# A COMPARATIVE ANALYSIS OF THE POTENTIAL OF ROBUST CODING SCHEMES IN EMERGING WIRELESS TECHNOLOGIES

# BY

# LIVINGSTONE LISTONE KALUUBA Dipl.-Ing.[Comm.Eng.](Karlsruhe), B.Sc.[EE]Hons.(Mak).

A THESIS SUBMITTED IN FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN COMMUNICATIONS ENGINEERING OF MAKERERE UNIVERSITY, UGANDA

## **DECEMBER 2010**

#### DECLARATION

I declare that, except where specifically acknowledged, the work contained in this thesis is my original work. It is being submitted for the Degree of Doctor of Philosophy of Makerere University. It has not been submitted before for any degree or examination at any other University.

#### Livingstone Listone Kaluuba

(Name of Candidate)

.....

(Signature)

## 28<sup>th</sup> day of December, 2010

#### DEDICATION

In remembrance of my late parents: **Janet Matama Nabirye** and **Enock Bagalira**, who although themselves had received minimal education, just being able to read and write, toiled, brought me up and supported my childhood, which enabled me to become what I am today.

#### ACKNOWLEDGEMENT

I would like to express my profound gratitude to my Ph.D supervisors Prof. Dr. Livingstone Luboobi and Dr. Tabani-Wani Gyavira for their invaluable guidance and continuous encouragement during the entire period of this research work. Their technical acumen, precise suggestions and timely discussions whenever I was in some problem is whole heartedly appreciated.

I would like to thank Prof. P. Mugambi who was instrumental during the phase of my choosing a premier mathematical and yet practical application of mathematics in emerging wireless technologies.

I am thankful to Makerere University Administration for having granted me a waver of tuition fees during the period I spent doing my research as a postgraduate student and full-time lecturer in the Department of Electrical Engineering. I would also like to thank the Carnegie Foundation for its financial assistance which enabled me to acquire the necessary equipment and software for this research.

I would like to thank all fellow members of staff in the faculty of Technology and those in the Department of Mathematics and Physics at Makerere University with whom I often had some academic exchanges and valuable advice.

Thanks to my wife Florence for her patience and to my family as a whole for the encouragement always extended to me with assurance that the task had to be accomplished.

## TABLE OF CONTENTS

DECLARATIONii			
DEDI	DEDICATIONiii		
ACKN	ACKNOWLEDGEMENTiv		
TABL	E OF CONTENTS	v	
LIST	OF FIGURES i	X	
LIST	OF ACRONYMS AND ABBREVIATIONS	ĸi	
ABST	RACT	1	
CHAP	TER ONE	2	
INTRO	DDUCTION	2	
1.1	BACKGROUND	2	
1.2 CHAN	CLASSICAL METHODS USED TO MITIGATE WIRELESS INEL PROBLEMS	3	
1.2. 1.2.	INCREASING SIGNAL POWER     DECREASING SIGNAL NOISE	.3 .4	
1.2.	3 INTRODUCING DIVERSITY [43]	.4	
1.3	ROBUST CODING SCHEMES	6	
1.4	THE CONCEPTUAL-THEORETICAL FRAMEWORK	6	
1.5	STATEMENT OF THE PROBLEM	9	
1.6	RESEARCH APPROACH 1	0	
1.6. 1.6. 1.6.	1       OBJECTIVES       1         2       METHODOLOGY       1         3       JUSTIFICATION       1	0 1 1	
1.7	SIGNIFICANCE OF THE STUDY 1	1	
1.8	SCOPE OF THE RESEARCH 1	2	
1.9	THESIS OUTLINE 1	2	
CHAP	TER TWO 1	4	
LITER	ATURE REVIEW 1	4	
2.1	INTRODUCTION 1	4	
2.2 COMN	INFORMATION THEORY: THE FOUNDATION OF MODERN MUNICATIONS 1	5	
2.2. 2.2.	1       CLAUDE SHANNON AND THE CHANNEL CAPACITY THEOREM	6 20	
2.3	FORWARD ERROR CORRECTING CODES 2	1	
2.4	SIGNAL QUALIFICATION AND QUANTIFICATION METRICS 2	5	
2.4. 2.4.	1       QUALITY METRIC       2         2       QUANTITY METRIC OR INFORMATION CONTENT       2	26 26	

<ul> <li>2.4.3 INFORMATION CONTENT OF A SOURCE</li> <li>2.4.4 INFORMATION CAPACITY OF A CHANNEL</li></ul>	28 28 29
2.5 INFORMATION-THEORETIC TOOLS	. 30
2.5.1The Binary Symmetric Channel2.5.2The Additive White Gaussian NoiseChannel	32 33
2.6 CHANNEL CODING PROCESSES	. 38
<ul> <li>2.6.1 WHY CHANNEL CODING IS NECESSARY</li></ul>	40 40 43 43 45
2.7 BLOCK CODES	. 47
<ul> <li>2.7.1 LINEAR BLOCK CODES</li> <li>2.7.2 BINARY LINEAR CODES</li> <li>2.7.3 HAMMING DISTANCE</li> </ul>	47 48 49
2.8 CONVOLUTION CODES	. 51
<ul> <li>2.8.1 The D-Transform and Generator Polynomials of Recursive Processes</li> <li>2.8.2 Recursive Systematic Convolution Encoders</li> <li>2.8.3 Constraint Length</li> </ul>	5.54 56 57
2.9 CYCLIC BLOCK CODES	. 59
2.10 SYSTEMATIC LINEAR CODES	. 61
2.11 EXAMPLES OF COMMON LINEAR CODES	. 62
<ul> <li>2.11.1 Hamming Codes</li> <li>2.11.2 The Parity Check Equation</li> <li>2.11.3 Hamming Bound</li> </ul>	63 64 66
2.12 BLOCK CODES AND THE GENERATOR MATRIX	. 69
2.13 STOCHASTIC DECODING METHODS	. 78
2.14 THE SEARCH FOR GOOD CODES	. 81
2.15 ROBUST CODES	. 86
2.16 THE ROLE OF COMPUTATIONAL SCIENCE: MODELING AND SIMULATION	) . 89
2.17 CHAPTER SUMMARY	. 91
CHAPTER THREE	. 92
WIRELESS CHANNEL MODELS	. 92
3.1 WIRELESS SIGNAL TRANSMISSION	. 92
<ul> <li>3.1.1 WIRELESS CHANNEL MODELING AND SIMULATION</li> <li>3.1.2 THE SIGNAL FADING PHENOMENON</li></ul>	94 97 98
3.2 THE WIRELESS CHANNEL MODEL	100
3.2.1 MATHEMATICAL MODELING OF THE WIRELESS CHANNEL	.101

3.2 3.2 3.2 SPR 3.2 3.2 3.2	.2 .3 .4 EAD AI .5 .6 .7	THE PHYSICAL BASIS OF MULTIPATH FADING THE MATHEMATICAL MODEL OF MULTIPATH FADING CHARACTERIZATION OF THE CHANNEL IN TIME AND FREQUENCY: DO ND TIME SPREAD FREE-SPACE POWER LOSS THE TWO RAY MODEL EMPIRICAL AND SEMI-EMPIRICAL MODELS	
3.3	WI	RELESS MOBILE CHANNEL DESIGN	116
CONS	SIDE	RATIONS	116
3.3 3.3	.1 .2	The Rayleigh Fading Model [20] The Rician Fading Model	117 118
3.4	NO	ISE AND INTERFERENCE MODELS	119
3.4	.1	Noise	120
3.5	WI	RELESS CHANNEL PERFORMANCE METRICS	124
3.5 3.5	.1 .2	BIT ERROR PROBABILITY WIRELESS CHANNEL DESIGN TRADEOFFS	126 126
3.6	LIM	IITATIONS OF CHANNEL MODELS	128
3.7	CH	APTER SUMMARY	129
CHAF	PTER	FOUR	131
THE	ΓURE	30 CODE CONCEPT	131
4.1	INT	RODUCTION TO TURBO CODES	131
4.2	HO	W TURBO CODES WORK	133
4.2 4.2 4.2	.1 .2 .3	The Role of the Interleaver Turbo Code Generation Processes Channel Model	134 136 139
4.3	TUF	RBO CODE DECODERS	139
4.4	THE	E CODING DILEMMA [28]	141
4.4 4.4	.1 .2	TURBO CODE PERFORMANCE FACTORS AND TRADEOFFS PRACTICAL ISSUES	
4.5	PEF	RFORMANCE OF TURBO CODES IN FADING CHANNE	LS144
4.6	TU	RBO CODE DESIGN COMPLEXITY	145
4.7 TEC⊦	TUI INOL	RBO CODE APPLICATIONS IN EMERGING WIRELESS	<b>;</b> 146
4.7 4.7 4.7 4.7 4.7 4.7	.1 .2 .3 .4 .5 .6	CELLULAR MOBILE RADIO COMMUNICATIONS DIGITAL VIDEO BROADCASTING LONG-HAUL TERRESTRIAL MICROWAVE LINKS MILITARY APPLICATIONS IMAGE PROCESSING WIRELESS LOCAL AREA NETWORKS (WLANS)	146 146 147 147 147 147
4.8	CH	AFIEK SUNINIAKY	14/

CHAPTER FIVE149			
INVES BASE	STIG. D GU	ATION OF RADIO PROPAGATION MODELS USING MATLAB- IS	149
5.1	INT	RODUCTION	149
5.2	LAR	GE SCALE MODELS	150
5.2. 5.2. 5.2.	1 2 3	THE FREE SPACE MODEL GUI THE OKUMURA-HATA MODEL GUI GUI FOR HF AND MF GROUND WAVE MODEL FOR URBAN AREAS	151 152 154
5.3	SMA	ALL SCALE MODELS	155
5.3. 5.3.	2	The Rician Fading Distribution GUI The Nakagami Fading Model GUI	158 161
5.4	LOC	G-NORMAL DISTRIBUTION	161
5.5	CHA	APTER SUMMARY	163
CHAP	TER	SIX	165
TURB	0 CC	DE SIMULATION EXPERIMENTS AND ANALYSIS OF RESULT	S 165
6.1	INT	RODUCTION	165
6.2	SIM	ULATION SETUP FOR TURBO CODES	165
6.2. 6.2. 6.2.	1 3 4	TURBO CODE SIMULATION SOFTWARE TURBO CODE PERFORMANCE AS A FUNCTION OF THE NUMBER OF ITERATIONS INTERLEAVER DESIGN CONSIDERATIONS	166 169 170
6.3 CHAPTER SUMMARY173			
CHAPTER SEVEN			
RESE. RECO	ARCI MME	H FINDINGS, OBSERVATIONS, CONCLUSIONS AND	175
7.1	RES	EARCH FINDINGS	175
7.2	OBS	SERVATIONS	175
7.2. GENI 7.2.	1 ERATIC 2	MERITS OF TURBO CODES COMPARED TO CONVOLUTION CODES FOR THIRD N WIRELESS SYSTEMS COMPARISON OF LOW DENSITY PARITY CHECK CODES AND TURBO CODES	175 177
7.3	CO	NCLUSIONS	178
7.4	REC	COMMENDATIONS	180
7.4. 7.4. 7.4.	1 2 3	WHY LDPC IS A STRONG CANDIDATE FOR EMERGING WIRELESS SYSTEMS CALL FOR ADDITIONAL DESIGN CRITERION FUTURE RESEARCH ENDEAVORS	181 181 182
REFERENCES			
APPENDICES			189

## LIST OF FIGURES

Fig. 1.1	The Conceptual -Theoretical Framework	Page
Eia 01	The Communication System Model	10
FIg. 2.1	The Generic Communication System Model	18
Fig. 2.2	Characteristics of a binary symmetric channel (BSC)	29
Fig. 2.3	Additive White Gaussian Noise Channel	
Fig. 2.4	Definition of code robustness	34
Fig. 3.1	Multipath Radio Signal Propagation	40
Fig. 3.2	Mathematical model of the wireless channel	45
Fig. 3.3	Sketch of signal arrival components.	48
Fig. 3.4	Diagram illustrating propagation of the signal assuming a two-ray	y
r	nodel	51
Fig. 3.5	N-path Mobile Channel Attenuation and Noise Model	63
Fig. 4.1	Coding Gain in AWGN Channels	78
Fig. 4.2	Convolution encoder with 3 memory registers, with 1 input and 3	output
b	pits	84
Fig.4.3 /	Another Convolution encoder configuration	86
Fig. 4.4	Combined feed-back and feed-forward encoder	
Fig. 4.5	Recursive systematic turbo code encoder	
Fig. 5.1	Serial Concatenated Encoding and Decoding	116
Fig. 5.2	A typical (generic) structure for generating turbo codes	117
Fig. 5.3	Linear Feedback Shift Register (LFSR)	120
Fig. 5.4	Turbo decoder	123
Fig. 6.1	A GUI for a free-space model	135
Fig. 6.2	Graphic User Interface for the Knife-Edge Model	136
Fig. 6.3	A GUI for the Okumura-Hata model	137
Fig. 6.4	A GUI for the new HF/MF Ground-wave model	138
Fig. 6.5	A GUI to simulate Rayleigh fading	141

Fig. 6.6 A more comprehensive simulation of Raleigh fading	142
Fig. 6.7 A GUI for simulating Rician Fading	143
Fig. 6.8 A GUI for simulating the Rician PDF and CDF	144
Fig. 6.9 A GUI for simulating Nakagami model distribution	145
Fig. 6.10 A GUI for Log-Normal fading distribution.	146
Fig. 7.1 AWGN and Rayleigh Fading Channel Models.	149
Fig. 7.2 Impact on performance for different information block lengths (frame-size) used.	151
Fig.7.3 Effect of varying the number of iterations, assuming a	
constant information block length of 4000	152
Fig. 7.4 Effect of different RSC encoder configurations on performance of the resulting turbo code	153
Fig. 7.5 Comparison of performance between the unpunctured and punctured turbo code generation configurations	154
Fig. 7.6 Performance dependence on code rate, information block size, and puncturing	154

# LIST OF ACRONYMS AND ABBREVIATIONS

ASIC	Application Specific Integrated Circuits
APP	A Posteriori Probability
ARQ	Automatic Repeat Request
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate or Bit Error Ratio
ВСН	Bose-Chaudhuri-Hocquenghen code
BSC	Binary Symmetric Channel
BPSK	Binary Phase Shift Keying
CCSDS	Consultative Committee for Space Data Systems
CD	Compact Disk
CDF	Cumulative Distribution Function
CD-ROM	Compact Disk- Read Only Memory
CDMA	Code Division Multiple Access
CDMA2000	Code-Division Multiple Access version of the IMT-2000 standard developed by the ITU
COST Research	European Cooperation in the Field of Scientific and Technical
DMC	Discrete Memory Channel
DSCS	Defense Satellite Communications System
3GPP	Third Generation Partnership Project
DVB	Digital Video Broadcasting: European standard for sound/video broadcasting to fixed and mobile devices
DVB-S2	Digital Video Broadcasting – standard 2

DVB-RCS	Digital Video Broadcasting Return Channel Satellite
DPSK	Dual Phase Shift Keying
EIRP	Effective Isotropically Radiated Power
$E_b/N_o$	Energy per bit divided by the one-sided spectrum noise density
EM	Electromagnetic
EVDV	Evolution- Data and Voice
ETSI	European. Telecommunications Standards Institute
EXOR	Exclusive-OR function
FDMA	Frequency Division Multiple Access
FSK	Frequency Shift Keying
FEC	Forward Error Correction
FECC	Forward Error Control Codes
FSK	Frequency Shift Keying
FSM	Finite-State Machine
GF	Galois Field
GSM	Global System for Mobile Communications
GUI	Graphic User Interface
HSDPA	High Speed Downlink Packet Access A 3G high-speed digital data service provided by cellular carriers worldwide that use the GSM technology, including AT&T.
ISI	Inter-Symbol Interference
IEEE	Institute of Electrical and Electronic Engineers
JTIDS	Joint Tactical Information Distribution System
JRSC	Jam Resistant Secure Communications
LOS	Line of sight
LDPC	Low Density Parity Check code

- LPI Low Probability of Intercept
- LFSR Linear Feedback Shift Register
- MAP Maximum-A-Posteriori
- MBS Mobile Broadband System
- MER Message Error Rate
- MIT Massachusetts Institute of Technology
- MHD Minimum Hamming Distance
- MHW Minimum Hamming weight
- MDS Maximum Distance Separable
- ML Maximum Likelihood
- MATLAB Matrix Laboratory Software
- NLOS Non-Line-of-sight
- P<sub>b</sub> Bit error probability
- P<sub>bl</sub> Block error probability
- PDF Probability Density Function
- $P_e$  Probability of error
- PCCCs Parallel Concatenated Convolution Codes
- PCS Personal Communications Systems
- QAM Quadrature Amplitude Modulation
- QPSK Quadrature Phase Shift Keying
- RAM Random Access Memory
- RSC Recursive Systematic Coding
- SCCCs Serially Concatenated Convolution Codes
- SER Symbol Error Rate

SOVA	Soft Output Viterbi Algorithm
S/N	Signal-to-Noise Ratio (also denoted as SNR)
SNR SNIR	Signal-to-Noise Ratio (also denoted as S/N) Signal-to-Noise-and-Interference Ratio
TCM	Trellis Coded Modulation
TURBO8.EXI	E Eight – state turbo code executable program
TV	Television
ТСМ	Trellis Coded Modulation
UCC	Uganda Communications Commission
UMTS	Universal Mobile Telecommunications Service: a standard for third-generation mobile networks.
VLSI	Very Large Scale Integration
W-CDMA	Wideband Code Division Multiple Access
WLAN	Wireless Local Area Network

#### ABSTRACT

This study investigates the critical design parameters for robust codes used for data transmission systems in a wireless environment. Robustness refers to the ability of a given code to withstand transmission channel impairments, and thus deliver a useable signal despite the unavoidable signal corruption of the channel. One of the major obstacles to the accurate delivery of information from the source to some destination is the unpredictable and random behaviour of the wireless communication channel. There are several factors/processes which influence the performance of the communication channel, namely: noise, distortion, shadowing, multipath fading, and many others. Most of these are stochastic in nature and we have often to deal with probabilistic variables. The main culprit of the channel impairments in a mobile environment, however, is multipath fading. Several methods exist for combating the multipath problem, such as space diversity, frequency diversity, polarization diversity, and adaptive modulation, but we believe that forward error coding is the most versatile method for mobile communications.

This thesis deals with the fundamental issues related to the recovery of a useful signal from the corrupted signal using the forward error correction coding technique in a wireless environment. This is only possible through the use of robust codes whose mathematical origin is rather interesting, extending from the binary number system, to information theory and finally coding theory. The thesis discusses the background to the wireless transmission problem, how it manifests itself in terms of path loss, shadowing, multipath attenuation, wave absorption, etc., mathematical modeling of these processes, multipath fading mitigation methods, and finally gives a comparative presentation of the critical design parameters of turbo codes as a representative example of robust codes.

From the error performance analysis and results, it is evident that turbo codes are quite suitable for the emerging wireless communications technologies and applications. However, low density parity check codes are preferred to turbo codes in some applications because of their more efficient implementation as well as better performance.

# CHAPTER ONE INTRODUCTION

#### **1.1 BACKGROUND**

Transmission of information from a source to a destination by wireless means involves many hidden issues, such as channel noise, interference, and fading phenomena. When the recipient of the message fails to decipher the true meaning intended, he/she realizes that the system he/she is dealing with is quite complex. Unfortunately, not everyone is in a position to diagnose that problem.

The last ten years have been marked by a phenomenal development in high-tech mobile and personal communications systems and their application on a world-wide basis. A whole range of new wireless services and products have been developed, spanning from low mobility indoor wireless local area networks (WLANs) to high mobility outdoor mobile broadband systems (MBSs). This development is the motivation for an investigation into the fundamental limits of data transmission over the wireless channel [73].

A common feature of all wireless operating environments is that, the received signal undergoes random variations; *the problem we wish to address deals with the difficulties encountered with the transmission of messages due to the inherent problems of noise and fading phenomena, which are associated with the transmission channel* : during the process of transmission, there is usually some altering of the message due to fading (weakening of the signal), sporadic electrical bursts (interference) and other naturally occurring noise that creeps into the wireless transmission medium.

Thus, a fundamental characteristic of wireless communications is that, the channel is time-varying. This occurs due to the mobility of the user or of objects in the

propagation path. In addition, multiple scatterers, such as buildings and forests cause the received signal to contain time-shifted versions of the transmitted signal. This delay spread translates into inter-symbol interference (ISI), in digital communication. We also note that cellular networks are dependent on a scheme of frequency reuse, whereby, similar sets of frequencies are reused by cells, which are geographically well separated in order to use the frequency spectrum more efficiently. However, this scheme has the unavoidable effect of co-channel interference due to interfering signals from outside the cell of interest. Therefore, an important aspect of wireless communication is to understand the impact of interference on reliable communications.

The solution to the problem is to insure that the intended message is obtainable from whatever is actually received. This presents wireless system designers with a challenge to design powerful error control techniques, that can minimize the effect of the hostile propagation conditions. Many different disciplines come together to successfully recover the corrupted signals: electronic engineering, computing and mathematics. This multidisciplinary interest in the problem explains why several approaches have been used, leading to useful contributions towards the solution to the problem of transmitting an information signal, through the hostile wireless environment.

Nevertheless, the problem still exists because the causes of corruption of the signal are random in nature and often very difficult to predict accurately.

#### 1.2 CLASSICAL METHODS USED TO MITIGATE WIRELESS CHANNEL PROBLEMS

#### 1.2.1 Increasing Signal Power

Increasing signal power is similar to talking louder in a noisy room. We intuitively know that as the signal to noise ratio is increased, the errors are bound to decrease. Assuming that we can do nothing to the environment, the most obvious thing we can do is to increase the transmitter power. However, examples of power-limited devices exist, such as mobile phones and satellites. Both of these have a fixed amount of power and can not increase their transmitting power beyond a certain point. In fact most are designed to operate at their maximum power and have no spare power available.

There are actually some disadvantages associated with using this scheme. When amplifier characteristics are not linear, increasing the power means both the signal and noise are amplified making the situation worse. You can see this mechanism at work in a radio. When you increase the volume, the noisiness of the signal goes from bad to intolerable [43].

#### 1.2.2 Decreasing Signal Noise

If we are at a party and the room is noisy for conversation, we may have to move to a quieter corner, where there is less noise. If you turn on the fan in the room and the TV reception goes bad, you turn off the offending item to improve the reception. These are some means of noise reduction. In a communication device, the only noise that is under the designer's control is thermal noise, and inter-modulation noise of the system. Assuming that, the system is designed to minimize these, we do not have any way of reducing any other external noise. We are at the mercy of the environment and have to accept this as it is. Nevertheless, we can do something to mitigate the effects of noise and other adverse effects as discussed in the following subsections.

#### 1.2.3 Introducing Diversity [43]

Various *diversity techniques* exist which can be employed to correct for burst errors. Examples include:

- Space diversity,
- Frequency diversity, and
- Time diversity.

All these systems have some redundancy associated with them, in that, data has to be transmitted twice or more times! This has cost implications, with reference to the wireless channel and terminal equipment.

When listening to a radio, you find that the signal is waxing and waning. Your instinctive response is to move the radio to a different location. With a mobile phone, we may switch channels, or ask the caller to call us on a different line. In satellite communications, in wet and rainy areas, we often have two ground stations separated by some distance, both receiving the same signal so as to increase the signal-to-noise ratio,(SNR). Different

polarizations which are used to expand the useable spectrum in satellite communications can also be used to send the same information as a means of error correction. All these techniques fall under the category of diversity, the main purpose of these is to improve signal quality by taking advantage of redundancy. Another form of diversity comes to use from an unexpected source in the case of mobile phones. The bouncing around of the signals causes amplitude reduction but with sophisticated signal processing, we can actually use these low-power signals to combine and improve the SNR [43].

In *space diversity* two or more antennas are used, each connected to a separate receiver. These antennas are sited sufficiently far apart to decorrelate fading at the outputs of the corresponding receivers. *Frequency diversity* is a technique whereby two different frequencies are used to transmit the same information. Frequency diversity can be in-band or out-of-band depending upon the distance between the carrier frequencies. In *time diversity* systems the same message is transmitted more than once. The most common diversity example is *dual diversity* (employing two antennas, two unique frequency allocations or two entirely separate transmissions of the same information). Over a wide range of applications a 3 dB signal-to-noise (SNR) improvement is achieved by use of dual diversity.

#### 1.2.4 Forward Error Correction

Multipath fading, interference, and noise effects severely degrade the average biterror rate, (BER), or probability of error, ( $P_e$ ), performance of a wireless digital radio transmission link. In order to achieve a highly reliable data transmission without excessively increasing both transmitter power and co-channel reuse distance, it is indispensable to adopt an auxiliary technique that can cope with the fast fading effect and the noise and interference associated with the channel.

When a duplex line is not available or is not practical, a form of error correction called *Forward Error Coding* (FEC) is used. The receiver has no real-time contact with the transmitter and can not verify exactly whether the received code is correct. It must make, however, a decision about the received data and do whatever it can to either fix it or declare it unsuitable.

#### **1.3 ROBUST CODING SCHEMES**

Ordinary use of the word *robustness* as commonly defined in dictionaries describes the state of being strong and healthy, full of vigour and hardy. In the context of communications, the usage is not too different [1]. *Robustness characterizes a signal's ability to withstand impairments from the channel, such as noise, jamming, fading, and so on.* A signal configured with multiple replicate copies, each transmitted on a different frequency (frequency diversity), has a greater likelihood of survival than does a single such signal with equal total power. The greater the diversity (multiple transmissions, at different frequencies, spread in time), the more robust the signal against random interference.

With reference to coding schemes, a code used to convey information via a corruptible channel, but with the ability of enabling recovery of a useful signal from the corrupted signal, is generally referred to as a *robust code*.

A given channel code is said to be robust if the transmitted message across a noisy or generally corruptible channel can be recovered from the received and often corrupted code word at the receiver with minimum effort. This definition does not formally exist anywhere in the recently published literature.

*Coding Strength*: It should be noted that there are not many codes which are capable of detecting and correcting all kinds of error. In other words, all error patterns cannot be correctly decoded. *The error correction capability of a code, hereby, referred to as the coding strength, can be investigated by first defining the physical structure of the code.* 

Modern emerging wireless digital communication systems have to adapt robust coding schemes to combat the effects of noise, interference, and multipath fading.

#### **1.4 THE CONCEPTUAL-THEORETICAL FRAMEWORK**

Much of modern communications rests on the well-established, but still vital disciplines of signal generation, modulation, coding, demodulation, detection, compression, equalization, signal processing, etc.. Figure 1.1 illustrates a conceptual

framework of the data transmission problem and our motivation for research in this area. The challenge to the design engineer is to ensure that the recipient at the destination is able to extract a useful signal from what ever is available at the output of the communication channel.



**Figure 1.1 The Conceptual - Theoretical Framework** 

Here our emphasis is on the propagation of the information signal from the source to the destination. We are aware that the communication channel between the two extreme ends is full of random and often time varying impairments. While we follow what happens to the signal from the input of the channel to the output of the channel, our main interest is the adaptation of suitable coding techniques, referred to as **robust codes**, to mitigate the problems of noise and multipath fading, which are actually a nuisance, as far as accuracy of the delivered information signal is concerned. The motivation for research into *robust codes* for wireless applications could not be carried out in isolation. *Robust codes are a product of coding theory, which is itself a subset of the wider knowledge area in mathematics, generally referred to as field theory.* **Source coding** and **channel coding** are two prominent subsets of the coding theory specialization. Each component has certain crucial applications in digital signal processing. Source coding is necessary because it is used to remove redundancy in the source signal before transmission which has great implications in saving transmission time, and conservation of space in storage media. Related topics in this area are the source coding theorem, Huffman codes, and data compression algorithms.

Channel coding facilitates recovery of a useful signal from a corrupted signal from the wireless communication channel. Channel coding is comprised of various categories of codes including *block codes*, *linear block codes*, *convolution codes*, *concatenated codes*, *Reed-Solomon codes*, and the more recently developed, *low density parity check codes* (LDPC), and *turbo codes*. Important performance metrics for these categories include: the *Hamming distance*, and *channel capacity*, taking into account the effects of noise, multipath fading propagation, and bandwidth limitations, *bit error rates*, and *coding gain*.

The salient properties of a robust code can only be arrived at after a thorough understanding of the major problems associated with the wireless communication channel. We are especially interested in the wireless channel, where the problems of noise and multipath fading phenomena are more pronounced and lead to the distortion of the useful signal.

Because of the gravity of this problem several approaches have been tried in practice, all geared to achieving a good signal at the output of a corrupted wireless communication channel. For instance:

- Noise mitigation through appropriate modulation techniques
- Flat fading mitigation through various type of diversity and adaptive antenna systems

• Intersymbol interference (ISI) mitigation through equalization, adaptive coding, multi-carrier modulation and spread spectrum techniques.

We are also cautious about traditional error-detecting approaches which assume error models and distributions. Such approaches may not be ideally suited for protection of data from some wireless devices. We therefore propose modifications of traditional methods which aim at providing uniform protection against all errors without (or which minimize) any assumptions on the error distributions.

In this thesis we propose forward error correction schemes based on a class of *nonlinear systematic error-detecting codes*, called *robust codes*. These nonlinear codes are robust in terms of equal protection against all errors. Examples of robust codes include turbo codes, low density parity check codes, Reed-Solomon codes, and Golay codes.

#### **1.5 STATEMENT OF THE PROBLEM**

Modern emerging wireless digital communication systems have to adapt robust coding schemes to combat the effects of noise, interference, and multipath fading, which lead to bit errors at the receiver.

An important question is:

#### What are the critical design parameters of a robust code?

In other words: Under what circumstances, if any, is it theoretically possible to design a channel coding system (encoder plus decoder) such that the overall transmission of information from a source to a user can become as reliable as we desire?

The fundamental problem is finding codes with both content and reasonable error handling ability. Whether this is possible, the answer is "yes". The existence of such codes is a consequence of the channel coding theorem from Shannon's 1948 paper [3]. Finding these codes is another question. Once we know that good codes exist, we

pursue them, hoping to construct practical codes that solve more precise versions of the *fundamental problem*. This is the quest of the coding theory.

#### **1.6 RESEARCH APPROACH**

#### **1.6.1** Objectives

In this thesis, we believe that the causes of stochastic distortions in the information signal can be overcome or minimized by the use of *robust coding techniques*, such that, despite the presence of some corruption in the received signal, the useful information can be derived from the corrupted signal.

An *error-correcting code* is a way of adding redundancy to information, so that a useful component can be recovered even if some of it is corrupted in transmission.

The overall objective of our research is to undertake a comprehensive study of code design, comparative performance analysis, coding and decoding methods, and emerging applications. The specific objectives of the study are:

- Investigation of the fading channel phenomena in mobile wireless communication systems and description of these phenomena using models.
- Investigation of various coding schemes employed in wireless transmission of digital data in order to establish the salient parameters that influence the wireless channel performance in the presence of noise and fading phenomena in a mobile environment.
- Evaluation of code performance, which leads to the identification of optimum codes for wireless channels.
- Establishment of the fundamental performance bounds and capacity limits at all levels and consideration of several design alternatives.
- To use simulation processes in order to compare our results with those obtained by other researchers using different approaches and methods.

Other issues include investigation of impact of channel performance on the output signal, noise and fading phenomena mitigation measures and fundamental channel capacity limits. The research has to a great extent fulfilled most of the above investigation objectives, and with the help of simulation using MATLAB software, we

were able to achieve satisfactory results comparable to those from other comparable studies elsewhere.

#### 1.6.2 Methodology

In order to be able to design channel codes capable of surviving the unpredictable corruption tendencies of the channel, the first step to take is to fully understand the stochastic behavior of the channel parameters and the main culprits contributing to the predicament. From the preliminary literature survey conducted earlier, it was established that scrupulous noise, multipath radio propagation, and intersymbol interference are the major culprits. This research is therefore focused on the analysis of signals in the noise environment, and multipath fading phenomena. The analysis is followed by simulation of various models, and comparison of our results with those results obtained by other researchers, who have carried out test-bed measurements in typical application environments. We then turn our attention to various categories of channel codes, with the aim of establishing the cardinal parameters, which can be optimized for a given code to be able to survive the onslaught of the channel impairments. This endeavor incorporates a thorough study of channel codes and simulation of the behavior of these codes under different scenarios of noise and multipath phenomena.

#### 1.6.3 Justification

Currently, some robust coding schemes do indeed exist. *Their potential at influencing the performance of wireless technologies has not been yet fully exploited. This research attempts to unravel the potential of robust coding schemes in facilitating overall performance improvement in emerging wireless technologies.* 

The research reviewed recent activities in coding theory and identified the salient design parameters of robust coding schemes and how these schemes can be adapted in emerging wireless technologies for improved performance.

#### **1.7 SIGNIFICANCE OF THE STUDY**

No electronic data transmission or storage system is perfect. Each system makes errors at a certain rate. As data transfer rates and storage densities increase, the raw error rate also increases. Error correction is needed to ensure the accuracy and integrity of data [2].

Information access by anyone anywhere has emerged as a strategic goal for a whole new wave of technologies, with networking being at the core. The implementation of the networking vision for the future – concerning seamless access to information by anyone -- requires a high speed/bandwidth wire-line infrastructure coupled to a wireless one, supporting a multitude of mobile end-users' access. The technology for maximizing the capacity and performance of wireless communication systems will be robust coding.

The rising demands for mobile phones and higher data rates introduce new challenging requirements on radio spectrum allocations. As the radio spectrum is an immensely scarce resource, there is an increasing realization that it needs to be used more efficiently. Existing radio technology cannot offer satisfactory link and network performance required for the high data rates in the third and fourth generations of cellular mobile systems. Currently, there are research activities around the world focused on expanding channel capacities in the future mobile communications. One of the most promising approaches in this area is the application of robust coding and modulation schemes jointly.

#### **1.8 SCOPE OF THE RESEARCH**

The research covered both theoretical and practical applications of robust coding schemes. However, greater emphasis is placed on the theoretical analysis and simulation of models, rather than on physical setups involving test beds for the various parameters of interest.

The research is focused on code design, performance analysis, coding and decoding methods, modeling and simulation techniques of transmission channels. We are interested in understanding the physical processes involved and how they influence the performance of a given code and hence the different measures to take in order to ensure the robustness of a given code in a given transmission environment.

#### **1.9 THESIS OUTLINE**

The research presented in this thesis is primarily concerned with aspects of reliable communication in the presence of partially known interference and channel fading phenomena. By analyzing the salient radio signal propagation characteristics of the wireless channel and examining the achievable performance of various transmission channel codes and detection schemes, we make some recommendations about robust communication structures in form of codes suitable for the wireless channel.

The thesis is organized as follows:

*Chapter One:* Comprises the introduction and presents the conceptual-theoretical framework.

*Chapter Two*: Reviews the major milestones of information theory, coding theory, and establishes the cardinal theoretical concepts on which the design of channel codes is based.

*Chapter Three*: Discusses wireless channel signal propagation phenomena, modeling and simulation aspects. It dwells at great length on various aspects including fading phenomena, signal propagation models, wireless mobile channel design issues, and analysis and modeling of noise and interference models.

*Chapter Four*: This chapter discusses recent advances in channel coding, with particular emphasis on the turbo code concept. Various cardinal parameters of turbo codes are identified and their mathematical relationships are derived.

*Chapter Five*: This chapter presents and discusses radio propagation simulation using graphic user interfaces developed using MATLAB. This simulation exercise aims at illustrating the propagation channel characteristics, such as fading and noise in the simplest way using a computer.

*Chapter Six*: This is a presentation of the turbo code simulation experiments and the analysis of results. The chapter demonstrates how transmission errors can be mitigated by varying various turbo code parameters.

*Chapter Seven*: presents the research findings, conclusions and recommendations, and proposals of future research work in this field.

# CHAPTER TWO LITERATURE REVIEW

#### 2.1 INTRODUCTION

Error-correcting codes are at the heart of most modern data transmission and storage systems, including wired and wireless, optical and magnetic recording systems. Our current research effort focuses on robust coding for emerging wireless data communication systems. The basic idea of coding is to add redundancy to data for transmission so that even in the presence of some noise and distortion introduced by the transmission channel or storage system, the original data can be recovered error-free.

A well-designed coding system adds the minimum amount of redundancy to achieve the desired level of robustness. Most traditional error correcting codes are based on algebraic techniques, i.e., the redundant information is added according to some cleverly chosen algebraic equations. These equations are then exploited at the decoder in order to recover the original information from the possibly noisy or corrupted received code word. The binary (7,4) Hamming code[4] and the (23,12) Golay code [5] are some of the classic algebraic codes examples, which are also some of the very few codes known as "perfect codes".

The purpose of this literature review is to show the connections between information theory, coding theory, and how these theories have been exploited in the development of robust coding techniques for reliable data transmission in a hostile wireless environment, namely a transmission channel faced with problems of multipath fading, interference from other radio signal sources and noise.

#### 2.2 INFORMATION THEORY: THE FOUNDATION OF MODERN COMMUNICATIONS

Information theory, also referred to as communication theory is a branch of mathematics that deals with the information content of messages. It is concerned with the amount of information and with the accuracy of its transmission. Information theory addresses two aspects of communication:

- "How we define and measure information", and
- "The maximum information, that we can send through a communications channel"

Information theory involves the quantification of data with the goal of enabling as much data as possible to be reliably communicated over a channel and/or stored in a medium.

The measure of data, known as *information entropy*, is usually expressed by the average number of bits needed for storage or communication.

Information theory presents the fundamental concepts and theorems which are at the very heart of modern communications and information technology. The results of information theory tell us [6]:

- How to quantify the information content in a set of data;
- How to model and analyze a wide range of communications channels and their capacity for transmitting information;
- Conditions under which error- free representation and transmission of information is possible, and when it is strictly impossible;
- Conditions for the design of good ways of (codes for) representing information so as to achieve data compaction and compression, and channel error robustness;
- Which minimal quality reduction we may expect for a given transmission rate, a given information source, and a given communications channel;
- How we may split our communication systems into subsystems, in order to simplify design without the loss of theoretical performance.

Coding theory is a branch of information theory concerned with finding explicit methods, called, **codes**, of *increasing the efficiency* and *reducing the net error rate of data communication* over an *impairments prone* channel. These codes can be

subdivided into two main sub-categories, namely *data compression codes* (associated with **source coding**) and *error-correction codes* (associated with **channel coding**). While source coding and channel coding are the fundamental concerns of information theory, this research is mainly focused on channel coding techniques.

Design engineers are interested in the basic concepts and the most famous theorems of information theory because they are very useful in explaining certain processes in the design and analysis of communication and information storage systems.

#### 2.2.1 Claude Shannon and the Channel Capacity Theorem

**Claude Shannon** is regarded as the father of **Information Theory** because he laid the foundations of Information Theory in his famous paper "The Mathematical Theory of Communication (1948)" [3] while working at Bell Labs in the United States. He thereby established himself as the creator of digital communications. His paper was a culmination of research on how fast the telegraph could operate. The main result of his formulations is that *the only way to get the most storage capacity in a storage device or the fastest transmission through a given communications channel is through the use of very powerful error-correcting systems*. In his monumental paper, Shannon showed roughly that up to a special measure called the channel capacity, it is possible to transmit information with an arbitrarily small probability of error by using *long enough* code words.

Unfortunately, Shannon used probabilistic techniques in his paper and did not provide any method for actually constructing such codes. We can appropriately, explicate Shannon's theory in common terms that if our information source is sending data at a rate less than what the communications channel can handle, we can add some extra bits to the data stream to push the error rate down to an arbitrarily low level! The down side of this is that communications delay is increased as we achieve lower and lower data error rates. However, for every situation, there are enough choices of channel coding that there is likely some satisfactory compromise between delay and error performance. Before Shannon's discovery, it was believed that channel noise (alone) prevented error-free communications. Shannon, however, showed that channel noise only limited the transmission rate, and not the error probability. Shortly after the publication of Shannon's work, many engineers and mathematicians got to work finding out how to create arbitrarily good communications links. Some of those techniques are discussed in this thesis.

Earlier on in 1928, Harry Nyquist had published his work in a paper on, "Certain Topics in Telegraph Transmission Theory", in which he observed that a channel's signaling rate was a function of the channel's bandwidth. Moreover, he proved that the signaling rate must be at least twice the bandwidth of the signal being transmitted. Nyquist also explored signal spacing and signal shaping, but he never looked at how interference affected the signals.

Shannon expanded the work of Nyquist to include interference, or noise. He postulated that the goal of a communications system was to get the signal through a noisy channel to the receiver. Noise is random, variable, and unpredictable, whereas distortion is not. Distortion can be measured and the channel conditioned to mitigate distortion's adverse effects. A multipath propagation phenomenon is quite similar to noise in that it is random in nature and unpredictable as well.

At the core of Shannon's work, however, is the equation for a channel's information carrying capacity:

*C* is the capacity of the channel in bits per second, *W* is the bandwidth of the channel in Hz, S is the signal power in watts, and *N* is the noise power of the channel in watts. *S/N* is the channel's signal-to-noise ratio. *The equation states that the balance between the power and the bandwidth drives the channel*. For example, increasing the bandwidth allows us to decrease the power while maintaining the same capacity. The power/bandwidth tradeoff is an important part of wireless communications.

Shannon also explored the effects of source compression and channel coding (error detection/correction codes) on information carrying capacity. The Shannon bound is

the minimum bit energy required for reliable data signal transmission, i.e.,  $\frac{E_b}{N_o} \ge \ln(2) = -1.59 dB$ , which is the absolute minimum signal energy to noise spectral density ratio required to reliably transmit one bit of information, assuming infinite amounts of bandwidth.

Other implications of the Shannon theorem (c.f. equation (2.1)) are

 $\lim_{N \to \infty} C = 0$ , assuming S is fixed, and C = W if S = N.

The first result is quite practical and plausible, but the second has no practical interpretation.

From the 1950s through the 1970s, Shannon's theorem was viewed as an engineering limit rather than an information carrying philosophy [8]. In the 1980s, when we moved to digital wireless to deal with the lack of cellular capacity, Shannon's work became the reality of wireless communications. In 2003, Calhoun [7] introduced the notion of post Shannon to categorize the relevance of Shannon's work to the capacity-challenged and difficult-to-manage wireless channels. Moreover, this new look at Shannon's work raises questions about the basic theory. For example, noise addition has proven to be beneficial in some environments.

Claude Elwood Shannon's work was referred to by Senator John D. Rockfeller in the US congress after Shannon's death, on February 24, 2001, as "The Magna Carta of the Information Age". Shannon's ideas first presented in his famous paper have been crucial in enabling the information and communication technological advances which have created today's information society. We think of Newton's Laws, Maxwell's Equations, and Einstein's Relativity as groundbreaking, enabling all sorts of advances [42]. Shannon's theory of communication is definitely of that order.

*The Channel Coding Theorem*: Let C(S) be the capacity of a memoryless channel at a cost *S*. For any information rate R < C(S), and for any desired reliability of transmission over the channel (as measured through probability of channel decoding error) there exists a channel code of rate *R* (information bits per channel symbol) such that the desired reliability or robustness may be obtained. For rates R > C(S) no such

codes exist, and there will always be a nonzero probability of decoding errors. Such robustness is essential for a wireless channel, just as it is for any good quality machine; properly building such robustness into the code is done by channel coding.

*Shannon's Noisy Channel Theorem* states that: For every channel there is a rate of transmission, C, called the capacity of the channel, such that, if it is acceptable to have a rate R of transmission less than C, then there exists an encoding scheme that will reduce the probability of decision error to any desired (low) level. We note that the channel coding theorem and Shannon's noisy channel theorem refer to the same thing.

#### Interpretation of Shannon's Channel Theorem

What was completely novel about this result, compared to the usual line of thinking before Shannon's papers, was that it showed that *the achievable transmission rate was not a function of the desired degree of communication reliability*. Either communication can be made reliable, or it cannot. In the case that it can, it can be made as reliable as we want: Methods for achieving any desired reliability have been proven to exist for transmission rates all the way up to the (constant) channel capacity – at the cost of increased system complexity[6].

Conversely, it has been proven that there are no transmission schemes to be found, which can guarantee any degree of reliability if one attempts to transmit above capacity.

The channel coding theorem effectively consists of two parts: a *direct part* which says that for a rate R < C there exists a coding scheme with arbitrarily low block and bit error rates as we let the codelength *n* tend to infinity, and a *converse part* which states that for  $R \ge C$  the bit and block error rates are strictly bounded away from zero for any coding scheme. The channel coding theorem establishes rigid limits on the maximal supportable transmission rate of an AWGN channel in terms of power and bandwidth.

The channel coding theorem holds for both discrete and continuous channels. It is, perhaps, chiefly an *existence* theorem; in other words, it *does not exactly tell us how to construct our channel coding systems* – beyond imposing necessary constraints on the channel symbol distribution. Several methods of channel coding are used in

practice including *block coding*, *convolution coding*, and *turbo coding*, and more specifically for data transmission the following code modulation techniques are used: Quadrature Amplitude Modulation, Frequency Shift Keying (FSK), Dual Phase Shift Keying (DPSK), and Quadrature Phase Shift Keying(QPSK) [18].

#### 2.2.2 The Generic Communication System Model

Central to the development of information theory is the notion of a *generic communication system model* which can be used as a unifying framework suitable for describing a wide range of real-world systems. In the generic model proposed by Shannon, information is transmitted from an *information source* to a *user*, by means of a *transmitter*, a *communications channel*, and a *receiver*. This is illustrated in Fig. 2.1 below.

Examples of possible information sources in this context are: human speakers, video cameras, musical instruments, microphones, loudspeakers, and computer keyboards. The transmitter and receiver perform information coding and decoding respectively, which means processing the messages generated by the information source in order to [6]:

- Represent (encode) the messages in a suitable way during transmission over the channel, and
- Regenerate (decode) the messages at the receiver end, with as little deviation from what was originally transmitted as required for the service under discussion.



Fig. 2.1 The Generic Communication System Model

The term transmission here is intended to cover transmission both in space (between two different locations) and in time (i.e., storage of data on an imperfect medium).

Examples of physical communication channels thus range from wireless channels such as satellite links, mobile radio channels, and broadcast channels, via wired links such as optical fibres and copper transmission lines, to magnetic storage media and CDs.

Prior to Shannon's seminal papers of the 40s and 50s, there had been simply no satisfactory way of modeling and analyzing the process of information generation, transfer, and reception from a transmitter to a receiver over a noisy communications channel – which actually is a generic description of how all practical communication systems work. With Shannon's introduction of a generic communication system model, his view of information as a probabilistic entity (sidestepping its actual semantic meaning), the insight that the process of information transmission is fundamentally stochastic in nature, and his invention of precise mathematical tools to give a complete performance analysis of his model, the door was suddenly opened to a much more fundamental understanding of the possibilities and limitations of communications systems.

A former contemporary of Shannon, David Slepian (1973) says, "Probably no single work in this century has more profoundly altered man's understanding of communication than C.E. Shannon's article, "A mathematical theory of Communication".

We, however, believe that information theory can be used to reveal a performance potential beyond what was previously thought possible, and aid in the design of systems realizing this potential. For example, Shannon's theory enabled scientists to design more efficient communication and storage systems by demonstrating the enormous gains achievable through coding, and by providing the intuition for the correct design of coding systems. The sophisticated coding schemes used in systems as diverse as deep-space communication systems, and home compact disk audio and video systems, owe their success to the insights provided by Shannon's theory.

#### 2.3 FORWARD ERROR CORRECTING CODES

Shannon showed [2] that every communication channel has a capacity, C, (measured in bits per second), and as long as the transmission rate, R, (also in bits per second) is

less than *C*, it is possible to design a virtually error-free communications system *using error control codes*. Shannon's challenge was to prove the existence of such codes. He did not manage to tell us how to find them. But nevertheless, he had set the ball rolling:

Motivation for this research emanates from this question. Our research is an investigation of the critical wireless channel code design parameters, which influence the performance under application of error-control coding techniques.

A major concern of channel coding is the control of errors so that reliable communications can be obtained, i.e., the output signal  $\tilde{s}$  is as close to the input signal s as possible. There are many coding schemes available. However, turbo coding is the most exciting and this is a potentially important development in recent years. The turbo code is capable of achieving near Shannon capacity performance [12].

After the publication of Shannon's famous paper [2], researchers struggled to find codes that would produce the very small probability of error that he predicted. It was not an easy task throughout the 1950s when only a few weak codes were found [9]. During the 1960s, more enthusiasm was gained in the subject, the research community split into two groups between the so-called algebraists who concentrated on a class of codes called *block codes* and the stochastics (or probabilistic), who were concerned with understanding encoding and decoding as a random/stochastic process.

The stochastics (probabilistic) eventually discovered a second class of codes, called *convolution codes*, and designed powerful decoders for them. In the 1970s [9] the two research groups agreed to cooperate. Their joint effort led to the development of several efficient decoding algorithms. This development led to some types of *algebraic coding* (linear block code) schemes which were most effective in combating "bursty" errors (errors that arrive in bursts).

Convolution coding is generally more robust when faced with random errors or white noise; however, any decoding errors occurring in the convolution decoder are more likely to occur in bursts. We note that a single error correction code does not always provide enough error protection with reasonable complexity, and therefore often two
or more convolution encoders have to be concatenated to create a much more powerful code.

Meanwhile, around the same period, a contemporary of Claude Shannon, **Richard Hamming** discovered and implemented a single-bit error-correcting code. Richard Hamming is referred to as the father of Coding Theory because he initiated concepts of error-correcting codes [2]. The basis of modern forward error correction coding techniques is, however, the five-page paper, "Polynomial Codes Over Certain Finite Fields", published in 1960 by Irving Reed and Gustave Solomon in the Journal of the Society for Industrial and Applied Mathematics. This paper was a major fundamental change in the way information could be handled during transmission and storage to ensure reliable reception of messages. Currently, Reed-Solomon codes are integral parts of several digital systems including CD players, digital audio tape, digital television, mobile phone systems, digital imaging systems, and so forth.

In 1960, researchers including Irving Reed and Gustave Solomon discovered how to construct error correcting codes that could correct for an arbitrary number of bit errors or an arbitrary number of "bytes", where a "byte" means a group of "eight" bits. Even though the codes were discovered at this time, there was no way known to decode the codes. Reed and Solomon then staff members at MIT's Lincoln Laboratory, introduced ideas that form the core of current error-correcting techniques. For instance, everything from computer hard disk drives to CD players and also Reed-Solomon codes (plus a lot of ingenuity) made it possible to receive the stunning pictures of the outer planets sent back by Voyager II. They also make it possible to recover almost perfect music to enjoy from a scratched compact disc today. It is also predictable that within a foreseeable future, these forward error control codes (FECC) techniques will enable the profit mongers of cable television to squeeze more than 500 channels into their systems.

Over the years, isolated groups of coding theory researchers looked at the computational aspects of error-correcting codes, beginning with a paper by Robert Gallagher [10] of MIT in the 1960s. But only with the technological revolution of the 1990s did computational issues come to the forefront of code design. During that time, French engineers devised *turbo codes*, which have extremely fast (linear-time)

encoding and decoding algorithms and, in practice, correct a large fraction of errors. At the time, however, rigorous analysis of the performance of these algorithms eluded researchers. Naturally, the new algorithmic focus of coding theory caught the attention of the theoretical computer science community. Soon than later, Michael Sisper and Danielman of MIT used so-called expander graphs, a combinatorial tool widely used in computer science, to construct codes whose decoding efficiency was intricately intertwined with their ability to error-correct.

In 1968, Elwyn Berlekamp and James Massey discovered algorithms needed to build decoders for multiple error-correcting codes. They came to be known as *the Berlekamp-Massey algorithms for solving the key decoding equation* [11]. It was later established by researchers after some years that the Berlekamp-Massey algorithm is after all, a variation of an ancient algorithm discovered in Egypt around 300BC by Euclid and known as the Euclid's extended algorithm for finding the greatest common divisor of two polynomials [11].

In 1974, Joseph Odenwalder combined the two coding techniques mentioned above to form a *concatenated code*, now referred to as a *turbo code*. In this arrangement, *the encoder linked together an algebraic code followed by a Convolution code*. Performance was further enhanced by using an interleaver between the two encoding stages to mitigate any bursts that might be too long for the algebraic decoder to handle. This particular structure demonstrated significant improvement over previous coding systems. In 1993, Claude Berrou and his associates perfected the turbo code and is currently the most powerful forward error-correction code [12].

The importance of turbo codes is that they enable reliable communications with power efficiencies close to the theoretical limit predicted by Claude Shannon. Since their introduction, turbo codes have been proposed for low-power applications such as deep-space and satellite communications, as well as for interference prone applications such as third generation cellular phone and personal communication services. The major objective here is to achieve maximal information transfer over a limited-bandwidth communication link in the presence of data-corrupting factors, like noise. The unprecedented revolution in the design and application of very large scale integrated (VLSI) circuits combined with the development of powerful computation algorithms realizable both in hardware and software culminated in the advent of cheap microelectronics, and finally decoders became practical as early as 1981, and the entertainment industry adopted a very powerful error control scheme for the then new compact disc (CD) players [13].

Today, error-control coding is used in many forms in almost every new military communications system including the Joint Tactical Information Distribution System (JTIDS) and the Jam Resistant Secure Communications (JRSC) system employed on the Defense Satellite Communications System (DSCS), and for civilian applications such as the Global System for Mobile Communications (GSM) wireless data transmission systems, UMTS, etc.

# 2.4 SIGNAL QUALIFICATION AND QUANTIFICATION METRICS

The impairments to wireless transmitted messages may differ a lot depending on the type of channel, but may include:

- thermal noise and atmospheric noise (additive noise);
- interference from other sources and system users;
- reflections and scattering of transmitted radio wave power during propagation through the terrain;
- signal attenuation due to path loss in the radio channels or transmission line resistance;
- inter-symbol interference due to lack of sufficient available bandwidth;
- Doppler shifts due to relative movements between receiver and transmitter;
- Non-linear effects, e.g. due to nonlinear power amplifier characteristics;
- Stains or scratches on a CD-ROM.

The aim of information theory is to model these impairments in a *quantitative* way, mainly by statistical models. The modeling is used to adduce the *qualitative* or *performance limits*, and to devise methods for efficient transmission of data over the channel – that is, *coding algorithms*.

The ultimate goal of the coding is to exploit the communication channel as well as possible. *In this endeavor, our major goal is to spend as little as possible of the limited physical resources at our disposal, including time, bandwidth, transmit power, or disk space - on the transmission or storage of information, in order to maximize the number of users, systems, or services that are able to share resources* [6]. We need at the same time, to ensure that the *quality* of the information retrieved at the receiver end is satisfactory.

# 2.4.1 Quality Metric

By "quality" of a received signal, we usually mean something like "degree of similarity to the transmitted message". The appropriate measure of, and typical demands on quality, is dependent on the type of information transmitted and on the application or service:

- For data communications, the criterion might be that the *average information bit error probability* (or bit error rate BER) should be less than, say, 10<sup>-9</sup> whereas for speech and video communications, the aural or visual quality perceived by human ears or eyes is the most important thing, and a higher BER can usually be accepted.
- The accepted perceptual quality range also vary with the application, and is different, e.g. for mobile telephony (low-to-medium quality application) and high fidelity audio (very high quality application).
- Real-time applications such as two-way speech communication also place constraints on average delay, buffering, probability of no transmission ("outage"), etc.

The single most important parameter to optimize for the FEC block is arguably the bit and/or symbol error rate, and we adopt this as our criterion for the goodness of an FEC system. We note however, that this is not necessarily the most meaningful measure in all cases. For example, if we consider pulse code modulated (PCM) speech, an error in the most significant bit is clearly more detrimental than an error in the least significant bit.

# 2.4.2 Quantity Metric or Information Content

The questions we are most likely to face in practice are usually, "What is the information content of a given source?" or "What is the capacity of a given channel?"

When discussing these issues it is useful to distinguish between two different perspectives:

- How to quantify the information content in the data produced by a source.
- How to quantify the information content transmitted from a source to a user by means of a communication channel.

Since generation of information is a stochastic process, an event has a low information content if its "future" can be forecasted with a high degree of accuracy, based on knowledge of its "past". In this case an observer's *uncertainty* about the future of the event is low, which means he will not receive much new information by continued observation of the event.

Thus, an event's information content is intimately linked to the a priori degree of randomness and uncertainty associated with the event: The more predictable an event is, the less knowledge we need to describe or predict it; thus the less information we receive by observing it.

Conversely, if an event is highly unpredictable, a detailed observation of its actual development is needed in order to describe it, which means that its information content is high. This holds regardless of the physical content of the event in question – sometimes referred to as the semantic meaning of the information.

The above can be restated as saying that, the actual semantic meaning in a message is of no consequence for the amount of information carried by the message. All that matters, is the degree of predictability; i.e., its statistical properties.

In a communication context, at least from the second of the two perspectives, the "events" under study are usually information messages passed from a source to a user. The degree of uncertainty on the receiver side, and the a priori predictability of the messages, will then depend on the statistical properties of the information source under study, the impairments introduced by the physical communications channel, and the way the transmitter and receiver are designed.

#### 2.4.3 Information Content of a source

The output of an information source can be modeled as a random process. The information content of this source is described by its *entropy* and is defined as

$$H(s) = -\sum p_j \log_2 p_j \qquad (2.2)$$

with j = 1, 2, ..., k, and H(s) measured in *information bits per source symbol* with  $s = \{x_1, x_2, ..., x_k\}$ , such that  $p(x_j) = p_j$ , the probability of occurrence of symbol  $x_j$ . The source entropy as defined above is a measure of average information per symbol produced by the source. The entropy of a source provides a fundamental bound on the number of bits required to represent that source for full recovery. In other words, the average number of bits per source output required to encode a source for error-free recovery can be made as close to the entropy of the source as we desire, but can not be less than the entropy.

### 2.4.4 Information Capacity of a Channel

The amount of *useful source information received by a user* is referred to as the *mutual information* between the source and the user. It is important here to emphasize the *usability* of the information received, as the "*total information content*". Mutual information is one of the most fundamental information theoretic concepts. It can be used to describe not only the capacity properties of communication channels, but also the compression possibilities for a given source when a certain amount of error (distortion) is accepted in the source representation.

Consider two given discrete, possibly statistically dependent random variables,  $X \in \{x_1, x_2, ..., x_k\}$  and  $Y \in \{y_1, y_2, ..., y_k\}$ . We can think of X as the input to, and Y as the output from a communication channel; i.e., Y is a noisy version of X. X and Y have probability distributions  $\mathbf{p} = [p_1, ..., p_k]^T$  and  $\mathbf{q} = [q_1, ..., q_k]^T$  respectively.

The conditional probability distribution of  $x_j$ , given  $y_k$ , is denoted  $P_{j|k}$ .  $P_{j|k}$  will then describe the probability of  $x_j$  being transmitted if  $y_k$  is observed at the receiver.

The *mutual information* between X and Y is now defined as

$$I(X,Y) = H(X) - H(X | Y),$$
 (2.3)

where H(X) is the entropy of the source that outputs X, and, by definition,

$$H(X | Y) = \sum_{k=0}^{K-1} q_k$$
.  $H(X | y_k)$  of some information – in this case Y.

It is simply a measure of the average information content (uncertainty) which is left in X when Y is known.

i.e., 
$$H(X | Y) = -\sum_{j=0}^{J-1} \sum_{k=0}^{K-1} q_k P_{j|k} \log_2 P_{j|k}$$
 .....(2.4)

H(X | Y) can be interpreted as "the entropy of X when Y is observed". It is a conditional entropy, which means that it is based on the a priori knowledge of some information - in this case Y. It is simply a measure of the average information content (uncertainty) which is left in X when Y is known.

Mutual information is simply a measure of how much information can be obtained about one random variable by observing another. The mutual information of X relative to Y (which represents conceptually the average amount of information about X, that can be gained by observing Y) is given by equation (2.3) above.

The mutual information I(X, Y) can be thought of as the information Y gives about X – the average reduction in the observer's uncertainty about X, which is brought about by observing Y. If X is the input to, and Y is the output from a given communications channel, H(X|Y) can then be thought of as information "lost" by the channel.

We note that the choice of logarithmic base in Shannon's equations for information content and mutual information, determines the *unit* of *information entropy* that is used. The most common unit of information in current use is the *bit*, based on the *binary logarithm*.

## 2.4.5 Channel Capacity of a Memory less Channel

A discrete-memoryless channel (DMC) is completely described by its input and output alphabets, and the channel transition probability matrix. One special case of a DMC is the binary symmetric channel (BSC) that can be considered as a mathematical model for binary transmission over a Gaussian channel with hard decision at the output. This channel is discussed in detail in section 2.5.1.

Most physical channels used to transmit information are subject to noise and distortion of various kinds, resulting in errors in the received waveforms/channel symbols. For simplification purposes, let us consider *memoryless* channels, which transmit discrete- or continuous-valued symbols in *discrete* time intervals. *Memorylessness* of a channel means that the added noise at a given time instance does not influence the channel output in any *other* time instances.

Let us also introduce the notion of a memoryless channel with a channel capacity at a cost S. The most obvious "cost" in a communication system is perhaps an upper limit on the average symbol power available for transmission over a channel. Another possible cost is that of bandwidth. For simplicity, we consider the case where the bandwidth is given, and thus is not a subject for optimization. The capacity at cost S is still defined in terms of a maximum of the mutual information I(X, Y) between the channel input and the output, as

$$\max_{X \in X_S} I(X, Y)$$

bits per channel use, where,  $X_S$  is the set of all possible channel symbol distributions (discrete or continuous depending on the channel) such that the average cost per channel use, E[s], is less than or equal to the constant *S*.

### 2.5 INFORMATION-THEORETIC TOOLS

Harnessing information-theoretic tools for the investigation of wireless channels has not only resulted in enhanced understanding of the potential and limitations of those channels, but also in fact provided on many occasions the right guidance to the specific design of efficient communications systems.

The rapid advancement in technology and the exploding demand for efficient highquality and volume of digital wireless communications over almost every possible media and for a variety of applications (such as cellular, personal, data networks, etc.) reveals the dramatic role information theory has been playing in this trend. Channel capacity is one of the main design parameters of a wireless channel, and it is the most important performance measure.

We also use information theoretic tools to establish how the structure of fading channels can affect code design. Application of fading channel equalization techniques is only possible after thorough understanding of the theoretic basis of the multipath channel behaviour.

In the recent past, the world has experienced a rapid and tremendous growth in digital communications, especially in the fields of cellular and personal communication services, satellite, and computer communication. In these communication systems, information is represented as a sequence of binary bits. The binary bits are then mapped (modulated) onto analog signal waveforms (carriers) and transmitted over a communication channel. The wireless communication channel is characterized by the multipath signal propagation problem and noise and interference are added, leading to the corruption of the transmitted signal. At the receiver, the channel-corrupted and transmitted signal has to be mapped back to binary bits.

The received binary information is an estimate of the transmitted binary information. Bit errors may result due to the transmission and the number of bit errors depends on the amount of noise and interference, and the impact of the multipath phenomena in the wireless communication channel.

Channel coding is often used in digital communications to protect the digital information from the effects of multipath phenomena, noise and interference and thus reducing the number of bit errors. Channel coding is mostly accomplished by selectively introducing redundant bits into the transmitted information stream. These additional bits facilitate detection and correction of bit errors in the received data stream and provide more reliable information transmission.

The cost of using channel coding to protect the information is a reduction in data rate or an expansion in bandwidth. According to Shannon's capacity theorem, the capacity of a given communication channel depends on the bandwidth of the channel, and the signal-to-noise ratio of the system. The maximum rate at which information bits can be transmitted along the wireless channel will depend on the capacity of that channel. That is why information theory is relevant in order to be able to apply informationtheoretic tools to the digital communication problem.

Communications over a channel, such as the wireless channel or Ethernet cable, is the primary motivation of information theory. We are all familiar with such channels, however, such channels often fail to produce exact reconstruction of a signal; noise, outage (periods of silence), and other forms of signal corruption often degrade quality. How much information can one hope to communicate over such a noisy or rather imperfect channel? Under such stochastic constraints, we would wish to maximize the amount of information or the signal we can communicate over that channel. The appropriate measure for this is the mutual information and maximum mutual information is referred to as the *channel capacity* as expressed in equation (2.1).

## 2.5.1 The Binary Symmetric Channel

The simplest example of a communication channel is the *binary symmetric channel* (*BSC*). This channel, which models well e.g. the storage on a CD, or an optical fibre with binary modulation, takes binary symbols (denoted 0 and 1, for simplicity) as input and transmits them with a symmetric probability of error, denoted p. That is to say, the probability of a transmitted 0 being received as a 1 is p, as is the probability of a 1 being received as a 0. Note that p is limited to [0,0.5] since, if p were higher than 0.5, the bit error rate could be improved by inverting all the received bits.

For a *binary symmetric channel*, the maximum mutual information or channel capacity is given by the expression

$$C = l + p \log_2(p) + (1-p) \log_2(1-p), \qquad (2.5)$$

[information bits per channel symbol].

If p = 0.01, then  $C \approx 0.919$ .

Unfortunately, Shannon's noisy channel does not indicate how the good codes cannot be obtained in practice. The probability of error can be made arbitrarily small at the expense of the rate of transmission (R = k/n). *Other Desirable Qualities for Encoding Schemes*: Apart from probability of decision error, to be as small as possible, we desire to have encoding schemes which [1]:

- are easy to implement and lead to easy decoding
- lead to code words of moderate lengths (c.f. repetition codes which require very long code words to achieve small decision probability of error).

The higher the probability p is (up to 0.5), the more noise there is on the channel, and the smaller the capacity is.



Fig.2.2 Characteristics of a binary symmetric channel (BSC)

## 2.5.2 The Additive White Gaussian Noise Channel

Another important channel model is the band-limited additive white Gaussian noise (AWGN) channel with an input power constraint. A schematic model for this channel is shown in Fig. 2.3 below.



s(t): Input signal; H(f): Channel transfer function, n(t): Noise; y(t): Output signal

#### Fig. 2.3: Additive White Gaussian Noise Channel

If we consider a *memoryless continuous-amplitude* channel with *additive Gaussian zero-mean noise (AWGN)* of power (variance) *N*, we have the so called *Gaussian memoryless channel*, and the noise is statistically independent of the channel input. The noise power is uniformly distributed in frequency (hence *white* noise), while the noise samples follow a Gaussian distribution.

This *noise model* is valid for all cases where there are many independent sources of noise; in particular, it models thermal noise in electronic components well. In cellular mobile radio systems, with many active users transmitting in the same frequency band, it is also a good approximation when modeling interference between users.

We can assume further that the channel is *bandlimited* to W Hz and that the signals to be sent are critically sampled, i.e., at  $f_s = 2W$  Hz. This is a good model e.g. for a wireless mobile telephone channel. For such a channel, Shannon derived the following famous formula for the capacity at received signal power S (which is the same as transmitted power if, as usually done for this channel model, we assume that no signal attenuation is present in the channel):

 $C(S) = W \cdot \log_2(1 + SNR) W \cdot \log_2(1 + SNR), \dots (2.6)$ 

measured in information bits per second.

SNR denotes channel signal-to-noise ratio; i.e., the ratio between received signal power and noise power.

The weakness of this expression for C(S) is that it does not take into account the practical fact that *thermal* (*white*) noise power in a communication system is proportional to the system bandwidth. Thus, operating at the same SNR at different bandwidths means that the *transmit power* is different for each bandwidth.

Shannon's theorem, is a statement in information theory that expresses the maximum possible data speed that can be obtained in a data channel because according to this theorem, *the highest obtainable error-free data speed, expressed in bits per second* (*bps*), *is a function of the bandwidth and the signal-to-noise ratio*.

Among several other remarkable and fundamental insights, Shannon proved that it is possible to communicate without error (or with an arbitrarily low error rate) on a noisy channel as long as just one requirement is met: *the special ratio*  $E_b/N_0$  *must be greater than* ln(2), *the natural logarithm of 2. Numerically, that is about 0.693, or* 

-1.6 decibels (dB) in electrical engineering terms. This number is now known naturally enough as the Shannon bound. This is a theoretical limit; any real receiver will always do worse. No matter how clever you are, and no matter how much computer power you have, you cannot exceed the Shannon limit and communicate without error.

A more fair comparison [6] from a resource point of view is obtained by normalizing the SNR with respect to the bandwidth, i.e., expressing the SNR as the ratio of transmitted signal power S to noise power per Hertz bandwidth, denoted by  $N_o$ [W/Hz]. Doing this yields the capacity of an AWGN channel as follows:

$$C = B \log_2\left(1 + \frac{S}{N_o B}\right) \qquad \text{[information bits per second]} \dots (2.7)$$

which implies a finite (assuming finite transmit power) asymptotic upper bound on capacity as the bandwidth is increased to infinity: i.e.,

$$C(B - - > \infty) = \frac{S}{N_o \ln(2)}$$
. (2.8)

Proof:

$$C = B \log_2 \left( 1 + \frac{S}{N_o B} \right) = \left( \frac{S}{N_o} \right) \left( \frac{N_o B}{S} \right) \log_2 \left( 1 + \frac{S}{N_o B} \right)$$
$$= \left( \frac{S}{N_o} \right) \log_2 \left[ \left( 1 + \frac{S}{N_o B} \right)^{(N_o B/S)} \right].$$

Let:

$$x = S/N_o B$$

so that

$$C = \left(\frac{S}{N_o}\right) \log_2\left[\left(1+x\right)^{1/x}\right];$$

since

$$\lim_{x \to 0} (1+x)^{1/x} = e_{x}$$

we have

$$C(B \to \infty) = \left(\frac{S}{N_o}\right) \log_2 \left[\lim_{x \to 0} (1+x)^{1/x}\right]$$
$$= \left(\frac{S}{N_o}\right) \log_2 e = \left(\frac{S}{N_o}\right) \frac{\log_e e}{\log_e 2} = \left(\frac{S}{N_o}\right) \frac{1}{\log_e 2},$$

or

$$C (B \rightarrow \infty) = \left(\frac{S}{N_o}\right) \frac{1}{\log_e 2} = 1.44 \frac{S}{N_o}.$$

The *Shannon limit* for a given noisy channel, yields the maximum rate at which error-free communication is possible.

For a desired actual transmission rate  $R \le C$  [information bits/second] this can be used to obtain a lower bound on the *transmit energy per information bit* which must be used if error-free transmission is to be at this rate. The energy spent per channel symbol is

$$E_b = \frac{S}{R}$$
 [Joule/information bit].

Thus,

$$S = E_b R$$
,

so, from equation (2.7)

$$R/B \leq C/B = log_2\left(1 + \frac{E_b R}{N_o B}\right),$$
 (2.9)

Or,

$$E_b \geq N_o \frac{2^{R/B} - 1}{R/B}.$$
 (2.10)

The absolute lower bound for error-free transmission over the channel using any transmission scheme is obtained from this equation by letting the actual transmission rate go to zero:

Using L'Hopital's rule

or -1.6 dB on the decibel scale.

Shannon's channel capacity theorem states that the channel capacity of a continuous channel of bandwidth *B* Hz, perturbed by bandlimited Gaussian noise of power spectral density  $N_o/2$ , is given by

$$C(S) = B \cdot \log_2\left(1 + \frac{S}{N}\right)$$
 bits /s .....(2.12)

where S is the average transmitted signal power and the average noise power

where f is the instantaneous signal frequency and  $N_o$ , the spectral noise density is constant.

If B = 3 kHz and S/N is maintained at 30dB for a typical telephone channel, the channel capacity C(S) is about 30 kbits/s. According to Claude Shannon, every channel has associated with it a capacity, C(S), measured in bits per second (modulated symbol). Channel capacity is an upper bound on information rate, r. There exists a code of rate r < C(S) that achieves reliable communications. Reliable implies an arbitrarily small error probability.

The theorem implies that error-free transmission is possible if we do not send information at a rate greater than the channel capacity. Thus, *the channel capacity theorem defines the fundamental limit on the rate of error-free transmission for a power limited, band-limited Gaussian channel.* The essential elements of Shannon's formula are:

- Proportionality of capacity to bandwidth B
- Signal power S
- Noise power *N*, and
- The logarithmic function.

The channel bandwidth sets a limit to how fast symbols can be transmitted over the channel. The signal to noise ratio (S/N) determines how much information each symbol can represent. The signal and noise power levels are, of course, expected to be measured at the receiver end of the channel. Thus, the power level is a function both of transmitted power and the attenuation of the signal over the transmission medium (channel). The choice of the logarithm base determines the unit of information entropy that is used, here being the *bi*nary digit or **bit**.

Early space probes like Mariner used a type of error-correcting code called a block code, and more recent space probes use Convolution and turbo codes. Error-correcting codes are also used in CD players [13], high speed modems, and cellular phones. Modems use error detection when they compute checksums, which are sums of the digits in a given transmission modulo some number. The International Standard Book Number (ISBN) used to identify books and the Universal Product Codes (bar codes), also incorporate a check digit.

# 2.6 CHANNEL CODING PROCESSES

In data communications, *coding* is used for controlling transmission errors induced by channel noise or other impairments such as fading and interference, so that error-free communication can be achieved. In data storage systems, coding is used for controlling storage errors (during retrieval) caused by storage medium defects, dust particles and radiation so that error-free storage can be achieved. Ideally, we would like to communicate with the least probability of error and at a substantially high speed (rate).

Coding theory uses mainly algebraic and geometric tools to contrive efficient codes for various situations. The major objective of coding theory is to enable the transmission of data over a noisy channel with as few errors as possible. Redundancy is introduced in the codes to facilitate the detection and/or correction of errors. In digital electronic systems, information is carried by signals with discrete amplitudes so that the individual message bits can be more easily distinguished from one another. But when this digital information needs to be stored or transmitted over long distances or at very high speeds, noise and other hazards become an issue. Sources of such unwanted hazards include thermal noise due to motion of electrons, damage to the storage media, multipath fading or coupling from other energy sources.

In communication systems, one way to decrease the probability of bit errors is to increase the power of the transmitted signal until it is much higher than the noise. But the amount by which the signal power can be increased may be limited by the rating of the electronic circuits in the transmitter and even at the receiving end. This power level may also be regulated, as in the case of radio signals whose levels are specified by the Uganda Communications Commission (UCC) here in Uganda.

So, clearly we need some other means of controlling the probability of error. *Forward error correction* (FEC), or *channel coding*, provides this added dimension. By adding redundant symbols to the transmitted or stored digital information, we can achieve not only a means of error detection, but error correction as well.

An *error-correcting code* is an *algorithm* for expressing a sequence of numbers such that any errors which are introduced can be detected and corrected (within certain limitations) based on the remaining numbers. The study of error-correcting codes and the associated mathematics is known as *coding theory*.

Error detection is much simpler than error correction, and one or more "check" digits are commonly embedded in credit card numbers in order to detect mistakes.

Coding is achieved by adding properly designed *redundant digits* (bits) to each message. The redundant digits (bits) are used for detecting and/or correcting transmission (or storage) errors. Time diversity or in-band frequency diversity can be used to improve a receiver's performance. Diversity is a brutal force use of redundancy in which each bit or symbol is repeated several (n) times. From the coding point of view, diversity involves the use of a trivial code of rate 1/n. Using a

sophisticated code like the turbo code leads to a more efficient system while maintaining the benefits of the diversity concept.

# 2.6.1 Why channel coding is necessary

Since the beginning of the 1980s great strides made in the development of large scale integrated circuits led to tremendous growth in digital communications especially in the fields of cellular/PCS, satellite, and computer communication. In these communications systems, the information is represented as a sequence of binary bits. The binary bits are then mapped (modulated) to analog signal waveforms and transmitted over a communication channel. The communication channel introduces noise, multipath phenomena and interference, which corrupts the transmitted information signal. At the receiver, the corrupted signal has to be mapped back to the original message. However, due to the corruption experienced over the channel, the received binary information is often an approximation of the transmitted binary information. Bit errors may occur due to the channel impairments and the number of bit errors depends on the extent of this problem.

Channel coding is often used in digital communication systems to protect the digital information from noise and interference and to reduce the number of bit errors. Channel coding is mostly accomplished by selectively introducing redundant bits into the transmitted information stream. The additional bits facilitate the detection and correction of bit errors in the received data stream and provide more reliable information transmission and storage. The cost of using channel coding to protect the information is a reduction in data rate and/or an undesirable expansion in the required bandwidth.

# 2.6.2 Types of Channel Codes

There are several types of channel codes. The first major classification is *block codes versus convolutional (Trellis) codes*. Each of these groups can be further divided into *linear* versus *non-linear* codes. Linear codes of interest are encoded using the methods of linear algebra and polynomial arithmetic. Alternatively, we talk of *block codes* versus *convolution codes*. A *block code* is a type of *channel coding* that adds *redundancy* to a message so that, at the receiver, one can decode the code with minimal (theoretically zero) errors. Block codes operate in a block-by-block fashion and each code word depends only on the current input message block.

The main characterization of a block code is that it is a *fixed length* channel code. Typically, a block code takes a k-digit information word, and transforms this into an n-digit code word. The (n-k) parity bits of a linear block code are linear combination sums of the k message bits.

The first block code was developed by Richard W. Hamming in 1947 and is called the *Hamming code*. It actually consists of a whole class of codes with the following characteristics:

- Block length:  $n = 2^m 1$ ; m is the number of parity check bits
- Information bits:  $k = 2^m m 1$
- Parity check bits: n k = m
- Correctable errors: t = 1

These conditions [42] are true for m > 2. For example, with m = 4, there are n = 15 total bits per block or code word, k = 11 information bits, n - k parity check bits, and the code can correct t = 1 error. A representative Hamming code example is

1 00101001 01 0010, where, the four bits on the right (0010) are the parity checkbits. By choosing the value of m, we can create a single error correcting code that fits our block length and correction requirements. This one is customarily denoted as a (15,4) code, telling us the total number of bits in a code word (15) and the number of inormation bits (4).

The *Golay code* is another block code, more powerful than the Hamming code, and geometrically interesting. This is a (23,12) code discovered by Marcel J.E. Golay in 1949. It may also be extended using an overall parity bit to make a (24,12) code. Its minimum distance is seven, so it can detect up to six errors, or correct up to t = (7-1)/2 = 3 errors. There is one aspect of the Golay and Hamming codes that makes them interesting, they are basically *perfect*. With any of these codes, the code words can be considered to reside within spheres packed into a region of space. The entire space is GF(2<sup>m</sup>). Each sphere contains a valid code word at its centre and also all the valid code words that correct to the valid code word, those being a distance of three or fewer bits from the centre in the case of the Golay code (t = 3). If there are orphan [42] binary words outside the spheres, then the code is termed *imperfect*.

The *BCH code* is a block code discovered by Bose and Chaudhuri (1960), and independently by Hocquenghem (1959). BCH codes are multiple error correcting codes and a generalization of the Hamming codes. Possible configurations of BCH codes for  $m \ge 3$  and  $t < 2^{m-1}$  are:

- Block length :  $n = 2^m 1$
- Parity check bits :  $n k \le mt$
- Minimum distance :  $d \ge 2t + 1$

The code words are formed by taking the remainder after dividing a polynomial representing the information bits by a generator polynomial. The generator polynomial is selected to give the code its characteristics. All code words are multiples of the generator polynomial [42].

A convolution code (alternatively referred to as a linear Trellis code) is a forward error-correction scheme, whereby the coded sequence is algorithmically achieved through the use of current data bits plus some of the previous data bits from the incoming stream. Linear Trellis codes are known as convolution codes because the code sequence can be viewed as the discrete-time convolution of the message sequence with the impulse response of the encoder.

Linear codes are defined by a linear mapping over an appropriate algebraic system, such as *Galois Fields*, from the space of input messages to the space of output messages. Galois fields make use of mathematical constructs known as finite fields. This algebraic structure allows significant simplification of encoding and decoding equipment.

A convolution code is generated when:

- Each k-bit information symbol (each k-bit string) to be encoded is transformed into an n-bit symbol, where k/n is the code rate (n >= k) and
- The transformation is a function of the last m information symbols, where m is the constraint length of the code.

Convolution codes operate on streams of data bits continuously, inserting redundant bits used to detect and correct errors. Block codes differ from convolution codes in that the data is encoded in discrete blocks, and not continuously [42]. The basic idea employed is to break the information to be transmitted into chunks, appending redundant check bits to each block, these being used to detect and correct errors. Each data plus check bits block is called a code word. *A code is linear when each code word is a linear combination of one or more other code words*. This is a concept originating from linear algebra and often the code words are referred to as vectors for that reason.

Another characteristic of some block codes is a cyclic nature. This means that any cyclic shift of a code word is also a code word. So linear, cyclic, block-code code words can be added to each other and shifted circularly in any way, and the result is still a code word.

# 2.6.3 Coding For Wireless Channels [18]

Coding allows bit errors introduced by transmission of a modulated signal through a wireless channel to be either detected or corrected by a decoder in the receiver. Coding can be considered as the embedding of signal constellation points in a higher dimensional signaling space than needed for communications. By going to a higher dimensional space, the distance between points can be increased, which provides for better error detection and correction.

There is a difference between codes designed for purely AWGN channels and for fading channels. Codes designed for AWGN channels do not typically work well on fading channels since they cannot correct for long error bursts that occur in deep fading. Normally, a code designed for an AWGN channel only, has to be modified by combining it with some *interleaving* in order to make it suitable for fading channels. Thus, the criterion for the code design has to change to provide for fading diversity. Other coding techniques to combat performance degradation due to fading include *unequal error protection codes* and *joint source and channel coding*.

### 2.6.4 Code Design Parameters

The basic design parameters for both additive white Gaussian noise (AWGN) and fading environments include *minimum distance*, *coding gain*, *bandwidth expansion*, and *diversity order*. While code designs for AWGN environments are not directly

applicable to fading channels, codes for fading channels and other codes used in wireless systems (e.g. spreading codes in CDMA) require the background in code design fundamental techniques for AWGN channels.

We also need to understand the theory of concatenated codes and their evolution to turbo and low density parity check (LDPC) codes for AWGN channels. These, rather extremely powerful codes, exhibit near-capacity performance with reasonable complexity levels [18].

Normally, a code designed for an AWGN channel only, has to be modified by combining it with some *interleaving* in order to make it suitable for fading channels. Thus, the criterion for the code design has to change to provide for fading diversity [18].

Interleaving is a practical way of enhancing the error correcting capability of a given code. It is a process of rearranging the ordering of a data sequence in a one to one deterministic format. Code designs for fading channels combine block and convolution codes with interleaving, and modify the coding process to provide for maximum fading diversity.

Although diversity gains can also be obtained by combining coded modulation with symbol or bit interleaving, bit interleaving is generally preferred because it provides much higher diversity gains. Coding combined with interleaving provides diversity gain in the same manner as other forms of diversity, with the diversity order built into the code design. The interleaver spreads out bursts of errors over time, so it provides a form of time diversity. This diversity is exploited by the inherent diversity in the code.

*Unequal error protection* is an alternative to diversity in channel fading mitigation. In these codes bits are prioritized, and high priority bits are encoded with stronger error protection against deep fades. Since bit priorities are part of the source code design process, unequal error protection is a special case of joint source and channel coding error mitigation.

#### 2.6.5 Code Design for Error Correction

The main reason why we apply error correction coding in a wireless system is to reduce the probability of bit or block error. The errors may be due to noise or due to fading phenomena or any other causes.



Fig. 2.4 Coding gain in AWGN channels

The *bit error probability*,  $P_b$  for a coded system is the probability that a bit is decoded in error. The *block error probability*  $P_{bl}$ , also called the *packet error rate*, is the probability that one or more bits in a block of coded bits are decoded in error.

Block error probability is useful for packet data systems where bits are encoded and transmitted in blocks. The amount of error reduction provided by a given code is typically characterized by its *coding gain* in AWGN and its *diversity gain* in fading channel environments.

Coding gain in AWGN is defined as the amount by which the SNR can be reduced under the coding technique for a given  $P_b$  or  $P_{bl}$ .

Fig.2.4 illustrates the probability of bit error curves for uncoded and coded cases. The gain  $C_{g1}$  at  $P_b = 10^{-4}$  is less than the gain  $C_{g2}$  at  $P_b = 10^{-6}$ , and there is negligible coding gain at  $P_b = 10^{-2}$ .

The coding gain in AWGN channels is generally a function of *the minimum Euclidean distance of the code*, which *equals the minimum distance in the signal space between code words or error events*.

Thus codes designed for AWGN channels maximize their Euclidean distance for good performance.

Error probability with or without coding tends to fall off with SNR as a waterfall shape at low to moderate SNRs as depicted in Fig.2.4. While this waterfall shape holds at all SNRs for uncoded systems, coded systems exhibit error floors as SNR grows. The error floor also, kicks in at a threshold SNR which depends on the code design. For SNRs above this threshold, error probability falls off much more slowly, due to the fact that minimum distance error events eventually dominate code performance in this SNR regime [18].

#### Code Imperfectness

We define the *imperfectness* of a given code as the difference between the code's required  $E_b/N_o$  to attain a given word error probability  $(P_w)$ , and the minimum possible  $E_b/N_o$  required to attain the same  $P_w$ , as implied by the sphere-packing bound [18] for codes with the same block size k and code rate r.

The performance limit corresponding to the sphere-packing bound [18] would be reached with equality only if the code were a *perfect* code for the AWGN channel, i.e., if equal-size cones could be drawn around every code word so as to completely fill *n*-dimensional space without intersecting. Note that perfectness for the unconstrained-input AWGN problem requires that the entire continuum of *n*-dimensional Euclidean space be filled by these non-intersecting cones, not just the discrete points that might be occupied by binary code words. Thus, under this definition, even the (7,4) Hamming code and the (23,12) Golay code, which are rare examples of perfect binary codes, do not qualify as perfect codes for the unconstrained-input AWGN channel. Indeed, Shannon mentions in his 1959 paper that such codes only exist if k=1 or n=1 or 2.

# 2.7 BLOCK CODES

If a source message of k digits is mapped into a *structured sequence* referred to here as a channel code or code word of n digits (n > k), the mapping operation is called *block coding*. If the encoding operation is independent of the past encodings, it is said to be memoryless. The collection of all possible code words from the source is called a *block code* in this case.

One major characteristic of block codes is that they are based rigorously on finite field arithmetic and abstract algebra. They can be used to either detect or correct errors. Block code encoders accept a block of k information bits and produce a block of n coded bits. Predetermined rules are used to add n - k redundant bits to the k information bits to form the n coded bits. These codes are commonly referred to as (n, k) block codes. Examples of some of the commonly used block codes are Hamming codes, Golay codes, BCH codes, and Reed-Solomon codes (use non-binary symbols). There are many ways of decoding block codes and estimating the k information bits.

*Algebraic coding* (also known as block coding) was the only type of forward errorcorrection coding in use when Claude Shannon published his seminal paper, "Mathematical Theory of Communications", in 1948. With this technique, the encoder intersperses parity bits into the data sequence using a particular algebraic algorithm to identify and correct any errors caused by channel corruption.

## 2.7.1 Linear Block Codes

A block code is called a linear code when the mapping of the k information bits to the n code word symbols is a linear mapping. Linear block codes are defined simply as a subspace of a vector space. Binary linear codes are subspaces of a vector space over the finite Galois field, GF(2). These codes are the most widely used and can be represented in several ways. The most useful representation is through a parity-check matrix (see section 2.12 later). A parity check matrix is a matrix whose rows generate a subspace that is orthogonal to the subspace that represents the code.

Linear block codes are conceptually simple codes that are basically an extension of single-bit parity check codes for error detection. A single-bit parity check code uses one extra bit in a block of n data bits to indicate whether the number of 1s in a block

is odd or even. Thus, if a single error occurs, either the parity bit is corrupted or the number of detected 1s in the information bit sequence will be different from the number used to compute the parity bit: in either case the parity bit will not correspond to the number of detected 1s in the information sequence, so the single error is detected. Linear block codes extend this notion by using a larger number of parity bits to either detect more than one error or correct for one or more errors. Unfortunately, linear block codes, along with Convolution codes, trade their error detection or correction capability for either bandwidth expansion or a lower data rate.

## 2.7.2 Binary Linear Codes

Let us restrict ourselves to the simpler case of binary codes, where the original information and the corresponding code consist of bits taking a value of either 0 or 1, and also assume that data is given in form of blocks with symbols taken from an alphabet **A**, i.e.,  $\mathbf{A} = \mathbf{Z}_2$  for a binary system and the blocks will be in form of bit strings (binary codes).

A binary block code generates a block of *n* coded bits from k information bits [18]. We call this an (n,k) binary code. The coded bits are also called *code word symbols*. The *n* code word symbols can take on  $2^n$  possible values corresponding to all possible combinations of the *n* binary bits. We select  $2^k$  code words only from these  $2^n$  possibilities to form the code, such that each *k* bit information block is uniquely mapped to one of these  $2^k$  code words.

Assume the following notation: For  $k \ge 1$ , denote  $\mathbf{A}^k$  as the set of strings  $a_1, \dots, a_k$  of length k of symbols from  $\mathbf{A}$ .

The message can be thought of as a string  $\mathbf{a} = a_1, \dots, a_k$  from  $\mathbf{A}^k$ . "Redundancy" is added by "encoding"  $\mathbf{a}$  into a string  $\mathbf{c} = c_1, \dots, c_n \in \mathbf{A}^n$  of some longer length n > k in a *certain* way. We call  $\mathbf{c}$  a *code word*.

A block code is called a linear code when the mapping of the k information bits to the n code word symbols is a linear mapping. The set of all binary *n*-tuples  $A^n$  is a vector space over the binary field, which consists of two elements 0 and 1. The binary field is characterized by two operations: binary addition (modulo 2 and standard

multiplication. A subset S of  $A^n$  is called a subspace if it satisfies the following conditions [18]:

- 1. The all zero vector is in *S*.
- 2. The set S is closed under addition, such that if  $S_i \in S$  and  $S_i \in S$ , then  $S_i + S_i \in S$ .

An (n,k) block code is linear if the  $2^k$  length-n code words of the code form a subspace of  $A^n$ . Thus, if  $C_i$  and  $C_i$  are two code words in an (n,k) linear block code, then  $C_i + C_i$  must form another code word of the code.

### 2.7.3 Hamming Distance

Hamming distance is a fundamental parameter associated with an [n, M] – block code C, where M refers to the number of members or elements in C. Before we can define the Hamming distance for a code, we must define the Hamming distance between two code words.

**Definition**: The Hamming distance  $d(\mathbf{a}, \mathbf{b})$  between two code words  $\mathbf{a}$  and  $\mathbf{b}$  is the number of coordinate positions in which they differ.

The Hamming distance d(<u>a</u>, <u>b</u>) between two strings

$$\underline{\mathbf{a}} = a_1 a_2 \dots a_n,$$
$$\mathbf{b} = b_1 b_2 \dots b_n$$

а.

both  $\in A^n$  is defined as the number of entries j such that  $a_i \neq b_i$ .

**Definition**: For the [n, M] - block code  $C \in A^n$ , the Hamming distance d for C is defined as the minimum of  $d(\mathbf{a}, \mathbf{b})$  over all  $\mathbf{a}, \mathbf{b} \in \mathbf{C}$ , with  $\mathbf{a} \neq \mathbf{b}$ .

Alternatively, the Hamming distance d of the block code C is

 $d = \min\{d(\mathbf{a}, \mathbf{b}): \mathbf{a}, \mathbf{b} \text{ belong to } \mathbf{C}, \mathbf{a} \neq \mathbf{b}\}.$ 

In other words, the Hamming distance of a code is the minimum distance between two distinct code words, over all pairs of code words in C. Example: For the [5, 4] – binary code C = [00000, 10101, 01011, 11110], the Hamming distances between distinct code words are;

d(00000, 10101) = 3; d(00000, 01011) = 3;d(00000, 11110) = 4; d(10101, 01011) = 4;d(10101, 11110) = 3; d(01011, 11110) = 3;

Thus C has Hamming distance d = 3.

If the size of C gets larger, this method of computing the Hamming distance between each of the (M  $\|2$ ) pairs of code words of an [n, M] – code becomes very time – consuming. In this case, a more efficient method is available to deal with linear codes and uses a *parity check matrix* as discussed in section 4.3.2.

We can also define the Hamming distance between two code words  $C_i$  and  $C_j$ , denoted as  $d(C_i, C_j)$  or  $d_{ij}$ , as the number of elements in which they differ:

$$d_{ij} = \sum_{l=1}^{n} C_{i}(l) + C_{j}(l) \qquad (2.14)$$

where  $C_m(l)$  denotes the l-th bit in  $C_m$ . Since the Hamming distance between any two code words equals the weight of their sum, we determine the minimum distance between all code words in a code by just looking at the minimum distance between all code words and the all zero code word.

Intuitively, the greater the distance between code words in a given code, the less chance that errors introduced by the channel will cause a transmitted code word to be decoded as a different code word and therefore, *the minimum distance of a linear block code is a critical parameter in determining its probability of error*.

Some properties of the Hamming distance on A<sup>n</sup> are:

- $d(\underline{a}, \underline{b}) \ge 0$  with equality if and only if  $\underline{a} = \underline{b}$ .
- $d(\underline{a}, \underline{b}) = d(\underline{b}, \underline{a})$  for all  $\underline{a} = \underline{b}$ .
- $d(\underline{a}, \underline{b}) + d(\underline{b}, \underline{c}) \ge d(\underline{a}, \underline{c})$  for all  $\mathbf{a}, \underline{\mathbf{b}}, \underline{\mathbf{c}}$  (the Triangle Inequality).

Any function satisfying these three properties is called a *metric*, so the Hamming distance d is a *metric*.

If we are encoding k – bit strings (messages) into n – bit code words then we call R = k/n information bits per code word symbol, the *rate*, or *rate of transmission* of the code, and n - k, the *redundancy* of the code. If we assume that code word symbols are transmitted across the channel at a rate of  $R_s$  symbols/second, then the information rate associated with an (n,k) block code is  $R_b = R.R_s = (k/n)R_s$  bits/second. Thus we see that block coding reduces the data rate compared to what we obtain with uncoded modulation by the code rate R.

We note that as *n* increases, the redundancy of the code increases, however, the rate decreases, and error correction capability increases. This implies that improving error correction properties may be at the expense of a drop in efficiency. The *code to be used* will need to be chosen after considering the *cost* and *error correction needs*.

Note: The properties of concern in using codes include:

- Efficiency
- Cost of transmitting data, and
- Error detection/correction capabilities.

Rate (R) and redundancy give a rough measure of the first two properties.

### 2.8 CONVOLUTION CODES

Convolution encoders process the incoming bits in streams rather than in blocks. The paramount feature of convolution codes is that the encoding of any bit is strongly influenced by the bits that preceded it (that is, the memory of past bits). A convolution decoder takes into account such memory when trying to estimate the most likely sequence of data that produced the received sequence of the code bits.

Historically, the first type of convolution decoding, known as sequential decoding, used a systematic procedure to search for a good estimate of the message sequence; however, if such an information sequence from the source is divided into (short) blocks of k digits each, and each k-digit sequence (message) is encoded into an n-digit coded block, such that the n-digit coded block does not only depend on the k-digit message block, but also on m ( $\geq 1$ ) previous message blocks, the encoder is said to possess memory of order m. In this case, the information is encoded into a coded sequence. The collection of all possible code sequences is called an (n,k,m) *convolutional code*. The ratio k / n is called the *code rate*. Convolutional codes were introduced in 1955 and are currently widely used in practice.

A convolution code can also be defined as a type of error-correction code in which procedures employed require a great deal of memory, and typically suffer from buffer overflow and non-graceful degradation:

- Each *k*-bit information symbol (each *k*-bit string) to be encoded is transformed into an *n*-bit symbol, where *n* > *k* and
- The transformation is a function of the last *m* information symbols, where *m* is the *number of memory registers* used in the implementation of the code.

Convolution codes are commonly specified by the three parameters, (n,k,m) where m is the number of shift memory registers used in the implementation of the code. These codes can also be described using the two parameters: the *code rate* and the *constraint length*.

The *code rate*, k/n, is a measure of the efficiency of the code. Commonly, the *k* and *n* parameters range from 1 to 8, and m from 2 to 10, and the code rate from 1/8 to 7/8 except for deep space applications where code rates as low as 1/100 or even longer have been employed.

The *constraint length* parameter, K, denotes the "length" of the convolution encoder, i.e. how many k-bit stages are available to feed the combinatorial logic that produces the output symbols. Closely related to K is the parameter m, which indicates how many encoder cycles an input bit is retained and used for encoding after it first appears at the input to the convolution encoder. The m parameter can be thought of as the memory length of the encoder.

Manufacturers of convolution code chips specify the code differently, namely as (n,k,L). Here the quantity L is called the constraint length, and is defined by the expression:



Fig.2.5 Convolution encoder with 3 memory registers, 1 input and 3 output bits.

The constraint length L represents the number of bits in the encoder memory that affect the generation of the n output bits. The convolution code structure can easily be deduced from its parameters. Begin by drawing m boxes to represent the m memory registers. Then draw n modulo-2 adders to represent the n output bits. Now connect the memory registers to the adders using a generator polynomial as illustrated in Fig.2.5.

This is a rate 1/3 code. Each input bit is coded into 3 output bits. The constraint length of the code is 2. The 3 output bits are produced by the 3 modulo-2 adders by adding up certain bits in the memory registers. The selection of which bits are to be added to produce the output bit is called the generator polynomial (g) for that output bit. For example, the first output bit has a generator polynomial of (1,1,1). The output bit 2 has a generator polynomial of (0,1,1) and the third output bit has a polynomial of (1,0,1). The output bit is just the sum of these bits.

$$v_{1} = (u_{1} + u_{0} + u_{-1}) \mod_{2}$$
  

$$v_{2} = (u_{0} + u_{-1}) \mod_{2}$$
 and ....(2.16)  

$$v_{3} = (u_{1} + u_{-1}) \mod_{2}$$

The polynomials give the code its unique error protection quality. One (3,1,4) code can have completely different properties from another one depending on the polynomial chosen.

#### The D-Transform and Generator Polynomials 2.8.1 of Recursive Processes

A recursive process involves a system with an output variable, which is not only dependent on current values of the input variables but also dependent on previous inputs or state values. For instance, a convolution encoder is an example of a recursive finitestate machine (FSM) used to generate a convolution code. We can represent a typical convolution encoder using the block diagram in Fig.2.6. This FSM has a state  $\sigma_i = [u_{i-1}, u_{i-2}, u_{i-3}]$ , which consists of the past three information symbols. It is a discrete feed-forward filter (in this case) [53].

The output symbols are given by

$$x_{2i} = u_i + u_{i-2} + u_{i-3} \mod_2$$
, and  
 $x_{2i-1} = u_i + u_{i-1} + u_{i-2} + u_{i-3} \mod_2$ .

There are in general  $L_t$  information bits and T=3 dummy tail bits that encode a sequence, i.e.,



#### Fig.2.6 A typical convolution encoder configuration

The encoder is started in the all-zero state,  $\sigma_1 = [0,0,0]$ , and the dummy tail bits drive

it back into the all-zero state [53]. This is called *encoder termination*.

The code rate of this convolution encoder is given by:

$$R = \frac{1}{2} \frac{L_t}{L_t + T}$$
 (2.17)

In practice,  $L_t >> T$  and the rate loss due to the termination can be ignored.

The operation of the encoder can be described algebraically by the D-transform of the connector polynomials, i.e.

$$g^{(1)}(D) = 1 + D^2 + D^3$$
,

and

$$g^{(2)}(D) = 1 + D + D^2 + D^3$$

This means that the top output  $x_{2i}$  connects  $u_i, u_{i-2}$ , and  $u_{i-3}$ , and that the bottom output  $x_{2i-1}$  connects  $u_i, u_{i-1}, u_{i-2}$ , and  $u_{i-3}$ . Using the D-transform notation:

$$[u_0, u_1, u_2, \dots] = u_0 + u_1 D + u_2 D^2 + \dots$$

we can write:

and

$$x^{(1)}(D) = u(D)g^{(1)}(D)$$
$$x^{(2)}(D) = u(D)g^{(2)}(D)$$

as polynomial products, or in matrix form:

$$x(D) = u(D)[g^{(1)}(D), g^{(2)}(D)]^{T} = u(D)G(D).$$

The code x(D) can be realized in a systematic feedback format by writing

$$x(D) = u(D)G_2(D) = u(D)(1 + D^2 + D^3) \left[1, \frac{1 + D + D^2 + D^3}{1 + D^2 + D^3}\right]$$



Fig.2.7 Combined feed-back and feed-forward encoder



Fig.2.8 Recursive systematic turbo code encoder

We note that  $u'(D) = u(D)(1 + D^2 + D^3)$  is simply a scrambled new information sequence from which the u(D) can be easily recovered. Division by  $1 + D^2 + D^3$  describes a feed-back circuit, while the numerator  $1 + D + D^2 + D^3$  describes the new inputs into the circuit.

The x(D) circuit implementation shown in Fig.2.7 performs the product through a division process of two polynomials, i.e.,  $g^{(2)}(D)/g^{(1)}(D)$ . Generally, the encoder (for a rate R=1/2 code) can be written as

$$x(D) = u(D) \left[ 1, \frac{h^{(1)}(D)}{h^{(0)}(D)} \right],$$

where,  $h^{(0)}(D)$  is the feed-back polynomial and  $h^{(1)}(D)$  is the feed-forward polynomial.

## 2.8.2 Recursive Systematic Convolution Encoders

With the advent of *turbo codes*, another implementation of recursive convolution encoders has become popular.

We start by defining the division sequence:

$$w(D)=\frac{u(D)}{h^{(0)}(D)}\,.$$

For example, let  $w(D)(1 + D^2 + D^3) = u(D) \Longrightarrow w(D) = (D^2 + D^3)w(D) + u(D)$ .

This encoder can be realized as depicted in Fig.2.8 above. The parity check sequence of this encoder is given as:

$$p^{(2)}(D) = x^{(2)}(D) = w(D)h^{(1)}(D) = u(D)\frac{h^{(1)}(D)}{h^{(0)}(D)}$$

and is hence identical to that of the (systematic) feedback encoder.

### 2.8.3 Constraint Length

Constraint length is a measure of the size of the encoder state space, and, in certain cases, also of the decoder state space. It measures the delay elements needed in a particular encoder implementation. The constraint length of a convolution encoder is

defined as 
$$v = \sum_{i=1}^{k} \max_{j} (\deg(g_{ij}(D))) = \sum_{i=1}^{k} v_i$$
 .....(2.18)

A minimum basic encoder is one which realizes the code with the minimum number of delay elements. By definition a minimal basic encoder is a basic encoder that has the smallest constraint length among all equivalent encoders.

Convolution codes operate on serial data, one or a few bits at a time, while block codes operate on relatively large (typically, up to a couple of hundred bytes) message blocks. There are a variety of useful convolution and block codes, and a variety of algorithms for decoding the received coded information sequences to recover the original data.

In 1967, Andrew Viterbi developed a decoding technique (The Viterbi-Algorithm), that has since become the standard for decoding convolution codes. At each bitinterval, the Viterbi decoding algorithm compares the actual received code bits with the code bits that might have been generated for each possible memory-state transition. It chooses, based on metrics of similarity, the most likely sequence within a specific time frame. The Viterbi decoding algorithm requires less memory than sequential decoding because unlikely sequences are dismissed early, leaving a relatively small number of candidate sequences that need to be stored.

*Convolution encoding with Viterbi decoding* is a forward-error-control (FEC) technique, that is particularly suited to a channel in which the transmitted signal is corrupted mainly by additive white Gaussian noise (AWGN). *You can think of AWGN as noise whose voltage distribution over time has characteristics that can be* 

described using a Gaussian, or normal, statistical distribution, i.e. a bell curve. This voltage distribution has zero mean and a standard deviation that is a function of the signal-to-noise ratio (SNR) of the received signal. Let's assume for the moment that the received signal level is fixed. Then if the SNR is high, the standard deviation of the noise is small, and vice-versa. In digital communications, SNR is usually measured in terms of  $E_b/N_0$ , which stands for energy per bit divided by the one-sided noise density.

Convolution codes are often used to improve the performance of digital radio, mobile phones and satellite links.

As a result of the wide acceptance of convolution codes, there have been many advances to extend and improve this basic coding scheme. This advancement resulted in two new coding schemes, namely, trellis coded modulation (TCM) and turbo codes. TCM adds redundancy by combining coding and modulation into a single operation (as the name implies). The unique advantage of TCM is that, there is no reduction in data rate or expansion in bandwidth as required by most of the other coding schemes. In this thesis, however, our further discussions are more focused on linear block codes rather than convolution codes.

A free distance (d) is a minimal *Hamming distance* between different encoded sequences. A code correcting capability (t) of a convolution code is a number of errors that can be corrected by the code. It can be calculated as

$$t = \left[\frac{d-1}{2}\right] \tag{2.19}$$

Since a Convolution code doesn't use blocks, processing instead a continuous bitstream, the value of t applies to a quantity of errors, located relatively near to each other. That is, multiple groups of t errors can usually be fixed when they are relatively far.

Free distance can be interpreted as a minimal length of an erroneous "burst" at the output of a convolution decoder. The fact that errors appears as "bursts" should be accounted for when designing a *concatenated code* with an inner convolution code. The popular solution for this problem is to *interleave* data before convolution
encoding, so that outer block (usually *Reed-Solomon*) code can correct most of the errors.

## 2.9 CYCLIC BLOCK CODES

Cyclic codes are a subclass of linear block codes, where all code words in a given code are cyclic shifts of one another. Specifically, if the code word  $\mathbf{c} = (c_0c_1....c_{n-1})$  is a code word in a given code set, then a cyclic shift by 1 denoted as  $\mathbf{c}^{(1)}$  and equal to  $\mathbf{c}^{(1)} = (c_{n-1}c_0.....c_{n-2})$  is also a code word [18]. More generally, any cyclic shift  $\mathbf{c}^{(i)} = (c_{n-i}c_{n-i+1}.....c_{n-i-1})$  is also a code word. The cyclic nature of cyclic codes creates a nice structure that allows their encoding and decoding functions to be of much lower complexity than the matrix multiplications associated with encoding and decoding for general linear block codes.

Cyclic codes are generated via a *generator polynomial instead* of a generator matrix. The generator polynomial g(X) for an (n,k) cyclic code has degree n-k and is of the form:

where  $g_i$  is binary (0 or 1) and  $g_0 = g_{n-k} = 1$ . The *k*-bit information sequence  $(s_0, \dots, s_{k-1})$  is also written in polynomial form as the *message polynomial*:

The code word associated with a given *k*-bit information sequence is obtained from the polynomial coefficients of the generator polynomial times the message polynomial, i.e., the code word  $\mathbf{c} = (c_0 \dots c_{n-1})$  is obtained from:

A code word described by a polynomial c(X) is a valid code word for a cyclic code with generator polynomial g(X) if and only if g(X) divides c(X) with no remainder(no remainder polynomial terms), i.e.

$$\frac{c(X)}{g(X)} = q(X) \tag{2.23}$$

or a polynomial q(X) of degree less than k.

#### Example:

Let us consider a (7,4) cyclic code with generator polynomial  $g(X) = 1 + X^2 + X^3$ . Determine if the code words described by polynomials  $c_1(X) = 1 + X^2 + 2X^3 + X^5 + X^6$  and  $c_2(X) = 1 + X^2 + X^3 + X^5 + X^6$  are valid code words for this generator polynomial.

*Solution*: Division of binary polynomials is similar to division of standard polynomials except that under binary addition, subtraction is the same as addition. Dividing  $c_1(X) = 1 + X^2 + 2X^3 + X^5 + X^6$  by  $g(X) = 1 + X^2 + X^3$ , we have

$$\begin{array}{c}
X^{3} + 1 \\
\underline{X^{3} + X^{2} + 1} \\
\underline{X^{6} + X^{5} + 2X^{3} + X^{2} + 1} \\
\underline{X^{6} + X^{5} + X^{3}} \\
\underline{X^{3} + X^{2} + 1} \\
\underline{X^{3} + X^{2} + 1} \\
0
\end{array}$$

Since g(X) divides c(X) with no remainder, it is a valid code word. In fact, we have  $c_1(X) = (1 + X^3) g(X) = s(X) g(X)$ , so the information bit sequence corresponding to  $c_1(X)$  is s = [1001] corresponding to the coefficients of the message polynomial

$$s(X) = 1 + X^{3}$$

Dividing, we have:

$$c_2(X) = 1 + X^2 + X^3 + X^5 + X^6$$

by

$$g(X) = 1 + X^2 + X^5$$
,

gives:

$$\begin{array}{c} X^{3} \\ (X^{3} + X^{2} + 1) \\ \hline X^{6} + X^{5} + X^{3} + X^{2} + 1 \\ \hline X^{6} + X^{5} + X^{3} \\ \hline X^{2} + 1 \end{array}$$

where, we note that there is a remainder of  $X^2 + I$  in the division. Thus,  $c_2(X)$  is not a valid code word for the code corresponding to this generator polynomial.

## 2.10 SYSTEMATIC LINEAR CODES

Systematic linear codes have the first k code word symbols equal to the information bits, and the remaining code word symbols equal to the parity bits. A cyclic code can be converted into systematic form by first multiplying the message polynomial s(X) by  $X^{n-k}$ , yielding

This shifts the message bits to the k rightmost digits of the code word polynomial. If we next divide (2.24) by g(X), we obtain

$$\frac{X^{n-k}s(X)}{g(X)} = q(X) + \frac{p(X)}{g(X)}, \qquad (2.25)$$

where, q(X) is a polynomial of degree at most k-1 and p(X) is a remainder polynomial of degree at most n-k-1. Multiplying (2.25) through by g(X) we obtain:

$$X^{n-k}s(X) = q(X)g(X) + p(X) \dots (2.26)$$

Adding p(X) to both sides yields

$$p(X) + X^{n-k}s(X) = q(X)g(X)$$
....(2.27)

(modulo 2 Arithmetic used).

This implies that  $p(X) + X^{n-k} s(X)$  is a valid code word since it is divisible by g(X) with no remainder. The code word is described by the n coefficients of the code word polynomial  $p(X) + X^{n-k} s(X)$ . Note that we can express p(X) (of degree *n-k-1*) as:

Combining (2.22) and (2.27) (through addition) we get:

$$p(X) + X^{n-k}s(X) = p_0 + p_1X + \dots + p_{n-k-1}X^{n-k-1} + s_0X^{n-k} + s_1X^{n-k+1} + \dots + s_{k-1}X^{n-1}$$
....(2.29)

Thus, the code word corresponding to this polynomial has the first *k* bits consisting of the message bits  $[s_0 \dots s_k]$  and the last *n*-*k* bits consisting of the parity bits  $[p_0 \dots p_{n-k-1}]$ , as required for the systematic form.

We note that the systematic code word polynomial is generated in three steps: first multiplying the message polynomial s(X) by  $X^{n-k}$ , then dividing  $X^{n-k}s(X)$  by g(X) to get the remainder polynomial p(X) (along with the quotient polynomial q(X), which is not used), and finally adding p(X) to  $X^{n-k}s(X)$  to get (2.28). The polynomial multiplications are straightforward to implement, and the polynomial division is easily implemented with a feedback shift register. Thus, code word generation for systematic cyclic codes has very low cost and low complexity.

# 2.10.1 Characterization of channel errors for cyclic codes

The code word polynomial which corresponds to a transmitted code word is of the form:

c(X) = s(X) g(X).....(2.30)

The received code word can also be written in polynomial form as

r(X) = c(X) + e(X) = s(X) g(X) + e(X).....(2.31)

where e(X) is the error polynomial of degree n-1 with coefficients equal to 1 where errors occur. For example, if the transmitted code word is  $\mathbf{c} = [1011001]$  and the received code word is  $\mathbf{r} = [1111000]$  then  $e(X) = X + X^{n-1}$ . The **syndrome polynomial** *sypol(X)* for the received code word is defined as the remainder when r(X) is divided by g(X), so sypol(X) has degree n-k-1. But by eq. (2.28), e(X) = g(X)sypol(X). Therefore, the syndrome polynomial sypol(X) is equivalent to the error polynomial e(X) modulo g(X). Moreover, we obtain the syndrome through a division circuitry similar to the one used for generating the code. As mentioned before, this division circuit is typically implemented using a feedback shift register, resulting in a low-cost low-complexity implementation [18].

## 2.11 EXAMPLES OF COMMON LINEAR CODES

Examples of common linear codes include:

- Hamming codes
- Golay Codes
- Bose-Chadhuri-Hocquenghem (BCH) Codes, and Low density parity check (LDPC) codes.

#### 2.11.1 Hamming Codes

In 1948 Hamming [53] found the first error control code, now known as the [7,4] Hamming code; he was trying to extend the concept of parity checks to correcting errors. The [7,4] Hamming code is composed of the *binary code words* 

 $\underline{x} = [x_1, x_2, x_3, x_4, x_5, x_6, x_7]$ 

of length 7, fulfilling the following parity check equations:

$$x_4 + x_5 + x_6 + x_7 = 0 \mod_2$$
  
 $x_2 + x_3 + x_6 + x_7 = 0 \mod_2$ 

and

 $x_1 + x_3 + x_5 + x_7 = 0 \mod_2$ , .....(2.32)

at the sending end.

This leaves only four free choices among the seven binary symbols, these choices are arbitrary and are the *information bits*. (Note: here addition is binary addition, also known as the EXOR (exclusive-OR) function). Let us choose  $x_3$ ,  $x_5$ ,  $x_6$ , and  $x_7$  as the information bits.

If x is transmitted and received as y, the code can be able to tolerate the presence of a single bit error and still deliver the correct information bits  $x_3$ ,  $x_5$ ,  $x_6$ , and  $x_7$ . This is achieved in the following simple way: let

$$y_4 + y_5 + y_6 + y_7 = \alpha$$
,  
 $y_2 + y_3 + y_6 + y_7 = \beta$ ,

and

 $y_1 + y_3 + y_5 + y_7 = \gamma$ , .....(2.33)

at the receiving end. Binary addition applies in all cases.

The vector  $(\alpha, \beta, \gamma)^T$  gives the *location of the error in binary forms*, i.e.,  $(1,0,1)^T$  means the fifth symbol is *in error* [53]. i.e.,  $(1\ 0\ 1)_2 \rightarrow 1.2^2 + 0.2^1 + 1.2^0$ = 4 + 1 = 5.

In 1949 Golay [53] did not only discover two perfect codes, i.e., the [23,12] binary (c.f. GF(2)) Golay code and the [11,6] ternary (GF(3)) Golay code, but also extended Hamming's [7,3] code to the infinite family:

 $[2^{m}-1, 2^{m}-(m+1),3], \forall m > 1 \text{ of Hamming codes.}$ 

## 2.11.2 The Parity Check Equation

For the [7,4] Hamming code

This algebraic formulation immediately reveals some fundamental principles [53]: Namely, that the code is linear, i.e.,  $Hx_1 = 0$ ,  $Hx_2 = 0$ ,  $\rightarrow H(x_1 + x_2) = 0$ , and  $x_3 = (x_1 + x_2)$  is also a code word. We conclude logically, that the family of  $[2^m - 1, 2^m - (m+1), 3]$ . Hamming codes has parity check matrices **H**, the matrices

whose columns consist of all  $2^{m}$ -1 non-zero vectors of length m. e.g. if m = 3, we have the [7,4,3] - Hamming code  $\rightarrow$  [n,k,d] - code, where d is the minimum distance or Hamming distance of the code. Hamming codes are single-error correcting codes.

A single error is identified by  $h_j$ , where *j* is the location of the error and all  $h_j$  are unique. Conversely, double errors in positions *i* and *j* are not identifiable since  $h_i + h_j = h_k$ , look like a single error in position *k*.

Any parity check matrix can be arranged such that

 $\underline{\mathbf{H}} = [\underline{\mathbf{A}} \mid \underline{\mathbf{I}}_{\mathbf{n}\cdot\mathbf{k}}] . \qquad (2.36)$ 

Through column and row permutations and linear combinations of rows, where  $\underline{\mathbf{I}}_{n-k}$  is the n-k identity matrix and  $\underline{\mathbf{A}}$  has dimension  $(n-k) \times k$ . Example:

From this form we can obtain the systematic *k x n code generator matrix* 

through simple matrix manipulation.

For the [7,4] – Hamming code

The code generator matrix has dimensions  $k \ge n$  and is used to generate directly the code words  $\underline{x}$  via

$$\underline{\boldsymbol{x}}^{\mathrm{T}} = \underline{\boldsymbol{u}}^{\mathrm{T}} \underline{\boldsymbol{G}}, \qquad (2.40)$$

where  $\underline{u}$  is the information k – tuple. Algebraically, the code word  $\underline{x}$  lies in the row space of  $\underline{G}$ .

The Hamming code is an error-detecting and binary error correcting code used in data transmission systems that can

• Detect all single- and double-errors and

• Correct all single-bit errors.

Hamming codes are identified by the ordered set (n, k). The Hamming code (7,4) is the classic example used which describes a word of 4 data bits long and 3 error parity check bits.

## 2.11.3 Hamming Bound

A t-error correcting code with M code words must fulfill the inequality

where, n is the code length and M is the number of code words.

*Mathematical Proof of the Hamming Bound*: There are  $2^n$  possible binary vectors of length n. Each of the *M* code words needs a sphere of distance *t* around itself which cannot contain another code word in order to tolerate *t* errors and still be uniquely identifiable.

**Definition**: A code, which fulfills the *Hamming Bound* with equality is called a perfect code.

The family of Hamming codes and the two codes discovered by Golay are the only existing perfect codes.

A Hamming code satisfies the relation  $(k + p + 1) \ge 2^p$ , with (k + p) = n, where n is the total number of bits in a block, k is the number of information bits in the block, and p is the number of parity check bits in the block, such that p = (n - k).

A set of Hamming codes exists and is referred to as the 'forward error correction code set'; it has the capability of enabling the receiving station to correct a transmission error. While it takes more bits to send the information, it means fewer retransmissions and thus can actually speed up a noisy or hazardous channel.

The number of parity bits in the Hamming code is given by the Hamming rule. The rule is represented by the inequality  $k + p + 1 \ge 2^p$ . The hamming code word is

created by multiplying the data bits by a generator matrix using modulo-2 arithmetic. The result of this is called a *code word vector*, which consists of the original data bits and the parity bits.

The generator matrix used in constructing the Hamming code consists of two components, namely, the identity matrix component I, and the parity check generation matrix P. For a data size of 4 bits the following matrix is created:

Multiplying a 4 – bit vector (d1, d2, d3, d4) by G results in a 7-bit vector of the form (d1, d2, d3, d4, p1, p2, p3). The P component is what generates the parity bits. If the selection of the columns of P are unique, it is true that (p1, p2, p3) is the parity calculation of three distinct subsets of the original data.



#### 2.11.4 Golay Codes

Binary Golay code refers to two so-called closely related error-correcting codes. One is called *extended binary Golay code*, which is an error-correcting code that encodes 12 bits of data in a 24-bit word, in such a way that any triple-bit error can be corrected and any quadruple-bit error can be detected. The other is called the *perfect binary Golay Code*, which has code words of length 23 and is obtained from the extended binary Golay code, by deleting one coordinate position. Conversely, the extended binary Golay code can be obtained from the perfect binary Golay code by adding a parity bit.

There are two essentially distinct versions of the Golay code: a binary version and a ternary version. The binary version  $G_{23}$  is a (23,12,7) binary linear code consisting of  $2^{12} = 4096$  code words of length 23 and minimum distance 7. The ternary version is a (11,6,5) ternary linear code, consisting of  $3^6 = 729$  code words of length 11 with minimum distance 5.

By adding a parity check bit to each code word in  $G_{23}$ , the extended Golay code  $G_{24}$ , which is a nearly perfect [24,12,8] binary linear code, is obtained.

## 2.11.5 Bose-Chaudhuri-Hocquenghem (BCH) Codes

Bose-Chaudhuri-Hocquenghem (BCH) Code is a large family of powerful cyclic block forward error correction codes used in the transmission of data. For any positive integers m, m > 3, and t <  $2^{m-1}$ , there is a binary BCH code with a block length n equal to  $2^m$  - 1 and n - k < m\*t parity check bits, where k is the number of information bits. The BCH code has a minimum distance of at least 2t + 1.

## 2.11.6 Low Density Parity Check (LDPC) Codes

While LDPC and other error correcting codes cannot guarantee perfect transmission, the probability of lost information can be made as small as desired. LDPC was the first code to allow data transmission rates close to the theoretical maximum, the Shannon Limit. Impractical to implement when developed in 1963, LDPC codes were forgotten. LDPC codes are capacity-approaching codes, which means that practical constructions exist that allow the noise threshold to be set very close (or even *arbitrarily* close) to the theoretical maximum (the Shannon limit) for a symmetric memory-less channel. The noise threshold defines an upper bound for the channel noise, up to which the probability of lost information can be made as small as desired.

The explosive growth in information technology has produced a corresponding increase of commercial interest in the development of highly efficient data transmission codes as such codes impact everything from signal quality to battery life. LDPC codes are finding increasing use in applications where reliable and highly efficient information transfer over bandwidth or return-channel constrained links in the presence of data-corrupting noise is desired. Although implementation of LDPC codes has lagged that of other codes, notably turbo codes, the absence of encumbering software patents has made LDPC attractive to some.

Impractical to implement when first developed by Gallager in 1963, LDPC codes were forgotten, but they were rediscovered in 1996. Turbo codes, another class of capacity-approaching codes discovered in 1993, became the coding scheme of choice

in the late 1990s, used for applications such as deep space satellite communications. However, in the last few years, the advances in low-density parity-check codes have seen them surpass turbo codes in terms of error floor and performance in the higher code rate range, leaving turbo codes better suited for the lower code rates.

## 2.12 BLOCK CODES AND THE GENERATOR MATRIX

The encoding of k information (message) bits into n physical bits can be described by a *generator matrix*, <u>**G**</u>, with n rows and k columns, where n > k, and where each matrix entry is either 0 or 1. The generator matrix is a compact description of how code words are generated from information bits in a linear block code. The design goal in linear block codes is to find generator matrices such that their corresponding codes are easy to encode and decode yet have powerful error correction/detection capabilities. We represent the encoding process as a set of n equations defined by:

$$t_{ij} = s_{i1}g_{1j} + s_{i2}g_{1j} + s_{i3}g_{1j} + \dots + s_{ik}g_{kj}, \quad j = 1,\dots,n,\dots(2.43)$$

where,  $g_{ii}$  is binary (0 or 1) and standard binary multiplication is used.

With reference to channel coding, the transmitted stream  $\underline{\mathbf{t}}$  is related to the source stream  $\underline{\mathbf{s}}$  by the linear relation:  $\underline{\mathbf{t}} = \underline{\mathbf{G}}^{\mathrm{T}}\underline{\mathbf{s}}$  .....(2.44)

where,  $\underline{\mathbf{G}}$  is called the generator matrix.

#### 2.12.1 The Replication Process and Code

Replication is similar to repetition. For example, a repetition code for 1 logical bit, replicated 3 times, can be described as follows:

$$\underline{\mathbf{s}} = (\mathbf{x}_0) ; \text{ Let } \underline{\mathbf{G}} = (1 \ 1 \ 1) ;$$
  
then  $\underline{\mathbf{G}}^{\mathrm{T}} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} ;$ 

and

$$\underline{\mathbf{t}} = \underline{\mathbf{G}}^{\mathrm{T}} \underline{\mathbf{s}} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} (\mathbf{x}_{0}) = \begin{pmatrix} x_{0} \\ x_{0} \\ x_{0} \end{pmatrix}$$

To encode a pair of logical bits into a triple of physical bits, we can use the generator matrix:

$$\underline{\mathbf{G}} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix};$$

and

$$\underline{\mathbf{G}}^{\mathrm{T}} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{pmatrix}; \text{ the source vector } \underline{\mathbf{s}} = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix}, \text{ so that we get the}$$

transmitted vector (matrix) as:

or

$$\mathbf{\underline{t}} = \mathbf{\underline{G}}^{\mathrm{T}} \mathbf{\underline{s}} = \begin{pmatrix} x_{0} \\ x_{0} \\ x_{0} \\ x_{1} \\ x_{1} \\ x_{1} \\ x_{1} \end{pmatrix}$$
 (2.46)

Let us look at another example: if

$$\underline{\mathbf{G}} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{pmatrix},$$

with

$$\underline{\mathbf{s}} = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix},$$

we have

If  $x_0 = 0$ , it implies that

$$\mathbf{\underline{t}} = \begin{pmatrix} 0\\0\\0\\x_1\\x_1\\x_1\\x_1 \end{pmatrix},$$

which is a replication code;

and if  $x_0 = 1$ , then we have:

which is another variant of a replication code.

The *replication* or *repetition* code has the following implications:

• Encoding the original source data by introducing redundancy *may reduce the probability of a decision error*. E.g., the probability of making a decision error using a triple repetition code is given by:

$$\mathbf{p}_{\text{error}} = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \mathbf{p}^2 (1-\mathbf{p}) + \begin{pmatrix} 3 \\ 3 \end{pmatrix} \mathbf{p}^3 ,$$

or

$$p_{error} = 3p^2 - 2p^3 \dots (2.49)$$

• Therefore, the *probability of a decision error* can be made arbitrarily small at the expense of lowering the *rate of transmission (or code rate* R = k/n).

## 2.12.2 Testing for Errors Using the Orthogonal Property of Matrices

The above formulation of the channel encoding process does not tell us how errors which occur along the channel can be detected and/or corrected at the destination.

To be able to detect errors, another matrix called a *parity check matrix*,  $\underline{H}$ , is used. The parity check matrix  $\underline{H}$  is related to the generator matrix  $\underline{G}$ , through the *orthogonal property*: i.e.,

Matrix <u>G</u>, which encodes a k – bit message into n physical bits, where n > k, is an *n* x k matrix, i.e., it has n rows and k columns. Consequently, to make the orthogonal relation (2.50) valid, the parity check matrix <u>H</u> must have n columns. We also require that it should have (n - k) rows. Thus, the parity check matrix <u>H</u> in (2.50) is an (n - k) by *n* matrix.

Matrix	Rows	Columns	Designation
<u>G</u>	п	k	n x k
<u>H</u>	n-k	p=n	( <i>n-k</i> ) x <i>p</i>

The product  $\underline{A}$  .  $\underline{B}$  of two matrices is defined whenever the number of columns of the first factor is the same as the number of rows in the second.

Thus with reference to the product  $\underline{H}$ .  $\underline{G} = \underline{0}$ , the number of columns of  $\underline{H}$  must be equal to the number of rows of  $\underline{G}$ .

Matrix	Rows	Columns	Designation
<u>H</u>	n-k	p=n	$(n-k) \ge p$
<u>G</u>	n	k	$n \ge k$

In this case, the number of columns of <u>H</u> must be equal to number of rows of <u>G</u> for the relation (2.50) to be true (valid). The product of <u>H</u> and <u>G</u> is an  $(n - k) \times k$  matrix.

The parity check matrix  $\underline{H}$  is referred to as the orthogonal complement of matrix  $\underline{G}$  if the relation (2.50) above is valid.

Let

$$\underline{\boldsymbol{G}} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}; \qquad \underline{\boldsymbol{H}} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

and

$$\underline{\boldsymbol{H}} \cdot \underline{\boldsymbol{G}} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Because matrices  $\underline{G}$  and  $\underline{H}$  complement each other, their roles can be reversed. This means that you can use  $\underline{G} = \underline{H}^T$  as a generator matrix and  $\underline{H} = \underline{G}^T$  as a parity check matrix. Matrix  $\underline{G}$  and  $\underline{H}$  are said to be dual to each other.

The parity check matrix is used to decode linear block codes with generator matrix **G**. The parity check matrix  $\underline{H}$  corresponding to a generator matrix  $\underline{G} = [\underline{I}_k / \underline{P}]$  is defined as:

$$\underline{H} = [\underline{P}^{T} / \underline{I}_{n-k}]. \qquad (2.51)$$

It can easily be shown that  $\underline{GH}^T = \underline{0}_{k,n-k}$ , where  $\underline{0}_{k,n-k}$  denotes an all-zero  $k \ x \ (n-k)$  matrix. Recalling that a given code word  $\mathbf{T}_i$  in the code is obtained by multiplication of the information bit sequence  $\underline{S}_i$  by the generator matrix  $\underline{G}: \underline{T}_i = \underline{S}_i \underline{G}$ , we have:

$$\underline{T}_{i} \underline{H}^{T} = \underline{S}_{i} \underline{G} \underline{H}^{T} = \underline{0}_{n-k} , \qquad (2.52)$$

for any input sequence  $T_i$ , where  $\underline{O}_{n-k}$  denotes the all-zero row vector of length n - k. Thus multiplication of any valid code word with the parity check matrix results in an all-zero vector. This property is used to determine whether the received vector is a valid code word or has been corrupted, based on the notation of syndrome testing.

#### Example:

Let

$$\underline{\boldsymbol{G}} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}$$

and

		(0	0	0	1	1	1	1)	
<u>H</u>	=	0	1	1	0	0	1	1	;
		(1)	0	1	0	1	0	1)	

Matrix	Rows	Columns	Designation
<u>G</u>	7	4	(7 x 4)
<u>H</u>	3	7	(3 x 7)

Since all our additions in the above operations are modulo 2, the matrix  $\underline{H}$  is indeed perpendicular (orthogonal) to  $\underline{G}$  because  $\underline{H}.\underline{G} = \underline{0}$ .

## 2.12.3 Checking for Channel Transmission Errors

To check if a given block code  $\underline{x}$  has been correctly received at the destination as  $\underline{y}$ , we calculate the product  $\underline{H}$ .  $\underline{y}$ , and if  $\underline{H}$ . $\underline{y} \neq \underline{0}$ , we suspect that an error occurred during transmission.  $\underline{H} \cdot \underline{y}$  is called the **error syndrome**.

In general, we can describe the error syndrome finding procedure as follows:

If  $\underline{G} \cdot \underline{x} = \underline{y}$ , then if an error occurs, we can represent it by e such that  $\underline{y}' = \underline{y} + \underline{e}$ . The error syndrome then returns:

(Modulus 2). An important category of linear codes is the so called Hamming Codes.

The parity check matrix  $\underline{H}$  for Hamming codes is a 3 x 7 matrix that is constructed by *making its columns be the numbers in binary representation* from 1 to 2<sup>r</sup> - 1, where r is the number of rows in the parity check matrix.

For example, if r = 3, we have the numbers 1, 2, 3, 4, 5, 6, and 7 and their representation in binary form is shown below:

Number	Binary Representation
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Therefore, for r = 3 the parity check matrix for the Hamming Code looks as follows:

		(0)	0	0	1	1	1	1)	
<u>H</u>	=	0	1	1	0	0	1	1	•
		(1	0	1	0	1	0	1)	

The corresponding generator matrix to  $\underline{H}$  is obtained from the relation  $\underline{G} = \underline{H}^{T}$  and is a 7 x 4 matrix:

$$\underline{\boldsymbol{G}} = \begin{pmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \end{pmatrix}.$$

Because matrices  $\underline{G}$  and  $\underline{H}$  complement each other, their roles can be reversed. This means that you can use  $H^T$  as a generator matrix and  $\underline{G}^T$  as a parity check matrix. This new encoding is different from the original one given by  $\underline{G}$  and  $\underline{H}$ , but both codes are related. We say that they are *dual* to each other.

Message Word $\underline{\mathbf{x}}$	Code word	y
0000	0000000	
0001	1010001	
0010	1110010	
0011	0100011	
0100	0110100	
0101	1100101	
0110	1000110	
0111	0010111	
1000	1101000	
1001	0111001	
1011	1001011	
1100	1011100	
1101	0001101	
1110	0101110	
1111	1111111	

 Table 2.1 : Conversion of a message word into a code word for transmission

The notation that accompanies this duality is as follows: If the pair { $\underline{G}$ ,  $\underline{H}$ } defines (information/source) bits into n physical bits (n > k), then the pair { $\underline{H}^T$ ,  $\underline{G}^T$ } defines a code  $\mathbf{C}^{\perp}$  [n, n - k], which encodes n - k logical (information/source) bits into n physical bits.

One example of an error-control code is the (7, 4) Hamming block code. (7, 4) means that 4 message bits, represented by the vector  $\underline{x}$ , are converted into a 7 – bit code word, represented by the vector  $\underline{y}$ , according to Table 2.1 above. This coding process can be conveniently represented in matrix form as:

 $\underline{\mathbf{y}}' = \underline{\mathbf{x}} \underline{\mathbf{G}}, \text{ (modulo 2)}$  .....(2.54)

where,  $\underline{G}$  is a "generator matrix" given by:

 $\underline{\boldsymbol{G}} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} ;$ 

#### 2.12.4 Modelling Effects of a Hazardous Channel

After the message bits pass through a hazardous (e.g. noisy) channel (medium) that introduces unknown errors, we can model the received vector  $\underline{v}$  as follows:

$$\underline{\mathbf{y}} = \underline{\mathbf{y}}' + \underline{\mathbf{e}} \pmod{2.55}$$

In modulo-2 addition we have 0 + 0 = 1 + 1 = 0; 0 + 1 = 1 + 0 = 1. Therefore, we imagine the error vector  $\underline{e}$  as a binary vector, where a component of  $\underline{e}$  is 1, when the received  $\underline{y}$  vector differs from the originating  $\underline{x}$  vector.

The receiver then needs to decode the data in an attempt to recover the original message bits. This produces a "syndrome",  $\underline{s}$ , which determines what the error vector was (if no errors or only a single bit error has occurred). This is again conveniently represented in matrix notation, though physical electronics may implement it differently:

 $\underline{s} = \underline{y} \underline{H}^T$  (modulo-2) .....(2.56)

where,  $\underline{\mathbf{H}}^{T}$  is the transpose of the "parity-check" matrix,  $\underline{\mathbf{H}}$ , given by:

		(1	0	0	1	0	1	1)	
<u>H</u>	=	0	1	0	1	1	1	0	(2.57)
		(0	0	1	0	1	1	1)	

 $\underline{s} = \underline{y} \underline{H}^{\mathrm{T}}$ ;  $\underline{y} = \underline{y}' + \underline{e}$  .....(2.58)

A 3 - bit syndrome uniquely determines what the error pattern was, for single bit errors. So, since we know what the error was, we can correct it and decrease the

probability of a received bit error for the system. The table below defines the syndrome to error pattern relationship. But if the syndrome is perceived as a binary number, it can be converted to decimal and used as an index into the following error-pattern look-up matrix (table). The decimal value of the syndrome now points to the row of  $\underline{\mathbf{e}}$  that contains the corresponding error pattern.

Syndrome	Error Patterr			
000	0000000			
100	1000000			
010	0100000			
001	0010000			
110	0001000			
011	0000100			
111	0000010			

The (7, 4) Hamming code is such that, when there is a *single error* in  $\underline{\mathbf{v}}$ , the row of  $\underline{\mathbf{e}}$  chosen with the syndrome matches the <u>*e*</u> used above to model the relationship between  $\underline{\mathbf{x}}$  and  $\underline{\mathbf{v}}$ . Therefore, given  $\underline{\mathbf{v}}$ , the decoding procedure is

 $\underline{y}$  (corrected) =  $\underline{y}' + \underline{e}$  (modulo-2)

since addition modulo-2 of any binary number vector to itself results in a vector of zeros.

#### 2.13 STOCHASTIC DECODING METHODS

The basic wireless communications system is illustrated in fig.2.1. In a digital communications system, the *encoder* is part of the transmitter and the *decoder* is part of the receiver. The encoder adds redundancy to a sequence u of input information symbols. The *channel* is a probabilistic medium governed by conditional probabilities  $p(y_r | x_r)$ , of individual output symbols,  $y_r$ , given the input symbol,  $x_r$ . The decoder is an algorithm, which reconstructs u from the received word y as the estimate,  $\hat{u}$ .

#### 2.13.1 Optimal Decoding

A "good" decoder reconstructs u with a small probability of error,

$$P_{B} = P(\hat{u} \neq u). P(\hat{u} \neq u) = \sum_{y} P(\hat{u} = u \mid y) \dots(2.60)$$

This is the *maximum a posteriori* decoding principle which by definition states that: A *maximum a posteriori* (MAP) *decoder* selects the information word whose code word maximizes the a posteriori probability given the channel observation, i.e.,

 $\hat{u} = \arg \max_{u} P(x(u) | y) \dots (2.61)$ 

## 2.13.2 Maximum Likelihood Decoding

We can develop the a posteriori probability as follows:

$$P(x(u) \mid y) = \frac{P(y \mid x(u))P(x(u))}{P(y)} \quad \propto \quad P(y \mid x(u))P(x(u)) \dots \dots (2.62)$$

The *a posteriori* probability depends both on the channel probabilities, P(y | x(u)) and the code word *a priori* probabilities, P(x(u)). If all code words are equally likely, the following event prevails:

A *maximum likelihood* (ML) *decoder* selects the information word, whose code word maximizes the conditional channel probability, given the channel observation, i.e.,

$$\hat{u} = \arg \max P(y | x(u)).$$
 (2.63)

#### 2.13.3 Minimum Distance Decoding

One of the simplest channel models is the binary symmetric channel, where both of two possible input symbols are converted into the other symbol with an error probability  $\varepsilon$ . In this case, given that  $\underline{y}$  and  $\underline{x}$  differ in  $d(\underline{x},\underline{y})$  positions

If  $\varepsilon < 0.5$  this equation is maximized by minimizing d(x,y). A minimum distance decoder selects the information word, whose code word minimizes the Hamming distance d(x,y), i.e.,

$$\hat{u} = \arg \min_{u} d(x, y)$$
. ....(2.65)

The Hamming distance  $d(x_1, x_2)$  between two code words  $x_1$  and  $x_2$  is the number of symbols where these two code words differ. The *minimum Hamming distance* (MHD) of a code *C* is defined as

$$d_{min} = \min_{x_1, x_2} d(x_1, x_2), \ \forall \ x_1, x_2 \in C, \ x_1 \neq x_2. \dots (2.66)$$

## 2.13.4 Minimum Hamming Weight (MHW)

The Hamming weight,  $w(x_1)$  of a code word,  $x_1$  is the number of non-zero entries of  $x_1$ .

The MHW of a code *C* is:

$$w_{\min} = \min_{x_1} w(x_1) = \min_{x_1,0} d(x_1,0), \quad \forall x_1 \in C, \quad x_1 \neq 0.$$
 (2.67)

Furthermore [53], for a linear code C:

$$d_{\min} = w_{\min}$$
.

A code with minimum Hamming distance (MHD)  $d_{\min}$  can correct  $\left\lfloor \frac{d_{\min}}{2} \right\rfloor$  errors [53].

## 2.13.5 Decoding on General Channels

The maximum likelihood (ML) decoder selects

$$\begin{aligned}
\hat{x} &= \arg \max_{x} P(y \mid x) \\
&= \arg \max_{x} \sum_{i=1}^{n} \log(P_{Y|X}(y_{i} \mid x_{i})) \\
&= \arg \max_{x} \sum_{i=1}^{n} \alpha[\log(P_{Y|X}(y_{i} \mid x_{i}) + f(y_{i}))] \\
&= \arg \max_{x} \sum_{i=1}^{n} m_{i}(x) \dots (2.68)
\end{aligned}$$

The addition of the constants  $\alpha$  and  $f(y_i)$  have no effect on the selection of  $\tilde{x}$ . The term  $m_i(x)$  is called the matrix of code word x at time i. The AWGN channels are characterized by the equations:

$$y_i = x_i + n_i;$$
  $P_N(n_i) = \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{n_i^2}{N_0}\right).$ 

Therefore,

$$P_Y(y_i) = \frac{1}{\sqrt{\pi N_0}} \exp\left(-\frac{(y_i - x_i)^2}{N_0}\right),$$

and the *Gaussian performance metric* is given by  $m_i(\mathbf{x}) = (y_i - x_i)^2$ , which is called the *minimum distance metric*.

## 2.14 THE SEARCH FOR GOOD CODES

Let *C* be a block code (not necessarily linear) over a q-ary alphabet, i.e., *C* is a subset of V(n,q). If *C* has *m* code words and of minimum distance d, it is called an (n, m, d) - code.

For a fixed n, the parameters m and d work against one another – the bigger m is, the smaller is d and vice-versa. This is unfortunate since for practical reasons, we desire a large number of code words with high error-correcting capability (large m and large d). The search for good codes always involves some compromise between these parameters.

Since V(n, q) has a metric defined on it, it makes sense to talk about spheres centered at a vector with a given radius.

Thus, 
$$B(x, r) = \{y \text{ in } V(n, q) \mid d(x, y) < = r \}$$

.....(4.34)

is the sphere of radius r centered at x. The *covering radius* of a code C is the smallest radius s so that

$$V_n(q) \subseteq \bigcup_{x \in C} B(x, s)$$
 (4.35)

i.e., every vector of the space is contained in some (at least one) sphere of radius s centered at a code word.

A code is said to correct e errors if a decoder using the nearest neighbour decoding scheme is capable of correcting any pattern of e or fewer errors, introduced by the channel. In this case, the decoder can correct any transmitted code word, which has been altered in e or fewer coordinate positions.

Let c be one code word of C and S be the set of all n-tuples over the alphabet of C and define

 $S(c) = \{a \text{ belongs to } S : d(a, c) \le e\}$  .....(4.36)

S(c) is called the *sphere of radius e* about the code word c. It consists of all *n*-tuples within distance *e* of the code word *c*, which we think of as being at the centre of the sphere.

The *packing radius* of a code *C* is the largest radius t so that the spheres of radius t centered at the code words are disjoint. Obviously, t < = s. When t = s, we say that C is a *perfect code*.

**Definition1**: A code is said to correct  $\mathbf{e}$  errors if using nearest neighbour decoding, it is possible to decode correctly any transmitted code word, which has been altered in  $\mathbf{e}$  or fewer coordinate positions by the channel.

**Definition2**: A code is said to correct k errors if, given any transmitted code word, which has been altered, but only in k or fewer coordinate positions, it is possible to recognize correctly that "some errors have occurred".

The *minimum distance*, d, of a perfect code *must be odd*. If it were even, there would be vectors at an equal distance from two code words and spheres around those code words *could not be disjoint* if they had to contain these vectors.

So, d = 2e + 1, where e is the number of errors in a given code word and it is easy to see that for a perfect code t = s = e.

Furthermore, we can count the number of vectors in a sphere of radius e and obtain the

**proposition**: A q – ary (n, m, d) – code is perfect if and only if d = 2e + 1 and

*Designing a good code is a fairly involved process. Many factors need to be considered.* For example, the code should be designed appropriately in practice taking into account the expected error rates for the particular channel being used [10].

For example, if it is known that the probability that the channel will introduce two errors into any transmitted code word is extremely small, then it will not be necessary to construct a code that will correct all two-bit error patterns. A single-error correcting code will likely be good enough. Conversely, if double errors are frequent and a single error correcting code is being used, then decoding errors will be frequent. This brief discussion raises a number of questions:

- Given n, M and d, can we determine if an [n, M] block code with distance d exists?
- Assuming such a code exists, how would one be constructed in practice?
- How should information k-tuples be associated with code word n-tuples to facilitate efficient channel encoding?
- How should channel decoding be performed?

The issue of channel decoding needs to be considered in greater detail. When some word is received and it is not a valid code word, finding the closed code word involves computing the distance from the received word to each code word in a set, requiring M comparisons. While this might be acceptable for a small M, in practice we find that M is usually quite large.

Suppose a code C is used with  $M = 2^{50}$  code words, which is quite realistic! If we could carry out one million distance computations per second, it would take around 20 years to make a single correction. Clearly, this is not tolerable and more efficient techniques are required. Wishing to retain the nearest neighbour strategy, we seek more efficient techniques for its implementation as discussed below.

#### Nearest Neighbour Decoding

If we have an [n, M] – block code C with distance d, we could adapt a strategy for the decoder at the end of the communication channel. When this decoder receives an n-tuple r, it must make some decision. This decision may be one of the following possibilities:

- no errors have occurred: accept r as a valid code word.
- Errors have occurred: correct r to some code word  $c \in C$ .
- Errors have occurred: no correction is possible.

In general, the decoder will *not always* make the correct decision; for example, we can consider the possibility of an error pattern occurring which changes a transmitted code word into another code word. The goal is that the decoder should take a course of action which has the greatest probability of being correct. We hereby assume that errors are introduced by the channel at random, and that the probability of an error in one coordinate is independent of errors in adjacent coordinates.

The decoding strategy normally adapted is called *nearest neighbour decoding*, which is specified as follows:

If an n-tuple r is received, and there is a unique code word c that belongs to C such that d(r, c) is a minimum, then correct r to the c. If no such c exists, report that errors have been detected, but no correction is possible. By nearest neighbour decoding, a received vector is decoded to the code words "closest" to it, with respect to Hamming distance.

Using nearest neighbour decoding, what the decoder actually does is to maximize the probability P(r | c) that r is received, given that c is sent, i.e., choosing the nearest code word is equivalent to choosing the most likely input message c given the received tuple r. This decoding strategy is also referred to as **maximum likelihood decoding**.

Suppose that a code word from a code  $C \in A^n$  (element or subset of  $A^n$ ) is transmitted and a word  $r \in A^n$  is received:

- 1. if r is a code word, then decode as r.
- 2. if  $r \notin C$ , and if there is a *unique* code word  $r_0$  that is nearest to r with respect to Hamming distance, then decode as  $r_0$ .
- if r ∉ C, and there are at least 2 code words at the minimum Hamming distance from r, then we cannot decode r by this decoding scheme.

Note: Hamming distance is highly significant for determining how good a code is at error correction or detection.

#### **Gilbert-Varshamov Bound**

The Gilbert-Varshamov bound [53] quantifies limits of an error control's error correcting capability as a function of the code's size and rate. The *Gilbert-Varshamov Bound* states that there exists a binary linear code of length n and at most r parity check bits and minimum distance at least d, given that

$$1 + {\binom{n-1}{1}} + \dots + {\binom{n-1}{d-2}} < 2^r$$

The proof for this depends on the fact that  $d_{\min}$  equals the smallest number of columns of H that form a linearly dependent set. We construct the parity check matrix H for this code as follows: we find the number I of the total  $2^r - 1$  possible vectors of size r columns of H, such that no d-1 are *linearly dependent*. There are at most

$$\binom{i}{1} + \dots + \binom{i}{d-2}$$

distinct linear combinations of d-2 or fewer of these columns. If this number is less than the total number of vectors,  $2^r - 1$ , another linearly independent column can be added to **H**. We can continue until i = n, which proves the bound.

*Singelton Bound*: A similar approach as above is used to prove the *Singelton bound*. If C is an [n, k, d] linear code, then

$$d-1 \le n-k.$$

A code which achieves equality in the Singleton bound is called *maximum distance separable* (MDS). Here n-k is the rank of **H** and therefore the maximum number of linearly independent columns.

#### **2.15 ROBUST CODES**

The word *robust* is used here not only to refer to codes that perform well under ordinary conditions, but also under unusual conditions that stress the designers' assumptions. Here we address the problem of robust channel coding in which the signal information should be preserved in spite of intrinsic noise and fading phenomena experienced by the signal. We present a theoretical analysis and provide insights into optimal coding strategies. Normally, any robust code implementation scheme makes use of an arbitrary cooperative number of coding units to minimize the reconstruction error. This is evident in the design of the famous turbo code encoder. One common characteristic of robust encoders is that they *introduce* extra redundancy in the code in order to compensate for channel noise and fading phenomena.

## 2.15.1 Mathematical Formulation of Robust Codes [44]

Let  $C \subseteq GF(q^n)$  be an (n, M)-code, where M = |C|.

#### **Definition** 1:

The code *C* is *robust* with respect to its error-masking probability if and only if the probability Q(e) of missing an error *e* is less than one for all nonzero errors *e*:

$$Q(e) = \frac{|\{w \mid w \in C, w + e \in C\}|}{|C|} < 1, \ e \neq 0,$$

.....(2.14)

where  $w, e \in GF(q^n)$ .

#### **Definition 2**:

The code *C* is *uniformly robust* with respect to its error-masking probability iff the probability Q(e) is constant independently of the nonzero error e:

$$Q(e) = \frac{|\{w | w \in C, w + e \in C\}|}{|C|} = const., \ e \neq 0, \ const < 1.$$
(2.15)

#### **Definition 3:**

A robust code, where  $R = \max |\{w/w \in C, w+e \in C\}|$  for all errors,  $e \neq 0$  is called a *R*-*Robust code and is denoted by*  $C_R$ .

A graphic depiction of the definitions of the properties of a robust code is shown in Figure 2.5. Let *C* be the set of all code words of length *n*, and let  $\tilde{C}$  be the set of all code words of set *C* shifted by an element  $e \in GF(q^n)$ ,  $\tilde{C} = \{\tilde{w} | \tilde{w} = w + e, w \in C\}$ , then the code *C* is *robust* if for any  $e \in GF(q^n)$ ,  $e \neq 0$  the intersection of the two sets *C* and  $\tilde{C}$  is less than the size of the code:  $|C| > |C \cap \tilde{C}|$ . Additionally, if for any *e*, the size of the intersection is less than or equal to R < |C|, then the code is a *R*-robust code. If R < |C| is constant for all shifts of the code, then the code is said to be *uniformly robust*.



Fig. 2.4 Definition of code robustness

The construction of systematic robust error detecting codes is based on the use of perfect nonlinear functions [44]. While nonlinearity is a necessary condition for robustness, there is a direct relationship between good robust codes and good linear codes. For instance, in the design and construction of turbo codes, a combination of two linear codes is used to generate a nonlinear turbo code, with very good error detection and correction capabilities.

## 2.15.2 The Potential of Robust Coding Techniques

In order to compare different robust coding schemes, we need a parameter which expresses the system performance level. It is called the information bit error probability  $P_b$ , and typically falls into the range  $10^{-3} \ge P_b \ge 10^{-6}$ .

The traditional role for error-control coding has been to make a troublesome channel acceptable by lowering the frequency of error events [9]. The error events could be

bit errors, message errors, or undetected errors. The traditional role has rather now expanded tremendously and today coding can do many other things, including:

- *Reducing the occurrence of undetected errors*: This was one of the first uses of error-control coding. Today's error detection codes are so effective that the occurrence of undetected errors is, for all practical purposes, eliminated.
- *Reducing the cost of communications systems:* Transmitter power is expensive, especially on satellite transponders. Coding can reduce the satellite's power needs because messages received as close to the thermal noise level can still be recovered correctly.
- *Overcoming Jamming:* Error-control coding is one of the most effective techniques for reducing the effects of the enemy's jamming. In the presence of pulse jamming, for example, coding can achieve coding gains of over 35 dB.
- *Eliminating Interference:* As the electromagnetic spectrum becomes more crowded with man-made signals, error-control coding will mitigate the effects of unintentional interference.
- Uniformly robust codes provide for equal error detection against all errors (assuming uniform distribution of data messages) independently of error distributions. Application of nonlinearity to linear codes decreases the number of undetectable errors and redistributes the error detecting power of the original linear code reducing the potential weaknesses.
- *Robust codes* offer a significant improvement over linear codes with respect to the three criteria listed earlier by decreasing the number of undetectable errors without changing the redundancy of the code.

## 2.15.3 Robust Coding Limitations

Robust coding schemes are not a solution to all problems! Despite all the new uses of error-control coding [9], there are limits to what coding can do.

On the Gaussian noise channel, for example, Shannon's capacity formula sets a lower limit on the signal-to-noise ratio that we must achieve to maintain reliable communications. Shannon's lower limit depends on whether the channel is powerlimited or bandwidth-limited. The deep space channel is an example of a powerlimited channel because bandwidth is an abundant resource compared to transmitter power. Telephone channels, on the other hand, are considered bandwidth-limited because the telephone company adheres to a strict 3.1 kHz channel bandwidth.

- For strictly *power-limited (unlimited bandwidth) channels*, Shannon's lower bound on E<sub>b</sub> / N<sub>o</sub> is 0.69, or -1.6 dB [16]. In other words, we must maintain E<sub>b</sub> / N<sub>o</sub> of at least -1.6 dB to ensure reliable communications, no matter how powerful an error-control code we use.
- For *bandwidth-limited channels with Gaussian noise*, Shannon's capacity formula can be written as follows [16]

$$\frac{E_b}{N_o} \ge \frac{2^r - 1}{r} , \qquad (2.16)$$

where, r is the spectral bit rate in bits/s/Hz. For example, consider a bandwidthlimited channel operating with un-coded quadrature phase shift keying (a common modulation technique with 2 bits/symbol and a maximum spectral bit rate of r = 2 bits/s/Hz) and a required BER of  $10^{-5}$ . We know that without coding, this communications system requires an  $E_b / N_o$  of 9.6 dB [1]. Shannon's formula above says that to maintain reliable communications at an arbitrarily low BER, we must maintain (for r = 2 bits/s/Hz) an  $E_b / N_o$  of at least 1.5 (1.8 dB). Therefore, if we need to lower the required  $E_b / N_o$  by more than 7.8 dB, coding can not do it. We must resort to other measures, like increasing transmitter power. In practice, the situation is worse because we have no practical code that achieves Shannon's lower bound. A more realistic coding gain for this example is 3 dB rather than 7.8 dB.

• Another limitation to the performance of error-control codes is the modulation technique of the communication system. Coding must go hand-in-hand with the choice of modulation technique for the channel. Even the most powerful codes cannot overcome the consequences of a poor modulation choice.

## 2.16 THE ROLE OF COMPUTATIONAL SCIENCE: MODELING AND SIMULATION

Computational science is a novel area of basic and applied research in highperformance computing, applied mathematics, intelligent systems and information technologies. It is a new and cost-effective way of solving problems and the outcome is powerful software tools for students, lecturers, government researchers, and industrial scientists. Computational science requires basic understanding of fundamental concepts like modeling, analytical solution, numerical solution, analytical- and numerical-based modeling, simulation, model validation, code verification through canonical tests or comparisons, accreditation, etc.

An *analytical solution* is the solution based on a mathematical model (usually in differential and/or integral forms) of the physical problem in terms of known, easily computable mathematical functions, such as sine, cosine, Bessel, Hankel functions, etc.[17]. A *numerical solution* is a solution based on direct discretization of the mathematical representations by using numerical differentiation, integration, etc. A semi-analytical solution is in between these two and is the solution based on partially derived mathematical forms that are computed numerically.

A *model* is defined as a *physical* or *mathematical abstraction* of a real world process, device, or concept. Simulation is concerned with modeling of real-world problems. Simulation in engineering usually refers to the process of representing the dynamical behavior of a real system in terms of the behaviour of an idealized, more manageable model-system implemented through computation via a simulator. A *system simulator* is a program or a series of programs developed to implement and execute simulations with well-defined relations between model objects. These relations are, by definition, mathematical relations, numerical or not.

The advantages and disadvantages of computer simulations are somewhat obvious. On the plus side, simulations are cheap (generally- supercomputer time can be expensive) and just cost computer time. Simulations can be used to model population dynamics without going out in the field to count the animals. For the researcher, simulations allow total control over every parameter in the system. On the other hand (down side), simulations only model what they are told to model, not necessarily reality. The simulation may even work in special cases, but changing any of the parameters might cause it to behave unpredictably. The cornerstone of good simulation science is constant communication with experimentalists. A healthy respect for the limitations of the simulation, and efforts to verify simulated models with real data, strengthen both the validity of the simulation, and the understanding of the experimental results. The research conducted here involves the modeling and simulation of the transmission channel of a wireless radio communication system.

## 2.17 CHAPTER SUMMARY

Error-correcting codes are at the heart of most modern data transmission and storage systems. Our current research effort focuses on robust coding for emerging wireless data communication systems. The basic idea of coding is to add redundancy to data for transmission so that even in the presence of some noise and distortion introduced by the transmission channel or storage system, the original data can be recovered error-free. A well-designed coding system adds the minimum amount of redundancy to achieve the desired level of robustness.

The purpose of the literature review has been to show the connections between information theory, coding theory, and how these theories have been exploited in the development of robust coding techniques for reliable data transmission in a hostile wireless environment, namely a transmission channel faced with problems of multipath fading, interference from other radio signal sources and noise.

The aim of information theory is to model the wireless channel impairments in a quantitative way, mainly by statistical models. The modeling is used to adduce the performance limits, and to devise methods for efficient transmission of data over the channel – that is, *coding algorithms*.

The ultimate goal of the coding is to exploit the communication channel as well as possible. Our major goal is to spend as little as possible of the limited physical resources at our disposal, including time, bandwidth, transmit power, or disk space - on the transmission or storage of information, in order to maximize the number of users, systems, or services that are able to share resources [6]. We need at the same time, to ensure that the *quality* of the information retrieved at the receiver end is satisfactory.

## CHAPTER THREE WIRELESS CHANNEL MODELS

## 3.1 WIRELESS SIGNAL TRANSMISSION

Within the last ten years, we have witnessed an explosive increase of various communication applications operating over the wireless media. It has therefore become crucial to investigate the fundamental performance limits on reliable communication over wireless channels, in which the phenomenon of fading poses critical challenges as well as promising opportunities to communication engineers. Fading is one of the major forms of volatility in a wireless environment, taking place in all dimensions, i.e., time, frequency, and space. Due to fading transmitted signals can occasionally suffer from deep signal-to-noise ratio (SNR) attenuation, severe phase distortion, and intersymbol interference, all of which notoriously degrade the quality of communication. On the other hand, by intelligently exploiting the inherent diversity of fading processes, the chance of successful reception as well as the achievable rate of transmission can both be dramatically improved [18].

The wireless channel presents a fundamental technical challenge to reliable communications, constrained by propagation conditions and the Shannon capacity equation. Signal transmission over the wireless channel is done by modulating a radio frequency carrier with the message waveform. This signal arrives at the destination (the receiver) via multiple paths . These multiple path signal components, caused by reflection off objects in the transmission environment, can arrive with different delays (see Fig.3.1) and from different directions. This results in an overall received signal having delay spread and an angular spread [19].

Another characteristic of wireless signal propagation is that the transmitter or the receiver or the reflecting objects in the environment can be moving. This motion

results in a Doppler frequency shift [19] in the received signal. This mobility (along with the multipath) causes a Doppler (or frequency) spread in the received signal.

Therefore multipath propagation results in several transmission impairments. It results in a Doppler spread due to channel time-variation. It also results in a delay spread and an angular spread in the received signal.

There are other effects on the received signal that are due to average propagation loss arising from the square law spreading, absorption by objects in the environment, etc. Long term channel variations (also called shadowing) are caused by signal attenuation arising from buildings and natural features (such as mountains, trees, etc.). This also occurs when new reflecting objects appear in the propagation environment.



Fig. 3.1 Multipath Radio Signal Propagation

Another characteristic of wireless signal transmission is that the frequency spectrum is shared between several users. In practice, this is done by separating the users by using either *time-division multiple access* (TDMA), *frequency division multiple*  *access*(FDMA) or *code division multiple access*(CDMA). In addition, in a cellular structure, different geographical areas re-use the spectrum if they are separated far enough apart.

The result of these schemes is the presence of interference from other users in the received signal. *Therefore, the major impairments in wireless communication arise from mobility (channel time variation), multipath propagation (delay spread and angular spread) and co-channel interference due to spectral sharing (transmission band overlap).* 

## 3.1.1 Wireless Channel Modeling and Simulation

In order to explore the limits of transmission in a wireless environment and also to develop suitable algorithms, we need to understand the mathematical model of the propagation environment. A radio signal propagation model is a set of mathematical expressions, diagrams and algorithms used to represent the behaviour of radio waves in an indoor or outdoor communication channel. Propagation models are used in the design and development of wireless communication networks. The ability to accurately predict radio propagation behaviour for wireless personal communication systems such as cellular mobile radio is crucial to system design.

Since site measurements are costly, propagation models have been developed as suitable, low cost, and convenient system design alternatives. Channel modeling is for instance required to predict path loss associated with the design of cellular phone base stations, as this tells design engineers how much a transmitter needs to radiate so as to service a given region.

A typical network consists of a transmitter, a receiver, and the surrounding environment. A model can be used for a certain frequency band to predict with a high degree of accuracy the radio signal behaviour of the surrounding atmosphere. Fig. 3.1 illustrates the transmitter-receiver chain in a wireless channel. Let us assume that the impulse responses of the transmitter and receiver systems are g(t) and f(t)respectively. Let us further assume that the impulse response of the physical channel is given by  $c(t, \tau)$ .
The overall channel response including the transmit and receive filters is given by,

$$h(t,\tau) = \int_{\varsigma} \int_{\beta} f(t-\varsigma)c(\varsigma,\beta)g(\tau+\varsigma-t-\beta)d\beta d\varsigma \dots (3.1)$$

This is the general form of the time-varying impulse response at time t due to an impulse at t- $\tau$ . The physical channel response  $c(t, \tau)$  captures the effects of channel time-variation, multipath delay spread and angle spread of the propagation environment.

The performance of a communications system (or any other system) depends on *design parameters* whose values can be selected by the system designer and *environmental parameters* over which the designer has no control. The relationship between these parameters and performance metrics of interest is usually complex. In general, changing any single design parameter tends to impact on all performance metrics of interest, and simultaneously, changing multiple design parameters typically affects performance metrics in ways that cannot be predicted from knowledge of the single parameter effects alone.

The goal of a *channel simulation process* is to select the design parameters so as to achieve specific performance levels (or the best performance possible) subject to constraints on system cost (cost can thus be viewed as another performance metric). Some of the choices the designer must make are essentially discrete or integer valued, i.e., a selection among a small (or at least finite) set of alternatives.

Error performance simulation and prediction is quite essential, because the fading and noise phenomena, which greatly influence the performance of the wireless communication channel, are pretty complex, and they are also usually accompanied by other random environmental parameters. The simulation process does not only lead to quick results, but also enables design tradeoffs to be investigated in a cheaper way.

Simulation is increasingly applied in wireless system design problems, due to various reasons:

- The designer is faced with a huge design space (each design parameter can be thought of as one dimensional in a multidimensional space). Exhaustive exploration of this space is typically impractical. Thus, the designer must rule out many alternatives early in the design process on the basis of experience (his own or others') in order to consider a smaller, more manageable set of alternatives that can be evaluated through simulation.
- Even without detailed modeling of the physical layer, high-fidelity simulations of large networks tend to require large amounts of computation. One cannot scale down networks for purposes of performance evaluation because the behaviour of networks involving small numbers of nodes may be very different.
- The external environment in which a system must operate is often highly uncertain. Terrain type, presence of interfering equipment, jamming, and other external factors can all impact performance, but are difficult to accurately characterize and model. In the case of jamming, uncertainty about the threat is a major issue.

Although simulation is an essential part of the process by which competing design alternatives are compared and inferior ones winnowed out [1], simulations of wireless communications links and networks are notoriously unreliable, and the results obtained from such simulations should be assessed carefully with caution.

For any type of stochastic system, there are at least four general ways in which simulation studies can lead to incorrect design choices:

- Errors in the underlying mathematical models of the system and its environment, in associated data, or in the process by which models are fitted to the available data. Errors in modeling include unwarranted approximations and simplifications.
- Faulty implementation of models in code (e.g., programming errors or use of faulty random number generators).
- Improper inputs, e.g., inputs that violate the range of validity of the underlying models, or insufficient exploration of the parameter space.
- Errors in statistical processing and interpretation of the simulation outputs. Examples include:
  - Insufficient sample sizes (numbers of runs and/or run lengths),
  - Failure to account for dependence in simulation outputs, and

• Untested assumptions about the behaviour of the model, e.g., that interaction between design parameters can be ignored.

Our focus here is on the specific aspect of the first bullet above (errors in the underlying models) – *the problem of characterizing the external environment in which a wireless system/network must operate.* The external environment here is the *communications channel* between a pair of antennas. The channel accounts for propagation effects such as ordinary  $1/d^2$  free space path loss, rain absorption, multipath fading, diffraction, refraction, and scattering, as well as general background noise. In the widest sense, the channel may also account for sources of interference when these are treated in aggregate. In any case, sources of interference, whether friendly (unintentional) or jammers, are part of the external environment.

Wireless channels are difficult and capacity-limited communications media. They differ a lot from wired channels, due to their unreliable (random or stochastic) behaviour compared to wired channels. In wireless channels, the state of the channel may change within a very short time span. This random and drastic behaviour of wireless channels turns communication over such channels into a difficult task. With wireless channels, many different propagation environments can be identified, such as urban, suburban, indoor, underwater or orbital environments, which differ in various ways.

In this research, we however, focus our attention on factors which influence the performance of terrestrial wireless channels. We consider analytic models of basic propagation effects encountered and show how they translate into the performance of different communication systems. For example, this knowledge is crucial in facilitating the design and parameterization of simulation models of wireless channels.

### 3.1.2 The Signal Fading Phenomenon

Fading is the variation in the signal strength as the signal propagates from the transmitter to the receiver, and this is accompanied by varying signal levels at different points in the channel. In a mobile radio channel, the transmission path varies from a simple line-of-sight to one severely obstructed by buildings, foliage, hills and

mountains. The term fading is used to describe the rapid fluctuation of amplitudes, phases or multipath delays of a radio signal over a short period of time or travel distance, so that the large-scale path loss effects may be ignored. Multipath propagation is caused by interference between two or more versions of the transmitted signal which arrive at the receiver at slightly different times. These waves, called *multipath waves*, combine at the receiver antenna to give a resultant signal which can vary widely in amplitude and phase, depending on the distribution of the intensity and relative propagation time of the waves and the bandwidth of the transmitted signal.

Due to multipath propagation, multiple paths have varying path length, experience varying interactions with objects in the paths and varying path loss. The extent of the fading depends on the vegetation types, terrain, weather conditions such as humidity and precipitation in the channel, etc. The mechanisms of electromagnetic wave propagation include reflection, diffraction, shadowing and scattering.

#### 3.1.3 Signal Propagation Models

Propagation models are employed to predict the average received signal strength at a given distance from the transmitter, as well as the variability of the signal strength in close spatial proximity to a particular location. Fading can be classified as either *small- scale fading* or *large-scale fading* [19].

**Small-scale fading** refers to rapid fluctuations of the amplitude of the received signal strength over very short travel distances or short periods of time. This fading can be modeled using small-scale fading models. The statistical time-varying nature of the received signal envelope in small scale fading can be described using either the *Rayleigh* or the *Rician* fading distribution models. The distribution is Rician when there is a dominant stationary (direct) signal component such as a line-of-sight propagation path. In this case, random multipath components arrive at different angles and are superimposed on the dominant signal component. If a dominant signal component is absent, the fading is described using a Rayleigh distribution. Multipath propagation creates small-scale fading effects in the wireless channel. The Rayleigh distribution is commonly to describe the statistical time varying nature of the received envelope of a flat fading signal, or the envelope of an individual multipath component.

The three most important factors which contribute to the small-scale fading process include:

- Rapid changes in signal strength over a small travel distance or time interval.
- Random frequency modulation due to varying phase shifts on different multipath signal components. The relative motion between the base station and the mobile cellular phone results in the random frequency modulation and if objects in the radio channel are in motion, they induce a time-varying Doppler shift on the multipath signal components.

**Large scale fading** refers to variation of the signal strength over large distances, and long periods of time between the transmitter and the receiver. This fading can be described using large scale propagation models. Large scale propagation models predict the mean signal strength for an arbitrary transmitter-receiver distance. They characterize signal strength over large transmitter-receiver separation distances and are used in estimating the radio coverage area of a transmitter. Examples of large – scale fading models include:

- The free-space propagation model
- The ground reflection ray (2-ray) model
- The *log-distance* and *log-normal* shadowing models [19],[27].

#### 3.1.4 The Mobile Communications Channel

Mobile cellular communication channels are modeled as fading dispersive channels. Communication over such channels is severely degraded because of the time-and frequency-selective nature of such channels. Fading dispersive channels are modeled as time-varying filters and are completely characterized by their *scattering functions*. The scattering function  $\sigma(\tau, f)$  measures the average received power as a function of the time delay,  $\tau$ , and the Doppler spread *f*. Widths of the scatter function in time L and in frequency B reflect the spread of the channel in time and frequency respectively [24].

The spread in time is due to the multipath effect that is inherent in the channel while the spread in frequency is due to the time variation of the channel characteristics. Channels with B=0 are spread only in time and are known as time-flat or frequencyselective channels, while channels with L = 0 are dispersive only in frequency and are labeled as frequency flat or time-selective [41]. The quantity S = BL measures the overall channel spread. Channels with overall spread S < 1, are called under spread channels. Examples of such channels are long distance ionospheric HF propagation for which S=0.6. Channels with  $S \ge 1$  are known as overspread channels. Non-dispersive, but fading, channels are characterized by zero spreads in both time and frequency. Such channels are, in essence, the random phase Rayleigh fading channels discussed extensively in section 3.3.1.

#### **3.2 THE WIRELESS CHANNEL MODEL**

Assuming ideal antennas, the propagation channel becomes identical to the radio or wireless channel. The wireless channel attenuates the received signal by a time varying factor, denoted by a(t). This attenuation may be compensated by the modulation channel, since amplifiers are employed here to boast the received signal. However, at the modulation channel, random time varying *noise* n(t) also enters the system, adding a distorting element to the signal. If the attenuated signal is largely amplified, the noise will also be amplified strongly. Therefore, reliable detection methods have to be used as part of the demodulator to extract the transmitted information signal from the noisy signal. In addition to the noise which is always experienced at this stage of the communications system, there is also a possibility of electromagnetic waves from other sources, interfering with the carrier signal. These unwanted signals are usually referred to as *interference*. Interference has a significant impact on the performance of the channel, similar to noise. The interfering signal is also time variant and is denoted by i(t). The resulting mathematical model of the received signal y(t), depending on the sent signal s(t) and all influencing factors is given in Fig. 3.2;



Fig.3.2 Mathematical model of the wireless channel.

The specific propagation effects that must be accounted for in a channel model depend on the type of system (including the frequency of operation; symbol rate, modulation, coding, and other waveform characteristics; antenna types, and antenna heights), the terrain, rates of movement, and other geometric factors (e.g., distances between antennas and distances to reflective surfaces). Diffraction, refraction, and scattering are often of secondary importance and can be ignored for many applications.

Multipath propagation phenomena are, however, an important effect that is often disregarded or improperly modeled. It is recognized as the major cause of bit errors at especially high data transmission rates.

For low-data-rate systems, multipath fading can often be represented as *flat fading*, i.e., as *a time-varying attenuation that affects the amplitudes but not the shapes of the received signal pulses*. For higher-data- rate systems, however, multipath phenomena cause distortion of pulses and intersymbol interference; these effects can be crucial in determining waveform parameters and receiver characteristics, including equalization, rake reception, error control coding and interleaving, and the use of spread spectrum techniques.

## 3.2.1 Mathematical Modeling of the Wireless Channel

The wireless channel influences the received signal by a multiplicative factor, here referred to as the attenuation a(t). However, this attenuation factor is a product of several components. Analytically a(t) represents the overall attenuation of the transmitted signal and it is constituted by three different effects: namely *path loss*, *shadowing* and *multipath fading*.

**Path loss** is a deterministic effect depending only on the distance between the transmitter and the receiver and in most situations does not change significantly on smaller time scales.

**Shadowing** is not deterministic and causes fluctuations of the received signal strength at points with the same distance from the transmitter. Shadowing is sometimes referred to as slow fading, while **multipath fading** is referred to as fast fading.

Multipath fading is also stochastic in nature, and leads to significant attenuation changes within smaller time scales such as milliseconds or even greater. The three attenuation factors combined result in the actual experienced attenuation of the wireless channel. This attenuation is an aggregated function

$$a(t) = a_{PL}(t).a_{SH}(t).a_{MF}(t), \qquad (3.3)$$

where,  $a_{PL}(t)$  is the path loss factor,  $a_{SH}(t)$  is the shadowing attenuation factor, and  $a_{MF}(t)$  is multipath fading factor. Multipath fading may have a time-varying or frequency varying attenuation impact on the transmitted signal.

#### 3.2.2 The Physical Basis of Multipath Fading

The physical basis of multipath fading is exemplified by the reception of multiple copies of the transmitted signal, each having followed a different path. In a typical environment, each path has a different path length  $l_i$ . Due to this difference in path length, each signal traveling along a path arrives with a different *delay*,  $\tau_i = l_i / c$ , where, c is the speed of light. Some signal copies traveling along shorter paths will arrive later.

Physically, this is similar to an echo encountered in a canyon. The channel is said to have a memory, since it is able to store signal copies for a certain time span. Besides this multipath propagation, each signal copy is attenuated differently, since the signal paths have to pass different obstacles like windows, building walls of different materials, trees of different sizes and so on.

Let us denote the attenuation factor of path i by  $a_i$ . Taking all this into account, the multipath propagation of a transmitted radio wave results in an interference pattern, where at certain points, the wave components interfere constructively, while at other points they interfere destructively.

If each element within the propagation environment (transmitter, scatterers, and receiver) do not move, the propagated signal will only suffer from the delay spread and the varying attenuation. In this case, the interference situation of the channel stays constant and therefore the channel is said to be *time invariant*.

In contrast, if any kind of movement is encountered in the propagation environment, all or some paths change in time, such that all  $a_i$  and  $\tau_i$  change in time. As a consequence, the wireless channel becomes *time variant*. Here, along with a constant changing delay spread, the receiver also experiences varying signal strength due to its movement through the interference pattern, therefore, the received signal fades.

# 3.2.3 The Mathematical Model of Multipath Fading

Let us consider the transmission of a band pass signal at carrier frequency,  $f_c$ , with

complex envelope s(t). This transmitted band pass signal can be represented by the equation

$$s(t) = \operatorname{Re}\left\{\bar{s}(t) \exp(j2\pi f_c t)\right\} \dots (3.4)$$

The received band pass signal is given by the equation

$$y(t) = \operatorname{Re}\left\{\bar{y(t)} \exp(j2\pi f_c t)\right\} \dots (3.5)$$

We are interested in deriving a mathematical model of the received bandpass signal taking into account the effect of multipath propagation. Let us consider first the case where we do not encounter motion in the environment. Each path will then be associated with a different path length,  $l_i$ , and a different attenuation,  $a_i$ . Therefore, the received signal y(t) is the superposition of all copies, as given below.

$$y(t) = \sum_{\forall i} a_i \cdot s\left(t - \frac{l_i}{c}\right) \qquad (3.6)$$

$$= \operatorname{Re}\left\{\sum_{\forall i} \bar{a_i} \cdot \bar{s}(t - \frac{l_i}{c}) \cdot \exp(j2\pi f_c(t - \frac{l_i}{c}))\right\} \quad \dots \quad (3.7)$$

Since wavelength,  $\lambda = \frac{c}{f_c}$ , the complex envelope of the signal can be expressed as:

$$\overline{y}(t) = \sum_{\forall i} a_i . \exp(-j2\pi \frac{l_i}{\lambda}) . \overline{s}\left(t - \frac{l_i}{c}\right)$$

$$= \sum_{\forall i} a_i . \exp(-j\varphi_i) . \bar{s}(t-\tau_i) , \qquad (3.8)$$

where,  $\varphi_i = 2\pi \frac{f_c l_i}{c} = 2\pi \frac{l_i}{\lambda}$  is the phase shift of the carrier frequency caused by the different length of each path, and  $\tau_i = \frac{l_i}{c}$  is the signal delay along the *i*-th path. Now, let us consider the effect of motion: denote  $\gamma_i$ , as the angle of arrival of path *i* with respect to the direction of motion of the receiver.



Fig. 3.3 Sketch of signal arrival components.

The path length change, as a function of speed, v, and time, t, is given by the expression:

$$\Delta l_i = -v.\cos(\gamma_i)t. \qquad (3.9)$$

The complex envelope of the received signal can now be expressed as

$$\overline{y}(t) = \sum a_i . \exp(-j2\pi \frac{l_i + \Delta l_i}{\lambda}) . \overline{s}(t - \frac{l_i + \Delta l_i}{c})$$

or

$$\bar{y}(t) = \sum_{\forall i} a_i . \exp(-j\varphi_i) . \exp(-j2\pi . \cos(\gamma_i) t . \frac{v}{\lambda}) . \bar{s}(t - \tau_i + \frac{v . \cos(\gamma_i) t}{c})$$
.....(3.10)

Further simplification of equation (3.10) is possible if we introduce a **Doppler** frequency variable,  $f_d = \frac{f_c}{c} \cdot v = \frac{v}{\lambda}$  and a Doppler shift  $v_i = \cos(\gamma_i) \cdot f_d$ . With these two, we have:

$$\bar{y}(t) = \sum_{\forall i} \bar{A}_i . \exp(j2\pi . \cos(\gamma_i) t. f_d) . \bar{s}(t - \tau_i)$$

The influence of the motion of the receiver in combination with the *i*-th scatterer is embedded in the amplitude of the carrier ( $\overline{A_i}$ ), in the carrier frequency ( $v_i$ ) and in the delay ( $\tau_i$ ) of the envelope.

In practice, the multipath fading phenomena has a frequency-varying or a timevarying attenuating impact on the transmitted signal.

## 3.2.4 Characterization of the Channel in Time and Frequency: Doppler Spread and Time Spread

Although many factors influence the quality of the signal received in a multipath propagation environment, motion, frequency offset (Doppler shift) of the carrier and time delay of the envelope are the major causers of signal damage. This is because the shifted and delayed wave components often interfere destructively, causing severe attenuation.

In practice, wireless transmission of a signal in an environment, which includes some motion of objects is described by two variables – the Doppler spread,  $\Delta f_d$ , and the delay spread,  $\Delta \tau$ . Both spreads result from multipath reception of the carrier signal (and in the case of the Doppler spread, also from the velocity involved), where each path may be characterized by a different Doppler shift (due to a different receive angle) and time delay. While Doppler spread is caused by the motion of objects within the environment (which might be the transmitter, the receiver or scatters), the delay spread is caused by the topology of the environment itself [21].

Although the Doppler spread is a phenomenon in the frequency domain (generating a Doppler shift – which is a shift in frequency), the overall result is a time selective behaviour. For the delay spread, this is exactly the opposite. While the delay spread is a phenomenon in time, the resulting impact on the received signal is a frequency selective behaviour.

Considering a receiver moving through a multipath environment with a fixed speed, and assuming all path delays in this environment are negligibly small, such that  $\bar{s}(t - \tau_i) \approx s(t)$ , the complex envelope of the received signal will then be given by the expression:

$$\bar{y}(t) = \bar{s}(t) \sum_{\forall i} \bar{A}_i \exp(j2\pi\cos(\gamma_i) \cdot f_d \cdot t) \dots (3.12)$$

or,

$$\bar{y}(t) = \bar{s}(t).\bar{g}(t)$$
 .....(3.13)

where,  $\bar{g}(t)$  is called the *complex gain* of the channel. We note that here the input  $\bar{s}(t)$  and the output  $\bar{y}(t)$ , of the channel are related by a simple multiplicative operation.

#### 3.2.5 Free-Space Power Loss

The attenuation of a signal propagating in free space over a distance of d meters between two antennas can be derived from Maxwell's equations [21]:

$$\frac{P_{o}}{P_{t}} = \left(\frac{\lambda}{4\pi d}\right)^{2} g_{Tx} g_{Rx} \qquad (3.14)$$

or in dB

where,  $P_o$  is the received power,  $P_t$  is the transmitted power,  $\lambda$  is the wavelength of the signal,  $g_{Tx}$  is the gain of the transmitter antenna and  $g_{Rx}$  is the gain of the receiver antenna (both gains being in the direction of the straight line that connects the two antennas in space. The received power is inversely proportional to the square of the distance *d* and the square of the signal frequency.

#### 3.2.6 The Two Ray Model

Since most wireless communications happen close to the earth's surface, the scenario of free-space power loss is unrealistic. The two-ray model, also known as plane earth [21], is a simple model based on physical-optics theory, which takes the reflection of the earth's surface into account. It also assumes a line-of-sight (LOS) communication

and no other influence on the wave propagation besides the earth's surface. It is a useful starting point for the study of propagation of radio waves for personal communications.

The derivation of the two ray model takes into account only two waves: the direct wave, from the transmitter to the receiver, and the reflected wave from the earth's surface. This scenario is illustrated in Fig.3.4.

From the free-space propagation model, the power of the direct wave at the receiver is given by the expression:

$$P_{R1} = P_t \left(\frac{\lambda}{4\pi d_1}\right)^2 g_{Tx} g_{Rx} \qquad (3.16)$$

The power of the wave component reflected from the earth's surface at the receiver is calculated using laws of reflection of plane waves [21] and is given by the expression:

$$P_{R2} = P_t R(\vartheta, \varepsilon_{eff}) \left(\frac{\lambda}{4\pi d_2}\right)^2 g_{Tx} g_{Rx} \qquad (3.17)$$

where  $\varepsilon_{eff}$  is the effective electric permittivity of space and the magnetic permeability of free space being assumed to be 1 (i.e., no magnetic properties present);  $R(\mathcal{G}, \varepsilon_{eff})$  is the effective reflection factor, which takes on values near -1 at the surface.



# Fig. 3.4 Diagram illustrating propagation of the signal assuming a two-ray model.

The total power received  $(P_{R1} + P_{R2})$  is then given by

$$P_{R} = P_{t} \left(\frac{\lambda}{4\pi}\right)^{2} g_{Tx} g_{Rx} \left[\frac{1}{d_{1}} + R(\mathcal{G}, \mathcal{E}_{eff}) \frac{e^{j\Delta\Phi}}{d_{2}}\right]$$
....(3.18)

by applying the superposition principle to the arriving electric field strengths, where  $\Delta \Phi = 2\pi (d_2 - d_1)\lambda$  is the phase difference between the two waves, and  $d_1$  and  $d_2$  are distances covered by the direct and reflected wave fronts. Further reduction of the expression in equation (3.18) is possible [21] by assuming that the angle of incidence of the reflected ray is very near 90 degrees, leading to a simplified version (3.19):

$$P_{R} = P_{t} \left(\frac{\lambda}{4\pi d}\right)^{2} g_{Tx} g_{Rx} \cdot 2 \cdot (1 - \cos \Delta \Phi)$$

.....(3.19)

or

$$P_{R} = P_{t} \left(\frac{\lambda}{4\pi d}\right)^{2} g_{Tx} g_{Rx} \cdot 4.\sin^{2} \frac{2\pi h_{Tx} h_{Rx}}{\lambda d}$$

.....(3.20)

$$= P_t \left(\frac{\lambda}{4\pi d}\right)^2 g_{Tx} g_{Rx} \cdot 4.\sin^2(\Delta \Phi/2),$$
.....(3.21)

for horizontal polarization.

For values of  $\Delta\Phi$  smaller than 0.6 radian,  $\sin \Delta\Phi/2 \approx \Delta\Phi/2$  and the expression can be simplified to the known  $4^{th}$ -power-law form:

$$P_{R} = P_{t} g_{Tx} g_{Rx} \left( \frac{h_{Tx} h_{Rx}}{d^{2}} \right)^{2}, \qquad (3.22)$$

where the dependence on frequency vanishes.

Plots of the received power,  $P_R$ , can be plotted as a function of the distance between the transmitter and the receiver, according to the free-space loss, the two-ray model, and the 4<sup>th</sup>-power law.

Parameters used for example are:

- Antennas with unit gain: i.e.,  $g_{Tx} = g_{Rx} = 1$ .
- Antenna heights:  $h_{Tx} = 25$ m;  $h_{Rx} = 1.5$ m
- Transmitted power: 0dB (1mW)
- Transmission frequency (carrier) : 2.4 GHz.

Simulation results indicate that for the two-ray model there are clearly two different areas: near the transmitter (before a breakpoint), where the received power decreases according to the square sinus function, with peak value following the square of the distance; and after the breakpoint, when the phase difference between direct and reflected rays is smaller than 0.6 radian, the second approximation (3.20) becomes valid, and the received power decreases with the 4<sup>th</sup> power of the distance (so that the difference between the curves can no longer be seen) [21]. The breakpoint can be calculated according to

$$d_{Breakpoint} = d \mid_{\Delta \Phi \le 0.6} = \frac{2\pi h_{Tx} h_{Rx}}{o.6\lambda}$$
(3.23)

The theoretical results obtained from the simulation exercise are in agreement with measured results in practice [24]. For instance, the 2-ray model fits quite well with the actual path loss in line-of-sight (LOS) environments, with few or no reflectors and scatters, e.g., on highways.

Assumptions made in connection with the above three models discussed so far:

- LOS propagation was assumed.
- No other objects surrounding the path or the transmitter and receiver.
- There was no relative motion between the transmitter and the receiver.

These assumptions are not valid in many realistic environments like urban, suburban and indoor environments, where non-LOS (NLOS) is a common feature and mobility could be involved. In a more realistic situation, a multitude of physical phenomena influence the propagation of electromagnetic waves and the number of possible propagation paths is very high.

# 3.2.7 Empirical and Semi-Empirical Models

Radio signal propagation models can be categorized as either analytical models or empirical models. Analytical models are obtained from analyzing electromagnetic (EM) wave propagation phenomena. Examples of analytical models are the free space model and the two-ray model. Empirical models are obtained from data measurements. They implicitly take into consideration many known and/or unknown effects. They are good for the experimented frequency/environment. Composite models are a combination of analytical and empirical models.

Ray-tracing models are accurate path models which calculate the ever possible path loss (attenuation) suffered in each path, and finally add all the signal components which arrive at the receiver. Path loss models play a significant role in radio systems planning. They predict large scale coverage for mobile systems; they are used to estimate and predict signal-to-noise ratios (SNR) and are used for link-budget design.

However, these methods not only require exact data about the terrain, the buildings and vegetation, but are also very demanding in terms of computing capacity to process all the data and therefore extremely time consuming.

Due to these reasons, empirical and semi-empirical models were developed [21] to calculate the path loss between a transmitter and a receiver at a certain distance from each other in specific environments for different frequencies.

Whereas ray-tracing models were based on extensive measurement campaigns in different environments, empirical models are based on a mix of empirical and theoretical data. For every new area, calibration measurements are required to calculate correction factors for the general models. The models are usually of the form:

$$\frac{P_R}{P_t} = K \cdot \frac{1}{d^{\gamma}} \tag{3.24a}$$

or, in dB,

$$\frac{P_R}{P_t}[dB] = 10.\log(K) - 10.\gamma.\log(d), \qquad (3.24b)$$

where, the constants *K* and  $\gamma$  are fitted to measurement results according to the areas under consideration. The factor *K* usually depends on the frequency used, as well as height of the base station antenna and the wireless terminal. The distance d is in units referenced to some reference distance, and has to be defined along with the path loss exponent  $\gamma$ .

Note that no difference is made between LOS and NLOS anymore, since the models are obtained under both conditions, and the different propagation effects are approximately condensed in a single parameter,  $\gamma$ . Typical values of  $\gamma$  for various environments are given in the table below.

Some of the well-known and widely used empirical models are

- The Okumura-Hata Model, and
- The Lee Model.

# The Okumura-Hata Model [21]

The Okumura-Hata model is the most popular of the empirical models. It is based on extensive measurements made by Okumura in Japan and on a formula developed by Hata which approximates the measured statistics. It is valid for the following values of the parameters:

•	Frequency:	150	-	1000 MHz
•	Distance:	1	-	20 km
	<b>T</b>		20	• • • •

- Transmitter height over ground: 30 200 m
- Receiver height over ground: 1 10 m

The expression for the path loss is:

 $A[dB] = 69.55 + 26.16 \log (f[MHz]) - 13.82 \log(h_{Tx}[m])$ 

+  $(44.9 - 6.55 \log (h_{Tx}[m])).\log(d[m] - \beta$ 

.....(3.25a)

where,  $\beta$  is a correction factor, which depends on the environment and takes the following values:

 $\beta$  = [1.1 log ( f[MHz]) - 0.7].  $h_{Rx}[m]$  - [1.56 log (f[MHz] - 0.8]

for small cities.

 $\beta$  = 8.29 [log(1.54  $h_{Rx}[m])$ ]<sup>2</sup> - 1.1

for urban areas where  $f \leq 300$  MHz; and

 $\beta$  = 3.2 [log (11.75  $h_{Rx}[m]$ )]<sup>2</sup> - 4.97....(3.25b)

for urban areas , where  $f \ge 300$  MHz.

Three further categories of environments have corresponding correction factors to the corresponding urban formula, namely:

$$\beta_{urban} - 2.[\log(f[MHz]/28)]^2 - 5.4$$
 ....

(for suburban areas)

$$\beta_{urban} - 4.78 \ [\log(f[MHz])]^2 + 18.33 \log(f[MHz]) - 35.94$$

(.....for almost open Rural Areas)

$$\beta_2 = \beta_{urban} - 4.78 \left[ \log(f[MHz]) \right]^2 + 18.33 \log(f[MHz]) - 40.94$$
.....(3.25c)

(for open rural areas).

# Table 3.1: Typical values of the path loss exponent for various environments[27]

	γ		
Outdoor	Free space	2	
	Shadowed urban area	2.7 - 5.0	
In buildings	Line-of-sight	1.5 - 1.8	
	Obstructed	4 - 6	

The Okumura-Hata model was extended by COST (European Cooperation in the Field of Scientific and Technical Research) to the COST231 – Hata model for frequencies between 1500 MHz and 2000 MHz:

$$A[dB] = 46.3 + 33.9 \log (f[MHz]) - 13.82 \log (h_{Tx}[m])] + [44.9 - 6.55 \log (h_{Tx}[m])] . \log (d[km]) - \beta + \begin{cases} 3 \\ 0 \end{cases}$$

for big city centres and other cases respectively,

where,

 $\beta = [1.1\log(f[MHz]) - 0.7].h_{Rx}[m] - [1.56.\log(f[MHz]) - 0.8].$ 

.....(3.26)

#### **The Lee Model** [21, 23]

This model is quite popular because its parameters can easily be adapted to a new environment using measurement results. The model consists of two parts:

- The point-to-point model which takes the terrain into account; and
- The area-to-area model, based on the previous one, which reflects the effects of constructions. Details of the Lee Model are found in [21, 23].

Besides the models presented here, there are other models, which have been developed in various environments in several countries to better adapt the parameters of the general model form to the real propagation environment, since building materials, urban planning and vegetation differ from one country to another.

#### Shadowing

The Okumura-Hata and Lee path loss models discussed earlier aim at a deterministic calculation of the path loss for determined positions of the transmitter and the receiver. This is, however, not the reality in practice. In reality the position of a receiver involves also the influence of the objects surrounding it and the transmission path, as well as the terrain. Therefore, when measurements are made under several varying conditions, statistical variations are observed. For a fixed frequency and distance between a transmitter and a receiver, different values of the received signal power are measured.

Thus for a given fixed distance, frequency and transmission power, the received signal power is not deterministic, but varies due to the objects in and around the signal path. These stochastic, location dependent variations are deemed to be due to shadowing, denoted by  $\alpha_{SH}(t)$  in equation (3.2).

We note that the effect of shadowing is stationary with time, if no receiver movement and environment change are involved. Shadowing approximately accounts for the difference between the theoretically derived value for the path loss and the actual measured value at a certain point from the transmitter. We further note that, the effect of shadowing is an abstraction which reflects the result of a sum of several propagation phenomena, which occur when an electromagnetic wave propagates in an environment [21]: reflections, (e.g., on buildings and ground), refraction (e.g., through walls or windows), scattering (e.g., on buildings, trees or ground) and absorption (e.g., on forest or parks). *The calculation of the effects of every one of these phenomena for each location is not feasible (sometimes even impossible) both due to complexity and time limitations*. Therefore, shadowing is used to describe the aggregated effects of all these phenomena.

The free space model and the two-ray model predict the received power as a deterministic function of distance. They both represent the communication range as an ideal sphere. In reality, the received power at certain distances is a random variable due to multipath propagation effects, which is also known as fading effects.

The shadowing model consists of two parts: The first one is known as the path loss model, which predicts the mean received power at a distance d, denoted by  $\overline{P_r(d)}$ . It depends on a close-in distance  $d_o$ , (referred to in equation (3.20) as the break point) and used here as a reference.  $\overline{P_r(d)}$  is computed relative to  $P_r(d_o)$ , from equation (3.22) such that

$$\frac{P_r(d_o)}{\overline{P_r}(d)} = \left(\frac{d}{d_o}\right)^{\gamma} \tag{3.27}$$

The path loss is usually measured in dB. So from equation (3.27) we have

$$\left[\frac{\overline{P_r(d)}}{P_r(d_o)}\right]_{dB} = -10\gamma \log\left(\frac{d}{d_o}\right)$$
....(3.28)

The second part of the shadowing model reflects the variation of the received power at a certain distance. It is a log-normal random variable, that is, it is of Gaussian distribution if measured in dB. The overall shadowing model is represented by

$$\left[\frac{\overline{P_r(d)}}{P_r(d_o)}\right]_{dB} = -10\gamma \log\left(\frac{d}{d_o}\right) + X_{dB}$$

.....(3.29)

where,  $X_{dB}$  is a Gaussian random variable with zero mean and standard deviation  $\sigma_{dB}$ .  $\sigma_{dB}$  is called the shadowing deviation, and is also obtained by measurement [27]. Equation (3.29) is also known as a log-normal shadowing model. The shadowing model extends the ideal circle model to a richer statistical model: nodes can only probabilistically communicate when near the edge of the communication range.

From measurements of path loss for a variety of environments and distance [21], the variations of the measured signal level relative to the average predicted path loss are found in practice to exhibit a normal distribution with 0 mean in dB, which implies a log-normal distribution of the received power around the mean value corresponding to the path loss.

The shadowing variations of the path loss can, therefore, be calculated from the distribution[27]:

$$p(a_{SH}) = \frac{1}{\sigma_{SH}\sqrt{2\pi}} \exp\left(\frac{-a_{SH}^{2}}{2\sigma_{SH}^{2}}\right),$$
.....(3.30)

where,  $\sigma_{\rm SH}$ , is the variability of the signal and all variables are expressed in dB.

The value of the variation due to shadowing is then added to the path loss value to obtain the overall attenuation (variation):

$$a[dB] = 10\log \frac{P_R}{P_t} = a_{PL}[dB] + a_{SH}[dB]$$
 ....(3.31)

# 3.3 WIRELESS MOBILE CHANNEL DESIGN CONSIDERATIONS

The design of wireless communication networks requires accurate prediction of electromagnetic signal propagation in the targeted areas of operation. This is essential for optimal cell sizes, mobility planning, proper coverage in each of the cells and ensuring of reliable links between transmitters and mobile handsets, even for low power handset operation [25].

Many models for providing radio wave propagation characteristics are available and are used for radio coverage planning. However, these models are just approximations to the real situations and are not equally accurate in different locations or situations. Many of the available models are purely statistical and are represented by expressions that in effect are probability distribution functions of radio signal attenuation, in space and/or time.

In theory, if a model exactly represents the propagation factors at a given location, then it is possible to attain zero standard deviation, i.e.,  $\sigma = 0$ , for the variables involved. However, since by definition, models are approximations, all such models deviate from reality with deviations that reduce with the accuracy of the techniques used.

Although propagation behaviour depends on many factors, there are three major factors that are critical which influence the propagation characteristics of signals from static transmitters and receivers communicating using antenna systems that have clear heights advantage. *The three are large scale path loss, shadowing, and multipath fading.* However, for mobile terminals, other factors come into play, so that analysis based on the assumption of the prevalence of only the above three factors is no longer accurate.

The standard deviation obtained from practical measurements of the instantaneous signal attenuation depends on the resolution used for approximating the area-mean

power. Multipath fading involves interference of many scattered signals arriving at an antenna. It is responsible for the most rapid and violent changes of signal strength as well as its phase. Multipath fading phenomena is regarded as the most dominant propagation parameter. However, owing to technology limitations, system margin can no longer be considered as the sole mean of compensating for propagation disturbances at any instant in time. From a practical point of view, margins are designed in order to ensure minimum outage duration of the service for a given availability [26].

# 3.3.1 The Rayleigh Fading Model [20]

Considering propagation in a mobile environment, the receiving antenna is assumed to receive a number of reflected and scattered waves, and because of the varying path lengths, the phases of the different components are random, and consequently the instantaneous received power is also a random variable.

Assuming an unmodulated carrier, the transmitted signal frequency,  $\omega_c$ , reaches the receiver via a number of paths, the *i*<sup>th</sup> path having an amplitude  $a_i$ , and a phase  $\varphi_i$ . If we further assume that there is no direct path or line-of-sight (LOS) component, the received signal s(t) can be expressed as

$$s(t) = \sum_{i=1}^{N} a_i \cos(\omega_c t + \varphi_i),$$

.....(3.32)

where, N is the number of paths. The phase  $\varphi_i$  depends on the varying path length, phasing by  $2\pi$ , when the path length changes by a wavelength. Therefore, the phases are uniformly distributed over  $[0, 2\pi]$ .

When there is relative motion between the transmitter and the receiver, equation (3.32) can be modified to include the effects of motion-induced frequency and phase shifts.

Let the *i*<sup>th</sup> reflected wave with amplitude  $a_i$  and phase  $\varphi_i$  arrive at the receiver from an angle  $\psi_i$  relative to the direction of motion of the antenna. The Doppler shift of this wave is given by

where, v is the velocity of the mobile system, c is the speed of light (3 x  $10^8 \text{ ms}^{-1}$ ), and the  $\psi_i$ 's are uniformly distributed over [0,  $2\pi$ ]. The received signal, s(t), can now be written as

$$s(t) = \sum_{i=1}^{N} a_i \cos(\omega_c t + \omega_{di} t + \varphi_i) \qquad (3.34)$$

Expressing the signal in terms of in-phase and quadrature components, from (3.31) we have

$$s(t) = I(t)\cos\omega_c t - Q(t)\sin\omega_c t \qquad (3.35)$$

where, in-phase and quadrature components are respectively given as

and

$$Q(t) = \sum_{i=1}^{N} a_i \sin(\omega_{di}t + \varphi_i) \qquad (3.37)$$

The envelope, R, is given by

When N is large in-phase and quadrature components are Gaussian [22]. The probability density function (pdf) of the received signal envelope, p(r), has a Rayleigh distribution given by the expression:

$$p(r) = \frac{r}{\sigma^2} \exp\left\{\frac{-r^2}{2\sigma^2} - \right\}, r \ge 0$$
 (3.39)

The multipath faded signal is simulated using MATLAB to understand the relationship between the number of paths (N) and the statistics of the received signal.

#### 3.3.2 The Rician Fading Model

The Rician model is characterized by the presence of a direct path (line-of-sight - LOS) signal in addition to the multiple components. In the presence of both a line-of-

sight component and the multipath components, the received signal can be expressed as

$$s(t) = \sum_{i=1}^{N-1} a_i \cos(\omega_c t + \omega_{di} t + \varphi_i) + k_d \cos(\omega_c t + \omega_d t)) \dots (3.40)$$

where the constant  $k_d$  is the strength of the direct component,  $\omega_d$  is the Doppler shift along the LOS path, and  $\omega_{di}$  are the Doppler shifts along the indirect paths as given by equation (3.33) above. The envelope in this case has a Rician distribution density function given by

$$p(r) = \frac{r}{\sigma^2} \exp\left\{-\frac{r^2 + k_d^2}{2\sigma} - \right\} \cdot \mathbf{I}_0\left[\frac{rk_d}{\sigma^2}\right], r \ge 0 \quad \dots \qquad (3.41)$$

where  $I_o(x)$  is the *zeroth* order of the modified Bessel function of the first kind. The cumulative distribution of the Rician random variable is given by

$$F(r) = 1 - Q\left[\frac{k_d}{\sigma}, \frac{r}{\sigma}\right], \quad r \ge 0 \quad \dots \quad (3.42)$$

where,  $Q(r, \sigma)$  is the Marcum's Q function.

The Rician distribution is often described in terms of the Rician factor K, defined as the ratio between the deterministic signal power (from the direct path) and the diffuse signal power (from the indirect paths). K is usually expressed in decibels as

$$K(dB) = 10\log_{10} \left[ \frac{k_d^2}{2\sigma^2} \right] \qquad (3.43)$$

If in equation (3.41)  $k_d$  goes to zero  $(k_d/2\sigma^2 \ll r^2/2\sigma^2)$  the direct path is eliminated and the envelope becomes Rayleigh fading model.

To simulate the presence of a direct component, the received signal is modeled using equation (3.40). The RF signal and the envelope corresponding to N = 10 can then be plotted using MATLAB.

#### 3.4 NOISE AND INTERFERENCE MODELS

One of our major goals is to develop a solid understanding of the characteristics of mobile communication channels as well as models used to represent such channels for simulation purposes. The major wireless channel impairments are path loss, multipath propagation, and noise and interference.

While the wireless channel propagation phenomena result in an overall attenuating impact on the signal from the transmitter to the receiver, the channel noise and interference have an additive impact on that signal and this results into distortion of the signal. The most famous and important equation in communications systems design and research is that, due to Claude Shannon:

$$C = B \cdot \log_2\left(1 + \frac{S}{N}\right)$$
 bits/second (3.44)

This is often also referred to as the Shannon-Hartley equation and states that the capacity of error-free communications is limited and is both proportional to the bandwidth that the signal occupies and to the ratio of the received signal power to the received noise power. The signal-to-noise ratio term simply expresses the need to reduce natural and random noise relative to the man made communications signal. This can be done through various ways, including among others, increasing the transmitter power, using adaptive modulation and coding systems, and by improving the antenna system.

If the required information transfer rate is less than the capacity as defined by the Shannon-Hartley equation, then error free communication is possible. If information transfer at a rate greater than this limit is attempted, errors in transmission will occur no matter how well the equipment is designed. The Shannon-Hartley equation is a very good first step in evaluating the feasibility of any digital communications system design. It provides an upper bound, only achievable with infinite signal processing resources. The development of strategies to reach the Shannon limit and to overcome these issues is a very important theme of wireless communications research today. Such strategies include the development of new modulation and coding schemes.

#### 3.4.1 Noise

We generally define noise as any unwanted electrical signals which negatively affect the quality of a desired signal. But we can also classify noise in different categories such as man-made noise, thermal noise, impulse noise, and interference from other radio frequency sources. Noise and interference are generally stochastic in nature and both vary with time.

Fig. 3.5 shows an N-path mobile channel attenuation and noise model with path gain, path delay and overall additive white Gaussian noise.

Noise is always present in the modulation channel and comes from several sources, e.g., atmospheric disturbances, electronic circuitry, human-made machinery, etc. Atmospheric noise and noise from electronic devices is generally referred to as thermal noise, because it is due to movement of charged particles inside electronic components, which exist in every receiver system and is therefore unavoidable. This kind of noise is called *white* because it contains all frequencies of light.



Fig. 3.5 N-path mobile channel attenuation and noise model

Thermal noise sources have a power spectral density which depends on *the working temperature*. The *effective noise temperature* and *noise equivalent bandwidth* can be found in the relevant datasheets of electronic components. However, in mobile communications, these values are usually not relevant since the limitations to system performance are mainly due to interference and man-made noise.

Man-made noise is that produced by machines operated by humans. It is noise because it is energy radiated by machines which are not supposed to generate that energy. Their main purpose is some other function, but due to turning on and off of electrical and electronic components or to wiring, they radiate electromagnetic energy [22], which can disturb wireless communication nearby.

The characterization of man-made noise is very complicated, since many parameters have to be measured and they are always specific to the measurement environment. The parameters to be measured include average total power, power spectrum, probability distribution of the noise voltage, the pulse heights, widths and rates, as well as system specific parameters like dependence on antenna polarization, height and directivity, and long-time dependence on time and location.

Assuming that BPSK signals are transmitted and that the channel is frequency nonselective, under these conditions the channel will result in a multiplicative distortion of the transmitted signal s(t). Furthermore, if we assume that channel fading is slow, the resulting multiplicative process can be regarded as a constant during at least a one bit signaling interval. If the transmitted signal is s(t), the received equivalent lowpass signal is:

$$y(t) = a.e^{-j\varphi}s(t) + n(t)$$
  $y(t) = a.e^{-j\varphi}s(t) + n(t)$  ..... (3.45)

where n(t) represents the complex AWGN process that corrupts the signal with mean zero and variance  $\sigma_n^2 = N_o/2$ . The channel gain *a* is described by a probability density function with Rayleigh distribution. It is normalized to have a mean-square value of  $E(a^2) = 1$ , indicating that the expected received average signal energy will be  $E_c$ , which is the coded bit energy.

#### 3.4.2 Interference

Interference can be due to other systems operating in the same frequency band as your receiver, in case of unlicensed bands, or co- and adjacent-channel interference in licensed bands. *Co-channel interference* happens due to frequency re-utilization in a cellular environment and is unavoidable. Co-channel interference occurs if two

transmission devices operating within the same radio frequency band are active and a receiver, originally trying to receive a signal from one transmitter also receives a (weak) signal from the second transmitter. In cellular systems, co-channel interference is an important factor which limits the systems performance, and is more important than noise. One important fact to note about co-channel interference in a cellular environment is that, increasing the transmission power at each base station (so at each transmitter) provides no system improvement regarding the interference power level [21]. In unlicensed bands, co-channel interference is also an important issue to take into consideration. Here, the same assumption may be made as with a cellular system: if multiple interference sources exist, the interference can be modeled as white Gaussian noise, which makes the consideration of interference much easier. The interference power level can be incorporated in the path loss equations then.

*Adjacent-channel interference* is due to realistic filters letting some power to be transmitted in the sidebands. Adjacent channel interference is encountered in cellular systems as well as in unlicensed bands.

Interference is always the result of other wireless systems operating in the same or operating in nearby frequencies. The quality of demodulation and decoding of a signal depends on the difference between the power of the received signal and the power of other signals with power in the same frequency band – interference – added to the power of the noise.

In general, interference is usually due to a source of noise which primarily does not intend to produce electromagnetic disturbance patterns, for example, microwave ovens or other electrical or electronic equipment. Besides these sources of signal distortion, there are other communications systems which might be active in the environment. Such sources, which have the primary goal to produce electromagnetic radiation for communication purposes are not regarded as noise but are instead referred to simply as interference.

Like noise, interference has an additive distorting impact on the signal. For example, interference occurs in cellular systems, due to the fact that bandwidth is limited and system operators have to reuse certain spectra of the overall frequency bandwidth.

Frequency planning is one method to control interference in cellular systems. However, in unlicensed bands, interference may stem from local wireless networks, which just happen to be deployed quite close to each other.

#### 3.5 WIRELESS CHANNEL PERFORMANCE METRICS

A couple of performance metrics are commonly used in assessing a wireless system performance. These include [20] *symbol error rates,* (SER), and/or *bit error rates,* (BER), or *probability.* Both performance metrics relate to the digital channel: the BER relates to the interpreted bit stream, while the SER relates to the stream of symbols, not being interpreted yet. Both metrics depend on the *instantaneous power ratio* between the received signal power,  $y^2(t)$ , and the noise and interference power,  $n^2(t)$  and  $i^2(t)$ . This instantaneous power ratio is given by the *Signal-to-Noise-and-Interference Ratio* (SNIR). Note that the attenuating influence of the wireless channel is already included in the received signal power,  $y^2(t)$ . If the average SNIR of a given link is available, the average error rates like symbol rate (SER) or bit error rates or error probabilities is not linear; it is instead highly complex and depends on a lot of details. Therefore, an exhaustive understanding of the influence of all effects on the receiver SNIR is necessary when carrying out investigations in the performance of any wireless communications systems.

For any channel where data is transmitted without retransmissions, error rates are critically important. Error rates are typically low for wired connections, but vary enormously for wireless links. Depending on the formatting and content of the data, the relevant measure of performance for a wireless communications system would be:

- End-to-end bit error rate (BER)
- Symbol error rate
- Message error rate, or
- Line error rate (for video).

These are some of the high-level performance measures used as criteria for comparing mobile wireless systems.

Complex digital communications systems are easier to design if "what-if" simulations are used early in the design process. The speed of getting results can be greatly accelerated by focusing the simulation. In designing a simulation, it is important to focus the model on the design task it supports – usually, one component or subsystem of the overall system. A focused, structured simulation accelerates achievement of results (be it improvements or otherwise) without sacrificing accuracy.

From the mathematical model of Fig. 3.2 we have:

y(t) = a(t).s(t) + n(t) + i(t) .....(3.46)

where a(t) is the overall attenuation experienced by the transmitted signal, s(t); n(t) is the random noise and i(t) represents the channel interference signals.

A commonly used measure of the performance of a given channel is the Signal-to-Noise-and Interference Ratio (SNIR), usually measured in dB.

$$SNIR(t) = \frac{a^2(t).s^2(t)}{2.(n^2(t) + i^2(t))}$$
(3.47)

With reference to a digital communication system, we note that a sequence of analog pulses has to be converted back into a digital bit sequence which originated it. This process is generally referred to as *demodulation*. However, demodulation can be separated into two phases: the demodulation itself, and the detection stage.

The digitally modulated signal can be seen as a linear combination of functions which form an orthonormal base of a *vector space* (the inner product space). The function of the demodulator during each symbol period is to decompose the received waveform into a vector in that space. This vector contains the value of the projections of the received waveform as functions in the orthonormal base of the vector space. This can be achieved either by a matched filter or a correlator, depending on the type of signal being demodulated. The detector's function is to compare the vector output from the demodulator with values of the M possible waveforms. The distances in the vector space are compared, and the "closest" possible waveform is accepted as the transmitted sequence. The detection is made possible such that the probability of making a wrong decision is minimized.

# 3.5.1 Bit Error Probability

Error-control coding is one of the most complex functions to simulate in a digital communications system. It is even more of a challenge to simulate error-control coding as part of a system that uses a particular modulation technique over a specified channel. Success lies in the ability to determine how much effect the rest of the system has on the error-control component and, therefore, how best to model the rest of the system.

This is because many variable parameters are involved, and you have to account for the performance of several components: the modulator-channel-demodulator chain although you may not necessarily have to include the physical models for the modulator, channel, and demodulator in your simulations. We are restricting ourselves to the digital signal performance simulation and evaluation with reference to fading and noise phenomena only. For instance, one can remove the modulation and demodulation stages and insert errors directly in the data stream to the decoder. We then derive the statistical characteristics of these errors from the modulation scheme. In effect, we replace the modulator-channel demodulator chain with an equivalent channel.

### 3.5.2 Wireless Channel Design Tradeoffs

Performance optimization usually involves a judicious trade-off to be made between the power, the bandwidth and the complexity of the signal processing required, maintaining the transmission errors of the source data below some given data. The most suitable channel coding and modulation techniques for a given application must, as a rule, take into account the characteristics of the channel through which the transmission occurs.

Consider an ideal band-limited channel of bandwidth W Hertz in which the signal is corrupted only by additive white Gaussian noise (AWGN) having a one-sided power spectral density level of N<sub>o</sub> watt/Hertz. (This is a simple model and does not apply to channels that corrupt signals in more complex ways, e.g., by multipath or mutual interference.) Let  $E_b$  denote the energy per information bit at the receiver and let *C* denote the *channel capacity*, i.e., the maximum average rate at which information can be transferred over this channel. The following formula, derived from Claude Shannon's capacity formula, relates the maximum achievable spectral efficiency C/W to the signal-to-noise ratio (SNR)  $E_b/N_o$ :

$$\frac{E_b}{N_o} = \frac{W}{C} [2^{C/W} - 1]. \qquad (3.48)$$

A graph of spectral efficiency C/W as a function of the signal-to-noise ratio (SNR)  $E_b/N_o$  shows that there is a tradeoff between power efficiency and spectral efficiency. It also reveals that waveforms that achieve very high spectral efficiency (e.g., high-order quadrature-amplitude modulation) require high SNR, while the most power-efficient waveforms (e.g., orthogonal frequency-shift keying), are wasteful of spectrum.

The above equation also reveals that in order to achieve a spectral efficiency C/W of 6 bps/Hertz, the minimum required signal-to-noise ratio  $E_b/N_o = 10.5$  or 10.2 dB. Note that although this value of SNR is a practical operating point for many wireless systems, most existing wireless systems that operate at SNRs in this neighbourhood actually achieve spectral efficiencies less than 1.0, and many military wireless systems and satellite systems have spectral efficiencies less than 0.1. Thus, there is clearly substantial room for improvement.

The simplest way to achieve higher spectral efficiency is by increasing  $E_b/N_o$ , which in turn implies increasing the transmitted signal EIRP in the direction of the receiver (recall that EIRP is the product of transmitted power and antenna gain), increasing the receiving antenna gain, decreasing the receiving system noise figure, or some combination of these measures. The price of simultaneous power and spectral efficiency is a substantial increase in complexity. Nevertheless, combined modulation/coding techniques that achieve fairly high power and spectral efficiency simultaneously have been developed in recent years, and commercial ASICs that implement some of these techniques are now available.

### 3.5.3 Capacity and Spectral Efficiency Issues

The information theoretic definition of capacity of a given channel is the maximum information rate, also equals the maximum error-free user data rate that could be achieved with ideal forward error control (FEC) coding. The term "capacity" is very

often incorrectly used as a substitute for "maximum user data rate". There is an increasing number of users who desire access, as well as a demand for more bandwidth (higher data rates) per user. Higher capacities can be achieved by a combination of:

- More efficient use of the frequency spectrum and
- Greater exploitation of frequencies at X-band (8 12 GHz) and above. Since even at the higher frequencies spectrum is limited, more efficient spectral utilization will be a high priority for all wireless channel users.

A deceptively simple question in mobile communications networking is this: "How does one assess the 'capacity' (maximum throughput) of a given channel?" There are a number if different ways of doing this, and the results can vary significantly, depending on:

- Whether one considers per-user throughput or total channel throughput. Total channel throughput is (at most) the sum of all user transmit data rates at a given time.
- The maximum tolerable bit error rate (BER) or message error rate (MER). For systems in which users or channels generate mutual interference, the maximum number of simultaneous transmissions increases with the maximum error rate that one is willing to accept. Thus, total channel throughput also depends on the maximum error rate.

# 3.6 LIMITATIONS OF CHANNEL MODELS

There are at least four major problems associated with the channel models used in wireless communications system simulations:

- There is a lack of standard reference channel models that can be used for making fair comparisons between competing system concepts. Contractors and other proponents of systems are free to choose the external environment models against which their systems will be evaluated. There is a strong disincentive to choose an external environment model that is more stressing than a proposed system can tolerate.
- Most wireless communications simulations lack adequate representations for multipath fading and distortion, and for jamming other than broadband noise

jamming. These effects can often be much more important for overall system performance than background noise.

• Channel models are often inextricably interwoven with the system model, i.e., elements of the system (modulation, demodulation, and other analog signal processing) are lumped together with the channel to form a single "discrete" channel model, a black box, whose inputs and outputs are symbols (bits or groups of bits). The primary disadvantage of such models is that one cannot separate the system from the channel in order to compare different systems against the same channel.

Even when there is a clear separation between the communications system and the channel in the model, the coding of the interface between the two may not be clean and, in any case, varies from one simulation to the next. This prevents one from easily removing the channel part of a simulation in order to substitute a different channel (or removing the system part in order to substitute a different system). The goal of being able to make fair comparisons of competing systems using existing simulations (for the future) without extensive recoding, will not be realized until all the four above problems have been addressed.

# 3.7 CHAPTER SUMMARY

In this chapter, analytical and empirical models for radio wave propagation were discussed, and the behaviour of the wireless channel as it is observed in reality, was explained. We note that, for many applications, the wireless channel is still the first choice of transmission medium, due to lack of alternatives. The behaviour of the wireless channel may be divided into two parts: *attenuating factors* and *additive factors*. Both contribute to the stochastic behaviour of channel. The attenuation factors, residing in the radio channel, can be analytically decomposed into three components. One of them (path loss) is *deterministic* and depends solely on the distance between the transmitter and the receiver. The other two (shadowing and multipath fading) are *stochastic*, but might be modeled according to their primary and secondary statistics. They depend on the propagation environment as well as on the time scale and sampling rate with which the wireless channel is observed.

The additive factors, residing in the transmission channel, may be modeled by two components, which are both stochastic. The first one, noise, is an omnipresent effect and can not be excluded from any system analysis. It is not a characteristic for wireless channels, but is a limiting factor in any communications system. The second source, interference, depends again on the considered scenario. Since interference is caused by devices transmitting radio frequency signals, it is important to clarify first, if these signals are transmitted within the same frequency bands or not and further, how far such disturbing sources are away from the actual receiver considered. There exist multiple scenarios, where interference has no significant impact.

The analysis in this chapter provides a framework in order to classify different, considered transmission scenarios and determining in advance, which effects might be of interest for a specific scenario and which one is not. Thus, it forms a fundamental block for simulating a certain scenario. We have considered several models, but we restrict our simulation in chapter five to the following models:

- The free-space power model
- The Okumura-Hata model
- A new HF/MF Ground-wave Model for Urban Areas in Uganda
- The Rayleigh fading model, the Rician fading model, and
- The noise and interference models.

However, turning these analytical models into running simulations requires a further step *-- algorithm development and programming*. This is treated in chapter five, which guides the reader to implementable simulation models of wireless channels.

Using propagation models, we can provide installation guidelines, mitigate interference, and design better wireless systems.
# CHAPTER FOUR THE TURBO CODE CONCEPT

### 4.1 INTRODUCTION TO TURBO CODES

All radio communications systems users want the impossible: worldwide, error-free communications. Shannon's theorem gives us some insights into why this is difficult or impossible to achieve but propagation and information theory specialists have been striving to push the physical laws to the limit. The last fifteen years has witnessed a revolution in error control coding, sparked off by the invention of *turbo coding*.

Turbo codes are a class of powerful error correction codes that were introduced in 1993 by a group of researchers in France, along with a practical decoding algorithm [12]. Powerful error correcting codes like turbo codes are indispensable because they offer the reliable data and multimedia service required currently, and in the next generation mobile communication systems. The importance of turbo codes is that they enable reliable communications, with power efficiencies close to 0.5dB of the theoretical capacity limit predicted by Shannon. Since their introduction, turbo codes have been applications, proposed for low-power such as deep-space, and satellite communications, as well as for interference prone applications, such as third generation cellular phone and personal communication services. The major objective here is to achieve maximal information transfer over a limited-bandwidth communication link in the presence of data-corrupting factors, like noise and multipath propagation phenomena. Turbo decoders make use of very powerful soft-input/soft-output iterative decoding algorithms that are however numerically intensive [36]. Turbo codes achieve a performance very close to the Shannon limit, but at the expense of considerable processing complexity and decoding delays. With modern very large scale integration (VLSI) techniques, however, this complexity is tractable. There is a fundamental tradeoff between the bandwidth needed for transmitting signals and information and the amount of information and signal processing required (which also needs battery power) for coding and compression. The major features of turbo codes include:

• Parallel or serial concatenated encoding

- Recursive convolution encoders are used
- Pseudo-random interleaving is employed, and
- Iterative decoding is used.

We note that a single error correction code does not always provide enough error protection with reasonable complexity. Since linear block codes (algebraic codes) are most effective in combating "bursty" errors (errors that arrive in bursts), and Convolution codes are generally more robust when faced with random errors or white noise, we believe that a much powerful code will be obtained using an encoder which links together an algebraic code followed by a convolution code. Serial concatenation was proposed by Forney in 1966 (see fig.4.1).

In 1974, Joseph Odenwalder combined linear block coding and convolution coding techniques to form a *concatenated code*, now referred to as a *turbo code*. In this arrangement, *the encoder linked together an algebraic code followed by a convolution code*. Performance was further enhanced by using an interleaver between



Fig. 4.1 Serial Concatenated Encoding and Decoding

the two encoding stages to mitigate any bursts that might be too long for the algebraic decoder to handle [50].

In 1993 [12], Claude Berrou and his associates perfected the turbo code and is currently the most powerful forward error-correction code.

### 4.2 HOW TURBO CODES WORK

One of the most interesting characteristics of a turbo code is that it is not just a single code [28]. It is, in fact, normally a combination of two codes that work together to achieve a synergy that would not be possible by merely using one code by itself. In particular, a turbo code is formed by the *parallel concatenation* of two *constituent* codes separated by an *interleaver*. Each constituent code may be any type of FEC code used for conventional data communications. Although the two constituent encoders may be different, in practice they are normally identical.



Fig. 4.2 A typical (generic) structure for generating turbo codes.

The overall encoder is said to be systematic because one of the outputs is the same as the input. Fig. 4.2 shows a typical parallel concatenated (generic) structure for generating turbo codes.

We note that the input data stream is applied directly to one of the encoders and the input to the second encoder is applied to its input through the interleaver, leading to

non-identical parity-check bits being produced by the two identical constituent encoders, denoted by Encoder-A and Encoder-B respectively. A relatively stronger code is thus created by encoding in parallel concatenation with a non-uniform interleaver, which is used to scramble the ordering of bits at the input of the second encoder. A pseudo-random interleaving pattern is used. It is very unlikely that both encoders will produce low weight code words. Three copies of the input are generated. First, the parity is computed on the original input data. Secondly, the original data is transformed by the interleaver, and its parity computed. Thirdly, the original data is appended. A word consisting of these three elements (parity of original data, parity of interleaved data, and original data) is then sent.

The input data stream and the parity outputs of the two parallel encoders are then serialized (multiplexed) into a single turbo code word. The data bits are transmitted together with the parity bits generated by the two encoders. Thus, the overall code rate of the encoder is r = 1/3, excluding tail bits. The multiplexer can be used to increase the code rate from  $\frac{1}{3}$  to  $\frac{1}{2}$ . The output of the turbo code encoder is a sequence stream of three components, namely two coded bit streams, and one systematic (uncoded) bit stream. The randomly permutated bit order is transmitted over the channel.

Although each component encoder may employ algebraic coding or convolution coding, the overall encoder can be considered a block encoder because data are processed in blocks. The size of these blocks is dictated by the length of the interleaver that separates each component encoder. In a coded system, performance is characterized by low weight code words. A *good* code produces low weight outputs with very low probability. A *recursive systematic convolution* code (RSC) produces low weight outputs with fairly low probability, although some inputs still cause low weight outputs. However, due to the presence of the interleaver, the probability that both encoders have inputs that cause low weight outputs is very low. Therefore, the parallel concatenation of both encoders will produce a *good* code.

### 4.2.1 The Role of the Interleaver

Interleaving is a practical way of enhancing the error correcting capability of a given code. It is a process of rearranging the ordering of a data sequence in a one to one deterministic format [49]. The size and structure of interleavers greatly influence the performance of turbo codes. There are several types of interleavers which can be implemented. These include random interleavers, block interleavers, diagonal interleavers, and circular-shift interleavers.

The random interleaver uses a fixed random permutation and maps the input sequence according to the permutation order. The block interleaver is the most commonly used interleaver in communication systems. It writes in column wise from top to bottom and left to right and reads out row wise from left to right and top to bottom. A diagonal interleaver writes in column wise from top to bottom and left to right and reads out row has a permutation of the permutation order.

$$P(i) = (a * i + s) \operatorname{mod} L \tag{4.1}$$

such that a < L, a is relatively prime to L, and s < i where *i* is the index, *a* is the step size, and *s* is the offset.

There are two related features of turbo codes [29] that make them different from the more traditional error-correcting codes of the 20<sup>th</sup> century, namely:

- The key insight is the realization that instead of producing a stream of binary digits from the signal it receives, the front-end of the decoder can be designed to produce a likelihood measure for each bit.
- The decoder and encoder are designed so that they take advantage of this extra information.

The encoder sends three sub-blocks of bits. The first sub-block is the *k*-bit block of payload data. The second sub-block is n/2 parity bits for the payload data, computed using the first recursive systematic convolution (RSC) encoder. The third sub-block is n/2 parity bits for a known permutation of the payload data, computed using the RSC encoder - that is, two redundant but different sub-blocks of parity bits for the sent payload. The complete block has n + k bits of data with a code rate of k/(n + k).

For data transmission purposes over a mobile wireless channel BPSK is assumed, along with either an AWGN or Rayleigh flat-fading channel.

### 4.2.2 Turbo Code Generation Processes

The theory of turbo code design and operation is based on the mathematics of recursive sequences. A sequence can be realized as the output of a linear feedback shift register. The terms "shift register sequence" and "recursive sequence" are mutually synonymous, and are used here interchangeably to refer to the same thing.

A linear feedback shift register (LFSR) [51] of length n is a time-dependent device (running on a clock) of n memory cells each capable of holding a value from some field F, such that with each clock cycle the contents of the memory cells are shifted cyclically by one position (to the right, say). While the LFSR discards (or outputs) the rightmost entry  $b_o$  (and replaces it by  $b_1$ ), it computes the linear function

$$c_1 b_{n-1} + \dots + c_n b_o \tag{4.2}$$

of the present state vector  $(b_0, ..., b_{n-1})$  and the feedback coefficients  $(c_1, ..., c_n)$ , (see Fig.(4.3)). Thus, the box with the entry "ADD" represents an adder over F, and the circle with entry  $c_i$  indicates multiplication by  $c_i \in F$  [51]. In practice, the case of the binary field GF (2) is by far the most important one, but the general notion of a LFSR serves as a good intuitive way of modelling recursive sequences.



Fig. 4.3 Linear Feedback Shift Register (LFSR).

Given the initial conditions ( $\alpha_0, ..., \alpha_{n-1}$ ), after *t* clock cycles the LFSR will hold the state vector  $\mathbf{a}^{(t+1)} = (\alpha_t, ..., \alpha_{t+n-1})$ , where [51].

$$a_{t+n-1} = c_1 a_{t+n-2} + \ldots + c_n a_{t-1} .$$
(4.3)

Thus, the shift register sequence  $\mathbf{a} = (\alpha_k)$  produced by the LFSR will satisfy a linear recurrence relation of order  $\mathbf{n}$ ; namely, for  $\mathbf{k} \ge \mathbf{n}$ :

$$a_k = \sum_{i=1}^n c_i a_{k-i} . \tag{4.4}$$

Assuming by convention that  $c_0 = -1$ , one can define the feedback polynomial of the LFSR as

$$f(\mathbf{x}) = -c_0 - \dots - c_n \mathbf{x}^n;$$
 (4.5)

and its reciprocal polynomial

$$f^{*}(\mathbf{x}) = \mathbf{x}^{n} - c_{1}\mathbf{x}^{n-1} - \dots - c_{n-1}\mathbf{x} - c_{n}$$
(4.6)

is called the characteristic polynomial of the LFSR. Using its companion matrix

	0	0			0	$c_n$	
(	1	0			0	$c_{n-1}$	l
4_	0	1	·		÷	:	
л_	÷	0	۰.	۰.	Ξ	:	'
	÷	÷		۰.	0	cz	
Ì	0	0			1	$c_1$	
	(					L	1

the recursion (4.4) can be rewritten in terms of the state vectors as

$$\mathbf{a}^{(t+1)} = \mathbf{a}^{(t)} A \text{ for } t \ge 0.$$
(4.8)

A is usually called the feedback matrix of the LFSR, and it satisfies the equation  $m_A = \chi_A = f^*$ , where  $\chi_A$  and  $m_A$  denote the characteristic and the minimal polynomial of A, respectively.

One may characterize the shift register sequences over F by associating an arbitrary sequence  $\mathbf{a} = (a_k)_{\text{over }} F$  with the *formal power series* 

$$\alpha(x) = \sum_{k=0}^{\infty} \alpha_k x^k \in F[[x]], \qquad (4.9)$$

Then **a** is a shift register sequence if and only if  $\alpha(x)$  belongs to the field F(x) of rational functions over F. More precisely, **a** can be obtained from the LFSR of length n with feedback polynomial  $f \in F[x]$  if and only if

$$\alpha(\mathbf{x}) = \frac{g(\mathbf{x})}{f(\mathbf{x})} \tag{4.10}$$

for a suitable polynomial  $g \in F[x]$  with deg g < n, and this correspondence between shift register sequences **a** belonging to f and polynomials g is a bisection. For instance, the Fibonacci sequence, defined by the recursion  $a_k = a_{k-1} + a_{k-2}$  with initial conditions  $(a_0, a_1) = (1, 1)$  over the rational numbers, belongs to the feedback polynomial  $f(x) = 1 - x - x^2$ , and the polynomial g(x) is simply g(x) = 1. Thus, the formal power series describing **a** is [51]

$$a(x) = \frac{1}{1 - x - x^2} = 1 + x + 2x^2 + 3x^3 + 5x^4 + 8x^5 + \dots + 13x^6 + 21x^7 + 34x^8 + \dots$$
(4.1)

1)

There exists a uniquely determined polynomial m, such that, a given shift register sequence **a** can be obtained from the LFSR with characteristic polynomial  $f^*$  if and only if  $f^*$  is a multiple of m; this polynomial is called the *minimal polynomial* of the shift register sequence **a**. In other words, m is the *characteristic polynomial* of the linear recurrence relation of least order that is satisfied by **a**. If  $\mathbf{a} = (\alpha_k)$  belongs to a LFSR of length n, with characteristic polynomial  $f^*$ , then  $f^*$  is actually the minimal polynomial of **a**, if and only if the first n state vectors  $\mathbf{a}^{(0)}, \dots, \mathbf{a}^{(n-1)}$  are linearly independent.

The global turbo code is built by combining two constituent RSC codes with interleaved versions of the same information sequence u to be transmitted. This is accomplished using a pseudo-random interleaver; the interleaver implements a random permutation of the latter input sequence. In other words, the two constituent encoders are coding the same information sequence u but in a different order. For each input binary information symbol  $u_i$ , we keep the systematic output  $x_i^s = u_i$  of the first RSC encoder, and the parity outputs  $x_i^{1p}$  and  $x_i^{2p}$  of both RSC encoders[52]. The three out symbols are then multiplexed in order to form the following turbo-coded sequence:

$$\{.., u_i, x_i^{1p}, x_i^{2p}, u_{i+1}, x_{i+1}^{1p}, x_{i+1}^{2p}, u_{i+2}, x_{i+2}^{1p}, x_{i+2}^{2p}, ...,\}.$$
 (4.12)

This design results in a code rate of r = 1/3. In practice, it is often necessary to increase the code rate via a puncturing technique which enables to select the coded bits following a particular pattern. The code rate can for instance be increased to  $\frac{1}{2}$  by selecting the sequence:

$$\{\dots, u_i, x_i^{1p}, u_{i+1}, x_{i+1}^{2p}, u_{i+2}, x_{i+2}^{1p}, u_{i+3}, x_{i+3}^{2p}, \dots\}$$
(4.13)

The data bits are transmitted together with the parity bits generated by the two encoders. Thus, the overall code rate of the encoder is r = 1/3, excluding tail bits. Let us assume that the number of input data bits is K, where  $40 \le K \le 5114$ , the first 3K output bits of the overall encoder are in the form:  $X_1, Z_1, Z'_1, X_2, Z_2, Z'_2, \ldots, X_K, Z_K, Z'_K$ , where  $X_k$  is the kth systematic (i.e., data) bit,  $Z_k$  is the parity output from the upper (uninterleaved) encoder, and  $Z'_k$  is the parity output from the lower (interleaved) encoder.

### 4.2.3 Channel Model

Binary phase-shift keying (BPSK) modulation is assumed, along with either an AWGN or a flat-fading channel. The output of the receiver's matched filter is  $Y_k = a_k S_k + n_k$ , where  $S_k = 2X_k - 1$  for the systematic bits,  $S_k = 2Z_k - 1$  for the upper encoder's parity bits,  $S_k = 2Z_k' - 1$  for the lower encoder's bits,  $a_k$  is the channel gain ( $a_k = 1$  for AWGN and is Rayleigh random variable for Rayleigh flat-fading),  $n_k$  is Gaussian noise with variance[52]:

$$\sigma^{2} = \frac{1}{(2E_{s} / N_{o})} = \frac{(3K + 12)}{(2K(E_{b} / N_{o}))}$$
(4.14)

 $E_s$  is the energy per code bit,  $E_b$  is the energy per data bit, and  $N_o$  is the one-sided noise spectral density. Details of the decoding process are contained in reference [52].

### 4.3 TURBO CODE DECODERS

Turbo codes get their name because the decoder uses feedback, like a turbo engine. A turbo code decoder (like the one illustrated in Fig. 4.4 below) operates in an iterative manner. Full iteration consists of two half-iterations, one for each constituent RSC code. The operation of RSC decoders is well articulated in [37].

To decode the (n + k) - bit block of data, the decoder front-end creates a block of likelihood measures, with one likelihood measure for each bit in the data stream. There are two parallel decoders, one for each of the n/2 – bit parity sub-blocks. Both decoders use the sub-block of k likelihood of the payload data. The decoder working on the second parity sub-block knows the permutation the encoder used for this sub-block.

The receive process, requires an analogue input signal, two identical decoders, and an interleaver. It produces confidence value strings. The analogue input signal is transformed into digital confidence values, as well as parity check values. Confidence values are exchanged between the decoders in an iterative data-exchange process. A strong 1 in one detector influences the other; both decoders continue to exchange data until their highest-likelihood solutions converge, which generally occurs in 4 to 10 steps.

*A 'good' linear code is one that has mostly high-weight code words* (except, of course, the mandatory all-zeros code word). High weight code words are desirable because it means that they are more distinct, and thus the decoder will have an easier time distinguishing among them. A turbo code generator is used to reduce the number of low-weight code words [28].



APP = a posteriori probability; DeMUX = Demultiplexor

Fig. 4.4: Turbo decoder.

Since the weight of the turbo code word is simply the sum of the weights of the input and the parity outputs of the two constituent words, we can allow one of these parity outputs to have low weight (as long as the other has high-weight).

Because the second encoder's input has been scrambled by the interleaver, its parity output is usually quite different from the first encoder's. Thus, although it is possible that one of the two encoders will occasionally produce a low-weight output, the probability that both encoders simultaneously produce a low-weight output is extremely small. This improvement is called *interleaver gain* and is one of the main reasons why turbo codes perform so well.

For transmission in a multipath environment, the primary benefit of an interleaver is to provide time diversity (when used along with error-correction coding)[19]. The larger the time span over which the channel symbols are separated, the greater chance there is that contiguous bits (after deinterleaving) will have been subjected to uncorrelated fading manifestations; thus the greater the chance there is to achieve effective diversity. It is important to note [19] that the interleaver provides no benefit against multipath unless there is motion between the transmitter and receiver (or motion of objects within the signal-propagating paths). As the motion increases in velocity, so does the benefit of a given interleaver to the error-performance of the system.

Turbo codes make it possible to increase data rates without increasing the power of a transmission, and they can be used to decrease the amount of power used to transmit at a certain data rate. The main drawback of turbo codes is the high decoding complexity and a relatively high latency, which makes it unsuitable for some applications. For satellite use, this is not of great concern; since the transmission distances it self introduces latency due to the limited speed of light.

### 4.4 THE CODING DILEMMA [28]

Shannon argued that large block-length random codes could achieve channel capacity. We note, however, that random codes are not feasible in practice because the code must have enough structures that permit decoding with hardware or software

of reasonable complexity. Nevertheless, codes with structure do not perform as well as random codes. This leads to the "*Coding Dilemma*": namely that "*all codes are good, except those that we can think of*". Therefore, the best solution is to make the code appear random, while maintaining enough structure to permit reliable decoding. This is why a pseudo-random interleaver must be incorporated in the turbo code generation system. Although turbo codes possess random-like properties, decoding is possible since the interleaving pattern is known.

# 4.4.1 Turbo Code Performance Factors and Tradeoffs

Analysis of simulation results using data from various researchers [31, 32] of turbo code performance reveals the following:

- The turbo code is a powerful error correcting technique under noisy and fading environments. Its performance approaches the Shannon limit within 0.5 dB.
- However, there are several factors which must be considered in the turbo code design: a trade-off between the BER and the number of iterations to be used must be considered first, and secondly, the effect of the frame size on the BER must be taken into account. We note that although the turbo code with a larger frame size has better performance, the output delay in this case is longer.
- Thirdly, the code rate is another factor that must be considered. A higher coding rate demands more bandwidth. For a fixed constraint length, a decrease in code rate improves the performance, whereas for a fixed code rate, an increase in constraint length improves the performance.
- We also note that the behaviour of the turbo code decoder is quite different under different channel environments; for example, the performance of the turbo code is much worse under correlated Rayleigh fading channels than with AWGN or uncorrelated Rayleigh fading channels.
- The major drawbacks of a turbo code design are its complexity and latency (decoding) time. Turbo code decoding is generally computationally intensive; therefore most of the simulated performance results are for high code rates, short constraint lengths, and small frame sizes.

### 4.4.2 Practical Issues

The performance of a turbo code may be more affected by the various parameters of the component codes, block size, interleaver design and weight spectrum [12, 33]. The bit error rate (BER) curve of a turbo code is divided into two regions. The first region is called the "waterfall region", in which the BER decreases rapidly at low signal-to-noise ratios (SNR) and the second region is called the "error-floor region", whereby the BER decreases at a low rate at high SNRs. In the waterfall region the performance depends on the existence of low weight code words. Low weight code words reduce the decoding convergence, thus the BER decreases rapidly and the number of iterations required in the decoding process will also be reduced. The error-floor region occurs due to the presence of a few low weight code words. At low SNR, these code words are insignificant, but as the SNR increases they begin to dominate the performance of the code.

Although turbo codes have the potential to offer unprecedented energy efficiencies, they have some peculiarities that should be taken into consideration [28]. First, while the BER curve falls off sharply with increasing SNR for moderate error rates (e.g., BER>10<sup>-5</sup>), the BER curve begins to flatten at higher SNR. This characteristic is observed for the case where the BER was simulated down to very small values. The region where the BER curve flattens out is called the *error floor* and hinders the ability of a turbo code from achieving extremely small bit-error rates. The error floor is due to the presence of a few low-weight code words. At low SNR, these code words are insignificant, but as SNR increases, they begin to dominate the performance of the code.

The *error flooring effect* can be combated in several ways. One interesting approach is to use two different RSC encoders. One RSC encoder is optimized to perform well at low SNR, while the other is optimized to reduce the error floor. The resulting *asymmetrical turbo encoder* provides a reasonable combination of performance at both a low and high SNR. Unfortunately, although the error floor is reduced, it is still present. Another way to reduce the error floor is to arrange the two constituent encoders in a serial concatenation, rather than in a parallel concatenation. Such a serially concatenated Convolution codes (SCCCs) offer excellent performance at high SNR, as the error floor is virtually eliminated. However, performance at low SNR is considerably worse than it is for parallel concatenated codes (also called parallel

concatenated Convolution codes (PCCCs). An alternative to choosing between SCCCs and PCCCs is to use *hybrid turbo codes*, which combine features of each type of code.

# 4.5 PERFORMANCE OF TURBO CODES IN FADING CHANNELS

Turbo codes are reported [33] to be very powerful in additive white Gaussian noise (AWGN) channels. Turbo codes have also been shown to perform very well in rapidly fading channels [34], but to perform less well in slow fading channels. In rapidly fading channels, coding together with interleaving techniques are used to spread consecutive code bits over multiple independently fading blocks to improve performance. However, in slow fading channels coding together with interleaving techniques cannot in general be used in an effective manner because delay and latency considerations limit the depth of interleaving. This situation compromises in particular the performance of turbo codes because occasional deep fades cause severe error propagation in the iterative decoding process [35]. Multimedia systems require varying quality of service, and therefore we need to consider various performance factors. For instance, for voice communication and teleconferencing low latency is desirable, whereas for data transmission, low bit/frame error rates (BER or FER) are desirable. Fortunately, the tradeoffs inherent in turbo codes match with the tradeoffs required by multimedia systems. Although for data transfer large frame sizes are used, with low bit error rates, the associated long latency has to be tolerated with. For voice communication small frame sizes are used, leading to short latency, however, the associated bit error rates are much higher.

In fading channels errors associated with the demodulator tend to occur in bursts, corresponding to the times when the channel is in deep fade [18]. Most codes designed for AWGN channels cannot correct for the long bursts of errors exhibited in fading channels. Therefore, it is not surprising that codes designed for AWGN channels only, can exhibit worse performance in fading channels than an uncoded system. To improve performance of coding in fading channels, coding is typically combined with interleaving to mitigate the effects of error bursts [18]. The basic premise of coding and interleaving is to spread error bursts due to deep fades over

many code words such that each received code word only exhibits at most a few simultaneous symbol errors, which can be corrected for. The spreading out of burst errors is accomplished by an interleaver and the error correction is accomplished by the code. The size of the interleaver must be large enough so that fading is independent across a received code word. Slowly fading channels require large interleavers, which in turn can lead to large delays.

Coding and interleaving [18] is a form of diversity, and performance of coding and interleaving is often characterized by the diversity order associated with the resulting probability of error. This diversity order is typically a function of the minimum Hamming distance of the code. Thus, designs for coding and interleaving on fading channels must focus on maximizing the diversity order of the code, rather than on metrics like Euclidean distance which are used as performance criterion in AWGN channels.

### 4.6 TURBO CODE DESIGN COMPLEXITY

The turbo code is a very complex channel coding scheme. The turbo encoder is a parallel concatenation of two recursive systematic convolution (RSC) codes. The turbo code decoder is an iterative serial concatenation of two soft output Viterbi algorithm (SOVA) decoders. In addition, the presence of interleavers in both the encoder and the decoder further complicates this coding scheme [40].

The two common methods of evaluating the performance of turbo codes are using theoretical analysis and using computer simulation. Theoretical analysis of turbo codes, however, is very difficult due to the structure of the coding scheme. A few researchers have attempted to analyze turbo codes using the theoretical approach; however, their results do not match closely to computer simulation results. The theoretical analyses presented in their publications are not convincingly presented and are thus difficult to follow.

# 4.7 TURBO CODE APPLICATIONS IN EMERGING WIRELESS TECHNOLOGIES

### 4.7.1 Cellular Mobile Radio Communications

Wireless mobile communications are often faced with the problem of unpredictable and time-varying fading phenomena and therefore demand more robust systems to provide the required immunity. For applications where delay versus performance is critical, turbo codes offer a wide trade-off space at decoder complexities equal to or better than those of conventional convolution or block code performance. The major benefit is that turbo codes can work with smaller constraint-length encoders. The major drawback of turbo encoders/decoder systems is the decoder latency. Turbo codes with short delay are being heavily researched. Turbo codes generally outperform convolution and block codes when interleavers exceed 200 bits in length [46]. Since 1999, in the area of third-generation mobile networks, UMTS in Europe and CDMA2000 in the USA and in Asia have been using Turbo Codes for broadband data services that require a guaranteed transmission quality with a bandwidth of more than 64 kbps.

CDMA2000 is an important third generation cellular standard, which was formulated by the third generation partnership project (3GPP). As in Universal Mobile Telephone Service (UMTS) system, CDMA2000 systems use turbo codes for forward error correction (FEC). While the turbo codes used by these two systems are very similar, the differences lie in the interleaving algorithms, the range of allowable input word length and the rate of constituent RSC encoders [46].

# 4.7.2 Digital Video Broadcasting

In 2000, in the field of digital video broadcasting (DVB), Turbo Codes were chosen for return channels that allow interactive services either by satellite (DVB-RCS: Return Channel Satellite) or hertz (DVBRTC: Return Channel Terrestrial). Similarly, broadband wireless local-loop systems in Europe (ETSI Hiperaccess) and in America (IEEE 802.16.1) also chose this technology as optional coding to increase transmission rates. Currently turbo codes are used as part of the Digital Video Broadcasting (DVB) standards.

# 4.7.3 Long-Haul Terrestrial Microwave Links

Microwave towers are spread across the countryside, usually on hilltops and communications between them is subjected to weather induced fading, shadowing and vegetation profiles. Since these links utilize high data rates, turbo codes with large interleavers are used to combat the fading, while adding only insignificant delays. Furthermore, substantial power savings can be realized for towers on towers in remote areas especially, when turbo codes are used [46].

# 4.7.4 Military Applications

Since turbo codes are applicable in spread-spectrum systems, this provides increased opportunity for anti-jam and low probability of intercept (LPI) communications. In particular, very steep BER versus  $E_b/N_o$  curves lead to a sharp demarcation between geographic locations that can receive communication with just sufficient  $E_b/N_o$  and those where  $E_b/N_o$  is insufficient. Turbo codes are used in a number of military and defence application.

### 4.7.5 Image Processing

Embedded image codes are very sensitive to channel noise because a single bit error can lead to irreversible loss of synchronization between the encoder and the decoder. Turbo codes are used for forward error correction in robust image transmission. They are suited to protection of visual signals, since these signals are typically represented by a large amount of data even after compression.

# 4.7.6 Wireless Local Area Networks (WLANs)

Turbo codes facilitate a better performance of a WLAN compared to the case when traditional Convolution codes are used. Using turbo codes, however, an 802.11a system must be configured for high performance. The resulting benefits to the WLAN system are that it requires less power, and it can transmit over a greater coverage area. The turbo code solution is used to reduce power and boost performance in the transmit portion of mobile devices in a wireless local area network (WLAN) [46].

### 4.8 CHAPTER SUMMARY

In this chapter we have shown that turbo codes are a natural forward error-correction scheme for third-generation high-speed wireless data services and their application in the next generation equipment is assured because of their good performance. A turbo code encoder produces *low weight outputs with very low probability*. It consists of two RSC encoders and an interleaver between the systematic input and the input of the second RSC encoder. Because of the interleaver, the probability of both encoders having inputs that would cause low weight outputs is very low. Therefore the parallel concatenation of both encoders produces a robust code. Simulation results confirm that turbo codes have good performance and flexibility and are therefore suitable for wireless local area networks (WLANs) applications and for next generation mobile phones. The coding gain in the waterfall region ranges from 0.5 to 0.7 dB at a BER of  $10^{-4}$  depending on the channel conditions. So it is important to define the BER very well, in order to make a proper code selection. Other concerns, such as latency and availability of low cost, low power components, may be of greater concern than the small performances differences at a particular  $E_b/N_o$  operating point. Turbo codes are mainly attractive for high-data-rate services due to the relatively long interleaver. For extremely short interleavers, Convolution codes outperform turbo codes.

Turbo codes have extraordinary performance at low SNR, namely, performance is very close to the Shannon limit. This is due to a low multiplicity of low weight code words [39]. However, turbo codes have a BER "floor". This is due to their low minimum distance. Performance of turbo codes improves for larger block sizes. However, larger block sizes mean more latency (delay). Nevertheless larger block sizes are not more complex to decode. We also note that the BER floor is lower for larger frame/interleaver sizes. Finally, turbo codes have been compared for short frame sizes, which is another possible application for turbo codes in third-generation wireless systems.

# CHAPTER FIVE INVESTIGATION OF RADIO PROPAGATION MODELS USING MATLAB-BASED GUIS

### **5.1 INTRODUCTION**

The propagation of radio waves through a wireless environment is influenced by various mechanisms which affect the fidelity of the received signal. Accurate prediction of these effects is essential in the design and development of a wireless communications system. Accurate prediction of these propagation effects allows communications engineers to address the trade-off between radiated power and signal processing by developing an optimum system configuration in terms of modulation schemes, coding, frequency band and bandwidth, antenna design, and transmission power.

The mobile radio channel places fundamental limitations on the performance of wireless communication systems. The transmission path between the transmitter and receiver can vary from simple line of sight to one that is severely obstructed by buildings, mountains and foliage. Unlike wired channels which are stationary and predictable, radio channels are extremely random and do not offer easy analysis. Even the speed of motion impacts how rapidly the signal fades as a mobile terminal moves in space. Modelling the channel has historically been one of the most difficult parts of mobile radio system design, and is typically done in a statistical fashion, based on measurements made specifically for an intended communication system or spectrum allocation.

To be able to design good and reliable wireless communication systems, a wireless engineer must be well versed with what happens to the signal as it travels from the transmitter to the receiver. Simulation of radio-frequency (RF) signals with appropriate statistical properties can readily facilitate this process. Statistical testing can subsequently be used to establish the validity of the fading models used. This chapter presents the pathloss models and fading models. It then concentrates on the approaches used to simulate these models using MATLAB.

One of the objectives of this research was to develop a software tool to determine signal strength and pathloss at any distance between the transmitter and receiver in any given environment and to simulate the variation of the signal over short distances and short time durations (fading). This software has been designed and implemented successfully.

### 5.2 LARGE SCALE MODELS

In the process of designing GUIs for the study of large scale models, analytic equations specific to each model were used to carry out the computations for the plots. These equations are available in literature and have already been quoted in this thesis. These models give variations of the signal strength and pathloss as a function of distance. For each model, a clear understanding of the equations and accompanying facts and assumptions is obtained first and algorithms and MATLAB code development follow, to complete the design.

Examples of large scale models considered here include:

- The free space/path propagation model,
- The Okumura-Hata model, and
- The HF and MF Ground wave models for Urban areas.

Two frequencies are used for comparison purposes. 900 MHz and 1800 MHz are chosen as typical values because these frequency bands are used for cellular networks in Uganda. However, these values can be changed depending on the user's requirements. The antennas at both ends are taken to have unity gain. The gains can be changed as well. Also included is a calculation section which gives the user answers for received power and pathloss at any distance without reading from the GUI. Note that the distance entered in the calculation section does not necessarily have to be within the range entered for the distance of coverage. For example, at a distance of 4 km, the values are shown at 900 MHz. The system loss L is usually

kept as 1 which indicates that there is no loss in the system. In practice, this value is greater than 1.

### 5.2.1 The Free Space Model GUI

The attenuation of a signal propagating in free space over a distance of d meters between two antennas is according to section 3.2.5 expressed as:

$$\frac{P_{o}}{P_{t}} [dB] = 10 \log \frac{P_{o}}{P_{t}} = 20 \log \left(\frac{\lambda}{4\pi d}\right) + 10 \log (g_{Tx}) + 10 \log (g_{Rx}) \dots (5.1)$$

where  $P_o$  is the received power,  $P_t$  is the transmitted power,  $\lambda$  is the wavelength of the signal,  $g_{Tx}$  is the gain of the transmitter antenna and  $g_{Rx}$  is the gain of the receiver antenna (both gains being in the direction of the straight line that connects the two antennas in space. The received power is inversely proportional to the square of the distance *d* and the square of the signal frequency.



Figure 5.1: A GUI for a free-space model

Figure 5.1 shows a GUI that has been designed for the free space model. The freespace power loss model MATLAB code is found in Appendix C of the thesis. Note that the input for distance does not start at zero. This is because d must be in the far field. Given antenna size and wavelength, the far field distance can be obtained from

the condition  $d \ge \frac{2D^2}{\lambda}$ , where D is the largest linear dimension of the antenna.

The **Simulate** button enables the user to obtain plots for the received signal strength and path loss where as the **Close** button closes the window. When pressed, it first prompts the user to conform whether or not he wants to close the window. The **Reset** button takes the user back to the default values in the different edit boxes.

Two frequencies are considered for comparison purposes. 900 MHz and 1800 MHz are chosen as base values because these are the two frequency bands used for cellular networking in Uganda. However, these values can be changed as appropriate to the user's requirements. The antennas at both ends are taken to have unity gain. The gains can be changed as well. Note that the distance entered in the calculation section does not necessarily have to be within the range entered for the distance of coverage. For example, at a distance of 4 km, the values are shown at 900 MHz. These values must of course be similar to those obtained from the graph. The system loss L is usually kept as 1 which indicates that there is no loss in the system. In practice, this value is greater than 1.

#### 5.2.2 The Okumura-Hata Model GUI

Because the Hata model is an empirical formulation of the graphical pathloss data provided by Okumura, the two models were combined in a single GUI represented by the design equation

$$L_{urban}(dB) = 69.55 + 26.16 \log f_c - 13.82 \log h_{te} - a(h_{re}) + (44.9 - 6.55 \log h_{te}) \log d$$
....(5.2)

where  $f_c$  is the frequency which varies from 150 MHz to 1500 MHz,  $h_{te}$  and  $h_{re}$  are the effective heights of the base station and the mobile antennas (in meters) respectively, *d* is the distance from the base station to the mobile antenna,  $a(h_{re})$  is the correction factor for the effective antenna height of the mobile unit, which is a function of the size of the area of coverage [32].

The GUI simulates power received as well as path loss for propagation in different areas. The areas or environments considered by the GUI were large city, mediumsized city, small city as well as sub-urban area and open countryside (rural area). A calculation section is provided to give accurate values of pathloss and received power at any distance (even outside the scale of the graphs). Two transmitting antenna heights are input for comparison purposes. It is evident that the higher the antenna, the greater the signal strength at the receiver, and consequently the pathloss suffered.



Figure 5.3: A GUI for the Okumura-Hata model

# 5.2.3 GUI for HF and MF Ground Wave Model for Urban Areas

This model looks at ground wave propagation in urban areas at HF and MF assuming various weather conditions. Kampala City was used as a case study. The key novelty in the new model is the inclusion of three building-complex parameters, i.e. building density parameter  $r_b$ , sight parameter  $r_s$  and environmental parameter  $r_e$ , and a new height-gain factor  $G_h$ . This is because buildings are seen as the primary barriers which affect propagation of ground-waves at MF and HF in urban areas. The effect of buildings can be broken down as follows:

- The number of buildings in an area (building density)
- The height of buildings with respect to the transmitting antenna height (sight parameter).
- The environment around the receiver as far as buildings are concerned (environmental factor).



Figure 5.4: A GUI for the new HF/MF Ground-wave model

The above caption shows the GUI. The signal characteristics at both MF and HF are plotted on the same graphs for comparison purposes. As expected, there is more pathloss and hence less received power strength at HF than at MF as expected.

# 5.3 SMALL SCALE MODELS

Small scale propagation models characterize the rapid fluctuations of the received signal strength over very short travel distances (a few wavelengths) or short time durations (on the order of seconds). In other words, these models quantify the variability of the signal strength in close proximity to a particular location.

Small-scale models also describe the rapid fluctuation in the amplitude of a radio signal over a short period of time or travel distances and therefore large scale effects may be ignored. Fading is caused by the interference between two or more versions of the transmitted signal which arrive at the receiver at slightly different times. These waves, called multipath waves, combine at the receiver antenna to give a resultant signal which can vary widely in amplitude and phase depending on the intensity and relative propagation time of the waves and the bandwidth of the transmitted signal.

Radio frequency signals with appropriate statistical properties can readily be simulated. Examples of small scale models simulated using MATLAB include:

- The Rayleigh Fading Model
- The Rician Fading Model
- The Nakagami Distribution, and
- The Log-Normal fading distribution model.

# 5.3.1 The Rayleigh Fading Model

The multipath faded signal is simulated using MATLAB to understand the relationship between the number of paths (N) and the statistics of the received signal. The GUI developed allows the user to vary the carrier frequency so as to compare simulations at different frequencies of propagation. N can also be varied at the user's discretion and a large value gives you a better Gaussian distribution approximation.

For a given time instant, the received signal in the case of a stationary receiver is generated using equation (5.3).

$$R(t) = \sum_{i=1}^{N} a_i \cos(\varpi_c t + \varphi_i)$$
(5.3)

If the Doppler-effect induced by motion is considered we use equation (5.4).

$$R(t) = \sum_{i=1}^{N} a_i \cos(\varpi_c t + \varpi_{di} + \varphi_i) \qquad (5.4)$$

to describe the received signal, where  $\varpi_{di} = \frac{\varpi_c V}{c} \cos \psi_i$  is the angular Doppler shift

of the signal.

The path amplitudes are taken to be Weibull distributed and generated using the function *weibrnd()* from the MATLAB Statistics Toolbox. The 2-parameter Weibull distribution allows the flexibility of making it easy to see the effects of varying scattering amplitudes. For example, a function *weibrnd(0.5,0.5,[1 (N+1)])* produces N random values that can be taken to represent the random amplitudes.

The phases are also taken to be uniform in the range  $[0,2\pi]$  and are generated using the function *unifrnd()*, *also* from the Statistics Toolbox. For example, a function *unifrnd(0,1:(N+1))* generates N random values that can be used to represent the random phases. Subsequently, the envelope is calculated using equation (5.5), which represents the received signal envelope:

$$R = \sqrt{[I(t)]^2 + [Q(t)]^2}$$
(5.5)

When N is large, the in-phase and quadrature components will be Gaussian.

The probability density function (pdf) of the received signal envelope can be shown to be Rayleigh distributed:

$$f(r) = \frac{r}{\sigma^2} \exp\left[-\frac{r^2}{2\sigma^2}\right], \ r \ge 0$$
(5.6)

Note that  $\sigma$  is the *rms* value of the received voltage signal before envelope detection, and  $\sigma^2$  is the time-average power of the received signal before envelope detection. The probability that the envelope of the received signal does not exceed a specified value R is given by the corresponding cumulative distribution function (CDF) below:

$$F(R) = \Pr(r \le R) = \int_{0}^{R} f(r) dr = 1 - \exp\left(-\frac{R^{2}}{2\sigma^{2}}\right)....(5.7)$$

The mean value  $r_{mean}$  of the Rayleigh distribution is given by:

$$r_{mean} = E[r] = \int_{0}^{\infty} rf(r)dr = \sigma \sqrt{\frac{\pi}{2}} = 1.2533\sigma$$
 ....(5.8)

Two GUIs for Rayleigh fading are depicted in Fig. (5.5) and (5.6). The first simulates the fading phenomena and the Rayleigh envelope as well as the probability and cumulative distribution functions (as given by equations (5.6) and (5.7)). The second GUI is more comprehensive. In addition to these, it also simulates the outage probability as well as showing the in-phase and quadrature components separately.



Fig. 5.5 A GUI to simulate Rayleigh fading

#### **Outage Probability**

In a fading radio channel, it is likely that a transmitted signal will suffer deep fades that can lead a complete loss of the signal or outage of the signal. The outage probability is a measure of the quality of the transmission in a mobile radio channel. Outage is said to occur when the received signal power goes below a certain threshold level (a constant value shown by a straight line on the simulations). This probability can be calculated as the integral of the received signal power R(t) as

$$P_{out} = \int_{0}^{P_{out}} R(t) dt, \qquad (5.9)$$

 $P_{th}$  is the threshold power.

The concept of outage can be demonstrated with MATLAB using the results from the previous sections. The procedure to find the outage probability is as follows:

- 1. Calculate the received signal power as given in specific equations previously.
- 2. Set a threshold power level for the received signal relative to the average signal power.
- 3. Count the number of times in the sample interval that the received signal power goes below this threshold.
- 4. Using the basic concept of probability, the outage is then calculated by taking the ratio of the count in step 3 to the total number of samples.

### 5.3.2 The Rician Fading Distribution GUI

The main difference between Rayleigh and Rician fading is that where as Rayleigh fading considers a number of multipath components only, giving the total received signal, in the Rician model, there exists a direct path between the transmitter and receiver. This direct path or LOS gives rise to a dominant signal component which, in addition to the multipath components, yields the total received signal. GUIs designed for the Rician fading model are depicted in Figures (5.7) and (5.8). Fig. (5.7) shows simulations for the signal fading and received envelope while figure (5.8) shows the Rician PDF and CDF for the specified parameter inputs.



Figure 5.6: A more comprehensive simulation of Raleigh fading



Figure 5.7: A GUI for simulating Rician Fading



Fig. 5.8: A GUI for simulating the Rician PDF and CDF

### 5.3.3 The Nakagami Fading Model GUI

Parameter  $\Omega$  controls the distribution spread and is given by  $\Omega = E\{r^2\}$ . The corresponding Nakagami-*m* cumulative distribution function is given by:

P(,) is the incomplete gamma function. Note that when m=1, the Nakagami model reduces to the Rayleigh model. This is a special case of the model. When m>1, the fluctuations in signal strength reduce and the models gradually tends to Rician as m increases.

The GUI shown in Figure (5.9) is a simple simulation that produces plots of the Nakagami probability and cumulative distribution functions when the envelope size is specified. The PDF is similar in shape to that of the Rayleigh and Rician distributions. It is possible to describe both Rayleigh and Rician fading with the help of a single model using the Nakagami distribution. The fading model for the received signal envelope, proposed by Nakagami, has the PDF given by:

$$f(r) = \frac{2m^m r^{2m-1}}{\Gamma(m)\Omega^m} \exp\left[-\frac{mr^2}{\Omega}\right], \ r \ge 0$$
(5.11)

 $\Gamma$ () is a gamma function and m is the shape factor (constrained to  $m \ge 0.5$ ) given by

$$m = \frac{E^2 \{r^2\}}{E \{r^2 - E^2 \{r^2\}\}^2}$$
(5.12)

### 5.4 LOG-NORMAL DISTRIBUTION

This distribution is obtained from the large-scale model that predicts the local signal strength in an area by the use of a reflection exponent n as discussed earlier in section 3.2.7. The fading over large distances causes random fluctuations in the mean signal power. Evidence suggests that these fluctuations are log-normally distributed. A heuristic explanation for encountering this distribution is as follows:

The transmitted signal undergoes multiple reflections at the various objects in its path, before reaching the receiver. Then it splits up into a number of paths, which finally

combine at the receiver. The expression for the transmitted signal is the same as that given in equation (5.5), except that the path amplitudes are themselves the products of the amplitudes due to the multiple reflections. They are given by:

$$a_i = \prod_{j=1}^N a_{ji}$$
 (5.13)

N is the number of multiple reflections per path. Multiplication of the signal amplitude leads to a log-normal. The fact that the mean of the envelope is lognormal is well established in literature which gives the lognormal PDF as:

$$f(r) = \frac{1}{r\sigma\sqrt{2\pi}} \exp\left[-\frac{(\ln r - \mu)^2}{2\sigma^2}\right], \ r \ge 0$$
 (5.14)



Fig. 5.9: A GUI for simulating Nakagami model distribution.



Fig. 5.10: A GUI for Log-Normal fading distribution

For this distribution, log(r) has a normal distribution with  $\mu$  as its mean and  $\sigma^2$  as its variance.

# 5.5 CHAPTER SUMMARY

The mobile radio channel places fundamental limitations on the performance of wireless communication systems. The transmission path between the transmitter and receiver can vary from simple line of sight to one that is severely obstructed by buildings, mountains and foliage. Unlike wired channels which are stationary and predictable, radio channels are extremely random and do not offer easy analysis.

To be able to design good and reliable wireless communication systems, a wireless engineer must be well versed with what happens to the signal as it travels from the transmitter to the receiver. Simulation of radio-frequency (RF) signals with appropriate statistical properties can readily facilitate this process. Statistical testing can subsequently be used to establish the validity of the fading models used. This chapter presents the path loss models and fading models. It then concentrates on the approaches used to simulate these models using MATLAB.

Graphic user interfaces (GUIs) are a simple way of illustrating the effect of varying the values of the various parameters, which influence the behaviour of the information signal, especially due to fading and noise. MATLAB code for the following graphic user interfaces (GUIs) was developed:

- Free-space model
- Okumura-Hata model
- HF and MF ground-wave model for urban areas
- Rayleigh fading model
- Rician fading model
- Nakagami fading model, and the
- Log-normal fading distribution model.

# CHAPTER SIX TURBO CODE SIMULATION EXPERIMENTS AND ANALYSIS OF RESULTS

### **6.1 INTRODUCTION**

In this chapter we investigate the performance of turbo codes in the wireless environment. We especially study the effects of the code block length (block size), interleaver size, the number of decoding iterations, and the code rate on bit error probability (BER).

### 6.2 SIMULATION SETUP FOR TURBO CODES

The simulation setup is composed of three distinct parts, namely the encoder (as discussed in sections 4.1 and 4.2), the channel (whose details are found in chapters three and four) and the decoder (discussed in section 4.3). The simulation of the turbo code encoder is based on its description in section 4.2. The simulated turbo code encoder is composed of two identical RSC component encoders. These two component encoders are separated by a random interleaver. The random interleaver is a random permutation of bit order in a bit stream. This permutation of bit order is stored in memory so that the interleaved bit stream can be deinterleaved at the decoder. The output of the turbo code encoder is described by three streams, one systematic (uncoded) bit stream and two coded bit streams, as illustrated in Fig. 4.2. The systematic bit stream can only have one set of m tail bits from one of the two recursive encoders.

In its basic form, the turbo code encoder [40] is a rate 1/3. However, in many turbo code publications a rate of  $\frac{1}{2}$  is indicated. This is accomplished in practice by puncturing the coded bit streams of the turbo code. The puncturing pattern is that for every one coded bit stream, the odd bits are punctured out, and for the other coded bit stream, the even bits are punctured out. The BER performance of the turbo codes are studied by simulation using an AWGN channel and a Rayleigh fading channel. The

discrete model of the AWGN channel is given by the expression  $y_k = x_k + z_k$ , where  $x_k$  is the transmitted symbol and  $z_k$  is a Gaussian random vector representing channel noise with independent and identically distributed (i.i.d.) components with mean zero and variance  $N_o/2$ . Similarly, the discrete model of the Rayleigh fading channel is given by the expression  $y_k = a_k x_k + z_k$ , where  $x_k$  and  $z_k$  are the same as above, and the  $a_k$ 's are i.i.d. random variables with a Rayleigh distribution of the form  $f(a) = 2ae^{-a^2}$  for  $a \ge 0$ .

For the Rayleigh fading channel, the decoder incorporates the  $a_k$ 's into the decoding algorithm. In other words, it is assumed that the receiver can determine the multiplicative fading factors. The discrete models for the AWGN and Rayleigh channels are illustrated in Fig. 6.1.



Fig. 6.1 AWGN and Rayleigh Fading Channel Models

### 6.2.1 Turbo Code Simulation Software

The TURBO8.EXE program developed by the Communications Research Centre, Canada, was used to generate the simulation results for various turbo code generation configurations. But any other suitable simulation tool, for example, MATLAB Simulink could be used. The TURBO8.EXE is a stand-alone executable program for simulating the performance of binary rate 1/3 (or higher with puncturing) 8-state turbo codes. The program is invoked by calling it at a command line with the name of
a file (referred to as a parameter file) that specifies the parameters of the desired simulation. The turbo decoder uses iterative soft-in/soft-out a posteriori probability (APP) decoding principles based on low-complexity enhanced max-log-APP processing with scaled extrinsic information, as described in [54]. The turbo decoder is iterative. A half iteration is defined as the APP processing of one of the two RSC codes. The turbo decoder has an early stopping feature to reduce the average number of iterations performed. Simulation results are appended in Appendix A, and MATLAB code has been designed (see Appendix B) for plotting the results graphically.

### 6.2.2 Turbo Code Performance as a Function of Interleaver Size and Frame Size

The choice of the interleaver size is dependent on the expected maximum size of the input frame. When the input frame size is increased, the interleaver is made large, and this adds to the complexity of the design. In turn this is accompanied by increased decoding latency and power consumption. The probability of bit error  $(P_b)$  for a turbo code is inversely related to the interleaver length as given by the following equation:

Therefore, for everything else equal, a turbo code with a long interleaver length will outperform a turbo code with a short interleaver length in terms of BER. However, this performance improvement comes at a cost since it was found that processing time increases with interleaver length.

Interleaving is the process of rearranging the ordering of an incoming bit stream in some deterministic order specified by the interleaver module. The inverse of this process is called deinterleaving and restores the received sequence to its original order. Interleaving is a practical way of enhancing the error-correcting capability of channel coding. In turbo coding the interleaving process is implemented before the data to be transmitted is encoded by the second RSC code encoder. The unique role of the interleaver is to construct a long block code from short Convolution codes, which enables the resulting long codes to approach the Shannon capacity limit. The interleaver also spreads out the burst errors, as mentioned in section 4.5. Interleaving

provides scrambled information data to the second RSC code encoder and decorrelates inputs to the two component decoders so that an iterative suboptimum-decoding algorithm based on uncorrelated information exchange between the two decoders can be applied [49].

Since the turbo code is a block code, it causes a time delay before getting the complete decoding output, and therefore increasing the frame size also increases the delay time. From Fig. 6.2 we are able to establish that a turbo code with a larger frame size has better BER performance compared to short ones. However, since the turbo code is a block code, it causes a time delay before getting the complete decoding output, and therefore increasing the frame size also increases the delay time.



Fig. 6.2 Impact on performance for different information block lengths (frame-size) used.

### 6.2.3 Turbo Code Performance as a Function of the Number of Iterations

We choose the turbo code (64,54,4000) as the basic reference parameter change, whereby the block length is kept constant at 4000, and the other basic conditions are kept the same while the number of decoding iterations is varied. We perform numerous simulations with varying numbers of MAP decoding iterations. We then draw graphs (Fig.6.3) to depict the relationships for the bit error rate under different values of the signal-to-noise ratio.



# Fig.6.3 Effect of varying the number of iterations, assuming a constant information block length of 4000.

From the graph of Fig.6.3 we establish that increasing the number of iterations leads to a performance improvement, however, this is at the expense of the added complexity. Definitely, increasing the number of iterations also leads to increased latency in the process of obtaining the decision output after completion of the decoding. This has also some impact on the power consumption considering the design at the implementation level.

#### 6.2.4 Interleaver Design Considerations

The performance of turbo codes is dependent on various factors like frame-size, number of iterations, selection of various types of encoders and use of different interleavers.



Fig. 6.4 Effect of different RSC encoder configurations on performance of the resulting turbo code.

The effect of various RSC encoder configurations on the performance of the resulting turbo codes is demonstrated by plotting graphs of bit error probability (BER) against the signal-to-noise ratio ( $E_b/N_o$ ) as depicted in Fig.6.4 for various encoder types and use of various interleavers. At low signal-to-noise ratios ( $E_b/N_o$ ) the BER of turbo codes using random interleavers is lower than that obtained using structured interleavers. However, at high SNR it is in general the other way round by proper design.

Serially concatenated systems tend to have larger free Euclidean distances than parallel concatenated systems, and hence lower error floors. Parallel concatenated systems tend to do better in the waterfall (turbo cliff) region. Large interleavers are required to achieve precise turbo cliff behavior.

The excellent BER performance of turbo codes is enhanced when the length of the interleaver is significantly increased. The interleaving block and its corresponding deinterleaver in the decoder, does not much increase the complexity of the turbo scheme, but it does introduce a significant delay in the system, which in some cases can be a strong drawback, depending on the application.





The puncturing technique is used to improve the rate of a given turbo code. The puncturing selection process is performed by periodically eliminating one or more of the outputs generated by the constituent RSC encoders. Fig.6.5 illustrates the performance of turbo codes with unpunctured and punctured turbo code encoders of similar block lengths. The performance of the unpunctured configuration is better

than that of the punctured configuration, but the time delay for the decoding process is greater in the unpunctured case compared to the punctured situation.

Fig. 6.6 reveals that for large code rates greater signal-to-noise ratios are required for both unpunctured and punctured scenarios whereas for the same block size the required signal-to-noise ratios using lower code rates are comparatively much lower. At low signal-to-noise ratios  $(E_b/N_o)$  the BER of turbo codes using random interleavers is lower than that obtained using structured interleavers. However at high SNR, it is in general the other way round by proper design. Through a combination of these two kinds of interleavers, this thesis proposes a new type of pseudo-random structured interleaver.



Fig.6.6 Performance dependence on code rate, information block size, and puncturing.

#### 6.3 CHAPTER SUMMARY

The performance of turbo coding systems is characterized by two distinct regions, namely, a *turbo cliff* (or waterfall) region and an *error floor/flare* region. Within the turbo cliff the bit error rate drops within a fraction of a dB of signal-to-noise ratio (SNR) to very low values, and the error floor/flare region is characterized by a slow decrease of the error rate with increasing SNR. The turbo cliff is dependent on the statistical behaviour of the individual component decoders, whereas the error floor/flare is determined by the free Euclidean distance asymptote [53]. Serially concatenated systems tend to have larger free Euclidean distances than parallel concatenated systems, and hence lower error floors. Parallel concatenated systems tend to do better in the waterfall (turbo cliff) region. Large interleavers are required to achieve precise turbo cliff behavior. Both serially concatenated and parallel concatenated codes can virtually achieve the Shannon bound, however, as in the limit performance is pushed to the Shannon limit, the ensuing complexity is overwhelming due to the number of iterations required.

To obtain significant improvements in performance at high  $E_b/N_o$ , one needs to increase the length of the code, and hence, increasing the complexity of the system. Another disadvantage of using longer codes is the necessity to have large interleavers, resulting not only in increased implementation complexity but also in large delays. We note, however, that concatenation of two or more codes allows a significant reduction in complexity over single level codes that would provide the same overall code rate. Concatenation is an alternative way of improving the performance without increasing the length of the code. Concatenation can be viewed as a means to add, explicit diversity to the channel; the overall order of diversity for a given channel is the sum of the added explicit diversity and the implicit diversity introduced by the channel itself. The gain achieved using concatenated codes is proportional to the relative increase in diversity. For channels with small spread, concatenation is a powerful means of achieving small bit error rates at reasonable values of signal-tonoise ratios.

There are many factors that need to be considered in turbo code design. In the first instance, a trade-off between the BER and the number of iterations needs to be made,

e.g., more iterations will get lower BER, but the decoding delay is also longer. Secondly, the effect of the frame size on the BER also needs to be considered. Thirdly, the code rate is another factor that needs to be considered. A higher coding rate needs more bandwidth. The number of decoding iterations have the effect of improving the error performance at each additional iteration.

# CHAPTER SEVEN RESEARCH FINDINGS, OBSERVATIONS, CONCLUSIONS AND RECOMMENDATIONS

### 7.1 RESEARCH FINDINGS

The contribution of this thesis is a detailed analysis of the critical design parameters of robust codes, and the demonstration of the key design issues involved and derivation of results of turbo codes developed by simulation using MATLAB. Error correction techniques play an important role in making wireless communications ever and ever more efficient and reliable. Emerging wireless technologies have already adopted some of the most robust channel error coding techniques. These include turbo codes and low density parity check codes. This thesis has demonstrated that the choice of the type of robust coding technique to use in a particular application depends on several factors, such as channel noise, multipath fading, and co-channel interference. We have established that the performance of a given turbo code will depend on the size of the input message or frame size, the number of decoding iterations, and the design and size of interleavers.

# 7.2 OBSERVATIONS

# 7.2.1 Merits of turbo codes compared to convolution codes for third generation wireless systems

There are several reasons why turbo codes are especially suited for high-speed data services in wireless systems of the third-generation and above [38].

- At high speeds, sufficiently long blocks of data can be accumulated without causing substantial delay in the system.
- Error-free data transmission is typically by an automatic repeat request (ARQ) protocol implemented in higher layers. As such the more appropriate figure of merit is the frame error rate (FER) rather than the bit error rate (BER).
- As the information frame size increases from 512 to 3072, the frame error rate for turbo codes decreases sharply, at least in the waterfall region, while that of the convolution codes is essentially uniform across the frame and is a constant independent of frame size. Thus, for a given  $E_b/N_o$ , the expected number of bit

errors increases with frame size, and the frame error rate worsens. For turbo codes, however, the power of the code increases significantly as the frame size increases due to spectral thinning. This increase in power is more than sufficient to overcome the burden of protecting a larger frame of data.

- Fast power control is employed in third-generation systems. Indeed, without any power control, the performance advantage of turbo codes over convolution codes decreases considerably. At relatively short frame sizes (e.g., 512 bits) the BER and FER performance of turbo codes and convolution codes are so similar that the extra complexity of the turbo decoder would not be worthwhile. However, use of fast power control restores the performance advantage of turbo codes to gains close to those achievable on the additive white Gaussian noise (AWGN) channel.
- Turbo codes are mainly attractive for high-data-rate services due to the relatively long interleaver. Initially, the standard bodies limited turbo code only to high-data-rate services. Results show that turbo codes still offer some modest gains with respect to convolution codes with a frame size as low as 100 bits. For extremely short interleavers, convolution codes outperform turbo codes. Theoretically, it is best to switch to convolution codes when the amount of data to be transmitted is small. However, this switching typically requires signaling; it incurs extra delay and overhead. Due to this reason, third generation systems allow turbo codes to be sued across almost all data rates.
- The constraints on bandwidth, power, and time in many image communication systems prohibit transmission of uncompressed raw image data. Compressed image representation, however, is very sensitive to bit errors, which can severely degrade the quality of the image at the receiver. Therefore, application of channel coding is required before transmission of data over noisy and fading channels. Turbo codes have been tested [39] and proved to be quite reliable for robust transmission of compressed images over noisy and fading channels. Turbo codes are characterized by a large interleaver and their performance improves with increasing interleaver size. Thus, the large number of bits in an image representation makes turbo codes naturally suitable for image transmission.

### 7.2.2 Comparison of Low Density Parity Check Codes and turbo codes

Although implementation of LDPC codes has lagged that of other codes, notably the turbo code, the absence of encumbering software patents has made LDPC attractive to some, and LDPC codes are positioned to become a standard in the developing market for highly efficient data transmission methods. LDPC codes are selected as the DVB-S2 standard over 7 other turbo code based candidates [47] because of their more efficient implementation as well as better performance- this is mainly because LDPC have a much lower latency compared to turbo codes although at higher BER rates compared to turbo codes.

Turbo Codes are high-performance error-correcting codes that are good choices for limited-bandwidth, high-noise communications. They come closer to the Shannon limit than was originally thought possible, and the feedback technique they use has inspired new near-Shannon limit coding techniques. The primary advantage of turbo codes is their ability to increase the usable bit rate of a signal without increasing transmission power, or similarly, maintaining the bit rate of a signal while decreasing the transmission power. Few coding schemes have ventured closer to the Shannon limit of usable performance compared to turbo codes. The disadvantages of turbo codes are high decoding complexity and high decoding latency. These properties make turbo codes suboptimal for low-latency, battery-limited voice applications, but make them just fine for high-latency applications like NASA satellites and earth-orbit satellite TV systems.

Shannon observed that the longer the code word, the more difficult it was for noise to cause errors. By producing arbitrarily long code words, one can approach the Shannon limit. Long code words, however, have impractical space requirements, and the many bits transmitted per input bit reduce the useful transmit rate. Instead of producing a stream of bits from the signal, a turbo code receiver produces a likelihood measure for each bit. An iterative process of comparing parts of the code words turns an impractical code word space requirement into a tractable one that requires a number of steps. The two decoders converge on a solution, and report the result as a block of bits.

The impact of turbo codes extends beyond just the high-performance codes themselves. Their unexpected creation showed information theorists that the creation of higher-performance error correction codes was indeed possible, and has inspired new coding techniques to deal with multipath propagation. In addition, they have inspired low-density parity check (LDPC) codes, which come even closer to the Shannon limit and are unencumbered by patents.

### 7.3 CONCLUSIONS

In emerging digital wireless communication systems, the purpose of channel coding is to add redundancy to the binary data stream to combat the effect of signal degradation of the channel. Signal degradation is usually due to noise and fading phenomena. Ideally, channel codes should meet the following requirements:

- Channel codes should be high rate to maximize data throughput.
- Channel codes should have good bit error rate (BER) performance at the desired signal-to-noise ratios (SNR) to minimize the energy needed for transmission.
- Channel codes should have low encoder/decoder complexity to limit the size and cost of the transceivers.
- Channel codes should only introduce minimal delays, especially in voice transmission, so that no degradation in signal quality is detectable. [48]

These requirements are very difficult to obtain simultaneously, because excellent performance in one requirement usually comes at a price of reduced performance in another. However, for mobile cellular voice and data communications, it is desirable that all these requirements are met, which makes design of mobile cellular communication systems for both voice and data, quite a challenging task. There are two categories of channel codes which qualify to be regarded as *robust codes* because of their salient features and current applicability and reliability levels. These include *turbo codes* and *low density parity check codes*. Our comprehensive research is here limited to turbo codes only for obvious reasons.

We have established that various parameters affect the performance of turbo codes. These parameters include:

- The size of the input message or frame size.
- The number of decoding iterations.

• The design and size of the interleaver used.

Our results are presented for the cases of either additive white Gaussian noise (AWGN) or Rayleigh flat fading channels. Normally, a code designed for an AWGN channel only, has to be modified by combining it with some *interleaving* in order to make it suitable for fading channels. Thus, the criterion for the code design has to change to provide for fading diversity [18]. Code designs for fading channels combine block and Convolution codes with interleaving, and modify the coding process to provide for maximum fading diversity.

Simulation results show that turbo codes are powerful error correcting codes under noisy and fading environments. However, there are many factors which need to be considered in the turbo code design process. First, a tradeoff between BER and the number of iterations need to be made, e.g., more iterations lead to lower BER, while the decoding delay gets longer. Secondly, the effect of the frame size on the BER also needs to be considered. Although the turbo code with larger frame size has a better performance, the output delay is longer. Thirdly, the code rate is another factor to be considered.

A higher coding rate leads to greater bandwidth requirements.

Simulation results reveal that turbo code performance increases with increasing numbers of iterations over the AWGN channel and the Rayleigh channel. The improvement in performance with increased iterations, however, comes at the expense of complexity and time delay (latency). The higher the number of iterations involved the higher the complexity. From the error performance results, it is evident that turbo codes are quite suitable for the emerging wireless communications applications taking into account of the fore-mentioned requirements.

Turbo codes introduced in 1993 and low-density parity-check (LDPC) codes, introduced by Gallager in the sixties and revisited after turbo codes were invented, are the most exciting and important development in coding theory in many years. Researchers around the world have been able to extend the basic idea to other forms of code concatenations, with various applications for transmission over fading channels, band-limited satellite channels, and channels with inter-symbol interference.

After slightly more than a decade from their date of discovery, turbo and LDPC codes have been accepted as robust coding standards for 3G wireless communications systems, like CDMA2000 and UMTS, for satellite and deep space applications as the new Consultative Committee for Space Data Systems (CCSDS) telemetry channel coding standard, for the new digital video broadcasting by satellite DVBS-2, and many others. Both classes of codes rely on the application of soft, decentralized decoding algorithms, like the BCJR for turbo codes and the "message passing" for LDPC codes.

### 7.4 RECOMMENDATIONS

The major disadvantage pertaining to additional complexity and delay of turbo codes has to be dealt with in order to achieve desirable objectives through better hardware and software design approaches.

From the error performance analysis and results, it is evident that turbo codes are quite suitable for the emerging wireless communications technologies and applications, assuming that the disadvantages mentioned above could be minimized with other developments in hardware design.

Random coding theory states that almost all randomly designed codes are good, as long as they are sufficiently long [55]. However, just a few of them in terms of the parity check bits make decoding simple to implement. They were for some period in the 1960s, and even in the 1980s still rather too complex to implement. The early 1990s saw the discovery of turbo codes by Berrou, *et al* whose performance is built on large random interleavers, and iterative decoding. Recently, Neal and McKay "rediscovered" the low density parity check codes employing iterative decoding to achieve turbo-like performance [55]. To design a good LDPC code, efficient use of modern random access memory (RAM) architecture is the key. Design alternatives of LDPC codes that have sufficient structure to allow efficient read/write operations, while retaining sufficient "randomness" to retain coding gain are still needed. LDPC codes are preferred to turbo codes in some applications because of their more efficient implementation as well as better performance.

It is quite clear that simulation tools while difficult to design, offer cheaper and fairly good alternatives to electrical laboratories which involve high capital costs and often requiring connecting together several hardware components to form a complete system.

# 7.4.1 Why LDPC is a strong candidate for Emerging Wireless Systems

Since emerging wireless systems typically require higher data throughput in a given bandwidth, this implies that:

- High rate forward-error correction (FEC) systems with higher order modulation types (e.g. CDMA2000, EVDV, and HSDPA) will be required.
- Applications requiring high speed, by default should permit use of long blocks. [55].

Since turbo codes tend to lose performance at high code rates due to excessive puncturing, and turbo trellis codes become rather complex to implement for higher order modulation, the LDPC code appears to be a strong candidate for next generation wireless systems applications. It is envisaged that relative gain of LDPC codes over 3G turbo codes in the AWGN and fading channels will also be preserved when power control is applied.

Low-density parity check codes (LDPCC) are favoured over turbo codes for large bock sizes due to their superior error correction performance.

# 7.4.2 Call for Additional Design Criterion

The performance of a turbo code is dependent mainly on two properties: its distance spectrum and its suitability to be iteratively decoded. Both of these properties are influenced by the choice of the interleaver used in the turbo encoder. Turbo codes are however decoded iteratively, which is suboptimal to maximum likelihood decoding. This calls for an additional design criterion which can minimize performance deterioration due to iterative decoding.

Both implementation complexity and latency associated with the decoding process, are some of the major bottlenecks which need further research to improve overall performance of turbo codes.

# 7.4.3 Future Research endeavors

The future research directions proposed here for the furtherance of robust code performance research are as follows:

- Examining the BER performance of a more realistic fading channel model, such as the Jake's model [21]. Our BER performance simulations have been limited to AWGN and Rayleigh models.
- Investigating the use of other modulation techniques or signal constellations. Our research has been limited to BPSK modulation systems and 8 signal constellations.
- Performing simulations to obtain the BER performance curves for BERs below 10<sup>-5</sup> in order to study the "error floor" area of the BER curves. It is suspected that the performance in this area of the BER curves might oscillate slightly when the symbol size is increased.
- The literature review has revealed several fundamental channel code performance bounds and capacity limits at various levels, and the need for considering several design alternatives. Examples of these performance bounds and limits include:
  - The Shannon capacity limit and bound
  - The Singelton bound
  - The Gilbert-Varshamov bound, and
  - The Hamming bound.

There is a need for clarifying on the convergence or divergence of these concepts to a similar point and their comparative advantages with reference to established channel code performance metrics.

\_\_\_\_\_

#### REFERENCES

- Bernard Sklar, "Digital Communications Fundamentals and Applications", Englewood Cliffs, New Jersey: Prentice Hall, 1988.
- [2] "Error Correction Power for the Future," ECC Technologies, Inc., 1996 2001.
- [3] C.E. Shannon, A Mathematical Theory of Communication", Bell Systems Technical Journal No. 27, 379-423(1949).
- [4] R.W. Hamming, "Error detecting and error correcting codes," Bell System Technical Journal, vol.29, pp.147-160, 1950.
- [5] "Golay Codes", http://www-math.mit.edu/phase2/UJM/vol1/MKANEM~1.PDF
- [6] Geir E. Qien, Information Theory: The Foundation of Modern Communications, Telektronikk, 1.2002.
- [7] G. Calhoun,"Third Generation Wireless Systems, Volume 1: Post-Shannon Signal Architectures." Norwood: Artech House, 2003.
- [8] Paul Whalen, "Signaling Processes in the Third Generation Wireless Systems", Hill Associates, April, 2003.
- [9] Jay M. Jacobsmeyer, "Introduction to Error-Control Coding", Pericle Communications Company, 2004.
- [10] R. Gallager, "A simple derivation of the coding theorem and some applications, "*IEEE Trans. on Inform. Theory*, vol. 11, pp. 3-18, Jan. 1965.
- [11] "A relationship between the Berlekamp Massey and Euclidian Algorithms", <u>http://www.springerlink.com/index/c441782958971I62/...</u>
- Berrou, Glavieux, and Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding : Turbo-Codes", Proceedings of the IEEE International Communications Conference, ENST Bretagne, France 1993.
- [13] F. Guterl,"Compact Disc." IEEE Spectrum, Vol. 25, No.11, pp.102-101, 1988.
- [14] R.G. Gallager, "Information Theory and Reliable Communications", Wiley, 1968.
- [15] "Perfect Codes",

http://tesla.csl.uiuc.edu/~koetter/ece459/seminars2002/swartz.doc

- [16] R.E. Blahut, "Theory and Practice of Error Control codes", Reading, Massachusetts: Addison- Wesley Publishing Company, 1983.
- [17] Levent Sevgi, "Modeling and Simulation Concepts in Engineering
   Education: Virtual Tools", Dogus University, Electronics and Communication
   Engineering Departments, 34722 Istanbul, TURKEY.
   http://journals.tubitak.gov.tr/elektrik/issues/elk-06-14-1/elk-14-1-12-0509-12.pdf
- [18] Andrea Goldsmith, "Wireless communications." 2005 Cambridge University Press.
- [19] Bernard Sklar, "Mitigating the Degradation Effects of Fading Channels", http://www.informit.com/content/images/art\_sklar6\_mitigating/
- [20] Ana Aguiar, James Gross, "Wireless Channel Models", TKN Technical Report TKN-03-007, Telecommunication Networks Group, Technical University Berlin, April 2003.
- [21] W. C. Jakes, "Microwave Mobile Communications.", IEEE Press, Wiley Interscience, 1994.
- [22] Li Tang Zhu Hongbo, "Analysis and Simulation of Nakagami Fading Channel with MATLAB", Asia-Pacific Conference on Environmental Electromagnetics, CEEM'2003 Nov. 4-7, 2003 Hangzhou, CHINA.
- [23] W.C.Y. Lee. Mobile Cellular Telecommunications. McGraw-Hill International editions, 1995.
- [24] R. Steele and L. Hanzo, editors: Mobile Radio Communications. J. Wiley & Sons Ltd, 2000.
- [25] J.M. Kasana Isaac, "Evaluation of RF Propagation Models for GSM Planning for Kampala", October, 2003, M.ENG.[EE] Thesis, Dept. of Electrical Engineering, Faculty of Technology, Makerere University, Uganda.
- [26] Adaptive Impairments Mitigation Techniques, http://www.estec.esa.nl/xewww/cost255/part1.pdf
- [27] Log-normal shadowing path loss model, http://www.isi.edu/~weiye/pub/propagation\_ns.pdf
- [28] Matthew C. Valenti and Jian Sun, "Turbo Codes", DOWLA: Handbook of RF and Wireless Technologies, 2003. <u>http://www.cs.wvu.edu/~mvalenti/documents/DOWLA-CH12.pdf</u>
- [29] Turbo Codes, Wikipedia Encyclopedia, <u>http://en.wikipedia.org/wiki/Turbo\_code</u>, 06.11.2006.

- [30] **Reed-Solomon Error Correction** Wikipedia, the free encyclopidea http://en.wikipedia.org/wiki/Reed-Solomon\_error\_correction 10/9/2006.
- [31] Matthew Valenti, Turbo Codes and Iterative processing, IEEE New Zealand Wireless Communications Symposium, November 1998.
- [32] Dowla, F., Handbook of RF and Wireless Technologies, Newnes 2004, ISBN 0750676957
- [33] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: Turbo codes," IEEE Transactions on Communications, Vol. 44, pp.1261-1271, October 1996.
- [34] J.P. Woodard and L. Hanzo, "Comparative study of turbo decoding techniques: An overview, " IEEE Transactions on Vehicular Technology, vol. 49, pp.2208-2233, November 2000.
- [35] M.R.D. Rodrigues, I. Chatzgeorgiou, I.J. Wassell and R. Carrasco, "On the Performance of Turbo Codes in quasi-Static Fading Channels." Laboratory for Communication Engineering, University of Cambridge, U.K.
- [36] K. Ramasamy, Mohammad Umar Siddiqi, Mohamad Yusoff Alias and A. Arunagiri, "Performance comparison of Convolution and block turbo codes", IEICE Electron. Express, Vol. 3, No.13, pp.322-327 (2006).
- [37] M.C. Valenti, and J. Sun, "The UMTS Turbo Code and an Efficient Decoder Implementation Suitable for Software-Defined Radios", International Journal of Wireless Information Networks, Vol. 8, No.4, October 2001.
- [38] Lin-Nan Lee, A. Roger Hammons, Jr., Feng-Wen Sun, and Mustafa Eroz,
   "Application and Standardization of Turbo Codes in Third-Generation High-Speed Wireless Data Services", IEEE Trans. On Vehicular Technology, Vol. 49, No. 6, November2000.
- [39] Jiali He, Daniel J. Costello Jr., et al, "On the Application of Turbo Codes to the Robust Transmission of Compressed Images", Dept. of Electrical Engineering, University of Notre Dame, Notre Dame, IN 46556 USA.
- [40] Fu-hua Huang, "**Performance Analysis of Turbo Codes**", <u>http://scholar.lib.vt.edu/thesis/available/etd-71897-15815únrestricted/</u>
- [41] Samir S. Soliman, and Karim Mokrani, "Performance of Coded Systems Over Fading Dispersive Channels", IEEE Transactions on Communications, VOL. 40. No.1, January 1992.

- [42] Hank Wallace, "Error Detection and Correction Using the BCH Code", Copyright © 2001 Hank Wallace.
- [43] Charan Langton," Intuitive Guide to Principles of Communication, <u>http://www.complextoreal.com</u>
- [44] Kulikowski, K.J., Karpovsky, M.G., Alexander Taubin, "Robust Codes and Robust, Fault Tolerant Architectures of Advanced Encryption Standard", Reliable Computing Laboratory, Boston University, Boston, MA02215.
- [45] Performance Measures and Design Tradeoffs, <u>http://www.rand.org/publications/MR/MR960/MR960.chap2.pdf</u>, 19/10/2005.
- [46] Rajeshwari Banakar, M. "A Low Power Design Methodology for Turbo Encoder and Decoder", Ph.D Dissertation, Department of Electrical engineering, Indian Institute of Technology, Delhi, July 2004.
- [47] Lin-Nan Lee, LDPC Codes, Application to Next Generation Communication Systems, Hughes Network Systems, Germantown, Maryland 20854, October 8, 2003.
- [48] Mark Bingeman, "Symbol-Based Turbo Codes for Wireless
   Communications", Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, Ontario, Canada, 2002.
- [49] Shobha, R., Subharani, S., and Shanmugam, A., "Optimal Choice of Interleaver for Turbo Codes", Academic Open Internet Journal, Volume 15, 2005.
- [50] Branka Vucetic and Jinhong Yuan,"Turbo Codes Principles and Applications", Buton: Kluwer Academic Publishers, 2000.
- [51] Springer Online Works: "Shift Register Sequences", http://eom.springer.de/s/s130270.htm - 33k 30/09/2007
- [52] Teodor Iliev, "A study of turbo codes for UMTS third generation cellular standard.", International Conference on Computer Systems and Technologies – CompSysTech'o7, University of Rousse, Bulgaria.
- [53] Christian Schlegel, Error Control Coding Lecture Notes, <u>http://www.ece.ualberta.ca/~hcdc/ErrorContrCoding.html</u>, 22/04.2005.
- [54] Christian Schlegel, Error Control Coding : Beginnings http://www.ece.ualberta.ca/~hcdc/library/ErrorContrClass/BlockCodeBasic.pdf
- [55] Crozier, S., Hunt, A, and Lodge, J., "Method of enhanced max-log-a posteriori probability processing", U.S. Patent 6,145,114, November 7, 2000.

- [56] Lin-Nan Lee, "LDPC Codes, Application to Next Generation Communication Systems", Hughes Network Systems, Germantown, Maryland 20854, October 8, 2003.
- [57] G.G. Raleigh, and J.M.Cioffi, "Spatio-Temporal Coding for wireless Communications", IEEE Transactions on Communications, Vol.46, No.3, March1998.
- [58] Chip Fleming, "A Tutorial on Convolutional Coding with Viterbi Decoding", Spectrum Applications 1999 – 2003.
- [59] Bernard Sklar, "Rayleigh Fading Channels in Mobile Digital Communication Systems, Part II : Mitigation", Communications Engineering Services, IEEE Communications Magazine, July 1997, p.102 – 109.
- [60] Joao Martins, A. Giulietti, M. Strum, "Performance Comparison of Convolutional and Block Turbo Codes for WLAN Applications", 4. IEEE International Caracas Conference on Devices, Circuits and Systems, Aruba, April 17. – 19., 2002.
- [61] A. Burr, "**Turbo Codes : the Ultimate Error Control Codes**?" http://www.elec.york.ac.uk/comms/papers/burr01\_ecej.pdf
- [62] M.C. Valenti, and Jian Sun, "Turbo Codes", DOWLA: Handbook of RF and Wireless Technologies, 2003/9/25.
- [63] David J.C. MacKay (2003), "Information theory, inference and learning algorithms", CUP, ISBN 0-521-64298-1.
- [64] Todd K. Moon(2005), "Error Correction Coding, Mathematical Methods and Algorithms." Wiley, ISBN 0-471-64800-0 (Includes code).

[65] David J.C. MacKay and Radford M. Neal, "Near Shannon Limit Performance of Low Density Parity Check Codes", Electronics Letters, July 1996, available online at http://wikipedia.org

[66] Third Generation Partnership Project2 (3GPP2), "Physical Layer Standard for CDMA2000 Spread Spectrum Systems, Release C", 3GPP2C, S0002 – C, Version 1.0 (May 28, 2002): pp.115 – 122; available online at http://www.3gpp2.org.

[67] Thomas J. Richardson and Rüdiger L. Urbanke, "Efficient Encoding of Low-Density Parity- Check Codes", IEEE Transactions in Information Theory, 47(2), February 2001.

[68] Y. Kou, S. Lin and M. Fossorier, "**Low-Density Parity-Check Codes Based on Finite Geometries: A Rediscovery and New Results**", IEEE Transactions on Information Theory, vol. 47, no.7, Nov. 2001, pp2711 – 2736.

[69] Sae-Young Chung, G., David Forneyt, Jr., Thomas J. Richardson, and Rüdiger Urbanke, "**On the Design of Low-Density Parity-Check Codes within 0.0045 dB of the Shannon Limit**", IEEE Communication Letters, vol.5, pp. 58-60, Feb2001, ISSN 1089 – 7798.

[70] European Telecommunications Standards Institute (ETSI), "Universal Mobile
Telecommunications System (UMTS): Multiplexing and Channel Coding
(FDD)", 3GPPTS 125.212 available online at <a href="http://www.3gpp.org">http://www.3gpp.org</a>

- [71] **Turbo8 An 8-State Turbo Code Design & Simulation Tool** <u>http://www.crc.gc.ca/en/html/fec/home/turbo8/turbo8</u>
- [72] **Coding Theory** From Wikipedia, the free encyclopedia, <u>http://en.wikipedia.org/wiki/Coding\_theory 11.12.2010</u>
- [73] Peter Stuckmann, Rainer Zimmermann, "**Towards Ubiquitous and Unlimited-Capacity Communication Networks**" – European Research in Framework Program 7, European Commission, 2007-2013.

# APPENDICES

# **APPENDIX A**

### **Simulation Results of Turbo Code Performance**

Same Block Length = 512 bits;		Same code rate $r = 1/2$		
Unpunctured Code		Punct	Punctured Code	
t8k5	12r12	t8k512r12dp		
$E_b/N_o(dB)$	Bit Error	$E_b/N_o(dB)$	Bit Error	
	Rate(BER)		Rate(BER)	
0.5	1.03e-001	0.5	1.47e-001	
0.75	6.51e-002	0.75	9.70e-002	
1.00	2.78e-002	1.00	4.27e-002	
1.25	8.65e-003	1.25	1.67e-002	
1.50	1.77e-003	1.50	3.40e-003	
1.75	2.03e-004	1.75	4.43e-004	
2.00	1.69e-005	2.00	4.24e-005	
2.25		2.25	1.60e-006	
Time elapsed = $376.250$ sec.		Time elaps	ed = 475.172 sec.	

Same Block Length = 1024 bits;		Same code rate $r = 1/2$	
Unpunctured Code t8k1024r12		Punctured Code t8k1024r12dp	
E <sub>b</sub> /N <sub>o</sub> (dB)	Bit Error Rate(BER)	E <sub>b</sub> /N <sub>o</sub> (dB)	Bit Error Rate(BER)
0.5	1.10e-001	0.5	1.55e-001
0.75	6.01e-002	0.75	9.17e-002
1.00	1.97e-002	1.00	3.08e-002
1.25	2.66e-003	1.25	4.58e-003
1.50	1.61e-004	1.50	2.92e-004
1.75	2.99e-006	1.75	5.10e-006
Time elapsed = $298.797$ sec.		Time elapsed	= 459.828 sec.

Same Block Length = 1504 bits;		Same code rate $r = 1/2$	
Unpunctured Code t8k1504r12		Punctured Code t8k1504r12dp	
E <sub>b</sub> /N <sub>o</sub> (dB)	Bit Error Rate(BER)	$E_b/N_o(dB)$	Bit Error Rate(BER)
0.5	1.09e-001	0.5	1.48e-001
0.75	5.68e-002	0.75	8.80e-002
1.00	1.18e-002	1.00	1.80e-002
1.25	7.69e-004	1.25	1.34e-003
1.50	1.34e-005	1.50	2.34e-005
1.75	1.91e-007	1.75	2.19e-006

Same Block Length = 1504 bits;		Same code rate $r = 1/3$	
Unpunctured Code t8k1504r13		Punctured Code t8k1504r13dp	
E <sub>b</sub> /N <sub>o</sub> (dB)	Bit Error Rate(BER)	E <sub>b</sub> /N <sub>o</sub> (dB)	Bit Error Rate(BER)
0.0	1.04e-001	0.0	1.05e-001
0.25	3.06e-002	0.25	5.29e-002
0.5	3.14e-002	0.5	1.76e-002
0.75	8.48e-005	0.75	4.20e-003
1.00	3.93e-007	1.00	5.74e-004
1.25	-	1.25	4.57e-005
Time elapsed = $3806.422$ sec.		Time elapsed = $423.453$ sec.	
Same Block Length = 1504 bits;		Same code :	rate $r = 1/3$
Unpunctured Code t8k1504r13		Punctured Code t8k1504r13dp	
E <sub>b</sub> /N <sub>o</sub> (dB)	Bit Error Rate(BER)	$E_b/N_o(dB)$	Bit Error Rate(BER)
0.0	1.04e-001	0.0	1.05e-001
0.25	3.06e-002	0.25	5.29e-002
0.5	3.14e-002	0.5	1.76e-002
0.75	8.48e-005	0.75	4.20e-003
1.00	3.93e-007	1.00	5.74e-004

1.25	-		
Time elapsed = $3806.422$ sec.		Time elapsed =	423.453 sec.

Same Block Length = 1024 bits;		Same code rate $r = 2/3$	
Unpunctured Code t8k1024r23		Punctured Code t8k1024r23dp	
$E_b/N_o(dB)$	Bit Error Rate(BER)	$E_b/N_o(dB)$	Bit Error Rate(BER)
1.50	4.73e-002	1.50	6.72e-002
1.75	1.74e-002	1.75	2.65e-002
2.00	3.72e-003	2.00	4.86e-003
2.25	4.41e-004	2.25	5.20e-004
2.50	2.54e-005	2.50	1.98e-005
2.75	1.23e-006	2.75	
Time elapsed = $2671.593$ sec.		Time elap	sed = 4988.312sec.

Same Block Length = 1024bits;		Same code rate $r = 4/5$	
Unpunctured Code t8k1024r45		Punctured Code t8k1024r45dp	
E <sub>b</sub> /N <sub>o</sub> (dB)	Bit Error Rate(BER)	$E_b/N_o(dB)$	Bit Error Rate(BER)
2.50	2.00e-002	2.50	4.10e-002
2.75	7.23e-003	2.75	1.51e-002
3.00	1.64e-003	3.00	2.95e-003
3.25	2.23e-004	3.25	3.41e-004
3.50	2.41e-005	3.50	2.12e-005
3.75	2.86e-006	3.75	7.63e-007
Time elapsed = $6876.078$ sec.		Time elapsed = $36$	668.563 sec.

Same Block Length = 1504bits;		Same code rate $r = 8/9$	
Unpunctured Code t8k1504r89		Punctured Code t8k1504r89dp	
$E_b/N_o(dB)$	Bit Error Rate(BER)	E <sub>b</sub> /N <sub>o</sub> (dB)	Bit Error Rate(BER)
3.00	2.77e-002	3.00	7.23e-002
3.25	1.77e-003	3.25	5.80e-002
3.50	6.69e-003	3.50	3.43e-002
3.75	2.02e-003	3.75	1.29e-002
4.00	4.10e-004	4.00	2.28e-003
4.25	4.72e-005	4.25	2.37e-004
4.50	6.55e-006	4.50	1.09e-005
4.75	1.25e-006	4.75	-

Г

Time elapsed = $7758.360$ sec.	Time elapsed = $2587.343$ sec.
Time elapsed = $7758.360$ sec.	Time elapsed = $2587.343$ sec.

Varying the number of iterations, while the block length is kept constant at 4000 bits, and other parameters are assumed fixed:

A. No. of iterations = 1 $E_b/N_o(dB)$ 0.2 0.6 0 0.4 0.8 1.0 1.2 1.4 0.1316 0.1519 0.1298 0.1081 0.0926 0.0719 **Bit Error** 0.0663 0.0503 Rate(BER) B. No. of iterations = 4 $E_b/N_o(dB)$ 0 0.2 0.4 0.6 0.8 1.0 1.2 Bit Error 0.1616 0.1198 0.0602 0.0321 0.0106 0.0011 9.914e-4 Rate(BER) C. No. of iterations = 80.2 0.4 1.0 1.4  $E_{\rm b}/N_{\rm o}(dB)$ 0.6 0.8 1.2 0 0.0783 0.0650 0.0453 0.0173 0.0068 0.0027 6.1713e-2.6687e-Bit Error 4 Rate(BER) 4 1.8 2.0 2.1021e-5 3.7793e-6 Comparison of results assuming different information block lengths are used and other basic conditions remain the same, i.e., No. of iterations = 8. A. Length of block: Block Length = 1000 bits  $E_b/N_o(dB)$ 0.2 0.4 0.6 0.8 1.0 1.2 0 0.1225 0.0729 0.0208 0.0034 0.0014 Bit Error 0.0942 2.7042e-4 Rate(BER) B. Length of block: Block\_Length = 4000 bits  $E_b/N_o(dB)$ 0 0.2 0.4 0.6 0.8 1.0 0.0783 0.0453 0.0173 0.0044 0.0017 Bit Error 0.0650 Rate(BER) 1.8 1.2 1.4 2.0 6.1713e-4 2.6687e-4 6.1021e-5 3.7793e-6 C. Length of block: Block\_Length = 8000 bits  $E_b/N_o(dB)$ 0.2 0 0.4 0.6 0.8 1.0 Bit Error 0.1242 0.0997 0.0491 0.0059 3.1577e-6 1.0004e-6 Rate(BER)

### APPENDIX B

### MATLAB CODE FOR PLOTTING TURBO-CODE PERFORMANCE GRAPHS

EbNo = [0 0.2 0.4 0.6 0.8 1.0 1.2 1.5 2.0];

BER = [0.1303 0.0964 0.0583 0.0064 1.1675e-4 2.3166e-6 6.5346e-7 5.785e-7 2.2556e-7]; semilogy(EbNo, BER,'--r\*') xlabel('Eb/No(dB)') ylabel('Bit Error Probability')

$$\begin{split} & EbNo1 = [0\ 0.2\ 0.4\ 0.6\ 0.8\ 1.0\ 1.2\ 1.5\ 2.0]; \\ & BER1 = [0.1303\ 0.0964\ 0.0583\ 0.0064\ 1.1675e-4\ 2.3166e-6\ 6.5346e-7\ 5.785e-7\ 2.2556e-7]; \\ & EbNo2 = 0:0.2:1.4; \\ & EbNo3 = 0:0.2:1.2; \\ & BER2 = [0.1316\ 0.1519\ 0.1098\ 0.1081\ 0.0926\ 0.0719\ 0.0663\ 0.0503]; \\ & BER3 = [0.1616\ 0.1298\ 0.0602\ 0.0321\ 0.0106\ 0.0011\ 2.6551e-4]; \\ & semilogy(EbNo2,BER2,'-b',EbNo3,BER3,'-g',EbNo1,BER1,'--r*') \end{split}$$

title('\fontname{Ariel}Performance Comparison of Different Iteration Times.','FontSize',14); xlabel('Eb/No(dB)') ylabel('Bit Error Probability') legend('nIterate=1','nIterate=8')

EbNo1 = [0 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6 2.0]; BER1 =  $[0.1126\ 0.0931\ 0.0485\ 0.0210\ 0.0086\ 0.002\ 8.5064e-5\ 9.8758e-6\ 4.0158e-6\ 3.7793e-6];$ EbNo2 = [0 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.8 2.0]; BER2 = [0.0783 0.0650 0.0453 0.0173 0.0055 0.0017 6.1713e-4 2.6687e-4 6.1021e-5 1.6653e-5]; EbNo3 = [0 0.2 0.4 0.6 0.8 1.0 1.2 1.5 2.0]; BER3 =  $[0.1303\ 0.0964\ 0.0583\ 0.0064\ 1.1675e-4\ 2.3166e-6\ 6.5346e-7\ 5.785e-7\ 2.2556e-7];$ semilogy(EbNo1,BER1,'--b\*',EbNo2,BER2,':k.', EbNo3, BER3,'-gd') title('\fontname{Ariel}Comparison of Performance of Different RSC Encoder Configurations.','FontSize',14); xlabel('Eb/No(dB)') ylabel('Bit Error Probability') legend('(64,6)','(44,54)','(64,54)') EbNo1 = [0 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.6 2.0]; BER1 = [0.1126 0.0931 0.0485 0.0210 0.0086 0.002 8.5064e-5 9.8758e-6 4.0158e-6 3.7793e-6]; EbNo2 = [0 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.8 2.0]; BER2 =  $[0.0783 \ 0.0650 \ 0.0453 \ 0.0173 \ 0.0055 \ 0.0017 \ 6.1713e-4 \ 2.6687e-4 \ 6.1021e-5 \ 1.6653e-5];$ EbNo3 = [0 0.2 0.4 0.6 0.8 1.0 1.2 1.5 2.0]; BER3 = [0.1303 0.0964 0.0583 0.0064 1.1675e-4 2.3166e-6 6.5346e-7 5.785e-7 2.2556e-7]; semilogy(EbNo1,BER1,'--b\*',EbNo2,BER2,':k.', EbNo3, BER3,'-gd') title('\fontname{Ariel}Comparison of Performance of Different RSC Encoder Configurations.','FontSize',14); xlabel('Eb/No(dB)') ylabel('Bit Error Probability') legend('(64,6)','(44,54)','(64,54)') EbNo1 = [0.5 0.75 1.00 1.25 1.50 1.75 2.00]; BER1 = [1.03e-001 6.51e-002 2.78e-002 8.65e-003 1.77e-003 2.03e-004 1.69e-005];

 $EbNo2 = [0.5 \ 0.75 \ 1.00 \ 1.25 \ 1.50 \ 1.75 \ 2.00 \ 2.25];$   $BER2 = [1.47e-001 \ 9.70e-002 \ 4.27e-002 \ 1.67e-002 \ 3.40e-003 \ 4.43e-004 \ 4.24e-005 \ 1.60e-006];$   $EbNo3 = [0.5 \ 0.75 \ 1.00 \ 1.25 \ 1.50];$ 

BER3 = [1.10e-001 6.01e-002 1.97e-002 2.66e-003 1.61e-004]; EbNo4 = [0.5 0.75 1.00 1.25 1.50]; BER4 = [1.55e-001 9.17e-002 3.08e-002 4.58e-003 2.92e-004];  $EbNo5 = [0.5 \ 0.75 \ 1.00 \ 1.25 \ 1.50];$ BER5 = [1.48e-001 8.80e-002 1.80e-002 1.34e-003 2.34e-005]; EbNo6 = [0.5 0.75 1.00 1.25 1.50 1.75]; BER6 = [1.09e-001 5.68e-002 1.18e-002 7.69e-004 1.34e-005 1.91e-007]; semilogy(EbNo1,BER1,'--b\*',EbNo2,BER2,':k.',EbNo3, BER3,'-g+',EbNo4,BER4,':r\*',EbNo6,BER6,'-.g.',EbNo5,BER5,'-.md') title('\fontname{Ariel}Comparison of Unpunctured and Punctured Configurations.','FontSize',14); xlabel('Eb/No(dB)') vlabel('Bit Error Probability') legend('t8k512r12','t8k512r12dp','t8k1024r12','t8k1024r12dp','t8k1504r12','t8k1504r12dp')  $EbNo1 = [0\ 0.2\ 0.4\ 0.6\ 0.8\ 1.0\ 1.2\ 1.4];$ BER1 =  $[0.1316 \ 0.1519 \ 0.1298 \ 0.1081 \ 0.0926 \ 0.0719 \ 0.0663 \ 0.0503]$ ; %No. of iterations = 1  $EbNo2 = [0\ 0.2\ 0.4\ 0.6\ 0.8\ 1.0\ 1.2];$ BER2 = [0.1616 0.1198 0.0602 0.0321 0.0106 0.00411 9.914e-4]; % No. of iterations = 4 EbNo3 = [0 0.2 0.4 0.6 0.8 1.0 1.2 1.4 1.8 2.0]; BER3 = [0.0783 0.0650 0.0453 0.0173 0.0068 0.0027 6.1713e-4 2.6687e-4 2.1021e-5 3.7793e-6]; % No. of iterations = 8semilogy(EbNo1,BER1,'--b\*',EbNo2,BER2,':k.',EbNo3, BER3,'-g+') title('\fontname{Ariel}Comparison of varying the number of iterations.','FontSize',14); xlabel('Eb/No(dB)') ylabel('Bit Error Probability') legend('nIterations = 1', 'nIterations=4', 'nIterations=8') EbNo1 = [0.00 0.25 0.50 0.75 1.00]; BER1 = [1.04e-001 3.06e-002 3.14e-003 8.48e-005 3.93e-007]; EbNo2 = [0.00 0.25 0.50 0.75 1.00 1.25]; BER2 = [1.05e-001 5.29e-002 1.76e-002 4.20e-003 5.74e-004 4.57e-005]; EbNo3 = [3.00 3.25 3.50 3.75 4.00 4.25 4.50 4.75 5.00]; BER3 = [2.77e-002 1.77e-002 6.69e-003 2.02e-003 4.10e-004 4.72e-005 6.55e-006 1.25e-006 3.37e-0071: EbNo4 = [3.00 3.25 3.50 3.75 4.00 4.25 4.50]; BER4 = [7.23e-002 5.80e-002 3.43e-002 1.29e-002 2.28e-003 2.37e-004 1.09e-005]; EbNo5 = [2.50 2.75 3.00 3.25 3.50 3.75]; BER5 = [2.00e-002 7.23e-003 1.64e-003 2.23e-004 2.41e-005 2.86e-006 ]; EbNo6 = [2.50 2.75 3.00 3.25 3.50 3.75]; BER6 = [4.10e-002 1.51e-002 2.95e-003 3.41e-004 2.12e-005 7.63e-007]; semilogy(EbNo1,BER1,'--b\*',EbNo2,BER2,':k.',EbNo3, BER3,'-g+',EbNo4,BER4,':r\*',EbNo6,BER6,'-.g.',EbNo5,BER5,'-.md') title('\fontname{Ariel}Comparison of Turbo Codes of Different Code Rates and Block size.','FontSize',14); xlabel('Eb/No(dB)') ylabel('Bit Error Probability') legend('t8k1504r13','t8k1504r13dp','t8k1504r89','t8k1504r89dp','t8k1024r45','t8k1024r45dp') EbNo1 = [3.00 3.25 3.50 3.75 4.00 4.25 4.50 4.75]; BER1 = [6.94e-002 5.65e-002 3.40e-002 1.34e-002 3.91e-003 5.66e-004 7.06e-005 6.62e-006]; EbNo2 = [3.00 3.25 3.50 3.75 4.00 4.25 4.50]; BER2 = [7.23e-002 5.80e-002 3.43e-002 1.29e-002 2.28e-003 2.37e-004 1.09e-005]; semilogy(EbNo1,BER1,'--b\*',EbNo2,BER2,':k.') title('\fontname{Ariel}Turbo codes of different block sizes but having same code rate.','FontSize',14); xlabel('Eb/No(dB)') ylabel('Bit Error Probability') legend('t8k1024r89dp','t8k1504r89dp')

# APPENDIX C

### MATLAB CODE EXAMPLE FOR RADIO PROPAGATION GRAPHIC USER INTERFACE (GUI)

### Free-Space Power Loss Model Matlab Code

function varargout = free\_space\_edit(varargin) % FREE\_SPACE\_EDIT M-file for free\_space\_edit.fig FREE\_SPACE\_EDIT, by itself, creates a new FREE\_SPACE\_EDIT or raises % the existing singleton\*. % % % H = FREE SPACE EDIT returns the handle to a new FREE SPACE EDIT or the handle to the existing singleton\*. % % FREE SPACE EDIT('CALLBACK', hObject, eventData, handles,...) calls the % local function named CALLBACK in FREE\_SPACE\_EDIT.M with the given input % arguments. % FREE\_SPACE\_EDIT('Property', 'Value',...) creates a new FREE\_SPACE\_EDIT % or raises the % existing singleton\*. Starting from the left, property value pairs are applied to the GUI before free space edit OpeningFunction gets called. An % % unrecognized property name or invalid value makes property application stop. All inputs are passed to free space edit OpeningFcn via varargin. % % % \*See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one % instance to run (singleton)". % % See also: GUIDE, GUIDATA, GUIHANDLES % Edit the above text to modify the response to help free\_space\_edit % Last Modified by GUIDE v2.5 27-Apr-2007 18:32:24 % Begin initialization code - DO NOT EDIT gui\_Singleton = 1; gui State = struct('gui Name', mfilename.... 'gui\_Singleton', gui\_Singleton, ... 'gui OpeningFcn', @free space edit OpeningFcn, ... 'gui\_OutputFcn', @free\_space\_edit\_OutputFcn, ... 'gui LayoutFcn', [], ...

```
'gui_Callback', []);
if nargin & isstr(varargin{1})
  gui_State.gui_Callback = str2func(varargin{1});
end
if nargout
  [varargout{1:nargout}] = gui mainfcn(gui State, varargin{:});
else
  gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
% --- Executes just before free_space_edit is made visible.
function free_space_edit_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to free space edit (see VARARGIN)
% Choose default command line output for free_space_edit
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
% UIWAIT makes free_space_edit wait for user response (see UIRESUME)
% uiwait(handles.figure1);
% --- Outputs from this function are returned to the command line.
function varargout = free space edit OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;
% --- Executes during object creation, after setting all properties.
function d input CreateFcn(hObject, eventdata, handles)
% hObject handle to d input (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
      See ISPC and COMPUTER.
%
if ispc
  set(hObject,'BackgroundColor','white');
else
```

set (hObject, `BackgroundColor', get (0, 'defaultUicontrolBackgroundColor')); end

function d\_input\_Callback(hObject, eventdata, handles) % hObject handle to d\_input (see GCBO) % eventdata reserved - to be defined in a future version of MATLAB % handles structure with handles and user data (see GUIDATA) % Hints: get(hObject, 'String') returns contents of d input as text str2double(get(hObject,'String')) returns contents of d\_input as a double % % --- Executes during object creation, after setting all properties. function f input CreateFcn(hObject, eventdata, handles) % hObject handle to f input (see GCBO) % eventdata reserved - to be defined in a future version of MATLAB % handles empty - handles not created until after all CreateFcns called % Hint: edit controls usually have a white background on Windows. See ISPC and COMPUTER. % if ispc set(hObject,'BackgroundColor','white'); else set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor')); end function f input Callback(hObject, eventdata, handles) % hObject handle to f\_input (see GCBO) % eventdata reserved - to be defined in a future version of MATLAB % handles structure with handles and user data (see GUIDATA) % Hints: get(hObject, 'String') returns contents of f\_input as text % str2double(get(hObject, 'String')) returns contents of f\_input as a double f\_input = str2double(get(hObject, 'String')); if isnan(f\_input) set(hObject,'String',0); errordlg('Input must be a Number','Error!'); end data = getappdata(gcbf,'user data'); data.f input = f input; setappdata(gcbf,'user data',data); % --- Executes during object creation, after setting all properties. function p\_input\_CreateFcn(hObject, eventdata, handles) % hObject handle to p\_input (see GCBO) % eventdata reserved - to be defined in a future version of MATLAB % handles empty - handles not created until after all CreateFcns called

```
% Hint: edit controls usually have a white background on Windows.
      See ISPC and COMPUTER.
%
if ispc
  set(hObject,'BackgroundColor','white');
else
  set(hObject, 'BackgroundColor',get(0, 'defaultUicontrolBackgroundColor'));
end
function p_input_Callback(hObject, eventdata, handles)
% hObject handle to p input (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of p_input as text
%
       str2double(get(hObject,'String')) returns contents of p input as a double
p_input = str2double(get(hObject,'String'));
if isnan(p input)
  set(hObject,'String',0);
  errordlg('Input must be a Number','Error!');
end
data = getappdata(gcbf,'user_data');
data.p_input = p_input;
setappdata(gcbf,'user_data',data);
% --- Executes during object creation, after setting all properties.
function tg_input_CreateFcn(hObject, eventdata, handles)
% hObject handle to tg input (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles
           empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
      See ISPC and COMPUTER.
%
if ispc
  set(hObject,'BackgroundColor','white');
else
  set(hObject, 'BackgroundColor',get(0, 'defaultUicontrolBackgroundColor'));
end
function tg input Callback(hObject, eventdata, handles)
% hObject handle to tg input (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hints: get(hObject, 'String') returns contents of tg_input as text
       str2double(get(hObject,'String')) returns contents of tg_input as a double
%
tg_input = str2double(get(hObject,'String'));
if isnan(tg_input)
```

```
set(hObject,'String',0);
  errordlg('Input must be a Number','Error!');
end
data = getappdata(gcbf,'user_data');
data.tg input = tg input;
setappdata(gcbf,'user_data',data);
% --- Executes during object creation, after setting all properties.
function rg input CreateFcn(hObject, eventdata, handles)
% hObject handle to rg input (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
      See ISPC and COMPUTER.
%
if ispc
  set(hObject,'BackgroundColor','white');
else
  set(hObject, 'BackgroundColor',get(0, 'defaultUicontrolBackgroundColor'));
end
function rg input Callback(hObject, eventdata, handles)
% hObject handle to rg_input (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hints: get(hObject,'String') returns contents of rg_input as text
%
       str2double(get(hObject, 'String')) returns contents of rg_input as a double
rg_input = str2double(get(hObject, 'String'));
if isnan(rg input)
  set(hObject,'String',0);
  errordlg('Input must be a Number','Error!');
end
data = getappdata(gcbf,'user_data');
data.rg_input = rg_input;
setappdata(gcbf,'user_data',data);
% --- Executes during object creation, after setting all properties.
function l input CreateFcn(hObject, eventdata, handles)
% hObject handle to 1 input (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called
% Hint: edit controls usually have a white background on Windows.
      See ISPC and COMPUTER.
%
if ispc
  set(hObject,'BackgroundColor','white');
else
```

```
set (hObject, 'BackgroundColor', get (0, 'defaultUicontrolBackgroundColor')); \\ end
```

```
function l_input_Callback(hObject, eventdata, handles)
% hObject handle to 1 input (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Hints: get(hObject, 'String') returns contents of l_input as text
%
       str2double(get(hObject, 'String')) returns contents of 1 input as a double
l_input = str2double(get(hObject,'String'));
if isnan(l_input)
  set(hObject,'String',0);
  errordlg('Input must be a Number', 'Error!');
end
data = getappdata(gcbf,'user_data');
data.l input = l input;
setappdata(gcbf,'user_data',data);
% --- Executes on button press in plot button.
function plot_button_Callback(hObject, eventdata, handles)
% hObject handle to plot_button (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% Change inputs from strings to double values
d = eval(get(handles.d_input,'String'));
f = str2double(get(handles.f input, 'String')); % F1
f1 = str2double(get(handles.f1_input,'String')); % F2
P = str2double(get(handles.p_input,'String'));
tg_db = str2double(get(handles.tg_input,'String'));
rg_db = str2double(get(handles.rg_input,'String'));
L = str2double(get(handles.l_input,'String'));
% Wait while the processing takes place
h = waitbar(0, 'Please Wait...');
for i = 1:800, % Computation being done here
waitbar(i/100)
end
close(h)
```

% Change gains from dB to watts

 $tg = (10^{(tg_db/10)});$  $rg = (10^{(rg_db/10)});$
M1 = (300/f); % M1 is the wavelength of F1 M2 = (300/f1); % M2 is wavelength of F2

$$\begin{split} \text{N1} &= (\text{P*tg*rg*}(\text{M1.}^2)); \ \% \ \text{Numerator of the FRIIS equation for F1} \\ \text{N2} &= (\text{P*tg*rg*}(\text{M2.}^2)); \ \% \ \text{Numerator of the FRIIS equation for F2} \\ \text{S} &= (1000000*(16*(d.^2)*(\text{pi.}^2)*\text{L})); \ \% \ \text{Denominator of the FRIIS equation} \\ \text{Pr1} &= (\text{N1.}/\text{S}); \ \% \ \text{Received signal in watts} \\ \text{Pr2} &= (\text{N2.}/\text{S}); \ \% \ \text{Received signal in watts} \\ \text{Pr_dB1} &= (10*(\log 10(\text{Pr1}))); \ \% \ \text{Converts the signal strength to dB for F1} \\ \text{Pr_dB2} &= (10*(\log 10(\text{Pr2}))); \ \% \ \text{Converts the signal strength to dB for F2} \end{split}$$

 $\begin{aligned} Q1 &= (32.44 + (20*\log 10(f)) + (20*\log 10(d)) - (10*\log 10(tg)) - (10*\log 10(rg))); \% \\ Computes the path loss in dB for F1 \\ Q2 &= (32.44 + (20*\log 10(f1)) + (20*\log 10(d)) - (10*\log 10(tg)) - (10*\log 10(rg))); \% \\ Computes the path loss in dB for F2 \end{aligned}$ 

% Plot the Curves - specfy axes

axes(handles.power\_axes) plot(d,Pr\_dB1,'b',d,Pr\_dB2,'r') xlabel('Distance(KM)') ylabel('Received Power(dB)') legend('F1 MHz','F2 MHz') title('Signal Strength At Any Distance') grid on

axes(handles.loss\_axes) plot(d,Q1,'b',d,Q2,'r') xlabel('Distance(KM)') ylabel('Path Loss(dB)') legend('F1 MHz','F2 MHz',4) title('Free Space Path Loss At Any Distance') grid on

% Calculation Section

D = handles.X\_INPUT;

val = get(handles.popupmenu1,'Value');

if val==1

PL = (32.44 + (20\*log10(f)) + (20\*log10(D))-(10\*log10(tg))-(10\*log10(rg)));% PL gives the path loss at a distance d

 $Y = (P*tg*rg*(M1.^2)); \%$  Numerator of the FRIIS equation  $Z = 100000*(16*(D.^2)*(pi.^2)*L); \%$  Denominator of the FRIIS equation, L==1 RP = 10\*(log10(Y./Z)); % Converts the received signal strength to dB

else

 $\label{eq:PL} \begin{array}{l} PL = (32.44 + (20*log10(f1)) + (20*log10(D)) - (10*log10(tg)) - (10*log10(rg))); \\ \% \ PL \ gives the path loss at a distance d \\ Y = (P*tg*rg*(M2.^2)); \ \% \ Numerator of the FRIIS equation \\ Z = 1000000*(16*(D.^2)*(pi.^2)*L); \ \% \ Denominator of the FRIIS equation, L==1 \\ RP = 10*(log10(Y./Z)); \ \% \ Converts the received signal strength to dB \end{array}$ 

end

ans1 = round(100\*RP)/100; ans2 = round(100\*PL)/100;

set(handles.power,'String',ans1);
set(handles.loss,'String',ans2);

% --- Executes on button press in reset\_button.
function reset\_button\_Callback(hObject, eventdata, handles)
% hObject handle to reset\_button (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

initialize\_gui(gcbf, handles);

function initialize\_gui(fig\_handle, handles)
data.f\_input = 900;
data.p\_input = 10.0;
data.tg\_input = 0.0;
data.rg\_input = 0.0;
data.l\_input = 1.0;
data.fl\_input = 1800;
setappdata(fig\_handle,'user\_data',data);

set(handles.f\_input,'String',data.f\_input); set(handles.p\_input,'String', data.p\_input); set(handles.tg\_input,'String',data.tg\_input); set(handles.rg\_input,'String',data.rg\_input); set(handles.l\_input,'String',data.l\_input); set(handles.fl\_input,'String',data.fl\_input);

% --- Executes on button press in close\_button.
function close\_button\_Callback(hObject, eventdata, handles)
% hObject handle to close\_button (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

```
selection = questdlg('Are You Sure You Want To Close This Window?',...
'Close Request Function',...
'Yes','No','Yes');
switch selection,
```

```
case 'Yes',
delete(gcf)
case 'No'
return
end
```

% --- Executes during object creation, after setting all properties. function X\_INPUT\_CreateFcn(hObject, eventdata, handles) % hObject handle to X INPUT (see GCBO) % eventdata reserved - to be defined in a future version of MATLAB % handles empty - handles not created until after all CreateFcns called % Hint: edit controls usually have a white background on Windows. See ISPC and COMPUTER. % if ispc set(hObject,'BackgroundColor','white'); else set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor')); end function X INPUT Callback(hObject, eventdata, handles) % hObject handle to X\_INPUT (see GCBO) % eventdata reserved - to be defined in a future version of MATLAB % handles structure with handles and user data (see GUIDATA) % Hints: get(hObject,'String') returns contents of X\_INPUT as text % str2double(get(hObject, 'String')) returns contents of X\_INPUT as a double NewStrVal = get(hObject, 'String'); NewVal = str2double(NewStrVal); handles.X INPUT = NewVal; guidata(hObject,handles); % --- Executes during object creation, after setting all properties. function f1\_input\_CreateFcn(hObject, eventdata, handles) % hObject handle to f1\_input (see GCBO) % eventdata reserved - to be defined in a future version of MATLAB empty - handles not created until after all CreateFcns called % handles % Hint: edit controls usually have a white background on Windows. % See ISPC and COMPUTER. if ispc set(hObject,'BackgroundColor','white'); else set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor')); end function f1\_input\_Callback(hObject, eventdata, handles)

```
% hObject handle to f1_input (see GCBO)
```

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject, 'String') returns contents of f1\_input as text

% str2double(get(hObject,'String')) returns contents of f1\_input as a double

```
f1_input = str2double(get(hObject,'String'));
if isnan(f1_input)
    set(hObject,'String',0);
    errordlg('Input must be a Number','Error!');
end
```

```
data = getappdata(gcbf,'user_data');
data.f1_input = f1_input;
setappdata(gcbf,'user_data',data);
```

% --- Executes during object creation, after setting all properties.
function popupmenu1\_CreateFcn(hObject, eventdata, handles)
% hObject handle to popupmenu1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: popupmenu controls usually have a white background on Windows.% See ISPC and COMPUTER.

if ispc

```
set(hObject,'BackgroundColor','white');
```

else

```
set(hObject,'BackgroundColor',get(0,'defaultUicontrolBackgroundColor'));
end
```

% --- Executes on selection change in popupmenu1.

function popupmenu1\_Callback(hObject, eventdata, handles)

% hObject handle to popupmenu1 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% Hints: contents = get(hObject,'String') returns popupmenu1 contents as cell array % contents{get(hObject,'Value')} returns selected item from popupmenu1

```
val = get(handles.popupmenu1,'Value');
```

% -----

function file\_Callback(hObject, eventdata, handles)

% hObject handle to file (see GCBO)

% event data reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% ------

function Untitled\_4\_Callback(hObject, eventdata, handles)
% hObject handle to Untitled\_4 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB % handles structure with handles and user data (see GUIDATA)

% -----function userguide Callback(hObject, eventdata, handles) % hObject handle to userguide (see GCBO) % eventdata reserved - to be defined in a future version of MATLAB % handles structure with handles and user data (see GUIDATA) % -----function about\_Callback(hObject, eventdata, handles) % hObject handle to about (see GCBO) % eventdata reserved - to be defined in a future version of MATLAB % handles structure with handles and user data (see GUIDATA) % -----function Untitled 7 Callback(hObject, eventdata, handles) % hObject handle to Untitled 7 (see GCBO) % eventdata reserved - to be defined in a future version of MATLAB % handles structure with handles and user data (see GUIDATA) % -----function space1\_Callback(hObject, eventdata, handles) % hObject handle to space1 (see GCBO) % eventdata reserved - to be defined in a future version of MATLAB % handles structure with handles and user data (see GUIDATA) % -----function space2 Callback(hObject, eventdata, handles) % hObject handle to space2 (see GCBO) % eventdata reserved - to be defined in a future version of MATLAB % handles structure with handles and user data (see GUIDATA) % ----function space3\_Callback(hObject, eventdata, handles) % hObject handle to space3 (see GCBO) % eventdata reserved - to be defined in a future version of MATLAB % handles structure with handles and user data (see GUIDATA) % -----function space4 Callback(hObject, eventdata, handles) % hObject handle to space4 (see GCBO) % eventdata reserved - to be defined in a future version of MATLAB % handles structure with handles and user data (see GUIDATA) % -----function open\_Callback(hObject, eventdata, handles) % hObject handle to open (see GCBO) % eventdata reserved - to be defined in a future version of MATLAB % -----

function new\_Callback(hObject, eventdata, handles)

- % hObject handle to new (see GCBO)
- % eventdata reserved to be defined in a future version of MATLAB
- % handles structure with handles and user data (see GUIDATA)