 134|
| 4.2 Correlative-level codes for binary data systems | 141|
Chapter 5: NOISE-ADAPTIVE NEAR-MAXIMUM LIKELIHOOD DECODING SCHEMES

5.1 System A1

5.2 System A2

5.3 Comments on the decoder complexity

Chapter 6: ROTATIONALLY INARIANT CONVOLUTIONALLY CODED 8 AND 16 PSK SIGNALS

6.1 Differentially coded system model

6.2 Rotationally invariant codes

6.3 Rate-2/3 rotationally invariant convolutional codes that are transparent to phase changes of \( \frac{\pi}{2} \) and \( \pi \) radians

6.4 Rate-3/4 rotationally invariant convolutional codes that are transparent to phase changes of \( \frac{\pi}{4} \), \( \frac{\pi}{2} \), \( \frac{3\pi}{4} \), and \( \pi \) radians

6.5 A phase error correction system

Chapter 7: DISCUSSION OF RESULTS

7.1 Comparisons of the more promising decoding schemes

7.2 Assessments of coded MPSK signals

7.3 Originality of the study

7.4 Recommendations for further work

Chapter 8: CONCLUSIONS
REFERENCES

APPENDIX A1: Representations of bandpass signals and systems 326

APPENDIX A2: The theorem of maximum likelihood decoding in the presence of additive white Gaussian noise 338

APPENDIX A3: Asymptotic performances of coded signals with the maximum likelihood decoder 347

APPENDIX A4: A mathematical treatment of Shannon's channel capacity concept 356

APPENDIX A5: Definitions and simulation techniques 362

APPENDIX B1: Program for the uncoded QPSK system with coherent detection 374

APPENDIX B2: Program for rate-2/3 convolutionally coded 8PSK signals with the Viterbi decoding 378

APPENDIX B3: Program for rate-2/3 convolutionally coded 8PSK signals with the decoding scheme of system A 381

APPENDIX B4: Program for an exhaustive search for the best correlative code for binary systems 386

APPENDIX B5: Program for rate-2/3 convolutionally coded 8PSK signals with the decoding scheme of system A1 391

APPENDIX B6: Program for rate-2/3 convolutionally coded 8PSK signals with the decoding scheme of system A2 399
The aim of this investigation is to determine the combination of signal coding and modulation for satellite links, that, for a given degree of equipment complexity needed for the detection of the received signal, achieves the best tolerance to noise. Computer simulation tests and theoretical analyses are used to compare the various proposed signal designs.

The trellis coded M-ary phase-shift-keyed (MPSK) modulation method is introduced as the scheme for which different codes are to be devised.

A class of known binary convolutional codes for 8 and 16 PSK signals is studied, and new correlative-level codes using modulo-M arithmetic are designed for MPSK signals.

The soft-decision maximum likelihood Viterbi decoding algorithm is considered for the two proposed signal designs, and a more conventional near-maximum likelihood (reduced-state Viterbi) decoding scheme is also investigated for both types of coded signals.

Two novel decoding schemes, derived from a more conventional near-maximum likelihood decoder, are proposed for coded 8PSK signals. In both decoders the amount of computation involved in decoding each data-symbol is adjusted to meet the prevailing noise level in transmission. Results of extensive computer simulation tests for both decoding schemes are presented. These results suggest that the new schemes come very close to achieving the maximum likelihood decoding of the coded signals without, however, requiring nearly as much
storage and computation per decoded data symbol as does the Viterbi decoder.

The carrier-phase synchronisation problem in a coherent trellis coded MPSK system is investigated. Eight new rotationally invariant rate-2/3 and rate-3/4 convolutional codes for 8 and 16 PSK signals are designed. The new coded MPSK signals, when combined with a simple phase-error correction system proposed for the receiver, are able to tolerate the likely carrier-phase changes in the reference carriers of the coherent demodulation process and therefore avoid the prolonged error bursts that are otherwise caused in the decoded data symbols by such phase shifts. The asymptotic coding gains of the majority of the new codes here are either the same as, or come close to, those of the best known but not rotationally invariant convolutional codes of the same rates and constraint lengths.
ACKNOWLEDGEMENTS

I would like to express my deep gratitude to my supervisor, Professor A.P. Clark, from the Department of Electronic & Electrical Engineering, Loughborough University of Technology, for his constant encouragement and useful guidance throughout this work.

The financial support of Plessey Network & Office System Ltd., Beeston, Nottingham, is gratefully acknowledged.

Many thanks also go to my friends Lesley and Ebrahim Haji-gholan, Ean Goh, and Francis Lee for their patience in reading this thesis.

Finally, I would like to thank my parents, my parents-in-law, my grandmother, my sisters, and my husband for their moral support and encouragement.
GLOSSARY OF IMPORTANT SYMBOLS AND TERMS

\( b_i \)  
L-component vector, \( \{ b_i(1), b_i(2), \ldots, b_i(L) \} \), of binary data symbols

\( b'_i \)  
decoded value of \( b_i \)

\( c_i \)  
cost of the stored vector \( Z_i \), at time \( t=iT \)

\( d_{\text{free}} \)  
minimum free unitary distance of a coded signal

\( d_i \)  
\( 2^L \)-level differentially coded data symbol determined by the vector \( u_i \)

\( d'_i \)  
a possible value of \( d_i \)

\( E_b \)  
average signal energy per information bit

\( E_s \)  
average energy per signal element or symbol

\( G_c \)  
asymptotic coding gain of a code, measured in dB

\( g \)  
number of bits in the memory of a code

\( h(t) \)  
impulse response of the baseband channel

\( (K+1) \)  
constraint length of the code, measured in the number of symbols

\( L \)  
number of different possible data symbols

\( M \)  
total number of distinct phase positions in a PSK signal

\( n \)  
delay in detection (measured in symbols)

\( q_i \)  
a possible value of \( q_i \)

\( q_i \)  
complex-valued symbol fed to the modulator at time \( t=iT \), whose value is determined by the code symbol \( z_i \)

\( r(t) \)  
complex-valued baseband waveform at the output of the demodulator

\( r_i \)  
sample value of \( r(t) \), at time \( t=iT \)

\( s_i \)  
\( 2^L \)-level data symbol determined by the vector \( b_i \)
s'_i
S_i
T
u_i
u'_i
v_i
w(t)
w_i
x_i
z_i
Z_i
(1/2)N_0
\nu
\sigma^2
AWGN:
BER:
decoded data symbol determined by the vector b'_i
the state of a coder, at time t=iT
symbol interval in seconds
L-component vector, \{ u_i(1) u_i(2) ... u_i(L) \}, of differentially coded binary data symbols
decoded value of u_i, or else a possible value of u_i
n-component vector, \{ v_i(1) v_i(2) ... v_i(\nu) \}, of binary code symbols
baseband Gaussian noise waveform at the output of the demodulator
sample value of w(t) at time t=iT
a possible value of b_i or u_i
code symbol (as an integer) determined by the vector v_i at time t=iT
(n+1)-component vector, \{ x_{i-n} x_{i-n+1} ... x_i \}
two-sided power spectral density of the additive Gaussian noise
total number of binary code symbols in the coded vector v_i
variance of real or imaginary part of w_i
additive white Gaussian noise
bit error rate
CHAPTER ONE

INTRODUCTION
1.1 BACKGROUND

The capacity and capability of satellite communication have been growing rapidly since the first commercial geostationary satellite, the Early Bird (Intersat 1), was launched in 1965 [6-13]. With more than twenty years of developments, experiments, and operation experience, the communication satellite technology appears now to have matured. Today there are over 80 satellite communication systems in the world either in active operation or under construction [11], and the use of satellites in many varieties of applications continues to expand steadily.

Traditionally, communication satellite systems have been established mainly to provide long-distance telephone, telegram and telex trunk services, including the distribution of television (T.V.) programs. These services are largely based on the geostationary satellites operating with transparent transponders, travelling-wave-tube (TWT) power amplifiers and relative simple antennas, together with the use of very large and costly ground stations feeding back into a national or regional telecommunication network. Today communication satellites bear over 60% of the total international trunk traffic [13], and it is generally believed that these services will remain dominant in the foreseeable future. Increasingly there has also been an emerging demand for new services, such as the high speed transmission of business documents (texts, pictures and data), inter-computer links, banking, and video-conferencing [6-13].

Although satellite communication started with a fixed
earth station service, their greatest potential applications are in mobile and broadcast services. For these services, the inherent geometric advantages of satellites are overwhelming. Although it is generally accepted that submarine cables, fibre optics, and microwave radios now provide effective competition to satellites in the fixed service, there seems to be no alternative to satellites for the provision of wideband transmission to mobile terminals. Direct broadcast by satellites (DBS) of T.V. signals to homes, where T.V. programs originating in metropolitan centres are transmitted to the satellite by a large earth terminal and broadcasted by the satellite directly to ordinary small home receivers, is now a well established concept and its technical feasibility has been demonstrated beyond doubt.

Another rapidly growing field is that of private network facilities provided by satellites [13]. This encompasses the applications which public networks do not generally serve and commonly use fixed small antenna earth receivers close to users' premises. These important applications include the point to multipoints business services and intermittent short duration connections for high speed data transmission between offices, normally at a transmission rate of several Mbits per second (e.g., 4 to 8 Mbits/second). Clearly, the new facilities which are now becoming feasible with satellites are gradually moving away from the traditional trunk services, in that they are much closer to the end-users, and thus require much smaller and cheaper ground terminals.

Most of existing satellite communication systems today operate below 15 GHz (i.e., the established bands at 4, 6, 11,
12 and 14 GHz), where the earth-space propagation is reasonably good and space-qualified hardware is available [6-13]. As a consequence, some segments of the geostationary orbit are "over-crowded". The growing demand for the conventional facilities has alone caused the serious congestion in the geostationary orbit. The rising demand for the new facilities will inevitably cause more problems.

Broadly speaking, three basic approaches could be adopted to alleviate this problem [36,40]. Firstly, additional spectral resources could be made available for satellite transmission. This would lead to the use of higher frequency bands such as the 30/20 GHz band and/or the multiple reuse of the same band. The frequency reuse is achieved either by using carrier waves with orthogonal polarization or else by employing a multiple-beam antenna on the satellite board.

The second approach is the multiple reuse of the orbital slots. This can be done by clustering a number of satellites operating in different frequency bands into the same geostationary orbit slot. The other method is to employ a large spacecraft platform on which multiple antennas and communication payloads are operated at orthogonal frequency bands, and to assign it to a geostationary orbit slot. The satellite systems adopting frequency and/or orbital reuses will be virtually power-limited. The non-ideal antenna beam and polarization isolation both on the satellite board and at the earth station inevitably increase the co-channel interference level. In addition, signal fading and depolarization due to rainfall also become much more severe, particularly at the higher frequency bands such as 14/12 and 30/20 GHz.
The third approach for alleviating the problem of orbital congestion is to use more efficient transmission techniques. Since the available frequency spectrum and transmission power are both limited for the new generation of communication satellite systems, a detailed review of available transmission and modulation techniques has been motivated. Searching for more band- and power-efficient modulation schemes has been one of the most important and active research areas in recent years. It is quite obvious that whereas the approaches of frequency and orbital reuse are geared toward making more resources available for satellite transmission, the approach of adopting more efficient transmission techniques mainly concerns the conservation of available spectrum.

During the late 70's and early 80's, most major operational satellites used analogue frequency modulation (FM) techniques [7,8]. The predominant access method is Frequency-Division-Multiple-Access (FDMA) [7] in which a number of signals occupying non-overlapping frequency bands are combined to share a single transponder in the satellite. This is based on the extensive experience with the frequency division multiplex technique in terrestrial systems which was available when satellite communication started. It is relatively simple to implement, has lower initial cost, and is less demanding of the repeater and earth station parameters. However, the trend in new developments is to employ digital methods. With the rapid advancement of large-scale-integrated (LSI) circuits technology, leading to the high speed processing of digital signals, it is now possible to implement highly sophisticated techniques of modulation, demodulation and detection for
digital signals cheaply and compactly. Over the last few years extensive research has been carried out into more efficient modulation and detection methods, leading to a wide range of novel systems [3-4]. It is expected that by 1993 almost all new satellite communication system additions will become digital [10].

Digital modulation and digital buffers are also essential for compatibility with the Time-Division-Multiple Access (TDMA) technique [7]. TDMA is the sharing of a satellite transponder by several earth stations which transmit digitally modulated carriers in bursts timed and interleaved so as not to overlap each other at a transponder in the satellite. Since a TDMA system can be reconfigured and programmed for large changes and demand assignment in real time, the flexibility and performance are very attractive. For large systems, a significantly higher capacity can be achieved in comparison to TDMA [7]. Furthermore, since only one signal is processed by the transponder at a time, TDMA can operate close to the saturation level of the TWTA without generating significant interference.

Another innovation becoming feasible with digital communication satellite systems is the use of on-board signal processing regenerative satellite repeaters [7-10] instead of the conventional translating repeaters. The regenerative satellite demodulates the incoming uplink signals into baseband signals and then remodulates them for downlink transmission. By splitting the total link into two distinct sections, the regenerative satellite prevents the accumulation of noise, co-
CHAPTER 1

channel and adjacent interference, leading to the provision of the same performance with reduced earth station and/or satellite power level. This is a very significant step towards reducing the earth station hardware by performing more complex functions on the satellite. The "switchboard-in-sky" concept of regenerative satellite has clearly further enhanced the operational flexibility of digital satellite systems.

One more significant feature of digital communication systems is the possibility of using error control, permitting users the potential of low error rates and high reliability. Error control can be achieved by employing forward error correction (FEC) coding [7,8,15-18]. This investigation concerns one very important aspect of the new generation of digital satellite communication systems, the FEC coding/decoding technique. In particular, it is aimed at determining the most suitable combination of signal coding and modulation together with the most cost-effective decoding/detection scheme, given a degree of equipment complexity needed for the detection of the received signal to achieve the best tolerance to noise.

The remaining part of this section presents an elementary literature review of the available coding, modulation, and coding/detection techniques for digital signals transmitted over satellite links.

In the most general terms, FEC coding may be described as protecting digital information against errors by inserting redundant digits into the information sequence (message). The redundancy is determined by the message to be transmitted in such a way that it improves the "uniqueness" of the given
message. This improvement is normally measured by an increase in the minimum free Hamming distance between different possible (binary) code sequences relative to that between the different possible messages (also in binary). In this way the transmission channel is less likely to corrupt enough symbols of the code sequence to destroy its "uniqueness". Another way to describe FEC coding is that coding introduces a dependence (correlation) between successive symbols of the code sequence, so that each data symbol is the dependent of several code symbols. Even in the case where one or more of these symbols are corrupted by noise during transmission, the message sent can still be extracted correctly from the remaining symbols.

FEC coding improves the tolerance of a digital data transmission system to additive white Gaussian noise, relative to that of the corresponding uncoded system, by permitting a reduction in the signal/noise ratio needed to obtain a desired bit error rate. For satellite communication systems, this gain can result in a reduced transmission power and/or a reduced antenna size at the earth station, and is clearly very attractive.

Following the publication of Shannon's celebrated coding theorem [14], extensive studies have been carried out to discover powerful FEC codes and decoders. The first codes investigated were the block codes, which were mainly derived from rich algebraic mathematics. In these codes, the data symbols (normally binary) which carry the message to be transmitted are grouped into blocks of L symbols. For each block, (v-L) redundant bits, referred to as parity-check digits, are inserted to form a v-bit codeword. Such a code has
a code rate of \( L/v \), which at the same time represents the amount of information carried by each binary code symbol. The redundant data is uniquely determined by the corresponding \( L \) data bits, and it is used by the decoder at the receiver to detect and/or correct errors through some algebraic operations. Block codes have been widely studied and many powerful block codes and decoders have been found [7,15-18]. Some of these, such as the \((128,112)\) double-error-correcting/triple-error-detection extended BCH (Bose-Chaudhuri-Hocquenhem) codes, are particularly suitable for satellite applications [7].

Another important class is the convolutional code [7,8,16-28], where the redundant data is determined by several blocks of data bits rather than one block. For a general rate-\( L/v \) convolutional code, a \( v \)-bit codeword is uniquely determined by the current block of \( L \) data bits and the previous \( K \) blocks of data symbols. It has been readily shown that, when used in conjunction with a good decoder, convolutional codes outperform block codes with the same order of decoder complexity [7]. The error correctibility of convolutional codes normally increases with the code constraint length \((K+1)\), but the decoder complexity will also rise correspondingly. Because of the high coding gains available, convolutional codes appear to be much more attractive. High rate convolutional codes have been used for communication satellite systems [27].

In addition, several concatenated FEC systems have also developed particularly for deep space applications and satellite communications [29-31]. The code concatenation is usually achieved by using an inner short convolutional code with an outer multi-level RS (Reed-Solomon) code [30]. A net
gain of up to 9 dB in tolerance to noise at very high signal/noise ratios has been achieved this way [29].

It must be stressed that these coding gains provided by the FEC coding techniques considered so far can only be achieved at the expense of increased transmission rate or bandwidth expansion. To transmit the redundant digits in addition to the original data bits, the digital modulator must now operate at a higher symbol rate, which correspondingly increases the signal bandwidth, if the information rate is to be maintained. For this reason, the block and convolutional codes of potential interest for bandlimited satellite channels must be those of high code rates, with which a useful improvement in tolerance to noise can be achieved at a moderate bandwidth expansion [27].

In recent years, however, a joint FEC coding and modulation approach has been widely considered for satellite communication systems [32-51, 60-78]. This approach is commonly known as coding with expanded channel symbol alphabet (or simply coding with expanded signal sets). This approach has two very significant features. Firstly, the FEC coding and modulation are no longer treated as two strictly separate items. This permits the optimisation of the system to be carried out for a more effective utilisation of bandwidth and power. It implies that the codes for a particular modulation method will be designed to maximise the minimum Euclidean distance between the different possible modulated carrier waveforms rather than the traditional Hamming distance between different possible binary code sequences. Secondly, the approach offers the possibility of achieving an improved
tolerance to noise without sacrificing either the information rate or bandwidth efficiency. The FEC coding can be created by employing a larger set of signals than that required by non-redundant (uncoded) transmission. This means that the redundant data is now carried by additional signal levels rather than by additional signal elements. For this reason, such codes are also known as the bandwidth-efficient codes. The various coded modulation schemes proposed will be further discussed when the corresponding modulation method is introduced.

In order to reduce the nonlinear distortion introduced by the transmitter high power amplifier (HPA) of the earth station, and thus to simplify the HPA for an acceptable performance of the digital transmission system, constant, or at least near-constant, envelope signals must be used [5]. For this reason, only phase or frequency modulated signal, such as phase-shift-keyed (PSK) and frequency-shift-keyed (FSK) signals are to be considered for satellite links.

At present, a bandlimited quaternary PSK (QPSK) scheme is the prevalent modulation format in use for digital satellite communication systems. Digital FDM/QPSK/FDMA systems have proved themselves in trunk services with high capacity [7,8]. A carefully designed bandlimited QPSK signal can provide a moderate efficiency in the utilisation of both signal bandwidth and power. It has been shown that a filtered QPSK signal in real systems can achieve a bandwidth efficiency of 1.4 bits per Hz of channel bandwidth [66]. In addition, a bandlimited QPSK signal also provides a reasonably good noise and inter-
ference immunity, thus yielding a good compromise within the constraints given. To increase the bandwidth efficiency, multi-phase-shift-keyed (MPSK) schemes such as 8PSK modulation can be employed [7,8].

To improve the power efficiency, a joint convolutional coding and PSK modulation scheme has been devised [33-43]. In particular, the technique of using a rate-2/3 convolutional code, followed by a nonlinear mapping of three binary code symbols into an octal symbol for 8PSK signal generation, has been considered for satellite systems [33-40]. Such a coded 8PSK signal provides a similar bandwidth efficiency as that of the QPSK signal, but a useful improvement in tolerance to noise of above 5 dB, relative to that of the QPSK system, can now be achieved. The technique has also been extended to rate-3/4 convolutionally coded 16PSK signals [41-43].

However, the filtered PSK signals do not have a true constant envelope in that the envelope does falls to zero momentarily at the phase reversal. The major problem in the application of these signals for narrow-band nonlinear satellite channels is that they require significant bandpass filtering before being fed to the HPA at the earth station. The bandlimiting introduces a ripple in the envelope of the signal. When such a signal is fed through the HPA close to saturation level, the envelope ripple is greatly reduced, but the bandwidth is increased correspondingly [5]. However, with an appropriate output back-off from the saturation and by predistorting the signal at the input to the HPA, such that its waveform at the output approaches that of the required bandlimited signal, the nonlinear distortion normally
introduced by the HPA can be largely eliminated, or at least greatly reduced, provided only that the signal is nominally constant envelope [97].

Nevertheless, these schemes are relatively simple to implement in that the conventional modulation, demodulation, and carrier-phase acquisition processes can be used, with hardware that is readily available. In particular, these signals offer the choice of using error control.

Among various constant or near-constant envelope signals proposed, the signals of considerable interest include offset-QPSK signal, minimum shift keyed (MSK) and intersymbol interference and jitter free offset QPSK (IJF-OQPSK) signals [52-59]. These signals can provide improved spectral properties and performances relative to the bandlimited QPSK signals over bandlimited nonlinear satellite channels.

In particular, the MSK signal is a member of a large class of signals where the instantaneous frequency of the carrier signal (that is the rate of change of phase) is constant over the duration of any signal element. The information to be transmitted is now carried by the rate of change in phase of the carrier signal. Many signals of this class also have a continuous phase. Such a signal is named as the continuous phase frequency-shift-keyed (CPFSK) signal [67]. It has been recognised that, the performance of a CPFSK signal can be improved by using a detection process which operates over more than one symbol interval to produce an estimate of the phase trajectory.

Convolutional coding can also be applied to CPFSK signals [76] and an asymptotic coding gain of up to 6 dB may
be expected with such a scheme. The approach here is quite similar to that applied to PSK modulation, although it is now much more complex with the continuous phase signals.

Another signal of this class is the so-called multi-h FSK signal [34,66,67], where the modulation index (that is the h) is allowed to take on a set of fixed values in a cyclic manner over consecutive symbol durations. This is also a coding technique where error control is created by employing a larger set of possible signal shapes in the modulating waveform than that required by non-redundant (uncoded) transmission. An advantage of up to 4 or 5 dB in tolerance to noise can be achieved over the binary FSK signal in this way [34].

The FSK signals previously considered have a constant value of the instantaneous frequency over the duration of any signal element. A much wider range of FSK signals is obtained by permitting the instantaneous carrier frequency to vary over the duration of any signal element. A further development of such signals may be obtained by extending the duration of the frequency modulation pulse so that it extends over more than one symbol interval. These signals, including the majority of those previously reviewed, are generally known as continuous phase modulation (CPM) [60-77]. Since there is often a significant dependence (correlation) between the waveforms of neighbouring signal elements for CPM signals, these also employ correlative phase modulation.

A wide range of CPM schemes, with various degrees of bandwidth efficiency and power efficiency, have been proposed in recent years [67]. For these signals, an improved spectral property can often be obtained by appropriate shaping of the
frequency and phase pulses of the modulated carrier.

On the other hand, a better power efficiency may be achieved by employing a larger set of signal shapes in the modulating waveform than that required by non-redundant transmission. In the latter case, error control can be achieved by incorporating FEC coding explicitly, such as the correlative PSK (CORPSK) signals [65,66]. It can also be achieved by permitting the frequency modulation pulse to extend over several signal intervals. A typical example is the so-called partial response signal [63]. Both techniques are, in fact, equivalent. When an appropriate detector which can exploit this dependence is employed at the receiver, a useful improvement in tolerance to noise may be achieved. Unfortunately, the major disadvantage of these signals is that they generally require a very complex receiver and an accurate carrier phase estimation process.

For many of the signals considered so far, some employing coding explicitly while others do not, there is a significant dependence (correlation) between the waveforms of neighbouring signal elements. When such a signal is received in additive white Gaussian noise, a maximum likelihood detection process selects, as the detected message, the sequence of data symbols which gives the minimum mean-square difference (and therefore the minimum Euclidean distance) between the corresponding received modulated carrier, given the degree of signal distortion in the absence of noise, and the signal actually received [2-3]. If the different possible messages are equally likely, the maximum likelihood detector
minimises the probability of error in the detection of the received message [2-3]. The process itself can, under certain conditions, be implemented by means of the Viterbi Algorithm [2-3,7,79-82]. The Viterbi detector may, however, become unduly complex. Various much simpler detection processes have been developed for a wide range of signals [83-96], and the tolerance to noise of many of these systems is quite close to that of the optimum detector.

For the new satellite communication systems considered, relatively smaller and cheaper earth terminals are required. Thus, simplicity of the receiver is essential. In addition, since the new systems are under the constraints of limited channel bandwidth and transmission power, the modulation schemes (may also employ coding) with improved bandwidth and power efficiencies, relative to the bandlimited QPSK signal, are clearly desirable.

Compared with other schemes considered, the bandlimited MPSK modulation can give a good compromise within the constraints given. In particular, FEC coding may be combined with these signals, which offers the possibility of achieving an improved performance at no bandwidth expansion. Although the multi-h scheme can also provide a considerable large improvement in tolerance to noise, it is rather complex to implement, and in particular, it requires a sophisticated carrier-phase synchronisation process. CPM schemes are also unduly complex, as already mentioned earlier on. For these reasons, the chosen modulation methods in this study are the 8PSK and 16PSK schemes.

Finally, since the performance of any coded signal
depends largely on the type of the detector used, investigation leading to a more cost-effective decoding/detection algorithm should also be undertaken.

The following section of this chapter presents an outline of the investigation. In this work, the various coded modulation and decoding schemes are studied mainly by means of theoretical analysis and computer simulation. The important criterion used to define the cost-effectiveness of any scheme is to achieve the maximum tolerance of the overall system to noise, for a given degree of equipment complexity needed by the decoder/detector, over the range of bit error rates of $10^{-3}$ and $10^{-4}$. 
1.2 OUTLINES OF THE INVESTIGATION

The investigation, of which results are to be presented in this thesis, concerns one aspect of the new generation of digital satellite communication systems, the error control coding/decoding technique. In particular, it is aimed at determining the most suitable combination of signal coding and modulation together with the most cost-effective decoding/detection scheme, given a degree of equipment complexity needed for the detection of the received signal to achieve the best tolerance to noise.

Having already discussed the characteristics of the new satellite systems and the available digital modulation techniques suitable for satellite links, 8 and 16 PSK schemes are chosen as the primary modulation formats for which different codes are to be devised. Both theoretical analysis and computer simulation tests are carried out in the study of various proposed codes and decoding algorithms.

The model assumed for the digital data transmission system is discussed in Chapter 2. The system is here described by a linear baseband channel model. This is followed by a discussion of the characteristics and channel impairments of practical satellite links, including effects of the nonlinear HPA, the signal fading, Doppler shifts, inter-modulation noise, co-channel and adjacent channel interference. The standard MPSK modulation and demodulation processes are then described, followed by uncoded QPSK and 8PSK systems, particularly the coherent threshold-level detection of these signals. The performances of two uncoded systems with coherent detection
will be taken as the standards with which different coded systems are to be compared.

Chapter 3 first reviews the principles of convolutional coding. A well known scheme of combining rate-$L/(L+1)$ binary convolutional codes with $2^{L+1}$-ary PSK modulation is then described. A theoretical analysis of the potential available gains achieved by this approach is presented, and the design technique which ensures good minimum distance properties of coded signals is discussed. Also included in Chapter 3 is the maximum likelihood decoding, achieved by the well known Viterbi algorithm, which is described for convolutionally coded 8PSK signals. A conventional near-maximum likelihood (reduced-state Viterbi) scheme, known as system $A$, which was originally proposed for the detection of digital baseband signals in the presence of intersymbol interference, is also considered for coded 8PSK signals.

Chapter 4 deals with another class of coded PSK signals. These codes differ from the traditional convolutional codes in that they are generally defined over an integer number field, as opposed to a binary field over which a convolutional code is defined. The coding introduces a dependence (or correlation) into the successive information carrying phases of the PSK signal, so that, by incorporating the maximum likelihood decoding/detection process at the receiver, a useful improvement in tolerance to noise is achieved. The designs of codes for both binary and quaternary data symbols are presented, and the new coded signals are also considered for system $A$.

Chapter 5 presents two novel near-maximum likelihood
decoders for coded 8PSK signals. The new schemes are designed to perform a variable amount of computation over each symbol interval, in that the number of operations per decoded data symbol is appropriately adjusted to suit the prevailing noise in transmission. Both systems are derived from the more conventional near-maximum likelihood system A decoder and are therefore named as system A1 and system A2. The philosophy of adopting this "noise-adaptive" strategy is first analysed, which is followed by a detailed description of each decoder. Also in Chapter 5, results of extensive computer simulation tests for the new decoders are presented and optimisation of the system parameters is discussed. Consideration is also given to the processing speed and implementation complexity of the new decoders.

To obtain the best performance of a digital data transmission system, coherent detection must be used. This requires the correct generation of reference carriers at the receiver, especially for those systems using PSK modulation where correct signal detection largely depends on the accurate estimation of the carrier-phase. Chapter 6 examines the carrier-phase synchronisation problem of the coded MPSK system. To resolve the M-fold phase ambiguity in the received MPSK signal, the operation of differential coding and decoding is analysed in the first part of Chapter 6. A rotationally invariant convolutional code that can tolerate part of this ambiguity is then defined, and the procedures for designing rate-2/3 and rate-3/4 rotationally invariant coded 8 and 16 PSK signals are presented. Also in Chapter 6 a trial-by-error carrier-phase correction system, which resolves the remaining phase ambiguity
is proposed for the new coded MPSK signals.

In Chapter 7, assessments and performance evaluations are given for the different coded signals and decoding schemes studied so far. A comment on the originality of this study and an outline of the possible future research area linked with the present work are also presented.

Appendices include descriptions of bandpass signals and systems, the theorem of maximum likelihood decoding in the presence of additive white Gaussian noise, the asymptotic performances of different coded signals with such a decoder, and the Shannon's channel capacity concept. In addition, an introduction of computer simulation techniques is also given, which is followed by a set of computer programs used in the tests.
CHAPTER TWO

MODEL OF THE DATA TRANSMISSION SYSTEM
This chapter is divided into four parts. Section 2.1 describes the general model of the digital data transmission system used in the computer simulation tests of Chapters 3, 4, and 5. Section 2.2 examines the characteristics and channel impairments of practical satellite links. Various assumptions made for the transmission link in Section 2.1 are also discussed. In Section 2.3, MPSK modulation and demodulation processes are presented as the standard schemes with which different codes are devised. Finally in Section 2.4, uncoded QPSK, 8PSK systems and coherent decision threshold-level detection of uncoded QPSK signals are described.

2.1 GENERAL SYSTEM MODEL

A general model of the synchronous $2^L$-level data transmission system used throughout this work is shown in Figure 2.1.1 [1-3]. The baseband signal from the data source in Figure 2.1.1 is the sequence of $2^L$-level data symbols $\{s_i\}$, that carries the information, or message, to be transmitted, where

$$s_i = 0, 1, 2, \ldots, 2^L-1,$$  \hspace{1cm} (2.1.1)

for $i \geq 0$. The $\{s_i\}$ are statistically independent and equally likely to have any of their $2^L$ different possible values. It is assumed that $s_i=0$ for $i<0$, and $s_i$ is the $(i+1)^{th}$ transmitted data symbol at time instant $t=iT$, where $T$ is the symbol interval in seconds. In addition, each data
symbol $s_i$ is itself uniquely determined by the L-component vector of binary data symbols

$$b_i = \{ b_i(1), b_i(2), \ldots, b_i(L) \}$$

(2.1.2)

for $i \geq 0$. The components $b_i(1), b_i(2), \ldots, b_i(L)$ of the vector $b_i$ are statistically independent binary numbers with possible values 0 and 1. Clearly, both $s_i$ and $b_i$ carries $L$ bits of information. The sequence of data symbols $\{s_i\}$ is fed to a coder whose operation is described in Chapters 3 and 4. Coding is generally used to improve the tolerance of the system to noise, over the corresponding uncoded system.

For the given sequence of data symbols $\{s_i\}$ fed to the input, the coder generates the sequence of code symbols (as integers) $\{z_i\}$, where

$$z_i = 0, 1, 2, \ldots, (M-1).$$

(2.1.3)

for $i \geq 0$. $M$ is a positive integer, and $M > 2^L$. The $M$-level code symbol $z_i$ is itself uniquely determined by the $\nu$-component vector of binary code symbols

$$v_i = \{ v_i(1), v_i(2), \ldots, v_i(\nu) \}$$

(2.1.4)

for $i \geq 1$, where $\nu = \log_2 M$. The components $v_i(1), v_i(2), \ldots, v_i(\nu)$ of the vector $v_i$ are binary numbers with possible values 0 and 1. In general, $z_i$ is determined by the vector $v_i$ via the mapping function.
The code symbol $z_i$, at the output of the coder at time instant $t=\tau T$, is converted to the corresponding complex-valued symbol $q_i$, using equation 2.1.6.

$$q_i = \cos(2\pi z_i/M) + j \sin(2\pi z_i/M)$$  \hspace{1cm} (2.1.6)

where $j=\sqrt{-1}$. Thus, in all cases,

$$|q_i|^2 = (\text{Re}(q_i))^2 + (\text{Im}(q_i))^2$$
$$= 1.0$$  \hspace{1cm} (2.1.7)

where $\text{Re}(.)$ and $\text{Im}(.)$ represent the real and imaginary parts of the corresponding quantity respectively.

The corresponding baseband signal waveform, at the input of the linear modulator, may be generally expressed as a sequence of impulses

$$\sum_{i=0}^{\infty} q_i \delta(t-iT)$$  \hspace{1cm} (2.1.8)

regularly spaced at intervals of $T$ seconds. Thus, the signal symbol rate is $1/T$ symbols per second, or bauds. Since, $\delta(t-iT)$ is a unit impulse at time instant $t=iT$, the value or area of each impulse $q_i \delta(t-iT)$ is given by the corresponding complex-valued symbol $q_i$.

Although in practice the signal fed into the modulator at the transmitter is always in the form of a rectangular or rounded waveform, the above assumption is used to simplify the
theoretical analysis of the system. It should be noted that the appropriate change of the transmitter filter must be made in practice to give the same actual transmitted waveform.

The linear modulator at the transmitter is a combination of in-phase and quadrature suppressed carrier amplitude modulators. It also incorporates all transmitter equipment filters, e.g. the appropriate pre-modulation (or post-modulation) filtering. The latter filter is normally required to restrict the bandwidth of the transmitted signal. The term "linear modulation" implies that the frequency spectrum of the baseband signal (that has been appropriately low-pass filtered) is linearly translated (shifted) to the available band of the given bandpass transmission path, and is later shifted back to the original baseband (lowpass band) through a linear coherent demodulation process at the receiver. The operation of modulation and coherent demodulation processes is examined in Section 2.3.

The modulator, the bandpass transmission path formed by the satellite link, and a linear coherent demodulator at the receiver, together form the linear baseband channel. In practice the satellite link is basically a nonlinear channel. However, since the primarily concern of this investigation is to determine the most cost-effective coding/decoding scheme for satellite links, the channel is taken to be linear in order to simplify the channel model used in the computer simulation tests. The nonlinearity of the satellite channel is discussed in Section 2.2, where a more detailed description of the link model is given.
The channel model shown in Figure 2.1.1 assumes that, over the frequency band of the transmitted signal, the transmission path does not introduce any significant signal distortions. However, it does introduce a time-varying phase change in the signal carrier. Furthermore, the only noise introduced by the system is assumed to be wideband stationary White Gaussian noise with zero mean and a constant two-sided power spectral density of $(1/2)N_0$. The white Gaussian noise waveform is added to the signal waveform at the output of the transmission path. Other types of interference that may occur in practical satellite channels are ignored. Some of the important interfering signals are discussed briefly in Section 2.2.

A bandpass filter is normally employed at the receiver input to remove the noise components outside the frequency band approximately corresponding to the bandwidth of the received signal, without excessively bandlimiting the signal itself. The demodulator at the receiver includes two linearly coherent demodulators whose reference carriers are in phase quadrature at exactly the same frequency as that of the received signal. The two demodulators are followed by two appropriate lowpass filters. The baseband waveforms at the outputs of the in-phase and quadrature demodulators are taken to be the real and imaginary parts of the corresponding complex-valued demodulated signal. Perfect timing and carrier frequency synchronisation is assumed throughout, and unless otherwise stated, the correct carrier phase synchronisation is also taken to be achieved at the receiver.

The model of the resultant linear baseband channel used
in the computer simulation tests is now defined. The given channel has a transfer function \( H(f) \) and an impulse response \( h(t) \), where \( h(t) \) is the inverse Fourier Transform (F.T.) of \( H(f) \). The resultant baseband signal, at the output of the demodulator, is the complex-valued waveform

\[
\begin{align*}
\mathbf{r}(t) &= \sum_{i=0} q_i \delta(t-iT) * h(t) + w(t) \\
&= \sum_{i=0} q_i h(t-iT) + w(t) 
\end{align*}
\]  

(2.1.9)

where \( w(t) \) is the complex-valued noise waveform at the output of the demodulator, and \( * \) denotes the convolution operation. The waveform \( \mathbf{r}(t) \) is sampled once per signal element, at time instants \( \{iT\} \), to give the sample values \( \{r_i\} \), where \( r_i = \mathbf{r}(iT) \).

In order for a received sample \( r_i \) to contain no intersymbol interference, the resultant transfer function \( H(f) \) of the linear baseband channel must satisfy Nyquist's vestigial symmetry theorem [1-2,8]. This states that, if \( H(f) \) is a real-valued even function of \( f \) (symmetrical about \( f=0 \)) and if it has odd symmetry about the nominal cut-off frequencies \( \pm 1/2T \) Hz, then the corresponding impulse response \( h(t) \) is a real-valued even time-function (symmetrical about \( t=0 \)) and \( h(iT)=0 \) for all nonzero integer values of \( i \). A transfer function that satisfies Nyquist's vestigial symmetry theorem is often approximated by equation 2.1.10 and is shown in Figure 2.1.2.
\[
H(f) = \begin{cases} 
T/2(1-\sin \frac{\pi(|f|-1/2T)}{2f_\alpha}) & \text{for } |f|<1/2T-f_\alpha \\
0, & \text{otherwise}
\end{cases}
\]

For mathematical convenience, the time delay in transmission is neglected, so that \(h(t=0)=0\), and ideally, \(h(t=0)=1\). It now follows that the received sample, at time \(t=iT\), is given by

\[
r_i = q_i + w_i
\] (2.1.11)

It can be shown that \([3]\) the real and imaginary parts of the noise components \(\{w_i\}\) are statistically independent Gaussian random variables with zero mean and fixed variance, the real and imaginary parts of any \(w_i\) being statistically independent of each other. This is a highly desirable situation where FEC coding is used to generate the \(\{q_i\}\) from the \(\{s_i\}\).

Furthermore, it is taken that the resultant filtering of the baseband signal is equally shared between the transmitter and receiver filters, and the two filters are designed to have a bandwidth close to \(1/2T\) Hz, so that the sampling rate of \(1/T\) samples per second is quite close to the Nyquist rate for \(r(t)\).

The sequence of sampled values \(\{r_i\}\) is next fed to a decoder/detector, whose output is the sequence of decoded data symbols \(\{s'_i\}\). In the absence of noise, the latter sequence of \(\{s'_i\}\) must be the same as the \(\{s_i\}\) fed to the coder at the transmitter. The terms "decoder" and "detector" are made inter-changeable in this thesis, for the fact that the detection
of the given coded signals must also include the appropriate decoding operation. Finally, soft-decision decoding of the received signal has been generally assumed throughout, and this means that the received samples \( \{r_1\} \) are infinitely quantised.
Fig. 2.1.1: General model of the data transmission system

Data source $\{s_i\}$ → Coder $\{z_i\}$ → Mapping process $\{q_i\}$ → Transmitter filter & modulator → Receiver filter & coherent demodulator

User $\{s_i\}$ → Decoder /detector $\{r_i\}$ → $r(t)$

White Gaussian noise → Transmission path
Fig. 2.1.2: Baseband channel transfer function giving a signal spectrum with a sinusoidal roll-off
2.2 **SATellite Links**

The satellite link in real systems is basically a nonlinear channel. The link normally contains, at the transmitting earth station, a frequency translation equipment called the up-converter, a bandpass filter, and a high power amplifier (HPA). The up-converter is needed to change the carrier frequency to that required for the transmitted signal, thus permitting the modulation process itself to be carried out at a lower frequency. The bandpass filter removes the spurious frequency components generated at the modulator, and finally the HPA feeds the transmitter output signal to the antenna.

A typical HPA introduces both amplitude and phase distortions to the transmitted signal. Figure 2.2.1 shows an example of how the amplitude and phase of the output carrier from a HPA varies with the amplitude of the input carrier, resulting in AM-AM and AM-PM conversion effects. It is evident that, if an unacceptable level of signal distortion is to be avoided, amplitude modulation can only be used here with a large reduction (back-off) in the output signal level relative to the maximum level passed by the HPA. The latter is set to unity in Figure 2.2.1 and is called the saturation level. The need to operate with a large back-off increases the cost of the HPA, for a given signal level, and also reduces its power efficiency. Therefore, it is highly desirable that the transmitted signal has a constant or, at least, a near-constant envelope, enabling the HPA to be used close to saturation. It is of course assumed here that only a single modulated carrier signal is transmitted through the HPA.
The modulated carrier signal travels in the form of 
electro-magnetic radiation from the transmitting earth station 
antenna to the satellite, where a transponder amplifies and 
retransmits the signal towards the receiving earth station, 
with usually a different carrier frequency from the received 
signal at the satellite.

At the receiving earth station it is the receiver 
antenna which feeds the received signal to the receiver input 
amplifier. The latter is followed by a down-converter that is 
needed to change the carrier frequency to that required by the 
demodulator.

Considerable power is normally available for the signal 
transmitted from the transmitting earth station to the sate-
llite (the up-link), whereas only a relative low power level 
can be used for the signal transmitted from the satellite to 
the receiving earth station (the down-link). Therefore, most 
of the more serious additive interference in transmission 
originates in the down-link. The signal is attenuated by its 
passage through the atmosphere, the attenuation being increased 
significantly by heavy rainfall. The attenuation usually 
varyes over a range of only a few dB, but may sometimes vary by 
as much as 20 dB. Nevertheless, in relation to the trans-
mission rate of the digital signal, measured in elements per 
second, the fading rate is very small. The effect of signal 
fading is therefore neglected in this work.

Any movement of the satellite relative to the earth 
station introduces a Doppler shift into the received signal. 
This is a time modulation of the received signal, which 
introduces a frequency offset (change in frequency) into the
CHAPTER 2

signal carrier and also slightly changes the signal element rate. Whereas a geostationary satellite does not usually introduce any significant Doppler shifts, the satellite which travels closer to the surface of the earth over part of its orbit can introduce considerable Doppler shifts. The frequency shift introduced into the received signal can exceed 500 kHz, with a maximum rate change of around 5 kHz/sec or more. Special arrangements must therefore be made at the receiver to handle the severe time-varying Doppler-shifts introduced by the satellite. In all the channel models used in this work, exact correction of any frequency offset in the received signal (such as can be caused by a Doppler-shift) is taken to be achieved in the coherent demodulator at the receiver.

The applications to be considered in this work are those where Frequency Division Multiple Access (FDMA) is used. In a FDMA system, several signals originating from different earth stations and occupying different frequency bands are processed simultaneously by a single transponder on the satellite. The amplifier here can be a travelling-wave-tube (TWTA), whose characteristics is similar to those shown in Figure 2.2.1. It is assumed that a reasonable large number of signals are frequency-division multiplexed and a sufficient output back-off is used in the TWTA, so that, for practical purposes, the nonlinear distortions in forms of the AM-AM and AM-PM conversion effects indicated by Figure 2.2.1 are not introduced into individual signals by the satellite. Consequently, the satellite link can be considered to approximate to a linear channel, with the exception of the HPA at the output of the earth station transmitter. However, the nonlinear character-
istics of the TWTA introduces inter-modulation noise in a FDMA system, resulting in the generation of a series of new frequency components (not received by the satellite), and these frequency components may interfere with some of the signals received by the earth station. This is one type of additive noise.

The bandlimiting of a modulated carrier signal, even when this initially has a constant envelope, results in a ripple in the envelope. If such a signal is fed through an HPA at close to saturation, the envelope ripple is greatly reduced and the bandwidth is correspondingly increased. However, with an appropriate output back-off from saturation and by pre-distorting the signal at the input of the HPA, such that its waveform at the output approaches to that of the required bandlimiting signal, the nonlinear distortion introduced by the HPA can be greatly reduced, so long as that the signal is nominally with a constant envelope, such as where phase or frequency modulation is used. The satellite link may, from now on, be considered as a linear channel, for all practical purposes. Consequently, the baseband channel in Figure 2.1.1 is taken to be a linear channel.

Apart from the channel impairments discussed above, in a satellite link there may be adjacent-channel interference [8], which is the interference from a signal occupying a frequency band nominally adjacent to the wanted signal, and there may also be co-channel interference [8], which is the interference from transmitted signal occupying the same frequency band. Finally, there is always present a wideband random noise, originating from various sources that include the
external sources, the receiving antenna and the input circuits of the earth station. This noise has an essentially constant power spectral density over the frequency band of the received signal and has some properties similar to those of additive white Gaussian noise. This noise is of great importance since it places a strict lower limit on the received signal level that can be handled without an excessive error rate in the detected data symbols.

No doubt the satellite link also introduces different types of additive noise. However, because of the differing natures of the various sources of additive noise, its resultant characteristics may well vary from one location and/or time to another, so that no precise model of the noise is entirely satisfactory, particularly if it is to be amendable to theoretical calculation. For these reasons, the various systems considered in this work are compared on the basis of their tolerances to additive white Gaussian noise, which is easy to analyse theoretically and also easy to simulate on a computer. Not only is it similar to the wideband background noise, but it is also the least predictable of the different types of stationary noise and is therefore, in this sense, the most random.

Furthermore, such evidence as is available tends to suggest that, although the tolerance of a data transmission system to additive white Gaussian noise is not necessarily a good measure of its actual tolerance to any particular type of additive noise, the relative tolerances to white Gaussian noise of different systems are a reasonably good measure of their relative overall tolerances to the additive noise likely to be
experienced in practice.

Most of the information provided in this section is attributed to the author's supervisor, Professor A.P.Clark, (see reference 97 for details), and with his permission, the material is presented here for the completeness of this thesis.
Fig. 2.2.1: Characteristics of HPA
2.3 BANDPASS MODEL FOR MPSK SIGNALS

The modulation method considered throughout this work is the multi-phase shift keyed (MPSK) scheme, where $M=4$, 8 and 16, unless otherwise stated. The detailed bandpass model for MPSK signals is shown in Figure 2.3.1.

The total transmitter filter in Figure 2.1.1 is now given by the resultant of two lowpass filters and bandpass filter at the transmitter in Figure 2.3.1, while the total receiver filter in Figure 2.1.1 is given by the two lowpass filters and bandpass filter at the receiver in Figure 2.3.1. The two multipliers at the transmitter form the linear modulator, whereas the linear coherent demodulator comprises of two multipliers, together with the two following lowpass filters at the receiver. The white Gaussian noise waveform $n(t)$ in Figure 2.3.1 is real-valued and has a constant two-sided power spectral density of $(1/2)N_0$.

Each of the two lowpass filters in the transmitter has a real-valued impulse response $a(t)$ and transfer function $A(f)$, whereas each of the two lowpass filters at the receiver has a real-valued impulse response $b(t)$ and transfer function $B(f)$. The two bandpass filters and the bandpass transmission path together form a linear bandpass channel, which carries the FSK signal, the transfer function of the given channel being $C(f)$. The transmitter bandpass filter has a transfer function $P(f)$ and is required to remove the spurious frequency components generated in the modulator, whereas the receiver bandpass filter has a transfer function $Q(f)$ and is used to remove the noise components outside the frequency band of the received
signal. Given that the transfer function of the bandpass transmission path is $L(f)$, the transfer function of the given channel is

$$C(f) = P(f) L(f) Q(f)$$  \hfill (2.3.1)

For MPSK schemes, an ideal modulated carrier has $M$ distinct phase states, the given phase being maintained as constant during each symbol interval $T$. The corresponding complex-valued symbol $q_1$, at the input to the linear baseband channel in Figure 2.1.1, is given by

$$q_1 = \cos \phi_1 + j \sin \phi_1$$  \hfill (2.3.2)

where $j = \sqrt{-1}$, and $\phi_1 = 0, \frac{2\pi}{M}, \frac{4\pi}{M}, \ldots, (M-1)\frac{2\pi}{M}$ radians. Following equation 2.1.7, the average energy per signal element or symbol, defined as the mean-square value of $q_1$, is

$$E_s = E(|q_1|^2) = 1.0$$  \hfill (2.3.3)

where $E(.)$ represents the expected value of the corresponding quantity. Clearly, the $M$ distinct values of $q_1$ may be represented by a set of points regularly spaced on a unit circle in the complex number plane. The phase locations, also known as the signal constellations, for 4, 8 and 16 PSK modulation schemes are shown in Figure 2.3.2.

The real and imaginary parts of the complex-valued signal elements $\{q_1(t-iT)\}$, are fed separately to the two low-
pass filters at the transmitter. For an ideal MPSK signal, each of these two filters has an impulse response

\[ a(t) = \begin{cases} \frac{1}{\sqrt{T}} & \text{for } 0 \leq t \leq T \\ 0 & \text{elsewhere} \end{cases} \]

(2.3.4)

The PSK signal generated in this way has a constant envelope but a rather wide bandwidth. In practice, the appropriate bandpass (post-modulation) filtering is normally applied to the modulated carrier to restrict the radiated bandwidth of an ideal (unfiltered) PSK signal. The MPSK signal considered in this work is a combination of in-phase and quadrature double-sided suppressed carrier amplitude modulated signals, where the two baseband modulating waveforms (that correspond to the real and imaginary parts, respectively, of the baseband signal) are appropriately shaped to give the required (filtered) transmitted MPSK signal.

The corresponding baseband modulating waveforms, at the outputs of the two lowpass filters in the transmitter, are \( \cos \phi_i a(t-iT) \) and \( \sin \phi_i a(t-iT) \), respectively. These two signals are multiplied separately by two sinusoidal carriers that are in phase quadrature but have the same instantaneous frequency \( f_c \). The signal at the output of the adder (input to the bandpass filter) in Figure 2.3.1 is the real-valued waveform

\[
x(t) = \sqrt{2} \sum_{i=0}^{I} \left[ \cos \phi_i a(t-iT) \cos(2\pi f_c t) - \sin \phi_i a(t-iT) \sin(2\pi f_c t) \right]
\]

\[
= \sqrt{2} \sum_{i=0}^{I} \text{Re}[q_i a(t-iT) \exp(j2\pi f_c t)]
\]

(2.3.5)
Using the well known identity

\[ \text{Re}(z) = \frac{(z + z^*)}{2} \]  \hspace{1cm} (2.3.6)

where \( z^* \) is the complex conjugate of \( z \).

\[ x(t) = \left(\frac{1}{\sqrt{2}}\right) \sum_{i=0}^{\infty} q_i a(t-iT) \exp(j2\pi f_c t) \]
\[ + \left(\frac{1}{\sqrt{2}}\right) \sum_{i=0}^{\infty} q^*_i a^*(t-iT) \exp(-j2\pi f_c t) \]  \hspace{1cm} (2.3.7)

The Fourier Transform (F.T.) of \( x(t) \) is

\[ X(f) = \left(\frac{1}{\sqrt{2}}\right) \sum_{i=0}^{\infty} q_i A(f-f_c) \exp[-j2\pi(f-f_c) i T] \]
\[ + \left(\frac{1}{\sqrt{2}}\right) \sum_{i=0}^{\infty} q^*_i A(f+f_c) \exp[-j2\pi(f+f_c) i T] \]  \hspace{1cm} (2.3.8)

where \( A(f) \) is the F.T. of \( a(t) \).

In the absence of noise, the F.T. of the real-valued waveform \( y(t) \), at the output of the receiver bandpass filter, is

\[ Y(f) = X(f)C(f) \]  \hspace{1cm} (2.3.9)

where \( C(f) \) is, of course, the transfer function of the given bandpass channel, formed by two bandpass filters and the bandpass transmission path in Figure 2.3.1.

In the bandpass model of MPSK signals in Figure 2.3.1, the imaginary values are ascribed to the signals following the second of the two multipliers in the demodulator (that where the signal is multiplied by \( \sin(2\pi f_c t + \phi) \)). The model assumes that the correct reference carrier frequency is generated at
the receiver. The two multipliers in the demodulator together multiply the received signal $y(t)$ by

$$
\sqrt{2}\cos(2\pi f_c t + \phi) - j\sqrt{2}\sin(2\pi f_c t + \phi)
= \sqrt{2}\exp[-j(2\pi f_c t + \phi)]
= \exp(-j2\pi f_c t)\sqrt{2}\exp(-j\phi)
$$

(2.3.10)

The multiplication of $y(t)$ by $\exp(-j2\pi f_c t)$ causes $Y(f)$ to be shifted in the negative direction by $f_c$ Hz and hence to be replaced by $Y(f+f_c)$. The following lowpass filter removes the high frequency (bandpass) component of $Y(f+f_c)$, centred on $-2f_c$, leaving the low frequency (baseband) component appropriately modified.

Let the low frequency component of $Y(f+f_c)$ be $Y'(f)$. It can be shown [97] that, following equations 2.3.8 and 2.3.9, the F.T. of the demodulated baseband signal $r(t)$, in the absence of noise, is

$$
R(f) = Y'(f)B(f)\sqrt{2}\exp(-j\phi)
= X'(f)C'(f)B(f)\sqrt{2}\exp(-j\phi)
= (1/\sqrt{2}) \sum q_i \exp(-j2\pi f_i T) A(f)C'(f)B(f)\sqrt{2}\exp(-j\phi)
= \sum q_i \exp(-j2\pi f_i T) A(f)C'(f)B(f)\exp(-j\phi)
$$

(2.3.11)

where $X'(f)$ and $C'(f)$ are the low frequency components of $X(f+f_c)$ and $C(f+f_c)$, respectively. Bearing in mind that

$$
q_i \exp(-j2\pi f_i T)
$$

(2.3.12)

is the F.T. of the impulse
being fed to the input of the linear baseband channel in Figure 2.1.1, it is clear that the transfer function of the given baseband channel must be

\[ H(f) = A(f)C'(f)B(f)\exp(-j\phi) \]  

(2.3.14)

and in the absence of noise, the corresponding demodulated baseband signal is given by

\[
\begin{align*}
r(t) &= \sum_{i=0} q_i \delta(t-iT) * h(t) \\
&= \sum_{i=0} q_i h(t-iT)
\end{align*}
\]  

(2.3.15)

where \( h(t) \) is the inverse F.T. of \( H(f) \). The result here follows that given by equation 2.1.9.

Now, let \( P'(f), L'(f) \) and \( Q'(f) \) be the low frequency components of \( P(f+f_c) \), \( L(f+f_c) \) and \( Q(f+f_c) \), respectively, then, following equation 2.3.1,

\[ C'(f) = P'(f)L'(f)Q'(f) \]  

(2.4.16)

Substituting for \( C'(f) \) into equation 2.3.14, the baseband channel transfer function is

\[ H(f) = A(f)P'(f)L'(f)Q'(f)B(f)\exp(-j\phi) \]  

(2.3.17)
It follows that a linear bandpass transmission path, preceded by a linear modulator, and followed by a linear coherent demodulator, can be considered as the corresponding baseband channel, whose transfer function is determined in the following manner. The frequency response of each of the separate bandpass filters (which, when connected in cascade, form the resultant bandpass channel) is converted into the corresponding baseband characteristic by deleting the negative frequency part of the frequency response and shifting the positive frequency part down in frequency by $f_c$ Hz, to give a baseband characteristic, which is the required transfer function. This concept is further examined in Appendix A1.

Consequently, the transfer function of the transmitter filter in Figure 2.1.1 is now given by

$$D(f) = A(f)P'(f)$$  \hspace{1cm} (2.3.18)

and the transfer function of the receiver filter in Figure 2.1.1 is

$$E(f) = Q'(f)B(f)$$  \hspace{1cm} (2.3.19)

so that

$$H(f) = A(f)P'(f)L'(f)Q'(f)B(f)\exp(-\phi)$$

$$= D(f)L'(f)E(f)\exp(-\phi)$$  \hspace{1cm} (2.3.20)

Since the given bandpass transmission path is assumed to be an ideal channel in that it introduces no delay, attenu-
ation, and signal distortions, over the frequency band of the transmitted signal,

\[ |L(f)| = 1 \]  

(2.3.21)

over the transmitted signal frequency band. Therefore,

\[ |L'(f)| = 1 \]  

(2.3.22)

is assumed over the corresponding baseband. Furthermore, if the transmission path introduces a phase advance of \( \phi \) radians into \( r(t) \), this is equivalent to multiplying the received signal by \( \exp(j\phi) \). However, the phase advance of \( \phi \) radians in the reference carriers at the receiver relative to the transmitted PSK signal, introduces the factor \( \exp(-j\phi) \) into the received signal \( r(t) \). This corrects the phase advance of \( \phi \) radians introduced by the transmission path, so that correct carrier-phase synchronisation is achieved at the demodulator. Consequently,

\[ H(f) = D(f)E(f) \]  

(2.3.23)

It has been assumed throughout the previous analysis that there is no additive noise. In the presence of the additive white Gaussian noise, the output of demodulator in Figure 2.3.1 becomes

\[ r(t) = \sum_{i=0}^{\infty} q_i h(t-iT) + w(t) \]  

(2.3.24)
where $w(t)$ is the complex-valued baseband noise waveform. The important assumption here is that the transfer function $E(f)$ of the receiver filter in Figure 2.1.1, which is formed from the combination of the bandpass filter and two lowpass filters at the receiver in Figure 2.3.1, is a real and even function. That is, $E(f)$ is symmetrical about $f=0$. Under such conditions, any sample of the real part of $w(t)$ and any sample of the imaginary part of $w(t)$, regardless of their time instants, are statistically independent Gaussian random variables with zero mean and fixed variance [1-3]. This is a highly desirable situation where FEC codes are used to generate the $\{q_i\}$ from the $\{s_i\}$.

Suppose next the two-sided power spectral density of the wideband real-valued Gaussian noise waveform $n(t)$ in Figure 2.3.1 is $(1/2)N_0$, then the two-sided power spectral density of each of the real and imaginary parts of the bandlimited noise waveform $w(t)$, at the output of the coherent linear demodulator can be shown to be [19]

$$\frac{1}{2}N_0 |E(f)|^2$$

(2.3.25)

Thus, the auto-correlation function of each of the real and imaginary parts of $w(t)$, for a time separation of $t$ seconds between the two sampling instants, is

$$N(t) = \frac{1}{2}N_0 \int_{-\infty}^{+\infty} |E(f)|^2 \exp(j2\pi ft) \, df$$

(2.3.26)

Since it is assumed that the resultant filtering of the
baseband signal is equally shared between the transmitter and receiver filters.

\[ E(f) = D(f) = H(f)^{1/2} \]  

(2.3.27)

Thus, \[ N(t) = \frac{1}{2}N_0 \int_{-\infty}^{+\infty} H(f) \exp(j2\pi ft) df \]

\[ = \frac{1}{2N}N_0 h(t) \]  

(2.3.28)

Finally, for the channel transfer-function shown in Figure 2.1.2, \( h(t) = 1 \), at \( t=0 \). Consequently, the variance of each of the real and imaginary parts of the noise components \( \{w_i\} \) is given by

\[ \sigma^2 = N(t=0) \]

\[ = \frac{1}{2}N_0 \]  

(2.3.29)

Furthermore, for any nonzero integer \( i \),

\[ N(it) = \frac{1}{2}N_0 \cdot h(it) = 0 \]  

(2.3.30)

But the sampling instants for any two noise samples \( w_i \) and \( w_j \) are separated by a multiple of \( T \) seconds, and the noise samples have zero mean. Thus any two noise samples \( w_i \) and \( w_j \) are sample values of uncorrelated and statistically independent random variables [1-3].
Fig. 2.3.1: Bandpass model for M-ary PSK signals

\[ \sum \text{Re}(q_i) \delta(t - iT) \rightarrow \sqrt{2} \cos(2\pi f_c t) \]

\[ \sum \text{Im}(q_i) \delta(t - iT) \rightarrow -\sqrt{2} \sin(2\pi f_c t) \]

\[ \sqrt{2} \cos(2\pi f_c t + \phi) \]

\[ L(t); X(t) \]

\[ \text{LPF} \ a(t); A(f) \]

\[ \text{LPF} \ b(t); B(f) \]

\[ \text{LPF} \ c(t); C(f) \]

\[ \text{LPF} \ d(t); D(f) \]

\[ \text{BPF} \ P(f) \]

\[ \text{BPF} \ Q(f) \]

\[ n(t) \]

\[ r(t); R(f) \]

\[ j = \sqrt{-1} \]

CHAPTER 2
Fig. 2.3.2: Constellations of 4, 8, and 16 PSK signals

**QPSK**

**8PSK**

**16PSK**
2.4 UNCODED QPSK AND 8PSK SYSTEM MODELS

Without the use of FEC coding, the block "coder" in Figure 2.1.1 is not incorporated, and the $2^L$-level data symbols $\{s_i\}$ are directly converted to the corresponding complex-valued symbols $\{q_i\}$ via an appropriate mapping process (the differential precoding operation is not considered here). The mapping is such that $q_i$ is uniquely determined by a particular value of $s_i$, in the complex number plane. Thus $q_i$ also has $2^L$ different possible values, the $\{s_i\}$ and $\{q_i\}$ being statistically independent and equally likely to have any of their $2^L$ possible values.

The uncoded QPSK and 8PSK systems, where each signal element carries 2 and 3 information bits, respectively, are taken to be the standard schemes with which the performances of different coded systems are to be compared. In the case of a QPSK signal where $L=2$, the data symbol $s_i$ is uniquely determined by the 2-component vector of binary data symbols

$$b_i = [ b_i(1) \ b_i(2) ]$$

via the Gray coding function defined by Table 2.4.1. Similarly, for an 8PSK signal (where $L=3$), each $s_i$ is determined by the 3-component vector of binary data symbols

$$b_i = [ b_i(1) \ b_i(2) \ b_i(3) ]$$

via the Gray coding function defined in Table 2.4.2. In both cases, the corresponding complex-valued signal symbol $q_i$ is
determined by

\[ q_i = \cos\left(\frac{2\pi s_i}{2^L}\right) + j\sin\left(\frac{2\pi s_i}{2^L}\right) \]  

(2.4.3)

where \( j = \sqrt{-1} \), \( L = 2 \) and 3. It is clearly seen from Tables 2.4.1 and 2.4.2 that, with the Gray coding function, the two vectors \( \{b_i\} \) must differ in one of their \( L \) binary components only, if their corresponding transmitted symbols \( \{q_i\} \) are the two adjacent signals in the complex number plane.

Now, following equation 2.1.11, the sample value of the demodulated baseband signal \( r(t) \), at time \( t = iT \), is

\[ r_i = q_i + w_i \]  

(2.4.4)

where \( q_i \) is uniquely determined by the transmitted data symbol \( s_i \), at time \( t = iT \). The sequence of received samples \( \{r_i\} \) is next fed to a detector, whose output at time \( t = iT \) is the detected data symbol \( s'_i \).

It can be shown (see Appendix A2 for the details) that, in the presence of additive white Gaussian noise, and when the \( \{s_i\} \) are equally likely to have any of the \( 2^L \) possible values, as is assumed here, the detector which minimises the probability of detection error, selects as the detected data symbol \( s'_i \), the possible value of \( s_i \) that corresponds to the transmitted symbol \( q_i \), where \( q_i \) is closest to the received sample \( r_i \) in the complex number plane. Such a detector is known as the maximum likelihood detector.

The maximum likelihood detection process is now briefly described for uncoded QPSK signals. The detection of uncoded
8PSK signals follows the same procedure.

For QPSK signals, the \( q_1 \) which is closest to the received sample \( r_1 \), in the complex number plane, is determined by using two decision thresholds in the complex number plane, as shown in Figure 2.4.1. It also includes the Gray coding of \( \{ [b_1(1), b_1(2)] \} \) to the \( \{ s_1 \} \) and thus \( \{ q_1 \} \). The detector operates such that if, for instance, \( r_1 \) falls into the region between the thresholds where the value of \( q_1 \) is mapped from the data symbol \( s_1 = 0 \), then the detected data symbol \( s'_1 \) is taken to be \( s'_1 = 0 \).

At high signal/noise ratios, by far the most likely error in the detection of a quaternary element is the one which involves the crossing of the nearest decision threshold. With Gray coding, as defined in Figure 2.4.1, such a detection error will only cause one of the two binary components in the vector \( b_1 \) being incorrect. Consequently, the overall bit error rate of a QPSK signal is only slightly greater than of a binary PSK signal, at a given signal/noise ratio.

In general, if \( s_1 \) (and thus \( q_1 \)) is equally likely to have any of its \( 2^L \) possible values, it can be shown [7, 8] that the average probability of error event in the detection of \( s_1 \) is

\[
P(E) = 2Q\left( \frac{d_{\text{uncoded}}}{2\sigma} \right) \tag{2.4.5}
\]

where \( d_{\text{uncoded}} \) is the Euclidean distance between any two nearest symbols \( \{ q_1 \} \) (uncoded) in the complex number plane, \( \sigma \) is the variance of real and imaginary parts of the noise components \( \{ w_1 \} \) in the \( \{ r_1 \} \), and
\[ Q(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \exp\left(-\frac{y^2}{2}\right) \, dy \]  

(2.4.6)

At high signal/noise ratios, most of such errors will only cause a single bit error. Thus, the probability of bit error, \( P_b \), can be approximated to

\[ P_b = \frac{2}{Q\left(\frac{d_{\text{uncoded}}}{2\sigma}\right)} \]  

(2.4.7)

which can be taken to be

\[ P_b = Q\left(\frac{d_{\text{uncoded}}}{2\sigma}\right) \]  

(2.4.8)

at very low bit error rates (such as \( P_b < 10^{-4} \)). From Figure 2.3.2, it is clear that

\[ d_{\text{uncoded}}^2 = 2.0 \, E_s \]  

(2.4.9)

for QPSK signals, and

\[ d_{\text{uncoded}}^2 = 0.5858 \, E_s \]  

(2.4.10)

for 8PSK signals, where \( E_s = 1.0 \) (see equation 2.1.7). In addition the average signal energy per data bit, \( E_b \), is defined to be

\[ E_b = \frac{(1/L)E_s}{2^L} \]  

(2.4.11)

for a general \( 2^L \)-level data transmission system. Therefore,
\[ E_b = (1/2)E_s \]  
(2.4.12)

for QPSK signals, and

\[ E_b = (1/3)E_s \]  
(2.4.13)

for 8PSK signals. From equation 2.3.29,

\[ \sigma^2 = (1/2)N_o \]

Now, following equations 2.4.8 to 2.4.13, the corresponding bit error probability \( P_b \) is

\[ P_b = Q\left( \frac{2.012E_b}{\sqrt{4\times(1/2)N_o}} \right) = Q\left( \frac{2E_b}{\sqrt{N_o}} \right) \]  
(2.4.14)

for QPSK signals, and

\[ P_b = Q\left( \frac{0.58583E_b}{\sqrt{4\times(1/2)N_o}} \right) = Q\left( \frac{0.88E_b}{\sqrt{N_o}} \right) \]  
(2.4.15)

Figure 2.4.2 presents the variations of bit error rate (BER) with the signal/noise ratio, defined as \( E_b/N_o \), for uncoded QPSK and 8PSK signals with coherent decision threshold-level detection. These results are obtained from a number of computer simulation tests carried out at the various \( E_b/N_o \) ratios. The program used in the tests is given in Appendix B1. The accuracy of the two curves in Figure 2.4.2, over the range of bit error rates of \( 10^{-3} \) to \( 10^{-4} \), is ±0.10 dB. Appendix A5 discusses the simulation techniques and confidence limits of
the results.

As can be seen in Figure 2.4.2, compared to the QPSK system, the 8PSK system can provide a higher information transmission rate (of 3 bits per symbol interval), at the expense of increased transmitted signal power, for a given bit error rate. Take the example at BER=$10^{-4}$. The $E_b/N_0$ ratio required for QPSK system is 8.4 dB, and that for 8PSK system is 11.7 dB, corresponding to an additional transmission power of 3.3 dB. These results also follow the theoretical prediction given by equations 2.4.14 and 2.4.15. The two curves are taken to be the standards with which the performances of various coded systems are to be compared.
### Table 2.4.1: Gray coding function for QPSK signals

<table>
<thead>
<tr>
<th>$b_i$</th>
<th>Data symbol</th>
<th>Transmitted symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$b_i(1)$</td>
<td>$b_i(2)$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### Table 2.4.2: Gray coding function for 8PSK signals

<table>
<thead>
<tr>
<th>$b_i$</th>
<th>Data symbol</th>
<th>Transmitted symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$b_i(1)$</td>
<td>$b_i(2)$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Fig. 2.4.1: Gray mapping and decision thresholds for a QPSK signal
Fig. 2.4.2: Performances of uncoded QPSK and 8PSK systems with coherent threshold detection
CHAPTER THREE

CODING AND DECODING TECHNIQUES
3.1 CONVOLUTIONAL CODES

An (L, v, g) convolutional coder, over the field of binary numbers GF(2) [18], generally consists of a shift register with the storage of g binary elements, v modulo-2 adders, and a set of connections between the shift register and the adders. The 2^L-level data symbol s_i fed to the coder, at time instant t=iT, where T is the symbol interval in seconds, is generally described by the L-component vector of binary data symbols,

$$\mathbf{b_i} = [ b_i(1) \ b_i(2) \ldots \ b_i(L) ]$$

(3.1.1)

for i≥0. It is assumed that the components b_i(1), b_i(2), ... b_i(L) of the vector b_i are zero for i<0, and for i≥0, they are statistically independent binary numbers and equally likely to have either the values 0 and 1. Thus b_i carries L bits of information, for i≥0. The binary data symbols \{b_i(h)\} are shifted into and along the shift register at L bits per symbol interval.

Upon each given input vector b_i, the convolutional coder generates and outputs the v-component vector of binary code symbols

$$\mathbf{v_i} = [ v_i(1) \ v_i(2) \ldots \ v_i(v) ]$$

(3.1.2)

for i≥0, where v>L and v_i(1), v_i(2), ... v_i(v) are binary numbers with possible values 0 and 1. The redundant data is here carried by the additional (v-L) binary digits, so that each binary code symbol, v_i(h), for h=1, 2, ... v, bears L/v bits of information only. The code rate is therefore
The convolutional coder operates such that the coded vector \( v_i \) at time \( t=iT \), is a function of the current input vector \( b_i \) and one or more of the \( K \) previous vectors \( b_{i-1}, b_{i-2}, \ldots, b_{i-K} \), of the storage elements in the shift register. It is quite often (but not always true) that \( K=g/L \), where \( g \) is an integer. The memory of the code is \( g \) bits, or \( K \) symbols, the constraint length of the code being \((g+L)\) bits, or \((K+1)\) symbols.

A simple example of rate-2/3 convolutional coders, where \( L=2 \), \( v=3 \), and \( g=3 \), is shown in Figure 3.1.1. This example is also used in the following descriptions.

One method for describing a convolutional code is to give its generator matrix \([7]\). In general this matrix is semi-infinite since the input sequence is semi-infinite in length. Let the information sequence, or message, to be transmitted, be a semi-infinite sequence

\[
B = [ b_0 \ b_1 \ b_2 \ldots ] \tag{3.1.4}
\]

and when fed to the coder, it gives rise to a semi-infinite code sequence

\[
v = [ v_0 \ v_1 \ v_2 \ldots ] \tag{3.1.5}
\]

The coding equation in matrix form is
\[ V = B \begin{bmatrix} G \end{bmatrix} \] (3.1.6)

where \([G]\) is the semi-infinite generator matrix, and

\[
\begin{bmatrix} G \end{bmatrix} = \begin{bmatrix}
g_0 & g_1 & g_2 & \ldots & g_K & 0 & 0 & 0 & \ldots \\
0 & g_0 & g_1 & \ldots & g_{K-1} & g_K & 0 & 0 & \ldots \\
0 & 0 & g_0 & \ldots & g_{K-2} & g_{K-1} & g_K & 0 & \ldots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots \\
\end{bmatrix}
\] (3.1.7)

\(G_h,\) for \(0 \leq h \leq K,\) is a \((L \times \nu)\) sub-matrix, and

\[
G_h = \begin{bmatrix}
g_{h}(1,1) & g_{h}(1,2) & \ldots & g_{h}(1,\nu) \\
g_{h}(2,1) & g_{h}(2,2) & \ldots & g_{h}(2,\nu) \\
\vdots & \vdots & \ddots & \vdots \\
g_{h}(L,1) & g_{h}(L,2) & \ldots & g_{h}(L,\nu) \\
\end{bmatrix}
\] (3.1.8)

Modulo-2 addition of all components is assumed.

The \(i\)th row and \(j\)th column element of the sub-matrix \(G_h,\) for \(1 \leq i \leq L\) and \(1 \leq j \leq \nu,\) is \(g_{h}(i,j),\) where \(g_{h}(i,j)\) can have either value 0 or 1. Notice that, in general, the constraint length \((K+1)\) symbols of a convolutional code is such that \(K\) is the smallest integer for which \(g_{K}(i,j)=1,\) for some \(i\) and \(j.\) But \(g_{h}(i,j)=0,\) for all \(h>K,\) \(i\) and \(j.\)

The elements \(\{g_{h}(i,j)\},\) where \(1 \leq i \leq L\) and \(0 \leq h \leq K,\) are usually used to determine the \(j\)th sub-generator of the code. This is best illustrated with the example of rate-2/3 codes shown in Figure 3.1.1. Here, \(L=2, \ \nu=3,\) and \(K=2.\) The sub-
generator of the code is given by

\[ G^*(j) = [g_0(2,j)\ g_1(2,j)\ g_2(2,j)\ g_0(1,j)\ g_1(1,j)\ g_2(1,j)] \]  

(3.1.9)

for \( j = 1, 2 \) and 3. In fact,

\[ G^*(1) = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \]  

(3.1.10)

\[ G^*(2) = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \]  

(3.1.11)

\[ G^*(3) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \]  

(3.1.12)

In general, \( G^*(j) \) is an \( L(K+1) \)-component row vector. In the example here, \( L(K+1) = 2(2+1) = 6 \). Notice that the value of \( g \) in the example above is such that \((g+L)\) is the smallest integer for which the \((g+L)\)th component of \( G^*(j) \) in equation 3.1.9 equals to 1, for some \( i \) and \( j \). But \( g_{h,i,j} = 0 \), for all \( h>(g+L) \), \( i \) and \( j \). It is clear that \( g*LK \). This is also true for some other convolutional codes studied in this work. In the example above, \( g+L=5 \), thus \( g=3 \). But \( LK=2*2=4 \).

In general, the vector of binary code symbols, \( v_1 \), at time \( t=iT \), is determined by

\[ v_1 = \sum_{h=0}^{K} b_{i-h}g_{h} \]  

(3.1.13)

where each binary component \( v_1(j) \) of the vector \( v_1 \) is given by

\[ v_1(j) = \sum_{h=0}^{K} \sum_{l=0}^{L} b_{i-h}(l)g_{h}(l,j) \]  

(3.1.14)

for \( j=1, 2, \ldots, v \). Modulo-2 addition is assumed in both equations 3.1.13 and 3.1.14. For the coder shown in Figure 3.1.1.
\[ v_1(j) = \sum_{h=0}^{K} [b_{1-h}(1)g_{h}(1,j) + b_{1-h}(2)g_{h}(2,j)] \] (3.1.15)

for \( j = 1, 2 \) and \( 3 \).

In general, the convolutional code with a memory of \( g \) bits, or \( K \) symbols, may be viewed as a finite state machine with a specific number of states and state transitions [7]. The machine, at time \( t=iT \), rests in the state \( S_i \), which is one of its \( 2^g \) different possible states. The coder forms the corresponding coded vector \( v_1 \) and changes to its next state \( S_{i+1} \) according to the current input vector \( b_i \), where \( S_{i+1} \) is again, one of the \( 2^g \) states. The coded vector \( v_1 \) is therefore uniquely determined by the original and final states, and so by the corresponding state transition. Since, for a rate-\( \frac{L}{N} \) code, \( b_i \) has \( 2^L \) different possible values, there are the corresponding \( 2^L \) possible state transitions from each state \( S_i \). Finally, the state \( S_i \) may be generally defined as a \( g \)-component vector

\[ S_i = [s_{i}(1) \ s_{i}(2) \ ... \ s_{i}(g)] \] (3.1.16)

where \( s_{i}(1), s_{i}(2), \ldots, s_{i}(g) \) are independent binary numbers with possible values of 0 and 1, so that \( S_i \) has a total of \( 2^g \) distinct values. Note that \( s_{i}(j) \) is not to be confused with \( S_i \).

The coded vector \( v_1 \) at the output of a convolutional coder may also be represented by a tree diagram, which simply shows all possible sequences of coded vectors corresponding to \( N \) input data symbols in a \( N \)-level tree structure [7]. The tree
diagram is now described for the rate-2/3 convolutional code in Figure 3.1.1. The state of the code, at time \( t = iT \), is shown here as:

\[
S_i = [b_{i-2}(2) \ b_{i-1}(1) \ b_{i-1}(2)]
\]  

(3.1.17)

Supposing the coder rests in the state \( S_0 \), at time \( t=0 \), as shown in Figure 3.1.2, this is assumed to be an all zero state before the start of transmission. The vertical axis in Figure 3.1.2 shows the possible combinations of \( b_0, b_1, ..., b_{N-1} \), for \( N=2 \), whereas the horizontal axis denotes time \( t \), measured as an integer multiple of \( T \) seconds. Each node in the diagram has one predecessor, except the root node at time \( t=0 \), and four \( (2^L) \) branches are extended from each node. For \( N=2 \), 16 \( (2^{LN}) \) branches result from the corresponding 16 different possible combinations of the vectors \( b_0 \) and \( b_1 \). The code sequence \( \{v_i\} \), uniquely determined by any given combination of \( b_0 \) and \( b_1 \), must be one of the 16 paths shown on the tree diagram.

As can be seen from Figure 3.1.2, each node in the tree, in fact, corresponds to a state of the coder. For a sequence of \( N \) input vectors \( \{b_i\} \), or \( LN \) data bits, there are \( 2^{LN} \) terminal nodes in the \( N \)-level tree, but there are only \( 2^L \) different possible states. Clearly, when \( LN > 2^L \), the number of nodes in the tree diagram exceeds the number of different possible states of the code. Several nodes must therefore correspond to the same state. After the input data symbols have been identical for \( K \) consecutive symbols, where \( K \) is the code memory in symbols, the coded vectors \( \{v_i\} \) must now be the
same. The path is said to remerge. The redrawing of the tree diagram with the merging paths is called a trellis diagram. The graph in Figure 3.1.2 can therefore be reduced to that given in Figure 3.1.3.

In Figure 3.1.3, the vertical axis denotes the possible state of the code, whereas the horizontal axis denotes time \( t \), measured as an integer multiple of \( T \) seconds. The nodes are drawn in a rectangular grid and the number of nodes in each column is fixed, representing all different possible states of the code. The configuration of the branches connecting each column of nodes to the next column of nodes at the right is the same for every column, showing the fact that each state is determined only by the corresponding previous state and the current vector of binary data symbols. In this particular example, the four branches leaving each node correspond to \( b_1 = [00], [01], [10] \) and \( [11] \), respectively.

The trellis representation of convolutional codes uses the fact that the coder is a finite state machine, and therefore it gives a more constructive way of looking at the codes. Because of the trellis structure of convolutional codes, they are also known as a class of "trellis codes".

The minimum distance properties of convolutional codes play a very important role in determining the error-correction capability of the codes [7,16-18]. Two minimum distance parameters, each has its own importance, are now defined for rate-\( L/v \) binary convolutional codes. Let

\[
B = [ b_0 \ b_1 \ b_2 \ \ldots ]
\]  

(3.1.18)
and \[ B' = [ b'_0 \ b'_1 \ b'_2 \ ... ] \] (3.1.19)

be any two possible messages that differ in their first vectors (data symbols) \( \{b_0\} \), and let

\[
V = [ v_0 \ v_1 \ v_2 \ ... ] \quad (3.1.20)
\]

\[
V' = [ v'_0 \ v'_1 \ v'_2 \ ... ] \quad (3.1.21)
\]

be the corresponding code sequences of the messages \( B \) and \( B' \), respectively. Now, consider

\[
V(l) = [ v_0 \ v_1 \ ... \ v_{l-1} ] \quad (3.1.22)
\]

and \( V'(l) = [ v'_0 \ v'_1 \ ... \ v'_{l-1} ] \quad (3.1.23) \)

where \( V(l) \) and \( V'(l) \) take on the first \( l \) components of the code sequences \( V \) and \( V' \), respectively.

The Hamming distance between the two binary-valued code sequences \( V(l) \) and \( V'(l) \) is denoted by \( d_H[V(l),V'(l)] \), and it is defined as the total number of positions (binary digits) where \( V(l) \) and \( V'(l) \) differ.

The \( l^{th} \) column distance function \( \eta_l \) of the code is the minimum Hamming distance between all pairs of code sequences of length \( l \) symbols that differ in their first coded vectors \( \{v_0\} \). This is given by

\[
\eta_l = \min \{ d_H[V(l),V'(l)] \} \quad (3.1.24)
\]

The most important distance parameter for binary convolutional codes is the minimum free Hamming distance,
\[ d_H(\text{free}) = \lim_{L \to \infty} \{ \eta_L \} \]
\[ = \min \{ d_H(V, V') \} \quad (3.1.25) \]

The minimisation is carried out over all valid code sequences \( \{V\} \) and \( \{V'\} \) of arbitrary length that differ in their first code symbols. The second most important distance parameter of the convolutional codes is the distance profile \( \Omega \), where \( \Omega \) is the \((g+L)\)-tuple column distance function

\[ \Omega = \{ \eta_1, \eta_2, \ldots, \eta_{g+L} \} \quad (3.1.26) \]

\( \eta_L \), for \( L=1, 2, \ldots, (g+L) \), is defined by equation 3.1.24. \( g \) is the code memory in bits. Clearly, the distance profile of the code is a measure of the rate of the growth in the column distance function.

Most convolutional codes published so far have been designed to achieve the optimum minimum free Hamming distances for the given code rates and constraint lengths. The examples of such codes are given in references 22 to 24. The codes with the best Hamming distance profiles are also quoted in the published literature [25-26]. These codes are normally obtained by employing exhaustive computerised searches.

However, the use of conventional binary convolutional codes requires the transmission of redundant data bits. Consequently, the actual transmission speed of the system must be increased by a factor of \( \nu/L \), for the given information rate, leading to the corresponding expansion in the signal.
bandwidth. Otherwise, the information transmission rate of the system would have to be reduced by a factor of \( \frac{L}{v} \). Furthermore, hard-quantised decoding normally causes the loss of information for about 3.0 dB.

In the next section, the approach of combining the convolutional coding with MPSK modulation is described. Such schemes have been proven to achieve a considerable gain in tolerance to noise without, however, reducing the information rate or increasing the signal bandwidth. Soft-decision decoding is also made possible with such coded signals.
Fig. 3.1.1: A rate-2/3, 8-state, convolutional coder
Fig. 3.1.2: Tree representation of the coder in Fig. 3.1.1
Fig. 3.1.3: Trellis representation of the coder in Fig. 3.1.1
3.2 CONVOLUTIONAL CODING WITH MULTI-PHASE SIGNALS

This part of investigation concerns the theoretical study of a class of well known convolutional codes, initially proposed by G. Ungerboeck in 1976 [33]. These codes are commonly known as the convolutional codes with expanded signal sets, or else known as the alphabet redundant codes, characterised by the fact that the redundancy required by coding is provided by additional signal levels rather than by transmitting additional signal elements. It has been shown that a useful improvement in tolerance to noise of several dB can be achieved in this way without, however, actually reducing the information rate or increasing the bandwidth of the transmitted signal. These codes are particularly attractive for applications over power- and band-limited satellite links.

The approach combines binary convolutional coding with a specific mapping of binary code symbols onto a set of multi-level signals. The constellation of the latter signal is determined by the particular modulation format considered. The modulation scheme of primarily concern in this work is the MPSK scheme of Section 2.3, for M=8 and 16.

The complete coding and mapping process is shown in Figure 3.2.1, where the vector of binary code symbols

\[ v_1 = [ v_1(1) \ v_1(2) \ ... \ v_1(n) ] \]

(3.2.1)

at the output of the rate-L/v convolutional coder and time t=iT, is fed to a specific mapping process, whose output is the corresponding complex-valued signal symbol q_1. The mapping is
one-to-one, so that \( q_i \) also has \( 2^\nu \) (and \( M=2^\nu \)) different possible values. The coded vectors \( \{v_i\} \) are mapped onto the complex number plane according to a given mapping function \( M_1(v_i) \), as follows. For a given \( v_i \), the function \( M_1(v_i) \) is an integer in the range 0 to \( 2^\nu-1 \), so that

\[
z_i = M_1(v_i) \quad (3.2.2)
\]

Unless otherwise stated, the function \( M_1(.) \) is defined as

\[
z_i = 2^{\nu-1}v_i(1) + 2^{\nu-2}v_i(2) + \ldots + 2^0v_i(\nu) \quad (3.2.3)
\]

The phase angle of the corresponding transmitted symbol \( q_i \), in the complex number plane, is determined by

\[
\phi_i = 2\pi z_i/M \quad \text{(radians)} \quad (3.2.4)
\]

for \( M=2^\nu \). Now,

\[
q_i = \cos\phi_i + j\sin\phi_i \quad (3.2.5)
\]

where \( j=\sqrt{-1} \).

Following equation 2.1.11, the sample of the complex-valued demodulated baseband signal \( r(t) \), at time \( t=iT \), is

\[
p_i = q_i + w_i \quad (3.2.6)
\]

where the real and imaginary parts of the noise components \( \{w_i\} \) are sample values of statistically independent Gaussian random
variables with zero mean and fixed variance $\sigma^2$.

The minimum free unitary distance squared, $d_{\text{free}}^2$, of the convolutionally coded signal is now defined. Let $\{q_1\}$ and $\{q'_1\}$ be any two possible (valid) sequences of complex-valued code symbols of arbitrary length that differ in their first symbols, then

$$d_{\text{free}}^2 = \min \left\{ \sum_{i=0}^{\infty} |q_i - q'_i|^2 \right\}$$

(3.2.7)

The minimisation is carried out over all valid coded signal sequences. It can be shown (see Appendix A3) that, in an additive white Gaussian noise (AWGN) channel, when all different possible messages at the transmitter are equally likely, as is assumed here, and when a soft-decision maximum likelihood decoder is incorporated at the receiver, the error-event probability for the decoded data symbols, as the signal/noise ratio increases, approaches a value which is determined by the minimum free unitary distance between the different possible coded signal sequences. In fact, the bit error probability of the given coded system is bounded (see Appendix A3) by

$$P_b > N_1 Q\left( d_{\text{free}}/2\sigma \right)$$

(3.2.8)

$N_1$ is the average number of bit errors associated with an incorrect selection of the decoded data sequence at the distance $d_{\text{free}}$, and $\sigma^2$ is the variance of real and imaginary parts of $\{w_j\}$. The asymptotic gain in tolerance to noise, achieved by this system, over the corresponding uncoded system,
is given by

\[ G_c = 10 \log_{10} \left( \frac{d^2_{\text{free}}}{d^2_{\text{uncoded}}} \right) \text{ (dB)} \]  

(3.2.9)

where \( d_{\text{uncoded}} \) is the minimum free unitary distance for the corresponding uncoded system.

Clearly, the conventional binary convolutional codes with the optimum minimum free Hamming distance (such as those presented in references 22 to 24) may not necessarily correspond to the optimum minimum free unitary distance in the complex number plane. The codes therefore need be redesigned.

Before proceeding to the design of convolutional codes with expanded signal sets, it is appropriate here to examine the potential gains available in terms of channel capacity. The model of data transmission system is described in section 2.1, and a soft-decision maximum likelihood detector is assumed at the receiver.

The well known channel capacity equation for a memoryless AWGN channel with a discrete input and a continuous-valued output, as the channel model assumed here, may be given as [33, 39]

\[ C^* = \max h=0 \sum_{k=0}^{M-1} P(k) f(r/q(k)) \log_2 \left( \frac{f(r/q(h))}{\sum_{k=0}^{M-1} P(k) f(r/q(k))} \right) dr \]

(3.2.10)

where

\[ C^* = \text{number of information bits per signal interval} \]  

(bits/T), or equivalently, the number of bits per transmitted signal element.
M = number of discrete signal elements at the input of the channel.

q(h) = a possible value of q₁, for h=0, 1, ..., (M-1).

r = continuous-variable received sample.

f(r/q(h)) = conditional probability density function of r, for a given value of q(h).

P(h) = probability that code symbol q(h) is transmitted.

In an AWGN channel,

\[ f(r/q(h)) = \frac{1}{2\pi\sigma^2} \exp\left( -\frac{|r - q(h)|^2}{2\sigma^2} \right) \]  \hspace{1cm} (3.2.11)

bearing in mind that both r and q(h) are complex-valued.

If all code symbols \( \{q_i\} \) are equally likely to have any of the M different possible values, the maximisation over \( \{P(h)\} \) can be omitted. Now,

\[ C^* = \log_2 M - \sum_{h=0}^{M-1} \sum_{k=0}^{M-1} \text{E}\{\log_2 \exp(-\frac{|q(h)+w-q(k)|^2-|w|^2}{2\sigma^2})\} \]  \hspace{1cm} (3.2.12)

where the integration is replaced by the expectation over the Gaussian distributed complex-valued noise variable w [33,39]. The deviation of equation 3.2.12 is given in Appendix A4, which also includes a mathematical treatment of the Shannon’s channel capacity concept.

In Figure 3.2.2, \( C^* \) (bits/T) is plotted against the signal/noise ratio, defined here as \( E_s/N_0 \), for different MPSK signals [33,39]. The value of \( E_s/N_0 \) at which a symbol error rate of \( 10^{-5} \) is achieved for uncoded transmission is also
indicated (marked as a small circle on each curve). $E_s$ is the average energy per signal element and $(1/2)N_0$ is the two-sided power spectral density of the white Gaussian noise. The variation of $C^*$ with $E_s/N_0$ is also shown for the Shannon limit $\log_2(1+E_s/N_0)$ (see Appendix A4), where an infinitely large signal set is assumed. The channel capacity $C^*$ (bits/T) represents the amount of information per symbol interval which can be transmitted over the given channel without any error, as a net result of extensive coding and decoding employed. $C^*$ is also known as the information transmission rate, or simply the information rate.

It can be seen from Figure 3.2.2 that the 8PSK system theoretically achieves an information rate of 3 bits/T at a sufficient large $E_s/N_0$. For instance, a symbol error rate of $10^{-5}$ is achieved at $E_s/N_0=18.2$ dB, as can be seen in Figure 3.2.2. But below 12 dB, $C^*$ decreases rapidly, and at $E_s/N_0=5.7$ dB, $C^*$ falls to 2 bits/T. On the other hand, the QPSK system achieves the symbol error rate of $10^{-5}$ only if $E_s/N_0>13.1$ dB. The evidence here suggests that, instead of employing a QPSK signal to transmit two information bits per symbol interval at $E_s/N_0=13.1$ dB, the same error rate can be achieved by employing an 8PSK signal to carry the two information bits together with one redundant bit (that is generated by FEC coding). In the latter case, however, $E_s/N_0=5.7$ dB is sufficient. It follows that, while maintaining the information rate of 2 bits/T, a gain in tolerance to noise of up to 7.4 dB may be obtained in this way, at a BER of $10^{-5}$, as a net result of extensive coding and decoding.
The second interesting point from Figure 3.2.2 is that almost all that is to be gained in terms of achievable channel capacity could be obtained by doubling the total number of possible transmitted symbol values, relative to that for the uncoded system. Any further increase in the number of signal levels will only yield a small (if any) additional gain in tolerance to noise. Take the example at the information rate of 2 bits/T, in Figure 3.2.2. The step from QPSK to 8PSK signals can yield a theoretical advantage of 7.4 dB, whereas an additional gain of only 1.2 dB may be achieved by going to an infinitely large signal set. The conclusion here is that, the majority of available gains obtained by going from QPSK to 8PSK signals can be achieved with a corresponding rate-2/3 convolutional code.

The other interesting application is the rate-3/4 convolutional code that operates on an 8-level data signal to generate a coded 16PSK signal. The latter signalling provides a theoretical information rate of 3 bits/T. From Figure 3.2.2, the step from 8PSK to 16PSK signals yields a theoretical advantage of 7.0 dB, while an additional gain of 3.2 dB may be achieved by going to an infinitely large signal set.

The first step in the design procedure is to determine the code structure that ensures a good minimum free unitary distance. The technique of "mapping by set partitioning", introduced by G. Ungerboeck [33,39], shows how the transmitted symbols \(q_i\), which are uniquely determined by the vectors \(v_i\), should be assigned to the state transitions in achieving the best minimum distance. The codes of primary concern in
this work are those with the code rate \( \frac{L}{L+1} \), for \( L=2 \) and 3. The set partitioning of an 8PSK signal is shown in Figure 3.2.3, and that for a 16PSK signal is given in Figure 3.2.4.

Generally speaking, the \( 2^{L+1} \) different possible signals \( \{q_i\} \) (or vectors \( \{v_i\} \)) are partitioned into successively smaller sets of size \( 2^L \), ..., 8, 4, and 2, with a steady increase in minimum unitary distance between the signals within each set. These sets are known as subsets, and the size of a subset is, of course, the number of different signals in that set. From Figures 3.2.3 and 3.2.4 it is clear that the first two subsets have a size of \( 2^L \), the following four subsets have a size of \( 2^{L-1} \), and so on. Furthermore, the minimum distance between any two signals in each of the two first subsets is equal to that of the corresponding uncoded system.

A signal \( q_i \) or a vector \( v_i \) is assigned to each state transition according to the following three coding rules.

Coding Rule (1): Any state transitions originating from, or terminating into, the same states, are assigned signals from the same subset of size \( 2^L \) (which is one of the first two subsets). Note that the number of different possible state transitions from any given state \( S_i \) is \( 2^L \).

Coding Rule (2): Any transitions having the same originating and terminating states (known as parallel transitions) are assigned signals from the same subset of size \( 2^{L-1} \) (which is one of the following four subsets).

Coding Rule (3): All signals should appear with equal frequency, and there should be a fair degree of symmetry in the states and state transitions.

Following Coding Rule (1), each of the \( 2^L \) state trans-
itions originating from (or terminating into) the same state is
assigned one of the $2^L$ signals from either of the first two
subsets. This ensures good distance properties for the portion
of two sequences of the coded signals $\{q_i\}$, that starts with
the symbol at which the two sequences diverge and ends with the
symbol at which they converge. In fact, this ensures a minimum
free unitary distance squared of at least twice that of the
uncoded system, and thus provides an asymptotic coding gain of
at least 3 dB, so long as the other two coding rules are
appropriately followed.

The set partitioning technique is now described for
coded 8PSK signals. As can be seen from Figure 3.2.3, the
complete signal set $A(0)$ of an 8PSK signal consists of eight
elements,

$$A(0) = \{ q(0), q(1), ..., q(7) \} \quad (3.2.13)$$

where $q(h) = \cos(2\pi h/M) + j\sin(2\pi h/M)$

and $q(h)$ represents a possible value of the transmitted symbol
$q_i$, for $h=0, 1, ..., 7$. The minimum unitary (or Euclidean)
distance squared between any two signals in the set $A(0)$ is

$$\Gamma_0 = 0.5858 \quad (3.2.15)$$

Following the technique of "set partitioning", $A(0)$ is
partitioned into two subsets $B(0)$ and $B(1)$, each having four
elements, as such that
\[ B(0) = \{ q(0), q(2), q(4), q(6) \} \]  \hspace{1cm} (3.2.16)

and \[ B(1) = \{ q(1), q(3), q(5), q(7) \} \]  \hspace{1cm} (3.2.17)

respectively. The minimum unitary distance squared between any two signals in each subset here is

\[ \Gamma_1 = 2.0 \]  \hspace{1cm} (3.2.18)

Each of the two subsets \( B(0) \) and \( B(1) \) is further partitioned into the two subsets of two signal elements. These are

\[ C(0) = \{ q(0), q(4) \}, \quad C(2) = \{ q(2), q(6) \} \]  \hspace{1cm} (3.2.19)

for subset \( B(0) \), and

\[ C(1) = \{ q(1), q(5) \}, \quad C(3) = \{ q(3), q(7) \} \]  \hspace{1cm} (3.2.20)

for subset \( B(1) \). Clearly, the minimum unitary distance squared between the two signals in each of these four subsets is

\[ \Gamma_2 = 4.0 \]  \hspace{1cm} (3.2.21)

The design procedure for rate-2/3 convolutional coded 8PSK signals is now described with the example of a 4-state code. The trellis diagram of the code is shown in Figure 3.2.5, where the state of the coder, \( S_1 \), is a 2-component vector

\[ S_1 = [ b_{1-2}(2) \ b_{1-1}(2) ] \]  \hspace{1cm} (3.2.22)
Following Coding Rule (1), each of four transitions from the state [00] and the state [10] is assigned one of four signals from the subset B(0), whereas each of four transitions from the state [01] and the state [11] is assigned one of four signals from subset B(1). Coding Rule (2) only applies to parallel state transitions. Take the example in Figure 3.2.5, where the transition from \( S_i=[00] \) to \( S_{i+1}=[00] \) with input \( b_i=[00] \), and that from \( S_i=[00] \) to \( S_{i+1}=[00] \) with \( b_i=[10] \), are clearly in parallel. Applying Coding Rule (2), these two transitions are assigned the signals \( q(0) \) and \( q(4) \), respectively. Notice that \( q(0) \) and \( q(4) \) are the two elements in subset \( C(0) \), which is one of the second subsets.

Now, take the all zero path, where \( q_i=q(0) \) for \( i \geq h \), as the reference path. Correspondingly, this all zero path begins with the zero state [00], at time \( t=hT \), and is generated by the all zero input sequence in which \( b_i=[00] \) for each \( i \). As can be seen in Figure 3.2.5, from this reference path, the shortest path starting with \( S_h=[00] \) must be that generated by the input sequence in which \( b_h=[10] \) and \( b_i=[00] \) for \( i \geq h \). The unitary distance squared between this path and the all zero path must be given by

\[
\begin{align*} 
  d^2 &= |q(4) - q(0)|^2 \\
  &= r_2 \\
  &= 4.0
\end{align*}
\]

(3.2.23)

From the all zero path, the second shortest path starting with \( S_h=[00] \) must be that generated by the input sequence in which
and $b_i = [00]$ for $i > h$. From Figure 3.2.5, the unitary distance squared between this path and the all zero path is

$$d^2 = |q(2) - q(0)|^2 + |q(1) - q(0)|^2 + |q(2) - q(0)|^2$$

$$= \Gamma_1 + \Gamma_0 + \Gamma_1$$

$$= 2.0 + 0.5858 + 2.0$$

$$= 4.5858$$  \hspace{1cm} (3.2.24)$$

So long as Coding Rule (3) is properly followed, equation 3.2.23 must also give the minimum free unitary distance between all possible (valid) coded signal sequences that differ in their first symbols.

Figure 3.2.6 shows the trellis diagram of a rate-2/3, 8-state, convolutional code for 8PSK signals. The state of the coder, at time $t = iT$, is defined by

$$S_i = [ b_{i-2}(2) \ b_{i-1}(1) \ b_{i-1}(2) ]$$  \hspace{1cm} (3.2.25)$$

As a net result of similar signal assignments as those for the 4-state code in Figure 3.2.5, both transitions from the state $[000]$ to $[001]$ and from the state $[100]$ to $[000]$ receive the signal $q(2)$. The transition in between, which is that from the state $[001]$ to $[100]$, receives the signal $q(1)$. The unitary distance squared between this path (which is also the shortest path of all), and the all zero path, is

$$d^2 = |q(2) - q(0)|^2 + |q(1) - q(0)|^2 + |q(2) - q(0)|^2$$

$$= \Gamma_1 + \Gamma_0 + \Gamma_1$$

$$= 2.0 + 0.5858 + 2.0$$
Since no parallel transitions are involved, Coding Rule (2) need not be considered. Thus, so long as Coding Rule (3) is properly followed, equation 3.2.26 must also give the minimum free unitary distance between all possible (valid) coded signal sequences that differ in their first symbols.

For those with relatively short constraint lengths, the design of codes may be carried out by hand, as demonstrated above. Otherwise, computer aided searches of the best codes (see reference 39) must be undertaken. Table 3.2.1 presents five best known rate-2/3 convolutional codes for 8PSK signals which have been quoted in the published literature [33-40]. These codes are designed to achieve the optimum minimum free unitary distances for the given code constraint lengths. The relative asymptotic gain of each code over uncoded QPSK signals is also shown. In table 3.2.1, the sub-generators of each code are given by

\[ g^*(j) = [g_0(2,j) \ g_1(2,j) \ g_{k_2(2,j)} \ g_0(1,j) \ g_1(1,j) \ g_{k_1(1,j)}] \]

(3.2.27)

for \( j = 1, 2 \) and 3, where \( g = k_1 + k_2 \). \( g \) is the memory of the code in bits. The corresponding binary code symbols \( \{v_1(j)\} \) are determined by equation 3.2.28.

\[
v_1(j) = \sum_{h=0}^{k_1} b_{1-h}(1) g_h(1,j) + \sum_{h=0}^{k_2} b_{1-h}(2) g_h(2,j)
\]

(3.2.28)

for \( j = 1, 2 \) and 3, where modulo-2 addition is assumed. The
mapping function $M_1(v_1)$ for 8PSK signals is defined to be

$$z_1 = 4v_1(1) + 2v_1(2) + v_1(3) \quad (3.2.29)$$

and now,

$$q_1 = \cos(2\pi z_1/M) + j\sin(2\pi z_1/M) \quad (3.2.30)$$

The same design principle can also be applied to rate-3/4 convolutional coded 16PSK signals. The coder structure considered is shown in Figure 3.2.7. Such a coder offers a maximum asymptotic gain of $10\log_{10}(2.0/0.5858) = 5.3$ dB, and includes all short codes of practical interest.

The arrangement in Figure 3.2.7 has the important advantage that the maximum likelihood Viterbi decoder (see Section 3.3 for details of the decoder) operates on a binary signal. The simple form of the decoder results from the fact that only one of the three binary components in the input vector $b_1$ ($b_1(3)$ in Figure 3.2.7) is convolutionally coded in forming the coded vector $v_1$ and the transmitted symbol $q_1$. This point is further discussed in the appropriate part of Section 3.3. The two non-convolutionally coded bits $b_1(1)$ and $b_1(2)$ are Gray-coded, as can be seen in Figure 3.2.8, which also includes the mapping function $M_1(v_1)$ for coded 16PSK signals.

Table 3.2.2 presents four best known rate-3/4 codes for 16PSK signals [41-43] together with their asymptotic coding gains over the uncoded 8PSK signals. In Table 3.2.2, the sub-generators of each code are given by

$$G^*(j) = [ \xi_0(j) \xi_1(j) \ldots \xi_g(j) ] \quad (3.2.31)$$
for \( j = 3 \) and \( 4 \). The third and fourth components of the coded vector \( v_i \) are determined by equation 3.2.32.

\[
v_i(j) = \sum_{h=0}^{g} b_{i-h}(3) g_h(3,j)
\]

for \( j = 3 \) and \( 4 \), respectively, where modulo-2 addition is assumed. The first two components \( v_i(1) \) and \( v_i(2) \) of the vector \( v_i \) are taken to be the corresponding values of \( b_i(1) \) and \( b_i(2) \), respectively, as shown in Figure 3.2.7.

In the next section, the soft-decision maximum likelihood decoding, achieved by the well known Viterbi decoding algorithm, is described for rate-2/3 convolutionally coded 8PSK signals. The performance results of various coded 8PSK and 16PSK signals using the Viterbi decoding are also presented. The theorem of maximum likelihood decoding is described in Appendix A2, and Appendix A3 examines the asymptotic performances of various coded signals with such a decoder.
<table>
<thead>
<tr>
<th>Code</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>k2</td>
<td>2</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>g=k1+k2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Number of states</td>
<td>4</td>
<td>8</td>
<td>16</td>
<td>32</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>Code memory in bits.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Code sub-generators</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>G*(1)</td>
<td>[000 1]</td>
<td>[100 0]</td>
<td>[100 11]</td>
<td>[1000 001]</td>
<td>[1110 1011]</td>
<td></td>
</tr>
<tr>
<td>G*(2)</td>
<td>[101 0]</td>
<td>[001 01]</td>
<td>[001 100]</td>
<td>[0011 1010]</td>
<td>[0001 1010]</td>
<td></td>
</tr>
<tr>
<td>G*(3)</td>
<td>[010 0]</td>
<td>[010 00]</td>
<td>[010 000]</td>
<td>[0100 000]</td>
<td>[0000 0110]</td>
<td></td>
</tr>
<tr>
<td>G*(4)</td>
<td>[010 0]</td>
<td>[010 00]</td>
<td>[010 000]</td>
<td>[0100 000]</td>
<td>[0000 0110]</td>
<td></td>
</tr>
<tr>
<td>G*(5)</td>
<td>[010 0]</td>
<td>[010 00]</td>
<td>[010 000]</td>
<td>[0100 000]</td>
<td>[0000 0110]</td>
<td></td>
</tr>
<tr>
<td>G*(6)</td>
<td>[010 0]</td>
<td>[010 00]</td>
<td>[010 000]</td>
<td>[0100 000]</td>
<td>[0000 0110]</td>
<td></td>
</tr>
<tr>
<td>G*(7)</td>
<td>[010 0]</td>
<td>[010 00]</td>
<td>[010 000]</td>
<td>[0100 000]</td>
<td>[0000 0110]</td>
<td></td>
</tr>
<tr>
<td>Corresponding d2 free and Gc (dB) over QPSK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>d2 free</td>
<td>4.0</td>
<td>4.5858</td>
<td>5.1716</td>
<td>5.7674</td>
<td>6.3432</td>
<td></td>
</tr>
<tr>
<td>Gc</td>
<td>3.0</td>
<td>3.6</td>
<td>4.1</td>
<td>4.6</td>
<td>5.0</td>
<td></td>
</tr>
</tbody>
</table>
Table 3.2.2: Best known rate-3/4 convolutional codes for 16PSK signals

<table>
<thead>
<tr>
<th>Code</th>
<th>Code memory in bits</th>
<th>Number of states</th>
<th>Code sub-generators</th>
<th>Corresponding $d_2^{\text{free}}$ and $G_c$ (dB) over 8PSK</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC(6)</td>
<td>2</td>
<td>4</td>
<td>$G^*(3) = [101]$</td>
<td>$d_2^{\text{free}} = 1.3238$ $G_c = 3.6$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$G^*(4) = [010]$</td>
<td></td>
</tr>
<tr>
<td>UC(7)</td>
<td>3</td>
<td>8</td>
<td>$G^*(3) = [1011]$</td>
<td>$d_2^{\text{free}} = 1.7640$ $G_c = 4.0$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$G^*(4) = [0100]$</td>
<td></td>
</tr>
<tr>
<td>UC(8)</td>
<td>4</td>
<td>16</td>
<td>$G^*(3) = [11001]$</td>
<td>$d_2^{\text{free}} = 1.6282$ $G_c = 4.4$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$G^*(4) = [00100]$</td>
<td></td>
</tr>
<tr>
<td>UC(9)</td>
<td>5</td>
<td>32</td>
<td>$G^*(3) = [101001]$</td>
<td>$d_2^{\text{free}} = 1.7804$ $G_c = 4.8$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>$G^*(4) = [010010]$</td>
<td></td>
</tr>
</tbody>
</table>
Fig. 3.2.1: Complete coding process

\[ q_i = \cos(2\pi z_i/M) + j\sin(2\pi z_i/M) \]
Fig. 3.2.2: Variation of channel capacity with Es/No

$$\log_2 (1 + \text{Es/No})$$

capacity (bits/s/Hz)

Es/No (dB)
Fig. 3.2.3: Set partitioning of an 8PSK signal
Fig. 3.2.4: Set partitioning of a 16PSK signal
Fig. 3.2.5: Trellis representation of a rate-$2/3$, 4-state, convolutional coder

\[ S_i = \begin{bmatrix} b_{x_i}(2) & b_{y_i}(2) \end{bmatrix} \]

\[
\begin{bmatrix}
0 & 0 \\
0 & 1 \\
1 & 0 \\
1 & 1 \\
\end{bmatrix}
\]
Fig. 3.2.6: Trellis representation of a rate-2/3, 8-state, convolutional coder

\[ S_i = \begin{bmatrix} b(2) & b(1) & b(2) \end{bmatrix} \]

\[
\begin{array}{c}
[0 0 0] \\
[0 0 1] \\
[0 1 0] \\
[0 1 1] \\
[1 0 0] \\
[1 0 1] \\
[1 1 0] \\
[1 1 1]
\end{array}
\]
Fig. 3.2.7: A rate-3/4 convolutional coder
Fig. 3.2.8: 16PSK signal constellation and mapping
3.3 VITERBI-ALGORITHM DECODER

In this section, the Viterbi-algorithm decoding scheme is described for the rate-2/3 convolutionally coded 8PSK signals of Section 3.2. This decoding algorithm, invented by Viterbi, A.J. in 1967 [79], utilises the structural and distance properties of convolutional codes systematically, and it selects as the decoded message, the possible sequence of data symbols that has the maximum likelihood function conditioned upon the received signal. The theorem of maximum likelihood decoding is presented in Appendix A2.

The description of the Viterbi algorithm begins with a description of the received signal. The decoder is then described in terms of its output, a set of stored vectors and costs. The algorithm repeated during each symbol interval, which uses the stored values to generate the decoded data symbol $s'_1$, is also described. The assumed model of the data transmission system is described in Section 2.1. The Viterbi decoding of other trellis coded signals will follow the description given here.

From equation 2.1.11, the sample of the demodulated baseband signal $r(t)$, at the output of the demodulator and time $t=iT$, is

$$r_i = q_i + w_i$$  \hspace{1cm} (3.3.1)

$q_i$ is the complex-valued code symbol uniquely determined by the the coded vector $v_i$, at the output of the coder (see Figure 3.2.1), and $w_i$ is a sample value of the Gaussian noise waveform $w(t)$ at the demodulator output. $T$ is the symbol
interval in seconds. The real and imaginary parts of \( \{w_i\} \) are statistically independent Gaussian random variables with zero mean and fixed variance \( \sigma^2 \).

The decoder, as shown in Figure 2.1.1, operates on the sequence of received samples \( \{r_i\} \). The corresponding output sequence from the decoder comprises the 2-component vectors

\[
b'_i = [ b'_i(1) \ b'_i(2) ]
\]  

(3.3.2)

where \( b'_i(h) \) is the decoded binary data symbol, and \( b'_i(h)=0 \) or 1, for \( h = 1 \) and 2. In the absence of noise, \( b'_i=b_i \), for each \( i \), where \( b_i \) is the vector of binary data symbols being fed to the convolutional coder at time \( t=iT \), and

\[
b_i = [ b_i(1) \ b_i(2) ]
\]  

(3.3.3)

The vector \( b_i \) uniquely determines the corresponding 4-level data symbol \( s_i \), whose possible values are 0, 1, 2 and 3, respectively. Similarly, the vector \( b'_i \) at the decoder output uniquely determines the decoded data symbol \( s'_i \), the relationship between \( s'_i \) and \( b'_i \) being the same as that between \( s_i \) and \( b_i \).

Now, consider a possible sequence of \( \{b_i\} \),

\[
x^* = [ x_0 \ x_1 \ \ldots \ x_{i-1} \ x_i ]
\]  

(3.3.4)

where \( x_i \) represents a possible value of \( b_i \), for \( i \geq 0 \). Thus \( x_i \) is also a 2-component vector.
where \( x_1(1) \) and \( x_1(2) \) are possible values of the components \( b_1(1) \) and \( b_1(2) \), respectively, of the vector \( b_1 \). Clearly, \( x_1 \) can have any of the four values, 00, 01, 10 and 11.

Associated with any sequence \( X^* \) is the corresponding sequence of complex-valued code symbols \( \{p_i\} \), where \( p_1 \) takes on a possible value of \( q_1 \). From equations 3.2.27 to 3.2.30, \( p_1 \) may be determined as follows.

\[
\begin{align*}
0 \\
3.3.5
\end{align*}
\]

\[
\begin{align*}
x_1 = [ x_1(1), x_1(2) ] \\
\text{where } x_1(1) \text{ and } x_1(2) \text{ are possible values of the components } b_1(1) \text{ and } b_1(2), \text{ respectively, of the vector } b_1. \text{ Clearly, } x_1 \text{ can have any of the four values, 00, 01, 10 and 11.}
\end{align*}
\]

\[
\begin{align*}
\text{Associated with any sequence } X^* \text{ is the corresponding sequence of complex-valued code symbols } \{p_i\}, \text{ where } p_1 \text{ takes on a possible value of } q_1. \text{ From equations 3.2.27 to 3.2.30, } p_1 \text{ may be determined as follows.}
\end{align*}
\]

\[
\begin{align*}
\begin{align*}
k1 \\
k2
\end{align*}
\end{align*}
\]

\[
\begin{align*}
\nu_1(j) = \sum_{h=0}^{k1} x_{1-h}(1) g_{h}(1,j) + \sum_{h=0}^{k2} x_{1-h}(2) g_{h}(2,j) \quad (3.3.6)
\end{align*}
\]

\[
\begin{align*}
\text{for } j=1, 2 \text{ and } 3, \text{ where } x_1(k)=0, \text{ for } i<0, k=1 \text{ and } 2.
\end{align*}
\]

\[
\begin{align*}
x_1 = 4v_1(1) + 2v_1(2) + v_1(3) \quad (3.3.7)
\end{align*}
\]

\[
\begin{align*}
\nu_1 = \cos(\pi z_1/4) + jsin(\pi z_1/4) \quad (3.3.8)
\end{align*}
\]

where \( j=\sqrt{-1} \). Modulo-2 addition is assumed in equation 3.3.6.

The \( \{g_{h}(i,j)\} \) together determine three sub-generators of the corresponding code, and \( g \) (where \( g=k1+k2 \)) is the memory of the code in bits.

The unitary distance squared between the sequence of received samples \( \{r_1\} \) and the corresponding code symbols \( \{p_1\} \) of a given sequence \( X^* \), is determined by the cost

\[
\begin{align*}
c_1 &= \frac{1}{\sum_{h=0}^{1} |r_h - p_h|^2} \\
&= c_{i-1} + |r_1 - p_1|^2
\end{align*}
\]
of the particular sequence $x^*$. $\text{Re}(.)$ and $\text{Im}(.)$ represent here the real and imaginary parts of the corresponding quantity.

For any rate-2/3 convolutional code with a memory of $g$ bits (where $g=k_1+k_2$) in Table 3.2.1, and just prior to the receipt of the sample $r_1$, the corresponding Viterbi algorithm decoder at the receiver holds in store a set of $2^g$ $n$-component vectors $(X_{i-1})$ at the receiver. Each $X_{i-1}$ is formed by the last $(n+1)$ components of the corresponding $(i+1)$-component sequence $x^*$ in equation 3.3.4, with $x_1$ omitted, as such that

$$X_{i-1} = \begin{bmatrix} x_{i-n} & x_{i-n+1} & \ldots & x_{i-2} & x_{i-1} \end{bmatrix} \quad (3.3.10)$$

bearing in mind that $x_1$ takes on a possible value of $b_1$. $n$ is an appropriate positive integer and $n>K$, where $K$ is the memory of the code in symbols. The corresponding cost $c_{i-1}$ is now said to be the cost of the vector $X_{i-1}$. Associated with the $2^g$ vectors $(X_{i-1})$ are stored their costs $(c_{i-1})$.

Each of the $2^g$ vectors $(X_{i-1})$ stored in the decoder here has the lowest cost for the corresponding one of $2^g$ possible states of the coder, where the given state is uniquely determined by a $g$-component vector (and $g=k_1+k_2$)

$$S_i = [x_{i-k_1}(2) \ldots x_{i-2}(2) x_{i-1}(2) x_{i-k_2}(1) \ldots x_{i-2}(1) x_{i-1}(1)] \quad (3.3.11)$$

On the receipt of the sample $r_1$, each stored vector $X_{i-1}$ is expanded to form four $(n+1)$-component vectors $(Z_i)$, where
\[ Z_i = \begin{bmatrix} x_{i-n} & x_{i-n+1} & \ldots & x_{i-1} & x_i \end{bmatrix} \quad (3.3.12) \]

The components \( x_{i-n}, x_{i-n+1}, \ldots, x_{i-1} \) here are as in the original vector \( X_{i-1} \), and the last component \( x_i \) takes on its four possible values in the four \( Z_i \). The corresponding code symbol \( p_i \) and thus the cost \( c_i \), of each expanded vector \( Z_i \), is determined by equations 3.3.6 to 3.3.9. There are now \( 4 \cdot 2^g \) vectors \( (Z_i) \) and the associated costs \( (c_i) \).

Bearing in mind that \( S_i \) is as given in equation 3.3.11, the corresponding state at time \( t=(i+1)T \), must be

\[ S_{i+1} = [x_{i-k_1+1}(2), x_{i-1}(2), x_i(2), x_{i-k_2+1}(1), x_{i-1}(1), x_i(1)] \quad (3.3.13) \]

Now, for each of \( 2^g \) different possible values of \( S_{i+1} \) defined by equation 3.3.13, the decoder selects the vector \( Z_i \) for which cost \( c_i \) is smallest. The resulting \( 2^g \) vectors \( (Z_i) \) and their costs \( (c_i) \) are stored. The decoder then selects from the \( 2^g \) "surviving" \( (Z_i) \), the vector \( Z_i \) with the minimum \( c_i \), and takes its first component \( x_{i-n} \) as the decoded value \( b'_{i-n} \) of \( b_{i-n} \). After this, the first components of the \( (Z_i) \) are omitted to give the corresponding \( 2^g \) \( n \)-component stored vectors \( (X_i) \) (without changing their costs). Finally, the lowest cost \( c_i \) is subtracted from each cost, setting the smallest cost to zero. The process continues in this way.

Since the convolutional coder only has \( 2^g \) different possible states and the Viterbi decoder here holds the lowest cost vector associated with each state, the vector \( X_i \) with the minimum cost, over all stored vectors, can be shown to give the
last \( n \) components of the required maximum likelihood sequence of decoded data symbols [2,3]. The decoder therefore achieves the maximum likelihood (and also the minimum distance) decoding of coded signals.

Theoretically, the final choice of the decoded message is not reached until the entire sequence \( (r_i) \) has been received (see Appendix A2 for details). But this usually means an unacceptable delay in detection. In practice, the delay of \( n \) symbols is introduced into the decoding process, where \( n \) is sufficiently large to avoid any significant increase in decoding errors due to early detection. Furthermore, in order to avoid an unacceptable increase in the costs, over a long message, the smallest cost \( c_i \) is subtracted from each cost, before the decoder starts to operate on the next received sample \( r_{i+1} \). Thus the decoder reduces its smallest cost to zero without changing the differences between the various stored costs.

In general, the Viterbi decoder must calculate \( 4 \times 2^g \) costs and carry out \( 2^g \) cost ranking operations, each involving four costs, in decoding each received sample. It must also store \( 2^g \) \( n \)-component vectors. Since \( 2^g \) is an exponential function of \( g \), both the amount of storage and the number of operations per decoded data symbol will become unacceptably large when \( g \gg 1 \).

The Viterbi algorithm decoding of rate-\( 3/4 \) convolutionally coded 16PSK signals is very similar to that for the rate-\( 2/3 \) codes just given, except the following modifications
now to be described.

The rate-3/4 convolutional codes considered in this work have an important advantage in that the corresponding Viterbi decoder operates on a binary signal only. The simple form of the decoder results from the fact that only one of the three bits in each vector of binary data symbols (that is, \( b_i(3) \) in Figure 3.2.7) is convolutionally coded in forming the coded vector \( v_i \) and the transmitted symbol \( q_i \). For a given value of \( b_i(3) \), an appropriate decision threshold detector can be used to determine the combination of \( b_i(1) \) and \( b_i(2) \), that corresponds to the coded signal \( p_i \), where \( p_i \) is closest to the received sample \( r_i \) in the complex number plane. The operation of the latter detector is similar to that for uncoded QPSK signals (see section 2.4), together with the two adjustable thresholds defined by the coded bit \( b_i(3) \). Thus, in effect, the receiver first divides the possible signals \( \{ p_i \} \) into a group of subsets (two subsets in this case). It then selects the best signal within each subset, and finally applies the Viterbi algorithm to the resulting signals. The two non-convolutionally coded bits \( b_i(1) \) and \( b_i(2) \) are Gray-coded, as shown in Figure 3.2.8.

The decoding process for rate-3/4 convolutionally coded 16PSK signals is now briefly outlined. On the receipt of the sample \( r_i \), two expanded vectors \( \{ Z_i \} \) are derived from each of the \( 2^6 \) stored vectors \( \{ X_{i-1} \} \), where the last component \( x_i \) takes on two of its eight possible values in the two \( \{ Z_i \} \). For a given \( X_{i-1} \), these two selected values of \( x_i \) (and thus the corresponding \( \{ z_i \} \)) are determined in the following manner. Firstly, in one of the two selected \( \{ x_i \} \), the last component
$x_i(3)$ takes on 0, whereas in the other $x_i$, $x_i(3)$ takes on 1. Secondly, for each of two possible values of $x_i(3)$, the decoder selects from the four possible combinations of $x_i(1)$ and $x_i(2)$, the $[x_i(1) x_i(2)]$ that corresponds to the coded signal $p_i$, where $p_i$ is closest to the received sample $r_i$ in the complex number plane. The given $[x_i(1) x_i(2) x_i(3)]$ is then taken to be the $x_i$ in the corresponding expanded vector $Z_i$.

These together result in $2 \times 2^g$ expanded vectors $\{Z_i\}$ for any code with a memory of $g$ bits. The decoder now carries out the rest of operations with the $2 \times 2^g$ expanded vectors, exactly as that for the rate-$2/3$ convolutionally coded 8PSK signals. However, the Viterbi decoder of this kind only needs to calculate $2 \times 2^g$ costs, and to carry out $2^g$ cost ranking operations, each involving two costs, in decoding each received sample.

The performances of the Viterbi decoder with different coded signals are now presented as graphs of bit error rate (BER) against the signal/noise ratio, $E_b/N_0$, in Figures 3.3.1 to 3.3.4. $E_b$ is the average signal energy per data bit and $(1/2)N_0$ is the two-sided power spectral density of the additive white Gaussian noise. The results were obtained from a number of computer simulation tests. Appendix B2 shows an example of the computer simulation programs used in the tests.

Figures 3.3.1 to 3.3.3 present the performance results of five rate-$2/3$ convolutional coded signals (see Table 3.2.1), whereas Figure 3.3.4 presents those of the four rate-$3/4$ convolutional coded 16PSK signals (see Table 3.2.2). The performance of the corresponding uncoded system with coherent detection is also shown on each diagram. The accuracy of each
curve on these graphs, is ±0.2 dB, over the range of bit error rates of $10^{-3}$ to $10^{-4}$. The simulation techniques and the accuracy of the results are discussed in Appendix A5.

The five rate-2/3 codes used in the tests correspond to 4, 8, 16, 32 and 64 states, respectively, giving theoretical asymptotic coding gains of 3.0 dB, 3.6 dB, 4.1 dB, 4.6 dB and 5.0 dB, over the uncoded QPSK system. Table 3.3.1 outlines the performances of each code at different bit error rates. From Table 3.3.1 it can be seen that, at a BER of $10^{-4}$, the simple scheme incorporating code UC(1) and the Viterbi decoder of 4 stored vectors provides an attractive gain in tolerance to noise of 2.25 dB, over the uncoded QPSK system. From 4 stored vectors to 64 stored vectors, an additional gain of 1 dB is achieved at a BER of $10^{-4}$. However, the equipment complexity of the decoder is now approximately sixteen times of that for 4 stored vectors. It appears that, at a BER of $10^{-4}$, the shortfall in the actual gain of each code, from its asymptotic value, increases with the code constraint length.

The other interesting comparison between the different codes is their average numbers of bit errors per burst at a given BER. The definition of burst errors is given in Appendix A5. Table 3.3.2 shows the burst error characteristics of each code at the various bit error rates.

Figures 3.3.2 and 3.3.3 show the variation of BER with the delay in detection, $n$ (measured in symbols). The two curves in Figure 3.3.2 are those obtained with code UC(1) (a 4-state code), at $E_b/N_0=4.5$ dB and 6.0 dB, respectively. From Figure 3.3.2 there is a sharp increase in bit error rates when
n is reduced to less than 16. The two curves in Figure 3.3.3 are those obtained with code UC(5) (a 64-state code), at $E_b/N_0=4.2$ dB and 5.2 dB, respectively. But here BER increases rapidly when n is less than 32 symbols. The evidence suggests that, if any unacceptable increase in decoding errors due to early detection is to be avoided, the delay in detection must increase correspondingly with the constraint length of the code. The results shown in Figure 3.3.1 are those obtained with a detection delay of 48 symbols ($n=48$).

Figure 3.3.4 shows the performance results for four rate-3/4 convolutionally coded 16PSK signals. Table 3.3.3 outlines the results at different bit error rates, whereas Table 3.3.4 shows the burst error characteristics of each code. The codes used in the computer simulation tests are the codes UC(6) to UC(9) in Table 3.2.2. Theoretically, these four codes are expected to gain 3.6 dB, 4.0 dB, 4.4 dB and 4.8 dB, respectively, over the uncoded 8PSK system, at very low bit error rates. From Table 3.3.3 the four codes gain 1.95 dB, 2.2 dB, 2.4 dB and 2.5 dB, respectively, at a BER of $10^{-4}$. The shortfall in the actual gain of each code, from its asymptotic value, appears to be quite significant, particularly for those codes of large constraint lengths, such as code UC(9). However, these four rate-3/4 codes have the important advantage, in that the corresponding Viterbi decoder operates on a binary signal only, leading to a potential reduction in the decoder complexity.

In the next section, a conventional near-maximum likelihood decoding scheme is considered for rate-2/3 convolutionally coded 8PSK signals.
Table 3.3.1: Performances of various rate-2/3 convolutionally coded 8PSK signals with the Viterbi decoding

<table>
<thead>
<tr>
<th>Code states</th>
<th>2^k</th>
<th>BER=0.01</th>
<th>BER=0.001</th>
<th>BER=0.0001</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC(1)</td>
<td>4</td>
<td>0.25</td>
<td>1.4</td>
<td>2.25</td>
</tr>
<tr>
<td>UC(2)</td>
<td>8</td>
<td>0.1</td>
<td>1.5</td>
<td>2.45</td>
</tr>
<tr>
<td>UC(3)</td>
<td>16</td>
<td>0.15</td>
<td>1.7</td>
<td>2.7</td>
</tr>
<tr>
<td>UC(4)</td>
<td>32</td>
<td>0.15</td>
<td>1.8</td>
<td>3.0</td>
</tr>
<tr>
<td>UC(5)</td>
<td>64</td>
<td>0.3</td>
<td>2.05</td>
<td>3.25</td>
</tr>
</tbody>
</table>
Table 3.3.2: Burst error characteristics of convolutionally coded 8PSK signals with the Viterbi decoding

<table>
<thead>
<tr>
<th>Code</th>
<th>Number of states</th>
<th>Average number of bit errors per burst at a given bit error rate, (approximately)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>2ε BER=0.01, 0.005, 0.001, 0.0005, 0.0001</td>
</tr>
<tr>
<td>UC(1)</td>
<td>4</td>
<td>5  5  4  3  2</td>
</tr>
<tr>
<td>UC(2)</td>
<td>8</td>
<td>10 10 7  5  4</td>
</tr>
<tr>
<td>UC(3)</td>
<td>16</td>
<td>10  9  9  7  11</td>
</tr>
<tr>
<td>UC(4)</td>
<td>32</td>
<td>12 10 9  -  14</td>
</tr>
<tr>
<td>UC(5)</td>
<td>64</td>
<td>13  -  9  7  7</td>
</tr>
</tbody>
</table>
Table 3.3.3: Performances of various rate-3/4 convolutionally coded 16PSK signals with the Viterbi decoding

<table>
<thead>
<tr>
<th>Code</th>
<th>Number of states,</th>
<th>Gain in tolerance to noise over 8PSK signals at a given bit error rate (dB),</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>BER=0.01, BER=0.001, BER=0.0001</td>
</tr>
<tr>
<td>UC(6)</td>
<td>4</td>
<td>-0.45, 1.1, 1.95</td>
</tr>
<tr>
<td>UC(7)</td>
<td>8</td>
<td>-0.4, 1.25, 2.2</td>
</tr>
<tr>
<td>UC(8)</td>
<td>16</td>
<td>-0.35, 1.4, 2.4</td>
</tr>
<tr>
<td>UC(9)</td>
<td>32</td>
<td>-0.3, 1.5, 2.5</td>
</tr>
</tbody>
</table>

Table 3.3.4: Burst error characteristics of convolutionally coded 16PSK signals with the Viterbi decoding

<table>
<thead>
<tr>
<th>Code</th>
<th>Number of state,</th>
<th>Average number of bit errors per burst at a given bit error rate, (approximately)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>BER=0.01, 0.005, 0.001, 0.0005, 0.0001</td>
</tr>
<tr>
<td>UC(6)</td>
<td>4</td>
<td>8, 8, 6, 6, 3</td>
</tr>
<tr>
<td>UC(7)</td>
<td>8</td>
<td>11, 9, 6, 6, 7</td>
</tr>
<tr>
<td>UC(8)</td>
<td>16</td>
<td>11, 8, 8, 6, 9</td>
</tr>
<tr>
<td>UC(9)</td>
<td>32</td>
<td>12, 7, 8, 6, 6</td>
</tr>
</tbody>
</table>
Fig 3.3.1: Performances of rate-2/3 convolutionally coded 8PSK signals with Viterbi decoding

Legend
- Δ Uncoded QPSK
- × UC[4]: 32 stored vectors
- ○ UC[5]: 64 stored vectors
- □ UC[2]: 8 stored vectors
- ♦ UC[3]: 16 stored vectors
- ◯ UC[1]: 4 stored vectors

Common attributes: n=48
Fig. 3.3.2: Variation of BER with detection delay

Legend

\[ \Delta \] at \( \text{Eb/No} = 4.5 \text{ dB} \)

\[ \times \] at \( \text{Eb/No} = 6.0 \text{ dB} \)

Common attributes: \( \text{UC(1)}: \) Viterbi decoding with 4 stored vectors
Fig. 3.3.3: Variation of BER with detection delay

Legend

\[ \triangle \text{ at } \frac{E_b}{N_0} = 4.2 \, \text{dB} \]

\[ \times \text{ at } \frac{E_b}{N_0} = 5.2 \, \text{dB} \]

Common attributes: UG(5); Viterbi decoding with 64 stored vectors
Fig. 3.3.4: Performances of rate-3/4 convolutionally coded 16PSK signals with the Viterbi decoding.
3.4 A NEAR-MAXIMUM LIKELIHOOD DECODING SCHEME (SYSTEM A)

In this section, one of known near-maximum likelihood decoding systems, called system A, is described for the rate-
2/3 convolutionally coded 8PSK signals of Section 3.2.

System A is one of the schemes developed by Clark, A.P. et al [85-93]. This system was originally proposed for
detecting the received signal over a linear channel with inter-
symbol interference and has gone through several modifications
since then. Many of these detectors come close to achieving
the maximum likelihood (minimum distance) detection of the
received signal without, however, requiring nearly as much
computation and storage as does the Viterbi detector, and have
therefore been known as the near-maximum likelihood detectors.

In all cases the detectors operate with fewer stored
vectors than the corresponding Viterbi detector. The method of
selecting the stored vectors differs from one scheme to
another, but it generally uses, as a measure of the goodness of
any stored vector, the unitary distance squared between the
sequences of samples actually received and those would have
been received in the absence of noise, had the transmitted
sequence of data symbols been given by the particular stored
vector. The greater the unitary distance squared, known as the
cost of the given stored vector (see Section 3.3), the less
likely is the vector to be correct. The vectors with the
highest costs are simply discarded.

The description of the received signal has already been
given in Section 3.3, where the Viterbi algorithm decoding is
described. In this section, system A is first described in
terms of its stored vectors and costs. The algorithm repeated during each symbol interval, which uses these stored values to generate the decoded data symbol $s'_1$, is then given. The results of the computer simulation tests for system A with different rate-2/3 convolutionally coded 8PSK signals are later presented.

Just prior to the receipt of the sample $r_1$, where, following equation 2.1.11,

$$ r_1 = q_1 + w_1 $$  \hspace{1cm} (3.11.1)

the decoder holds in store a set of $m$ different $n$-component vectors $(X_{i-1})$ of possible data symbols, and

$$ X_{i-1} = [ x_{i-n} \, x_{i-n+1} \, \ldots \, x_{i-2} \, x_{i-1} ] $$  \hspace{1cm} (3.11.2)

$m$ is an appropriate positive integer. Each component $x_i$ in the vector $X_{i-1}$ is a possible value of the 2-component vector of binary data symbols $b_i$ (see equation 3.3.3), so that

$$ x_i = [ x_i(1) \, x_i(2) ] $$  \hspace{1cm} (3.11.3)

$x_i(h) = 0$ or $1$, for $h=1$ and 2. Thus $x_i$ can take on any of the four values 00, 01, 10 and 11. Associated with each vector $X_{i-1}$ is stored its cost $c_i$. The definition of the cost $c_i$ has already been given in the appropriate part of Section 3.3.

On the receipt of $r_1$, each vector $X_{i-1}$ is expanded to form four $(n+1)$-component vectors $(Z_i)$, where
\[ Z_i = [ x_{i-n} \ x_{i-n+1} \ \ldots \ x_{i-1} \ x_i ] \] (3.4.4)

The components \( x_{i-n} \), \( x_{i-n+1} \), \ldots, \( x_{i-1} \) here are as in the original vector \( X_{i-1} \), and \( x_i \) takes its four different possible values in the four \( \{Z_i\} \). The associated costs \( \{c_i\} \) are calculated using equations 3.3.6 to 3.3.9, exactly as in case of the corresponding Viterbi decoder. There are now \( 4m \) expanded vectors \( \{Z_i\} \) together with their costs \( \{c_i\} \). The system A described so far is the same as the Viterbi decoder except for the number of stored vectors in the decoder.

The decoder next selects the vector \( Z_i \) with the lowest cost \( c_i \), over all \( 4m \) expanded vectors, and takes the first component \( x_{i-n} \) of this vector as the decoded value \( b'_{i-n} \) of \( b_{i-n} \). All vectors \( \{Z_i\} \) for which \( x_{i-n} = b'_{i-n} \) are then discarded. This prevents the merging (becoming the same) of the stored vectors, since it ensures that if all the vectors are different at the start of transmission, no two of them will subsequently become the same. The procedure is known as the anti-merging procedure.

The first components of all remaining expanded vectors \( \{Z_i\} \) are now omitted to give the corresponding \( n \)-component vectors \( \{X_i\} \) (without changing their costs). The decoder then selects from these vectors, the \( m \) vectors \( \{X_i\} \) with lowest costs \( \{c_i\} \) (including the lowest-cost vector just selected). Each selected vector and its cost are stored. None of the expanded vectors can be selected more than once. Therefore, after a vector has been selected, it must be excluded from the remaining selection processes. Finally, the decoder subtracts the smallest cost from each of the \( m \) stored cost values, as
does the Viterbi decoder. The decoder is now ready for the next received sample.

At the start of transmission, the \( m \) stored vectors are set to the same value (which should, if at all possible, be correct). A cost of zero is allocated to one of these vectors, whereas all the other vectors are assigned a very high cost. In this way, only after a few symbol intervals, all the stored vectors will be those derived from the original vector with zero cost and will all be different.

It is clear that, for a given value of \( m \), system A must evaluate \( 4m \) costs and carry out \( m \) cost ranking processes, each involving \( 4m \) costs, in decoding each data symbol. The decoder must also store \( m \) \( n \)-component vectors. In practice, \( m \) is often much smaller here than \( 2^k \), which is the total number of stored vectors in the corresponding Viterbi decoder. This will lead to a considerable reduction in equipment complexity of the decoder.

Figures 3.4.1 to 3.4.3 present the performance results for different arrangements of system A with various rate-2/3 convolutionally coded 8PSK signals. These results are obtained from a number of computer simulation tests. Appendix B3 presents an example of the computer simulation programs used the tests. In all cases the detection delay is 48 symbols (\( n=48 \)). The performances of the corresponding Viterbi decoding scheme and uncoded QPSK system with coherent detection are also shown on each diagram. In all graphs, the bit error rates of various schemes are plotted against the signal/noise ratio, \( E_b/N_0 \). The accuracy of each curve, over the range of bit error
rates of $10^{-3}$ to $10^{-4}$, is $\pm 0.2$ dB.

Table 3.4.1 summarizes the performances of different arrangements of system A at various bit error rates, whereas Tables 3.4.2 outlines the burst error characteristics for these systems. From Table 3.4.2 it is clear that, at a given bit error rate (BER), the average number of bit errors per burst for system A, in each case, is significantly larger than that for the corresponding Viterbi decoder (see Table 3.3.2), particularly at low signal/noise ratios.

Figure 3.4.1 shows the performance results of system A with code UC(3) (a 16-state code), using $m=4, 8$ and 16, respectively. From Figure 3.4.1 it is clear that, with 4 stored vectors only ($m=4$), the degradation in tolerances to noise, compared to the optimum values (achieved by the corresponding Viterbi decoder), is quite significant, over the range of bit error rates of $10^{-3}$ to $10^{-4}$. But, when $m$ increases to 16, the performance of system A approaches to that of the Viterbi decoder over the same range of bit error rates.

Figure 3.4.2 shows the performance results for system A with code UC(4) (a 32-state code), using $m=8$ and 16, whereas Figure 3.4.3 shows that with code UC(5) (a 64-state code), using $m=16$. From the two graphs here it is clear that, when the number of stored vectors $m$ is reduced to a half of $2^g$, the tolerance to noise can be reduced by up to 0.35 dB, relative to that of the corresponding Viterbi decoder, whereas if $m$ is reduced to a quarter of $2^g$, the corresponding degradation can be as high as 1.2 dB. Table 3.4.3 presents a more detailed performance comparison of the two decoding schemes.

As is also shown in Figures 3.4.1 and 3.4.2, the
performance of system A can be improved significantly by employing a larger value of $m$, together with an inevitable increase in the number of operations per decoded data symbol. Take the example in Figure 3.4.2. Here, an increase in $m$ from 8 to 16 in system A improves the performance by up to 1.1 dB, at a BER of $10^{-4}$. However, this also means that the amount of storage required and computation per decoded data symbol must now be doubled.

Figures 3.4.4 to 3.4.6 contrast the performances of system A with those of the Viterbi decoder having the same numbers of stored vectors. The comparisons made here are based on the assumption that the two decoding schemes using the same number of stored vectors have approximately the same degree of equipment complexity. From Figures 3.4.4 and 3.4.5 it is clear that, with 4 or 8 stored vectors, the Viterbi decoding scheme appears to be superior than those incorporating system A. From Figure 3.4.6, system A with code UC(4) and $m=16$ slightly outperforms (about 0.2 dB better) other schemes, at BER=$10^{-4}$. The conclusion is that, if simple decoding systems with 4 to 16 stored vectors are desirable, the scheme which incorporates a convolutional code of short constraint length with the Viterbi decoding is generally superior than those incorporating system A and convolutional codes of large constraint lengths.

There are two main weaknesses of system A that appear to be responsible for its general poor performances when applied to coded 8PSK signals.

Firstly, it is quite possible at any time for all the $m$ stored vectors $\{X_i\}$ to have the same value of $x_{i-j}$, for $j=J$, $J+1$, ..., $n$, so that $s_{i-j}$ is effectively decoded as $x_{i-j}$ after
only the corresponding small delay of J symbols. The shorter the period of time, \((i-J)T < t < iT\), for which the \(m\) stored vectors differ, the smaller will be the cost differences between these stored vectors even in the absence of noise. In such a case it is more likely, over this period of time, that the correct maximum likelihood vector is discarded in the presence of noise. This can noticeably reduce tolerance of the system to noise.

The second weakness of system A is that, in order to keep the amount of computation per decoded data symbol and the storage required at a reasonably low level, the number of stored vectors in system A must be kept small (typically \(m\) is in the range of 4 to 16), and very often this is only up to a fraction of that required by the corresponding Viterbi decoder. This means that during each decoding process, only a small portion of all different combinations of \(x_i, x_{i-1}, \ldots, x_{i-K}\) where \(K\) is the code memory in symbols, will be examined in selecting the \(m\) more likely vectors of data symbols (Notice that, following equation 3.3.13, the combination of \(x_i, x_{i-1}, \ldots, x_{i-K+1}\) defines a possible state value of the coder at time \(t=(i+1)T\)). Clearly, the latter selected \(m\) stored vectors with lowest costs are only locally most likely.

Furthermore, although the anti-merging procedure prevents any two stored vectors eventually becoming the same, it is quite possible at any time for two (or more) of the stored vectors to have the same combination of \(x_i, x_{i-1}, \ldots, x_{i-K+1}\) and thus correspond to the same state. However, following the theorem of maximum likelihood decoding [2,3], only the vector with the smallest cost among these needs, in
This will effectively reduce the number of stored vectors further by one (or more). Consequently, when the noise in transmission is sufficiently high to cause the correct maximum likelihood vector to be eliminated from the decoder, it may be quite some time before the correct sequence is restored. This normally results in a significantly longer error-burst compared to that in the corresponding Viterbi decoder.

Table 3.4.1: Performances of various rate-2/3 convolutionally coded 8PSK signals with system A decoding

<table>
<thead>
<tr>
<th>Code</th>
<th>Number of stored vectors in system A, 2^l</th>
<th>Number of vectors in system A, m</th>
<th>Gain in tolerance to noise over QPSK signals at a given bit error rate (dB), BER=0.01, BER=0.001, BER=0.0001</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC(3)</td>
<td>16</td>
<td>4</td>
<td>-1.6, -0.2, 0.65</td>
</tr>
<tr>
<td>UC(3)</td>
<td>16</td>
<td>8</td>
<td>-0.5, 1.2, 2.35</td>
</tr>
<tr>
<td>UC(4)</td>
<td>32</td>
<td>8</td>
<td>-0.95, 0.6, 1.8</td>
</tr>
<tr>
<td>UC(3)</td>
<td>16</td>
<td>16</td>
<td>0.0, 1.65, 2.7</td>
</tr>
<tr>
<td>UC(4)</td>
<td>32</td>
<td>16</td>
<td>-0.2, 1.55, 2.9</td>
</tr>
<tr>
<td>UC(5)</td>
<td>64</td>
<td>16</td>
<td>-0.5, 1.35, 2.6</td>
</tr>
</tbody>
</table>
### Table 3.4.2: Burst error characteristics of convolutionally coded 8PSK signals with system A decoding

<table>
<thead>
<tr>
<th>Code</th>
<th>Number of stored vectors as a fraction of $2^k$</th>
<th>Average number of bit errors per burst at a given bit error rate, (approximately)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>BER=0.05, 0.01, 0.005, 0.001, 0.0005, 0.00001</td>
</tr>
<tr>
<td>UC(3) 1/4</td>
<td>&gt;100  &gt;160  69</td>
<td>-  -  -</td>
</tr>
<tr>
<td>UC(3) 1/2</td>
<td>47  37  28  22</td>
<td>9  6</td>
</tr>
<tr>
<td>UC(3) 1</td>
<td>27  22  16  12</td>
<td>11  8</td>
</tr>
<tr>
<td>UC(4) 1/4</td>
<td>96  46  35  51</td>
<td>66  55</td>
</tr>
<tr>
<td>UC(4) 1/2</td>
<td>41  44  36  26</td>
<td>-  10</td>
</tr>
<tr>
<td>UC(5) 1/4</td>
<td>&gt;100  &gt;100  78  69</td>
<td>-  -</td>
</tr>
</tbody>
</table>
### Table 3.4.3: Relative complexities and performances of the different decoders with convolutionally coded 8PSK signals

<table>
<thead>
<tr>
<th>Code</th>
<th>Decoder</th>
<th>Number of stored vectors as a fraction of $2^8$</th>
<th>Tolerance to noise relative to that of the corresponding Viterbi decoder at a given error rate (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC(3)</td>
<td>Viterbi 1</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>system A 1/4</td>
<td>-1.75</td>
<td>-1.9</td>
</tr>
<tr>
<td></td>
<td>system A 1/2</td>
<td>-0.65</td>
<td>-0.5</td>
</tr>
<tr>
<td></td>
<td>system A 1</td>
<td>-0.15</td>
<td>-0.05</td>
</tr>
<tr>
<td>UC(4)</td>
<td>Viterbi 1</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>system A 1/4</td>
<td>-1.1</td>
<td>-1.2</td>
</tr>
<tr>
<td></td>
<td>system A 1/2</td>
<td>-0.35</td>
<td>-0.25</td>
</tr>
<tr>
<td>UC(5)</td>
<td>Viterbi 1</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td></td>
<td>system A 1/4</td>
<td>-0.8</td>
<td>-0.7</td>
</tr>
</tbody>
</table>

BER=0.01, BER=0.001, BER=0.0001
Fig. 3.4.1: Performance of system A with UC[3]

Legend
- △ Uncoded QPSK
- × m=4; n=48
- ◻ m=8; n=48
- ☆ m=16; n=48
- ✪ Viterbi; 16 stored vectors
Fig. 3.4.2: Performance of system A with UC[4]
Fig. 3.4.3: Performance of system A with UC[5]

**Legend**
- △ Uncoded QPSK
- × m=16; n=48
- □ Viterbi; 64 stored vectors
Fig. 3.4.4: Performance comparison between the different decoders of 4 stored vectors

Legend
- △ Uncoded QPSK
- × UC[3]; system A; m=4
- ○ UC[1]; 4-state; Viterbi

Common attribute: rate-2/3 convolutionally coded 8PSK signal
Fig. 3.4.5: Performance comparison between the different decoders of 8 stored vectors

Legend
- \( \triangle \) Uncoded QPSK
- \( \times \) UC[3]; system A; \( m=8 \)
- \( \square \) UC[4]; system A; \( m=8 \)
- \( \nabla \) UC[2]; 8-state; Viterbi

Common attribute: rate-2/3 convolutionally coded 8PSK signal
Fig. 3.4.6: Performance comparison between the different decoders of 16 stored vectors

Legend
- Uncoded QPSK
- UC[3]; system A; m=16
- UC[4]; system A; m=16
- UC[5], system A; m=16
- UC[3]; 16-state; Viterbi

Common attribute: rate-2/3 convolutionally coded 8PSK signal
Fig. 4.1.1: Modulo-M correlative coding process

\[ q = \cos(2\pi z_\ell / M) + j \sin(2\pi z_\ell / M) \]
CHAPTER FOUR

MODULO-M CORRELATIVE-LEVEL CODES

FOR MPSK SIGNALS
This chapter describes another class of trellis codes, known as the correlative-level codes. The codes are different from the binary convolutional codes of Chapter 3 in that such a coder is, in general, a simple type of finite-state machine defined over the infinite real-number field, as opposed to the Galois field GF(2) [18] over which a convolutional coder is defined [32].

Section 4.1 introduces the approach of combining the correlative-level coding with modulo-M processing of coded signals. This scheme was originally proposed by Clark, A.P. et al [48], and it is now applied to MPSK signals. Section 4.2 presents the procedure for designing the codes for binary data systems, whereas Section 4.3 presents that for quaternary (4-level) data systems. Finally in Section 4.4, the new codes are re-examined and those giving the best performance for the near-maximum likelihood decoding scheme of system A in Section 3.4, are determined.

4.1 CORRELATIVE CODED MPSK SIGNALS

The modulo-M correlative-level coding scheme was initially developed for coding a digital data signal for transmission over a linear baseband channel [48]. The codes are nonlinear, and the redundancy required by coding is provided by additional possible data-symbol values rather than
by transmitting additional signal elements. Clearly, this approach generates a class of bandwidth-efficient codes.

The modulo-M correlative-level coding, applied to MPSK signals, is now described. The assumed model of the data transmission system is presented in Section 2.1.

The original \( L \)-component vectors of binary data symbols \( \{b_i\} \) (see equation 2.1.2) from the data source in Figure 2.1.1 are first converted, via Gray coding, into a sequence of \( 2^L \)-level data symbols \( \{s_i\} \), where

\[
s_i = 0, 1, 2, \ldots, 2^L-1. \tag{4.1.1}\]

for \( i > 0 \). The Gray coding functions for 4- and 8-level data symbols are shown in Tables 2.4.1 and 2.4.2, respectively.

In general, a correlative-level coder may be implemented as a linear feedforward transversal digital filter with \((k+1)\) taps, each having a tap gain \( h \), for \( h=0, 1, \ldots, K \), as seen in Figure 4.1.1. \( K \) is, of course, the memory of the code in symbols. The coder is completely defined by the sampled impulse response of the given filter

\[
Y = [y_0 \ y_1 \ \ldots \ y_K] \tag{4.1.2}
\]

where the components \( y_0, y_1, \ldots, y_K \) are integer numbers, and the greatest common divisor of \( y_0, y_1, \ldots, y_K \) is 1. The vector \( Y \) is named as the coding vector.

At time instant \( t=iT \), where \( T \) is the symbol interval in seconds, the output from the given filter, or coder, is an integer number which is a function of \( s_i \) and one or more of the
data symbols $s_{i-1}, s_{i-2}, \ldots, s_{i-K}$. Coding is followed by a modulo-$M$ operation, as such that the code symbol $z_i$ (as an integer) at time $t=iT$, is given by

$$z_i = \sum_{h=0}^{K} s_{i-h} r_h \mod M$$  \hspace{1cm} (4.1.3)$$

where $M > 2^L$. The arithmetic operations (i.e., the multiplications and additions) in equation 4.1.3 are those with ordinary numbers, and the modulo-$M$ operation is as such that $z_i$ is the positive-valued remainder of

$$\sum_{h=0}^{K} s_{i-h} r_h$$  \hspace{1cm} (4.1.4)$$

upon division by $M$, so that

$$z_i = 0, 1, 2, \ldots, (M-1),$$  \hspace{1cm} (4.1.5)$$

$z_i$ is next mapped onto the complex number plane, to give the corresponding transmitted symbol $q_i$. The mapping is as such that, for a given $z_i$, the phase angle of $q_i$ in the complex number plane is determined by

$$\phi_i = \frac{2\pi z_i}{M}$$  \hspace{1cm} (4.1.6)$$

and now,

$$q_i = \cos \phi_i + j \sin \phi_i$$  \hspace{1cm} (4.1.7)$$

where $j = \sqrt{-1}$. The sequence of transmitted symbols $\{q_i\}$ is fed to the linear modulator, as shown in Figure 2.1.1.

For mathematical convenience, the information sequence or message at the input to the coder, may also be generally
expressed as an infinite power series in the delay operator \( D \)

\[
s(D) = s_0 + s_1 D + s_2 D^2 + s_3 D^3 + \ldots
\]  
(4.1.8)

and the correlative coder is described by a polynomial in \( D \)

\[
Y(D) = y_0 + y_1 D + y_2 D^2 + \ldots + y_K D^K
\]  
(4.1.9)

where, equivalently, \( D \) represents a delay of \( T \) seconds in the time domain. The sequence of code symbols \( \{ z_1 \} \) may therefore be expressed as an infinite power series in \( D \)

\[
z(D) = z_0 + z_1 D + z_2 D^2 + z_3 D^3 + \ldots
\]

\[
= s(D)Y(D) \mod M
\]

(4.1.10)

and the phase angles of transmitted symbols \( \{ q_i \} \), in the complex number plane, are given by

\[
\Phi(D) = (2\pi/M)z(D) \text{ (radians)}
\]

(4.1.11)

In the remaining part of this section, the minimum free unitary distance, \( d_{\text{free}} \), of a correlative coded MPSK signal is defined.

Suppose that \( \{ q_i \} \) and \( \{ q'_i \} \) are the two possible (valid) coded signal sequences of length \( N \) symbols that differ in their first code symbols. Also let the corresponding phase angles of \( q_i \) and \( q'_i \) be \( \Phi_i \) and \( \Phi'_i \), respectively, for each \( i \). The unitary distance squared between \( \{ q_i \} \) and \( \{ q'_i \} \) is

\[
d^2(q_i,q'_i) = \sum_{i=0}^{N-1} \left| q_i - q'_i \right|^2
\]

\[
= \sum_{i=0}^{N-1} \left[ (\cos\Phi_i - \cos\Phi'_i)^2 + (\sin\Phi_i - \sin\Phi'_i)^2 \right]
\]
\[ N-1 = \sum_{i=0}^{N-1} \left[ 2 - 2\cos(\phi_i - \phi'_i) \right] \quad (4.1.12) \]

Following equations 4.1.3 and 4.1.6,

\[ d^2(q_i, q'_i) = \sum_{i=0}^{N-1} \left[ 1 - \cos(2\pi(z_i - z'_i)/M) \right] \]

\[ = \sum_{i=0}^{N-1} \sum_{h=0}^{K} \left[ 1 - \cos(2\pi (s_i - s'_i) y_{ih}/M) \right] \quad (4.1.13) \]

It is assumed that \( s_i \) and \( s'_i = 0 \) for \( i < 0 \), but \( s_i \) and \( s'_i = 0, 1, 2, \ldots, 2^{L-1}, \) for \( i \geq 0 \). Consequently, the minimum free unitary distance squared of the given correlative coded signal must be

\[ d^2_{\text{free}} = \min_{N \to \infty} \{ d^2(q_i, q'_i) \} \quad (4.1.14) \]

The minimisation is carried out over all pairs of possible (valid) coded signal sequences of arbitrary length that differ in their first code symbols.

For the correlative coded MPSK signal generated by the coding vector \( Y \), \( d_{\text{free}} \) may be determined by equations 4.1.12 to 4.1.14. Notice that, in equation 4.1.13, the term \( (s_i - s'_i) \) represents the difference between any two possible data-symbol values and can therefore be replaced by an integer number \( a_i \), where

\[ a_i = s_i - s'_i \quad (4.1.15) \]

Clearly, \( a_i \) can take on any of the values \( 0, \pm 1, \pm 2, \ldots, \pm 2^{L-1} \), except for \( a_0 \). Since it is assumed that \( s_0 \neq s'_0 \), \( a_0 \neq 0 \).
Therefore, the minimisation in equation 4.1.14 is, in fact, carried out over all possible (valid) coded signal sequences \( \{q_i\} \) that are generated by the different possible sequences of \( \{a_i\} \), where, \( a_i = 0 \) for \( i < 0 \), \( a_0 = 1, 2, \ldots, 2^L-1 \), and \( a_i = 0, \pm 1, \pm 2, \ldots, \pm 2^{L-1} \), for \( i > 0 \).

For any given sequence of \( \{a_i\} \), the phase angles of the corresponding transmitted symbols \( \{q_i\} \), in the complex number plane, are determined from

\[
\phi_i = \frac{2\pi z_i}{M} \quad \text{(radians)} \tag{4.1.16}
\]

where

\[
z_i = \sum_{h=0}^{K} a_i-h y_h \mod M \tag{4.1.17}
\]

In the following sections, the procedures for designing correlative coded MPSK signals are presented. The design of codes begins with a consideration of binary-valued data symbols (where \( L=1 \)), and it is then extended to quaternary data symbols (where \( L=2 \)).
The procedure for designing the correlative coded MPSK signals begins with a consideration of the simplest case of all, where the data symbol $s_i$ can only take on two possible values,

$$s_i = 0, 1 \quad (11.2.1)$$

Binary signaling has not been one of the primary schemes proposed for satellite communication services in recent years, simply because a better bandwidth efficiency has been required due to the limited available frequency-band for today's satellite transmission. However, designing codes for binary data symbols, in particular the computation involved in the determination of the minimum free unitary distance of the code, is considerably simpler than that for higher level data symbols. For this reason, the investigation with binary data symbols was first carried out, in order to find the code structure with good distance properties.

The minimum free unitary distance for uncoded binary PSK (BPSK) signals is given by the Euclidean distance between the two possible transmitted symbols that correspond to the phase states of 0 and $\pi$ radians, respectively. Bearing in mind that

$$|q_i|^2 = 1.0 \quad (11.2.2)$$

(see equation 2.1.7),

$$d^2_{\text{uncoded}} = [1.0-(-1.0)]^2$$
$$= 4.0 \quad (11.2.3)$$
for BPSK signals.

As already mentioned in Section 3.1, most good convolutional codes are determined by the computerized searches of a large number of codes, to locate those with good distance properties. This is due to the fact that constructing a convolutional code using its algebraic properties is relatively difficult to achieve. For a similar reason, computer aided searches have also been carried out here, in order to locate those coding vectors with good distance properties.

The search for good codes was carried out in the following manner. For a given code of constraint length $K$ and a given value of $M$, an exhaustive search may be carried out over all different possible combinations of

$$Y_0, Y_1, Y_2 \ldots, Y_K$$  \hspace{1cm} (4.2.4)

where $y_h=0, \pm 1, \pm 2, \ldots, \pm M/2$, for $h=1, 2, \ldots, (K-1)$, but $y_h=1, 2, \ldots, (M-1)$, for $h=0$ and $K$.

$M$ is here assumed to be an even number. The following two cases, however, are excluded from the search. Firstly, any coding vector for which the largest common divisor of $Y_0, Y_1, \ldots, Y_K$ and $M$ is greater than 1, is automatically excluded, since such a coder produces catastrophic errors in the decoded data symbols. Secondly, any coding vector for which $Y_0 \neq Y_K$ is also excluded from the search. This is explained as follows. Suppose the two coding vectors $Y$ and $Y'$, where

$$Y = [ Y_0 \ Y_1 \ \ldots \ Y_K ]$$  \hspace{1cm} (4.2.5)
and  \( y' = [v'_0 \ v'_1 \ldots \ v'_K] \)  

(4.2.6)

are as such that,  \( v_0 \ v_1 \ldots \ v_K \) is the time inverse of  \( v'_0 \ v'_1 \ldots \ v'_K \), that is  \( v_0 = v'_K \),  \( v_1 = v'_{K-1} \), and so on. As far as the calculation of  \( d^2_{\text{free}} \) is concerned, the two coding vectors are exactly identical (see equation 4.1.13). Hence, only one of these two vectors need be considered. This is done by automatically excluding any vectors for which  \( v_0 \leq v_K \) in the computer aided search.

For each coding vector tested, the minimum free unitary distance squared,  \( d^2_{\text{free}} \), is calculated using equations 4.1.14 to 4.1.17. Any vector for which  \( d^2_{\text{free}} \) is smaller than the threshold level,  \( d_{\text{th}} \), will be automatically rejected. The threshold value  \( d_{\text{th}} \) indicates the level at which  \( d^2_{\text{free}} \) is to be expected for the given values of  \( K \) and  \( M \), and  \( d_{\text{th}} \) may be easily obtained in the following manner. Suppose the search begins with  \( K=1, M=4 \), and  \( d_{\text{th}}=0.0 \), the largest  \( d^2_{\text{free}} \) found in this search is taken to be the value of  \( d_{\text{th}} \) for the next search, where  \( K=2 \) and  \( M=4 \). The largest  \( d^2_{\text{free}} \) then found will be the new threshold in the next search, where  \( K=2 \) and  \( M=6 \), and so on. The search continues in this way. In each case only those coding vectors giving a larger value of  \( d^2_{\text{free}} \) compared to those with shorter constraint lengths and/or fewer signal levels, need be recorded.

The theoretical prediction of Section 3.2 states that the majority of achievable coding gains in terms of channel capacity can be obtained by doubling the number of possible data-symbol values. The design of correlative codes with binary data symbols has therefore been concentrated on  \( M=4 \)
(that generates a coded QPSK signal). A range of different values of $M$, up to $M=64$, have also been considered. Some of the more important results, obtained from the extensive searches described above, are now presented.

It was found with the aid of a computer search program (see Appendix B4 for the program with $K=2$ and $M=4$) that, when $K=2$, the coding vector

\[ Y = [2 \ 1 \ 2] \quad (4.2.7) \]

has a minimum free unitary distance squared of

\[ d_{\text{free}}^2 = 4.0 + 2.0 + 4.0 = 10.0 \quad (4.2.8) \]

which corresponds to an asymptotic coding gain of $10.0/4.0=4.0$ dB. Suppose $M$ may take on any even number, and increases to 10, the coding vector

\[ Y = [5 \ 3 \ 5] \quad (4.2.9) \]

was found to offer a minimum distance squared of 10.61, leading to an additional asymptotic coding gain of 0.2 dB. Any further increase in $M$, up to 64, was not found to give any additional improvement in the minimum free distance of the code.

Notice that, for the coding vector $[5 \ 3 \ 5]$ with $M=10$, the code symbol $z_i$ has four different possible values, that are 0, 3, 5 and 8, respectively, with the phase angles of $q_i$ being $0$, $3\pi/5$, $\pi$ and $8\pi/5$ radians. Clearly, the constellation of the given coded PSK (4-level) signal is asymmetric, as opposed to
any conventional QPSK signals.

The search continued to $K=3$ and $M=4$. The best code then found has a minimum free distance squared of 12.0. When $M=10$, this distance was increased to 13.24, leading to an additional asymptotic gain of 0.4 dB. Further increase in $M$, up to 64, did not appear to improve the minimum free distance any further.

The results of the computer aided searches so far appeared to follow the theoretical prediction of Section 3.2, in that the majority of the achievable gains is obtained by doubling the number of possible transmitted symbol values. From now on, the design is concentrated on where $M$ is in the range of 4 to 16. Furthermore, any code with $M>16$ is not of potential interest, since it requires an accurate estimation of the carrier-phase at the receiver.

A number of correlative codes with good minimum free unitary distances for the given code constraint lengths are summarized in Table 4.2.1. The coding vector of each code is presented in the form of a polynomial in the delay operator $D$. Furthermore, the codes need not only be implemented in the form of a feedforward transversal filter, but can also employ a feedback transversal filter, or a combination of both. A few examples of such codes with good distance properties are shown in Table 4.2.2.

All the vectors shown in Table 4.2.1 are characterised by the fact that, the first and last components of each coding vector are always equal to $M/2$. Now, reconsider the Coding Rule (1) of Section 3.2, where convolutional coded MPSK signals
are described. Coding Rule (1) ensures that the unitary distance squared between the portion of any two convolutionally coded signal sequences, starting from the symbol at which they diverge and ending in the symbol at which they converge, is at least twice of that of the uncoded system. For a correlative code with binary data symbols, however, if the first and last components of the given coding vector are set to M/2 (as those shown in Table 4.2.1), the unitary distance squared between the portion of any two correlative coded signal sequences here, starting from the symbol at which they diverge and ending in the symbol at which they converge, must also be at least twice of that of the uncoded system. In this sense, both coding techniques appeared to be equivalent.

The second interesting point in Table 4.2.1 is that, for the coding vectors [5 3 5] and [5 2 5 5], the redundancy required by coding is still provided by doubling the number of possible transmitted symbol values, but the constellation of the corresponding PSK signal is now asymmetric. However, these two codes have been shown to provide the largest minimum free unitary distances for the given code constraint lengths. The evidence here suggests that, although the symmetric signal constellation has been proven to give the optimum performance for uncoded signals, it may not necessarily be so with coded signals.
### Table 4.2.1: Best modulo-M correlative-level codes for binary data symbols

<table>
<thead>
<tr>
<th>Coding vector as a polynomial in D, $Y(D)$</th>
<th>Memory of the code in symbols, $K$</th>
<th>Number of signals in expanded signal set, $M$</th>
<th>Minimum unitary distance squared, $d^2_{\text{free}}$</th>
<th>Asymptotic coding gain over binary PSK signals, $G_c$ (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$2+D+2D^2$</td>
<td>2</td>
<td>4</td>
<td>10.0</td>
<td>4.0</td>
</tr>
<tr>
<td>$5+3D+5D^2$</td>
<td>2</td>
<td>10</td>
<td>10.61</td>
<td>4.24</td>
</tr>
<tr>
<td>$2+D+2D^2+2D^3$</td>
<td>3</td>
<td>4</td>
<td>12.0</td>
<td>4.77</td>
</tr>
<tr>
<td>$5+2D+5D^2+5D^3$</td>
<td>3</td>
<td>10</td>
<td>13.24</td>
<td>5.2</td>
</tr>
<tr>
<td>$2+D+2D^2+2D^3$</td>
<td>4</td>
<td>4</td>
<td>14.0</td>
<td>5.44</td>
</tr>
<tr>
<td>$2+2D^2+2D^3+D^4+2D^5$</td>
<td>4</td>
<td>4</td>
<td>16.0</td>
<td>6.0</td>
</tr>
</tbody>
</table>

### Table 4.2.2: Correlative codes implemented as a combination of feedforward and feedback transversal filters

<table>
<thead>
<tr>
<th>Coding vector as a polynomial in D, $Y(D)$</th>
<th>Memory of the code in symbols, $K$</th>
<th>Number of signals in expanded signal set, $M$</th>
<th>Minimum unitary distance squared, $d^2_{\text{free}}$</th>
<th>Asymptotic coding gain over binary PSK signals, $G_c$ (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$4+2D+3D^2$</td>
<td>4</td>
<td>8</td>
<td>12.58</td>
<td>5.0</td>
</tr>
<tr>
<td>$8+4D+5D^2$</td>
<td>5</td>
<td>16</td>
<td>15.95</td>
<td>6.0</td>
</tr>
<tr>
<td>$4+2D+D^2+3D^3$</td>
<td>5</td>
<td>8</td>
<td>14.34</td>
<td>5.54</td>
</tr>
</tbody>
</table>
4.3 CORRELATIVE CODES FOR QUATERNARY DATA SYSTEMS

This section describes the modulo-M correlative codes for quaternary data symbols \( \{ s_i \} \), where

\[
s_i = 0, 1, 2 \text{ and } 3. \quad (4.3.1)
\]

The original vectors of binary data symbols \( \{ b_i \} \), for which

\[
b_i = [ b_i(1) \ b_i(2) ] \quad (4.3.2)
\]

are first converted, via Gray coding, into the sequence of 4-level data symbols \( \{ s_i \} \). The Gray coding function for quaternary symbols is shown in Table 2.4.1.

Following the previous investigation with binary data symbols (the results of which are presented in Section 4.2), the similar computer aided searches of good coding vectors were next carried out with quaternary data symbols. For each given value of the code constraint length \( K \) and expanded signal level \( M \) considered, an exhaustive search was carried out over all different possible combinations of

\[
y_0, y_1, y_2, \ldots, y_K \quad (4.3.3)
\]

where \( y_h = 0, 1, 2, \ldots, \lfloor M/4 \rfloor \), for \( h = 1, 2, \ldots, (K-1) \), and \( y_h = M/4 \), for \( h = 0 \) and \( K \).

Only the cases where \( M \) is an integer multiple of 4 were considered. Again, any coding vector for which the greatest common divisor of \( y_0, y_1, \ldots, y_2 \), and \( M \), is greater than 1,
was excluded from the search. For each tested coding vector \( y \), the minimum free unitary distance squared, \( d_{\text{free}}^2 \), is calculated using equations 4.1.14 to 4.1.17.

The searches were concentrated on \( M=8 \), which generates a coded 8PSK signal, as in cases of the rate-2/3 convolutional codes of Section 3.2. In addition, different values of \( M \), up to \( M=64 \), have also been considered. However, no improvement in the minimum distances, compared to those of best correlative coded 8PSK signals, was ever found.

Table 4.3.1 presents a number of correlative codes with the best minimum free unitary distances for the given constraint lengths. The table also shows, for each code, the minimum free unitary distance squared \( d_{\text{free}}^2 \), the asymptotic coding gain over the uncoded QPSK system \( G_c \) (dB), and the asymptotic gain relative to that of the corresponding rate-2/3 convolutional coded 8PSK signal. From Table 4.3.1 the majority of the correlative codes here has an asymptotic coding gain which is about 0.5 dB inferior to that of the corresponding best known convolutional code.

Clearly, the correlative-level codes need not only be implemented in the form of a feedforward transversal filter, but also they can employ a feedback transversal filter, or a combination of both. An example of such codes with good distance properties is shown in Table 4.3.2. This code offers a very large asymptotic coding gain (about 6.3 dB) over the uncoded QPSK system. However, the corresponding 64PSK signal requires a very sophisticated carrier-phase estimation process at the receiver. Furthermore, due to the large constraint length of the code, the Viterbi algorithm decoding of the given
coded signal is also difficult, if not impossible, to achieve in practice.

Figure 4.3.1 presents the performance results, obtained from a number of computer simulation tests, for the first three codes CC(1) to CC(3) shown in Table 4.3.1. The decoding scheme employed in the tests is the Viterbi decoding algorithm of Section 3.3. The performance of the uncoded QPSK system with coherent detection is also shown on the graph. The performance results are shown as graphs of bit error rate (BER) against the signal/noise ratio, $E_b/N_0$. $E_b$ is the average signal energy per data bit and $(1/2)N_0$ is the two-sided power spectral density of the additive white Gaussian noise. The accuracy of each curve, in the range of bit error rates of $10^{-3}$ to $10^{-4}$, is ±0.2 dB. The simulation techniques and the confidence limits of the results are discussed in Appendix A5. Due to the limited computation speed of the computer, simulation tests on the various new codes with more than 64 states in Table 4.3.1 were not carried out.

Table 4.3.3 outlines, at each given BER, the tolerances of each new code to noise relative to that of uncoded QPSK signals, and relative to that of the convolutional code. From Table 4.3.3 the three new codes gain 2.8 dB, 3.1 dB and 3.25 dB, respectively, over the uncoded QPSK system. These are certainly close to those of the convolutional codes (see Table 3.3.1), despite the fact that the latter codes are expected to be 0.5 dB better at very high signal/noise ratios. It appears that, at a BER of $10^{-4}$, the shortfall in the actual coding gain of each new code, compared to its asymptotic value, is not so significant as that for the corresponding convolutionally coded
BPSK signal.

The second interesting point shown in Figure 4.3.1 is that, although the codes CC(2) and CC(3) provided the same theoretical coding gain of 4.1 dB, at high signal/noise ratios, the performance of code CC(3) appeared to be slightly better than that of CC(2), over the range of bit error rates of $10^{-3}$ to $10^{-4}$.

A possible explanation for these results is that, at the practical error rates of $10^{-3}$ to $10^{-4}$, the probability of error in the decoded data symbols is not only determined by the minimum free unitary distance of the code, but it also depends on the second, the third, and so on, smallest unitary distances of the given coded signal (see Appendix A3 for details).

The other fact which must also have a bearing on the performance of any code is the average number of bit errors associated with an incorrect selection of the decoded data sequence at each of these distances. The evidence here suggests that, for different coded signals tested so far, these distances and their associated average numbers of bit errors must somehow be quite different, even though the codes may have the same code constraint length and/or the same $d^2_{\text{free}}$. This will certainly affect the performances of the corresponding coded signals, over the range of bit error rates of $10^{-3}$ to $10^{-4}$.

Table 4.3.4 shows the burst error characteristics of the three new codes tested. The definition of burst errors is given in Appendix A5. It is clear that, on average, the numbers of bit errors per burst for each of the correlative codes here, at a BER of $10^{-4}$, are very similar (or slightly
smaller) to those for the corresponding convolutional codes (see Table 3.3.2).

Figure 4.3.2 presents the variations of BER with the delay in detection $n$ (symbols), for code $CC(3)$ (a 64-state code), at $E_b/N_o=4.0$ dB and 5.2 dB, respectively. The two curves here are similar to those for the convolutional code $UC(5)$ (which also has 64 states) shown in Figure 3.3.3. The performance results presented in Figure 4.3.1 are those obtained with a delay of 48 symbols ($n=48$).

<table>
<thead>
<tr>
<th>Code</th>
<th>Coding vector as a polynomial in $D$, $Y(D)$</th>
<th>Memory of the code in symbols, $K$</th>
<th>Minimum unitary coding gain $d^2_{free}$</th>
<th>Asymptotic coding gain $G_c$ (dB)</th>
<th>Gain relative to convolutionally coded 8PSK</th>
</tr>
</thead>
<tbody>
<tr>
<td>$CC(1)$</td>
<td>$2+D+2D^2$</td>
<td>2</td>
<td>4.5858</td>
<td>3.6</td>
<td>-0.5</td>
</tr>
<tr>
<td>$CC(2)$</td>
<td>$2+D+2D^2+2D^3$</td>
<td>3</td>
<td>5.1716</td>
<td>4.1</td>
<td>-0.5</td>
</tr>
<tr>
<td>$CC(3)$</td>
<td>$2+D+D^2+2D^3$</td>
<td>3</td>
<td>5.1716</td>
<td>4.1</td>
<td>-0.5</td>
</tr>
<tr>
<td>$CC(4)$</td>
<td>$2+D+D^2+D^3+2D^4$</td>
<td>4</td>
<td>5.7613</td>
<td>4.6</td>
<td>-0.5</td>
</tr>
<tr>
<td>$CC(5)$</td>
<td>$2+D+2D^2+D^4+2D^5$</td>
<td>5</td>
<td>7.1716</td>
<td>5.5</td>
<td>-0.7</td>
</tr>
</tbody>
</table>
Table 4.3.2: A coder implemented as a combination of feed-forward and feedback transversal filters

<table>
<thead>
<tr>
<th>Coding vector as a polynomial in D, ( Y(D) )</th>
<th>Memory of the code in symbols, ( K )</th>
<th>Number of signals in expanded signal set, ( M )</th>
<th>Minimum unitary distance squared, ( d_{free}^2 )</th>
<th>Asymptotic coding gain over QPSK signals, ( G_c ) (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 16+4D+17D^2 )</td>
<td>8</td>
<td>64</td>
<td>8.50</td>
<td>6.28</td>
</tr>
<tr>
<td>( 1-4D^3 )</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4.3.3: Performances of correlative coded 8FSK signals with the Viterbi decoding

<table>
<thead>
<tr>
<th>Number of states, ( 4^K )</th>
<th>Gain in tolerance to noise over QPSK signals, at a given bit error rate, (dB)</th>
<th>Coding gain relative to that of the corresponding rate-2/3 convolutional code, (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BER=0.01, 0.001, 0.0001, 0.01, 0.001, 0.0001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CC(1) 16</td>
<td>0.2 1.8 2.8 0.05 0.0 0.1</td>
<td></td>
</tr>
<tr>
<td>CC(2) 64</td>
<td>0.35 2.05 3.1 0.05 0.0 -0.15</td>
<td></td>
</tr>
<tr>
<td>CC(3) 64</td>
<td>0.35 2.15 3.25 0.05 0.1 0.0</td>
<td></td>
</tr>
</tbody>
</table>
Table 4.3.4: Burst error characteristics of correlative coded 8PSK signals with the Viterbi decoding

<table>
<thead>
<tr>
<th>Code</th>
<th>Number of Code states</th>
<th>Average number of bit errors per burst at a given bit error rate (approximately)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4K</td>
<td>BER=0.01, 0.005, 0.001, 0.0005, 0.0001</td>
</tr>
<tr>
<td>CC(1)</td>
<td>16</td>
<td>11, 7, 6, 5, 3</td>
</tr>
<tr>
<td>CC(2)</td>
<td>64</td>
<td>16, 8, 7, 5, 4</td>
</tr>
<tr>
<td>CC(3)</td>
<td>64</td>
<td>11, 10, 8, 8, 6</td>
</tr>
</tbody>
</table>
Fig. 4.3.1: Performances of correlative coded 8PSK signals with Viterbi decoding

Legend

- Uncoded QPSK
- CC[1]; 16 stored vectors
- CC[2]; 64 stored vectors
- CC[3]; 64 stored vectors

Common attribute: $n=48$
Fig. 4.3.2: Variation of BER with detection delay

Legend

△ at Eb/No = 4.0 dB
× at Eb/No = 5.2 dB

Common attributes: CC[3]; 8PSK; Viterbi decoding with 64 stored vectors
4.4 SUB-OPTIMAL CORRELATIVE CODES FOR SYSTEM A

This section examines the modulo-M correlative codes that give the best performance for the near-maximum likelihood decoding scheme, system A, of Section 3.4. However, the new codes may now be sub-optimal in terms of minimum free unitary distances for the given code constraint lengths.

From Section 3.4, system A uses, as a measure of goodness of any stored vector of data symbols, the unitary distance squared (known as the cost of the given vector) between the sequences of the sample values actually received and the sample values that would have been received in the absence of noise, had the transmitted data sequence been given by the particular stored vector. System A simply operates as follows. The decoder stores a set of m lowest cost vectors. On the receipt of each received sample, the vectors are expanded, new costs are calculated, and a set of m lowest cost vectors are then selected from the expanded vectors.

Clearly, system A has two main weaknesses. Firstly, it is quite possible at any time for all the m stored vectors to have the same value of $x_{i-j}$, for $j=J, J+1, \ldots, n$, so that $s_{i-j}$ is effectively decoded as $x_{i-j}$ only after a small delay of $J$ symbols. Secondly, only a fraction of all different possible combinations of $s_1, s_{1-1}, \ldots, s_{1-K}$, where $K$ is the memory of the code in symbols, is examined during each decoding process. These result in prolonged error bursts and thus severely degrade the performance of the system. An obvious solution is to employ a larger value of $m$, together with an inevitable increase in the number of operations per decoded data symbol.
In this section, however, a slightly different approach is considered. The new correlative codes are re-examined to locate those giving the best performance for a given number of stored vectors in system A.

First of all, the distance profiles of correlative codes need be considered. The distance profile of a binary convolutional code is defined in Section 3.1 (see equation 3.1.26). The distance profile of a correlative code, when the unitary distance measure is employed, may be defined as follows.

\[ \Omega = \{ \eta_1, \eta_2, \ldots, \eta_{K+1} \} \tag{4.4.1} \]

where (K+1) is the code constraint length in symbols, and \( \eta_h \) is the minimum unitary distance squared between any two possible (valid) coded signal sequences of length \( h \) symbols that differ in their first code symbols.

The distance profile of any code is clearly a measure of the rate of growth in unitary distance between any two coded signal sequences that differ in their first symbols. If this distance increases only slowly with time, the costs of the corresponding two stored vectors of data symbols in system A, where one of them is correct, will be very similar over quite a long period of time even in the absence of noise. This will effectively increase the probability of discarding the correct maximum likelihood vector in the presence of noise and also the time taken to restore the correct vector, once it is falsely eliminated from the decoder at some stage. Clearly, if system A incorporates the correlative codes with generally good distance profiles, the performance of the system will almost
certainly be improved. Therefore, the coding vectors with good distance profiles need be allocated.

The investigation begins with a consideration of binary data symbols. Computer aided searches similar to those shown in Section 4.2 may be carried out for different values of $K$ and the number of signal levels $M$. The searches included those over all different possible combinations of

$$
\gamma_0, \gamma_1, \gamma_2, \ldots, \gamma_K
$$

(4.4.2)

where $\gamma_h = 0, 1, 2, \ldots, M/2$, for $h = 1, 2, \ldots, (K-1)$, $\gamma_h = 1, 2, \ldots, M/2$, for $h = 0$ and $K$, and $\gamma_0 \neq \gamma_K$.

Again, any coding vector for which the greatest common divisor of $\gamma_0, \gamma_1, \ldots, \gamma_K$, and $M$ is greater than 1 is excluded from the search. In all cases, $\eta_{K+1}$ is calculated for each coding vector tested. Following equations 4.1.13 to 4.1.17,

$$
\eta_{K+1} = \sum_{i=0}^{K+1} \cos\left(\frac{2\pi i}{M} \sum_{h=0}^{K} a_{i-h} \gamma_h\right)
$$

(4.4.2)

where $a_i = 0$ for $i < 0$, $a_0 = 1$, and $a_i = 0$, $i \neq 1$, for $i > 0$. The coding vectors with the best $\eta_{K+1}$ for the given values of $K$ and $M$ are recorded.

Tables 4.4.1 to 4.4.3 presents the various correlative codes found with good distance profiles and/or minimum free distances for the given code constraint lengths. The coding vector of each code is shown with the corresponding distance profile and minimum free unitary distance. Tables 4.4.4 to 4.4.6 show the burst error characteristics for these codes incorporating different arrangements of system A. These
results are obtained from a number of computer simulation tests (see Appendix B3 for the program of system A). The model of the data transmission system used in the tests had been described in section 2.1.

The codes Y₁ to Y₅ (that with a constraint length of 4 symbols) in Table 4.4.1 were tested using m=4. The codes Y₆ to Y₈ (that with a constraint length of 5 symbols) in Table 4.4.2 were tested for m=4 and 8, and finally, codes Y₉ to Y₁₂ (that with a constraint length of 6 symbols) in Table 4.4.3 were tested using m=8 and 16. From Tables 4.4.5 and 4.4.6 it is clear that, if system A employs a larger value of m, the corresponding average number of bit errors per burst for each code, at a given bit error rate (BER), can be significantly reduced.

From Table 4.4.1, code Y₁ appears to have the largest d² free but the poorest distance profile, when compared to other four codes of the same constraint length. From Table 4.4.4 the average number of bit errors per burst of this code, at each given BER, is also significantly larger than those of the other four codes. On the other hand, the average numbers of bit errors per burst for codes Y₂ and Y₅, at each given BER, appeared to be relatively smaller. But, the qₙ for Y₂ and Y₅ also grows faster with the path length n than the other three codes (see Table 4.4.1). The evidence here suggests that the distance profile of any code must have a bearing on its burst error performance. Among the codes Y₂ and Y₅, Y₅ appears to be more attractive while at the same time provides a better d² free and thus a larger asymptotic gain in tolerance to noise.
The second interesting point shown in Table 4.4.1 is that the distance profiles of the codes Y₃ and Y₄ are very similar. But, from Table 4.4.4, the average number of bit errors per burst for code Y₃, at each given BER, is relatively smaller than that for Y₄.

A possible explanation is that, as a net result of employing modulo-M arithmetic in coding, a large number of different combinations of \( s_i, s_{i-1}, \ldots, s_{i-K} \) must give the same complex-valued code symbol \( q_i \) (see equations 4.1.3 to 4.1.7). However, as far as the calculations and comparisons of cost values in the decoder is concerned, the expanded vectors corresponding to these combinations of \( s_i, s_{i-1}, \ldots, s_{i-K} \) are identical (if they have a similar previous cost). Consequently, if system A only holds a small number of stored vectors, then, during each symbol interval, only a small fraction of these combinations of \( s_i, s_{i-1}, \ldots, s_{i-K} \) will be examined in selecting the \( m \) "surviving" vectors. This may affect the probability of discarding the correct maximum likelihood vector and also the time taken to resume correct decoding.

The evidence here suggests that, although the majority of the achievable coding gains can be obtained by doubling the number of possible transmitted symbol values, an additional increase in the redundant signal level (without greatly reducing the \( d_{\text{free}}^2 \)), may effectively reduce the different combinations of \( s_i, s_{i-1}, \ldots, s_{i-K} \) that give the same code symbols \( \{q_i\} \). This might lead to a notable reduction in the average number of bit errors per burst, at a given BER.

Similar conclusions to those above can also be obtained
from the results shown in Tables 4.4.5 to 4.4.6, which present the burst error characteristics of the codes with K=4 and 5, respectively, in Tables 4.4.2 and 4.4.3.

The other interesting point in Tables 4.4.1 to 4.4.3 is that, except for code $Y_1$ which has a relative poor distance profile, the coding vectors of all other codes (those with good distance profiles) are such that $y_0=m/2$, and the first components $y_1$ in the corresponding distance profile is 4.0. Notice that $y_1$ is now equal to the $d_{\text{free}}^2$ of the uncoded BPSK signals (see equation 4.2.3). This ensures that the tolerance of any coded system to noise is, at least, as good as that of the corresponding uncoded system.

The distance profiles of various coding vectors are next examined for quaternary data symbols. However, it was later found that, unlike those with binary data symbols, the $y_h$ here generally grows very slowly with the path length $h$ for all the codes tested, as can be seen in Table 4.4.7. The table shows the distance profiles of the correlative codes in Table 4.3.1, together it shows those of the three new codes with $M=16$ or 64. The new codes, CC(6) to CC(8), obtained from a number of computer aided searches similar to those for binary data symbols, also appear to be superior in terms of minimum free unitary distances. Table 4.4.8 presents the minimum free unitary distance squared and asymptotic coding gain of each code in Table 4.4.7.

Figure 4.4.1 presents the performance results for codes CC(1), CC(6) and CC(7), using system A of 8 stored vectors ($m=8$). Figure 4.4.2 presents those for codes CC(2), CC(3) and
CC(8), using m=16. The results are shown as graphs of BER against the signal/noise ratio, $E_b/N_0$. The accuracy of each curve, over the range of bit error rates of $10^{-3}$ to $10^{-4}$, is ±0.2 dB. Tables 4.4.9 and 4.4.10 show, respectively, the burst error characteristics of these codes.

Code CC(5) was not used in these tests for the following reasons. As already mentioned earlier on (see Section 3.4), the decoding scheme of system A considered in this study can have 4 to 16 stored vectors only, in order to reduce the complexity of the decoder. With such a simple decoder, however, the performance code of CC(5) (that with a constraint length of 6 symbols) is expected to be very poor over the range of bit error rates of $10^{-1}$ to $10^{-4}$, despite the fact that it can provide a theoretical coding gain of 5.3 dB at very high signal/noise ratios.

From Figure 4.4.1 the three codes with 16 states appear to have similar tolerances to noise, over the range of the bit error rates of $10^{-3}$ to $10^{-4}$. From Table 4.4.9 the average number of bit errors per burst for each code here, at a given BER, is significantly large, even for code CC(7) (where M=64). This is perhaps due to the fact that, with the modulo-M arithmetic operation, there are relatively large numbers of different combinations of $s_1, s_{i-1}, \ldots, s_{i-k}$ giving the same code symbols ($q_i$), even when M increases to 64. From Figure 4.4.2 the code CC(8) slightly outperforms (about 0.2 dB) the other two codes, at a BER of $10^{-4}$, with M increasing to 64. However, such a small gain in tolerance to noise, compared to the increase in equipment complexity of the receiver, does not offer any gain in real terms.
Figures 11.11.3 and 11.11.11 contrast the performances of various arrangements of system A with those of the Viterbi decoder. It can be seen from Figure 11.11.3 that, for code CC(1) and m=8, system A degrades by about 0.7 dB, relative to the Viterbi decoder, at a BER of $10^{-4}$. From Figure 11.11.4, for code CC(2) and m=16, the degradation is as high as 0.8 dB.

A possible explanation for the general poor performance of the coded signals with system A is that, the correlative codes incorporated by the system here generally have a relatively poor distance profile. Hence, the minimum unitary distance between any two coded signal sequences that differ in their first symbols, increases very slowly with time even in the absence of noise.

Consequently, the costs of the corresponding two stored vectors of data symbols, where one of them is the correct maximum likelihood vector, will be quite similar over a long period of time in the presence of noise. In such a case, it is more likely for the decoder to falsely discard the correct vector, and when this happens, it will also take a significantly longer time to resume correct decoding.

The second factor is perhaps due to the large number of different vectors of data symbols that correspond to a very similar cost as those of the two vectors above. It is quite clear that the larger this number is, the more likely it is that the selected vector in system A will be one of these vectors, rather than the correct vector.
### Table 4.4.1: Distance profiles of correlative-level codes for binary data symbols, with K=3

<table>
<thead>
<tr>
<th>Code</th>
<th>Coding vector</th>
<th>Number of elements in the expanded signal set</th>
<th>Distance profile of the code, Ω</th>
<th>Minimum free unitary distance squared, $d^2_{\text{free}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y_1$</td>
<td>(11 6 4 8)</td>
<td>16</td>
<td>2.8 2.9 3.1 3.4</td>
<td>12.12</td>
</tr>
<tr>
<td>$Y_2$</td>
<td>(8 4 2 1)</td>
<td>16</td>
<td>4.0 6.0 6.7 6.7</td>
<td>6.74</td>
</tr>
<tr>
<td>$Y_3$</td>
<td>(4 2 1 3)</td>
<td>8</td>
<td>4.0 6.0 6.6 7.2</td>
<td>10.00</td>
</tr>
<tr>
<td>$Y_4$</td>
<td>(2 1 2 2)</td>
<td>4</td>
<td>4.0 4.0 6.0 8.0</td>
<td>12.00</td>
</tr>
<tr>
<td>$Y_5$</td>
<td>(8 4 2 5)</td>
<td>16</td>
<td>4.0 6.0 7.8 9.1</td>
<td>9.35</td>
</tr>
</tbody>
</table>

### Table 4.4.2: Distance profiles of correlative-level codes for binary data symbols, with K=4

<table>
<thead>
<tr>
<th>Code</th>
<th>Coding vector</th>
<th>Number of elements in the expanded signal set</th>
<th>Distance profile of the code, Ω</th>
<th>Minimum free unitary distance squared, $d^2_{\text{free}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y_6$</td>
<td>(2 0 1 2 2)</td>
<td>4</td>
<td>4.0 4.0 6.0 6.0 6.0</td>
<td>14.0</td>
</tr>
<tr>
<td>$Y_7$</td>
<td>(4 1 2 4 4)</td>
<td>8</td>
<td>4.0 4.6 6.6 6.6 7.2</td>
<td>13.76</td>
</tr>
<tr>
<td>$Y_8$</td>
<td>(16 8 12 2 19)</td>
<td>32</td>
<td>4.0 6.0 6.6 7.8 9.4</td>
<td>13.2</td>
</tr>
</tbody>
</table>
Table 4.4.3: Distance profiles of correlative-level codes for binary data symbols, with \( K=5 \)

<table>
<thead>
<tr>
<th>Code</th>
<th>Coding vector, ( \mathbf{Y} )</th>
<th>Number of elements in the expanded signal set, ( M )</th>
<th>Distance profile of the code, ( \Omega )</th>
<th>Minimum free unitary distance squared, ( d^2_{\text{free}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Y_9 )</td>
<td>(2 0 2 2 1 2)</td>
<td>4</td>
<td>4</td>
<td>4.0</td>
</tr>
<tr>
<td>( Y_{10} )</td>
<td>(8 4 7 1 2 6)</td>
<td>16</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>( Y_{11} )</td>
<td>(16 8 14 1 4 113)</td>
<td>32</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>( Y_{12} )</td>
<td>(32 16 24 4 8 31)</td>
<td>64</td>
<td>4</td>
<td>6</td>
</tr>
</tbody>
</table>

Table 4.4.4: Burst error characteristics of the codes shown in Table 4.4.1, with system \( A \) of 4 stored vectors

<table>
<thead>
<tr>
<th>Code</th>
<th>Number of elements in the expanded signal set, ( M )</th>
<th>Number of bit errors per burst at a given bit error rate, (approximately)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Y_1 )</td>
<td>16</td>
<td>34</td>
</tr>
<tr>
<td>( Y_2 )</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>( Y_3 )</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>( Y_4 )</td>
<td>4</td>
<td>15</td>
</tr>
<tr>
<td>( Y_5 )</td>
<td>16</td>
<td>4</td>
</tr>
</tbody>
</table>
Table 4.4.5: Burst error characteristics of the codes shown in Table 4.4.2, with system A of 4 and 8 stored vectors

<table>
<thead>
<tr>
<th>Code</th>
<th>Number of elements in the expanded signal set</th>
<th>Number of bit errors per burst at a given value of m and bit error rate, (approximately)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4 stored vectors</td>
<td>8 stored vectors</td>
</tr>
<tr>
<td></td>
<td>$\text{BER}=0.04, 0.008, 0.0006$</td>
<td>$\text{BER}=0.01, 0.003, 0.0005$</td>
</tr>
<tr>
<td>$Y_6$</td>
<td>4</td>
<td>28 36 44</td>
</tr>
<tr>
<td>$Y_7$</td>
<td>8</td>
<td>25 28 40</td>
</tr>
<tr>
<td>$Y_8$</td>
<td>32</td>
<td>16 16 40</td>
</tr>
</tbody>
</table>

Table 4.4.6: Burst error characteristics of the codes shown in Table 4.4.3, with system A of 8 and 16 stored vectors

<table>
<thead>
<tr>
<th>Code</th>
<th>Number of elements in the expanded signal set</th>
<th>Number of bit errors per burst at a given value of m and bit error rate, (approximately)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8 stored vectors</td>
<td>16 stored vectors</td>
</tr>
<tr>
<td></td>
<td>$\text{BER}=0.01, 0.002, 0.0005$</td>
<td>$\text{BER}=0.01, 0.001, 0.0003$</td>
</tr>
<tr>
<td>$Y_9$</td>
<td>4</td>
<td>66 63 -</td>
</tr>
<tr>
<td>$Y_{10}$</td>
<td>16</td>
<td>12 9 8</td>
</tr>
<tr>
<td>$Y_{11}$</td>
<td>32</td>
<td>13 10 10</td>
</tr>
<tr>
<td>$Y_{12}$</td>
<td>64</td>
<td>17 6 7</td>
</tr>
</tbody>
</table>
Table 4.4.7: Distance profiles of the correlative codes for quaternary data symbols

<table>
<thead>
<tr>
<th>Code</th>
<th>Coding vector,</th>
<th>Number of elements in the expanded signal set,</th>
<th>Distance profile of the code, ( \Omega )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( Y )</td>
<td>( M )</td>
<td>( \eta_1 )</td>
<td>( \eta_2 )</td>
</tr>
<tr>
<td>CC(1)</td>
<td>(2 1 2)</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>CC(2)</td>
<td>(2 1 2 2)</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>CC(3)</td>
<td>(2 1 1 2)</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>CC(4)</td>
<td>(2 1 1 -1 2)</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>CC(5)</td>
<td>(2 1 2 0 1 2)</td>
<td>8</td>
<td>2</td>
</tr>
<tr>
<td>CC(6)</td>
<td>(12 5 4)</td>
<td>16</td>
<td>2</td>
</tr>
<tr>
<td>CC(7)</td>
<td>(16 28 15)</td>
<td>64</td>
<td>2</td>
</tr>
<tr>
<td>CC(8)</td>
<td>(16 11 4 15)</td>
<td>64</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 4.4.8: Minimum free distances and asymptotic coding gains of the codes shown in Table 4.4.7

<table>
<thead>
<tr>
<th>Code</th>
<th>Coding vector, ( Y )</th>
<th>Number of elements in the expanded signal set, ( M )</th>
<th>Minimum free unitary distance squared, ( d_{\text{free}}^2 )</th>
<th>Asymptotic coding gain over uncoded QPSK signals, ( G_c ) (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC(1)</td>
<td>(2 1 2)</td>
<td>8</td>
<td>4.5858</td>
<td>3.6</td>
</tr>
<tr>
<td>CC(2)</td>
<td>(2 1 2 2)</td>
<td>8</td>
<td>5.1716</td>
<td>4.1</td>
</tr>
<tr>
<td>CC(3)</td>
<td>(2 1 1 2)</td>
<td>8</td>
<td>5.1716</td>
<td>4.1</td>
</tr>
<tr>
<td>CC(4)</td>
<td>(2 1 1 -1 2)</td>
<td>8</td>
<td>5.757</td>
<td>4.6</td>
</tr>
<tr>
<td>CC(5)</td>
<td>(2 1 2 0 1 2)</td>
<td>8</td>
<td>7.1716</td>
<td>5.3</td>
</tr>
<tr>
<td>CC(6)</td>
<td>(12 5 4)</td>
<td>16</td>
<td>4.152</td>
<td>3.2</td>
</tr>
<tr>
<td>CC(7)</td>
<td>(16 28 15)</td>
<td>64</td>
<td>4.726</td>
<td>3.7</td>
</tr>
<tr>
<td>CC(8)</td>
<td>(16 11 4 15)</td>
<td>64</td>
<td>5.013</td>
<td>4.0</td>
</tr>
</tbody>
</table>
Table 4.4.9: Burst error characteristics of various 16-state correlative codes with system A of 8 stored vectors

<table>
<thead>
<tr>
<th>Code</th>
<th>Number of states, $u_K$</th>
<th>Number of elements in the expanded signal set, M</th>
<th>Average number of bit errors per burst at a given bit error rate, (approximately) BER=0.05, 0.02, 0.006, 0.002</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC(1)</td>
<td>16</td>
<td>8</td>
<td>56 39 40 19</td>
</tr>
<tr>
<td>CC(6)</td>
<td>16</td>
<td>16</td>
<td>42 34 33 28</td>
</tr>
<tr>
<td>CC(7)</td>
<td>16</td>
<td>64</td>
<td>37 35 27 28</td>
</tr>
</tbody>
</table>

Table 4.4.10: Burst error characteristics of various 64-state correlative codes with system A of 16 stored vectors

<table>
<thead>
<tr>
<th>Code</th>
<th>Number of states, $u_K$</th>
<th>Number of elements in the expanded signal set, M</th>
<th>Average number of bit errors per burst at a given bit error rate, (approximately) BER=0.1, 0.06, 0.02, 0.002</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC(2)</td>
<td>64</td>
<td>8</td>
<td>&gt;100 &gt;100 74 41</td>
</tr>
<tr>
<td>CC(3)</td>
<td>64</td>
<td>8</td>
<td>&gt;100 &gt;100 86 92</td>
</tr>
<tr>
<td>CC(8)</td>
<td>64</td>
<td>64</td>
<td>79 61 - 20</td>
</tr>
</tbody>
</table>
Fig. 4.4.1: Performances of correlative coded MPSK signals with system A of 8 stored vectors

Legend
△ Uncoded QPSK
× CC[1]; 8PSK
□ CC[6]; 16PSK
◊ CC[7]; 64PSK

Common attributes: 16-state codes; n=48
Fig. 4.4.2: Performances of correlative coded MPSK signals with system A of 16 stored vectors

Legend
- Uncoded QPSK
- CC[2]; 8PSK
- CC[3]; 8PSK
- CC[8]; 64PSK

Common attributes: 64-state codes; n=48
Fig. 4.4.3: Performance comparison of the two decoding schemes with code CC[1]
Fig. 4.4.4: Performance comparison of the two decoding schemes with code CC[2]

Legend
- Uncoded QPSK
- System A; m=16
- Viterbi; $2^5=64$

Common attributes: 8PSK; n=48
CHAPTER FIVE

NOISE-ADAPTIVE NEAR-MAXIMUM LIKELIHOOD DECODING SCHEMES
The decoding algorithms considered so far perform the same operation over each symbol interval. This means that each decoded data symbol is determined with the same amount of computation, regardless the noise level in transmission. It is quite obvious that part of this computational effort must be wasteful during the period of time when the noise is low, since one naturally expects that, if it is less noisy in transmission, fewer operations per decoded data symbol would be sufficient to achieving correct decoding. Clearly, if the amount of computation per decoded data symbol can be made appropriately adjustable to the prevailing noise level, such that a larger amount of computation is involved only when the noise is relatively high, then a substantial reduction in the decoder complexity may be achieved with little reduction in tolerance to noise.

The first two sections of this chapter describe two such noise-adaptive decoding schemes proposed for coded 8PSK signals. The two decoders are called system A1 and system A2, for the fact that they both are the modified versions of system A described in Section 3.4. The modifications consist the use of a variable number of stored vectors from one decoding process to another, together with a more flexible arrangement for generating these vectors.

The penalty paid for using such a noise-adaptive scheme is the two buffer stores required at the receiver. One of the buffer stores is needed at the decoder output to provide a continuous, constant-rate, sequence of decoded data symbols, whereas the other buffer store is required to hold a number of the received samples at the decoder input. The buffer store
requirement for systems A1 and A2 is further discussed in Section 5.3.

5.1 SYSTEM A1

Now, consider the costs of the stored vectors in system A, compared with the zero-cost attributed to the lowest cost vector. The cost of any stored vector here is clearly a measure of how likely it is that the given vector stores the last n-components of the correct maximum likelihood sequence of data symbols. A low cost implies high likelihood, and a high cost implies low likelihood. Those vectors with very high costs can be discarded from the subsequental decoding processes without seriously affecting the probability of rejecting the correct vector of maximum likelihood data symbols.

The normal operation of system A is simply as follows. For a given number of stored vectors (that is m), each stored vector is expanded, and the associated costs are calculated, resulting in 4m expanded vectors and their costs. In the next stage, m lowest cost vectors are selected from the expanded vectors, and they are stored for the next decoding process. The results of extensive computer simulation tests presented in Section 3.4 have suggested that the performance of system A, if employing a sufficient large value of m, approaches that of the corresponding optimum decoder. Take the example in Figure 3.4.1, which shows the performances of different arrangements of system A incorporating code UC(3) (a 16-state code).
Clearly, when $m=16$, the tolerances of the system to noise are very close to those of the corresponding Viterbi decoder, over the range of bit error rates of $10^{-3}$ to $10^{-4}$.

Now, consider the case where system A employs a large value of $m$, as in the example above. It is certainly possible at any time for some of these $m$ stored vectors to have a very high cost, particularly if the noise in transmission is low. These high cost vectors have a low possibility of being correct and are therefore not worthy of any further consideration by the decoder. Clearly, the elimination of these vectors will not seriously affect the probability of rejecting the correct maximum likelihood vector. An immediate suggestion here is that, a cost-effective decoder, rather than selecting a fixed number of vectors with lowest costs, should only store those vectors whose costs are below a threshold level, say $C_m$. For a given signal/noise ratio, $C_m$ will be chosen as a cost-effective compromise between the implementation complexity and the error rate performance of the decoder. The smaller is the $C_m$, the fewer it will be the number of stored vectors employed in each decoding process and thus the fewer operations per decoded data symbol. However, this also means a higher possibility of rejecting the correct maximum likelihood vector and consequently a higher probability of decoding error.

Notice that a forced decision in selecting the stored vectors, using the threshold $C_m$, does not generally guarantee that the correct maximum likelihood vector is always included, and therefore the optimality of the decoder is not guaranteed even at high signal/noise ratios (see reference 91 for details on the optimality of near-maximum likelihood decoders). How-
ever, one could generally expect the performance of such a system to be close to that of the optimum decoder, so long as the threshold value $C_m$ is appropriately chosen, and a finite period of time is taken for the correct vector to remerge, should it at some stage be rejected from the decoder. But this conjecture must be verified by computer simulation tests.

A particular implementation of such a scheme, called system A1, is now described for rate-2/3 convolutionally coded 8PSK signals. The description begins with a description of the received signals. The decoder is then described in terms of its output, a set of stored vectors and costs. The algorithm repeated during each decoding process, which uses these stored values to generate the decoded data symbol $s'_i$ is also given. System A1 employs two thresholds, $C_m$ and $C'_m$, where $C'_m > C_m$, in selecting stored vectors. Both $C_m$ and $C'_m$ are positive constants and must be determined experimentally for any given code. The model of the data transmission system is given in Section 2.1 and the rate-2/3 convolutional codes for 8PSK signals are described in Section 3.2.

Following equation 2.1.11, the received sample fed to the decoder, at time instant $t=iT$, is given by

$$r_i = q_i + w_i$$  \hspace{1cm} (5.1.1)

$T$ is the symbol interval in seconds. $q_i$ is the complex-valued code symbol fed to the modulator at the transmitter and $w_i$ is a sample value of the Gaussian noise waveform $w(t)$ at the demodulator output. The real and imaginary parts of the $\{w_i\}$ are statistically independent Gaussian random variables with
zero mean and fixed variance $\sigma^2$.

The decoder operates on the sequence of received samples \( \{ r_i \} \). The corresponding output from the decoder comprises the 2-component vectors

\[ b'_i = [ b'_i(1), b'_i(2) ] \] (5.1.2)

where \( b'_i(1) \) and \( b'_i(2) \) are the decoded binary data symbols. In the absence of noise, \( b'_i = b_i \) for each \( i \), where \( b_i \) is the corresponding vector of binary data symbols fed to the coder at the transmitter, at time \( t=iT \). \( b'_i \) uniquely determines the decoded data symbol \( s'_i \), whose possible values are 0, 1, 2 and 3, respectively, the relationship between \( b'_i \) and \( s'_i \) being the same as that between \( b_i \) and \( s_i \). \( s_i \) is, of course, the quaternary data symbol determined by the vector \( b_i \).

Just prior to the receipt of the sample \( r_i' \), system A1 holds in store a set of \( m_i \) different \( n \)-component vectors of possible data symbols,

\[ X_{i-1} = [ x_{i-n}, x_{i-n+1}, \ldots, x_{i-1} ] \] (5.1.3)

where \( x_i \) is a possible value of the vector \( b_i \) and thus can take on any of the four values, 00, 01, 10 and 11. \( n \) is an appropriate positive integer, representing the total detection delay measured in symbols.

Associated with each vector \( X_{i-1} \) is stored its cost \( c_{i-1} \). In general, the cost \( c_i \) of any particular stored vector of data symbols is defined to be the unitary distance squared
between the sequences of received samples \( \{ r_i \} \) and the complex-valued code symbols \( \{ p_i \} \) corresponding to the given vector of data symbols. Thus, \( p_i \) is a possible value of \( q_i \), and by definition,

\[
c_{i-1} = \sum_{h=0}^{i-1} |r_h - p_h|^2
\]

(5.1.4)

Furthermore, in order to avoid an unacceptable increase in the costs, over a long message, the lowest \( c_{i-1} \) is always subtracted from each stored cost before the decoder starts to operate on the current received sample \( r_i \). Thus the decoder reduces its smallest cost to zero without changing the differences between the various stored costs.

The description given so far for system A1 is exactly the same as that for system A of Section 3.4. However, with system A1, only the vectors \( \{ X_{i-1} \} \) for which \( c_{i-1} < C_m \) have now been selected and stored in the decoding of the previous received sample \( r_{i-1} \). The total number of vectors held by the decoder before the start of \( (i+1) \)th decoding process is, of course, \( m_i \). The maximum permitted value of \( m_i \), known as \( M_m \), is chosen to limit the maximum number of operations per decoded data symbol in system A1. The decoder therefore selects up to \( M_m \) vectors with smallest costs in decoding each data symbol.

On the receipt of the sample \( r_i \), each stored vector \( X_{i-1} \) is expanded to form four \( (n+1) \)-component vectors \( \{ Z_i \} \), where

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

(5.1.5)

The components \( x_{i-n}, x_{i-n+1}, \ldots, x_{i-1} \) are here as in the
original vector $X_{i-1}$ of equation 5.1.3, and the last component $x_i$ takes on its four possible values in the four $\{Z_i\}$. The corresponding cost $c_i$, of each expanded vector $Z_i$, is calculated as follows. The binary code symbols $\{v_i(j)\}$ are first determined by

$$v_i(j) = \sum_{h=0}^{k1} x_{i-h}(1) g_h(1,j) + \sum_{h=0}^{k2} x_{i-h}(2) g_h(2,j)$$  \hspace{1cm} (5.1.6)$$

for $j=1, 2$ and $3$. In equation 5.1.6, modulo-2 addition is assumed, and the $\{g_h(i,j)\}$ together define sub-generators of the given rate-2/3 convolutional code. The sub-generators of the various rate-2/3 codes considered in this section are shown in Table 3.2.1. Furthermore, $g = k1 + k2$, where $g$ is the code memory in bits. Now,

$$z_i = 4v_i(1) + 2v_i(2) + v_i(3)$$  \hspace{1cm} (5.1.7)$$

and

$$p_i = \cos(\pi z_i/4) + j\sin(\pi z_i/4)$$  \hspace{1cm} (5.1.8)$$

where $j = \sqrt{-1}$. The cost $c_i$ is now determined by

$$c_i = c_{i-1} + |p_i - p_1|^2$$

$$= c_{i-1} + (\text{Re}(r_i - p_1))^2 + (\text{Im}(r_i - p_1))^2$$  \hspace{1cm} (5.1.9)$$

where $\text{Re}(.)$ and $\text{Im}(.)$ represent, respectively, the real and imaginary parts of the corresponding quantity. For a given value of $m_i$, there are now $4m_i$ expanded vectors $\{Z_i\}$ together with their costs $\{c_i\}$.

The threshold $C'_m$ is next used to eliminate the vectors
\{Z_i\}$ with relatively high costs. This is done by immediate discarding of any vector $Z_i$ for which $c_i > c'_m$. This is called the initial selection procedure. This procedure is needed to eliminate the least likely expanded vectors at very early state of the decoding process, so that fewer comparisons are required, on average, in the later selection of $m_{i+1}$ stored vectors.

The decoder now selects the minimum cost vector from the "surviving" vectors $\{Z_i\}$, and takes its first component $x_{i-n}$ as the decoded value $u'_{i-n}$ of $u_{i-n}$, exactly as in system A. All vectors for which $x_{i-n} u'_{i-n}$ are then discarded (this has been known as the anti-merging procedure), and the first components of the remaining vectors are also omitted to give the corresponding $n$-component vectors $\{X_i\}$ (without changing their costs).

The $m_{i+1}$ vectors stored for the next decoding process are determined in the following final selection procedure. At first, from costs $\{c_i\}$ of the vectors $\{X_i\}$ is subtracted the lowest cost, thus reducing the smallest cost to zero. The decoder starts by selecting the zero cost vector, and then it selects the vector with the second smallest cost, then the third, and so on, until either the vectors $\{X_i\}$ for which $c_i < c_m$ have all been stored, or else $m_i$ exceeds $M_m$. In the latter case, only the $M_m$ vectors $\{X_i\}$ with lowest costs $\{c_i\}$ are stored. The decoder is now ready for the next received sample.

A similar starting procedure as that for system A is employed in system A1. At the start of transmission, the decoder holds a single stored vector with zero cost (which should, if at all possible, be correct). In this way, only
after a few decoding processes, all the stored vectors will be those derived from this vector and will all be different.

The performance results of the various arrangements of system A1, obtained from a number of computer simulation tests, are presented in Figures 5.1.1 to 5.1.6, where they are shown as graphs of bit error rate (BER) against the signal/noise ratio, $E_b/N_0$. $E_b$ is the average signal energy per data bit and $(1/2)N_0$ is the two-sided power spectral density of the additive white Gaussian noise. Appendix B5 presents an example of the computer programs used in the tests. The accuracy of each curve here, in the range of bit error rates of $10^{-3}$ to $10^{-4}$, is ±0.2 dB.

Throughout these tests, the maximum permitted number of operations per decoded data symbol (that is, $M_m$) is chosen to be at least the same as that for the corresponding Viterbi decoder, and in many cases, $M_m$ is twice of that in the Viterbi decoder. This is reasoned as follows. Under a very noisy condition where a large number of vectors are used to resume correct decoding, system A1 will operate as the conventional system A with a fixed number of $M_m$ stored vectors. However, as already shown in Section 3.4, if system A uses the same number of stored vectors as the corresponding Viterbi decoder, its performance also becomes close to that of the Viterbi decoder. Therefore, it is quite adequate to have

$$M_m > 2g$$

where $g$ is the memory of the given code in bits ($g = k_1 + k_2$).

The second threshold level $C'_m$ employed in system A1 is
determined in the following manner. Let the lowest value in costs \( \{c_1\} \) of the expanded vectors \( \{Z_i\} \) be \( \Delta_i \), during the \((i+1)\)th decoding process, then any \( Z_i \) for which

\[
(c_1 - \Delta_i) \geq C_m
\]  
(5.1.11)

will be discarded from the decoder later in the final selection procedure. This means that, for a given value of \( \Delta_i \), any vector \( Z_i \) for which

\[
c_1 > (C_m + \Delta_i)
\]  
(5.1.12)

need not, in fact, be considered any further by the decoder. These vectors can therefore be eliminated in the initial selection procedure, using the threshold level \( C'_m \). Ideally, for a given value of \( \Delta_i \),

\[
C'_m = C_m + \Delta_i
\]  
(5.1.13)

would be sufficient. The results from a number of computer simulation tests suggest that, at reasonably high signal/noise ratios, the probability that \( \Delta_i \) is greater than 0.5 is below 10\(^{-4}\). Therefore, in most cases, \( C'_m \) is chosen to be

\[
C'_m = C_m + 0.5
\]  
(5.1.14)

Figure 5.1.1 shows the performance results for the various arrangements of system A1 with code UC(3) (a 16-state
code), and Table 5.1.1 outlines the performance of each system, together with the corresponding equipment complexity required, at different bit error rates. The latter complexity is shown by the corresponding average number of stored vector per decoded data symbol. \( \tilde{m}_1 \), at each given BER, averaged over the transmission of 100,000 data symbols. In all cases, \( n=48 \) (which is the same as that for the corresponding system A), \( M_m=32 \) (which is twice of that in the corresponding Viterbi decoder), and

\[
C'_m = C_m + 0.5 \quad (5.1.15)
\]

From Table 5.1.1 the parameter \( C_m \) is clearly a compromise between the performance of the decoder and the average number of operations required per decoded data symbol, for a given BER. Take the example with \( C_m=1.0 \), at a BER of \( 10^{-3} \). The average number of stored vectors here is about 1.6, the gain in tolerance to noise being 0.5 dB, over the uncoded QPSK system. But, with \( C_m=1.4 \), a coding gain of 1.55 dB is achieved with \( \tilde{m}_1=4.4 \). From \( C_m=1.4 \) to \( C_m=1.6 \), an additional gain of 0.15 dB is obtained with \( \tilde{m}_1 \) increasing from 4.4 to 7.0.

Figures 5.1.2 and 5.1.3 present the test results with codes UC(4) (a 32-state code) and UC(5) (a 64-state code), respectively, whereas Tables 5.1.2 and 5.1.3 outline the performances and equipment complexities of these systems at different bit error rates. Similar conclusions to those above can also be obtained from these results.

Figure 5.1.4 summarizes the performances of those arrangements of system A1 that, following the results of extensive computer simulation tests so far, appear to be most-
ly cost-effective for the three rate-2/3 convolutionally coded 8PSK signals (that are generated by codes UC(3), UC(4), and UC(5)). Table 5.1.4 outlines the performance and equipment complexity of each system in Figure 5.1.4, at different bit error rates. Table 5.1.5 shows their burst error characteristics and Table 5.1.6 presents the system parameters.

The optimisation of system parameters for each arrangement in Table 5.1.6 is made, for the given code, as a cost-effective compromise between the gain in tolerance to noise and the decoder complexity. The criterion used to define this compromise is that the degradation in tolerances to noise of approximately 0.2 dB, compared to the Viterbi decoder incorporating the same code, is achieved over the range of bit error rates of $10^{-3}$ to $10^{-4}$, for as few average numbers of stored vectors per decoded data symbol as possible. The other factor which must also have a bearing on the decoder complexity is the total number of cost evaluations performed in decoding each data symbol. Following equation 5.1.9, each of these calculations involves two squaring operations. The total number of cost evaluations in $(i+1)^{th}$ decoding process is denoted by $f_i$. Clearly, with system A1, $f_i = 4m_i$ for each $i$. In Table 5.1.4, the corresponding average number of cost evaluations per decoded data symbol $\bar{f}_i$, averaged over the transmission of 100,000 data symbols at each given BER, is also shown for each system.

From Table 5.1.4 it is clear that, at a BER of $10^{-4}$, the given arrangements of system A1 gain 2.45 dB, 2.75 dB and 3.0 dB, respectively, over the uncoded QPSK system, the average numbers of stored vectors here being 2.9, 5.5, and 10.0. These
certainly compare well with those of the corresponding Viterbi
decoder, which are 2.7 dB, 3.0 dB and 3.25 dB, respectively,
the numbers of stored vectors being 16, 32, and 64 (see Table
3.3.1). The tolerance of system A1 to noise here is only
reduced by up to about 0.25 dB, but the number of stored
vectors per decoded data symbol, on average, is reduced by up
to three-quarters. The comparison here is, of course, based on
the assumption that the two decoding schemes with the same
number of stored vectors (or average number of stored vectors)
have approximately the same equipment complexity.

Figure 5.1.5 presents the performance results for the
correlative coded 8PSK signals of Section 4.3, using the
decoding scheme of system A1. Table 5.1.7 outlines the
performances and equipment complexities for each system at
different bit error rates, whereas Table 5.1.8 presents the
system parameters. The two codes used in the tests are codes
CC(2) and CC(3) in Table 4.3.1, each having 64 states. The
operation of the decoder is very similar to that just described
for convolutional codes, except a few modifications now to be
described.

Firstly, each stored vector \(X_{i-1}\) here holds \(n\) possible
values of data symbols \(\{s_i\}\), where \(s_i\) has four possible values
0, 1, 2 and 3. Therefore each component \(x_i\) in equation 5.1.3
can now take on any of these four values. Secondly, the cost
\(c_i\) of any expanded vector \(Z_i\) in equation 5.1.4 is determined as
follows. For a given coding vector of

\[ Y = [y_0 \ y_1 \ y_2 \ y_3] \] (5.1.16)
where \( Y = [2 \ 1 \ 2 \ 2] \) for CC(2) and \( Y = [2 \ 1 \ 1 \ 2] \) for CC(3), the cost \( c_i \) of any particular expanded vector \( Z_i \) is given by

\[
z_i = \sum_{h=0}^{3} x_i - hY h \mod 8 \quad (5.1.17)
\]

\[
p_i = \cos(\pi z_i/4) + jsin(\pi z_i/4) \quad (5.1.18)
\]

and

\[
c_i = c_{i-1} + |p_i - p_1|^2 \quad (5.1.19)
\]

Finally, the corresponding \( x_{i-n} \) in the lowest cost vector \( Z_i \) must give the decoded data symbol value \( s'_{i-n} \).

It is interesting to see from Figure 5.1.5 that, over the range of bit error rates of \( 10^{-3} \) to \( 10^{-11} \), the tolerances to noise of system A1 with code CC(2), using \( C_m = 1.8 \) and \( C'_m = 2.3 \), are within 0.1 dB of those using \( C_m = 2.0 \) and \( C'_m = 3.0 \), but the corresponding \( \overline{m} \) is only two-thirds of the latter system (see Table 5.1.7). Clearly, further increasing \( \overline{m} \) and hence the amount of computation per decoded symbol does not always offer a notable improvement in tolerance to noise.

Figure 5.1.6 contrasts the performance of a correlative coded 8PSK signal with that of a convolutionally coded signal, using the same arrangement of system A1. The two codes under comparison are codes CC(2) and UC(5), each having 64 states. Although the asymptotic coding gain of code CC(2) is 0.5 dB inferior to that of code UC(5) (see Table 4.3.1), the tolerances of both coded systems to noise are very close over the range of bit error rates of \( 10^{-3} \) to \( 10^{-4} \), as can be seen in Figure 5.1.6.

A very important conclusion from the results shown so far is that the average number of stored vectors in system A1
(that is, $\overline{m}_1$) decreases as the signal/noise ratio increases. This means that the decoder uses fewer stored vectors at the higher signal/noise ratios, so that the number of operations per decoded data symbol is adjusted adaptively to suit the noise level in transmission. Thus system A1 may occasionally uses several additional vectors without, however, significantly increasing $\overline{m}_1$.

Notice that, in practice, the two thresholds $C_m$ and $C'_m$, and the quantities $M_m$ and $n$, employed in system A1, should be optimised for the most likely signal/noise ratio, for each particular code under consideration. Furthermore, supposing $C_m$ can be adjusted as such that it appropriately increases as the signal/noise ratio increases, a very similar amount of reduction in the average number of stored vectors can be achieved over the range of bit error rates of $10^{-3}$ to $10^{-4}$. 


Table 5.1.1: Performances and equipment complexities of different arrangements of system A1 with code UC(3)

<table>
<thead>
<tr>
<th>$C_m$</th>
<th>BER=0.01</th>
<th>0.001</th>
<th>0.0001</th>
<th>BER=0.01</th>
<th>0.001</th>
<th>0.0001</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>-0.95</td>
<td>0.5</td>
<td>1.5</td>
<td>2.5</td>
<td>1.6</td>
<td>&lt;1.6</td>
</tr>
<tr>
<td>1.2</td>
<td>-0.25</td>
<td>1.3</td>
<td>2.25</td>
<td>4.5</td>
<td>2.5</td>
<td>&lt;2.3</td>
</tr>
<tr>
<td>1.4</td>
<td>0.0</td>
<td>1.55</td>
<td>2.45</td>
<td>6.9</td>
<td>4.4</td>
<td>2.9</td>
</tr>
<tr>
<td>1.5</td>
<td>0.1</td>
<td>1.65</td>
<td>2.55</td>
<td>9.5</td>
<td>6.5</td>
<td>3.5</td>
</tr>
<tr>
<td>1.6</td>
<td>0.2</td>
<td>1.7</td>
<td>2.65</td>
<td>11.0</td>
<td>7.0</td>
<td>4.0</td>
</tr>
</tbody>
</table>

Common attributes: 8PSK; $M_m=32; n=48; C'_m=C_m+0.5$

Table 5.1.2: Performances and equipment complexities of different arrangements of system A1 with code UC(4)

<table>
<thead>
<tr>
<th>$C_m$</th>
<th>BER=0.01</th>
<th>0.001</th>
<th>0.0001</th>
<th>BER=0.01</th>
<th>0.001</th>
<th>0.0001</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5</td>
<td>0.0</td>
<td>1.7</td>
<td>2.7</td>
<td>12.0</td>
<td>6.5</td>
<td>4.5</td>
</tr>
<tr>
<td>1.6</td>
<td>0.1</td>
<td>1.75</td>
<td>2.75</td>
<td>13.0</td>
<td>7.8</td>
<td>5.5</td>
</tr>
<tr>
<td>1.7</td>
<td>0.2</td>
<td>1.85</td>
<td>2.85</td>
<td>17.0</td>
<td>10.0</td>
<td>7.0</td>
</tr>
</tbody>
</table>

Common attributes: 8PSK; $M_m=64; n=48; C'_m=C_m+0.5$
Table 5.1.3: Performances and equipment complexities of different arrangements of system A1 with code UC(5)

<table>
<thead>
<tr>
<th>Gain in tolerance to noise</th>
<th>Average number of stored vectors at a given BER, $\tilde{m}_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_m$ BER=0.01, 0.001, 0.0001</td>
<td>BER=0.01, 0.001, 0.0001</td>
</tr>
<tr>
<td>---------------------------</td>
<td>----------------------------------------------------------</td>
</tr>
<tr>
<td>1.7</td>
<td>0.05, 1.75, 2.8, 17.5, 9.0, 6.0</td>
</tr>
<tr>
<td>1.8</td>
<td>0.2, 1.9, 2.9, 19.6, 11.7, 7.7</td>
</tr>
<tr>
<td>1.9</td>
<td>0.25, 1.9, 3.0, 23.7, 14.8, 10.0</td>
</tr>
</tbody>
</table>

Common attributes: 8PSK; $M_m=64$; $n=64$; $C'_m=C_m*0.5$

Table 5.1.4: Performances and equipment complexities of the various arrangements of system A1 in Figure 5.1.4

<table>
<thead>
<tr>
<th>Code</th>
<th>Gain in tolerance to noise, $\tilde{m}_i$, $\tilde{r}_i$ (dB)</th>
<th>Gain in tolerance to noise, $\tilde{m}_i$, $\tilde{r}_i$ (dB)</th>
<th>Gain in tolerance to noise, $\tilde{m}_i$, $\tilde{r}_i$ (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC(3)</td>
<td>6.9, 27.6, 0.0, 4.4, 17.6, 1.55</td>
<td>2.9, 11.6, 2.45</td>
<td></td>
</tr>
<tr>
<td>UC(4)</td>
<td>13.0, 52.0, 0.1, 7.8, 31.2, 1.75</td>
<td>5.5, 22.0, 2.75</td>
<td></td>
</tr>
<tr>
<td>UC(5)</td>
<td>23.7, 95.3, 0.25, 14.8, 60.1, 1.9</td>
<td>10.0, 39.7, 3.0</td>
<td></td>
</tr>
</tbody>
</table>
Table 5.1.5: Burst error characteristics of the various arrangements of system A1 in Figure 5.1.4

Average number of bit errors per burst at a given bit error rate, (approximately)

<table>
<thead>
<tr>
<th>Code</th>
<th>BER=0.05, 0.01, 0.005, 0.001, 0.0005, 0.0001</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC(3)</td>
<td>25  19  13  14  8  6</td>
</tr>
<tr>
<td>UC(4)</td>
<td>32  22  19  16  17  7</td>
</tr>
<tr>
<td>UC(5)</td>
<td>25  16  12  11  10  5</td>
</tr>
</tbody>
</table>

Table 5.1.6: System parameters for the various arrangements of system A1 shown in Figure 5.1.4

<table>
<thead>
<tr>
<th>Code</th>
<th>Number of states, $2^g$</th>
<th>$M_m$</th>
<th>$n$</th>
<th>$C_m$</th>
<th>$C'_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC(3)</td>
<td>16</td>
<td>32</td>
<td>48</td>
<td>1.4</td>
<td>1.9</td>
</tr>
<tr>
<td>UC(4)</td>
<td>32</td>
<td>64</td>
<td>48</td>
<td>1.6</td>
<td>2.1</td>
</tr>
<tr>
<td>UC(5)</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>1.9</td>
<td>2.4</td>
</tr>
</tbody>
</table>
Table 5.1.7: Performances and equipment complexities of the various arrangements of system A1 in Figure 5.1.5

<table>
<thead>
<tr>
<th>Code</th>
<th>$\tilde{m}_i$</th>
<th>$\tilde{f}_i$ (dB)</th>
<th>$\tilde{m}_i$</th>
<th>$\tilde{f}_i$ (dB)</th>
<th>$\tilde{m}_i$</th>
<th>$\tilde{f}_i$ (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC(2)</td>
<td>19.8</td>
<td>80</td>
<td>12</td>
<td>48</td>
<td>8</td>
<td>32</td>
</tr>
<tr>
<td>CC(2)</td>
<td>27.5</td>
<td>114</td>
<td>19</td>
<td>76</td>
<td>12</td>
<td>48</td>
</tr>
<tr>
<td>CC(3)</td>
<td>27</td>
<td>112</td>
<td>19</td>
<td>76</td>
<td>14</td>
<td>52</td>
</tr>
</tbody>
</table>

Common attribute: correlative coded QPSK;

Table 5.1.8: System parameters for the various arrangements of system A1 shown in Figure 5.1.5

<table>
<thead>
<tr>
<th>Code</th>
<th>Number of states, $2^G$</th>
<th>$M_m$</th>
<th>$n$</th>
<th>$C_m$</th>
<th>$C'_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC(2)</td>
<td>$64$</td>
<td>$64$</td>
<td>$64$</td>
<td>1.8</td>
<td>2.3</td>
</tr>
<tr>
<td>CC(2)</td>
<td>$64$</td>
<td>$64$</td>
<td>$64$</td>
<td>2.0</td>
<td>3.0</td>
</tr>
<tr>
<td>CC(3)</td>
<td>$64$</td>
<td>$64$</td>
<td>$64$</td>
<td>2.0</td>
<td>3.0</td>
</tr>
</tbody>
</table>
Fig. 5.1.1: Performance of system A1 with UC[3]
Fig. 5.1.2: Performance of system A1 with UC[4]

Legend
- Uncoded QPSK
- Cm = 1.5
- Cm = 1.6
- Cm = 1.7

Common attributes: 8PSK; Mm = 64; n = 48; Cm = Cm + 0.5
Fig. 5.1.3: Performance of system A1 with UC[5]

**Legend**
- △ Uncoded QPSK
- × \(C_m = 1.7\)
- □ \(C_m = 1.8\)
- ⊗ \(C_m = 1.9\)

Common attributes: 8PSK; \(M_m = 64\); \(n = 64\); \(C_m = C_m + 0.5\)
Fig. 5.1.4: Performances of rate-2/3 convolutionally coded 8PSK signals with system A1 decoding

Legend
- △ Uncoded QPSK
- ✗ UC[3]; Mm=32; n=48; Cm=1.4
- ■ UC[4]; Mm=64; n=48; Cm=1.6
- ✧ UC[5]; Mm=64; n=64; Cm=1.9

Common attribute: Cm=Cm+0.5
Fig. 5.1.5: Performances of correlative coded 8PSK signals with system A1 decoding

Legend
- Uncoded QPSK
- CC[2]; C_m=1.8; C_m'=2.3
- CC[2]; C_m=2.0; C_m'=3.0
- CC[3]; C_m=2.0; C_m'=3.0

Common attributes: M_m=64; n=64;
Fig. 5.1.6: Performance comparison of different coded 8PSK signals with system A1 decoding

Legend
- △ Uncoded QPSK
- × CC[2]: correlative code
- □ UC[5]: convolutional code

Common attributes: \( M_m=64; n=64; C_m=1.8; C'_m=2.3 \)
5.2 **SYSTEM A2**

System A2 is the second noise-adaptive decoding scheme developed for coded 8PSK signals. The operation of system A2 is very similar to the system A1 presented in Section 5.1, except for the further modifications now to be described.

On the receipt of the sample \( r_1 \), each vector \( X_{i-1} \) (see equation 5.1.3) is expanded to form \( J_1 \) vectors \( \{ Z_i \} \) having different values of \( x_i \), where \( J_1 \) takes on the value 4, 2, or 1, depending upon the cost \( c_{i-1} \) of the corresponding vector \( X_i \). The value of \( J_1 \) is determined as follows.

\[
J_1 = 4 \quad \text{if} \quad c_{i-1} \leq \text{ex}_1 \\
J_1 = 2 \quad \text{if} \quad \text{ex}_1 < c_{i-1} \leq \text{ex}_2 \\
J_1 = 1 \quad \text{if} \quad \text{ex}_2 < c_{i-1} \\
\]

(5.2.1)

where \( \text{ex}_1 \) and \( \text{ex}_2 \) are positive constants, and \( \text{ex}_1 < \text{ex}_2 \). Both constants must determined experimentally for any given code.

This process of adjustable expansion may be illustrated by the simple diagram given in Figure 5.2.1. The stored vectors \( \{ X_{i-1} \} \) are here placed in order of increasing cost, the cost of the first stored vector being zero. The vectors are classed into three groups, depending upon whether \( c_{i-1} \leq \text{ex}_1 \), \( \text{ex}_1 < c_{i-1} \leq \text{ex}_2 \), or \( \text{ex}_2 < c_{i-1} \), as shown in Figure 5.2.1. On the receipt of the sample \( r_1 \), each stored vector \( X_{i-1} \) in the first group (for which \( c_{i-1} \leq \text{ex}_1 \)) is expanded to form four vectors \( \{ Z_i \} \), having four different possible values of \( x_i \), as before. The costs \( \{ c_i \} \) of these four vectors \( \{ Z_i \} \) are calculated using equations 5.1.6 to 5.1.9. Each stored vector \( X_{i-1} \) in the
second group (for which \( x_1 < c_1 - 1 < x_2 \)) is expanded to form two vectors \( \{Z_i\} \), where the last components \( \{x_i\} \) in the two \( \{Z_i\} \) are those giving the lowest costs. Finally, each stored vector \( X_{i-1} \) in the third group (for which \( x_2 < c_1 - 1 \)) is expanded to one vector \( Z_i \), whose last component \( x_i \) takes on the one of its four possible values giving the lowest cost.

Since the costs \( \{c_i\} \) of four different possible vectors \( \{Z_i\} \) originating from any given \( X_{i-1} \) differ only in the values \( \{|r_i - p_i|^2\} \) (see equation 5.1.9), the vector \( Z_i \) having the lowest \( c_i \) must also have the smallest \( |r_i - p_i| \). Thus, the value of \( x_i \) for which \( |r_i - p_i| \) and therefore the corresponding cost \( c_i \) is smallest can be determined from \( |r_i - p_i| \) without actually calculating the four \( \{c_i\} \). The minimum \( |r_i - p_i| \) of the four expanded vectors \( \{Z_i\} \) originating from any given stored vector \( X_{i-1} \) can be determined as follows. For each received sample \( r_i \), the phase angle of \( r_i \), in the complex number plane, is determined by

\[
\text{ph}(r_i) = \arctan[\text{Im}(r_i)/\text{Re}(r_i)] \quad \text{(radians)}
\]

for \( 0 \leq \text{ph}(r_i) < 2\pi \), where \( \text{Re}(.) \), \( \text{Im}(.) \) and \( \text{ph}(.) \) represent the real, imaginary, and phase angle of the corresponding quantity, respectively, in the complex number plane. Following equation 5.1.8, the phase angle of the corresponding code symbol \( p_i \), of any particular vector \( Z_i \), is given by

\[
\text{ph}(p_i) = \pi z_i / 8 \quad \text{(radians)}
\]

where \( z_i \) is determined by equation 5.1.7. Now, let
\[ D'_p = (\text{ph}(r_i) - \text{ph}(p_i)) \mod 2\pi \] (5.2.4)

and
\[ D_p = D'_p \quad \text{if } 0 \leq D'_p < \pi \]
\[ D_p = 2\pi - D'_p \quad \text{if } \pi \leq D'_p < 2\pi \] (5.2.5)

It is quite clear that the code symbol \( p_i \) which is closest to \( r_i \), in the complex number plane, must also correspond to the smallest phase difference \( D_p \). Hence, for each stored vector \( X_{i-1} \), the \( J_1 \) selected expanded vectors \( \{Z_i\} \) with smallest \( |r_i - p_i| \) and thus lowest costs \( \{c_i\} \) can be determined as those with the lowest \( \{D_p\} \) defined by equation 5.2.5.

However, the determination of \( D_p \) requires the knowledge of the phase angle of \( r_i \), thus requiring additional computation. A further simplification may be achieved as follows. For each stored vector \( X_{i-1} \) in the second group (for which \( e_1 < c_{i-1} \leq e_2 \) and \( J_1 = 2 \)), the first selected expanded vector \( Z_i \) is determined as the vector for which

\[ D_a = |\text{Re}(r_i) - \text{Re}(p_i)| + |\text{Im}(r_i) - \text{Im}(p_i)| \] (5.2.6)

is smallest, over the four possible vectors \( \{Z_i\} \) originating from the given \( X_{i-1} \). The second selected expanded vector \( Z_i \) is that for which \( D_a \) in equation 5.2.6 has its second smallest value. The two selected \( \{Z_i\} \) here are often (but not always) those with the smallest costs. Clearly, only the costs \( \{c_i\} \) of the two selected vectors \( \{Z_i\} \) need be calculated. The process of adjustable expansion given above determines, for each stored vector \( X_{i-1} \) with \( e_1 < c_{i-1} \leq e_2 \), the two possible values of \( X_i \).
and hence the two expanded vectors \( \{Z_i\} \) that are worthy of further consideration by the decoder.

Similarly, each stored vector \( X_{i-1} \) in the third group (for which \( \text{ex}_2 < c_{i-1} \) and \( J_1 = 1 \)) is expanded to give the vector \( Z_i \), that corresponds to the code symbol \( p_i \) for which \( D_a \) in equation 5.2.6 is smallest. Again, only the cost \( c_i \) of the selected expanded vector \( Z_i \) requires evaluation.

Figure 5.2.2 shows, in the complex number plane, the two distance measures \( D_P \) and \( D_a \), defined by equations 5.2.5 and 5.2.6. Both equations can be used in the adjustable expansion process of system A2.

For a given number of stored vectors \( m_i \), the decoder now requires to calculate \( f_i \) costs values, where \( f_i \) is less than \( 4m_i \). The decoder next carries out the rest of the decoding process with \( f_i \) selected expanded vectors \( \{Z_i\} \), using the two thresholds \( C_m \) and \( C'_m \), exactly as for system A1.

The adjustable expansion procedure employed in system A2 has two significant features. Firstly, the algorithm uses the fact that the stored vectors \( \{X_i\} \) with low costs \( \{c_i\} \) are more likely to be the correct maximum likelihood vector of data symbols than those with relatively high costs. Therefore, \( J_1 \) is made large for low cost vectors and small for high cost vectors. From Figure 5.2.1, none of the four expanded vectors of any vector \( X_{i-1} \) with \( c_{i-1} \leq \text{ex}_1 \) is discarded from the decoder, whereas three out of the four expanded vectors of any vector \( X_{i-1} \) with \( \text{ex}_2 < c_{i-1} \) will be eliminated. Secondly, for any given stored vector \( X_{i-1} \), the \( J_1 \) selected expanded vectors \( \{Z_i\} \) are those with lowest costs over all possible expanded vectors. Furthermore, these selected expanded vectors are determined by
a simple threshold comparison process based on the computation in equation 5.2.6, thus avoiding any unnecessary calculation of the costs. Notice that each cost evaluation involves two squaring operations, whereas the determination of $D_a$ in equation 5.2.6 only requires additions and subtractions and is therefore relatively simpler.

The starting procedure for system A2 is the same as that employed in system A1.

The performance results for the various arrangements of system A2, obtained from a number of computer simulation tests, are presented in Figures 5.2.3 to 5.2.10, where they are shown as graphs of bit error rate (BER) against the signal/noise ratio, $E_b/N_0$. In each case, the total numbers of stored vectors and cost evaluations per decoded data symbol, averaged over the transmission of 100,000 data symbols (denoted by $\bar{m}_1$ and $\bar{f}_1$, respectively), are recorded for each signal/noise ratio tested. Appendix B6 presents an example of the computer programs used in the tests. The accuracy of each curve here, in the range of bit error rates of $10^{-3}$ to $10^{-4}$, is ±0.2 dB.

The results of computer simulation tests suggest that, so long as the parameters $e_{x_1}$ and $e_{x_2}$ are appropriately chosen such that $\bar{f}_1$ is approximately twice of $\bar{m}_1$ or more, the performance of system A2 can be made as close to that of the corresponding system A1 as possible. In other words, if an appropriate arrangement of system A2 is used in place of system A1, the average number of cost evaluations per decoded data symbol may now be reduced from $4\bar{m}_1$ to about $2\bar{m}_1$, without any significant degradation in the performance of the system.
In practice, the parameters $e_1$ and $e_2$ employed in system $A_2$ should be optimised for the most likely signal/noise ratio, for any given code.

Figure 5.2.3 shows the results obtained for the various arrangements of system $A_2$ with code UC(3) (a 16-state code), and Table 5.2.1 outlines the performances of these systems together with the corresponding decoder complexities, at different bit error rates. The latter complexity is shown by the average number of stored vectors (that is, $m_1$), together with the average number of cost evaluations per decoded data symbol (that is, $\overline{r}_1$), at each given BER. In all cases, $M_m=32$, $n=48$, $C_m' = C_m + 0.5$, $e_1=0.5$ and $e_2=1.0$. Figure 5.2.4 and Table 5.2.2 present those for the various arrangements of system $A_2$ with code UC(4) (a 32-state code). In each case here, $M_m=64$, $n=48$, $C_m' = C_m + 0.5$, $e_1=0.5$ and $e_2=1.2$. For all the arrangements in Figures 5.2.3 and 5.2.4, the distance measure used in the adjustable expansion process is defined by equation 5.2.6, and is denoted as $\text{Dist}=D_a$ on both graphs.

Comparisons between those shown in Tables 5.2.1 and 5.1.1, and between those shown in Tables 5.2.2 and 5.1.2, are undertaken. The important result from these comparisons is that, in both cases, the tolerance of system $A_2$ to noise is within about 0.1 dB of that of the corresponding system $A_1$, at each given BER. However, the corresponding $\overline{r}_1$ for system $A_2$ is reduced to approximately a half of that for system $A_1$, at a given BER.

Figure 5.2.5 contrasts the performances of system $A_2$, using the two different distance measures, $D_p$ and $D_a$, in the adjustable expansion process. $D_p$ and $D_a$ are defined by
equations 5.2.5 and 5.2.6, respectively. Clearly, these two measuring schemes provide similar tolerances to noise over the range of bit error rates of $10^{-1}$ to $10^{-4}$. For the fact that the determination of $D_a$ does not require as much additional computation as does the $D_p$, $D_a$ is used throughout.

Figure 5.2.6 summarizes the performances of those arrangements of system A2 that, according to the results of extensive computer simulation tests so far, appear to be mostly cost-effective for the three rate-2/3 convolutionally coded 8PSK signals (those generated by codes UC(3), UC(4) and UC(5)). The optimisation of each system is made, for the given code, as a cost-effective compromise between the gain in tolerance to noise and the decoder complexity. The criterion used to define this compromise is that the degradation in tolerances to noise of approximately 0.2 dB, compared to the Viterbi decoder incorporating the same code, is achieved over the range of bit error rates of $10^{-3}$ to $10^{-4}$, for as few average numbers of stored vectors per decoded data symbol as possible. Table 5.2.3 outlines the performances and equipment complexities of each system above, at different bit error rates. Table 5.2.4 shows the burst error characteristics and Table 5.2.5 presents the system parameters, for each of these systems.

From Table 5.2.3, at a BER of $10^{-4}$, the three systems gain 2.45 dB, 2.7 dB and 3.0 dB, respectively, in tolerance to noise over the uncoded QPSK system, the $m_1$ being 2.8, 5.0, 10.1, and $f_1$ being 6.9, 10.3, 17.6. These certainly compare quite well with those for the corresponding Viterbi decoder, which are 2.7 dB, 3.0 dB and 3.25 dB, respectively (see Table
3.3.1), the $\bar{m}_1$ being 16, 32, 64, and $\bar{f}_1$ being 64, 128, 256. In all cases, the tolerance of system A2 to noise is within 0.3 dB of that of the corresponding Viterbi decoder, but $\bar{m}_1$ is now reduced to about 1/6, and $\bar{f}_1$ to about 1/12, of those of the Viterbi decoder. Suppose next that the two decoding schemes with the same average number of cost evaluations per decoded symbol have approximately the same degree of equipment complexity, the conclusion here is that a carefully designed system A2 can reduce the decoder complexity to about 1/12 of that of the Viterbi decoder, with a small degradation in tolerance to noise of no more than 0.3 dB.

The performance of system A2 also compares well with that of the corresponding system A1. Take the example at BER of $10^{-4}$, where the given arrangements of system A1 gain 2.45 dB, 2.75 dB and 3.0 dB, respectively, over the QPSK system, the $\bar{f}_1$ being 11.6, 22.0 and 39.7 (see Table 5.1.4). At the same BER, the corresponding gains in tolerance of system A2 to noise are 2.45 dB, 2.7 dB and 3.0 dB, respectively, the $\bar{f}_1$ now being 6.9, 10.3, and 17.6 only (see Table 5.2.3). Clearly, a reduction of around a half in $\bar{f}_1$ is achieved with a degradation in tolerance to noise of within 0.1 dB. In other words, for a given amount of computation per decoded data symbol, system A2 can employ a significantly larger value of $\bar{m}_1$ and can therefore achieve a better tolerance to noise than can system A1.

Figure 5.2.7 presents the performance results for the correlative coded 8PSK signals of Section 4.3, using the decoding scheme of system A2. Table 5.2.6 outlines the performances and equipment complexities of each system at
different bit error rates, and Table 5.2.7 presents the system parameters for each decoder. The three codes used in the tests are the codes CC(1), CC(2) and CC(3) shown in Table 4.3.1.

Figure 5.2.8 shows the performances of system A2 with code CC(2), using two different distance measures in the adjustable expansion process. It is clearly that, for correlative coded 8PSK signals, these two measuring schemes (defined by equations 5.2.5 and 5.2.6) also give the very similar performance, over the range of bit error rates of $10^{-3}$ to $10^{-4}$. Again, $D_a$ is preferred.

Figures 5.2.9 and 5.2.10 contrast the performances of correlative coded 8PSK signals with those of convolutionally coded signals, using the same arrangements of system A2. The two codes under comparison in Figure 5.2.9 are codes CC(1) and UC(3), each having 16 states, whereas the two codes in Figure 5.2.10 are codes CC(3) and UC(5), each having 64 states. From Table 4.3.1, the asymptotic coding gains of codes CC(1) and CC(3) are about 0.5 dB inferior to those of UC(3) and UC(5). But, from Figures 5.2.9 and 5.2.10, the tolerances of each correlative coded system to noise, over the range of bit error rates of $10^{-3}$ to $10^{-4}$, are very close to those of the convolutionally coded signals.
### Table 5.2.1: Performances and equipment complexities of different arrangements of system A2 with code UC(3)

<table>
<thead>
<tr>
<th>$C_m$</th>
<th>$\bar{m}_1$</th>
<th>$\bar{r}_1$ (dB)</th>
<th>$\bar{m}_1$</th>
<th>$\bar{r}_1$ (dB)</th>
<th>$\bar{m}_1$</th>
<th>$\bar{r}_1$ (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.4</td>
<td>6.8</td>
<td>14.2</td>
<td>-0.05</td>
<td>4.3</td>
<td>9.4</td>
<td>1.55</td>
</tr>
<tr>
<td>1.5</td>
<td>9.0</td>
<td>18.0</td>
<td>0.0</td>
<td>5.0</td>
<td>10.0</td>
<td>1.6</td>
</tr>
<tr>
<td>1.6</td>
<td>11.0</td>
<td>20.0</td>
<td>0.1</td>
<td>6.0</td>
<td>11.0</td>
<td>1.65</td>
</tr>
</tbody>
</table>

Common attributes: 8PSK; $M_m=32$; $n=48$; $C'_m=C_m+0.5$; Dist=$D_a$

### Table 5.2.2: Performances and equipment complexities of different arrangements of system A2 with code UC(4)

<table>
<thead>
<tr>
<th>$C_m$</th>
<th>$\bar{m}_1$</th>
<th>$\bar{r}_1$ (dB)</th>
<th>$\bar{m}_1$</th>
<th>$\bar{r}_1$ (dB)</th>
<th>$\bar{m}_1$</th>
<th>$\bar{r}_1$ (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5</td>
<td>11.0</td>
<td>19.0</td>
<td>0.05</td>
<td>6.0</td>
<td>12.0</td>
<td>1.7</td>
</tr>
<tr>
<td>1.6</td>
<td>13.0</td>
<td>23.1</td>
<td>0.15</td>
<td>7.6</td>
<td>14.4</td>
<td>1.8</td>
</tr>
<tr>
<td>1.7</td>
<td>17.0</td>
<td>29.4</td>
<td>0.2</td>
<td>9.0</td>
<td>15.0</td>
<td>1.8</td>
</tr>
</tbody>
</table>

Common attributes: 8PSK; $M_m=64$; $n=48$; $C'_m=C_m+0.5$; Dist=$D_a$
Table 5.2.3: Performances and equipment complexities of the various arrangements of system A2 in Figure 5.2.6

<table>
<thead>
<tr>
<th>Code</th>
<th>BER=0.01.</th>
<th>BER=0.001.</th>
<th>BER=0.0001.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$m_1$,</td>
<td>$\tilde{m}_1$, (dB)</td>
<td>$m_1$, $\tilde{m}_1$, (dB)</td>
</tr>
<tr>
<td>UC(3)</td>
<td>6.8</td>
<td>14.2</td>
<td>-0.05</td>
</tr>
<tr>
<td>UC(4)</td>
<td>13.0</td>
<td>23.1</td>
<td>0.15</td>
</tr>
<tr>
<td>UC(5)</td>
<td>21.0</td>
<td>37.2</td>
<td>0.15</td>
</tr>
</tbody>
</table>

Table 5.2.4: Burst error characteristics of the various arrangements of system A2 in Figure 5.2.6

<table>
<thead>
<tr>
<th>Code</th>
<th>Average number of bit errors per burst at a given bit error rate, (approximately)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Code</td>
<td>UC(3)</td>
</tr>
<tr>
<td>Code</td>
<td>$2^g$</td>
</tr>
<tr>
<td>UC(3)</td>
<td>16</td>
</tr>
<tr>
<td>UC(4)</td>
<td>32</td>
</tr>
<tr>
<td>UC(5)</td>
<td>64</td>
</tr>
</tbody>
</table>
Table 5.2.5: System parameters for the various arrangements of system A2 shown in Figure 5.2.6

<table>
<thead>
<tr>
<th>Code</th>
<th>Number of states, $2^n$</th>
<th>$M_m$</th>
<th>$n$</th>
<th>$C_m$</th>
<th>$C'_m$</th>
<th>dist</th>
<th>$e_{x1}$</th>
<th>$e_{x2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>UC(3)</td>
<td>16</td>
<td>32</td>
<td>48</td>
<td>1.4</td>
<td>1.9</td>
<td>$D_a$</td>
<td>0.5</td>
<td>1.0</td>
</tr>
<tr>
<td>UC(4)</td>
<td>32</td>
<td>64</td>
<td>48</td>
<td>1.6</td>
<td>2.1</td>
<td>$D_a$</td>
<td>0.5</td>
<td>1.2</td>
</tr>
<tr>
<td>UC(5)</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>1.9</td>
<td>2.4</td>
<td>$D_a$</td>
<td>0.5</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Table 5.2.6: Performances and equipment complexities of the various arrangements of system A2 in Figure 5.2.7

<table>
<thead>
<tr>
<th>Code</th>
<th>Gain in tolerance to noise, $m_i$, $f_i$, (dB)</th>
<th>Gain in tolerance to noise, $m_i$, $f_i$, (dB)</th>
<th>Gain in tolerance to noise, $m_i$, $f_i$, (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC(1)</td>
<td>8 16 0.0</td>
<td>4 9 1.6</td>
<td>2.5 6.8 2.5</td>
</tr>
<tr>
<td>CC(2)</td>
<td>19 38 0.1</td>
<td>12 24 1.9</td>
<td>7.0 14.0 3.0</td>
</tr>
<tr>
<td>CC(3)</td>
<td>23 52 0.05</td>
<td>14 33 1.95</td>
<td>12.1 27.6 3.25</td>
</tr>
</tbody>
</table>

Common attributes: correlative coded 8PSK; dist=$D_a$
<table>
<thead>
<tr>
<th>Code</th>
<th>Number of states, 2^n</th>
<th>M_m</th>
<th>n</th>
<th>C_m</th>
<th>C'_m</th>
<th>dist</th>
<th>ex_1</th>
<th>ex_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC(1)</td>
<td>16</td>
<td>32</td>
<td>48</td>
<td>1.4</td>
<td>1.9</td>
<td>Da</td>
<td>0.5</td>
<td>1.0</td>
</tr>
<tr>
<td>CC(2)</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>1.8</td>
<td>2.3</td>
<td>Da</td>
<td>0.5</td>
<td>1.5</td>
</tr>
<tr>
<td>CC(3)</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>2.0</td>
<td>3.0</td>
<td>Da</td>
<td>0.5</td>
<td>1.5</td>
</tr>
</tbody>
</table>
Fig. 5.2.1: Expanded vectors \( \{Z_i\} \) in system A2 at time \( t=iT \)

\[
Z_i = [x_{i-n}, \ldots, x_{i-2}, x_{i-1}, x_i]
\]

- \( c_{i,r} < e_{x_i}; J_i = 4 \)
- \( e_{x_i} < c_{i,r} < e_{x_i}; J_i = 2 \)
- \( e_{x_i} < c_{i,r}; J_i = 1 \)

\( t=(i-n)T \quad \ldots \quad t=(i-2)T \quad t=(i-1)T \quad t=iT \)
Fig. 5.2.2: Different distance measures employed in the adjustable expansion process of system A2

\[ D_c = a + b \]
Fig. 5.2.3: Performance of system A2 with UC[3]

Common attributes: 8PSK; Mm=32; n=48; Cm=Cm+0.5; dist=Do; ex=0.5; ex₂=1.0

Legend
- Uncoded QPSK
- Cm =1.4
- Cm =1.5
- Cm =1.6
Fig. 5.2.4: Performance of system A2 with UC[4]

Legend

- △ Uncoded QPSK
- × Cm = 1.5
- □ Cm = 1.6
- ⊙ Cm = 1.7

Common attributes: 8PSK; Mm = 64; n = 48; Cm = Cm + 0.5;
dist = Da; ex = 0.5; ex_a = 1.2
Fig. 5.2.5: Performances of system A2 with different distance measures in the adjustable expansion process

Common attributes: \( UC[4]; 8PSK; M_m=64; n=48; C_m=1.6; C_m=2.1; e_x=0.5; e_x=1.2 \)
Fig. 5.2.6: Performances of rate-2/3 convolutionally coded 8PSK signals with system A2 decoding

Legend

- △ Uncoded QPSK
- × UC[3]; Mm=32; n=48; Cm=1.4; ex₁=1.0
- □ UC[4]; Mm=64; n=48; Cm=1.6; ex₂=1.2
- ☐ UC[5]; Mm=64; n=64; Cm=1.9; ex₃=1.5

Common attributes: Cm=Cm+0.5; dist=Da; ex₂=ex₁+0.5
Fig. 5.2.7: Performances of correlative coded 8PSK signals with system A2 decoding

\[ \begin{align*}
\text{Legend} \\
\Delta & \text{ Uncoded QPSK} \\
\times & \text{ CC[1]; } C_m=1.4; \ C'_m=1.9; \ e_x=1.0 \\
\square & \text{ CC[2]; } C_m=1.8; \ C'_m=2.3; \ e_x=1.5 \\
\boxdot & \text{ CC[3]; } C_m=2.0; \ C'_m=3.0; \ e_x=1.5
\end{align*} \]

Common attributes: \( M_m=64; \ n=64; \ dist=Da; \ e_x=0.5 \)
Fig. 5.2.8: Performances of system A2 with different distance measures in the adjustable expansion process.

Legend
- $\triangle$ Uncoded QPSK
- $\times$ dist = Da
- $\square$ dist = Dp

Common attributes: CC[3]; 8PSK; Mm = 64; n = 64; Cm = 2.0; Cm = 3.0; $e_x = 0.5$; $e_x = 1.5$
Fig. 5.2.9: Performance comparison of different coded 8PSK signals with system A2 decoding

Legend

- △ Uncoded QPSK
- × CC[1]; correlative code
- □ UC[3]; convolutional code

Common attributes: \( M_m=32; n=48; C_m=1.4; C_m=1.9; \) dist=Da

\( \text{ex }=0.5; \text{ex }=1.0 \)
Fig. 5.2.10: Performance comparison of different coded 8PSK signals with system A2 decoding

Legend
- Δ Uncoded QPSK
- × CC[2]; correlative code
- □ UC[5]; convolutional code

Common attributes: \( M_m = 64; n = 64; C_m = 1.8; C'_m = 2.3; \) dist = 0a;
\( e_x = 0.5; e'_x = 1.5 \)
5.3 COMMENTS ON THE DECODER COMPLEXITY

The important property of a decoding scheme, for a given degree of equipment complexity, is its tolerance to noise at a given bit error rate (BER). This complexity is normally measured by the processing speed required by the decoder to generate each decoded data symbol, which, in turn, reflects the cost in implementing the given scheme. It is reasonable to assume that a higher processing speed required generally means a higher cost in implementation. In this work, the factors that affect the speed of a decoder are considered to be follows. The first factor is the total number of stored vectors that the decoder uses in each decoding process, and the second factor is the total number of cost evaluations involved for each decoded data symbol. Clearly, the larger are these two numbers, the longer it will take to decode the given received sample, and correspondingly the decoder must operate at a higher processing speed.

The results of extensive computer simulation tests presented in the previous two sections have shown that, the arrangements of system A1 and system A2, whose performances are shown in Figures 5.1.4 and 5.2.6, are mostly cost-effective for the three rate-2/3 convolutionally coded 8PSK signals tested (those generated by codes UC(3), UC(4) and UC(5)). Figure 5.3.1 shows the block diagram of a more detailed arrangement for systems A1 and A2, where two appropriate buffer stores are employed at the input and output of the given decoder.

In this section, two types of graphs are given to
provide the statistical information for each of the best systems defined. The first type of graphs provides, for each scheme, the distributions of $m_i$ or $f_i$ at different bit error rates, average over the transmission of 100,000 data symbols. $m_i$ and $f_i$ represent, respectively, the number of stored vectors and the number of cost evaluations involved in decoding the $(i+1)^{th}$ received sample. These are shown in Figure 5.3.2 to 5.3.7. In each graph here, the horizontal axis denotes the possible number of stored vectors (or the number of cost evaluations per decoded symbol), whereas the vertical axis denotes the percentage of total transmission time when the given number stored vectors (or the given number of cost evaluations) is employed by the decoder.

Figures 5.3.2 to 5.3.4 show, for the three arrangements of system $A_1$ in Figure 5.1.4, the distributions of $m_i$ at the different bit error rates. Bearing in mind that, with system $A_1$, $f_i=4m_i$ for each $i$. The distribution curve for the corresponding $f_i$ must also have a similar shape as that for the $m_i$ and is therefore omitted. Figures 5.3.5 to 5.3.7 show, for the three arrangements of system $A_2$ in Figure 5.2.6, the distributions of $f_i$ at the different bit error rates. Since the two thresholds $C_m$ and $C'_m$ employed in system $A_2$ are chosen to be the same as those for the corresponding system $A_1$, the distribution curves for the corresponding $m_i$ must also have a similar shape as those shown in Figures 5.3.2 to 5.3.4. Thus, these curves for $m_i$ are also omitted.

From Figures 5.3.2 to 5.3.7 it is clear that, systems $A_1$ and $A_2$ may occasionally use a large number of vectors, which correspondingly increases the number of operations per decoded
data symbol over that period of time. Fortunately, the two decoders use only a small number of vectors for the majority of the time. This is particularly true at the lower bit error rates.

In order to exploit this property of systems A1 and A2, the computation involved in decoding any received sample must not be confined to the corresponding symbol interval T. Instead, the time available for any one decoding process must be a variable, so that by spending less than T seconds on each of the majority of decoding processes, that require only a few vectors, considerable more than T seconds can be made available for each of the minority of decoding processes, that require a large number of vectors. For this to be possible, two buffer stores are required, one to hold a number of the received samples \{r_i\} at the input of the decoder, and the other to hold a number of the decoded data symbols \{s'_i\} at the output of the decoder, as shown in Figure 5.3.1. While the decoder operates with only a few vectors, the input buffer store slowly empties while the output buffer store slowly fills, and while the decoder operates with a large number of vectors, the input buffer store slowly fills while the output buffer store slowly empties.

The approximate size of the input and output buffer stores, for each best arrangement of the new decoder, is now determined. The following analysis is based on the assumption that the time taken to decode any received sample is determined by the total number of cost evaluations involved (which is also the total number of selected expanded vectors used), in
decoding the given sample.

From Figure 5.3.1, the received samples \( r_j \) arrive at the input buffer at a constant rate of \( 1/T \) samples per second, and the similarly, the decoded data symbols \( s'_j \) leave the output buffer also at a constant rate of \( 1/T \) symbols per second. Suppose that, on average, the total computation and processing time for a single selected expanded vector is \( T_c \) seconds, then the time taken to generate the \((j+1)^{th}\) decoded data symbol, where a total of \( f_j \) expanded vectors is involved, can be approximately to

\[
T_j = f_j \times T_c \quad (5.3.1)
\]

Suppose also that the processing speed of the given decoder is as such that it is able to handle the maximum of \( F \) expanded vectors, over each symbol interval \( T \), then

\[
F \times T_c = T \quad (5.3.2)
\]

Let the total number of samples accumulated in the input buffer at the end of \((j+1)^{th}\) decoding process be \( \lambda_j \), for \( j = 0, 1, 2, \ldots \), then

\[
\lambda_j = \text{int}\left[ \sum_{h=k}^{j} T_h / T \right] + 1 - (j+1-k) \quad (5.3.3)
\]

It is assumed in equation 5.3.3 that \( \lambda_k \geq 0 \), at the end of some \( k^{th} \) decoding process earlier on, and \( \text{int}[.] \) represents the integer part of the corresponding real-valued quantity. Equation 5.3.3 holds valid for all \( j > k \), until either the end of entire transmission period, or else the next time when the given buffer store becomes empty again. Clearly, the \( \lambda_j \) defined by equation 5.3.3 is always a non-negative integer.
number. Now, following equations 5.3.1 to 5.3.3,

$$\lambda_j = \text{int} \left( \frac{\sum_{h=k}^{j} f_h \times T_C}{F \times T_C} \right) + 1 - (j+1-k)$$

$$\lambda_j = \text{int} \left( \frac{\sum_{h=k}^{j} f_h / F}{F} \right) + 1 - (j+1-k)$$ (5.3.4)

The second type of graphs, as shown in Figures 5.3.8 to 5.3.13, provides a measure of the buffer store requirements for the different arrangements of systems A1 and A2, whose performance are shown in Figure 5.1.4 and 5.2.6. In each graph here, the horizontal axis represents the possible number of storage elements in the buffer, denoted by N, whereas the vertical axis represents the percentage of total transmission time when the number of samples stored in the buffer (that is the \( \lambda_j \) defined by equation 5.3.4) is equal or greater than the given value of N. Under the assumed conditions, each curve here must also show, for the given arrangement and processing speed (that is, the given value of F) of the decoder, the probability that the buffer is filled for the given buffer size of N (samples).

The conclusion of the results shown in Figures 5.3.8 to 5.3.13 is that, if the given buffer store can hold around 250-300 samples, then the decoder must be able to handle at least approximately 1.2\( \bar{f}_1 \) expanded vectors over each symbol interval (that is \( F=1.2\bar{f}_1 \)), in order to reduce to a negligible value the probability that the input buffer store is filled. When the latter occurs, the maximum permitted number of expanded vectors (and hence the cost evaluations) is reduced to
a value a little below the number that can be handled within the sampling interval, and it is held at this value until the input buffer store is about two-thirds full. Normal operation of the system is then resumed.

It has, for example, been shown that, with code UC(3), system A1, and at a BER of $10^{-3}$, where $f_i = 17.6$ (see Table 5.1.4), the decoder must be able to process at least 22 expanded vectors within each symbol interval, for the given size of the buffer stores. Again, with code UC(3), system A2 and at a BER of $10^{-3}$, where $f_i = 9.4$, (see Table 5.2.3), the decoder must now be able to process 12 vectors over each symbol interval. These certainly compare well with that for the corresponding Viterbi decoder, which requires a processing speed of 64 expanded vectors per symbol interval. Take another example with code UC(5) and at a BER of $10^{-3}$. Clearly, for system A1, $f_i = 60.1$, the decoder must be able to handle at least 72 expanded vectors per symbol interval. For system A2, $f_i = 24.1$, the decoder must now be able to handle about 30 to 32 expanded vectors. However, the corresponding Viterbi decoder requires a processing speed of 256 expanded vectors per symbol interval. A reduction of more than three-quarters in the processing speed of the decoder can be generally expected.

Finally, no serious degradation in performance is likely to be experienced, if the storage capacity of each buffer store is reduced from 300 samples to, say, 200 samples, with now the occasional reduction in the maximum permitted number of expanded vectors. Thus the size of the buffer stores and the resulting increase in the delay in detection need not to be excessive.
Fig. 5.3.1: Butter stores for systems A1 and A2

Output buffer store at a constant rate of \( \frac{1}{T} \) symbols/second

System A1 or system A2 both at a variable rate

Input buffer store at a constant rate of \( \frac{1}{T} \) symbols/second

at a constant rate of \( \frac{1}{T} \) symbols/second
Fig. 5.3.2: Distributions of $m_i$ at different bit error rates, for system A1 with UC[3]

Legend

- $\Delta$ $E_b/N_0=4.3$ dB; $BER=0.01$; $\bar{m}_i=6.9$
- $\times$ $E_b/N_0=5.1$ dB; $BER=0.001$; $\bar{m}_i=4.4$
- $\square$ $E_b/N_0=6.0$ dB; $BER=0.0001$; $\bar{m}_i=2.9$

Common attributes: BPSK; $M_m=32$; $n=48$; $C_m=1.4$; $\tilde{C}_m=1.9$
Fig. 5.3.3: Distributions of $m_i$ at different bit error rates, for system A1 with UC[4]

Legend

- $\triangle$ $Eb/No=4.2$ dB; $BER=0.01; m_i = 13.0$
- $\times$ $Eb/No=5.0$ dB; $BER=0.001; m_i = 7.8$
- $\square$ $Eb/No=5.7$ dB; $BER=0.0001; m_i = 5.5$

Common attributes: 8PSK; $M_m=64; n=48; C_m=1.6; C_m=2.1$
Fig. 5.3.4: Distributions of $m_i$ at different bit error rates, for system A1 with UC[5]

Legend

- $\Delta$ $Eb/No=4.1$ dB; BER=0.01; $m_i = 23.7$
- $\times$ $Eb/No=4.8$ dB; BER=0.001; $m_i = 14.8$
- $\square$ $Eb/No=5.5$ dB; BER=0.0001; $m_i = 10.0$

Common attributes: 8PSK; $M_m=64$; $n=64$; $C_m=1.9$; $C_m=2.4$
Fig. 5.3.5: Distributions of $f_i$ at different bit error rates, for system A2 with UC[3]

Legend
- $\triangle$ $Eb/No=4.3$ dB; BER=0.01; $f_i=14.2$
- $\times$ $Eb/No=5.1$ dB; BER=0.001; $f_i=9.4$
- $\Box$ $Eb/No=6.0$ dB; BER=0.0001; $f_i=6.9$

Common attributes: 8PSK; $M_m=32$; $n=48$; $C_m=1.4$; $C_m'=1.9$; $dist=Da$; $e_x=0.5$; $e_x'=1.0$
Fig. 5.3.6: Distributions of $f_\kappa$ at different bit error rates, for system A2 with UC[4]

Legend

- $\triangle$ $Eb/No=4.2$ dB; $BER=0.01$; $f_\kappa=23.1$
- $\times$ $Eb/No=5.0$ dB; $BER=0.001$; $f_\kappa=14.4$
- $\square$ $Eb/No=5.8$ dB; $BER=0.0001$; $f_\kappa=10.3$

Common attributes: 8PSK; $M_m=64$; $n=48$; $C_m=1.6$; $C_m=2.1$; $dist=Da$; $ex_1=0.5$; $ex_2=1.2$
Fig. 5.3.7: Distributions of $f_i$ at different bit error rates, for system A2 with UC[5]

Legend

$\triangle$ $Eb/No=4.1$ dB; BER=0.01; $f_i=37.2$

$\times$ $Eb/No=4.9$ dB; BER=0.001; $f_i=24.1$

$\square$ $Eb/No=5.5$ dB; BER=0.0001; $f_i=17.6$

Common attributes: 8PSK; $M_m=64$; $n=64$; $C_m=1.9$; $C_m'=2.4$; dist=Da; $e_x=0.5$; $e_x'=1.5$
Fig. 5.3.8: Percentage of total transmission time when $\lambda_i$ is greater or equal to $N$, for system A1 with UC[3], at BER=0.001

Legend
- $\Delta$ when $F=16$
- $\times$ when $F=18$
- $\square$ when $F=20$
- $\otimes$ when $F=22$
- $\bigotimes$ when $F=24$

Common attributes: 8PSK; $E_b/N_0=5.1$ dB; $M_m=32$; $n=48$; $C_m=1.4$; $C_m=1.9$; $m_i=4.4$; $f_i=17.6$
Fig. 5.3.9: Percentage of total transmission time when \( \lambda \) is greater or equal to \( N \), for system A1 with UC[4], at BER=0.001

Common attributes: 8PSK; \( \frac{E_b}{N_0}=5.0 \) dB; \( M_m=64; n=48; C_m=1.6; C_n=2.1; \overline{m}_r=7.8; T_r=31.2 \)
Fig. 5.3.10: Percentage of total transmission time when $\lambda_j$ is greater or equal to $N$, for system A1 with UC[5], at BER=0.001

Legend

- $\triangle$ when $F=58$
- $\times$ when $F=60$
- $\square$ when $F=62$
- $\times$ when $F=64$
- $\times$ when $F=66$
- $\times$ when $F=68$
- $\phi$ when $F=70$
- $\oplus$ when $F=72$

Common attributes: 8PSK; $E_b/N_0=4.8$ dB; $M_m=64$; $n=64$; $C_m=1.9$; $C_m=2.4$; $m_x=14.8$; $f_c=60.1$
Fig. 5.3.11: Percentage of total transmission time when $\lambda_i$ is greater or equal to $N$, for system A2 with UC[3], at BER=0.001

Common attributes: 8PSK; $E_b/N_o=5.1$ dB; $M_m=32$; $n=48$; $C_m=1.4$; $C_m=1.9$; dist=Da; $x_1=0.5$; $x_2=1.0$; $M_1=4.3$; $f_s=9.4$
Fig. 5.3.12: Percentage of total transmission time when \( \lambda_i \) is greater or equal to \( N \), for system A2 with UC[4], at BER=0.001.

Legend

- \( \Delta \) when \( F=16 \)
- \( X \) when \( F=18 \)
- \( \square \) when \( F=20 \)

Common attributes: 8PSK; \( E_b/N_0=5.0 \) dB; \( Mm=64; n=48; Cm=1.6; \\
\( Cm=2.1; \) dist=Da; \( e_x=0.5; e_x^*=1.2; m_x=7.6; T_x=14.4 \)
Fig. 5.3.13: Percentage of total transmission time when $\lambda_i$ is greater or equal to $N$, for system A2 with UC[5], at BER=0.001

Legend
- $\Delta$ when $F=24$
- $\times$ when $F=26$
- $\Box$ when $F=28$
- $\nabla$ when $F=30$

Common attributes: 8PSK; $Eb/No=4.9$ dB; $M_m=64$; $n=64$; $C_{m}=1.9$; $C_{m}=2.4$; dist=da; $ex_i=0.5$; $ex_j=1.5$; $m_i=13.8$; $T = 24.1$
CHAPTER SIX

ROTATIONALLY INVARIANT CONVOLUTIONALLY

CODED 8 AND 16 PSK SIGNALS
The channel model used in the work so far assumes correct carrier-phase synchronisation at the receiver. However, in the presence of impulsive noise and/or sudden phase changes introduced by the transmission path, the correct phase tracking at the coherent demodulator is difficult to achieve in practice. This chapter examines the problem of carrier-phase synchronisation for convolutionally coded MPSK systems under these conditions.

Section 6.1 presents the general system model in which the differential precoding operation is used for avoiding the prolonged error bursts that are otherwise caused by the impulsive noise and sudden phase changes. Section 6.2 describes the rotationally invariant convolutional codes which are particularly suitable for such a system. The design of the codes here is carried out along the same line as that proposed for QAM signals by L.F. Wei, in 1983 [51]. Sections 6.3 and 6.4 describe the procedures for designing rate-2/3 and rate-3/4 rotationally invariant coded 8 and 16 PSK signals, and also present the performance results for these new codes. In Section 6.5, a simple carrier-phase correction system is proposed for the receiver. When combined with such a phase correction system, the new coded PSK signals of Sections 6.3 and 6.4 are able to tolerate any likely phase changes in the reference carriers of the coherent demodulation process.
6.1 DIFFERENTIALLY CODED SYSTEM MODEL

Now, return the bandpass model of MPSK signals in Figure 2.3.1. Supposing the transmission path introduces a time-varying phase advance of $\phi_t$ radians into the demodulated baseband signal $r(t)$, this is equivalent to multiply the $r(t)$ by $\exp(j\phi_t)$. But the reference carriers at the coherent demodulator in Figure 2.3.1 have a phase advance of $\phi$ radians relative to that at the transmitter, and this is equivalent to multiply $r(t)$ by $\exp(-j\phi)$. Clearly, when the reference carriers are precisely phase locked onto the received signal, $\phi_t=\phi$, and correct carrier-phase synchronisation is achieved. This condition has been assumed in the work so far.

Following equation 2.1.9, the complex-valued baseband waveform, at the output of the demodulator, is

$$r(t) = \sum_{i=0}^{\infty} q_i h(t-it) + w(t) \quad (6.1.1)$$

with the correct carrier-phase synchronisation at the receiver. $h(t)$ is the impulse response of the given baseband channel, and $w(t)$ is the complex-valued noise waveform at the demodulator output. In addition, the real and imaginary parts of $w(t)$ are statistically independent Gaussian random variables with zero mean and a constant two-sided power spectral density of $(1/2)N_0$. $T$ is the symbol interval in seconds.

For the baseband channel transfer-function shown in Figure 2.1.2, the sample value of $r(t)$, at time instant $t=iT$, is

$$r_i = q_i + w_i \quad (6.1.2)$$
where the real and imaginary parts of \( \{ w_i \} \) are statistically independent Gaussian random variables with zero mean and fixed variance \( \sigma^2 \).

Unfortunately, the correct phase tracking assumed in the bandpass model of Section 2.3 is rather difficult to achieve in practice, especially when the transmission path introduces sudden phase changes, called phase jumps. Impulsive noise may also cause an incorrect carrier-phase synchronisation at the receiver. In practice, most of carrier-phase synchronisation circuits introduce a phase ambiguity into the reference carriers generated at the receiver [7,99] (further details of these techniques are, however, beyond the scope of this discussion). The phase ambiguity occurs in the sense that the two reference carriers at the receiver are phase locked onto the received signal, as such that they have one of the different possible phases of this signal. What happens now is that the phase jump or impulsive noise causes a temporary failure of the carrier-phase synchronisation circuits at the receiver, and when synchronisation is regained, it is locked onto any one of \( M \) different possible phases of the received signal. Under these conditions, the demodulated baseband signal becomes

\[
r'(t) = \sum_{i=0}^{M-1} q_i h(t-iT) \exp(j2\pi h/M) + w'(t)
\]

for \( h=0, 1, 2, \ldots, (M-1) \). The factor \( \exp(j2\pi h/M) \) (which is unknown to the receiver) rotates the complex-valued baseband signal at the output of the demodulator by \( 2\pi h/M \) radians, anti-clockwise in the complex number plane, without changing...
the signal level. Therefore, the real and imaginary parts of
the noise waveform \( w'(t) \) in equation 6.1.3 are also
statistically independent Gaussian random variables with zero
mean and a two-sided power spectral density of \((1/2)N_0\), exactly
as the \( w(t) \) in equation 6.1.1. The corresponding sample value
of \( r'(t) \), at time \( t=iT \), now becomes

\[
r'_i = q_i \exp(j2\pi h/M) + w'_i
\]

(6.1.4)

for \( h=0, 1, 2, ..., (M-1) \). Such a system has an inherent phase
ambiguity of \( M \) possible angles, that are multiples of \( 2\pi/M \)
radians. Notice that the rotation of \( 2\pi h/M \) radians in \( q_i \) must
always result in one of its \( M \) possible (valid) values. When
\( h=0 \), where the correct phase tracking is achieved, the \( r'_i \)
defined by equation 6.1.4 becomes the \( r_i \) in equation 6.1.2.

In general, a fixed phase rotation of a multiple of \( 2\pi/M \)
radians is introduced into each of the received samples. When
\( h \neq 0 \), this phase rotation will cause errors in the subsequently
decoded/detected data symbols even in the absence of noise,
until end of the received message, and may therefore severely
degrade the performance of the system. In the systems without
coding, this problem is generally solved by using differential
coding operation as in any conventional DQPSK (differentially
coded QPSK) and DBPSK (differentially coded BPSK) systems,
where a differential decoder follows the coherent detection
process to eliminate the effect of any phase shifts [7,8,10].

Figure 6.1.1 shows the block diagram of the data trans-
mmission system which incorporates the differential precoder and
decoder outside the complete process of convolutional coding,
modulation, demodulation, and decoding. Here, the $2^L$-level data symbol $s_i$, that is uniquely determined by the $L$-component vector of binary data symbols

$$b_i = [b_i(1) b_i(2) \ldots b_i(L)] \quad (6.1.5)$$

is defined as a complex number and may therefore be represented by a point in the complex number plane. Figure 6.1.2 shows, for rate-2/3 convolutionally coded BPSK signals, the constellations of the $4$-level data symbols $\{s_i\}$ and the corresponding $8$-level code (transmitted) symbols $\{q_i\}$, in the complex number plane. Figure 6.1.3 shows those for rate-3/4 coded 16PSK signals.

The sequence of $\{s_i\}$ in Figure 6.1.1 is first fed to a differential precoder, to give the corresponding differentially coded data symbols $\{d_i\}$, where $d_i$ is also complex-valued and may take on any of the $2^L$ different possible values of $s_i$. $d_i$ is itself uniquely determined by another $L$-component vector

$$u_i = [u_i(1) u_i(2) \ldots u_i(L)] \quad (6.1.6)$$

where $u_i(1), u_i(2), \ldots, u_i(L)$ are statistically independent binary numbers with the possible values of 0 and 1.

The differential precoder operates as follows. Since the absolute value of $s_i$ is held constant, as can be seen in Figures 6.1.2 and 6.1.3, the message to be transmitted is now carried by the phase angles of complex-valued data symbols $\{s_i\}$. The differential coding operation is as such that, at
time $t=iT$, the phase angle of the differentially coded symbol $d_i$ in the complex number plane, is given by the sum of the phase angles of $s_i$ and the previous differentially coded symbol $d_{i-1}$. That is,

$$\text{ph}(d_i) = (\text{ph}(d_{i-1}) + \text{ph}(s_i)) \mod 2\pi \quad (6.1.7)$$

where $\text{ph}(\cdot)$ represents the phase angle (in radians) of the complex-valued quantity. The message to be transmitted is now carried by the phase differences in the consecutive complex-valued symbols $\{d_i\}$. The sequence of $\{d_i\}$ is next fed to the convolutional coder in the place of $\{s_i\}$.

At the output of the Viterbi decoder in Figure 6.1.1 is the sequence of decoded (but differentially-coded) symbols $\{d'_i\}$. The sequence of $\{d'_i\}$ is next fed to a differential decoder, to give the decoded data symbols $\{s'_i\}$. The phase angle of each $s'_i$, in the complex number plane, is determined by the difference between the phase angles of decoded symbols $d'_i$ and $d'_{i-1}$. That is,

$$\text{ph}(s'_i) = (\text{ph}(d'_i) - \text{ph}(d'_{i-1})) \mod 2\pi \quad (6.1.8)$$

The differential decoding process removes any constant phase rotation in the $\{d'_i\}$, that is caused by the likely shift in phases of the reference carriers at the receiver. In the absence of noise, $\{s'_i\}$ must be the $\{s_i\}$ fed to the differential precoder at the transmitter.

The analysis so far assumes that the decoder at the receiver (before the differential decoding process) in Figure
6.1.1 is transparent to any likely phase rotations in the received signal. This means that a rotated version of a coded signal sequence, where the carrier phase has been changed through a fixed angle (see equation 6.1.4), must be another valid code sequence as far as the decoder is concerned. Unfortunately, this is not generally true with the optimum rate-2/3 and rate-3/4 convolutional coded 8 and 16 PSK signals shown in Tables 3.2.1 and 3.2.2. In the following sections, rotationally invariant codes are defined and the new coded 8 and 16 PSK signals are presented.

Figure 6.1.11 presents the performances of uncoded DQPSK and D8PSK (differentially coded QPSK and 8PSK) systems using coherent detection. The latter detector has been described in Section 2.11. The performance results for the uncoded QPSK and 8PSK systems are also shown. The results are presented as graphs of bit error rate against the signal/noise ratio, $E_b/N_0$. $E_b$ is the average signal energy per data bit and $(1/2)N_0$ is the two-sided power spectral density of the additive white Gaussian noise. The accuracy of each curve, in the range of bit error rates of $10^{-3}$ to $10^{-11}$, is approximately ±0.1 dB.

From equation 6.1.8 it is evident that each single error in the detected symbols $\{d'_i\}$ will cause two consecutive detected data symbols $\{s'_i\}$ being incorrect, at the output of the differential decoder. This approximately doubles the error rate in the $\{s'_i\}$. From Figure 6.1.4, the performance of each differentially coded system, at a bit error rate of $10^{-4}$, is about 0.3 dB inferior to that of the corresponding non-differentially coded system.
Fig. 6.1.1: General model of the data transmission system with sudden carrier–phase changes

\[ r(t) = \exp(j2\pi h/M) \]

for \( h=0,1, \ldots, (M-1) \)
Fig. 6.1.2: 8PSK signal constellation and mapping

**Data Signal** $s_i$

**Coded Signal** $q_i$
Fig 6.1.3: 16PSK signal constellation and mapping

data signal $s_i$

coded signal $q_i$
Fig. 6.1.4: Performances of uncoded DQPSK, QPSK, D8PSK and 8PSK systems with coherent threshold detection.
6.2 **ROTATIONALLY INVARIANT CODES**

The conclusion of Section 6.1 is that an appropriate arrangement of differential precoding/decoding must be used in the coherent data transmission system, in order to prevent the prolonged error bursts that are otherwise caused by phase jumps. To achieve the best performance, the precoding/decoding must be implemented outside the complete process of convolutional coding, modulation, demodulation, and decoding. Unfortunately, with phase coded MPSK signals, this approach requires the total transparency of the decoder to any likely rotations in phase of the received signal. It means that, if a fixed likely phase change (see equation 6.1.4) is introduced into each signal \( q_i \) of a coded sequence, then another valid coded signal sequence must be obtained with, furthermore, a constant phase change in each signal of the resulting decoded message.

Assuming that the carrier phase only changes very slowly with time, a linear first-order decision-feedback carrier-phase tracking system was proposed by G. Ungerboeck [39] for \( \pi \) radians rotationally invariant codes. These codes must operate correctly with the given phase angle and also with a phase shift of \( \pi \) radians. It is found that, among the nine best known convolutional codes for 8 and 16 PSK signals shown in Tables 3.2.1 and 3.2.2, code UC(1) and codes UC(5) to UC(9) are \( \pi \) radians rotationally invariant. Therefore, they are suitable for the above scheme.

However, it is rather unfortunate that such a system does not generally tolerate arbitrary phase jumps. The results
of a number of computer simulation tests have shown that, for the 8PSK system incorporating code UC(1), the given scheme constantly failed to track any sudden phase jump which are greater than \( \frac{1\pi}{8} \) radians. The aim of this study, whose results are presented in this chapter, is to develop new rate-2/3 coded 8PSK and rate-3/4 16PSK systems that can tolerate these phase jumps.

The model of the complete coding process is given in Figure 6.1.1, where the differential precoder operates on data symbols \( \{s_i\} \) to form differentially-coded data symbols \( \{d_i\} \). \( \{d_i\} \) are uniquely related to the vectors \( \{u_i\} \) (see equation 6.1.6). The operations of differential precoder and decoder are described in Section 6.1, where \( s_i \) and \( d_i \) are both defined to be complex numbers.

Since the absolute values of \( s_i \) and \( d_i \) are held constant, and the phase angle of \( d_i \), in the complex number plane, is given by the sum of the phase angles of \( s_i \) and \( d_{i-1} \) (see equation 6.1.7), any change in \( d_i \) from one of its possible values to another may be represented as a rotation in \( d_i \), in the complex number plane, the amount of the rotation being given by a particular value of \( s_i \). Since there are \( 2^L \) different possible values of \( s_i \), there are the corresponding \( 2^L \) different possible rotations in \( d_i \). These \( 2^L \) rotations in \( d_i \) are confined to the \( 2^L \) different possible rotations of an individual \( d_i \), which result in another valid symbol \( d_i \), such that the rotations are given by the possible values of \( s_i \). Notice that the possible rotations in \( d_i \) may also be represented by the equivalent changes in the vector \( u_i \), which in turn, are given by the possible values of the vector \( b_i \).
For convenience, this change is referred to as a "rotation", bearing in mind that the rotation is that in the associated complex-valued symbol and not in the vector itself.

A rotationally invariant coder defined in this work must have a structure such that, when its input symbols \( \{d_i\} \) are rotated through a constant phase angle, the same constant phase rotation also appears in each signal \( q_i \) of the corresponding code sequence. Since, for any rate-L/(L+1) code, there are only \( 2^L \) possible rotations in \( d_i \), but there are \( 2^{L+1} \) possible rotations in \( q_i \), only half of all possible rotations in phase of the received signal is included and thus the rotationally invariant codes defined here are only transparent to half of all possible phase shifts in \( r_i \) at the receiver. This point is further considered in Section 6.5.

The reasons for adopting this particular approach are as follows. If the codes are made fully transparent to all possible phase rotations in the received signal, that is, to a phase rotation (phase slip) of any multiple of \( \pi/4 \) radians in the case of a rate-2/3 code and of any multiple of \( \pi/8 \) radians in the case of a rate-3/4 code, an undue degradation in tolerance to noise is usually experienced relative to the corresponding optimum but not rotationally invariant code (Reference 100 examines full transparency for MPSK signals, but only few codes were found there).

On the other hand, under the assumed noise conditions, a conventional code that is, say, transparent to only a phase rotation of \( \pi \) radians, involves a fairly complex sequence of trial and error correction processes, in order to restore the correct phases to the reference carriers in the coherent
demodulator, once a phase slip has occurred. The techniques studied in this work have the advantage that they achieve near-optimum gain in tolerance to noise, and following the detection of any phase slip in the reference carriers, that leads to the incorrect operation of the decoder, this phase slip can be corrected with no further checks or phase changes being required to correct the given phase error.

Now, let

\[ U_k = [ u_h \ u_{h+1} \ldots \ u_k ] \]  \hspace{1cm} (6.2.1)

be any possible sequence of differentially coded vectors, and let

\[ V_k = [ v_{h+K} \ v_{h+K+1} \ldots \ v_k ] \]  \hspace{1cm} (6.2.2)

be the corresponding sequence of convolutionally coded vectors. \( K \) is the memory of the code, measured in the number of symbols. Suppose also that

\[ Q_k = [ q_{h+K} \ q_{h+K+1} \ldots \ q_k ] \]  \hspace{1cm} (6.2.3)

is the sequence of complex-valued transmitted symbols which is uniquely determined by the \( V_k \). Again, let

\[ U'_k = [ u'_h \ u'_{h+1} \ldots \ u'_k ] \]  \hspace{1cm} (6.2.4)

be another sequence of differentially coded vectors derived from the \( U_k \), with each component \( u_1 \) changed, such that the
complex-valued symbol $d_i$ corresponding to each $u_i$ in equation 6.2.1, for $i=h, h+1, \ldots, l$, is rotated through a fixed angle of $\Theta$ radians, in the complex number plane. The latter is assumed to carry $d_i$, and therefore $u_i$, into another one of its $2^L$ possible values. Clearly,

$$\Theta = \left(\frac{2\pi k}{2^L}\right) \mod 2\pi \tag{6.2.5}$$

where $k$ is an integer. The $d'_i$ corresponding to the vector $u'_i$ in equation 6.2.4 must be given by

$$d'_i = d_i \exp(i\Theta) \tag{6.2.6}$$

for $i=h, h+1, \ldots, l$. Now, suppose that

$$v'_l = [v'_{h+K} \ v'_{h+K+1} \ \ldots \ v'_{l}] \tag{6.2.7}$$

is the code sequence of $U'_l$, and let

$$q'_l = [q'_{h+K} \ q'_{h+K+1} \ \ldots \ q'_{l}] \tag{6.2.8}$$

be the corresponding sequence of transmitted symbols. If the convolutional coder is transparent to the given phase rotation of $\Theta$ radians, then each component $q'_i$ of the latter sequence $Q'_l$ must be given by

$$q'_i = q_i \exp(i\Theta) \tag{6.2.9}$$

for $i=h+K, h+K+1, \ldots, l$. 
Finally, if the above property holds true for all possible $(U_l)$ and $(U'_l)$, the given convolutional coder is said to be transparent to all $2^L$ possible rotations in the $(d_l)$. This means that, in the absence of noise, if the carrier phase of the received signal is rotated by $\theta$ radians (see equation 6.2.5), then the correct decoded sequence at the output of the Viterbi decoder must be the corresponding $U'_l$, bearing in mind that $U'_l$ is formed from the $U_l$ with each component $u_i$ rotated (changed) by $\theta$ radians. This constant phase change of $\theta$ radians in each subsequent decoded (but differentially coded) data symbol can then be removed by the differential decoding process following the decoder.

The investigation has been concentrated on finding the convolutional code for a given constraint length, such that the coder is transparent to the $2^L$ possible rotations in $(d_l)$, while still keeping the good distance properties of conventional codes. This means that certain rules must be obeyed in addition to the three coding rules given in Section 3.2. The work here follows the approach suggested by L.F. Wei, where rotationally invariant QAM codes were considered [51].

Coding Rule (4)

Now consider a state transition from state $S_i$ to the next state $S_{i+1}$, with the corresponding input vector $u_i$ and output vector $v_i$. For a state transition from $S'_i$ to $S'_{i+1}$ with the input vector $u'_i$, where $S'_i$, $S'_{i+1}$ and $u'_i$ are the corresponding equally rotated (changed) forms of $S_i$, $S_{i+1}$ and $u_i$, respectively, the coded vector must be $v'_i$, where $v'_i$ is the $v_i$ rotated (changed) by the same amount and in the same
direction as \( u_i \). This relationship is here known as the rule of equal rotation.

Since the state \( S_i \) of the coder is generally given by the combination of storage elements \( u_{i-1}, u_{i-2}, \ldots, u_{i-K} \) in the coder, a rotated (changed) form of \( S_i \) must, in fact, be given by the combination of the corresponding equally rotated (changed) versions of the individual elements \( \{u_i\} \).

The Coding Rules (1) to (3) of Section 3.2 together with the Coding Rule (4) just given must all be followed in designing the rotationally invariant codes. Whereas the first three rules ensure a good minimum free distance property of the code, the rule of equal rotation plays a fundamental role in determining the transparency of the coder to the \( 2^L \) possible rotations in \( \{d_i\} \) (and the vectors \( \{u_i\} \)).

The differential precoder and convolutional coder may, of course, be combined as a single finite state feedback network, with built-in transparency to the phase shifts of \( \Theta \) radians. However, in order to simplify the design process, the two functions are considered separately.
6.3 RATE-2/3 CONVOLUTIONAL CODES THAT ARE ROTATIONALLY INVARIANT TO PHASE CHANGES OF $\pi/2$ AND $\pi$ RADIANS

In the case of a rate-2/3 convolutionally coded 8PSK signal, the four possible phase rotations in a differentially coded data symbol $d_1$ are 0, $\pi/2$, $\pi$ and $3\pi/2$ radians, respectively, that is

$$\theta = (k\pi/2) \text{ modulo-}2\pi \text{ (radians)} \quad (6.3.1)$$

where $k$ is an integer number. These four possible rotations may also be represented by the binary-coded numbers 00, 01, 10 and 11. The truth tables for a typical differential precoder and its corresponding decoder are given in Table 6.3.1, where the four possible rotations in $(d_1)$, as given by the $(s_1)$, are represented by the changes in the vectors $(u_1)$ which, in turn, are given by the vectors $(b_1)$.

From equation 6.1.4, the eight possible phase rotations in a transmitted symbol $q_1$ are given by

$$\psi = k\pi/4 \text{ modulo-}2\pi \text{ (radians)} \quad (6.3.2)$$

for any integer $k$. Clearly, the four possible rotations in $d_1$ only includes half of all eight possible rotations of $q_1$ at the receiver. This point is further considered in Section 6.5.

If a convolutional coder is transparent to the four possible rotations given by equation 6.3.1, then a phase rotation of $\theta$ radians in $d_1$, that results in $d_1\exp(j\theta)$ (which gives a valid symbol-value of $d_1$), must also appear in the
corresponding code symbol $q_1$, to give $q_1 \exp(j\theta)$ (which is also a valid symbol-value of $q_1$). In addition, from Figure 6.1.2 it is clear that, a phase rotation of $\theta$ radians in $q_1$ changes the corresponding coded vector $v_1$, to any one of the four possible values 000, 010, 100 and 110, or else any one of four remaining values 001, 011, 101, and 111.

A example is given in this section to demonstrate the procedure for designing rate-$2/3$ rotationally invariant codes with an $8$PSK signal. The definition of the minimum free unitary distance, $d_{\text{free}}$, of the coded signals is given in Section 3.2.

A 4-state code was first designed. However it was then found that it was not possible to achieve the optimum asymptotic coding gain of 3.0 dB (see Table 3.2.1) and at the same time to satisfy the rule of equal rotation. The best 4-state code obtained has a minimum free unitary distance squared of $(2.0 + 1.1716)^2 = 3.1716$, which offers an asymptotic coding gain of 2.0 dB. Its transition and block diagrams are given in Figure 6.3.1.

An 8-state rotationally invariant code, with a maximum asymptotic coding gain of 3.0 dB, is shown in Figure 6.3.2. A particular implementation of this, that achieves the asymptotic coding gain of 3.0 dB, is shown in Figure 6.3.3. No improvement in minimum distance has been found with any other 8-state rotationally invariant codes tested. From Figure 6.3.2, the input vector to the convolutional coder is

$$u_1 = [ u_1(1) \quad u_1(2) ] \quad (6.3.3)$$
and the state of the coder $S_4$ is a 3-component vector

$$S_4 = [ u_{i-3}(2) \ u_{i-2}(2) \ u_{i-1}(2) ]$$  (6.3.4)

held in the shift register. Notice that the three components of $S_4$ comprises the second components of $u_{i-1}$, $u_{i-2}$, and $u_{i-3}$ only.

The effect of different phase rotations in the vectors $\{u_4\}$ is shown in Table 6.3.2, and Table 6.3.3 shows the effect of phase rotations on $u_4(2)$ alone. It can be seen here that a phase rotation of 0 and $\pi$ radians does not change $u_4(2)$, whereas a phase rotation of $\pi/2$ and $3\pi/2$ radians replaces $u_4(2)$ by its binary complement. Now return to equation 6.3.4. It is evident that a constant phase rotation of 0 or $\pi$ radians does not change $S_4$, whereas a constant phase rotation of $\pi/2$ or $3\pi/2$ radians replaces $S_4$ by its binary complement. This fact can also be seen in Table 6.3.4. Clearly, with the differential coding of $\{b_4(1)\}$ and $\{b_4(2)\}$ shown in Table 6.3.1, a rotationally invariant code which is transparent to phase rotations of multiples of $\pi/2$ radians is now obtained. Its block diagram is shown in Figure 6.3.3. The minimum distance squared of the code is 4.0, which corresponds to asymptotic coding gain of 3.0 dB.

Codes with other than 8 states have also been designed in the same way. It has been found that, with the appropriate design, the coding gain increases with the number of states of the coder. The best codes found for the given code constraint lengths are listed in Table 6.3.5. The table here shows, for each new code, the minimum free unitary distance squared
\( d^2 \text{free} \), the asymptotic coding gain over the uncoded QPSK system \( G_c \) (dB), and the asymptotic gain relative to that of the corresponding best known but not rotationally invariant code. From Table 6.3.5 it is clear that, with more than 4 states, the new codes lose only up to about 0.5 dB in the asymptotic coding gain, compared to the optimum codes of the same constraint lengths.

Clearly, the coder need not only to be implemented in the form of a feedforward transversal filter, but can also employ a feedback transversal filter, or else a look-up table. Various such forms have been tested in the search for better codes. An example is the 16-state code given in Table 6.3.5. This code requires the use of a look-up table. However, it offers an asymptotic coding gain of 3.6 dB, whereas the best possible rotationally invariant code, employing a 16-state feedforward transversal filter, has an asymptotic coding gain of 3.0 dB only.

Tables 6.3.6 to 6.3.9 present the truth tables of the four best rotationally invariant codes in Table 6.3.5. Each table here shows, for a given input vector

\[
u_i = [ u_i(1) \ u_i(2) ] \quad (6.3.5)
\]

the corresponding code symbol \( z_i \), where

\[
z_i = 4v_i(1) + 2v_i(2) + v_i(3) \quad (6.3.6)
\]

and the corresponding state transition (that from the state \( S_i \)
to $S_{i+1}$. In order to give a simple presentation, the values of $S_i$, $S_{i+1}$, and the input vector $u_i$ are given in each table as follows. From equation 3.1.16, the state $S_i$ of a coder may be generally defined by a $g$-component binary-coded vector

$$S_i = [s_i(1) s_i(2) \ldots s_i(g)]$$  \hspace{1cm} (6.3.7)

where $g$ is the memory of the code in bits. In Tables 6.3.6 to 6.3.9, each $S_i$ is represented by the corresponding integer value $S''_i$, where $S''_i$ is defined to be

$$S''_i = 2^{g-1}s_i(1) + 2^{g-2}s_i(2) + \ldots + 2^0s_i(g)$$  \hspace{1cm} (6.3.8)

Take the example in Table 6.3.6, where

$$S_i = [u_{i-2}(2) u_{i-1}(2)]$$  \hspace{1cm} (6.3.9)

Clearly,

$$S''_i = 2u_{i-2}(2) + u_{i-1}(2)$$  \hspace{1cm} (6.3.10)

The four possible values of $S_i$, which are [00], [01], [10] and [11], are here represented by the four integer values of $S''_i$, that are 0, 1, 2 and 3, respectively.

Similarly, the four possible values of the input vector $u_i$ are also represented by the four corresponding integer numbers $\{u''_i\}$. For a given value of $u_i$,

$$u''_i = 2u_i(1) + u_i(2)$$  \hspace{1cm} (6.3.11)

Figure 6.3.4 presents the performance results of the
four rate-2/3 rotationally invariant codes given in Tables 6.3.6 to 6.3.9, and contrasts these with that of the coherent DQPSK system. The results, obtained from a number of computer simulation tests, are shown as graphs of bit error rates against signal/noise ratio, $E_b/N_0$. $E_b$ is the average signal energy per data bit and $(1/2)N_0$ is the two-sided power spectral density of the additive white Gaussian noise. The accuracy of each curve, in the range of bit error rates of $10^{-3}$ to $10^{-4}$, is ±0.2 dB. The model of the data transmission system used in the computer simulation tests is described in Section 6.1, together it is further assumed that any phase slip of $\pm \pi/4$ radians in the $\{r'_1\}$ (which can not be tolerated by the decoder) is instantaneously corrected at the receiver.

Table 6.3.10 outlines the performances of each code at different bit error rates. From Table 6.3.10, at a bit error rate (BER) of $10^{-4}$, the four new codes offer the relative coding gains of 1.55 dB, 2.5 dB, 2.8 dB and 3.15 dB, respectively, over the uncoded DQPSK system. These compare well with those for the corresponding best known codes, which are 2.5 dB, 2.65 dB, 2.75 dB and 3.2 dB, respectively. Clearly, except for code RC(1) (a 4-state code), the performances of rotationally invariant codes are very close to those of the optimum codes.

The rate-2/3 rotationally invariant coded 8PSK signals here are transparent to phase rotations of multiples of $\pi/2$ radians. However, the reference carriers in the demodulation process may now locked onto a phase angle which is a multiple of $\pi/4$ radians. This leaves an irreducible phase error of $\pi/4$ radians, which can be eliminated by the phase error correction.
scheme described in Section 6.5.

Table 6.3.1: Differential coding and decoding

<table>
<thead>
<tr>
<th>$u_{i-1}$</th>
<th>$b_i$</th>
<th>$u_i$</th>
<th>$u'_{i-1}$</th>
<th>$u'_i$</th>
<th>$b'_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
<td>0 0</td>
</tr>
<tr>
<td>0 0</td>
<td>0 1</td>
<td>0 1</td>
<td>0 0</td>
<td>0 0</td>
<td>0 1</td>
</tr>
<tr>
<td>0 0</td>
<td>1 0</td>
<td>1 0</td>
<td>0 1</td>
<td>0 0</td>
<td>1 0</td>
</tr>
<tr>
<td>0 0</td>
<td>1 1</td>
<td>1 1</td>
<td>0 1</td>
<td>0 0</td>
<td>1 1</td>
</tr>
<tr>
<td>0 1</td>
<td>0 0</td>
<td>0 0</td>
<td>0 1</td>
<td>0 0</td>
<td>0 1</td>
</tr>
<tr>
<td>0 1</td>
<td>0 1</td>
<td>1 0</td>
<td>0 1</td>
<td>0 0</td>
<td>1 0</td>
</tr>
<tr>
<td>0 1</td>
<td>1 0</td>
<td>1 1</td>
<td>0 1</td>
<td>0 0</td>
<td>1 1</td>
</tr>
<tr>
<td>1 0</td>
<td>0 0</td>
<td>1 0</td>
<td>0 1</td>
<td>1 0</td>
<td>0 1</td>
</tr>
<tr>
<td>1 0</td>
<td>0 1</td>
<td>1 1</td>
<td>0 0</td>
<td>1 0</td>
<td>1 0</td>
</tr>
<tr>
<td>1 0</td>
<td>1 0</td>
<td>0 0</td>
<td>0 1</td>
<td>1 0</td>
<td>0 0</td>
</tr>
<tr>
<td>1 0</td>
<td>1 1</td>
<td>0 1</td>
<td>0 0</td>
<td>1 0</td>
<td>0 1</td>
</tr>
<tr>
<td>1 1</td>
<td>0 0</td>
<td>1 1</td>
<td>0 0</td>
<td>1 0</td>
<td>0 1</td>
</tr>
<tr>
<td>1 1</td>
<td>0 1</td>
<td>0 0</td>
<td>0 1</td>
<td>1 0</td>
<td>0 1</td>
</tr>
<tr>
<td>1 1</td>
<td>1 0</td>
<td>0 1</td>
<td>0 0</td>
<td>1 0</td>
<td>0 1</td>
</tr>
<tr>
<td>1 1</td>
<td>1 1</td>
<td>1 0</td>
<td>0 0</td>
<td>1 0</td>
<td>0 1</td>
</tr>
</tbody>
</table>
Table 6.3.2: Effects of phase rotations on differentially coded vector $u_1$

<table>
<thead>
<tr>
<th>$u_1$</th>
<th>Phase rotations (radians)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0, $\pi/2$, $\pi$, $3\pi/2$</td>
</tr>
<tr>
<td>$u_1(1)$</td>
<td>$u_1(2)$</td>
</tr>
<tr>
<td>0 0</td>
<td>0 0 0 1 1 0 1 1</td>
</tr>
<tr>
<td>0 1</td>
<td>0 1 1 0 1 1 0 0</td>
</tr>
<tr>
<td>1 0</td>
<td>1 0 1 1 0 0 0 1</td>
</tr>
<tr>
<td>1 1</td>
<td>1 1 0 0 0 1 1 0</td>
</tr>
</tbody>
</table>

Table 6.3.3: Effects of phase rotations on $u_1(2)$ alone

<table>
<thead>
<tr>
<th>$u_1(2)$</th>
<th>Phase rotations (radians)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0, $\pi/2$, $\pi$, $3\pi/2$</td>
</tr>
<tr>
<td>0</td>
<td>0 1 0 1</td>
</tr>
<tr>
<td>1</td>
<td>1 0 1 0</td>
</tr>
</tbody>
</table>
Table 6.3.4: Effects of phase rotations on the state of the coder, \( S_i \)

<table>
<thead>
<tr>
<th>State of coder, ( S_i )</th>
<th>Phase rotations (radians)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0, ( \pi/2, \pi, 3\pi/2 )</td>
</tr>
<tr>
<td>0 0 0</td>
<td>0 0 0 ( 1 1 1 ) 0 0 0 ( 1 1 1 )</td>
</tr>
<tr>
<td>0 0 1</td>
<td>0 0 1 ( 1 1 0 ) 0 0 1 ( 1 1 0 )</td>
</tr>
<tr>
<td>0 1 0</td>
<td>0 1 0 ( 1 0 1 ) 0 1 0 ( 1 0 1 )</td>
</tr>
<tr>
<td>0 1 1</td>
<td>0 1 1 ( 1 0 0 ) 0 1 1 ( 1 0 0 )</td>
</tr>
<tr>
<td>1 0 0</td>
<td>1 0 0 ( 0 1 1 ) 1 0 0 ( 0 1 1 )</td>
</tr>
<tr>
<td>1 0 1</td>
<td>1 0 1 ( 0 1 0 ) 1 0 1 ( 0 1 0 )</td>
</tr>
<tr>
<td>1 1 0</td>
<td>1 1 0 ( 0 0 1 ) 1 1 0 ( 0 0 1 )</td>
</tr>
<tr>
<td>1 1 1</td>
<td>1 1 1 ( 0 0 0 ) 1 1 1 ( 0 0 0 )</td>
</tr>
</tbody>
</table>

Table 6.3.5: Best rotationally invariant rate-2/3 codes for 8PSK signals

<table>
<thead>
<tr>
<th>Code</th>
<th>Number of states in the coder, ( 2^g )</th>
<th>Minimum free unitary distance, ( d_{\text{free}}^2 )</th>
<th>Asymptotic coding gain over uncoded QPSK signals, ( G_c ) (dB)</th>
<th>Coding gain relative to that of the best known code, (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC(1)</td>
<td>4</td>
<td>3.1716</td>
<td>2.0</td>
<td>-1.0</td>
</tr>
<tr>
<td>RC(2)</td>
<td>8</td>
<td>4.0</td>
<td>3.0</td>
<td>-0.6</td>
</tr>
<tr>
<td>RC(3)</td>
<td>16</td>
<td>4.5858</td>
<td>3.6</td>
<td>-0.5</td>
</tr>
<tr>
<td>RC(4)</td>
<td>32</td>
<td>5.1716</td>
<td>4.1</td>
<td>-0.5</td>
</tr>
</tbody>
</table>
Table 6.3.6: Truth table for the best rate-2/3, 4-state, rotationally invariant code RC(1)

<table>
<thead>
<tr>
<th>Current state, S_1</th>
<th>Next state S''_1, u''_1 = 0, 1, 2, 3</th>
<th>Current output z_1, u''_1 = 0, 1, 2, 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 1 0 1</td>
<td>0 2 4 6</td>
</tr>
<tr>
<td>1</td>
<td>2 3 2 3</td>
<td>1 3 5 7</td>
</tr>
<tr>
<td>2</td>
<td>0 1 0 1</td>
<td>1 3 5 7</td>
</tr>
<tr>
<td>3</td>
<td>2 3 2 3</td>
<td>0 2 4 6</td>
</tr>
</tbody>
</table>

Table 6.3.7: Truth table for the best rate-2/3, 8-state, rotationally invariant code RC(2)

<table>
<thead>
<tr>
<th>Current state, S''_1</th>
<th>Next state S''_1, u''_1 = 0, 1, 2, 3</th>
<th>Current output z_1, u''_1 = 0, 1, 2, 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 0 1</td>
<td>0 2 4 6</td>
<td></td>
</tr>
<tr>
<td>2 3 2 3</td>
<td>1 3 5 7</td>
<td></td>
</tr>
<tr>
<td>4 5 4 5</td>
<td>3 1 7 5</td>
<td></td>
</tr>
<tr>
<td>6 7 6 7</td>
<td>2 0 6 4</td>
<td></td>
</tr>
<tr>
<td>0 1 0 1</td>
<td>6 4 2 0</td>
<td></td>
</tr>
<tr>
<td>2 3 2 3</td>
<td>7 5 3 1</td>
<td></td>
</tr>
<tr>
<td>4 5 4 5</td>
<td>1 3 5 7</td>
<td></td>
</tr>
<tr>
<td>6 7 6 7</td>
<td>0 2 4 6</td>
<td></td>
</tr>
</tbody>
</table>
Table 6.3.8: Truth table for the best rate-2/3, 16-state, rotationally invariant code RC(3)

<table>
<thead>
<tr>
<th>Current state, $S''_i$</th>
<th>Next state $S''_i$, $u''_i = 0, 1, 2, 3$</th>
<th>Current output $z_i$, $u''_i = 0, 1, 2, 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 1 5 6</td>
<td>0 2 4 6</td>
</tr>
<tr>
<td>1</td>
<td>8 9 13 14</td>
<td>1 3 5 7</td>
</tr>
<tr>
<td>2</td>
<td>2 3 7 4</td>
<td>1 3 5 7</td>
</tr>
<tr>
<td>3</td>
<td>10 11 15 12</td>
<td>0 2 4 6</td>
</tr>
<tr>
<td>4</td>
<td>1 15 12 0</td>
<td>0 2 4 6</td>
</tr>
<tr>
<td>5</td>
<td>11 5 6 10</td>
<td>0 2 4 6</td>
</tr>
<tr>
<td>6</td>
<td>3 13 14 2</td>
<td>1 3 5 7</td>
</tr>
<tr>
<td>7</td>
<td>9 7 4 8</td>
<td>1 3 5 7</td>
</tr>
<tr>
<td>8</td>
<td>13 14 8 9</td>
<td>1 3 5 7</td>
</tr>
<tr>
<td>9</td>
<td>5 6 0 1</td>
<td>0 2 4 6</td>
</tr>
<tr>
<td>10</td>
<td>15 12 10 11</td>
<td>0 2 4 6</td>
</tr>
<tr>
<td>11</td>
<td>7 4 2 3</td>
<td>1 3 5 7</td>
</tr>
<tr>
<td>12</td>
<td>4 8 9 7</td>
<td>1 3 5 7</td>
</tr>
<tr>
<td>13</td>
<td>14 2 3 13</td>
<td>1 3 5 7</td>
</tr>
<tr>
<td>14</td>
<td>6 10 11 5</td>
<td>0 2 4 6</td>
</tr>
<tr>
<td>15</td>
<td>12 0 1 15</td>
<td>0 2 4 6</td>
</tr>
</tbody>
</table>
Table 6.3.9: Truth table for the best rate-2/3, 32-state, rotationally invariant code RC(4)

<table>
<thead>
<tr>
<th>Current state, $S''_i$</th>
<th>Next state $S''_{i+1}$, $u''_i = 0, 1, 2, 3$</th>
<th>Current output $z_i$, $u''_i = 0, 1, 2, 3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 1 2 3</td>
<td>0 2 4 6</td>
</tr>
<tr>
<td>1</td>
<td>4 5 6 7</td>
<td>1 3 5 7</td>
</tr>
<tr>
<td>2</td>
<td>8 9 10 11</td>
<td>2 0 6 4</td>
</tr>
<tr>
<td>3</td>
<td>12 13 14 15</td>
<td>3 1 7 5</td>
</tr>
<tr>
<td>4</td>
<td>16 17 18 19</td>
<td>3 1 7 5</td>
</tr>
<tr>
<td>5</td>
<td>20 21 22 23</td>
<td>4 6 0 2</td>
</tr>
<tr>
<td>6</td>
<td>24 25 26 27</td>
<td>5 7 1 3</td>
</tr>
<tr>
<td>7</td>
<td>28 29 30 31</td>
<td>2 0 6 4</td>
</tr>
<tr>
<td>8</td>
<td>0 1 2 3</td>
<td>2 0 6 4</td>
</tr>
<tr>
<td>9</td>
<td>4 5 6 7</td>
<td>3 1 7 5</td>
</tr>
<tr>
<td>10</td>
<td>8 9 10 11</td>
<td>0 2 4 6</td>
</tr>
<tr>
<td>11</td>
<td>12 13 14 15</td>
<td>3 5 3 1</td>
</tr>
<tr>
<td>12</td>
<td>16 17 18 19</td>
<td>5 7 1 3</td>
</tr>
<tr>
<td>13</td>
<td>20 21 22 23</td>
<td>2 0 6 4</td>
</tr>
<tr>
<td>14</td>
<td>24 25 26 27</td>
<td>3 1 7 5</td>
</tr>
<tr>
<td>15</td>
<td>28 29 30 31</td>
<td>4 6 0 2</td>
</tr>
<tr>
<td>16</td>
<td>0 1 2 3</td>
<td>4 6 0 2</td>
</tr>
<tr>
<td>17</td>
<td>4 5 6 7</td>
<td>5 7 1 3</td>
</tr>
<tr>
<td>18</td>
<td>8 9 10 11</td>
<td>6 4 2 0</td>
</tr>
<tr>
<td>19</td>
<td>12 13 14 15</td>
<td>7 5 3 1</td>
</tr>
<tr>
<td>20</td>
<td>16 17 18 19</td>
<td>7 5 3 1</td>
</tr>
<tr>
<td>21</td>
<td>20 21 22 23</td>
<td>0 2 4 6</td>
</tr>
<tr>
<td>22</td>
<td>24 25 26 27</td>
<td>1 3 5 7</td>
</tr>
<tr>
<td>23</td>
<td>28 29 30 31</td>
<td>6 4 2 0</td>
</tr>
<tr>
<td>24</td>
<td>0 1 2 3</td>
<td>6 4 2 0</td>
</tr>
<tr>
<td>25</td>
<td>4 5 6 7</td>
<td>7 5 3 1</td>
</tr>
<tr>
<td>26</td>
<td>8 9 10 11</td>
<td>4 6 0 2</td>
</tr>
<tr>
<td>27</td>
<td>12 13 14 15</td>
<td>5 7 1 3</td>
</tr>
<tr>
<td>28</td>
<td>16 17 18 19</td>
<td>1 3 5 7</td>
</tr>
<tr>
<td>29</td>
<td>20 21 22 23</td>
<td>6 4 2 0</td>
</tr>
<tr>
<td>30</td>
<td>24 25 26 27</td>
<td>7 5 3 1</td>
</tr>
<tr>
<td>31</td>
<td>28 29 30 31</td>
<td>0 2 4 6</td>
</tr>
</tbody>
</table>
### Table 6.3.10: Performances of rate-2/3 rotationally invariant coded 8PSK signals at different bit error rates

<table>
<thead>
<tr>
<th>Number of states in the coder, $2^g$</th>
<th>Gain in tolerance to noise over DQPSK signals at a given bit error rate, (dB)</th>
<th>Coding gain relative to the corresponding rate-2/3 convolutional code, (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BER=0.01, 0.001, 0.0001, 0.01, 0.001, 0.0001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RC(1) 4</td>
<td>0.05 1.25 1.55</td>
<td>-0.55 -0.75 -0.95</td>
</tr>
<tr>
<td>RC(2) 8</td>
<td>1.0 2.0 2.5</td>
<td>0.05 -0.05 -0.15</td>
</tr>
<tr>
<td>RC(3) 16</td>
<td>1.05 2.2 2.8</td>
<td>0.1 -0.1 -0.15</td>
</tr>
<tr>
<td>RC(4) 32</td>
<td>1.05 2.35 3.15</td>
<td>0.1 -0.05 -0.05</td>
</tr>
</tbody>
</table>
Fig. 6.3.1: A rate-2/3, 4-state, rotationally invariant convolutional coder for 8PSK signals
Fig. 6.3.2: A rate-2/3 coder which can achieve a maximum asymptotic coding gain of 3.0 dB.
Fig. 6.3.3: An 8-state rotationally invariant coder for 8PSK signals
Fig. 6.3.4: Performances of rate-2/3 rotationally invariant convolutionally coded 8PSK signals with Viterbi decoding.
6.4 RATE-3/4 CODES THAT ARE ROTATIONALLY INVARIANT TO PHASE CHANGES OF $\pm \pi/4$, $\pm \pi/2$, $\pm 3\pi/4$, AND $\pi$ RADIANS

In this section, rotationally invariant rate-3/4 convolutional codes for 16PSK signals are described. A coded 16PSK signal here carries three information bits and one redundant bit per symbol. The rotationally invariant codes are transparent to the phase rotation of $\theta$ radians, where

$$\theta = (k\pi/4) \mod 2\pi \text{ (radians)} \quad (6.4.1)$$

$k$ is an integer number. This includes half of all 16 possible rotations in a transmitted symbol $q_i$, which are

$$\psi = (k\pi/8) \mod 2\pi \text{ (radians)} \quad (6.4.2)$$

for any integer number $k$. This point will be further discussed in Section 6.5.

The basic structure of the coder considered here is shown in Figure 6.4.1. This offers a maximum asymptotic coding gain of $10\log_{10}(2.0/0.5858)=5.3$ dB, and includes all short codes of practical interest. The codes have a very similar structure as those of the conventional rate-3/4 convolutional codes in Section 3.2, in that the corresponding Viterbi decoder also operates on a binary signal. The two non-convolutionally coded bits $u_i(1)$ and $u_i(2)$ are also Gray-coded, as shown in Figure 6.1.3.

The design of the above codes employs the same techniques as those applied to the rate-2/3 codes with an 8PSK
signal. The codes were designed by hand and their minimum free unitary distances can be checked with the aid of a computer program. The coding gains of the best codes obtained for various constraint lengths are listed in Table 6.4.1, where they are compared with the corresponding optimum rate-3/4 convolutional codes (that are not constrained to be rotationally invariant). Very attractive codes with 8 and 32 states have been found. From Table 6.4.1 these two codes achieve the optimum minimum free unitary distances and, in addition, they are rotationally invariant to phase changes that are multiples of $\pi/4$ radians.

Tables 6.4.2 to 6.4.5 present the truth tables for those codes in Table 6.4.1. Each table here shows, for a given input vector $u_i$, where

$$u_i = [ u_i(1) \ u_i(2) \ u_i(3) ] \quad \ldots (6.4.3)$$

the corresponding code symbol

$$z_i = M_1(v_i) \quad (6.4.4)$$

and the corresponding state transition (that from the state $S_i$ to $S_{i+1}$). The mapping function $M_1(.)$ is defined in Figure 6.1.3. In order to give a simple presentation, each $S_i$, which is normally defined by a $g$-component binary coded vector

$$S_i = [ s_i(1) \ s_i(2) \ \ldots \ s_i(g) ] \quad (6.4.5)$$

for any code with a memory of $g$ bits, is represented in each
table here by the corresponding integer number $S''_1$. For a given value of $S_1$,

$$S''_1 = 2^{g-1}s_1(1) + 2^{g-2}s_1(2) + \ldots + 2^0s_1(g) \quad (6.4.6)$$

Take the example in Table 6.4.2, where

$$S_1 = [ u_{i-2}(3) \ u_{i-1}(3) ] \quad (6.4.7)$$

Here, $S''_1 = 2u_{i-2}(3) + u_{i-1}(3) \quad (6.4.8)$

The four possible values of $S_1$, that are [00], [01], [10] and [11], are presented in the table by the corresponding integer numbers 0, 1, 2 and 3, respectively. Similarly, each input vector $u_i$ is given by the corresponding integer value $u''_i$. The mapping of binary-coded vectors $\{u_i\}$ to the integer numbers $\{u''_i\}$ is also shown in Figure 6.1.3.

Figure 6.4.2 presents the performance results of the four rate-3/4 rotationally invariant codes shown in Tables 6.4.2 to 6.4.5, and contrasts these with that of the coherent D8PSK system. The results, obtained from a number of computer simulation tests, are shown as graphs of bit error rates against signal/noise ratio, $E_b/N_0$. The accuracy of each curve, over the range of bit error rates of $10^{-3}$ to $10^{-4}$, is ±0.2 dB. The model of the data transmission system used in the computer simulation tests is described in Section 6.1, together it is further assumed that any fixed phase shift of $\pm \pi/8$ radians in the received signal is instantaneously corrected at the receiver.
Table 6.4.6 outlines the performances of each code at different bit error rates. From Table 6.4.6, at a BER of $10^{-4}$, the four new codes provide the relative coding gains of 1.4 dB, 2.5 dB, 2.6 dB and 2.75 dB, respectively, over the uncoded D8PSK system, and those of the optimum codes are 2.3 dB, 2.6 dB, 2.8 dB and 2.9 dB, respectively. Clearly, except for the code RC(4) (a 4-state code), the performances of rotationally invariant codes here are close to those of the optimum codes.

The rate-$3/4$ rotationally invariant coded 16PSK signals described in this section are transparent to phase rotations that are multiples of $\pi/4$ radians. However, the reference carriers in the demodulation process may now be locked onto a phase angle which is a multiple of $\pi/8$ radians. This leaves an irreducible phase error of $\pi/8$ radians, which may be eliminated by the scheme described later in Section 6.5.
Table 6.4.1: Best rotationally invariant rate-3/4 convolutional codes for 16PSK signals

<table>
<thead>
<tr>
<th>Code</th>
<th>Number of states in the coder, 2^k</th>
<th>Minimum free unitary distance, d^2_free</th>
<th>Asymptotic coding gain ( G_c ) (dB)</th>
<th>Coding gain relative to that of the best known code, (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC(4)</td>
<td>4</td>
<td>0.8902</td>
<td>1.8</td>
<td>-1.7</td>
</tr>
<tr>
<td>RC(5)</td>
<td>8</td>
<td>1.4760</td>
<td>4.0</td>
<td>0.0</td>
</tr>
<tr>
<td>RC(6)</td>
<td>16</td>
<td>1.4760</td>
<td>4.0</td>
<td>-0.4</td>
</tr>
<tr>
<td>RC(7)</td>
<td>32</td>
<td>1.7804</td>
<td>4.8</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 6.4.2: Truth table of the best rate-3/4, 4-state, rotationally invariant code, RC(5)

<table>
<thead>
<tr>
<th>Current state, ( S''_1 )</th>
<th>Next state ( S''<em>1 ), ( u''</em>{i=0}, 1, 2, 3, 4, 5, 6, 7 )</th>
<th>Current output ( z_1 ), ( u'_1=0, 1, 2, 3, 4; 5, 6, 7 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0 1 0 1 0 1 0 1</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
<tr>
<td>1</td>
<td>2 3 2 3 2 3 2 3</td>
<td>1 3 5 7 9 11 13 15</td>
</tr>
<tr>
<td>2</td>
<td>0 1 0 1 0 1 0 1</td>
<td>1 3 5 7 9 11 13 15</td>
</tr>
<tr>
<td>3</td>
<td>2 3 2 3 2 3 2 3</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
</tbody>
</table>
Table 6.4.3: Truth table of the best rate-3/4, 8-state, rotationally invariant code RC(6)

<table>
<thead>
<tr>
<th>Current state</th>
<th>Next state ( S_i^\prime )</th>
<th>Current output ( z_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( S_i ) ( u_i = 0 )</td>
<td>( u_i = 1 )</td>
<td>( u_i = 0 )</td>
</tr>
<tr>
<td>0 0 1 0 1 0 1 0 1 0 2 4 6 8 10 12 14</td>
<td>1 2 3 2 3 2 3 2 3</td>
<td>1 3 5 7 9 11 13 15</td>
</tr>
<tr>
<td>1 4 5 4 5 4 5 4 5</td>
<td>2 0 6 4 10 8 14 12</td>
<td>14 4 2 8 6 12 10 0</td>
</tr>
<tr>
<td>2 6 7 6 7 6 7 6 7</td>
<td>1 3 5 7 9 11 13 15</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
<tr>
<td>3 0 1 0 1 0 1 0 1</td>
<td>14 4 2 8 6 12 10 0</td>
<td>15 5 3 9 7 13 11 1</td>
</tr>
<tr>
<td>4 2 3 2 3 2 3 2 3</td>
<td>1 3 5 7 9 11 13 15</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
<tr>
<td>5 4 5 4 5 4 5 4 5</td>
<td>2 0 6 4 10 8 14 12</td>
<td>14 4 2 8 6 12 10 0</td>
</tr>
<tr>
<td>6 6 7 6 7 6 7 6 7</td>
<td>1 3 5 7 9 11 13 15</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
<tr>
<td>7 8 9 8 9 8 9 8 9</td>
<td>2 0 6 4 10 8 14 12</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
<tr>
<td>8 10 11 10 11 10 11 10 11</td>
<td>1 3 5 7 9 11 13 15</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
<tr>
<td>9 12 13 12 13 12 13 12 13</td>
<td>3 1 7 5 11 9 15 13</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
<tr>
<td>10 14 15 14 15 14 15 14 15</td>
<td>2 0 6 4 10 8 14 12</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
<tr>
<td>11 12 13 12 13 12 13 12 13</td>
<td>3 1 7 5 11 9 15 13</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
<tr>
<td>12 14 15 14 15 14 15 14 15</td>
<td>2 0 6 4 10 8 14 12</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
<tr>
<td>13 16 15 16 15 16 15 16 16</td>
<td>1 3 5 7 9 11 13 15</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
<tr>
<td>14 17 16 17 16 17 16 17 17</td>
<td>1 3 5 7 9 11 13 15</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
<tr>
<td>15 18 17 18 17 18 17 18 18</td>
<td>1 3 5 7 9 11 13 15</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
</tbody>
</table>

Table 6.4.4: Truth table of the best rate-3/4 16-state rotationally invariant code RC(7)
Table 6.4.5: Truth table of the best rate-$3/4$, 32-state, rotationally invariant code RC(8)

<table>
<thead>
<tr>
<th>Current state, $S''_i$</th>
<th>Next state $S'''_i$, $u''_i$</th>
<th>Current output $z'_i$, $u'_i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$u''_i=0, 1, 2, 3, 4, 5, 6, 7$</td>
<td>$u'_i=0, 1, 2, 3, 4, 5, 6, 7$</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>0 1 0 1 0 1 0 1</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
<tr>
<td>1</td>
<td>2 3 2 3 2 3 2 3</td>
<td>1 3 5 7 9 11 13 15</td>
</tr>
<tr>
<td>2</td>
<td>4 5 4 5 4 5 4 5</td>
<td>2 0 6 4 10 8 14 12</td>
</tr>
<tr>
<td>3</td>
<td>6 7 6 7 6 7 6 7</td>
<td>3 1 7 5 11 9 15 13</td>
</tr>
<tr>
<td>4</td>
<td>8 9 8 9 8 9 8 9</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
<tr>
<td>5</td>
<td>10 11 10 11 10 11 10 11</td>
<td>1 3 5 7 9 11 13 15</td>
</tr>
<tr>
<td>6</td>
<td>12 13 12 13 12 13 12 13</td>
<td>2 0 6 4 10 8 14 12</td>
</tr>
<tr>
<td>7</td>
<td>14 15 14 15 14 15 14 15</td>
<td>3 1 7 5 11 9 15 13</td>
</tr>
<tr>
<td>8</td>
<td>16 17 16 17 16 17 16 17</td>
<td>1 3 5 7 9 11 13 15</td>
</tr>
<tr>
<td>9</td>
<td>18 19 18 19 18 19 18 19</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
<tr>
<td>10</td>
<td>20 21 20 21 20 21 20 21</td>
<td>3 1 7 5 11 9 15 13</td>
</tr>
<tr>
<td>11</td>
<td>22 23 22 23 22 23 22 23</td>
<td>2 0 6 4 10 8 14 12</td>
</tr>
<tr>
<td>12</td>
<td>24 25 24 25 24 25 24 25</td>
<td>1 3 5 7 9 11 13 15</td>
</tr>
<tr>
<td>13</td>
<td>26 27 26 27 26 27 26 27</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
<tr>
<td>14</td>
<td>28 29 28 29 28 29 28 29</td>
<td>3 1 7 5 11 9 15 13</td>
</tr>
<tr>
<td>15</td>
<td>30 31 30 31 30 31 30 31</td>
<td>2 0 6 4 10 8 14 12</td>
</tr>
<tr>
<td>16</td>
<td>0 1 0 1 0 1 0 1</td>
<td>14 4 2 8 6 12 10 0</td>
</tr>
<tr>
<td>17</td>
<td>2 3 2 3 2 3 2 3</td>
<td>15 5 3 9 7 13 11 1</td>
</tr>
<tr>
<td>18</td>
<td>4 5 4 5 4 5 4 5</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
<tr>
<td>19</td>
<td>6 7 6 7 6 7 6 7</td>
<td>1 3 5 7 9 11 13 15</td>
</tr>
<tr>
<td>20</td>
<td>8 9 8 9 8 9 8 9</td>
<td>14 4 2 8 6 12 10 0</td>
</tr>
<tr>
<td>21</td>
<td>10 11 10 11 10 11 10 11</td>
<td>15 5 3 9 7 13 11 1</td>
</tr>
<tr>
<td>22</td>
<td>12 13 12 13 12 13 12 13</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
<tr>
<td>23</td>
<td>14 15 14 15 14 15 14 15</td>
<td>1 3 5 7 9 11 13 15</td>
</tr>
<tr>
<td>24</td>
<td>16 17 16 17 16 17 16 17</td>
<td>15 5 3 9 7 13 11 1</td>
</tr>
<tr>
<td>25</td>
<td>18 19 18 19 18 19 18 19</td>
<td>14 4 2 8 6 12 10 0</td>
</tr>
<tr>
<td>26</td>
<td>20 21 20 21 20 21 20 21</td>
<td>1 3 5 7 9 11 13 15</td>
</tr>
<tr>
<td>27</td>
<td>22 23 22 23 22 23 22 23</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
<tr>
<td>28</td>
<td>24 25 24 25 24 25 24 25</td>
<td>15 5 3 9 7 13 11 1</td>
</tr>
<tr>
<td>29</td>
<td>26 27 26 27 26 27 26 27</td>
<td>14 4 2 8 6 12 10 0</td>
</tr>
<tr>
<td>30</td>
<td>28 29 28 29 28 29 28 29</td>
<td>1 3 5 7 9 11 13 15</td>
</tr>
<tr>
<td>31</td>
<td>30 31 30 31 30 31 30 31</td>
<td>0 2 4 6 8 10 12 14</td>
</tr>
</tbody>
</table>
Table 6.4.6: Performances of rate-3/4 rotationally invariant coded 16PSK signals at different bit error rates

<table>
<thead>
<tr>
<th>Number of states in the coder, Code</th>
<th>Gain in tolerance to noise over D8PSK signals at a given bit error rate, (dB)</th>
<th>Coding gain relative to that of the corresponding rate-3/4 code, (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RC(5) 4</td>
<td>0.1 0.6 1.4</td>
<td>-0.45 -0.85 -0.9</td>
</tr>
<tr>
<td>RC(6) 8</td>
<td>0.3 1.45 2.5</td>
<td>-0.3 -0.15 -0.1</td>
</tr>
<tr>
<td>RC(7) 16</td>
<td>0.3 1.5 2.6</td>
<td>-0.35 -0.25 -0.2</td>
</tr>
<tr>
<td>RC(8) 32</td>
<td>0.3 1.55 2.75</td>
<td>-0.4 -0.3 -0.15</td>
</tr>
</tbody>
</table>

BER=0.01, 0.001, 0.0001

<table>
<thead>
<tr>
<th>BER=0.01, 0.001, 0.0001</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Fig. 6.4.1: A rotationally invariant rate-$3/4$ convolutional coder
Fig. 6.4.2: Performances of rate-3/4 rotationally invariant convolutionally coded 16PSK signals with the Viterbi decoding.
6.5 A CARRIER-PHASE CORRECTION SYSTEM

The rate-2/3 rotationally invariant coded 8PSK signals of Section 6.3 can tolerate phase shifts that are multiples of \( \pi/2 \) radians. However the reference carriers at the receiver may become locked onto a phase angle that differs from the correct phase position by any multiple of \( \pi/4 \) radians. This leaves an irreducible phase error of \( \pi/4 \) radians. The receiver must therefore be able to identify this phase error whenever it occurs.

A promising technique for achieving this is to obtain a measure of the short-term mean-square difference between the received samples \( \{r'_i\} \) at the decoder input and the corresponding code symbols \( \{d_i\} \) that would have been received in the absence of noise, had the transmitted sequence of data symbols (differentially-coded) being given by the decoded sequence \( \{u'_i\} \). An approximate measure of the required mean-square difference may be obtained in the following manner. The operation of the Viterbi decoder is described in Section 3.3.

Towards the end of \((i+1)\)th decoding process, the first component \( x_{i-n} \) of the lowest-cost vector \( Z_1 \), is taken to be the decoded value \( u'_{i-n} \) of \( u_{i-n} \) (see Section 3.3 for details). Now let the cost associated with this lowest-cost vector (before being set to zero) be \( \Delta_1 \), then required mean-square difference may be determined by equation 6.5.1.

\[
\beta_{i+1} = \beta_i + \varepsilon(\Delta_1 - \beta_i) = \beta_i(1-\varepsilon) + \varepsilon\Delta_i
\]

\( \beta_0 = 0.0 \)
where the constant $\varepsilon$ is a small positive number ($\varepsilon<<1$), and the appropriate value for $\varepsilon$ must be determined experimentally for any given code.

The phase-error correction system proposed for the coded 8PSK signals simply operates as follows. At the end of $(i+1)^{th}$ decoding process, $\Delta_i$ is determined, and $\beta_{i+1}$ is calculated using equation 6.5.1. Whenever $\beta_{i+1}$ exceeds a given threshold level $H$, it is assumed that a fixed phase rotation of $\pi/4$ radians has been introduced into the received samples $\{r'_i\}$. A phase rotation of $\pi/4$ radians in either direction is therefore applied to the $\{r'_i\}$, so that, if the assumption is correct, there is now a resultant phase rotation of a multiple of $\pi/2$ radians, which can be handled correctly by the decoder.

Clearly, the phase error correction system here needs only to consider two cases, that is, whether a phase slip of a multiple of $\pi/4$ radians has, or has not, been introduced to the reference carriers of the coherent demodulator. On the other hand, however, if a non-rotationally invariant code is used, although the phase slip can be recognised in the similar manner, its actual value must be determined before the phase slip can be corrected. Unfortunately, this is likely to involve considerably more time and additional equipment complexity.

For any given code, the two constants $\varepsilon$ and $H$ must be appropriately chosen to achieve the best compromise between the following two factors. The first factor is the probability of falsely detecting a phase error when none has actually occurred. This probability is denoted by $P(E/C)$. Notice that when this happens, a phase error of $\pi/4$ radians will be
introduced into the subsequent received samples \( r'_i \). The second factor concerned is the time taken to recognise a phase slip of a multiple of \( \pi/4 \) radians after one has been introduced. This time delay is denoted by \( T_d \) and is measured as an integer multiple of \( T \), where \( T \) is the symbol interval in seconds. Clearly, the optimisation of the system parameters must be made to minimise both \( P(E/C) \) and \( T_d \), for the most likely signal/noise ratio.

A number of computer simulations were carried out to test the performances of the system with different values of \( \epsilon \) and \( H \). The code RC(2) (a 8-state code, see Table 6.3.7) was used in these tests. Two types of statistical information obtained from these tests are presented. The first type was obtained in the absence of any phase slips and is shown in Figure 6.5.1, whereas the second type of results was obtained with a phase error of \( \pi/4 \) radians and is shown in Table 6.5.1.

Figure 6.5.1 shows, at each given value of \( \epsilon \), the percentage of total transmission time when \( \beta_1 \) (see equation 6.5.1) is greater or equal to the given value of \( b \), for \( 0 \leq b \leq 0.40 \), at \( E_b/N_0=5.3 \) dB (which approximates to a BER of \( 10^{-3} \)). For a given value of \( \epsilon \), the threshold level \( H \) must be chosen for which there is a low probability of \( \beta_1>H \) when no phase error has been introduced. Take the example with \( \epsilon=0.01 \), in Figure 6.5.1. Here, the threshold value \( H \) should not be less than 0.19 (approximately), in order to reduce \( P(E/C) \) to a negligible value. On the other hand, the threshold level \( H \) should also be chosen as low as it can be allowed, in order to minimise \( T_d \) whenever a phase slip of \( \pi/4 \) radians occurs.

Table 6.5.1 outlines, for each given combination of \( \epsilon \)
and $H$ tested, the approximate average time taken to recognise a phase slip of $\pi/4$ radians, after one has been introduced into the received samples $\{r'_1\}$. This average value for $T_d$ was obtained with the detection of around 2000 phase slips of any multiples of $\pi/4$ radians in the received samples. The results here suggest that, with the appropriately selected constants $\varepsilon$ and $H$, a phase slip of $\pi/4$ radians can be detected, on average, within 100 to 200 data symbols, using the given technique. Take the example with $\varepsilon=0.04$ and $H=0.25$. A phase error can, in fact, be detected within the average of about 125 symbols.

The rate-$3/4$ rotationally invariant coded 16PSK signals of Section 6.4, on the other hand, are transparent to phase shifts that are multiples of $\pi/4$ radians. However, the reference carriers in the coherent demodulation process may now lock onto a phase angle which is a multiple of $\pi/8$ radians. This leaves an irreducible phase error of $\pi/8$ radians, which can also be eliminated by the technique just described for the rate-$2/3$ codes.

Similar computer simulation tests can also be carried out to determine the suitable values of $\varepsilon$ and $H$, for the various rate-$3/4$ codes of Section 6.4. It is expected that a phase slip of any multiple of $\pi/8$ radians in the reference carriers of the coherent demodulator may be detected with a similar delay as that for the rate-$2/3$ codes above.

In practice, it is unlikely that more than one or two phase slips will normally occur during any one transmission of around $10^6$ data symbols, with normally very many more data symbols being received between adjacent phase slips [101]. Again, since a phase slip does not usually occur
instantaneously, it is likely to be accomplished by several errors in the decoded data symbols, even when the phase slip is corrected very rapidly [101]. Therefore, a delay of some 200 or 300 symbol intervals in the correction of a phase error is not likely a serious disadvantage of the system.

Table 6.5.1: Performances of the phase error correction system with different values of $\varepsilon$ and $H$

<table>
<thead>
<tr>
<th>System parameters, $\varepsilon$, $H$</th>
<th>Average time delay required to recognise a phase slip of $\pi/4$ radians, $T_d$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.005 0.18</td>
<td>217T</td>
</tr>
<tr>
<td>0.01 0.20</td>
<td>144T</td>
</tr>
<tr>
<td>0.02 0.23</td>
<td>155T</td>
</tr>
<tr>
<td>0.04 0.25</td>
<td>125T</td>
</tr>
<tr>
<td>0.04 0.26</td>
<td>183T</td>
</tr>
<tr>
<td>0.06 0.28</td>
<td>210T</td>
</tr>
<tr>
<td>0.08 0.30</td>
<td>253T</td>
</tr>
<tr>
<td>0.08 0.32</td>
<td>680T</td>
</tr>
</tbody>
</table>

Common attributes: RC(2); 8PSK; $E_b/N_0=5.3$ dB; BER=0.001
Fig. 6.5.1: Percentage of total transmission time when $\beta_i$ is greater or equal to $b$, for code RC[$2$] and at BER=0.001

Legend

- \(\triangle\) when $\epsilon = 0.005$
- \(\times\) when $\epsilon = 0.01$
- \(\square\) when $\epsilon = 0.02$
- \(\blacktriangle\) when $\epsilon = 0.04$
- \(\blacktriangledown\) when $\epsilon = 0.06$
- \(\times\) when $\epsilon = 0.08$

Common attributes: rate-2/3 coded 8PSK; Eb/No=5.3 dB
Viterbi decoding; $n=32$
CHAPTER SEVEN

DISCUSSION OF RESULTS
This chapter consists of four parts. Section 7.1 compares the performances and equipment complexities of the four decoding schemes studied in this work, and Section 7.2 presents a performance evaluation for the two types of coded MPSK signals considered. In Section 7.3, a comment is given on the originality of this investigation, and finally Section 7.4 outlines possible further research work linked with the present study.

7.1 COMPARISONS OF THE MORE PROMISING DECODING SCHEMES

For coded 8PSK signals, four different types of decoding schemes have been investigated in this work. These include the well known maximum likelihood Viterbi algorithm, and a more conventional near-maximum likelihood scheme (system A) which was originally proposed for the detection of a digital signal over a linear channel with intersymbol interference. Also included in this study are the two novel noise-adaptive system A-type decoders, known as systems A1 and A2. In this section, these four decoding techniques are compared in terms of their tolerances to noise, for a given bit error rate and for a given degree of decoder complexity.

Figures 7.1.1 to 7.1.4 contrast the performances of the four decoding schemes incorporating the same coded 8PSK signals. The performances are shown as graphs of bit error rate (BER) against the signal/noise ratio, defined as $E_b/N_0$. $E_b$ is the average signal energy per data bit and $(1/2)N_0$ is the two-sided power spectral density of the white Gaussian noise.
The accuracy of each curve, in all graphs, is ±0.2 dB, over the range of bit error rates of $10^{-3}$ to $10^{-4}$. Four codes are used in the comparisons. These include three rate-2/3 convolutional codes, UC(3), UC(4), and UC(5) (see Table 3.2.1), and one modulo-8 correlative-level code, CC(2) (see Table 4.3.1). Among these, codes UC(3) and UC(4) have 16 and 32 states, respectively, whereas codes UC(5) and CC(2) have 64 states.

Clearly, an important property of any cost-effective decoding scheme is its tolerance to additive white Gaussian noise. This is measured by the signal/noise ratio in the received signal, which is required to achieve a desired BER, for the given degree of equipment complexity of the decoder. The latter complexity has been described in this work by the total number of stored vectors used in the decoder together with the total number of cost evaluations that the decoder must be able to perform, over each symbol interval $T$ (seconds).

Tables 7.1.1 to 7.1.4 summarizes, at the given BER of $10^{-3}$, the tolerance of each decoder to noise, the relative equipment complexity, and the corresponding average number of bit errors per burst. The definition of error bursts is given in Appendix A5, and the relative equipment complexities of the different decoders are presented in the following manner.

Firstly, the comparison of the decoder complexities is made on the basis of the corresponding total numbers of cost evaluations involved (which are also the total numbers of expanded vectors that the decoders must handle), within each symbol interval, for achieving the given tolerances to noise. Consequently, the integer numbers shown under the column "Relative complexity of the decoder", in each table here, in
fact, represent the corresponding numbers of cost evaluations per symbol interval, for the different decoders. Take the example with the Viterbi decoder in Table 7.1.1. Here, the Viterbi decoder performs \(4 \times 16 = 64\) cost evaluations per symbol interval, with a total of 16 stored vectors.

Secondly, the comparison made in each table allows systems A1 and A2 to use two buffer stores (see Section 5.3 for details), each having a storage of 300 elements (samples), at no extra implementation cost. The corresponding integer numbers shown for systems A1 and A2, under the column "Relative complexity of the decoder" of each table, actually represent the minimum numbers of cost evaluations per symbol interval, that system A1 or A2 must be able to handle with a low probability of buffer overflow, for the given size of the buffer stores. The conclusion of Section 5.3 is that, with the size of 300 samples, the number of cost evaluations that can be handled by system A1 or A2, within the given symbol interval \(T\), must not be less than \(1.2 \overline{f}_1\) (where \(\overline{f}_1\) is the average number of cost evaluations per decoded data symbol), in order to reduce to a negligible value the probability that the input buffer store is filled.

From Tables 7.1.1 to 7.1.4 it is clear that, system A2 is potentially more cost-effective than the other three decoders, in that it can provide a considerable reduction in the complexity of the decoder with little degradation in tolerance to noise, relative to that of the optimum decoder. For instance, with code UC(5) (see Table 7.1.3), a reduction of more than three-quarters in the decoder complexity is achieved, relative to that of the Viterbi decoder. But the corresponding
degradation in tolerance to noise is only up to 0.25 dB. Take another example with code UC(3) in Table 7.1.1. Again, a degradation of 0.15 dB in tolerance to noise is accomplished by a reduction of up to about three-quarters in complexity.

Another interesting comparison between the various decoders is their burst error characteristics. Compared to the other schemes, system A generally experiences severely prolonged error bursts, except for the cases where it uses the same number of stored vectors as the corresponding Viterbi decoder (see Table 7.1.1). On the other hand, systems A1 and A2 have a quite similar burst error characteristic as the Viterbi decoder. It is evident that, by employing the "noise-adaptive" technique, the burst error characteristics of the decoder can be greatly improved without significantly increasing the average number of operations per decoded data symbol. The penalty paid for using the "noise-adaptive" scheme, however, is the two buffer stores that are required at the input and output of the decoder.
Table 7.1.1: Comparisons of the four decoding schemes incorporating code UC(3), at BER=10^{-3}

<table>
<thead>
<tr>
<th>Decoder,</th>
<th>Number of stored vectors</th>
<th>Gain in tolerance to noise over QPSK signals, (dB)</th>
<th>Relative complexity of the decoder,</th>
<th>Average number of bit errors per burst,</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viterbi</td>
<td>$2^g=16$</td>
<td>1.7</td>
<td>64</td>
<td>9</td>
</tr>
<tr>
<td>System A</td>
<td>$m=4$</td>
<td>-0.2</td>
<td>16</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>$m=8$</td>
<td>1.2</td>
<td>32</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>$m=16$</td>
<td>1.65</td>
<td>64</td>
<td>12</td>
</tr>
<tr>
<td>System A1</td>
<td>$\bar{m}_1=4.4$</td>
<td>1.55</td>
<td>22</td>
<td>14</td>
</tr>
<tr>
<td>System A2</td>
<td>$\bar{m}_1=4.3$</td>
<td>1.55</td>
<td>10</td>
<td>16</td>
</tr>
</tbody>
</table>

Table 7.1.2: Comparisons of the four decoding schemes incorporating code UC(4), at BER=10^{-3}

<table>
<thead>
<tr>
<th>Decoder,</th>
<th>Number of stored vectors</th>
<th>Gain in tolerance to noise over QPSK signals, (dB)</th>
<th>Relative complexity of the decoder,</th>
<th>Average number of bit errors per burst,</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viterbi</td>
<td>$2^g=32$</td>
<td>1.8</td>
<td>128</td>
<td>9</td>
</tr>
<tr>
<td>System A</td>
<td>$m=8$</td>
<td>0.6</td>
<td>32</td>
<td>51</td>
</tr>
<tr>
<td></td>
<td>$m=16$</td>
<td>1.55</td>
<td>64</td>
<td>26</td>
</tr>
<tr>
<td>System A1</td>
<td>$\bar{m}_1=7.8$</td>
<td>1.75</td>
<td>38</td>
<td>16</td>
</tr>
<tr>
<td>System A2</td>
<td>$\bar{m}_1=7.6$</td>
<td>1.8</td>
<td>18</td>
<td>12</td>
</tr>
</tbody>
</table>
### Table 7.1.3: Comparisons of the four decoding schemes incorporating code UC(5), at BER=10^{-3}

<table>
<thead>
<tr>
<th>Decoder,</th>
<th>Number of stored vectors,</th>
<th>Gain in tolerance to noise over QPSK signals, (dB)</th>
<th>Relative complexity of the decoder,</th>
<th>Average number of bit errors per burst,</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viterbi</td>
<td>2^8=64</td>
<td>2.05</td>
<td>256</td>
<td>9</td>
</tr>
<tr>
<td>System A</td>
<td>m=16</td>
<td>1.35</td>
<td>64</td>
<td>69</td>
</tr>
<tr>
<td>System A1</td>
<td>m=14.8</td>
<td>1.9</td>
<td>72</td>
<td>11</td>
</tr>
<tr>
<td>System A2</td>
<td>m=13.8</td>
<td>1.85</td>
<td>28</td>
<td>12</td>
</tr>
</tbody>
</table>

### Table 7.1.4: Comparisons of the four decoding schemes incorporating code CC(2), at BER=10^{-3}

<table>
<thead>
<tr>
<th>Decoder,</th>
<th>Number of stored vectors,</th>
<th>Gain in tolerance to noise over QPSK signals, (dB)</th>
<th>Relative complexity of the decoder,</th>
<th>Average number of bit errors per burst,</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viterbi</td>
<td>2^8=64</td>
<td>2.05</td>
<td>256</td>
<td>7</td>
</tr>
<tr>
<td>System A</td>
<td>m=16</td>
<td>1.2</td>
<td>64</td>
<td>41</td>
</tr>
<tr>
<td>System A1</td>
<td>m=12</td>
<td>1.8</td>
<td>58</td>
<td>9</td>
</tr>
<tr>
<td>System A2</td>
<td>m=12</td>
<td>1.9</td>
<td>28</td>
<td>10</td>
</tr>
</tbody>
</table>
Fig. 7.1.1: Performance comparison between the various decoding schemes with code UC[3]

Common attribute: rate-2/3 convolutionally coded 8PSK signal
Fig. 7.1.2: Performance comparison between the various decoding schemes with code UC[4]

Legend
- Δ Uncoded QPSK
- × Viterbi; $2^g = 32$
- □ system A; $m=8$
- ◊ system A; $m=16$
- ★ system A1
- × system A2

Common attribute: rate-2/3 convolutionally coded 8PSK signal
Fig. 7.1.3: Performance comparison between the various decoding schemes with code UC[5]

Legend

- △ Uncoded QPSK
- × Viterbi; 2 = 64
- ○ system A; m = 16
- ◊ system A1
- ■ system A2

Common attribute: rate-2/3 convolutionally coded 8PSK signal
Fig. 7.1.4: Performance comparison between the various decoding schemes with code CC[2]

Legend
- Uncoded QPSK
- Viterbi, $Z_0^q = 64$
- System A, $m=16$
- System A1
- System A2

Eb/No (dB)

Bit error rate (BER)

Common attribute: correlative coded BPSK signal
A class of well known convolutional codes with expanded signal sets, for MPSK modulation, is considered in this work. The particular applications concerned are those of rate-2/3 codes with an 8PSK signal (see Table 3.2.1) and those of rate-3/4 codes with a 16PSK signal (see Table 3.2.2). The performances of these codes with the Viterbi decoding have already been widely studied and tested by many research workers under different channel conditions (33-43). The main objective of this investigation is to develop a near-optimum decoding scheme (with a reduced decoder complexity) for such coded signals. Nevertheless, as a part of the research exercises, the performances of these codes with the Viterbi decoder over an additive white Gaussian noise channel, were tested by computer simulations. The results obtained are taken to be the standards with which the performances of other coded MPSK signals or decoding schemes are compared.

A class of new trellis codes, known here as the modulo-M correlative-level codes, has been proposed for MPSK signals in this work. The new codes are generally defined over a modulo-M integer number field, as opposed to the binary field GF(2) over which a conventional convolutional code is defined. The design work is presented in Chapter 4, and the more promising codes found, which operate on a quaternary data signal to generate an 8PSK signal, are shown in Table 4.3.1. Figures 7.2.1 and 7.2.2 compare the performances of some of these codes with those of the corresponding rate-2/3 convolutionally coded 8PSK signals.

The two codes under comparison in Figure 7.2.1 are the
correlative code CC(1) and the convolutional code UC(3), both having 16 states. Furthermore, they both operate on a quaternary data symbol to generate an 8PSK signal. Likewise, the three codes in Figure 7.2.2 are the correlative codes CC(2) and CC(3), and the convolutional code UC(5), all having 64 states. From these graphs it can be seen that, although the convolutional codes theoretically outperform the correlative codes by about 0.5 dB, at very high signal/noise ratios, both types of coded 8PSK signals provide very similar gains in tolerance to noise, over the range of bit error rates of $10^{-1}$ to $10^{-4}$.

Following the results of extensive computer simulation tests presented in Section 3.4 and Section 4.4, the performances of both types of coded signals are generally considered to be unacceptably poor if the system incorporates the decoding scheme of system A. The reason for this is that the characteristics of coded 8PSK signals generally cause prolonged error bursts in the coded data symbols. A part of this investigation was to locate those coded MPSK signals giving the best performance for system A. The approach considered was either to use the codes with improved distance profiles, or else to use the codes with significant larger sets of redundant data-symbol values (see Section 4.4 for details). Unfortunately, the results have been less promising in that such an approach does not generally provide any effective improvement in tolerance to noise for quaternary data systems, even with a 64PSK signal, where a much more sophisticated carrier-phase estimation process is required at the receiver.

Another class of codes, known as the rotationally
invariant convolutional codes, have also been developed for 8 and 16 PSK signals. These codes are particularly attractive for satellite applications in that they are transparent to half of all likely phase rotations in the received signal. When combined with a simple phase error correction system proposed at the receiver (see Section 6.5 for details), such a coded signal can tolerate any likely phase change in the reference carriers of the coherent demodulation process. These carrier-phase changes are commonly experienced in the transmission system where the given channel introduces sudden phase jumps and/or impulsive noise.

The new codes are not made fully transparent to all possible carrier phases; so that the undue degradation in tolerance to noise will not be experienced relative to the corresponding optimum but not rotationally invariant codes. On the other hand, these codes are made transparent to part of the M-fold phase ambiguity in the received MPSK signal, so that, following the detection of a phase change (slip) in the reference carriers, that leads to the incorrect operation of the decoder, the phase slip can be corrected with no further checks or phase changes being required to correct the given phase error.

Eight new rotationally invariant coded 8 and 16 PSK signals (see Tables 6.3.5 and 6.4.1) have been designed. Their performances with the Viterbi decoding are now compared to those of the corresponding optimum (but not rotationally invariant) coded 8 and 16 PSK signals, in Figures 7.2.3 to 7.2.10. It can be seen from these graphs that, except for two 4-state codes, the tolerances of the other new codes to noise
are within 0.3 dB of those of the optimum codes, over the range of bit error rates of $10^{-3}$ to $10^{-4}$.

In particular, code RC(2) (a rate-2/3, 8-state code) appears to be very attractive for 8PSK signals, in that it can provide the tolerances to noise which are close to those of the optimum code, over the range of bit error rates of $10^{-3}$ to $10^{-4}$. RC(2) also has the desired structure (see Figure 6.3.2) with which the corresponding Viterbi decoder operates on a binary signal only, leading to a considerable reduction in the decoder complexity.

Among the four rate-3/4 codes, codes RC(5) and RC(8) achieve the optimum asymptotic coding gains. Over the range of bit error rates of $10^{-3}$ to $10^{-4}$, a degradation of about 0.2 dB in tolerances to noise, compared to the optimum codes, is usually expected.
Fig. 7.2.1: Performance comparison of different types of coded 8PSK signals with the Viterbi decoding of 16 stored vectors.

<table>
<thead>
<tr>
<th>Legend</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>△</td>
<td>Uncoded QPSK</td>
</tr>
<tr>
<td>×</td>
<td>Correlative code CC[1]</td>
</tr>
<tr>
<td>□</td>
<td>Convolutional code UC[3]</td>
</tr>
</tbody>
</table>

Bit error rate (BER) vs. Eb/No (dB)
Fig. 7.2.2: Performance comparison of different types of coded 8PSK signals with the Viterbi decoding of 64 stored vectors.

Legend
- △ Uncoded QPSK
- × Correlative code CC[2]
- □ Correlative code CC[3]
- ⊙ Convolutional code UC[5]
Fig. 7.2.3: Performance comparison of different rate-2/3, 4-state, convolutional coded 8PSK signals, using Viterbi decoding.

Legend

- Uncoded DQPSK
- \( \times \) rotationally invariant code RC[t]
- \( \square \) non-rotationally invariant code UC[t]
Fig. 7.2.4: Performance comparison of different rate-2/3, 8-state, convolutional coded 8PSK signals, using Viterbi decoding.

Legend:
- △ Uncoded DQPSK
- × rotationally invariant code RC[2]
- □ non-rotationally invariant code UC[2]
Fig. 7.2.5: Performance comparison of different rate-2/3, 16-state, convolutional coded 8PSK signals, using Viterbi decoding.
Fig. 7.2.6: Performance comparison of different rate-2/3, 32-state, convolutional coded 8PSK signals, using Viterbi decoding

Legend

- Δ Uncoded DQPSK
- × rotationally invariant code RC[4]
- □ non-rotationally invariant code UC[4]
Fig. 7.2.7: Performance comparison of different rate-3/4, 4-state, convolutional coded 16PSK signals, using Viterbi decoding.

Legend

- Δ Uncoded D8PSK
- × rotationally invariant code RC[5]
- □ non-rotationally invariant code UC[6]
Fig. 7.2.8: Performance comparison of different rate-3/4, 8-state, convolutional coded 16PSK signals, using Viterbi decoding.

Legend

- △ Uncoded D8PSK
- × rotationally invariant code RC[6]
- □ non-rotationally invariant code UC[7]
Fig. 7.2.9: Performance comparison of different rate-$\frac{3}{4}$, 16-state, convolutional coded 16PSK signals, using Viterbi decoding.
Fig. 7.2.10: Performance comparison of different rate-3/4, 32-state, convolutional coded 16PSK signals, using Viterbi decoding.

Legend
- △ Uncoded D8PSK
- × rotationally invariant code RC[8]
- □ non-rotationally invariant code UC[9]
7.3 ORIGINALLITY OF THE STUDY

This investigation was originated to study and compare various types of coded MPSK signals and the associated decoding schemes. It was hoped that such an investigation may lead to the development of more promising signal designs and/or more cost-effective decoding schemes for new digital satellite communication systems.

Four different types of decoding schemes have been considered. Except the well known Viterbi algorithm and system A which was originally proposed by A.P. Clark, et al [89,93] for detecting digital signals over a linear channel with intersymbol interference, the remaining two noise-adaptive decoders, called systems A1 and A2, were proposed by the author. In particular, the use of a variable number of stored vectors in the decoder and the use of two thresholds for achieving this are the author's own ideas, although the similar approach has also been applied to the sequential and Viterbi decoding by other people [91,94-96]. The use of an adjustable expansion procedure for system A2 was suggested by the author's supervisor. This proposal was further developed by the author with the use of two different distance measures (see equations 5.2.5 and 5.2.6) for achieving the adjustable expansions of stored vectors. The use of two buffer stores for systems A1 and A2 was suggested by the author's supervisor, and this was further tested by the author to determine the approximate size of the buffer stores required. A British patent application has been filed on system A2 by the Plessey Company.

The various modulo-M correlative coded MPSK signals
presented in Chapter 6 were designed by the author. This design work was carried out as an extension of the author's supervisor's original proposal for coding a digital baseband signal [48]. However, the use of a combination of feedforward and feedback transversal filters to achieving a large minimum free unitary distance is the author's contribution. In addition, during the extensive computer aided searches of new correlative codes, the author discovered that the use of asymmetric signal constellations may sometimes lead to a notable improvement in minimum distance for the given code constraint length. Unfortunately, only few such codes were actually found.

The design of rate-2/3 and rate-3/4 rotationally invariant convolutionally coded 8 and 16 PSK signals of Chapter 6 is the author's own work, although the similar design has been carried out for QAM signals before [50,51]. The idea of using an appropriate measure at the receiver to detect the presence of any phase slip (called the phase error) in the received signal, that leads to an incorrect operation of the decoder, was originated by the author's supervisor. This proposal was further developed and tested by the author with the use of the short-term mean-square difference defined by equation 6.5.1.

Finally, all computer simulation tests on the various coded systems and decoders were carried out by the author, and all the performance results presented in this thesis are attributed to the author. All computer programs used in the tests were also written by the author.
7.4 RECOMMENDATIONS FOR FURTHER WORK

The new correlative coded MPSK signals and rotationally invariant coded 8 and 16 PSK signals designed in this work have so far been tested over an additive white Gaussian noise channel only. To investigate the possible applications of these signals to practical satellite links, the effects of non-ideal channel characteristics must be considered. This would lead to a measure of performances of the new coded PSK signals over a more representative satellite link, that includes the effects of adjacent and co-channel interference, the nonlinear amplifications of the HPA at the earth station and of the TWTA in the satellite. The effects of carrier frequency and timing errors, together with the effect of bandlimiting 8 and 16 PSK signals, also require some investigation.

Furthermore, in order to prove the worth of the new rotationally invariant coded 8 and 16 PSK signals under non-ideal conditions, more detailed study and computer simulation tests are clearly required for the phase error correction system proposed in Section 6.5. This also includes the possibility of developing a more effective phase error tracking scheme for the new codes.

The two novel noise-adaptive system A-type decoding schemes proposed for coded 8PSK signals appear to be very promising in that the near-optimum performance of the coded signal can be achieved with a potential reduction in the equipment complexity, compared to that of the corresponding Viterbi decoder. Again, more detailed study and optimization of the system parameters are required under the non-ideal
channel conditions. In particular, additional statistical information is needed to ascertain the buffer storage requirement of the decoders. Furthermore, a detailed study concerning the possible implementation of the new decoding schemes is required and the processing speed of each decoder needs further consideration.

Finally, the possible applications of the new noise-adaptive decoding techniques to other coded modulation schemes, and also to the detection of digital baseband signals over a linear channel in the presence of intersymbol interference, certainly appear to be equally attractive. But this requires further study and development.
CHAPTER EIGHT

CONCLUSIONS
CHAPTER 8

8 CONCLUSIONS

Among the four decoding schemes studied in this work, the Viterbi algorithm decoder has been proved to be most promising for the convolutional codes with expanded signal sets of short constraint lengths, such as those with 4 and 8 stored vectors only. However, with more than 16 stored vectors, the Viterbi algorithm scheme generally becomes much too complex for coded 8 and 16 PSK signals. In these cases, alternative decoding schemes would have to be used instead.

The conventional near-maximum likelihood decoding scheme of system A is not suitable for coded 8PSK signals in that it provides a small reduction in equipment complexity of the decoder at the expense of a relatively large degradation in tolerance to noise, compared to that of the corresponding Viterbi decoder. The signal characteristics of coded 8PSK signals constantly cause prolonged burst errors in the decoded data symbols and thus severely degrade the performance of the system.

The two novel noise-adaptive system A-type schemes, known as systems A1 and A2, are very promising decoding techniques for coded 8PSK signals. For a given degree of equipment complexity needed by the decoder, a carefully designed new system can provide a tolerance to noise that compares well with those of the other decoders. The results of extensive computer simulation tests suggest that the new schemes achieve a useful reduction both in the amount of storage required and in the number of operations per decoded data symbol, relative to those of the corresponding Viterbi
decoder, without any significant degradation in the performance of the system. In particular, system A2 is potentially more cost-effective than system A1.

Within the range of bit error rates of $10^{-3}$ to $10^{-4}$, the newly designed modulo-8 correlative coded 8PSK signals provide very similar gains in tolerance to additive white Gaussian noise as the corresponding best known rate-2/3 convolutional coded 8PSK signals. However, the asymptotic performances of the new codes are generally expected to be about 0.5 dB inferior to those of the convolutional codes of the same constraint lengths.

Eight new rotationally invariant convolutional codes have been designed. Four of these are designed for rate-2/3 coded 8PSK signals, and the other four are designed for rate-3/4 coded 16PSK signals. The new coded 8 and 16 PSK signals are transparent to half of all possible carrier-phase rotations in the received signal, and the remaining phase ambiguity is resolved by a phase-error correction system proposed for the receiver. With a code memory of 3 or more bits, which corresponds to 8 or more stored vectors in the Viterbi decoder, the asymptotic coding gains of the new codes are within about 0.3 dB of the coding gains of the optimum (but not rotationally invariant) codes. The new coded 8 and 16 PSK signals are particularly suitable for applications over satellite and land mobile radio links. The use of these codes is subject to a more detailed design and development of the phase error correction system.
REFERENCES
REFERENCES

1 CLARK, A.P.: "Principles of digital data transmission"
   2nd Ed., (Pretech Press, 1983)

2 CLARK, A.P.: "Advanced data transmission systems"
   (Pretech-Hall, 1977)

3 CLARK, A.P.: "Equalizers for digital modems" (Pentech Press, 1985)


5 CLARK, A.P.: "Modulation and modems", Dept. of Electronic 
   & Electrical Engineering, Loughborough University of 
   Technology, 1986

6 SPILKER, J.J.: "Digital communications by satellites"
   (Prentice-Hall, 1977)

7 BHARGAVA, V.K., HACCOUN, D., MATYAS, R., and NUSPL, P.P.: 
   "Digital communications by satellite: Modulation,
   multiple access and coding" (John Wiley & Sons, Inc., 
   1981)

8 FEHER, K.: "Digital communications: Satellite/earth
   station engineering" (Prentice-Hall, 1983)

9 JANSKY, D.M. and JERUCHIM, M.C.: "Communication satellites 
   in the geostationary orbit" (Artech House, Inc., 1983)

10 WU, W.W.: "Elements of Digital Satellite Communication"
   (Computer Science Press, Inc., 1984)

11 PRITCHARD, W.L.: "The history and future of commercial 
   satellite communications", IEEE Communication Magazine, 
   1984, 22, pp.22-37

12 RUDGE, A.W.: "Sky-hooks, fish-warners and hub-caps:
15 GALLAGER,R.G.: "Information theory and reliable communication" (John Wiley, 1968)
17 BLAHUT,R.G.: "The theory and practice of error control coding" (Addison-Wesley, 1983)
24 JOHANNESSON,R.: "Robustly optimal one-half binary con-
volutional codes", IEEE Trans., 1975, IT-21, pp.464-468


27 LEE, L-N: "Error-correction coding for commercial satellite channels", IEEE Int. Conf. on Commun., Denver, CO, USA, 14-18, June 1981, pp.2.5/1-4


34 ANDERSON, J.B., and TAYLOR, D.P.: "A bandwidth-efficient
class of signal space codes”, IEEE Trans., 1978, IT-24, pp.703-712


36 HUI, J., and FANG, R.J.F.: "Convolutional code and signal waveform design for band-limited satellite channels”, ICC Int. Conf. on Commun., Rec., Denver, CO., USA, June 1981, vol.3, pp.47.5/1-10


43 WILSON, S.G., SCHOTTLER, P.J., and SLEEPER, H.A.: "Rate 3/4


50 IBM EUROPE, "Trellis coded modulation scheme with 8-state systematic encoder and 90° symmetry for use in data modems transmitting 3-7 bits per modulation interval", CCITT Com XVII-no.D180, Oct. 1983


60 RYDBECK,N. and SUNDBERG,C-E.: "Recent results on spectrally efficient constant envelope digital modulation methods", ICC'79, Int. Conf. on Commun., Pt.III, Boston, MA, USA, 10-14, June 1979, pp.42.1/1-6

61 AULIN,T. and SUNDBERG,C-E.: "M-ary CPSFK type of signalling with input data symbol pulse shaping: Minimum distance and spectrum", ICC'79, Int. Conf. on Commun., Pt.II, Boston, MA, USA, 10-14, June 1979, pp.42.3/1-6

62 AULIN,T., RYDBECK,N. and SUNDBERG,C-E.: "Bandwidth efficient digital FM with coherent phase tree demodulation", ICC'79, Int. Conf. on Commun., Pt.III, Boston, MA, USA, 10-14, June 1979, pp.42.4/1-6


67 AULIN,T., PERSSON,B., RYDBECK,N., and SUNDBERG,C-E.: "Spectrally-efficient constant-amplitude digital
modulation schemes for communication satellite applications, vol.1 and 2", ESA contract reports, 1982


FOSCHINI, G.J.: "A reduced state variant of maximum likelihood sequence detection attaining optimum performance for high signal-to-noise ratio", IEEE


CHEN, N.-Y. and HACCOUN, D.: "Stack algorithm for high rate convolutional codes", ACTA Electronica Sinica, 1984, 12, pp.7-14


STIFFLER, J.J: "Theory of synchronous communications" (Prentice-Hall, 1971)


This section provides the theoretical background for the bandpass channel model described in Section 2.3 of Chapter 2. A real-valued bandpass signal is first described in terms of complex-valued equivalent lowpass waveforms, and this is followed by a presentation of the linear bandpass channel in terms of equivalent lowpass models. The band-limited additive white Gaussian noise is also discussed.

An arbitrary real-valued signal \( x(t) \), with its frequency content concentrated in a narrow band of frequencies in the vicinity of a frequency \( f_c \), may be generally expressed in the form

\[
x(t) = m(t)\cos(2\pi f_c t + \theta(t))
\] (A1.1)

where \( m(t) \) and \( \theta(t) \) are both real-valued functions that represent, respectively, the envelope and phase angle of \( x(t) \) [8]. The frequency \( f_c \) is usually called the carrier of \( x(t) \), and it is quite common that the amplitude spectrum of \( x(t) \) is centred at the frequency \( f_c \). When the bandwidth of \( x(t) \) is much smaller than \( f_c \), \( x(t) \) is known as a bandpass signal. Following equation A1.1,

\[
x(t) = m(t)\cos\theta(t)\cos 2\pi f_c t - m(t)\sin\theta(t)\sin 2\pi f_c t
\]

\[
= x_c(t)\cos 2\pi f_c t - x_B(t)\sin 2\pi f_c t
\] (A1.2)

where, by definition,

\[
x_c(t) = m(t)\cos\theta(t)
\] (A1.3)
\[ x_g(t) = m(t) \sin \theta(t) \]  

(A1.4)

The signals \( x_c(t) \) and \( x_g(t) \) are both real-valued and are known as the "in-phase" and "quadrature" components of \( x(t) \). Now, let

\[ u(t) = x_c(t) + jx_g(t) \]  

(A1.5)

then

\[ x(t) = \text{Re}[u(t)e^{2\pi ft}] \]  

(A1.6)

where \( \text{Re}[] \) represents the real part of the corresponding complex-valued quantity.

From equation A1.6, the Fourier transform (F.T.) of \( x(t) \) is

\[
\begin{align*}
X(f) &= \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft} \, dt \\
&= \int_{-\infty}^{\infty} \text{Re}[u(t)e^{j2\pi ft}]e^{-j2\pi ft} \, dt \\
&= \int_{-\infty}^{\infty} (x(t) + jx_g(t))e^{-j2\pi ft} \, dt \\
&= (1/2) \int_{-\infty}^{\infty} \left[ X(f) + X(-f) \right] \, df \\
&= (1/2) \left[ U(f-f_c) + U^*(-f-f_c) \right] \\
\end{align*}
\]

(A1.7)

(A1.8)

(A1.9)

where \( U(f) \) is the F.T. of \( u(t) \). Bearing in mind that \( x(t) \) is a
narrow-band signal with its frequency content concentrated in the vicinity of \( f = f_c \). Equation A1.9 shows the fact that the frequency content of \( u(t) \) must be concentrated at low frequencies, in the vicinity of \( f = 0 \). Therefore \( u(t) \) hence \( x_c(t) \) and \( x_s(t) \) must be lowpass signals. Consequently, \( u(t) \) is known as the equivalent lowpass (baseband) signal, and it may generally be regarded as a complex-valued waveform. The signal energy in \( x(t) \) is given by

\[
E = \int_{-\infty}^{+\infty} x^2(t) \, dt
\]

\[
= \int_{-\infty}^{+\infty} \left( \text{Re}[u(t)\exp(j2\pi f_c t)] \right)^2 \, dt
\]

\[
= (1/4) \int_{-\infty}^{+\infty} [u(t)\exp(j2\pi f_c t) + u^*(t)\exp(-j2\pi f_c t)]^2 \, dt
\]

\[
= (1/2) \int_{-\infty}^{+\infty} |u(t)|^2 \, dt
\]

\[
+ (1/2) \int_{-\infty}^{+\infty} |u(t)|^2 \cos[4\pi f_c t + 2\theta(t)] \, dt
\]

\[(A1.10)\]

Since the envelope of \( u(t) \) varies very slowly relative to the rapid variation of the cosine function, the net area contributed by the second integral in equation A1.10 is very small relative to that given by the first integral and is therefore neglected. Consequently,

\[
E = (1/2) \int_{-\infty}^{+\infty} |u(t)|^2 \, dt
\]

\[(A1.11)\]

In conclusion, a bandpass signal \( x(t) \) may, for mathematical
convenience, be generally described in terms of its equivalent low-pass signal $u(t)$, the energy carried by the real-valued signal $x(t)$ being a half of that of the complex-valued waveform $u(t)$.

Notice that, in the modulation process described in Section 2.3, the real and imaginary parts of the complex-valued baseband signal are multiplied by $\sqrt{2}\cos(2\pi f_c t)$ and $\sqrt{2}\sin(2\pi f_c t)$, respectively. In this way, the resultant real-valued bandpass signal fed to the transmission path has the same average signal energy per data symbol as that of the baseband signal.

A linear bandpass channel (or filter) may be described either by its impulse response $c(t)$ or else by its frequency response $C(f)$, which is the F.T. of $c(t)$. For a practical transmission path, $c(t)$ must be real-valued, so that

$$C^*(f) = C(f) \quad \text{(A1.12)}$$

Now, let

$$H(f-f_c) = C(f) \quad f\geq 0$$

$$= 0 \quad f<0 \quad \text{(A1.13)}$$

then

$$H^*(-f-f_c) = 0 \quad f\geq 0$$

$$= C^*(f) \quad f<0 \quad \text{(A1.14)}$$

Following equations A1.13 and A1.14,

$$C(f) = H(f-f_c) + H^*(-f-f_c) \quad \text{(A1.15)}$$
and the inverse F.T. of $C(f)$ is

$$ c(t) = h(t) \exp(j2\pi f_c t) + h^*(t) \exp(-j2\pi f_c t) $$

$$ = \sqrt{2} \text{Re}[h(t) \exp(j2\pi f_c t)] \quad (A1.16) $$

$h(t)$ is the inverse F.T. of $H(f)$. In general, the linear band-pass channel $c(t)$ may be described by its equivalent lowpass system $h(t)$, and $h(t)$ is basically complex-valued.

Suppose now that a real-valued bandpass signal $x(t)$, whose equivalent lowpass signal is given by $u(t)$, is fed to the bandpass channel (filter), which is characterised by its impulse response $c(t)$, or the equivalent lowpass response $h(t)$. Also suppose that the resultant signal at the bandpass channel output is given by $y(t)$, where $y(t)$ may be expressed in the form

$$ y(t) = \text{Re}[r(t) \exp(j2\pi f_c t)] \quad (A1.17) $$

It can be shown that the operation of bandpass filtering

$$ y(t) = x(t) \ast c(t) \quad (A1.18) $$

can be generally described by the equivalent lowpass filtering

$$ r(t) = u(t) \ast h(t) \quad (A1.19) $$

The proof is now given. Let the F.T. of $x(t)$, $c(t)$, $y(t)$, $u(t)$, $h(t)$ and $r(t)$ be $X(f)$, $C(f)$, $Y(f)$, $U(f)$, $H(f)$ and $R(f)$, respectively. Consider
\[ Y(f) = X(f)C(f) \]  
(A1.20)

and following equations A1.9 and A1.15,

\[
Y(f) = \frac{1}{2} \left[ U(f-f_c)+U^*(-f-f_c) \right] \left[ H(f-f_c)+H^*(-f-f_c) \right]
\]

(A1.21)

Since \( U(f-f_c) \) and \( H(f-f_c) \) are both zero when \( f<0 \), but \( U(-f-f_c) \) and \( H(-f-f_c) \) are both zero when \( f>0 \), equation A1.21 may be reduced to

\[
Y(f) = \frac{1}{2} \left[ U(f-f_c)H(f-f_c)+U^*(-f-f_c)H^*(-f-f_c) \right]
\]

(A1.22)

But, using equation A1.9,

\[
Y(f) = \frac{1}{2} \left[ R(f-f_c)+R^*(-f-f_c) \right]
\]

(A1.23)

It now follows that,

\[
R(f) = U(f)H(f)
\]

(A1.24)

and therefore,

\[
r(t) = u(t)*h(t)
\]

(A1.25)

This simple relationship allows the study of any data transmission over a bandpass channel to be carried out in terms of equivalent lowpass models. Consequently, if the coherent demodulator at the receiver achieves the exact correction of any frequency offset in the received signal (such as can be caused by Doppler shifts), the linear frequency translation encountered in the modulation of a signal, for purposes of
matching its spectrum to the available frequency band of the
given transmission path, can be neglected. The linear baseband
channel model described in Section 2.1 has therefore been used
throughout this work.

It is mathematically convenient to describe a band-
limited white Gaussian process in terms of white Gaussian noise
(WGN) with zero mean and a flat two-sided power spectrum
density \((1/2)N_0\) over the entire frequencies, the WGN being fed
to an "ideal" bandpass filter [19]. The filter is assumed to
have a pass band of \(B\) Hz that includes the spectrum of the
transmitted signal but much wider. The filter introduces
negligible distortion to the transmitted signal but will
eliminate the noise component outside the pass band. The
resultant noise signal \(n(t)\) is a narrow band signal and may
therefore be described by its equivalent lowpass signal \(z(t)\) as
follows.

\[
n(t) = n_c(t) \cos 2\pi f_c t - n_s(t) \sin 2\pi f_c t
\]

\[
= \text{Re}[z(t) \exp(j2\pi f_c t)] \tag{A1.26}
\]

where \(z(t) = n_c(t) + jn_s(t)\) \tag{A1.27}

Following equation A1.11, the total energy in \(n(t)\) is half of
that in \(z(t)\). Thus \(z(t)\) has a constant power spectral density
of a total of \(N_0\) over the frequency band from approximately
\(-(1/2)B\) Hz to \((1/2)B\) Hz. In addition, each of the in-phase
and quadrature components \(n_c(t)\) and \(n_s(t)\) must have a flat two-
sided power spectral density of \((1/2)N_0\) from \(-(1/2)B\) Hz to
\((1/2)B\) Hz.
Now return to the bandpass model for MPSK signals in Figure 2.3.1. As already shown in Section 2.3, the receiver bandpass filter whose transform function is $Q(f)$ and the receiver lowpass filter whose transform function is $B(f)$ may be described together by the equivalent lowpass transform function $E(f)$. The resultant filtering on $n(t)$ gives in the corresponding complex-valued lowpass waveform $w(t)$, at the output of the demodulator. Clearly, $w(t)$ is given by

$$w(t) = z(t)*e(t)$$  \hspace{1cm} (A1.28)

$e(t)$ is the inverse T.F. of $E(f)$. Bearing in mind that the bandwidth of $E(f)$ is approximately equal to that of the transmitted signal and is therefore much smaller than $B$, each of the real and imaginary parts of $w(t)$ has a two-sided power spectral density function of [19]

$$(1/2)\sigma_0 |E(f)|^2$$  \hspace{1cm} (A1.29)

The result here follows that given by equation 2.3.25.
THEOREM OF MAXIMUM LIKELIHOOD DECODING IN THE PRESENCE OF ADDITIVE WHITE GAUSSIAN NOISE

The following analysis leads to the principle of optimum decoding/detection of a sampled baseband coded signal in the presence of additive white Gaussian noise. From the general system model shown in Figure 2.1, the received baseband signal $r(t)$, at the output of the coherent demodulator, is a complex-valued continuous waveform

$$ r(t) = \sum q_i h(t-iT) + w(t) \quad (A2.1) $$

where $h(t)$ is the impulse response of the given baseband channel, and $w(t)$ is the complex-valued baseband noise waveform at the output of the coherent demodulator. $\{q_i\}$ are the complex-valued code symbols which are uniquely determined by the sequence of $2^L$-level data symbols $\{s_i\}$ fed to the coder at the transmitter. In general, $\{s_i\}$ are statistically independent and equally likely to have any of $2^L$ possible values. The transmitted symbols $\{q_i\}$, on the other hand, are not generally statistically independent of each other. $q_i$ can here take on any of its $M$ possible values, where $M > 2^L$.

Over the entire transmission period $0 \leq t < NT$, the given information sequence, or message, to be transmitted over the baseband channel, may be generally represented by a $N$-component vector of data symbols

$$ I = [ s_0 \ s_1 \ ... \ s_{N-1} ] \quad (A2.2) $$

 Clearly, the message $I$ can have any of the $2^{LN}$ different
possible values \( \{I_j\} \) for \( j = 1, 2, \ldots, 2^{LN} \). For a given coding/mapping process incorporated by the data transmission system, the message \( I \) uniquely determines the \( N \)-component vector of complex-valued code symbols

\[
Q = [q_0 \ q_1 \ \ldots \ q_{N-1}] \quad (A2.3)
\]

using the coding/mapping function

\[
Q = F(I) \quad (A2.4)
\]

Clearly, the \( Q \) may have any of the corresponding \( 2^{LN} \) different possible values, \( \{Q_j\} \), for \( j = 1, 2, \ldots, 2^{LN} \), as defined by equation A2.4. The code symbols \( \{q_j\} \) are transmitted over the baseband channel in sequence, at time instants \( \{t = iT\} \), where \( T \) is the symbol interval in seconds.

It is assumed that the received waveform \( r(t) \) is sampled once per symbol interval, at time instants \( \{t = iT\} \), and is at close to the Nyquist rate, so that all the information in \( r(t) \) over the entire transmission period \( 0 < t < NT \), is contained in the vector of the resultant \( N \) sample values \( \{r(iT)\} \),

\[
R = [r_0 \ r_1 \ \ldots \ r_{N-1}] \quad (A2.5)
\]

where \( r_i = r(iT) \), for \( i = 0, 1, \ldots, N-1 \). For the channel transfer-function shown in Figure 2.1.2, the vector \( R \) is given by

\[
R = Q + W \quad (A2.6)
\]
\[ r_i = q_i + w_i \] 

for \( i = 0, 1, \ldots, N-1 \). \( W \) is the corresponding vector of the resultant \( N \) noise samples \( \{w_{iT}\} \).

\[ W = [w_0 \ w_1 \ \ldots \ w_{N-1}] \]  

(A2.8)

Throughout the following discussion, it is taken that the real and imaginary parts of the components \( \{w_i\} \) of the vector \( W \) are sample values of statistically independent Gaussian random variables with zero mean and fixed variance \( \sigma^2 \). In addition, \( \{w_i\} \) are statistically independent of the \( \{q_i\} \), so that the continuously variable random vector \( W \) is statistically independent of the code vector \( Q \).

The decoder at the receiver is assumed to have the prior knowledge of the \( 2^L \) different possible values of \( s_i \) (and so the \( 2^{LN} \) possible values of \( I \)), and it also has the prior knowledge of the coding/mapping function \( F(.) \) (and so the corresponding \( 2^{LN} \) possible values of \( Q \)). From any given received vector \( R \), the decoder must select one of the \( 2^{LN} \) vectors \( \{Q_j\} \) and therefore one of the \( 2^{LN} \) vectors \( \{I_j\} \), as the decoded message. Clearly, the decoder can make \( 2^{LN} \) hypotheses, \( H_j \), that \( Q=Q_j \) and therefore \( I=I_j \), for \( j = 1, 2, \ldots, 2^{LN} \). It requires a decision rule to decide which values of \( R \) lead to the acceptance of \( H_j \), for \( j = 1, 2, \ldots, 2^{LN} \).

The vectors \( R \), \( Q \) and \( W \) may be represented as points in an \( N \)-dimensional Euclidean vector space [2]. The orthogonal projections of any point in this vector space onto the \( N \) orthogonal axes give the \( N \) components of the corresponding
vector. The decoder uses the position of the received vector R in this vector space to determine the value of Q and thus I that has been received. It divides the vector space into $2^{LN}$ regions, \{D_j\}, for $j=1, 2, \ldots, 2^{LN}$. The $2^{LN}$ regions of the vector space are known as the decision regions, and the boundaries separating these regions are known as the decision boundaries.

If the vector R lies in some region $D_j$ of the vector space, it is written $R \in D_j$. When this happens, the decoder accepts the hypothesis $H_j$ and so decides the received message as $I_j$. It is, of course, assumed that R must lie in either one of the $2^{LN}$ regions, and it can not lie in more than one region, nor can it lie in neither of these. In other words, the $2^{LN}$ decision regions are disjoint and together they fill the whole of the N-dimensional vector space [2].

The optimum decision boundary for the decoder to minimise the probability of decoding error is now derived. In order to minimise the probability of error in a decoding process it is necessary to maximise the probability of a correct decision, $P(C)$, where

$$P(C) = \int \ldots \int P(C/R)f(r_0, r_1, \ldots, r_{N-1}) \, dr_0 \, dr_1 \ldots \, dr_{N-1}$$

$$= \int P(C/R)f(R) \, dR \quad \text{(A2.9)}$$

$P(C/R)$ is the conditional probability of a correct decision given the received vector R, and

$$f(R) = f(r_0, r_1, \ldots, r_{N-1}) \quad \text{(A2.10)}$$
is the value of the joint probability density function of the N random variables, corresponding to the N components of R, at a given value of R. Notice that R is here taken as a continuously random variable and may have any of its possible values.

Since f(R) is nonnegative, P(C) is maximised by maximising P(C/R) for every possible value of R. For any given received vector R, P(C/R) is maximised by selecting Ik as the decoded value of I, the vector Qk for which

$$\Pr(Q_k/R) > \Pr(Q_j/R), \quad \text{for } j=1, 2, \ldots , 2^{LN}, j \neq k$$  \hspace{1cm} (A2.11)

Pr(Q_j/R) is the conditional probability that Q=Q_j, given R, so that Pr(Q/R) is the a-posteriori probability of Q, and, of course, from equation A2.4,

$$Q_j = F(I_j) \quad \text{for } j=1, 2, \ldots , 2^{LN}$$  \hspace{1cm} (A2.12)

Therefore, for the given received vector R, the decoder that minimises the probability of error, must select from the L possible values of Q, the vector Q which maximises P(Q/R) and thus which has the maximum a-posteriori probability of being correct [2]. If the maximum a-posteriori probability is shared between two or more of the $2^{LN}$ vectors (Q_j), the decoder may select any of these vectors as the decoded vector of code symbols.

Now, following the Bayes' Theorem [2],
\[
\Pr(Q/R) = \frac{p(R/Q) \Pr(Q)}{f(R)}
\]  
(A2.13)

where \( \Pr(Q) \) is a-posteriori probability of \( Q \),

\[
f(R) = f(r_0, r_1, \ldots, r_{N-1})
\]  
(A2.14)

is the value of the joint probability density function of the random variables corresponding to \( r_0, r_1, \ldots, r_{N-1} \), at the given values of \( r_0, r_1, \ldots, r_{N-1} \).

and

\[
p(R/Q) = p(r_0, r_1, \ldots, r_{N-1}/q_0, q_1, \ldots, q_{N-1})
\]  
(A2.15)

is the value of the conditional probability density function of the random variables corresponding to \( r_0, r_1, \ldots, r_{N-1} \), at the given values of \( r_0, r_1, \ldots, r_{N-1} \), and the given values of \( q_0, q_1, \ldots, q_{N-1} \).

Following equations A2.11 and A2.13, the decoding process that minimises the probability of error, must select from the \( 2^{LN} \) vectors \( \{Q_j\} \), the vector \( Q_k \) for which

\[
\Pr(R/Q_k)\Pr(Q_k) > p(R/Q_j)\Pr(Q_j)
\]  
(A2.16)

for \( j=1, 2, \ldots, 2^{LN} \), and \( j \neq k \).

Notice that the decoding process defined by equation A2.16 minimises the probability of error for any joint probability density function of the \( N \) noise samples \( \{w_i\} \) and not just for the joint Gaussian probability density assumed here [2].

In the important case where all different possible
messages \( (I_j) \) are equally likely to be received, as is assumed here, the probability that \( Q_j \) occurs, where \( Q_j = F(I_j) \), is the same for all possible values of \( Q \). Thus, \( \Pr(Q=Q_j) \) is independent of \( Q_j \), and

\[
\Pr(Q=Q_j) = 1/2^{LN} \quad \text{for } j=1, 2, \ldots, 2^{LN} \quad \text{(A2.17)}
\]

Under these conditions, the decoder that minimises the probability of error, must select from the \( 2^{LN} \) vectors \( \{Q_j\} \), the vector \( Q_k \) for which

\[
P(R/Q_k) > P(R/Q_j) \quad \text{for } j=1, 2, \ldots, 2^{LN}, j \neq k \quad \text{(A2.18)}
\]

In conclusion, the decoder which satisfies equation A2.18 maximises the likelihood function \( p(R/Q) \), for the given value of \( R \), and is thus known as the maximum likelihood decoder.

Since the real and imaginary parts of the noise components \( \{w_i\} \) in the received vector \( R \) are sample values of statistically independent Gaussian random variables with zero mean and fixed variance \( \sigma^2 \), it follows from equation A2.7 that \( r_i \) is a sample value of a complex-valued Gaussian random variable with a mean value \( q_i \), and the real and imaginary parts of \( r_i \) both have a fixed variance of \( \sigma^2 \). In addition, the different \( \{r_i\} \) are sample values of statistically independent Gaussian random variables. Under these conditions, the conditional probability density function of \( R \), given \( Q \), becomes

\[
p(R/Q) = p(r_0, r_1, \ldots, r_{N-1}; q_0, q_1, \ldots, q_{N-1})
\]
\[
\begin{align*}
N-1 & \prod_{i=0}^{ \frac{1}{\sigma^2} \exp\left( - \frac{|r_i-q_i|^2}{2\sigma^2} \right) \\
& = \frac{1}{(2\pi\sigma^2)^N} \exp\left( - \frac{1}{2\sigma^2} \sum_{i=0}^{N-1} |r_i-q_i|^2 \right) \\
& = \frac{1}{(2\pi\sigma^2)^N} \exp\left( - \frac{1}{2\sigma^2} |R-Q|^2 \right) \quad (A2.19)
\end{align*}
\]

where \(|R-Q|\) represents the Euclidean distance between the two vectors \(R\) and \(Q\), in the \(N\)-dimensional Euclidean vector space.

Clearly, following equation A2.17, when \(p(R/Q)\) is maximum, \(R-Q\) must be minimum. Thus the decoded message must satisfy

\[
|R-Q_k| < |R-Q_j|, \quad \text{for } j=1, 2, \ldots, 2^{LN}, j \neq k \quad (A2.20)
\]

Therefore, under the assumed noise condition, the decoding process that minimises the probability of error, must select from the \(2^{LN}\) possible messages \(\{I_j\}\), the message \(I_k\) for which the corresponding code vector \(Q_k\) is closed to \(R\) in the \(N\)-dimensional vector space. Such a decoder is also known as the minimum distance decoder. From equation A2.19,

\[
|R-Q|^2 = \sum_{i=0}^{N-1} |r_i-q_i|^2 
\]

\(|R-Q|\) is also known as the unitary distance between the received sequence (vector) \(R\) and the corresponding code sequence (vector) \(Q\) of the message \(I\).

For the four decoding schemes investigated in this work, the unitary distance defined by equation A2.21 is used
throughout as a measure of goodness of any stored vector of data symbols. The larger is the unitary distance squared, known as the cost of the stored vector, the less likely it is that the given vector of data symbols is correct. The four decoders employ different decision rules to eliminate the least likely vectors of data symbols.

Appendix A3 presents the asymptotic performances of various coded MFSK signals studied in this work, using the maximum likelihood decoder defined above.
It is well known that the minimum distance property of trellis codes plays a very important role in determining the error-correction capability of the codes [7,16-18]. The two important minimum distance parameters, which are the minimum free unitary distance and the distance profile of a coded signal, have already been defined in Section 3.2 and Section 4.4, respectively. In this section, the bit error probability of the coded signal using the soft-decision maximum likelihood decoder is derived in the presence of additive white Gaussian noise. The deviation begins with a consideration of the first-error-event probability, as follows.

Supposing the transmission starts at time t=0, the first-error-event occurs at an arbitrary trellis level i, if the correct sequence (vector) of data symbols is eliminated from the decoder, for the first time at the end of i-th decoding process. For this event to occur, an incorrect vector of code symbols must be closer, in the unitary distance sense, to the received sequence (vector) of sample values than the correct code vector (see Appendix A2 for details).

Now, let the i-component vector of complex-valued code symbols at the transmitter be

\[ Q = [ q_0 \ q_1 \ ... \ q_{i-1} ] \]  (A3.1)

When Q is fed to the linear baseband channel with additive white Gaussian noise, it gives rise to the i-component vector
of received samples

\[ R = \{ r_0, r_1, \ldots, r_{i-1} \} \]  

(A3.2)

The decoder selects the incorrect vector

\[ Q' = \{ q'_0, q'_1, \ldots, q'_{i-1} \} \]  

(A3.3)

instead of the vector Q in equation A3.1, only if

\[ |R - Q'| - |R - Q| > 0 \]  

(A3.4)

In other words, the received vector \( R \) is closer to \( Q' \) than to the correct vector \( Q \) in the \( i \)-dimensional Euclidean vector space.

Now, consider the \( i \)-dimensional vector space in which vectors \( Q' \) and \( Q \) may be represented as two points. Following the analysis given in Appendix A2, the best decision boundary here must be a \((i-1)\)-dimensional hyperplane which perpendicularly bisects the line joining the two vectors \( Q \) and \( Q' \) \( [2] \). Both \( Q \) and \( Q' \) are at a distance \( d/2 \) from this hyperplane, where

\[
d^2 = |Q - Q'|^2
\]

\[
= \sum_{h=0}^{i-1} |q_h - q'_h|^2
\]  

(A3.5)

Clearly, \( d \) represents here the unitary distance between the two code sequences (vectors) \( Q \) and \( Q' \). An incorrect decision is made only when \( R - Q > d/2 \) and \( R \) lies on the wrong side of the
hyperplane. Following equation A2.6,

\[ R = Q + W \]  \hspace{1cm} (A3.6)

where \( W \) is the noise vector

\[ W = [ w_0 \ w_1 \ \ldots \ w_{i-1} ] \]  \hspace{1cm} (A3.7)

The noise vector \( W \) may be represented by \( i \) orthogonal components, with one of them chosen to be in parallel to the line joining \( Q \) and \( Q' \), so that remaining \((i-1)\) components are orthogonal to this line and therefore have no effect in shifting \( R \) toward or from the hyperplane [2]. Since the value of the orthogonal projection of the noise vector \( W \) onto any direction in the \( i \)-dimensional vector space is a sample value of a Gaussian random variable with zero mean and fixed variance \( \sigma^2 \), the value of the noise component which is parallel to the line joining \( Q \) and \( Q' \) and hence is responsible for shifting \( R \) toward or from the hyperplane must be the sample value of a Gaussian random variable with zero mean and variance \( \sigma^2 \) [2]. Thus, the probability that the first-error-event occurs at \( i^{th} \) step is given by

\[ P(E) = Q(d/2\sigma) \]  \hspace{1cm} (A3.9)

where

\[ Q(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp\left(-\frac{1}{2} y^2\right) dy \]  \hspace{1cm} (A3.10)

Now, let the smallest, the second, the third, and so on, smallest values of \( d \), for the given code, be \( x_1, x_2, x_3, \) and so
on, respectively. It follows that the bit error probability associated with any error event at any step may be given by

\[ P_b = N_1 Q(x_1/2\sigma) + N_2 Q(x_2/2\sigma) + N_3 Q(x_3/2\sigma) + \ldots < N_1 Q(x_1/2\sigma) \]  

(A3.11)

where \( N_1, N_2, N_3, \) and so on, represent the average numbers of bit errors associated with an incorrect decision at each of the corresponding distances. At low bit error rates, it is clear that by far the most of incorrect decisions must occur at the nearest decision boundary. Thus the asymptotic bit error rate for a given coded signal, when the maximum likelihood decoder is incorporated, can be approximated to

\[ P_b = N_1 Q(x_1/2\sigma) \]  

(A3.12)

But \( x_1 \) is the minimum free unitary distance of the given coded signal, \( d_{\text{free}} \). Hence, equation A3.12 becomes

\[ P_b = N_1 Q(d_{\text{free}}/2\sigma) \]  

(A3.13)

The asymptotic improvement in tolerance to noise achieved this way, over the corresponding uncoded system, is determined by

\[ G_c = 10 \log_{10} \left( \frac{d_{\text{free}}^2}{d_{\text{uncoded}}^2} \right) \]  

(dB)  

(A3.14)

where \( d_{\text{uncoded}} \) represents the minimum free unitary distance for the corresponding uncoded system.

For a conventional convolutional code over the binary
field GF(2), where the Hamming distance measure is employed, this asymptotic coding gain may be given by

\[ G_c = 10 \log_{10}(R_c \cdot d_H) \]  

(A3.15)

\( R_c \) is the code rate and \( d_H \) is the minimum free Hamming distance of the code [7,18]. To complete this section, equation A3.15 is derived as follows.

Consider a rate-\( R_c \) binary convolutional code with the minimum free Hamming distance of \( d_H \), and the code is incorporated by a system employing a binary PSK signal. Now, supposing the two possible transmitted symbol values are \( +\sqrt{E_s} \) and \( -\sqrt{E_s} \), where \( E_s \) is the average signal energy per transmitted symbol, the minimum free unitary distance squared between any two different coded signal sequences must be

\[ d_{\text{free}}^2 = (\sqrt{E_s} - (-\sqrt{E_s}))^2 \cdot d_H \]

\[ = 4E_s \cdot d_H \]

\[ = 4R_cE_b \cdot d_H \]  

(A3.16)

bearing in mind that the average signal energy per data bit, \( E_b \), is here given by \( E_s/R_c \). On the other hand, the Euclidean distance squared between the two uncoded binary PSK signals that correspond to the carrier phases of 0 and \( \pi \) radians is

\[ d_{\text{uncoded}}^2 = (\sqrt{E_b} - (-\sqrt{E_b}))^2 \]

\[ = 4E_b \]  

(A3.17)

Note that, in the uncoded BPSK system, \( E_b = E_s \). Therefore, the
asymptotic coding gain achieved by using the given code is

\[ G_c (\text{dB}) = 10 \log_{10} \left( \frac{4R_c E_b d_H}{4E_b} \right) \]

\[ = 10 \log_{10} (R_c \cdot d_H) \]  \hspace{1cm} (A3.18)

Clearly, for any code with a code rate of \( R_c \), the amount of additional bit signal/noise ratio in dB required, due to the \( R_c \), is considered as a loss of \( 10 \log_{10} (1/R_c) \) dB. Therefore, for such error-control coding schemes to be effective, the gain obtained in tolerance to noise due to the enlarged Hamming distance \( d_H \) must exceed the loss in energy per information bit, for a given error rate. The overall advantage achieved, at very low error rates, is the true asymptotic coding gain of the given code.
A4 A MATHEMATICAL TREATMENT OF THE CHANNEL CAPACITY

The Shannon's coding theorem [14] demonstrates the existence of codes that achieve a reliable communication if and only if the information transmission rate does not exceed the maximum rate $C$ (bits per second), called the channel capacity. This remarkable result indicates the fact that the ultimate limit of performance set by the noise on the channel is not the accuracy, as generally believed before the Shannon's work, but by the rate at which the information is reliably transmitted. The concept of the channel capacity is fundamental to the communication theory, for it can be generally applied to a large class of channel models. This section presents a mathematical treatment of the Shannon's channel capacity concept for an idealized satellite channel, modelled as a memoryless, additive white Gaussian noise channel with a set of discrete input signals and a continuous-variable output signal. Both the channel capacity expression in equation 3.2.10 and the Shannon's limit curve shown in Figure 3.2.3 are derived. Detailed treatments of the channel capacity can be found elsewhere [15].

A few important mathematical definitions used to define the channel capacity is now introduced. Suppose that the event $X$ has a total of $m$ possible outcomes, represented by $x_0$, $x_1$, $\ldots$, $x_{m-1}$, and the event $Y$ has $n$ possible outcomes, that are $y_0$, $y_1$, $\ldots$, $y_{n-1}$. Also let

$$P(x_i) = \text{probability that } x_i \text{ occurs},$$

(A4.1)
\[ P(Y_j) = \text{probability that } Y_j \text{ occurs.} \tag{A4.2} \]
\[ P(X_i, Y_j) = \text{joint probability of } X_i \text{ and } Y_j. \tag{A4.3} \]
\[ P(Y_j/X_i) = \text{conditional probability of } Y_j \text{ given } X_i. \tag{A4.4} \]
\[ P(X_i/Y_j) = \text{conditional probability of } X_i \text{ given } Y_j. \tag{A4.5} \]

for \( i=0, 1, \ldots, (m-1) \) and \( j=0, 1, \ldots, (n-1) \). The entropy (or uncertainty) of the event \( X \) is generally defined by

\[ H(X) = - \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} P(X_i, Y_j) \log_2 P(X_i) \tag{A4.6} \]

and the entropy (or uncertainty) of the event \( X \), given \( Y \), is generally defined by

\[ H_Y(X) = - \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} P(X_i, Y_j) \log_2 P(X_i/Y_j) \tag{A4.7} \]

[15]. Similarly, the entropy of the event \( Y \) is defined by

\[ H(Y) = - \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} P(X_i, Y_j) \log_2 P(Y_j) \tag{A4.8} \]

and the entropy of \( Y \), given \( X \), is defined by

\[ H_X(Y) = - \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} P(X_i, Y_j) \log_2 P(Y_j/X_i) \tag{A4.9} \]

Now, let \( X \) and \( Y \) represent the input and output signals of a discrete memoryless channel, respectively, the well-known expression for the information transmission rate of the given channel, \( R_T \), is [15]
This is to say that, $R_T$ is determined by the uncertainty of $X$ (which is the amount of information transmitted over the given channel) less the uncertainty of $X$, given $Y$ (that is the uncertainty of what was actually sent). It follows from equations A4.1 to A4.9 that,

\[
R_T = H(X) - H_Y(X)
= H(Y) - H_X(Y)
\tag{A4.10}
\]

The channel capacity, $C^*$ (bits per symbol interval), is generally defined as the maximum amount of information that can be transmitted over the given channel without error, and the maximisation is carried over all possible input probability assignments. That is,
\[ C^* = \max \{ H(X) - H_Y(X) \} \]

\[
= \max_{m-l} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} \frac{P(x_i, y_j / x_1) \log_2 \left( \frac{P(x_i, y_j)}{P(x_1)P(y_j)} \right)}{P(x_0) .. P(x_{m-1})} \]

where \( P(x_i) \), for \( i = 0, 1, ..., (m-1) \), is the a priori probability of \( x_i \).

The alternative way to define \( C^* \) involves the use of the expected mutual information between \( X \) and \( Y \), \( I(X;Y) \) [15], which is defined as follows. Let the mutual information between \( x_i \) and \( y_j \), for \( i = 0, 1, ..., (m-1) \), and \( j = 0, 1, ..., (n-1) \), be

\[
I(x_i; y_j) = \log_2 \frac{P(x_i, y_j)}{P(x_i)P(y_j)} \quad (A4.15)
\]

the average (or expected) mutual information between \( X \) and \( Y \) is

\[
I(X;Y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \frac{P(x_i, y_j)}{P(x_i)P(y_j)} \log_2 \frac{P(x_i, y_j)}{P(x_i)P(y_j)}
\]

\[
= \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \frac{P(x_i)P(y_j / x_1)}{P(x_i)P(y_j / x_1)} \log_2 \left( \frac{P(x_i)}{P(x_i)P(y_j / x_1)} \right)
\]

Finally, the channel capacity, measured by the total number of information bits per symbol interval, is defined to be
\[ C^* = \max \{ I(X;Y) \} \]
\[ = \max \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} P(x_i) P(y_j/x_i) \log_2 \frac{P(y_j/x_i)}{\sum_{k=0}^{m-1} P(x_k) P(y_j/x_k)} \]  
\[ (A4.17) \]

Clearly the definitions given by equations A4.14 and A4.17 lead to the same expression for \( C^* \).

Now, consider the satellite channel model described in Section 2.1. Here, the set of discrete signals (complex-valued) fed to the linear channel is \( \{q(h)\} \) (see also Section 3.2), for \( h = 0, 1, \ldots, (M-1) \), and the corresponding signal at the output of the channel, at any time instant, may be considered as a complex-valued continuous-variable \( r \). In addition, following equation 2.1.11,

\[ r = q(h) + w \]  
\[ (A4.18) \]

The real and imaginary parts of \( w \) are sample values of Gaussian random variables with zero mean and fixed variance \( \sigma^2 \). The probability density function of \( w \) may be generally expressed as

\[ f(w) = \frac{1}{2\pi\sigma^2} \exp\left( -\frac{|w|^2}{2\sigma^2} \right) \]  
\[ (A4.19) \]

Since \( r \) is a continuous variable, the summation over \( j = 0, 1, \ldots, (n-1) \), in equation A4.14, now becomes the integration over \( r \), and the channel capacity expression becomes
where \( P(q(h)) \) is the a-priori probability of \( q(h) \), and \( f(r/q(h)) \) is the conditional probability density function of \( r \), given \( q(h) \), for \( i=0, 1, \ldots, (M-1) \). Again, the maximisation is carried out over all possible probability assignments of \( P(q(h)) \), for \( i=0, 1, \ldots, (M-1) \).

The following mathematical analysis of the channel capacity expression (equation A4.20) leads to the deviation of equation 3.2.12 in Section 3.2. Since it is assumed that the transmitted signal is equally likely to have any of its \( M \) possible values \((q(h))\), the maximisation in equation A4.20 can be omitted, and the a-priori probability of \( q(h) \), for \( i=0, 1, \ldots, (M-1) \), is given by

\[
P(q(h)) = \frac{1}{M}
\]  

(A4.21)

Following equation A4.20,

\[
C^* = \max \sum_{h=0}^{M-1} \int_{-\infty}^{\infty} \frac{f(r/q(h)) \log_2 f(r/q(h))}{\sum_{k=0}^{M-1} f(r/q(k))} \, dr
\]  

(A4.20)
It follows from equations A4.12 and A4.13 that,
\[\int_{-\infty}^{+\infty} f(r/q(h)) \, dr = 1\] (A4.23)

Furthermore, for any given \(q(h)\), the conditional probability function \(f(r/q(h))\) must be given by
\[
f(r/q(h)) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|q(h)+w-q(h)|^2}{2\sigma^2}\right) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|w|^2}{2\sigma^2}\right)\] (A4.24)

and similarly, for any given \(q(h)\),
\[
f(r/q(k)) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{|q(h)+w-q(k)|^2}{2\sigma^2}\right)\] (A4.25)

so that,
\[
\frac{f(r/q(k))}{f(r/q(h))} = \exp\left(-\frac{|q(k)+w-q(h)|^2-|w|^2}{2\sigma^2}\right)\] (A7.26)

Consequently, equation A4.22 becomes
\[
C^* = \log_2 M - \sum_{h=0}^{M-1} \int_{-\infty}^{+\infty} f(r/q(h)) \log_2 \left( \sum_{k=0}^{M-1} \exp\left(-\frac{|q(h)+w-q(k)|^2-|w|^2}{2\sigma^2}\right) \right) \, dr
\] (A4.27)

Now, if the integration in equation A4.27 is replaced by the
corresponding expectation, it becomes

\[ C^* = \log_2 M - \frac{1}{M} \sum_{h=0}^{M-1} E( \log_2 \sum_{k=0}^{M-1} \exp(-\frac{|q(h)+w-q(k)|^2 - |w|^2}{2\sigma^2}) ) \]  

(A4.28)

The final part of this section presents the derivation of the well known Shannon limit expression.

Consider an arbitrary Gaussian-distributed variable \( z \) (also complex-valued), where the probability density function of \( z \) is given by

\[ f(z) = \frac{1}{\sqrt{2\pi N}} \exp\left(-\frac{z^2}{2N}\right) \]  

(A4.29)

Clearly,

\[-\log_2 f(z) = \log_2 (2\pi N) + \log_2 e(z^2/2N) \]  

(A4.30)

The entropy of \( z \) is generally defined by [15]

\[ H(z) = -\int f(z) \log_2 f(z) \, dz \]

\[ z \]

\[ = \log_2 (2\pi N) \int f(z) \, dz + \log_2 e/2N \left[ \int f(z)|z|^2 \, dz \right] \]

\[ z \]

\[ = \log_2 (2\pi N) + \log_2 e/2N \left[ 2N \right] \]

\[ = \log_2 (2\pi eN) \]  

(A4.31)

where \( 2N = \int p(z)|Z|^2 \, dz \)

\[ z \]

\[ = E( |z|^2 ) \]  

(A4.32)

From equation A4.18,

\[ r = q(h) + w \]  

(A4.33)
The expected $|r|^2$ and $|w|^2$ are here given by

$$E(|r|^2) = E(|q(h)|^2) + E(|w|^2)$$
\[= E + 2\sigma^2 \quad (A4.34)\]

and

$$E(|w|^2) = 2\sigma^2 \quad (A4.35)$$

respectively. Furthermore, from equation A4.9, the channel capacity can be else defined as

$$C^* = H(r) - H(r/q(h)) \text{ (bits/T)} \quad (A4.36)$$

Since $w$ is independent of $x$ for a Gaussian channel, $f(r/q(h))$ is a function of $w$ only, and thus

$$H(r/q(h)) = H(w) \quad (A4.37)$$

Therefore,

$$C^* = H(r) - H(w)$$
\[= \log_2(2\pi e[E_g + 2\sigma]) + \log_2(2\pi e 2\sigma) \quad (A4.38)\]
\[= \log_2(1 + E_g/2\sigma^2) \quad (A4.38)\]

Following equation 2.3.29,

$$\sigma^2 = (1/2)N_o \quad (A4.39)$$

so that

$$C^* = \log_2(1 + E_g/N_o) \text{ (bits/T)} \quad (A4.40)$$

This is the well known Shannon's limit expression and is as shown in Figure 3.2.3.

Clearly, the channel capacity, measured by the total number of information bits per second, is
and the average signal energy per data bit $E_b$, is given by

$$E_b = \frac{E_s}{L} \quad \text{(A4.42)}$$

where $L$ is the number of information bits per symbol, or per symbol interval. Thus, if the given data transmission system operates at the highest possible information rate $C$ (bits/second), then

$$L = C \times T \quad \text{(A4.43)}$$

Using equations A4.41 to A4.43,

$$C = \frac{1}{T \log_2(1 + \frac{E_b}{N_0} \times \frac{T}{C})} \quad \text{(bits/second)} \quad \text{(A4.44)}$$

Suppose that the system transmits the digital signal at the Nyquist rate, then the double-sided signal bandwidth $B$ may be expressed as [8],

$$B = \frac{1}{T} \quad \text{(A4.45)}$$

Thus,

$$C = B \log_2(1 + \frac{C}{B} \times \frac{E_b}{N_0}) \quad \text{(bits/second)} \quad \text{(A4.46)}$$

Equation A4.46 establishes the maximum rate at which information can be reliably transmitted over a memoryless Gaussian channel. The Shannon's channel capacity theorem
states that, whenever the information transmission rate $RT$ (bits/second) is less than $C$, some hypothetical coding, modulation, demodulation and decoding schemes must exist which yield an arbitrarily small error probability. Conversely, if $RT$ is greater than $C$, then regardless the coding or modulation technique used, the error probability will be bounded from zero [14].

Now, for an infinite bandwidth, the asymptotic value of $C$ is given by

$$C_\infty = \lim_{B \to \infty} C = \lim_{B \to \infty} \log_2(1 + \frac{C_\infty}{B} \times \frac{E_B}{N_0})$$

$$= \left(\frac{1}{\ln 2}\right) C_\infty \frac{E_B}{N_0} \quad (A4.47)$$

using the relation [7]

$$\lim_{n \to \infty} \ln(1 + x/n)^n = x$$

$$\lim_{n \to \infty} \ln(1 + x/n)^n = x \quad (A4.48)$$

Consequently, the minimum bound on $E_B/N_0$ is

$$E_B/N_0 = \ln 2 = -1.6 \text{ dB} \quad (A4.49)$$

This is called the Shannon's limit, and it represents the minimum $E_B/N_0$ required for an infinite signal bandwidth and unlimited coding/decoding effort, if the error probability is to be made as small as desired. Naturally, since the available power is limited on the satellite, it is desirable to design the transmission system that operates with an $E_B/N_0$ as close as possible to this limit. However, the selection of the particular coding, modulation, demodulation, and decoding schemes is an engineering problem which depends on the
theoretical and practical factors, such as the available signal bandwidth, implementation cost and complexity, and so on.

Another important parameter in establishing the efficiency of a communication system is the so-called bandwidth efficiency, defined as the number of information bits per Hertz of channel bandwidth. Clearly, the maximum bandwidth efficiency is given by

\[ C/B = \log_2[1 + (E_b/N_0)(C/B)] \quad \text{(bits/Hz)} \quad \text{(A4.50)} \]

or equivalently,

\[ E_b/N_0 = \frac{1}{C/B} \left( 2^{C/B} - 1 \right) \quad \text{(A4.51)} \]

Taking the bandwidth efficiency as a design criterion, different coding and modulation schemes can be compared to each other.
In this section, the definition of the signal/noise ratio is developed, the more important computer simulation techniques are introduced, and a burst error is defined. Finally, the confidence limits of computer simulation results are discussed.

The signal/noise in the received samples \( r_i \), at the output of the coherent demodulator, may be generally defined to be

\[
\text{snr} = 10 \log_{10} \left[ \frac{E(|q_i|^2)}{E(|w_i|^2)} \right]
\]  

(A5.1)

where \( E(.) \) represents the expect value of the corresponding quantity. \( q_i \) is the transmitted code symbol (complex-valued), and the real and imaginary parts of the noise components \( \{w_i\} \) are sample values of Gaussian random variables with zero mean and fixed variance \( \sigma^2 \). Following equation 2.1.7, the average signal energy per transmitted symbol is given by

\[
E(|q_i|^2) = E_b
\]

\[
= E(\text{Re}(q_i)^2) + E(\text{Im}(q_i)^2)
\]

\[
= 1.0
\]  

(A5.2)

where \( \text{Re}(.) \) and \( \text{Im}(.) \) represent, respectively, the real and imaginary parts of the corresponding quantity. For a general \( 2^L \)-level data transmission system, the average signal energy per data bit, \( E_b \), is given by
\[ E_b = \frac{E_s}{L} \]
\[ = \frac{E(|q_i|^2)}{L} \quad (A5.3) \]

The additive white Gaussian noise in the channel has a zero mean and a constant two-sided power spectral density of \((1/2)N_0\). Following equation 2.3.29, the variance of the noise samples \(\{w_i\}\) along either real and imaginary axes in the complex number plane is

\[ \sigma^2 = \frac{(1/2)N_0}{L} \quad (A5.4) \]

for the assumed linear baseband channel transfer-function shown in Figure 2.1.2. The total noise variance \(E(|w_i|^2)\) is now

\[ E(|w_i|^2) = E((Re(w_i))^2) + E((Im(w_i))^2) \]
\[ = \sigma^2 + \sigma^2 \]
\[ = 2\sigma^2 \quad (A5.5) \]

Consequently, equation A5.1 becomes

\[ \text{snr} = 10\log_{10}\left(\frac{E_s}{2\sigma^2}\right) \quad (dB) \]
\[ = 10\log_{10}\left(\frac{L \cdot E_b}{2(1/2)N_0}\right) \quad (dB) \]
\[ = 10\log_{10}\left(\frac{E_b}{N_0}\right) \quad (db) \quad (A5.6) \]

and thus,

\[ \text{snr} = 10\log_{10}(E_b/N_0) + 10\log_{10}L \quad (dB) \quad (A5.7) \]
In all graphs showing error rate performances of the systems, the signal/noise ratio defined above is adjusted to give the curves of bit error rate against $E_b/N_0$.

The computer simulation tests carried out in this work employ the standard NAG (Numerical Algorithm generator) random number generator library subroutines to generate the random data symbols $\{s_i\}$ and the Gaussian noise samples $\{w_i\}$. $\{s_i\}$ are generated with a uniform distribution, and $\{w_i\}$ are generated by the Gaussian random number generator with zero mean and a standard deviation $\sigma$. All simulation programs used in the tests are written in Fortran 77 language. For each particular system tested, the noise standard deviation $\sigma$ is varied to give the required signal/noise ratio, snr (dB), resulting correspondingly in the bit error rates of the decoded data symbols, over the range of $10^{-1}$ to $10^{-4}$.

A burst error is now defined as follows. Following an incorrect decoded data symbol, if 16 or more subsequent decoded symbols are correct, the next incorrect decoded symbol will be considered as the start of a new error burst. Otherwise this error symbol will be taken as part of the present error burst. The number 16 is considered here to be sufficient large for the first error symbol in an error burst to be independent of all errors in the previous burst.

The confidence limits of the simulation results presented in this thesis are now examined. In tests to determine the tolerance of a digital data transmission system to additive white Gaussian noise, a sufficient large number of data symbols must be transmitted in a test, in order to obtain
a reasonably accurate measure of the tolerance to noise, at a given bit error rate [97]. At each particular signal/noise ratio considered, the accuracy or reliability in a given measure of the bit error probability is determined by the total number of independent error bursts, \( n_b \), that occur in the test [97]. Supposing the measured bit error probability is \( P_b \), where \( P_b < 1.0 \) and \( n_b > 30 \), it can be shown that the 95% confidence limits for the value of \( P_b \) is approximately

\[
P_b \pm \frac{2}{\sqrt{n_b}} P_b \quad (A5.8)
\]

[97]. Since, at low bit error rates, a doubling of the error rate only corresponds to a change of well below 1 dB in the signal/noise ratio, the change reducing as the signal/noise ratio increases, quite reasonable confidence limits can be achieved when \( n_b > 30 \) [97].

In all tests carried out in this investigation, the total number of transmitted data symbols used in a test, at each particular signal/noise ratio, is adjusted to produce more than 50 (i.e. \( n_b > 50 \)) isolated error bursts whenever this is possible. Under these conditions, the 95% confidence limits for the value of \( P_b \) are approximately

\[
P_b \pm \frac{2}{\sqrt{n_b}} P_b = P_b (1.0 \pm 0.283) \quad (A5.9)
\]
**B1 PROGRAM FOR THE UNCODED QPSK SYSTEM WITH COHERENT DETECTION**

This program simulates an uncoded QPSK data transmission system over an additive white Gaussian noise channel. The coherent decision threshold detection process is incorporated at the receiver, and correct timing and carrier phase synchronisation is assumed.

```plaintext
double precision r1, r2, w1, w2, p1, p2, q1, q2, s
double precision sd, snr, power, pnoise, pi, angle, h
double precision g05daf, g05ddf
integer dets1, dets2, dphase, data, map, symer, biter
call g05cbf(1)

c Various system parameters are set.

no=100000
snr=10.00
pi=datan(1.0d00)*4.0
h=0.250
angle=pi*h
power=0.5
symer=0
biter=0
pnoise=power/(10.0**(snr/10.0))
sd=dsqrt(pnoise)

c The transmission begins.

do 350 ll=1, no, 1

   c The quaternary data symbol s_i is generated
   c and it is then mapped onto the complex number
   c plane to give the transmitted symbol q_i.

   s=g05daf(0.0d00,2.0d00)
   if(s-1.0) 100, 100, 110
60 ia1=0
61 go to 120
90 ia1=1
120 continue
   s=g05daf(0.0d00,2.0d00)
   if(s-1.0) 130, 130, 140
150 ia2=0
160 go to 150
190 ia2=1
200 continue
   is=map(ia1, ia2)
   ih=2*is-3
   p1=dcos(angle*ih)
   p2=dsin(angle*ih)
```

### Notes
- The code uses the `g05daf` and `g05ddf` functions for generating random numbers.
- The `data` variables are not defined in the provided code snippet.
- The `map` function maps the quaternary data symbol to the complex plane.
- The `dphase` variable is not defined in the provided code snippet.
- The `symer` variable is used to determine the symmetry of the data.
- The `biter` variable is used to determine the bit error rate.
The noise component $w_i$ is added onto $q_i$.

\[
\begin{align*}
\omega_1 &= g05ddf(0.0d00, sd) \\
\omega_2 &= g05ddf(0.0d00, sd) \\
r_1 &= p1 + w_1 \\
r_2 &= p2 + w_2
\end{align*}
\]

The detection starts

\[
\begin{align*}
&\text{if}(r_1 < 0.0) \quad 190,190,160 \\
&\text{if}(r_2 < 0.0) \quad 170,170,160 \\
&\text{idh} = 1 \\
&\text{go to 220} \\
&\text{idh} = 2 \\
&\text{go to 220} \\
&\text{idh} = 3 \\
&\text{go to 220} \\
&\text{idh} = 0 \\
&\text{continue}
\end{align*}
\]

\[
\begin{align*}
\text{dphase} &= \text{idh} \\
\text{dets1} &= \text{data(dphase, 1)} \\
\text{dets2} &= \text{data(dphase, 2)} \\
\text{is1} &= \text{data(is, 1)} \\
\text{is2} &= \text{data(is, 2)}
\end{align*}
\]

Bit errors and symbol errors are counted.

\[
\begin{align*}
&\text{if}(\text{dphase} < \text{is}) \quad 230,240,230 \\
&\text{symer} = \text{symer} + 1 \\
&\text{continue} \\
&\text{if}(\text{dets1} < \text{is1}) \quad 300,305,300 \\
&\text{biter} = \text{biter} + 1 \\
&\text{if}(\text{dets2} < \text{is2}) \quad 310,315,310 \\
&\text{biter} = \text{biter} + 1 \\
&\text{continue}
\end{align*}
\]

The process continues with the next data symbol $s_i$.

\[
\begin{align*}
\text{ber} &= \text{float(biter)} / \text{float(no)} * 0.50 \\
\text{symbol} &= \text{float(symer)} / \text{float(no)} \\
\text{write(*,4800)} \\
\text{write(*,4900)} \\
\text{write(*,4910)} \\
\text{write(*,5003)} \ h \\
\text{write(*,5005)} \ \text{snr} \\
\text{write(*,5010)} \ \text{ber} \\
\text{write(*,5015)} \ \text{symbol} \\
\text{write(*,5025)} \ \text{no} \\
\text{stop}
\end{align*}
\]

4800 format(2x,/) 4900 format(15x,32hthis is the uncoded QPSK system,/) 4910 format(15x,25husing threshold detection,/)
c This function generates the two binary data symbols for each given symbol value \( s_i \).

integer function data(x,y)
integer x,y
data=10000000

1000 go to (1000,1010,1020,1030) x+1
1000 if(y.eq.1) data=0
1000 if(y.eq.2) data=0
1000 go to 1050
1010 if(y.eq.1) data=0
1010 if(y.eq.2) data=1
1010 go to 1050
1020 if(y.eq.1) data=1
1020 if(y.eq.2) data=1
1020 go to 1050
1030 if(y.eq.1) data=1
1030 if(y.eq.2) data=0
1050 return
end

c This is the Gray coding function.

integer function map(x,y)
integer x,y
map=10000000
if(x.eq.0.and.y.eq.0) map=0
if(x.eq.0.and.y.eq.1) map=1
if(x.eq.1.and.y.eq.1) map=2
if(x.eq.1.and.y.eq.0) map=3
return
end
Program for Rate-2/3 Convolutionally Coded 8PSK Signals with the Viterbi Decoding

This program simulates a quaternary data transmission system over an additive white Gaussian noise channel. It incorporates the rate-2/3 convolutional code UC(2) (a 8-state code, see Table 3.2.1) and the Viterbi decoding scheme (see Section 3.2) at the receiver. The resultant signal is an 8PSK signal. The decoder has 8 stored vectors and the delay in detection is 48 symbols (n=48).

double precision r1, r2, w1, w2, p1, p2, q1, q2, c(64), cc(256)
double precision snr, power, pnoise, pi, angle, cost
double precision g05daf, g05ddf, ccc, s
integer is0(80), isl(80), ix0(64, 80), ixl(256, 80), ix0(64, 80)
integer dets1, dets2, dphase, s, mer, biter, ix0(256, 80)
integer burst, count, range, map
call g05cbf(78)

System parameters are set.

no=100000
n=48
n1=n-1
l=no+n
snr=7.50
pi=datan(1.0d00)*4.0
angle=pi/4.0
m=8
m4=4*m
power=0.5

do 10 i=1,n,1
  is0(i)=0
  isl(i)=0
10 continue
do 30 i=1,m,1
c(i)=1.0d07
30 continue
do 20 j=1,n,1
  ix0(i,j)=0
  ix1(i,j)=0
20 continue

c(1)=0.0
symer=0
biter=0
burst=0
count=0
pnoise=power/(10.0**((snr/10.0)))
sd=dsqrt(pnoise)

The transmission begins.
do 500 ll=1,1,1
Binary data symbols \(b_i(1)\) and \(b_i(2)\) are generated.

\[
s = \mathrm{g05daf}(0.0d00, 2.0d00)
\]

\[
\text{if}(s-1.0) \ 100, 100, 110
\]

\[
100 \ \text{ia1}=0
\]

\[
go \ \text{to} \ 120
\]

\[
110 \ \text{ia1}=1
\]

\[
120 \ \text{continue}
\]

\[
s = \mathrm{g05daf}(0.0d00, 2.0d00)
\]

\[
\text{if}(s-1.0) \ 130, 130, 140
\]

\[
130 \ \text{ia2}=0
\]

\[
go \ \text{to} \ 150
\]

\[
140 \ \text{ia2}=1
\]

\[
150 \ \text{continue}
\]

\[
is0(n)=\text{ia1}
\]

\[
is1(n)=\text{ia2}
\]

The coded vector \(v\) and the transmitted symbol \(q_i\) are obtained.

\[
\text{ik1}=is1(n)+is0(n-1)
\]

\[
\text{ik2}=is1(n-2)+is0(n)
\]

\[
\text{ik3}=is1(n-1)
\]

\[
\text{ik1}=\text{range}(2, \text{ik1})
\]

\[
\text{ik2}=\text{range}(2, \text{ik2})
\]

\[
\text{ik3}=\text{range}(2, \text{ik3})
\]

\[
\text{ik}=4*\text{ik1}+2*\text{ik2}+\text{ik3}
\]

\[
\text{pl} = \text{dcos}(\text{angle} \times \text{ik})
\]

\[
\text{p2} = \text{dsin}(\text{angle} \times \text{ik})
\]

The noise component \(w_i\) is added onto \(q_i\):

\[
\text{w1} = \mathrm{g05ddf}(0.0d00, \text{sd})
\]

\[
\text{w2} = \mathrm{g05ddf}(0.0d00, \text{sd})
\]

\[
\text{r1} = \text{pl} + \text{w1}
\]

\[
\text{r2} = \text{p2} + \text{w2}
\]

The detection process starts.

The 8 stored vectors \(\{X_{i,k}\}\) are expanded to form 32 vectors \(\{Z_i\}\).

\[
do \ 180 \ i=1,m,1
\]

\[
do \ 170 \ j=1,n1,1
\]

\[
\text{ixx}0(4*i-3,j)=\text{ixx}0(i,j+1)
\]

\[
\text{ixx}0(4*i-2,j)=\text{ixx}0(i,j+1)
\]

\[
\text{ixx}0(4*i-1,j)=\text{ixx}0(i,j+1)
\]

\[
\text{ixx}0(4*i,j)=\text{ixx}0(i,j+1)
\]

\[
\text{ixx}1(4*i-3,j)=\text{ixx}1(i,j+1)
\]

\[
\text{ixx}1(4*i-2,j)=\text{ixx}1(i,j+1)
\]

\[
\text{ixx}1(4*i-1,j)=\text{ixx}1(i,j+1)
\]

\[
\text{ixx}1(4*i,j)=\text{ixx}1(i,j+1)
\]
The associated costs \( c \cdot \) are calculated.

\begin{align*}
do & \ i=1, m4, 1 \\
i k1 &= i x x 1 (i, n) + i x x 0 (i, n-1) \\
i k2 &= i x x 1 (i, n-2) + i x x 0 (i, n) \\
i k3 &= i x x 1 (i, n-1) \\
i k1 &= \text{range}(2, i k1) \\
i k2 &= \text{range}(2, i k2) \\
i k3 &= \text{range}(2, i k3) \\
i k &= 4 * i k1 + 2 * i k2 + i k3 \\
q1 &= \cos(\text{angle}\cdot i k) \\
q2 &= \sin(\text{angle}\cdot i k) \\
c c (i) &= (r1-q1) \cdot (r1-q1) + (r2-q2) \cdot (r2-q2) \\
continue &
\end{align*}

The 8 stored vectors \( \{Z_i\} \) with the smallest costs for the corresponding states are selected.

\begin{align*}
do & \ i=1, m, 1 \\
c c c &= 1.0 d 1 0 \\
m3 &= 3 * m + i \\
do & \ j=i, m3, m \\
if & (c c (j)-c c c) 210, 220, 220 \\
210 & c c c = c c (j) \\
it &= j \\
220 & continue \\
do & \ j=1, n, 1 \\
i x 0 (i, j) &= i x x 0 (i t, j) \\
i x 1 (i, j) &= i x x 1 (i t, j) \\
230 & continue \\
c (i) &= c c (i t) \\
240 & continue \\
c &
\end{align*}

The decoded symbol \( s' \) is determined.

\begin{align*}
c c c &= 1.0 d 0 7 \\
do & \ i=1, m, 1 \\
if & (c (i)-c c c) 250, 260, 260 \\
\end{align*}
ccc=c(i)
it=it+1
continue
cost=c(it)

dets1=ix0(it,1)
dets2=ix1(it,1)
dphase=map(dets1,dets2)
iss1=is0(1)
iss2=is1(1)
idp=map(iss1,iss2)

c Bit errors and burst errors are counted.

if(dphase-idp) 270,290,270
  symer=symer+1
  if(count-16) 280,280,275
  burst=burst+1
  count=0
  go to 295
  count=count+1
  continue

if(dets1-iss1) 300,305,300
  biter=biter+1
  if(dets2-iss2) 310,315,310
  biter=biter+1
  continue

c The lowest cost is subtracted from each stored cost value.

do 320 i=1,m,1
  c(i)=c(i)-cost
  continue

c The process continues to the next data symbol s_i.

ber=float(biter)/float(no)*0.50
symbol=float(symer)/float(no)
burstrate=float(burst)/float(no)
write(*,4800)
write(*,4900)
write(*,5005) snr
write(*,5010) ber
write(*,5015) symbol
write(*,5020) burstrate
write(*,5025) no
write(*,5030) n
stop
4800 format(2x,/////)
4900 format(15x,43this is a convolutionally coded 8PSK system,/)  
4910 format(15x,26using the Viterbi decoding,/)
The signal to noise ratio is: \( f(8.3,///) \)

The bit error rate is: \( f(9.6,///) \)

The symbol error rate is: \( f(9.6,///) \)

The burst rate is: \( f(9.6,///) \)

The errors are counted from detection of \( i_{1}i_{2}i_{3}i_{4} \) received data symbols.

The effective delay is: \( i_{6} \)

This function carries out modulo-M operation.

```
integer function range(ix,iy)
range=10000000
3000 if(iy-O) 3010,3040,3020
3010 iy=iy+ix
go to 3000
3020 if(iy-ix) 3040,3030,3030
3030 iy=iy-ix
go to 3020
3040 range=iy
return
end
```

This is the Gray coding function.

```
integer function map(ix,iy)
if(ix.eq.0.and.iy.eq.0) map=0
if(ix.eq.0.and.iy.eq.1) map=1
if(ix.eq.1.and.iy.eq.1) map=2
if(ix.eq.1.and.iy.eq.0) map=3
return
end
```
%global ansi77,card

c B3 PROGRAM FOR RATE-2/3 CONVOLUTIONALLY CODED 8PSK

PROGRAM FOR RATE-2/3 convolutionally coded 8PSK signals with the decoding scheme of system A

This program simulates a quaternary data transmission system over an additive white Gaussian noise channel.
It incorporates the rate-2/3 convolutional code UC(3) (see Table 3.2.1) to generate an 8PSK signal and the decoding scheme of system A (Section 3.4) at the receiver.
The code UC(3) has 16 states and the decoder holds 4 stored vectors (m=4). The detection delay is 48 symbols (n=48).

double precision r1,r2,w1,w2,p1,p2,q1,q2,c(64),cc(256)
double precision sd,snr,power,pnoise,pi,angle,cost
double precision g05daf,g05ddf,h,ccc,s
integer isO(50),isl(50),ix0(16,50),ix1(16,50),ixx0(64,50)
integer dets1,dets2,phase,system,biter
integer burst,count,range,map,ixx(64,50)
call g05cbf(200)

The system parameters are set.

no=100000
n=48
n1=n-1
l=no+n
snr=9.50
pi=datan(1.0d00)*4.0
h=1.0/6.0
angle=pi*h*2.0
m=4
m4=4*m
power=0.5

do 10 i=1,n,1
   isO(i)=0
   isl(i)=0
10 continue
do 30 i=1,m,1
c(i)=1.0d07
30 continue
do 20 j=1,n,1
   ix0(i,j)=0
   ix1(i,j)=0
20 continue
do 20 j=1,n,1
30 continue
c(1)=0.0
system=0
biter=0
burst=0
count=0
pnoise=power/(10.0**(snr/10.0))
sd=dsqrt(pnoise)

The transmission begins.
do 500 ll=1,1,1
The binary data symbols $b_i(1)$ and $b_i(2)$ are generated.

```
s=g05daf(0.0d00,2.0d00)
if(s<1.0) 100,100,110
100 ia1=0
   go to 120
110 ia1=1
120 continue
   s=g05daf(0.0d00,2.0d00)
   if(s<1.0) 130,130,140
130 ia2=0
   go to 150
140 ia2=1
150 continue
   is0(n)=ia1
   is1(n)=ia2
```

The coded vector $v_i$ and transmitted symbol $q_i$ are generated.

```
   ik1=is1(n)+is0(n-1)+is0(n-2)
   ik2=is1(n-2)+is0(n)
   ik3=is1(n-1)
   ik1=range(2,ik1)
   ik2=range(2,ik2)
   ik=4*ik1+2*ik2+ik3
   p1=dcos(angle*ik)
   p2=dsin(angle*ik)
```

The noise component $w_i$ is added onto $q_i$.

```
   w1=g05ddf(0.0d00,1d0)
   w2=g05ddf(0.0d00,1d0)
   r1=p1+w1
   r2=p2+w2
```

The detection starts.

The $m$ stored vector $\{X_i\}$ are expanded to form $4m$ vectors $\{Z_i\}$.

```
do 170 i=1,m,1
do 160 j=1,n1,1
   ixx1(4*i-3,j)=ix1(i,j)
   ixx1(4*i-2,j)=ix1(i,j)
   ixx1(4*i-1,j)=ix1(i,j)
   ixx1(4*i,j)=ix1(i,j)
   ixx0(4*i-3,j)=ix0(i,j)
   ixx0(4*i-2,j)=ix0(i,j)
   ixx0(4*i-1,j)=ix0(i,j)
   ixx0(4*i,j)=ix0(i,j)
```


```
ixx0(4*i,j)=ix0(i,j)
continue
ixx1(4*i-3,n)=0
ixx0(4*i-3,n)=0
ixx1(4*i-2,n)=0
ixx0(4*i-2,n)=1
ixx1(4*i-1,n)=1
ixx0(4*i-1,n)=0
ixx1(4*i,n)=1
ixx0(4*i,n)=1
cc(4*i-3)=c(i)
cc(4*i-2)=c(i)
cc(4*i-1)=c(i)
cc(4*i)=c(i)
continue

c The 4m associated costs \{c_i\} are calculated.

do 180 i=1,m4,1
  ik1=ixx1(i,n)+ixx0(i,n-1)+ixx0(i,n-2)
  ik2=ixx1(i,n-2)+ixx0(i,n)
  ik3=ixx1(i,n-1)
  ik1=range(2,ik1)
  ik2=range(2,ik2)
  ik=4*ik1+2*ik2+ik3
  q1=dcos(angle*ik)
  q2=dsn(angle*ik)
  cc(i)=cc(i)+(r1-q1)*(r1-q1)+(r2-q2)*(r2-q2)
continue

cc The lowest-cost vector is selected and

c the decoded symbol \(s'_{-n}\) is determined.

ccc=1.0d10
do 230 i=1,m4,1
  if(cc(i)-ccc) 220,230,230
  ccc=cc(i)
  it=i
continue
  cost=cc(it)
  cc(it)=1.0d14

dets1=ixx0(it,1)
dets2=ixx1(it,1)
dphase=map(dets1,dets2)
iss1=is0(1)
iss2=is1(1)
idp=map(iss1,iss2)

cc The bit errors and burst errors are counted.

if(dphase-idp) 270,290,270
  symer=symer+1
  if(count=16) 280,280,275
  burst=burst+1
  count=0
```
go to 295
290  count=count+1
295  continue

if(dets1-iss1) 300,305,300
300  biter=biter+1
305  if(dets2-iss2) 310,315,310
310  biter=biter+1
315  continue

Antimerging procedure:

do 330 i=1,m4,1
ip=map(ixx0(i,1),ixx1(i,1))
if(ip-dphase) 320,330,320
320  cc(i)=1.0d14
330  continue

The lowest-cost vector is stored.

do 335 j=1,n1,1
ix0(i,j)=ixx0(it,j+1)
ix1(i,j)=ixx1(it,j+1)
335  continue

c(1)=0.0

The (m-1) smallest cost vectors are selected.

do 450 i=2,m,1
ccc=1.0d10
do 360 j=1,m4,1
if(cc(j)-ccc) 350,360,360
350  ccc=cc(j)
it=j
360  continue

do 390 j=1,n1,1
ix0(i,j)=ixx0(it,j+1)
ix1(i,j)=ixx1(it,j+1)
390  continue

c(i)=cc(it)-cost
cc(it)=1.0d14
450  continue

The process continues with the next data symbol s_i+1.

500  continue

ber=float(biter)/float(no)*0.50
symbol=float(symer)/float(no)
burate=float(burst)/float(no)
write(*,4800)
write(*,4900)
write(*,4910)
write(*,5003) h
write(*,5005) snr
write(*,5006) m
write(*,5010) ber
write(*,5015) symbol
write(*,5020) burate
write(*,5025) no
write(*,5030) n
stop

4800 format(2x,\\\\)
4900 format(15x,43htthis is a convolutionally coded 8PSK system,\\)
4910 format(15x,24htusing system A decoding,\\)
5003 format(2x,26hthe modulation index h is:,3x,f8.3,\\)
5005 format(2x,29hthe signal to noise ratio is:,f8.3,\\)
5006 format(2x,24hthe no. of stored vector is:,i5,\\)
5010 format(2x,22hthe bit error rate is:,7x,f9.6,\\)
5015 format(2x,25hthe symbol error rate is:,4x,f9.6,\\)
5020 format(2x,18hthe burst rate is:,11x,f9.6,\\)
5025 format(2x,40htthe errors are counted from detection of,\\)
\  
\  
\  
5030 format(2x,22htthe effective delay is,i6,\\)
end

This is the Gray coding function.

integer function map(x,y)
integer x,y
map=10000000
if(x.eq.0.and.y.eq.0) map=0
if(x.eq.0.and.y.eq.1) map=1
if(x.eq.1.and.y.eq.1) map=2
if(x.eq.1.and.y.eq.0) map=3
return
end

The function carries out modulo-M operation.

integer function range(ix,iy)
5000 if(iy-0) 5010,5050,5020
5010 iy=i+y+ix
   go to 5000
5020 if(iy-ix) 5050,5030,5030
5030 iy=i+y+ix
   go to 5020
5050 range=iy
return
end
PROGRAM FOR AN EXHAUSTIVE SEARCH FOR THE BEST CORRELATIVE CODE, FOR BINARY DATA SYSTEMS

This program shows how a computer aided search can be carried out to determine the correlative-level code with the best minimum free unitary distance for the given code constraint length \((K+1)\) and signal level \(M\). In this example, binary data symbols are assumed. \(K=2\), and \(M=4\) which generates a QPSK signal.

```fortran
integer a(100),y(50),p(100),range,g
double precision pi,angle,th,dcos,d,dth,small
g=3
m=4
dth=8.0
l=3
n=2*g+6
pi=datan(1.0d00)*4.0
angle=2*pi/m
m1=m-1
m2=m+1

do 280 j1=1,m1,1
  y(1)=j1
do 270 j2=1,m2,1
  y(2)=j2-1-m/2
do 260 j3=1,m1,1
  y(3)=j3
  if (y(3).gt.y(1)) go to 250
  if (abs(y(1)).eq.abs(y(2))) go to 10
  go to 20
10  if (abs(y(2)).eq.abs(y(3))) go to 250

20  For each coding vector considered, the minimum unitary distance is determined.

small=1000000.0
do 30 i=1,n,1
  a(i)=0
  p(i)=0
30  continue

  a(g)=1
  do 170 i3=1,1,1
    a(g+6)=i3-2
  do 165 i4=1,1,1
    a(g+1)=i4-2
  do 160 i5=1,1,1
    a(g+2)=i5-2
  do 155 i6=1,1,1
    a(g+3)=i6-2
  do 150 i7=1,1,1
    a(g+4)=i7-2

  do 50 i=g,n,1
    ik=0
```

...
do 40 j=1,g,1
ik=ik+a(i+1-j)*y(j)
40 continue
p(i)=range(m,ik)
50 continue

d=0.0
70 do i=g,n,1
5 d=d+2.0*(1.0-d*cos(angle*p(i)))
if (d.gt.small) go to 90
continue
if(d-small) 80,90,90
80 small=d
90 continue
150 continue
155 continue
160 continue
165 continue
170 continue
if(small-dth) 250,200,200
200 print*,g,m,small
print*, (y(j),j=1,g)
print*
250 continue
260 continue
270 continue
280 continue
stop
end

c This function carries out modulo-M operation.
integer function range(ix,iy)
2000 if(iy-0) 2010,2040,2020
2010 iy=iy+ix
2020 go to 2000
2040 iy=iy-ix
2030 go to 2020
2040 range=iy
return
end
This program simulates a quaternary data transmission system over an additive white Gaussian noise channel. It incorporates the rate-2/3 convolutional code UC(3) (a 16-state code generating an 8PSK signal, see Table 3.2.1), and the decoding scheme of system A1 (see Section 5.1) at the receiver.

double precision r1,r2,w1,w2,p1,p2,q1,q2,c(32),cc(128)
double precision sd,snr,power,pnoise,pi,angle,cost
double precision g05daf,g05ddf,ccc,s,d1,d2
integer is0(50),isl(50),ix0(32,50),ix1(32,50),ix0(128,50)
integer dets1,dets2,dphase,symer,biter,data
integer burst,count,range,map,ixx1(128,50)
call g05cbf(5)

Various system parameters are set.

no=100000
n=48
n1=n-1
l=no+n
snr=8.10
pi=datan(1.0d00)*4.0
angle=pi/4.0
m=4
mm=32
d1=1.40
d2=d1+0.50
power=0.5

do 10 i=1,n,1
is0(i)=0
isl(i)=0
10 continue
do 30 i=1,m,1
c(i)=1.0d07
do 20 j=1,n,1
ix0(i,j)=0
ix1(i,j)=0
20 continue
30 continue
c(1)=0.0
symer=0
biter=0
burst=0
count=0
pnoise=power/(10.0**(snr/10.0))
sd=dsqrt(pnoise)

The transmission begins.

icot=0
icot1=0
icot2=0
do 500 ll=1,1,1

do 60 i=1,n1,1
is0(i)=is0(i+1)
is1(i)=is1(i+1)
60 continue

c The two binary data symbols \( b_{1}(1) \) and \( b_{1}(2) \) are generated.

\[
s = g05daf(0.0d00, 2.0d00)
\]

if (s<1.0) 100,100,110

100 ia1=0
go to 120

110 ia1=1
120 continue

s = g05daf(0.0d00, 2.0d00)

if (s<1.0) 130,130,140

130 ia0=0
go to 150

140 ia0=1
150 continue

is0(n)=ia0
is1(n)=ia1

c The coded vector \( v_{s} \) and transmitted symbol \( q_{s} \) are determined.

\[
\begin{align*}
&i_{k1} = is1(n) + is0(n-2) \\
&i_{k2} = is1(n-2) + is1(n-3) + is0(n) \\
&i_{k3} = is1(n-1) \\
&i_{k1} = range(2, i_{k1}) \\
&i_{k2} = range(2, i_{k2}) \\
&i_{k3} = range(2, i_{k3}) \\
&ik = 4*i_{k1} + 2*i_{k2} + i_{k3} \\
p1 = dcos(\text{angle}*ik) \\
p2 = dsin(\text{angle}*ik)
\end{align*}
\]

c The noise component \( w_{s} \) is added onto \( q_{s} \).

\[
\begin{align*}
w1 &= g05ddf(0.0d00, sd) \\
w2 &= g05ddf(0.0d00, sd) \\
r1 &= p1+w1 \\
r2 &= p2+w2
\end{align*}
\]

c The decoding process now starts.

c Each stored vector \( X_{w} \) is expanded to form four vectors \( \{Z_{i}\} \) and their costs \( \{c_{i}\} \) are calculated.

c The expanded vectors \( \{Z_{i}\} \) with relative large costs \( \{c_{i}\} \) are eliminated in the initial selection procedure.

ic1=0
ic2=0
do 210 i=1,m,1
ikk1 = ix0(i,n-2)
ikk2 = ix1(i,n-2)+ix1(i,n-3)
akk3 = ix1(i,n-1)

do 200 iss=1,4,1

ia0=data(iss-1,1)
ial=data(iss-1,2)

ik1=ia1+ikk1
ik2=ia0+ikk2
ik1=range(2,ik1)
ik2=range(2,ik2)
q1=dcos(angle*ik)
q2=dsin(angle*ik)
ic1=ic1+1
ccc=c(i)+(r1-q1)*(r1-q1)+(r2-q2)*(r2-q2)

if(ccc-d2) 190,200,200

190 ic2=ic2+1
do 195 j=1,n1,1
ixx0(ic2,j)=ix0(i,j)
ixx1(ic2,j)=ix1(i,j)

195 continue

ixx0(ic2,n)=ia0
ixx1(ic2,n)=ia1
cc(ic2)=ccc

200 continue

210 continue

m2=ic1
m4=ic2

c The minimum cost vector is selected and the decoded symbol s' is determined.

ccc=1.0d10
do 230 i=1,m4

if(cc(i)-ccc) 220,230,230

220 cc=cc(i)
it=i

230 continue

cost=cc(it)
c(it)=1.0d14

dets1=ixx0(it,1)
dets2=ixx1(it,1)
dphase=map(dets1,dets2)

iss1=is0(1)
iss2=is1(1)
idp=map(iss1,iss2)

c The bit errors and burst errors are counted.

if(dphase-idp) 270,290,270
symer = symer + 1
if (count-16) 280, 280, 275
burst = burst + 1
280 count = 0
290 count = count + 1
295 continue

if (dets1 - iss1) 300, 305, 300
biter = biter + 1
305 if (dets2 - iss2) 310, 315, 310
310 biter = biter + 1
315 continue

do 330 i = 1, m4, 1
cc(i) = cc(i) - cost
ip = map (ixx0(i, 1), ixx1(i, 1))
if (ip - dphase) 320, 330, 320
320 cc(i) = 1.0d14
330 continue

do 335 j = 1, n1, 1
ix0(i, j) = ixx0(it, j + 1)
ix1(i, j) = ixx1(it, j + 1)
335 continue

c(1) = 0.0

The final selection procedure: (where the
vectors for which c.< Cm are stored)
ic3 = 1
340 continue
ccc = 1.0d10
do 360 j = 1, m4, 1
if (cc(j) - ccc) 350, 360, 360
350 ccc = cc(j)
it = j
360 continue
if (cc(it).ge.d1) go to 450
ic3 = ic3 + 1
360 continue

ic3 = ic3 + 1
do 390 j = 1, n1, 1
ix0(ic3, j) = ixx0(it, j + 1)
ix1(ic3, j) = ixx1(it, j + 1)
390 continue

c(ic3) = cc(it)
if (ic3.ge.mm) go to 450
cc(it) = 1.0d14
go to 340

450 continue
m = ic3
icot = icot + m
icot1 = icot1 + m2
icot2 = icot2 + m4

The process continues with the next data symbol s.
continue

\[
x_m = \frac{\text{float}(\cot)}{\text{float}(1)}
\]

\[
x_m2 = \frac{\text{float}(\cot1)}{\text{float}(1)}
\]

\[
x_m4 = \frac{\text{float}(\cot2)}{\text{float}(1)}
\]

\[
\text{ber} = \frac{\text{float}(\text{bit})}{\text{float}(\text{no})} \times 0.50
\]

\[
\text{symbol} = \frac{\text{float}(\text{symer})}{\text{float}(\text{no})}
\]

\[
\text{burate} = \frac{\text{float}(\text{burst})}{\text{float}(\text{no})}
\]

write(*,4800)
write(*,4900)
write(*,4910)
write(*,4970) d1
write(*,4980) d2
write(*,4990) mm
write(*,5005) snr
write(*,5006) xm
write(*,5007) xm2
write(*,5008) xm4
write(*,5010) ber
write(*,5015) symbol
write(*,5020) burate
write(*,5025) no
write(*,5030) n
stop

4800 format(2x,//) This is a convolutionally coded 8PSK system,
4900 format(15x,43hthis is a convolutionally coded 8PSK system,)
4910 format(15x,25husing system A1 decoding,)
4970 format(15x,6hd1 is:,f6.3,)
4980 format(15x,6hd2 is:,f6.3,)
4990 format(2x,19hthe max storage is:,i5,)
5005 format(2x,29hthe signal to noise ratio is:,f8.3,)
5006 format(2x,24hno. of stored vector is:,f6.2,)
5007 format(2x,27hno. of cost evaluations is:,f6.2,)
5008 format(2x,36hno. of selected expanded vectors is:,f6.2,)
5010 format(2x,25hthe bit error rate is:,7x,f9.6,)
5015 format(2x,25hthe symbol error rate is:,4x,f9.6,)
5020 format(2x,18hthe burst rate is:,11x,f9.6,)
5025 format(2x,40hthe errors are counted from detection of,
ii8,3x,21hreceived data symbols,)
5030 format(2x,22hthe effective delay is,i6,)
end

This function generates the two binary data symbols for each given symbol value.

integer function data(x,y)
integer x,y
data=10000000
go to (1000,1010,1020,1030) x+1
1000 if(y.eq.1) data=0
if(y.eq.2) data=0
go to 1050
1010 if(y.eq.1) data=0
if(y.eq.2) data=1
go to 1050
1020 if(y.eq.1) data=1
if(y.eq.2) data=0
go to 1050
1030 if(y.eq.1) data=1
if(y.eq.2) data=1
return
dend

c This is the linear mapping function.

integer function map(x,y)
integer x,y
map=100000000
if(x.eq.0.and.y.eq.0) map=0
if(x.eq.0.and.y.eq.1) map=1
if(x.eq.1.and.y.eq.0) map=2
if(x.eq.1.and.y.eq.1) map=3
return
dend

c This function carries out modulo-M operation.

integer function range(ix,iy)
5000 if(iy-0) 5010,5050,5020
5010 iy=iw+ix
go to 5000
5020 if(iy-ix) 5050,5030,5030
5030 iy=iw-ix
go to 5020
5050 range=iy
return
dend
This program simulates a quaternary data transmission system over an additive white Gaussian noise channel. It incorporates the rate-2/3 convolutional code UC(3) (a 16-state code generating an 8PSK signal, see Table 3.2.1) and the decoding scheme of system A2 (see Section 5.2) at the receiver.

The various system parameters are set.

```fortran
no=100000
n=48
nl=no+n
snr=8.50
pi=datan(1.0d00)*4.0
angle=pi/4.0
m=4
mm=32
d1=1.40
d2=d1+0.50
power=0.5

do 10 i=1,n,1
is0(i)=0
isl(i)=0
10 continue

do 30 i=1,m,1
  c(i)=1.0d07
20 continue
30 continue

c(1)=0.0
symer=0
biter=0
burst=0
count=0
pnoise=power/(10.0**(snr/10.0))
sd=dqrt(pnoise)
```

The transmission begins.

icot=0
icot1=0
icot2=0
do 500 i=1,1,1
  do 60 j=1,1,1
    is0(i)=is0(i+1)
    is1(i)=is1(i+1)
  60 continue

c The two binary data symbols \( b_1(1) \) and \( b_1(2) \) are generated.

\[ s=g05daf(0.0d00,2.0d00) \]
if(\( s-1.0 \)) 100,100,110
  100 ia1=0
  go to 120

  110 ia1=1
  120 continue

\[ s=g05daf(0.0d00,2.0d00) \]
if(\( s-1.0 \)) 130,130,140
  130 ia0=0
  go to 150

  140 ia0=1
  150 continue

is0(n)=ia0
is1(n)=ia1

c The coded vector \( v_i \) is generated and the transmitted symbol \( q_i \) is obtained.

\[ i_k1=\text{is1}(n)+\text{is0}(n-1)+\text{is0}(n-2) \]
\[ i_k2=\text{is1}(n-2)+\text{is0}(n) \]
\[ i_k3=\text{is1}(n-1) \]
\[ i_k1=\text{range}(2,i_k1) \]
\[ i_k2=\text{range}(2,i_k2) \]
\[ i_k3=\text{range}(2,i_k3) \]
\[ i_k=4*i_k1+2*i_k2+i_k3 \]
\[ p1=dcos(\text{angle}*i_k) \]
\[ p2=dsin(\text{angle}*i_k) \]

c The noise component \( w_i \) is added onto \( q_i \).

\[ w1=g05ddf(0.0d00,\text{sd}) \]
\[ w2=g05ddf(0.0d00,\text{sd}) \]
\[ r1=p1+w1 \]
\[ r2=p2+w2 \]

c The decoding process now starts.

c The adjustable expansion procedure: (where equation 5.2.6 is used to decide which expanded vectors \( \{Z_i\} \) are valid).

ic1=0
ic2=0
do 210 i=1,m,1
  i_kk1=ix0(i,n-1)+ix0(i,n-2)
ikk2=ixl(i,n-2)
iki=ixl(i,n-1)

do 160 iss=1,4,1
   ia0=data(iss-1,1)
   ia1=data(iss-1,2)
   ik1=ial+ikkk1
   ik2=ia0+ikkk2
   ik1=range(2,ik1)
   ik2=range(2,ik2)
   ik=4*ik1+2*ik2+ik3
   ikk(iss)=ik
   q1=dcos(angle*ik)
   q2=dsin(angle*ik)
   c0(iss)=dabs(r1-q1)+dabs(r2-q2)
160  continue

im=choice(c(i))

do 200 ii=1,im,1

   ccc=1.0d07
   do 180 iss=1,4,1
      if(c0(iss)-ccc) 170,180,180
   170  ccc=c0(iss)
      it=iss
   180  continue

   ik=ikk(it)
   q1=dcos(angle*ik)
   q2=dsin(angle*ik)
   ic1=ic1+1
   ccc=(r1-q1)*(r1-q1)+(r2-q2)*(r2-q2)
   c0(it)=1.0d14

c   The initial selection procedure: (where C'm
   c is used to eliminate the [Z_i] with relative
   c high costs).

   ccc=ccc+c(i)
   if(ccc-d2) 190,200,200
190  ic2=ic2+1
   do 195 j=1,n1,1
      ixx0(ic2,j)=ix0(i,j)
      ixx1(ic2,j)=ix1(i,j)
195  continue
   ixx0(ic2,n)=data(it,1)
   ixx1(ic2,n)=data(it,2)
   cc(ic2)=ccc

200  continue

210  continue
   m2=ic1
   m4=ic2

c   The lowest-cost vector is selected and the
   c decoded symbol s'_{i-n} is determined.
ccc=1.0d10
do 230 i=1,m4,1
  if(cc(i)-ccc) 220,230,230
  ccc=cc(i)
  it=i
  230 continue
  cost=cc(it)
  cc(it)=1.0d14
  dets1=ixx0(it,1)
  dets2=ixx1(it,1)
  dphase=map(dets1,dets2)
  iss1=isO(1)
  iss2=is1(1)
  idp=map(iss1,iss2)
  c The bit errors and burst errors are counted.
  if(dphase-idp) 270,290,270
  270 symer=symer+1
  if(count-16) 280,280,275
  275 burst=burst+1
  280 count=0
  go to 295
  290 count=count+1
  295 continue
  if(dets1-iss1) 300,305,300
  300 biter=biter+1
  305 if(dets2-iss2) 310,315,310
  310 biter=biter+1
  315 continue
  do 330 i=1,m4,1
  cc(i)=cc(i)-cost
  ip=map(ixx0(i,1),ixx1(i,1))
  if(ip-dphase) 320,330,320
  320 cc(i)=1.0d14
  330 continue
  do 335 j=1,n1,1
  ix0(i,j)=ixx0(it,j+1)
  ix1(i,j)=ixx1(it,j+1)
  335 continue
  c(1)=0.0
  c The final selection procedure:
  ic3=1
  340 continue
  ccc=1.0d10
  do 360 j=1,m4,1
  if(cc(j)-ccc) 350,360,360
  350 ccc=cc(j)
  it=j
  360 continue
if(cc(it).ge.dl) go to 450

ic3=ic3+1
do 390 j=1,n1,1
ix0(ic3,j)=ixx0(it,j+1)
ix1(ic3,j)=ixx1(it,j+1)
390 continue
cc(ic3)=cc(it)
if(ic3.ge.mm) go to 450
cc(it)=1.0d14
go to 340

450 continue
m=ic3
icot=icot+m
icotl=icotl+m2
icot2=icot2+m4

The process continues with the next data symbol s_i+1.

500 continue
xm=float(icot)/float(l)
xm2=float(icotl)/float(l)
xm4=float(icot2)/float(l)
ber=float(biter)/float(no)*0.50
symbol=float(symer)/float(no)
burate=float(burst)/float(no)
write(*,4800)
write(*,4900)
write(*,4910)
write(*,4970) d1
write(*,4980) d2
write(*,4990) mm
write(*,5005) snr
write(*,5006) xm
write(*,5007) xm2
write(*,5008) xm4
write(*,5010) ber
write(*,5015) symbol
write(*,5020) burate
write(*,5025) no
write(*,5030) n
stop

4800 format(2x,///)
4900 format(15x,43hThis is a convolutionally coded 8PSK system,///)
4910 format(15x,25husing system A2 decoding,///)
4970 format(15x,6hd1 is:,f6.3,///)
4980 format(15x,6hd2 is:,f6.3,///)
4990 format(2x,19hthe max storage is:,i5,///)
5005 format(2x,29hthe signal to noise ratio is:,f8.3,///)
5006 format(2x,24hno. of stored vector is:,f8.2,///)
5007 format(2x,27hno. of cost evaluations is:,f8.2,///)
5008 format(2x,36hno. of selected expanded vectors is:,f8.2,///)
5010 format(2x,22hthe bit error rate is:,7x,f9.6,///)
5015 format(2x,25hthe symbol error rate is:,4x,f9.6,///)
5020 format(2x,18hthe burst rate is:,11x,f9.6,///)
5025 format(2x,40hthe errors are counted from detection of,///
i8,3x,21hreceived data symbols,///)
c This function generates the two binary data symbols for each given data symbol value.

integer function data(x,y)
iinteger x,y
data= 1
1000 if(y.eq.1) data=0
if(y.eq.2) data=0
go to 1050
1010 if(y.eq.1) data=0
if(y.eq.2) data=1
go to 1050
1020 if(y.eq.1) data=1
if(y.eq.2) data=0
go to 1050
1030 if(y.eq.1) data=1
if(y.eq.2) data=1
1050 return
end

c This is the linear mapping function.

integer function map(x,y)
iinteger x,y
map= 1
if(x.eq.O.and.y.eq.O) map=O
if(x.eq.O.and.y.eq.1) map=1
if (x.eq.l .and.y.eq.O) map=2
if(x.eq.1.and.y.eq.1) map=3
return
end

c This function carries out modulo-M operation.

integer function range(ix,iy)
5000 if(iy-O) 5010,5050,5020
5010 iy=iy+ix
go to 5000
5020 if(iy-ix) 5050,5030,5030
5030 iy=iy-ix
go to 5020
5050 range=iy
return
end

c This function is defined by equation 5.2.1.

integer function choice(x)
double precision x
if(x.le.0.50) im=4
if(x.gt.0.50.and.x.le.1.00) im=2
if(x.gt.1.00) im=1
choice = im
return
end