

**TITLE PAGE**

**OPTIMISATION OF DATA THROUGHPUT  
FOR LEO SATELLITE DOWNLINK USING  
TRELLIS CODED MODULATION.**

**BY**

**MOHAMMED, UMAR MASKA  
PG/UD/M.ENG/07/73.**

**DEPARTMENT OF ELECTRONIC ENGINEERING,  
UNIVERSITY OF NIGERIA,  
NSUKKA**

**February, 2016.**

**CERTIFICATION**

This is to certify that **MOHAMMED, UMAR MASKA**, a post graduate student in the Department of Electronic Engineering, University of Nigeria, Nsukka, with Registration number PG/UD/M.ENG/07/73 has satisfactorily completed the requirements for course and research work for the degree of Masters of Engineering (M.Eng) in Telecommunication Engineering.

-----  
---

Professor A.N Nzeako  
Ani  
Supervisor.

-----  
---

Professor C. I.  
Head of Department.

-----  
---

Date

-----  
---

Date

-----

External Examiner.

-----

Date

**DECLARATION**

The work embodied in this thesis is original and has not been submitted in part or in full for any other Diploma or Degree in this University or any other University.

-----  
Mohammed, Umar Maska

-----  
Date

## **DEDICATION**

This thesis is dedicated to the Almighty God and all Praises belong to Him.

## **ACKNOWLEDGEMENT**

I wish to express my profound gratitude to my supervisor, Prof. A.N Nzeako for his inspirational guidance and encouragement which made this work a success. I am also grateful to the Head Department, Professor C. Ani who through his lectures gave insight on the significance of error control coding in Digital Communications.

My appreciation also goes to the former Director of Digital Bridge Institute Abuja, Dr U. Okechukwu who also gave me insight on research method and his administrative staff Mrs. Muslimat A. Taiwo for her support and encouragement toward the success of this work.

This acknowledgement will not be complete without commending my colleagues in the National Space Research and Development Agency (NASRDA), Eng. Akachukwu Chebe and Eng. O. Adetoro who shared their experience with me on the Operation of Nigeriasat-1 Ground Station.

Maska

Mohammed, Umar

## **TABLE OF CONTENTS**

### **Page**

TITLE PAGEi

CERTIFICATION AND APPROVAL

ii

DECLARATION

iii

DEDICATIONivACKNOWLEDGEMENTv

TABLE OF CONTENTSvi

LIST OF TABLESix

LIST OF FIGURES

xi

LIST OF ACRONYMS

xiv

ABSTRACTxvi

## **CHAPTER ONE**

### **GENERAL INTRODUCTION**

1.0 Introduction1

1.1 Motivation

1

1.2 Problem Statement

2

1.3 Objective of the Study

2

1.4 Significance of the Study21.5Scope of Study and Methodology

2

## **CHAPTER TWO**

### **LITERATURE REVIEW**

2.0 Introduction

4

2.1 Low Earth Orbit and Ground Track

4

2.2 Geometry of LEO Satellite Visibility

8

2.3 Downlink Budget Analysis of a LEO Satellite

12

2.4 Basic Digital Modulations for SatelliteCommunications16

2.5 Probability of Error in BPSK and QPSK Schemes

20

2.6 Error Control and Coding Gain

25

2.7 Spectral Efficiency in Power and Bandwidth Limited Link

26

2.8 Reviews on Past Publications on Optimisation of LEO satellite

Downlink 29

2.8.1 Adaptive Variable Data Rate: An IEEE Paper by M.A Matar	29
2.8.2 Optimisation Through Hybrid-ARQ: An IEEE Paper from University of Surrey	31
2.8.3 Adaptive Communication System for Implementation on Board a Future Algerian LEO Satellite, a 2007 IEEE Paper	34

## **CHAPTER THREE**

### **TRELLIS CODED MODULATION AND THE CONCEPTUAL MODEL**

3.0 Introduction	37
3.1 Euclidean Distance in MPSK Schemes	38
3.2 Convolution Coding Technique	39
3.2.1 Trellis Diagram	41
3.2.2 Coding Gain in Convolution Coding	43
3.3 Set Partition Theory	46
3.4 Trellis Coded Modulation (TCM) Scheme	48
3.4.1 Implementation of TCM/8-PSK	50
3.4.1 Coding Gain in Trellis Coded Modulation	54
3.4.3 High Gain TCM/8-PSK Schemes	56
3.5 The Conceptual Model: Integrated Multi-State TCM	64
3.5.1 Design of Multiple State Encoders	66

## **CHAPTER FOUR**

### **COMPUTER SIMULATION FOR CODING GAIN**

4.0 Introduction	89
4.1 MATLAB Bit Error Rate Analysis Tool (bertool)	90
4.2 Simulations of MPSK Theoretical Links	93
4.3 MATLAB Template for Downlink Mode	100
4.4 Building MATLAB Downlink Block Model	102
4.4.1 Settings for Monte Carlo Simulation	105

4.5	Simulations of MPSK Downlink Block Models	106
4.6	Simulations of TCM Block Downlink	115
4.7	Discrete Simulations of Multi-State TCM/8-PSK Encoders	120
4.8	Modification and Simulation of the Conceptual Model	127
4.9	Data Throughput Measurements for Optimisation	129

## **CHAPTER FIVE**

### **ANALYSIS OF SIMULATION RESULTS**

5.0	Introduction	133
5.1	Performance of Convolution coded MPSK on Theoretical Links	134
5.2	Performance of Convolution coded MPSK on Downlink Block Models	135
5.3	Performance of TCM/MPSK on Downlink Block Models	137
5.4	Discrete Performance of Multi-State TCM/8-PSK Encoders	139
5.5	Performance of Multiplexed Multi-State TCM/8-PSK Encoders	140
5.6	Data Throughput Comparison and Optimisation on LEO Downlink Block Models	141

## **CHAPTER SIX**

### **CONCLUSION AND RECOMMENDATION**

144

### **REFERENCES**

### **APPENDICES**



- A. Pulse Shaping and Nyquist Filters 148
- B. Analysis on High Power Amplifier 155

**LIST OF TABLES**

2.1 Linear Displacement of LEO Ground Track	8
2.2 Visibility Duration of LEO Satellite	
11	
2.3 Phase angle of I- Channel and Q-Channel in QPSK	18
2.4 QPSK Implementation	19
2.5 Adaptive Discrete Rate in Fading Region	
35	
3.1 State Transition Table of a 1/3- Rate Convolution Encoder	41
3.2A 1/3-Rate Convolution Encoder State - Output Table	
41	
3.3 State Table for a Simple TCM	
52	
3.4 Transition Tables for 8-State Feedback TCM/8-PSK	
57	
3.5 Transition State table for a 16-State Feedback TCM/8-PSK	60
3.6 Coding Gain Variation in Multiple Transition State TCM	63
3.7 Transition and Output Table for a 4-State TCM/8-PSK	
68	
3.8 Transition and Output Table for a 8-State TCM/8-PSK	
71	
3.9 Transition and Output Table for a 16-State TCM/8-PSK	
74	
3.10 Transition and Output Table for a 32-State TCM/8-PSK	
77	
3.11 Transition and Output Table for a 64-State TCM/8-PSK	80
3.12 Transition and Output Table for a 128-State TCM/8-PSK	
84	
4.1 Performance Summary of Uncoded, Convolution Coded and TCM on MPSK Downlink Block Model at BER=10 <sup>-6</sup>	113
4.2 Data Rate of Uncoded, Convolution Coded and TCM on MPSK Downlink Block Model at 5 x 10 <sup>6</sup> Sps and bandwidth 5.5MHz	114

4.3 Performance Summary of TCM/MPSK on Downlink Block Model at BER = $10^{-6}$	119
4.4 Performance Summary of Discrete Simulation of Multi-state Encoders on Downlink Block Model at BER = $10^{-6}$	126
4.5 Simulated Free Space Loss Insertion on Multiplexed Multi-State Downlink Model	127
4.6 Data Throughput of Un-coded MPSK Downlink Block Models	130
4.7 Data Throughput of Convolution Coded MPSK Downlink Block Model	131
4.8 Data Throughput of TCM/MPSK Downlink Block Models	131
4.9 Data Throughput on Multi-State TCM/8-PSK Encoders	132
5.1 Performance Analysis of Theoretical MPSK Links	134
5.2 Coding Gains of Convolution Codes on Theoretical MPSK Links	135
5.3 Performance Analysis of Uncoded and Convolution Coded MPSK on Downlink Block Models	136
5.4 Coding Gains of Convolution Codes on MPSK Downlink Block Models	137
5.5 Performance Analysis of TCM/MPSK on Downlink block model	138
5.6 Coding Gains of TCM/MPSK on Downlink Block Models	138
5.7 Coding Gain Comparison between Convolution code and TCM	138
5.8 Discrete Coding Gains of Multi-State TCM/8-PSK Encoders	140
5.9 Differential Coding Gains of Multi-State Encoders from 3dB	141
5.10 Data Throughput Comparison between Uncoded, Convolution Coded and TCM on MPSK Downlink Block Model	142
5.11 Optimisation of Data Throughput on Multi-State TCM/8-PSK	143

<b>LIST OF FIGURES</b>	<b>Page</b>
2.1 Low Earth Orbit Satellite and Ground Track Displacement	
62.2 Geometry of LEO Satellite Visibility	
92.3 Sketch of LEO Satellite Downlink and its Components	
122.4 Gain Pattern of QuadriFilar Helix (QFH) Antenna	
162.5 Block Diagram of Binary Phase Shift Keying	
Implementation	
162.6 Schematic of QPSK Implementation	
192.7 QPSK Signal Points	
202.8 Probability Density Function (PDF) of Antipodal	
Signal Point	
212.9 Graph of Probability of Error $P_e$ Vs $E_b/N_0$ for	
BPSK and QPSK	
242.10 Graph of Probability of Symbol Error in MPSK	
Schemes	
252.11 Graph of Probability of Bit Error in Uncoded	
and Coded BPSK	
262.12 Graph Spectral Efficiency in Power	
Limited Link	
282.13 Graph of Spectral Efficiency in Bandwidth	
Limited Link	
282.14 Geometry of Earth-Satellite Angular	
Relationship	
302.15 Flow Chart Diagram of Hybrid-ARQ	
Coding Scheme	
333.1 Euclidean Distance in QPSK and 8-PSK	
Signal Constellation	
383.2 A Simple 1/3-Rated Convolution	
Encoder	
39	
3.3 State Transition Diagram of 1/3-rated Convolution Encoder	40
3.4 Trellis Diagram of 1/3-rated Convolution Encoder	41

3.5	Diagram of Coding Process in 1/3-rated Encoder	42
3.6	Convolution Encoder of L Blocks of k-Stage Shift Registers	43
3.7	Diagram of 2/3 rated Convolution Encoder	44
3.8	Mapping of 2-bits and 3-bits Sequences	44
3.9	Trellis of a 2/3 Convolution Code and Coding Gain	45
3.10	Set Partitioning of an 8-PSK Signal Points	47
3.11	Examples of Trellis Coded Modulation Schemes	49
3.12	General Diagram of Trellis Coded Modulation Scheme	50
3.13	A Simple TCM and State Transition	51
3.14	Mapping of TCM/8-PSK Output ( $c_3c_2c_1$ ) in a Constellation	52
3.15	Trellis Diagram for a Simple TCM Scheme	53
3.16	Minimum Distance Transition Path in a TCM Scheme	55
3.17	Systematic Feedback of an 8-State Transition TCM/8PSK	56
3.18	Trellis Diagram of an 8-State Transitions TCM/8-PSK	58
3.19	Mapping for an 8-State TCM/8-PSK in Constellation	59
3.20	Systematic Feedback of a 16-State Transition TCM/8-PSK	
593.21	Trellis Diagram of a 16-State TCM/8PSK	
623.22	Mapping for a 16-State TCM/8PSK in Constellation	
633.23	The Conceptual Model . An Integrated Multi-State TCM/8-PSK	65
3.24A	1/2-Rate Convolution Encoder from <i>poly2trellis(5, [35 31])</i>	67
3.25A	4-State TCM/8-PSK Encoder: <i>poly2trellis([3 1],[0 7 1;1 0 0])</i>	68
3.26	Mapping for a 4-State TCM/8-PSK in Constellation	69
3.27	An 8-State TCM/8-PSK Encoder: <i>poly2trellis([2 3],[0 3 1; 4 0 3])</i>	71
3.28	Mapping for a 8-State TCM/8-PSK in Constellation	72
3.29A	16-State TCM/8-PSK Encoder: <i>poly2trellis([2 4],[0 3 1;10 1 5])</i>	73
3.30	Mapping for a 16-State TCM/8-PSK in Constellation	74
3.31A	32-State TCM/8-PSK Encoder: <i>poly2trellis([3 4],[0 5 3;10 113])</i>	76
3.32	Mapping for a 32-State TCM/8-PSK in Constellation	78
3.33	A 64-State TCM/8-PSK Encoder: <i>poly2trellis([4 4],[0 11 7; 10 4 3])</i>	79
3.34	Mapping for 64-State TCM/8-PSK in Constellation	82
3.35	128-State TCM/8PSK Encoder: <i>poly2trellis([5 4],[0 23 14; 10 15 3])</i>	84
3.36	Mapping for 128-State TCM/8-PSK in Constellation	88
4.1	Bit Error Rate Tool box (bertool) on MATLAB Execute Window	93
4.2	Plot of BER versus $E_b/N_o$ for Theoretical BPSK Link	94
4.3	Plot of BER versus $E_b/N_o$ for QPSK Scheme on Downlink	96

4.4	Plot of BER versus $E_b/N_0$ for 8-PSK on Downlink	97
4.5	Plot of BER versus $E_b/N_0$ for 16-PSK on Downlink	99
4.6	Plot of BER versus $E_b/N_0$ for 32-PSK Scheme on Downlink	100
4.7	MATLAB RF Satellite Downlink Template Model	
	102	
4.8	A Convolution Coded QPSK Downlink Block Model	
	104	
4.9	Convolution Coded QPSK Constellation at $E_b/N_0=10.7dB$	
	109	
4.10	Convolution Coded 8-PSK Constellation at $E_b/N_0=14.5dB$	110
4.11	Convolution Coded 16-PSK Constellation at $E_b/N_0=19.5dB$	112
	Convolution Coded 32-PSK Constellation at $E_b/N_0=28.5dB$	114
4.13	Spectrum Scope of Convolution Coded 32-PSK at 5.5MHz	115
4.14	ATCM/QPSK Downlink Block Model	117
4.15	TCM/QPSK Constellation at $E_b/N_0= 7.0dB$	118
4.16	TCM/8-PSK Constellation at $E_b/N_0= 9.5dB$	
	119	
4.17	TCM/16-PSK Constellation at $E_b/N_0= 13.0dB$	
	120	
4.18	TCM/32-PSK Constellation at $E_b/N_0= 18dB$	
	121	
4.19	TCM/8-PSK Downlink Block Model	122
4.20	Plot of BER versus $E_b/N_0$ for a 4-State TCM/8-PSK	
	123	
4.21	Plot of BER versus $E_b/N_0$ for a 8-State TCM/8-PSK	
	124	
4.22	Plot of BER versus $E_b/N_0$ for a 16-State TCM/8-PSK	125
4.23	Plot of BER versus $E_b/N_0$ for a 32-State TCM/8-PSK	125
4.24	Plot of BER versus $E_b/N_0$ for a 64-State TCM/8-PSK	126
4.25	Plot of BER versus $E_b/N_0$ for a 128-State TCM/8-PSK	
	127	
4.26	Multiplexed Multi-State TCM/8-PSK on Downlink	
	129	

## LIST OF ACRONYMS

ADR	Adaptive Discrete Rate
AM-DSBSC	Amplitude-Modulation with Double-Sideband-Suppressed-Carrier
ARQ	Automatic Retransmission Request
AWGN	Additive White Gaussian Noise
AX.25	Amateur form of X.25 (packet radio transmission)
BER	Bit Error Rate
bertool	bit error rate (analysis) tool
C/N	Carrier power to Noise power ratio
dB	Decibel
$E_b/N_0$	Energy per bit to Noise spectral density ratio
EIRP	Effective Isotropic Radiated Power
FEC	Forward Error Correction
FSL	Free Space Loss
GEO	Geosynchronous Orbit

Gbps	Giga bits per second
HPA	High Power Amplifier
Hz	Hertz
I-Channel	In phase channel (for QPSK scheme)
IEEE	Institute of Electrical and Electronics Engineers
LEO	Low Earth orbit
LLR	Log Likelihood Ratio
LNA	Low noise amplifier
MATLAB	
Mbps	Megabits per second
MHz	Mega Hertz
MPSK	Multi-Phase Shift Keying
MQAM	Multilevel Quadrature Amplitude Modulation
MSED	Minimum Squared Euclidean Distance
NASRDA	National Space Research and Development Agency
PDF	Probability Density Function
QAM	Quadrature Amplitude Modulation
Q-Channel	Quadrature Channel (for QPSK)
QFH	QuadriFilar Helix (antenna)
QPSK	Quadrature Phase Shift Keying
RF	Radio Frequency
SED	Squared Euclidean Distance
SNR	Signal to Noise ratio
Sps	Symbol per seconds
TCM	Trellis Coded Modulation

## ABSTRACT

The line of sight communication between an orbiting LEO satellite and its Ground Station is time constrained by duration of visibility and number of visible passes in a day. In Equatorial region like Nigeria, the average duration is short and number of passes is very few. In most cases, the satellite size is small and therefore power generated on-board is also small, consequently the downlink budget is power constrained. Transmission bandwidth is also constrained by Regulations and need for cost effective RF design. High resolution remote sensing LEO satellite that acquires large data to be downloaded to a Ground station requires high capacity downlink at minimum power that guarantee BER of  $10^{-6}$ . Application of high level MPSK for high data rate requires more power to reduce transmission errors. Convolution or Block codes when used for Error Correction adds overhead bits to the detriment of the link capacity and without significant coding gain for a power limited downlink. Trellis Coded Modulation (TCM) developed in 1982 by Ungerboeck et al, though still adds overhead bits, but generates significant coding gain that can be extended up to 6dB. This Thesis seeks to optimise LEO Satellite Downlink that is Time, Power and Bandwidth constrained, through trade-off between the coding gains of TCM for bandwidth efficient high level MPSK schemes.



## **CHAPTER ONE**

### **GENERAL INTRODUCTION**

#### **1.0 Introduction**

A remote sensing satellite is launched into Low Earth Orbit (LEO) to acquire data from the Earth surface. When viewed from the Earth, the satellite in LEO does not appear stationary as in the case of a satellite in Geostationary orbit (GEO), instead it is observed to rise from horizon and travel along its orbit until it sets on the other side of horizon.

Telemetry and acquired data on the satellite are communicated to Earth or Ground Station via Downlink, while command and control signals from the Ground Station are communicated to the satellite via Uplink. Both the Downlink and Uplink are established only within line of sight. Because of that antenna dish of the Ground Station tracks LEO satellite from horizon to horizon. The number of times a LEO satellite passes over a Ground Station are few in a day and its durations usually last a few minutes depending on altitude of the orbit and latitude of the Ground Station.

Selection of orbit altitude is determined by satellite mission and Ground stations located in Equatorial region have shorter duration and less number of satellite passes in a day than those in high latitude. There are various forms of low earth orbit orientation, but in general all have limited duration of visibility and number of passes in a day. [1]

In most remote sensing missions, LEO satellites are designed to be small in size which limits on-board power generation, consequently the downlink and other satellite subsystems are power limited. Also most LEO satellites are designed to be cost effective, for that reason wide bandwidths are avoided in the communication subsystem.

In a Downlink digital communication data bits are subjected to error control or channel coding before modulation. This is necessary in low power link and noisy channel to improve reliability of received data. However, coding in general requires overhead bits, which reduces data throughput of the link or requires additional bandwidth. To conserve bandwidth at improved data throughput some high level digital modulations with good spectral (bandwidth) efficiency are used, but this also requires more transmission power.

#### **1.1 Motivation**

When Nigeria launched its first remote sensing satellite Nigearsat-1, in September, 2003, it was launched into Low Earth Orbit of altitude 686 Km and inclination of  $98.19^\circ$ , with mission objective to obtain optical image of the Earth surface at 32 m resolution. A Ground station for the Satellite was built in Abuja, which experiences two or three communication passes daily. Many of the daily passes are too short for communication

purpose, the maximum duration was 10 minutes, which occurred at 5 days intervals.

A low power, high rate transmitter of 8Mbps was used to download image taken from any part of the globe. Optimisation of downlink capacity for LEO Satellite is therefore important in Nigeria and Equatorial region

### **1.2 Problem Definition**

The downlink of Nigeriasat-1 is constrained in power, bandwidth, and communication time. Daily data throughput on the downlink transmissions was not adequate to download the whole contents of the image recorder, the contents have to be fragmented and downloaded over many days.

In LEO satellite communications convolution coding and Multi-Phase Shift Keying (MPSK) are two preferred techniques used for error control coding and digital modulation respectively [1, 11]. Convolution coding has maximum gain of 3 dB which is not sufficient for the problem stated. Trellis Coded Modulation (TCM) is a technique that applies coding and modulation differently from conventional communication systems. It combined convolution coding and MPSK concurrently to yield high coding gain, while maintaining high spectral efficiency of MPSK.

### **1.3 Objective of the Study**

This research work seeks to harness communication time, coding gain and spectral efficiency in digital communications for optimisation of data throughput by:

- Evaluation of orbital factors that restrict visibility of LEO satellite to maximise communication time in Equatorial region.
- Evaluation and simulation of the limitation of convolution codes in LEO satellite downlink.
- Evaluation and simulation of the power constrain for using higher MPSK in LEO satellite downlink.
- Evaluation and Simulation of the potentials of TCM/MPSK for optimisation of data throughput in LEO satellite downlink.

### **1.4 Significance of the Study**

Technological trend in remote sensing satellite is moving toward using high resolution sensor or camera this implied increased downlink data size.[1] Optimisation of downlink data throughput therefore is imperative in the communication subsystem of future remote sensing satellites, especially for Earth Stations in regions that have few and short communication passes.

It is also expected that results of this research thesis would be useful in other wireless terrestrial transmission that are constrained in power, bandwidth and communication time.

### **1.5 Scope of the Study and Methodology**

The Shannon-Hartley Theorem stated that for error free communication the capacity of a transmission link can be increased by increasing its signal-to-noise ratio (SNR) and/or bandwidth. In LEO satellite downlink both the transmission power and bandwidth are limited. This work will seek apparent power and bandwidth from TCM/MPSK to increase the capacity of the downlink.

This research work will confine itself to the analytical method and MATLAB simulations of various downlink models to harness coding gain and spectral efficiency for the optimisation. The approach is similar to some past researches on optimisation of LEO satellite downlink in which modulations, error control codings and other communication protocols were harnessed. [7, 8, 9, 10]

## **CHAPTER TWO**

### **LITERATURE REVIEW**

#### **2.0 INTRODUCTION**

This chapter presents theoretical background on some major concepts of LEO satellite Communications. It begins with orbital mechanics to determine line of sight communication with the satellite. The communication channel for LEO Satellite in this work is considered to be Additive White Gaussian Noise channel (AWGN) with Large Free Space Loss (FSL) as a major link impairments. [1]

Link equation for the downlink, in terms of SNR or Carrier to Noise ratio (C/N), is presented. Then using Shannon's Channel Capacity Theorem, transmission bandwidth and C/N are identified as key parameters for optimization of transmission rate. Phase shift keying (PSK) is introduced as bandwidth efficient bandwidth digital modulation scheme, that is suitable for LEO satellite downlink

Errors in data communication due to white noise and weak signal are treated and Probabilities of error ( $P_e$ ) for some MPSK schemes were derived from the first principle. The concept Bit Error Rate (BER) as a measurable parameter and its relation to  $P_e$  is presented. A brief introduction is made on the concept of Error control and coding gain. The effect of spectral efficient schemes in power and bandwidth limited channel is discussed using Shannon's link capacity Theorem.

At second part of the chapter some past research works on optimization of LEO satellite communication link are presented.

#### **2.1 Low Earth Orbit and Ground Track**

The nature of Low Earth Orbit is determined by mission objectives and payload operational requirements. The altitude is determined through trade-offs between maintaining the satellite in its orbit in presence of atmospheric drag, space radiation and camera or sensor resolution. In practice, orbit altitude of 450-800 km range is required for LEO satellite mission expected to last 5 years. [1].

Orientation of the orbit also depends on the mission of the satellite. Typical orientations are Earth synchronous and Sun-synchronous [1]. Earth synchronous orbit is non-circular and the orbit plane is made to rotate at same rate with Earth rotation (i.e. 360° in 24 hours). This is useful for satellite mission that requires daily revisit to a particular area, but not applicable in remote sensing missions for global coverage. Sun-synchronous also known as Polar orbit, the orbit is circular, its plane is made to incline at close 90° to the equatorial plane, and rotate regressively at same rate with the Earth orbiting the Sun (making 360° rotation in 365.25 days). In this form, the orbit plane remains at same orientation to the Sun throughout the year. This is what makes it Sun-synchronous, it is very useful for optical remote sensing and missions with near global coverage, its disadvantages are short visibility duration and direct overhead (maximum duration) pass occurs at interval of 5 days. This thesis work is on Circular Sun-synchronous Low Earth Orbit.

In Sun-synchronous orientation, the satellite orbit the Earth in about 90 minutes, within this period the Earth rotate eastward so that after each orbit, its ground track is displaced westward. It is the extent of this displacement that determines the visibility of satellite in the next orbit. The ground track displacement is determined as follows.

Let  $P_E$  be period for earth rotation (one day) and  $P$  be satellite orbital period of orbit in figure 2.1. During orbital period  $P$ , the Earth rotate eastward by angle  $\theta_1$  and the satellite orbit plane regression (rotation) westward by angle  $\theta_2$ . Then westward angular displacement of the ground track due to earth rotation  $\theta_1$  is given by: [5]

$$\theta_1 = -2\pi \frac{P}{P_E} \dots \dots \dots 2.1$$

Assuming eastward displacement to be positive.

If  $P_{ES}$  is the period orbit plane rotation about its axis (one year), then ground track displacement eastward due to this rotation is  $\theta_2$  given by [5]:

$$\theta_2 = 2\pi \frac{P}{P_{ES}} \dots \dots \dots 2.2$$

Net ground track longitudinal displacement is

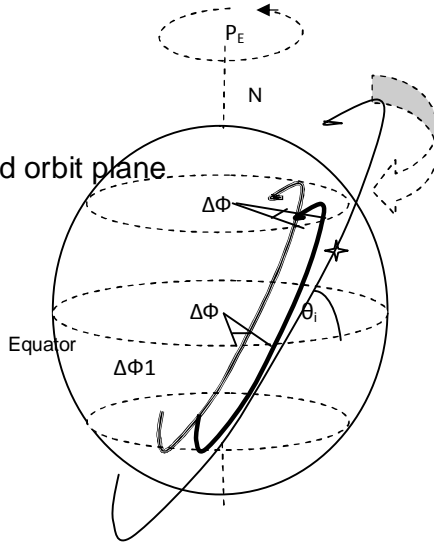
$$\Delta\Phi = \Delta\Phi_1 + \Delta\Phi_2 \dots \dots \dots 2.3$$

Eastward Earth rotation  $\Delta\Phi_1$

Latitude  $\theta_i$  Westward orbit plane

Rotation  $\Delta\Phi_2$

LEO satellite



Latitude  $\theta_i$

Figure 2.1 Low Earth Orbit Satellite and Ground Track Displacement.

Since  $P_E = 24 \times 60 \times 60$  sec and  $P_{ES} = P_E \times 365.25$  sec, then

$$\begin{aligned} \Delta\Phi &= 2\pi \left[ \frac{1}{P_E} + \frac{1}{365.25 P_E} \right] \\ &= 2\pi \frac{1}{P_E} \times 0.9973 \dots \dots \dots 2.4 \end{aligned}$$

According Kepler's Third law of orbital mechanics, which states that: The square of the orbit period P is proportional to the cube of the mean distance between the two bodies [3].

In our case the mean distance is from the Earth centre to the satellite, (i.e. Earth radius  $R_E$  plus height above the earth h)

Using the law for a circular orbit, P can be written as

$$P = 2\pi \sqrt{\frac{R_E^3}{\mu}} \dots \dots \dots 2.5$$

where,  $\mu$  is Kepler's constant (a product of universal gravitational constant G and mass of the Earth) =  $3.986 \times 10^5 \text{ km}^3/\text{kgs}^{-2}$  and Earth radius  $R_E = 6378.14 \text{ km}$ .

**a) Ground Track Linear Displacement along the Equator**

Let linear displacement of the ground track along the equator be  $x$ , then using geometry of the Earth

$$\Delta x = \Delta P \dots \dots \dots 2.6$$

Where  $R_E =$  radius of the equator

Using equation 2.04

$$\Delta x = 2\pi \frac{R_E}{P} \dots \dots \dots 2.7$$

Again substituting for P from equation 2.7

$$= 4\pi \frac{R_E}{P} \frac{R_E^3}{\mu} \dots \dots \dots 2.8$$

$$= 2.914 \frac{6378.14 + h}{3.986 \times 10^5}$$

Assuming orbit height  $h = 800\text{km}$

$$\therefore \Delta x = 2806.9 \text{ Km}$$

**b) Ground Track Linear Displacement along Latitude  $L_t$**

Let linear displacement of the ground track along latitude  $L_t$  be  $x_t$ , then using geometry of the Earth,

$$\text{Radius of a latitude } L_t = (R_E \cos L_t)$$

Therefore corresponding linear displacement  $\Delta x_t$  is given by:

$$\Delta x_t = \Delta x \cos L_t \dots \dots \dots 2.9$$

$$= \Delta x \cos L_t$$

For orbit height  $h = 800\text{km}$

$$\Delta x_t = 2806.9 \cos L_t$$

Ground track linear displacement between to two succeeding orbits at various latitudes are obtained using equation 2.9, the results are tabulated in table 2.1

Table 2.1 Linear Displacement of LEO Ground Track

S/N	Ground Station Latitude $L_t$	Ground Track linear Displacement $\Delta r'$ (km)
1	0°	2806.96
2	15°	2711.25
3	30°	2430.84
4	45°	1984.77
5	60°	1403.45
6	75°	726.479
7	85°	244.64

It can be seen from Table 2.1 that at higher latitudes the linear displacements is shorter than at Equator. This implied that at higher latitude the ground tracks of succeeding overlap more than at lower latitude e.g. in the Equator region. This implied there are more number of visible orbits on higher latitudes than on the Equatorial region [1].

Further orbital analysis to determine width of ground track is extensive and beyond the scope of this work [1].

## 2.2 Geometry of LEO Satellite Visibility

To obtain duration of visibility for a LEO satellite, a satellite pass directly overhead an Earth station at B is considered as shown in figure 2.02, if the Earth is assumed to be a perfect sphere of radius  $R_E$ , the satellite is visible on arc AE, which subtends angle  $2\theta$  at Earth centre C. When the satellite just appears above the horizon, communication along line of sight AB would be largely affected with blockage and multipath due surrounding hills and buildings. Therefore, no effective communication is established along orbit arc AD, until the satellite rises to certain angle of elevation above horizon [5]. A clear line of sight communication is established as the satellite moves along the arc DF, similarly along the arc EF there is also no communication.



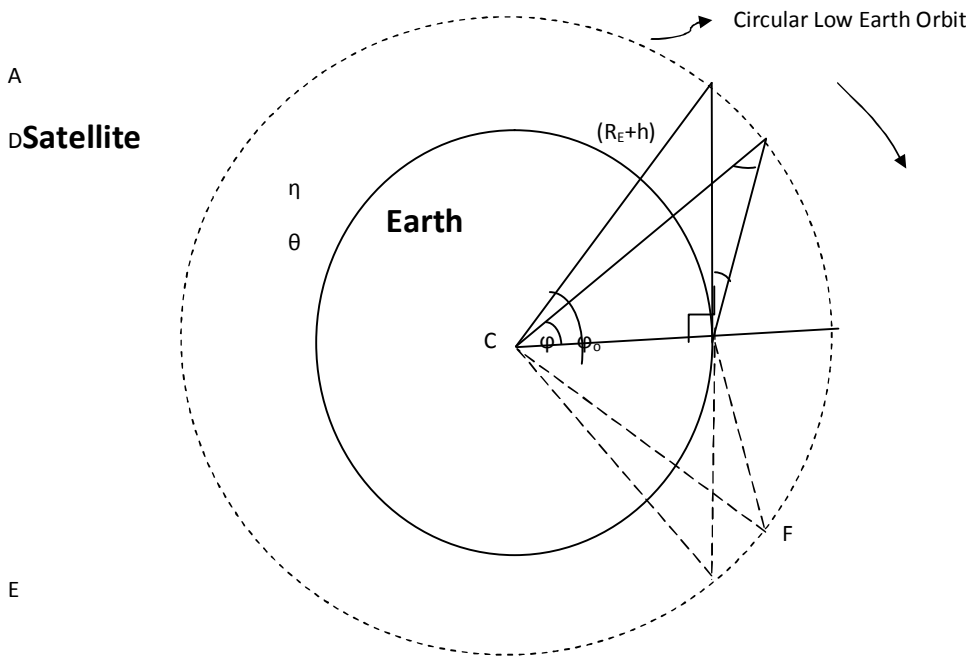


Figure 2.2 Geometry of LEO Satellite Visibility [5]

The two arcs AE and DF subtend angle  $2\phi_0$  and  $2\phi$  respectively at Earth centre C. To calculate visibility angle  $\eta$ , the Sine rule is applied to triangle BCD on figure 2.2, so that:

$$\frac{R_E + h}{\sin(90^\circ + \eta)} = \frac{R_E}{\sin \eta}$$

$$\therefore \frac{R_E + h}{\sin(90^\circ + \eta)} = \frac{R_E}{\sin \eta} \dots \dots \dots 2.10$$

But  $\eta = (180^\circ - (90^\circ + \eta)) = (90^\circ - \eta)$ ,

Simplifying equation 2.10

$$\frac{R_E + h}{\cos \eta} = \frac{R_E}{\cos(90^\circ - \eta)}$$

$$\therefore \cos(90^\circ - \eta) = \frac{R_E \cos \eta}{R_E + h}$$

$$\therefore \eta = -\eta + \cos^{-1} \left[ \frac{R_E \cos \eta}{R_E + h} \right] \dots \dots \dots 2.11$$

Also let  $\omega_s$ ,  $\omega_E$  and  $\omega_{ES}$  be angular velocity of the Satellite, Earth and relative angular velocity of the satellite to Earth. The three angular velocity are related as given by.[5]

$$\omega_{ES}^2 = \omega_s^2 + \omega_E^2 - 2\omega_s\omega_E \cos i \dots \dots \dots 2.12$$

$i$  is orbit inclination angle and in Sun synchronous orbit,  $i \sim 90^\circ$ .

$$\therefore 2\omega_s\omega_E \cos i \sim 0$$

But  $\omega_s = \frac{2\pi}{24 \times 60 \times 60} = 7.272 \times 10^{-5} \text{ rad/s}$  Since  $\omega_E$  is

very small, it is sufficient to use  $\omega_{ES} \approx \omega_s$  in equation 2.12

Angular velocity of a satellite in circular orbit of period P is

$$\omega_s = \frac{2\pi}{P}$$

Substituting for P using the Kepler's law from equation 2.07

$$\omega_s = \frac{2\pi}{\frac{2\pi R_E^3}{GM} + \frac{2\pi R_s^3}{GM}} \dots \dots \dots 2.13$$

And substituting for  $GM = 3.986 \times 10^5 \text{ km}^3/\text{s}^2$  and  $R_E = 6378.14 \text{ km}$  in equation 2.12

$$\omega_s = \frac{631.34}{\sqrt{6378.14 + R_s^2}}$$

To determine the maximum duration of visibility  $\Delta t$  for direct overhead pass full visibility angle  $= 2\theta$  is used and is given as [5]

$$\theta = \frac{2\theta}{2} \dots \dots \dots 2.14$$

$$\therefore \Delta t = \frac{2\theta \sqrt{6378.14 + R_s^2}}{631.34} \dots \dots \dots 2.15$$

Maximum durations of visibility are calculated for various orbital heights and tabulated in table 2.2. Range of orbit heights from 400 to 1000 km are considered from three visibility elevation angles of  $0^\circ$ ,  $5^\circ$  and  $10^\circ$  in table 2.2(a), (b) and (c) respectively. Below  $5^\circ$  elevation angle, the line of sight communication suffers blockage and multipath, but has longer periods of visibility. Above  $5^\circ$  is considered adequate in most cases, except in highly urbanised locations even at that the maximum duration of visibility/communication is less than 15 minute, which occurs rarely (one in 5 days). At elevation above  $10^\circ$ , the communication is free but with short durations, hence the need for optimisation of data throughput.

Table 2.2 Visibility Duration of LEO Satellite referenced to:

(a) Look-up Elevation Angle = 0

S/N	Orbit Height (km)	Visibility Angle at Earth centre 2 (degrees)	Visibility Angle at Earth Centre 2 (radians)	Duration of visibility (sec)	Duration of Visibility (Min)
1	400	39.56°	0.6905	610.33	10.17
2	500	43.96°	0.7673	693.27	11.55
3	600	47.96°	0.8354	771.33	12.85
4	700	51.39°	0.8970	846	14.1
5	800	54.61°	0.9532	918.24	15.3
6	900	57.59°	1.0052	988.56	16.47
7	1000	60.36°	1.0535	1057.51	17.62

(b) Look-up Elevation Angle = 5°

S/N	Orbit Height (km)	Visibility Angle at Earth centre 2 (degrees)	Visibility Angle at Earth Centre 2 (radians)	Duration of visibility (sec)	Duration of Visibility (Min)
1	400	29.56°	0.516	456	7.6
2	500	33.96°	0.593	535.67	8.93
3	600	37.87°	0.6609	610.22	10.17
4	700	41.39°	0.7224	681.43	11.36
5	800	44.61°	0.7787	750.11	12.5
6	900	47.59°	0.8306	816.911	13.62
7	1000	50.36°	0.8789	882.25	14.7

(c) Look-up Elevation Angle = 10°

S/N	Orbit Height (km)	Visibility Angle at Earth centre 2 (degrees)	Visibility Angle at Earth Centre 2 (radians)	Duration of visibility (sec)	Duration of Visibility (Min)
1	400	19.56°	0.3414	301.76	5.03
2	500	23.96°	0.4182	377.84	6.29
3	600	27.96°	0.488	450.57	7.51
4	700	31.39°	0.5479	516.75	8.61
5	800	34.61°	0.6041	581.88	9.69
6	900	37.59°	0.6561	645.26	10.75
7	1000	40.36°	0.7044	707.12	11.78

### 2.3 Downlink Budget Analysis of a LEO Satellite

The performance of a digital communication link is measured by its capacity and probability of error, which are both determined by received Signal to Noise Ratio (SNR) or Carrier to Noise ratio (C/N).[8] The aim in satellite downlink analysis is to find value of C/N that guarantee low probability of error for a determined link capacity.

A schematic diagram of a LEO satellite downlink is shown in figure 2.3. The satellite is at distance  $S$  from its Earth Station (ES). If the satellite transmitter power is  $P_T$ , antenna gain  $G_T$  and feeder Loss  $L_T$ . At its Earth station the receiver antenna gain is  $G_R$ , feeder loss  $L_R$ . And other losses on the link are free space loss  $L_S$  and atmospheric absorption  $L_A$ .

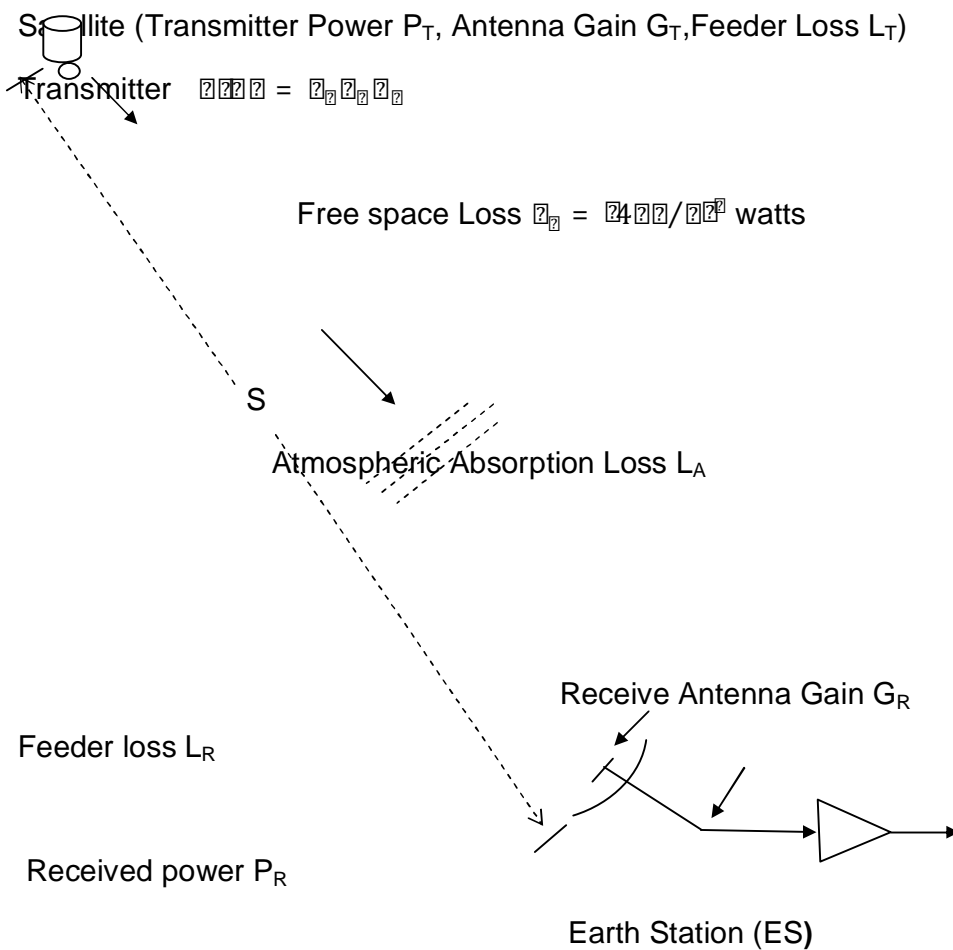


Figure 2.3 A Sketch of LEO satellite Downlink and its Components [3]

Using Friis's link equation.[3] Received power of the downlink of  $P_R$  is given by:

$$P_R = P_T \left( \frac{A_R}{4\pi R^2} \right) \left( \frac{A_T}{4\pi R^2} \right) \dots \dots \dots 2.16$$

Where transmitter Effective Isotropic Radiated Power  $P_{EIRP} = P_T A_T$  and Free Space loss  $L_S = 4\pi R^2 / \lambda^2$  with  $R$  is transmission path length and  $\lambda$  is transmission signal wavelength.

Then,

$$P_R = P_T \left( \frac{A_R}{4\pi R^2} \right) \left( \frac{A_T}{4\pi R^2} \right) \dots \dots \dots 2.17$$

The received power  $P_R$  is typically very small, in order of picowatts which tend to be corrupted by Thermal noise. Thermal noise arises from electromagnetic signals generated by Thermodynamic motions of electrons within the receiver active and passive components, and also antenna picked-up galactic radiation. Thermal noise is generally characterized to be white because of its uniform spread across wide RF frequency spectrum with random Gaussian distribution.

At elevation of 5-10° above the horizon, the link is free from terrestrial interference, so that only thermal noise is considered. The link is therefore characterised as operating in Additive White Gaussian Noise channel.

Thermal noise power  $N$  is calculated as function of noise temperature and receiver noise bandwidth.[1]

Noise power  $N$  is given by [1]:

$$N = k T_S B_n \dots \dots \dots 2.18[1]$$

Where

$k$  = Boltzmann's constant =  $1.39 \times 10^{-23}$  J/K

$T_S$  = Equivalent noise temperature of the system

$B_n$  = Receiver noise bandwidth measured in Hz

Then using equation 2.17 and 2.18, the downlink equation can be expressed as ratio of carrier power to noise power C/N:

$$\frac{C}{N} = \frac{P_R}{N} = \frac{P_T \left( \frac{A_R}{4\pi R^2} \right) \left( \frac{A_T}{4\pi R^2} \right) \dots \dots \dots}{k T_S B_n} \dots \dots \dots 2.19$$

For digital communication the equation is written in form energy per bit ( $E_b$ ), defined as product of received power  $P_R$  and bit duration  $T$ [1]

$$E_b = P_R T \dots \dots \dots 2.20$$

$$= \frac{P_R}{R} \dots \dots \dots 2.21$$

Where bit rate  $R = 1/T$

But in AWGN, noise powerspectral density ( $N_o$ ) is uniform:

$$N_o = \frac{P_n}{B} = \frac{P_n}{R} \dots \dots \dots 2.22$$

Then taking energy per bit to noise spectral density ratio ( $E_b/N_o$ ) as subject of equation 2.17[1]

$$\frac{E_b}{N_o} = \frac{P_R}{P_n} \times \frac{P_R}{R} = \frac{P_R^2 T}{P_n R} \dots \dots \dots 2.23$$

Substituting the value for  $k$  in equation 2.23 and taking the logarithm to express the parameters in decibel dB

$$\frac{E_b}{N_o} = 10 \log P_R + 10 \log T + 10 \log P_R - 10 \log R + 228.6 - 10 \log P_n - 10 \log B \dots \dots \dots 2.24$$

In this is downlink equation  $E_b/N_o$  is a function of transmission power (EIRP), losses ( $L_T, L_S, L_A, L_R$ ), receiving antenna sensitivity ( $G_R/T_s$ ), and transmission rate ( $R$ ).[1]

The major variables in link equation are:

- a. Transmitter EIRP; where transmitter power  $P_T$  is limited because power generated on a small satellite is small and has to be shared among all other on board satellite subsystems. Limited transmitter antenna gain  $G_T$  can be realized, because it has to be small proportional to the satellite size, therefore EIRP is limited.
- b. Receiving antenna gain  $G_R$ ; size of antenna at Earth Station is also limited for tracking purposes and to minimize antenna thermal noise. Thus antenna sensitivity  $G_R/T_s$  is also constrained.
- c. Transmission losses; Feeder losses ( $L_T$  and  $L_R$ ) can be minimized; while atmospheric loss is somewhat a random process. Free space loss ( $L_S$ ) is the major variable and it varies inversely to square of the range ( $S^2$ ) and transmitting frequency (which is predetermined by regulations). The distance  $S$  varies significantly with time, it makes the LEO communication channel to have wide variation in path loss [7].

Satellite distance is maximum ( $S_{max}$ ) when the satellite just appeared above the horizon or at a given elevation angle  $\theta$ , by using figure 2.02  $S_{max}$  is given by:

$$S_{max} = R_e + R_e \sin \theta / \cos \theta \dots \dots \dots 2.25$$

And the range is minimum ( $S_{min}$ ) when the satellite is directly overhead the Earth station

$$S_{min} = R_e \dots \dots \dots 2.26$$

Substituting for  $S_{max}$  and  $S_{min}$  in the formula for free space loss to get the variation as in [7]

$$\begin{aligned} L_{max} - L_{min} = \Delta L &= 20 \log \left( \frac{4\pi R_e^2 \sin^2 \theta}{\lambda^2} \right) - 20 \log \left( \frac{4\pi R_e^2}{\lambda^2} \right) \\ &= -20 \log \frac{\sin^2 \theta}{1} \dots \dots \dots 2.27 \end{aligned}$$

Using equations 2.26 and 2.27

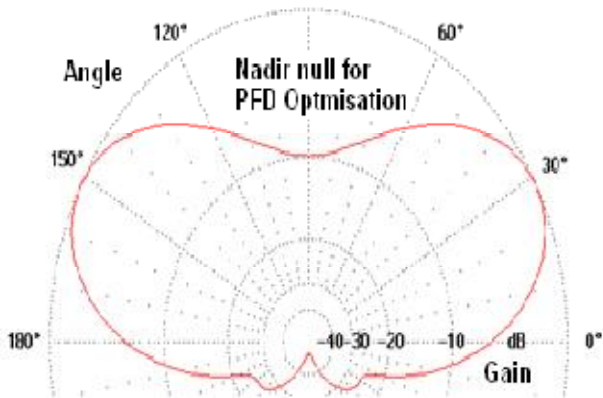
$$\Delta L = -20 \log \frac{R_e + R_e \sin \theta}{R_e \cos \theta} \dots \dots \dots 2.28$$

Thus, using table 2.02 and equation 2.28, a LEO of altitude  $h = 800\text{Km}$ , elevation  $\theta = 5^\circ$  and earth central angle  $\phi = 22.31^\circ$ , its variation in free space loss is given as

$$L_{max} - L_{min} = -10.83 \text{ dB}$$

This variation is very significant and Matar et al applied variable data rate to counter it [7].

In the case of Nigeriasat-1, a specially shaped antenna gain pattern was used to compensate for this large variation (up to 10dB). A Quadri Filar Helix Antenna (QFH) with gain pattern shown in figure 2.4 was used to deliver fairly constant SNR, as the satellite moves along its orbit during communication time



**Figure 2.4** Gain Pattern of QuadriFilar Helix Antenna [18]

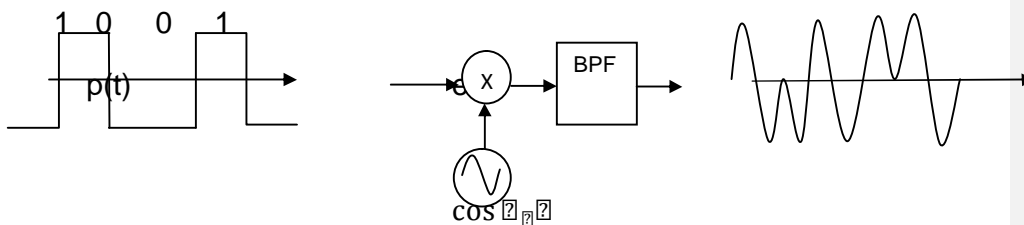
### 2.4 Basic Digital Modulations for Satellite Communications

To guarantee certain quality of signal  $Qos$  (in this case level of error rate), a minimum value  $E_b/N_0$  is required which is determined by the type of modulation used for the transmission.

In digital modulation, any parameters of the carrier - amplitude, frequency or phase - could be digitally altered or modulated. In satellite communication, phase and frequency modulation are generally preferred over amplitude modulation, because the former are less susceptible to power amplifier saturation distortion effect [1].

This thesis is based on digital phase modulation known as Multi-Phase Shift Keying (MPSK). Some basic forms of MPSK are:

- a) **Binary Phase Shift Keying (BPSK):** This is basically a phase reversal technique using bipolar digital signal on sinewave carrier. Basic amplitude-modulation with double-sideband-suppressed-carrier (AM . DSBSC) produces the effect of phase reversal, when non-return to zero pulses is used as modulating signal..



**Figure 2.5** Block Diagram of BPSK Implementation [2]



When binary signal is +1 as shown in figure 2.5, the carrier sinusoid is unchanged but when it is -1 the carrier sinusoid is changed in phase by 180°. The modulated wave e(t) is a linear product of the two inputs written as

$$e(t) = A_m \cos \omega_c t \dots \dots \dots 2.29$$

When  $m(t) = +1$ , then

$$e(t) = \cos \omega_c t$$

And when  $m(t) = -1$ , then

$$e(t) = -\cos \omega_c t$$

$$e(t) = A_m \cos(\omega_c t + 180^\circ) \dots \dots \dots 2.30$$

It is known from communication theory that bandwidth ( $B_{if}$ ) of modulated AM-DSBSC signal is twice the maximum frequency or bandwidth of the baseband signal ( $f_{max}$  or  $B$ ) and since BPSK is an AM-DSBSC, then it follows that the IF bandwidth of BPSK scheme  $B_{if} = 2B_{bb} = 2B$ .

The modulated signal is passed through a bandpass filter to remove harmonics. Filters in modulator and demodulator set the limit on bandwidth. In high rate transmission, error also arises in the demodulator due to pulse overlaps (also called Intersymbol Interference (ISI)), the baseband signal need to be passed through pulse shaping filters to remove the overlaps. This pulse shaping filters are designed to satisfy theoretically Nyquist criterion (see APPENDIX A). Therefore, bandwidth for digital transmission is limited by Nyquist filter criteria on ISI. Relationship between filter bandwidth  $B$  and symbol rate  $R_{sym}$  is given by [3]:

$$B = \frac{\alpha + 1}{2} R_{sym} \dots \dots \dots 2.31$$

Where  $\alpha$  is filter roll-off factor  $0 \leq \alpha \leq 1$

For BPSK the input data bit rate  $R_b$  is the modulating (symbol) rate  $R_{sym}$  i.e.  $R_b = R_{sym}$ .

$$\therefore B = \frac{\alpha + 1}{2} R_b$$

Since  $R_b = 2B$

$$\therefore B = 2B = \frac{\alpha + 1}{2} R_b \dots \dots \dots 2.32$$

Bandwidth utilization or spectral efficiency in BPSK is given as

$$\frac{R_b}{B} = \frac{1}{\frac{\alpha + 1}{2}} \dots \dots \dots 2.33$$

Which has maximum value of one.

**(b) Quadrature Phase-Shift Keying (QPSK):**

In this scheme, the phase of the carrier phase is shifted by 90° to four possible values by the baseband input bit stream p(t). So that the modulated signal is

$$s(t) = A \cos(\omega_c t + \theta) \dots \dots \dots 2.34$$

So that  $\theta$ , takes values of  $\pi/4, 3\pi/4, 5\pi/4, 7\pi/4$ .

By expansion

$$s(t) = A \cos \theta \cos \omega_c t + A \sin \theta \sin \omega_c t \dots \dots \dots 2.35$$

The first term of LHS is similar to equation 2.34 or BPSK signal; it is called in-phase or *I-channel*. The second term is also a BPSK signal, but in quadrature phase with the carrier; it is called quadrature or *Q-channel*. Thus a QPSK signal, is a combination of two BPSK signal in quadrature. Such that

$$s(t) = p_i \cos \omega_c t + p_q \sin \omega_c t \dots \dots \dots 2.36$$

Where  $p_i$  and  $p_q$  represent binary stream modulating the I-channel and q-channel respectively. Such that

$$p_i = A \cos \theta \dots \dots \dots 2.37$$

And

$$p_q = A \sin \theta \dots \dots \dots 2.38$$

Table 2.3 Phase angle of I- Channel and Q-Channel in QPSK

$p_i$	$p_q$	
1	1	$\pi/4$
-1	1	$3\pi/4$
-1	-1	$5\pi/4$
1	-1	$7\pi/4$

In both signals, a logical 1 corresponds to  $p_i$  or  $p_q = +1$  and a logical 0 corresponds to  $p_i$  or  $p_q = -1$ , as shown in table 2.3.

Figure 2.6 is a block diagram for implementing QSPK scheme. Incoming modulating bit stream p(t) is passed into a serial-to-parallel converter, which splits it into two parts, in-phase  $p_i$  and quadrature  $p_q$  streams. Such that bits  $p_i$  and  $p_q$  are selected alternately from the input bit stream. The signals are then separately mixed with two orthogonal sinusoidal carriers  $\cos \omega_c t$  and  $\sin \omega_c t$  thus producing two orthogonal BPSK waveforms which are then combined to produce the QSPK modulation.

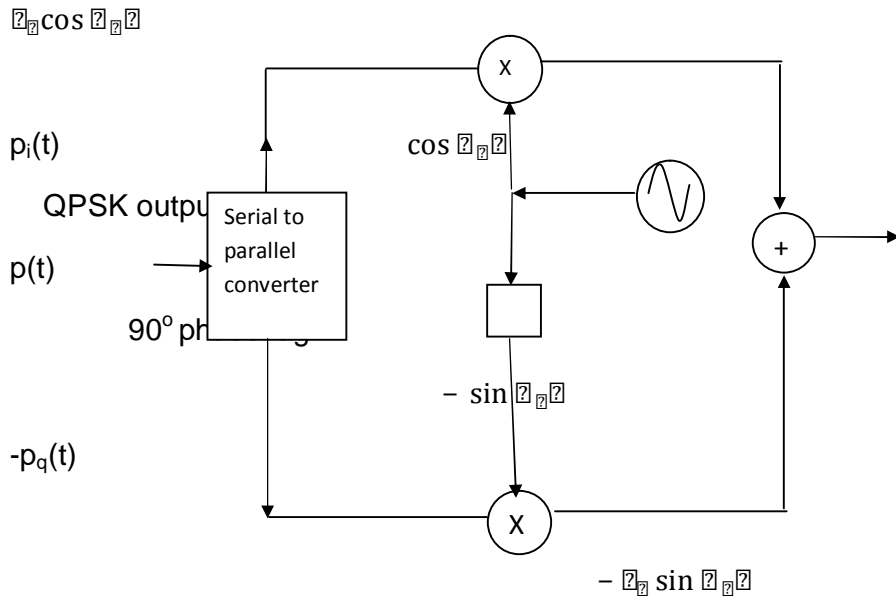


Figure 2.6 Schematic of QPSK Implementation

Table 2.4 is a phase table for the QPSK and its signal point constellation is shown in figure 2.7

Table 2.4 QPSK Implementation [2].

2-bit binary	$p_i$	$p_q$	QPSK
11	1	1	$\cos 2\pi f_c t \cos \frac{\pi}{4} - \sin 2\pi f_c t \sin \frac{\pi}{4} = \cos 2\pi f_c t + \frac{\pi}{4}$
01	-1	1	$\cos 2\pi f_c t \cos \frac{3\pi}{4} - \sin 2\pi f_c t \sin \frac{3\pi}{4} = \cos 2\pi f_c t + \frac{3\pi}{4}$
10	1	-1	$\cos 2\pi f_c t \cos \frac{7\pi}{4} - \sin 2\pi f_c t \sin \frac{7\pi}{4} = \cos 2\pi f_c t + \frac{7\pi}{4}$
00	-1	-1	$\cos 2\pi f_c t \cos \frac{5\pi}{4} - \sin 2\pi f_c t \sin \frac{5\pi}{4} = \cos 2\pi f_c t + \frac{5\pi}{4}$

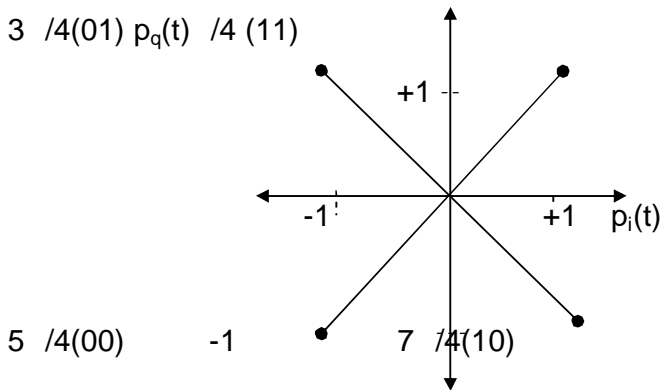


Figure 2.7 QPSK Signal points [2].

To calculate the spectral efficiency for QPSK, since QPSK is a combination of two BPSK and symbol rate is half of the bit rate ( $R_{sym}=R_b/2$ ), then using equation 2.23. [2]

$$\eta_{QPSK} = \frac{2 \log_2 4}{2} \eta_{BPSK}$$

$$\therefore \frac{\eta_{QPSK}}{\eta_{BPSK}} = \frac{2}{2} \dots \dots \dots 2.39$$

In ideal filter  $\eta_{BPSK} = 0$  and  $R_b/B_{IF} = 2$ .

Thus, spectral efficiency QPSK is twice that of BPSK.

In general (to be discussed in chapter 5), for M-PSK the spectral efficiency increases with modulation order. This thesis is aimed to exploit high spectral efficiency of M-PSK to increase channel capacity.

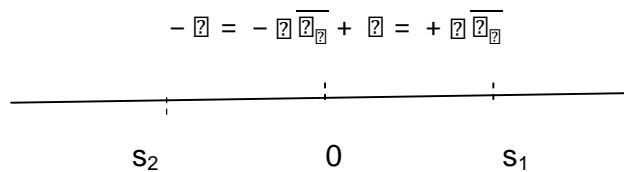
**2.5 Probability of Error (P<sub>e</sub>) in BPSK and QPSK Schemes**

To calculate probability of error due to the noise in the received BPSK signal, consider two voltage levels +V and -V to represent binary 1 and 0 transmitted and let noise voltage received be  $n(t)$ . Then output voltage of the demodulator for decision making is the sum of the signal voltage  $\pm V$  and the noise voltage  $n(t)$ . [6]

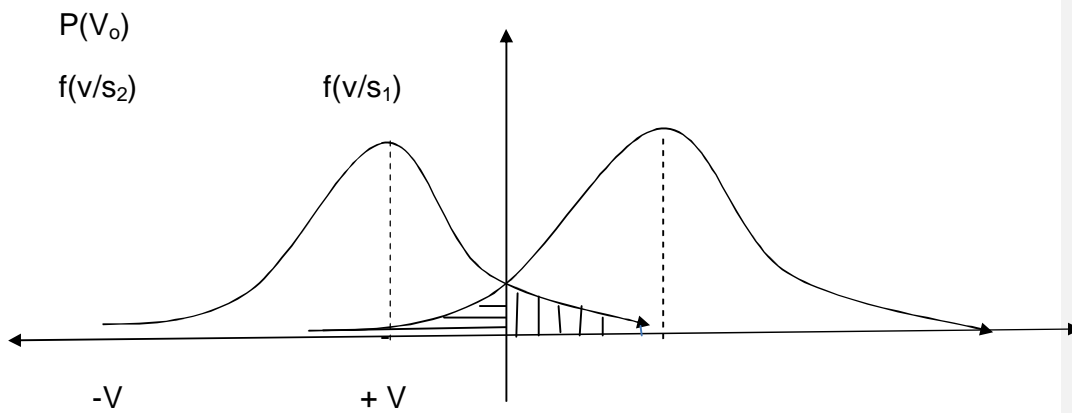
$$V_{out} = \pm V + n(t) \dots \dots \dots 2.40$$

$$V_{out} = \overline{\pm V + n(t)} = \overline{\pm V} + \overline{n(t)} \dots \dots \dots 2.41$$

In figure 2.8(a), the two decision thresholds are represented by antipodal signal points  $s_1$  and  $s_2$  corresponding to binary 1 and 0 respectively. Such that, when  $v > 0$  decision is made in favour of  $s_1$ , and when  $v < 0$  decision is made in favour of  $s_2$ . An error will occur, when the noise voltage  $n$  change sign of  $v$ . There are two possibilities for this error to occur: (a) If  $+V$  signal is sent at a time  $n$  is less than  $-V$ , then a wrong decision is made for negative  $v$ . (b) If  $-V$  signal is sent at a time  $n$  is greater than  $+V$ , then a wrong decision is made for positive  $v$ . Therefore, probability of error occurring  $P_e$  is given as:



(a) Antipodal Signal Points [6]



(b) Conditional PDF for  $s_1$  and  $s_2$  decision [6]

Figure 2.8 Probability Density Function of Antipodal Signal

$$P_e = \frac{1}{2} \int_{-V}^0 f(v/s_1) + f(v/s_2) dv + \frac{1}{2} \int_0^{+V} f(v/s_1) - f(v/s_2) dv \dots \dots 2.42$$

Since the two events are equally likely, it follows

$$P_e = \int_{-V}^0 f(v/s_1) + f(v/s_2) dv + \int_0^{+V} f(v/s_1) - f(v/s_2) dv \dots \dots 2.43$$

$$= \frac{1}{\sqrt{2\pi N_0}} \exp\left\{-\frac{V^2}{2N_0}\right\} \dots \dots \dots 2.44$$

The probability of error occurring in the transmission is now reduced to probability of noise voltage greater in wrong direction than the sample value ( $\pm V$ ). Since the noise is defined to be Gaussian.

The Gaussian PDF for random variable  $x$ , with mean  $\mu$  and variance  $\sigma^2$  is given as

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{(x-\mu)^2}{2\sigma^2}\right\} \dots \dots \dots 2.45$$

Consequently applying the Gaussian expression to this case, the two events have conditional probability density functions (PDF) as follows: [6]

$$f_1(v) = \frac{1}{\sqrt{2\pi N_0}} \exp\left\{-\frac{(v-V)^2}{2N_0}\right\} \dots \dots \dots 2.46$$

$$f_2(v) = \frac{1}{\sqrt{2\pi N_0}} \exp\left\{-\frac{(v+V)^2}{2N_0}\right\} \dots \dots \dots 2.47$$

Taking the noise component to have zero mean and variance of  $\sigma^2 = N_0/2$  Although the two PDF overlapped as shown in figure 2.8(b), but for error to occur, the probabilities for the two events are mutually exclusive as shown in the two shaded areas

Thus, given that  $s_1$  (+V) was transmitted, probability of an error in receiving  $s_1$  (i.e.  $v_o < 0$ ) is the integral of its PDF: [6]

$$P_e = \int_{-\infty}^0 f_1(v) dv \dots \dots \dots 2.48$$

$$= \frac{1}{\sqrt{2\pi N_0}} \int_{-\infty}^0 \exp\left\{-\frac{(v-V)^2}{2N_0}\right\} dv \dots \dots \dots 2.49$$

By changing of variables such that  $v = \frac{v-V}{\sqrt{N_0/2}}$

$$\Rightarrow v = \frac{v-V}{\sqrt{N_0/2}} \Rightarrow \frac{dv}{\sqrt{N_0/2}} = \frac{dv}{\sqrt{N_0/2}}$$

And differentiating,

$$P_e = \frac{1}{2} \int_{-\infty}^{\infty} \dots$$

Also for range of the integration when  $\gamma = 0$ ,  $\gamma = \frac{2E_b}{N_0}$

$$\therefore P_e = 0, \gamma = \frac{2E_b}{N_0}$$

And when  $\gamma = -\infty$ ,  $\gamma = -\infty$

$$\therefore P_e = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \dots \dots \dots 2.50$$

Since the PDF is symmetrical about +V then

$$P_e = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} \dots \dots \dots 2.51$$

The above expression is a statistical Q-function of  $\sqrt{2E_b/N_0}$  so that

$$P_e = Q\left(\sqrt{2E_b/N_0}\right) \dots \dots \dots 2.52$$

Similarly if  $s_2(-V)$  was transmitted,  $\gamma = -V + V$ , then probability of  $\gamma > 0$  is  $Q\left(\sqrt{2E_b/N_0}\right)$  and its PDF is a replica to that of  $Q\left(\sqrt{2E_b/N_0}\right)$

$$\therefore P_e = Q\left(\sqrt{2E_b/N_0}\right) \dots \dots \dots 2.53$$

Since  $s_1$  and  $s_2$  are exclusive, and equally likely events,

Then, the average probability of error  $P_e$  is

$$P_e = \frac{1}{2} P_e + \frac{1}{2} P_e \dots \dots \dots 2.54$$

$$= Q\left(\sqrt{2E_b/N_0}\right) \dots \dots \dots 2.55$$

This can also be expressed using complementary error function erfc as

$$P_e = \frac{1}{2} \text{erfc}\left(\sqrt{E_b/N_0}\right) \dots \dots \dots 2.56$$

A plot  $P_e$  against  $E_b/N_0$  (in dB) for BPSK scheme is shown in figure 2.09. It is established in Communication theory that probability of error for QPSK and BPSK schemes are equal, since QPSK is a combination of two orthogonal BPSK

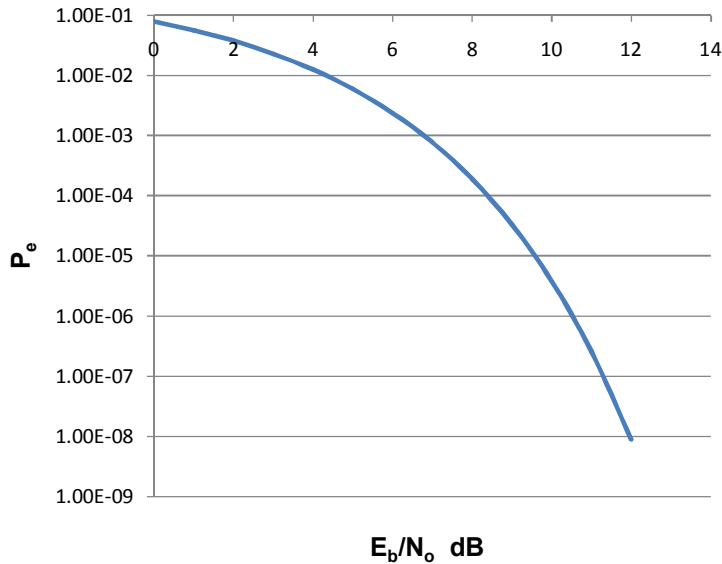


Figure 2.9 Graph of Probability of Error  $P_e$  Vs  $E_b/N_0$  for BPSK and QPSK

For high level MPSK scheme the probability for symbol error  $P_M$  is given as [6]

$$P_M = 2 \left( \frac{E_b}{N_0} \right)^{-1} \sin^2 \left( \frac{\pi}{2M} \right) \dots \dots \dots 2.57$$

Or  $P_M = \left( \frac{E_b}{N_0} \right)^{-1} \sin^2 \left( \frac{\pi}{2M} \right) \dots \dots \dots 2.58$

Plots of  $P_e$  versus  $E_b/N_0$  for QPSK, 8-PSK and 16-PSK are shown in figure 2.10. It is important to note that to maintain same  $P_e$ , as modulation level  $M$  increases more  $E_b/N_0$  is required. This is a major disadvantage for high level MPSK scheme



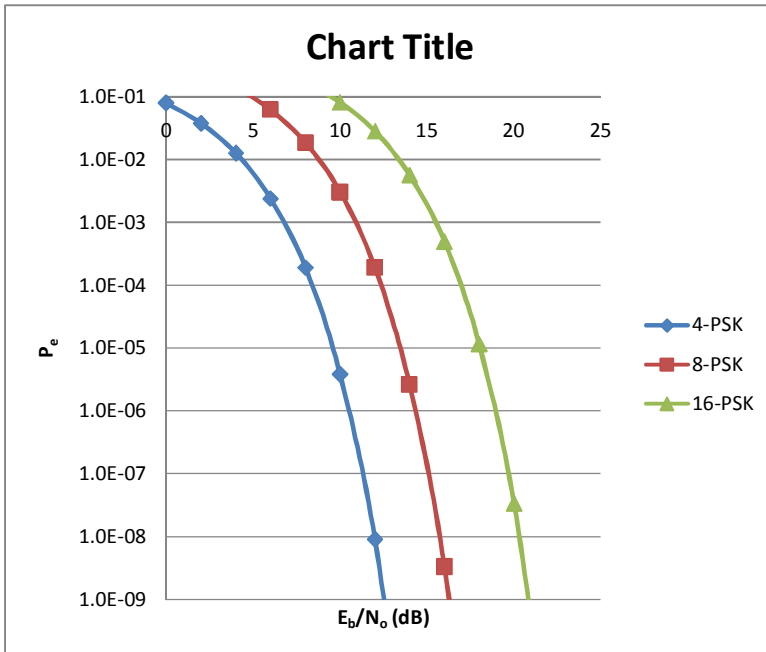


Figure 2.10 Probability of Symbol Error  $P_M$  versus  $E_b/N_0$  in M-PSK

## 2.6 Error Control and Coding Gain

One major advantage of digital communications over analog is possibility of improving quality of (corrupted) digital signal received by use of error control codes. The techniques may involve an addition of extra bits for error detection, as in automatic repeat request (ARQ), [8] in which case the receiver request for retransmission when error is detected; or combined error detection and correction, as Forward Error Correction (FEC). In conventional FEC encoding is implemented before the modulation while decoding is implemented after demodulation.

Where error control code is applied, a (measurable or countable) parameter bit error rate (BER) is used instead of  $P_e$ . In uncoded link  $P_e$  and BER are the same, but in coded link  $P_e$  is determined at demodulator output, while BER is determined at decoder output. When error control code is applied to a link, the required value of  $E_b/N_0$  for a certain BER is reduced as shown by the difference in the two plots of figure 2.11.

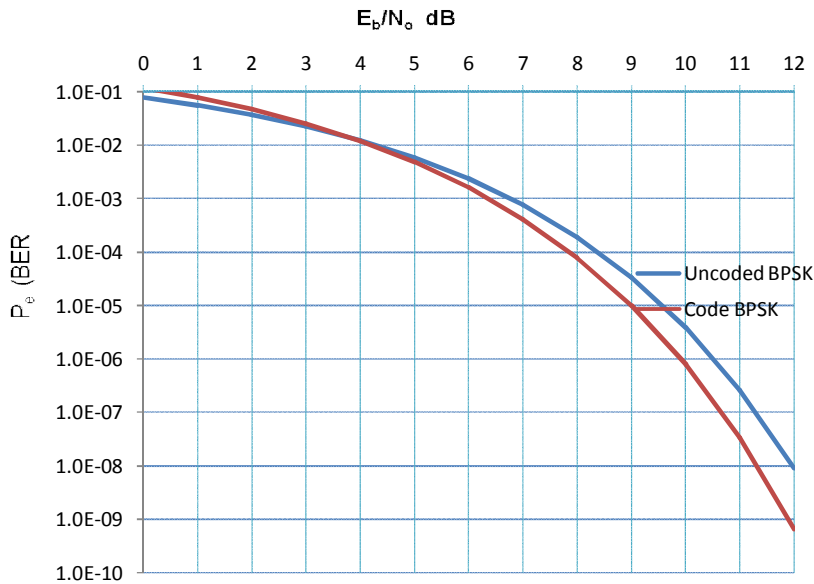


Figure 2.11 Probability of Bit Error in Uncoded and Coded BPSK

It is possible using certain conventional FEC to produce gain in  $E_b/N_0$  of up to 3dB, applying this gain is more convenient than to increase receiving antenna size by 41% or transmitter power by 100%.

### 2.07 Spectral Efficiency in Power and Bandwidth Limited Links

The Shannon's link capacity theorem has set limit for error free communication in an AWGN channel

$$C = B \log_2(1 + C/N) \dots \dots \dots 2.59$$

It expresses the relationship between the Transmission data rate R, channel bandwidth B, and received carrier-to-noise ratio C/N. [1]

Using the theorem, a ratio of data rate to bandwidth known as spectral efficiency is formed:

$$\frac{C}{B} = \log_2(1 + C/N) \dots \dots \dots 2.60$$

Therefore from equation 2.60 spectral efficiency and received C/N are two key parameters that counter balance each other in an AWGN channel. In practice a link is characterized to be limited in power or bandwidth or both.

Using equation 2.33, spectral efficiency can also be expressed in terms of  $E_b/N_o$  as[3]

$$\frac{R}{B} = \log_2(1 + \frac{E_b}{N_o} \frac{R}{B}) \dots \dots \dots 2.61$$

When spectral efficiency of a link is less one (i.e.  $R < B$ ), the link is characterized as power limited, because power is traded for bandwidth. A plot of  $\log_2(1 + \frac{E_b}{N_o} \frac{R}{B})$  versus  $\frac{E_b}{N_o} \frac{R}{B}$  in decibel for power limited link is shown in figure 2.12. The graph shows how  $E_b/N_o$  (power) is traded for low spectral efficiency. It set a lower theoretical limit on  $E_b/N_o$  at -1.6dB for any communication in AWGN regardless of spectral efficiency, it is called Shannon's limit. [4]

In practice some high gain FEC such as Turbo coding allows links to approach  $E_b/N_o$  value of -0.9 dB that is just 0.7 dB above Shannon limit.[3]

In case of  $R > B$ , the link is characterized as bandwidth limited, here bandwidth is traded for power. The spectral efficiency is always greater than one; plot of  $\log_2(1 + \frac{E_b}{N_o} \frac{R}{B})$  versus  $\frac{E_b}{N_o} \frac{R}{B}$  in decibel is shown in figure 2.13. As the ratio  $R/B$  increases (i.e. reduced bandwidth) the value of  $E_b/N_o$ (power) is required to achieve practical communication increases asymptomatic, simply put for higher spectral efficiency more power is required.

The downlink of LEO satellite is both power and bandwidth limited. This work seeks to apply high spectral efficiency of MPSK modulation schemes and harness power from high gain error control coding to optimise data rate.

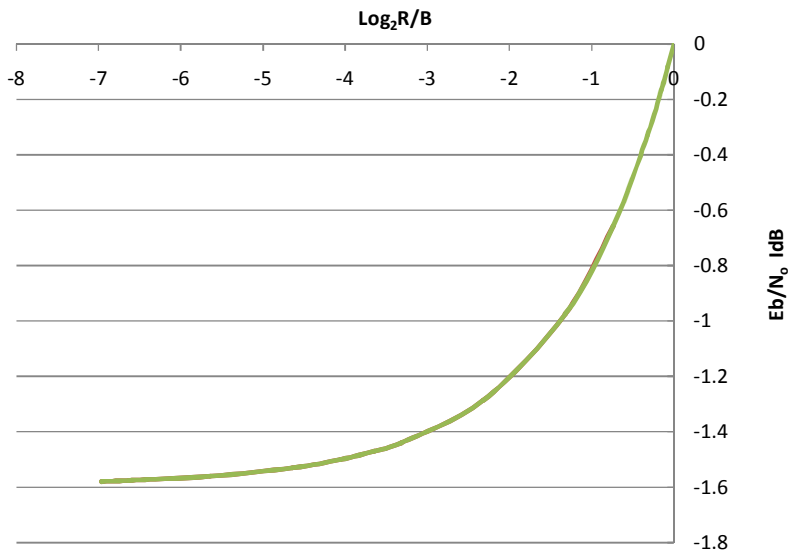


Figure 2.12 Plot  $E_b/N_0$  Versus  $\log_2(R/B)$  for Power Limited link ( $R < B$ )

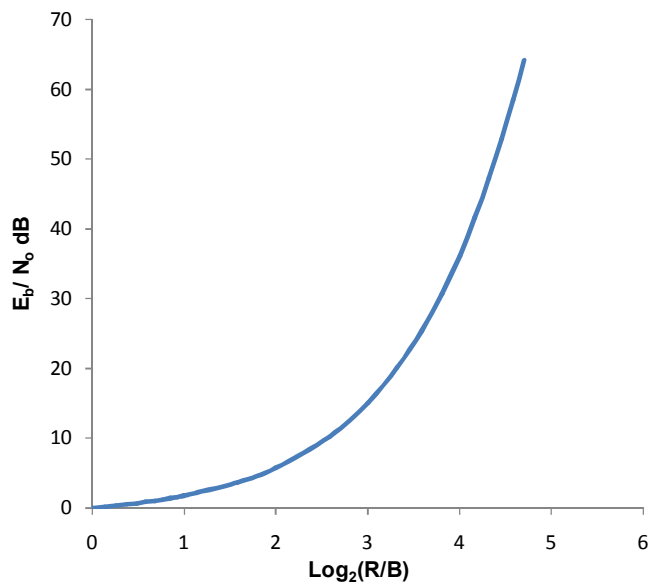


Figure 2.13 Plot of  $E_b/N_0$  Vs  $\log_2(R/B)$  for Bandwidth Limited link ( $R > B$ )

## 2.8 Reviews on Past Publications on Optimisation of LEO Downlink

Researches are still on-going on optimization of data communication between LEO satellites and Earth stations. While the results of some of these researches are commercial proprietary that remains exclusive to the sponsoring organisations, those that have been published in the past IEEE publications focused in the following areas:

- Adaptive Variable Transmission Rate.[7] (2003)
- Hybrid-ARQ Communication protocol. [8](1997)
- Adopting Raleigh and Rician Model for Downlink Channel.[9](2003)

### 2.8.1 Adaptive Variable Data rate

In a paper published in IEEE 2003 by M.A Matar titled "Throughput/Channel Capacity Improvement Issues for LEO satellite downlink" [7] An adaptive variable data transmission rate was proposed to counter the wide variation of free space loss during the communication time. This approach seeks to apply constant SNR by varying the transmission rate according to the satellite orbiting dynamic to maintain fixed BER. The author substituted for free space loss  $L_f = \frac{4\pi S^2}{\lambda^2}$  in the link

equation 2.23:

$$\frac{E_b}{N_b} = \frac{EIRP L_f G_r \lambda / 4\pi S^2 L_f}{T_b k R}$$

$$\frac{E_b}{N_b} = \frac{EIRP L_f}{4\pi S^2 R} \frac{G_r / T_b \lambda^2}{4\pi k L_f} \dots \dots \dots 2.62$$

During communication time, it is only the satellite range S that varies while other parameters in the link equation 2.62 are fixed. It is therefore, power wastage to apply same  $E_b/N_b$  to counter both the minimum and maximum free space losses. Rather, the data rate R can be increased while the free space loss is decreased as the satellite rises from the horizon. Such that, the term  $S^2 R$  is constant, and therefore all terms in equation 2.62 remain constant during communication time. A fixed value of  $E_b/N_b$  is kept at certain threshold to guarantee the required BER.

Such that:

$$S^2 R = \text{constant} = \frac{1}{\sigma} \dots \dots \dots 2.63$$

In figure 2.14, at Earth station B, the satellite A in orbits subtends an earth angle  $\phi$  at C. This earth angle is a function of time as in equation 2.64

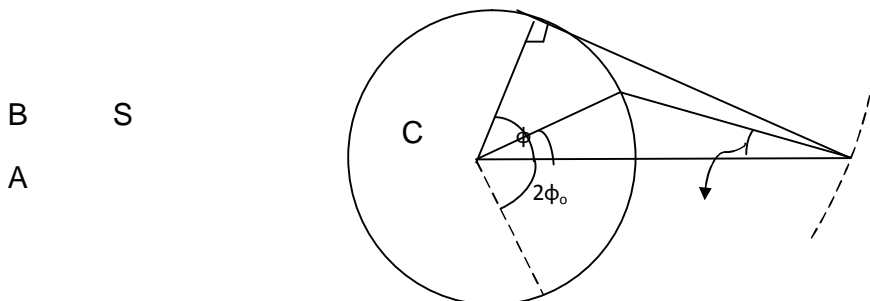


Figure 2.14 Geometry of Satellite-Earth Angular Relationship.

Using the triangle ABC in figure 2.14, the satelliterange AB=S is function of time given by:

$$S^2 = R_E^2 + R^2 + h^2 - 2R_ER \cos \phi \dots \dots \dots 2.64$$

To maintain equation 2.64 an Adaptive Rate Transmission (ART) process was proposed by adopting a strategy of varying symbol duration T with time, so that

$$\sigma S = \frac{1}{R} = T \dots \dots \dots 2.65$$

Such that

$$T = \sigma S \dots \dots \dots 2.66$$

Therefore, volume of data downloaded or throughput TP during the communication time is obtained as

$$TP = \int \frac{dt}{T(t)} = \int \frac{dt}{\sigma S} \dots \dots \dots 2.67$$

For a LEO altitude of  $h = 650\text{km}$  and  $R_E = 6378.14\text{km}$

$$R_E + h \approx 1.21R_E \dots \dots \dots 2.68$$

And if  $\frac{2.21R_b(1 - 0.995475 \cos \varphi(t))}{2.21\sigma R_b(1 - 0.995475 \cos \varphi(t))} \cong 1$

Then equation 2.66 reduces to

$$S^2 = 2.21R_b^2(1 - 0.995475 \cos \varphi(t)) \dots \dots \dots 2.69$$

$$\therefore TP_{var} = \int \frac{dt}{2.21\sigma R_b(1 - 0.995475 \cos \varphi(t))} \dots \dots \dots 2.70$$

The value of  $TP_{var}$  is obtain by evaluating the above integral over the communication period.

Inconclusion, the proposal yields throughput improvement compared to a constant rate throughput  $TP_{const}$  as

$$\frac{TP_{var}}{TP_{const}} = 4.7 \text{ to } 6.7\text{dB} \dots \dots \dots 2.71$$

### 2.8.2 Optimisation Through Hybrid-ARQ

This research work was published in IEEE communications Transaction of 1997 [8] by a group of research academics; VY Chu, Dr. W, Sun and M.N Sweeting of the Centre of Satellite Engineering Research in University of Surrey, United Kingdom. The research was conducted to upgrade selective repeat SR-ARQ protocol, one of the communication protocol applied at the centre.

The SR-ARQ protocol is a form of AX.25 (Amateur form of X.25 - radio packet transmission adopted to layer 2 of OSI protocols) adapted for advanced LEO small satellite communication. The satellite transmits message in form of packet broadcast to be received by visible Earth stations. Not all packets that make up the message will be received. Each missing packet represent a hole which can be filled through retransmission hole request made by an Earth station. This form of ARQ provided a robust link in the dynamic LEO downlink and near optimum throughput. However, when the retransmission request becomes high, as result of low margin of  $E_b/N_o$ , it slow down the throughput.

The retransmission requests can be reduced by inclusion of a forward error correction code (FEC) to correct the most frequent error pattern and the retransmission (ARQ) correct the less frequent error pattern. A hybrid-ARQ system is thus formed, so that retransmission requests are minimized, thereby harnessing communication time optimally. The link data throughput is improved and the solution is cost effective for low power margin and small Earth station terminals. Earlier some extensive researches were carried-out on hybrid-ARQ schemes.

1) In type-I hybrid, the data packets are encoded for error detection and correction. A receiver attempts to correct an erroneous packet, but if uncorrectable error pattern occurs, the packet is discarded and request for same packet is made. This scheme is suitable for communications systems that experience fairly constant and low noise level. In the event of unpredictable noise levels, the throughput degrades when the noise is high, but when the noise is relatively low, the extra parity bits represent useless overhead.

2) In a type-II hybrid redundancy symbols are transmitted only when required. However, the type-II hybrid has the upper limit of providing only two codes rates.

A new packet combining scheme, was proposed by the group that maintains high throughput in channels that exhibit wide range of noise levels. This is achieved through adaptive packet combining scheme. The underlying principle of packet combining is to form increasingly reliable estimate by combining each erratic packet with its predecessor, until the resultant combined packet can be reliably decoded. The repeats are not necessarily the message, but coded derivatives at different level.

The two combined schemes were proposed in the paper, these are code combining and diversity combining schemes.

In code combining scheme as presented by Kallel (IEEE 1990 Transaction on communications), code  $C_0$  is an  $(n,k)$  block code used for error detection and  $C_1$  is a  $(2,1,m)$  convolutional code for error correction as shown in figure 2.15.

Code  $C_0$  is an  $(n,k)$  block code used for error detection,  $C_1$  is the  $(2,1,m)$  convolution code for error correction. A message is first encoded with block code  $C_0$  into a code word  $I(X)$ , then forming sequences  $P_1(X)$  and



$P_2(X)$  using generator polynomial of  $C_1$ ,  $G_1(X)$  and  $G_2(X)$ . Sequences  $P_1(X)$  and  $P_2(X)$  are sent alternately.

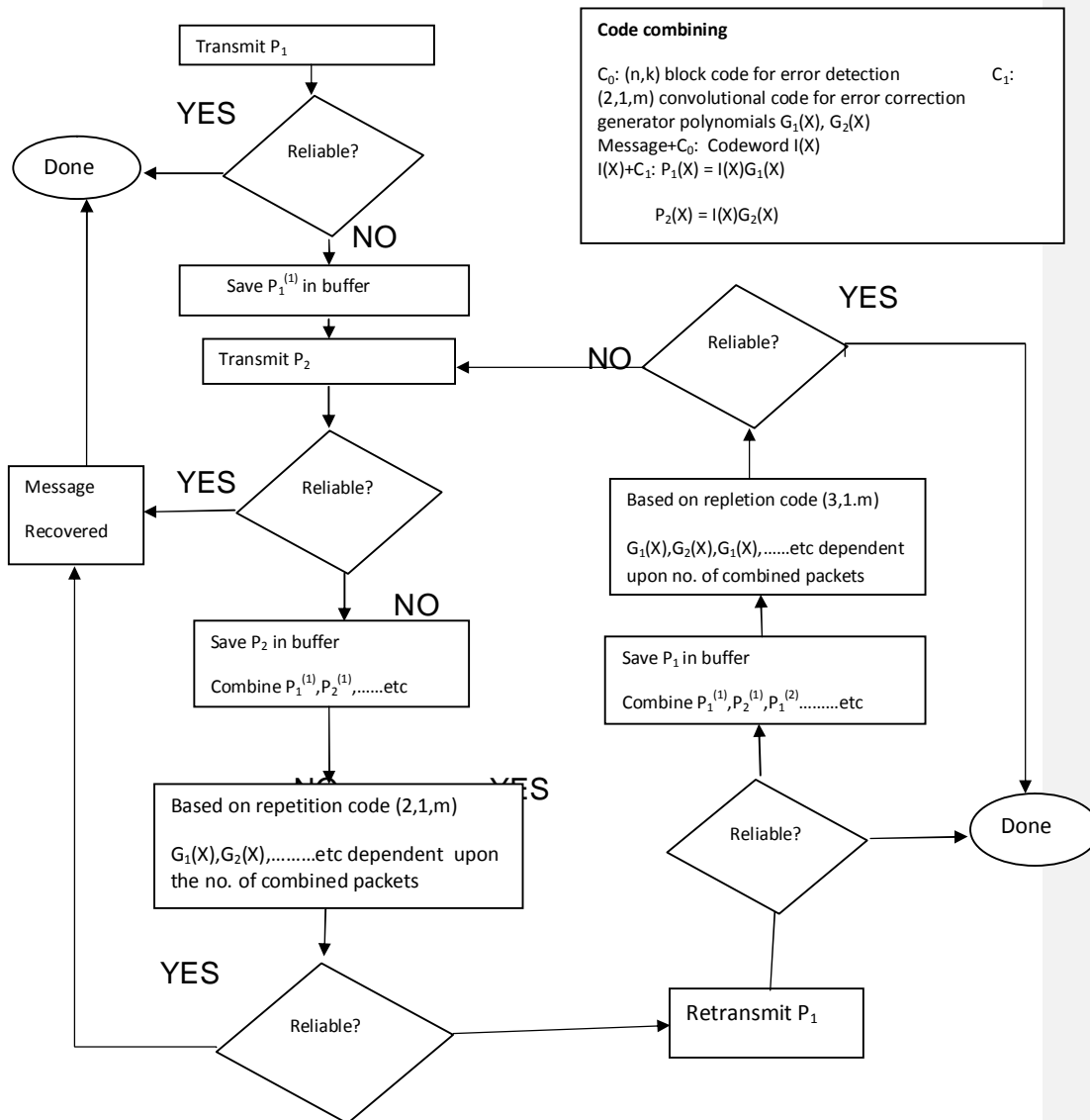


Figure 2.15 Flow chart Diagram of Hybrid-ARQ code combining scheme[8].

The receiver combines all the received sequences with all its repeated copies until the message is decoded error free.

In diversity combining scheme, punctured convolution code is used for both error detection (retransmit request) and correction. The punctured

code attempt to correct an erroneous packet, if the difference in the path (trellis correction path) metrics between a surviving path and best non-surviving path falls below certain threshold the survivor path is declared unreliable. A multiple of unreliable paths (packets) are combined to generate single retransmit request and in response a packet of the same length is retransmitted for correction of the unreliable packets.

Although there is no metric to measure the level of optimisation achieved in their research, its novelty lies in the attempts to address the recurrent overhead bits in FEC which becomes redundant during the time the channel is less noisy. The communication channel is dynamically characterised and adaptive retransmit request (ARQ) applied in a process that guarantees optimum throughput.

### **2.8.3 Adaptive Communication System for Algerian LEO Satellite**

Algeria like Nigeria launched its first remote sensing satellite into LEO in 2003 [9]. In the design proposed for future Algerian LEO satellite, the authors examined some critical issues involved in the communication channel. As most Earth stations are located in cities and the satellite spent a lot of time at low elevation angle, the researchers chooses to optimize communication time by inclusion of low elevation angle, where the channel is mostly affected by multipath fading.

The communication channel is thus, characterized to have maximum free space loss, multipath fading, Doppler effect and white Gaussian noise. The researchers therefore modeled it as Rician/Rayleigh channel.

To counter the dynamic effects of fading, use of Turbo Codes was proposed by the authors. The Turbo codes was developed in 1993 by Claude Berrou, it has excellent gain potentials of improving C/N to 0.7dB from Shannon's theoretical limit. The codes are made up of parallel concatenated convolution codes, it. Its key concept is to use a set of decoders working in a feedback connection where each decoder accepts an estimate of the Log Likelihood Ratio (LLR) of each information bit at its input, and improves the accuracy of this estimate. The LLR is a measure of total information about an information bit. This information comes from two sources, part of it from the received data bit, and is called the intrinsic information. The other part of this

information comes from the decoder and is called extrinsic information. The extrinsic information is passed between decoders whereas the intrinsic is not. The operation iterates a number of times in the feedback loop of the decoders, and finally a hard decision is made on each input bit.

An adaptive M-QAM modulation scheme was proposed for improved spectral efficiency and to counter wide variation in free space loss. The strategy adopted is as follows: The C/N range is divided into N+1 fading and loss regions, and the constellation size  $M_i$  ( $i = 1, 2, \dots, N$ ) is assigned to the  $n$ th region. When the received C/N is estimated to be in the  $n$ th region, a  $M_i$ -QAM constellation size is transmitted, thus adaptive discrete rate (ADR) multilevel modulation scheme is used.

The turbo encoder is made to have adaptive code rates that are also applicable to the N+1 fade regions. Thus, the two schemes form turbo coded adaptive modulator scheme (ADR-MQAM), that is implemented as shown in table 2.5.

Table 2.5 Adaptive Discrete Rate in Fading Regions

Fading Regions N	Code Rate	Spectral Efficiency	Constellation Size
1	1/2	1	4-QAM
2	2/3	2	8-QAM
3	3/4	3	16-QAM
4	4/5	4	32-QAM
5	5/6	5	64-QAM
6	6/7	6	128-QAM
7	7/8	7	256-QAM
8	8/9	8	512-QAM

The average bit error rate in a Rayleigh fading channel was quoted as:

$$P_b = \frac{2\sqrt{M} - 1}{M \log_2 M} \left[ 1 - \frac{3 \bar{\gamma} \log_2 M}{2\sqrt{M} - 1 + 3 \log_2 M} \right] \dots \dots \dots 2.72$$

Where  $\bar{\gamma}$  is the average C/N ratio over period of M constellation. This is applied to the ADR-MQAM techniques.

And BER of a coherent MQAM over an additive white Gaussian Noise (AWGN) channel as given by (L. Hanzo and W.T Webb, 2000):

$$\text{BER} = \frac{4\sqrt{M} - 1}{\sqrt{M} \log_2 M} Q\left(\sqrt{\frac{3\gamma}{M-1}}\right) \dots \dots \dots 2.73$$

Here  $\gamma$  is the carrier to noise ratio.

The authors concluded that by adopting the Rayleigh fading channel model on a LEO satellite link and applying the ADR-MQAM techniques, for average BER approach  $10^{-6}$ , the ADR-MQAM has more than 20dB power gain over a non-adaptive modulation.

## CHAPTER THREE

# TRELLIS CODED MODULATION AND THE CONCEPTUAL MODEL

### 3.0 Introduction

As mentioned in chapter two that transmission impairments introduces error in digital communication, which can be corrected with error control coding. In general this is achieved with addition of overhead bits, which in turn reduces data throughput. To maintain same throughput after coding is applied, the transmission rate has to be increased, which translates to demand for more bandwidth. In a bandwidth limited channel, like the LEO satellite downlink, the alternative is to use a more bandwidth efficient modulation scheme.

When Quadrature Phase Shift Keying (QPSK) is used to transmit 2 information bits per Hz and to maintain this same information data throughput when optimum code rate of  $2/3$  is applied, the requirement for bandwidth will increase by a factor of  $3/2$ . If however to avoid the demand for more bandwidth, an 8-PSK is used to transmit at 3 bps/Hz, so that the original 2 information bits/Hz is maintained, however the Euclidean distance in signal constellation of 8-PSK is closer than in QPSK and if same Bit Error Rate (BER) is to be maintained then more power is required on 8-PSK.

In this chapter it would be shown how conventional error coding such as convolution coding can be used to compensate for the power penalty and its limitation. [5] It would be shown that to generate sufficient gain to compensate for the penalty using conventional coding system requires large size constraint length on convolution encoders or large block size on block encoders.

The alternative is to use a modulation scheme that increases the distance between possible coded sequences. This process is called Trellis Coded Modulation (TCM) developed by Ungerboeck et al in 1982, it combines coding and modulation scheme concurrently. The process is implementable on simple convolution codes using the principle of set partitioning on signal points of multilevel or multiphase digital modulation.

### 3.1 Euclidean Distance in MPSK Scheme

Euclidean distance is the shortest distance between any two points i.e straight line distance unlike Hamming distance which is a digital metric that measures the number of difference between two sequences, Euclidean distance is an analog concept extended to digital signal constellation points [10]. Figure 3.1 (a) and (b) are two constellations of signal points for QPSK and 8-PSK respectively. The signal points are located each on a circle of radius equal to unity representing maximum amplitude of the signal.

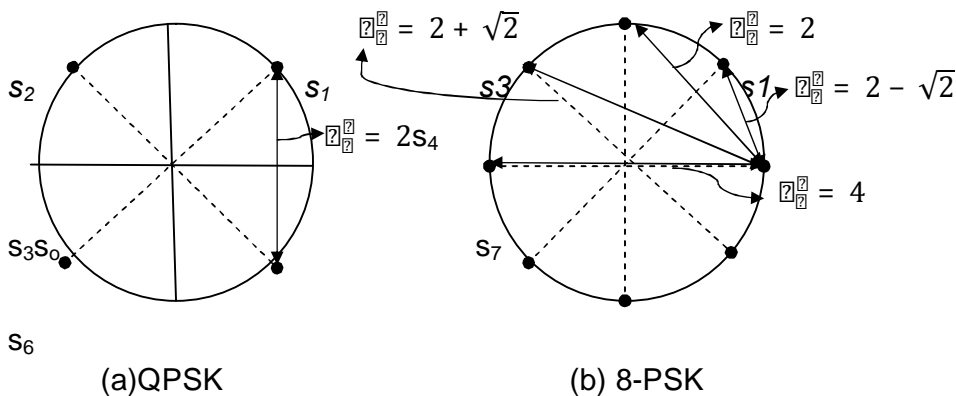


Figure 3.1 Euclidean Distance in QPSK and 8-PSK Schemes.

The position of each point on the constellation is a combination of the amplitude and phase. The distances marked  $d_1, d_2, d_3, \dots$  are the Euclidean distances for signals points  $s_1, s_2, s_3, \dots$  referenced to  $s_0$  respectively, normally expressed in squared form as Squared Euclidean Distance (SED). The smallest of these distances is called the Minimum Squared Euclidean Distance (MSED) represented as  $d_{min}^2$ .

On QPSK constellation:

$$d_{min}^2 = 2 \dots \dots \dots 3.1$$

And on 8-PSK constellation:

$$d_{min}^2 = 2 \sin^2 \frac{\pi}{8} = 2 - \sqrt{2} = d_2^2 \dots \dots \dots 3.2$$

Other SED on the 8-PSK constellation are

$$d_1^2 = 2 \dots \dots \dots 3.3$$

$$d_4^2 = 2 + \sqrt{2} \dots \dots \dots 3.4$$

And

$$\frac{E_b}{N_0} = 4 \dots \dots \dots 3.5$$

It should be noted that both SED and MSED are functions of radius of the circle or the signal amplitude.

The ability of a demodulator to distinguish a signal point from its neighbors is expressed as probability of bit error ( $P_e$ ) of the modulation scheme, which is sometimes expressed as a function of the MSED.[12]

$$\frac{E_b}{N_0} \sim \frac{E_b}{N_0} \dots \dots \dots 3.6$$

Where  $\sigma^2 = N_0/2$ . ( $N_0$  is the noise power).

### 3.2 Convolution Coding Technique

Convolution coding involves logical combination of present and previous inputs to generate a coded output bit, this is demonstrated in a simple convolution encoder of code rate =1/3 (input bit  $k = 1$ , output bit  $n = 3$ ) shown in figure 3.2, it has 3 arrange in stages ( $L = 3$ ), and 3 modulo-2 adders as the 3 output generators. The 3 generators  $g_1, g_2$  and  $g_3$  are connected to the cascaded shift registers according to 3 logic equations, given as:  $g_1=[101]$ ,  $g_2=[011]$  and  $g_3=[111]$ .

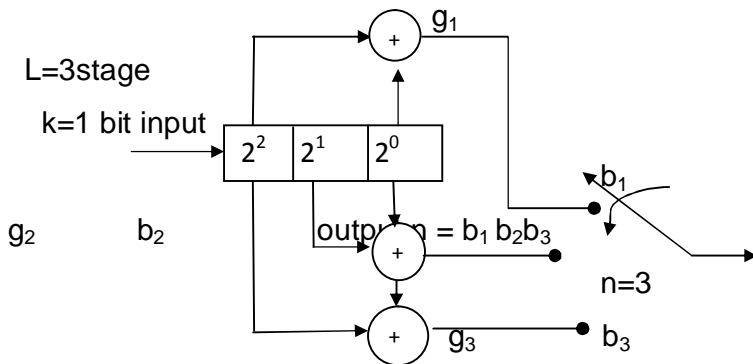
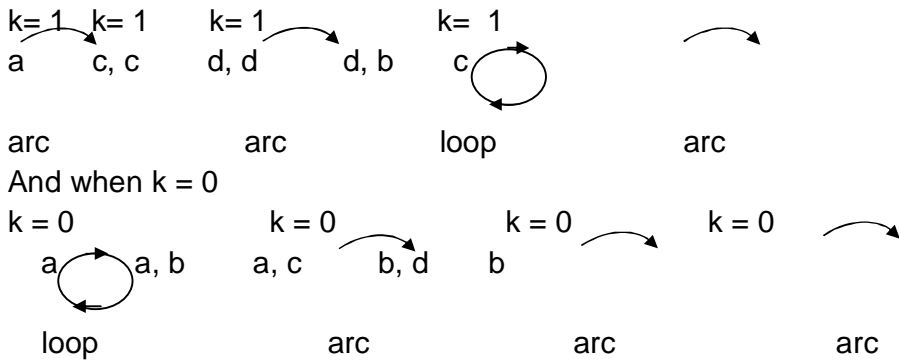


Figure 3.2A Simple 1/3-Rated Convolution Encoder

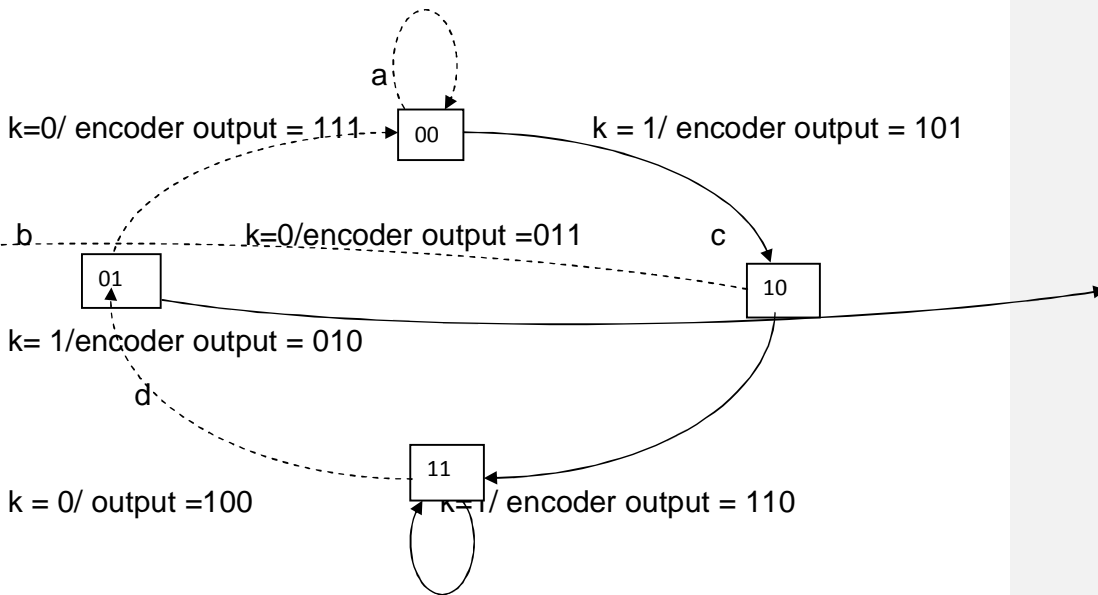
Output of the encoder depends on present input bit in the first register, and two previous inputs shifted and stored in the second and third registers- LSB stored in the third register.

The current states of the first two registers are critical in determining the next state of the three registers, if these states are labeled in boxes as  $a[00], b[01], c[10]$  and  $d[11]$ , then their transition to the next states can be specified by an arc for a new state and loop for same state. The Input  $k$ , the transitions and the possible next states are given below:



Thus, the encoder is a finite state machine in which all the registers possible states and the output can be drawn. The Operation of the encoder is graphically presented in figure 3.3. Transition from one state to another is represented by a long arc connecting the two boxes and a loop signifies transition to same state. The dotted arc or loop represent transition when input bit  $k=0$ /encoder output and solid arc/loop represent transition when input bit  $k=1$ /encoder output.

$k=0$ / encoder output = 000



$k=1$ /encoder output = 001

*Figure 3.3 Transition Diagram for a 1/3-Rate Convolution Encoder*

The operation is summarized in transition table 3.1 and output table 3.2.



Table 3.1 State Transition Table of a 1/3- Rate Convolution Encoder

Current state	Next state if input k=1	Next state if input k=0
00	10	00
10	11	01
11	11	01
01	10	00

Table 3.2 1/3-Rate Convolution Encoder State - Output Table

Current State	Output if input k=1	Output if input k=0
00	101	000
10	110	011
11	001	100
01	010	111

### 3.2.1 Trellis Diagram

A Trellis diagram is another graphical presentation of transitions and output as input are clock in, this is shown in figure 3.4[4]. The diagram specifies all the possible states of the 1/3 rate convolution encoder on the vertical axis and clock timing on the horizontal axis.

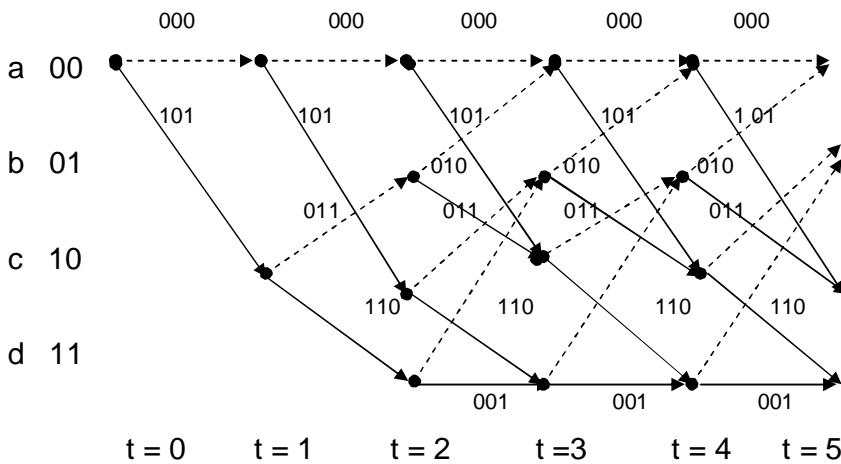


Figure 3.4 Trellis Diagram for a 1/3 Convolution Encoder

Transitions are denoted by straight line (solid line for  $k=1$  and dotted line for  $k=0$  bit) between two states, and outputs are indicated on top of the Transition lines. The time axis is scaled to correspond to clocking time of input  $k$ .

At  $t=0$  if the registers are at state = 00, when input bit  $k=1$ , the transition is to state  $c[10]$  and when  $k=0$ , the transition is remain in state  $a[00]$ .

At  $t=1$  depending on the present state  $a[00]$  or  $c[10]$  (i.e depending the previous input ) and present input, there are four possible transitions . these are:  $[a]$  to  $[a]$  or  $[a]$  to  $[c]$  and  $[c]$  to  $[d]$  or  $[c]$  to  $[b]$ .

At  $t=2$  depending on the present state (the registers have transited to any of the four possible states), consequently there are eight possibilities and the process goes for  $t = 3, 4, \dots, \infty$ .

If the above Encoder is use to code the information sequence 101001. Using the above procedure for each clock time, the encoding process is as shown in figure 3.5.

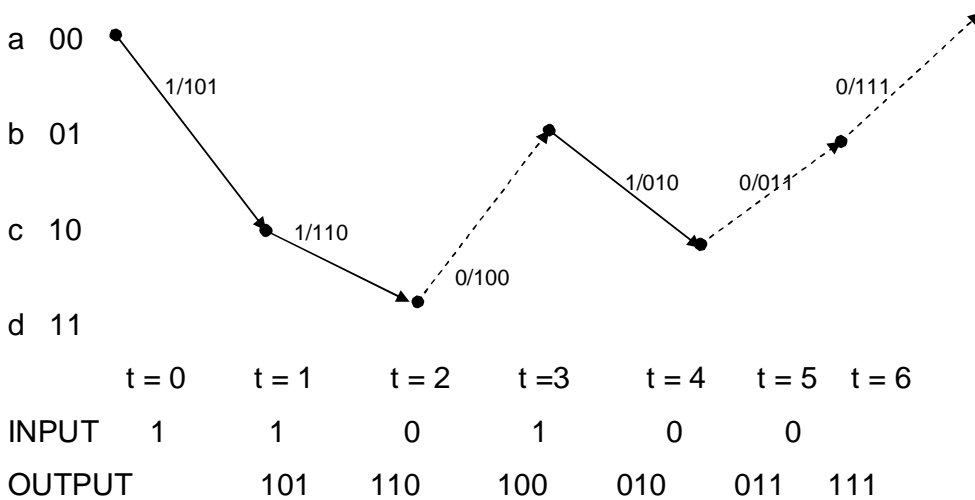


Figure 3.5 Coding Process of 1/3-rated Convolution

The encoder output produces three bits for each clock input bit, the coded output = 101110100010011111.

In general, a  $k/n$ -rate convolution encoder, can be generalized as shown in figure 3.6,  $k$  input bits are shifted into a block of  $k$  registers at each

clock time, L blocks of k registers are connected, so that an input is stored and relevant to the output for L clock times (L is the encoder constrain length). Series of k bits in L blocks are combined or convoluted using n number of exclusive-OR gate. Connection EX-OR gates to the registers are defined by sequence generator for each gate. The output n bits depend not only on the most recent input block, but also on contents stored in (L-1)k registers. Therefore, number of transition states in this encoder is  $2^{(L-1)k}$ .

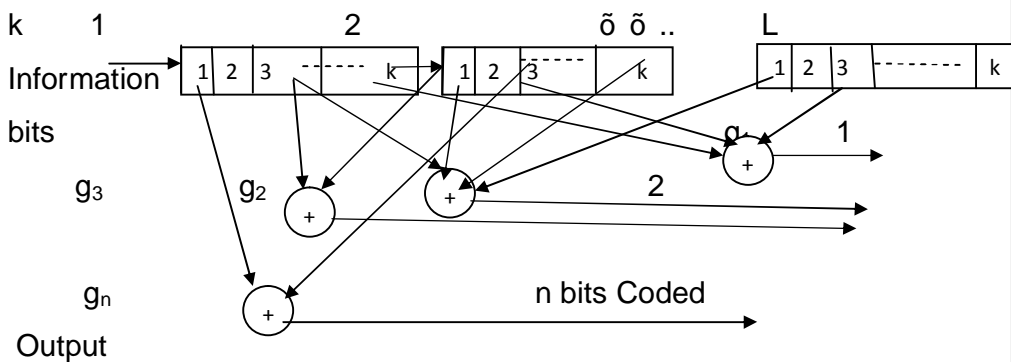


Figure 3.6 Convolution Encoder of L Blocks of k-stage Shift Registers.

The connections between exclusive-OR gates to Lk registers are not specific, but defined by convoluted logics of the generators  $[g_1, g_2, \dots, g_n]$  [5].

### 3.2.2 Coding Gain in Convolution Coding

Coding gain is a measure of BER improvement or reduction of C/N due to application of coding. It is the ratio of C/N of coded to that of uncoded link transmitting same information bit at the same BER. This expressed as:

$$\frac{E_b/N_0 \text{ (coded)}}{E_b/N_0 \text{ (uncoded)}} = \dots \dots \dots 3.7$$

$$= \frac{E_s/N_0 \text{ (coded)}}{E_s/N_0 \text{ (uncoded)}} \dots \dots \dots 3.8$$

Where  $E_s$  and  $E_b$  are energy per symbol for coded and uncoded system respectively. Coding gain is also deduced in some texts as [5]

$$\dots \dots \dots = \frac{E_s \text{ (coded)}}{E_s \text{ (uncoded)}} \dots \dots \dots 3.9$$

Where  $d_{min}^2$  is the squared minimum Euclidean distance in an uncoded constellation and the term  $d_{free}^2$  is called the free squared Euclidean distance, which is the smallest distance between allowed coded sequences this is taken as the distance between an all zero sequence and the one that diverted, and then re-merged with it (as in figure 3.5). This is based on the principles of viterbi decoding algorithm of convolution codes, in which erroneous diversion of path due to received error is corrected (i.e. remerge) after few subsequent transitions in the decoding process.

To illustrate the above principle, consider a 2/3-rate convolution encoder (with 3-bits output applied to an 8-PSK scheme) shown in figure 3.7.

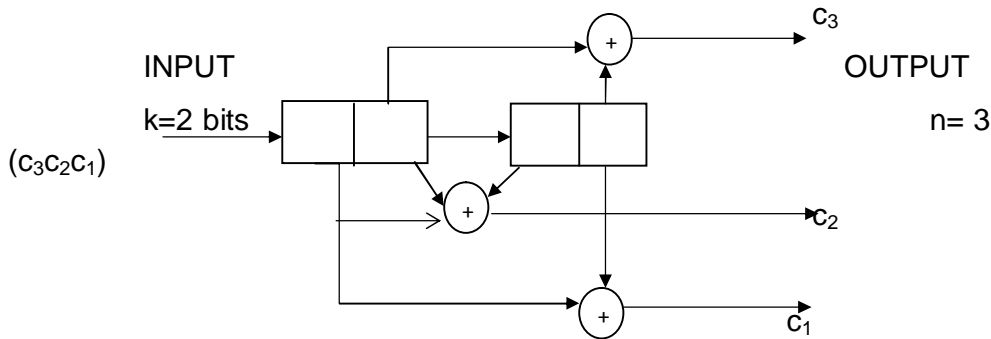
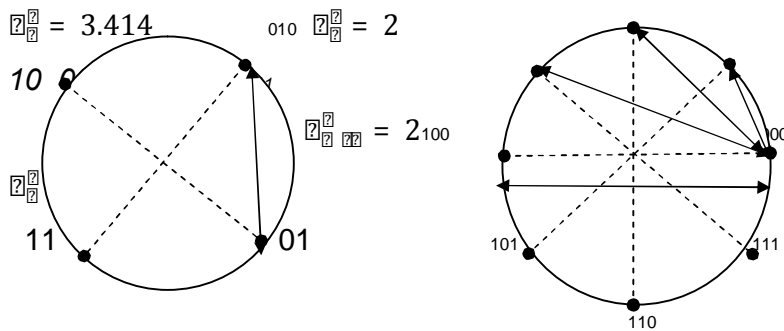


Figure 3.7 2/3 - Rate Convolution Encoder

The uncoded 2-bit inputs is mapped into a QPSK constellation of radius=1, while the coded 3-bit outputs is also mapped into an 8-PSK constellation of same radius. The squared minimum Euclidean distances of the uncoded input (on the QPSK) marked in figure 3.8(a) as  $d_{min}^2 = 2$  (equation 3.1)



(a) 2 bits input uncoded QPSK (b) 3 bits output coded 8-PSK  
Figure 3.8 Mapping 2-bits and 3-bits Sequences

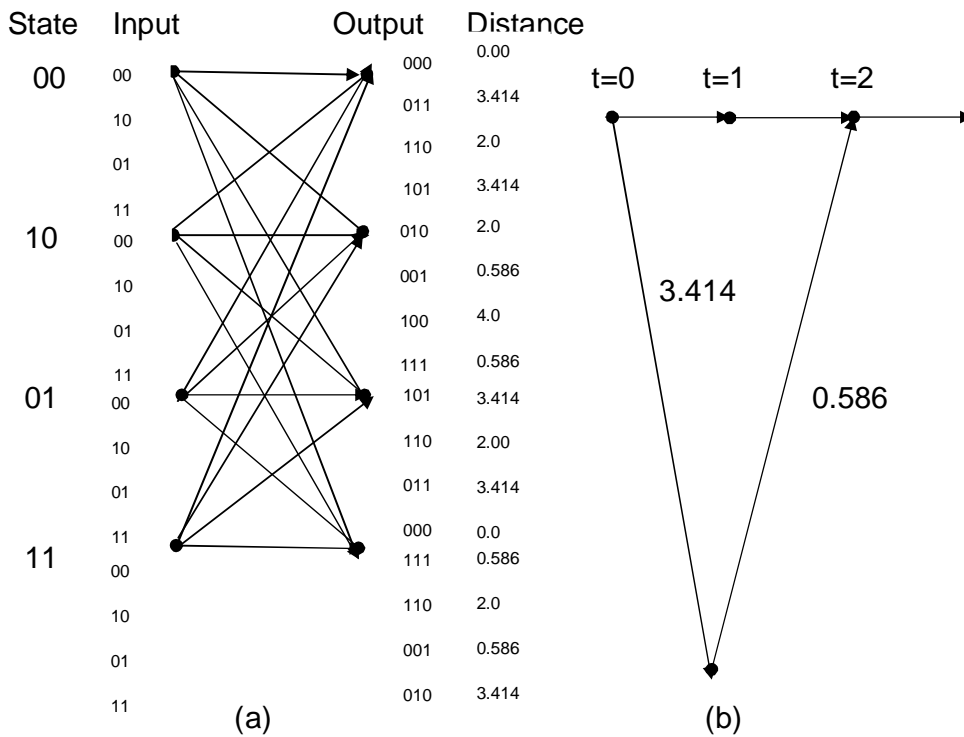


Figure 3.9 Trellis of a 2/3 Convolution Code and Coding Gain

Trellis diagram of this coding is given in figure 3.9(a). The squared Euclidean distance (on the 8-PSK constellation) from an all zero path (output  $s_0$  000) to all possible transition paths are recorded adjacent to each output.

To determine  $d_{2,2}^2$  of the code, a path with smallest total squared distance, which diverts from an all zero (000) and remerges after successive transitions is selected. As shown in figure 3.10(b), at state 00, the path that transitioned to state 11 has squared distance of 3.414 and returns to state 00 with squared distance of 0.586 has the smallest total squared distance of  $3.414 + 0.586 = 4.0$ , all other possible transitions from state 00 and back to 00 have total squared distance greater than 4. In this encoder therefore

$$d_{2,2}^2 = 3.414 + 0.586 = 4.0 \dots \dots \dots 3.10$$

Then, from equation 3.9 coding gain in dB of 2/3 rate convolution encoder can be written as:

$$\begin{aligned} \text{Coding gain (dB)} &= 10 \log \frac{2^2}{2^2} \\ &= 10 \log \frac{2^2}{2^2} \end{aligned}$$

From equations 3.1 and 3.10

$$\frac{4.0}{2.0} = 10 \log \frac{4.0}{2.0} = 3.0 \text{ dB}$$

This is happened to be the maximum theoretical coding gain of convolution coding on 8-PSK scheme.[5]

### 3.3 Set Partitioning Theory

This is a process of partitioning a signal points constellation into smaller congruent subsets, such that the minimum Euclidean distance ( $d_{\min}$ ) is increased with each partitioning.

An 8-PSK signal constellation of unit radius shown in figure 3.10 is to be partitioned. At first level the eight signal points are divided into two equal subsets  $A_0$  and  $A_1$  so that the minimum distance is increased

$$\text{from } \frac{\sqrt{2}}{2} \text{ to } \sqrt{2}$$

In the second partitioning each of the two subsets is further divided into two equal subsets each containing two signal points, so that there are four subsets  $B_0, B_1, B_2$  and  $B_3$ . This has increased the minimum distance to  $\frac{\sqrt{2}}{2} = 2$ .

In third partitioning which in this case is the last, the signal points are divided into eight points  $C_0, C_1, C_2, C_3, C_4, C_5, C_6$  and  $C_7$  where each subset contains only one signal point.

The Trellis Coded Modulation make use of the increase in minimum Euclidean distance with each partitioning, to reduce the probability of error (thus providing apparent power gain). The coded sequences are mapped into each partitioning point to form TCM constellation according to the rules spelled out by Ungerboeck et al to be presented in section 3.5[4]

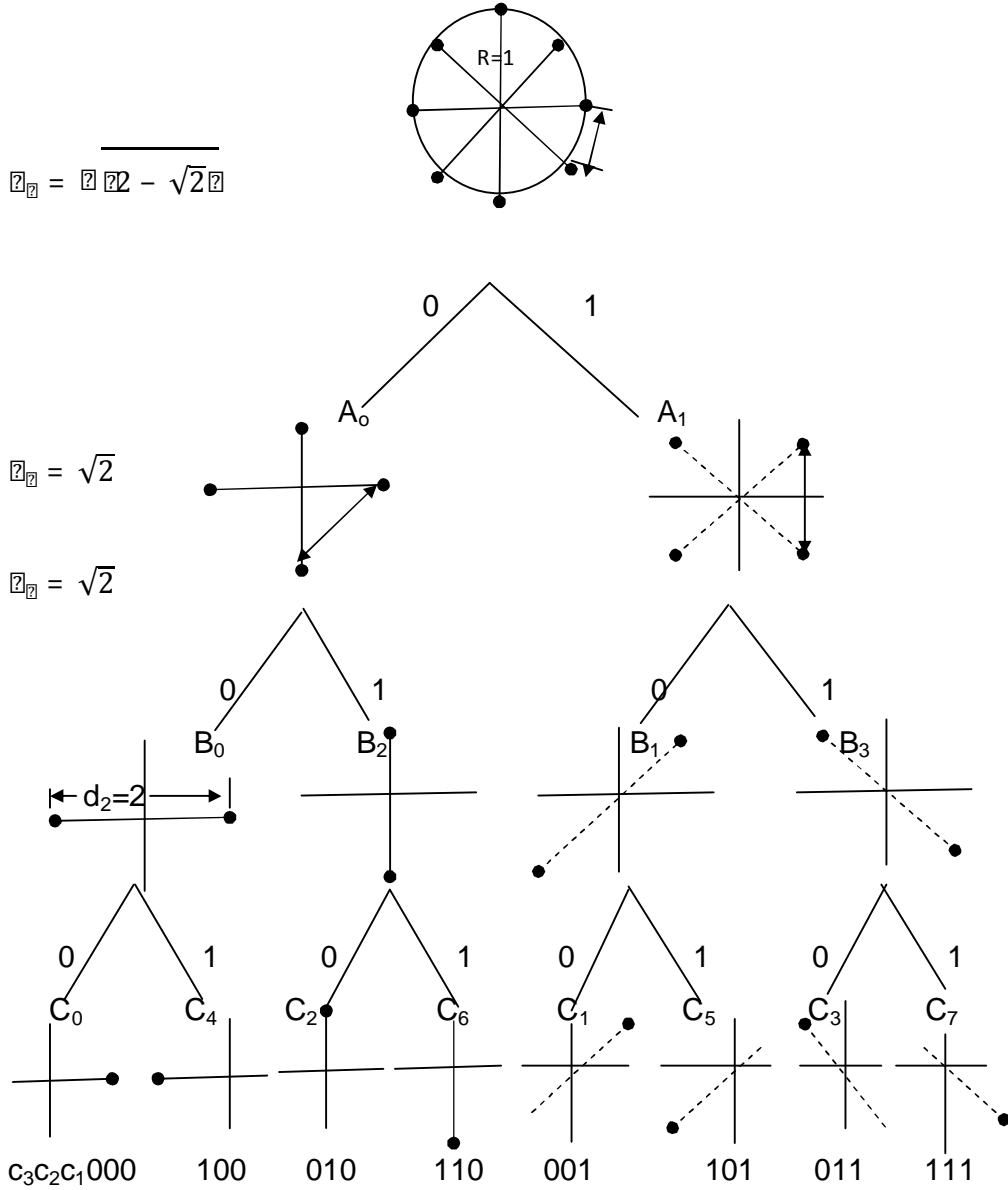


Figure 3.10 Set Partitioning of an 8-PSK signal points

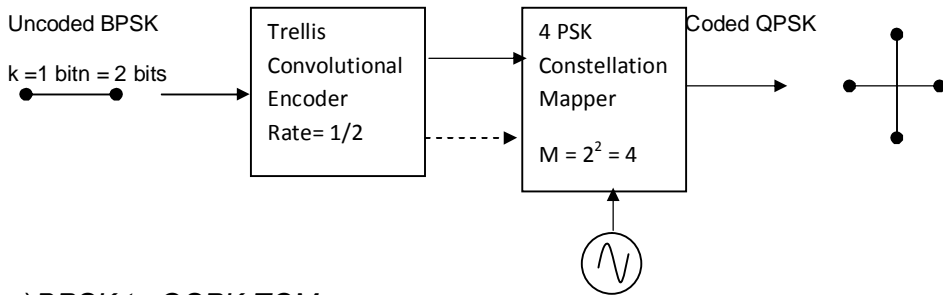
### 3.4 Trellis Coded Modulation Scheme

In Trellis Coded modulation (TCM) the two functions Trellis (convolution) coding and Constellation mapping (for multilevel or multiphase modulation) are performed concurrently[5]. During symbol transition, the TCM make use of transition constrain in convolution coding to map (set partition) the allowable output into constellation points with maximum Euclidean separation. In MPSK mapping this maximum Euclidean separation is  $180^\circ$  as shown in figure 3.10.

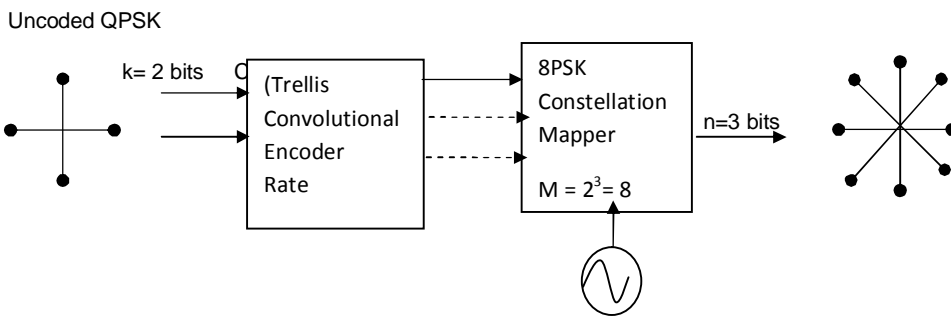
The block diagram implementation of TCM is shown in figure 3.11, in each case, the size of output constellation is twice the input size. From uncoded to coded constellation, TCM is therefore viewed as system for constellation upgrade [10]. The TCM produces coded QPSK from uncoded BPSK, coded 8-PSK from uncoded QPSK, and coded 16-PSK from uncoded 8-PSK etc. Convolution coding is more readily applicable to TCM than Block coding because of the availability for optimum soft decision decoding algorithm (Viterbi) of the former.[4]

As the constellation expands the symbol size increases, this implies increased throughput. Although, Euclidean separation between the symbols becomes smaller, but selective separation of the symbols enhances the performance.

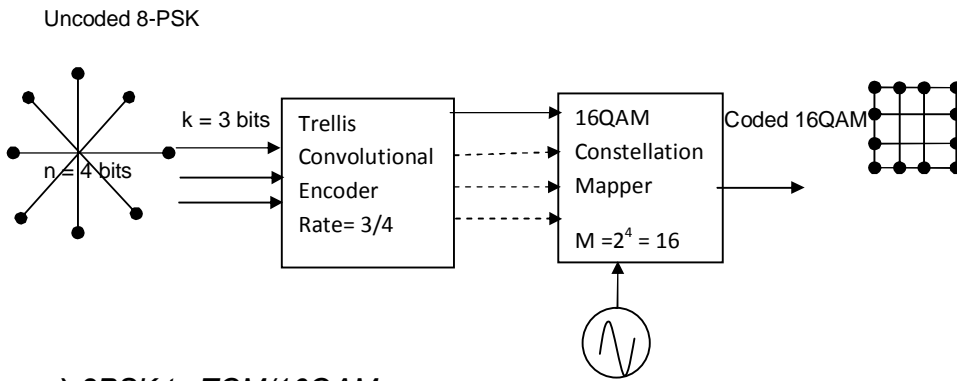




a) BPSK to QSPK TCM



(b) QPSK to TCM /PSK



c) 8PSK to TCM/16QAM

Figure 3.11 Examples of Trellis Coded Modulation [10]

A general block diagram for TCM implementation is shown in figure 3.12[4].

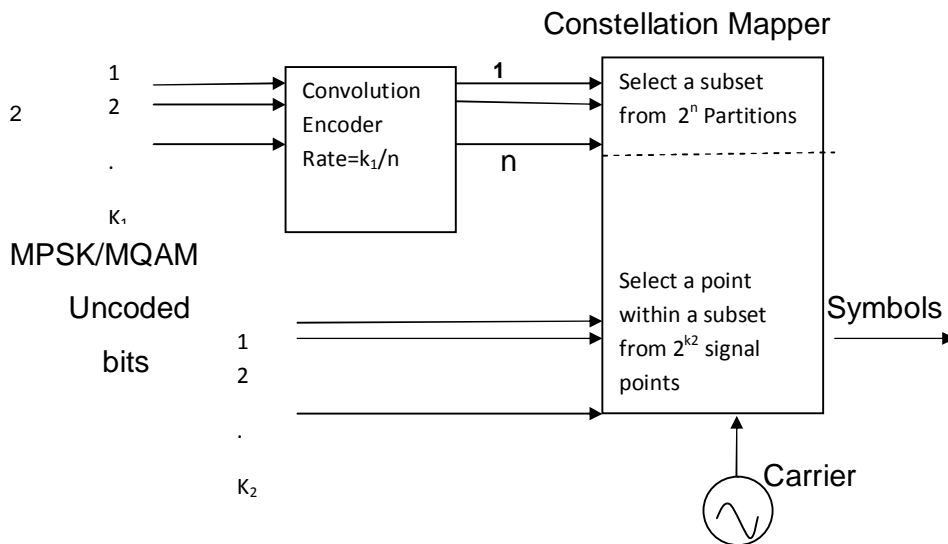


Figure 3.12 General Diagram of Trellis Coded Modulation Scheme

An input block symbol  $k$  length is broken into two sub-blocks  $k_1$  and  $k_2$ , such that  $k = k_1 + k_2$ . The sub-block  $k_1$  is encoded into  $n$  bits by a convolution encoder and the  $k_2$  sub-block is left uncoded. The encoded  $n$  bits are used to select a subset from  $2^n$  set partitions by the constellation mapper, while the uncoded  $k_2$  bits are used to select a signal point among  $2^{k_2}$  signal points in each of the  $2^n$  set partitions. By choosing  $n = k_1 + 1$  and  $k_2 = 1$ . Ungerboeck et al, has shown that with a simple encoder and a mapper that doubles the constellation points ( $M = 2^{k+1}$ ), coding gain between 3 and 6dB can be achieved.

With output constellation of  $M = 2^{k+1} = 2^n$  it implies that code rate of  $k/(k+1)$  is applied, this is optimum since only one overhead coding bit is added for all of  $k > 0$ .

TCM is therefore considered to be applicable in throughput optimization to links that are both bandwidth and power limited, like the LEO downlink.

### 3.4.1 Implementation of TCM/8-PSK

Consider a simple encoder diagram on figure 3.13(a) in which the input  $k$  is divided into two such that  $k = k_1 + k_2 = 2$ , so that  $k_1 = 1$ ,  $k_2 = 1$ . The input  $k_1$  is fed into a simple  $1/2$ -rate convolution encoder whose outputs are  $c_1 c_2$ . While the other input  $k_2$  is left uncoded as  $c_3$ .

Transition diagram for the convolution encoder is shown in figure 3.13(b), from each current state on the registers  $S_1S_0$  there two possible transitions and two possible outputs depending on the value of input  $k_1=0$  or 1. The operation is described as follows:

(i) At current state of  $S_1S_0=00$ , when  $k_1=0$  then output  $c_2c_1=00$  or  $B_0$  and next state of the registers remains at 00; but when  $k_1=1$  the output  $c_2c_1=10$  or  $B_2$  and next state of the encoder is 10.

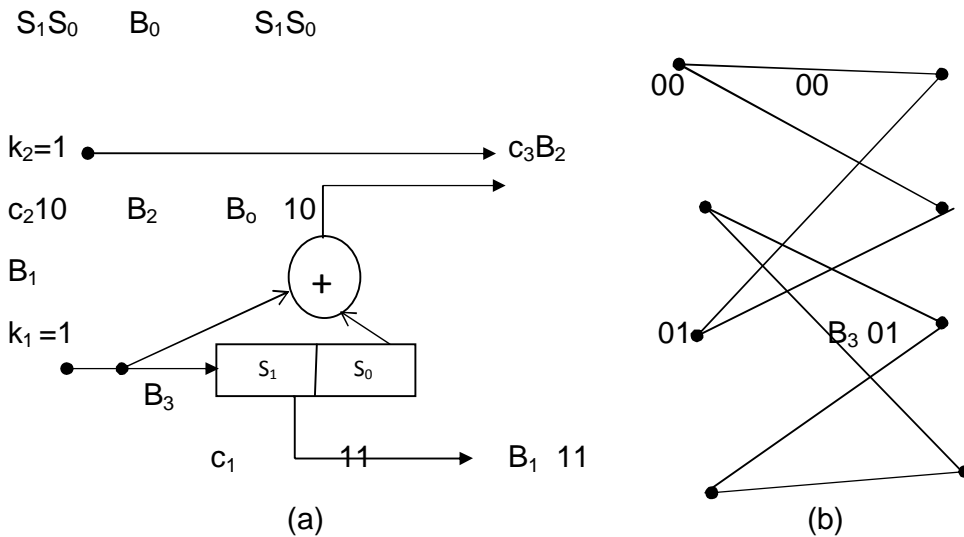


Figure 3.13A Simple TCM Scheme and State Transition

(ii) At current state  $S_1S_0=10$ , when  $k_1=0$  then the output  $c_2c_1=01$  or  $B_1$  and next state of the registers  $S_1S_0=01$ ; but if  $k_1=1$  then output  $c_2c_1=11$  or  $B_3$  and next state of the registers  $S_1S_0=11$ .

(iii) At current state  $S_1S_0=01$ , when  $k_1=0$  then the output  $c_2c_1=10$  or  $B_2$  and next state of the registers  $S_1S_0=00$ ; but if  $k_1=1$  then output  $c_2c_1=00$  or  $B_0$  and the next state of the registers  $S_1S_0=10$ .

(iv) At current state  $S_1S_0=11$ , when  $k_1=0$  then the output  $c_2c_1=11$  or  $B_3$  and next state of the registers  $S_1S_0=10$ ; but if  $k_1=1$  then output  $c_2c_1=01$  or  $B_1$  and next state of the registers remains at  $S_1S_0=11$ .

Table 3.3 State Table for a Simple TCM

output	$C_2C_1$	$C_3C_2C_1$	
		$c_3=0$	$c_3=1$
$B_0$	00	000	100
$B_1$	01	001	101
$B_2$	10	010	110
$B_3$	11	011	111

A TCM is formed when the outputs are combined in a sequence  $c_3c_2c_1$  such that  $c_3$  is the most significant bit (MSB). In which case, each subset  $c_2c_1$  produces two possible TCM outputs depending on the value of  $c_3$  as shown in table 3.3.

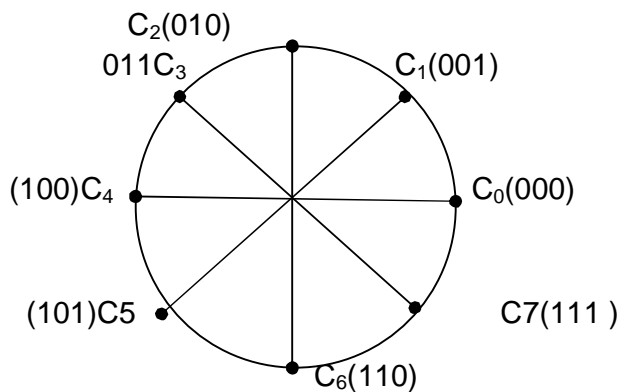


Figure 3.14 Mapping of TCM/8-PSK Output ( $c_3c_2c_1$ ) in a Constellation

If these two outputs are mapped into an 8-PSK constellation separated at  $180^\circ$  as in figure 3.14, then possibility of error is reduced. The advantage of TCM in reducing error in this case is that, the Euclidean distance between within a subset  $c_2c_1$  (output of convolution encoder) has been increased from  $\sqrt{2}$  in figure 3.9(a) to  $2$  in figure 3.9(b), under same average power.

The operation of the simple TCM, as a whole is shown in state transition diagram of figure 3.15. On the left hand side are the present states of the two registers  $S_1S_0$  before the arrival of the present input  $k_1k_2$ . On the right hand side are the outputs for each possible transition. There two possible transitions to same state from one state, which are represented as two parallel lines (solid and broken). The most significant bit (MSB

$c_3=k_2$ ) representing the uncoded bit is used to select one of the two parallel transition.

In the trellis diagram shown, transitions from a state are restricted to two next states only, it is this restriction that the TCM uses for mapping the output  $c_3c_2c_1$  as in figure 3.14.

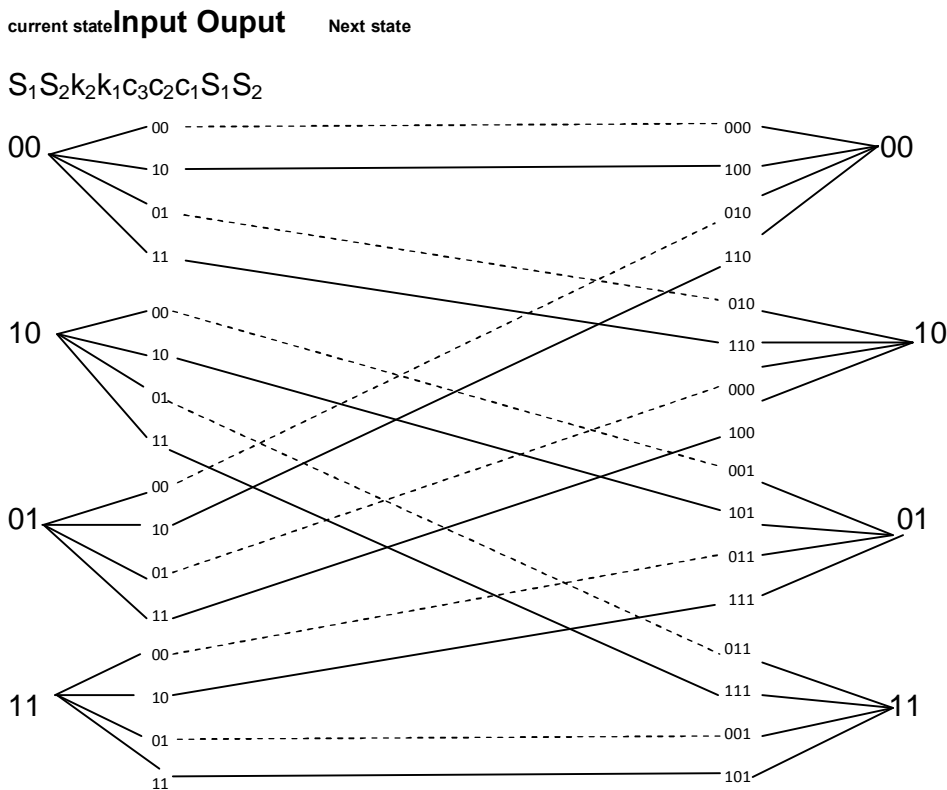


Figure 3.15 Trellis Diagram for a Simple TCM Scheme

The signal points in set partition are optimally assigned to trellis transition path according to three heuristic rules stated by Ungerboeck et al, these are: [4]

1. Use all subsets with equal frequency.
2. Transition originating from the same state or merging into same state are assigned subsets that have a single parent from the preceding stage in set partition. In the above case  $\{B_0, B_2\}$  and

$\{B_1, B_3\}$  are assigned to quadruples parent as  $A_0$  and  $A_1$  respectively, where the minimum Euclidean distance  $d_{\min}^2 = \sqrt{2}$ .

3. Parallel transitions, when they occur (characteristic of 4 state transition), are assigned signal points separated by maximum Euclidean distance  $d_{\min}^2 = 2$ . In the above case  $B_0 = \{C_0, C_4\}$ ,  $B_2 = \{C_2, C_6\}$ ,  $B_1 = \{C_1, C_5\}$  and  $B_3 = \{C_3, C_7\}$  corresponds to parallel transitions.

### 3.4.2 Coding Gain in TCM

Coding gain in TCM is determined as in section 3.5 using equation 3.5. For the 2/3- rated TCM, the smallest free squared Euclidean distance  $d_{\text{free}}^2$  obtained using figure 3.16. Paths that diverted from path 000 and then remerged are:

- (a) Transition path with output=100 parallel to path with output= 000 diverted at  $t=0$  and merged at  $t=1$ , Squared Euclidean distance between 100 and 000  $d_{\min}^2 = 4$ .
- (b) Transition paths with output=010 diverted from path with output = 000 at  $t=0$  to state= 10 at  $t=1$ , at squared distance of  $d_{\min}^2 = 2$ ; then from state=10 at  $t = 1$  it transitioned to state 01 at  $t=2$  with output = 001 which is at squared distance of  $d_{\min}^2 = 0.586$  from output =000; it then transitioned from state 10 to state 01 at  $t=2$  with output= 010; which finally transitioned from state 01 to state 00 at  $t = 3$  with output = 001, at squared distance of  $d_{\min}^2 = 2$  where merged with output = 000. Therefore total squared distance is  $d_{\min}^2 + 2d_{\min}^2 = 0.586 + 4 = 4.586$ .

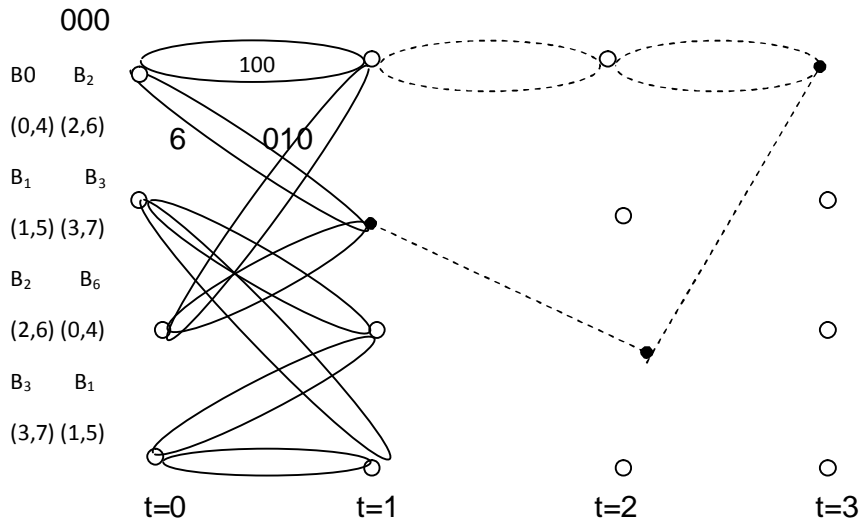


Figure 3.16 Minimum Distance Transition Path in a TCM Scheme

In this case, the parallel transition path with output =100 has the smallest total squared Euclidean distance of  $d_{T}^2 = 4$ ;

$$\Rightarrow d_{T}^2 = 4 \cdot \underbrace{\dots}_{\dots} = 4$$

Therefore by using equation 3.9, coding gain on 3 bits coded TCM/8-PSK relative to 2 bits uncoded QPSK where  $d_{T}^2 = 2$  is given by:

$$\begin{aligned}
 \text{Coding Gain} &= 10 \log \frac{d_{T}^2}{d_{T}^2} \\
 &= 10 \log \frac{4}{2} = 3.01 \text{ dB}
 \end{aligned}$$

This 3 dB gain can be seen to be limited by the distance of the parallel transition, hence if the parallel transition is eliminated by increasing the number of transition state beyond 4 states, higher gain could be realized.

### 3.4.3 High Gain TCM/8-PSK Schemes

#### (i) 8-State Transitions

When systematic feedback is applied to a  $2/3$  rate TCM, the number transition states increases to more than four states, thereby eliminating the parallel transitions [4]. A systematic feedback 8-PSK/TCM is shown in figure 3.17, with three shift registers ( $S_2, S_1$  and  $S_0$ ) placed along the feedback path.

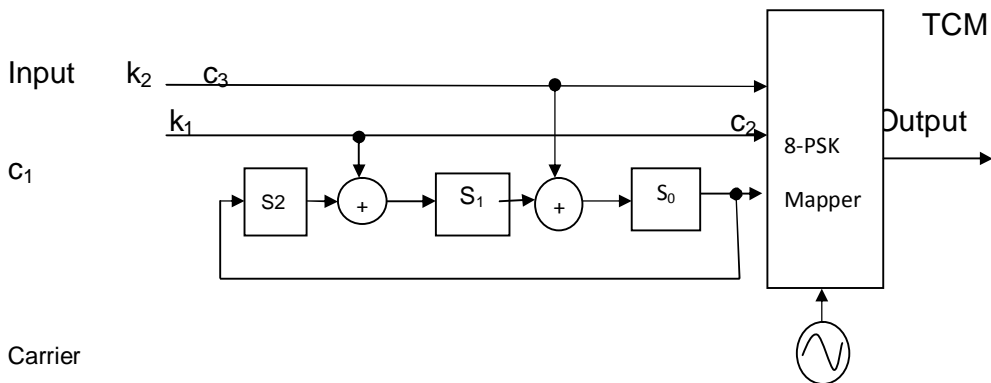


Figure 3.17 Systematic Feedback of an 8-State on 8-PSK TCM

The input  $k_1, k_2$  have unrestricted possible combination, but the bits output  $c_3, c_2, c_1$  are restricted by  $c_1$ , which depends on the state ( $S_2, S_1, S_0$ ) of the three registers. The next state or transition of the registers is turn restricted by modulo-2 combination of the input  $k_1, k_2$  and individual previous state of the registers connected. In this case the state of each register determines the transition and the output.

For each current state of registers  $S_2, S_1, S_0$ , there is an input-output . next state table one output only (thus eliminating the parallels). Since there are 8 possible state of  $S_2, S_1, S_0$  these are shown in table 3.4(a-h).



Table 3.4 Transition Tables for 8-State Feedback TCM/8-PSK  
 (a) Current State  $S_2S_1S_0=000$  (b) Current State  $S_2S_1S_0=001$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_2S_1S_0$
00	000	000
10	010	010
01	100	001
11	110	011

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_2S_1S_0$
00	001	100
10	011	110
01	101	101
11	111	111

c) Current State  $S_2S_1S_0=010$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_2S_1S_0$
00	000	001
10	010	011
01	100	000
11	110	010

(d) Current State  $S_2S_1S_0=011$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_2S_1S_0$
00	001	101
10	011	111
01	101	100
11	111	110

(e) Current State  $S_2S_1S_0=100$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_2S_1S_0$
00	000	010
10	010	000
01	100	011
11	110	001

(f) Current State  $S_2S_1S_0=101$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_2S_1S_0$
00	001	110
10	011	100
01	101	111
11	111	101

(g) Current State  $S_3S_2S_1=110$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_2S_1S_0$
00	000	011
10	010	001
01	100	010
11	110	000

(h) Current State  $S_2S_1S_0=111$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_2S_1S_0$
00	001	111
10	011	101
01	101	110
11	111	100

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Left, Relative to: Margin, Vertical: 0.1", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Left, Relative to: Margin, Vertical: 0.1", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Left, Relative to: Margin, Vertical: 0.1", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Left, Relative to: Margin, Vertical: 0.1", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Left, Relative to: Margin, Vertical: 0.1", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.35", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.35", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.35", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.35", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.35", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

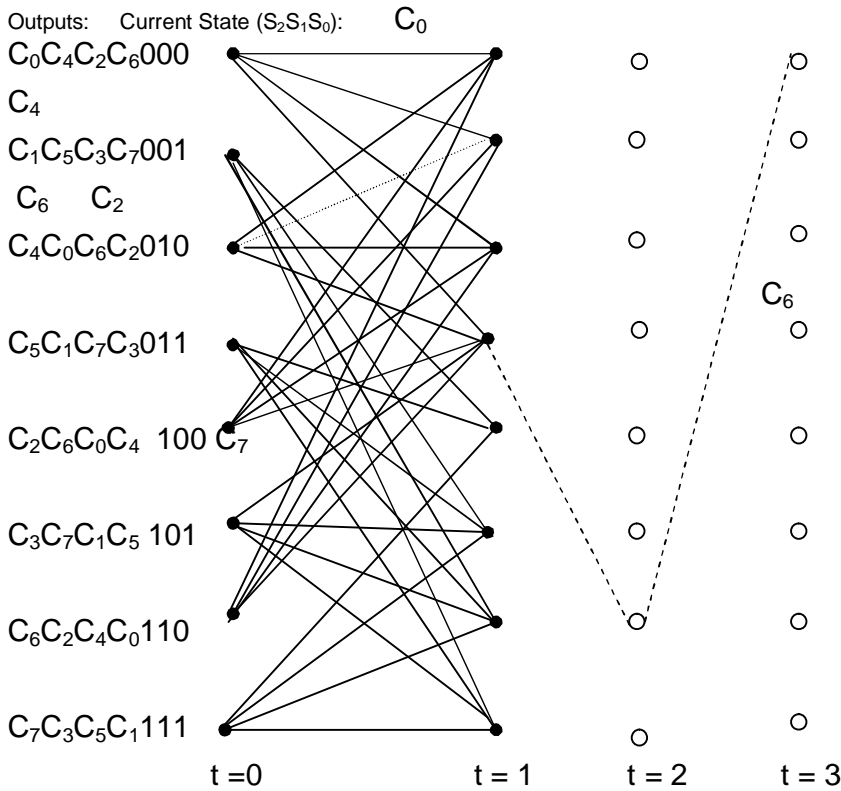


Figure 3.18 Trellis Diagram of an 8-State Transitions TCM/8-PSK

When the second heuristic rule is applied to these, the four outputs originating from same state are grouped into a parent subset (quadruple) as shown in constellation diagram of figure 3.19 and listed at LHS from  $t = 0$  of figure 3.18

To determine coding gain, the constellation diagram of figure 3.19 is used. The minimum square Euclidean distance in the quadruple subset is  $d_{\min}^2 = 2$ . To determine the path with smallest total free Euclidean distance  $d_{\text{free}}^2$ , all the paths that started from 000 state and subsequently remerged into same 000 state were considered, the path that gives the smallest total Euclidean distance is calculated (referenced to  $C_0$ ) as follows:

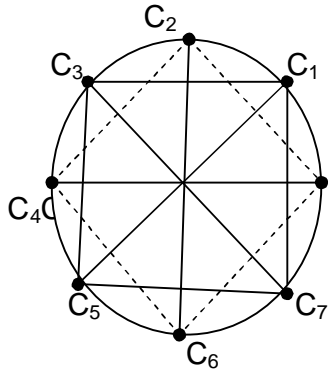


Figure 3.19 Mapping for an 8-State TCM/8PSK in Constellation

Transition from state 000 to 011 resulting to output  $C_6$  at squared distance of  $d_{00}^2 = 2$ ; transition from state 011 to 110 resulting to output  $C_7$  at squared distance of  $d_{01}^2 = 0.586$  and transition from state 110 to 000 resulting to output  $C_6$  at squared distance of  $d_{10}^2 = 2$ . Thus, smallest total free Euclidean distance squared  $d_{free}^2$  for this path is:

$$d_{free}^2 = d_{00}^2 + d_{01}^2 + d_{10}^2 + \dots \dots \dots 3.12$$

$$= 2 + 0.586 + 2 = 4.586$$

Using equation 3.9

$$\therefore \text{Gain} = 10 \log \frac{4.586}{2} = 3.6 \text{ dB}$$

This is an improvement over the 3dB gain obtained from the 4-State TCM/8-PSK, higher gain can be realized if more transitions are allowed.

**(ii) A 16-state 8-PSK TCM**

A 16-State is formed by increasing the number of shift registers along the feedback path as shown in figure 3.20.

8-PSK

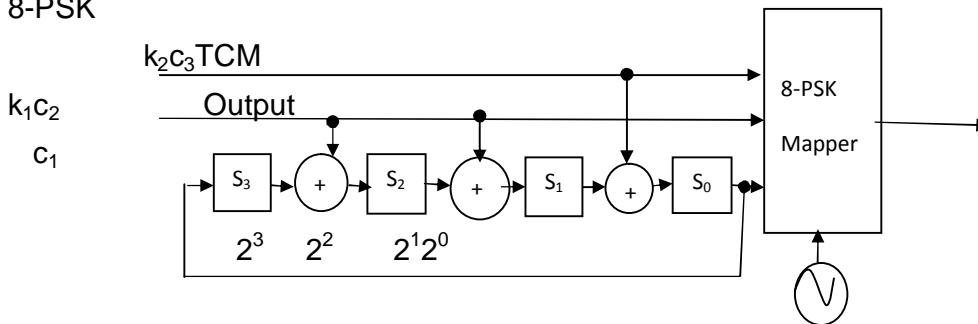


Figure 3.20 16-State Systematic Feedback for TCM/8-PSK

For each state of registers  $S_3S_2S_1S_0$  there is an input- output table as shown in table 3.5.

Table 3.5 Transition State table for a 16-State Feedback TCM/8-PSK

a) Current State  $S_3S_2S_1S_0=000$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_3S_2S_1S_0$
00	000	0000
10	010	0001
01	100	0010
11	110	0111

b) Current State  $S_3S_2S_1S_0=0001$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_3S_2S_1S_0$
00	001	1000
10	011	1010
01	101	1001
11	111	1011

c) Current State  $S_3S_2S_1S_0=0010$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_3S_2S_1S_0$
00	000	0001
10	010	0111
01	100	0000
11	110	0110

d) Current State  $S_3S_2S_1S_0=0011$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_3S_2S_1S_0$
00	001	1001
10	011	1111
01	101	1000
11	111	1110

e) Current State  $S_3S_2S_1S_0=0100$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_3S_2S_1S_0$
00	000	0000
10	010	0100
01	100	0011
11	110	0101

f) Current State  $S_3S_2S_1S_0=0101$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_3S_2S_1S_0$
00	001	1010
10	011	1010
01	101	1101
11	111	1101

g) Current State  $S_3S_2S_1S_0=0110$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_3S_2S_1S_0$
00	000	0011
10	010	0101
01	100	0010
11	110	0100

h) Current State  $S_3S_2S_1S_0=0111$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_3S_2S_1S_0$
00	001	1011
10	010	1101
01	100	1010
11	110	1100

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.33", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.33", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.33", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.33", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.33", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.3", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.3", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.3", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.3", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.3", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.34", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.34", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.34", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted** ... [1]

**Formatted** ... [2]

**Formatted** ... [3]

**Formatted** ... [4]

**Formatted** ... [5]

**Formatted** ... [6]

**Formatted** ... [7]

i) Current State  $S_3S_2S_1S_0=1000$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_3S_2S_1S_0$
00	000	0100
10	010	0010
01	100	0101
11	110	0011

k) Current State  $S_3S_2S_1S_0= 1010$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_3S_2S_1S_0$
00	000	0101
10	010	0011
01	100	0100
11	110	0010

m) Current State  $S_3S_2S_1S_0=1100$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_4S_3S_2S_1$
00	000	0110
10	010	0000
01	100	0111
11	110	0001

o) Current State  $S_3S_2S_1S_0= 1110$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_3S_2S_1S_0$
00	000	0111
10	010	0001
01	100	0110
11	110	0000

j) Current State  $S_3S_2S_1S_0=1001$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_3S_2S_1S_0$
00	000	1100
10	010	1010
01	100	1001
11	110	1011

l) Current State  $S_3S_2S_1S_0=1011$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_3S_2S_1S_0$
00	000	1101
10	010	1011
01	100	1100
11	110	1010

n) Current State  $S_3S_2S_1S_0=1101$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_3S_2S_1S_0$
00	000	1110
10	010	1000
01	100	1111
11	110	1001

p) Current State  $S_3S_2S_1S_0=1111$

Input $k_1k_2$	Output $C_n$ $c_3c_2c_1$	Next State $S_3S_2S_1S_0$
00	000	1111
10	010	1001
01	100	1110
11	110	1000

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: 0.11", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: 0.11", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: 0.11", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: 0.11", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: 0.11", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -0.11", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -0.11", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -0.11", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -0.11", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -0.11", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.35", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.35", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted:** Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.35", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Formatted** ... [8]

**Formatted** ... [9]

**Formatted** ... [10]

**Formatted** ... [11]

**Formatted** ... [12]

**Formatted** ... [13]

**Formatted** ... [14]

Outputs: Current State ( $S_3S_2S_1S_0$ ):

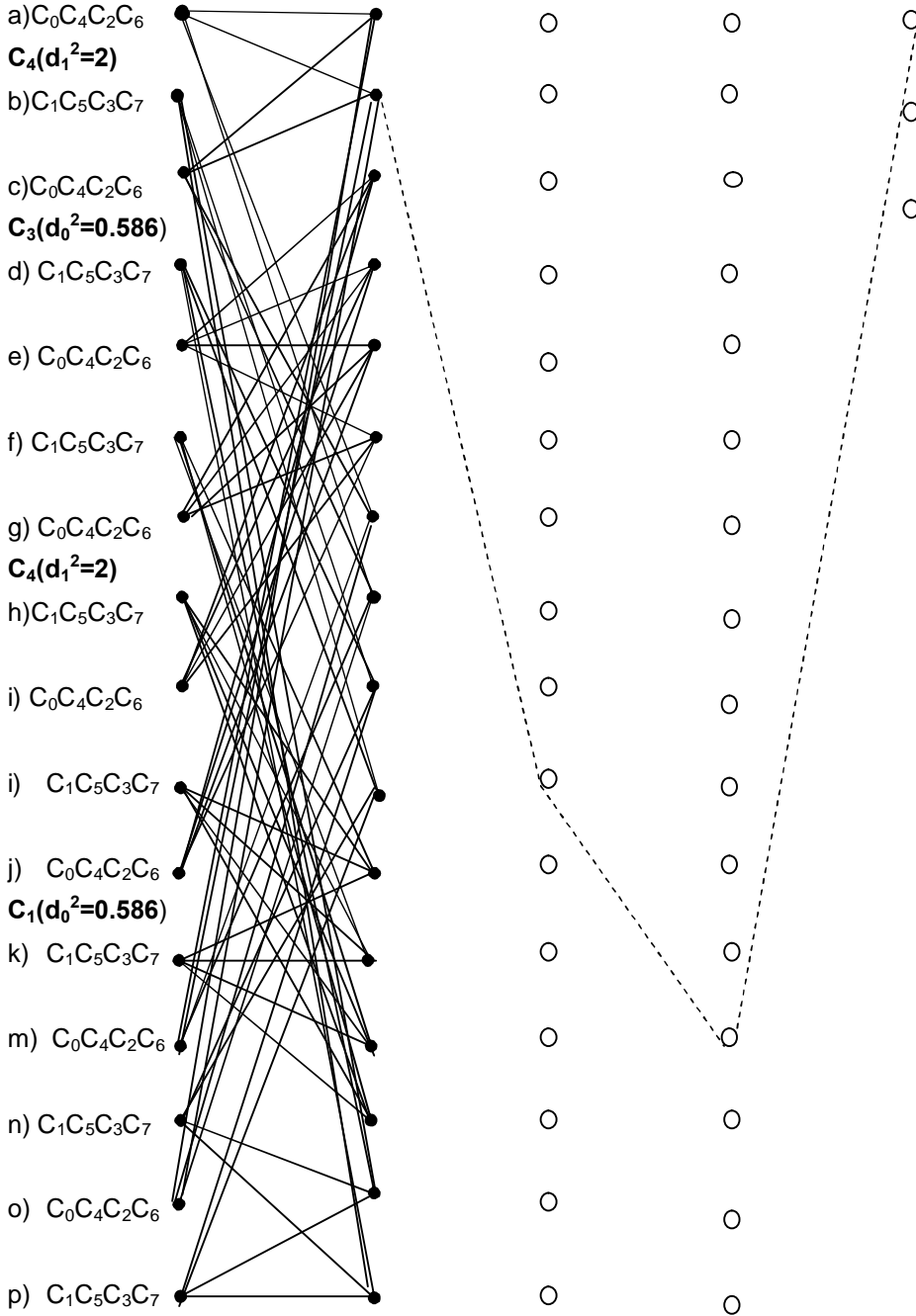


Figure 3.21 Trellis Diagram of a 16-State TCM/8-PSK [4]

Applying the second heuristic rule to the transitions, the four outputs originating from same state are grouped into a subset (quadruple) as  $C_0C_4C_2C_6$  and  $C_1C_5C_3C_7$  shown in constellation diagram of figure 3.22 and listed at LHS from  $t=0$  in trellis diagram of figure 3.21.

The output  $C_3C_2C_1$  are mapped into an 8-PSK constellation according to the second heuristic rule using the state table 3.5

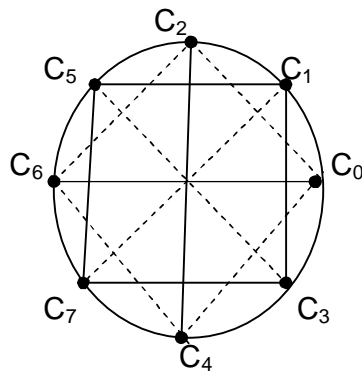


Figure 3.22 Mapping for a 16-State TCM/8-PSK in Constellation.

To determine coding gain, trellis diagram of figure 3.2 and the constellation diagram of figure 3.22 are used iteratively.

The quadruple subset output originating from same state in 8-PSK constellation have minimum Euclidean distance  $d_1^2 = 2$  (as in signal set partition). While the smallest total squared distance  $d_{2,2,2,2}^2$  (path that diverted from state 000 and then remerged into it) is determined as follows:

Transition paths are from state 0000 to 0001 resulting to output 100 or  $C_4$  on the constellation map, at distance of  $C_0C_4$  ( $d_1^2=2$ ), then from state 0001 to 1001 resulting to output 011 or  $C_3$  at distance of  $C_0C_3$  ( $d_0^2=0.586$ ); then from state 1001 to 1100 resulting to output 001 or  $C_1$  at distance of  $C_0C_1$  ( $d_0=0.586$ ) and  $C_4$  at distance of  $C_0C_4$  ( $d_1^2=2$ ), therefore

$$d_{2,2,2,2}^2 = d_{0,0}^2 + d_{0,0}^2 + d_{0,0}^2 + d_{0,0}^2 \dots \dots \dots 3.13$$

$$= 2 + 0.586 + 0.586 + 2 = 5.172$$

$$\therefore d_{2,2,2,2} = 10 \log \frac{5.172}{2} = 4.122$$

This 16-state TCM/8-PSK has yielded coding gain of 4.1dB. It is established in Communication Theory that by increasing the number of transition states of TCM to 32, 64, 128 and 256 more gains could be realized as shown in table 3.6.[4][10].

Table 3.6 Coding Gain Variation in Multiple Transition State TCM

S/N	Number of Transition states	Coding Gains for Trellis Coded Modulation (dB)				
		8PSK From QPSK	16-PSK From 8-PSK	16-QAM From 8-QAM	32-QAM From 16-QAM	64-QAM From 32-QAM
1	4	3.0	3.54	3.01	3.01	2.8
2	8	3.6	4.01	3.98	3.98	3.77
3	16	4.1	4.44	4.77	4.77	4.56
4	32	4	5.13	4.77	4.77	4.56
5	64	4.8	5.33	5.44	5.44	5.23
6	128	5	5.33	6.02	6.02	5.81
7	256	5.4	5.51	6.02	6.02	5.81

This Thesis work will simulate the higher coding gains of TCM/MSPK (M=2,4,16 and 32) using MATLAB in chapter 4, then select from the results appropriate MPSK for optimization of Data Throughput. A conceptual model that combined the multiple transition states in TCM/8-PSK scheme is proposed in which the differential coding gain is applied to counter the variations in Free Space Loss of a downlink during the satellite passage.

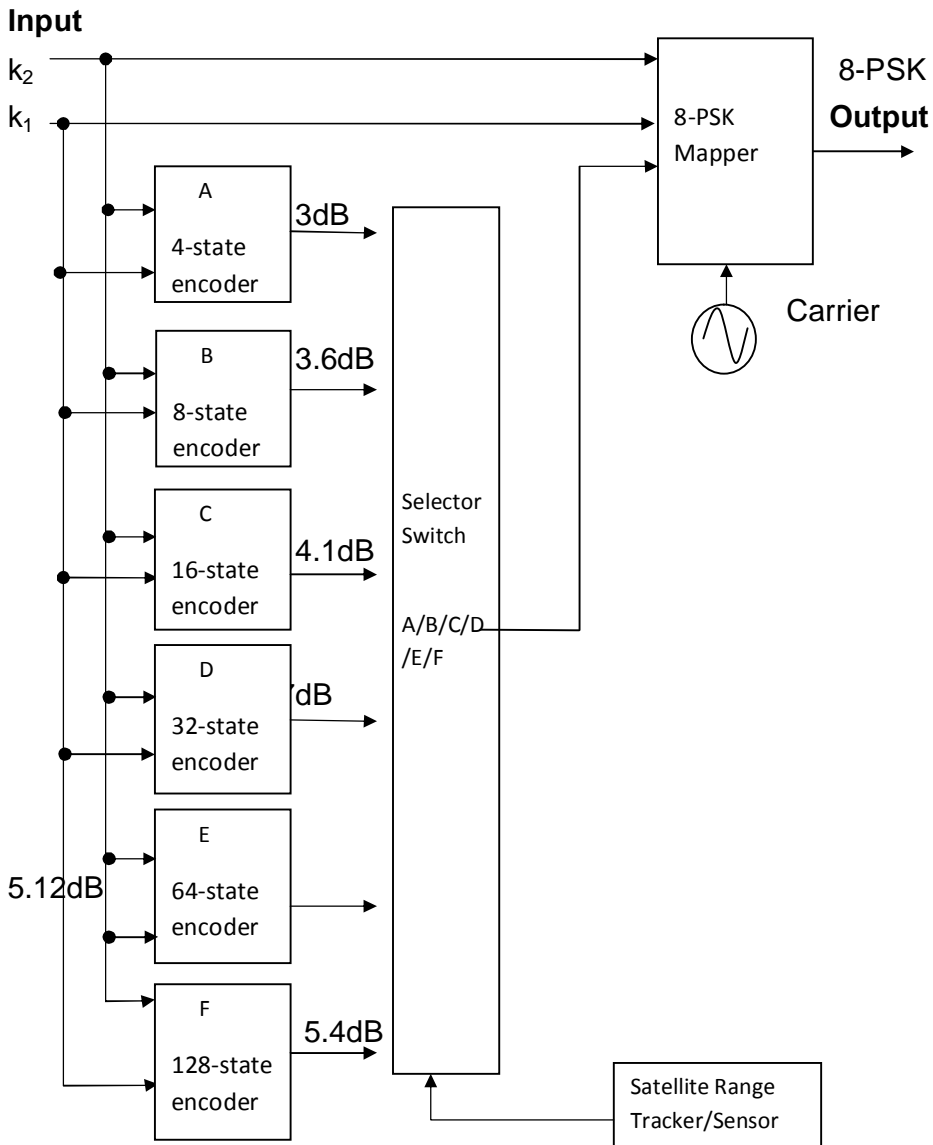
### 3.5 The Conceptual Model

#### An Integrated Multiple State TCM/8-PSK Encoders

A proposal is made to counter the variation of Free Space Loss (FSL) in the downlink (mentioned in chapter 2), by using the differential gains that exist in a TCM as in table 3.6. A model is designed with multiple TCM/8-PSK encoders as in figure 3.23, each has its number of state (4-, 8-, 16-, 32-, 64- and 128-state) transitions. The operation is to run in step, one encoder is to be selected at time to counter prevailing FSL with its coding gain. At low satellite elevation when maximum FSL is encountered, an encoder with maximum gain is selected to counter the loss. As the satellite elevation increases the loss reduces (equation 2.29) and the model's coding gain reduces too. The two input bits  $k_2, k_1$  of TCM/8-PSK are fed into all the multi-state (4, 8, 16, 64 and 128)



encoders as shown in the figure, a satellite range tracker or sensor that track motion of the satellite from horizon to horizon is to be connected to a selector-switch for theselection.



*Figure 3.23 An Integrated Multi-State TCM/8-PSK Model*

Due to large number of transition/output tables and complicated transition diagrams as seen in section 3.4.3, MATLAB software is used to design the multi-state encoders in section 3.5.1 for simplicity.

### 3.5.1 Design of Multi-State TCM/8-PSK Encoders with MATLAB

The MATLAB software make use of a polynomial called poly2trellis to specify code rate, number of shift registers and code generators of a convolution or Trellis Coded Modulation (TCM). When polynomial is properly stated and the encoder is realistic the MATLAB generates the Transition and Output tables. While the designer must properly enter polynomial in terms of number of line input /output, the number of shift registers and connection of the code generators, it is the MATLAB that calculate and determine if the encoder is practically realistic. Although the MATLAB brings out the Transition and Output table in decimal (not in binary as in table 3.4 &3.5) , but the polynomial must be stated in octal-decimal (base 8).

A  $\frac{1}{2}$ -rate encoder is specified using poly2trellis as follows:

Channel coding: **Convolution; poly2trellis(5, [35 31])**

Meaning one line input with constraint length = 5 (i.e. 4 shift registers + input bit).

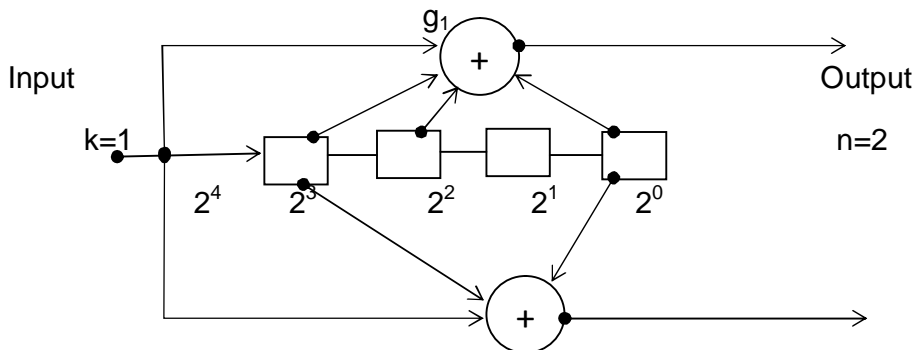
And there are two code generators stated as 35 and 31 (each digit is specified in base 8) indicating how the registers contribute to two modulo-2 adders that produces the two output bits.

Code Generators = [35 31]

Converting the octal numbers to binary

= [011 101 011 001] in binary-decimal

Thus, code generators are  $g_1 = 11101$  and  $g_2 = 11001$  The encoder is graphically represented as shown in figure 3.22



**Figure 3.24**  $\frac{1}{2}$ -Rate Convolution Encoder from poly2trellis(5, [35 31])

### a) A 4-State Transition TCM/8-PSK

To design a 4- state transitionTCM/8-PSKencoder, an optimum code rate for TCM/8-PSK of  $k/n= 2/3$  is applied and since there are two input  $k = k_1+ k_2$ (the input is split in two andeach line input hasits constraint length).Then, guided by MATLAB Software note, the encoder polynomial wasiteratively formed and entered into MATLAB command window as:

```
poly2trellis(3 1,[0 7 1;1 0 0])
```

The MATLAB software checked the polynomial and returned the followings:

trellis =

numInputSymbols: 4 (i.e. 4 binary combinations of  $k_1k_2$ )

numOutputSymbols: 8 (i.e. 8 binary combinations  $c_3 c_2c_1$ )

numStates: 4 (meaning 4 possible transition states)

nextStates: [4x4 double](i.e.4 possible next state of registers  $S_1S_0$  for each current state in decimal)

outputs: [4x4 double] (i.e.4 possible output of combination  $c_3 c_2c_1$ for each of current statein decimal)

The implementation block diagram of the 4-State TCM/8PSK encoder is shown in figure3.25. From two input lines its constraint length is  $(1+3)=4$

These two commandstrellis.nextStates(1:4,:) and trellis.outputs(1:4,:) are entered into the MATLAB Windows to generate the next states [4x4] and outputs [4x4] tables respectively. The MATLAB returns the two tables separately, each with entries in decimal number. The two tables are combined and aligned in sequential order of shift registers  $S_1S_0$  current transition statesas shown in table 3.7.

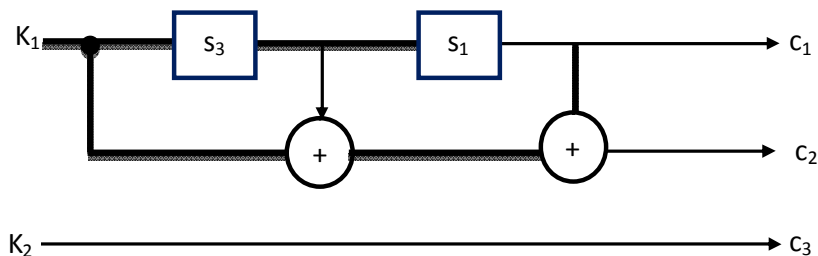


Figure 3.25A 4-State TCM/8-PSK Encoder Poly2trellis([3 1],[0 7 1;1 0 0])

Table 3.7 Transition and Output Table for a 4-State TCM/8-PSK

Current State	Next States	TCM Output	$S_1S_0$	$S_1S_0(c_3c_2c_1)$
00	0 0 2 2	0 4 2 6		
01	0 0 2 2	3 7 1 5		
10	1 1 3 3	2 6 0 4		
11	1 1 3 3	1 5 3 7		

The first line of the table is interpreted as: At state  $S_1S_0 = 00$ , the possible next states are 0, 0, 2 and 2 or 00, 00, 10 and 10 with corresponding MATLAB outputs  $c_3c_2c_1$  as 0, 4, 2 and 6 or 000, 100, 010 and 110. Although MATLAB does not produce a Trellis diagram but the parallel transition of a 4-state transition seen in figure 3.15 are obvious in the repetitive next state transitions in the table. The outputs are mapped into constellation points according to the rules of constellation mapping as shown in figure 3.26.

**C<sub>2</sub>(010)**

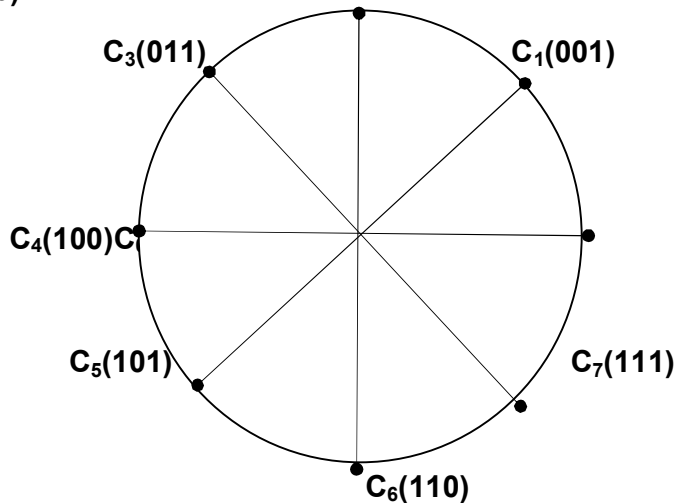


Figure 3.26 Mapping for a 4-State TCM/8-PSK in Constellation

For Coding gain calculation table 3.7 and figure 3.26 are used to select the minimum transition path  $D_{free}^2$  (transition that depart state 00 and return after two or more transitions to state 00). Since there is no trellis

diagram from the MATLAB, each transition path selected on the table is stated as: Current State(in binary) /Next State (in decimal)/TCM Output Symbol(in decimal) and an arrow to indicate a transition path. The following three candidate paths are considered:

a) Current State /Next State/TCM Output Symbol:  
 00/0/4 (parallel with 00/0/0)  
 Output symbols on this parallel path is  $C_4$  only  
 Using figure 3.26, the total path distance  $D_{free}^2 = C_4$   
 $D_{free}^2 = 4$

b) Current State/ Next State/Output:  
 00/2/2 10/1/2 001/0/3  
 Thus  $D_{free}^2 = (C_2 + C_2 + C_3)$   
 $= 2 + 2 + \sqrt{2} = 4.586$

c) Current State/ Next State/ Output:  
 00/2/6 → 10/1/6 01/0/7  
 thus using constellation map of figure 3.26  
 $D_{free}^2 = C_6 + C_6 + C_1$   
 $2 + 2 + 2 - \sqrt{2} = 4.586$

Option (a) is the path with minimum  $D_{free}^2 = 4.0$

$$\therefore \text{Rate} = 10 \log_{10} \frac{4.0}{2} = 3.0 \text{ dB}$$

**ii) 8-State Transition TCM /8-PSK Encoder**

The polynomial code to generate 8 transition states TCM on 8-PSK is iteratively formed by increasing constraint length by one (adding one shift register) to the 4-state TCM/8-PSK and entered in the MATLAB command window as:

```
trellis = tcm(3, 1, 4, 0, 3)
```

The MATLAB returns

trellis =

- numInputSymbols: 4 (4 logical combinations of  $k_1k_2$ )
- numOutputSymbols: 8 (8 logical combinations  $c_3c_2c_1$ )
- numStates: 8 (8 logical transition states  $S_2S_1S_0$ )

nextStates: [8x4 double] ( 4 possible next state of  $S_2S_1S_0$  from each of the 8 transitions)

outputs: [8x4 double] (4 possible output of  $c_1c_2c_3$  from each of the 8 transitions)

The implementation block diagram of the 8-StateTCM/8PSK encoder is shown in figure 3.27. From two input linesit constraints length is  $(2+3)=5$

The two commandstrellis.nextStates(1:8,:) and trellis.outputs(1:8,:) are entered into the MATLAB Windowsto generate the next states [8x4] and outputs [8x4] tables respectively. The MATLAB returns the two tables separately, the next state and output symbols are presented in decimal number. The two tables are combined and aligned in sequential order of current state of the Shift registers $S_2S_1S_0$ as shown in table 3.8

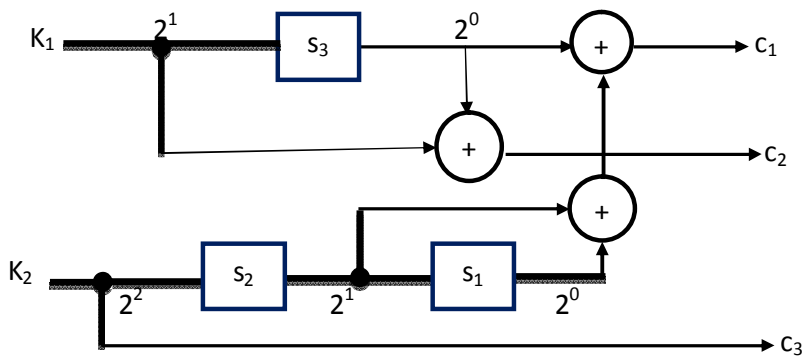


Figure 3.27An 8-State TCM/8-PSK Encoder:poly2trellis([2 3],[0 3 1; 4 0 3])

Table 3.88-State TCM/8-PSK Transition and Output table

Current State	Next States	TCM	Outputs	Symbols	Mapping
$S_2S_1S_0$	$S_2S_1S_0$	$(c_3c_2c_1)$			
0 0 0	0 4 1 5	0 4 2 6			
0 0 1	0 4 1 5	3 7 1 5			
0 1 0	0 4 1 5	5 3 7			
0 1 1	0 4 1 5	2 6 0 4			
1 0 0	2 6 3 7	1 5 3 7			
1 0 1	2 6 3 7	2 6 0 4			
1 1 0	2 6 3 7	0 4 2 6			
1 1 1	2 6 3 7	3 7 1 5			

For instance at state  $S_2S_1S_0 = 100$ , the possible next states are 2, 6, 3 and 7 or 010, 110, 011 and 111 with corresponding outputs  $c_3c_2c_1$  as 1, 5, 3 and 7 or 001, 101, 011 and 111. Thus applying the mapping rules to this set of outputs to obtain the constellation diagram of figure 3.28.

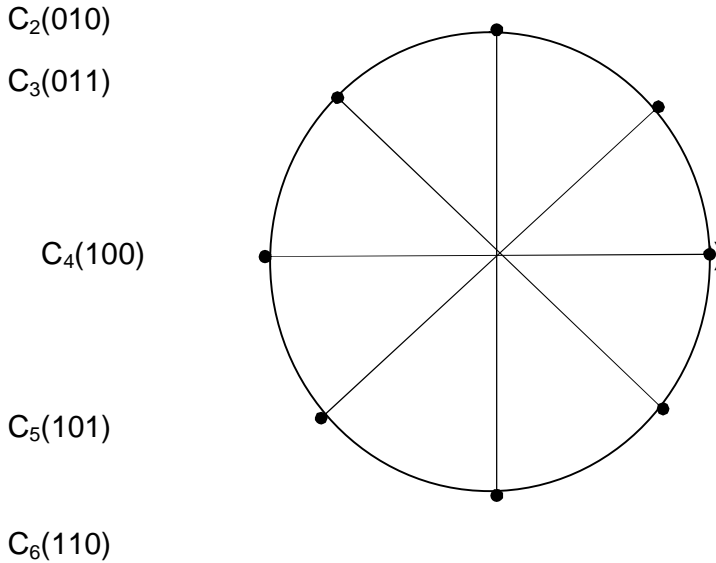


Figure 3.28 Mapping for an 8-State TCM/8-PSK in Constellation

To calculate the expected Coding Gain against uncoded QPSK using table 3.8 to determine the minimum  $D_{\text{free}}^2$  transition path, the following paths that depart State 000 and return to it with minimum length are considered:

- Current State /Next State/Output Symbol ( $c_3c_2c_1$ ) :  
 $000/4/4 \rightarrow 100/2/2 \rightarrow 010/0/1$   
 Thus, transition path length =  $(C_4 + C_2 + C_1)$   
 $D_{\text{free}}^2 = 4 + 2 + \sqrt{2} = 6.586$
  - Current State/ Next State/Output:  
 $000/1/2 \rightarrow 001/0/3 \rightarrow 001/0/4$   
 Thus, transition path length  $D_{\text{free}}^2 = (C_2 + C_3)$   
 $= 2 + \sqrt{2} = 5.414$
  - Current State/ Next State/ Output:  
 $000/5/6 \rightarrow 101/2/2 \rightarrow 010/0/1$   
 Thus, transition path length  $D_{\text{free}}^2 = C_6 + C_2 + C_1$   
 $= 2 + 2 + \sqrt{2} = 4.586$
- Option (c) is the path with  $D_{\text{free}}^2 = 4.586$

$$\therefore \text{Rate} = 10 \log_{10} \frac{2^4}{2^6} = 10 \log_{10} \frac{4.586}{2} = 3.6 \text{ dB}$$

### iii) 16-State Transition TCM/ 8-PSK

The polynomial code to generate 16 transition states TCM on 8-PSK is iteratively formed by increasing constraint length by one (adding one shift register) to the 8-state TCM/8-PSK and entered in the MATLAB command window as:

```
trellis = trellis(2, 4, 3, 1; 10 1 5 2)
```

The MATLAB checked the polynomial and returned the followings:

```
trellis =
  numInputSymbols: 4   (k1k2)
  numOutputSymbols: 8   (c3c2c1)
  numStates: 16        (meaning 16 possible transition state)
  nextStates: [16x4 double]
  outputs: [16x4 double]
```

The implementation block diagram of the 16-StateTCM/8PSK encoder is shown in figure 3.29.From the two input lines its constraint length is (2+4)=6.

These commandstrellis.nextStates(1:16,:) and trellis.outputs(1:16,:) are entered into the MATLAB Windows to generate the next states [16x4] and outputs [16x4] tables respectively. The MATLAB returns the two tables separately, with entries in decimal number representing a possible next states and corresponding outputs. The two tables are combined and aligned in sequential order of shift registers **S<sub>3</sub>S<sub>2</sub>S<sub>1</sub>S<sub>0</sub>**transition state as shown in table 3.9.



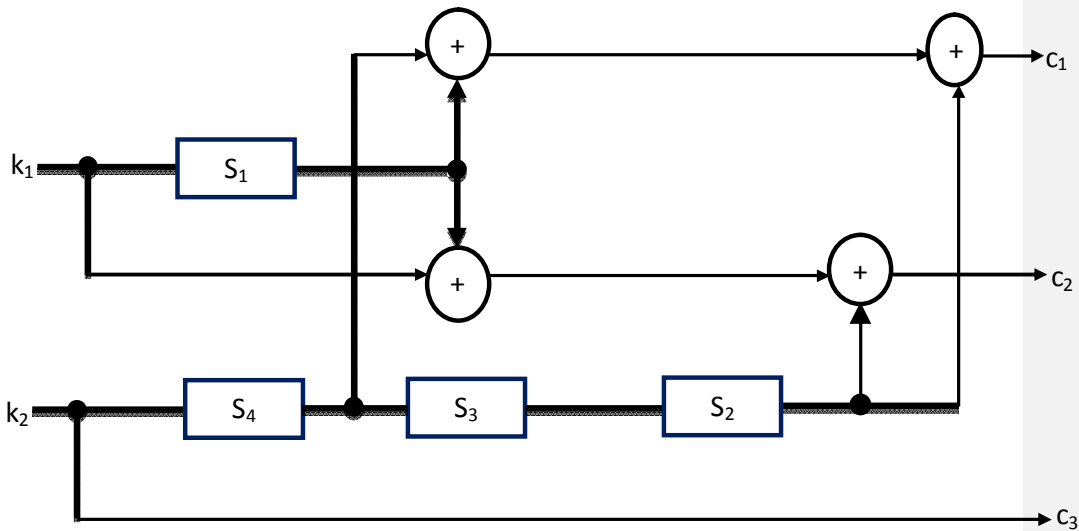


Figure 3.29A 16-State TCM/8-PSK Encoder:  $\text{poly2trellis}([2\ 4],[0\ 3\ 1;10\ 1\ 5])$

Table 3.916-State TCM/8-PSK Transition and Output table

Current State	Next States	TCM Outputs
$S_3 S_2 S_1 S_0$	$S_3 S_2 S_1 S_0$	$(c_3 c_2 c_1)$
0000	0 8 1 9	0 4 2 6
0001	0 8 1 9	3 7 1 5
0010	0 8 1 9	3 7 1 5
0011	0 8 1 9	0 4 2 6
0100	2 10 3 11	0 4 2 6
0101	2 10 3 11	3 7 1 5
0110	2 10 3 11	3 7 1 5
0111	2 10 3 11	0 4 2 6
1000	4 12 5 13	1 5 3 7
1001	4 12 5 13	2 6 0 4
1010	4 12 5 13	2 6 0 4
1011	4 12 5 13	1 5 3 7
1100	6 14 7 15	1 5 3 7
1101	6 14 7 15	2 6 0 4
1110	6 14 7 15	2 6 0 4
1111	6 14 7 15	1 5 3 7

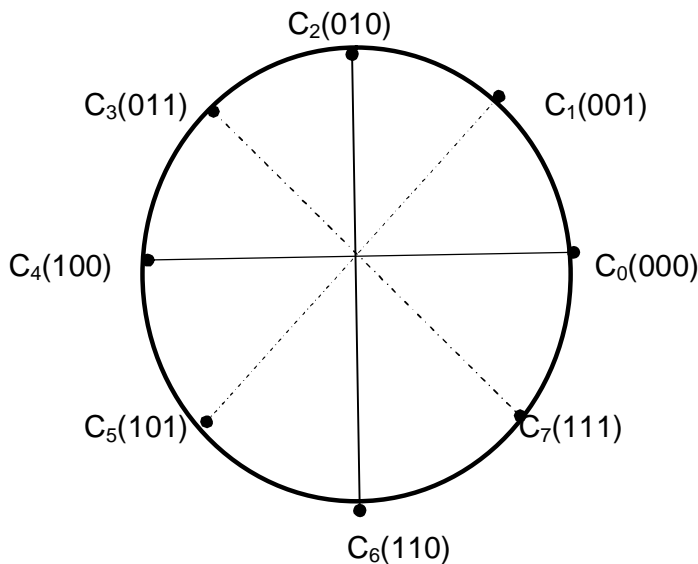


Figure 3.30 Mapping for a 16-State TCM/8-PSK in Constellation

For instance at state  $S_3S_2S_1S_0 = 0000$ , the possible next states are 0, 8, 1 and 9 or 0000, 1000, 0001 and 1001 with corresponding outputs  $c_3c_2c_1$  as 0, 4, 2 and 6 or 000, 100, 010 and 110. Thus applying the mapping rules to this set of outputs using constellation figure 3.30

From the table 3.09 and figure 3.30 paths that depart State 0000 available and return to it are as follows

a) Current State/ Next State/Output:

0000/8/41000/4/1 0100/0/2

Thus, transition path length  $D_{\text{free}}^2 = C_4 + C_1 + C_3$

$$= 4 + \sqrt{2} + \sqrt{2} = 8$$

b) Current State/Next State/Output:

0000/1/20001/0/30000

Thus, transition path length  $D_{\text{free}}^2 = C_2 + C_3$

$$\therefore \sqrt{2} + \sqrt{2} = 2 + \sqrt{2} = 5.414$$

c) Current State/ Next State/Output:

0000/9/6 1001/4/2 0100/1/2 0001/0/4

Thus  $D_{\text{free}}^2$  corresponding to the three outputs:

$$D_{\text{free}}^2 = C_6 + C_2 + C_2 + C_3$$

$$\therefore = 2 + 2 + 2 + \sqrt{2} + \sqrt{2} = 9.414$$

Option (b) has path with least  $D^2_{\text{free}} = 5.414$

$$\begin{aligned} \therefore \frac{5.414}{2} &= 10 \log_{10} \frac{2^{\frac{5.414}{2}}}{2^{\frac{5.414}{2}}} \\ \therefore \frac{5.414}{2} &= 10 \log_{10} \frac{5.414}{2} \\ &= 4.325 \end{aligned}$$

**iv) A 32- State Transition TCM/8PSK**

The polynomial code to generate 32 transition states TCM on 8-PSK is iteratively formed by increasing constraint length by one (adding one shift register) to the 16-state TCM/8-PSK and entered on the MATLAB command window as:

```
poly2trellis(3, [0 5 3; 10 11 3])
```

MATLAB checked the polynomial and returned the followings:

```
trellis =
  numInputSymbols: 4
  numOutputSymbols: 8
  numStates: 32      (meaning 32 possible transition states)
  nextStates: [32x4 double]
  outputs: [32x4 double]
```

The implementation block diagram of 32-State TCM/8PSK encoder is shown in figure 3.3. From two input lines its constraints length is (3+4)=7. The two commands `trellis.nextStates(1:32,:)` and `trellis.outputs(1:32,:)`

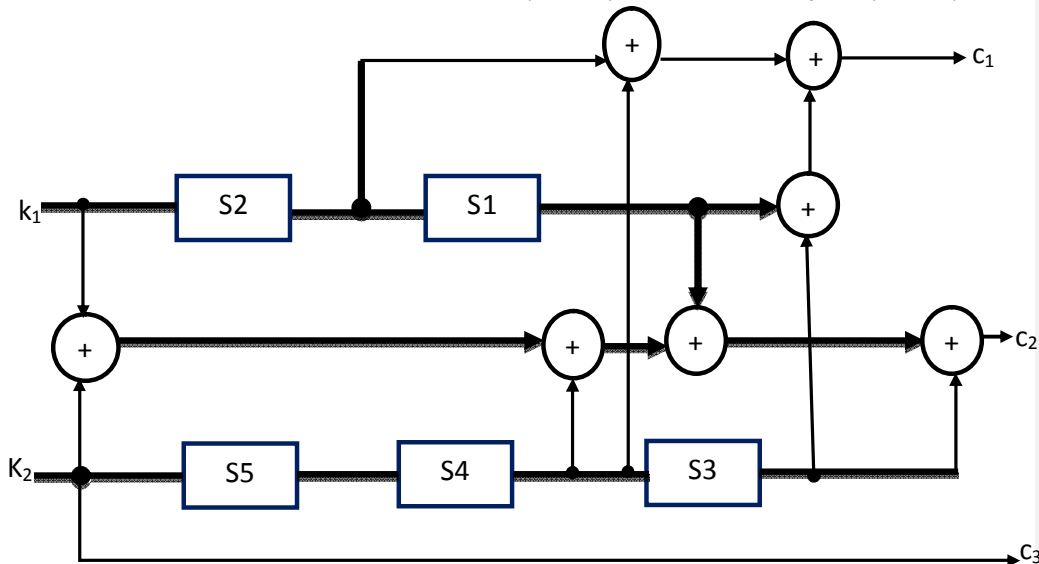


Figure 3.31A 32-State TCM/8-PSK Encoder: `poly2trellis([3 4],[0 5 3;10 11 3])`

are entered into the MATLAB Windowsto generate the next states [32x4] and outputs [32x4] tables respectively. The MATLAB returns the two tables separately, with entries in decimal number representing a possible next states and corresponding outputs. The two tables are combined and aligned in sequential order of shift registers  $S_4S_3S_2S_1S_0$  current transition state as shown in table 3.10.

Table 3.1032-State TCM/8-PSK Transition and Output table

<b>Current State</b>	<b>Next States</b>	<b>TCM Outputs</b>
$S_4S_3S_2S_1S_0$	$S_4S_3S_2S_1S_0$	$(c_3c_2c_1)$
0 0 0 0 0.	0 16 2 18	0 6 2 4
0 0 0 0 1.	0 16 2 18	3 5 1 7
0 0 0 1 0.	1 17 3 19	1 7 3 5
0 0 0 1 1	1 17 3 19	2 4 0 6
0 0 1 0 0	0 16 2 18	3 5 1 7
0 0 1 0 1	0 16 2 18	0 6 2 4
0 0 1 1 0	1 17 3 19	2 4 0 6
0 0 1 1 1.	1 17 3 19	1 7 3 5
0 1 0 0 0.	4 20 6 22	1 7 3 5
0 1 0 0 1.	4 20 6 22	2 4 0 6
0 1 0 1 0	5 21 7 23	0 6 2 4
0 1 0 1 1	5 21 7 23	3 5 1 7
0 1 1 0 0.	4 20 6 22	2 4 0 6
0 1 1 0 1.	4 20 6 22	1 7 3 5
0 1 1 1 0	5 21 7 23	3 5 1 7
0 1 1 1 1.	5 21 7 23	0 6 2 4
1 0 0 0 0.	8 24 10 26	0 6 2 4
1 0 0 0 1.	8 24 10 26	3 5 1 7
1 0 0 1 0.	9 25 11 27	1 7 3 5
1 0 0 1 1.	9 25 11 27	2 4 0 6
1 0 1 0 0.	8 24 10 26	3 5 1 7
1 0 1 0 1.	8 24 10 26	0 6 2 4
1 0 1 1 0.	9 25 11 27	2 4 0 6
1 0 1 1 1.	9 25 11 27	1 7 3 5
1 1 0 0 0.	12 28 14 30	1 7 3 5
1 1 0 0 1.	12 28 14 30	2 4 0 6
1 1 0 1 0.	13 29 15 31	0 6 2 4
1 1 0 1 1.	13 29 15 31	3 5 1 7
1 1 1 0 0.	12 28 14 30	2 4 0 6
1 1 1 0 1.	12 28 14 30	1 7 3 5
1 1 1 1 0.	13 29 15 31	3 5 1 7
1 1 1 1 1.	13 29 15 31	0 6 2 4

For instance at state  $S_4S_3S_2S_1S_0 = 00111$ , the possible next states are 1, 17, 3 and 19 or 00001, 10001, 00110 and 10011 with corresponding outputs  $c_3c_2c_1$  as 1, 7, 3 and 5 or 001, 111, 011 and 101. Thus applying the mapping rules to this set of outputs

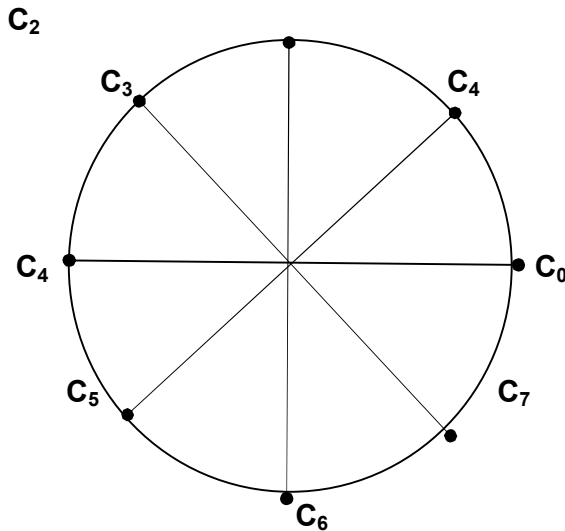


Figure 3.32 Mapping for a 32-State TCM/8-PSK in Constellation

From the tables 3.10 and figure 3.32 Transitions paths that depart State 0000 and return to it are as follows

- a) Current State/Next State/Output:  
 00000/16/6 10000/10/2 01010/7/2 00111/1/1 →  
 00001/0/3

Thus  $D_{\text{free}}^2$  corresponding to the five transition:

$$D_{\text{free}}^2 = C_6 + C_2 + C_2 + C_1 + C_3$$

$$= 2 + 2 + 2 + \sqrt{2} + \sqrt{2} + \sqrt{2} = 10$$

- b) Current States/Next State/ Output

0000/2/2 0001/0/1 0000/01/0/3

Thus  $D_{\text{free}}^2$  corresponding to the three outputs:

$$D_{\text{free}}^2 = C_2 + C_3 + C_1$$

$$\therefore = 2 + \sqrt{2} + \sqrt{2} + \sqrt{2} - \sqrt{2} = 6$$

- c) Current State/ Next State/Output:

00000/18/4 10010/9/1 1001/4/2 00100/0/3

Comment [D1]:

Thus  $D_{\text{free}}^2$  corresponding to the four transitions:

$$D_{\text{free}}^2 = C_4 + C_1 + C_2 + C_3$$

$$\therefore D_{\text{free}}^2 = 4 + 2\sqrt{2} - \sqrt{2} + 2 + 2\sqrt{2} + \sqrt{2} = 10$$

Option (b) has path with least  $D_{\text{free}}^2 = 6.0$

$$\therefore \text{Rate} = 10 \log_{10} \frac{2^4}{2^2 \cdot 2^2}$$

$$\begin{aligned} \therefore \text{Rate} &= 10 \log_{10} \frac{6.0}{2} \\ &= 4.771 \text{ bits/symbol} \end{aligned}$$

### v) A 64-State Transition TCM/8PSK

The polynomial code to generate 64 transition states TCM on 8-PSK is iteratively formed by increasing constraint length by one (adding one shift register) to the 32-state TCM/8-PSK and entered on the MATLAB command window as:

```
trellis = poly2trellis([4 4],[0 11 7; 10 4 3])
```

MATLAB checked the polynomial and returned the followings:

trellis =

```
numInputSymbols: 4
numOutputSymbols: 8
numStates: 64 (meaning 64 possible transition states)
nextStates: [64x4 double]
outputs: [64x4 double]
```

The block diagram of the 64-State Transition TCM/8PSK encoder is shown in figure 3.33. From two input lines its constraint length is  $(4+4)=8$

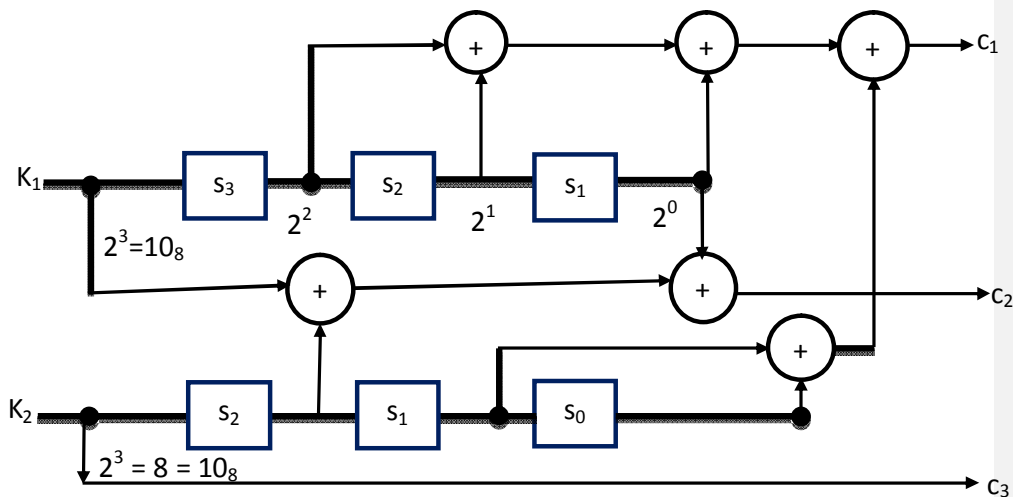


Figure 3.33A 64-State TCM/8-PSK Encoder: poly2trellis([4 4],[0 11 7; 10 4 3])

The two commands trellis.nextStates(1:64,:) and trellis.outputs(1:64,:) are entered into the MATLAB Windowsto generate the tables next states [64x4] and outputs [64x4] respectively. The MATLAB returns the two tables separately, with entries in decimal number representing a possible next states and corresponding outputs. The two tables are combined and aligned in sequential order of shift registers  $S_5S_4S_3S_2S_1S_0$  current transition state as shown in table 3.11.

For instance at state  $S_5S_4S_3S_2S_1S_0 = 010000$ , the possible next states are 8, 40, 12 and 44 or 001000, 101000, 001010 and 101100 with corresponding outputs  $c_3c_2c_1$  as 1, 5,3 and 7 or 001, 101,011 and 111. Thus applying the mapping rules to this set of outputs

Table 3.1164-State TCM/8-PSK Transition and Output table

Current State $S_5S_4S_3S_2S_1S_0$	Next States	TCM Outputs $(c_3c_2c_1)$
000000.	0 32 4 36	0 4 2 6
000001.	0 32 4 36	3 7 1 5
000010.	1 33 5 37	1 5 3 7
000011.	1 33 5 37	2 6 0 4
000100.	2 34 6 38	1 5 3 7
000101.	2 34 6 38	2 6 0 4
000110.	3 35 7 39	0 4 2 6
000111.	3 35 7 39	3 7 1 5
001000.	0 32 4 36	1 5 3 7
001001.	0 32 4 36	2 6 0 4
001010.	1 33 5 37	0 4 2 6
001011.	1 33 5 37	3 7 1 5
001100.	2 34 6 38	0 4 2 6
001101.	2 34 6 38	3 7 1 5
001110.	3 35 7 39	1 5 3 7
001111.	3 35 7 39	2 6 0 4
010000.	8 40 12 44	1 5 3 7
010001.	8 40 12 44	2 6 0 4
010010.	9 41 13 45	0 4 2 6
010011.	9 41 13 45	3 7 1 5

010100.	10	42	14	46	0	4	2	6
010101.	10	42	14	46	3	7	1	5
010110.	11	43	15	47	1	5	3	7
010111.	11	43	15	47	2	6	0	4
011000.	8	40	12	44	0	4	2	6
011001.	8	40	12	44	3	7	1	5
011010.	9	41	13	45	1	5	3	7
011011.	9	41	13	45	2	6	0	4
011100.	10	42	14	46	1	5	3	7
011101.	10	42	14	46	2	6	0	4
011110.	11	43	15	47	0	4	2	6
011111.	11	43	15	47	3	7	1	5
100000.	16	48	20	52	2	6	0	4
100001.	16	48	20	52	1	5	3	7
100010.	17	49	21	53	3	7	1	5
100011.	17	49	21	53	0	4	2	6
100100.	18	50	22	54	3	7	1	5
100101.	18	50	22	54	0	4	2	6
100110.	19	51	23	55	2	6	0	4
100111.	19	51	23	55	1	5	3	7
101000.	16	48	20	52	3	7	1	5
101001.	16	48	20	52	0	4	2	6
101010.	17	49	21	53	2	6	0	4
101011.	17	49	21	53	1	5	3	7
101100.	18	50	22	54	2	6	0	4
101101.	18	50	22	54	1	5	3	7
101110.	19	51	23	55	3	7	1	5
101111.	19	51	23	55	0	4	2	6
110000.	24	56	28	60	3	7	1	5
110001.	24	56	28	60	0	4	2	6
110010.	25	57	29	61	2	6	0	4
110011.	25	57	29	61	1	5	3	7
110100.	26	58	30	62	2	6	0	4
110101.	26	58	30	62	1	5	3	7
110110.	27	59	31	63	3	7	1	5
110111.	27	59	31	63	0	4	2	6
111000.	24	56	28	60	2	6	0	4



1 1 1 0 0 1.	24	56	28	60	1	5	3	7
1 1 1 0 1 0.	25	57	29	61	3	7	1	5
1 1 1 0 1 1	25	57	29	61	0	4	2	6
1 1 1 1 0 0.	26	58	30	62	3	7	1	5
1 1 1 1 0 1.	26	58	30	62	0	4	2	6
1 1 1 1 1 0.27	59	31	63	2	6	0	4	
1 1 1 1 1 1.	27	59	31	63	1	5	3	7

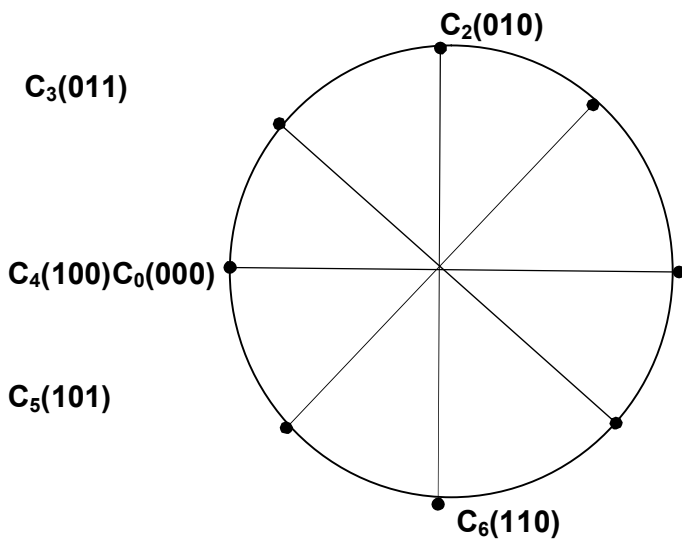


Figure 3.34 Mapping for 64-State TCM/8-PSK in Constellation

Using Mapping table 3.11 and figure 3.34 the paths with smaller  $D_{free}^2$  that depart State 000000 and return to it are as follows

- a) Current State/Next State/Output:  
 000000/32/4 → 000000/16/2010000/8/1 → 0010000/0/1

Thus  $D_{free}^2$  corresponding to the five outputs:

$$D_{free}^2 = C_4 + C_2 + C_1 + C_1 = 4 + 2 + 2\sqrt{2} - \sqrt{2} = 7.17157$$

- b) Current States/ Next State /Output:  
 000000/4/2 → 000100/2/1000010/1/1 → 000000/0/3

Thus  $D_{free}^2$  corresponding to the three outputs:

$$D_{free}^2 = C_2 + C_1 + C_1 + C_3 = 2 + 2\sqrt{2} - \sqrt{2} + \sqrt{2} + \sqrt{2} = 6.5858$$

Comment [D2]:

c) Current State/Next State/ Output:  
 000000/36/6 → 100110/18/3 → 010010/13/2 →  
 01101/2/30000 → 10/1/1000001/0/3 →

Thus  $D_{free}^2$  corresponding to the four transitions:  
 $D_{free}^2 = C_6 + C_3 + C_2 + C_3 + C_1 + C_3$   
 $\therefore D_{free}^2 = 2 + 4 + \sqrt{2} + 2 - \sqrt{2} = 6.5858$   
 Option (b) has the least  $D_{free}^2 = 6.5858$

$$\therefore R_{TCM} = 10 \log_{10} \frac{6.5858}{2} = 5.1759 \text{ dB}$$

**vi) A 128-State Transition TCM/8PSK**

The polynomial code to generate 128 transition states TCM on 8-PSK is iteratively formed by increasing the constraint length by one (adding one shift register) to the 64-state TCM/8-PSK and entered on the MATLAB command window as:

```
trellis = tcm(4, 23 14; 10 15 3)
```

MATLAB checked the polynomial and returned the followings:

```
trellis =
  numInputSymbols: 4
  numOutputSymbols: 8
  numStates: 128 (meaning 128 possible transition states)
  nextStates: [128x4 double]
  outputs: [128 x4 double]
```

The block diagram of the 128 -State Transition TCM/8PSK encoder is shown in figure 3.34. From two input lines its constraint length is (5+4)=9.

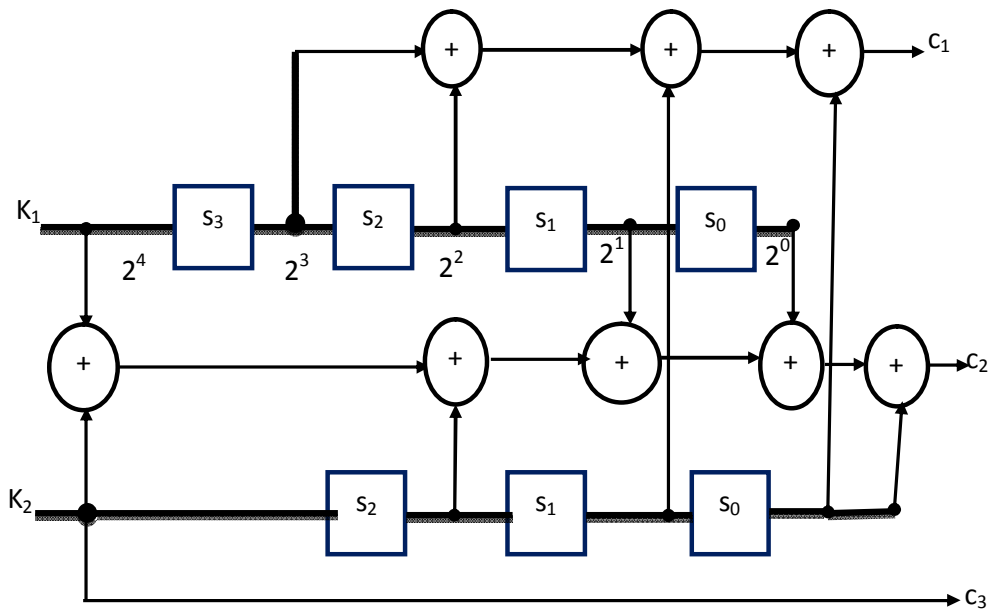


Figure 3.34 128-State TCM/8PSK Encoder:  $\text{poly2trellis}([5\ 4], [0\ 23\ 14; 10\ 15\ 3])$

The two command `trellis.nextStates(1:128,:)` and `trellis.outputs(1:128,:)` are entered into the MATLAB Window to generate the tables `next states [128x4]` and `outputs [128x4]` respectively. The MATLAB returns the two tables separately, with entries in decimal number representing a possible next states and corresponding outputs. The two tables are combined and aligned in sequential order of shift registers  $S_6 S_5 S_4 S_3 S_2 S_1 S_0$  current transition state as shown in table

For instance at state  $S_6 S_5 S_4 S_3 S_2 S_1 S_0 = 1110000$ , the possible next states are 48, 112, 56 and 120 or 0110000, 1110000, 0111000 and 1111000 with corresponding outputs  $c_3 c_2 c_1$  as 0, 6, 2 and 4 or 000, 110, 010 and 100. Thus applying the mapping rules to this set of outputs on each row as shown in table 3.12.

Table 3.12 128-State TCM/8-PSK Transition and Output table

Current State	Next States								TCM						
	S <sub>6</sub>	S <sub>5</sub>	S <sub>4</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	S <sub>6</sub>	S <sub>5</sub>	S <sub>4</sub>	S <sub>3</sub>	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	(C <sub>3</sub> C <sub>2</sub> C <sub>1</sub> )
0000000.	0	64	8	72											0 6 2 4
0000001.	0	64	8	72											2 4 0 6
0000010.	1	65	9	73											2 4 0 6
0000011.	1	65	9	73											0 6 2 4
0000100	2	66	10	74											1 7 3 5
0000101	2	66	10	74											3 5 1 7
0000110	3	67	11	75											3 5 1 7
0000111.	3	67	11	75											1 7 3 5
0001000.	4	68	12	76											1 7 3 5
0001001.	4	68	12	76											3 5 1 7
0001010.	5	69	13	77											3 5 1 7
0001011.	5	69	13	77											1 7 3 5
0001100.	6	70	14	78											0 6 2 4
0001101.	6	70	14	78											2 4 0 6
0001110.	7	71	15	79											2 4 0 6
0001111.	7	71	15	79											0 6 2 4
0010000.	0	64	10	72											3 5 1 7
0010001.	0	64	10	72											1 7 3 5
0010010.	1	65	11	73											1 7 3 5
0010011.	1	65	11	73											3 5 1 7
0010100.	2	66	10	74											2 4 0 6
0010101.	2	66	10	74											0 6 2 4
0010110.	3	67	11	75											0 6 2 4
0010111.	3	67	11	75											2 4 0 6
0011000.	4	68	14	76											2 4 0 6
0011001.	4	68	14	76											0 6 2 4
0011010.	5	69	15	77											0 6 2 4
0011011.	5	69	15	77											2 4 0 6
0011100.	6	70	14	78											3 5 1 7
0011101.	6	70	14	78											1 7 3 5
0011110.	7	71	15	79											1 7 3 5
0011111.	7	71	15	79											3 5 1 7
0100000.	16	80	24	88											1 7 3 5
0100001.	16	80	24	88											3 5 1 7

0100010.	17	81	25	89	3	5	1	7
0100011.	17	81	25	89	1	7	3	5
0100100.	18	82	26	90	0	6	2	4
0100101.	18	82	26	90	2	4	0	6
0100110.	19	83	27	91	2	4	0	6
0100111.	19	83	27	91	0	6	2	4
0101000.	20	84	28	92	0	6	2	4
0101001.	20	84	28	92	2	4	0	6
0101010.	21	85	29	93	2	4	0	6
0101011.	21	85	29	93	0	6	2	4
0101100.	22	86	30	94	1	7	3	5
0101101	22	86	30	94	3	5	1	7
0101110	23	87	31	95	3	5	1	7
0101111.	23	87	31	95	1	7	3	5
0110000.	16	80	24	88	2	4	0	6
0110001.	16	80	24	88	0	6	2	4
0110010.	17	81	25	89	0	6	2	4
0110011.	17	81	25	89	2	4	0	6
0110100.	18	82	26	90	3	5	1	7
0110101	18	82	26	90	1	7	3	5
0110110.	19	83	27	91	1	7	3	5
0110111.	19	83	27	91	3	5	1	7
0111000.	20	84	28	92	3	5	1	7
0111001.	20	84	28	92	1	7	3	5
0111010.	21	85	29	93	1	7	3	5
0111011.	21	85	29	93	3	5	1	7
0111100.	22	86	30	94	2	4	0	6
0111101.	22	86	30	94	0	6	2	4
0111110.	23	87	31	95	0	6	2	4
0111111.	23	87	31	95	2	4	0	6
1000000	32	96	40	104	2	4	0	6
1000001	32	96	40	104	0	6	2	4
1000010	33	97	41	105	0	6	2	4
1000011	33	97	41	105	2	4	0	6
1000100	34	98	42	106	3	5	1	7
1000101	34	98	42	106	1	7	3	5
1000110	35	99	43	107	1	7	3	5

1000111	35	99	43	107	3	5	1	7
1001000	36	100	44	108	3	5	1	7
1001001	36	100	44	108	1	7	3	5
1001010	37	101	45	109	1	7	3	5
1001011	37	101	45	109	3	5	1	7
1001100	38	102	46	110	2	4	0	6
1001101	38	102	46	110	0	6	2	4
1001110	39	103	47	111	0	6	2	4
1001111	39	103	47	111	2	4	0	6
1010000	32	96	40	104	1	7	3	5
1010001	32	96	40	104	3	5	1	7
1010010	33	97	41	105	3	5	1	7
1010011	33	97	41	105	1	7	3	5
1010100	34	98	42	106	0	6	2	4
1010101	34	98	42	106	2	4	0	6
1010110	35	99	43	107	2	4	0	6
1010111	35	99	43	107	0	6	2	4
1011000	36	100	44	108	0	6	2	4
1011001	36	100	44	108	2	4	0	6
1011010	37	101	45	109	2	4	0	6
1011011	37	101	45	109	0	6	2	4
1011100	38	102	46	110	1	7	3	5
1011101	38	102	46	110	3	5	1	7
1011110	39	103	47	111	3	5	1	7
1011111	39	103	47	111	1	7	3	5
1100000	48	112	56	120	3	5	1	7
1100001	48	112	56	120	1	7	3	5
1100010	49	113	57	121	1	7	3	5
1100011	49	113	57	121	3	5	1	7
1100100	50	114	58	122	2	4	0	6
1100101	50	114	58	122	0	6	2	4
1100110	51	115	59	123	0	6	2	4
1100111	51	115	59	123	2	4	0	6
1101000	52	116	60	124	2	4	0	6
1101001	52	116	60	124	0	6	2	4
1101010	53	117	61	125	0	6	2	4
1101011	53	117	61	125	2	4	0	6

1101100	54	118	62	126	3	5	1	7
1101101	54	118	62	126	1	7	3	5
1101110	55	119	63	127	1	7	3	5
1101111	55	119	63	127	3	5	1	7
1110000	48	112	56	120	0	6	2	4
1110001	48	112	56	120	2	4	0	6
1110010	49	113	57	121	2	4	0	6
1110011	49	113	57	121	0	6	2	4
1111000	50	114	58	122	1	7	3	5
1110101	50	114	58	122	3	5	1	7
1110110	51	115	59	123	3	5	1	7
1110111	51	115	59	123	1	7	3	5
1111000	52	116	60	124	1	7	3	5
1111001	52	116	60	124	3	5	1	7
1111010	53	117	61	125	3	5	1	7
1111011	53	117	61	125	1	7	3	5
1111100	54	118	62	126	0	6	2	4
1111101	54	118	62	126	2	4	0	6
1111110	55	119	63	127	2	4	0	6
1111111	55	119	63	127	0	6	2	4

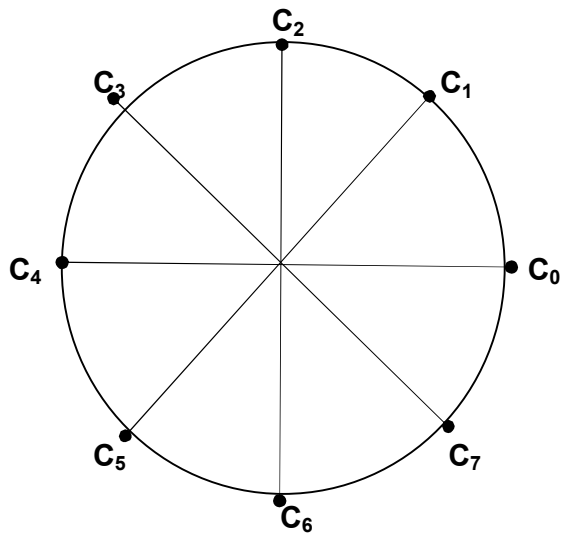


Figure 3.36 Mapping for 128-State TCM/8-PSK in Constellation

From the table 3.12 and figure 3.36 paths that depart State 0000 available and return to it are as follows:

a) Current State/Next State/Output:

0000000/64/6 → 1000000/32/2 → 0100000/16/1 → 0010000/0/3

Thus  $D_{free}^2$  corresponding to the five outputs:

$$D_{free}^2 = C_6 + C_2 + C_1 + C_3 = 2 + 2 + \sqrt{2} + \sqrt{2} + \sqrt{2} = 8$$

b) Current State/ Next State/Output:

0000000/8/2 → 0001000/4/1 → 0000100/2/1 → 0000010/1/2  
0000000/0/2

Thus  $D_{free}^2$  corresponding to the three outputs:

$$D_{free}^2 = C_2 + C_1 + C_1 + C_2 + C_2 = 2 + \sqrt{2} - \sqrt{2} + \sqrt{2} - \sqrt{2} + 2 + 2 = 7.1716$$

c) Current State/Next State/Output

000000/72/4 → 100000/44/1010110022/10010110/11/2 → 0001011/5/1  
0000101/2/3 → 0000010/1/2 0000001/0/2 0000000

Thus  $D_{free}^2$  corresponding to the four transitions:

$$D_{free}^2 = C_4 + C_1 + C_1 + C_2 + C_1 + C_3 + C_2 + C_2 = 4 + \sqrt{2} + \sqrt{2} + 3\sqrt{2} - \sqrt{2} + 6 = 15.1716$$

Option (c) has path with least  $D_{free}^2 = 7.1716$

$$\begin{aligned} \therefore \log_2 \frac{7.1716}{2} &= 10 \log_{10} \frac{7.1716}{2} \\ &= 5.546 \end{aligned}$$

In Chapter Four each of these encoders will be simulated in a MATLAB Downlink model to ascertain its gain before it is integrated into the model for final simulation.

## CHAPTER FOUR

Comment [D3]:



## COMPUTER SIMULATION FOR CODING GAIN

### 4.0 Introduction:

The Software used for computer simulation is MATLAB version R2010. An RF Satellite Link Block Model shown in figure 4.1 obtained from RF-linksample in the software is adopted as a template for development of LEO satellite downlink. The template is modified to focus on coding gain and bandwidth utilization of Multi-level PSK modulation scheme. In MATLAB software digital modulation schemes are implemented and presented at baseband with no carrier component. The modulated signals are therefore presented as simple baseband envelope of signal constellation points. This makes the simulation to run faster since less computation is involved.[13] Communication Theory allowed digital modulated signal to be presented at baseband for purpose of Signal Analysis and Simulation, so that the linear translation of frequency due to modulation is ignored [5]. In this chapter all modulators and demodulators are therefore in baseband, and five level of MPSK modulation ( $M = 2, 4, 8, 16$  and  $32$ ) are treated. Since the measurements are centered on Power ( $E_b/N_0$ ) against Bit Error Rate (BER),  $\mu$ bertool simulation tool in the MATLAB is extensively used for all the simulations. A standard BER of  $10^{-6}$  is adopted for all measurement, as it guarantee signal quality for remote sensing application.[1]

The chapter is divided into four stages, in each stage certain category of simulations are performed. In stage 1, uncoded and convolution coded Theoretical MPSK link models are simulated and plots of bit error rate (BER) against  $E_b/N_0$  are obtained. Then, LEO Satellite Downlink block models are formed using a template model and simulated. The aim in stage 1 is to compare performance of the block models with the theoretical. Parameter settings in the block models are fine-tuned until performance results that are comparable to theoretical are obtained. Convolution encoder/decoder block units are applied to all the models by specifying the appropriate polynomials. The simulation plots are superimposed for graphical comparison. In stage 2, TCM encoder/decoder block units are used to replace Convolution encoder/decoder and MPSK modulation/demodulator block units in the downlink block models. The TCM models are simulated for BER against  $E_b/N_0$  the plots are superimposed on the corresponding plots of stage 1.

In stage 3, the six multi-states TCM encoders proposed in section 3.9 are discretely simulated in a downlink block model for the expected gain. Attempts were made to simulate the proposed model as a whole, but due to unavailability of Satellite Tracking sensor/switch in the MATLAB, the model was modified to use a multiplexer to replace the switch.

In Stage 4, MPSK models on convolution coding and TCM are simulated for data throughput. Simulation run times of 4, 10 and 17 minutes are used, which correspond to the minimum, average and maximum LEO satellite visibility durations respectively.

## STAGE 1

### 4.1 MATLAB Bit Error Rate Analysis Tool (bertool).

A tool used in MATLAB for simulation of Bit Error Rate (BER) is called bertool. It generates simulation results and plots of BER against a specified range of SNR or  $E_b/N_o$ . To open the bertool window, simply enter the word: '**bertool**' on the MATLAB execute windows and the bertool window will open as shown in figure 4.1.

At the top of the window is the Data viewer for displaying simulation table of BER against  $E_b/N_o$ . Under the Data viewer are tabs for selecting three simulation methods labeled **Theoretical, Semi Analytical and Monte Carlo**. [13]:

a) **Theoretical Simulation** is performed on a theoretical link formed by specifying the following parameters in the bertool window:

- i) Received Signal power  $E_b/N_o$  range (**0:36 dB**)
- ii) Channel type (**AWGN, Rician and Raleigh**)
- iii) Modulation type (**PSK, FSK, QAM**)
- iv) Modulation order (**M**)
- v) Demodulation type (**Hard or Soft**)
- vi) Channel Coding (**Convolution or Block**)/None)

The theoretical simulation is run when **Plot** tab is selected on the windows. A graph of BER versus  $E_b/N_o$  is plotted by the software and stored. This method will be used extensively for comparison with block models in Stage I.

- b) The second tab is **Semi Analytic** method, this is for simulation of a link model formed using MATLAB software codes. It is more flexible than the Theoretical method as it allows more link parameters such as antenna gain, amplification gain, filter parameters etc. to be specified using MATLAB software codes. This method will not be used in this thesis work as all models are formed using block units only.
- c) The third method of BER analysis is on Monte Carlo tab, a link model is formed with block units in a MATLAB file, more detail link and block specific parameters are specified in each block unit. When Monte Carlo method is selected from bertool window, filename of the block model is specified and the simulation runs on the model at each specified SNR for specified number of errors. The simulation results from the block model is send to bertool Monte Carlo, as variables for plotting of BER versus  $E_b/N_0$  graph. This method will also be used extensively for BER simulations on block models.

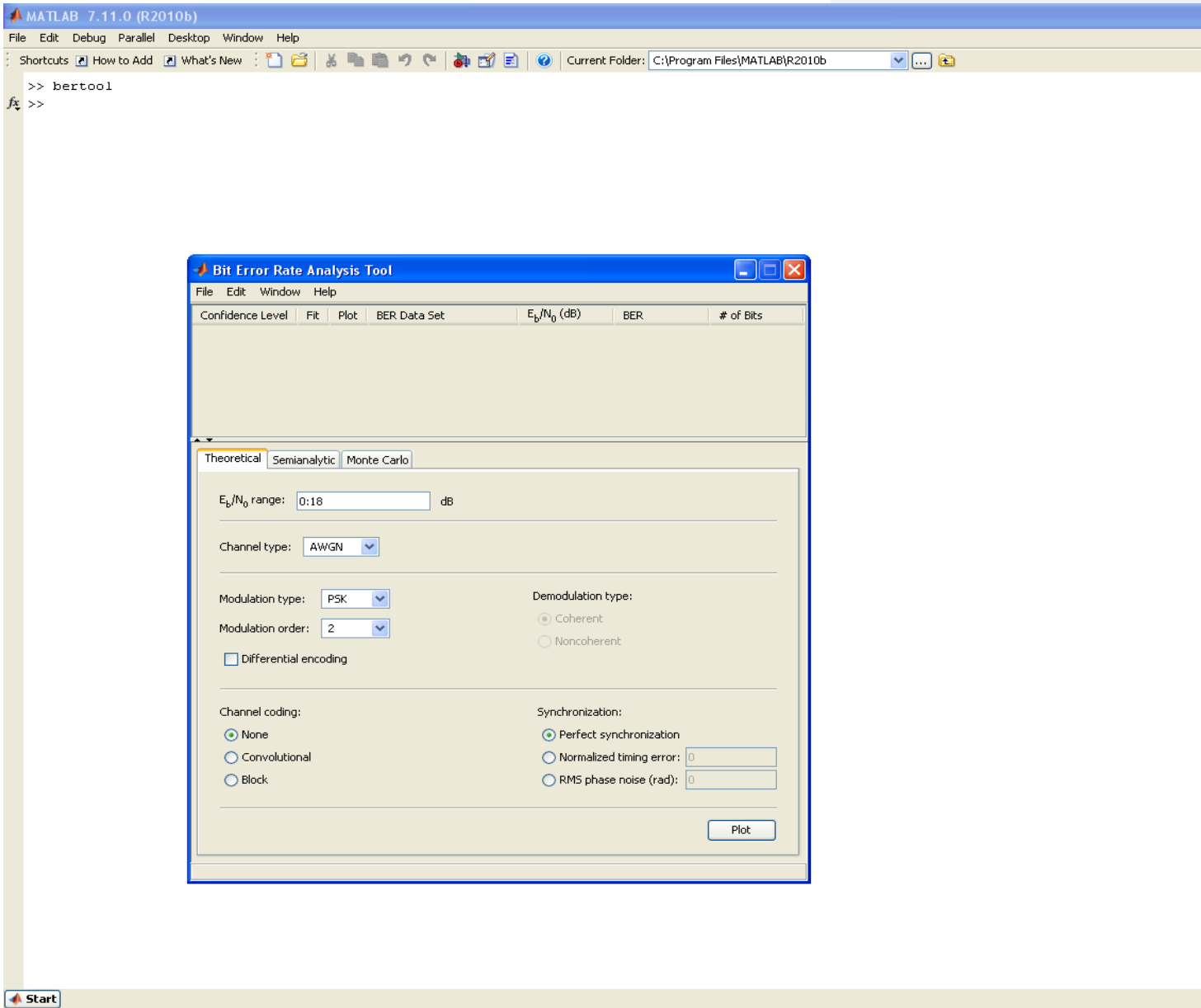


Figure 4.01 Bit Error Rate Analysis Tool (bertool) on MATLAB Execute window

## 4.2 Simulation of Theoretical MPSK Links.

### (a) Simulation of BPSK link

**Theoretical** tab is selected on the bertool window and the link parameters for BPSK scheme are entered as follows:

- (i)  $E_b/N_o$  range: **0: 18**(dB)
- (ii) Channel type: **AWGN**
- (iii) Modulation type: **PSK**
- (iv) Modulation order: **2**
- (v) Demodulation mode : **Hard**
- (vi) Channel coding: **None**

A click on **plot** tab

The bertool runs the simulation of uncoded BPSK link from  $E_b/N_o = 0$  to 18dB in an AWGN channel with demodulation of hard decision type. A graph of **BER** versus  $E_b/N_o$  is plotted as shown in figure 4.2

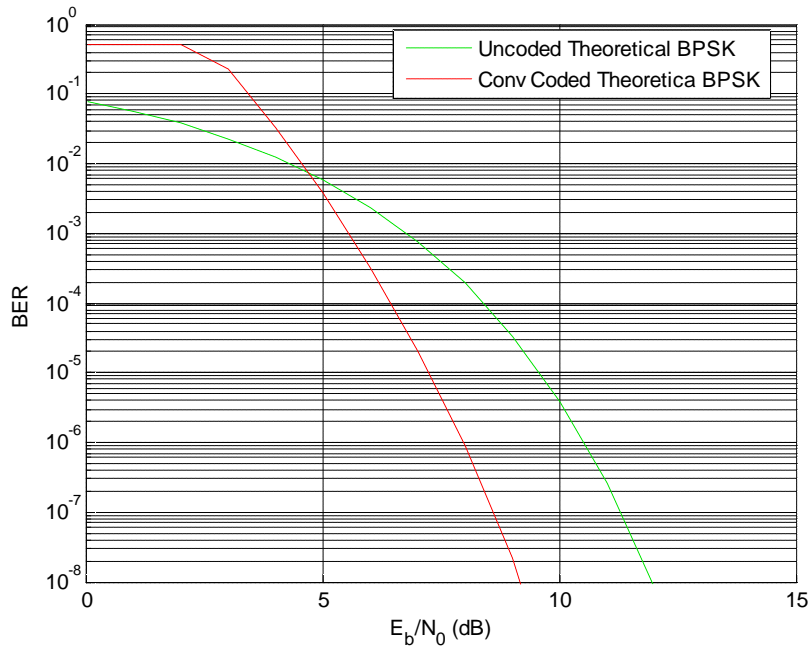


Figure 4.2 Plot of BER Versus  $E_b/N_o$  for BPSK Downlink

**Convolution code** is selected on channel coding. An optimum encoding rate applicable to a BPSK scheme is input bit  $k=1$  and output bit  $n=2$ .

A  $\frac{1}{2}$ -rate encoder is specified using poly2trellis (for more details refer to MATLAB literature) as follows:

Channel coding: **Convolution; poly2trellis(5, [35 31])**

Meaning constraint length = 5, (total number of bits in the encoder at a time i.e. four register plus the current one bit input).

The MATLAB software requires only the polynomial to implement the coding. By clicking **plot** tab on bertool window, the simulation for the coded BPSK scheme is run and graph of BER versus  $E_b/N_o$  is superimposed on plot of the uncoded BPSK, as shown in figure 4.2

#### **b) Simulation of QPSK Link**

**Theoretical** tab is selected and QPSK link parameters are entered on the BERTool window as follows:

- $E_b/N_o$  range: **0: 18**
- Channel type: **AWGN**
- Modulation type: **PSK**
- Modulation order: **4**
- Demodulation model : **Hard**
- Channel coding: **None**
- The uncoded link is simulated and graph of **BER** Versus  $E_b/N_o$  is plotted as shown in figure 4.3

**Convolution code** is now selected on Channel coding,

The optimum encoder rate for QPSK is also  $\frac{1}{2}$  rate (since QPSK requires 2 bits input to run). The same polynomial is as in (a) above:

Channel coding: **Convolution code**

Poly2trellis: **(5, [35 31])**

The coded QPSK downlink is simulated and graph of **BER** Versus  $E_b/N_o$  is superimposed on plot of the uncoded link by bertool as shown in figure 4.3.

It is important to note the similarity in the plots of BPSK and QPSK both the uncoded and coded, this was derived in chapter two on probabilities of error.

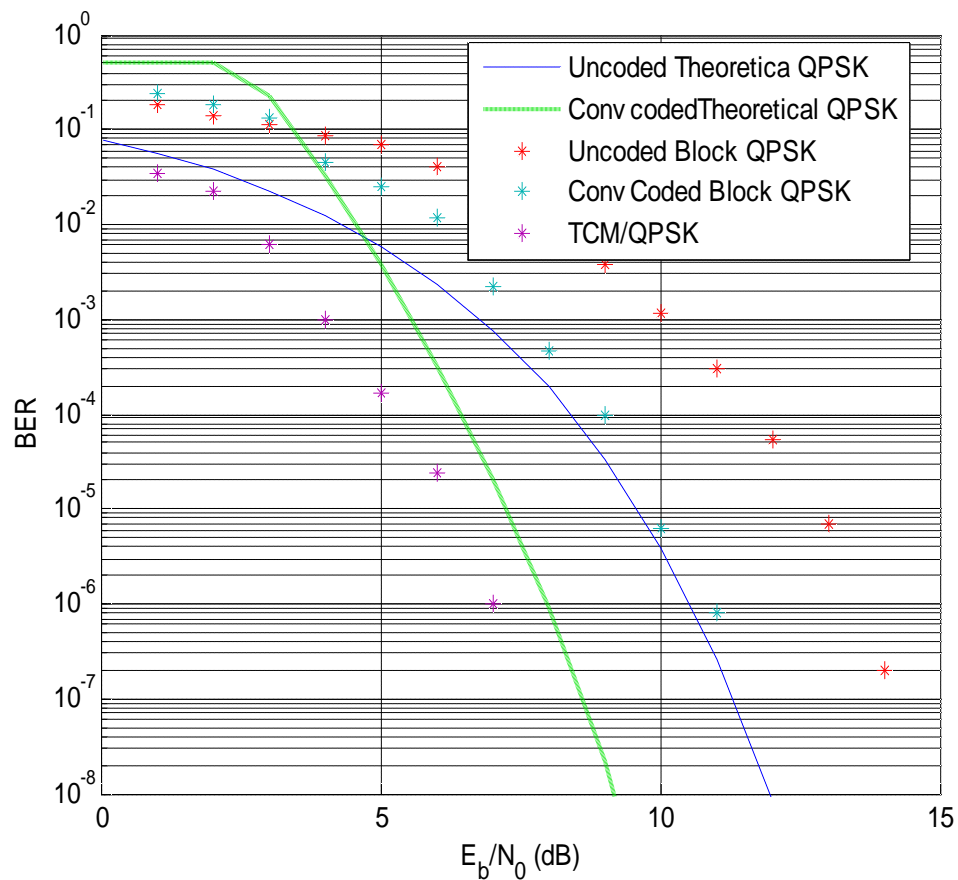


Figure 4.3 Plot of BER Versus  $E_b/N_0$  for QPSK Downlink

### c) Simulation 8-PSK link

Link parameters on 8-PSK are entered in the bertool window as follows:

- $E_b/N_0$  range: **0: 18**
- Channel type: **AWGN**
- Modulation type: **PSK**
- Demodulation mode: **Hard**
- Modulation order: **8**
- Channel coding: **None**

The uncoded link is simulated and graph of **BER** versus  $E_b/N_0$  is plotted by bertool as shown in figure 4.4.

On the bertool window, **convolution code** is selected under channel coding tab as :

Channel coding: **Convolution**

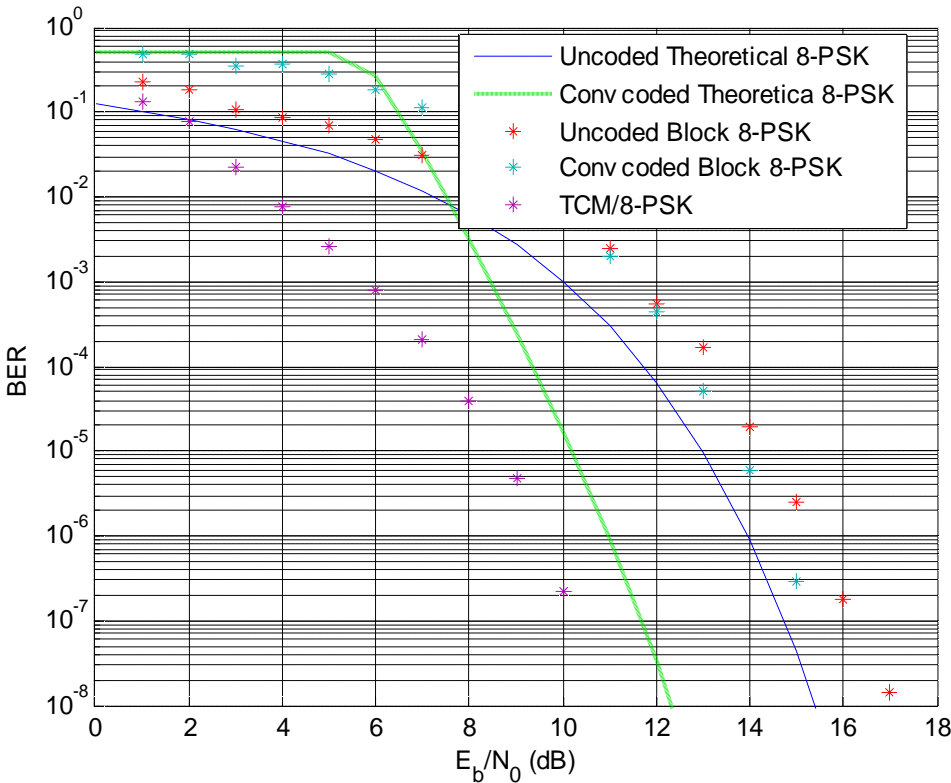


Figure 4.4 Plot of BER versus  $E_b/N_0$  for 8-PSK Downlink

A 2/3-rate encoder is selected as optimum rate to feed the 3-bit 8PSK modulator. MATLAB trellis structure for such code is given as:

```
Poly2trellis([3 2], [7 4 0; 0 2 3])
```

In this case the Constraint length of the encoder from two input lines is  $3+ 2=5$ , meaning that on the first input line there are 2 bits in two registers plus current bit and on the second input line 1bit in one register plus current bit.

The code generators are specified in octal using a 3x2 matrix. The two rows on the matrix specifies connections from each input lines, while the columns specifies connections to three modulo-2 adder that produces 3 bits output.

$$\begin{bmatrix} 7 & 4 & 0 \\ 0 & 2 & 3 \end{bmatrix}$$



Thus code generators are  $g_1 = [7,0] = [111, 00]$ ;  $g_2 = [4,2] = [100,10]$  and  $g_3 = [0,3] = [000, 11]$ .

The link is simulated and graph of **BER** Versus  $E_b/N_o$  is superimposed on plot of the uncoded 8-PSK by bertool as shown in figure 4.4.

**d) Simulation of 16-PSK link**

Link parameters on 16-PSK scheme are entered in the bertool window for theoretical simulation as follows:

- $E_b/N_o$  range: **0:22**
- Channel type: **AWGN**
- Modulation type: **PSK**
- Modulation order: **16**
- Demodulation model: **Hard**

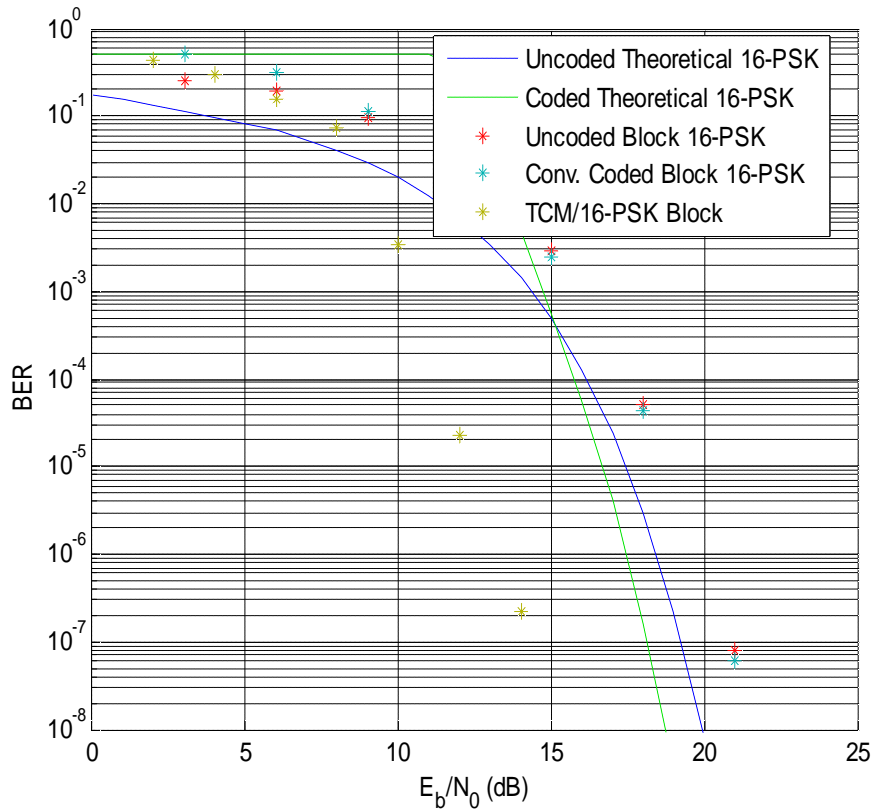


Figure 4.5 Plot of BER versus  $E_b/N_o$  for 16-PSK Downlink

The uncoded link is simulated and graph of **BER** Versus  $E_b/N_o$  is plotted by bertool as shown in figure 4.5.

On the bertool window, channel coding is selected and convolution code is entered. A  $\frac{3}{4}$  rate encoder is selected as optimum to feed 4-bit 16-PSK modulator. The MATLAB trellis structure of the code is as follows:

Channel coding: **Convolution**

Poly2trellis([**2 3 2**], [**3 2 0 0**; **2 6 0 1**; **0 0 3 2**])

The code generators which specifies contribution of each line input (in octal) to the four modulo-2 adders on the output lines are in 4x3 matrix.

$$\begin{bmatrix} 3 & 2 & 0 & 0 \\ 2 & 6 & 0 & 1 \\ 0 & 0 & 3 & 2 \end{bmatrix}$$

$g_1 = [3,2,0] = [11,010, 00]$ ;  $g_2 = [2,6,0] = [10,110,00]$ ;  $g_3 = [0,0,3] = [00,00,11]$  and  $g_4 = [0,1,2] = [00,001,10]$

Here the constraint length of the encoder is  $[2+ 3+ 2]$ , meaning that there is 1bit in a register plus current input bit on the first input line, 2 bits in two registers plus current bit on second input line and 1bit in a register plus current bit on the third input line.

The coded link is simulated and the graph of **BER** Versus  $E_b/N_o$  is superimposed on plot of the uncoded 16-PSK as shown in figure 4.5.

#### e) Simulation of 32-PSK link

Link parameters on 32-PSK scheme are entered in the bertool window for theoretical simulation as follows:

- $E_b/N_o$  range: **0:22**
- Channel type: **AWGN**
- Modulation type: **PSK**
- Modulation order: **32**
- Demodulation mode: **Hard**

The uncoded link is simulated and graph of **BER** Versus  $E_b/N_o$  is plotted by bertool as shown in figure 4.6

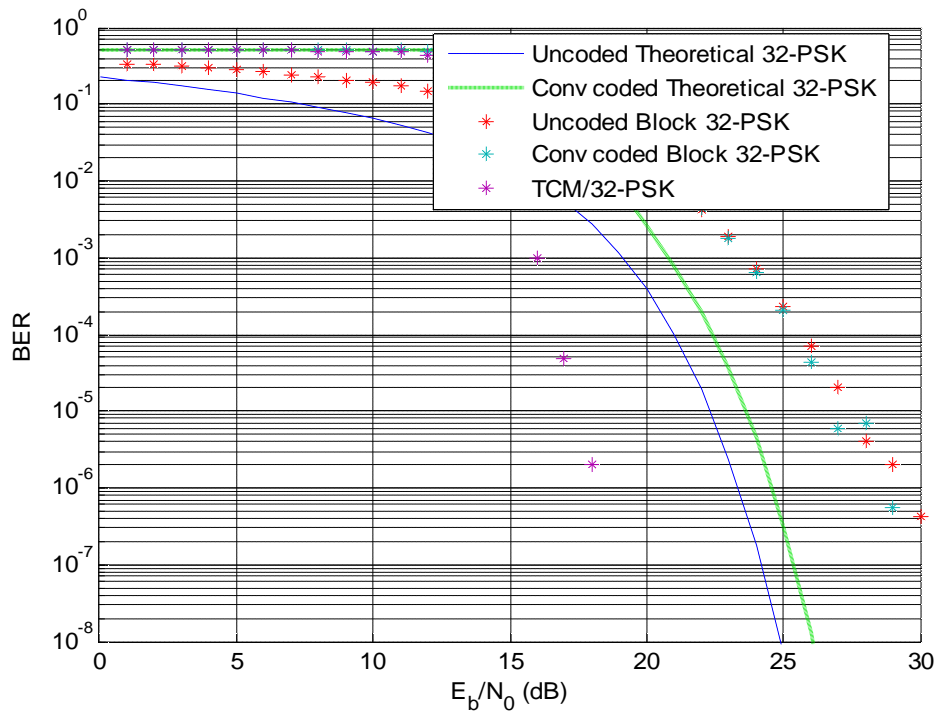


Figure 4.6 Plots of BER versus  $E_b/N_0$  for 32-PSK

On the bertool windows, channel coding is selected and convolution code is entered. A 4/5 rate encoder is selected as optimum to feed 5-bits 32-PSK modulator. The MATLAB trellis structure of the code is as follows:

Channel coding: **Convolution**

`poly2trellis([3 2 3 2], [5 0 2 0 1; 0 3 0 1 1; 1 0 5 0 1; 0 2 0 3 1])`

The code generators that specifies contribution of each input line to each modulo-2 adder on the 5 output lines are in 5x4 matrix

$$\begin{bmatrix} 5 & 0 & 2 & 0 & 1 \\ 0 & 3 & 0 & 1 & 1 \\ 1 & 0 & 5 & 0 & 1 \\ 0 & 2 & 0 & 3 & 1 \end{bmatrix}$$

$g_1=[5,0,1,0]=[101,00,001,00]$ ;  $g_2=[0,3,0,2]=[000,11,000,10]$ ;  $g_3=[2,0,5,0]$   
 $= [010,00,101,00]$ ;  $g_4=[0,1,0,3]=[000,01,000,11]$  and  $g_5$   
 $= [1,1,1,1]=[001,01,001,01]$ .

Here the constraint length is  $[3+ 2+ 3+ 2] = 10$ , meaning that there are 2 bits in two registers plus current input bit on the first input line, 1 bit in a register plus current input bit on the second input line, 2 bits in two registers plus current input bit on the third input line and 1 bit in a register plus current input bit on the fourth input line.

Channel coding is re-selected and convolution coding is entered. The coded link is simulated and the graph of **BER** Versus  $E_b/N_o$  is superimposed on plot of the uncoded 32-PSK as shown in figure 4.6.

#### **4.3 MATLAB Downlink Template Model**

An RF satellite downlink block model shown in figure 4.7 is selected from MATLAB link samples and adopted as template for building downlink block models in this research work. The model comprises of three sections, on-board transmitting section, downlink path and Ground station receiving section.

The transmitting section is made up of: Random integer generator used as data source for the link; Rectangular 16-QAM as digital modulator; square root raised cosine (RRC) filter as transmit filter; Saleh model (see APENDIX B) amplifier as high power amplifier (HPA); antenna gain block as transmitter antenna.

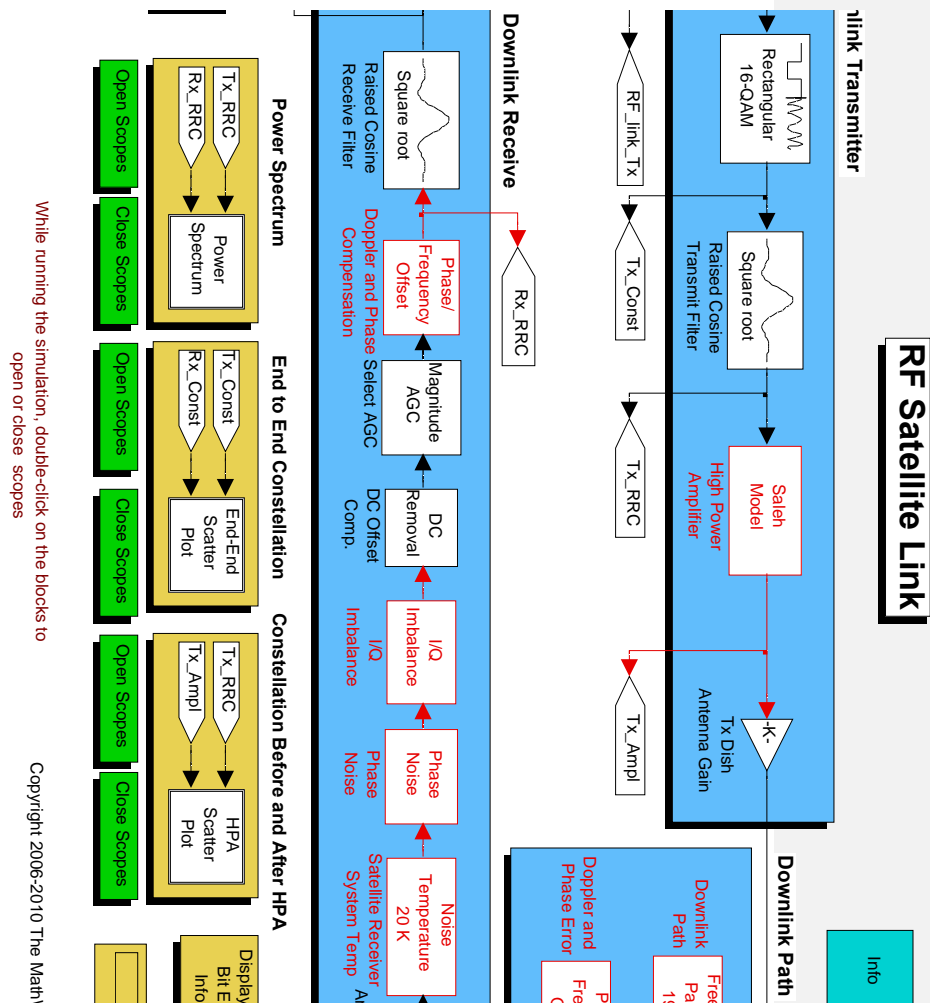


Figure 4.07 MATLAB RF Satellite Downlink Template Model

The down path or channel section is made up of: Free space loss, Doppler effect and Phase error blocks. The receiving section is made up of antenna gain block as receiving antenna; satellite receiver system noise temperature block as equivalent noise temperature of the receiving system; other impairment blocks are phase noise block, in-phase and quadrature imbalance block, DC removal block, AGC block, phase offset block to compensate for Doppler and phase noise; Square Root Raised Cosine Filter block as receiver filter and rectangular 16-QAM demodulator as discriminator.

Some measurements blocks used on the model are: Power spectrum block to measure signal bandwidth; Scatter plot block to display signal constellations and a Communication sink Display block to count the Total number of demodulated bits, Number of received errors and compute Bit Error Rate (BER).

The simulation in this work is on Bit error rate (BER) performance against received SNR or  $E_b/N_0$  and bandwidth utilization of MPSK downlink. The template model is modified to focus and measure the effect of coding gain on SNR or  $E_b/N_0$  and bandwidth utilization on data rate.

#### **4.4 Building MATLAB Downlink Block Model**

To build downlink block model, functional unit blocks are obtained from Communication blockset library of the MATLAB, while blocks for measurement and display are obtained from Simulink and Signal Analysis blockset Libraries. The modification on the template is shown in figure 4.8 and realized as follows:

In the Transmitter Section: A Random Integer Generator block, in conjunction with an Integer-to-Bit Converter block are used to form a Data Source. An MPSK Baseband Modulator block is used as a Modulator block and a Transmit Square Root Raised Cosine (RRC) Filter block is used for pulse shaping. An AWGN block is used as a channel for transmission (the downlink channel has been characterized as an AWGN channel in Chapter Two). In MATLAB received SNR or  $E_b/N_0$  can be specified among the parameters in the AWGN block, this means for our simulation purpose the transmitter HPA, transmit/receive

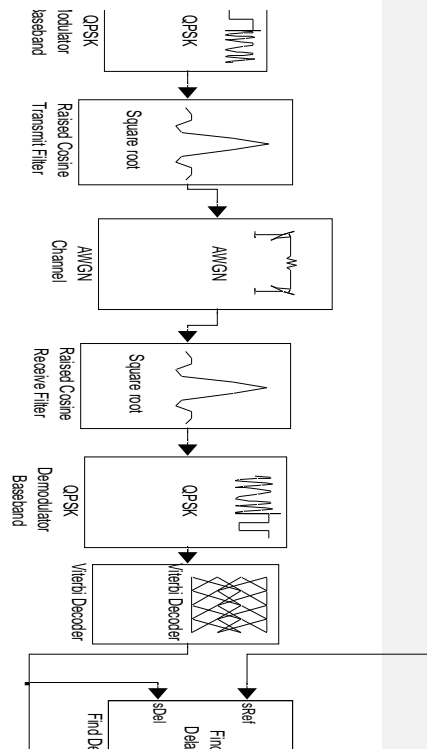


Figure 4.8A Convolution Coded QPSK Downlink Block Model

antenna gains and all other transmission losses are combined by specifying the delivered SNR or  $E_b/N_0$  at the receiver terminal. In the receiver section: Receive Square Root Raised Cosine Filter block is used to complement the Transmit RRC filter (see appendix A). MPSK Baseband Demodulator block is used as Demodulator.

Some measuring blocks used are: Find Delay block to measure the delay between received and transmitted signal bits this is necessary in counting bit error received. Error Calculation block is used to measure total number receive bits i.e. **Data Throughput, Number of Received Errors** and computation of **Bit Error Rate (BER)**, these measured parameters are sent to a Display Unit block and a Signal to Workspace block for Monte Carlo bertool simulation.

The parameter settings on each block unit are given below using a QPSK downlink as example.[13]

1.Data Source Block:

- Random Integer Generator Block set **M=4**.
  - Sampling Time  **$T_s=0.2\mu s$**
  - Integer to Bits Converter Block specify Number bits per input integer (symbol) **k=2**
2. Baseband Modulation Block:
- MPSK; **M = 4**
  - Constellation order = (Binary/Gray)
  - Input Data type: Binary bits
3. Square Root Raised Cosine (RRC) Transmit Filter block:
- Upsampling rate ; **N = 2**(minimum sampling rate)
  - Roll . off factor;  **$\alpha = 0.1$** (range: 0-1)
  - Group delay =3
4. AWGN Channel block: The AWGN block is used as means of specifying Received Signal Power to Noise Ratio in three modes: Signal Power to Noise Power Ratio **SNR**; Energy per Symbol to Spectral Noise Power density ratio  **$E_s/N_o$** ; and Energy per bit to Spectral Noise Power density ratio  **$E_b/N_o$** .
- Mode (of specifying power);  **$E_b/N_o$** (dB)
  - Input Signal power( watts) referenced to  $1\hat{o} = 1/ N = 0.5$  [13]
  - Number of bits per symbol: **n= 2** (n =k for uncoded)
  - Transmission Symbol Period  **$T_{sym} = T_s N = 0.2 \times 2 = 0.4\mu s$**
5. Square Root Raised Cosine Receive Filter Block:
- Input Sample per Symbol; **N = 2**
  - Group delay = 3
  - Roll-off factor:  **$\alpha = 0.1$**
  - Downsampling factor **L = 2**
6. Baseband Demodulator Block:
- Phase offset (constellation phase offset = pi/4 for QPSK)
  - Constellation ordering; **Binary/Gray**
  - Output type; **Bits**
  - Decision mode; **Hard/Soft** : Hard for Bits and Soft for Integer output type
7. Find Delay Block:
- Correlation window length (Number of sample) = 200  
Delay output is sent to a Display block.



8. Error Rate Calculation Block: Compares the received data with data transmitted to compute error between the two, the transmitted data is delayed for proper alignment. The parameters to set are:

- Receive Delay = Output of Find Delay block on Display 2
- Computation Delay = No. of samples to be delayed ( set to ignore transient effect at beginning of comparison)

These outputs send to:

i) Display Block:

Shows

- Number of Comparison = Number received bits
- Number of Errors from the comparison;
- Bit Error Rate = Number of received bits/Number of Errors

ii) MATLAB workspace with BER as variable for Monte Carlo simulation.

#### 4.4.1 Settings for Monte Carlo Simulation

To simulate block models using Monte Carlo method, the following steps are applied:

a) On the block model built as described in section 4.4 and the following changes are made:

- On Modulator/Demodulator Blocks set  $M = 2^k$  for uncoded and  $M = 2^n$  for coded (k/n-rate) model.
- In the Modulator/Demodulator block, set input/output type to **bits**.
- In the AWGN block, the set parameter  $E_b/N_0$  to **EbNo**.
- In the Error Calculation block as part of the iteration process, set the Maximum Number of Symbol to stop at **maxNumBits** and Target Number of Errors to stop at **maxNumErrs**.
- In the Signal to Workspace block connected to send results to Monte Carlo, the Variable name is set as **BER**.

The block model with its settings is saved as a File.

b) On the MATLAB execute command windows the iteration values are specified as:

**EbNo = 0; maxNumErrs = 100; maxNum Bits = 1e8;**

This command indicates that iterative simulation starts at  $E_b/N_0 = 0$  and stops each time number of errors = 100 or number of received bits =  $10^8$ .

c) The **BERTool** window is opened and **Monte Carlo** tab is selected.

The simulation parameters are entered as follows:

- $E_b/N_0$  range: **Start : Iteration step : Stop** values

- Simulation MATLAB Filename of block model: '.... '
- Variable name:**BER**.

The following MPSK block models are simulated using Monte Carlo as specified above.

#### 4.5 Simulation of MPSK Downlink Block Models

##### a) QPSK Downlink Block Model

An uncoded QPSK downlink block model is built as shown in figure 4.12 without the Encoder and Decoder blocks, the parameter settings are as follows:

- In Random Integer Generator block; set Integer **M = 4** to generate Integer (0,1,2and3).
- In Integer to Bit Converter Block specify Number bits per input integer (symbol) **k=2**.
- In **QPSK** Baseband Modulator/Demodulator block; set Input/Output type to bits and constellation set to binary.
- In AWGN Channel Block; set Number of bits per symbol = **2** and  $E_b/N_o = \mathbf{EbNo}$ .
- In Error Calculation block; set Receive Delay = **14** (measured from Find Delay block on test run simulation of the model) and Stop simulation set to = **maxNum Errs** or **maxNumBits**.
- In Signal to Workspace block; set Variable name to **BER**

The model is saved with name qpsk1.

The **BERTool** window is opened and **Monte Carlo** tab is selected. The simulation parameters are entered as follows:

- $E_b/N_o$  range:**0: 2: 18**
- Simulation MATLAB File : **qpsk1**
- Variable name:**BER**

The simulation is run and its result is a graph of BER Versus  $E_b/N_o$  superimposed on that of Theoretical QPSK as shown in figure 4.4 Convolution Encoder and Decoder blocks are added to the model before the modulator and after demodulator blocks respectively, with same coding structure as used on Theoretical link in section 4.2(b), i.e.

- In Encoder/Decoder blocks Polynomial set at: **poly2trellis(5, [35 31])**

The coded model is test run and new Receive delay is measured and adjusted

- In Error Calculation block; set Receive delay=**41**(measured)

The coded QPSK model formed is saved with filename'qpsklink2', Monte Carlo simulation run on it. Its BER performance plot is also superimposed on the plots of figure 4.3 and the constellation of modulated signal envelope is shown in figure 4.9. The dispersion of signal points on each symbol in the constellation is measure of noise level taken at  $E_b/N_0=10.7\text{dB}$ . It is the ability of QPSK demodulator to discriminate each signal point correctly and the convolution decoder to correct any error that is measured as BER of the link.

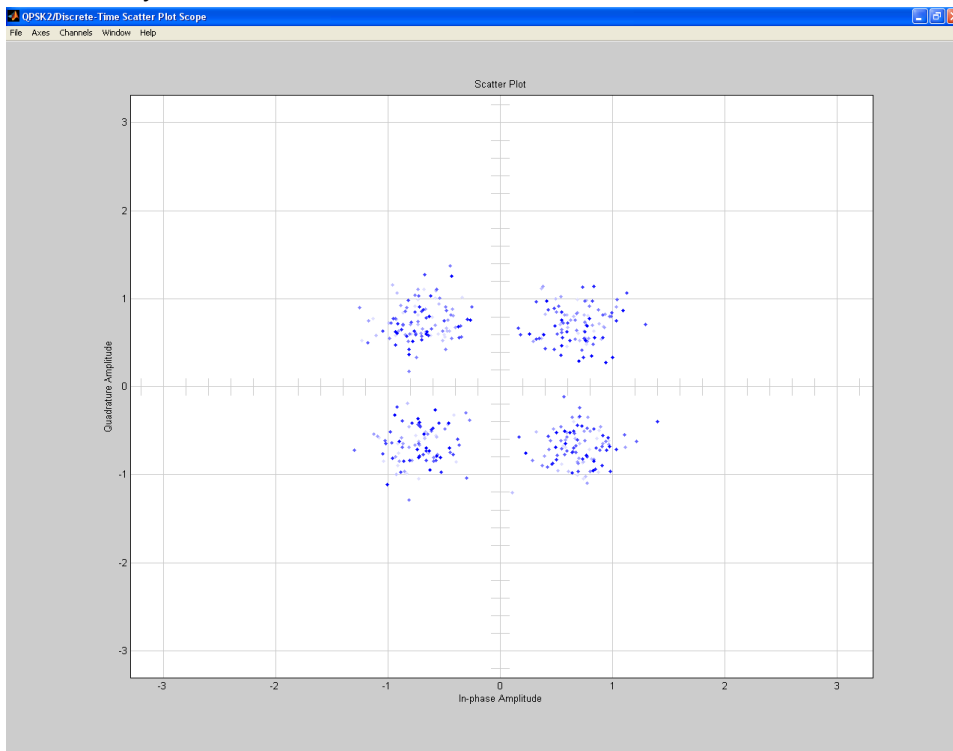


Figure 4.9 Convolution Coded QPSK Constellation at  $E_b/N_0=10.7\text{ dB}$

### (b) 8-PSK Block Model

An 8-PSK block model is built with the following changes in some affected blocks:

- **M = 8**
- Number of bits per symbol **k = 3**

- 8-PSK Baseband **Modulator/Demodulator**.  
Input/Output:**Bits** and Constellation:**Binary**
- AWGN Channel Block; Number of bits per symbol = 3;  
 $E_b/N_o = \mathbf{EbNo}$
- Receive Delay = **18**(measured)

The uncoded model is saved as 'psk8link1' as in (a) above.

**BERTool** window is opened and **Monte Carlo** tab is selected.

Parameters for simulation are entered as follows:

- $E_b/N_o$  range:**0: 2: 18**
- Simulation MATLAB File : **psk8link1**
- Variable name:**BER**

Monte Carlo simulation is run on the uncoded 8-PSK model and its BER performance plot is shown in figure 4.5.

Convolution encoder/ Viterbi decoder blocks of 2/3 rate are added, with same MATLAB Trellis structure as in section 4.2(c) and the following necessary changes are made:

- Random Integer Generator **M=4** (To match the 2-bit encoder input)
- Encoder/Decoder blocks: poly2trellis( **[3 2], [7 4 0; 0 2 3]**)
- Error Calculation Block: Receive Delay =**62** (Measured)

The coded model formed is saved as 'psk8link2' and Monte Carlo simulation run on it. Its performance plot is also superimposed on the plots of figure 4.4 and signal constellation of 8-PSK link is shown in figure 4.10.

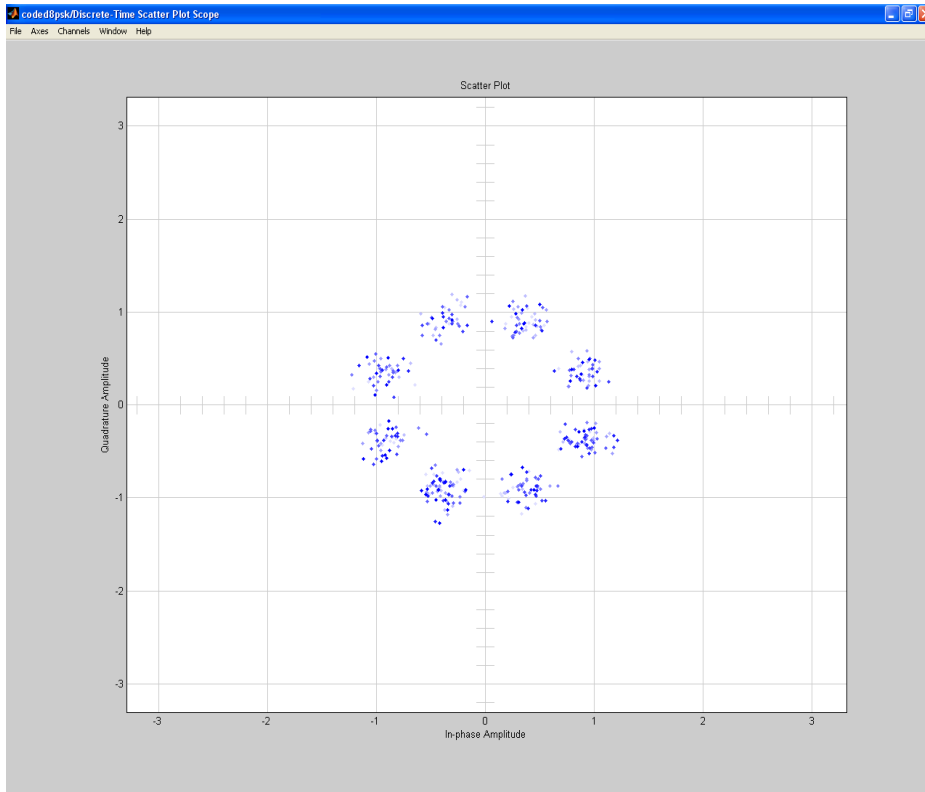


Figure 4.10 Coded Modulated 8-PSK Constellation at  $E_b/N_0=14.5\text{dB}$

The dispersion of signal points on each symbol in the constellation is a measure of noise level taken at  $E_b/N_0=14.5\text{dB}$ . It is the ability of 8-PSK demodulator to discriminate each signal point correctly and the convolution decoder to correct any error that is measured as BER of the link.

### (c) 16-PSK Block Model

An uncoded 16-PSK model is built with following parameter settings:

- Random Integer Generator Block: **M = 16**
- In Integer to Binary Converter Block: **k = 4**
- **16-PSK** Modulator/Demodulator:  
Input/Output: **Bits** and Constellation: **Binary**;
- AWGN Channel Block: Bits per symbol = **4** and  $E_b/N_0 = E_b N_0$
- Error Calculation Block: Receive Delay = **44**(measured)

Stop simulation set at **maxNumErrs** or **maxNumBits**.

- In Signal to Workspace block; set Variable name to **BER**.
- The uncoded model is saved as 'psk16link1'.

**BERTool** window is opened and **Monte Carlo** tab is selected. The simulation parameters are entered as follows:

- $E_b/N_0$  range: **0: 2: 18**
- Simulation MATLAB File : **psk16link1**
- Variable name: **BER**

Monte Carlo simulation is run on the uncoded 16-PSK block model and its BER plot is shown in figure 4.6.

Convolution encoder/ decoder blocks are added, with same MATLAB Trellis structure as the Theoretical 16-PSK link in section 4.3(d) and the following necessary changes are made:

- Random Integer Generator Block: **M = 8** ( for 3/4-rate encoder )
- Integer to Bit Converter Block: **k= 3**
- Encoder/Decoder Blocks:  
Poly2trellis([**2 3 2**],[**3 2 0 0; 2 6 0 1; 0 0 3 2**]).
- Error Calculation Block: Receive Delay= **63**(Measured)

Stop simulation set at **maxNumErrs** or **maxNumBits**.

- The coded model is saved as 'psk16link2'

The simulation is run on the model and BER performance plots is superimposed on that of Theoretical uncoded and coded 16-PSK as shown in figure 4.5. Its Signal constellation is shown in fig 4.12.

The dispersion of signal points on each symbol in the constellation is measure of noise level taken at  $E_b/N_0 = 19.5\text{dB}$ . It is the ability of 16-PSK demodulator to discriminate each signal point correctly and the convolution decoder to correct any error that is measured as BER of the link.

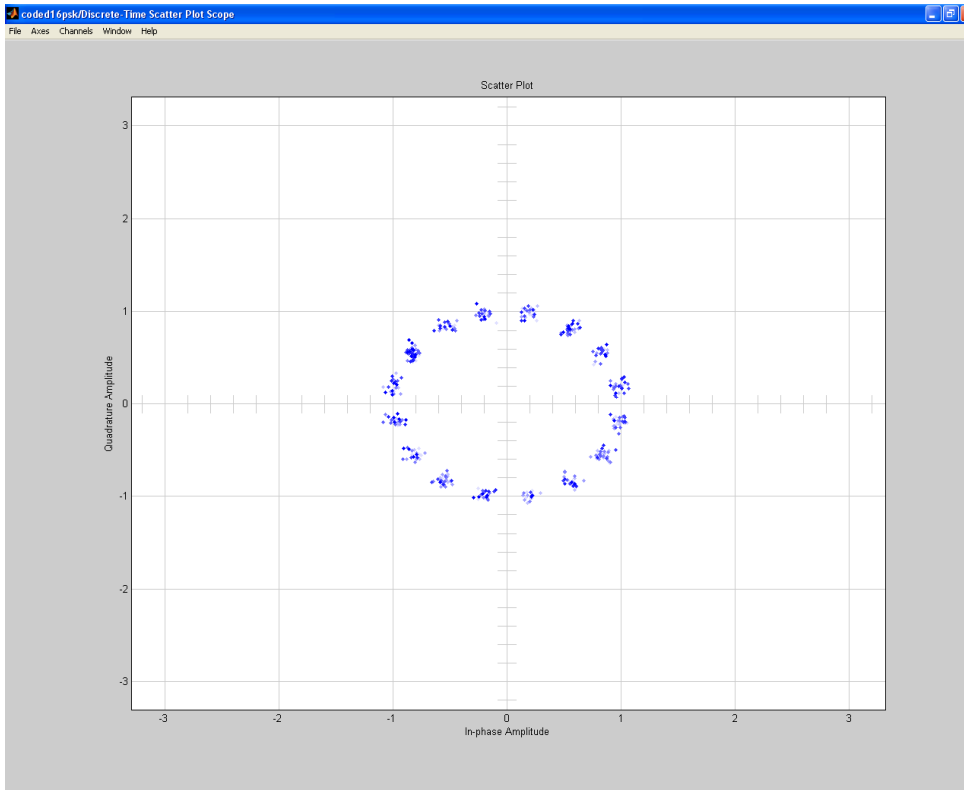


Figure 4.11 Convolution Coded 16-PSK Constellation at  $E_b/N_o = 19.5\text{dB}$

#### (d) 32-PSK Block Model

An uncoded 32-PSK block model is built with parameters settings of:

- Random Integer Generator Block: **M = 32**
- In Integer to Binary Converter Block: **k = 5**
- **32-PSK** Modulator/Demodulator Block  
Input/output type: **Bits** and Constellation: **Binary**
- AWGN Channel Block: Bits per symbol = **5**;  $E_b/N_o = \mathbf{EbNo}$
- Error Calculation Block : Receive Delay = **30** (measured)

Stop simulation: **maxNumErrs** or **maxNumBits**

- Signal to Workspace Block; Variable name : **BER**

**Monte Carlo** tab is selected. Its simulation parameters are :

- $E_b/N_o$  range: **0: 2: 18**
- Simulation MATLAB File : **psk8link1**
- Variable name: **BER**

Monte Carlo simulation is run on the uncoded 32-PSK model and its BER performance plot is shown in figure 4.7).

Convolution encoder/ decoder blocks of 4/5 are added, with same MATLAB Trellis structure as link in section 4.2(e) and the following necessary changes are made:

- Random Integer Generator Block: **M = 16**(to match 4-bit input of the encoder )
- Integer to Bit Converter Block: Number of bit/integer = **4**
- Encoder/Decoder Blocks:

poly2trellis(**[3 2 3 2]**,**[5 0 2 0 1; 0 3 0 1 1; 1 0 5 0 1; 0 2 0 3 1]**)

- Error Calculation Block: Receive Delay= **84** (Measured)

Stop simulation set at **maxNumErrors** or **maxNumBits**

The coded model is saved as 'psk32link2' and Monte Carlo simulation is run on it. Its BER performance plot is superimposed on figure 4.6. Modulated signal constellation of 32 PSK link is shown in figure 4.12. The dispersion of signal points on each symbol in the constellation is measure of noise level taken at  $E_b/N_0 = 28.5\text{dB}$ . It is the ability of 32-PSK demodulator to discriminate each signal point correctly and the convolution decoder to correct any error that is measured as BER of the link.



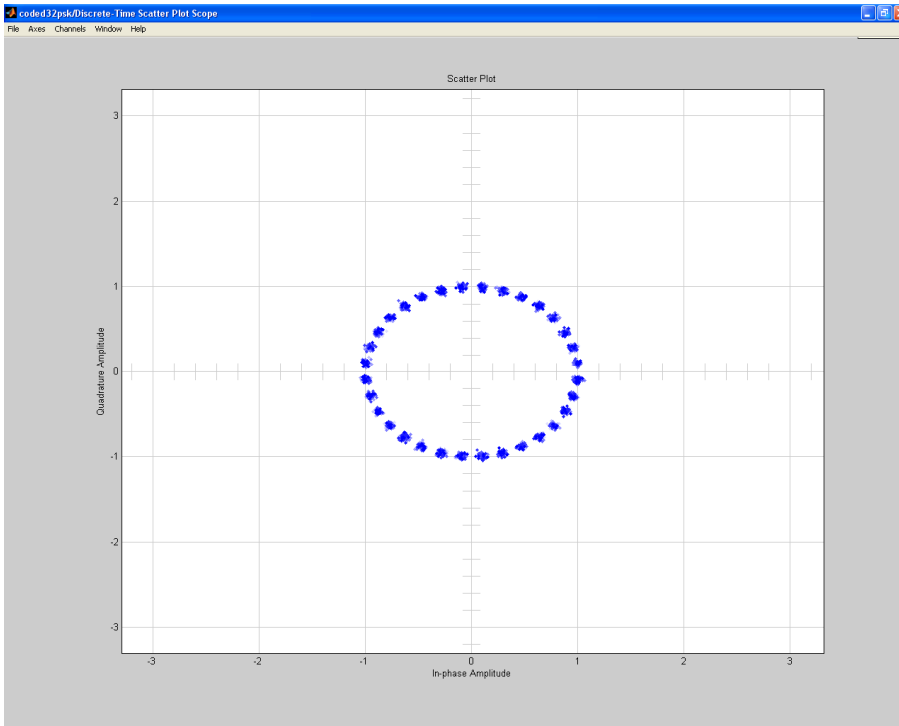


Figure 4.12 Convolution Coded 32-PSK constellation  $E_b/N_o$  28.5dB

Summary of  $E_b/N_o$  performances at BER of  $10^{-6}$  for the uncoded and convolution coded and TCM on MPSK downlink block models are shown in table 4.1.

Table 4.1 Performance Summary of Uncoded, Convolution Coded and TCM on MPSK Downlink Block Model at BER =  $10^{-6}$

MPSK Scheme	Inputk bits	Data Rate (Mbps)	Uncoded MPSK $E_b/N_o$ (dB)	Convolution Coded MPSK $E_b/N_o$ (dB)	TCM/MPSK $E_b/N_o$ (dB)
BPSK	1	5	13.5	10.7	N/A
QPSK	2	10	13.5	10.7	7
8-PSK	3	15	15.5	14.5	9.8
16 PSK	4	20	20.4	19.5	13
32 PSK	5	25	29.5	28.5	18.2

Notethe comparative advantages onpower, the coding gain would be calculated in chapter5.

All the simulations were conducted at symbol rate of  $5 \times 10^6$  symbol per seconds, each occupied bandwidth of 5.5 MHz as shown in figure 4.13. The transmission data rates on each MPSK scheme are shown in table 4.2.



Figure 4.13 Spectrum Scope of Convolution Coded 32-PSK at 5.5 MHz

Table 4.2 Data Rate of Uncoded, Convolution Coded and TCM on MPSK Downlink Block Model MPSK at  $5 \times 10^6$  Sps and bandwidth 5.5MHz.

MPSK Scheme	Input k bits	uncoded Data Rate (Mbps)	Code Rate k/n	Data Rate of Convolution Coded and TCM	
				Coded (Mbps)	Decoded (Mbps)
BPSK	1	5	$\frac{1}{2}$	5	5
QPSK	1	10	$\frac{1}{2}$	10	5
8-PSK	2	15	$\frac{2}{3}$	15	10
16P SK	3	20	$\frac{3}{4}$	20	15
32PSK	4	25	$\frac{4}{5}$	25	20

Note the difference in data rate between uncoded and coded, and its improvement with higher MPSK.

## STAGE 2

### 4.6 Simulations of TCM/MPSK Downlink Block Model

A block unit that combine and perform function of Trellis Coded Modulation (TCM) is also available in MATLAB Communication blockset library. A Downlink Block Model is built by replacing the Convolution Encoder and MPSK modulator blocks in figure 4.3 with a single TCM/MPSK that combines the functions of the two. Similarly MPSK Demodulator and Viterbi Decoder blocks are replaced with a single TCM Decoder block. A Downlink block model on TCM/MPSK scheme is shown in figure 4.14. The Monte Carlo simulation is run on TCM/MPSK block model for BER performance evaluation and comparison with that of convolution coding/MPSK scheme as follows:

#### a) TCM/QPSK on Downlink Model

A TCM/QPSK encoder block is used to replace the Convolution encoder /QPSK modulator blocks, and a TCM/QPSK decoder block is used to replace the demodulator and Viterbi decoder blocks. The TCM encoder and decoder blocks are set with same parameters as in section 4.5(a) viz ;

Trellis coded Modulation structure = `poly2trellis(5,[35 31])`

Modulation/Demodulation order **M = 4**

Thus, a TCM/QPSK Downlink model is formed as shown in figure 4.14 and saved as 'qpsktcm'. The model is run on Monte Carlo simulation, the BER performance plot is superimposed on the plot in figure 4.3 and its signal constellation taken at  $E_b/N_0 = 7\text{dB}$  is shown in figure 4.15 (b),

#### b) TCM/8-PSK/TCM on Downlink Model

A TCM/8PSK encoder block is used to replace the convolution encoder /modulator blocks and TCM/8-PSK decoder block is used to replace the demodulator and Viterbi decoder blocks in section 4.5(b). The TCM encoder and decoder are set as in section 4.5(b):

Trellis coding structure = `poly2trellis([3 2], [7 4 0; 0 2 3])`

Modulation order **M = 8**

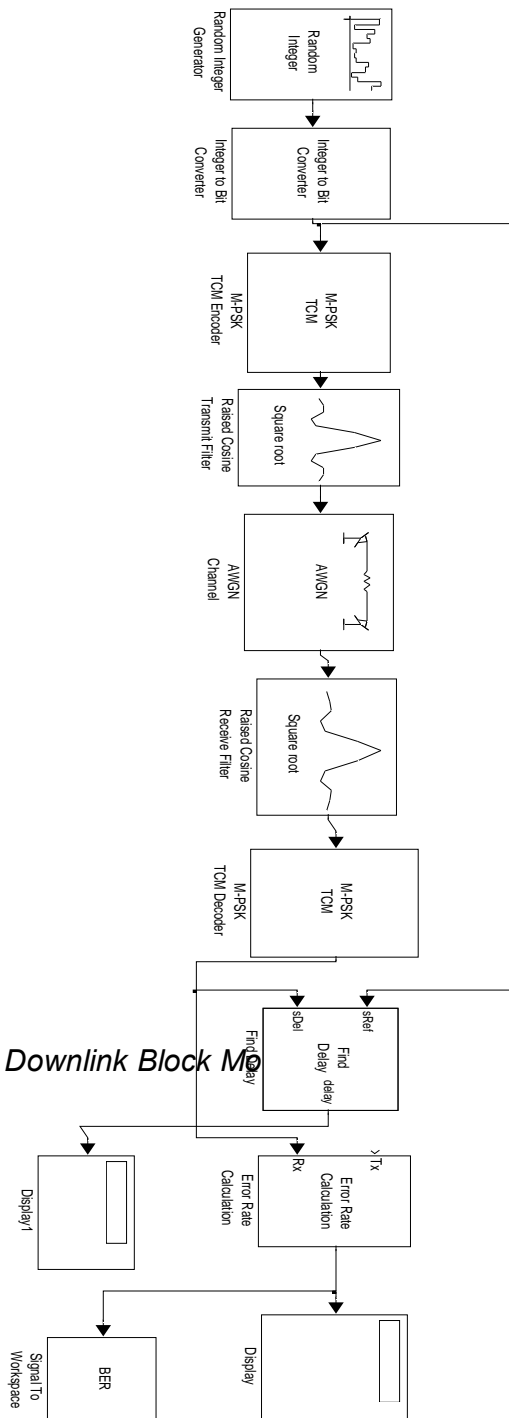
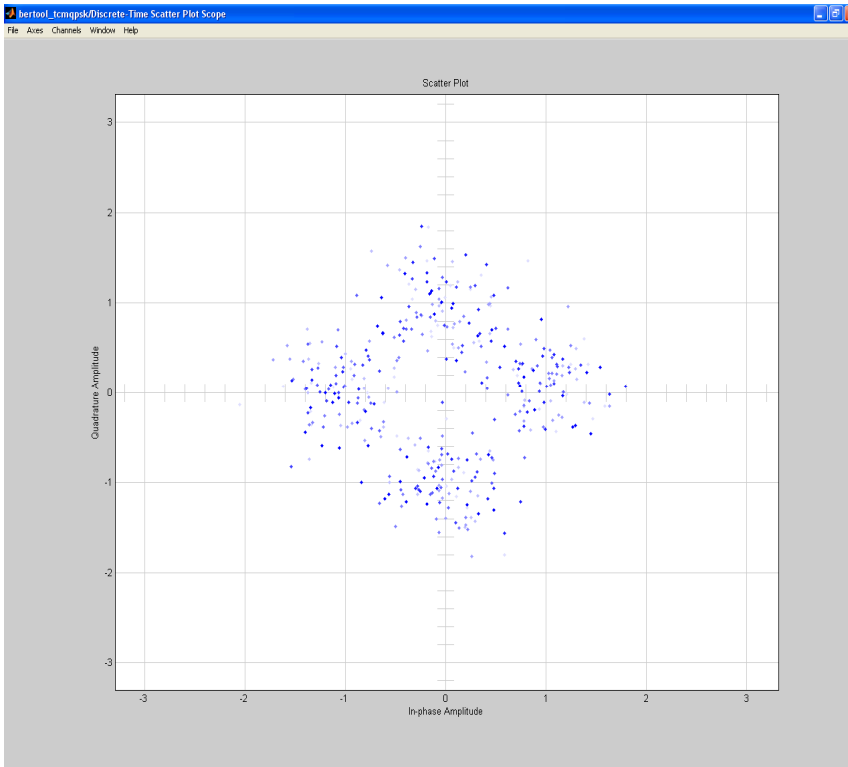


Figure 4.14 TCM/QPSK Downlink Block Model



*Figure 4.15 TCM-QPSK Constellation at  $E_b/N_0 = 7\text{dB}$*

The block model formed is saved as 'tcm8psk'. The model is run on Monte Carlo simulation, its BER plot is superimposed on plots in figure 4.4 and signal constellation is taken at  $E_b/N_0 = 9.5\text{dB}$  is shown in figure 4.16.

### c) TCM/16-PSK on Downlink Block Model

A TCM/16-PSK encoder block unit is used to replace convolution encoder and 16-PSK modulator blocks while a TCM/16-PSK decoder block unit is used to replace the demodulator and viterbi decoder blocks of section 4.5 (c). The TCM encoder and decoder blocks are set with parameters as in section 4.5 (c);

Trellis structure = poly2trellis ([2 3 2], [3 2 0 0; 2 6 0 1; 0 0 3 2])

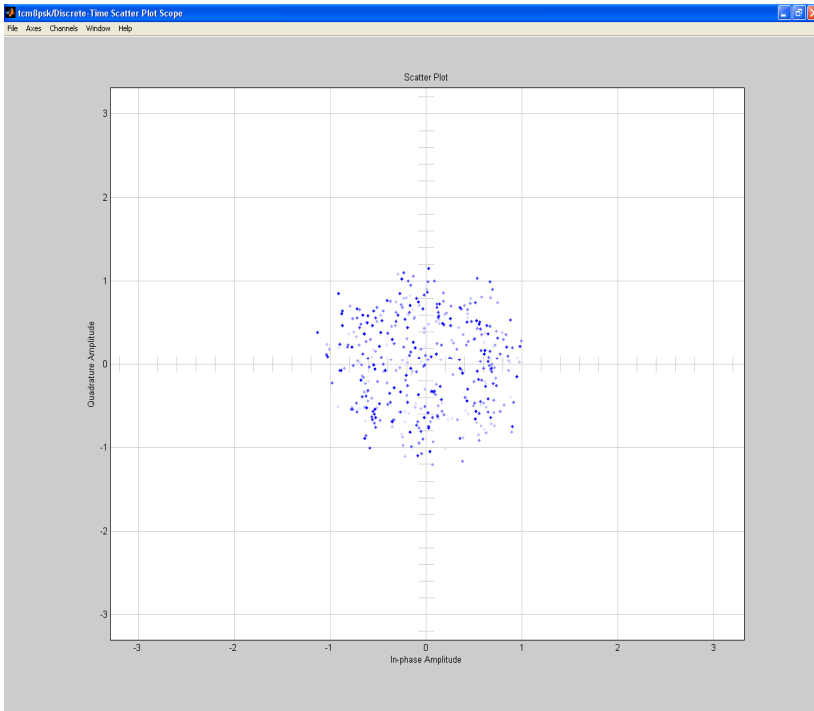


Figure 4.16 TCM/8-PSK Signal Constellation at  $E_b/N_o = 9.5\text{dB}$

Modulation order  $M = 16$

The TCM block model formed is saved as 'tcm16psk'. The model is run on Monte Carlo simulation, its BER plot is superimposed on that in figure 4.5 and signal constellation taken at  $E_b/N_o = 13.0\text{dB}$  is shown in figure 4.17

#### d) TCM/32-PSK on Downlink Block Model

A TCM/32-PSK encoder block unit is used to replace convolution encoder and a 32-PSK modulator blocks and a 32-PSK/TCM decoder block unit is used to replace the demodulator and viterbi decoder blocks of section 4.5 (d). The TCM encoder and decoder blocks are set as in section 4.5(d):

Trellis coding structure =

`poly2trellis([3 2 3 2], [5 0 2 0 1; 0 3 0 1 1; 1 0 5 0 1; 0 2 0 3 1])`

Modulation order  $M = 32$

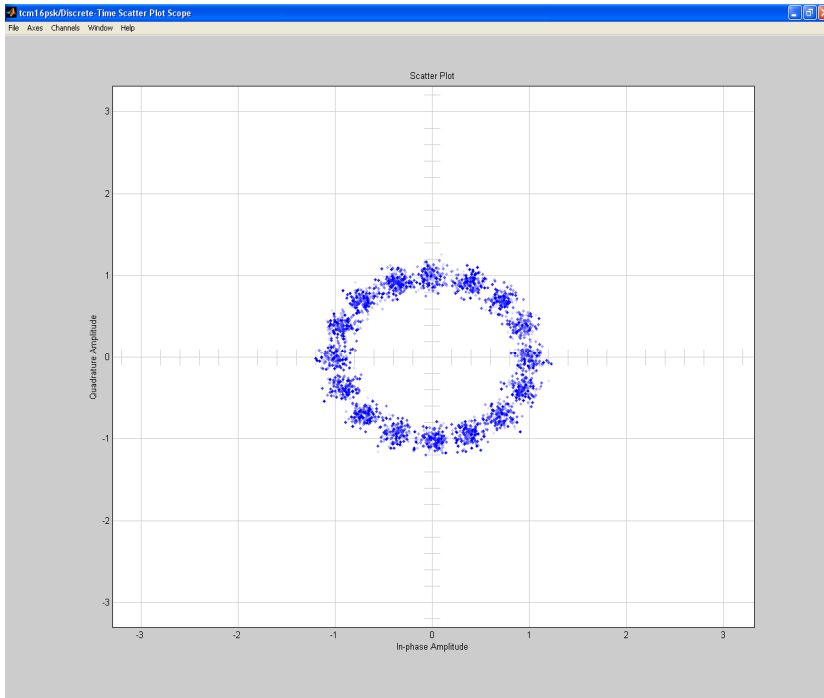


Figure 4.17 TCM/16-PSK Signal Constellation at  $E_b/N_0 = 13.0\text{dB}$

The TCM model formed is saved as 'tcm32psk'. The model is run on Monte Carlo simulation, its BER plot is superimposed on that in figure 4.6 and its signal constellation taken at  $E_b/N_0 = 18.2\text{dB}$  is shown in figure 4.18.

Summary of  $E_b/N_0$  performances at BER of  $10^{-6}$  for all the TCM/MPSK downlink block models simulated are in table 4.3

Table 4.3 Performance Summary of TCM/MPSK on Downlink Block Model at BER =  $10^{-6}$

S/N	TCM/MPSK	Simulated $E_b/N_0$ (dB)
1	TCM/QPSK	7
2	TCM/8-PSK	9.5
3	TCM/16-PSK	13.5
4	TCM/32-PSK	18.2

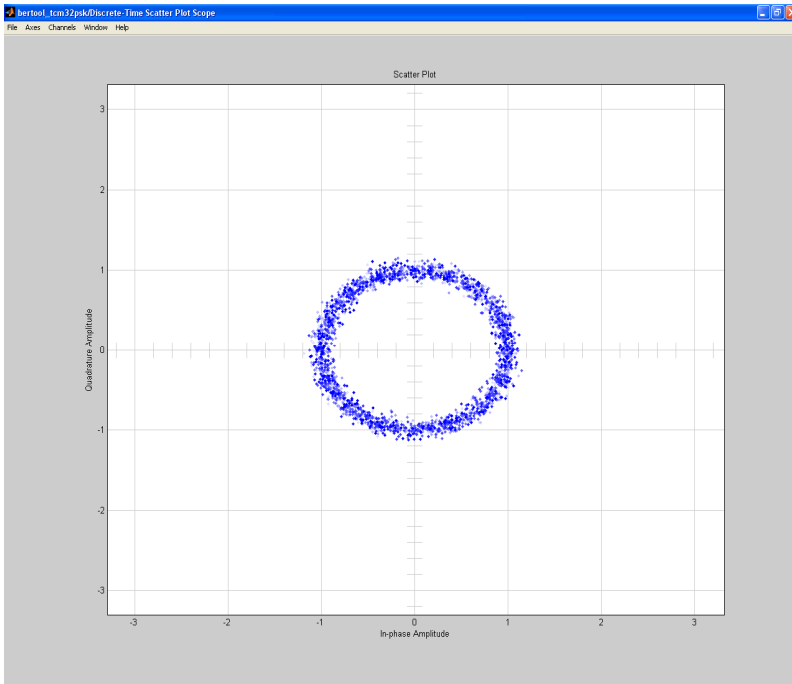


Figure 4.18 TCM/32-PSK Signal Constellation at  $E_b/N_0 = 18.2\text{dB}$

### STAGE 3

#### 4.7 Discrete Simulations of Multi-State TCM/8-PSK Encoders

The conceptual model in section 3.5.1 on TCM/8-PSK is designed to take advantage of variation of coding gain among the six encoders. The model is expected to provide coding gain variations of 3 to 5.4dB. An uncoded QPSK was simulated again and its BER plot shown in figure 4.20. Each of the six encoders was discretely simulated in a LEO Downlink model of figure 4.19 and the BER plots are superimposed on that of uncoded QPSK to measure coding gain.

##### a) 4-State Transition TCM/8-PSK

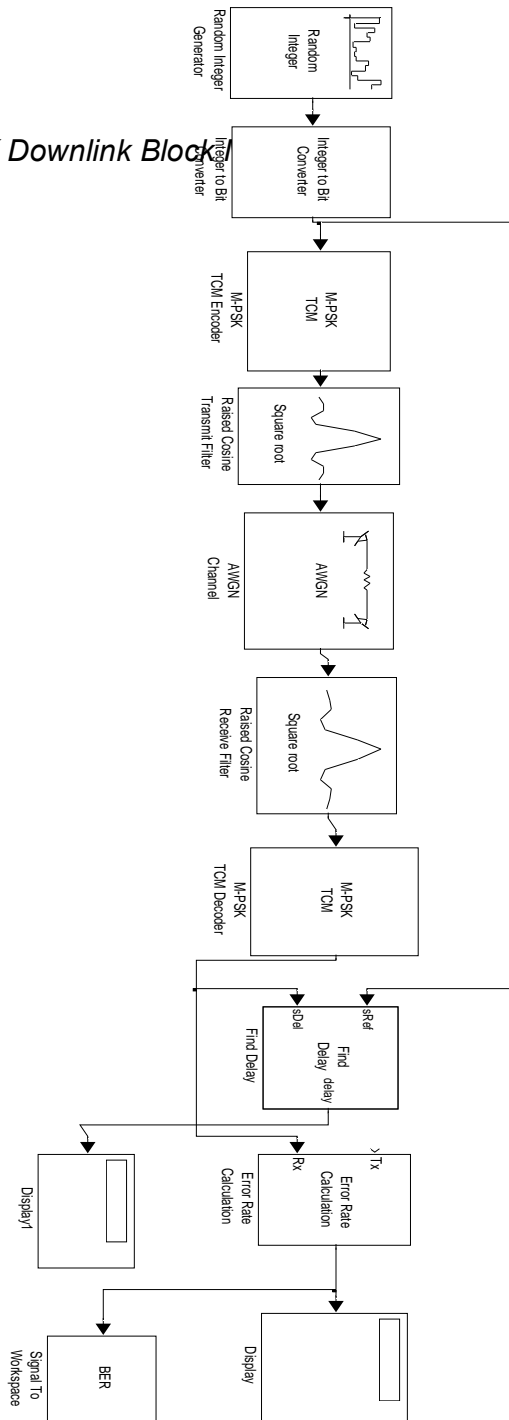
The TCM encoder/decoder blocks of figure 4.19 were set with the polynomial of 4-State Transition TCM/8-PSK obtained in section 3.10(a) as:

```
trellis = poly2trellis([3 1],[0 7 1; 1 0 0])
```

The `bertoolqsimulation` was conducted on the model and the BER plot obtained is shown in figure 4.20. From the plot at  $\text{BER} = 10^{-6}$ , the results are  $E_b/N_0 = 10.8\text{dB}$  for 4-State TCM against  $E_b/N_0 = 13.5$  for uncoded QPSK, these are recorded in table 4.4



Figure 4.19 TCM/8-PSK Downlink Block



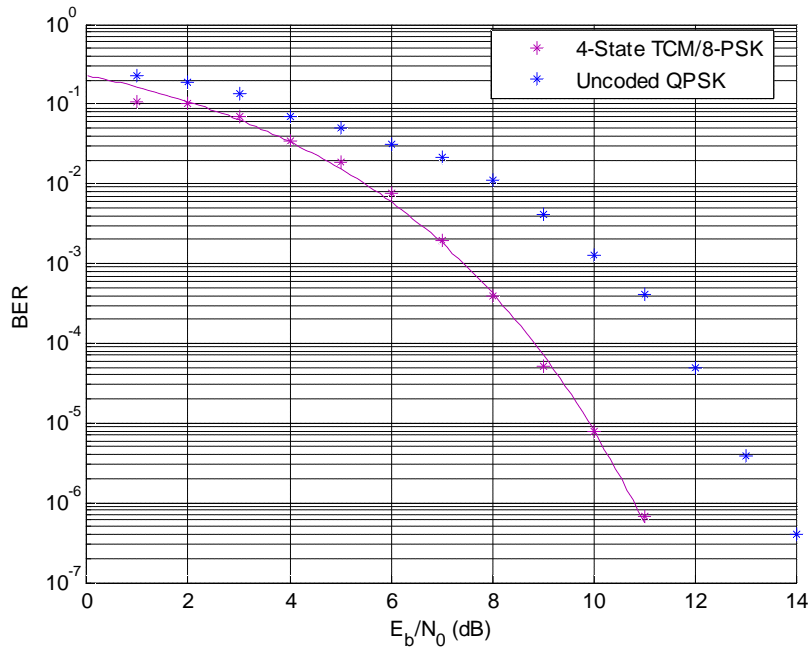


Figure 4.20 BER Versus  $E_b/N_0$  plot for a 4-State TCM/8-PSK  
 b) 8-State Transition TCM/8-PSK

The TCM encoder/decoder blocks of figure 4.19 were set with the polynomial of 8-State Transition TCM/8-PSK obtained in section 3.10(b) as:

$$\text{trellis} = \text{Poly2trellis} ([2 \ 3], [0 \ 3 \ 1; 4 \ 0 \ 3])$$

The `bertoolqsimulation` was conducted on the model and the BER plot obtained is shown in figure 4.21. From the plot at  $\text{BER} = 10^{-6}$ , the results are  $E_b/N_0 = 10.3\text{dB}$  for 8-State TCM against  $E_b/N_0 = 13.5$  for uncoded QPSK, these are recorded in table 4.4

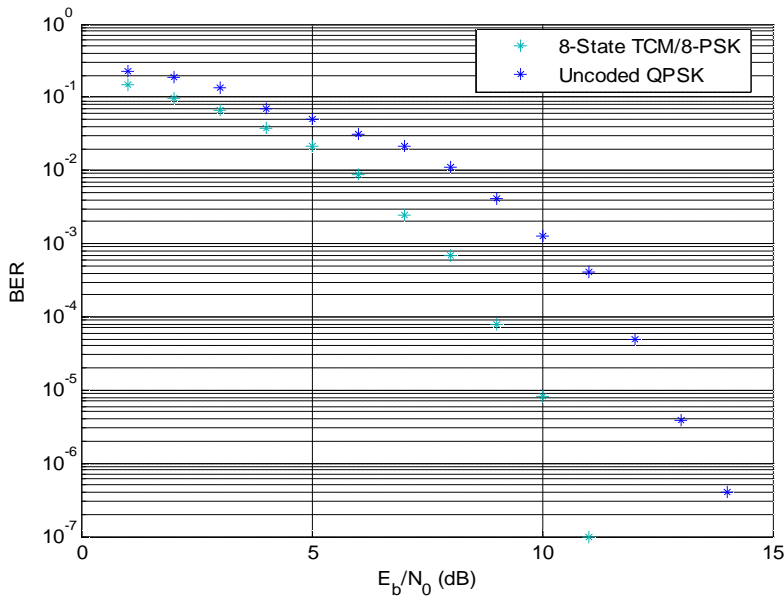


Figure 4.21 BER Versus  $E_b/N_0$  plot for a 8-State TCM/8-PSK

c) 16-State Transition TCM/8-PSK

The TCM encoder/decoder blocks of figure 4.19 were set with the polynomial of 16-State Transition TCM/8-PSK obtained in section 3.10(c) as:

```
trellis = poly2trellis([2 4],[0 3 1; 10 1 5])
```

The `bertoolqsimulation` was conducted on the model and the BER plot obtained is shown in figure 4.22. From the plot at  $BER = 10^{-6}$ , the results are  $E_b/N_0 = 9.2$  dB for 16-State TCM against  $E_b/N_0 = 13.5$  for uncoded QPSK, these are recorded in table 4.4

d) 32-State Transition TCM/8-PSK

The TCM encoder/decoder blocks of figure 4.19 were set with the polynomial of 32-State Transition TCM/8-PSK obtained in section 3.10(d) as:

```
trellis = poly2trellis([3 4],[0 5 3; 10 11 3])
```

The `bertoolqsimulation` was conducted on the model and the BER plot obtained is shown in figure 4.23. From the plot at  $BER = 10^{-6}$ , the results are  $E_b/N_0 = 8.8$  dB for 32-State TCM against  $E_b/N_0 = 13.5$  for uncoded QPSK, these are recorded in table 4.4.

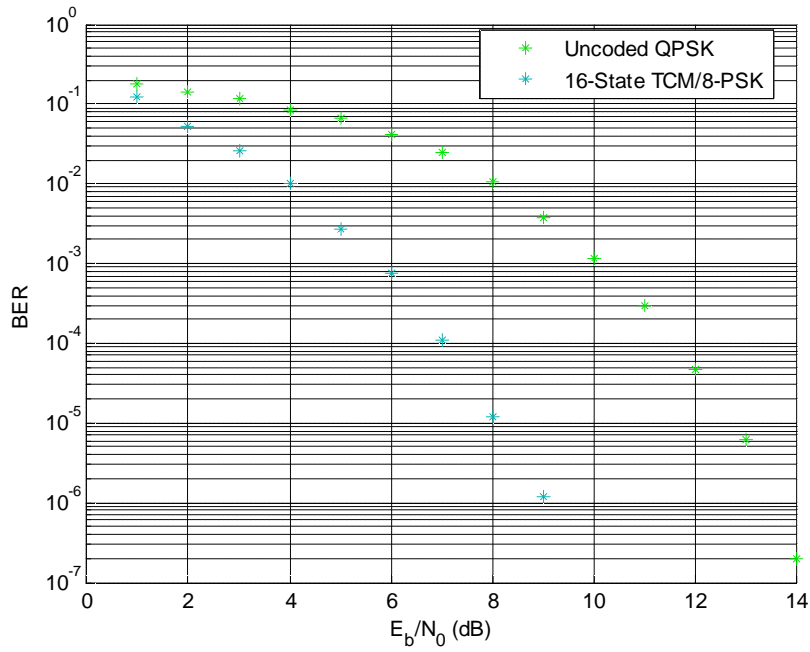


Figure 4.22 BER Versus  $E_b/N_0$  plot for a 16-State TCM/8-PSK

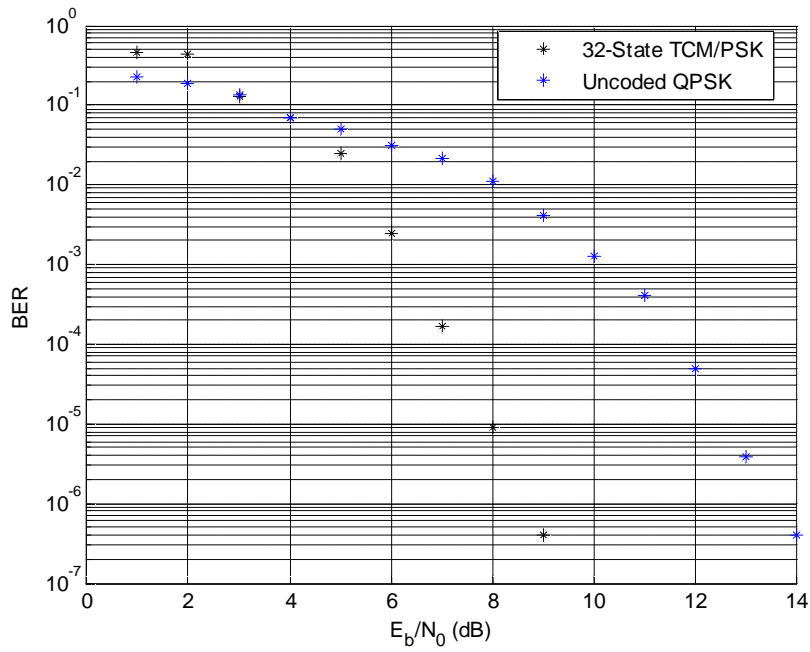


Figure 4.23 BER Versus  $E_b/N_0$  plot for a 32-State TCM/8-PSK

e) 64-State Transition TCM/8-PSK

The TCM encoder/decoder blocks of figure 4.19 were set with the polynomial of 64-State Transition TCM/8-PSK obtained in section 3.10(e) as:

```
trellis = poly2trellis([4 4],[0 11 7; 10 4 3])
```

The `beroolqsimulation` was conducted on the model and the BER plot obtained is shown in figure 4.24. From the plot, at BER =  $10^{-6}$  the results are  $E_b/N_0=8.5\text{dB}$  for 64-State TCM against  $E_b/N_0= 13.5$  for uncoded QPSK, these are recorded in table 4.4

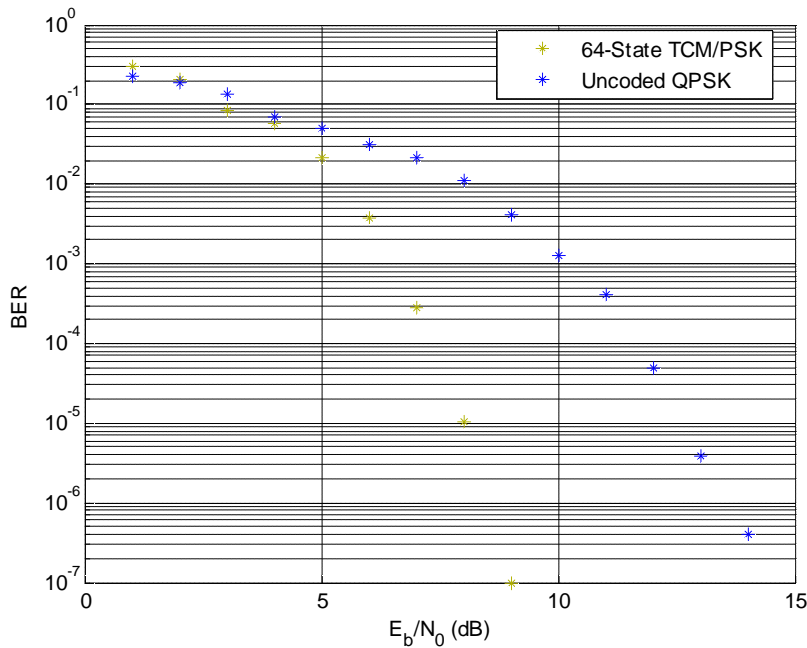


Figure 4.24 BER Versus  $E_b/N_0$  plot for a 64-State TCM/8-PSK

f) 128-State Transition TCM/8-PSK

The TCM encoder/decoder blocks of figure 4.19 were set with the polynomial of 128-State Transition TCM/8-PSK obtained in section 3.10(f) as:

```
trellis = poly2trellis([5 4],[0 23 14; 10 15 3])
```

The ~~h~~ertoolqsimulation was conducted on the model and the BER plot obtained is shown in figure 4.25. From the plot at BER =  $10^{-6}$ , the results are  $E_b/N_0=8.2\text{dB}$  for 128-State TCM against  $E_b/N_0= 13.5$  for uncoded QPSK, these are recorded in table 4.4

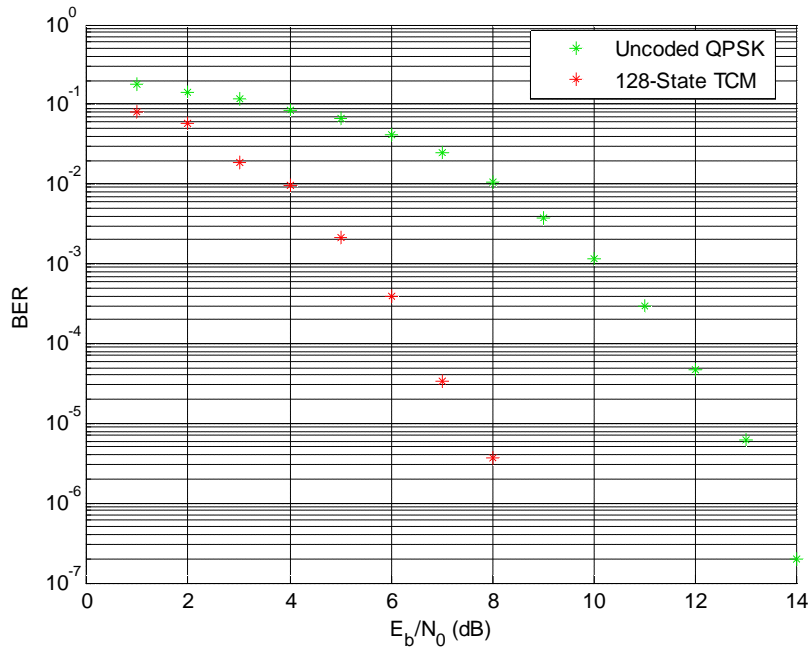


Figure 4.25 BER Versus  $E_b/N_0$  plot for a 128-State TCM/8-PSK

Table 4.4 Performance Summary of Discrete Simulation of Multi-state Encoders On Downlink Block Model at BER =  $10^{-6}$

S/N	Multi-State Encoder	Simulated $E_b/N_0$ (dB)
1	4-State	10.8
2	8-State	10.3
3	16-State	9.2
4	32-State	8.8
5	64-State	8.5
6	128-State	8.2

#### 4.8 Modification and Simulation of the Conceptual Model

All building units for the conceptual model have been simulated and found to be functional, except the Tracking Switch whose function is to track and tap the encoders for few minutes during LEO Satellite passage. In an effort to find a replacement for the Tracker, the simulation software rejected a Timer switch because of prolonged delay in switching as only few microseconds are accepted (i.e. in few multiples of source sampling time  $T_s$ ).

But since the objective is to compensate for varying transmission losses and simulate all the encoders under same BER  $E_b/N_0$ , the model was modified to incorporate the variation.

The model was simulated with the six TCM encoders concurrently switched ON and the expected Free Space Loss for each encoder was inserted along its transmission path. The encoder outputs together with insertion losses are combined in a Frequency Multiplexer using Vector Concatenated block (no carrier frequency is used) as shown in the figure 4.26. The composite was then passed through transmit RRC filter and at AWGN block a uniform  $E_b/N_0$  is applied on the composite. At this level the power of each signal in the composite is a combination of the uniform  $E_b/N_0$  and its inserted loss. The received composite was then filtered by receiver RRC filter block and then demultiplexed. The BER from each channel decoder is now a function of the inserted loss, the uniform  $E_b/N_0$  applied and the coding gain in that channel. When a baseline coding gain of 3dB (4-state coding gain) is assumed on each encoder and any extra or differential gain is removed as Free Space Loss insertion, these are recorded in table 4.5.

Table 4.5 Simulated FSL Insertion on Multiplexed Multi-State Downlink Model.

S/N	Multi-State Encoder	Simulated Insertion Loss (dB)
1	4-State	0.0
2	8-State	0.5
3	16-State	1.6
4	32-State	2.0
5	64-State	2.3
6	128-State	2.6

All encoders in the model were simulated at uniform  $E_b/N_0$  of 10.8 dB and coding gain of 3dB, while each inserted loss was iteratively adjusted until BER of  $10^{-6}$  is obtained from all the decoders. Such that Uniform Data Rate was achieved at uniform  $E_b/N_0$  and BER.

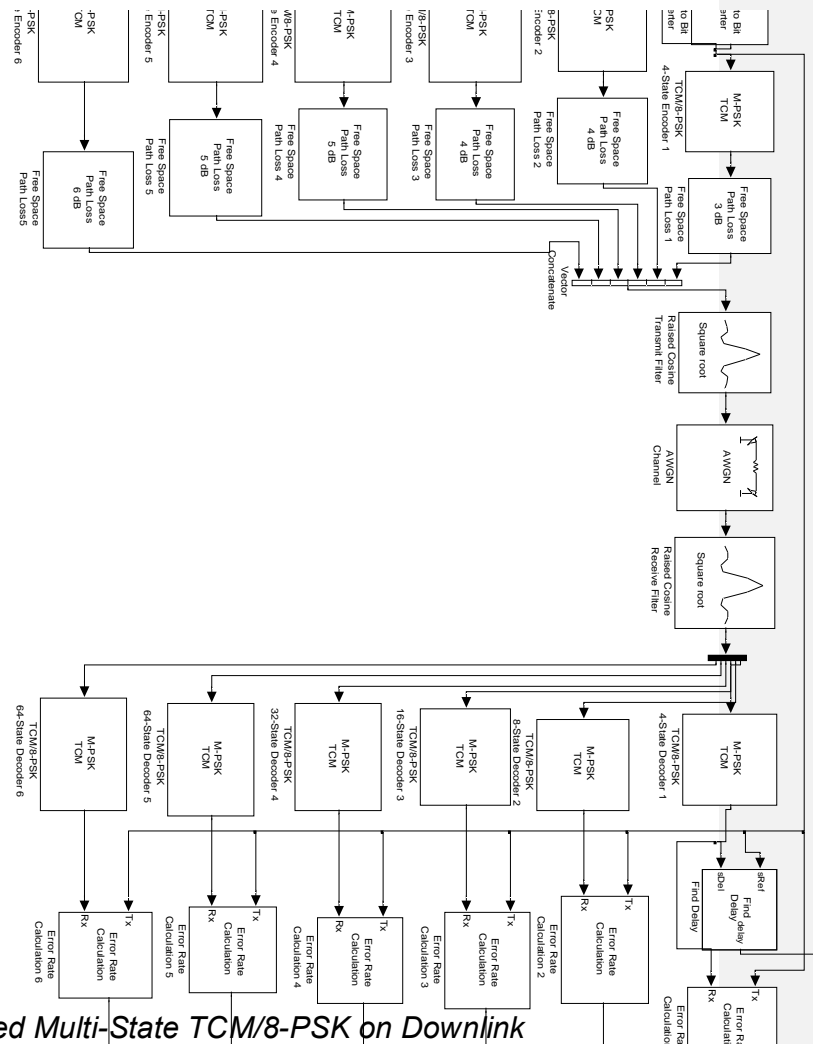


Figure 4.26 Multiplexed Multi-State TCM/8-PSK on Downlink

## STAGE 4

### 4.9 Data Throughput Measurements for Optimisation

In stage 1, 2 & 3 MATLAB bertool is used to simulate the relationship between received power (SNR or  $E_b/N_0$ ) and BER in convolution coding and Trellis coded modulation. The simulations were done at same symbol rate using source sampling time  $T_s = 0.2 \mu s$ , thus using same bandwidth  $B = 5 MHz$  throughout.



In this section bertool is not used, instead the simulations are performed directly on the open file of downlink block model. Simulation run time of 4, 10 and 17 minutes are used, which corresponds to the minimum, average and maximum communication period respectively (as in table 2.02). The parameters to measure are:

- i) Occupied **Bandwidth** ( $B_{occ}$  in MHz) signal to be measured using MATLAB spectrum scope display.
- ii) Received SNR (or  $E_b/N_o$  in dB) to be adjusted in AWGN block to attain at BER =  $10^{-6}$ .
- iii) Simulation run time T (scaled version of real clock time) to be displayed on the running file Task bar
- iv) **Number of decoded bits** received for within the specified period (4, 10 and 17 minutes) to be displayed on Error Calculation Block .

The variable parameter for this simulation is transmission symbol rate  $R_s$ , this is specified by Data Source Sampling Time  $T_s$  in Random Integer Generator Block, such that symbol rate  $R_s = 1/T_s$ .

#### (A) Uncoded MPSK on Downlink Block Models

Uncoded BPSK/QPSK/8-PSK/16-PSK/32-PSK downlink models are formed as in section 4.4 and the simulations are performed as follows:

Simulation Benchmarks:

- i) Data Source sampling time  $T_s = 2 \times 10^{-7}$  s or Symbol rate  $R_s = 5 \times 10^6$  sps
- ii) SNR for each model is adjusted until Bit Error Rate **BER =  $10^{-6}$** .
- iii) RRC filter roll-off factor  $\alpha = 0.1$
- iv) Simulation run time **T** on each model: **4, 10 and 17 minutes**

Results of the measurements are tabulated in table 4.6

Table 4.6 Data Throughput of Un-coded MPSK Downlink Block Models

MPSK	Data Rate R (Mbps)	$E_b/N_o$ (dB)	Received Bits (Gbits)		
			4 (Minutes)	10 (Minutes)	17 (Minutes)

BPSK	5	13.5	1.2	3	5.1
QPSK	10	13.5	2.4	6	10.2
8-PSK	15	15.5	3.6	9	15.3
16 PSK	20	21.0	4.8	12	20.4
32 PSK	25	29.5	6	15	25.5

The results in Table 4.6 above shows that Transmission Data Rate (R) and Number of received bits increases with M modulation level but at increasing demand of SNR or  $E_b/N_o$ . The Spectrums occupied by the MPSK modulated signals are the same as shown in figure 4.13.

### (B) Convolution Coded MPSK on Downlink Block Models

Convolution coded BPSK/QPSK/8-PSK/16-PSK/32-PSK downlink models are formed as in section 4.06 and the simulations are performed as follows:

Simulation Benchmarks: Same as in (A) above

Results of the measurements are tabulated in table 4.7

The results in table 4.7 shows that for convolution coded MPSK, there are small reductions on the required  $E_b/N_o$  for BER=  $10^{-6}$  due moderate coding gain of convolution code. However, received decoded bits or information data throughputs of the link have been reduced due to coding bits overhead. The bandwidth occupied by the coded modulated signals are the same with the un-coded modulated signal of figure 4.13.

Table 4.7 Data Throughput of Convolution Coded MPSK Downlink

Scheme	Code Rate $k/n$	Data Rate R(Mbps)	$E_b/N_o$ (dB)	Bits Decoded (Gbits)		
				4(Minutes)	10 (Minutes)	17(Minutes)
BPSK	1/2	10	10.7	1.2	3	5.1
QPSK	1/2	10	10.7	1.2	3	5.1

8-PSK	2/3	15	14.5	2.4	6	10.2
16-P SK	3/4	20	20.2	3.6	9	15.3
32-PSK	4/5	25	28.5	4.8	12	20.4

### C) Trellis Coded Modulation on Downlink Block Models

Trellis coded modulation on QPSK/8-PSK/16-PSK/32-PSK downlink models are formed as in section 4.8 and the simulations are performed as follows:

Simulation Benchmarks: Same as in (A) above.

Results of the Measurements are tabulated in table 4.8

Table 4.8 Data Throughput of TCM /MPSK Downlink Model

Scheme	Code Rate k/n	Data Rate R (Mbps)	$E_b/N_o$ (dB)	Decoded Bits (Gbits)		
				4 (Minutes)	10 (Minutes)	17 (Minutes)
QPSK/ TCM	1/2	10	7.0	1.2	3	5.1
8-PSK/ TCM	2/3	15	9.8	2.4	6	10.2
16PSK/ TCM	3/4	20	13.0	3.6	9	15.3
32PSK/ TCM	4/5	25	18.2	4.8	12	20.4

The results in Table 4.8 shows some significant reduction in the required Signal power  $E_b/N_o$  for TCM schemes compared to that of Convolution codes with same coding parameters. Bandwidths occupied by the TCM signals are same with that of un-coded and convolution coded signal. Received bits or Information data throughputs for both convolution coded and TCM are same.

### (D) Optimisation of Data Throughput on Multi-State TCM/8-PSK

Data Throughput simulations were run discretely on each of the Multi-State TCM/8-PSK encoder, without the FSL insertion as in section 4.7

Simulation Benchmarks: Same as in (A) above.

Results of the Measurements are tabulated in Table 4.8.

Table 4.9 Data Throughput on Multi-State TCM/8-PSK

Encoder	Code Rate k/n	Data Rate R (Mbps)	$E_b/N_0$ (dB)	Decoded Bits (Gbits)		
				4 (Minutes)	10 (Minutes)	17 (Minutes)
4-State	2/3	15	10.8	2.4	6	10.2
8-State	2/3	15	10.3	2.4	6	10.2
16-State	2/3	15	9.2	2.4	6	10.2
32-State	2/3	15	8.8	2.4	6	10.2
64-State	2/3	15	8.5	2.4	6	10.2
128-State	2/3	15	8.2	2.4	6	10.2

The results of multi-state encoders in table 4.8 have the potentials for optimisation on high data rate of 15Mbps at  $E_b/N_0$  of less 10dB. These results are analysed in details in Chapter 5.

## CHAPTER FIVE ANALYSIS OF SIMULATION RESULTS

### 5.0 Introduction

This chapter is to analyse the advantages of Trellis Coded Modulation over Convolution coded on MPSK by comparison of coding gains from the two techniques and perform optimisation of Data Throughput based on the highlighted constraints of LEO satellite downlink.

Theoretical MPSK and Convolution coded links were initially simulated at stage 1 of chapter 4 to obtain some theoretical link parameters of MPSK and Convolution codes. These parameters were used to choose and fine tune parameters of downlink models built with block units in the MATLAB. The simulation plots obtained for the downlink were superimposed on the plots of corresponding theoretical links for easy comparison. Similarly in stage 2 downlink block models built with TCM/MPSK schemes were simulated and the plots were superimposed again on that of the theoretical. In both convolution coding and TCM optimum code rates were applied to conserve bandwidth (i.e. code rate =  $k/n$ , with  $n=k+1$ ).

In this chapter coding gain is calculated from the simulation results of chapter four, it is the difference between received  $E_b/N_o$  of uncoded and coded links transmitting same number of information (source) bits per symbol plots at a given BER (see chapter 3). In this work the values  $E_b/N_o$  are obtained from interpolation at  $BER = 10^{-6}$  on the plots in figure 4.3-7

The simulation in stage 3 was performed on six multi-state TCM/8-PSK encoders, initially the encoders were simulated discretely and BER plot for each was obtained. The conceptual model could not be simulated as proposed due to unavailability of the Tracking sensor/ selector switch, as a result it was modified and the encoders were simulated in composite. A frequency multiplexer was used to form the composite and a power loss block was inserted along each path to reflect expected variation in FSL. Implications of the modified conceptual model and the simulated power loss insertions would be analysed.

In stage 4 of chapter 4, data throughput measurements were performed on all the models simulated including the discrete simulations on six multi-state TCM/8-PSK encoders, the results obtained would be compared for optimisation of the downlink.

### **5.1 Performance of Convolution coded MPSK Theoretical Link**

Theoretical MPSK links were simulated in section 4.2 and simulation plots are shown in figure 4.3-7. The plots of uncoded BPSK in figure 4.3 and that of uncoded QPSK in figure 4.4 are the same and have equal  $E_b/N_o$  requirements for all values of BER, this confirmed the assertion made on probabilities of error ( $P_e$ ) of the two schemes in chapter 2.

Table 5.1 Performance Analysis of Theoretical MPSK Link

	MPSK Scheme	Input k info bit per symbol	$E_b/N_o$ (dB) At BER = $10^{-6}$	
			Uncoded	Coded
1	BPSK	1	10.7	8
2	QPSK	2	10.7	8
3	8-PSK	3	14.0	11.0
4	16 PSK	4	18.2	17.3
5	32 PSK	5	23.5	24.5

The values of  $E_b/N_o$  at BER= $10^{-6}$  for uncoded and coded Theoretical MPSK links were read from the performance plots of shown in figure 4.03-07 and tabulated in table 5.01. In both BPSK and QPSK at BER= $10^{-6}$ , the required  $E_b/N_o = 10.5\text{dB}$ , this is same as predicted in theory in figure 2.09 of chapter two at  $P_e=10^{-6}$ . Theoretical uncoded 8-PSK on figure 4.05 requires  $E_b/N_o$  of 14dB at BER= $10^{-6}$  which was predicted in figure 2.10 at  $P_e =10^{-6}$  and same is true with 16-PSK on figure 4.6. These results confirmed that:

- i) Probability of error ( $P_e$ ) for uncoded digital link can be measured as Bit Error Rate (BER).
- ii) The obvious advantage of QPSK over BPSK, as both have same power requirements but the former has higher spectral efficiency of 2bits/Hz. For this reason this research work dropped further consideration on BPSK in on throughput optimization.

Comment [D4]:

When MPSK is used on convolution coded data, its BER plots also maintained the curvature shape as the uncoded with difference as shown in figure 4.3-7. On  $\frac{1}{2}$ -rate coded QPSK link for instance, the power requirements in  $E_b/N_o$  at BER= $10^{-6}$  dropped from 10.7 to 8dB as shown in table 5.1. To calculate coding gain of this convolution code on QPSK, the power requirement is compared with that of uncoded scheme that transmit same source bit per symbol, which in this case is uncoded BPSK and the difference of the two power requirements is taken as coding gain, in this case coding gain = (10.7 - 8) = 2.7dB. Similar comparisons are made on four levels of MPSK, each on optimal coded rate in table 5.2.

Table 5.2 Coding Gains of Convolution Codes on Theoretical MPSK Links

k	Uncoded MPSK Scheme	$E_b/N_o$ (dB)	Optimal Code Rate k/n	Coded (M+1)PSK at k+1 bit/symbol	$E_b/N_o$ (dB)	Coding Gain $G_{cc}$ (dB)
1	BPSK	10.5	1/2	QPSK	8	2.5
2	QPSK	10.5	2/3	8-PSK	11.0	-0.5
3	8-PSK	14	3/4	16-PSK	17.3	-3.3
4	16PSK	18.2	4/5	32PSK	24.5	-6.3

The results shows positive gain for  $\frac{1}{2}$ -rate convolution coded data on QPSK and negative gain for other optimally coded link on higher MPSK schemes. This implies that on higher MPSK, which are generally more bandwidth efficient requires more power is required to transmit source data than at lower uncoded (M-1)PSK. This verified the assertion made on inadequacy of Convolution Coding on power constrained link. in chapter 3.

## 5.2 Performance of Convolution Coded MPSK on Downlink Block Models

The results of Theoretical MPSK links are in agreement with the theories in chapter two on MPSK schemes mentioned, but still remained hypothetical. Performance of a communication model built with MATLAB block units in which link parameters such as data rate, filters, power gains and losses etc are specified is more realistic. Before any search work on optimization is done a Downlink block models, its validity are verified against the hypothetical results. The block model parameters are fine-tuned until its performances are within acceptable implementation margin.

Simulations of uncoded MPSK block model were conducted in section 4.5 and its BER plots were again superimposed on figure 4.3-7. The BER plots obtained for the uncoded block model also maintained the curvature of that from the Theoretical links. Values of  $E_b/N_o$  at  $BER=10^{-6}$  for the uncoded downlink block models are obtained from the plots and tabulated in table 5.3(a). Implementation margin in comparison with Theoretical values are calculated and tabulated against each model. The implementation margins varies among the MPSK schemes in uncoded models, 8-PSK has the least implementation margin of 1.5dB, while 32-

PSK has implementation margin 6.5dB, the results are within expected values due to application of non-ideal filters and the high probability of symbol error in 32-PSK (equation 2.58)

Table 5.3 Performance Analysis of Uncoded and Convolution Coded MPSK on Downlink Block Models

a) Uncoded

	MPSK Scheme	$E_b/N_o$ (dB) At BER = $10^{-6}$		Implementation Margin (dB)
		Theoretical Link	Block Downlink	
1	BPSK	10.5	13.5	3.0
2	QPSK	10.5	13.5	3.0
3	8-PSK	14.0	15.5	1.5
4	16 PSK	18.0	20.4	1.6
5	32 PSK	23.0	29.5	6.5

b) Convolution Coded

	MPSK Scheme	$E_b/N_o$ (dB) At BER = $10^{-6}$		Implementation Margin (dB)
		Theoretical Link	Block Downlink	
1	BPSK	8	10.7	2.7
2	QPSK	8	10.7	2.7
3	8-PSK	11.0	14.5	3.5
4	16 PSK	17.3	19.4	3.1
5	32 PSK	24.5	28.5	4.0

When block units of convolution encoders and decoders were added to the block model with same polynomial setting (as in Theoretical link), its performance plots were superimposed on figure 4.3-7. Values of  $E_b/N_o$  at BER= $10^{-6}$  for the coded models dropped slightly from the values of uncoded and implementation margin against coded Theoretical are tabulated in table 5.3(b). The discrepancy on implementation margins between the uncoded and coded tables indicates suitability of each convolution encoder on its MPSK. Coding gain for the convolution codes on the block models are calculated and tabulated in table 5.4. The results on coding gain showed positive coding gain on QPSK only, while on higher MPSK the gains are negative, this means for  $M > 4$  on the same



bandwidth, more power is required to transmit convolution coded bits ( $k+1$ ) than uncoded bits ( $k$ ). The discrepancy on coding gains of the same code between table 5.2 and 5.4, means that the Convolution code have its implementation margin between Theoretical and block models.

Table 5.4 Coding Gains of Convolution Codes on MPSK Block Downlink Block Models

k	Uncoded MPSK k bits/symbol	$E_b/N_o$ (dB) (Uncoded)	Coded (M+1)-PSK with k+1 bit/symbol	Coded $E_b/N_o$ (dB) (Coded)	Coding Gain (dB)
1	BPSK	13.5	QPSK	10.7	2.8
2	QPSK	13.5	8PSK	14.5	-1.0
3	8PSK	15.5	16PSK	19.4	-3.9
4	16PSK	20.5	32PSK	28.5	-8.0

### 5.3 Performance of TCM/MPSK on Downlink Block Models

MATLAB software does not support Theoretical TCM simulation; however it supports simulation of TCM on block model. Therefore TCM on MPSK downlink block models were formed and simulated in stage 2 of chapter 3. TCM simulation plots were also superimposed on the figure 4.3-7 for comparison. The values of  $E_b/N_o$  at BER =  $10^{-6}$  in table 5.5 for TCM/MPSK dropped more than in table 5.4 for Convolution codes.

Table 5.5 Performance Analysis of TCM/MPSK on Downlink block model

SCHEME	$E_b/N_o$ (dB) at BER = $10^{-6}$				
	BPSK	QPSK	8-PSK	16-PSK	32-PSK
Uncoded	13.5	13.5	17.2	22.3	28.8
TCM	-	7	9.5	13.0	18.2

The coding gain in TCM scheme is also calculated against uncoded scheme transmitting same information bits at same BER as in

convolution coded schemes. TCM coding gains are calculated and tabulated in table 5.6.

Table 5.6 Coding Gain of TCM/MPSK on Downlink Block Models

k	Uncoded MPSK $M=2^k$		TCM /MPSK Scheme $M=2^{k+1}$		
		$E_b/N_0$ (dB)		$E_b/N_0$ (dB)	Coding Gain(dB)
1	BPSK	13.5	QPSK	7	6.5
2	QPSK	13.5	8PSK	9.5	4.0
3	8PSK	15.5	16PSK	13.0	2.5
4	16PSK	20.40	32PSK	18.2	2.2

The TCM has more positive coding gain (at all modulation levels) than on convolution coded models, the advantage of the later over the former is manifested in comparison table 5.7. This has proof the validity of the built Downlink Block models and its readiness for use in Optimisation of Data Throughput.

Table 5.7 Coding Gain Comparison between Convolution coded and TCM

	MPSK	Convolution Coding Gain (dB)	TCM Coding Gain (dB)
1	QPSK	2.8	6.5
2	8-PSK	-1.0	4.0
3	16-PSK	-3.9	2.5
4	32-PSK	-8.0	2.2

It has been deduced furthermore in section 3.9, that coding gain on each TCM can be increased when number of its logical transition states is increased. Thus, the Conceptual model of section 3.10 was developed with multiple transition state encoders.

#### 5.4 Discrete Performance of Multi-State Encoders

Discrete simulations for the six Multi-state TCM/8-PSK encoders were performed in Section 4.7, BER performance plots obtained were shown in figure 4.19-25. The plots are used to obtain values of  $E_b/N_0$  at BER =  $10^{-6}$  and coding gains are calculated as follows:

a) 4-State TCM/8-PSK Encoder;

Reading from plot in figure 4.20

$E_b/N_0 = 13.5\text{dB}$  for uncoded QPSK at BER= $10^{-6}$

$E_b/N_0 = 10.8\text{dB}$  for 4-State encoder at BER =  $10^{-6}$

Coding Gain =  $(13.5 - 10.8) = 2.7\text{dB}$

b) 8-State TCM/8-PSK Encoder;

Reading from plot in figure 4.21

$E_b/N_0 = 13.5\text{dB}$  for uncoded QPSK at BER= $10^{-6}$

$E_b/N_0 = 10.3\text{dB}$  for 8-State TCM/8-PSK at BER= $10^{-6}$

Coding Gain =  $(13.5 - 10.3) = 3.3\text{dB}$ .

c) 16-State TCM/8-PSK Encoder;

Reading from plot in figure 4.22

$E_b/N_0 = 13.5\text{dB}$  for uncoded QPSK at BER= $10^{-6}$

$E_b/N_0 = 9.2\text{dB}$  for 16-State TCM/8-PSK at BER= $10^{-6}$

Coding Gain =  $(13.5 - 9.2) = 4.3\text{dB}$ .

d) 32-State TCM/8-PSK Encoder;

Reading from plot in figure 4.23

$E_b/N_0 = 13.5\text{dB}$  for uncoded QPSK at BER= $10^{-6}$

$E_b/N_0 = 8.8\text{dB}$  for 32-State TCM/8-PSK at BER= $10^{-6}$

Coding Gain =  $(13.5 - 8.8) = 4.7$

e) 64-State TCM/8-PSK Encoder;

Reading from plot in figure 4.24

$E_b/N_0 = 13.5\text{dB}$  for uncoded QPSK at BER= $10^{-6}$

$E_b/N_0 = 8.5\text{dB}$  for 64-State TCM/8-PSK at BER= $10^{-6}$

Coding Gain =  $(13.5 - 8.5) = 5.0\text{dB}$

f) 128-State TCM/8-PSK;

Reading from plot in figure 4.25

$E_b/N_0 = 13.5\text{dB}$  for uncoded QPSK at BER= $10^{-6}$

$E_b/N_0 = 8.2\text{dB}$  for 128-State TCM/8-PSK at BER= $10^{-6}$

Coding Gain =  $(13.5 - 8.2) = 5.3\text{dB}$ .

The above simulated coding gains and the Theoretical coding gains obtained in section 3.10 are tabulated in table 5.8, the implementation margins are calculated accordingly.

Table 5.8 Discrete Coding Gains of Multi-State TCM/8-PSK Encoders

	Theoretical Coding Gain (dB)	Simulated Coding Gain (dB)	Implementation Margin (dB)
4-State	3.0	2.7	-0.3
8-State	3.6	3.3	-0.2
16-State	4.32	4.3	-0.02
32-State	4.771	4.7	-0.07
64-State	5.128	5.0	-0.12
128-State	5.54	5.3	-0.24

The differences between simulated coding gains and Theoretical coding gains in table 5.8 are due to implementation margin, with 128-state and 32-State having the highest of 1.1dB and the lowest of 0.03dB respectively.

### 5.5 Performance of Multiplexed Multi-State TCM/8-PSK Model

The Conceptual model was conceived to counter the variation in Downlink Free Space Loss of LEO Satellite, it was modified in section 4.8 due to lack of satellite tracking sensor in the MATLAB to select the counteracting encoder gain. In the modified model all the encoders were simulated in composite, while the variation in the Free Space Loss was made to be equivalent to differential coding gains of the Multi-State encoders. The model was simulated at uniform  $E_b/N_0$ , a baseline coding gain for a 4-State encoder at 3dB was used for each encoder and the differential gain inserted as Free Space Loss.

Table 5.9 Differential Coding Gains of Multi-State Encoders from 3dB

S/N	TCM/8-PSK	Theoretical Coding Gain	Differential from 3dB		Simulated Coding Gain
			Theoretical	Simulated	
1	4-State	3.0	0.0	0.0	3.0
2	8-State	3.6	0.6	0.5	3.5

3	16-State	4.12	1.12	1.0	4.15
4	32-State	4.77	1.7	1.2	4.2
5	64-State	5.12	2.12	1.55	1.5
6	128-State	5.54	2.54	2.0	2.0

The inserted differential losses were obtained iteratively at uniform BER= $10^{-6}$ , these are tabulated in table 5.9. Theoretical differential gains in the table are calculated from expected coding gain in section 3.10. The results from all the encoders under the variation in Loss were: i) uniform  $E_b/N_0$  delivered uniform BER ii) uniform Bandwidth supplied uniform Data rate.

### 5.6 Data Throughput Comparison and Optimisation on LEO Downlink Block Models

In section 4.9 Data Throughput were measured for Uncoded MPSK, Convolution coded MPSK and Trellis coded modulation MPSK on Downlink Block models. A uniform symbol rate of 5 Msps was transmitted on Bandwidth of 5.5 MHz in all the models. Optimum code rates of  $k/k+1$  were used on both Convolution coded and Trellis coded modulation.

At BER= $10^{-6}$  Data bits Received/Decoded were counted for periods of 4, 10 and 17 minutes, the results are summarised in table 5.10.

Table 5.10 Data Throughput Comparison between Uncoded, Convolution Coded and TCM on MPSK Downlink Block Model

Scheme	Code Rate	$E_b/N_0$ (dB)	Transmission Data Rate R (Mbps)	Decoded Data Throughput (Gbits)		
				4	10	17 (Minu)

				(Minutes)	(Minutes)	tes)
BPSK	-	13.5	5	1.2	3	5.1
QPSK	-	13.5	10	2.4	6	10.2
8-PSK	-	15.5	15	3.6	9	15.3
16-PSK	-	20.5	20	4.8	12	20.4
32-PSK	-	29.5	25	6.0	15	25.5
Coded BPSK	1/2	10.7	5	0.6	1.5	2.55
Coded QPSK	1/2	10.7	10	1.2	3	5.1
Coded 8-PSK	2/3	14.5	15	2.4	6	10.2
Coded 16-PSK	3/4	19.5	20	3.6	9	15.3
Coded 32-PSK	4/5	28.5	25	4.8	12	20.4
TCM/BPSK	-	-	-	-	-	-
TCM/QPSK	1/2	7.0	10	1.2	3	5.1
TCM/8PSK	2/3	9.5	15	2.4	6	10.2
TCM/16PSK	3/4	13.0	20	3.6	9	15.3
TCM/32PSK	4/5	18.2	25	4.8	12	20.4

When volume of Data Throughput (counted Data) for each scheme is compared against its power ( $E_b/N_o$ ) requirements, the following deductions are made:

- i) To deliver Data Throughput of 1.2Gbits in minimum downlink communication time of 4 minutes, uncoded BPSK requires  $E_b/N_o=13.5\text{dB}$ ;  $\frac{1}{2}$  rate convolution coded QPSK requires  $E_b/N_o=10.7\text{dB}$ ; while  $\frac{1}{2}$  rate TCM/QPSK requires  $E_b/N_o=7.0\text{dB}$ .
- ii) To deliver 6 Gbits of data throughput in average downlink communication time of 10 minutes, uncoded QPSK requires  $E_b/N_o=13.5\text{dB}$ ;  $\frac{2}{3}$  rate convolution coded 8-PSK requires  $E_b/N_o=14.5\text{dB}$ ; while  $\frac{2}{3}$  rate TCM/8-PSK requires  $E_b/N_o=9.5\text{dB}$ .
- iii) To deliver 9Gbits of data throughput in average downlink time of 10 minutes, uncoded 8-PSK link requires  $E_b/N_o=15.5\text{dB}$ ;  $\frac{3}{4}$  rate convolution coded 16-PSK requires  $E_b/N_o=19.4\text{dB}$ ; while  $\frac{3}{4}$  rate TCM/16-PSK requires  $E_b/N_o=13.0\text{dB}$ .
- iv) To deliver 12Gbits of data throughput in maximum downlink time of 17minutes, uncoded 16-PSK link requires  $E_b/N_o=20.5\text{dB}$ ;  $\frac{4}{5}$  rate convolution coded 32-PSK requires  $E_b/N_o=28.5\text{dB}$ ; while  $\frac{4}{5}$  rate TCM/32-PSK requires  $E_b/N_o=18.2\text{dB}$ .

The highest data throughput recorded is on 32-PSK at high value of  $E_b/N_0=28.5$  for convolution coding and at  $E_b/N_b= 18.2\text{dB}$  for TCM/32-PSK. Convolution coding and TCM delivers same data throughput, but the latter has advantage to deliver same BER at lower  $E_b/N_0$ . For optimization of data throughput in a power limited downlink, TCM/8-PSK has moderate throughput and also moderate power requirement  $E_b/N_0=9.5\text{dB}$ .

Consequently, search for higher coding gain on TCM/8-PSK was conducted as shown in section 4.09(D).

Table 5.11 Optimisation of Data Throughput on Multi-State TCM/8-PSK

Encoder	Code Rate k/n	Data Rate R (Mbps)	$E_b/N_0$ (dB)	Decoded Bits (Gbits)		
				4 (Minutes)	10 (Minutes)	17 (Minutes)
4-State	2/3	15	10.8	2.4	6	10.2
8-State	2/3	15	10.3	2.4	6	10.2
16-State	2/3	15	9.2	2.4	6	10.2
32-State	2/3	15	8.8	2.4	6	10.2
64-State	2/3	15	8.5	2.4	6	10.2
<b>128-State</b>	<b>2/3</b>	<b>15</b>	<b>8.2</b>	<b>2.4</b>	<b>6</b>	<b>10.2</b>

The 128-State TCM/8-PSK encoder that offered the highest coding gain is therefore selected for Optimisation of Data Throughput on LEO Satellite Downlink. The result is shown in table 5.11

## CHAPTER SIX

### CONCLUSION AND RECOMMENDATION

#### CONCLUSION:

This Thesis has expounded the potentials of Trellis Coded Modulation scheme and applied it to LEO satellite downlink that is constrained in time, power and bandwidth. A TCM/8-PSK scheme was identified to be applicable to the optimization of data throughput because of its modest demand on power and good utilization of bandwidth. The scheme was explored with MATLAB downlink block models, it generated significant coding gain in the range of 2.7 to 5.4dB at BER=  $10^{-6}$ . On the other hand convolution code generated maximum coding gain 2.8dB on QPSK.

A 128-State Transition TCM/8-PSK encoder that generated coding gain of 5.4dB was specifically selected for the optimization. Data rate of 15 Mbps was realized with the TCM encoder on a downlink bandwidth of 5.5MHz, **at received  $E_b/N_0 = 8.2\text{dB}$**  and BER =  $10^{-6}$ . While same data rate of 15Mbps was realized with convolution encoder on 8-PSK on a downlink bandwidth of 5.5MHz, **but at received  $E_b/N_0 = 14.5\text{dB}$**  and BER =  $10^{-6}$ , a difference in  $E_b/N_0$  of 6.3dB was realized in this optimization work and the bandwidth efficiency of 8-PSK was conserved at approximately 3bits/Hz.

Thus, Optimisation of Data Throughput was achieved at optimum code rate (1-bit overhead) on the downlink of LEO satellite. Decoded information bits received at 2.4 Gbits, 6Gbits and 10.2Gbits which corresponds to minimum (4 minutes), average (10 minutes) and maximum (17 minutes) durations of satellite passage over a Ground station respectively. Although the results obtained in this work were not close to Shannon's limits as that was not the objective, the objective was to achieve high coding gain of TCM and high spectral efficiency of MPSK schemes for reliable LEO Satellite downlink at QoS of BER=  $10^{-6}$ . In general, coding techniques with very high coding gain that reduces SNR to near Shannon's limits, such as Turbo and Concatenated Coding add large number of overhead bits to reduce the information throughput.

Attempts were made to explore coding gain variations within a TCM/8-PSK scheme as predicted in table 3.6. A Conceptual model of Multi-State TCM/8-PSK downlink model was proposed in Chapter Three, its code structures were to change in steps during the satellite passage, each step has its coding gain to counter the downlink FSL. Coding structures for six steps were developed and simulated discretely, six



coding gains were distinctly realized (2.7 to 5.3 dB in table 5.08). However, steps simulation on the encoder could not be realized on the MATLAB, due to long delay on unswitched encoders detected by the Software. Consequently the model was modified with the six encoders simulated in composite using Frequency Multiplexer, a baseline coding gain of 2.7dB was reserved for each encoder and the differentials were used to neutralize variation in Free Space Loss. The simulation was performed at uniform bandwidth of 5.5MHz and uniform  $E_b/N_o$  of 10.8dB, the results achieved were uniform data rate of 15Mbps at uniform BER =  $10^{-6}$  across all the encoders despite the variation in FSL (see table 4.00).

**RECOMMENDATION:**

Although the neutralizations were short of the maximum variation in FSL of 10dB (deduced from equation 2.28), the principle is still applicable other wireless link with small FSL variation (2.4dB) e.g WI-FI where the distance range <10 Km.

**REFERENCES**

[1] James R. Wertz and Wiley J. Carson, Space Mission Analysis and Design, Third Edition, Published by Microcosm Press Torrance California 1999.

- [2] Denis Roddy, Satellite Communications, Third Edition, Published by MacGraw Hill Companies, Inc . 2001.
- [3] Timothy Pratt, Charles Bostian and Jeremy Allnutt, Satellite Communications, Published by John Wiley & Sons Inc. 2003.
- [4] John G. Proakis, Digital Communications, Published by McGraw Hills Companies Inc. 2001.
- [5] Peter Fortescue and John Stark, Spacecraft Systems Engineering Second Edition, Published by Wiley and Sons. 2001.
- [6] John G. Proakis and Masoud Salehi, Communication Systems Engineering, Published by Pearson Prentice Hall. 2004.
- [7] M.A Matar, Throughput /Channel Capacity Improvement Issues for LEO Satellite Based Remote Sensing Systems, IEEE Publication 2003.
- [8] VY Chu, W Sun, Professor M.N Sweeting; Optimising LEO Satellite Communication Link Through Hybrid-ARQ Techniques, Centre for Satellite Engineering Research University of Surrey, United Kingdom, IEEE Publication 1997.
- [9] L. HadjAbderrahman, D.E Baba Hameed and M Benyettou; Design of an Adaptive Communication System for Implementation on Board a Future Algerian LEO Satellite, Centre of Space Techniques Division of Space Instrumentation/University of Science and Technology Oran Algeria; IEEE publication 2007.
- [10] Onde Belce, Comparison of Advanced Modulation Schemes for LEO Satellite Downlink Communications, Tubitak Bilten, Ankara Turkey; IEEE 2003
- [11] Chip Fleming , A Tutorial on Convolution Coding with Viterbi Decoding Spectrum Applications 2002 (also available on the Internet).
- [12] Charan Lanton, Intuitive Guide to Principle of Communication, [www.complextoreal.com](http://www.complextoreal.com) 2004.
- [13] MATLAB R2010 Software Support Documentation.
- [14] Behzad Razavi, RF Micro Electronics, Pearson Education Asia Limited, 2003.

[15] D. Divisalar, S.SDolinar, F Pollara ; Serial Turbo Trellis Coded Modulation with Rate -1Inner Code Jet Propulsion Laboratory, California Institute of Technology, 2000 (published on the Internet).

[16] Peter Jantunen; Modelling of Amplifier Nonlinearities I Helsinki University of Technology Signal Processing Laboratory(Published on the Internet).

[17] Ghorbani andSheikhan , The Effect of Solid State Power Amlifier(SSPA) non linearities on MPSK and MQAM Transmission, IEEE publication 1991.

[18] Surrey Satellite Technology Documentation on Nigeriasat-1, NASRDA Library (Unpublished).

## **APPENDIX A**

### **Pulse Shaping and Nyquist Filters**

When time limited signal passes through bandwidth limited system the response is time unlimited signal, this is readily demonstrated by passing time limited rectangular pulses through bandwidth limited low pass filter (i.e. RC circuit), the output signal exhibit exponential tails corresponding to rise and fall of the input pulses as shown in figure 1A.

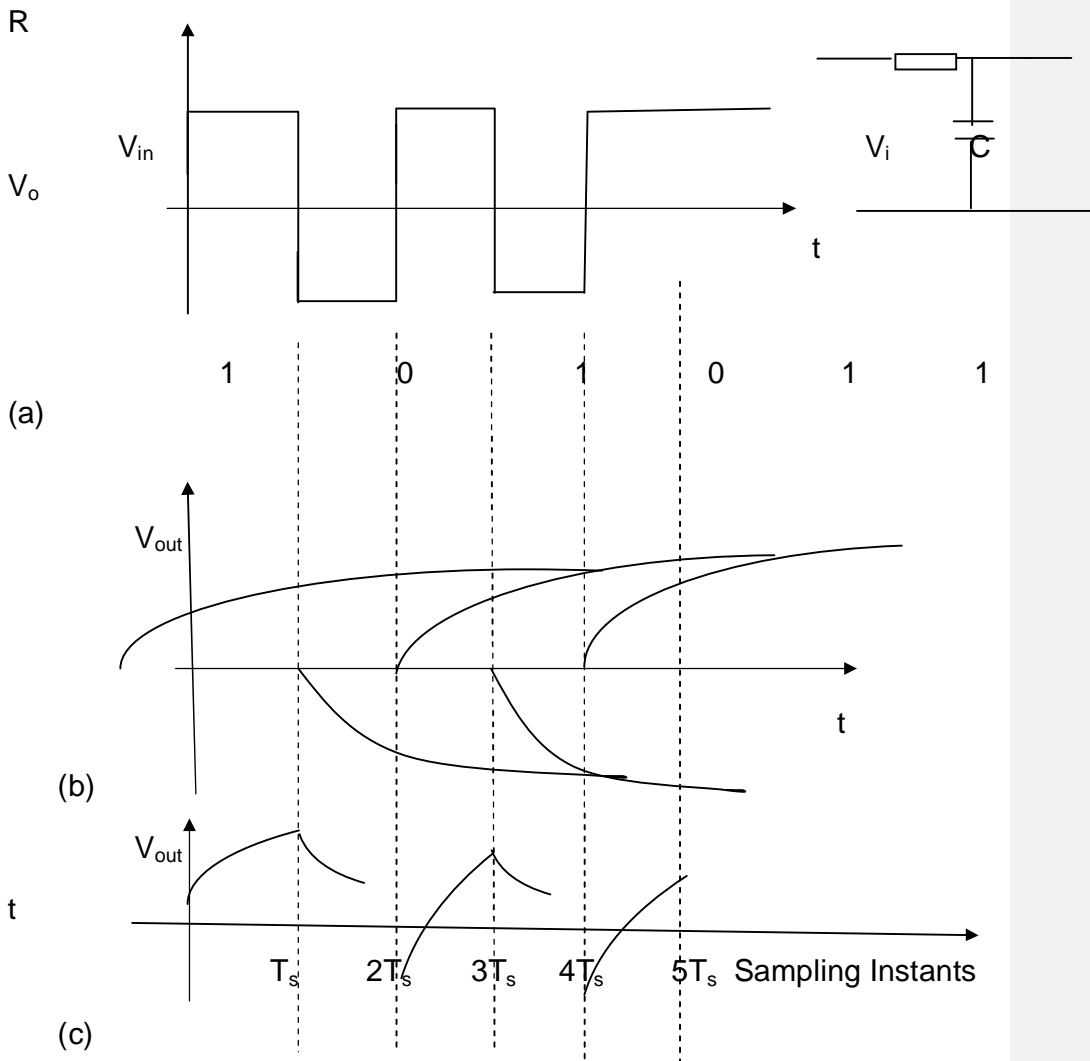


Figure 1A (a) NRZ pulse input to an RC Filter (b) Time Response of Individual pulse (c) Time Response of Pulse Train

The exponential portion from one transition extends to the next pulse interval resulting in the output waveform being smeared in time and where the digital pulse pattern is 10 or 01 the amplitude of the second

pulse is reduced by the presence of delayed portion of the preceding pulse this is called InterSymbol Interference (ISI). In addition to noise ISI also increases the likelihood of a receiver for digital communication to detect bit.

The frequency response of the output pulse  $V_o(f)$  at the receiver is determined by the frequency response of the input pulse  $V_i(f)$ , the transmit filter frequency response  $H_T(f)$ , the channel frequency response  $H_{ch}(f)$  and the receiver filter response  $H_R(f)$  as shown in figure 2A such that

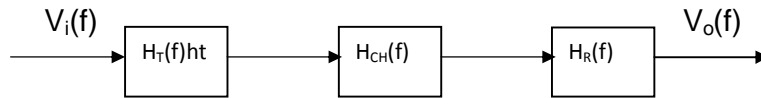


Figure 2A Digital Baseband Transmission Path

$$V_o(f) = V_i(f) H_T(f) H_{CH}(f) H_R(f) \dots 1a$$

For wireless channel as in satellite link, it is generally considered that  $H_{CH}(f) = 1$  that is the satellite link has flat transfer function and the phase response is linear,[3] thus

$$V_o(f) = V_i(f) H_T(f) H_R(f) \dots 2a$$

The inherent inductive and capacitive elements in the two filters ultimately lead to exchange of energy between the magnetic and electric fields so that the square pulses that enters the system may exhibits some ringing at its exits as shown in figure 3Aa. Since these ringing cannot be removed, a criteria was proposed by Nyquist, such that the pulses can be shaped so that sampling of a pulse occurs when the tails of the previous pulses are at zero crossover as shown in figure 3Ab. To achieve this Nyquist Criteria requires to create a pulse in the receiver of the type  $\sin x / x$  or  $\text{sinc } x$  which can be generated when input is an impulse ( $V_i(t) = \delta(t)$ ) so that  $V_i(f) = 1$  and its frequency response is shown in figure 3Ac,

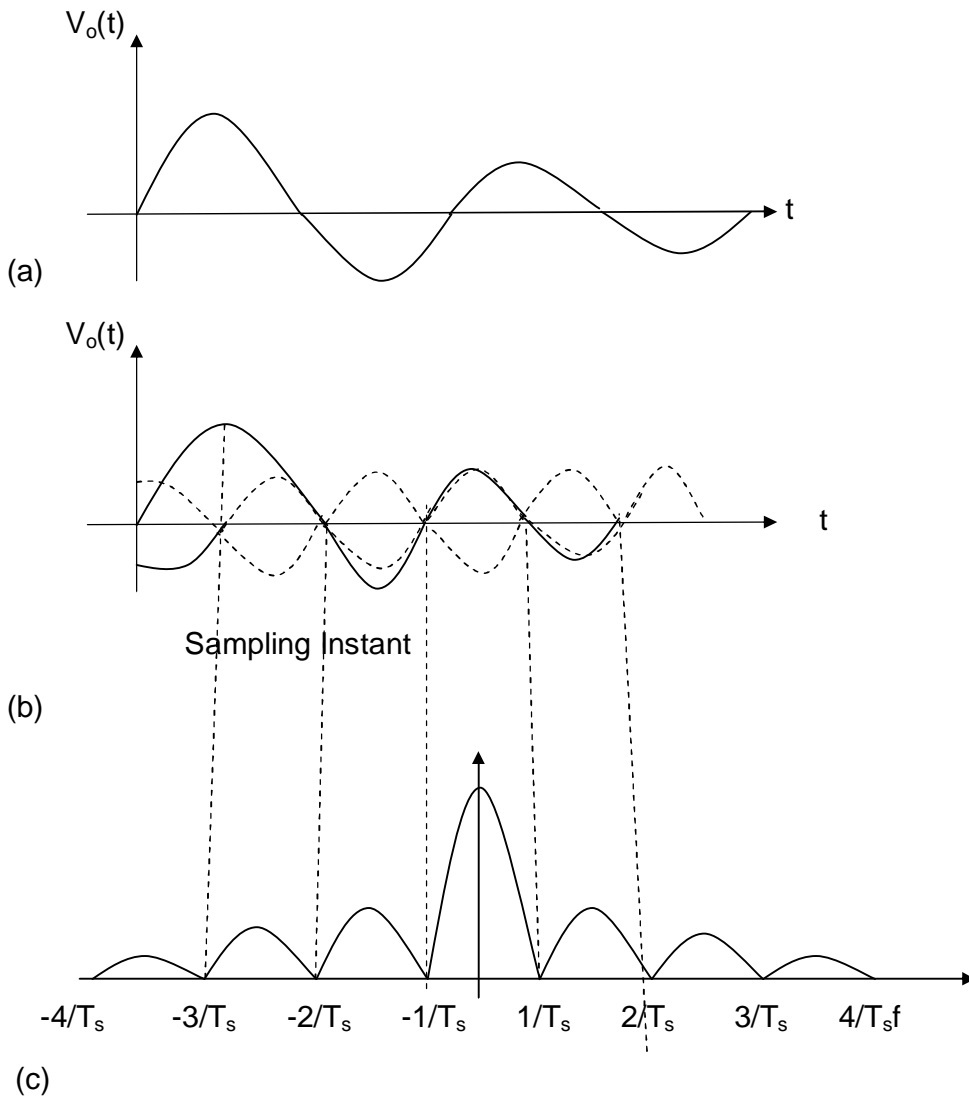


Figure 3A (a) Pulse Ringing Due inductive and Capacitive effect of Filter  
 (b) Determining Sampling Instant at Zero Crossover point tails  
 (c) Impulse Spectrum

Equation 2a therefore reduces to

$$V_o(f) = H_T(f) H_R(f) \dots 3a$$

By making the transmit and receive filters to match such that the two transfer functions are equal, so that:

$$|H_T(f)|^2 = |H_R(f)|^2 = \frac{1}{2} \sqrt{\frac{R_{sym}}{2B}} \dots \dots \dots 4a$$

A filter that satisfy the above condition and that

$$|H(f)|^2 = \begin{cases} 1 & f < f_1 \\ \frac{1}{2} \left[ 1 + \cos \left( \frac{\pi}{2} \left( \frac{R_{sym}/2 - f}{B - R_{sym}/2} \right)^2 \right) \right] & f_1 < f < R_{sym}/2 \\ 0 & f > R_{sym}/2 \end{cases} \dots \dots \dots 5a$$

Then such filters are Square Root Raised Cosine filters or simply Root Raised Cosine (RRC) Filters. The frequency  $f_1$  and B are determined by the symbol rate and a design parameter called Roll-off Factor ( $\alpha$ ) and

$$0 \leq \alpha \leq 1 \dots \dots \dots 6a$$

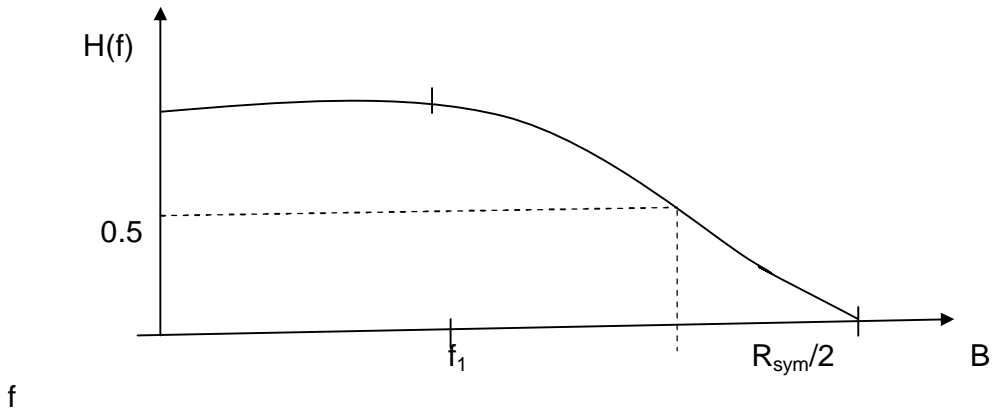


Figure 4A Frequency Response of Root Raised Cosine Filter

The frequency response of Root Raised Cosine Filter is shown in figure 4A

$$|H(f)|^2 = \frac{1}{2} \sqrt{\frac{R_{sym}}{2B}} \dots \dots \dots 7a$$

and

$$V_i(f) = \frac{1}{T} \sum_{n=-\infty}^{\infty} V_n(f) \text{ sinc}(f - n/T) \quad .8a$$

The above derivations are applicable to baseband digital transmissions with  $\delta(t)$  impulses as input to meet the sinc x spectrum requirements. However, in digital wireless satellite link, the baseband need to be modulated and transmission is conducted in passband. In such cases the baseband input is typically Non Return to Zero (NRZ) whose spectrum is given by  $V_i(f)$ :

$$V_i(f) = \frac{1}{T} \sum_{n=-\infty}^{\infty} V_n(f) \text{ sinc}(f - n/T) \quad .9a$$

Therefore, an equalizer must be use in the transmitter to reverse the spectrum to the condition for zero ISI, as shown in figure 5A

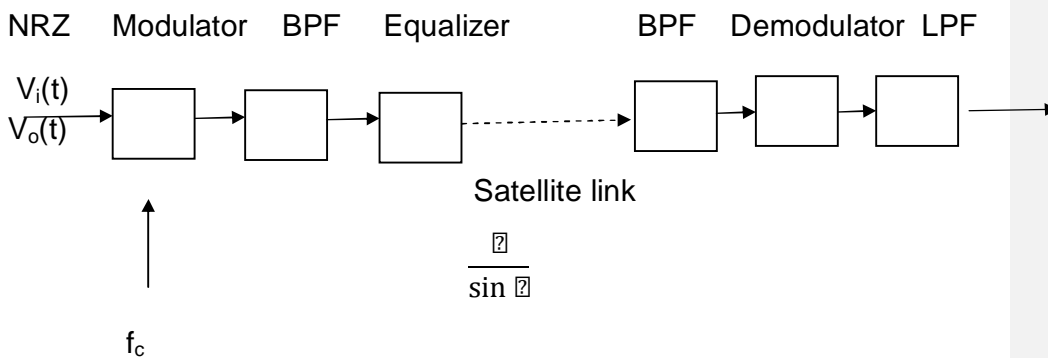


Figure 5A Transmission Path of Modulated NRZ pulses for Wireless Satellite Link

the spectrum of such equalizer is given as :

$$V_e(f) = \frac{1}{T} \sum_{n=-\infty}^{\infty} V_n(f) \text{ sinc}(f - n/T) \quad .10a$$



So that equation 1a becomes

$$H_{\text{RRC}}(f) = \frac{H_{\text{LP}}(f) H_{\text{LP}}(f - f_c)}{H_{\text{LP}}(f - f_c) H_{\text{LP}}(f_c - f)} \times H_{\text{LP}}(f) \times \frac{H_{\text{LP}}(f_c - f)}{H_{\text{LP}}(f_c - f)} \times H_{\text{LP}}(f_c - f) \dots \dots \dots .11a$$

or

$$H_{\text{RRC}}(f) = H_{\text{LP}}(f) H_{\text{LP}}(f_c - f)$$

Thus Nyquist criteria is applied as in baseband transmission and Root Raised Cosine Band Pass Filters (BPF) are used in the Transmitter and Receiver.

The transfer function of Root Raised Cosine Band Pass Filter is equal to that of two single sided (baseband) low pass filter with the centre frequency shifted from zero to carrier frequency.

$$H_{\text{RRC}}(f) = H_{\text{LP}}(f) + \frac{H_{\text{LP}}(f)}{2} + H_{\text{LP}}(f - f_c) - \frac{H_{\text{LP}}(f - f_c)}{2} - \frac{H_{\text{LP}}(f_c - f)}{2} - H_{\text{LP}}(f_c - f) \dots \dots \dots .12a$$

or

$$H_{\text{RRC}}(f) = H_{\text{LP}}(f) + H_{\text{LP}}(f_c - f) \dots \dots \dots .13a$$

### The Effect of Nyquist Filter on Spectral Efficiency

Equation 2.60 and 2.61 are theoretical spectral efficiency upper bounds for digital modulation schemes, in practice intersymbol interference (ISI) limits these theoretical upper bounds as mentioned in chapter two. The effect of ISI phenomenon can be displayed on oscilloscope in Eye diagram as shown in figure. The ISI causes the Eye to close thereby reducing the margin for additive white noise and increasing sensitivity to error in carrier synchronisation. To reduce the ISI, Nyquist filters are used to filter the digital baseband and the modulated signal. The Root Raised Cosine (RRC) satisfied the Nyquist criteria as highlighted in chapter two is usually split into two (transmit and receive ) sections as used in block models of chapter four. The relationship between Symbol rate  $R_s$  and bandwidth  $B_{\text{occ}}$  occupied by modulated signal passed through Nyquist filter is given by [ 2]

$$B_{\text{occ}} = R_s + B_{\text{guard}}$$

where  $R_s$  is symbol rate and  $\beta$  is filter roll-off factor with its range values at  $0 < \beta < 1$ .

But

$$B_{\text{RRC}} = (1 + \beta) R_s$$

Therefore equation 14a becomes

$$\frac{B_{\text{RRC}}}{R_s} = \frac{(1 + \beta) R_s}{R_s} = 1 + \beta$$

$$\therefore \frac{B_{\text{RRC}}}{R_s} = (1 + \beta) \dots \dots \dots 15$$

From above the RRC filter increases the bandwidth and decreases the spectral efficiency by factor of  $(1 + \beta)$ . As  $\beta$  decreases, the ISI becomes more significant causing more error, a value of  $\beta = 0.3$  is considered to be sufficient [14] and adopted to mitigate ISI in this work.

## APPENDIX B

### High Power Amplifier (HPA) Analysis

Travelling Wave Tube commonly used power amplifier on big communications Satellites on GEO are not applicable to small LEO

satellites, instead Solid State Power Amplifier (SSPA) are used as High Power Amplifiers (HPA) because of the weight and heat dissipation of the former. Power amplifiers are classified according its linearity, which decreases as its efficiency increases as shown in table 1B.

Table 1B Classification of Power Amplifiers

AMPLIFIER CLASS	MAXIMUM EFFICIENCY	CONDUCTION ANGLE CYCLE	LINEARITY
A	50%	2π	Linear
AB		< 2π	Partial Linear
B	78%		Partially Linear
C	100%	<	Non-Linear
D	100%		Non-linear

The non-linearity in power amplifier causes:

- i) Spectral spreading or regrowth of the amplified signal which may lead to adjacent channel interference.
- ii) Intermodulation effects where several (Multiplexed) channels are amplified by same HPA.
- iii) Interference between I and Q phase components causing phase distortion.

In a memoryless non-linear power amplifier, the output is a function of input at given time instant or fixed delay. A simple carrier modulated input signal  $x(t)$  of the form

$$x(t) = A(t) \cos(\omega_c t + \theta(t)) \quad (1)$$

when applied to a memoryless non-linear HPA such as SSPA, the output signal  $y(t)$  produced is given as

$$y(t) = g[A(t)] \cos(\omega_c t + \theta(t) + \phi(t)) \quad (2)$$

Where  $g[A(t)]$  is gain distortion function and  $\phi(t)$  is the phase distortion function, representing output amplitude distortion and output phase deviation due to increase of input beyond the amplifier compression point. The two expressions above can also be presented in complex envelopes without the carrier components as: [16 ]

$$r(t) = A(t) \cos(\omega_c t + \theta(t)) \rightarrow r(t) = A(t) \cos(\omega_c t) - A(t) \sin(\omega_c t) \theta(t) \dots \dots 3$$

The two distortion parameters  $g[A(t)]$  and  $\theta(t)$  are usually described as AM-AM conversion and AM-PM conversion respectively. The two distortion effects are more significant on QAM- where the modulating parameters are both Amplitude and Phase than in PSK . where the modulating parameter is Phase only. Detailed analysis on the effect of non-linearity on MPSK and MQAM are beyond the scope of this research work [17].

**Page 76: [1] Formatted** user 1/30/2016 3:33:00 PM

Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.34", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Page 76: [2] Formatted** user 1/30/2016 3:33:00 PM

Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.34", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Page 76: [3] Formatted** user 1/30/2016 3:33:00 PM

Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: 0.11", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Page 76: [4] Formatted** user 1/30/2016 3:33:00 PM

Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: 0.11", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Page 76: [5] Formatted** user 1/30/2016 3:33:00 PM

Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: 0.11", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Page 76: [6] Formatted** user 1/30/2016 3:33:00 PM

Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: 0.11", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Page 76: [7] Formatted** user 1/30/2016 3:33:00 PM

Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: 0.11", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Page 77: [8] Formatted** user 1/30/2016 3:33:00 PM

Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.35", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Page 77: [9] Formatted** user 1/30/2016 3:33:00 PM

Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.35", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Page 77: [10] Formatted** user 1/30/2016 3:33:00 PM

Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.36", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Page 77: [11] Formatted** user 1/30/2016 3:33:00 PM

Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.36", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Page 77: [12] Formatted** user 1/30/2016 3:33:00 PM

Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.36", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Page 77: [13] Formatted** user 1/30/2016 3:33:00 PM

Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.36", Relative to: Paragraph, Horizontal: 0.13", Wrap Around

**Page 77: [14] Formatted** user 1/30/2016 3:33:00 PM

Justified, Line spacing: single, Position: Horizontal: Right, Relative to: Margin, Vertical: -1.36", Relative to: Paragraph, Horizontal: 0.13", Wrap Around