



*Campus de la Universidad de Alcalá
Ctra. Madrid-Barcelona, km. 33,6
28805 Alcalá de Henares (Madrid)
Tel: 91 885 88 99
Fax: 91 885 66 99*

DEPARTAMENTO DE TEORÍA DE LA SEÑAL Y COMUNICACIONES

D. ÁNGEL M^a BRAVO SANTOS, Profesor Titular de Universidad del Área de Conocimiento Teoría de la Señal y Comunicaciones de la Universidad Carlos III de Madrid,

CERTIFICO

Que la Tesis “**Sequence Codes**”, presentada por D. Lionel G. A. Woodhead, realizada en el Departamento de Teoría de la Señal y Comunicaciones de la Universidad de Alcalá bajo mi dirección, reúne méritos suficientes para optar al grado de Doctor, por lo que puede procederse a su depósito y lectura.

Leganés, 29 de noviembre de 2012

Fdo. Dr. D. Ángel M^a Bravo Santos



Universidad
de Alcalá

*Campus de la Universidad de Alcalá
Ctra. Madrid-Barcelona, km. 33,6
28805 Alcalá de Henares (Madrid)
Tel: 91 885 88 99
Fax: 91 885 66 99*

DEPARTAMENTO DE TEORÍA DE LA SEÑAL Y COMUNICACIONES

D. Lionel G. A. Woodhead ha realizado en el Departamento de Teoría de la Señal y Comunicaciones y bajo la dirección del Dr. D. Ángel M^a Bravo Santos, la tesis doctoral titulada “**Sequence Codes**”, cumpliéndose todos los requisitos para la tramitación que conduce a su posterior lectura.

Alcalá de Henares, 29 de noviembre de 2012

EL DIRECTOR DEL DEPARTAMENTO

Fdo. Dr. D. Saturnino Maldonado Bascón



ESCUELA POLITÉCNICA SUPERIOR
DPTO. DE TEORÍA DE LA SEÑAL Y
COMUNICACIONES

PROGRAMA DE DOCTORADO EN ARQUITECTURA DE
COMPUTADORES Y TÉCNICAS DE TRATAMIENTO DE
SEÑAL APLICADAS A LAS TELECOMUNICACIONES

Memoria de tesis doctoral

Sequence Codes

Autor: D. Lionel G. A. Woodhead
Director: Dr. D. Ángel M^a Bravo Santos

Acknowledgements

The number of persons responsible for my current position are too many to be included here but I would like to give special thanks to a few. Firstly to my parents who I now know dedicated so much of their lives to me. Then Mr. R. Burgess who brought to me, a rebel, an interest in studying. Next the Post Office and BT (the same company) who took a semi educated young man and completed his education. To the late Mr. Jim Adams who took this man and gave him the chance to use his education. A very important man in this work was Peter (Dr. P. Sweeney) who, so long ago, showed a fascinated student how a simple process could perform miracles of error correction. To Maz (Dr. Mahmoud) who kept a student sane during some rough times. To Dr. C. Menu who supported this student as a friend. To Dr. Ángel Bravo Santos who encouraged me to continue through many difficult periods and who worked to correct this text over such a long period. To Julia (my wife) for her encouragement and patience whilst I was working towards this thesis.

Abstract

The objective of this work was the support of efficient and effective transportation of minimally delayed digital data, primarily via geostationary satellite, between two users. The applicable communication constraints are primarily the media effects related to the orbital distance of approximately 40000 kilometres, to and from the satellite. The media effects are largely related to the free space path loss of approximately 200dB and the satellite link time delay of approximately 0.27 seconds which is barely acceptable for speech.

To achieve this objective a short, efficient, error correction system has been designed. The effort was approached using QPSK modulation as a BPSK modulation sequence code similar, in principle, to Ungerboeck allowing optimization of the signal's Euclidean distance. Due to transmission delay constraints, no iterative techniques were considered. Instead a logically based short phase sequence was generated which, on reception, was operated on by multiple integrated decoders rather than the usual one shot correction techniques of most non iterative systems. The decoder/decodulator consists of an integrated serial/parallel concatenated Viterbi with a new phase sequence decoder (PSD) designed to recover the transmission to a state. This output is further concatenated with a newly developed Metric Analysis Optimise & Control Unit (MAOC) designed to recover to a bit. Separating the multiple integrated decoders to state from the bit correction unit simplifies the integration of the integrated decoder pair. This system is further concatenated with a short Reed Solomon coder. The results achieved are impressive and improvements are considered to be possible with further development.

Index terms: BPSK, QPSK, CPM, TFM, trellis code, tree code, convolutional code, Fano algorithm, coding, Ungerboeck code, Karnaugh map, Gaussian noise, Viterbi decoder, frame, window, trace-back, metric, Euclidian distance, Reed Solomon code, modulation, quantisation, codulation, decodulator, low density parity check code, raptor code, turbo code, iteration

Resumen

El objetivo general de esta Tesis es el de proporcionar los elementos que permitan un transporte eficiente, efectivo y con un mínimo retardo en comunicaciones digitales por satélite, y más específicamente, satélites en órbita geoestacionaria. Las limitaciones más importantes impuestas en las comunicaciones con estos sistemas se deben a la distancia orbital de 40.000 Km., aproximadamente, hacia y desde el satélite. Los efectos más importantes producidos por el medio están relacionados con la atenuación de espacio libre y el retardo en el enlace de 0,27 segundos, aproximadamente, que está en el límite de lo que es aceptable para voz.

Para conseguir este objetivo se ha diseñado un sistema de corrección de errores ligero y eficiente. Se ha utilizado una aproximación al problema similar, en principio, al modelo de Ungerboeck, haciendo uso de secuencias código basadas en modulaciones híbridas QPSK/BPSK que permiten optimizar la distancia mínima euclídea. No se han considerado técnicas iterativas al objeto de satisfacer las restricciones impuestas por el retardo. Por el contrario, se ha diseñado un sistema basado en lógica con el que se generan secuencias cortas de fases de la portadora. En recepción se emplean múltiples decodificadores integrados en vez del único detector generalmente usado en la mayor parte de los sistemas no iterativos. El demodulador/decodulador consiste en un nuevo decodificador de secuencias de fase diseñado para recuperar el estado de transmisión y concatenado con un procesador de Viterbi. La operación de ambos es integrada e intercambian información en paralelo y serie simultáneamente. La salida del conjunto se trata en la unidad MAOC, que se presenta y analiza en esta Tesis, para recuperar la información a nivel de bit. El separar el par de decodificadores de estado integrados de la unidad de corrección de bit simplifica su integración.

El sistema que aquí se presenta se ha concatenado con un código Reed-Solomon corto. Se han conseguido resultados notables y es posible conseguir nuevas mejoras con desarrollos adicionales.

Contents

1	Introduction	1
1.1	The problem	3
1.2	Proposed Solution	4
1.3	Thesis structure	5
2	Background and Related Work	7
2.1	Related Information Theory	8
2.2	Forward Error Correction (FEC)	9
2.2.1	Block Codes	11
2.2.2	Trellis Codes	12
2.2.3	Viterbi Algorithm	13
2.2.4	Modulation Codes	16
2.2.5	Concatenated Codes	20
2.2.6	Good Codes	20
2.3	Potential Modulation Techniques	27
2.3.1	Phase Modulation (PM)	29
2.3.2	Tamed FM (TFM)	33
2.4	Full/Partial Response	34
2.5	Recovery of Digital Signals	35
2.5.1	Envelope detection	35
2.5.2	Coherent (Synchronous) detection	35
2.6	Discussion	35

3	Development Background	37
3.1	Principles	37
3.1.1	The Error Correction Problem Revisited	38
3.1.2	A Potential Solution	39
3.2	Developing Sequence Coding	40
3.2.1	General	41
3.2.2	Pulse Shaping	44
3.2.3	Early Error Coding	44
3.3	New Codulation System	48
3.3.1	Comments on new codulation	50
4	Sequence codes	53
4.1	The Encoder	56
4.1.1	Phase Allocation	58
4.1.2	Implementation of the coder	60
4.2	The Receiver System	61
4.2.1	16 level Phase Quantiser	63
4.3	Open Window	64
4.4	The Decodulator	66
4.4.1	Integrated phase decoders (phase sequence & Viterbi)	67
4.4.2	Phase Sequence Decoder (PSD)	68
4.4.3	Viterbi decoder	72
4.4.4	The MAOC	76
4.4.5	Metric Allocation	77
5	Sequence Code Simulation	85
5.1	Metric Derivation and Unit Optimisation	86
5.2	Implementation of Phase Sequence Decoder	88
5.3	Implementation of the Viterbi Decoder.	89
5.4	MAOC.	92

5.5	Worked Example	96
5.6	Error Probability Bound	105
5.6.1	Simulation of Bound	107
5.7	Results	108
6	Conclusion	111
7	Thesis Contributions	113
8	Future Work	117
	Bibliography	120

List of Figures

2.1	A basic radio transmission system	7
2.2	Code classes	10
2.3	Non systematic, half rate convolutional encoder	14
2.4	Trellis of convolutional coder: Figure 2.3	15
2.5	Grey versus natural coding	17
2.6	4 state 8-PSK Ungerboeck coder	18
2.7	4 state 8-PSK Ungerboeck trellis	19
2.8	Serial concatenated coding	21
2.9	Unity rate convolutional coder	22
2.10	Turbo coder	22
2.11	Interleaver	23
2.12	Turbo decoder	24
2.13	Tanner graph of table 2.2	26
2.14	BPSK waveform	31
2.15	PSK distance of a unit energy signal	32
2.16	TFM encoder	34
3.1	Binary transmission in Gaussian noise	39
3.2	Spectrum of modulation techniques	40
3.3	Continuous phase	42
3.4	Shaped continuous phase	43
3.5	Root/raised cosine comparison	45
3.6	4 phase encoder and state sequence (Based on TFM principles)	46

3.7	DQPSK data modulator and 8 state sequence. The modulator output $g(t)$ has phase shown in the diagram.	50
3.8	Expanded TFM and DQPSK results	52
4.1	Sequence error correction schematic	54
4.2	4 States, 8 branches to 4 phases	55
4.3	Coder and related 8 state sequence allocation. The modulator output $g(t)$ has the internode phase shown in the state sequence	57
4.4	8 State trellis (broken line is the minimum distance state 0 to state 0 for the 8 state system of Figure 4.3	59
4.5	Receiver system	61
4.6	Received signal structure	62
4.7	The 16 Phase quantizer (Each phase window is $\frac{\pi}{8}$)	63
4.8	16 Phase quantisation constellation based on quantizer C	64
4.9	Open window distance & reception	65
4.10	Phase decoder elements for phase sequence and Viterbi decoders . . .	66
4.11	Relative bit error QPSK, BPSK	67
4.12	Phase sequence decoder	69
4.13	Fano algorithm	70
4.14	Phase trellis of 4 states with translation to 8 states	72
4.15	Viterbi decoder; $\hat{S}t_m$ is the initial proven state from the PSD; μ_{seq} is the associated PSD metric for the state $\hat{S}t_m$; $\hat{S}t_{m_v}$ is the potentially double proven state output of the Viterbi decoder; F_i is the frame number within the Viterbi decoder window with the associated state $S t_j$ sequence within that window	73
4.16	Viterbi decoder basic window structure.	74
4.17	Viterbi decoding sequence tree from a single original state with received phase sequence 0, 0, 0, 0, phase sequence code metric = 0 and linear metric 0 for true phase, 1 adjacent to true phase and 2 opposite	75
4.18	Metric allocation (optimum and sub-optimum - headings are related time references and columns are the states of a PSD sequence. The windows shown are the inter time reference phase windows)	80

4.19	Metric allocation (non-optimum - headings are related time references and columns are the states of a PSD sequence. The windows shown are the inter time reference phase windows)	82
5.1	Results without individual sequence coder elements	89
5.2	Viterbi first frame with expansion to second frame. The received quantised phase is $\frac{7\pi}{16}$ and the amplitude of the corresponding signal $ z_m = 1$	90
5.3	Example of the Viterbi dual routing problem	91
5.4	MAOC logic	94
5.5	Average sequence code error burst per RS symbol (Linear $\frac{E_b}{N_0}$ Scaling)	95
5.6	Signal space	106
5.7	Sequence code comparison with rate, K=4 soft decision Viterbi with short trace-back [46] including bound.	109

List of Tables

2.1	Example related phase to element outputs of Figure 2.6	20
2.2	Simple regular sparse parity check matrix	25
3.1	Bit sequence to relative phase	47
3.2	Input sequences and states	47
3.3	Error correction example	48
3.4	Bit sequence to relative phase	49
3.5	Non optimised phase allocation with sequence effects	51
3.6	Irreducible phase allocation logic for early sequence code	51
4.1	Bit sequence to phase	55
4.2	Phase allocation logic for sequence code	56
4.3	Bit sequence to phase sequence	56
4.4	Irreducible logic of Table 4.2	58
4.5	Alternate irreducible logic	60
4.6	Alternate irreducible logic of Table 4.5	60
4.7	Viterbi decoder frame metric allocation	78
4.8	Phase sequence decoder metric Allocation.	79
5.1	Early PSD and Viterbi metrics allocation.	86
5.2	Enhancement test results	87
5.3	quantiser test results	87
5.4	Initial metric test results.	88
5.5	Final test results Viterbi window lengths 5 and 6.	88

5.6	Viterbi decoder dual routing problem presentation	92
5.7	Viterbi dual routing states	92
5.8	PSD decoder results	96
5.9	Viterbi decoder window 1	98
5.10	Viterbi decoder window 2	99
5.11	Viterbi decoder window 3	100
5.12	Viterbi decoder window 4	101
5.13	Viterbi decoder window 5	102
5.14	Viterbi decoder window 6	103
5.15	Viterbi decoder window 7	104
8.1	Potential 4 bit logic	118

Glossary

ARQ	Automatic Repeat Request
ASK	Amplitude Shift Keying
AWGN	Additive White Gaussian Noise; In this channel mode the only impairment is the linear addition of white noise having a constant spectral density and a Gaussian distribution of amplitude
BCH	Bose, Chaudhuri and Hocquenghem
BER	Bit Error Rate
BPS	bits per second
BPSK	Binary Phase Shift Keying
CORPSK	Correlative Phase Shift Keying
CPFSK	Continuous Phase FSK
CPM	Continuous Phase Modulation
C/N	Carrier to Noise
DQPSK	Differentially Coded QPSK
E_b/N_0	Energy per Bit/Noise power spectral density
Euclidean Distance	The geometric distance between vectors
Fan in	No of inputs to a logic element
Fan out	No of outputs from a logic element
FEC	Forward Error Correction
FM	Frequency Modulation

FSK	Frequency Shift Keying
FFSK	Fast FSK
HPA	High Powered Amplifier
iid	Independent and Identically Distributed
Intermodulation noise	The result of the mixing of two, or more, signals in a non-linear amplifier forming new unwanted signals.
IRA	Irregular Repeat Accumulate.
ISI	Inter Symbol Interference
LDPC	Low Density Parity Check Code
MAOC	Metric Analysis Optimise & Control Unit
Metric	A metric, here, is an empirically derived reliability value related to the spatial distance of a received phase from that expected. Designated as μ .
ML	Maximum Likely-hood Decoder
MPSK	M'ary Phase Shift Keying
MSK	Minimum Shift Keying
Multilevel Signals	The use of an increased alphabet of m symbols such that each symbol can represent more than a single data bit.
Multipath	The simultaneous reception of a number of reflected signals which may interfere with the direct signal
NRZ	Non Return to Zero
OFSK	Orthogonal FSK

OOK	On-Off Keying
OSMSED	One Sided Minimum Squared Euclidian Distance
Out of Band Energy	Energy transmitted outside of bandwidth
PAM	Pulse Amplitude Modulation
PDF	Probability Density Function
PSD	Phase Sequence Decoder
PSK	Phase Shift Keying
QAM	Quadrature Amplitude Modulation
QASK	Quadrature Amplitude Shift Keying
Qphase-TFM	4 phase TFM
QPSK	Quadriphase Shift Keying
RS	Reed Solomon Code
St	State
<i>hatSt</i>	Estimate of State
SNR	Signal Noise Ratio
Spectral Efficiency	Defined as the number of bits per transmitted symbol
SS	Spread Spectrum
State Group	The states forming a single Viterbi decoder frame
TCM	Trellis Coded Modulation.

Systematic Code	Is a code where the original information bits form part of the codeword. A non-systematic code, therefore, does not contain the original information bits.
TFM	Tamed FM
XOR	Exclusive OR Logic Gate

Notation

$W_{\substack{\text{phase window dimension} \\ \text{time period}}}$	Window
β	Phase Deviation
Δf	Modulation Index of an FM signal
ϕ	Transmitted Phase Symbol
ϕ_i	Received Phase Symbol
$\tilde{\phi}_i$	Quantized Received Phase Symbol
$\bar{\phi}_i$	Enhanced Quantised Received Phase Symbol
$\hat{\phi}_i$	Estimate of $\hat{\phi}_i$
μ	Metric
μ_s	Phase Sequence Decoder Metric
μ_{sum_m}	Minimum Summation Metric
μ_{sum}	Summation Metric
μ_v	Viterbi Decoder Metric

Chapter 1

Introduction

Forney made the following points in his book concatenated codes (where concatenation refers to the operation of joining, serially and/or in parallel, two or more systems as an integrated functioning unit);

Within the framework of concatenation it is natural to consider these devices (Channel, Modulator, Demodulator Detector) as additional stages of concatenation and to draw the moral, which should have been plain all along, that coding and modulation ought to be designed together to maximise performance while minimising complexity.

In this document we extend this principle by integrating both coding and modulation into a single entity using an approach similar, though different, to Ungerboeck[67]. It is demonstrated that it is possible to combine error correction coding within a bandwidth efficient modulation. This bandwidth (or spectral) efficiency of M-ary phase shift keying being measured by the bit rate to bandwidth ratio such that [11]:

$$\frac{R}{B} = \log_2 M$$

where:

$M \geq 2$,

R is the bit rate (bit/s),

B the bandwidth (Hz),

M the number of phase points within the unit circle.

This work was initiated to consider problems associated with the transmission of time critical data between users via a geostationary satellite at a distance of over 40000 kilometres from both transmitter and receiver. At this distance the free space

propagation loss is given by[11]:

$$F_{fs} = \frac{4\pi d^2}{\lambda^2}$$

where:

d is the distance in meters between stations,

λ the wavelength in meters.

This represents a loss of $> 200dB$ at Ku band (14/12GHz) requiring considerable technical support.

The loop delay is given by:

$$t_{ud} = \frac{2d}{c}$$

where:

t_{ud} is the combined delay to and from the satellite,

c is the velocity of light 3×10^8 meters per second [11].

It is a basic assumption, in this thesis, that the transmission is digital i.e. the input signal appears in discrete steps rather than the continuous variability characterising analogue signals [33]. It is also assumed that the channel is memory-less and the probability of any symbol error, caused by the transmission medium, is unrelated to any other symbol. The elements considered to require support in this work are the inherent delay and BER (Bit Error Rate).

A relevant restriction is imposed by amplifiers in satellite communications operating in the non linear, near saturation region for greater efficiency. Operating in this region, unless the signal amplitude remains constant, results in transmissions broadening and distortion of the signal spectrum [63]. Because of this present day designers of satellite communication systems are constrained to constant, or near constant, envelope modulation [18, 11]. This work is, therefore, constrained to those elements of a sinusoidal transmission that will pass a non linear device without distortion, i.e. phase and frequency.

The overall service requirement is that required by the end user. Their information quality requirement is normally given in terms of the probability of bit error. To achieve their quality of service requirement it is necessary to obtain: [11]

$$\frac{C}{N_0} = \frac{E_b}{N_0} - G_c + R_d + M_o \text{ dB}$$

where:

$\frac{C}{N_0}$ is the energy per uncoded symbol to the power spectral density of noise to achieve the required BER; dB,

$\frac{E_b}{N_0}$ is the energy per uncoded bit to the noise power spectral density ratio; dB,

G_c the coding gain being dependant on the code type and rate; dB,
 R_d is the log code rate; dB,
 M_o represents the system margins (rain fade etc); dB.

The achievable data rate is dependant on the bandwidth and modulation type. For digital satellite communications modulation is usually a form of digital FSK(Frequency Shift Keying), or MPSK (M_{ary} Phase Shift Keying) outlined in Chapter 2.

In sequence coding, the basis of this thesis, the exact characterization of an errored phase is problematical, however, it is assumed that an error is the result of a separation from a predicted transmitted phase or a possibly previous erroneous recovered state that results in an incorrect solution within the recovered data sequence [72]. Unless otherwise stated, the term sequence shall be assumed to be a chronological sequence as defined in Collins Concise English Dictionary. This definition is quoted as; “The successive order of two or more things”. In addition the last received or transmitted bit is considered the least significant bit. This applies to all diagrams and tables.

1.1 The problem

In data transfers the transmission medium plays a significant role. All practical transmissions are affected, to some degree, by numerous forms of corruption. In satellite systems, however, it is usually possible to assume an AWGN (Additive White Gaussian Noise) channel and this work models the communication medium as such [11]. Different modulation types are adversely affected, to differing degrees, by different characteristics introduced by the medium. The modulation system is, therefore, an important system design element of error control.

Where data transmission speeds and/or spectral efficiency are required multi-level signals are often used [11, 4]. These transmissions, introduced due to bandwidth limitations, tend to reduce the physical distance of signal elements increasing end user errors often requiring bit error correction.

Bit error correction requires some form of redundancy [58]. If the number of transmitted information bits is k and the number of information plus redundancy bits is n , where $n \geq k$, then $n - k$ is the number of redundant bits. If $n > k$ and the information rate and bits per symbol remains unchanged, the bandwidth B is necessarily expanded by the factor $\frac{n}{k}$. Assuming the same transmit power the increased number of symbols, due to error control redundancy, reduces E_s/N_o by $\frac{n}{k}$ and the error correction system, to be beneficial, must achieve a greater gain to offset this reduction. The efficient use of redundancy in error correction systems

becomes an important design criterion for parameter limited digital transmissions.

1.2 Proposed Solution

The current work was initiated to study the problem of error control in an environment where delay and bandwidth are critical elements. During initial studies maximising Euclidean distance was established as a further critical objective and optimization of this parameter became an objective.

Cheap processing power has allowed a new look at coding. Early single shot mathematical techniques have, with Viterbi decoder [69, 23] including trace-back [2, 70], Turbo [75, 9, 14] and LDPC [38, 24, 26] codes, partially given way to analytical solutions. The iterative Turbo [59] and LDPC [66] codes are capable of approaching the Shannon limit but to achieve this they incur considerable decoding delays.

“Sequence codes” use short phase sequences designed for delay critical, near real time systems. Though similar to Ungerboeck [67] codes, sequence codes have significantly differences. Ungerboeck codes are linear codes constructed by set partitioning, logically connecting a bit sequence to a transmitted phase. Sequence codes are non-linear codes constructed by the logically irreducible allocation of phases to states within a sequence currently using Karnaugh mapping. In sequence coding Karnaugh mapping is used to ensure the maximum bit sequence separation of same phase transmissions. This use of Karnaugh mapping in this manner is considered unique.

The coder (codulator) converts a random binary bit sequence, via a state sequence, into a structured sequence of coded modulation symbols. Redundancy is introduced via an expanded phase constellation supported by a state dependant sequential methodology. The result is a coded QPSK sequence transmitting a single bit per symbol having an optimum Euclidean distance at each state transition in each bit period; coding with no bandwidth expansion.

Phases are allocated to overlapping three bit subsets of an input random bit sequence to distinguish sequences of sets $S = (s_0, s_1, \dots, s_{n-1})$ via a logic controlled Quadriphase Shift Keying (QPSK) modulator, Figure 4.2, where $s_i \in \{0, 1\}$ is a member of the random bit sequence. Each of the eight states of the set $St = \{0, 1, 2, 3, 4, 5, 6, 7\}$ are derived from three bit subsets of S as shown in Figure 4.3. St is separated into 4 two state subsets; $St_1 = \{0, 6\}$, $St_2 = \{1, 7\}$, $St_3 = \{2, 4\}$, $St_4 = \{3, 5\}$. Each subset is allocated a phase introducing a dependency into ϕ_i (phase) and St_j , sequences.

A multi-tiered concatenated approach to decoding was developed. First an inner serial/parallel, concatenated unit currently consisting of two integrated phase

decoders both using the same received phase data. This unit allocates, using data obtained from a 16 level phase quantising receiver, to each element of a recovered state sequence a reliability metric μ . This unit is concatenated with an outer Metric Analysis Optimise and Control (MAOC) decoder. This decoder uses the recovered states and associated reliability metrics to correct to bit level. This can result in short bit error bursts, Figure 5.5, and the sequence decoder is further concatenated with a (43, 63), over $GF(64)$ systematic RS outer code making an overall code rate of approximately $\frac{1}{3}$.

The metrics are derived empirically based on the translation, due to medium noise, of the received quantised phase vectors from values transmitted. Metric values are obtained using the simulator to minimise errors by iterating the metric values. The empirically derived metric values are obtained at low Signal to Noise Ratio, SNR, in order to reduce the number of samples required to obtain sufficient errors. Testing at higher SNR suggested the results, though not optimum, would be acceptable throughout the SNR range.

1.3 Thesis structure

Chapter 2 presents a project related review of information theory considering a definition of the quantity of information, channel capacity and coding. It considers modulation techniques where these relate to the current work. Chapter 3 presents the related background to the problem of coding and optimization of the modulation with a brief analysis of why a new coding technique is required. The early work into spectrally efficient modulation and coding which has developed into the sequence code is introduced. Chapter 4 describes the development, structure and elements of the sequence code. This includes the final logic allocation of phase to state and the open window technique. Chapter 5 introduces the simulator and the results achieved in both tabular and graphical form. Chapter 6 offers what is believed to be new and beneficial about this work. Chapter 7 communicates some ideas of further work which it is believed would extend the initial work. Chapter 8 presents some conclusions drawn from the work.

Chapter 2

Background and Related Work

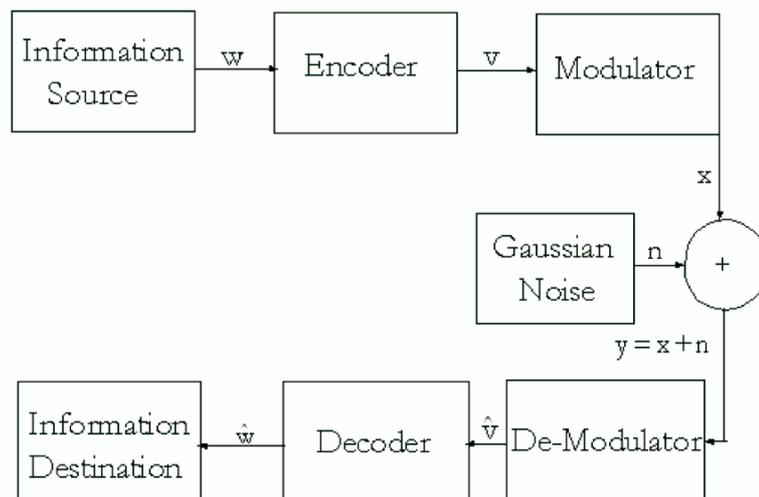


Figure 2.1: A basic radio transmission system

In digital communication systems the problem is to transmit digital data, in a usable form, from a source to a user where the source and user are physically separated [29]. This requires the transmitter to act as a data transformer and link interface known as a modulator and the receiver to act as an inverse data transformer and link interface known as a de-modulator. For each link type the form of modulation is likely to be different to meet the requirements of the medium [10]. Figure 2.1 shows a basic radio transmission system used to present the elements

of this review where:

w = Source information,

v = Coded information,

x = Signal,

n = Noise,

\hat{v} = Recovered estimate of coded data,

\hat{w} = Recovered estimate of source data.

2.1 Related Information Theory

Here is considered, in a semi-axiomatic manner, a definition of quantity of transmitted probabilistic information. If a source produces elements of a set of N symbols where the probabilities of occurrence of each element is p_1, p_2, \dots, p_n . The measure of uncertainty, or entropy, of the source is [46, 56]:

$$H = - \sum_{i=1}^n p_i \log p_i$$

If the probability of each source symbol is the same the entropy (the amount of information) provided by each symbol is maximised [11]. To achieve this ideal sometimes requires a source encoder to minimise (optimise) the original data redundancy within the transmission. A simple example of such an encoder is Run Length Encoding (RLE). If we take a black and white television and consider a single line on a screen where we can expect long runs of white pixels and black pixels letting B represent a black pixels and W represent a white:

WWWWWWWWWWWWBBBBBBBBBBWWWWWWWWWWBBBBBBBBBBBB

Applying the RLE data compression algorithm to the above we get:

12W10B9W11B

A considerable saving

Consider the number of bits that can be transmitted over the channel (Figure 2.1). Being a satellite channel it can be considered Gaussian and use made of the Shannon Hartley theorem to establish the Shannon capacity [11]:

$$C = B \log_2 \left(1 + \frac{S}{N} \right)$$

where:

C is the channel capacity in bits per second,
 B is the bandwidth of the channel in hertz,
 S is the signal power measured in watts,
 N is the noise power over the signal bandwidth measured in watts,
 $\frac{S}{N}$ is the signal to noise as a power ratio.

The proof of this theorem demonstrates that for any transmission rate $R \leq C$ there is a coding scheme that will give an arbitrarily low error probability. For any $R > C$ there is no coding scheme that will achieve reliable transfer of information [16]. Shannon also showed that averaging all possible length n codes the channel error rate is characterized by the probability of, *me*, message error [58].

$$P_{me} \leq e^{-nE(R_I)}$$

where:

E is a function of the information rate known as the random coding error exponent. This coding error exponent, or reliability function, exists for all codes and, within achievable limits, the higher the value the better the code.

R_I is the information transmission rate and providing this does not exceed the channel capacity a positive error exponent is achievable.

Bit error rates for both coded and un-coded communications reduces exponentially with increased $\frac{E_b}{N_0}$ (E_b being the energy per bit and N_0 the single sided power spectral density of the noise) being proportional to the error exponent. Provided a positive error exponent is obtainable increasing the code length decreases the error probability.

2.2 Forward Error Correction (FEC)

Forward error correction operates on a received signal $g(t)$, $0 \leq t \leq \tau$, to produce an estimate $\hat{w}_1, \hat{w}_2, \dots, \hat{w}_M$ of a transmitted sequence w_1, w_2, \dots, w_M . $g(t)$ is corrupted by additive noise hence it is not possible to guarantee to reproduce the originally transmitted sequence. The best that can be achieved is to minimise the probability of errors by using the best FEC coding scheme available [30].

All FEC systems add redundancy, to the originators information, designed to allow a remote decoder to correct errors introduced by corrupting elements within the channel medium. This redundancy results in more data being included in the transmitted message than existed in the originators information. ‘‘Redundancy’’; in this case means that for the original purpose of the originators information the additional data performs no function. One effect of this redundancy is that if a

random data sequence is block coded the code-word address space has 2^k elements (where k is the number of uncoded bits in the block) and not 2^n as the number of bits in the n code block would suggest.

There are two basic classes of FEC Codes, block and trellis codes, Figure 2.2

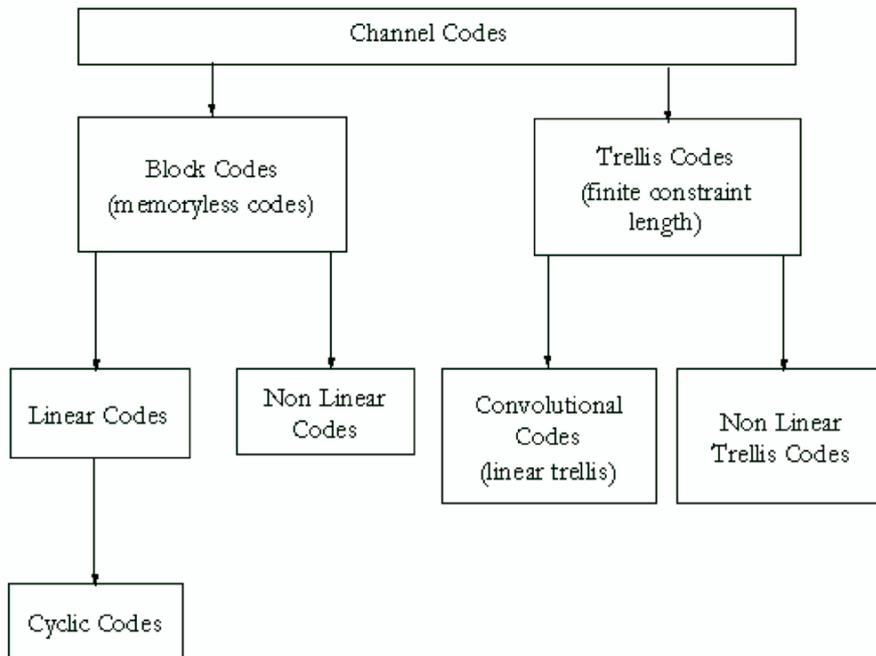


Figure 2.2: Code classes

Most of these error correction coding systems expand, by an encoding algorithm, the incoming k bits of random data, generated by an independent stochastic process, to form a pseudo random n bit data pattern where $n \geq k$. The ratio $\frac{k}{n}$ is referred to as the code rate and $n - k$ is the amount of redundancy. When coding is applied $\frac{k}{n}$ is a fraction but exceptionally, for example when the data is uncoded, $n = k$ and the code rate is unity.

The coded information is modulated onto a carrier and transmitted through a channel medium. After demodulation an inverse transformer, decoder, analyses the non-randomness of the received coded sequence. Due to a priori knowledge of the original encoding technique the original random information may be recovered at the receiver using the recovered syndrome [57, 75]. The syndrome is a unique series of bit values within the recovered data, within the known zero redundant bits, that gives an indication of the actual error. The recovery of the original data has

the proviso that the corruption of the bits, or symbols, caused by the transmission medium does not exceed the capability of the code.

At the receiver the reception of a non code-word indicates an error. The decoder can then either correct or detect errors (or both) to a limit dependant on the Hamming distance of the code [12] :

$$t = \lfloor \frac{d_{min}^H - 1}{2} \rfloor$$

where the function $\lfloor \cdot \rfloor$ is the largest integer less or equal to its argument.

The number of errors detectable is $d_{et} = d_{min}^H - 1$.

In error correction codes the redundancy is derived using the originators information in such a way that errors introduced during the course of the data transmission may, within the limit of the code, be detected and/or corrected. Errors, in this context, refers to the effect, on the transmitted data, of corruption introduced by the channel medium resulting in the demodulator and decoder introducing errors into the originators information.

The inefficiencies of such bounded distance techniques are self evident. The more bits transmitted the more likely an error. The lower effective $\frac{E_b}{N_0}$, consequential on the effect of redundancy, results in higher link error rates. The quality of a digital transmission is normally measured by this link BER.

2.2.1 Block Codes

A binary block code is a set:

$$C \subset V^n$$

where V^n is the set of n-tuples $V^n = \{0, 1\}^n$

Classical block codes associate a message $i \in I$ from a set of messages I to a fixed length codeword $c \in C$ of code C . The elements of the codeword can be taken from an arbitrary alphabet. If the code is binary the codeword is a vector of n bits. For some codes the message can be expressed as a pattern of k symbols. This is the most frequent case. The n bits/symbols of the codeword are modulated for the transmission medium and transmitted. The transmitted modulated symbols are corrupted by the transfer function of the transmission medium and the noise. On receipt, by the receiver, and demodulation each code-word is decoded independently of other code-words and, within the code block, is capable of correcting or detecting a specific number of errors.

Linear Block Codes may be defined as a linear mapping over a suitable alge-

braic system, for example a Galois field, from an input message space to an output data space. An example of a linear block codes is the Reed Solomon (RS) Code being a non-binary subset of BCH codes. The RS code can be specified as RS(n, k) symbols in an extension field GF(2^m) having parameters;

Symbol Length:	m bits
Block Length:	$n = (2^m - 1)$ symbols = $m(2^m - 1)$ bits
No of parity symbols:	$n - k = 2t$ symbols
Minimum Distance:	$d_{min} = 2t + 1$

Since $d_{min} = n - k + 1$ the RS code has the greatest minimum distance possible for any given rate [11].

Cyclic Codes are an important subclass of linear block codes. Most linear block codes in practical telecommunication applications are of this class. If C is a linear code over a finite field A having a coded block length n then if $(s_n, s_{n-1}, \dots, s_1) \in C$ is a codeword a cyclic shift of this codeword, $(s_1, s_n, \dots, s_{n-1}) \in C$ is also a codeword. The cyclic nature of this code gives a solid structure and the codes can be encoded and decoded using simple shift registers with combinational logic elements [49].

2.2.2 Trellis Codes

Trellis codes operate over the entire sequence of a transmission continuously processing the information according to a rule that outputs a coded sequence having more elements than the original information. The encoder breaks up the input information into a usually small number, k_0 , of symbol blocks. Using the current and several proceeding symbol blocks the encoder produces an n_0 symbol of a code sequence. The term trellis code, of Figure 2.2, comes from the fact that the encoding rule for this type of code can best be described by a trellis graph. The principle is the basis of the sequence code except that in this case the information operates directly on the modulated signal.

Linear Trellis Codes, better known as convolutional codes, can be viewed as a discrete time convolution of the message sequence with the impulse response of the encoder. With these codes each k_0 information symbol is transformed into a n_0 data symbols where $\frac{k_0}{n_0}$ is the code rate and $n_0 \geq k_0$. If v is the number of shift register stages in the encoder and K , referred to as the input constraint length, the total number of bits involved in the coding process, where $K = v + k_0$. The output constraint length, n , is the number of output bits for which the effect of any input bit persists, $n = (m + 1)n_0$ where m is the memory order this being the maximum number of shift registers in the path of any output bit [57]. The encoder of Figure 2.3 is considered the best $\frac{1}{2}$ rate, $K = 3$ coder.

Convolutional code can be envisaged as a very (infinitely?) long linear block code where the entire input sequence is encoded into one continuous code word. The encoder, Figure 2.3, is a $K = 3$ (7, 5) convolutional code. In this descriptive form the 7 and 5 are octal numbers related to the polynomials. The upper shift-register connections of Figure 2.3 are 111_2 or $g^{(1)} = x^2 + x + 1$ and the lower 101_2 or $g^{(2)} = x^2 + 1$. The inverse of the number of polynomials is the code rate; in this case a $\frac{1}{2}$ rate code.

Using Figure 2.3 the incoming data stream is clocked through the shift registers one bit at a time. At each clock pulse two code bits are cycled out. As the original data bits are not in the output data this convolutional coder is non-systematic. The convolutional code can be seen as finite state machine where the state corresponds to the value of bits in the encoder. An input bit can take the state to one of two possible other states. These state transitions then form a trellis where each path through the trellis represents a valid state sequence within the encoder output. The trellis of Figure 2.3 is shown in Figure 2.4.

Several decoding algorithms exist for convolutional codes. Probably the best known being the Viterbi algorithm providing maximum likelihood performance. This algorithm is one of the decoders used in sequence coding. The Viterbi decoding algorithm is limited, in practice, to relatively small values of K (normally a maximum of $K = 9$ [15]). For longer constraint lengths there are several sequential decoding algorithms the best known being the Fano [53] algorithm, Figure 4.13, which also operates on a frame by frame basis [57]. Sequential algorithms are not maximum likelihood but benefit from a complexity that increases only slightly with constraint length.

2.2.3 Viterbi Algorithm

In this description we refer to a trellis though the decoding algorithm is a tree that usually decays to a trellis with each transition being known as a frame. The number of frames involved in a decoded sequence is called the Viterbi decoder window. Individual routes through the trellis are called paths. This algorithm has two coding entities; hard decision and soft decision. Hard decision decoder use only the coded data. Soft decision decoders receive data from a channel in such a form that a reliability estimate of a received symbol can be made.

In principle the best way of decoding random errors is to compare a received sequence against all possible code sequences and chose as a solution that which has the minimum number of differences. The Viterbi algorithm [30, 69] attempts to achieve this objective using a trellis representation of a convolutional encoder data (Figure 2.4). From this trellis analysis is derived an accumulated metric (either hard

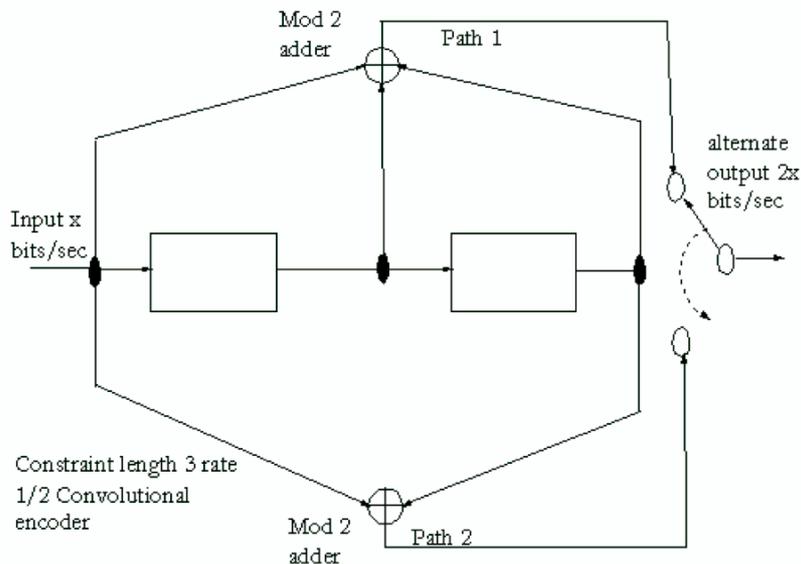


Figure 2.3: Non systematic, half rate convolutional encoder

or soft decision) based on some difference between every path through the trellis, on a frame by frame basis, from the received sequence. Each frame transition is given a metric value dependant on its distance from the received sequence and this metric is summed for every path. This operation usually results in a best summation metric related to an end state which is then considered the solution. The Viterbi decoder, therefore, recovers a maximum likelihood path from all possible paths through the trellis.

This decoding procedure can be considered as updating a table after each frame transition having the following parameters; Figure 2.3.

The number of states, i.e., the number of nodes of a particular frame, is 2^{m_T} , where m_T , is the total number of memory cells of the coder.

The branch metric (Related to the distance between the nearest predicted state/phase transition of the frame and related received state/phase transition of the frame provided in the form of a metric.)

The initial path metrics are the accumulated branch metrics to the current state transition.

The final path metric (called the summation metric) is the accumulated path metric at the end of the Viterbi decoder window) [57]

Decoding begins by initiating a summation metric based on transitions from an original state to a current state. This is continued over a number of received channel

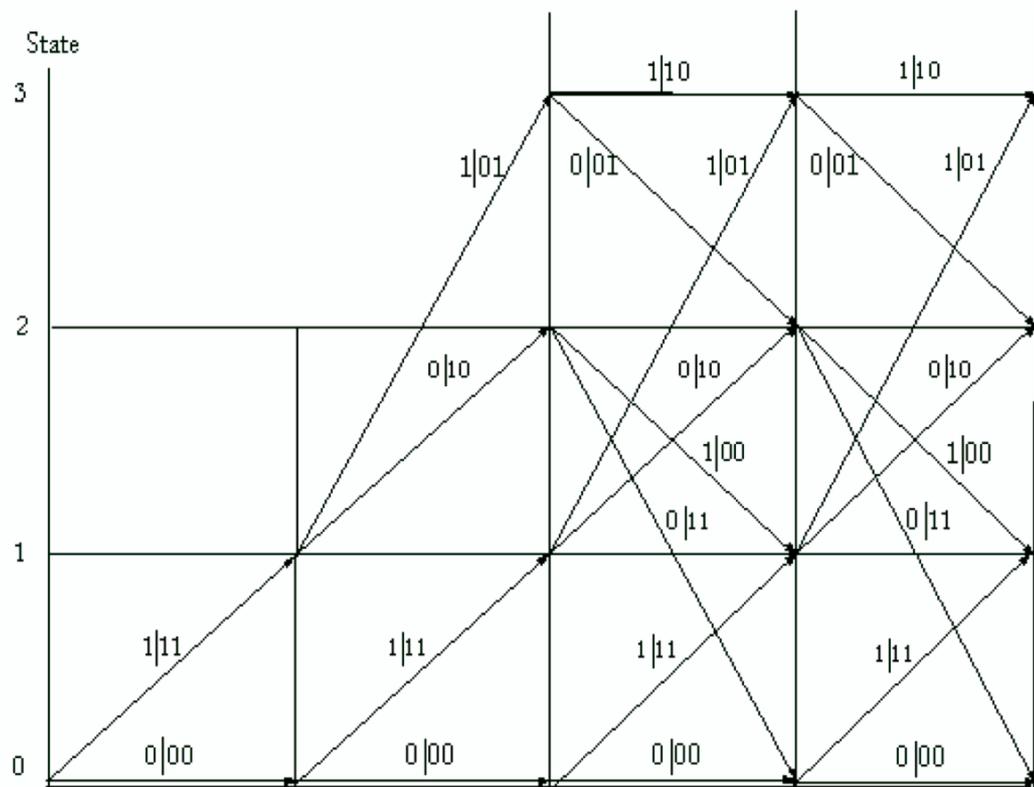


Figure 2.4: Trellis of convolutional coder: Figure 2.3

symbols with the accumulation of state histories preceding the current states at each time instant t . With this information the Viterbi decoder is ready to recreate a maximum likelihood estimate of the original bit sequence. This is achieved by selecting the state with the best (smallest or largest depending on the design) accumulated summation metric and saving that state with its related metric. Then iterate the following step until the beginning of the trellis:

Working backward through the state history table, for the selected state, select a new state which is listed in the state history table as being the predecessor to that state and save the state number of each selected state.

Work forward through the list of selected states saved in the previous steps. Look up what input bit corresponds to a transition from each predecessor state to its successor state. This is the bit that is assumed to have been encoded by the convolutional encoder.

This step is called trace-back. The length of any trace-back is the number of trellis frames processed before the decoder outputs a bit. By storing a frame history of all paths corrections can be made by choosing a best fit path from a greater path-length at the cost of output delay and complexity. Trace-back methods are constantly being upgraded and are the subject of several papers and patents [36, 2].

The complexity of the Viterby decoder is linear in the frame window and exponential in the total number of memory cells. This fact puts a limit to the complexity of the convolutional code. The trace-back needs memory for its implementation. For each state and frame an index must be stored. The total amount of needed memory is again exponential in the number of memory cells and linear in the number of frames.

The Viterbi decoder works on data blocks. It takes in and processes N (the number of encoded symbols) symbols.

A further discussion of the implemented Viterbi algorithm, used for this work, that makes use of a novel trace back technique rather than that described here is presented in section 4.6.2.

2.2.4 Modulation Codes

Though Forney introduced the principle of integrating modulation and coding in his book Concatenated Codes [23] it was Massey in 1974 who introduced the idea of treating coding and modulation as a single integrated entity [44]. Two problems were associated with this idea.

1. The construct of the bit-to-symbol mapping
2. The assignment of the coded bit sequence to the coded symbol sequence

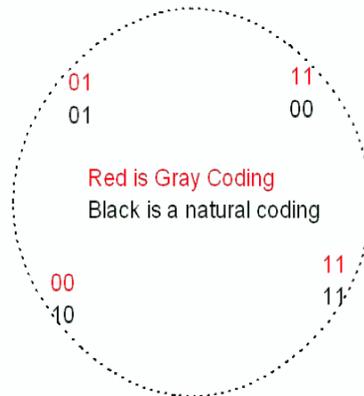


Figure 2.5: Grey versus natural coding

In the 1970-s two basic approaches were proposed to support the design of coded modulation systems;

1. Trellis Coded Modulation (TCM) applies a natural mapping of bits to signals using a technique called set partitioning. (Ungerboeck codes).
2. Multilevel modulation applies a mapping of code-words to bit position through binary partition.

A simple example of item 2 is Gray coding, Figure 2.5, where the QPSK constellation is arranged that only one bit changes per single errored phase shift. Imai and Hirakawa codes [31] expand the constellation to effectively increase Euclidean distance.

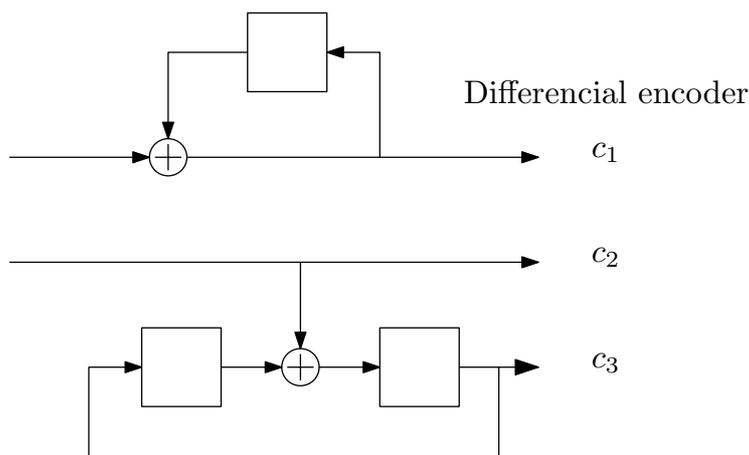
Trellis codes

Trellis-codes are multilevel modulation codes with expanded channel-signal sets. First introduced in 1976 [12, 67] they make use of redundant non-binary modulation using a finite-state encoder. The encoder controls the selection of the modulation signal using the incoming data symbol sequence. It is usually convolutional and generates a coded signal sequence by mapping the incoming bit stream onto the modulation sequence. At the receiver the corrupted signals are normally corrected using a soft decision, maximum likelihood, sequence decoding algorithm such as Viterbi.

These were the first to integrate coding and modulation into a single entity. This integration, and the realization of the dominance of Euclidean distance over

Hamming distance, gave a new direction for error correction coding. Before Ungerboecks 1982 paper [67] there were no combined power and spectrally efficient coding techniques.

Ungerboeck codes



Four State Ungerboeck encoder

Figure 2.6: 4 state 8-PSK Ungerboeck coder

The presentation of Trellis Coded Modulation (TCM) revolutionised the art of coding. In his 1982 paper Ungerboeck constructed trellis codes for Phase Shift Keying (PSK) and Quadrature Amplitude Modulation (QAM) now normally called lattice codes. The work was shown to be extendable to block codes by Soheil I. Sayegh in 1986 [62].

The coding is performed by mapping by set partitioning. An important requirement of this technique is insuring that the parallel transitions of the trellis are mapped into signals separated by a maximum distance in signal space. For the 8-PSK system of Figure 2.6 $d_{free}^2 = 4$, whereas for uncoded QPSK $d_{free}^2 = 2$ assuming unit energy in both cases. Hence this simple TCM scheme has a 3dB gain, at high SNR, over uncoded QPSK.

A simple four-state, 8-PSK code is shown in Figures 2.6 and 2.7 as an example. This version of an encoder [16] operates on two input bits, one differentially encoded to produce a single encoded bit. Differential encoding is required because of the “rotational invariance” of the code so that the correct information sequence may

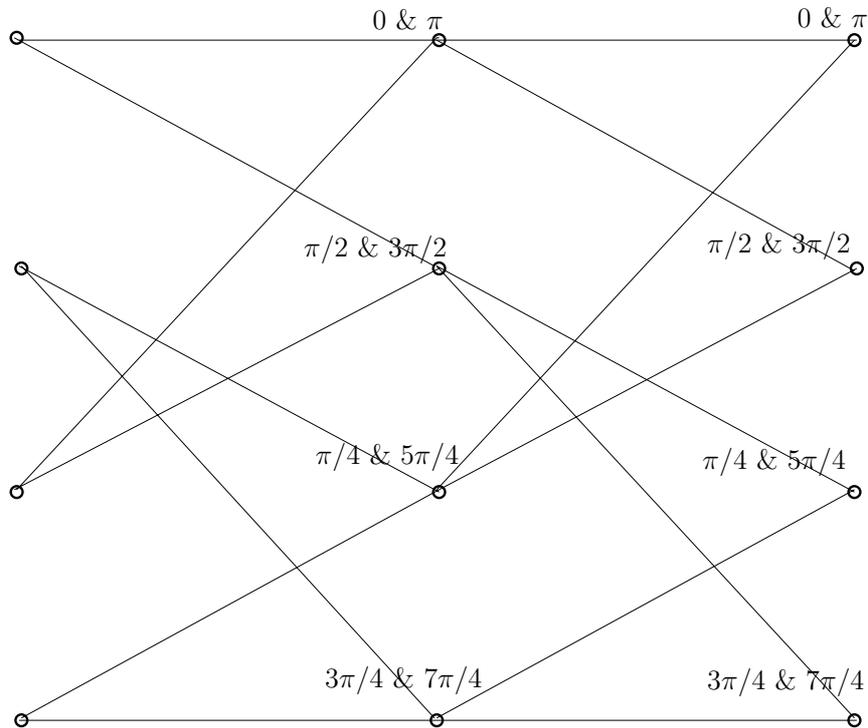


Figure 2.7: 4 state 8-PSK Ungerboeck trellis

be recovered after decoding in the event that the receiver's demodulator locks to the wrong phase. The other input bit is encoded using a systematic convolutional encoder producing two output bits.

Decoding is normally performed using a soft-decision Viterbi decoder operating directly on the received noisy 8-PSK signals. In Figure 2.7 the decoding trellis has four states. Because the code has a single uncoded information bit, the trellis is made up of “parallel transitions,” i.e., parallel paths of length one branch each connecting pairs of states at different time periods. Parallel transitions in the trellis diagram are a characteristic of all TCM schemes containing uncoded information bits. Ungerboeck had realized that for non binary signal constellations, used in TCM, minimum free squared distance in Euclidean space, rather than minimum free Hamming distance d_{free}^2 , was the primary parameter for this type of codes performance.

c1	0	0	0	0
c2	0	0	1	1
c3	0	1	0	1
Phase	$3\pi/4$	$\pi/4$	$\pi/2$	0
c1	1	1	1	1
c2	0	0	1	1
c3	0	1	0	1
Phase	$7\pi/4$	$5\pi/4$	$3\pi/2$	π

Table 2.1: Example related phase to element outputs of Figure 2.6

2.2.5 Concatenated Codes

Limitation, introduced by the error exponent, can be partly overcome using multi-stage, or concatenated codes [23]. Even with concatenated codes the error exponents obtainable are less than theoretically attainable values. Concatenation is, however, a practical method of approaching the predictions of information theory and sequence coding makes use of the concept.

Forney [23] showed that the benefits of long codes could be obtained by serially concatenating shorter more practical codes. Later work, for example Turbo codes, showed how parallel concatenation could be used. These codes create the effect of long codes using a combination of shorter codes. Figure 2.8 shows the basic principle where errors introduced by failures of the inner code are corrected by the outer code. It is usual with this technique to have an inner trellis code with a RS, burst error correcting, outer code.

2.2.6 Good Codes

Good codes are defined as those recently developed families of codes that can claim, in practical terms, to approximate Shannon capacity. Shannon used random codes in his original proof [64]. It has since been demonstrated, non-constructively, by many researchers that good codes of many classes do exist. Unfortunately [39] there exists, for these codes, no practical decoding algorithms.

It has recently been shown that convolution codes, when combined with interleaving in a concatenated scheme, are capable of approaching the Shannon limit [75]. Unfortunately the complexity of the decoding algorithm increases exponentially with constraint length. It is, therefore, currently impractical to build a convolutional decoder capable of approximating Shannon. Block codes can also theoretically ap-

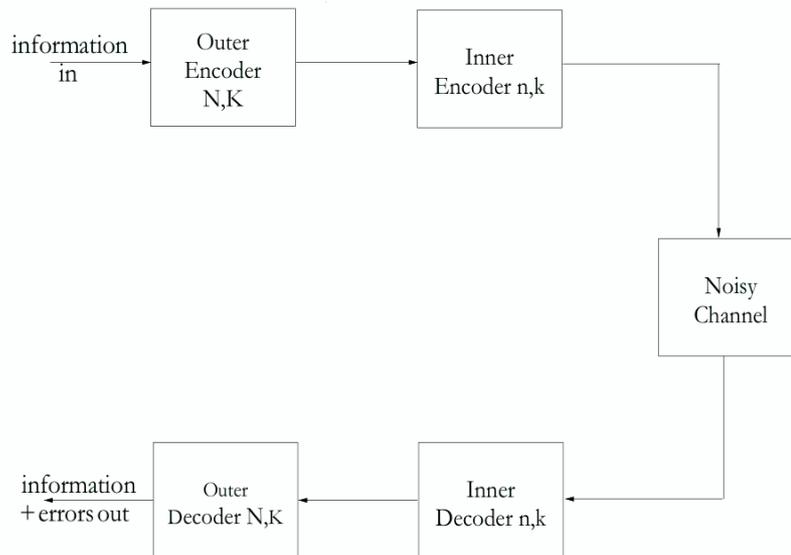


Figure 2.8: Serial concatenated coding

proach Shannon [25, 59] but only at infinite length.

Currently only three iterative families of code (4 if including Irregular Repeat Accumulate [IRA] codes which are examples of LDPC codes) have achieved the distinction of approximating the prediction of Shannon. The first are the parallel-concatenated recursive systematic convolutional codes [14, 9] better known as turbo codes. The second are the Sparse Graph Codes or Low Density Parity Check (LDPC) codes based on Gallager [24] codes. The last family is formed by the polar codes [5], that are currently the only practical codes able to reach capacity.

Turbo Codes

Developed by Berrou, Glavieux and Thitimajshima, [9] were presented in 1992 and eventually recognized as potentially approaching the Shannon limit within practical technology. These systematic codes are basically simple using known coding principles and based on a form of dual use of data.

Two (obviously non systematic) unity rate convolutional coders similar to that shown in Figure 2.9 are used in the encoder Figure 2.10. This encoding rate is normally of limited, if any, practical use but the data provided to one of the coders is modified by an interleaver (Figure 2.11). The interleaver, in this case, performs

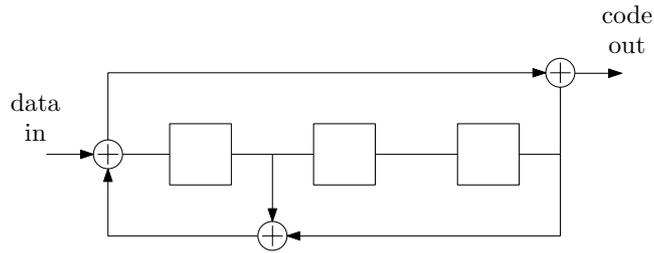


Figure 2.9: Unity rate convolutional coder

the task of manipulating the data into a pseudo random sequence such that the output parity streams of the two coders are different. As the two coders use the same information, which is also transmitted, the basic code is, overall, a $\frac{1}{3}$ rate systematic code.

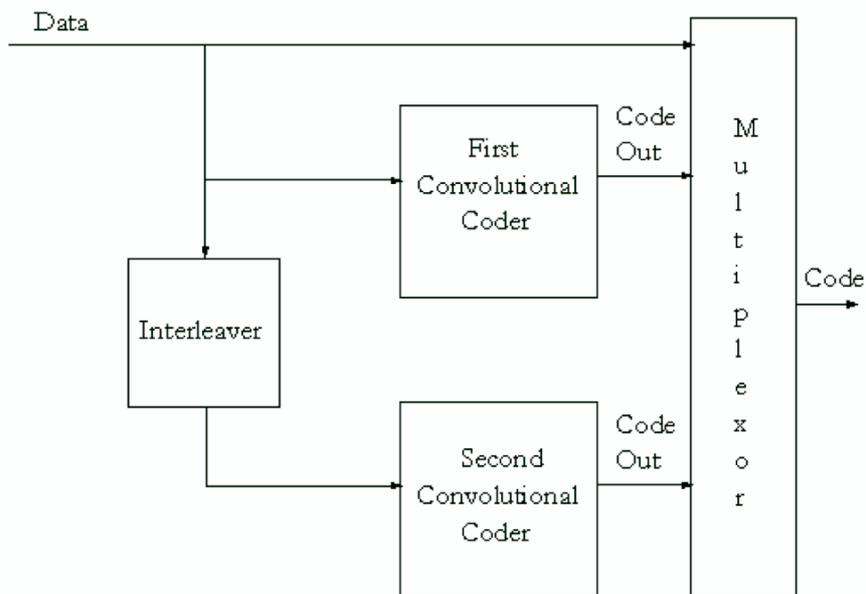


Figure 2.10: Turbo coder

The two parity streams are combined with the original information for transmission via a multiplexer. At the receiver the data is de-multiplexed separating the two parity streams and the data to form the original three information streams.

A simple Turbo decoder is shown as Figure 2.12. The data is passed to one

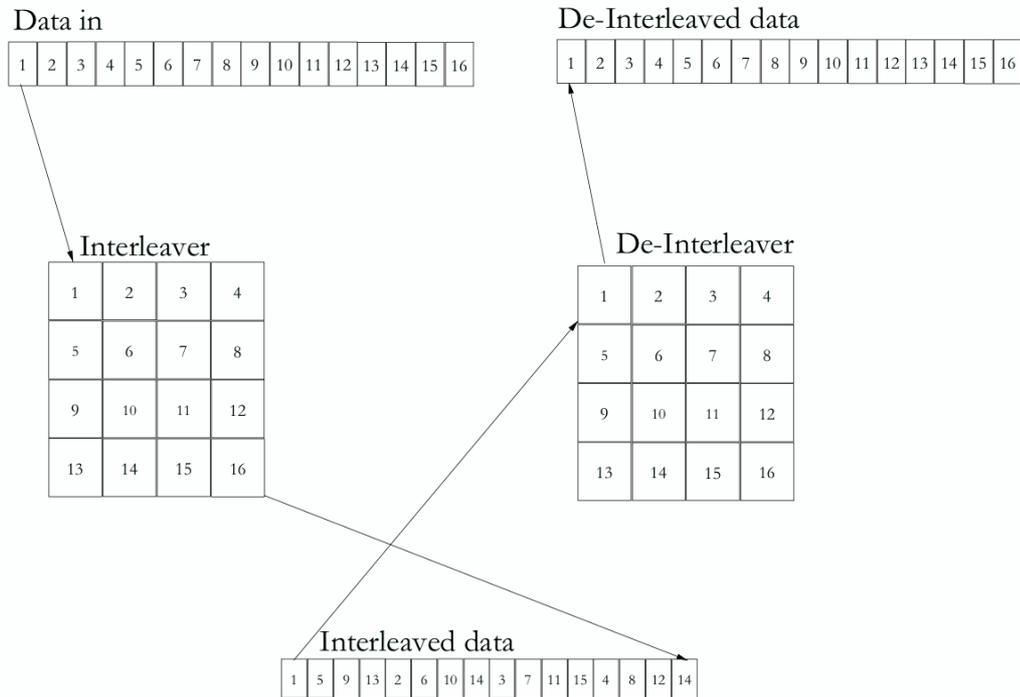


Figure 2.11: Interleaver

decoder via an interleaver acting as an inverse to the transmitter interleaver. To this decoder is passed the interleaved parity sequence (second decoder in Figure 2.12). The data is passed un-de-interleaved to the first decoder using the un-interleaved parity.

The parity from both decoders is held in a buffer and operated upon using the sum product algorithm, also used in LDPC codes, for several iterations. The technique is efficient and can achieve an error correction capability close to the Shannon limit but latency is inevitable due to interleaving and iteration delays. The coder described above would have to function several times faster than line speed to allow iterative decoding during the period of the single de-multiplexer buffer filling. This problem can be nullified by using a series of double decoders each having their own buffer.

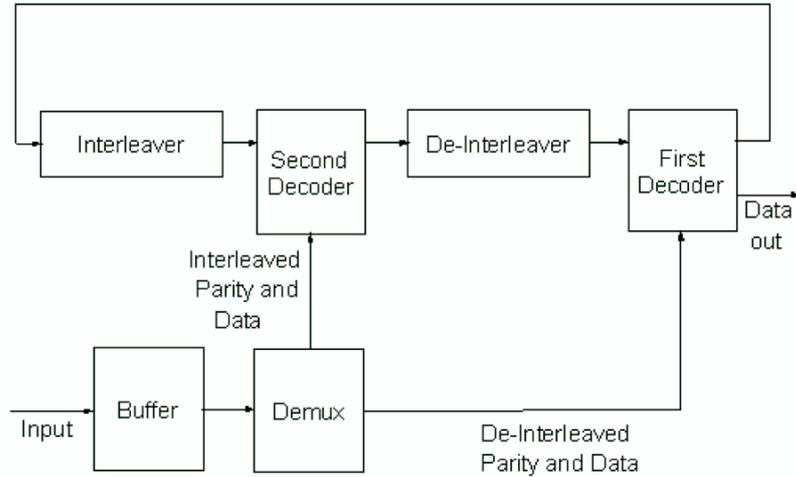


Figure 2.12: Turbo decoder

Low Density Parity Check (LDPC) codes

First presented by Robert Gallager in his 1962 [24] and 1963 MIT research paper [26]. They were forgotten for some time due to their computational requirements and the belief that concatenated RS/convolutional codes were sufficient. They were rediscovered in published work by two groups Richardson [60, 61] and Urbanke, MacKay and Neal in 1998 [38] though there is considerable evidence that Dr Mackay had been working on developing these codes considerably before the publication of this work. These codes are linear block codes having a sparse parity check matrix H , table 2.2.

The Tanner graph, a bipartite factor graph, of this matrix for a non systematic code is shown as Figure 2.13. The graph consist of two elements: circles that represent the variables, and factors than represent the functions: the constraints indicated by the equation: $H\mathbf{v}^T = \mathbf{0}$. Each row of H imposes a constraint to a set of variables, according to that equation and these variables are linked by this functional relation. Therefore in the Tanner graph these variables are connected to the same factor. If these codes are subject to a regularity constraint they have fixed weights for all rows W_r and columns W_c . The allocation of such parity bits can be set at random subject to these constraints. Though this presentation only describes codes having such regularity constraints non regular codes exist that are recognised

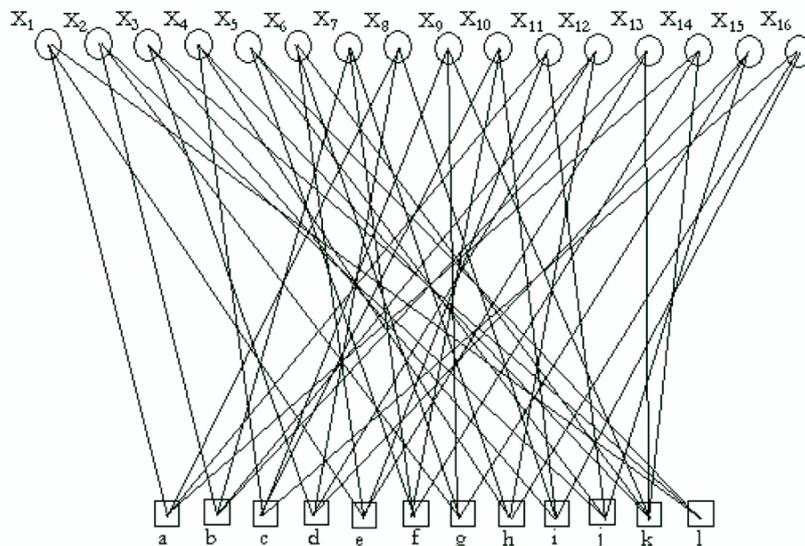


Figure 2.13: Tanner graph of table 2.2

the transmitter is analogous to a fountain of water that continuously produces a stream of water. It is from this analogy that the term Fountain Codes is derived. This coding system produces, for a set of k input symbols (x_1, \dots, x_k) a potentially limitless stream of output symbols $(z_1, z_2, \dots, \infty)$; The input and output vectors can be binary vectors of arbitrary length. Each output symbol is the arbitrary modulo x sum of an arbitrary subset of the input symbols. The information regarding the structure of the arbitrary subset is made available to the receiver. Work by Palanki and Yedidia showed, however, that although these codes had fairly good performance in the waterfall (initial rolloff) regime they suffer from "error floors" [51] where the waterfall region levels out to a limit.

A combination of a high rate LDPC code and an inner Fountain code is known as a raptor code. These require only a small number of "Exclusive OR" (XOR) gate operations per generated symbol for both encoding and decoding and do not suffer from the error floors initially found in Fountain codes [51]. The encoder operates with the k input bits being encoded into n bits which forms the outer LDPC codeword. This LDPC codeword is then encoded into a potentially infinite number of bits of a rateless fountain code. The decoder obtains a sufficient number of received symbols to construct a Tanner graph made up of the inner fountain code and the outer LDPC code. These codes have proven to be a considerable improvement on fountain codes and to operate successfully in a variety of channels

having demonstrated a waterfall region near the Shannon capacity.

IRA (Irregular Repeat Accumulate) codes introduced in 2000 by Khandekar and McEliece [34]. These codes can be represented by Tanner graphs having arbitrary connections between nodes of different degree being defined by the degree distribution of the nodes. They are decoded by a combination of the sum product and maximum a-posterior algorithms.

2.3 Potential Modulation Techniques

In digital satellite systems the modulator acts as a link interface and transforms an input alphabet $W = \{w_1, w_2, \dots, w_m\}$ to an output alphabet $X = \{x_1, x_2, \dots, x_n\}$. Each member of the transmitted alphabet contains information in the form of an analogue symbol.

All radio modulation systems transpose a base-band signal (analogue or digital) to some higher part of the frequency spectrum [11]. The transmitted sine-wave carrier is represented by;

$$y(t) = A \sin(2\pi f_c t + \theta_c)$$

where:

$y(t)$ = is the instantaneous value of the carrier voltage at time t

A = is the maximum amplitude of the carrier voltage

f_c = the carrier frequency

θ_c = the carrier phase

The modulation process involves a change, related to the characteristics of an input, of one, or more, of the three parameters of a carrier wave [33]:

1 Amplitude

2 Frequency

3 Phase

In satellite systems using non-linear HPAs the transmission is limited to the modulation of those constant amplitude elements of a transmission i.e. phase and/or frequency.

Frequency Modulation (FM)

FM represents information as variations of the instantaneous frequency of a carrier. In analogue systems the carrier frequency f_c is varied in direct relationship to the

amplitude changes of an input signal. Digital systems (described later under FSK) represent the data by shifting the carrier frequency among a set of discrete values.

The instantaneous frequency $f(t)$ of the frequency modulated wave is given by;

$$f(t) = f_c(1 + kV_m \cos \omega_m t)$$

where:

f_c = Unmodulated carrier frequency

k = Proportional constant

$kV_m \cos \omega_m t$ = The instantaneous modulating voltage

This can be rewritten as;

$$f(t) = f_c(1 + \Delta f \cos \omega_m t)$$

where the quantity Δf is the frequency deviation being the maximum deviation of the instantaneous frequency from the carrier frequency [29, 33] being proportional to the amplitude of the modulating signal and independent of its frequency.

A characteristic of FM modulation is that the phase of the carrier is changed by the information induced changes in frequency. Frequency modulation can therefore be used to smooth phase transitions over time reducing the instantaneous rate of change of the phase transitions.

Frequency Shift Keying (FSK) [3]

It is a digital form of FM which is most effective when designed as a special case of orthogonal signalling. In this approach only orthogonal groups of frequencies are used, usually individually, to represent a symbol. A binary symbol could be transmitted using two frequencies forming a set $F = \{f_1, f_2\}$. For a duo-binary symbol 4 frequencies are required making a set $F = \{f_1, f_2, f_3, f_4\}$. The number of frequencies required to transmit a symbol is 2^m where m is the number of binary bits per symbol i.e. a 5 bit symbol requires 32 frequencies. The Bandwidth expansion factor for a digital modulation schemes is given by $B_e = \frac{W}{R_b}$ where W is the overall bandwidth of a set of modulation waveforms denoted in Hertz and R_b is the information rate of the modulation in Bits/Second (BPS). For M-ary FSK with tones spaced at $1/2T$ Hz, the bandwidth is approximately $\frac{M}{2T}$ and T is the FSK pulse duration then $R_b = \frac{m}{T}$ where $2^m = M$.

For M-ary FSK signalling $B_e = \frac{M}{2 \log_2 M}$. As $M \rightarrow \infty$ $B \rightarrow \infty$. Although fast Fourier transforms and specialist high performance processors are making these techniques viable an additional disadvantage of large M-ary FSK is that the reception of large numbers of orthogonal signals still requires a great deal of complex circuitry [46]. Rectangular shaping can be described [29]:-

$$\begin{aligned} f_{1c}(t) &= A \cos \omega_1 t & -\frac{T}{2} \leq t \leq \frac{T}{2} \\ f_{2c}(t) &= A \cos \omega_2 t & -\frac{T}{2} \leq t \leq \frac{T}{2} \end{aligned}$$

where:

$$A = \sqrt{2 \frac{E_s}{T}}$$

E_s is the energy per symbol.

If a 1 corresponds to frequency f_1 then a 0 corresponds to f_2 . Alternatively it can be described by representing the FSK frequencies as $f_1 = f_c + \Delta f$ and $f_2 = f_c - \Delta f$. Then:-

$$f_c(t) = A \cos(\omega \pm \Delta\omega) \quad -\frac{T}{2} \leq t \leq \frac{T}{2}$$

The frequency deviates $\pm \Delta f$ [47, 29] about f_c .

Because coherent detection of FSK requires the recovery of reference carriers for each respective frequency the technique is complex and coherent detection of FSK is inherently inferior to PSK [48].

FSK is usually demodulated non-coherently. In non-coherent detection the detector operates without knowledge of the received carrier phase [46]. The frequencies must then be separated by integer multiples of the data-rate. If the two frequencies are multiples of the reciprocal of the data-rate T and synchronized in phase the FSK wave becomes continuous. This technique called Continuous Phase FSK (CPFSK). A version of CPFSK, MSK, can be seen as the superposition of two periodic On-Off Keyed (OOK) signals each delayed by T seconds relative to the other. This is advantageous as phase continuity results in a frequency spectrum that rolls off more rapidly than the spectra of other forms of modulation [46].

2.3.1 Phase Modulation (PM)

In PM the information is transmitted in variations of the carrier phase.

$$S(t) = A \sin(2\pi f_c t + m(t) + \varphi_c)$$

where:

φ_c the phase of the carrier

Phase modulation is commonly used in satellite digital communications because of its constant amplitude and robust and simple circuitry. The modulation of phase has given rise to a family of techniques used in modern digital communications.

Binary Phase Shift Keying (BPSK)

This is a coherently demodulated technique where the carrier phase is stepped π radians. The signal is given by:

$$f_c(t) = \pm A \cos 2\pi f_c t \quad -\frac{T}{2} \leq t \leq \frac{T}{2}$$

If rectangular shaping is used the phase, controlled by the input information, is stepped with the polarity of the rectangular input pulse controlled by the input information. The PSK signal is essentially a polar Non Return to Zero (NRZ) binary stream translated up in frequency.

The average probability of error for BPSK is given by:

$$P_e = 0.5 \operatorname{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right)$$

where:

$$\operatorname{erfc}(x) = \frac{2}{\pi} \int_x^\infty e^{-t^2} dx$$

Since there is only one bit per symbol P_e is also the symbol error probability. As the channel may introduce an arbitrary phase shift the transmission is often differentially encoded.

Figure 2.14 shows a BPSK waveform, assuming a square wave transition waveform. The discontinuous phase transitions that cause out of band noise are clearly shown. There are several variants of BPSK/QPSK to overcome the problems where the phase transitions are smoothed using waveform shaping [29].

Quadrphase PSK (QPSK)

With this modulation it is possible to transmit two binary bits per symbol having an address space of four binary pairs (00, 01, 10, 11) used to switch a carrier between four phases. This signal, an obvious extension of BPSK, can be given by:-

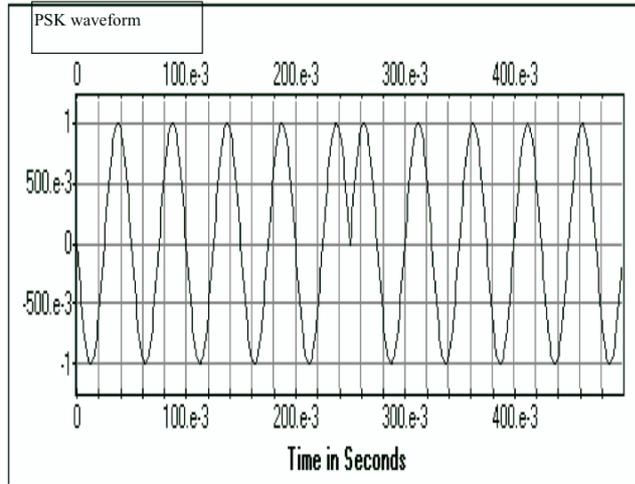


Figure 2.14: BPSK waveform

$$s_i(t) = A \cos(2\pi f_c t + \theta_i) \quad i = 1, 2, 3, 4 \quad -\frac{T}{2} \leq t \leq \frac{T}{2}$$

Rectangular shaping is assumed.

The phases are evenly spaced. In the case of conventional PSK systems both BPSK and QPSK exhibit the same error rate performance for the same E_b/N_0 . However, for the same data-rate QPSK requires only half the bandwidth.

Conventional QPSK requires a header to recover from the potential phase ambiguity of demodulation. By Differentially encoding QPSK (DQPSK), the phase can be resolved without this syndrome pattern [48].

Consider the QPSK signal constellation of Figure 2.15. The Euclidean distance of unit energy symbols allocated between 01 and 11 is 2. The distance 01 to 10 is $\sqrt{2}$. The bit error probability of QPSK modulation can be improved with Gray coding, named after Frank Grey of Bell Labs, (coloured red in Figure 2.5) the adjacent phase elements represent a single bit difference hence a single bit error on a symbol error transition. For this reason the bit error probability $P_b \approx P_s/k$, with P_s the symbol error probability and k the number of bits per symbol. The technique is also known as reflected binary code as the first half is the inversion of the second half; i.e. for 2 bits 00, 01, and 11, 10 shown graphically in Figure 2.5. This technique provides support for standard algebraic bit coding but not modulation codes.

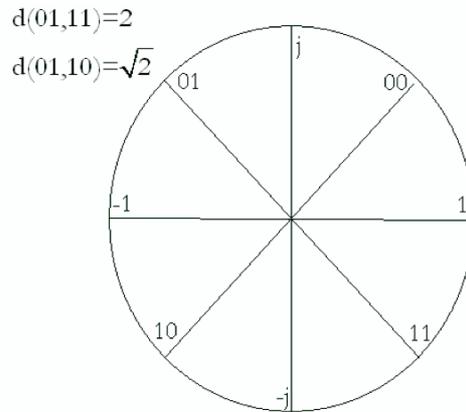


Figure 2.15: PSK distance of a unit energy signal

Continuous Phase Modulation (CPM)

With its constant envelope, continuous phase and compact spectrum is an attractive modulation technique for both satellite and mobile radio applications [13]. The CPM signal is described by:

$$s(t) = \cos(2\pi f_c t + \phi(t))$$

where:

$\phi(t)$ =phase variation

The phase is related to the information input to the modulator which can be considered as a sequence, a_k of K-ary digital symbols [6].

With symbol duration of T_s the input signal can be given as a function of time;

$$a(t) = \sum_k a_k q(t - KT_s)$$

where:

$q(t)$ is a phase smoothing pulse of length T_s .

This pulse can take many forms such as half sine wave, raised cosine etc. [6, 7]. CPM schemes achieve lower transmission bandwidths than ordinary PSK or FSK formats. The reduced transmission bandwidth is achieved by the improved spectral roll-off associated with continuous phase formats so that moderate band-limiting causes less distortion than more traditional schemes [54].

Minimum Shift Keying (MSK)

This is also known as Fast Frequency Shift Keying, (FFSK) where fast refers to the fact that more bits are transmitted in a channel bandwidth than with BPSK. Minimum refers to the minimum modulation index ($h=0.5$) for orthogonal signalling [11]. MSK is a special case of CPFSK with an ideal bit error rate performance equivalent to PSK [45, 48]. This coherently demodulated modulation technique can be considered either PSK or FSK [48]. The frequency deviation is given by:-

$$\Delta f = \frac{f_1 - f_2}{2} = \frac{1}{2T_b}$$

T_b being the bit duration. This illustrates the correlation between transmitted frequencies and bit rate that is required for the phase trajectories to be related.

Without sinusoidal smoothing the data changes linearly over the signalling interval and the phase changes are non-linear to the same point reducing frequency transitions. Either technique result in a signal that corresponds to the data streams differing in at least the first bit being separated in phase over at least two intervals. MSK receivers achieve BER performances comparable to QPSK schemes within an equivalent bandwidths [54].

An FSK signal is produced by the transmission of a sinusoid the frequency being shifted between two frequencies:-

$$f_1 = f_c - \Delta f = f_c - \frac{1}{4T_b}$$

$$f_2 = f_c + \Delta f = f_c + \frac{1}{4T_b}$$

Optimal performance for MSK is achieved by exploiting the fact that it can be constructed as two binary phase modulated pulse streams in phase quadrature. Each of the channels transporting shaped pulses of duration $2T$ with the two channels being offset by T seconds. The modulator output has either a positive or negative linear phase change relative to the carrier [19]. The signal is optimally demodulated using coherent quadrature channel methods (as O-QPSK) rather than FSK.

2.3.2 Tamed FM (TFM)

Developed by Frank de Jager and Cornelius B. Dekker [4, 32] provided improved spectral efficiency using coded partial response, has one of the narrowest bandwidths of the true constant envelope class and tolerates additive noise well [4, 8]. Correlation creates the desired variation of phase shift where the phase transitions are

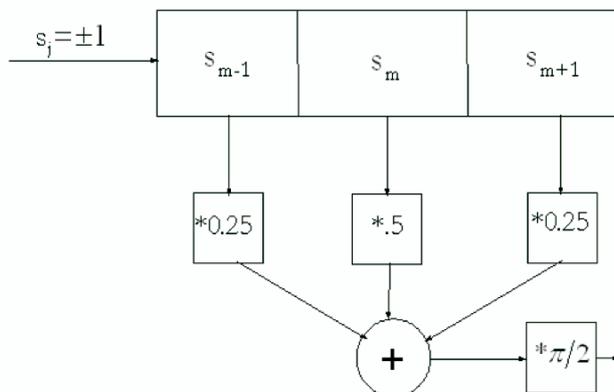


Figure 2.16: TFM encoder

both smoothed and correlated which results in the improved spectrum and power efficiency whilst maintaining a constant transmission envelope. This type of modulation is known as Correlative Phase Modulation (CORPSK).

The encoder shown in Figure 2.16, though not a classical convolutional encoder, has a similar structure to a non-systematic, unity rate, output constraint length 3, convolution encoder with a phase constellation of 8 points and 5 possible shift levels with phase intervals of $\frac{\pi}{4}$ and $\frac{\pi}{2}$ radians. The natural symbol error structure is correlated couplets due to the continuous phase demodulator considering symbol phase difference.

2.4 Full/Partial Response

If the transition period is the same as the symbol period T the system is said to be full response. A further development which extends the period of the input waveform over more than T is called partial response. A receiver that makes a decision based on two or more symbol intervals makes a stronger decision than one that makes symbol by symbol decisions. The advantage of the partial response technique is that by rounding and extending the input waveform $p(t)$ whilst maintaining its area there is a reduction in the bandwidth of the modulating waveform and the modulated carrier. There is, in addition, an increase in the correlation between neighbouring signal element waveforms. When maximum-likelihood detection is used this dependence can lead to an improved tolerance to noise.

2.5 Recovery of Digital Signals

Although FSK systems perform well in noisy environments PSK (ignoring inter-symbol interference and other distortions) is considered the optimal system for digital signalling in the presence of additive noise. There are two normally used demodulation techniques:-

2.5.1 Envelope detection

This avoids the synchronization problem. The high frequency signal is passed through a non linear device and a low pass filter. The detector recovers the envelope of the incoming high frequency wave.

2.5.2 Coherent (Synchronous) detection

Optimum detection of binary PSK signals is performed with a matched filter followed by a sampler. The filter is matched to either of the antipodal signals $s_0(t)$ and $s_1(t)$, being sinusoids of the same frequency having a phase separation of π . Using this technique assuming perfect phase coherence the probability of error is:-

$$P_e = 0.5 \operatorname{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right)$$

where:

$$\operatorname{erfc}(x) = \frac{2}{\pi} \int_x^\infty e^{-t^2} dx$$

2.6 Discussion

In this chapter is presented the systems, both modulation and coding, that were considered in the development of the sequence code. The TFM encoder was useful in the development of the simple sequence code encoder. Though similar to a standard convolutional encoder it uses no feedback loop instead, via a few simple logic gates, it has direct access to the modulator.

The Ungerboeck code was useful as it provided the original ideas for coding the modulation [73]. Ungerboeck uses a system of set partitioning which in short codes appears somewhat complex. The use of logic optimisation appeared to simplify the allocation of the phase to data and to optimise the Euclidean distance over the short code.

The Viterbi algorithm appeared to be the best solution for correcting the derived coding sequence. It was found that this algorithm had some problems related to error bursts starting prior to errors. This resulting error bursts are caused by the Viterbi window summation metric and it was considered that if some support could be provided to reduce this problem considerable gains could be achieved.

Chapter 3

Development Background

The most recognised techniques for decoding convolutional (sequence) codes are the iterative Fano algorithm [53] or the Viterbi algorithm. The Fano algorithm is a frame by frame tree searching algorithm. In figure 4.13 (Fano algorithm) x is the branch currently being decoded, T is a metric threshold and $t(x)$ is the current summation metric over a sequence length up to n . If $t(x)$ exceeds T then x is reduced and the threshold increased until a solution is found. The principle of the Fano algorithm is considered excessively complicated and being iterative -increasing delay- is unsuitable for use in current sequence coding.

PSK is the preferred modulation for digital data in low carrier to noise (C/N) environments having the lowest Bit Error Rate (BER), for the same C/N in a Gaussian environment, of any modulation scheme [48]. Early work on spectral optimisation examined the smoothing effect of FM generation of a phase shift using non rectangular pulses in both full and partial response modes. The pulse shaping was designed to reduce and smooth the rate of change of phase improving spectral efficiency. It was considered that improving PSK spectral presentation using FM generated changes of phase would offer practical advantages. In addition the development an efficient modulation coding scheme, including spectral efficiency, should provide additional gains. This work resulted in the development of an integrated modulation and coding technique (codulation) which then became the prime development of the project.

3.1 Principles

Early studies indicated that meeting the above requirements necessitated a simple coding technique based on partial response, the minimising of the phase constellation

to maximise Euclidian distance and the shaping of the data pulse into the modulator to improve spectral efficiency. The shaping of the phase generating pulse proved an early distraction, however, the use of demodulator information, using soft decision, [57, 70] would assist error correction. The technique makes use of the analogue information normally ignored in signal reception. This recovered information can be used by the decoder to assist the recovery of data by up to 2dB Signal to Noise Ratio (SNR) [57, 68] though Zaragoza [75] gives a maximum value of 3dB.

3.1.1 The Error Correction Problem Revisited

A binary transmission system is shown in Figure 3.1. The information source outputs a block of message symbols taken from the alphabet \mathcal{W} that are the input to the coder

$$\mathbf{w} = (w_0, w_1, \dots, w_{k-1}), w_i \in \mathcal{W}$$

The encoder forms the binary codeword

$$\mathbf{v} = (v_0, v_1, \dots, v_{n-1}), v_i \in \{1, -1\}$$

that is the input to the modulator. Based on the input codeword the modulator chooses the appropriate signals from a finite set and forms the modulated transmitted signal $g(t)$. The input to the demodulator is the transmitted signal plus noise. The attenuation of the channel, though not specifically stated in Figure 3.1, is implicitly accounted for in the energy of $r(t)$. The demodulator filters the received signal $r(t)$ with filters adapted to the transmitted signals. The sampled output of the filters forms the sequence of samples

$$\bar{v}_i = \tilde{u}_i + n_i, i = 0, \dots, n - 1,$$

where:

$$n_i \sim \mathcal{N}(0, N_0/2).$$

The ML detector finds in the code C the codeword $\hat{\mathbf{u}}$ that makes the received pattern $\bar{\mathbf{v}}$ the most probable,

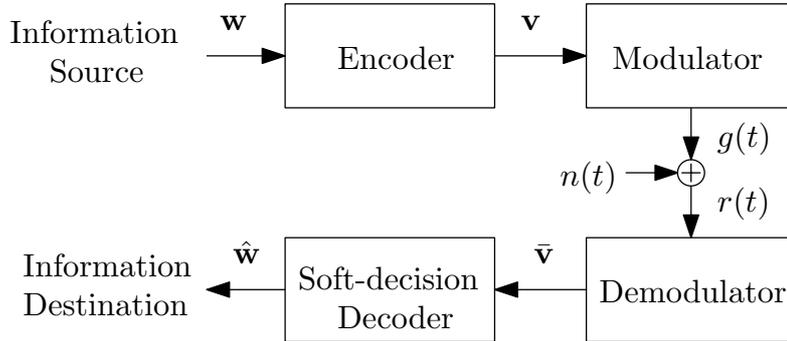


Figure 3.1: Binary transmission in Gaussian noise

$$\begin{aligned} \hat{\mathbf{u}} &= \arg \max_{\mathbf{u} \in C} P(\bar{\mathbf{v}} | \mathbf{u}) \\ &= \arg \max_{\mathbf{u} \in C} \frac{1}{\sqrt{\pi N_0}} e^{-\frac{1}{N_0} \sum_{i=0}^{n-1} (\bar{v}_i - u_i)^2} \end{aligned} \quad (3.1)$$

$$= \arg \min_{\mathbf{u} \in C} \sum_{i=0}^{n-1} |\bar{v}_i - u_i|^2 \quad (3.2)$$

The ML detector is a minimum distance detector.

The demodulator supplies information, related to the reliability of the received analogue symbols, to the decoder. This information provides an indication of the relative strength or weakness of the received analogue symbols giving an indication of a potential error and its position.

3.1.2 A Potential Solution

Trellis Coded Modulation (TCM) techniques have been examined by Ungerboeck [67, 68] and others [44, 52, 65]. Such techniques usually involve expanded signal sets in excess of four phases considerably reducing the Euclidean distance.

A study of potential short delay error correction coding techniques suitable for satellite communications resulted in the development of a new TCM based analogue-symbol restructuring and error correction coding technique. Coding the analogue signal rather than the data has several advantages including enhanced modulation information recovery. Additionally, in the case of sequence coding, it proved possible to design an integrated demodulator incorporating concatenated decoders using

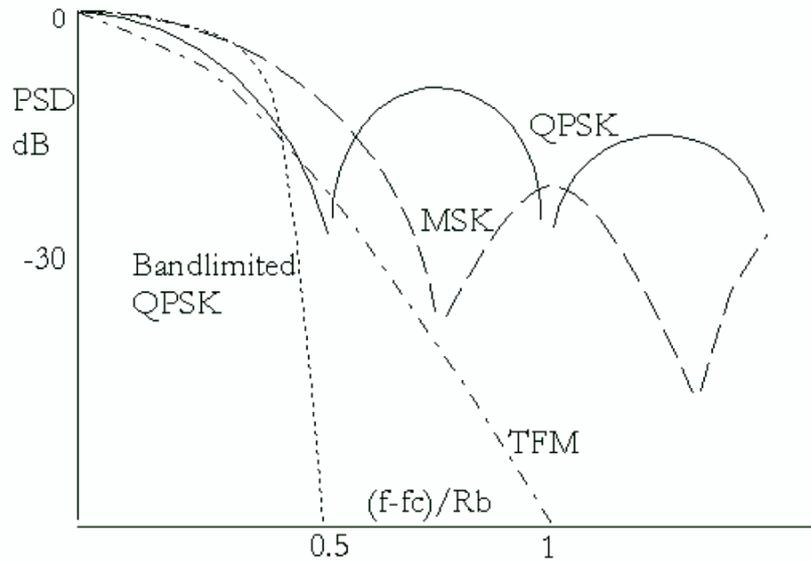


Figure 3.2: Spectrum of modulation techniques

the same quantized received phase symbol information in different ways in mutual support [73, 74].

To meet the requirement for short coding delays the sequence code uses no iteration. The technique introduces a decoding delay of less than ten symbols (plus any satellite delay) implying the potential for use in near real time applications. Sequence codes, due to the possibility of short error bursts, might require a supporting short burst error correcting outer code. For this a (63, 43) RS code was chosen resulting in an additional delay of several hundred symbols.

3.2 Developing Sequence Coding

To develop the sequence code required various studies. During the course of these studies considerable work was performed in the study of the modulation and coding techniques available [72]. An initial study, using the Elanix [17] simulation software, of pulsed FSK modulation demonstrated the bandwidth spreading factors due to the rapid transitions associated with rectangular pulses.

Shaping the pulse, reducing the transition rate, shown in Figures 3.3 and 3.4, reduced spectral spreading. In these figures the integral over the frequency band

presented in the lower graph indicates, by the transition slope, the amount of energy over the frequency band. FSK was considered and provided an improved bandwidth and spectral structure. This was further improved by shaping the information pulse into the FSK modem. Shaping the pulse reduced the out of band energy and became an aspect for consideration. The shaped information pulse should be smooth as the resulting energy spectrum will consist of lower frequency components. The spectrum of some constant envelope modulation techniques is shown in Figure 3.2.

3.2.1 General

The original objective of producing a bandwidth and spectrally efficient satellite communications technique was considered achievable if the coding and modulation could be integrated into a single entity. This proved possible as the natural memory, inherent in some modulation techniques, could be used to develop a coding structure. Bandwidth expansion could be reduced by the introduction of pulse shaping and such a system should be more efficient than data redundant coding schemes.

Use was made of those parameters that could be transmitted via a non-linear device, i.e. phase and frequency, to create a multi-dimensional modulation scheme with a structured multi-level data (partial response) scheme (based on the principle of TFM) using FM to generate a continuous phase QPSK or a Differentially encoded QPSK (DQPSK) signal. The rationale for FM-PM over direct PM was that FM phase transitions could be made continuous rather than stepped. These advantages could be brought at the expense of tight phase synchronization. Some simulation studies, using the Elanix software, of these phase transitional effects were performed using random data into a two frequency FSK modulator. With discontinuous phase the roll-off is relatively poor, Figure 3.3. The slope of the integral of the upper waveform indicates the proportion of the energy is within the useful zone.

Next was simulated the base-band spectral shaping (Raised Cosine) of the continuous phase of Figure 3.3. The result, Figure 3.4, demonstrated a significant improvement in roll-off and a corresponding reduction in out of band spectral content, shown in the integral curve. Almost all the transmitted energy is within the useful zone. This demonstrates that the transitions, phase or frequency, can be suppressed by control of the base-band phase itself or further improved when used in conjunction with pulse shaping. Filtering could perform the same function. Unfortunately, a filtered waveform passed through a non-linear device has a tendency to reinstate those spectral elements filtered out [18, 29].

The use of partial response techniques results in data decisions being less positive during a single element period. Such an approach requires an extended decision methodology, i.e. a maximum likelihood or trellis type technique.

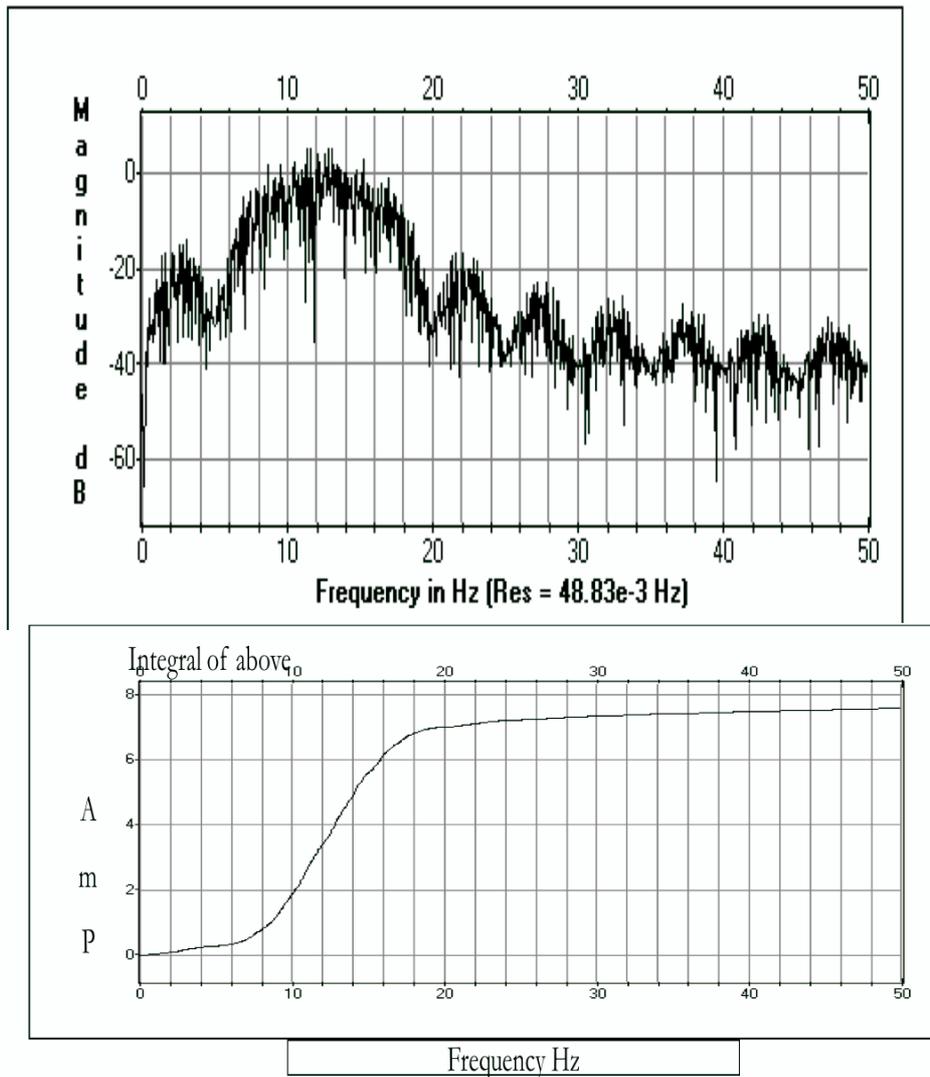


Figure 3.3: Continuous phase

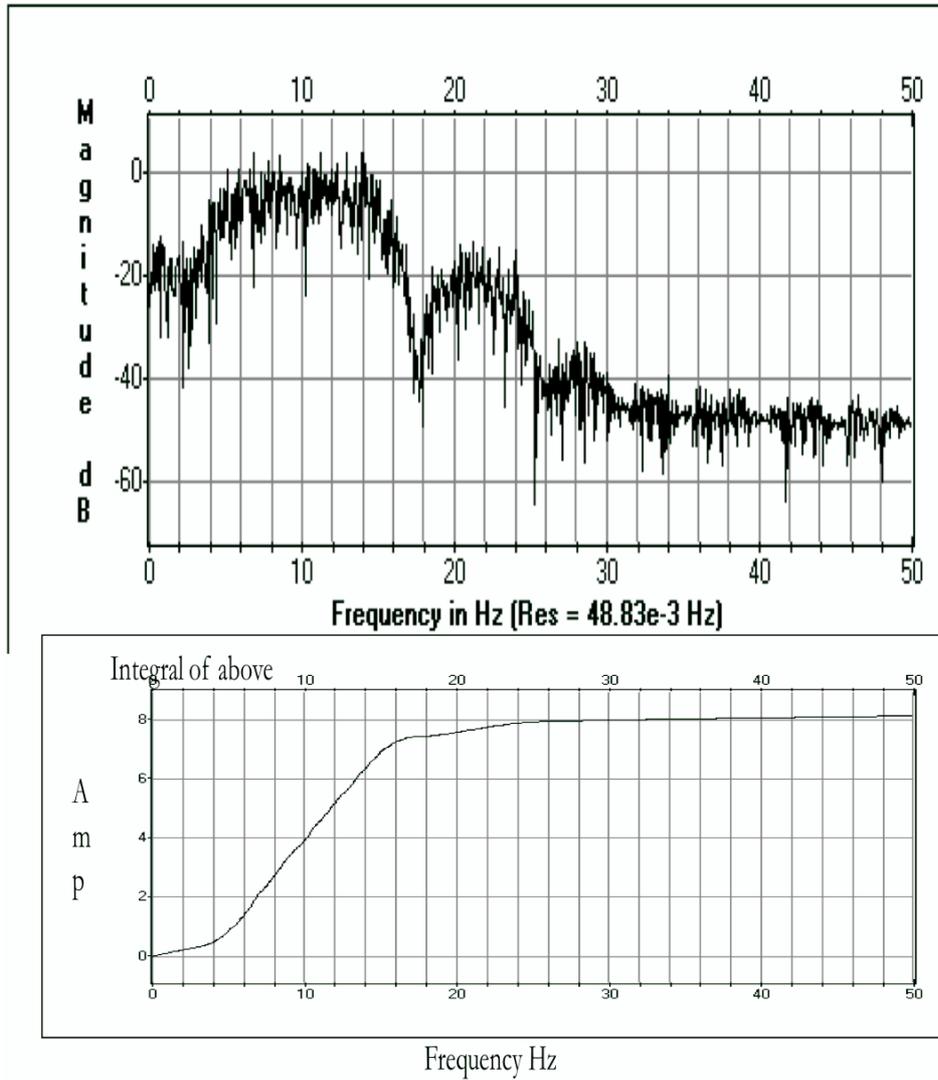


Figure 3.4: Shaped continuous phase

3.2.2 Pulse Shaping

In order to increase the data transmission rate through a bandwidth allocation without affecting BER there is the option of considering pulses requiring less overall bandwidth (including guard-band) than the rectangular pulse. Ideally the pulse shape should be smooth and introduce no Inter-Symbol Interference (ISI). Correctly shaping the MPSK data pulses reduces the rate of phase transitions reducing the out of band spectral components.

Any equal area pulse into an FM modulator will result in an equal phase change [4]. From this the optimum shape for minimum bandwidth having the greatest area over a fixed time period and amplitude would appear to be rectangular having no ISI and all information within the pulse period. Rectangular shaping, however, produces the maximum instantaneous frequency transitions. With CPM FM these transitions can be limited, at worst, to the instantaneous difference between the maximum and minimum frequencies of the modulation.

In the early work the problem was to find a spectrally optimum shape, i.e. to minimize the rate of phase transitions whilst improving bandwidth efficiency. This could be achieved in many ways, for example, by predefining a known area waveform or pre-filtering the rectangular shaped elements with a low pass filter [32].

It was necessary, therefore, to find the increased level of signal allowing the same change of phase when converting from rectangular to, for example, a sinc pulse. A sinc pulse in the time domain becomes a rectangular pulse in the frequency domain. This has the advantage of low bandwidth and no ISI though is subject to timing jitter. The raised cosine requires more bandwidth than the sinc having no ISI and is almost free of jitter at the receiver [4].

For comparison purposes simulations were performed to compare different shaping with both phase and frequency modulations with respect to their spectral effects. These simulations showed that given the criteria of limited bandwidth utilization with low out of band noise the best waveform was a Root Cosine 0.4 with frequency modulation. Figure 3.5 shows the result of comparing the spectral results of the best frequency modulated system (Root Cosine 0.4) with the best phase modulated shape (Raised Cosine). The Root Cosine resulted in the best spectral results.

3.2.3 Early Error Coding

The first attempts at a coding and integrated decoding scheme were developed using the Entegra simulator. The decoding schemes had a trellis structure using a Maximum Likelihood Sequence Estimation (MLSE) route trajectory technique

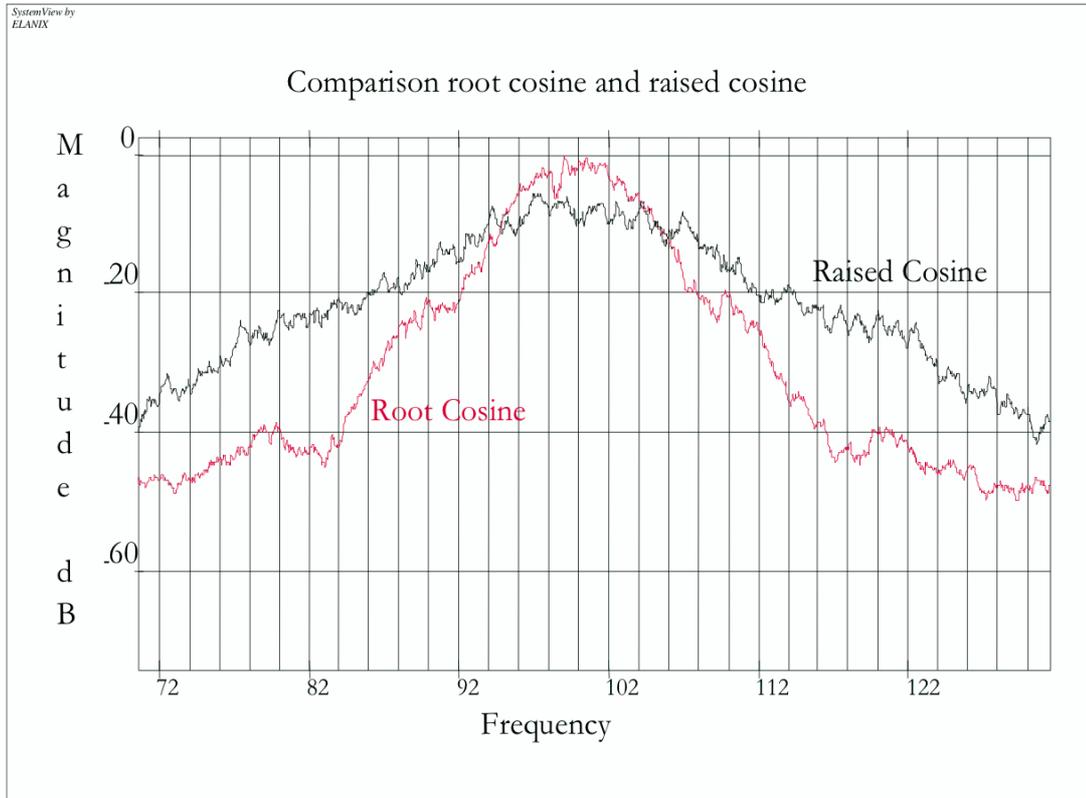


Figure 3.5: Root/raised cosine comparison

so called as it views the entire sequence of channel signals and makes a maximum likelihood estimate of the transmitted signal [4]. The signal available for observation was:

$$f(t) = g(t, \beta) + n(t) \quad -\infty < t < \infty$$

where $n(t)$ represents a Gaussian random process having zero mean and a one sided power spectral density $\frac{N_0}{2}$, β is the transmitted sequence and $g(t, \beta)$ is the modulated signal. A detector, which minimizes the probability of error, must observe the received signal over the complete time axis and, in practice, choose a long sequence β that minimizes the error probability [12].

The original simulations were performed to examine possibilities based on existing techniques. The first sequence code simulation was a partial response sequence technique based on TFM. From now on it will be called expanded TFM even though it is not a true TFM modulation. This technique, the encoder and state structure being shown in Figure 3.6, was potentially spectrally efficient.

A shift register structures the data into 8 potential states having 4 phase levels, Table 3.1, utilising π and $\frac{\pi}{2}$ phase elements. The modulation decoder has a trellis structure but was found to be too short for the double symbol error of differentially encoded modulation. Fortunately using structured modulation within a sequence rather than structured data provided some additional information within the modulation, in the form of the relative quality of the phase vector, to assist error correction. In addition, as the structure is part of a spectrally efficient modulation scheme symbol error correction came as a bonus.

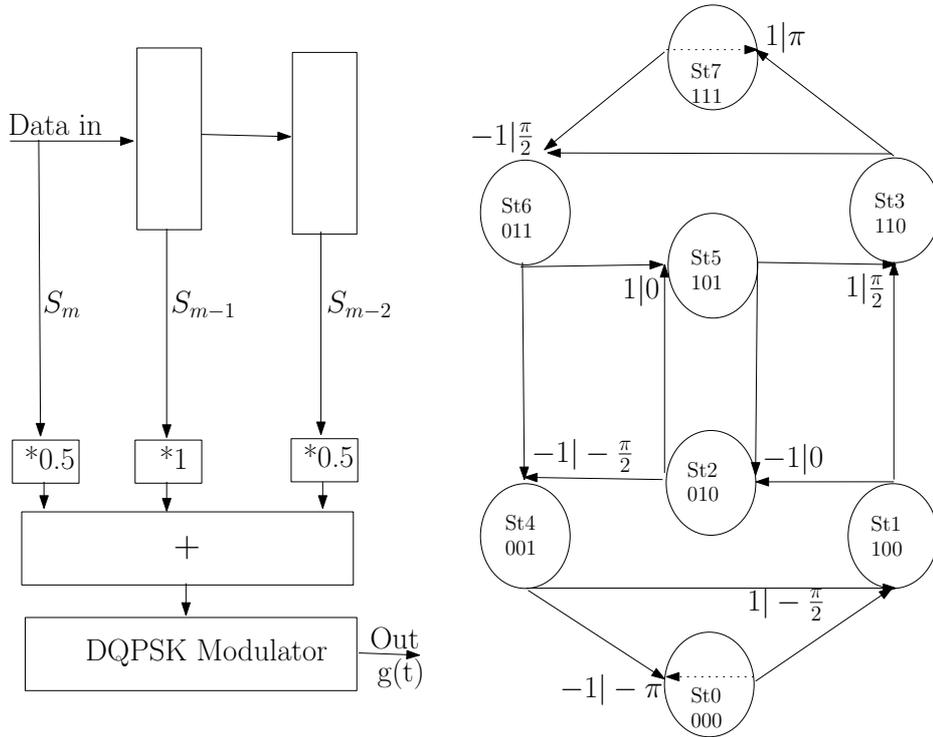


Figure 3.6: 4 phase encoder and state sequence (Based on TFM principles)

The input to the encoder, Figure 3.6, consists of 3 bits s_m, s_{m-1}, s_{m-2} where s_m is the last received bit and each output signal element can be valued within the set;

$$A = \{-2, -1, 0, 1, 2\}.$$

Each value can be derived from;

$$i = 0.5s_m + s_{m-1} + 0.5s_{m-2}$$

s_{m+1}	s_m	s_{m-1}	Relative Carrier Phase	State (St)
-1	-1	-1	$-\pi$	0
1	-1	-1	$-\frac{\pi}{2}$	1
-1	1	-1	0	2
1	1	-1	$\frac{\pi}{2}$	3
-1	-1	1	$-\frac{\pi}{2}$	4
1	-1	1	0	5
-1	1	1	$\frac{\pi}{2}$	6
1	1	1	π	7

Table 3.1: Bit sequence to relative phase

where: $s_i \in \{+1, -1\}$.

The coding rule for this modulation can be written [32];

$$\varphi(mT + T) - \varphi(mT) = \frac{\pi}{2} \left(\frac{s_m}{2} + s_{m-1} + \frac{s_{m-2}}{2} \right)$$

The binary to phase representations of the five values of A are;

A vaule		Bit Sequence		Relative Phase
2	\leftrightarrow	(111)	\leftrightarrow	π
1	\leftrightarrow	(11-1),(-111)	\leftrightarrow	$\frac{\pi}{2}$
0	\leftrightarrow	(-11-1),(1-11)	\leftrightarrow	0
-1	\leftrightarrow	(1-1-1),(-1-11)	\leftrightarrow	$-\frac{\pi}{2}$
-2	\leftrightarrow	(-1-1-1)	\leftrightarrow	$-\pi$

Table 3.2: Input sequences and states

The relative phase changes are double those of regular TFM. The modulation being DQPSK; the digital information is coded by the phase changes $\Delta\phi_n$.

The relative phases to states and those states are shown in Figure 3.6 and Table 3.2. The diagram corresponds with a Moore state machine diagram. In this figure the states are shown as states 0 to 7. The output relative phase and the binary input (-1 and 1) are also indicated.

The symbol sequence at each successive sampling moment is dependant on the previous state and the phases changes are restricted by that dependence. Hence although $\pm\pi$ are, as far as a phase detector is concerned, the same point the two possibilities are separated by three states within the sequence hence the direction of

Digital data in	1	-1	-1	-1	-1	1	-1	1	-1	1
Relative phase sent	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$-\frac{\pi}{2}$	π	π	$-\frac{\pi}{2}$	0	0	0	0
Relative phase received	$\frac{\pi}{2}$	$\frac{\pi}{2}$	$-\frac{\pi}{2}$	$\frac{\pi}{2}$	$-\frac{\pi}{2}$	$-\frac{\pi}{2}$	0	0	0	0
Error indication	1	2	2	2	2	1	0	0	0	0
State out	3	6	4	3	4	1	2	5	2	5
Sent state	3	6	4	0	0	1	2	5	2	5

Table 3.3: Error correction example

phase rotation can be resolved. A small amount of intelligence, related to the phase sequence, in the detector can differentiate between direction of arriving to phase value π within the sequence. Certain received errors were found able to overcome the decoder intelligence, however, if the error can be correctly placed within the sequence then correction would not be difficult. An error correction example is shown in Table 3.3. The start state is 3.

The error indication is derived from the Viterbi metric. This error indication is obtained from a step through a simplified Viterbi decoder. An error was noted when the received relative phase was unexpected. In the above a relative phase shift of $\frac{\pi}{2}$ instead of the two true continuums $-\frac{\pi}{2}$ or $-\pi$. The Viterbi decoder in this environment was unable to correct errors but assisted in detecting and positioning them using the error indication (metric). This, by itself, is a weak test as the Viterbi decoder is liable, as shown in table 3.3 and later in section 5.3, to indicate an error in advance of the true erroneous symbol position and without application of further support, such as trace-back, this can be critical.

The correction was made from the first appearance of an error indication and an unexpected phase. In the example the error is obvious given that state 4 has no direct connection to state 3. The invalid state continuum is another indication of error.

The start of error can be confused. Also important is the direction of rotation which is indicated by a "–" for the reverse direction in Figure 3.6. In the detector some errors appear due to ambiguities that could be resolved with a longer code but together with other complications, due to the receiver intelligence, causing problems a new technique based on the sequence principle was sought.

3.3 New Codulation System

Due to the problems in the previous section the technique was modified. The new technique, we have called DQPSK after the modulation technique employed, main-

tains the encoding principle of a derived state sequence, the standard three bit state sequence (s_m, s_{m-1}, s_{m-2}) where $s_i \in \{1, 0\}$. The new phase rotation at any state is shown in Figure 3.7 and Table 3.4. The states in Table 3.4 are obtained as

$$St = s_m 2^0 + s_{m-1} 2^1 + s_{m-2} 2^2$$

where $s_i \in \{0, 1\}$.

This table is arranged in a diagram that is similar to a Karnaugh mapping and it is shown in Table 3.5. It ensures a non-ambiguous length three symbol sequence for single couplet error correction. This also ensures a Hamming distance of at least 2 in each three bit sequence element allocated to the same phase. This proved, empirically, to be a requirement during a study of error structure. An ambiguous arrangement, such as table 3.5, led to potentially ambiguous results for the sequence search leading to duplicate sequence metrics and higher BERs.

In Table 3.5 the allocation of the phase element, π , is seen as dependant solely upon the $s_m = 1$ and $s_{m-2} = 0$. Due to this ambiguity the recovered bit sequence could either 100 or 110. A similar problem applies to the other sequences associated with phase 0 (010 or 011) and $\frac{3\pi}{2}$ (101 or 001) both having a Hamming distance of one. The allocation of phase, therefore, became an element in the future sequence code. Because of the mentioned reasons the assignment of phases to the 3 bit sequence (s_m, s_{m-1}, s_{m-2}) was changed to the values gathered in Table 3.6.

s_m	s_{m-1}	s_{m-2}	Relative Carrier Phase	State (St)
0	0	0	0	0
1	0	0	π	1
0	1	0	$\frac{\pi}{2}$	2
1	1	0	$\frac{3\pi}{2}$	3
0	0	1	$\frac{3\pi}{2}$	4
1	0	1	$\frac{\pi}{2}$	5
0	1	1	0	6
1	1	1	π	7

Table 3.4: Bit sequence to relative phase

The new system is four phase, removing the difficulty of the five levels of the previous system, but retained the DQPSK transmission. Each state transition now had a separation of π . In this case an error is recognized by a transition of greater than $\frac{\pi}{2}$ radians from that expected. Using a previously recovered state as a precursor an error is noted when the phase is greater than $\pm\frac{\pi}{4}$ from that expected within the sequence. This erroneous phase symbol is then modified by $\pm\frac{-\pi}{4}$ in the appropriate direction. The following phase symbol is then modified by a $\mp\frac{\pi}{4}$. This information

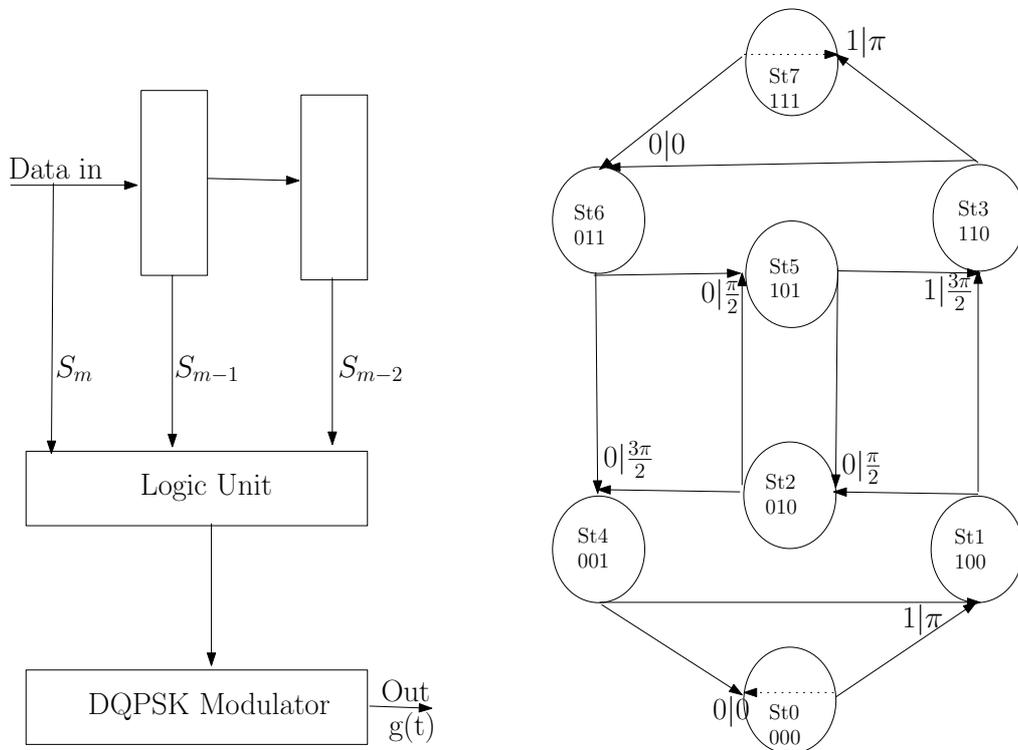


Figure 3.7: DQPSK data modulator and 8 state sequence. The modulator output $g(t)$ has phase shown in the diagram.

resulting in four state sequences that can then be tested against possible valid state routings within the sequence. The original state routing can then be chosen from the limited number of unambiguous choices. The results of this simple operation were moderately successful, Figure 3.8.

3.3.1 Comments on new modulation

The early system had 5 level steps with 4 phases and hence the decoder was more complex and less efficient than the DQPSK system. The redundancy is in the modulation instead of the data and each symbol is within a continuing sequence of 8 states. Due to the state sequence, an error could be detected and easily corrected. The principle of the new software was simple being based on a continuous sequence.

$\mathbf{s}_m = \mathbf{1}$		$\mathbf{s}_m = \mathbf{0}$		$\mathbf{s}_m/\mathbf{s}_{m-1}$
π	$\pi/2$	0	0	$\mathbf{s}_{m-1} = \mathbf{1}$
π	$3\pi/2$	$3\pi/2$	$\pi/2$	$\mathbf{s}_{m-1} = \mathbf{0}$
$\mathbf{s}_{m-2} = \mathbf{0}$	$\mathbf{s}_{m-2} = \mathbf{1}$	$\mathbf{s}_{m-2} = \mathbf{0}$	$\mathbf{s}_{m-2} = \mathbf{1}$	$\mathbf{s}_{m-1}/\mathbf{s}_{m-2}$

3 Bit Sequence	Phase
(01-)	\leftrightarrow 0
(000),(111)	\leftrightarrow $\pi/2$
(1-0)	\leftrightarrow π
(-01)	\leftrightarrow $3\pi/2$

An “-” indicates an indeterminate decision

Table 3.5: Non optimised phase allocation with sequence effects

$\mathbf{s}_m = \mathbf{1}$		$\mathbf{s}_m = \mathbf{0}$		$\mathbf{s}_m/\mathbf{s}_{m-1}$
$3\pi/2$	π	0	$\pi/2$	$\mathbf{s}_{m-1} = \mathbf{1}$
π	$\pi/2$	$3\pi/2$	0	$\mathbf{s}_{m-1} = \mathbf{0}$
$\mathbf{s}_{m-2} = \mathbf{0}$	$\mathbf{s}_{m-2} = \mathbf{1}$	$\mathbf{s}_{m-2} = \mathbf{0}$	$\mathbf{s}_{m-2} = \mathbf{1}$	$\mathbf{s}_{m-1}/\mathbf{s}_{m-2}$

3 Bit Sequence	Phase
(011),(000)	\leftrightarrow 0
(101),(010)	\leftrightarrow $\pi/2$
(100),(111)	\leftrightarrow π
(110),(001)	\leftrightarrow $3\pi/2$

Table 3.6: Irreducible phase allocation logic for early sequence code

Once an error was found in the received version of the transmitted phase ϕ_{m-1} being $\bar{\phi}_{m-1}$ by analysing the possibility that the sequence $\bar{\phi}_{m-1} \rightarrow \bar{\phi}_{m-1}$ is valid. In the event of an error this analysis would provide two possible corrections. The nearest correction value is assumed and the initial correction performed by addition and subtraction of the discovered error from the $\bar{\phi}_{m-1}$ and addition of this error to $\bar{\phi}_m$ and the result tested against $\bar{\phi}_{m+1}$ providing protected bit correction. In the event that this correction does not meet the adjusted $\bar{\phi}_{m-1} \rightarrow \bar{\phi}_{m+1}$ sequence the alternative correction is tried. If this fails an error in $\bar{\phi}_{m-1}$ is assumed and the sequence correction continues at $\bar{\phi}_{m-1} \rightarrow \bar{\phi}_{m+1}$.

It appeared that this simple short codulation technique, in the environment in which it was tested, was potentially competitive with equivalent in data soft decision Viterbi decoding. The Viterbi decoder used was disappointing in this environment;

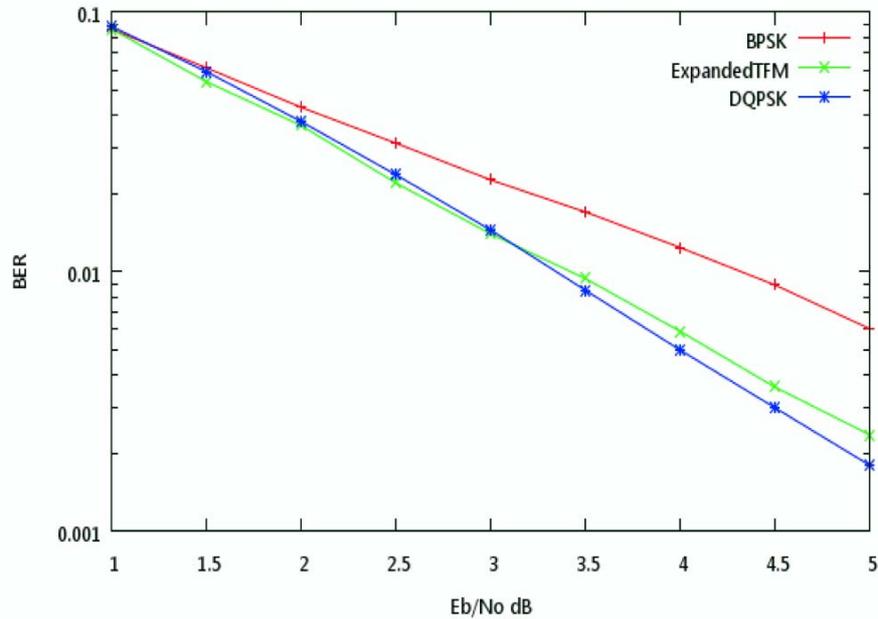


Figure 3.8: Expanded TFM and DQPSK results

probably due to its short length within an error environment which in FM derived (DPSK) phase modulation schemes appears bursty. Symbol errors are constrained to occur in couplets or longer sequences. At high BER the chances of a second error couplet occurring adjacent to another error couplet, thus overcoming the code, is relatively high.

The modulation code was designed to correct any separated couplet errors and actually corrected many of the connected couplet errors. When bursts occur the technique corrects many of the errors at the bit relationship. The technique, on double errors, goes round the states the alternative way producing, normally, 2 bit errors when correction fails. The technique has problems with connected couplets where the centre of the three symbol errors is, by the addition of errors, corrected making it appear as a couplet error with a central correct symbol. This error process produces 2 bit errors. This problem occurs because the technique has been designed for couplet errors. It was believed that by studying the error structure further improvements might have been possible but it was considered that the time would be better spent improving and simplifying the code.

Chapter 4

Sequence codes

The previous chapter presented early work on a short, phase modulated, symbol sequence, error correction code within a DQPSK environment. During this work problems, related primarily to the use of short modulation codes in a duplex error environments were apparent. The early versions of sequence code offered a simple to implement, short delay, multiple, integrated decoder, approach to the error correction problem. The original work continues in this chapter using the original encoding principles within a QPSK environment; Figure 4.1.

The necessary improvements required only a few simple modifications to the principle, however, the changes necessitated the creation of a dedicated simulator, Chapter 5, as the upgraded decoder system could not be simulated within the available Elanix or Matlab environments. The upgraded code uses redundancy within an encoded QPSK constellation allowing a new concatenated decoder system to use an open window technique. The open window is a symbol predictive technique based on a continuing encoded phase sequence, Section 4.3, allowing the reception window, of the phase decoders, to be significantly increased. [71, 73, 74]

Two received phase representative descriptions are used here; $\tilde{\phi}_m$ and $\bar{\tilde{\phi}}_m$. The element $\tilde{\phi}_m$ is a quantized representation of the transmitted ϕ_m and $\bar{\tilde{\phi}}_m$ indicates that $\tilde{\phi}_m$ has passed through the enhancer though if the enhancement conditions are not met the phase representation $\bar{\tilde{\phi}}_m$ will not have been enhanced.

The encoder bit, s_m , to state, St_m , to phase, ϕ_m , transitions are shown in Figure 4.3 and Table 4.1. Each state transition $St_{x-1} \rightarrow St_x$ is allocated a phase based on the irreducible logic shown in Table 4.2. The phases are taken from a QPSK alphabet; $Y = \{0, \frac{\pi}{2}, \pi, \frac{3\pi}{2}\}$. partitioned into subsets; $Y_1 = \{0, \pi\}$ and $Y_2 = \{\frac{\pi}{2}, \frac{3\pi}{2}\}$; each subset element having a separation of π .

The code can be resolved as a 4 state, 8 branch version, Figure 4.2, or an 8

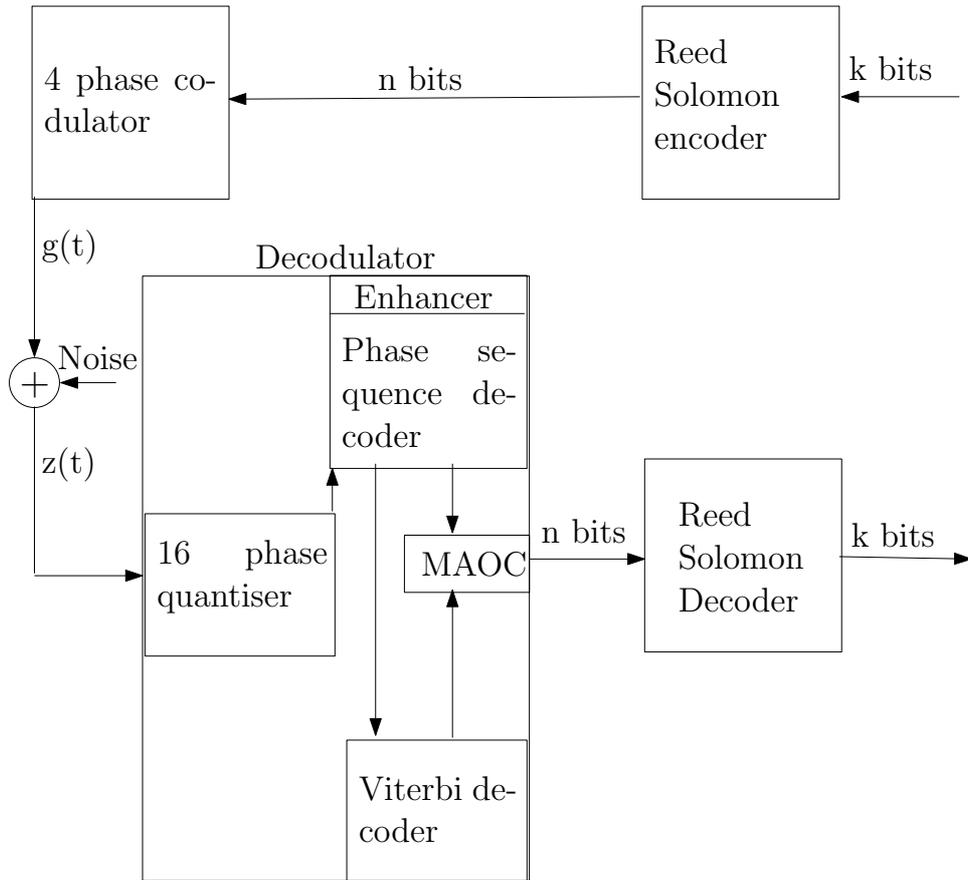


Figure 4.1: Sequence error correction schematic

state version, Figure 4.3. The simulation (Chapter 5) uses the 8 state version with multiple concatenated inner decoders and an outer $(63, 43)$, over $GF(64)$ systematic RS code having a d_{min} of 20 symbols and capable of correcting 10 erroneous 6 bit symbols.

Phase to state recovery is performed by two integrated decoders consisting of a phase sequence decoder concatenated with a specialised Viterbi decoder. Both decoder use the same received QPSK sequence to recover an estimate of the original state sequence. Each state recovery decision giving a metric indication of the reliability of a state recovery decision. The outer decoder, the MAOC, uses these metrics with associated states to recover an estimate of the original bit sequence. Any remaining bit errors appearing as short error bursts, Figure 5.5.

It would be possible to pass state and metric information to the, currently, RS

s_m	s_{m-1}	s_{m-2}	Carrier Phase	State
0	0	0	0	0
1	0	0	π	1
0	1	0	$3\pi/2$	2
1	1	0	$\pi/2$	3
0	0	1	$3\pi/2$	4
1	0	1	$\pi/2$	5
0	1	1	0	6
1	1	1	π	7

Table 4.1: Bit sequence to phase

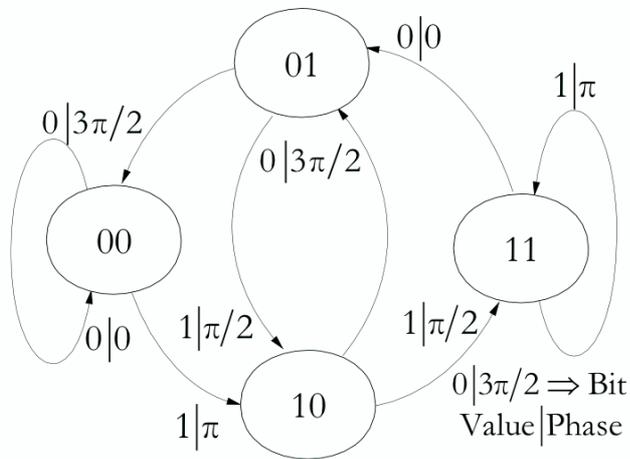


Figure 4.2: 4 States, 8 branches to 4 phases

$\mathbf{s}_m = \mathbf{1}$		$\mathbf{s}_m = \mathbf{0}$		$\mathbf{s}_m/\mathbf{s}_{m-1}$
$\pi/2$	π	0	$3\pi/2$	$\mathbf{s}_{m-1} = \mathbf{1}$
π	$\pi/2$	$3\pi/2$	0	$\mathbf{s}_{m-1} = \mathbf{0}$
$\mathbf{s}_{m-2} = \mathbf{0}$	$\mathbf{s}_{m-2} = \mathbf{1}$	$\mathbf{s}_{m-2} = \mathbf{0}$	$\mathbf{s}_{m-2} = \mathbf{1}$	$\mathbf{s}_{m-1}/\mathbf{s}_{m-2}$

Table 4.2: Phase allocation logic for sequence code

Bit Sequence	Resulting Phase
$s_{m-3}, s_{m-4}, s_{m-5}$	ϕ_{m-3}
$s_{m-2}, s_{m-3}, s_{m-4}$	ϕ_{m-2}
$s_{m-1}, s_{m-2}, s_{m-3}$	ϕ_{m-1}
s_m, s_{m-1}, s_{m-2}	ϕ_m

Table 4.3: Bit sequence to phase sequence

outer code. This is not currently performed as there is a difficulty translating a bit metric to a RS symbol metric. For this reason it has been left to future work.

The concatenated decoder group performs well with marginally increased complexity relative to a similar half-rate Viterbi decoder. The increased complexity being the phase sequence decoder and the transfer of the error correction capability, of both decoders, to a separate MAOC. These increases in complexity are largely offset by the sequence code inner integrated decoding pair not correcting bit errors or using standard trace-back techniques.

4.1 The Encoder

The simple encoder is designed to optimize the Euclidian distance within a bit to state to QPSK symbol sequence. The resulting technique, Figure 4.3, uses a sequence of states, formed from the incoming random data bits, to which a logical mapping, Table 4.1, allocates a phase sequence. The unit is simple and has high speed operational potential.

The resulting low pass equivalent modulated signal is;

$$g(t) = \sum_i s_i p(t - iT) \quad s_i \in \{\pm\sqrt{-1}, \pm 1\}$$

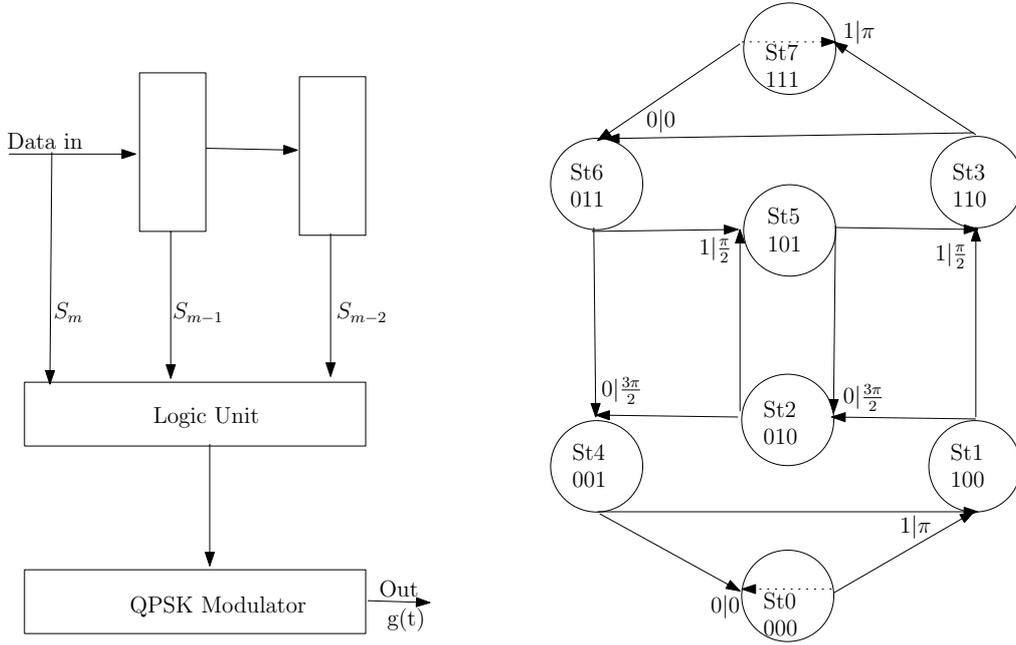


Figure 4.3: Coder and related 8 state sequence allocation. The modulator output $g(t)$ has the internode phase shown in the state sequence

where

$$p(t) = \pi \left(\frac{t}{T} \right), \quad \pi(t) = \begin{cases} 1 & 0 \leq t \leq 1 \\ 0 & \text{otherwise} \end{cases}$$

The band-pass signal is:

$$\begin{aligned}
 S(t) &= \text{Re}\{g(t)e^{j\omega t}\} \\
 &= \text{Re}\left\{\sum_k s_k p(t - kT)e^{j\omega t}\right\} \\
 &= \text{Re}\left\{\sum_k p(t - kT)e^{j(\omega t - \phi_k p(t - kT))}\right\} \\
 &= \text{Re}\left\{e^{j\omega t} \sum_k p(t - kT)e^{-j\phi_k p(t - kT)}\right\} \\
 &= \text{Re}\left\{\sum_k e^{j(\omega t - \sum \phi_k p(t - kT))}\right\},
 \end{aligned} \tag{4.1}$$

where j indicates imaginary unit.

The binary logic tools have been valuable for allocating the phase values. The individual phase element, ϕ_m , is allocated, based on the Karnaugh mapping of Table 4.2, via a state formed, from the values of three bits within the encoder register, as;

Logic	Combined Logic	f	d _h
$\bar{s}_m \cdot s_{m-1} \cdot s_{m-2} + \bar{s}_m \cdot \bar{s}_{m-1} \cdot \bar{s}_{m-2}$	$\bar{s}_m(s_{m-1} \cdot s_{m-2} + \bar{s}_{m-1} \cdot \bar{s}_{m-2})$	0	2
$s_m \cdot s_{m-1} \cdot \bar{s}_{m-2} + s_m \cdot \bar{s}_{m-1} \cdot s_{m-2}$	$s_m(s_{m-1} \cdot \bar{s}_{m-2} + \bar{s}_{m-1} \cdot s_{m-2})$	$\frac{\pi}{2}$	2
$s_m \cdot s_{m-1} \cdot s_{m-2} + s_m \cdot \bar{s}_{m-1} \cdot \bar{s}_{m-2}$	$s_m(s_{m-1} \cdot s_{m-2} + \bar{s}_{m-1} \cdot \bar{s}_{m-2})$	π	2
$\bar{s}_m \cdot s_{m-1} \cdot \bar{s}_{m-2} + \bar{s}_m \cdot \bar{s}_{m-1} \cdot s_{m-2}$	$\bar{s}_m(s_{m-1} \cdot \bar{s}_{m-2} + \bar{s}_{m-1} \cdot s_{m-2})$	$\frac{3\pi}{2}$	2

Table 4.4: Irreducible logic of Table 4.2

$$\phi_m \equiv f(s_m, s_{m-1}, s_{m-2}).$$

At each bit period, $m + x$, the value of x is decremented by 1 and a new phase element transmitted based on the new bits within the encoder register. This results in a binary phase switch at each state transition, Table 4.1. The mapping rule $f(\dots)$ is derived from the Karnaugh map, Table 4.2. $s_i \in \{0, 1\}$ is an element of the incoming bit sequence $S = (s_0, s_1, \dots, s_{n-1})$. The Karnaugh mapping technique was chosen for its simplicity over Boolean algebra for data elements exceeding 2 bits and its potential future extendibility to 8 bits or possibly more.

In this encoder the resulting phase sequence within the signal $g(t)$, Figure 4.3, is made up of the phase elements ϕ_m derived from a continuum of overlapping three bit sequences, Tables 4.3 & 4.2. Here the time reference, Table 4.3, is related to the last received bit into the coder hence ϕ_{m-3} is encoded at the time of s_{m-3} . The received bits have been described as data as the sequence code, being the project, is an inner code of a systematic RS code.

4.1.1 Phase Allocation

The bit sequence $S = (s_0, s_1, \dots, s_{n-1})$ contains the continuous random binary bit $s_m \in S - P(s_m = 0) = P(s_m = 1) = \frac{1}{2}$, — that is the input to the coder. Three continuous bit subsets of S , Figure 4.3, form the code states. The state values change as binary steps related to the contents of the register. The coder converts this sequence into a pseudo random phase sequence (Table 4.3). The mapping of the three bit subset, (s_m, s_{m-1}, s_{m-2}) to phase elements ϕ_m is designed to optimise Euclidian distance, within the phase sequences, and to maximise the Hamming distance (shown as column d_h in Table 4.4), related to the 3 bits representing the states associated with each same phase element. The Hamming distance is used by the MAOC. In Figure 4.4 is shown the phase trellis of the code starting and ending at 0 state 0 being a minimum distance.

The allocation of phases to bit subsets and state transitions is shown in Tables 4.2 and 4.1 representing the finite values of a function of binary arguments. This

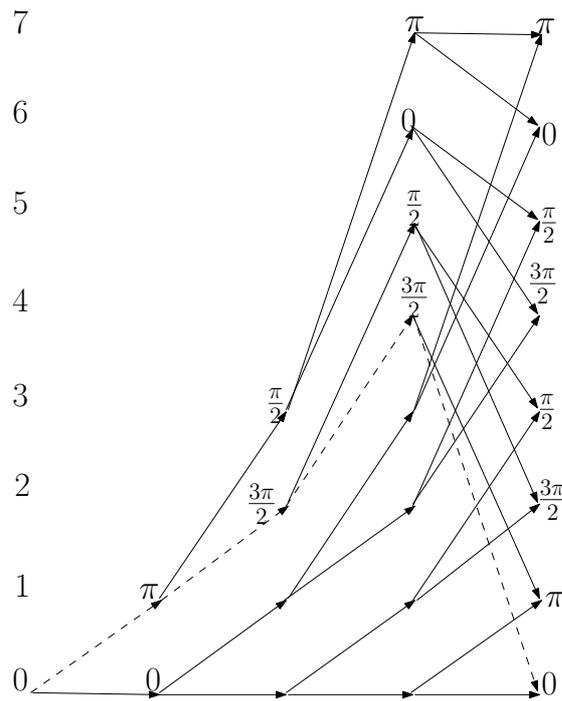


Figure 4.4: 8 State trellis (broken line is the minimum distance state 0 to state 0 for the 8 state system of Figure 4.3

Logic	f	d _h
$s_m \cdot \bar{s}_{m-1} \cdot \bar{s}_{m-2} + \bar{s}_m \cdot s_{m-1} \cdot s_{m-2}$	0	3
$s_m \cdot s_{m-1} \cdot \bar{s}_{m-2} + \bar{s}_m \cdot \bar{s}_{m-1} \cdot s_{m-2}$	$\frac{\pi}{2}$	3
$s_m \cdot s_{m-1} \cdot s_{m-2} + \bar{s}_m \cdot \bar{s}_{m-1} \cdot \bar{s}_{m-2}$	π	3
$s_m \cdot \bar{s}_{m-1} \cdot s_{m-2} + \bar{s}_m \cdot s_{m-1} \cdot \bar{s}_{m-2}$	$\frac{3\pi}{2}$	3

Table 4.5: Alternate irreducible logic

$s_m = 1$		$s_m = 0$		s_m/s_{m-1}
$\pi/2$	π	0	$3\pi/2$	$s_{m-1} = 1$
0	$3\pi/2$	$\pi/2$	π	$s_{m-1} = 0$
$s_{m-2} = 0$	$s_{m-2} = 1$	$s_{m-2} = 0$	$s_{m-2} = 1$	s_{m-1}/s_{m-2}

Table 4.6: Alternate irreducible logic of Table 4.5

function can be decomposed into the functions: $f_0()$, $f_{\frac{\pi}{2}}()$, $f_{\pi}()$, $f_{\frac{3\pi}{2}}()$ corresponding to their phase values. Each function value is binary having, at each test, two possible results: a value or its complement. As shown in Tables 4.1 and 4.4 with Figure 4.3 the bit allocation of each of the states mapped to the same phase value cannot be within the same state sequence continuums.[71, 73, 74]

4.1.2 Implementation of the coder

The coder, Figure 4.3, referred to as the codulator (coder/modulator), consists of a two element shift register with the current input a single bit: $k = 1$. The fan in (number of inputs to a logic element) and fan out (number of outputs from a logic element) of each transition is 2. At each clock pulse the state transition $St_{m-1} \rightarrow St_m$ results in the transmission of a phase element ϕ_m , representing that transition. A single transmitted phase is sent from one of the subsets of Y . With $k = 1$ and $n = 1$ (though dimensionally different) no apparent redundancy exists. From each state there are 2 possible π separated phases, from a constellation of 4, to transmit a single bit hence a half rate redundancy exists within the modulation.

The current logic allocates the phases as:

$$\phi_m = \begin{cases} \pi \text{ or } \frac{\pi}{2} & s_i = 1 \\ 0 \text{ or } \frac{3\pi}{2} & s_i = 0 \end{cases}$$

Leading to the possibility of a single phase element test for a bit.

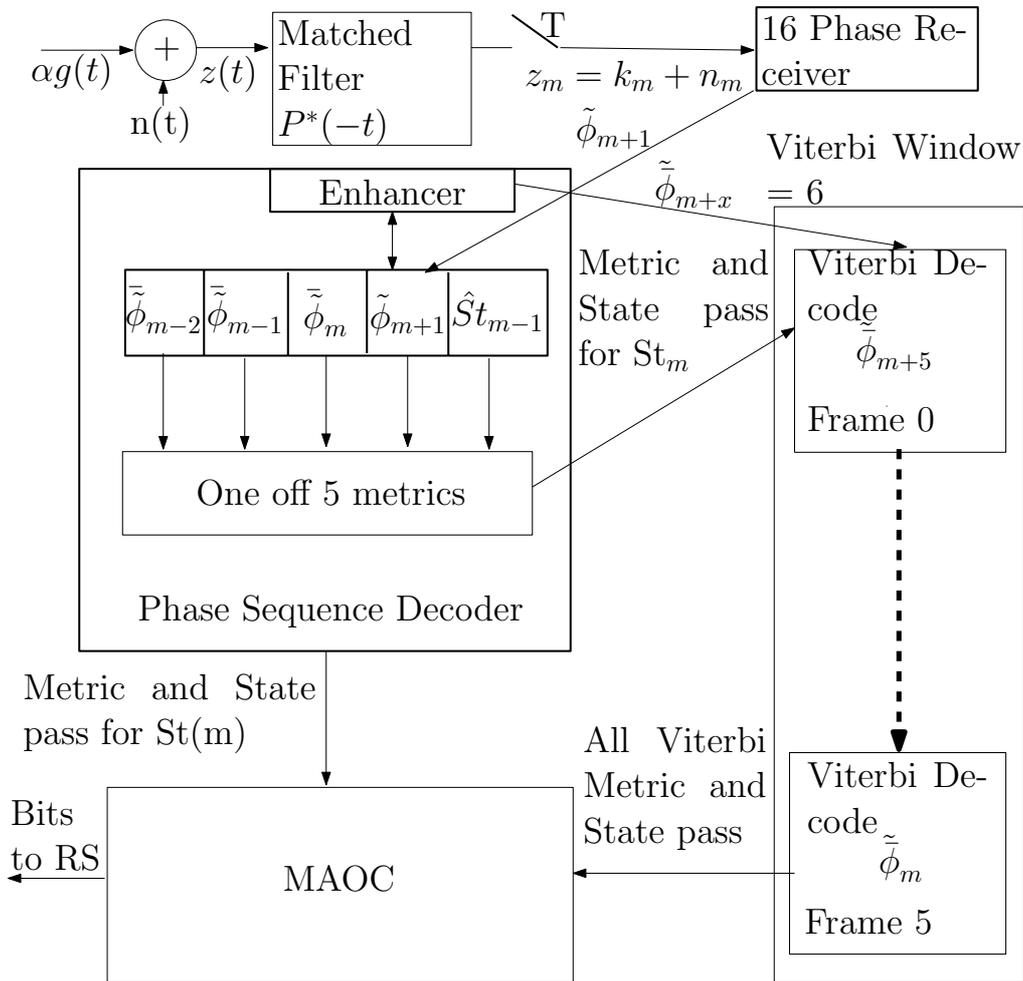


Figure 4.5: Receiver system

4.2 The Receiver System

By itself a receiver decision based on a single phase element is weak; being limited for a QPSK system to the decision region included in a single $\frac{\pi}{2}$ window. This can be improved, in sequence coding, as the transmitted phase and state sequences, based on the bit sequence S , introduce a sequential dependency. This allows a prediction of the Y subset from a previously proven state. This, potentially, opens the receiver phase acceptance window to π . The current bit sequence to phase assignment logic, Table 4.2, is only one of several possible irreducible options: e.g. Table 4.5. This table, however, with a Hamming distance, related to individual phases, of 3 (column d_h), proved difficult to implement with advantage.

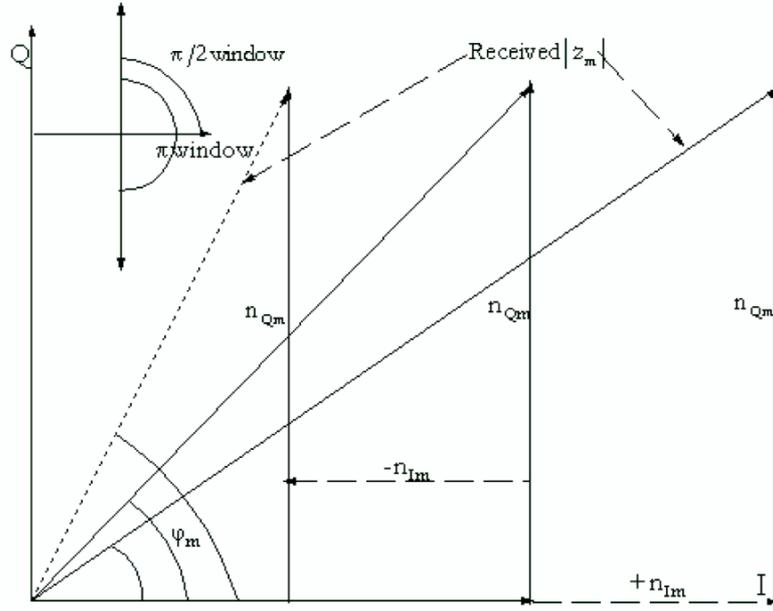


Figure 4.6: Received signal structure

The modulator output $g(t)$, Figure 4.1, contains the ϕ_m representation of the original three-bit subset; s_m, s_{m-1}, s_{m-2} . If the channel is assumed Gaussian with a real constant attenuation α , then the modulated signal at the receiver is $k(t) = \alpha g(t)$. This is combined, during transmission, with a complex $n(t)$ representing the channel and front end receiver noise. The received noise corrupted signal becomes;

$$z(t) = k(t) + n(t) \quad (4.2)$$

where $n(t)$ is a zero mean Gaussian random process having a power spectral density σ^2 . $z(t)$, Figure 4.5, is passed through a matched filter $p^*(-t)$ and sampled with a period T producing;

$$z_m = k_m + n_m \quad (4.3)$$

with $n_m = n_{Im} + jn_{Qm}$

The current ϕ_m is the protégé of an earlier phase sequence. This dependency, based on the metric defence of $St_{m-3} \rightarrow St_{m-2} \rightarrow St_{m-1}$ allows, by prediction of the next Y phase subset ϕ_m , the opening of the phase reception window. At the open window limit of π only the negative values of n_{Im} , Figure 4.6, of the noise move the received signal out of the $\pm \frac{\pi}{2}$ window increasing the symbol error probability of the phase sequence decoder.

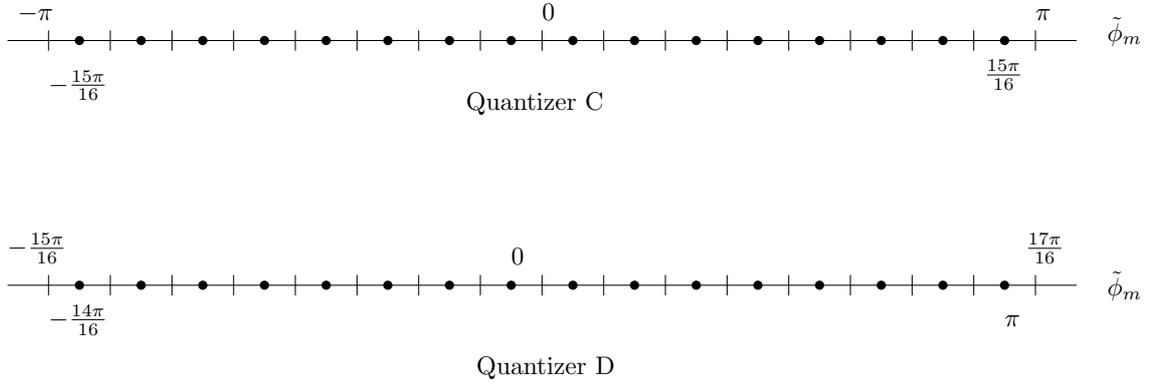


Figure 4.7: The 16 Phase quantizer (Each phase window is $\frac{\pi}{8}$)

Assuming a decision region contained in phase window of π centred in the transmitted phase, $\phi_m = 0$ in this case, then to produce a symbol error by transition of the $\pm\frac{\pi}{2}$ boundary of an individual phase symbol it is necessary that;

$$k_m + n_{Im} \leq 0$$

A positive n_{Im} component, of the noise vector, increases the modulus of the received signal $|z_m|$ reducing the effect of the n_{Qm} component potentially improving the state metrics. An increased n_{Qm} , Figure 4.6, will never exceed the $\pm\frac{\pi}{2}$ transition from the transmitted phase boundary. This boundary transition is important in sequence codes having the potential to initiate an erroneous state sequence. Such errors are normally correctable as the state metric tends to be poor.

4.2.1 16 level Phase Quantiser

The output of the sampler is followed by phase quantiser with 16 phase levels, 16 phase receiver in Figure 4.5, that quantizes the phase of the sampled signal. Two quantizing strategies are adopted: quantiser models **C** and **D**

- Quantiser **C**

$$\tilde{\phi}_n = -\frac{15\pi}{16} + \frac{n\pi}{8} \quad n \in 0 \dots 15$$

- Quantiser **D**

$$\tilde{\phi}_n = -\frac{14\pi}{16} + \frac{n\pi}{8} \quad n \in 0 \dots 15$$

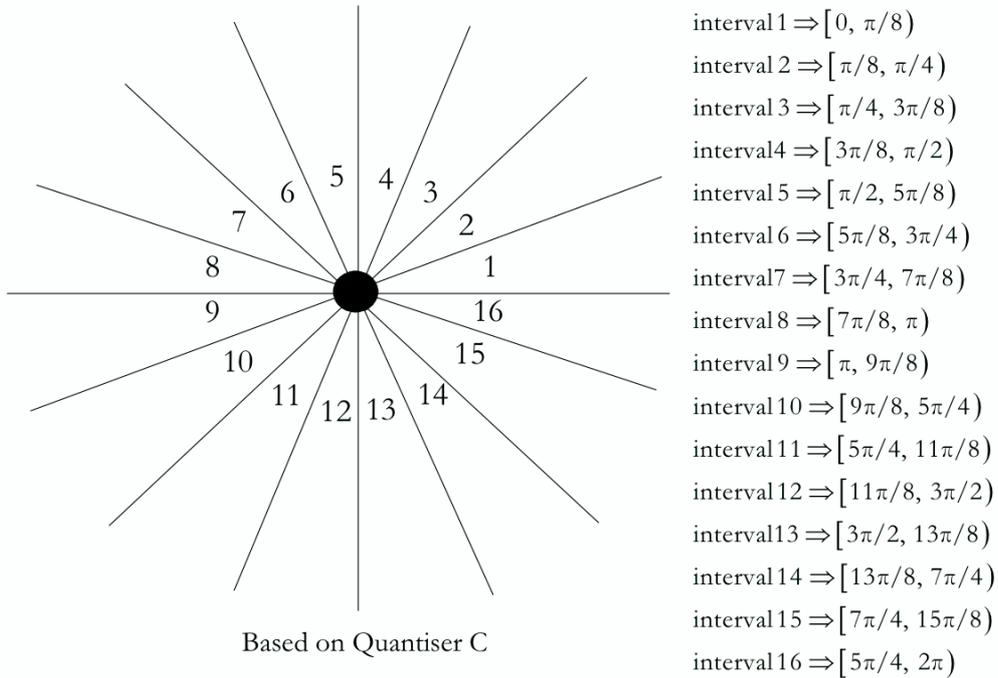


Figure 4.8: 16 Phase quantisation constellation based on quantiser C

The 16 phase quantiser takes the corrupted signal z_m and outputs $\tilde{\phi}_m$: a quantised representation of the noise corrupted transmitted ϕ_m . If the signal corruption translates $\tilde{\phi}_m$ across a quantised boundary, Figures 4.7 and 4.8, an inferior metric results.

The phase quantiser provides two items of information to the integrated decoders;

1. One of sixteen $\tilde{\phi}_m$, Figure 4.8.
2. The modulus $|z_m|$, Figure 4.6, of the sampled signal.

4.3 Open Window

This, Figure 4.9, is an important extension of the sequence coding technique [71]. To use the facility requires a phase sequence decoder estimate of the previous state

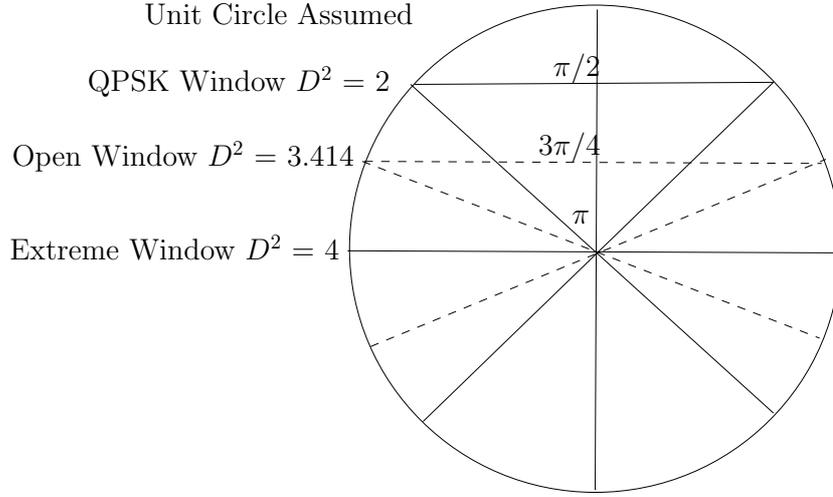


Figure 4.9: Open window distance & reception

$\hat{S}t_{m-1}$. Figure 4.10 shows the phase analysis of both the phase sequence and Viterbi decoders. With an estimate of the state $\hat{S}t_{m-1}$ the phase sequence decoder can predict, due to the phase and state sequences, with a reliability metric obtained in the unit PSD unit, that will be explained later, and allocated to $\hat{S}t_{m-1}$, the following phase subset Y_m associated with the state transition $\hat{S}t_{m-1} \rightarrow St_m$.

The bit error rate of optimally detected BPSK is [46]:

$$P_2 = \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right)$$

where:

$$\operatorname{erfc}(x) = \frac{1}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt \quad (4.4)$$

The optimal bit error rate for detected QPSK [46]:

$$P_4 = \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right) \left(2 - \frac{1}{4} \operatorname{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right) \right)$$

Dividing P4 by P2 gives;

$$P_{inc} = 2 - \frac{1}{2} \operatorname{erfc} \left(\sqrt{\frac{E_b}{N_0}} \right)$$

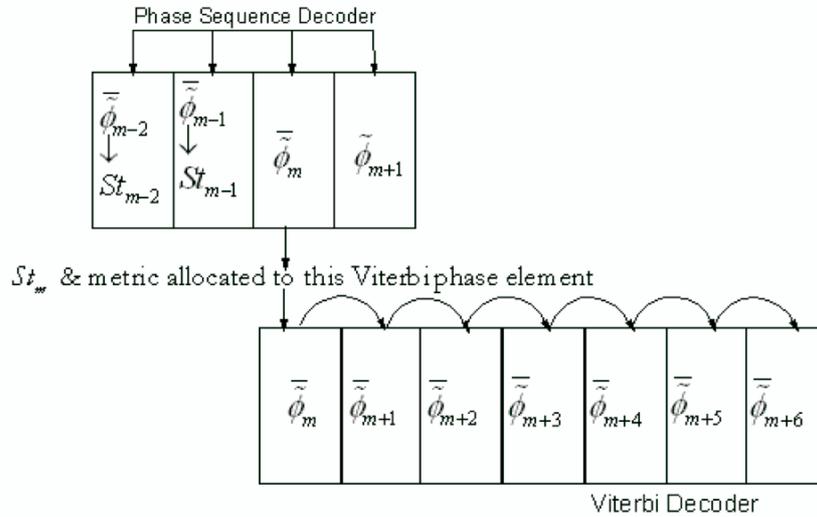


Figure 4.10: Phase decoder elements for phase sequence and Viterbi decoders

Using this result, Figure 4.11, demonstrates that the relative bit error probability QPSK - BPSK using the same Signal to Noise ($\frac{S}{N}$) ratio, is asymptotic on two with BPSK achieving superior gains over QPSK at low $\frac{S}{N}$. By developing a half rate coded QPSK transmission that allows the opening of the phase acceptance interval - Open Window - towards that of BPSK, reducing the apparent phase redundancy providing a performance enhancement, Figure 4.9.

4.4 The Decodulator

The decodulator consists of four functional units, Figures 4.1 and 4.5;

- The quantizing phase receiver.
- The phase sequence decoder including enhancer.
- The Viterbi decoder.
- The MAOC

These units are considered here as to their functionality.

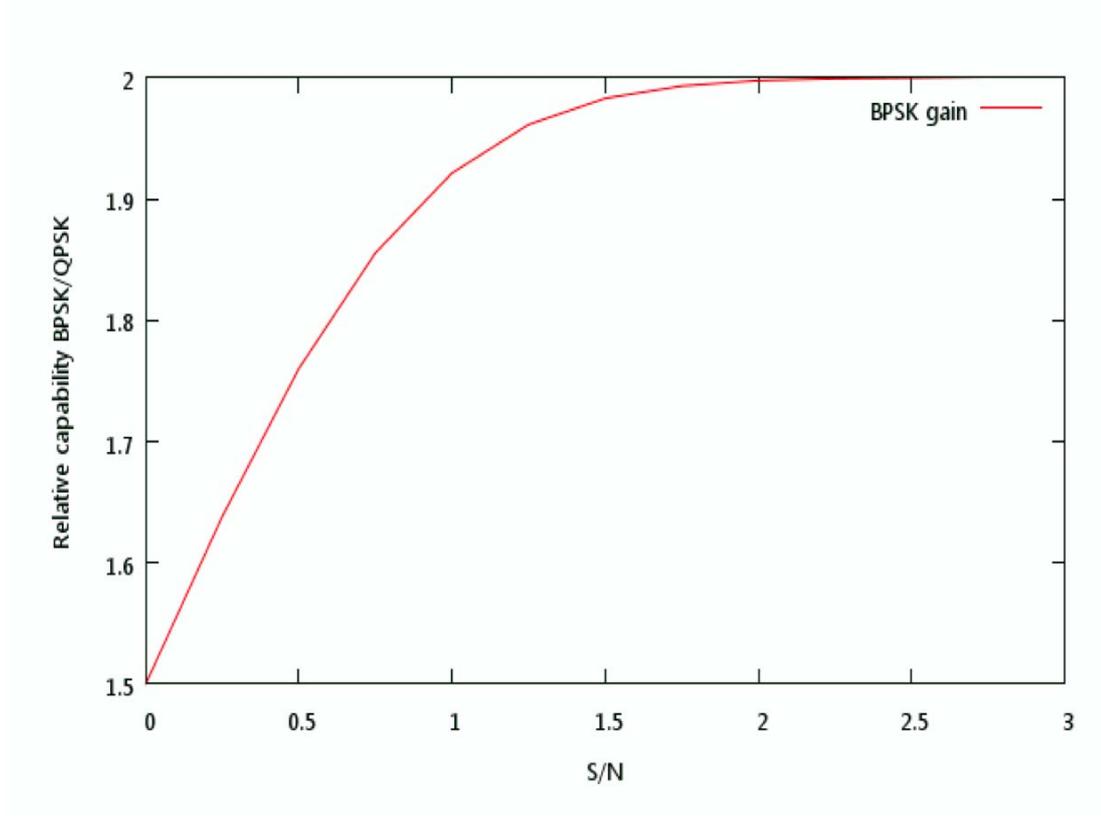


Figure 4.11: Relative bit error QPSK, BPSK

4.4.1 Integrated phase decoders (phase sequence & Viterbi)

Some of the best known trellis decoders are based on optimizing a metric dependent on the perceived link noise. In the current work this metric value is obtained empirically for both integrated decoders. Assuming a real symbol has been transmitted, the received information k_m (4.3) is also real. In this case only the real component of the noise, n_{Im} , can transit the received signal out the acceptance region. If the acceptance window has a width of π , centred on the x-axis, the imaginary component of the noise n_{Qm} has no affect on such transitions. The phase φ_m of the received signal z_m is bounded.

$$\left| \tan^{-1} \left(\frac{n_{Qm}}{k_m + |n_{Im}|} \right) \right| \leq \varphi_m \leq \left| \tan^{-1} \left(\frac{n_{Qm}}{k_m - |n_{Im}|} \right) \right| \quad (4.5)$$

Figure 4.6 applies

The smaller the error angle from the nearest ϕ_m of the predicted subset of Y_m the better the metric. This is dependant on the quality of the prediction of the current phase subset Y_m related to the $\hat{S}t_{m-1} \rightarrow St_m$ transition.

The link being Gaussian the attenuation α is a constant resulting in a constant $|k_m| \cdot |z_m|$ can, therefore, be considered an indication of the reliability of the received data and is used as a metric multiplier supporting the data reliability analysis. This is valid even if n_{Qm} is extreme; in which case the metric will tend to be poor as the phase approaches, without crossing, the $\pm\frac{\pi}{2}$ phase transition boundary and the multiplication effect largely neutralised.

The Viterbi decoder is concatenated with a specially developed inner phase sequence decoder (PSD). The two decoders operate as an integrated pair. The PSD was developed to provide phase sequence information, as a single state and metric, to the Viterbi decoder and to a simple, non iterative, bit recovery facility - the MAOC.

Within the integrated decoder pair the system is based on a Viterbi decoder frame by frame transition assessment approach based on an enhanced phase sequence that will be explained later;

$$\bar{\phi}_m, \bar{\phi}_{m+1}, \dots, \bar{\phi}_{m+w-1}$$

where w is the Viterbi decoder window length, Figure 4.10. The Viterbi decoder is supported by the short phase sequence decoder assessment based on;

$$\bar{\phi}_{m-2}, \bar{\phi}_{m-1}, \bar{\phi}_m, \bar{\phi}_{m+1}$$

The individual elements of the recovered, quantised phase data, though the same, are extracted and used differently by each of the two decoders increasing the apparent length of the code without increasing delay. The resulting states and metrics are passed to an MAOC which uses the state and metric information to correct to bit level.

4.4.2 Phase Sequence Decoder (PSD)

This decoder, shown in Figure 4.12 uses a simple continuous sequential algorithm being less complex than the iterative Fano algorithm, Figure 4.13, as it is required only to analyze, according to simple comparison rules, the possibility that a sequence of received phases, $\bar{\phi}_i$, represents the transmitted ϕ_i sequence and provide's a metric value for a single $\bar{S}t_m$ based on the possibility that $\bar{\phi}_m$ represents a valid transition phase value within the $\hat{S}t_{m-1} \rightarrow \bar{S}t_m$ continuum.

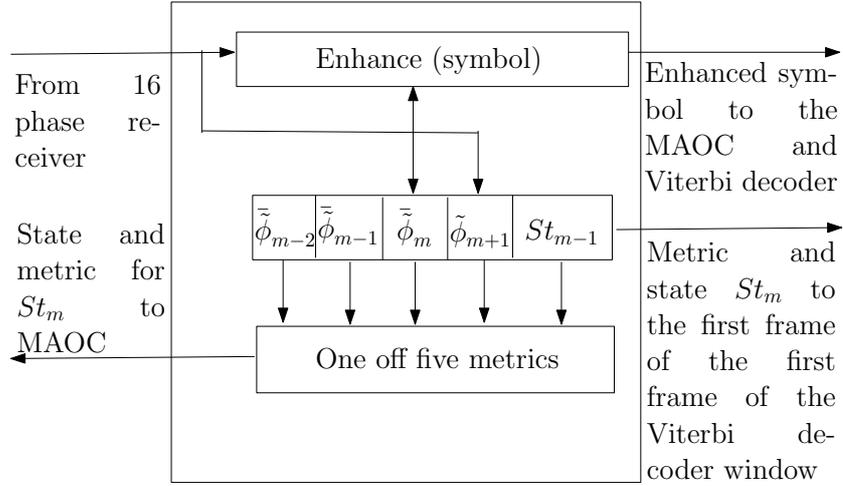


Figure 4.12: Phase sequence decoder

The unit consists of 2 elements; an enhancer and decoder. In a practical system the enhancer would form an integral part of the PSD. This separation of the two elements facilitates the removal of the phase symbol enhancer allowing a simple test for benefit within the simulation; results provided in Figure 5.1. The enhancer upgrades the received quantised representation of $\tilde{\phi}_i$ according to a simple set of rule.

The following definitions are given;

- $W_m^\theta(\tilde{\phi}_m)$: phase interval of length θ centred on the estimated phase value $\tilde{\phi}_m$ in the time period m .
- $W_m^{\frac{\pi}{2}+\frac{\pi}{2}}(\tilde{\phi}_m)$: union of two intervals each of length $\frac{\pi}{2}$, whose centres are separated by π with one constituent centred on $\tilde{\phi}_m$.

Hereafter, in order to alleviate notation the center of the above defined windows is omitted when it can be deduced from the context.

The operation of the enhancer is the following,

- if the phase sequence satisfies:

$$\tilde{\phi}_{m-2} \in W_{m-2}^{\frac{3\pi}{4}}, \tilde{\phi}_{m-1} \in W_{m-1}^{\frac{3\pi}{4}}, \tilde{\phi}_m \in W_m^{\frac{3\pi}{4}}$$

the element $\tilde{\phi}_m$ is shifted $\pi/8$ towards the expected phase. The shifted value is called $\tilde{\phi}_m$.

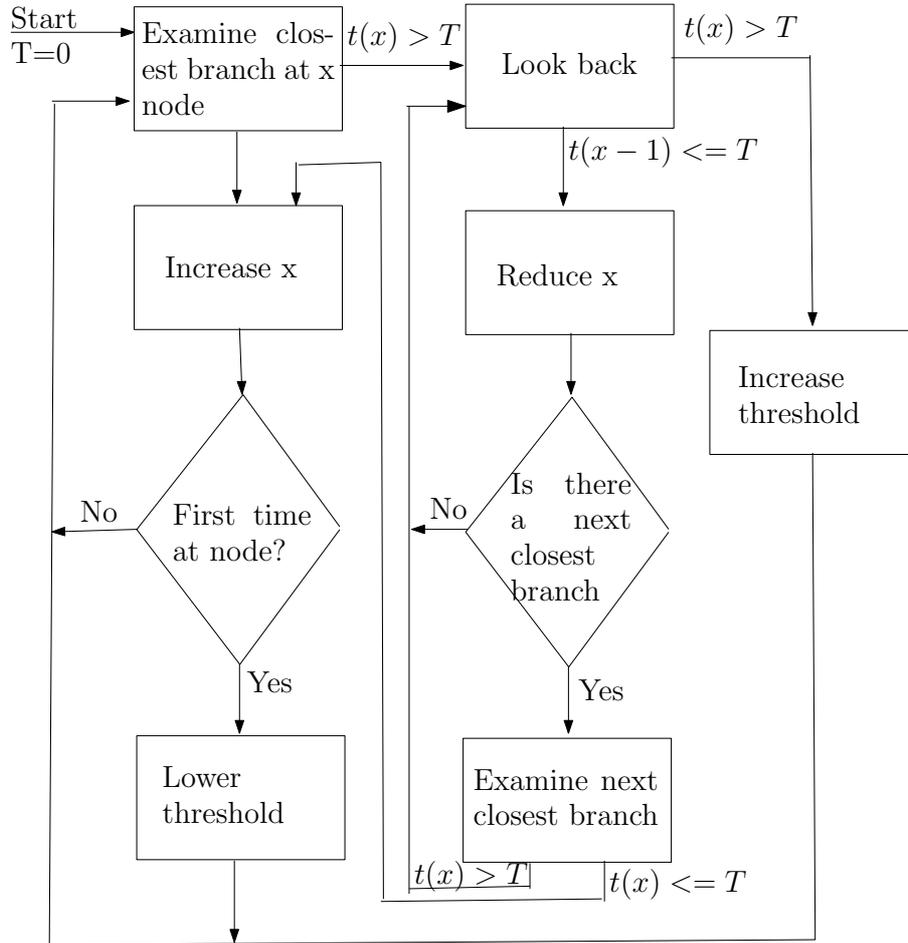


Figure 4.13: Fano algorithm

- if

$$\bar{\phi}_{m-2} \in W_{m-2}^{\frac{3\pi}{4}}, \bar{\phi}_{m-1} \in W_{m-1}^{\frac{3\pi}{4}}, \tilde{\phi}_m \in W_m^{\frac{3\pi}{4}}, \tilde{\phi}_{m+1} \in W_{m+1}^{\frac{\pi}{2} + \frac{\pi}{2}}$$

the element $\tilde{\phi}_m$ is shifted $\pi/4$ towards the expected phase. The shifted value is called $\bar{\phi}_m$.

The algorithm 4.1 gives a step by step logical sequence of the PSD. The PSD corrects no bit errors but provides the first frame of the Viterbi decoder window and the MAOC with a single state and associated metric for each received quantised phase. The metric allocation is described in Section 4.4.5. In addition it associates the following decoder phase element with a state.

Algorithm 4.1 Recovery of the state St_m with the associated metric

Obtain \hat{St}_{m-1}
if $\tilde{\phi}_{m-2} \in W_{m-2}^{\frac{3\pi}{4}}$ and $\tilde{\phi}_{m-1} \in W_{m-1}^{\frac{3\pi}{4}}$ **then**
 if $\tilde{\phi}_m \in W_m^{\frac{3\pi}{4}}$ of one of the elements of the predicted subset of Y **then**
 if $\tilde{\phi}_{m+1} \in W_{m+1}^{\frac{\pi}{2} + \frac{\pi}{2}}$ (A predicted extension of St_m) **then**
 Allocate St_m an optimum metric.
 end if
 if $\tilde{\phi}_{m+1} \in W_{m+1}^{\frac{\pi}{2} + \frac{\pi}{2}}$ is not within a predicted extension of St_m **then**
 Allocate St_m a suboptimum metric.
 end if
 end if
 if $\tilde{\phi}_m \in W_m^{\frac{\pi}{2}}$ is an element of the predicted subset of Y **then**
 Allocate St_m with a metric 3
 end if
 if $\tilde{\phi}_m \in W_m^{\frac{3\pi}{4}}$ is an element of the predicted subset of Y **then**
 Allocate St_m with a metric 4
 end if
 if $\tilde{\phi}_m \in W_m^{\pi}$ is a predicted subset extension from $\hat{\phi}_{m-1}$ **then**
 if $\tilde{\phi}_{m+1} \in W_{m+1}^{\frac{\pi}{2} + \frac{\pi}{2}}$ is a predicted subset of Y being an extension from ϕ_m **then**
 Allocate St_m a metric 5
 end if
 end if
 if $\tilde{\phi}_m \in W_m^{\pi}$ is an element of the predicted subset of Y **then**
 Allocate St_m with no metric
 end if
Send recovered St_m and metric to the first frame of the Viterbi decoder and the MAOC

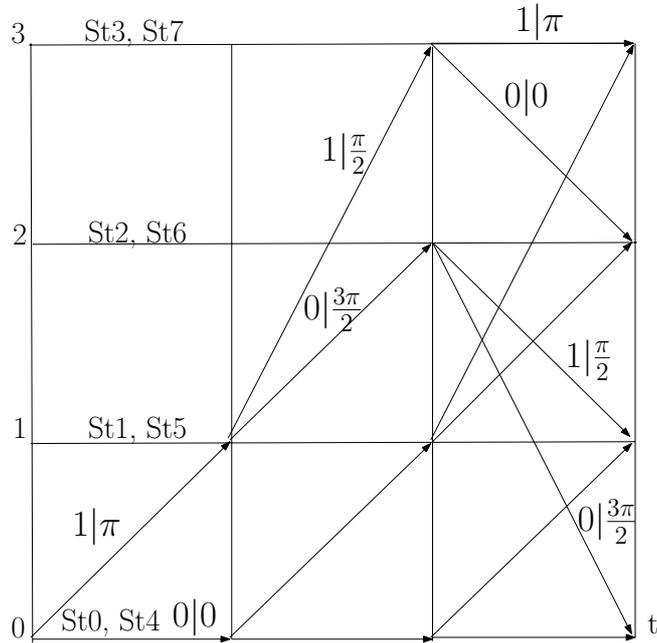


Figure 4.14: Phase trellis of 4 states with translation to 8 states

The structure of error output is important as the RS decoder functions most efficiently when used with defined length and positioned burst errors. The sequence code cannot arrange the position of error bursts though the average length of burst is reduced by this system to the advantage of the RS decoder.

4.4.3 Viterbi decoder

This decoder functions co-operatively with the PSD to produce an overall summation metric. The decoder differs from the classical Viterbi being a very short windowed, soft output system. The primary differences relate to the phase to state translation and the reception of metric reliability data associated with single state from the PSD allowing this decoder, with no trace-back facility, to usefully extend from 4 to 8 states; Figure 4.14. In addition the decoder does not correct bit errors but passes correction data to the MAOC. The data passed to the MAOC by the Viterbi decoder are the overall Viterbi frame summation metrics, and the corresponding states at the beginning of the frame. It is the function of the MAOC to correct to bit level.

The algorithm is implemented in a window of 6 frames, as shown in Figure

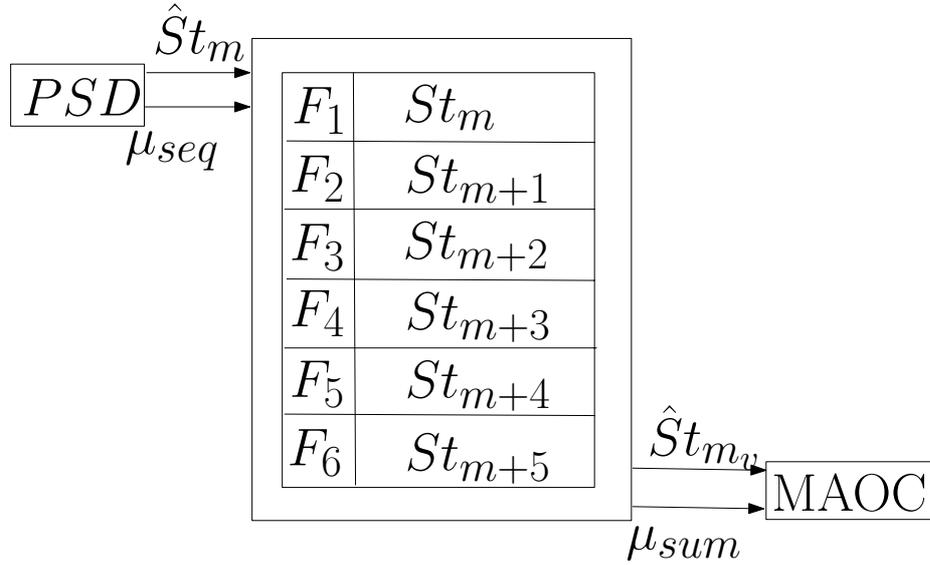


Figure 4.15: Viterbi decoder; $\hat{S}t_m$ is the initial proven state from the PSD; μ_{seq} is the associated PSD metric for the state $\hat{S}t_m$; $\hat{S}t_{m_v}$ is the potentially double proven state output of the Viterbi decoder; F_i is the frame number within the Viterbi decoder window with the associated state St_j sequence within that window

4.15. In the first frame of the window, the algorithm receives, from the PSD, the estimated state $\hat{S}t_m$ with associated metric μ_{seq} . The algorithm starts to form the base of the tree in the first frame with all 8 initial states. One important different between the algorithm presented in this Thesis and in the classical Viterbi is that one initial state does not have the initial metric attached to it. In the sequence code Viterbi one state in the base of the tree receives a PSD derived metric. This inter-decoder collaboration allows the short windowed Viterbi to operate as an 8 state system.

The reliability indication of individual transitions is provided by a metric related to the possibility that the $\tilde{\phi}_n$ value could be associated with individual transitions $\hat{S}t_j \rightarrow St_i$ within the current frame, Figure 4.16, where i relates to the current state and j its predecessor.

The number of paths (tree branches) increases exponentially (doubling in the case of the half rate decoder) with the number of frames in the decoder window. In this decoder a path from an original node to all state numbered nodes will occur, Figure 4.17. Thus each original state will be associated with all states in frame $m+3$ and above. In random error channels non-optimal paths can never become optimal [57]. In this case only optimal paths (original node to all state nodes) should be

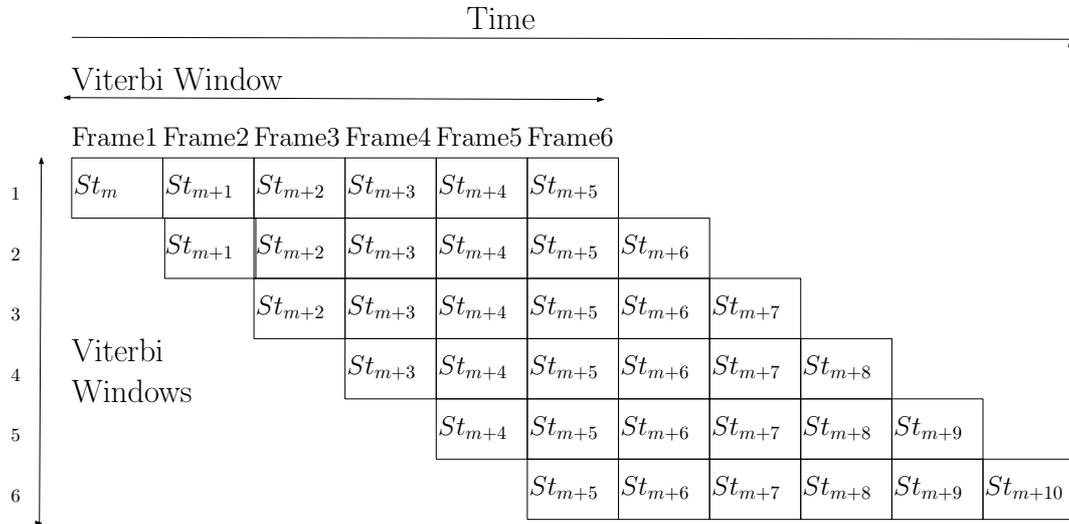


Figure 4.16: Viterbi decoder basic window structure.

maintained. The Viterbi decoder maintains a maximum of n^2 ($n = \text{no of states} = 8$ in current sequence code) paths i.e., the tree decays to a trellis at $m + 4$ and above allowing the support of a single path to each of the 8 states from each original state.

At the end of each frame, $m + 4$ and above, within the Viterbi decoder window, the decision on which of the two paths (original state node to one of the two same numbered state current nodes), Figure 4.17, to keep is based on minimum summation path metric values to that point. This figure shows that at frame $m + 4$ a single original node (in this case state 0) has expanded to 16 paths (nodes). This results in 2 same state nodes for all 8 possible states. A decision, based on the comparison of summation metrics to that point, regarding which of the two current same state nodes to original state to retain is made at this and each succeeding frame of the Viterbi decoder window.

The sequence code Viterbi decoder does not use standard route trace-back techniques. For transfers between frames only the original state, for that path, and the associated summation metric, Figure 4.17, to that point are saved. A very simple alternative facility, checking the state sequence continuity and using the phase sequence decoder results, is provided by the MAOC; Section 4.7.

The Viterbi decoder uses a tree decaying to trellis coding algorithm based on the state sequence to derive an original state associated accumulated metric for every state transition within the Viterbi window. The metrics are related to the separation of the quantised phase, $\tilde{\phi}_n$ ($n = \text{frame element}$), from each of the two

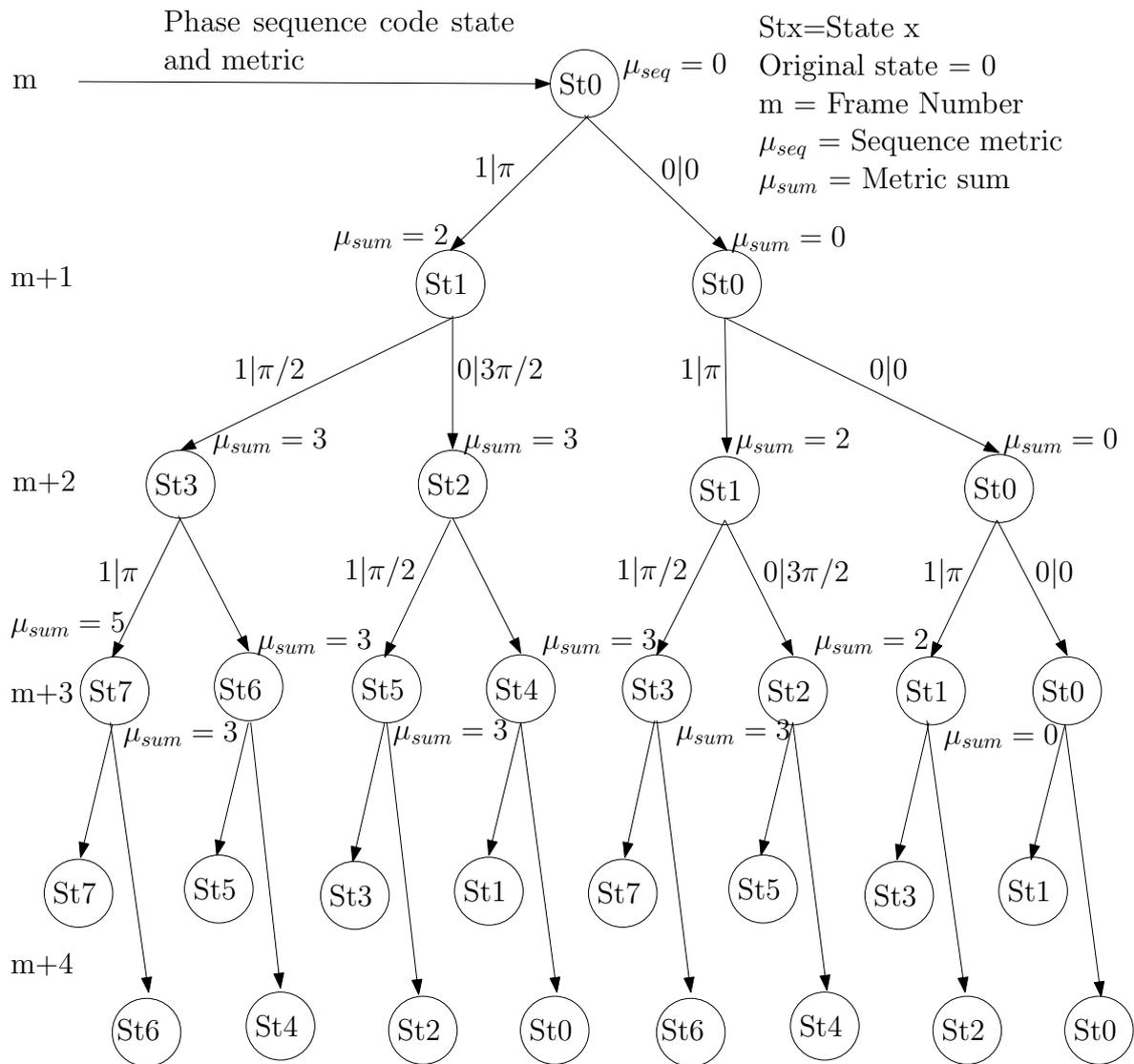


Figure 4.17: Viterbi decoding sequence tree from a single original state with received phase sequence 0, 0, 0, 0, phase sequence code metric = 0 and linear metric 0 for true phase, 1 adjacent to true phase and 2 opposite

$\bar{\phi}_{n+1}$ elements of the next predicted subset of Y . This is performed for each path through the Viterbi window Table 4.7. The Viterbi decoder finds an optimum path through a Markov graph being a sequence of states governed by a Markov chain [30].

Studies of the Viterbi algorithm showed that the Viterbi decoder outputs a minimum summation metric attached to incorrect states in short bursts. These erroneous state sequences usually starts at a Viterbi decoder window termination prior to the position of the actual corrupted phase element or elements that resulted in the poor metrics. The Viterbi decoder metrics are based on frame transitions within a Viterbi decoder window and the overall summation metric is obtained from the sum of the metrics of individual $\hat{S}t_{m+x} \rightarrow \hat{S}t_{m+x+1}$ transitions within frames. Each individual metric relates to the tested reliability of a single received $\bar{\phi}_m$ for a state transition. Assuming that a corrupted phase is sufficient to reduce a metric element for a single Viterbi decoder frame; this single poor metric will affect all summation metric values in all Viterbi decoder windows including that frames metric.

The reception of several poor metric values within a Viterbi decoder window could result in a recovered minimum summation metric, μ_{sum_m} , attached to an incorrect state. It was also found that slightly corrupted, phase sequences producing fair metric values in incorrect routings were sometimes sufficient to produce an erroneous minimum summation metrics in several Viterbi decoder windows potentially creating an erroneous sequence. The phase sequence decoder, however, using a-priori (to the point of error) sequence information would often be unaffected at this point. It is shown in, Section 5.4, that the phase sequence decoder metric in the MAOC can sometimes correct such an error.

4.4.4 The MAOC

The MAOC receives a single phase sequence decoder state and metric at each phase transition and a state and summation metric for every possible transition from every initial state to every possible end state at the end of the Viterbi decoder window. The MAOC uses simple machine intelligence rules on the received metric and state information, from the inner decoders, to transform the recovered state sequence to a corrected bit sequence.

The MAOC, shown as an element in Figure 4.5 and logically in Figure 5.4, is the state sequence to bit recovery unit. It uses as criterion for its decisions the discovered minimum summation metric, μ_{sum_m} , and associated $\hat{S}t_{m_v}$ (the original Viterbi state) from the Viterbi decoder and associated metric and state from the PSD to perform error correction to bit level. The potentially bursty bit erroneous

output of the MAOC is passed to an outer RS decoder for further correction.

The unit operates by finding one or more minimum summation metrics, μ_{sum_m} , from the 64 summation metric values available at the end of the Viterbi decoder window. Each summation metric is associated with an original state being one of the eight states of Figure 4.3. The MAOC replaces the route trace-back function of standard Viterbi decoders by ensuring, as far as possible, that the recovered state sequence is within the continuum of states to the \hat{St}_{mv} point. In the event that the recovered state does not fit the sequence then the MAOC uses alternative algorithms to provide a best fit solution.

4.4.5 Metric Allocation

A metric system for testing the reliability of decoding results is a powerful tool providing, when properly implemented, considerable gains for trellis based systems [71, 73, 74]. Table 4.7 gives the Viterbi decoder metrics order (0=optimum, 1 sub-optimum etc) for quantiser **C**, Figure 4.7. Table 4.8 gives the quantiser **C** metric allocation for the PSD.

For the Viterbi decoder windows the $\bar{\phi}$ column, of Table 4.7, contains the type **C** quantiser values, Figure 4.7. The column *Metric* \rightarrow *StateSequence* lists the order of metric for each state to state transition for each received quantised phase interval. For example receiving a quantised value $\frac{7\pi}{16}$ for a $St_0 \rightarrow St_0$ transition results in a level 3 metric whereas the $St_1 \rightarrow St_3$ transition provides an optimum metric. The quantised value $\frac{9\pi}{16}$ for $St_1 \rightarrow St_3$ the table indicates an optimum metric. The numbers in the tables representing the related metrics is not the actual metric value but the order of the metric minimum number representing the better metric.

The optimum and suboptimum metric values for the PSD are obtained based on the criterion of an acceptable phase sequence. The set inclusion in phase windows defined in sub-section 4.4.2 is the criterion adopted for calculating the metrics.

The PSD metrics are defined as follow:

- An optimum metric, metric 0 (μ_0), for the PSD is obtained if the sequence of quantised and enhanced satisfies;

$$\hat{\phi}_{m-2} \in W_{m-2}^{\frac{3\pi}{4}}, \hat{\phi}_{m-1} \in W_{m-1}^{\frac{3\pi}{4}}, \bar{\phi}_m \in W_m^{\frac{3\pi}{4}}, \tilde{\phi}_{m+1} \in W_{m+1}^{\frac{\pi}{2} + \frac{\pi}{2}}$$

where:

$$\hat{\phi}_{m-2}, \hat{\phi}_{m-1}, \bar{\phi}_m, \tilde{\phi}_{m+1}$$

$\tilde{\phi}$	Metrics \rightarrow State Sequence															
	0	0	1	1	2	2	3	3	4	4	5	5	6	6	7	7
	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow
	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7
$\frac{\pi}{16}$	0	4	4	3	4	3	0	4	0	4	4	3	4	3	0	4
$\frac{3\pi}{16}$	1	4	4	2	4	2	1	4	1	4	4	2	4	2	1	4
$\frac{5\pi}{16}$	2	4	4	1	4	1	2	4	2	4	4	1	4	1	2	4
$\frac{7\pi}{16}$	3	4	4	0	4	0	3	4	3	4	4	0	4	0	3	4
$\frac{9\pi}{16}$	4	3	4	0	4	0	4	3	4	3	4	0	4	0	4	3
$\frac{11\pi}{16}$	4	2	4	1	4	1	4	2	4	2	4	1	4	1	4	2
$\frac{13\pi}{16}$	4	1	4	2	4	2	4	1	4	1	4	2	4	2	4	1
$\frac{15\pi}{16}$	4	0	4	3	4	3	4	0	4	0	4	3	4	3	4	0
$\frac{17\pi}{16}$	4	0	3	4	3	4	4	0	4	0	3	4	3	4	4	0
$\frac{19\pi}{16}$	4	1	2	4	2	4	4	1	4	1	2	4	2	4	4	1
$\frac{21\pi}{16}$	4	2	1	4	1	4	4	2	4	2	1	4	1	4	4	2
$\frac{23\pi}{16}$	4	3	0	4	0	4	4	3	4	3	0	4	0	4	4	3
$\frac{25\pi}{16}$	3	4	0	4	0	4	3	4	3	4	0	4	0	4	3	4
$\frac{27\pi}{16}$	2	4	1	4	1	4	2	4	2	4	1	4	1	4	2	4
$\frac{29\pi}{16}$	1	4	2	4	2	4	1	4	1	4	2	4	2	4	1	4
$\frac{31\pi}{16}$	0	4	3	4	3	4	0	4	0	4	3	4	3	4	0	4

Table 4.7: Viterbi decoder frame metric allocation

is a PSD phase sequence.

- A sub optimum metric, metric 1 (μ_1), is obtained if the sequence of recovered, quantised and enhanced phase element $\tilde{\phi}_i$ satisfies:

$$\hat{\phi}_{m-2} \in W_{m-2}^{\frac{3\pi}{4}}, \hat{\phi}_{m-1} \in W_{m-1}^{\frac{3\pi}{4}}, \tilde{\phi}_m \in W_m^{\frac{3\pi}{4}}$$

- Metric 2 (μ_2) and metric 3 (μ_3) are based on the value of the current phase.
 - If $\tilde{\phi}_m \in W_m^{\frac{\pi}{2}}$ metric 2 (μ_2) results.
 - If $\tilde{\phi}_m \in W_m^{\frac{3\pi}{4}}$ metric 3 (μ_3) results.

The value of $\tilde{\phi}_m$ being taken from the subset of Y predicted as an extension from the earlier confirmed $\hat{S}t_{m-1}$.

- Metric 4 (μ_4) results if;

m - 2	m - 1	m	m + 1	Metric
$\frac{3\pi}{4}$	$\frac{3\pi}{4}$	$\frac{3\pi}{4}$	$\frac{\pi}{2} + \frac{\pi}{2}$	Optimum Metric μ_0
$\frac{3\pi}{4}$	$\frac{3\pi}{4}$	$\frac{3\pi}{4}$		Sub Optimum Metric μ_1
		$\frac{\pi}{2}$		Metric 2 μ_2
		$\frac{3\pi}{4}$		Metric 3 μ_3
		π	$\frac{\pi}{2} + \frac{\pi}{2}$	Metric 4 μ_4
		π		No Metric

Table 4.8: Phase sequence decoder metric Allocation.
In recovery mode the decoder operates at μ_2 without \hat{St}_{m-1}

$\tilde{\phi}_{m+1} \in W_{m+1}^{\frac{\pi}{2} + \frac{\pi}{2}}$, and $\bar{\phi}_m \in W_m^\pi$ are valid for the confirmed \hat{St}_{m-1} .

- On failure of these related to the closest Y_x subset element;

$$\bar{\phi}_m \in W_m^\pi$$

is used and the state recovered, based on this window, with no or zero metric being provided to the MAOC

Table 4.8 summarises the metric assignment adopted in the PSD. The windows are indicated by their associated width and the double windows are indicated by two values. Each row of the table contains the phase reception window range with its associated metric.

In all cases, except recovery mode, the previously obtained PSD \hat{St}_{m-1} is required for St_m recovery. As \hat{St}_{m-1} is assumed known (with a reliability indicated by its associated PSD metric) the following expected phase subset, Y_x , of the QPSK transmission is identifiable.

An example of the PSD metric calculation with a brief description of the enhancer operation is shown in Figure 4.18. In the example it is assumed that the recovered \hat{St}_{m-1} is 7 and the phases to state transitions are:

$$\begin{aligned} \hat{St}1 &\rightarrow \hat{St}3, & \bar{\phi}_{m-2} &= \frac{\pi}{2} \\ \hat{St}3 &\rightarrow \hat{St}7, & \bar{\phi}_{m-1} &= \pi \\ \hat{St}7 &\rightarrow St6, & \bar{\phi}_m &= 0 \\ St6 &\rightarrow St5, & \bar{\phi}_{m+1} &= \frac{5\pi}{8} \end{aligned}$$

In Figure 4.18 the states are represented by circles and the state transitions are represented by arrows. At the bottom of the figure the windows centred in the

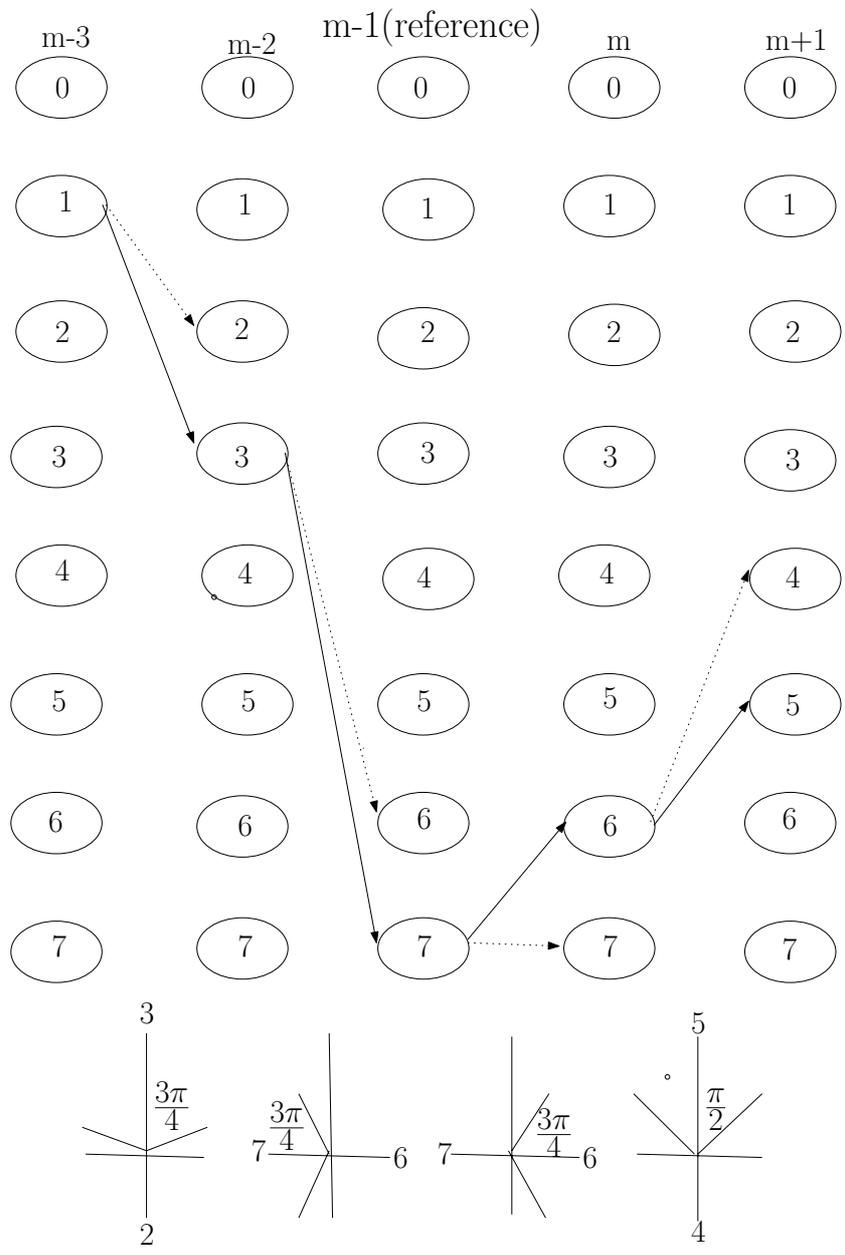


Figure 4.18: Metric allocation (optimum and sub-optimum - headings are related time references and columns are the states of a PSD sequence. The windows shown are the inter time reference phase windows)

expected phase are shown. The phase $\tilde{\phi}_{m+1}$ has been indicated by a \cdot in the phase transition between $St6 \rightarrow St5$.

The phase sequence;

$$\tilde{\phi}_{m-2}, \tilde{\phi}_{m-1}, \tilde{\phi}_m, \tilde{\phi}_{m+1}$$

fits into the set of windows

$$W_{m-2}^{\frac{3\pi}{4}}, W_{m-1}^{\frac{3\pi}{4}}, W_m^{\frac{3\pi}{4}}, W_{m+1}^{\frac{\pi}{2}+\frac{\pi}{2}}$$

as it is shown at the bottom of Figure 4.18 . Therefore the optimum, μ_0 , metric is allocated indicating that the sequence is recognised as

$$\hat{St}_{m-2} = 3, \hat{St}_{m-1} = 7, \text{ and } St_m = 6.$$

If $\tilde{\phi}_{m+1}$ is outside the acceptance window then a sub optimum, μ_1 , metric is allocated for St_6 .

Additionally $\tilde{\phi}_{m-1}$ has been enhanced; $\tilde{\phi}_{m-3}, \tilde{\phi}_{m-2}, \tilde{\phi}_{m-1}$ being within the $\frac{3\pi}{4}$ phase acceptance windows these being those predicted, at each stage, by the previous phase and state sequences. $\tilde{\phi}_m$ was enhanced as $\tilde{\phi}_{m-2}, \tilde{\phi}_{m-1}, \tilde{\phi}_m$ are within the $\frac{3\pi}{4}$ phase acceptance windows. The amount enhanced is then dependant on $\tilde{\phi}_{m+1}$. There are two possible results shown;

- $\tilde{\phi}_{m+1} \in W_{m+1}^{\frac{\pi}{2}+\frac{\pi}{2}}$, represented by the \cdot inside the window in Figure 4.18. In this case $\tilde{\phi}_m$ would have been advanced $\frac{\pi}{4}$ towards an optimum value or until $\tilde{\phi}_m$ was optimum.
- $\tilde{\phi}_{m+1} \notin W_{m+1}^{\frac{\pi}{2}+\frac{\pi}{2}}$. In this case $\tilde{\phi}_m$ would have been advanced $\frac{\pi}{8}$ towards an optimum value unless $\tilde{\phi}_m$ was already optimum.

These enhancements result in potentially improved transfer metrics for the MAOC and Viterbi decoders.

A further PSD example is presented in Figure 4.19 that considers some variations of the example in Figure 4.18. Now $\tilde{\phi}_{m-2} \in W_{m-2}^{\frac{3\pi}{4}}$ but $\tilde{\phi}_{m-1} \notin W_{m-1}^{\frac{3\pi}{4}}$, what has been indicated with a 'x' in the transition between time $m-2$ and $m-1$. A consequence of this is that the metric of the PSD can not be optimum or suboptimum. Algorithm 4.2 shows the possible states and metric values associated with different values of $\tilde{\phi}_m$, as indicated in Figure 4.19.

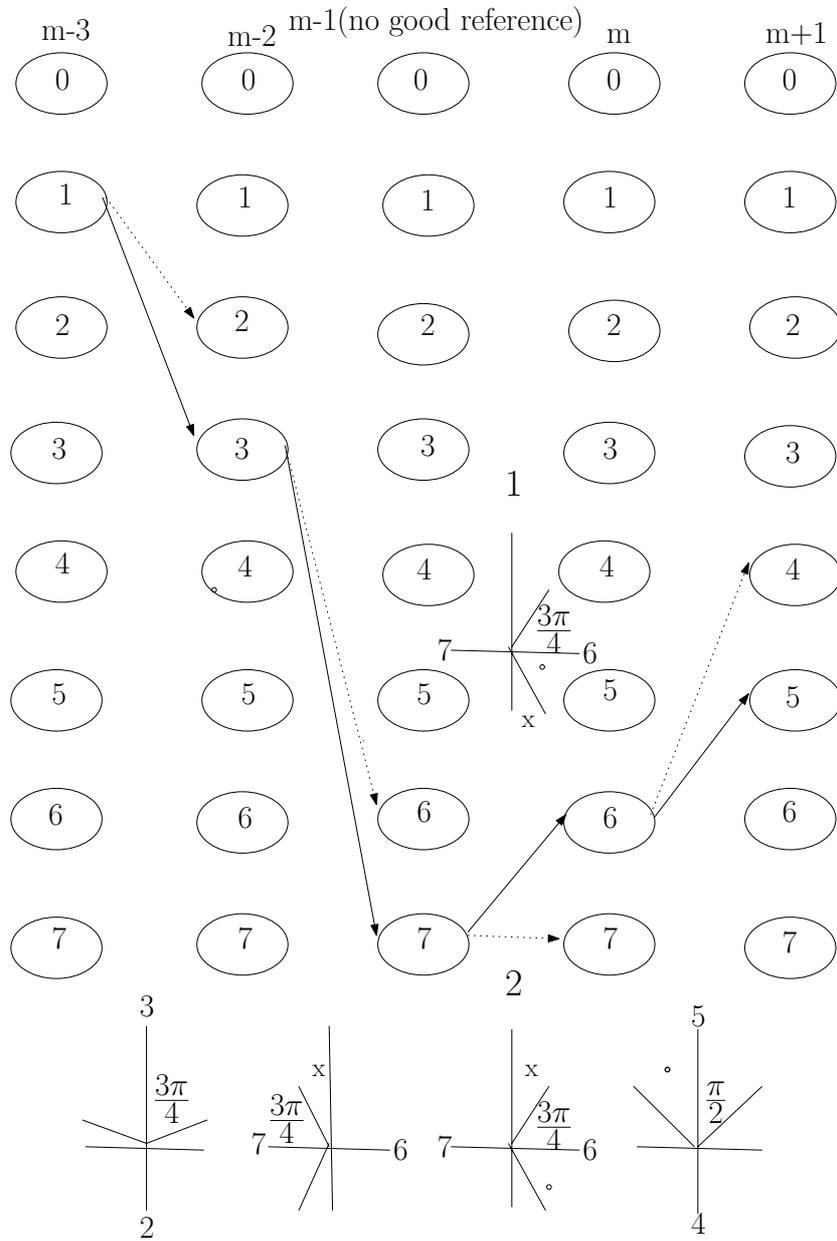


Figure 4.19: Metric allocation (non-optimum - headings are related time references and columns are the states of a PSD sequence. The windows shown are the inter time reference phase windows)

Algorithm 4.2 Metric values and states of the example of Figure 4.19

if $\tilde{\phi}_m$ is within the $\frac{\pi}{2}$ phase window **then**
 $\tilde{\phi}_m$ will be allocated St_6 with a μ_2 metric.
end if
if $\tilde{\phi}_m$ is outside the $\frac{\pi}{2}$ window but inside the $\frac{3\pi}{4}$ window **then**
 St_6 is allocated a μ_3 metric.
end if
if $\tilde{\phi}_m$ is outside the $\frac{3\pi}{4}$ phase window but within the π window **then**
 if $\tilde{\phi}_{m+1}$ is within the appropriate $\frac{\pi}{2}$ window **then**
 Allocate St_6 with a μ_4 metric.
 end if
end if
Allocate St_6 with no metric based on the closest $\tilde{\phi}_m$ value to an element of Y_x ,
the set of expected phase values.

Chapter 5

Sequence Code Simulation

Neither of the available Matlab or Entegra simulators provided the flexibility required for the upgraded sequence code simulation. It was, therefore, necessary to develop a technique specific simulator. This involved some implementation risk which was controlled, to the extent possible, by using, where available, standard routines. Additionally the developed simulator demonstrated that the BPSK symbol error rate closely match theoretical values. The simulators decoders operate over random error sequences within the BPSK/QPSK environment giving some justification to the simulation results.

To produce the random transmitter information required an acceptable random number generator. The simulation of Gaussian noise for the communication link required a Gaussian noise generator. A requirement of this Gaussian noise generator was that it could make unmodified use, as an input, of the same random number generator used for the generation of the information sequence.

A good random number generator should have a long period and the subsequences of pseudo random numbers should be uniform and uncorrelated [55]. Another important consideration, in view of the number of times this random number generator is used in the simulator on a P4 computer, is that the generator should be economic with the available resources. There were several options that met these requirements and a random number algorithm from [55] was chosen. This provided an acceptable, standard, random number generator in relatively few lines of efficient code. For an acceptable Gaussian random number generator, with the same limitations, the polar form of the Box Muller algorithm from the same source [55] was chosen. Both algorithms being mathematically robust and operationally efficient.

The integrated decoder pair, PSD and Viterbi, function as probabilistic or inference decoders based on the empirically derived metrics. These metrics allow some matching of the channel to the decoders and once optimized, to the link,

PSD Metrics			Viterbi Metrics					BER	No of Samples
μ_0	μ_1	μ_2	μ_0	μ_1	μ_2	μ_3	μ_4		
-9	-1	-1	-11	-9	-4	-2	0	$5.40 * 10^{-5}$	$2 * 10^6$
-9	-2	-1	-11	-8	-4	-2	0	$4.85 * 10^{-5}$	$2 * 10^6$
-9	-3	-1	-11	-8	-4	-2	0	$5.50 * 10^{-5}$	$2 * 10^6$
-2	-2	-1	-11	-8	-4	-2	0	$5.05 * 10^{-5}$	$2 * 10^6$
-2	-1	-1	-11	-8	-4	-2	0	$6.25 * 10^{-5}$	$2 * 10^6$
-11	-2	-1	-11	-8	-4	-2	0	$4.85 * 10^{-5}$	$2 * 10^6$
-9	-2	0	-11	-8	-4	-2	0	$4.10 * 10^{-5}$	$2 * 10^6$
-9	-2	0	-11	-8	-4	-1	0	$4.15 * 10^{-5}$	$2 * 10^6$
-9	-2	0	-11	-8	-3	-2	0	$4.10 * 10^{-5}$	$2 * 10^6$
-9	-2	0	-11	-8	-4	-3	0	$3.99 * 10^{-5}$	$2 * 10^6$

Table 5.1: Early PSD and Viterbi metrics allocation.

were found to be almost invariant to SNR. In addition optimisation of all items, including **C** or **D** receiver choices Figure 4.7, Viterbi decoder windows, the PSD, the enhancement routine and the metrics were performed with the simulator.

5.1 Metric Derivation and Unit Optimisation

The metric values for sequence codes are derived empirically based on the angular separation of $\tilde{\phi}_m$ from the closest predicted Y_m phase subset element. Relatively small changes of metrics can have substantial effects on the results. This is demonstrated in some early coder metric tests (3 metrics phase sequence 5 metrics Viterbi decoders) in Table 5.1.

The metric values are obtained iteratively using repeated runs of the simulator to obtain the best obtainable BER results by upgrading preset metric values being usually initialised at -2 . The metrics are obtained at low SNR to reduce the number of samples required to provide sufficient errors. Testing the resultant metric at higher SNR suggested these BER results, though not the best obtainable, would be acceptably throughout the SNR range.

The enhancement routine, within the phase sequence decoder, was examined being the first action taken by the sequence decoder simulator on the received quantised data. The system was tested with and without the enhancement routine and the Sequence code and RS errors noted in Table 5.2.

Based on these results the enhancement routine was included.

Tested Unit	Sequence Code Errors	Bit Errors After RS
With Enhancement	2205	195
Without Enhancement	2505	330

Table 5.2: Enhancement test results

Tested Unit	Sequence Code Errors	Bit Errors After RS
Receiver C	1800	165
Receiver D	2205	195

Table 5.3: quantiser test results

The choice of phase quantiser, Figure 4.7, was examined. Based on the results shown for sequence code errors and RS bit errors in table 5.3 the **C** version of the receiver was included. This surprised as the **D** receiver, with its potential for a wider overall interval, would have been expected to provide better results. It is intended that work be continued on this element as future work.

In Table 5.4 the phrases *SDecode* and *Enhance* represent the two phase window sets used in the PSD. In table 5.5 *SDecode* represents the PSD and *Window* the Viterbi window.

The resulting best phase window sets were obtained by simulation. For the PSD the intended set of acceptance windows is:

$$W_{m-2}^{\varphi_{m-2}}, W_{m-1}^{\varphi_{m-1}}, W_m^{\varphi_m}$$

where the length of the phase windows;

$$(\varphi_{m-2}, \varphi_{m-1}, \varphi_m)$$

are varied through the potential quantisation range to obtain the best available error figure. The same optimisation tests are performed with the three phase window set of the enhancer. The results of the window simulations and the resultant error figures are shown in Table 5.4. In this table the first column gives the interval of phase windows for the decoder and next column the intervals of the phase windows for the enhancer. The columns represent the metric values (μ_x) for the PSD and Viterbi decoders where x indicates the metric type with 0 representing the optimum metric, 1 the sub-optimum metric etc..

The RS, and Code columns indicate the errors of the RS code and the received symbol errors respectively and the best initial results are provided in Table 5.4. To

SDecode	Enhance	PSD Metric					Viterbi Metric					Code	RS
		μ_0	μ_1	μ_2	μ_3	μ_4	μ_0	μ_1	μ_2	μ_3	μ_4		
$\frac{\pi}{2} \frac{\pi}{2} \frac{\pi}{2}$	$\frac{\pi}{2} \frac{\pi}{2} \frac{\pi}{2}$	-8	-3	2	-1	-3	-6	-4	-3	-3	-1	10405	874
$\frac{\pi}{2} \frac{\pi}{2} \frac{3\pi}{4}$	$\frac{3\pi}{4} \frac{3\pi}{4} \frac{\pi}{2}$	-8	-3	1	-1	-3	-6	-4	-3	-3	-1	9471	517
$\frac{\pi}{2} \frac{3\pi}{4} \frac{3\pi}{4}$	$\frac{3\pi}{4} \frac{3\pi}{4} \frac{\pi}{2}$	-8	-2	2	-1	-3	-6	-4	-3	-3	-1	9856	596
$\frac{3\pi}{4} \frac{3\pi}{4} \frac{\pi}{2}$	$\frac{3\pi}{4} \frac{3\pi}{4} \frac{\pi}{2}$	-5	-5	2	-1	-3	-6	-4	-3	-3	-1	10632	867
$\frac{3\pi}{4} \frac{4}{4} \frac{2}{4}$	$\frac{3\pi}{4} \frac{3\pi}{4} \frac{\pi}{2}$	-8	-3	2	-1	-3	-6	-4	-3	-3	-1	10432	859
$\frac{3\pi}{4} \frac{2}{4} \frac{2}{4}$	$\frac{3\pi}{4} \frac{3\pi}{4} \frac{\pi}{2}$	-5	-5	2	2	-3	-6	-4	-3	-3	-1	9565	457
$\frac{3\pi}{4} \frac{3\pi}{4} \frac{3\pi}{4}$	$\frac{3\pi}{4} \frac{3\pi}{4} \frac{\pi}{2}$	-5	-5	2	2	-3	-6	-4	-3	-3	-1	9565	457

Table 5.4: Initial metric test results.

SDecode	Window	PSD Metric					Viterbi Metric					Code	RS
		μ_0	μ_1	μ_2	μ_3	μ_4	μ_0	μ_1	μ_2	μ_3	μ_4		
$\frac{3\pi}{4} \frac{3\pi}{4} \frac{3\pi}{4}$	5	-7	-5	2	2	-3	-6	-3	-2	-3	1	10021	599
$\frac{3\pi}{4} \frac{3\pi}{4} \frac{3\pi}{4}$	6	-7	-3	2	2	-3	-6	-3	-2	-3	-1	9873	535
$\frac{3\pi}{4} \frac{3\pi}{4} \frac{3\pi}{4}$	6	-5	-5	2	2	-3	-6	-4	-3	-3	-1	9565	457

Table 5.5: Final test results Viterbi window lengths 5 and 6.

obtain these metrics the data sequence used was the same in each case - the same initial seed being provided to the random number generators at each iteration. The length of the Viterbi window was 6.

The metrics obtained providing the best RS results, though not optimum over the whole SNR range, were used. More work is needed on this topic. The PSD metrics μ_2 and μ_3 appear to be out of order with μ_4 suggesting that the PSD metrics have not been properly ordered. This was studied and the order giving the best result is that provided. The tests were performed with Viterbi decoder window lengths of five to six frames. The results are given in Table 5.5. The system used the $(\frac{3\pi}{4}, \frac{3\pi}{4}, \frac{3\pi}{4})$ PSD window lengths with $(\frac{3\pi}{4}, \frac{3\pi}{4}, \frac{\pi}{2})$ enhancement lengths with a Viterbi decoder window length of 6.

5.2 Implementation of Phase Sequence Decoder

As part of the PSD design and implementation a study was conducted in order know the influence of its elements on the error probability [71].

The enhancement procedure, described in Section 4.4.2 , potentially improves $\tilde{\phi}_m$ resulting in improved PSD and Viterbi decoder metrics.

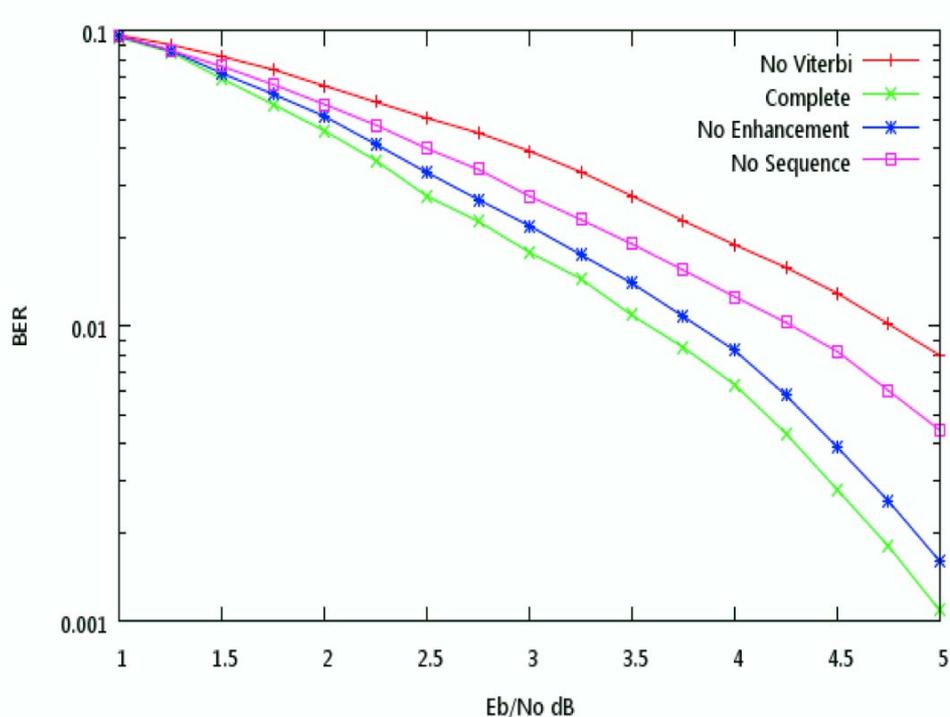


Figure 5.1: Results without individual sequence coder elements

These metric improvements reduce the effect of poor phase symbols within the decoders by biasing the decoders with improved metrics prior to any poorer quality phase symbols. Figure 5.1 shows the effect of removing individual elements from the sequence decoder and the benefit of including the enhancement element.

The PSD state, St_m , solution with its related metric are allocated to the first frame of the Viterbi decoder window as $\tilde{\phi}_m$ passes into the Viterbi decoder, Figure 4.15. The same PSD information is passed to the MAOC and stored in a buffer until the end of the Viterbi decoder window. It can then be used as a solution in the event that the Viterbi decoders solution proves not to fit the sequence; Section 5.4.

5.3 Implementation of the Viterbi Decoder.

The implementation of the Viterbi decoder is similar to the implementation of a classical Viterbi decoder. There are, however, some differences introduced due to the lack of traceback requirement. The main differences are the receipt of the PSD

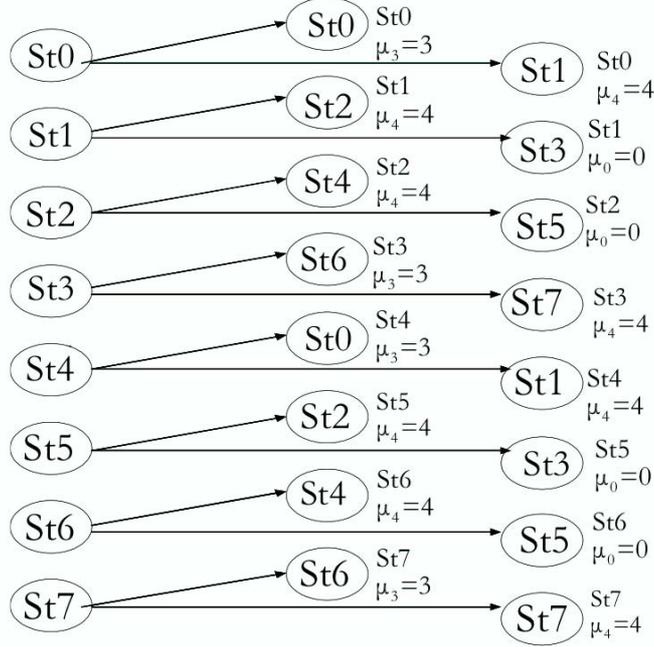


Figure 5.2: Viterbi first frame with expansion to second frame. The received quantised phase is $\frac{7\pi}{16}$ and the amplitude of the corresponding signal $|z_m| = 1$

metric in a single state into the first frame of the Viterbi decoder window and the algorithm itself is implemented with the minimum of data transfer between frames.

At each frame of the Viterbi decoder window the algorithm allocates one of, currently, up to 5 (tested to 6) metrics each related to the probability that the received quantised enhanced phase $\tilde{\phi}_m$ can be correctly assigned to a transition from $\hat{S}t_{m-1} \rightarrow St_m$, Fig 5.2. This probability relates to the separation of the quantised enhanced phase $\tilde{\phi}_m$ from that predicted by a continuum from the previously accepted $\tilde{\phi}_{m-1}$ and state $\hat{S}t_{m-1}$.

For each maintained route through the Viterbi decoder window the integrated decoders provide, via the Viterbi decoder, an overall summation metric, μ_{sum} , made up by the Viterbi and phase sequence decoders summation path metric

$$\mu_{sum}^{(m)} = \mu_{s0}^{(m)} + \mu_{v0}^{(m)} |z_m| + \mu_{v1}^{(m+1)} |z_{m+1}| + \dots + \mu_{vw-1}^{(m+w-1)} |z_{m+w-1}|, \quad (5.1)$$

where $\mu_{s0}^{(m)}$ is the PSD metric multiplied by the related magnitude of $|z_m|$, Figure 4.6, provided to the first frame of the Viterbi decoder window and the MAOC,

$\mu_{vi}^{(m+i)}$, $i \in \{0, w - 1\}$, are the individual frame Viterbi decoder metrics, w is the Viterbi decoder window length and $\mu_{sum}^{(m)}$ is the metric summation over the Viterbi decoder window. A minimum value represents the highest probability that a state (the original state of the Viterbi window) is valid. Hereafter the time superscript indices of eq. (5.1) are dropped in order to clarify the presentation.

The Viterbi decoding procedure updates a table, at each frame $m + 3$ and above, Figure 4.17, containing the following parameters:

- Current state (64 values, 8 for each of 8 original states).
- Start state (64 same values attached, 8 for each of 8 original states).
- Current μ_{sum} (64 Values 8 associated with each 8 original states)

This information is passed, at the end of each Viterbi window, to the MAOC which performs the correction of the received data to bit level.

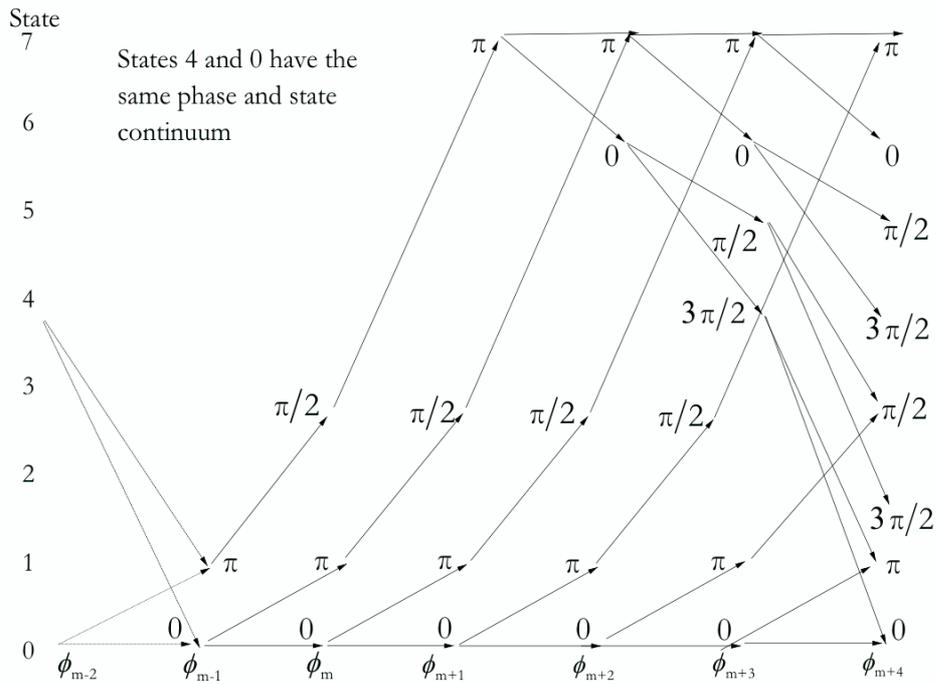


Figure 5.3: Example of the Viterbi dual routing problem

Figure 5.3 shows the Viterbi decoder routing from $\hat{S}t4$ and $\hat{S}t0$. The two continuums illustrate the equivalence of the sequences from the two state groups

shown in Table 5.6. The metric allocation to these two state continuums would, in the Viterbi decoder, be the same resulting in a dual routing shown in Tables 5.6 & 5.7. The PSD provides a single metric and associated state to the first frame of the Viterbi decoder allowing the MAOC to differentiate between the dual start states by biasing the summation metric, at each Viterbi decoder window initialisation, towards the PSD solution.

Pair 1
From $St3 \rightarrow 0$ radians to $St6$ or π radians to $St7$
From $St7 \rightarrow 0$ radians to $St6$ or π radians to $St7$
Pair 2
From $St1 \rightarrow \frac{3\pi}{2}$ radians to $St2$ or $\frac{\pi}{2}$ radians to $St3$
From $St5 \rightarrow \frac{3\pi}{2}$ radians to $St2$ or $\frac{\pi}{2}$ radians to $St3$
Pair 3
From $St2 \rightarrow \frac{3\pi}{2}$ radians to $St4$ or $\frac{\pi}{2}$ radians to $St5$
From $St6 \rightarrow \frac{3\pi}{2}$ radians to $St4$ or $\frac{\pi}{2}$ radians to $St5$
Pair 4
From $St4 \rightarrow 0$ radians to $St0$ or π radians to $St1$
From $St0 \rightarrow 0$ radians to $St0$ or π radians to $St1$

Table 5.6: Viterbi decoder dual routing problem presentation

<i>State 3</i> \leftrightarrow <i>State 7</i>
<i>State 1</i> \leftrightarrow <i>State 5</i>
<i>State 2</i> \leftrightarrow <i>State 6</i>
<i>State 4</i> \leftrightarrow <i>State 0</i>

Table 5.7: Viterbi dual routing states

5.4 MAOC.

On initialization the MAOC enters a series of steps to recover one or more minimum summation metrics, μ_{sum_m} , each with an associated original state $St_{m_o r}$. At the initiation of a transmission the MAOC knows the non-tested, previously recovered state is zero; the registers being initially preset to zero. In this case the first recovered state is either 0 or 1 dependant on the initial data bit into the phase sequence encoder.

On initiation of a transmission, the Viterbi decoder window length-1 recovered states and metrics are ignored. When the first Viterbi decoder window of a transmission is completed all the summation metrics, with associated original states for that window, are transferred to the MAOC.

The MAOC examines all 64 Viterbi decoder window summation metrics for a, or more than one, minimum summation metric, μ_{sum_m} . This examination is performed at each Viterbi window termination. The recovered minimum summation metric, or metrics, are then stored, with associated state or states, for the second stage of correction.

Figure 5.4 presents the MAOC logic; Assume the MAOC finds a single minimum summation metric, μ_{sum_m} , associated with a state \hat{St}_m . The related bit value for s_m can then be derived directly from this state \hat{St}_m . The probability of this bit value being correct is indicated by the associated summation metric.

Poor enhanced quantised phases, $\tilde{\phi}_i$, either singly or in groups, can produce errors in all Viterbi decoding windows that include this corrupted phase element or elements within their window. The resultant of individual phase to state transitions within the Viterbi decoder window can result in erroneous minimum metric summation values.

The MAOC checks the sequence continuity of the Viterbi decoders state output. If the recovered state, associated with the minimum metric summation, does not fit the existing state sequence the MAOC enters a test routine. The single state, \hat{St}_m , derived by the PSD is tried. If this state is a continuation of the state $(\dots, \hat{St}_{m-2}, \hat{St}_{m-1}, \hat{St}_m)$ sequence this value becomes the solution. This state, derived from the PSD solution, is entered into the continuous state sequence, potentially correcting or, at least, reducing the length of an erroneous state sequence.

If these options, Viterbi and PSD, fail to produce an acceptable state continuum within the sequence the Viterbi decoders dual phase routing error possibility is considered. This can occur when the corrupted $\tilde{\phi}_m$ is at the start of the Viterbi decoders sequence affecting both the PSD and Viterbi decoder results. The Viterbi decoder might also be affected by further corrupted quantized phase symbols within the window. In this case the Viterbi decoders dual state routing, Tables 5.6 and 5.7, alternative to the received state will be considered. For example, if the minimum metric analysis gives a $St3$ then $St7$ could be the solution. If this also fails the previous state is repeated and the PSD attempts a rapid recovery. This results in short error bursts, Figure 5.5, but provides an indication of error which could be forwarded to the outer RS decoder. This possibility has been left to future work.

If there is more than a single minimum metric summation the initial tests remain the same. All \hat{St}_m values associated with the multiple minimum metric

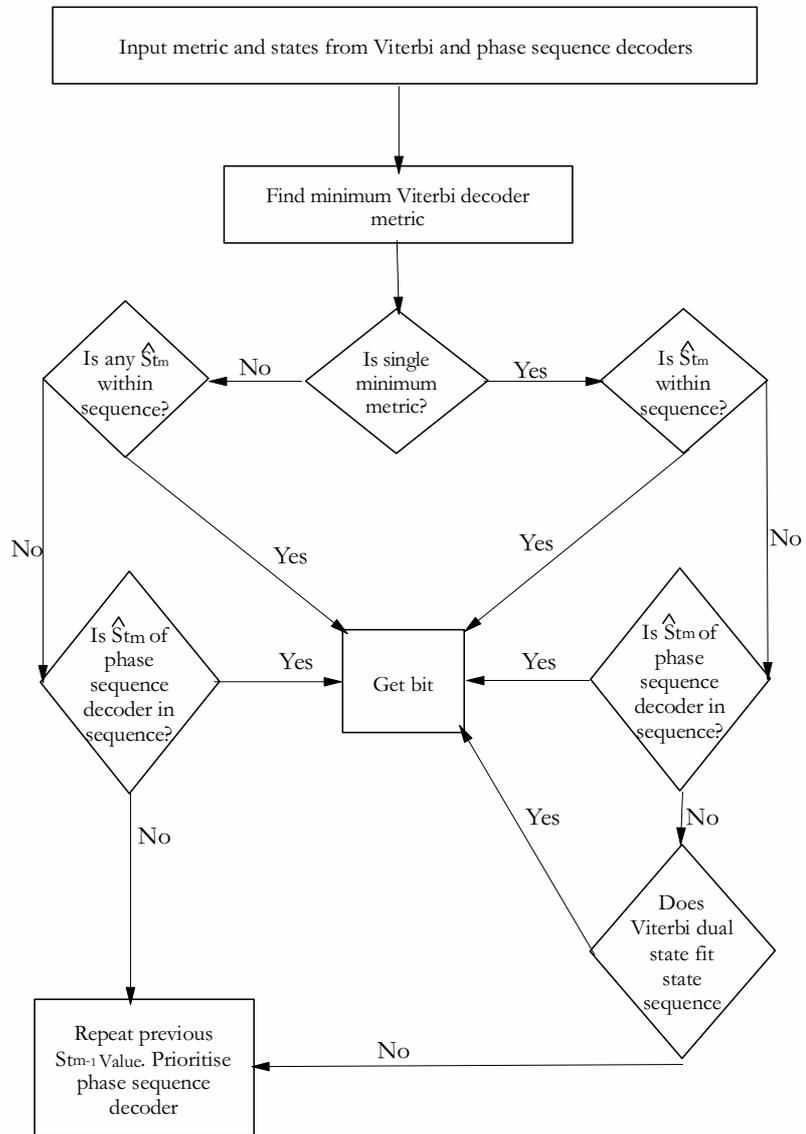


Figure 5.4: MAOC logic

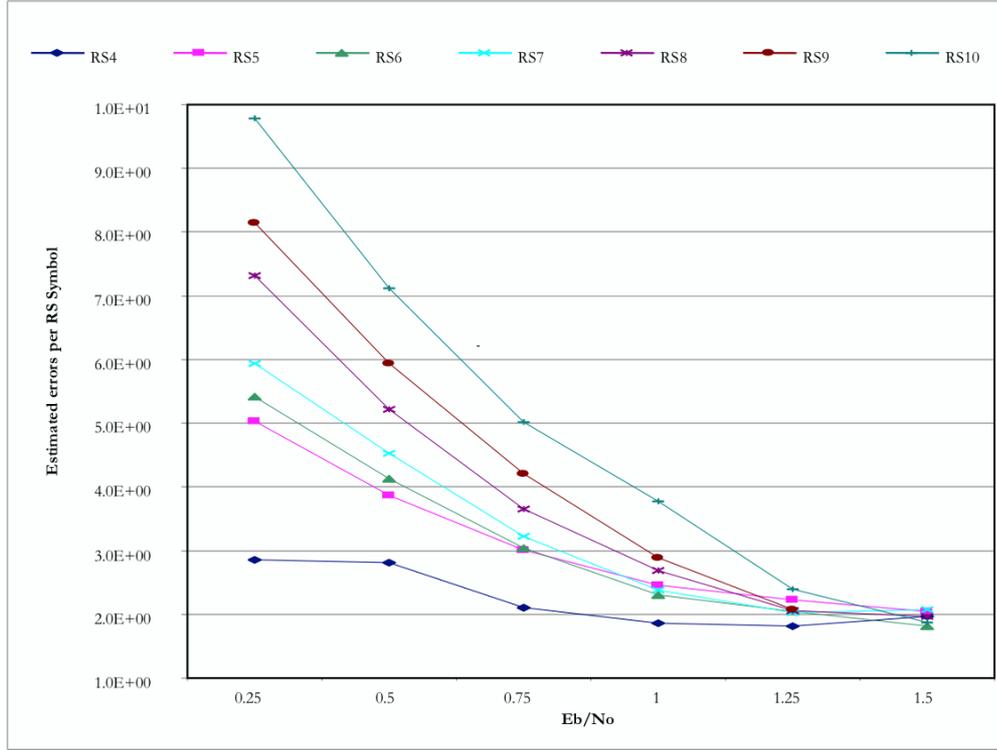


Figure 5.5: Average sequence code error burst per RS symbol (Linear $\frac{E_b}{N_0}$ Scaling)

summations are tested for a possible fit within the current state sequence. Only two state values will fit the sequence continuum. These two state values, being separated by π radians, cannot have the same metric. If none of the states associated with a minimum metric summation are valid the PSD solution is then tried. If this single state value fits the state sequence it is used and entered into the continuing state sequence of the MAOC. The extension to Viterbi decoders dual state routing has not been used as this proved, on the rare times it was used, liable to extend the error bursts.

In the event of coding loss, recovery is achieved by prioritising the PSD results noting the received phase symbol sequence until the Viterbi decoder results are confirmed. In this case, for example, receiving a phase within the $\frac{\pi}{2}$ radian acceptance interval would suggest a state. Receiving a decoded quantised phase sequence of $\tilde{\phi}_m = \pi, \tilde{\phi}_{m-1} = 0, \tilde{\phi}_{m-2} = \frac{\pi}{2}$ in Figure 4.3 would indicate a high probability of *St5* which is sent to the now reinstated Viterbi decoder. The extension of Viterbi decoders dual state routing to multiple μ_{sum_m} is left to future work. The Figures

used for analysing this data are 4.2 and 4.7.

5.5 Worked Example

The following worked example is structured from the output of a 3.5 dB $\frac{E_b}{N_0}$ program run from the beginning of a data sequence. In Table 5.8 are shown the results obtained by this decoder using the first seven received symbols.

Table 5.8: PSD decoder results

	Symb1	Symb2	Symb3	Symb4	Symb5	Symb6	Symb7
Original bit	0	1	0	0	1	1	1
Original St	0	1	3	6	5	3	7
ϕ_m	0	π	$\frac{\pi}{2}$	0	$\frac{\pi}{2}$	$\frac{\pi}{2}$	π
$\tilde{\phi}_m$	$\frac{29\pi}{16}$	$\frac{15\pi}{16}$	$\frac{7\pi}{16}$	$\frac{29\pi}{16}$	$\frac{7\pi}{16}$	$\frac{9\pi}{16}$	$\frac{17\pi}{16}$
$\bar{\phi}_m$	$\frac{\pi}{16}$	$\frac{17\pi}{16}$	$\frac{9\pi}{16}$	$\frac{\pi}{16}$	$\frac{7\pi}{16}$	$\frac{9\pi}{16}$	$\frac{17\pi}{16}$
$ z_i $	1.3629	1.1587	1.2855	0.6281	1.1668	0.9014	1.0675
μ_n PSD	-11.93	-10.14	-11.25	-5.496	-10.21	-7.887	-9.341
Sequence	400	001	013	136	365	653	537
Derived St	0	1	3	6	5	3	7

The variables presented in Table 5.8 for each symbol are;

Original bit: The random data bit into the encoder. This element is not recovered by the sequence decoder. Bit recovery is accomplished by the MAOC.

Original St: The element recovered by the phase sequence and Viterbi decoders

ϕ_m : The transmitted representation of the State (St).

$\tilde{\phi}_m$: The original recovered quantised phase element.

$\bar{\phi}_m$: The enhanced quantised phase value.

$|z_i|$: The modulus of the received sample (figure 4.6).

μ_n **PSD:** Metric values of PSD.

Sequence: Some test information being the last two recovered states with the current recovered state.

Derived St: The state value recovered by the PSD. The related metric value with this state is sent to the relevant first frame of the Viterbi decoder window and the MAOC. This metric value is added to the related state within this Viterbi decoding window to indicate the probable correct state of the dual phase routings recovered by the Viterbi decoder.

The PSD results presented in Table 5.8 shows the steps related to the recovery of individual metric values related to individual states on an individual phase symbol basis. To the transmitted ϕ_m , the medium through which the signal passes, adds noise resulting in a received $\tilde{\phi}_m$ which is improved by the enhancer routine to $\bar{\phi}_m$. Using $\bar{\phi}_m$ and the previously recovered state St_{m-1} the values of $|z_i|$ and the μ_n for the PSD are recovered. The sequence row shows the value of the recovered state sequence and then the derived state being the last row of the table. The state value and the μ_n for the PSD are sent to the first frame of the Viterbi decoders window and the MAOC.

The data structure from the Viterbi decoder windows is the μ_{sum} - summation metric and associated state St - the recovered Viterbi state. At the end of each Viterbi decoder window the Viterbi decoder transfers all 64 states (8 associated with each of the 8 original states) with their associated metrics to the MAOC. The state sequence recovered using the metrics provided by the Viterbi and PSD is used by the MAOC to recover the bit sequence. The minimum summation metric, μ_{summ} , is then discovered by the MAOC and is presented in enhanced text in each of the provided Viterbi output windows.

Considering the Viterbi dual phase routing problem of Table 5.6 the minimum metric related to state 1 (-59.804131) in window 2, differs from the alternative phase routing metric related to state 5 (-49.665497) by the PSD metric (shown in approximation as μ_n in Table 5.8) that was applied to frame 0 of Viterbi window 2. The bit sequence recovered by the MAOC is shown in "Recovered bit" at each of the Viterbi decoder windows of tables 5.9 to 5.15

Table 5.9: Viterbi decoder window 1

Metric Values μ_{sum} and associated initial states St_m							
μ_{sum}	St_m	μ_{sum}	St_m	μ_{sum}	St_m	μ_{sum}	St_m
-31.251997	7	-31.241997	7	-27.901739	7	-33.986382	7
-26.758766	7	-32.843407	7	-29.102596	7	-20.764383	7
-22.975163	6	-37.374054	6	-29.035839	6	-16.890520	6
-17.199406	6	-23.284046	6	-34.765862	6	-26.427649	6
-32.761345	5	-24.423134	5	-20.328341	5	-26.412983	5
-29.235603	5	-35.320244	5	-28.682682	5	-20.344469	5
-52.036240	4	-43.698029	4	-31.552711	4	-37.637352	4
-27.942390	4	-34.027031	4	-41.558632	4	-33.220421	4
-39.580208	3	-31.241997	3	-27.901739	3	-33.986382	3
-26.758766	3	-32.843407	3	-29.102596	3	-20.764383	3
-37.374054	2	-29.035839	2	-16.89052	2	-22.975163	2
-17.199406	2	-23.284046	2	-34.765862	2	-26.427649	2
-32.761345	1	-24.423134	1	-20.328341	1	-26.412983	1
-29.235603	1	-35.320244	1	-28.682682	1	20.344469	1
-63.970581	0	-55.632370	0	-43.487053	0	-49.571693	0
-39.876732	0	-45.961372	0	-53.492973	0	-45.154762	0
Derived Results							
Minimum summation metric μ_{sum_m}						-63.970581	
Associated Viterbi state St						0	
Original Bit						0	
Recovered bit						0	

Table 5.10: Viterbi decoder window 2

Metric Values μ_{sum} and associated initial states St_m							
μ_{sum}	St_m	μ_{sum}	St_m	μ_{sum}	St_m	μ_{sum}	St_m
-29.512600	7	-27.334333	7	-18.996159	7	-26.202332	7
-15.255349	7	-22.461525	7	-31.615633	7	-21.740505	7
-35.846298	6	-25.971170	6	-26.928696	6	-34.134869	6
-20.291134	6	-27.497307	6	-29.497934	6	-19.622807	6
-49.665497	5	-39.790371	5	-29.850784	5	-37.056957	5
-27.711384	5	-34.917557	5	-35.266609	5	-25.391479	5
-38.278191	4	-28.403063	4	-18.463478	4	-25.669651	4
-26.374313	4	-33.580486	4	-29.512600	4	-19.637472	4
-37.209461	3	-27.334333	3	-18.996159	3	-26.202332	3
-15.255349	3	-22.461525	3	-31.615633	3	-21.740505	3
-35.846298	2	-25.971170	2	-26.928696	2	-34.134869	2
-20.291134	2	-27.497307	2	-29.497934	2	-19.622807	2
-59.804131	1	-49.929005	1	-39.989418	1	-47.195591	1
-37.850018	1	-45.056190	1	-45.405243	1	-35.530113	1
-38.278191	0	-28.403063	0	-18.463478	0	-25.669651	0
-26.374313	0	-33.580486	0	-29.512600	0	-19.637472	0
Derived Results							
Minimum summation metric μ_{sum_m}						-59.804131	
Associated Viterbi state St						1	
Original Bit						1	
Recovered bit						1	

Table 5.11: Viterbi decoder window 3

Metric Values μ_{sum} and associated initial states St_m							
μ_{sum}	St_m	μ_{sum}	St_m	μ_{sum}	St_m	μ_{sum}	St_m
-32.626149	7	-47.098202	7	-33.871918	7	-28.216843	7
-29.348156	7	-23.693081	7	-26.740114	7	-34.489662	7
-21.663206	6	-29.412752	6	-19.977016	6	-14.321940	6
-30.331985	6	-24.676910	6	-20.520292	6	-28.269838	6
-25.529455	5	-33.27900	5	-21.028959	5	-15.373883	5
-23.579483	5	-17.924408	5	-23.818026	5	-31.567572	5
-32.596149	4	-40.345699	4	-32.296841	4	-26.641768	4
-28.228954	4	-22.573879	4	-19.987612	4	-27.737158	4
-50.596836	3	-58.346386	3	-45.120102	3	-39.465027	3
-40.596340	3	-34.941265	3	-37.988297	3	-45.737846	3
-21.663206	2	-29.412752	2	-19.977016	2	-14.321940	2
-30.331985	2	-24.676910	2	-20.520292	2	-28.269838	2
-25.529455	1	-33.279003	1	-21.028959	1	-15.373883	1
-23.579483	1	-17.924408	1	-23.818026	1	-31.567572	1
-32.596149	0	-40.345699	0	-32.296841	0	-26.641768	0
-28.228954	0	-22.573879	0	-19.987612	0	-27.737158	0
Derived Results							
Minimum summation metric μ_{sum_m}						-58.346386	
Associated Viterbi state St						3	
Original Bit						1	
Recovered bit						1	

Table 5.12: Viterbi decoder window 4

Metric Values μ_{sum} and associated initial states St_m							
μ_{sum}	St_m	μ_{sum}	St_m	μ_{sum}	St_m	μ_{sum}	St_m
-32.322547	7	-24.478367	7	-36.693050	7	-22.723446	7
-30.757820	7	-16.788216	7	-22.272421	7	-32.466457	7
-30.407692	6	-40.601727	6	-54.392189	6	-40.422585	6
-41.165905	6	-27.196299	6	-25.548700	6	-35.742737	6
-21.036833	5	-31.230869	5	-36.352547	5	-22.382940	5
-27.460087	5	-13.490482	5	-18.974688	5	-29.168724	5
-18.158972	4	-28.353008	4	-42.143463	4	-28.173859	4
-34.094608	4	-20.125002	4	-13.299978	4	-23.494015	4
-14.284330	3	-24.478367	3	-36.693050	3	-22.723446	3
-30.757820	3	-16.788216	3	-22.272421	3	-32.466457	3
-24.911472	2	-35.105507	2	-48.895966	2	-34.926361	2
-35.669682	2	-21.700077	2	-20.052481	2	-30.246517	2
-21.036833	1	-31.230869	1	-36.352547	1	-22.382940	1
-27.460087	1	-13.490482	1	-18.97468	1	-29.168724	1
-18.158972	0	-28.353008	0	-42.143463	0	-28.173859	0
-34.094608	0	-20.125002	0	-13.299978	0	-23.494015	0
Derived Results							
Minimum summation metric μ_{sum_m}						-54.392189	
Associated Viterbi state St						6	
Original Bit						0	
Recovered bit						0	

Table 5.13: Viterbi decoder window 5

Metric Values μ_{sum} and associated initial states St_m							
μ_{sum}	St_m	μ_{sum}	St_m	μ_{sum}	St_m	μ_{sum}	St_m
-32.372181	7	-22.959070	7	-25.872196	7	-28.340504	7
-30.993872	7	-33.462181	7	-18.947956	7	-20.749155	7
-27.511440	6	-29.312639	6	-20.056479	6	-22.524786	6
-33.846939	6	-36.315247	6	-29.220078	6	-31.021276	6
-40.120502	5	-41.921700	5	-39.956596	5	-42.424904	5
-53.747059	5	-56.215366	5	-39.377186	5	-41.178383	5
-30.848225	4	-32.649422	4	21.139694	4	23.608002	4
-27.393473	4	-29.861780	4	-19.152239	4	-20.953438	4
-21.157871	3	-22.959070	3	-25.872196	3	-28.340504	3
-30.993872	3	-33.462181	3	-18.947956	3	-20.749155	3
-27.51144	2	-29.312639	2	-20.056479	2	-22.524786	2
-33.846939	2	-36.315247	2	-29.220078	2	-31.021276	2
-29.910744	1	-31.711943	1	-29.746838	1	-32.215145	1
-43.537296	1	-46.005604	1	-29.167421	1	-30.968620	1
-30.848225	0	-32.649422	0	-21.139694	0	-23.608002	0
-27.393473	0	-29.861780	0	-19.152239	0	-20.953438	0
Derived Results							
Minimum summation metric μ_{sum_m}						-56.215366	
Associated Viterbi state St						5	
Original Bit						1	
Recovered bit						1	

Table 5.14: Viterbi decoder window 6

Metric Values μ_{sum} and associated initial states St_m							
μ_{sum}	St_m	μ_{sum}	St_m	μ_{sum}	St_m	μ_{sum}	St_m
-26.461873	7	-39.877892	7	-34.897453	7	-48.021431	7
-18.638687	7	-31.762665	7	-18.978781	7	-28.555738	7
-17.688988	6	-27.265945	6	-22.285509	6	-35.409485	6
-19.431366	6	-32.555344	6	-19.771460	6	-29.348417	6
-20.610577	5	-30.187534	5	-25.207098	5	-38.331078	5
-18.111929	5	-31.235907	5	-17.784914	5	-27.361872	5
-22.279184	4	-31.856142	4	-26.208597	4	-39.332573	4
-14.698865	4	-27.822844	4	-15.038960	4	-24.615917	4
-38.188435	3	-47.765388	3	-42.784954	3	-55.908932	3
-26.526188	3	-39.650166	3	-26.866282	3	-36.443237	3
-17.688988	2	-27.265945	2	-22.285509	2	-35.409485	2
-19.431366	2	-32.555344	2	-19.771460	2	-29.348417	2
-20.610577	1	-30.187534	1	-25.207098	1	-38.331078	1
-18.111929	1	-31.235907	1	-17.784914	1	-27.361872	1
-22.279184	0	-31.856142	0	-26.208597	0	-39.332573	0
-14.698865	0	-27.822844	0	-15.038960	0	-24.615917	0
Derived Results							
Minimum summation metric μ_{sum_m}						-55.908932	
Associated Viterbi state St						3	
Original Bit						1	
Recovered bit						1	

Table 5.15: Viterbi decoder window 7

Metric Values μ_{sum} and associated initial states St_m							
μ_{sum}	St_m	μ_{sum}	St_m	μ_{sum}	St_m	μ_{sum}	St_m
-53.019005	7	-44.639843	7	-48.112297	7	-37.810955	7
-43.131863	7	-32.830521	7	-48.243683	7	-62.360340	7
-12.598558	6	-26.715214	6	-33.734695	6	-23.433350	6
-20.703274	6	-10.401932	6	-25.815098	6	-39.931755	6
-13.160099	5	27.276756	5	-30.749212	5	-20.447868	5
-25.101667	5	-14.800323	5	-30.213491	5	-44.330147	5
-13.917994	4	-28.034649	4	-24.752785	4	-35.054131	4
-24.784294	4	-14.482950	4	-29.896116	4	-44.012772	4
-21.181849	3	-35.298504	3	-38.770962	3	-28.469618	3
-33.790527	3	-23.489182	3	-38.902348	3	-53.019005	3
-12.598558	2	-26.715214	2	-33.734695	2	-23.433350	2
-20.703274	2	-10.401932	2	-25.815098	2	-39.931755	2
-13.160099	1	-27.276756	1	-30.749212	1	-20.447868	1
-25.101667	1	-14.800323	1	-30.213491	1	-44.330147	1
-13.917994	0	-28.034649	0	-35.054131	0	-24.752785	0
-24.784294	0	-14.482950	0	-29.896116	0	-44.012772	0
Derived Results							
Minimum summation metric μ_{sum_m}						-62.360340	
Associated Viterbi state St						7	
Original Bit						1	
Recovered bit						1	

5.6 Error Probability Bound

Obtaining a closed form of the bit error probability is problematical. Instead a bound of the error probability, based on the work of Giuseppe Caire et al. [27], has been obtained.

A vector representation has been adopted for the signals and noise in the modulation and demodulation making things easier. Vector \mathbf{c} represents the modulated signal $g(t)$ in Figure 4.1, \mathbf{y} represents $z(t)$ in the same figure and \mathbf{n} is the noise having a circular Gaussian distribution. The sequence of bits to be transmitted is stacked in a vector \mathbf{b} , hence $\mathbf{c} = f(\mathbf{b})$.

The error probability bound is usually obtained through the union bound [56] involving binary decisions. Assuming that the information \mathbf{b} was input to the modulator, the resultant modulated signal is \mathbf{c} and another possible modulated signal is \mathbf{d}' . Considering only the signal vectors \mathbf{c} and \mathbf{d}' the error region with ML detection, $\Gamma_{\mathbf{c}\mathbf{d}'}$, is the set of received points \mathbf{y} closer to \mathbf{d}' than to \mathbf{c} , as shown in Figure 5.6. This figure is a multi-dimensional representation of transmitted vectors in a signal space. In this representation \mathbf{c} is the transmitted vector sequence whose elements are c_i and \mathbf{y} is the received vector, $\mathbf{y} = \mathbf{c} + \mathbf{n}$, where \mathbf{n} is the complex zero mean Gaussian noise.

The error probability can be obtained by considering the error event as formed by the union of the error events corresponding to all possible modulated vectors, i.e.

$$P(e) = P\left(\bigcup_{\mathbf{d} \neq \mathbf{c}} \{y \in \Gamma_{\mathbf{c}\mathbf{d}}\}\right).$$

Using the union bound.

$$P(e) \leq \sum_{\mathbf{d} \neq \mathbf{c}} P\{\mathbf{y} \in \Gamma_{\mathbf{c}\mathbf{d}}\} \quad (5.2)$$

The probability $P\{\mathbf{y} \in \Gamma_{\mathbf{c}\mathbf{d}}\}$ is obtained in the same way as in a binary decision problem. The bound in equation (5.2) can be loose as some terms are redundant. To demonstrate this the summation of equation (5.2) is expanded as

$$P(e) \leq P\{\mathbf{y} \in \Gamma_{\mathbf{c}\mathbf{d}}\} + P\{\mathbf{y} \in \Gamma_{\mathbf{c}\mathbf{d}'}\} + P\{\mathbf{y} \in \Gamma_{\mathbf{c}\mathbf{d}''}\} + \dots \quad (5.3)$$

Proposition: For a transmitted \mathbf{c} the term $P\{\mathbf{y} \in \Gamma_{\mathbf{c}\mathbf{d}}\}$ can be removed from the summation (5.3) if the following conditions are met:

- There are points \mathbf{d}' and \mathbf{d}'' in the constellation of modulated vectors such that the terms $P\{\mathbf{y} \in \Gamma_{\mathbf{c}\mathbf{d}'}\}$ and $P\{\mathbf{y} \in \Gamma_{\mathbf{c}\mathbf{d}''}\}$ are found in (5.3).

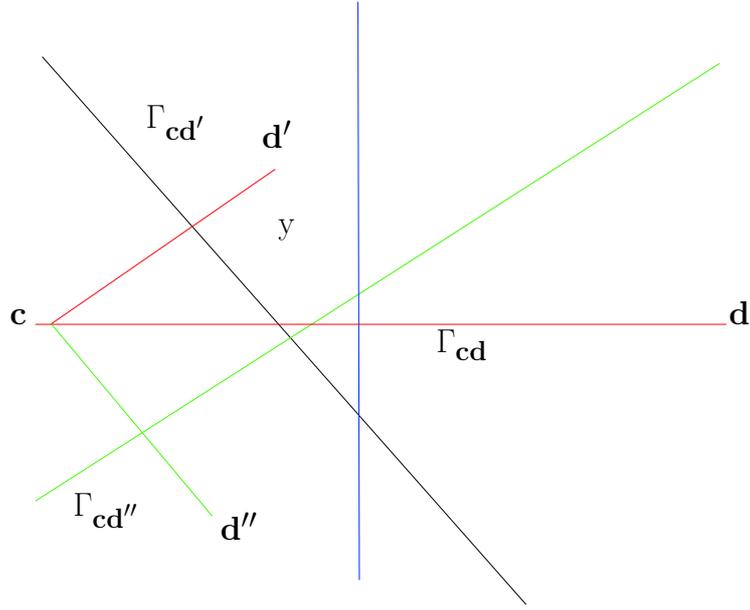


Figure 5.6: Signal space

- The following equations are true

$$\begin{aligned} \mathbf{R}_e\{(\mathbf{d}' - \mathbf{c}) \cdot (\mathbf{d}'' - \mathbf{c})\} &\geq 0 \\ \mathbf{d} &= \mathbf{d}' + \mathbf{d}'' - \mathbf{c}, \end{aligned} \quad (5.4)$$

where \cdot is the inner product operator in the Hilbert space of complex vectors.

Proof: In order to simplify the notation the following variables are introduced:

$$\mathbf{y}' = \mathbf{y} - \mathbf{c}; \quad \epsilon = \mathbf{d} - \mathbf{c}; \quad \epsilon' = \mathbf{d}' - \mathbf{c}; \quad \epsilon'' = \mathbf{d}'' - \mathbf{c}.$$

The condition of equation (5.4) becomes

$$\begin{aligned} \mathbf{R}_e\{\epsilon' \cdot \epsilon''\} &\geq 0 \\ \epsilon &= \epsilon' + \epsilon'' \end{aligned} \quad (5.5)$$

If

$$\begin{aligned} \mathbf{y} \in \Gamma_{cd'} &\Rightarrow \|\mathbf{y} - \mathbf{c}\|^2 > \|\mathbf{y} - \mathbf{c} - (\mathbf{d}' - \mathbf{c})\|^2 \\ &\Rightarrow \|\mathbf{y}'\|^2 > \|\mathbf{y}' - \epsilon'\|^2 \\ &\Rightarrow 0 > \|\epsilon'\|^2 - (\mathbf{y}' \cdot \epsilon'^*) - (\mathbf{y}'^* \cdot \epsilon') \end{aligned} \quad (5.6)$$

Similarly

$$\mathbf{y} \in \Gamma_{cd''} \Rightarrow 0 > \|\epsilon''\|^2 - (\mathbf{y}' \cdot \epsilon''^*) - (\mathbf{y}'^* \cdot \epsilon'') \quad (5.7)$$

Again

$$\mathbf{y} \in \Gamma_{\mathbf{cd}} \Rightarrow 0 > \|\epsilon\|^2 - (\mathbf{y}' \cdot \epsilon^*) - (\mathbf{y}'^* \cdot \epsilon) \quad (5.8)$$

Using the second equality in (5.5):

$$0 > \|\epsilon'\|^2 + \|\epsilon''\|^2 + \epsilon' \cdot \epsilon''^* + \epsilon'^* \cdot \epsilon'' - (\mathbf{y}' \cdot (\epsilon' + \epsilon'')^*) - (\mathbf{y}'^* \cdot (\epsilon' + \epsilon'')) \quad (5.9)$$

or

$$0 > (\|\epsilon'\|^2 - \mathbf{y}' \cdot \epsilon'^* - \mathbf{y}'^* \cdot \epsilon') + (\|\epsilon''\|^2 - \mathbf{y}' \cdot \epsilon''^* - \mathbf{y}'^* \cdot \epsilon'') + (\epsilon' \cdot \epsilon''^* + \epsilon'^* \cdot \epsilon'') \quad (5.10)$$

We distinguish 3 terms in (5.10). The first term is less than 0 if $\mathbf{y}' \in \Gamma_{\mathbf{cd}'}$, see (5.6). Similarly the second term is negative if $\mathbf{y}'' \in \Gamma_{\mathbf{cd}''}$, see (5.7). If $\mathbf{y} \in \Gamma_{\mathbf{cd}}$ then the inequality (5.10) is satisfied. When the third term in (5.10) is greater than 0, as the second condition of (5.4) is satisfied, the first, second or both terms must be less than zero showing that whenever $\mathbf{y} \in \Gamma_{\mathbf{cd}}$, then $\mathbf{y} \in \Gamma_{\mathbf{cd}'}$, or $\mathbf{y} \in \Gamma_{\mathbf{cd}''}$, or both, i.e. $\Gamma_{\mathbf{cd}} \subset \Gamma_{\mathbf{cd}'} \cup \Gamma_{\mathbf{cd}''}$ and the term $P\{\mathbf{y} \in \Gamma_{\mathbf{cd}}\}$ is accounted for in $P\{\mathbf{y} \in \Gamma_{\mathbf{cd}'}\}$ and $P\{\mathbf{y} \in \Gamma_{\mathbf{cd}''}\}$. The probability of a wrong decision when transmitting \mathbf{c} and deciding \mathbf{d} is given by;

$$P\{\mathbf{y} \in \Gamma_{\mathbf{c},\mathbf{d}}\} = Q \left(\sqrt{\frac{d^2(\mathbf{c}, \mathbf{d})}{2N_0}} \right) \quad (5.11)$$

where $d^2(\mathbf{c}, \mathbf{d})$ is the distance between vectors \mathbf{c} and \mathbf{d} .

5.6.1 Simulation of Bound

Bits into the sequence encoder result in a code in the form of a continuing sequence. A phase sequence is derived from this block code sequence by further logic coding. This bit sequence to phase sequence has been included in the Matlab generation of the bound as all possible bit sequences can be easily generated. The phase sequences generated from these bit sequences are a true representation of reality.

The receiver includes two metric based mutually supporting phase decoders (Viterbi and PSD) working in a parallel concatenation structure concatenated serially into a MAOC. This decoding element uses data derived from a quantising sixteen phase receiver to recover a coded QPSK transmission allowing the regeneration of a bit sequence. Such information is not, however, required when generating the bound. In addition the code is non linear hence no single path can be assumed without loss of generality.

The generation of the bound is based on the above propositions using 12 bit \mathbf{b} vectors and considering all possible initial sequences for the union bound. The relatively short sequence was necessary as any longer sequence would have required excessive time with the facilities available.

Although a QPSK transmission the sequence generated by the coder is derived from two binary phase subsets Y_1 and Y_2 of the 4 phase address space of Y . Each element of each subset of the phase sequence having a phase separation of radians. The sixteen level quantised receiver, together with the introduced sequence of the code, allows, using the open window procedure, a minimum acceptance squared Euclidian distance, for each element of the phase sequence decoder, of $(\frac{3\pi}{4})$. The samples are independent Gaussian random variables having a mean of 0 and a variance of $\frac{N_0}{2}$.

The bound can be considered as a valid estimate based on the phase sequence to state sequence transition and does not include the bit sequence derived from the state sequence within the MAOC decoder. This has resulted in a bound based on the standard $\frac{3\pi}{4}$ window that follows reasonably closely the results achieved by the current sequence code which, it is believed, can be improved upon using sequence techniques. In Figure 5.7 the technique has been compared with a K=4 soft decision Viterbi with traceback and not a similar length Ungerboeck code as this would be limited, at best, to a 3dB gain [58].

5.7 Results

An attempt is made to validate the basic simulation by providing the BPSK results in Figure 5.7. These correspond closely with theoretical values. Using the derived metric values as a basis the system was tested against the various possibilities within the software. For all the following tests $\frac{E_b}{N_0}$ is the same as $\frac{E_s}{N_0}$; a single bit being transmitted per symbol.

Figure 5.5 considers the average number of bits per burst error into the RS decoder with symbols between length 10 to 4. This was performed to analyse the resulting bit error bursts out of the MAOC. The result of this analysis shows that the average number of bit errors per erroneous RS symbol is asymptotic on approximately two. The restructuring of the Gaussian random errors to burst errors after error correction, though possibly controversial, is considered one of the gains of sequence coding.

Figure 5.1 compares the results of removing different elements (Viterbi decoder, Phase sequence decoder and enhancement routine) of the sequence code decodulator. It is demonstrated that each element of the decodulator benefits, to varying degrees, the overall result. As expected the Viterbi decoder has the biggest impact

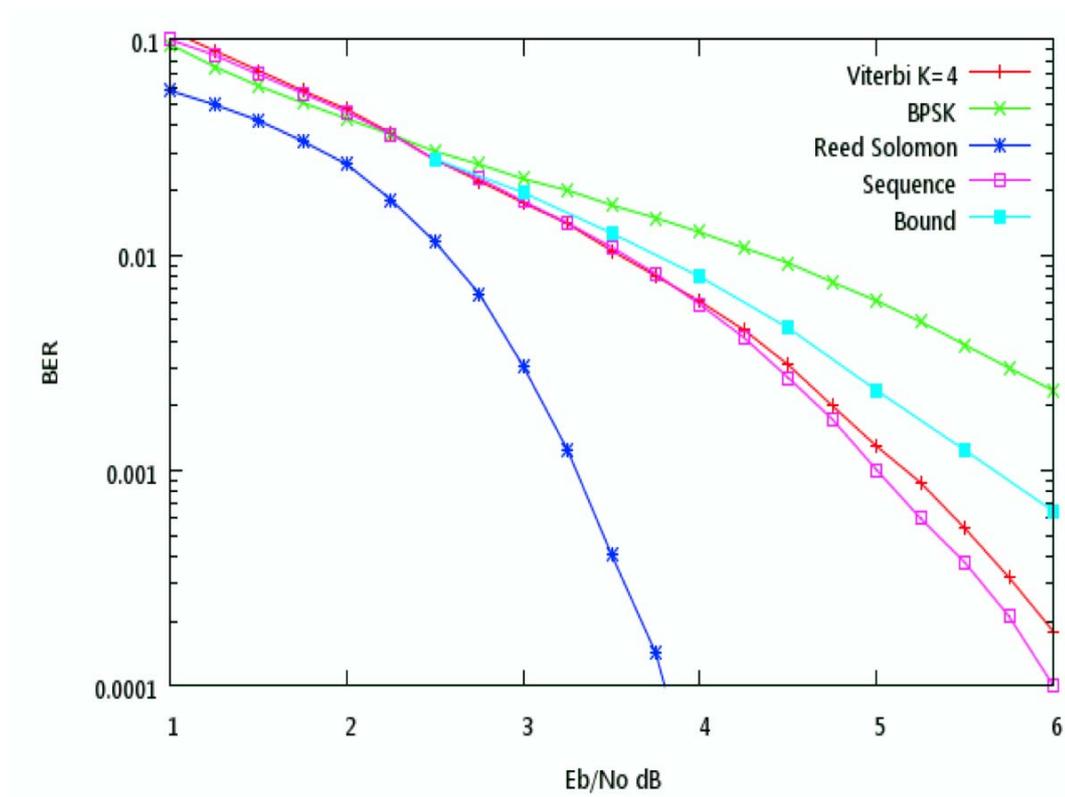


Figure 5.7: Sequence code comparison with rate, $K=4$ soft decision Viterbi with short trace-back [46] including bound.

for the overall system though the support of the phase sequence decoder was significant. The enhancer provided less improvement but being part of the phase sequence decoder adds little complexity to the system.

The results, Figure 5.7, shows that the short $K = 3$ sequence code compares favourable against the longer, soft decision $K = 4$ Viterbi including a complex trace-back system. Direct comparison with Ungerboeck codes in the literature is more problematical due to the very short nature of the codes used but sequence codes compares favourable, especially at lower $\frac{E_b}{N_0}$, with a three level coded 8 phase Ungerboeck coded modulation [12, 49]. This comparison is enhanced when taking into consideration the potential benefits of the possibility for metric transfers to a future outer decoder.

These results demonstrate that sequence coding the modulation has some merit. The current technique can be improved in many ways especially the enhancement and restructuring elements of the sequence code. In addition it is reasonable to believe that considerable gains can be achieved by including the RS, or alternative decoder, into the metric system.

Chapter 6

Conclusion

This short and simple to implement modulation technique using analysis techniques for decoding has demonstrated benefits in a Gaussian environment. It has been shown that it is possible to produce a coding technique, using logic techniques, to optimise and simplify the encoder, and integrate both code and Euclidian distance. The use of metrics values, obtained empirically, allows the two integrated decoders, of the decodulator to act in combination against the Gaussian environment.

In sequence coding the modulation is coded. The integrated decoders use the same quantised representation of the analogue phase data in two, potentially more, decoders to increase capability. An extension to the phase sequence decoder, called enhancement, acts to improve the received quantised analogue symbol data prior to decoding; at no increased delay and only slightly increased complexity.

The decoders pass only messages related to states and their metrics (reliability). This metric data, together with associated original state values, allows the sequence code's outer MAOC decoder to convert a recovered state sequence to corrected bits. The outer RS code attempts to correct any resulting short bit error bursts.

The simulation results demonstrate that it is possible to have encoding simplicity, improved error performance and near real time operation without sacrificing data rate or bandwidth. Figure 5.1 shows the effect of removing some elements from the decodulator demonstrating that each element plays a functional part in the overall unit. The results, Figure 5.7, demonstrate that the technique is potentially useful comparing favourably with a rate $\frac{1}{2}$, K=4 soft decision Viterbi with short trace-back [46] and short Ungerboeck codes [49]. The system requires further work to complete especially in the regions of extending the number of decoder concatenations, possibly with a reverse Viterbi and integrating a new outer code into the metric system. This is required as it proved difficult to transfer a bit metric

to a RS symbol metric. The use of a rated non-iterated LDPC type code is being studied which, it is believed, will provide considerable improvement over the current performance; potentially up to 2dB based on soft decision results.

The results demonstrate that, within the current work, an easy to implement coding technique has been produced. The system requires further work to complete especially in the region of integrating an outer code into the metric system as it has proved difficult to transfer a sequence code bit metric to a RS symbol metric.

Although the modulation technique described here is based on QPSK a sequence code could be designed for other multi-level modulation schemes. The technique uses principles of partial response [20] and Tamed FM [32, 4] provided ideas which, though not directly related to spectral efficiency, were used in the development of the technique.

Chapter 7

Thesis Contributions

A new, simple in design and implementation, currently very short, concatenated modulation error correction coding scheme, based on phase, state and bit sequences has been presented. This system uses analysis techniques supported by machine intelligence based on a study of error structure.

The encoder is designed to set up a state sequence generated by an overlapping three bit subset of a bit sequence to produce a logically derived modulated phase symbol sequence designed to maximise the Euclidean distance over a sequence. Both the encoder and decodulator are simple in design and efficient in operation requiring little more than a shift register, a few simple gates and a modulator for the encoder and a 16 phase receiver and some comparators and software for the decodulator.

The detection and decoding procedures are new. The present decodulator uses a sixteen phase quantiser on a received QPSK transmission. The encoder design allows the modulation data to be decoded by a specially designed integrated, concatenated (both serially and in parallel), phase decodulator. This simple decodulator consists of a newly designed PSD and a modified Viterbi algorithm serially concatenated with a MAOC unit.

The PSD provides a unique state solution, with its associated metric, to both the first frame of the Viterbi decoder window and the MAOC. This metric allows the normally 4 state Viterbi decoder to operate as an eight state system potentially improving the overall system gain. The integrated decoders (Viterbi and phase sequence) do not correct to bit level but translate a received quantised phase sequence to a recovered state sequence supported by reliability metrics.

The concatenation of the two integrated decoders, of the decodulator, is unique in using, without modification, the same quantized phase symbol data in mutual support. The original, received Gaussian distribution of potential transmission errors,

where they are not corrected, are restructured via the MAOC into short bursts of bit errors having an average length of approximately 2 bits, Figure 5.5, allowing the potential for improved gains to be achieved by a short outer RS decoder. This short length RS decoder introduces a delay of only 384 bits.

A common problem with concatenated codes is that of balancing the inner and outer elements of the codes. This is facilitated in sequence coding using metric passing techniques. The empirical development of the metrics effectively creates a balance between the elements of the integrated decoder.

By using simple empirical techniques the integrated decoder attempts to balance the two sets of metric values (phase sequence and Viterbi decoders) both to the link and the interaction of the decoders. To achieve this balance, in practice, would require only a few thousand data bits from a pre-defined bit source known to each end. Each end of the link would have such information probably in the form of the internal software. Such information would be transmitted at a low energy level, possibly during polling, to allow the required metric optimization. This metric optimisation would be unique to the link hence that obtained would be link balancing (equalisation) metrics.

The sequence code itself is more efficient than standard Viterbi as no traceback is required such that the additional simple PSD and enhancement routine with the MAOC are still less complex than the traceback element of standard Viterbi. The lack of traceback reduces the memory requirements as the data required in each Viterbi window is the initial state, the current state, the current Viterbi metric sum during each frame and the PSD metric and state. The information passed to the MAOC allows this element to perform the basic functions of traceback.

The technique is believed to be the first to use the magnitude $|z_i|$ of the received signal as an element in an error correction system. This value provides additional reliability information regarding the phase symbol.

The integrated decoder is believed to be the first to take into consideration a peculiarity of the Viterbi decoders output error structure whereby the Viterbi decoder outputs errors in positions prior to the true error position. This peculiarity of the Viterbi decoder means that the phase sequence decoder, using the same phase information, can provide state correction information in those state positions where the Viterbi decoder has been demonstrated, within the MAOC, using the sequence continuum, to have failed.

The following papers related to sequence coding have been published by learned bodies as suitable for publication and presentation.

The following two papers were presented in IEE Journals:

L. Woodhead, Angel M Bravo Santos, "Tunable error correction and link restruc-

turing,” *IEE Communications Engineer*, pp. 32-35, Oct./Nov. 2003

Angel Bravo Santos, L. Woodhead & V. K. Bhargava, “Spherical sequences with low aperiodic cross-correlation,” *IEE Proceedings on Communications* vol. 151, No. 6, pp. 601-604, (2004).

The following two papers were presented at international conferences:

L. Woodhead, Angel Bravo Santos, “Inference and Ungerboeck coded modulation,” “*TELECOM4 International Conference, Telecommunication, Electronics and Communications*,” at Santiago de Cuba, (Cuba), Jul. 2004.

L. Woodhead & Angel Bravo Santos, “ Sequence coded modulation,” *Fifth IEE International Conference on 3G Mobile Communication Technology*, (IEE, Savoy Place, London.), Oct. 2004.

Chapter 8

Future Work

This work attempts to demonstrate the benefits of encoding the modulation with designed logic based phase sequences within a state structure using basic machine intelligence to recover the original data by analysis of the phase and state sequences. The current work leaves much to be completed.

Although several alternative options for the quantiser were considered, which might improve results, these would require considerable development. One example, that was simulated, was a 32 phase quantiser using a simulated pair of parallel operating 16 phase quantisers in different modes. The study of the quantiser modes requires further study.

Within any sequence reducing the quantised interval might improve the results and it is considered that the offset quantiser “**D**” should provide a possibility of achieving this. As the simulations are working within a sequence the possibility of opening the operational window to exceed the π threshold using additional Viterbi decoder metrics should be considered.

The current logic arrangement of the sequence code associates the phase elements to bits. This simplified the recovery software and helped the code loss recovery but it is possible that an improvement could be achieved by removing such an association. An example of a non-associated logic system is provided as Table 4.5 though this version is non viable within the current decoding arrangement.

In the current sequence code the sequence length of the encoder has been limited to three bits and the RS symbol length was, by the minimum delay specification of the project limited to six bits. Increasing the sequence code to four bits or more would be possible. An example of an unoptimised four bit coder logic scheme is shown in Table 8.1. Longer sequences with more data bits per phase symbol, i.e. 8PSK with $4 * \frac{\pi}{2}$ phase separated pairs, and different codes are possible and would

$\mathbf{s}_m/\mathbf{s}_{m-3}$	$\mathbf{s}_m = \mathbf{1}$		$\mathbf{s}_m = \mathbf{0}$		$\mathbf{s}_m/\mathbf{s}_{m-1}$
$\mathbf{s}_{m-3} = \mathbf{0}$	$\pi/2$	π	0	$3\pi/2$	$\mathbf{s}_{m-1} = \mathbf{1}$
$\mathbf{s}_{m-3} = \mathbf{1}$	π	$\pi/2$	$3\pi/2$	0	
	0	$3\pi/2$	$\pi/2$	π	$\mathbf{s}_{m-1} = \mathbf{0}$
$\mathbf{s}_{m-3} = \mathbf{0}$	$3\pi/2$	0	π	$\pi/2$	
$\mathbf{s}_{m-3}/\mathbf{s}_{m-2}$	$\mathbf{s}_{m-2} = \mathbf{0}$	$\mathbf{s}_{m-2} = \mathbf{1}$	$\mathbf{s}_{m-2} = \mathbf{0}$	$\mathbf{s}_{m-2} = \mathbf{1}$	$\mathbf{s}_{m-1}/\mathbf{s}_{m-2}$

Table 8.1: Potential 4 bit logic

possibly be beneficial in some environments. It is intended that such techniques be developed.

An extension of the number of phase confirmation sequence checks, to $\tilde{\phi}_{m+2}$ will be performed. These confirmation checks are performed on the future dual $\frac{\pi}{2}$ intervals being separated by an angle π . These could be increased to two symbols;

$$\left(\left(\tilde{\phi}_{m-2}, \tilde{\phi}_{m-1}, \tilde{\phi}_m \right), \left(\tilde{\phi}_{m+1}, \tilde{\phi}_{m+2} \right), St_{m-1} \right).$$

Though some benefit is to be expected previous work suggests that the gain would not be significant.

The extension of Viterbi decoder dual phase routing to the multiple state problem requires study. The effect, however, of not using this possibility does not appear to be significant as the facility tends not to be regularly accessed.

A look at the possibility of using a reverse Viterbi decoder might be beneficial. In this case the Viterbi would operate on the received phases;

$$\tilde{\phi}_{m-(w+1)} + \tilde{\phi}_{m-(w+2)}, \dots, \tilde{\phi}_m$$

where w is the Viterbi window length. This would be an additional decoder allowing an extension of the effective length of the Viterbi decode window without increasing code delay; though problematic within the current simulation it would be a fairly simple matter within a new simulation requiring only minor software extensions.

It is intend that the current system be develop by changing the current outer RS decoder with the objective of developing an erasure decoder being a version of a fixed rate LDPC type code. It is believed that using these codes could provide considerable gains in this environment. The development of a decoder would be a prime objective as it would have to be designed to fit into the sequence decodulators metric structure and be of minimum length. As a further development of this technique it is intend to develop a simple means to use the metric values to flag

erasures in the outer decoder using error forecasting techniques [16, 50]. The possible transfer of a related metric, via a sequence, to a bit to a (decoder) symbol requires further study.

The metric system appears not to be fully optimized and although many options have been tested many other possibilities have not been fully examined. The empirically derived metric values have been shown to be non-continuous in value through the metric sequence indicating that further optimization of the phase sequence code structure is possible.

Work is required to reduce the current complexity of a system being, at present, at a design stage. The software has been written with the idea of converting to hardware and much of the code would be simplified by such a conversion.

Bibliography

- [1] Ali Abedi & Amir K. Khandani, “An analytical method for approximate performance evaluation of binary linear block codes,” *IEEE Transactions on Communications*, vol. 52, No. 2, pp. 228-235, Feb. 2004.
- [2] Chien-Meen Hwang, Christine Lee & Howard Hicks of Advanced Micro Devices Inc. (Sunnyvale, CA, US), “Viterbi decoder utilizing partial backtracing,” United States Patent 7248637.
- [3] F.Amoroso, “Pulse and spectrum manipulation in the minimum (frequency) shift keying (MSK) Format,” *IEEE Transactions on Communications*, vol. 24, No. 3, pp. 381-384, Mar. 1976.
- [4] John B. Anderson, Tor Aulin & Carl-Erik Sundberg, *Digital Phase Modulation*. Plenum Press, New York and London, 1986.
- [5] Erdal Arikan, “Channel polarization: A method for constructing capacity achieving codes for symmetric binary-input memoryless channels,” *IEEE Transactions on Information Theory*, vol. 55, No. 7, pp. 1-23, Jul. 2009.
- [6] Tor Aulin, Nils Rydbeck & Carl -Erik W. Sundberg, “Continuous phase modulation,” *Part 1 & 2, IEEE Transactions on Communications*, vol. 29, No. 3, pp. 196-225, Mar. 1981.
- [7] Mark C. Austin & Ming U. Chang, “Quadrature overlapped raised-cosine modulation,” *IEEE Transactions on Communications*, vol. 29, No. 3, pp. 237-249, Mar. 1981.
- [8] Sandro Bellini, Maurizio Sonzogni & Guido Tartara, “Noncoherent detection of tamed frequency modulation,” *IEEE Transactions on Communications*, vol. 32, No. 3, pp. 218-224, Mar. 1984.
- [9] Berrou, C. Glavieux & A. Thitimajshima, “Near Shannon limit error-correcting coding: Turbo codes,” in *Proc. IEEE Int conf. Communications*, Geneva, Switzerland, May 1993.

- [10] J.A. Betts, *Signal Processing Modulation and Noise*, Hodder and Stoughton, 1983.
- [11] Vijay K. Bhargava, David Haccoun, Robert Matyas & Peter P. Nuspl, *Digital Communications By Satellite*, A. Wiley, Interscience Publication, 1981
- [12] Ezio Biglieri, Dariush Divsalar, Peter J. McLane & Marvin K. Simon, *Introduction to Trellis-Coded Modulation with Applications*, Maxwell MacMillan International Editions, 1991.
- [13] Li Bin, "The noncoherent least squares phase difference sequence estimator for continuous phase modulation," *IEEE Transactions on Communications*, vol. 43, No. 2/3/4, pp. 718 -721, Feb./Mar./Apr., 1995.
- [14] A. Burr, "Turbo -codes: the ultimate error control codes?," *IEE Electronics & Communications Engineering Journal*, pp. 155-165, Aug. 2001.
- [15] Chip Fleming, *A Tutorial on Convolutional Coding with Viterbi Decoding*, Spectrum Applications, 2006.
- [16] Daniel J. Costello, Jr., Joachim Hagenauer, Hideki Imai & Stephen B. Wicker, "Applications of error-control coding," *IEEE Transactions on Information Theory*, vol. 44, No. 6, pp. 2531-2560, Oct. 1998.
- [17] SystemView Simulator; Elanix incorporated 5655 Lindero Canyon Road, Suite 721, Westlake Village, CA91262, www.Eagleware.com
- [18] B.G. Evans (editor), *Satellite Communication Systems*, Peter Peregrinus Ltd, 1987.
- [19] K. Feher, *Advanced Digital Communications: Systems and Signal Processing Techniques*, Englewood Cliffs, N.J. Prentice Hall, 1987.
- [20] Bartolomeu, F. Ucha Filho, Mark A. Herro, & Daniel J. Costello, Jr., "A multilevel approach to constructing trellis-matched codes for binary-input partial response channels," *IEEE Trans on Information Theory*, vol.45, No. 7, pp. 2582-2591, Nov. 1999.
- [21] G.D. Forney. Jr., "Coset codes part II binary lattice and related codes," *IEEE Transactions on Information Theory*, vol. 34, No. 5, pp. 1152-1187, Sept. 1988.
- [22] G.D. Forney. Jr., "The Viterbi algorithm," *Proc IEEE*, vol. 61, No. 3, pp. 268-276, May 1973.
- [23] G.D. Forney. Jr., *Concatenated Codes*, Cambridge, MA: MIT Press, 1966.

- [24] R.G. Gallager, "Low density parity check codes," *IRE Transactions on Information Theory*, IT-8, pp. 21-28, Jan. 1962.
- [25] R.G. Gallager, "A simple derivation of the coding theorem and some applications," *IEEE Transactions on Information Theory*, vol. 11, pp. 3-18, Jan. 1965.
- [26] R.G. Gallager, *Low Density Parity Check Codes*, Number 21 in MIT Research Monogram Series, MIT press, 1963.
- [27] Giuseppe Caire, Giorgio Taricco & Ezio Bigliere, "Bit-interleaved coded modulation", *IEEE Transactions on Information Theory*, vol. 44, No. 3, pp. 927-946, May 1998.
- [28] Hamming R. W., "Error detecting and error correcting codes", *Bell System Technical Journal*, No. 2 pp. 147-160, 1950.
- [29] Simon Haykin, *Communication Systems*, John Wiley & Sons, 1994.
- [30] Jeremiah F. Hayes, "The Viterbi algorithm applied to digital data transmission," *IEEE Communications Magazine*, vol. 40, No. 5, pp. 26-32 May 2002.
- [31] Hideki Imai & Shuji Hirasawa, "A new multilevel coding method using error correcting codes," *IEEE Transactions on Information Theory*, vol. 23, No.3, pp. 371-377, May 1977.
- [32] Frank de Jager & Cornelis B. Dekker, "Tamed frequency modulation, a novel method to achieve spectrum economy in digital transmission," *IEEE Transactions on Communications*, vol. 26, No. 5, pp. 534-542, May 1978.
- [33] George Kennedy, *Electronic Communication Systems (fourth Edition)*, McGraw Hill, International Editions, 1992.
- [34] Frank Kienle & Norbert Wehn, "Joint graph-decoder designs of IRA codes on scalable architectures," *Institute of Microelectronic Systems, University of Kaiserslautern, Erwin-Schrodiner-Strabe, 67663 Kaiserslautern, Germany*, 2004.
- [35] Frank R. Kschischang, Brendan J. Frey & Hans -Andrea Loeliger, "Factor graphs and sum-product algorithm," *IEEE Transactions on Information Theory*, vol. 47, No.2, pp. 498-519, Feb. 2001.
- [36] David Kwan & Samir Kallel, "A truncated best-path algorithm," *IEEE Transactions on Communications*, vol. 46, No. 5, pp. 568-572, May 1998.

- [37] Patrick S.K. Leung & Dr. Kamilo Feher, "F-QPSK a superior modulation for future generations of high capacity microcellular PCS systems," in *43rd IEEE Vehicular Technology Conference*, pp. 38-41, 18-20, May 1993.
- [38] David J. C. MacKay & Neal R. M., "Good codes based on very sparse matrices," in *Cryptography and Coding. 5th IMA Conference 1995*, LNCS 1025, edited by C. Boyd, pp. 100-111.
- [39] David J. C. MacKay, "Good error correction codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, No. 2, pp. 399-431, Mar. 1999.
- [40] James Martin, *Communications Satellite Systems*, Prentice Hall Inc., 1978.
- [41] Robert Matyas, "Effect of noisy phase references on coherent detection of FFSK signals," *IEEE Transactions on Communications*, vol. 26, No. 6, pp. 807-815, Jun. 1978.
- [42] Robert J. McEliece, "On the BCJR trellis for linear block codes," *IEEE Transactions on Information Theory*, vol.42, No. 4, pp. 1072-1092, Jul. 1996.
- [43] J.L Massey & D.J Costello, Jr., "Non-systematic convolutional codes for sequential decoding in space applications," *IEEE Transactions on Communications Technology*, Vol. 19, No. 5, pp. 806-813, Oct. 1971.
- [44] J.L. Massey, "Coding and modulation in digital communications," in *Proc.1974 International Zurich Seminar Digital Communications*, Zurich, Switzerland.
- [45] Robert Matyas, "Effect of noisy phase references on coherent detection of FFSK signals," *IEEE Transactions on Communications*, vol. 26, No. 6, pp. 807-815, Jun. 1978.
- [46] Arnold M. Michelson & Allen H. Levesque, *Error Control Techniques for Digital Communication*, John Wiley & Sons Ltd, 1985.
- [47] Mischa Schwartz, *Information Transmission, Modulation and Noise*, McGraw Hill, 1990.
- [48] K. Miya, *Satellite Communications Technology*, KDD Engineering and Consulting Inc., 1980.
- [49] Robert H. Morelos -Zaragoza, *The Art of Error Correcting Coding*, John Wiley & Sons Ltd, 2004.

- [50] E. Paaske, "Improved decoding for a concatenated coding system recommended by CCSDS," *IEEE Transactions on Communications*, vol. 38, No. 8, pp. 1138-1144, Aug. 1990.
- [51] Ravi Palanki & Jonathan S. Yedidia, *Rateless Codes on Noisy Channels*, Internet Mitsubishi Electric Research Laboratories.
- [52] Robert Padovani & Jack K. Wolf, "Coded phase/frequency modulation," *IEEE Transactions on Communications*, vol. 34, No. 5, pp. 446-453, May 1986.
- [53] W. Wesley Peterson & E. J. Weldon, Jr., *Error Correcting Codes 2nd Edition*, The MIT press, 1981.
- [54] Al-Nasir Premji & Desmond P. Taylor, "A practical receiver structure for multi-h CPM signals," *IEEE Transactions on Communications*, vol. 35, No. 9, pp. 901-908, Sep. 1987.
- [55] William Press, Saul A. Teukolsky, William T. Vetterling & Brian P. Flannery, *Numerical Recipes in C++ the art of Scientific Computing Second Edition*, Cambridge University Press, 2007.
- [56] John G. Proakis, *Digital Communications Fourth edition*, McGrawHill, 2001.
- [57] Peter Sweeney, *Error Correction Coding (An Introduction)*, Prentice Hall, 1991.
- [58] Peter Sweeney, *Error Control Coding (from theory to practice)*, John Wiley & Sons Ltd, 2004.
- [59] Ramesh Mahendra Pyndiah, "Near-optimum decoding of product codes: block turbo codes," *IEEE Transactions on Communications*, vol. 46, No. 8, pp. 1003-1010, Aug. 1998.
- [60] T. Richardson & R Urbanke, "The Capacity of Low-Density Parity Check Codes under Message Passing decoding," *IEEE Transactions on Information Theory*, vol. 47, No. 2, pp. 599-618, 2 Feb. 2001.
- [61] T. Richardson & R Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Transaction Information Theory*, vol 47, No.2, pp. 638 -656, Feb. 2001.
- [62] Soheil I. Sayegh, "A class of optimum block codes in signal space," *IEEE Transactions on Communications*, vol. 34, No.10, pp.1043-1045, Oct. 1986.
- [63] Mischa Schwartz, *Information Transmission, Modulation and Noise*, McGrawHill, 1990.

- [64] C.E. Shannon, "A mathematical theory of communication," *The Bell Technical Journal*, vol. 27, pp. 379-423, 623-656, July, October, 1948.
- [65] M. Siala & G.K. Kaleh, "Block & trellis codes for binary (1 -D) partial response channel with simple maximum likelihood decoders," *IEEE Transactions on Communications*, vol.44, No.12, pp. 1613-1616, Dec. 1996.
- [66] R. Michael Tanner, Deepak Sridhara, Arvind Sridharan, Thomas E Fuja & Danial J. Costello, Jr., "LDPC block and convolutional codes based on circulant matrices," *IEEE Trans on Information Theory*, vol.50, No.12, pp. 2966-2984, Dec. 2004.
- [67] Gottfried Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory*, vol. 28, No. 1, pp. 55-67, Jan. 1982.
- [68] Gottfried Ungerboeck, "Trellis coded modulation with redundant signal sets part 1 and 2," *IEEE Communications Magazine*, vol. 25, No. 2, pp. 5-21, 1987.
- [69] Andrew J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, No. 2, Apr. 1967.
- [70] Stephen B Wicker, *Error Control Systems for Digital Communications and Storage*, Prentice Hall, 1995.
- [71] L. Woodhead & Angel M Bravo Santos, "Tunable error correction and link restructuring," *IEE Communications Engineer*, pp. 32-35, October/Nov. 2003.
- [72] Angel Bravo Santos, L. Woodhead & V. K. Bhargava, "Spherical sequences with low aperiodic cross-correlation," *IEE Proceedings on Communications*, vol. 151, No. 6, pp. 601-604, Dec. 2004.
- [73] L. Woodhead & Angel Bravo Santos, "Inference and Ungerboeck coded modulation," in *TELECO4 International Conference, Telecommunication, Electronics and Communications*, at Santiago de Cuba, Jul. 2004.
- [74] L. Woodhead & Angel Bravo Santos, " Sequence coded modulation," in *Fifth IEE International Conference on 3G Mobile Communication Technology*, at IEE, Savoy Place, London, pp. 596-599, Oct. 2004.
- [75] Robert H. Morelos -Zaragoza, *The Art of Error Correcting Coding*, John Wiley & Sons Ltd; second reprint, Jul. 2004.