

Graduate Theses, Dissertations, and Problem Reports

2007

Capacity -based parameter optimization of bandwidth constrained CPM

Rohit Iyer Seshadri West Virginia University

Follow this and additional works at: https://researchrepository.wvu.edu/etd

Recommended Citation

Seshadri, Rohit Iyer, "Capacity -based parameter optimization of bandwidth constrained CPM" (2007). *Graduate Theses, Dissertations, and Problem Reports.* 2593. https://researchrepository.wvu.edu/etd/2593

This Dissertation is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Dissertation in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Dissertation has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact researchrepository@mail.wvu.edu.

Capacity-Based Parameter Optimization of Bandwidth Constrained CPM

by

Rohit Iyer Seshadri

Dissertation submitted to the College of Engineering and Mineral Resources at West Virginia University in partial fulfillment of the requirements for the degree of

> Doctor of Philosophy in Electrical Engineering

Erdogan Gunel, Ph.D. Daryl Reynolds, Ph.D. Natalia Schmid, Ph.D. Brian Woerner, Ph.D. Matthew C. Valenti, Ph.D., Chair

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia 2007

Keywords: Continuous phase modulation, BICM, capacity, differential phase detection

Copyright 2007 Rohit Iyer Seshadri

Abstract

Capacity-Based Parameter Optimization of Bandwidth Constrained CPM

by

Rohit Iyer Seshadri Doctor of Philosophy in Electrical Engineering

West Virginia University

Matthew C. Valenti, Ph.D., Chair

Continuous phase modulation (CPM) is an attractive modulation choice for bandwidth limited systems due to its small side lobes, fast spectral decay and the ability to be noncoherently detected. Furthermore, the constant envelope property of CPM permits highly power efficient amplification. The design of bit-interleaved coded continuous phase modulation is characterized by the code rate, modulation order, modulation index, and pulse shape. This dissertation outlines a methodology for determining the optimal values of these parameters under bandwidth and receiver complexity constraints. The cost function used to drive the optimization is the information-theoretic minimum ratio of energy-per-bit to noise-spectral density found by evaluating the constrained channel capacity. The capacity can be reliably estimated using Monte Carlo integration. A search for optimal parameters is conducted over a range of coded CPM parameters, bandwidth efficiencies, and channels. Results are presented for a system employing a trellis-based coherent detector. To constrain complexity and allow any modulation index to be considered, a soft output differential phase detector has also been developed.

Building upon the capacity results, extrinsic information transfer (EXIT) charts are used to analyze a system that iterates between demodulation and decoding. Convergence thresholds are determined for the iterative system for different outer convolutional codes, alphabet sizes, modulation indices and constellation mappings. These are used to identify the code and modulation parameters with the best energy efficiency at different spectral efficiencies for the AWGN channel. Finally, bit error rate curves are presented to corroborate the capacity and EXIT chart designs.

Acknowledgements

I would first like to thank the professors on my committee for their invaluable assistance and penetrating insight into my research. I have been fortunate to take courses under all of the committee members and their lectures have been instrumental in shaping my understanding of the subject.

I would like to thank my advisor and committee chair Dr.Valenti for inviting me to join his research group and greatly appreciate the many opportunities he has made available to me. Dr.Valenti has been a terrific mentor and I have benefited tremendously from our interaction. I hope that our relationship has not been without some degree of reciprocal utility.

Next, I would also like to thank the students of the wireless communications research laboratory (WCRL) for many a stimulating discussion. I would especially like to thank my colleague Shi Cheng, who has provided valuable feedback on my research and has helped verify some of the results presented in Chapter 6. Finally, on a personal note, I will always remain indebted to my family for their unwavering encouragement and support.

Contents

A	cknov	vledgements		iii
Li	st of	Figures		vii
Li	st of	Tables		xiii
N	Notation			xiv
1	Intr	oduction, Problem Statement and Methodology		1
	1.1	Introduction		1
		1.1.1 Building Blocks in a Digital Communication System		1
		1.1.2 Channel Capacity		2
		1.1.3 Error Correction Codes		3
		1.1.4 The Coding Paradox and Capacity Approaching Codes	• •	6
		1.1.5 Modulation \ldots		7
	1.2	Problem Statement	• •	9
	1.3	Methodology		12
2	Cha	nnel Coding for Modern Communication Systems		15
	2.1	Convolutional Codes		16
		2.1.1 Encoding Convolutional Codes		16
		2.1.2 State Diagram and Trellis		18
		2.1.3 Decoding Convolutional Codes		20
		2.1.4 Code Puncturing		24
	2.2	Low-Density Parity-Check Codes		25
		2.2.1 Review of Linear Block Codes		25
		2.2.2 Decoding LDPC Codes		26
		2.2.3 Constructing LDPC Codes		30
		2.2.4 Encoding LDPC Codes		32
	2.3	Turbo Codes		33
		2.3.1 Encoding Turbo Codes		34
		2.3.2 Decoding Turbo Codes		36
		2.3.3 Turbo codes as a type of LDPC codes		36
	2.4	Chapter Summary		38

3	Mo	dulation for Bandwidth Limited Systems	40
	3.1	Binary and Quadrature Phase Shift Keying	41
	3.2	Minimum Shift Keying	43
	3.3	Continuous Phase Modulation	47
	3.4	Tilted Phase Representation of CPM	51
	3.5	Bandwidth of CPM Signals	53
	3.6	Bandwidth of Coded CPM Signals	56
	3.7	Chapter Summary	57
4	Det	ector Designs For CPM	59
	4.1	System Model	60
		4.1.1 Transmitter	60
		4.1.2 Channel	60
		4.1.3 Receiver \ldots	61
	4.2	Coherent Detection of CPM	62
		4.2.1 Coherent Front-End	62
		4.2.2 Soft-Output Coherent Detector	63
		4.2.3 Bit Error Rates for Uncoded Systems with Coherent Detection	64
	4.3	Differential Phase Detection of CPM	65
		4.3.1 Differential Front-End	67
		4.3.2 Hard-Decision Differential Phase Detector	69
		4.3.3 Soft-Output, Soft-Decision, Differential Phase Detector	69
		4.3.4 Bit Error Rates for Uncoded Systems with Differential Detection	74
		4.3.5 Applying SO-SDDPD to Bluetooth	77
	4.4	Coherent Detection versus Differential Detection	84
	4.5	Chapter Summary	86
5	Mu	tual Information as a Performance Measure	87
	5.1	The Unconstrained Shannon Capacity	88
	5.2	Modulation Constrained Capacity	90
		5.2.1 Capacity of Coded Modulation	90
		5.2.2 Capacity of Bit-Interleaved Coded Modulation	92
		5.2.3 A Computationally Feasible Method for Evaluating Capacity	95
		5.2.4 Capacity Results	96
	5.3	Extrinsic Information Transfer Charts	99
		5.3.1 Detector Transfer Characteristics	102
		5.3.2 Decoder Transfer Characteristics	103
	5.4	Chapter Summary	106
6	Opt	imization Results for Bit-Interleaved Coded CPM	108
	6.1	Capacity-Based Parameter Optimization of BICCPM with Coherent Detection	n 112
		6.1.1 Design of CPFSK under Bandwidth Constraints	112
	6 9	6.1.2 Design of Partial Response CPM under Bandwidth Constraints	116
	0.2	tion	118

		6.2.1	Information Rate Loss Relative to Coherent Detector	119
		6.2.2	Design of Partial Response CPM under Bandwidth Constraints	120
	6.3	Valida	ting Design Parameters using BER Simulations	124
	6.4	Design	of Coded CPM with Iterative Detection and Decoding	126
		6.4.1	Design of BICCPFSK-ID with Coherent Detection	129
		6.4.2	Design of BICCPM-ID with Differential Detection	130
	6.5	Chapt	er Summary	132
7	Con	clusio	a	135
	7.1	Summ	ary	135
	7.2	Ideas f	for Future Research	137
		7.2.1	Optimization in Non-Ergodic Channels	137
		7.2.2	Effect of Finite Codeword Lengths on Capacity	139
		7.2.3	Optimization in the Face of Uncertainty	141
	А	Review	v of Relevant Information Theoretic Concepts	143
		A.1	Entropy of Random Variables	143
		A.2	Mutual Information	144
		A.3	Data Processing Inequality	144
		A.4	Capacity of Independent Parallel Channels	145
	В	MATL	AB Module for Computing CM Capacity using Monte Carlo Integration	n 145
	С	MATL	AB Modules for Generating Detector Transfer Characteristics	147
R	efere	nces		152

Contribution

160

List of Figures

$1.1 \\ 1.2 \\ 1.3 \\ 1.4$	Simplified block diagram of a digital communication system. \dots The Shannon capacity in bps/Hz as a function of the average SNR. \dots The Shannon capacity in bps as a function of the bandwidth. \dots BER comparison of uncoded MSK with coded CPFSK with parameters M =2 and h =1/8. The coded system uses an outer rate 1/2 CDMA 2000 turbo code. Coherent CPM detection and AWGN channel are assumed. \dots	2 3 4 11
2.1	A rate $1/2$, constraint length 3 nonrecursive convolutional encoder with octal generators [7, 5].	17
2.2	A rate 1/2, constraint length 3 recursive convolutional encoder with feedback, feedforward octal generators [7, 3] respectively.	18
2.3	State diagram for a rate $1/2$, constraint length 3 nonrecursive convolutional code with generators $[7, 5]$	19
2.4	Trellis diagram for a rate $1/2$, constraint length 3 nonrecursive convolutional code with generators [7, 5] respectively. Dashed arrows denote branches cor-	
	responding to an input symbol 0	19
2.5	The Tanner graph of a $(7, 4)$ Hamming code	27
2.6	A length four cycle in the Tanner graph of a $(7, 4)$ Hamming code. The edges correspond to the italicized entries in (2.28) .	27
2.7	Encoder for the Ex-IRA type code.	33
2.8	Frame error rate performance of the $n = 64800$ (normal frame) DVB-S2 LDPC code. The decoder used 100 iterations of the sum-product algorithm in the	
	log-domain.	34
2.9	Frame error rate performance of the $n = 16200$ (short frame) DVB-S2 LDPC code. The decoder used 100 iterations of the sum-product algorithm in the	
	log-domain.	35
2.10	Bit error rate performance of the $n = 2304$ WiMAX LDPC code. The decoder	
	used 100 iterations of the sum-product algorithm in the log-domain	35
2.11	Frame error rate performance of the $n = 2304$ WiMAX LDPC code. The decoder used 100 iterations of the sum-product algorithm in the log-domain.	36
2.12	A turbo encoder.	37
2.13	A turbo decoder.	37

3.1	Binary phase shift keying (BPSK) modulation. $x(t)$ is the BPSK waveform obtained by modulating the carrier $\cos(2\pi f_c t)$ with the symbol sequence a .	41
3.2	Quadrature phase shift keying (QPSK) modulation. $x(t)$ is the QPSK wave- form obtained by modulating $\cos(2\pi f t + \pi)$ and $\sin(2\pi f t + \pi)$ with \mathbf{a}_{t} and	
	ion obtained by modulating $\cos(2\pi f_c t + 4)$ and $\sin(2\pi f_c t + 4)$ with a_f and a_f	/3
33	Constellation diagram for OPSK modulation. Dotted arrows denote possible	40
0.0	transitions from $(1, 1)$ for non-offset QPSK which can occur every T seconds. Dashed arrow denotes possible transition from $(1, 1)$ for OQPSK which can	
	occur every T_b seconds.	44
3.4	OQPSK modulation. $x(t)$ is the QPSK waveform obtained by modulating $\cos\left(2\pi f_c t + \frac{\pi}{4}\right)$ and $\sin\left(2\pi f_c t + \frac{\pi}{4}\right)$ with \mathbf{a}_I and \mathbf{a}_Q respectively. \mathbf{a}_I and \mathbf{a}_Q	
0 5	have an offset of T_b	44
3.5	Minimum shift keying modulation. The MSK waveform $x(t)$ is generated by modulating the I and Q carriers with the pulse shaped even and odd data	10
2.0	streams respectively.	46
3.6	Phase trajectory for MSK.	46
3.7	Time variant phase trellis for MSK.	47
3.8	Power spectral densities in dB of BPSK, QPSK and MSK	48
3.9	Rectangular pulse shape $g(t)$ and its integral $q(t)$.	49
3.10	Raised cosine pulse shape $g(t)$ and its integral $q(t)$	50
3.11 2.10	Gaussian pulse snape $g(t)$ and its integral $q(t)$	50 50
3.12 3.13	CPM as a continuous phase encoder (CPE) followed by a memoryless modu-	52
014	lator (MM). \dots	53
3.14	Fractional out-of-band powers in dB for BPSK, QPSK and MSK versus nor- malized frequency fT_b .	54
3.15	Fractional out-of-band power for binary CPFSK at different values of h	54
3.16	Normalized 99% power bandwidth as a function of h , for different CPFSK alphabet sizes.	55
3.17	Fractional out-of-band power for quaternary CPM with BC pulse shaping at	00
0	different values of L .	55
3.18	Fractional out-of-band power for uncoded MSK and coded MSK with $r = 1/2$.	56
3.19	Fractional out-of-band power for uncoded MSK, coded CPM with $r = 1/2$.	
	The uncoded and coded signals have the same 99% bandwidth	57
4.1	Transmitter block diagram.	60
4.2	Receiver block diagram.	62
4.3	Receiver with soft-output coherent detection	62
4.4	BER for uncoded 2-CPFSK with different h and coherent detection in AWGN.	65
4.5	BER for comparison between MSK and 3 RC CPM with $h = 1/2$ and coherent	
	detection in AWGN.	66
4.6	Receiver with hard-decision differential phase detection	67
4.7	Receiver with soft-output, soft-decision differential phase detection.	70
4.8	Minimum separation between differential phase angles for symbols ± 1 (D_{min}) in degrees for binary CPFSK as a function of h .	76

4.9	BER curves for uncoded binary CPFSK with hard decision, symbol-by-symbol	
4 10	DPD in AWGN at select values of h	((
4.10	symbol-by-symbol DPD in AWGN at select values of h	78
4.11	BER comparison between the DPD and SO-SDDPD for uncoded 2-GFSK	.0
	with $B_aT = 0.5$ at different h. The SDDPD uses $R = 40$ uniform phase	
	regions. The channel is AWGN.	78
4.12	BER comparison between the DPD and SO-SDDPD for uncoded 2-GFSK	
	with $h = 0.5$ and $B_q T = 0.25$. The SO-SDDPD uses $R = 26$ uniform phase	
	regions. Three different K-factors are considered $K = -\infty$ dB (Rayleigh),	
	$K = 6$ dB and $K = \infty$ dB (AWGN). The maximum Doppler frequency is	
	assumed to be $f_d = 0$	79
4.13	Dotted curve is the BICM capacity in Rician channel with $K = 2$ dB, using	
	SISO-SDDPD. Six simulated points are shown for DM1 packets, representing	
	minimum \mathcal{E}_b/N_0 (dB) to achieve BER = 10 ⁻⁴ , from top to bottom: (1) LDI-	
	HDD (2) LDI-HDD with bit-interleaving (3) SDDPD-HDD (4) SDDPD-HDD	
	with bit-interleaving (5) SO-SDDPD-SDD (6) BICM receiver. All SDDPD	
	systems use $R = 24$ uniform phase regions. Modulation index $h = 0.315$ is	00
111	assumed	80
4.14	PER for DM1 packet types in Richan channel with $K = 2$ dB. All SDDPD systems use $P = 24$ uniform phase regions. Detted surves indicate systems	
	systems use $N = 24$ uniform phase regions. Dotted curves indicate systems with bit interleaving. Modulation index $h = 0.315$ is assumed	82
4 15	Throughput for DM1 DM3 and DM5 packet types in Rician channel with	02
1.10	K = 2 dB SDDPD systems use $B = 24$ uniform phase regions. Dotted curves	
	indicate systems with bit-interleaving. Modulation index $h = 0.315$ is assumed.	83
4.16	BER comparison between the DPD and the sequence based coherent detector	
-	for uncoded binary-CPFSK at different h in AWGN	85
4.17	BER comparison between the SO-SDDPD and the sequence based coherent	
	detector for uncoded binary 3-RC CPM in AWGN. The SDDPD used $R = 40$	
	uniform phase regions.	85
۳ 1		01
0.1 5 0	Block diagram of system with hit interleaved added modulation	91
0.2 5.3	BPSK capacity as a function of the number of simulated symbols per SNR	95
$5.0 \\ 5.4$	Simulation time as a function of the number of simulated symbols per SNR.	30
0.4	for computing the BPSK capacity	97
5.5	The unconstrained Shannon capacity for 2 dimensional signal sets and and	01
0.0	CM capacity for different linear modulations as a function of \mathcal{E}_b/N_0 . The	
	CM capacities were calculated using Monte-Carlo integration with 2 million	
	symbols generated per SNR. The modulation signal set is assumed to have	
	equally likely symbols.	98
5.6	The BICM capacity of 16-PSK with natural and gray labelling in AWGN. Also	
	shown is the CM capacity. Monte-Carlo integration with 2 million symbols	
	generated per \mathcal{E}_s/N_0 was used to compute the capacity	98

5.7	The BICM capacity of 16-QAM with set partitioning and gray labelling in AWGN. Also shown is the CM capacity. Monte-Carlo integration with 2	
58	million symbols generated per \mathcal{E}_s/N_0 was used to compute the capacity Average time required per SNP in seconds to compute the capacity and bit	99
5.8	error rate by Monte Carlo trials. Simulations were performed using 5 million	
	modulated symbols per SNR point.	100
5.9	Block diagram of system with bit-interleaved coded modulation with iterative	
	decoding.	101
5.10	BER after 20 BICM-ID iteration for 4-CPFSK with $h = 1/3$ and rate $1/2$ NRC code with octal generators [7, 5]. Two different bit-to-symbol mappings are considered. 4-CPFSK is coherently detected. The channel is AWGN and	
	interleaver size is 100000 bits	102
5.11	Mutual information of Gaussian distributed <i>a priori</i> information as a function of the variance.	103
5.12	Detector mutual information transfer characteristics for 16-QAM modulation in an AWGN channel, at different \mathcal{E}_s/N_0 and bit-to-symbol mapping	104
5.13	Decoder mutual information transfer characteristics for rate $1/2$, non-recursive convolutional codes with constraint lengths $K = 2, 3$ and 5.	105
5.14	Decoder mutual information transfer characteristics for rate $1/2$, NRC and BSC codes	105
5.15	Extrinsic information transfer chart for 4-CPFSK a rate 1/2 NRC code with	100
	octal generators $[7, 5]$.	106
6.1	Capacity in bits per channel use for <i>M</i> -ary BICCPFSK with $h = 3/4, 1/2, 1/5$, and $1/10$ and coherent detection, in AWGN	110
6.2	Information-theoretic \mathcal{E}_b/N_0 versus code rate for binary CPFSK with $h = 3/4$, $1/2, 1/5$, and $1/10$ with coherent detection, in AWGN.	110
6.3	Minimum allowable code rate as a function of h at $\eta = 3/4$ bps/Hz for M -ary CPFSK	113
6.4	Minimum \mathcal{E}_b/N_0 for reliable signaling required by binary BICCPFSK with coherent detection as a function of h , at different η , in AWGN	114
6.5	Minimum \mathcal{E}_b/N_0 for reliable signaling required by $M = 4$ BICCPFSK with	
0.0	coherent detection as a function of h , at different η , in AWGN	114
0.0	coherent detection as a function of h , at different η , in AWGN	114 115
6.6 6.7	coherent detection as a function of h , at different η , in AWGN Minimum \mathcal{E}_b/N_0 for reliable signaling required by $M = 8$ BICCPFSK with coherent detection as a function of h , at different η , in AWGN Minimum \mathcal{E}_b/N_0 for reliable signaling required by $M = 16$ BICCPFSK with coherent detection as a function of h at different η in AWGN.	114 115
6.66.76.8	coherent detection as a function of h , at different η , in AWGN Minimum \mathcal{E}_b/N_0 for reliable signaling required by $M = 8$ BICCPFSK with coherent detection as a function of h , at different η , in AWGN Minimum \mathcal{E}_b/N_0 for reliable signaling required by $M = 16$ BICCPFSK with coherent detection as a function of h , at different η , in AWGN Effect of bit-to-symbol mapping on the energy efficiency of $M = 4$ BICCPFSK with coherent detection	114115115116
6.66.76.86.9	coherent detection as a function of h , at different η , in AWGN Minimum \mathcal{E}_b/N_0 for reliable signaling required by $M = 8$ BICCPFSK with coherent detection as a function of h , at different η , in AWGN Minimum \mathcal{E}_b/N_0 for reliable signaling required by $M = 16$ BICCPFSK with coherent detection as a function of h , at different η , in AWGN	 114 115 115 116
 6.6 6.7 6.8 6.9 6.10 	coherent detection as a function of h , at different η , in AWGN Minimum \mathcal{E}_b/N_0 for reliable signaling required by $M = 8$ BICCPFSK with coherent detection as a function of h , at different η , in AWGN Minimum \mathcal{E}_b/N_0 for reliable signaling required by $M = 16$ BICCPFSK with coherent detection as a function of h , at different η , in AWGN Effect of bit-to-symbol mapping on the energy efficiency of $M = 4$ BICCPFSK with coherent detection	 114 115 115 116 117
 6.6 6.7 6.8 6.9 6.10 6.11 	coherent detection as a function of h , at different η , in AWGN	 114 115 115 116 117 117

х

6.12	Minimum \mathcal{E}_b/N_0 required at different spectral efficiencies for coherently de-	
	tected BICCPM, in AWGN. Binary partial response signaling using 3 RC and	
	3 REC pulse shapes give significant improvement in the energy efficiency over	
	binary CPFSK	119
6.13	Information-theoretic minimum \mathcal{E}_b/N_0 as a function of h at different spectral	
	efficiencies with coherent and differential detection. The channel is AWGN.	
	The modulation is binary CPFSK.	120
6.14	Minimum required \mathcal{E}_b/N_0 as a function of normalized bandwidth B_{coded} for	
	BICCPM in Rician fading ($K = 6$ dB). The code rate is $r = 2/3$, GFSK mod-	
	ulation used with $M \in \{2, 4\}$, and SO-SDDPD. The numbers denote modula-	
	tion indices corresponding to GFSK parameters with the lowest information-	
	theoretic limit on \mathcal{E}_b/N_0 at different B_{coded} .	122
6.15	Minimum required \mathcal{E}_b/N_0 as a function of code rate for BICCPM with GFSK	
	signaling in Rayleigh fading with SO-SDDPD and under bandwidth constraint	
	$B_{coded} = 0.8$. The legend specifies the GFSK parameters $(M, h, B_g T)$ that	
	achieve this minimum. Under the given constraints, the design $\{r = 3/4, M =$	
	$4, h = 0.25, B_g T = 0.5$ has the best energy efficiency	123
6.16	Bit error rate in AWGN for bit-interleaved coded, 2-CPFSK with $h = 1/10$,	
	1/7 and $1/2$ using a rate $1/2$ CDMA 2000 code after 10 turbo decoder it-	
	erations. The vertical lines denote the information theoretic \mathcal{E}_b/N_0 in dB	
	to achieve an arbitrarily low BER for the respective h and $r = 1/2$. The	
	interleaver size is 24576 bits	125
6.17	BER of coded (solid line) and uncoded (dotted line) GFSK in Rayleigh fad-	
	ing under bandwidth constraint $B_{coded} = 0.9$ using SO-SDDPD. The coded	
	(BICCPM) system system uses a rate $r = 2/3$, length $N_b = 6720$ turbo code,	
	16 decoder iterations, $R = 26$ phase regions, and GFSK parameters $M = 4$,	
	$h = 0.24$, and $B_g T = 0.5$. The uncoded system uses $R = 40$ phase regions	
6 1 0	and GFSK parameters $M = 2$, $h = 0.5$ and $B_gT = 0.3$	126
6.18	BER of coded (solid line) and uncoded (dotted line) GFSK in Rician fading	
	$(K = 6 \text{ dB})$ under bandwidth constraint $B_{coded} = 0.9$ using SO-SDDPD. The	
	coded (BICCPM) system system uses a rate $r = 3/4$, length $N_b = 6720$ turbo	
	code, 16 decoder iterations, $R = 26$ phase regions, and GFSK parameters	
	$M = 4, h = 0.285$, and $B_gT = 0.5$. The uncoded system uses $R = 40$ phase	107
C 10	regions and GFSK parameters $M = 2$, $h = 0.5$ and $B_g T = 0.3$.	127
6.19	Minimum \mathcal{E}_b/N_0 in dB required for reliable signaling as a function of h for	
	concreting detected CPFSK at $\eta = 1/2$ bps/Hz. The channel is AWGN.	
	I ne dotted curves denote BICCPFSK convergence thresholds found from the	
	thresholds with a NDC [7, 5] conversional code musticed by EVIT short	
	thresholds with a NRC $[7, 5]$ convolutional code, predicted by EAT chart analyzig. The alphabet gizes are $M \in \{2, 4, 16\}$ with natural bit to symbol	
	analysis. The alphabet sizes are $M \in \{2, 4, 10\}$ with natural bit-to-symbol	190
6 20	EXIT surves for 4 CDESK with $h = 1/2$ gray labelling and various outer	120
0.20	EXECUTED IN THE CONTROL WITH $n = 1/2$, gray fabelling and values of the convolutional codes. The code rate is $n = 0.64$. The channel is AWCN. The	
	for the particular r the NRC [7–5] code violds the lowest	
	inguite indicates that for the particular i , the Nite $[i, j]$ code yields the lowest convergence threshold	130
		100

6.216.22	BER after 25 iterations for 4-CPFSK with $h = 3/7$, natural labelling and two outer convolutional codes. The code rate is $r = 0.6$. The channel is AWGN. The vertical lines indicate convergence thresholds, predicted using EXIT charts	131
	K = 2 dB, $R = 24$ uniform phase regions. Note that the decoder's EXT curve intersects $(0.5, R')$, where $R' = 10/15$.	133
7.1	Information outage probability with code combining in block fading at $F = 1$ and $F = 100$ for BICCPM using SO-SDDPD. The combination of code rates and GFSK parameters are selected such that $B = 0.9$	138
7.2	GFSK parameters with the lowest information theoretic minimum \mathcal{E}_b/N_0 (dB) for various code rates at $B = 0.9$ in AWGN and Rayleigh channels for BIC-	
	CPM with SO-SDDPD.	139
7.3	BER curves for coherently detected MSK using a CDMA 2000 turbo code, in	
	AWGN	140
7.4	Minimum \mathcal{E}_b/N_0 required for $P_e = 10^{-4}$ at blocklengths $N_b = 1024$ and 16384	
	for 16-QAM in AWGN. Also shown is the CM capacity of 16-QAM	141

List of Tables

4.1	θ_i in degrees at different $B_g T$ for GFSK with $h = 0.5$.	71
4.2	θ_i in degrees at different L for RC pulse shape with $h = 0.5.$	71
4.3	Differential phase angles in radians for binary CPFSK at different $h. \ldots$	75
4.4	Differential phase angles in radians for binary GFSK with $B_gT = 0.5$, $h = 0.5$.	75
6.1	Combination of code rates and CPM parameters with lowest information the- oretic minimum \mathcal{E}_b/N_0 under the constraint of using SO-SDDPD in Rayleigh fading at different B_{coded} .	123
6.2	Combination of code rates and CPM parameters having lowest information theoretic minimum \mathcal{E}_b/N_0 under the constraint of using SO-SDDPD in Rician fading $(K = 6 \ dB)$ at different $B_{-1,1}$	194
6.3	Information theoretic minimum \mathcal{E}_b/N_0 in dB for non-iterative BICM in AWGN at different η .	131
6.4	Minimum \mathcal{E}_b/N_0 in dB for BICM-ID in AWGN with outer convolutional codes	199
	at different η	197

Notation

We use the following notation and symbols throughout this dissertation.

- $(\cdot)^H$: Complex conjugate transpose
- $(\cdot)^*$: Complex conjugate
- $E[\cdot]$: Expectation operator
- p(X) : Probability density function (pdf) of a random variable X
- $\|\cdot\|$: Euclidian norm
- $\Re\{\cdot\}$: Real part of the argument
- $\Im\{\cdot\}$: Imaginary part of the argument
- $|\cdot|$: Cardinality of a set

Bold upper case letters denote matrices and bold lower case letters denote vectors.

Chapter 1

Introduction, Problem Statement and Methodology

1.1 Introduction

Over the last few decades, digital communication systems have become ubiquitous. Unlike analog systems, which have infinite variations on the information carrying signal, in digital systems the number of possible signals is finite. This has many advantages, the most obvious being that digital signals are easier to reconstruct since the receiver simply has to select from a finite number of hypothesis. Furthermore, signal processing techniques (error correction, interleaving, spreading, equalization, etc.) make it possible to signal at extremely low error rates even in harsh environments.

1.1.1 Building Blocks in a Digital Communication System

An elementary block diagram of a digital communication system is shown in Fig. 1.1. The source (either analog or discrete) generates information-bearing messages which have to recovered with some reliability at the sink. If the source message is analog, the source encoder digitizes it by first sampling and then quantizing using the minimum number of bits required to meet a distortion measure. Digital messages may be compressed by the source encoder using a compression algorithm such as Huffman coding or the Lempel-Ziv algorithm. The channel encoder adds controlled redundancy to the source encoder's output. The modulator



Figure 1.1: Simplified block diagram of a digital communication system.

maps the coded symbols onto analog waveforms taken from a finite set which meet certain requirements imposed by the channel. The modulated waveform is transmitted through the channel which distorts the signal. The demodulator generates estimates of the modulated symbols, which are then fed to the channel decoder. The redundancy introduced by the channel code is exploited by the decoder to generate estimates of message bits that minimize the error rate. The source decoder reconstructs the message, which is then delivered to the sink. It must be mentioned that the above description is very simplistic. Modern digital communication systems are in fact extremely sophisticated and it is common practice to have the demodulator and channel decoder, source and channel decoder work jointly to recover the message (a process which involves an iterative exchange of the probabilistic estimates of the transmitted message).

1.1.2 Channel Capacity

In his ground breaking work [1], Shannon derived the theoretical limits on reliable communication. He introduced the concept of *channel capacity*, which is the maximum rate at which information can be transmitted reliably through a noisy channel.

Let X, Y be random variables denoting the input and output of a channel with bandwidth W. The input is assumed to be corrupted by zero mean, additive white Gaussian noise (AWGN) with variance $\frac{N_0}{2}$. The channel capacity is as derived by Shannon is given by

$$C = W \log\left(1 + \frac{P}{WN_0}\right),\tag{1.1}$$

3



Figure 1.2: The Shannon capacity in bps/Hz as a function of the average SNR.

where P is the average signal power. Shannon postulated that as long as the data rate (i.e the rate at which the source generates information bits) $r_b < C$, there exist channel codes that allow signaling with arbitrarily low bit error rates. When the logarithm has base 2, the capacity (1.1) has units of bits per second (bps) and when the logarithm has base e, the capacity is in nats per second.

Fig. 1.2 shows the the normalized capacity C/W as a function of the average signalto-noise ratio (SNR). This shows that for a fixed bandwidth and noise power, the capacity increases monotonically with increasing signal power. Fig. 1.3 shows the channel capacity at a fixed SNR as a function of the bandwidth (W). As W approaches infinity, the channel capacity approaches its asymptotic value of $\frac{SNR}{\log(2)}$. Detailed derivations leading up to (1.1) are given in Chapter 5.

1.1.3 Error Correction Codes

The signal at the output of the channel is distorted due to the presence of noise, fading, and interference. This signal degradation adversely affects the error rate. Error correction



Figure 1.3: The Shannon capacity in bps as a function of the bandwidth.

codes are introduced to combat this degradation by adding controlled redundancy to the information bits. The decoder exploits this redundancy in order to improve the error rate. Error correction codes can be broadly classified as *block codes* or *convolutional codes*.

Block Codes

A (n, k) block code is formed by grouping blocks of k (q-ary) data symbols to produce a codeword of size n. A block code is linear if the modulo-q addition of any two codewords produces a valid codeword. Pioneering work on block codes was done by Richard Hamming with the introduction of a class of single error correcting, binary linear block codes, popularly known as *Hamming codes* [2]. *Linear block codes* (LBCs) are characterized by a $k \times n$ generator matrix **G**. A codeword **c** is formed by multiplying groups of k data symbols (**u**) by **G**.

$$\mathbf{c} = \mathbf{u}\mathbf{G}.\tag{1.2}$$

The parity check matrix of a LBC is a matrix **H** such that if **c** is a valid codeword, then $\mathbf{cH^T} = \mathbf{0}$, where $\mathbf{H^T}$ is the transpose of **H**. The rate of the code is defined as $R_c = k/n$. The distance between two codewords is characterized by the *Hamming distance* (for binary LBCs, this is simply the number of bits by which two codewords differ). The minimum distance d_{min} of a code is the smallest Hamming distance between two distinct codewords. If the code is linear, then the minimum distance is the minimum weight amongst all non-zero codewords. For a code to perform well, the codewords must be as distinct from each other as possible. It is hence desirable for codes to have a large d_{min} . Hamming codes, Golay codes [3], Reed-Solomon codes [4], Reed-Muller codes [5], BCH codes [6] and low-density parity-check codes [7] are examples of widely used block codes.

Convolutional Codes

In block codes, the codeword at a particular time instance depends only on the current input and not on past inputs, i.e. block codes are memoryless. In contrast, convolutional codes [8] are codes with memory. A codeword for a (n, k, m) convolutional code is formed by the linear combination of k current input bits and m past bits which are stored in a shift register. The constraint length K_c of the code is given by $K_c = m + 1$, where m is the code memory. The rate of the code is simply the ratio of the k inputs during one coding interval to the n outputs generated during the same interval i.e. r = k/n. Since convolutional codes can be defined by a finite state machine, the encoding and decoding can be represented using a *trellis*. A direct consequence of the trellis representation is that it facilitates maximum likelihood sequence estimation (MLSE) and maximum a posteriori probability (MAP) estimation. MLSE is performed using the Viterbi algorithm [9] which finds the most likely transmitted sequence corresponding to a received noisy sequence of data. The BCJR algorithm [10] on the other hand, is used to estimate the symbol a posteriori probabilities at each symbol interval, for the given noisy sequence of data.

1.1.4 The Coding Paradox and Capacity Approaching Codes

In his noisy channel coding theorem, Shannon demonstrated the existence of codes that make it possible to achieve arbitrarily low bit error rates provided $r_b < C$ and $n \to \infty$. He showed that under the above constraints, a codeword selected randomly from an ensemble of codes would, with high probability, yield performance approaching capacity. Completely random codes having large block lengths are impractical to decode. Computational feasibility demands that some structure be introduced to the code, which can be exploited to simplify the encoding/ decoding. However, since the code is no longer random, it cannot come close the performance limits predicted by Shannon. This is the coding paradox, which was summarized by Wolfowitz as follows:

Almost all codes are good, except those we can think of.

As a consequence, the search for codes that perform close to the Shannon limit led to the development of codes with higher and higher complexity.

Turbo codes (parallel concatenated convolutional codes) [11] turned previously existing code design principles on their head. Instead of designing very complex codes, with elaborate decoding algorithms, turbo codes concatenate two relatively simple, recursive convolutional codes using a nonuniform interleaver. Since ML decoding is not feasible, the turbo decoder uses *iterative decoding*, which under certain assumptions can closely approximate optimum ML decoding. Well designed turbo codes allow signaling within mere fractions of a dB from the Shannon limit.

Long before the arrival of turbo codes, Gallager [7] in 1960 invented a class of linear block codes with sparse parity check matrices known as *low-density parity-check* (LDPC) codes. Increasing the dimensions of the sparse **H** matrix results in a code with a large d_{min} , which in turn improves the error rate performance of the code. These codes were largely ignored due to the lack of sufficient computing power required for their operation. The arrival of turbo codes and advances in computing resulted in a revival of LDPC codes in the 1990's led primarily by MacKay [12]. Like turbo codes, LDPC codes can be decoded iteratively. However, instead of on a trellis, the decoding proceeds on a *Tanner graph* using the *sumproduct algorithm*. Well designed LDPC codes have been known to perform within 0.0045 dB of the Shannon capacity [13]. LDPC codes and their decoding algorithms are described in more detail in Chapter 2.

1.1.5 Modulation

The modulator groups $\log_2 M$ bits from the channel encoder's output into one of M possible waveforms. M is known as the modulation order or the alphabet size. The purpose of the modulator is to first shape the power spectrum of the baseband signal and then to translate the center frequency to match the channel. For instance, if the channel is wireless, the transmission of radio waves is accomplished using antennas. The size of the antenna is proportional to the wavelength λ and since $\lambda = 3 \times 10^8/f$, a low frequency (baseband) signal would require a prohibitively large antenna. It is therefore necessary to frequency translate the baseband signal using a high frequency carrier before transmission.

Modulators can be broadly classified as *linear* or *nonlinear* depending upon whether the superposition theorem applies to the baseband waveforms or not. *Phase shift keying* (PSK), *pulse amplitude modulation* (PAM), *quadrature amplitude modulation* (QAM) are all examples of linear modulations [14], whereas continuous phase modulation (CPM) [15] is a popular example of nonlinear modulation.

Over the last few years, there has been a surge in the popularity of wireless standards such as Bluetooth [16] and Global System for Mobile communications (GSM). For instance, the number of GSM users have more than doubled from one billion in 2004 to over 2.3 billion, with GSM handset sales exceeding over 980 million units in a single year (2006)¹. However, the ever increasing number of users, combined with the insatiable need for high data rates places considerable strains on the quality-of-service (QOS) and the available (limited) radio spectrum. Limited bandwidth resources make modern (terrestrial) communication systems susceptible to adjacent channel interference [17]. This occurs due to the energy of a signal "leaking" into to neighboring frequency bands and can hence be mitigated if the modulated signal has a power spectrum which exhibits small side-lobes and a fast-roll offs. Continuous phase modulation (CPM) [15] is ideally suited for radio environments suffering from spectral

¹http://www.gsmworld.com/about/history.ahtml

congestion. In CPM, the signal phase transitions are continuous from one symbol epoch to another. This phase continuity yields the much desired compact power spectrum, with small side-lobes and fast spectral roll-off.

Extended battery life is a key requirement in mobile communication devices (especially in military communication systems). The battery life in turn depends on the efficiency of the power amplifier. The amplifier efficiency increases with increasing input drive levels, which increases the amplifier nonlinearities. There is hence a tradeoff between the battery life and signal distortions caused by the amplifier nonlinearities [18]. Constant envelope modulations such as CPM are not affected by amplifier nonlinearities, thereby permitting more efficient power amplification relative to modulations such as QAM and APSK.

The optimum detector (in terms of energy efficiency) for CPM is *coherent* [15] which accurately tracks the signal phase and has perfect channel state information (CSI). However, coherent detection is often not feasible due to increased complexity or not possible due to rapidly varying channel conditions [19]. *Differential phase detectors* [20] and *Noncoherent detectors* [21] provide a more pragmatic alternative due to the absence of carrier phase recovery, albeit at the expense of energy efficiency. The combination of a compact power spectrum, constant envelope and the existence of low-complexity receivers makes CPM well suited for Bluetooth, GSM, spread spectrum communications and mobile satellite communications.

The performance of a CPM system can be improved by better detector design, in particular by designing a receiver to exploit the memory inherent in the modulation. The energy efficiency of CPM can also be improved by combining channel coding with CPM, for instance by using a binary convolutional code to increase the memory of the modulation [15, 22, 23]. Additional gains in energy efficiency can be made by using nonbinary (ring) convolutional codes [24]. Ungerboeck's trellis coded modulation (TCM) [25] paradigm was applied to trellis coded, continuous phase frequency shift keying (CPFSK, Chapter 3) with coherent reception in [26] and later to partial response CPM (TCCPM) with noncoherent reception in [27]. The widespread interest generated by turbo codes (capacity-approaching codes in general) have resulted in application of the "turbo principle" to CPM [28, 29, 30], by passing soft-information between the demodulator and the decoder. CPM has also been concatenated with turbo codes [31, 32], LDPC codes [33] and recently with irregular repeat accumulate (IRA) codes [34] in [35].

1.2 Problem Statement

A communication system with coding and modulation is typically designed based on the following factors [36]

- 1. Error rate: which measures how reliable the information transmission is.
- 2. Energy efficiency: which is typically expressed in terms of the average signal-to-noise ratio per data bit \mathcal{E}_b/N_0 .
- 3. Bandwidth efficiency: which is measured as bits per second/Hz (bps /Hz).
- 4. System complexity and hence the cost.

However, as seen in Section 1.1.2 these requirements are often conflicting. As an example, in order to increase the number bits transmitted per second for a fixed bandwidth, the \mathcal{E}_b/N_0 has to be increased to keep the error rate arbitrarily low (Fig. 1.2). Similarly, if the SNR is fixed, then the bandwidth must be increased in order to increase the data rate (Fig. 1.3). For a given SNR and bandwidth, the error rate can also be reduced by using sophisticated signal processing, which increases the system complexity and hence the cost. Hence designing a system with coding and modulation is based on tradeoffs between energy efficiency, bandwidth efficiency and complexity.

Our goal in this dissertation is to address the above tradeoffs while designing a coded CPM system. At first glance, it is tempting to dismiss the above problem as trivial. However, a system designer must contend with the following issues in order to arrive at a satisfactory solution:

1. There are two popular approaches to designing a system with channel coding. The first is known as *coded modulation* (CM), in which the channel code and modulator are defined over the same alphabet and are concatenated using a symbol-interleaver. Alternatively, we could concatenate a binary encoder and the *M*-ary modulator using

a bit-interleaver, and exchange bit-wise soft-information between the demodulator and the decoder. Such as design is known as *bit-interleaved coded modulation* [37].

2. In coded CPM systems, the energy and bandwidth efficiency are determined by the alphabet size *M*, the type and width of the pulse shape, modulation index *h*, code rate *r* and also on the choice of the CPM detector. There exists an inherent tradeoff between code rate and CPM parameters. For instance, if a lower rate code is used, then to maintain a specified bandwidth efficiency, the modulation must either have a smaller modulation index, use longer pulses, or use a smaller signal set. For any particular scenario, it is not clear if the coding gain due to using the lower rate code will offset the performance loss due to using modulation that is further from being orthogonal or due to the additional inter-symbol interference (ISI) induced by the longer pulses. The code and modulation parameters must hence be selected with some care.

To expound on the above statements, we consider the following simple example. Let us suppose our goal is to design a CPM system with a spectral efficiency of $\eta = 0.84$ bps/Hz. This can be achieved using uncoded minimum shift keying (MSK), which is a popular class of CPM with M = 2, h = 1/2 and rectangular pulse shape extending up to one symbol interval ². For ease of exposition, we assume coherent reception (Chapter 4) and an AWGN channel. Under these assumptions, uncoded MSK requires an $\mathcal{E}_b/N_0 = 9.6$ dB to signal at a bit error rate (BER) of 10^{-5} . For the coded system, we could arbitrarily select a set of modulation parameters and code rate that meet our bandwidth requirement, in conjunction with a powerful channel code, for instance by concatenating a rate 1/2 CDMA 2000 turbo code with binary CPFSK with h = 1/8. Fig. 1.4 shows BER for a coded system with the above mentioned parameters using the CDMA 2000 turbo code [38], with interleaver size 12282 (data) bits. Observe that the gain of the code is not enough to overcome the loss due to using nonorthogonal tones.

Hence in order to obtain the optimum combination of code and CPM parameters, a

²The class of CPM signals using rectangular pulse shape extending up to one symbol interval is known as continuous phase frequency shift keying (CPFSK)



Figure 1.4: BER comparison of uncoded MSK with coded CPFSK with parameters M = 2 and h = 1/8. The coded system uses an outer rate 1/2 CDMA 2000 turbo code. Coherent CPM detection and AWGN channel are assumed.

rigorous search must be performed over all (allowable) code and modulation parameters, using some performance metric which quantifies the suitability of a particular set of parameters. It is also noted that parameters such as M, h, r and pulse width can assume a wide range of values, due to which we have an infinitely large search space.

- 3. The next issue is the choice of the channel code. Here, we are presented with several choices such as convolutional codes, turbo codes etc. A good code allows us to meet our energy efficiency requirements without a significant increase in the system complexity. Additionally, we have the option of iteratively exchanging soft-information between the detector and decoder.
- 4. Designing detectors for CPM that are energy efficient and have low complexity is also a non-trivial task. Also, the detector complexity is linked to the choice of the CPM parameters and it is possible that the complexity can be greatly reduced by accepting design solutions that are less than ideal.

5. The final and perhaps the most critical issue is the choice of the performance metric. The search for coded CPM parameters with the best energy efficiency at different bandwidth efficiencies spans a very large search space. Hence, any performance metric or cost function that we use to perform this search must be feasible to compute for the different modulation parameters, code rates, channel conditions, and receiver formulations considered. Additionally, such a cost function should also be a realistic indicator of the system performance.

1.3 Methodology

In this dissertation, we focus on coded-CPM systems which employ bit-interleaved coded modulation [37]. The BICM approach to coded-system design offers several advantages. In BICM, the code alphabet and the modulation alphabet need not match. This simplifies system design when different modulation orders are employed. BICM also improves the temporal diversity of the system [39]. Additionally, capacity-approaching codes are predominantly binary, hence once the optimum design parameters are determined, an "off-the-shelf" capacity-approaching code (DVB-S2 LDPC code, UMTS turbo code, CDMA 2000 turbo code etc..) can be incorporated to get very good performance. Due to these advantages, the BICM paradigm finds widespread application in modern communication systems.

Because finding the most energy efficient combination of code rate and CPM parameters for a given bandwidth efficiency and receiver complexity is an *optimization problem*, the first step is to identify an appropriate *cost function*. Optimization of coherently detected, convolutional coded-CPM under bandwidth constraints has been previously investigated in [22] and more recently in [40] and [41]. The cost function used in these papers is based on the distance spectrum of the serially concatenated system. The resulting performance bounds predict the performance of system employing an ML receiver. Since practical systems only approximate ML decoding by turbo-style processing, the performance bounds are hence indicative of system performance primarily at high SNR. However, it is often of practical interest to optimize with respect to the minimum SNR required to signal at some infinitesimally low error rate. Furthermore, as mentioned in [29], such bounds are nontrivial to compute for more sophisticated coded CPM systems (such as those employing capacity-approaching codes with noncoherent/ differential detectors).

In this dissertation, we turn to information theory and propose using a cost function that is derived from the *constrained* capacity of the system. More specifically, the cost function is the information-theoretic minimum \mathcal{E}_b/N_0 required for reliable signaling under the constraints of a class of modulation, equally likely input signals, a desired bandwidth efficiency, and a particular receiver design. There are several benefits of performing a capacity-based optimization. The most obvious advantage is that capacity specifies the performance limits for a coded system and inherently takes into account the tradeoff between code rate and energy efficiency. This tradeoff can be easily extended in our case to account for the tradeoff between code rate and modulation parameters. It is a very practical predictor of system performance since either capacity approaching codes or powerful codes tailored to the specific modulation parameters [33], [35], [42] can be used to get close to the constrained capacity. While closed form solutions for constrained capacity are at times non-trivial to compute, due to the availability of cheap and fast processing, it is now feasible to accurately estimate the constrained capacity by using Monte Carlo integration with a very large number of trials [37], [43]. Our capacity calculations also take into account the design of the detector. Due to these significant advantages, capacity-driven optimizations have been previously performed for noncoherently detected orthogonal FSK [44] and coded APSK [45]. We are however the first (to the best of our knowledge) to consider consider such an optimization for coded-CPM, and also the first to consider an optimization for a CPM system employing differential phase detection.

One could argue that the threshold on \mathcal{E}_b/N_0 could instead be found by simulating the BER performance of the system, which includes the particular channel code, CPM modulator, desired channel, detector and decoder using Mote-Carlo trials. However BER simulations for coded systems (especially those incorporating capacity-approaching codes) are very time consuming. We show in Chapter 5 that capacity calculations by Monte-Carlo trials are several hundred times faster than BER simulations with capacity-approaching codes. This computational efficiency is a very significant advantage provided by the constrained-capacity cost function, simply because the optimization search space is typically very large. Another

limitation of a BER simulation based optimization is that the SNR threshold is tied to the choice of the channel code and the optimization performed with respect to a particular channel code ceases to be valid when a different code is employed. In contrast, the capacity-based threshold is independent of the choice of the channel code since any well designed code will almost certainly approach capacity.

The rest of the dissertation is organized as follows. Chapter 2 discusses some channel coding techniques often employed in modern communication systems. Chapter 3 describes the CPM signal and it's spectral characteristics. Coherent and differential detectors for CPM are described in detail in Chapter 4, as is a description of the novel, soft-output differential detector for CPM signals using what we term the *soft-output*, *soft-decision differential phase detector* (SO-SDDPD). Chapter 5 describes the methodology for finding the capacity under modulation constraints. The optimization undertaken and subsequent results are described in Chapter 6. Chapter 7 concludes the dissertation.

Chapter 2

Channel Coding for Modern Communication Systems

The signal at the output of the communication channel is distorted due to the presence of noise, fading, and interference. The signal degradation adversely affects the fidelity of the received information. Error correction codes are used to combat the effects of the channel by adding controlled redundancy to the transmitted information bits. The decoder exploits this redundancy in order to improve the signal fidelity.

In this chapter, we review some error correction (channel) codes that find widespread use in modern communication system. We begin with a review of the popular *convolutional codes* [8]. Pertinent topics such as their structure, encoding and trellis representation are discussed. The Viterbi algorithm [9] and BCJR algorithm [10], two algorithms widely used in decoding convolutional codes are described in detail.

Next, we describe a class of codes known as *capacity-approaching codes*. Capacity approaching codes can be described as codes that perform within 1 dB or less of the Shannon capacity (Chapter 5). The first reported capacity approaching code was the original *turbo code* by Berrou et. al. [11] which was 0.7 dB from the unconstrained capacity. Since then turbo codes have found extensive use in applications such as mobile telephony (UMTS [46], CDMA2000 [38]) second generation digital video broadcasting (DVB-RCS) [47], deep space communications (CCSDS) and broadband wireless access (WiMAX) [48] to name a few. *Low-density parity-check* (LDPC) codes are another class of capacity approaching codes.

Although invented by Gallager [7] in 1960, they did not generate sufficient interest due to the high complexity involved in their encoding and decoding. The arrival of turbo codes and advances in computing led to their resurgence in the mid 90's, triggered primarily by MacKay's paper [12]. We present an overview of capacity approaching codes with emphasis on LDPC codes. We review linear block codes of which LDPC codes are a special sub-class. Next, the decoding algorithms are described in detail following which, code construction and encoding are discussed. Select performance curves for the LDPC codes used in the second generation digital video broadcasting standard (DVB-S2) and mobile WiMAX (IEEE 802.16e) are given. The operation of turbo codes is reviewed before concluding the chapter.

2.1 Convolutional Codes

Convolutional codes [8] are codes with memory and are encoded using a shift register. The memory (m) of the code is the number of delay elements in the shift register. The *constraint length* K_c of the code is defined to be $K_c = m + 1$.

2.1.1 Encoding Convolutional Codes

A codeword for a (n, k, m) convolutional code is formed by the linear combination of k current input bits and m past bits which are stored in a shift register. The rate of the code is simply the ratio of the k inputs during one coding interval to the n outputs generated during the same interval i.e. r = k/n.

Fig. 2.1 shows the encoder for a rate 1/2 and $K_c = 3$ convolutional code. Observe that the shift register is feedforward. Convolutional codes in which there is no feedback from the output are known as *nonrecursive* convolutional (NRC) codes. Likewise, convolutional codes in which there is a feedback from one of the outputs are known as *recursive* convolutional codes Fig. 2.2. The encoder can be represented in terms of generator vectors, which denote the connections between the shift register elements and the adders [49]. The feedforward generator vectors for the NRC code shown in Fig. 2.1 are

$$g_0 = \left[\begin{array}{rrr} 1 & 0 & 1 \end{array} \right]$$



Figure 2.1: A rate 1/2, constraint length 3 nonrecursive convolutional encoder with octal generators [7, 5].

$$g_1 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$
(2.1)

The feedback g_r and feedforward g_f generator vectors for the recursive code shown in Fig. 2.2 are

$$g_r = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$$

$$g_f = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}$$
(2.2)

The recursive code is also *systematic*, since the first parity bit is simply the input bit and is hence known as a recursive systematic convolutional (RSC) code. As is common in the literature, the generator vectors are expressed in octal representation to give a compact notation for the convolutional codes. The code in Fig. 2.1 can be denoted as NRC [7, 5] and the code in Fig. 2.2 as RSC [7, 3].

Convolutional codes can continuously encode a stream (an infinitely long sequence) of data. In practice however, the convolutional code encodes non-overlapping blocks of data, resulting in finite length code sequences. In such scenarios, it is desirable to reset the encoder



Figure 2.2: A rate 1/2, constraint length 3 recursive convolutional encoder with feedback, feedforward octal generators [7, 3] respectively.

to the all zeros state prior the next block of data. This process is known as *trellis termination*. In NRC codes, this is accomplished by feeding m zeros into the encoder at the end of each data block. In RSC codes, termination is accomplished by my setting the input to the output of the feedback register that is fed back to the input. This resets the encoder to zero after m shift intervals. The m additional bits at the end of each code sequence are called the *tail-bits*.

2.1.2 State Diagram and Trellis

A binary convolutional code is a finite state machine with 2^m states. A state diagram shows how the encoder states are related by specifying the inputs required to move from one state to another and the corresponding outputs that are produced for any given set of state transitions.

Fig. 2.3 shows the state diagram for NRC [7, 5] code. Each node in the diagram denotes a possible encoder state. There are two branches entering and leaving each node. The branches are labelled (u / $p_1 p_2$), u is an input bit and p_1 , p_2 are the encoded parity bits.

A *trellis* is an expansion of the state diagram which explicitly shows how the state transitions evolve over time for different possible input sequences. Each distinct sequence of input bits corresponds to a unique path through the trellis. Fig. 2.4 shows the trellis diagram for the NRC [7,5] code.



Figure 2.3: State diagram for a rate 1/2, constraint length 3 nonrecursive convolutional code with generators [7, 5].



Figure 2.4: Trellis diagram for a rate 1/2, constraint length 3 nonrecursive convolutional code with generators [7, 5] respectively. Dashed arrows denote branches corresponding to an input symbol 0.

2.1.3 Decoding Convolutional Codes

Convolutional codes are most conveniently decoded on a trellis. Two popular approaches to trellis-based decoding are *maximum likelihood* (ML) decoding using the Viterbi algorithm [9] and *maximum a posteriori* (MAP) decoding using the BCJR [10] algorithm. Before going into the details of the above mentioned algorithms, we define the following notation:

- $\mathbf{u} = \{u_0, u_1, ..., u_{N_u-1}\}$ denotes a length N_u bit sequence.
- **b** denotes the encoder generated codeword corresponding to **u**. Its length is N_b .
- \mathbf{x} denotes the BPSK modulated symbol sequence generated as follows

$$\mathbf{x} = 2\mathbf{b} - 1. \tag{2.3}$$

Maximum Likelihood Decoding: The Viterbi Algorithm

A maximum likelihood decoder selects the most probable path through the trellis. This is done by comparing the received sequence against all possible paths through the trellis and selecting the path "closest" to the received sequence. A naive approach would be to implement ML decoding using a table look-up. However, since the size of this table would be 2^{N_b} , such a table look-up is computationally prohibitive for even moderate sequence lengths. The *Viterbi algorithm* [9] provides a computationally feasible method for performing ML decoding by exploiting the underlying structure of the convolutional code (it is the same structure that allows the code to be represented as a trellis).

An important component of the Viterbi algorithm is the assignment of a "weight" to each branch in the trellis, which indicates how different the received code bits are from the postulated transmitted bits associated with that branch. The Viterbi algorithm finds the lowest weight path through the trellis.

The BPSK modulated codeword \mathbf{x} is transmitted through an AWGN channel having noise variance $\sigma^2 = N_0/2$. The signal at the output of the channel is

$$\mathbf{r} = \mathbf{x} + \mathbf{n}_{\mathbf{w}}.\tag{2.4}$$

The following notation need to be defined before the algorithm can be described

- $\{S_k = s'\}$ for $s' = 0, 1, ..., 2^m 1$ denotes the trellis state at time instant k.
- $\{S_k = s'\} \rightarrow \{S_{k+1} = s\}$ denotes a state transition corresponding to an input u_k .
- $\gamma_{k+1}(s', s)$ is the branch metric i.e. the weight associated with a branch whose starting and terminal states are $\{S_k = s'\}$ and $\{S_{k+1} = s\}$ respectively.
- $\Gamma_{s',k}$ is the *path metric* stored in trellis state $\{S_k = s'\}$.

The goal of the Viterbi algorithm is to minimize the probability of making a *codeword error*. Under the assumption that the encoder begins and ends in the all zeros state, the Viterbi algorithm is as follows:

1. At k = 0, initialize

$$\Gamma_{s',0} = \begin{cases} 0 & s' = 0\\ \infty & \text{otherwise} \end{cases}$$
(2.5)

2. Compute $\gamma_{k+1}(s', s)$ for every branch in the trellis

$$\gamma_{k+1}(s',s) = \sum_{i=0}^{n-1} ||r_{kn+i} - b_{kn+i}(s' \to s)||^2, \qquad (2.6)$$

where $b_{kn+i}(s' \to s)$ for i = 0, 1, ..., n-1 are the code bits associated with the state transition $\{S_k = s'\} \to \{S_{k+1} = s\}.$

3. Each state s has two branches entering into it. Pick the branch for which the sum $\Gamma_{s',k} + \gamma_{k+1}(s',s)$ is the smallest. This branch corresponding to the state transition $\{S_k = s'\} \rightarrow \{S_{k+1} = s\}$ is known as the *surviving branch* and the path metric is now

$$\Gamma_{s,k+1} = \Gamma_{s',k} + \gamma_{k+1}(s',s).$$
(2.7)

The branch with the larger metric is deleted. In addition to storing the surviving path's metric, the algorithm needs to keep a track of the surviving *state sequence*, i.e. the sequence of trellis states from which the surviving branches emanate.
4. Increment time

$$k = k + 1. \tag{2.8}$$

Repeat from step 2 until the end of the trellis is reached.

5. The maximum likelihood path leads into the state $S_{N_u} = 0$. Using the stored surviving state sequence, the Viterbi algorithm traces the ML path by sweeping right to left across the trellis. The ML estimate of data bits are the input bits associated with each branch (state transition) of the ML path.

Maximum A Posteriori Decoding: The BCJR Algorithm

The goal of the BCJR algorithm is to minimize the bit error probability. Towards this end, the algorithm produces the log-likelihood ratio

$$z_k = \log \frac{P[u_k = 1 | \mathbf{r}]}{P[u_k = 0 | \mathbf{r}]}.$$
(2.9)

The following notation need to be defined before the algorithm can be described:

- $\mathcal{S}^{(1)}$ denote the set of state transitions $\{S_k = s'\} \to \{S_{k+1} = s\}$ corresponding to $u_k = 1, \mathcal{S}^{(0)}$ is defined similarly for $u_k = 0$.
- $P[\{S_k = s'\} \rightarrow \{S_{k+1} = s\} | \mathbf{r}]$ denotes the probability that the encoder made the transition from $S_k = s'$ to $S_{k+1} = s$, given the entire received codeword.
- $\alpha_k(s', s)$ is defined as

$$\alpha_k(s') = P[S_k = s', \mathbf{r}_0^{k-1}].$$
(2.10)

• $\beta_{k+1}(s)$ is defined as

$$\beta_{k+1}(s) = P[\mathbf{r}_{k+1}^{N_u - 1} | S_{k+1} = s].$$
(2.11)

• $\gamma_{k+1}(s', s)$ is defined as

$$\gamma_{k+1}(s',s) = P[S_{k+1} = s, r_k | S_k = s'].$$
(2.12)

It is easy to infer that

$$P[u_{k} = 1 | \mathbf{r}] = \sum_{S^{(1)}} P[S_{k} = s', S_{k+1} = s | \mathbf{r}]$$

=
$$\sum_{S^{(1)}} P[S_{k} = s', S_{k+1} = s, \mathbf{r}] / P[\mathbf{r}].$$
 (2.13)

Now

$$P[S_{k} = s', S_{k+1} = s, \mathbf{r}] = P[\mathbf{r}_{0}^{k-1}, r_{k}, \mathbf{r}_{k+1}^{N_{u}-1}, S_{k} = s', S_{k+1} = s]$$

$$= P[r_{0}^{k-1}, r_{k}, S_{k} = s', S_{k+1} = s]P[\mathbf{r}_{k+1}^{N_{u}-1}|r_{k}, \mathbf{r}_{0}^{k-1}S_{k} = s', S_{k+1} = s]$$

$$= P[S_{k} = s', r_{0}^{k-1}]P[S_{k+1} = s, r_{k}|S_{k} = s', r_{0}^{k-1}].$$

Since the finite state machine describing the convolutional encoder is a Markov process,

$$P[S_k = s', S_{k+1} = s, \mathbf{r}] = P[S_k = s', r_0^{k-1}]P[S_{k+1} = s, r_k | S_k = s']P[\mathbf{r}_{k+1}^{N_u - 1} | S_{k+1} = s].$$
(2.14)

(2.13) can now be written as

$$P[u_{k} = 1 | \mathbf{r}] = \sum_{\mathcal{S}^{(1)}} \alpha_{k}(s') \gamma_{k+1}(s', s) \beta_{k+1}(s)] / P[\mathbf{r}].$$
(2.15)

 $P[u_k = 0|\mathbf{r}]$ is similarly expressed by summing over S^0 . From (2.9), (2.13) and (2.15) the LLR is

$$z_{k} = \log \frac{\sum_{\mathcal{S}^{(1)}} \alpha_{k}(s') \gamma_{k+1}(s', s) \beta_{k+1}(s)}{\sum_{\mathcal{S}^{(0)}} \alpha_{k}(s') \gamma_{k+1}(s', s) \beta_{k+1}(s)},$$
(2.16)

 $\alpha_k(s)$ for $k = 1, ..., N_u + 1$ is calculated using the following forward recursion

$$\alpha_k(s) = \sum_{s'=0}^{2^m - 1} \alpha_{k-1}(s')\gamma_k(s', s), \qquad (2.17)$$

using the initialization

$$\alpha_0(s) = \begin{cases} 1 & s = 0 \\ 0 & \text{otherwise} \end{cases}$$
(2.18)

 $\beta_k(s)$ for $k = N_u - 1, N_u - 2, ..., 1$ is calculated using the following backward recursion

$$\beta_k(s) = \sum_{s'=0}^{2^m - 1} \beta_{k+1}(s')\gamma_{k+1}(s', s), \qquad (2.19)$$

$$\beta_{N_u}(s) = \begin{cases} 1 & s = 0 \\ 0 & \text{otherwise} \end{cases}$$
(2.20)

The BCJR algorithm is summarized as follows

- 1. In the forward sweep through the trellis, calculate $\gamma_k(s', s)$ (4.27) for each branch in the trellis and $\alpha_k(s)$ 2.10 using (2.17), for $s = 0, 1, ..., 2^m 1$.
- 2. In the backward sweep through the trellis, compute $\beta_k(s)$ (2.11) using (2.19) for $s = 0, 1, ..., 2^m 1$ and the LLR of the message bit at the k^{th} trellis section z_k (2.16).
- 3. z_k is compared to a zero threshold to obtain hard estimates of the data bits \hat{u}_k

$$\hat{u}_k = \begin{cases} 1 & z_k \ge 0 \\ 0 & \text{otherwise} \end{cases}$$
(2.21)

Due to the forward and backward sweep through the trellis, the BCJR algorithm is often called the "forward-backward" algorithm.

Note that $\gamma_{k+1}(s', s)$ can also be written as

$$\gamma_{k+1}(s',s) = P[S_{k+1} = s | S_k = s'] P[r_k | (S_k \to S_{k+1}) = (s' \to s)], \qquad (2.22)$$

When the MAP decoder doesn't receive *a priori* information **v** on the data bits, the decoder assumes that all bits are equally likely due to which $P[S_{k+1} = s|S_k = s']$ is a constant that can be dropped from (2.22) without affecting the overall LLR (2.16). If *a priori* information were delivered to the decoder, then **v** could be used to update the value of $P[S_{k+1} = s|S_k = s']$. *Extrinsic information* $\bar{\mathbf{z}}$ is obtained by subtracting the *a priori* information from the LLR

$$\bar{\mathbf{z}} = \mathbf{z} - \mathbf{v}.\tag{2.23}$$

2.1.4 Code Puncturing

The convolutional codes mentioned thus far are rate 1/2 codes, or more generally, rate 1/n codes. When higher code rates are required, one could use rate k/n codes where k > 1 [14]. Decoding these codes is computationally intensive since the decoder complexity increases

exponentially with k. A more pragmatic approach to obtaining higher rate codes is by *puncturing* rate 1/n codes. Puncturing involves periodically deleting specific code bits from the encoder's output. Since the mother code is still 1/n, the number of states/ branches in the trellis remains unchanged. Puncturing is described in terms of a *puncturing matrix* P [14]

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1p} \\ p_{21} & p_{22} & \dots & p_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \dots & p_{np} \end{bmatrix}.$$
(2.24)

Each column of P corresponds to the n parity check bits from the encoder and p is the puncturing period. The elements of P are set to 1 or 0, where 0 indicates that corresponding parity check bit is deleted. The punctured rate is p/n_p , where n_p is the number of ones in P. Puncturing matrices cannot be selected arbitrarily since a bad puncturing pattern could result in high error floors. Puncturing matrices for high rate NRC codes resulting in good free distances have been found in [50].

2.2 Low-Density Parity-Check Codes

2.2.1 Review of Linear Block Codes

Let \mathcal{V}_n be an *n*-dimensional vector space over $\{0, 1\}$. A (n, k) binary linear block code is a *k*-dimensional vector sub-space (\mathcal{C}_n) of \mathcal{V}_n , such that the modulo-2 sum of any two codewords is another valid codeword. The code is characterized by its $k \times n$ generator matrix **G** which consists of the basis vectors for \mathcal{C}_n . A codeword **b** is hence simply generated by the linear combination of the basis vectors. If **u** is a length k message vector, the codeword **b** is given by

$$\mathbf{b} = \mathbf{u}\mathbf{G}.\tag{2.25}$$

The dual-space \overline{C}_n of C_n is a n-k dimensional sub-space of \mathcal{V}_n such that for all $\mathbf{b} \in C_n$ and $\mathbf{v} \in \overline{C}_n$

$$\mathbf{b}\mathbf{v}^{\mathrm{T}} = 0. \tag{2.26}$$

The *parity-check matrix* \mathbf{H} contains the basis vectors for \overline{C}_n . Each row in \mathbf{H} specifies a parity check equation. The code bits in positions where the row entry is 1 must sum to 0. Hence if \mathbf{b} a valid codeword, then

$$\mathbf{b}\mathbf{H}^{\mathbf{T}} = \mathbf{0}. \tag{2.27}$$

The Hamming distance between two codewords is the number of bits by which they differ. The minimum distance d_{min} of a code is the smallest Hamming distance between two distinct codewords and is equal to the smallest number of linearly dependent columns in **H**.

Low-density parity-check codes are a class of linear block codes characterized by very sparse **H** matrices, i.e. the number of 1's is very small compared to the number of 0's. When n is very large, the sparseness of **H** results in large d_{min} , which results in very low bit error rates for the code. Also having a sparse **H** matrix, reduces the decoder complexity. Since LDPC codes are best understood in terms of their decoding, we first describe their decoding and then discuss their construction and encoding.

2.2.2 Decoding LDPC Codes

LDPC codes are decoded on a *Tanner graph* [51]. A Tanner graph is a bipartite graph describing the parity check matrix for a code . It has two classes of nodes, called the *check* nodes and variable nodes, connected by edges. The variable nodes (v-nodes) correspond to the bits in the codeword or equivalently, the columns of **H**. The check nodes (c-nodes) correspond the parity check equations or equivalently, the rows of **H**. This implies there are n v-nodes and at least n - k c-nodes. The i^{th} check node is connected to the j^{th} variable node if and only if the $(i, j)^{th}$ element of the parity check matrix is one, i.e. if $h_{ij} = 1$. Fig.



Figure 2.5: The Tanner graph of a (7, 4) Hamming code.



Figure 2.6: A length four cycle in the Tanner graph of a (7, 4) Hamming code. The edges correspond to the italicized entries in (2.28).

2.5 shows a Tanner graph for (7, 4) Hamming code with H matrix

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}.$$
 (2.28)

A cycle of length l in a Tanner graph is a path of l distinct edges which closes on itself. The girth of a Tanner graph is the minimum cycle length of the graph. The shortest possible cycle in a Tanner graph has length 4. Length four cycles manifest as four 1's in the corners of a sub-matrix in **H**. The italicized entries in (2.28) indicate the length 4 cycle shown in Fig. 2.6.

LDPC codes are decoded iteratively using a Tanner graph in which messages are ex-

changed between the v and c-nodes with the edges acting as information pathways. Softdecision decoding on LDPC codes is accomplished using the sum-product algorithm also known as the *message passing/ belief propagation* algorithm [52], [53] and is described below.

Let \mathbf{b} be the codeword. The BPSK modulated \mathbf{x} codeword is generated as follows

$$x_i = (-1)^{b_i}. (2.29)$$

x is transmitted through an AWGN channel having noise variance $\sigma^2 = N_0/2$. The signal at the output of the channel is hence

$$\mathbf{r} = \mathbf{x} + \mathbf{n}_{\mathbf{w}}.\tag{2.30}$$

The following notation needs to be defined before the algorithm can be described

- $Q_0 = P(b_i = 0|S_i), Q_i = P(b_i = 1|S_i)$. S_i is the event that the bits in **b** satisfy the check equations involving c_i .
- $\bar{q}_{ij}(d)$ is the extrinsic information to be passed from v-node *i* to c-node *j* and is the probability that $b_i = d$ given extrinsic information from check nodes and channel sample r_i .
- $\bar{r}_{ji}(b)$ is the extrinsic information to be passed from c-node j to v-node i and is the probability of the j^{th} check equation being satisfied give that $b_i = d$.
- $C_i = \{j : h_{ji} = 1\}$. This is the set of row location of the 1's in the *i*th column.
- $C_{i\setminus j} = \{j' : h_{j'i} = 1, j' \neq j\}$. The set of row locations of the 1's in the i^{th} column, excluding location j.
- $R_j = \{i : h_{ji} = 1\}$. This is the set of column location of the 1's in the j^{th} row.
- $R_{j\setminus i} = \{i' : h_{ji'} = 1, i \neq i\}$. The set of column locations of the 1's in the j^{th} row, excluding location i.

The sum-product algorithm is as follows

R. Iyer Seshadri Chapter 2. Channel Coding for Modern Communication Systems 29

1. Initialize

$$\bar{q}_{ij}(0) = 1 - p_i = \frac{1}{(1 + \exp(-2r_i/\sigma^2))},$$
$$\bar{q}_{ij}(1) = p_i = \frac{1}{(1 + \exp(2r_i/\sigma^2))}.$$
(2.31)

2. At each c-node, update r message

$$\bar{r}_{ji}(0) = \frac{1}{2} + \frac{1}{2} \prod_{i' \in R_{j' \setminus i}} \left(1 - 2\bar{q}_{j'i}(1) \right),$$
$$\bar{r}_{ji}(1) = 1 - \bar{r}_{ji}(0).$$
(2.32)

3. Update $\bar{q}_{ij}(0)$ and $\bar{q}_{ij}(1)$

$$\bar{q}_{ij}(0) = k_{ij}(1-p_i) \prod_{j' \in C_{i \setminus j}} (\bar{r}_{j'i}(0)),$$
$$\bar{q}_{ij}(1) = k_{ij}(p_i) \prod_{j' \in C_{i \setminus j}} (\bar{r}_{j'i}(1)).$$
(2.33)

4. Calculate the APPs

$$Q_{i}(0) = k_{ij}(1 - p_{i}) \prod_{j \in C_{i}} \bar{r}_{ji}(0),$$
$$Q_{i}(1) = k_{ij}(p_{i}) \prod_{j \in C_{i}} \bar{r}_{ji}(1).$$
(2.34)

5. Make hard-decisions

$$\hat{b}_i = \begin{cases} 1 & \text{if } Q_i(1) \ge 0\\ 0 & \text{otherwise.} \end{cases}$$
(2.35)

6. Repeat from step 2 until either $\hat{\mathbf{c}}\mathbf{H}^{\mathbf{T}} = \mathbf{0}$ or maximum number of iterations have been reached.

Since $\mathbf{bH^T} = \mathbf{0}$ for a valid codeword, error detection comes for free in LDPC codes. If the Tanner graph contains no cycles, then Q_i converges to the true APP as the number of iterations reaches infinity.

2.2.3 Constructing LDPC Codes

LDPC codes can be broadly classified as either *regular* or *irregular*. An LDPC code is regular if the rows and columns of **H** have *uniform weight*, i.e. all rows have the same number of ones and all columns have the same number of ones. An LDPC code is irregular if the rows or columns have *non-uniform weight*. The *degree* of a node is the number of edges emanating from that node. The *degree distribution pair* (λ, ρ) for a LDPC code is defined as

$$\lambda(x) = \sum_{i=1}^{d_v} \lambda_i x^i,$$

$$\rho(x) = \sum_{i=1}^{d_c} \rho_i x^i.$$
(2.36)

 λ_i (ρ_i) represent the fraction of edges emanating from variable (check) nodes of degree *i*. d_c is simply the row weight for regular codes and the maximum row weight for irregular codes. Similarly, d_v is the column weight for regular codes and the maximum column weight for irregular codes.

The original LDPC codes by Gallager and MacKay were regular codes. MacKay and Neal [12] described methods for constructing (almost) regular sparse **H** matrices. The idea is to randomly generate a $M \times N$ matrix **H** with weight d_v columns and weight d_c rows, subject to some constraints. In their construction 1*A*, the overlap (i.e. an entry of 1 at same row location) between any two columns is no greater than 1, which ensures that there are no length 4 cycles. In construction 2*A*, M/2 columns have $d_v = 2$, with no overlap between any pair of columns. Remaining columns have $d_v = 3$. As with 1*A*, the overlap between any two columns is no greater than 1. Construction 1*B* and 2*B* are obtained by deleting select columns from 1A and 2A. The resulting **H** matrix might have a reduced rank which results in a higher rate code. Although regular codes perform well, they are still about 1 dB from capacity. Performance improvements are possible using irregular LDPC codes.

Variable and check nodes have conflicting requirements. Variable nodes benefit from having a large degree. High degree variable nodes converge to the correct value quickly. This increases the quality of information passed to the check nodes, which in turn helps the lower degree variable nodes to converge. Conversely, LDPC codes perform better with check nodes having low degrees. This is because check nodes form their messages based on information from the v-nodes, which in turn depends upon the noisy channel output. Hence as the number of v-nodes connected to a c-node increases, the reliability of the message it produces could decrease. However due to the bipartite nature of the graph, whenever the v-node degree increases, the c-node degree also increases. Irregular LDPC codes introduced by Luby et.al. [54] help balance these conflicting requirements. In [54], the check node degree is kept as uniform as possible while the variable node degree is non-uniform. As an example, [54] defines an irregular code known as Code 14 which has $d_c = 14$ and variable nodes with degrees 5, 6, 21 and 23.

Designing a good LDPC code requires the following questions to be answered

- Given an irregular Tanner graph with a maximum d_v and d_c , what is the best degree distribution?
- How many of the v-nodes should be degree $d_v, d_{v-1}, d_{v-2}, \dots$ nodes?
- How many of the c-nodes should be degree d_c, d_{c-1}, \dots nodes

These questions are answered using *density evolution* [55], which is the process of tracking the evolution of the message distribution during belief propagation. For any LDPC code, there is a worst case channel parameter called the *threshold* such that the message distribution during belief propagation evolves in such a way that the probability of error converges to zero as the number of iterations tends to infinity. Density evolution is used to find the degree distribution pair (λ, ρ) that maximizes this threshold. Using density evolution, Richardson et. al. have identified a rate 1/2 code with which is 0.06 dB away from Shannon capacity.

Density evolution has been used to design a rate 1/2 code which is 0.0045 dB away from capacity in [13].

2.2.4 Encoding LDPC Codes

A linear block code is encoded by multiplying the message vector with the generator matrix **G**. A common method for finding **G** from **H** is to make the code systematic by adding rows and exchanging columns to get the **H** matrix in the form $\mathbf{H} = [\mathbf{P}^T \mathbf{I}]$ which gives $\mathbf{G} = [\mathbf{I} \mathbf{P}]$. However, the result of the row reduction is a non-sparse **P** matrix. The multiplication $\mathbf{c} = [\mathbf{u} \mathbf{u} \mathbf{P}]$ is therefore very complex. This is especially problematic since we are interested in large code lengths (> 10⁵).

Richardson et. al. [56] show that even for large n, the encoding complexity can be a (almost) linear function of n. Using only row and column permutations, **H** is converted to an approximately lower triangular matrix. Since only permutations are used, **H** is still sparse. The resulting encoding complexity is almost linear as a function of n.

A more elegant code construction termed *extended-irregular repeat accumulate code* (Ex-IRA) was proposed by Yang et.al. in [57]. The parity check matrix is given by

$$\mathbf{H} = \begin{bmatrix} \mathbf{H_1} & \mathbf{H_2} \end{bmatrix}. \tag{2.37}$$

 $\mathbf{H_1}$ is sparse and $\mathbf{H_2}$ is given by



Figure 2.7: Encoder for the Ex-IRA type code.

Note that H_2 has M - 1 degree 2 variable nodes. This stems from the fact that the optimal v-node degree distribution (for r > 1/2) usually implies that the number of degree 2 variable nodes be greater than M - 1. The generator matrix for a systematic code is given by

$$\mathbf{G} = [\mathbf{I} \quad \mathbf{H}_1^{\mathbf{T}} \mathbf{H}_2^{-\mathbf{T}}]. \tag{2.40}$$

 $\mathbf{H_2^{-T}}$ can be implemented with a differential encoder. The encoder structure hence simply involves multiplication by a sparse matrix followed by differential encoding as shown in Fig. 2.7.

Due to the relative ease with which these codes can be encoded, they find application is popular standards like DVB-S2 and WiMAX. The DVB-S2 standard specifies two blocklengths for the LDPC code: normal (n =64800) and short (n =16200). 11 different code rates are specified with the normal blocklengths and 10 code rates are specified for the short blocklengths. The WiMAX LDPC code uses blocklengths ranging from 576 bits to 2304 bits, in increments of 96 bits. The code rates are 1/2, 2/3 (A and B), 3/4 (A and B) and 5/6. We present select error rate curves for the LDPC codes used in the these standards. Interested readers are referred to [58] and [48] for details on the DVB-S2 and WiMAX LDPC codes.

2.3 Turbo Codes

Turbo codes (parallel concatenated convolutional codes) were introduced by Berrou et. al in [11]. The discovery of turbo codes was remarkable for several reasons: 1) Turbo codes turned previous existing code design principles on their head. Instead of designing very complex codes, with elaborate decoding algorithms, turbo codes obtain performances close to



Figure 2.8: Frame error rate performance of the n = 64800 (normal frame) DVB-S2 LDPC code. The decoder used 100 iterations of the sum-product algorithm in the log-domain.

the Shannon limit using relatively simply constituent codes. For instance, the original turbo code is a rate 1/2 code, obtained by concatenating two constraint length 5 RSC codes using an interleaver of length 65536 bits and performs within 0.7 dB of the Shannon limit. 2) Turbo codes introduced a very effective, sub-optimal approach to decoding i.e. the *turbo principle*, a process which involves iterative exchange of extrinsic information between different receiver modules. The turbo principle has now become a standard feature in modern communication systems. 3) Turbo codes triggered a renewed interest in the field of error correcting codes, thereby resulting in significant advances in code design which allow signaling close to the information theoretic limits, at reduced complexity.

2.3.1 Encoding Turbo Codes

The turbo encoder consists of two constituent recursive systematic convolutional (RSC) codes connected using an interleaver and is shown in Fig. 2.12. The information bits are fed to the first RSC encoder, which generates the corresponding check bits. The interleaver



Figure 2.9: Frame error rate performance of the n = 16200 (short frame) DVB-S2 LDPC code. The decoder used 100 iterations of the sum-product algorithm in the log-domain.



Figure 2.10: Bit error rate performance of the n = 2304 WiMAX LDPC code. The decoder used 100 iterations of the sum-product algorithm in the log-domain.



Figure 2.11: Frame error rate performance of the n = 2304 WiMAX LDPC code. The decoder used 100 iterations of the sum-product algorithm in the log-domain.

scrambles the ordering of bits at the input of the second encoder. The encoder output typically consists of the systematic and check bits from the first encoder and the check bits from the second encoder.

2.3.2 Decoding Turbo Codes

The turbo decoder is shown in Fig. 2.13. The decoding starts with the upper MAP decoder producing estimates of the data bits, based on channel observations. These estimates are passed to the lower MAP decoder, which produces its own estimates. The lower MAP decoder then passes extrinsic information to the upper decoder. This process is repeated until a certain number of iterations are completed.

2.3.3 Turbo codes as a type of LDPC codes

Due to the popularity of both turbo and LDPC codes, attempts have been made to understand the relationship between the two. MacKay [59] postulated that turbo codes were in fact a class of LDPC codes. The main benefit from this undertaking is that the



Figure 2.12: A turbo encoder.



Figure 2.13: A turbo decoder.

signal processing and hardware design advances made for turbo codes can now be utilized for LDPC codes and vice versa, i.e. turbo codes can be decoded with a LDPC decoder, which is highly parallel. The first step in understanding turbo codes as instances of LDPC codes would be to identify their generator and parity check matrices. Recently Jiang et. al have come up with a generalized description of serially concatenated turbo codes using generator and parity check matrices [60]. More recently Nimbalker et. al. [61] have developed a "turbolike" decoding algorithm for LDPC codes whose **H** matrices display partial dual-diagonal structures similar to (2.38).

2.4 Chapter Summary

The field of error correction coding has seen significant advances in the last decade. The goal of this chapter was to review some binary codes that are widely used in commercial communication systems.

The relative ease with which convolutional codes can be encoded and the availability of computationally feasible, optimal decoding algorithms have led to the widespread use of convolutional codes in various communication systems. For instance, the Voyager 2 mission employed a (2, 1, 6) Odenwalder convolutional code [62] with generators [133, 171]. The Galileo mission used a (4, 1, 14) convolutional code with generators [255, 233] [49]. GSM uses a (2, 1, 5) NRC code with generators [31, 33]. IEEE 802.16 uplink specifies mandatory coding using (2, 1, 6) convolutional code with generators [171, 133].

While convolutional codes are relatively simple to encoder and decode, stand alone convolutional codes do not perform close to the Shannon limit. In practice, convolutional codes are often concatenated with an outer code such as the Reed Solomon code to get performance close to the Shannon limit. Well designed turbo and LDPC codes allow one to signal still closer to the capacity limits.

High data rate and low cost requirements make it necessary to design codes with low complexity encoding/ decoding, without incurring a significant loss in performance. As seen earlier, the encoding in LDPC codes can be simplified using Ex-IRA type codes. Simplifications are also possible in the sum-product algorithm [63] for LDPC codes or the BCJR

algorithm for turbo codes [64], [65]. Recently, a class of codes known as *irregular repeat accumulate* [34] codes have generated considerable interest due to their linear time encoding/ decoding complexity. IRA codes have been shown to have impressive performance for MIMO systems [66], noncoherent orthogonal FSK [67] and continuous phase modulation [35].

Chapter 3

Modulation for Bandwidth Limited Systems

The ever increasing demand for bandwidth spawned by a wide variety of next generation wireless services places considerable strains on the available radio spectrum. Problems arising from spectral congestion (eg. adjacent channel interference) can be alleviated by using bandwidth efficient modulation techniques such as *continuous phase modulation* (CPM) [15]. With CPM, the signal phase is constrained to vary continuously from one symbol interval to the next. This phase continuity results in some very desirable spectral properties such as *small side lobes*, which in turn helps reduce the out-of-band interference. Another benefit of CPM is its *constant signal envelope* which makes it suitable for systems requiring power efficient amplifiers. The above mentioned advantages and the fact that CPM signals can be demodulated quite reliably using low-complexity detectors have resulted in the modulation's widespread use, as evidenced by the existence of 2 billion GSM phones ¹ and 1 billion Bluetooth devices ².

The goal of this chapter is to provide an overview of CPM. The chapter begins with a review of BPSK and QPSK. Next, a variation of QPSK with less abrupt phase variations, known as offset QPSK (OQPSK) is described. This is followed by a description of the popular minimum shift keying (MSK) which is both a variant of OQPSK and a specific type

¹http://news.soft32.com/two-billion-gsm-customers-are-around-the-world_1605.html

²http://www.therawfeed.com/2006/11/big-number-one-billion-bluetooth.html



Figure 3.1: Binary phase shift keying (BPSK) modulation. x(t) is the BPSK waveform obtained by modulating the carrier $\cos(2\pi f_c t)$ with the symbol sequence **a**.

of CPM. Rimoldi's well known and widely used *tilted phase representation* of CPM [68] is reviewed. The chapter concludes with a discussion on the spectral characteristics of CPM signals.

3.1 Binary and Quadrature Phase Shift Keying

Consider a binary source that generates a sequence of symbols $\mathbf{a} \in \{\pm 1\}$ every T_b seconds. The bandpass BPSK signal in the interval $kT_b \leq t \leq (k+1)T_b$ for k = 0, 1, 2, ... can be represented as

$$x(t) = \sqrt{2P_x} \cos\left(2\pi f_c t + \varphi(t)\right), \qquad (3.1)$$

where $\varphi(t) = 0$ or 180 (degrees) corresponding to an input symbol of +1 or -1 respectively, P_x is the symbol power and f_c is the carrier frequency. Fig. 3.1 shows the carrier $\cos(2\pi f_c t)$ by the modulated by an input sequence **a** and the resulting BPSK waveform x(t). The signal phase undergoes 180 degree shifts each time the input symbol changes. While BPSK is a simple and reliable modulation format, its main drawback is that it is not especially bandwidth efficient. The poor bandwidth efficiency of BPSK led to the search for modulation schemes that exhibit better spectral efficiency, without incurring a degradation in BER. These goals can be achieved using *quadrature phase shift keying* (QPSK). In QPSK, a pair of antipodal (i.e. BPSK) signals are transmitted using two orthogonal carriers. Due to their orthogonality, the bits modulating the two carriers can be recovered separately at the receiver. The BER is hence the same as in BPSK, but the spectral efficiency is now 2 bps/Hz instead of 1 bps/Hz for BPSK.

In QPSK, the input binary sequence consisting of symbols $a_k \in \{\pm 1\}$ for k = 0, 1, ... with bit rate $1/T_b$ is separated into two streams \mathbf{a}_I and \mathbf{a}_Q , corresponding to the even and odd numbered bits in the input bit sequence respectively. \mathbf{a}_I and \mathbf{a}_Q modulate the in-phase and quadrature components of the carrier such that the QPSK signal during $kT \leq t \leq (k+1)T$ can be represented as [69]

$$x(t) = \frac{1}{\sqrt{2}} a_I(t) \cos\left(2\pi f_c t + \frac{\pi}{4}\right) + \frac{1}{\sqrt{2}} a_Q(t) \sin\left(2\pi f_c t + \frac{\pi}{4}\right).$$
(3.2)

where $T = 2T_b$ is the symbol period. Fig. 3.2 shows the QPSK signal x(t) generated by modulating the I and Q carriers with \mathbf{a}_I and \mathbf{a}_Q respectively. Because the timing in the in-phase and quadrature bit streams are aligned, a phase change can occur at intervals of $2T_b$. This phase shift can be 0 if both \mathbf{a}_I and \mathbf{a}_Q remain unchanged, ± 90 if only one of them changes and 180 if both \mathbf{a}_I and \mathbf{a}_Q change. It is also possible to rewrite (3.2) as

$$x(t) = \sqrt{2P_x} \cos\left(2\pi f_c t + \varphi(t)\right), \qquad (3.3)$$

where $\varphi(t) = 0, \pm 90$ or 180.

A variation of QPSK known as offset QPSK (OQPSK) can be obtained by a shift or offset in the relative alignment of the in-phase and quadrature bit streams by a factor of T_b . Due to the offset, a_I and a_Q cannot change simultaneously. In non-offset QPSK a symbol transition across the QPSK constellation (shown in Fig. 3.3 using dotted arrows) is possible every T seconds and can be from any constellation point to any other constellation point (including the point itself). On the other hand, with OQPSK, a transition is possible every T_b seconds and must be from each point to either a nearest neighbor or itself (shown in Fig.



Figure 3.2: Quadrature phase shift keying (QPSK) modulation. x(t) is the QPSK waveform obtained by modulating $\cos\left(2\pi f_c t + \frac{\pi}{4}\right)$ and $\sin\left(2\pi f_c t + \frac{\pi}{4}\right)$ with \mathbf{a}_I and \mathbf{a}_Q respectively. \mathbf{a}_I and \mathbf{a}_Q are time-aligned.

3.3 using a dashed arrow). This eliminates any 180° phase shifts in the OQPSK waveform as seen in Fig. 3.4.

3.2 Minimum Shift Keying

The MSK signal during $kT \le t \le (k+1)T$ can be written as [69]

$$x(t) = a_I(t)\cos\left(\frac{\pi t}{2T_b}\right)\cos\left(2\pi f_c t\right) + a_Q(t)\sin\left(\frac{\pi t}{2T_b}\right)\sin\left(2\pi f_c t\right).$$
(3.4)

 \mathbf{a}_I and \mathbf{a}_Q are the even and odd bit streams which have a relative offset of T_b . MSK can hence be viewed as OQPSK with *sinusoidal pulse shaping*. Fig. 3.5 shows the MSK waveform from (3.4). In OQPSK and QPSK, \mathbf{a}_I and \mathbf{a}_Q directly multiply the carriers which results in abrupt phase changes. However, in MSK the even and odd bit streams undergo pulse shaping due to which there are no abrupt changes in the signal phase and amplitude. The MSK signal



Figure 3.3: Constellation diagram for QPSK modulation. Dotted arrows denote possible transitions from (1, 1) for non-offset QPSK which can occur every T seconds. Dashed arrow denotes possible transition from (1, 1) for OQPSK which can occur every T_b seconds.



Figure 3.4: OQPSK modulation. x(t) is the QPSK waveform obtained by modulating $\cos\left(2\pi f_c t + \frac{\pi}{4}\right)$ and $\sin\left(2\pi f_c t + \frac{\pi}{4}\right)$ with \mathbf{a}_I and \mathbf{a}_Q respectively. \mathbf{a}_I and \mathbf{a}_Q have an offset of T_b .

can also be written as [69]

$$x(t) = \sqrt{2P_x} \cos\left(2\pi f_c t + b_k(t)\frac{\pi t}{2T_b} + \delta_k\right),\tag{3.5}$$

where $b_k = -a_I a_Q$ and $\delta_k = 0$ or π for $a_I = 1$ or -1 respectively. It is quite clear from (3.5) that the MSK signal is also a binary *frequency shift keying* (FSK) signal with tones

$$f_{1} = f_{c} + \frac{1}{4T_{b}},$$

$$f_{2} = f_{c} - \frac{1}{4T_{b}}.$$
(3.6)

The separation between the two frequencies if $\Delta_f = \frac{1}{2T_b}$, which is also the minimum frequency spacing required for two signals to be coherently orthogonal. The *continuous phase FSK* (CPFSK) signal from (3.5) is hence known as *minimum* shift keying. MSK can be coherently detected using quadrature matched filters which base their bit decisions after observing the signal over $2T_b$ seconds. For such receivers, the bit error rate of MSK is the same as BPSK, QPSK and OQPSK. Alternatively, MSK can also be demodulated as a coherent orthogonal FSK signal with bit decisions made every T_b seconds, in which case the bit error rate is 3 dB worse than BPSK.

Alternatively, the MSK signal can also be represented as [70]

$$x(t) = \sqrt{2P_x} \cos\left(2\pi f_c t + \frac{\pi a_k}{2T_b}t + \delta_k\right),\tag{3.7}$$

where δ_k is a variable that is required for phase continuity at the bit transitions and in given by

$$\delta_k = \delta_{k-1} + (a_{k-1} - a_k) \frac{\pi k}{2}.$$
(3.8)

The baseband MSK phase is hence

$$\varphi(t, \mathbf{a}) = \delta_k + \frac{\pi a_k}{2T_b} t, \qquad (3.9)$$

The time varying MSK phase $\varphi(t, \mathbf{a})$ for all possible input symbols plotted as a function of time is the *phase trajectory* and is shown in Fig. 3.6. The MSK phase increases or decreases linearly by a factor of 90 degrees every bit interval depending upon the input bit a_k during



Figure 3.5: Minimum shift keying modulation. The MSK waveform x(t) is generated by modulating the I and Q carriers with the pulse shaped even and odd data streams respectively.



Figure 3.6: Phase trajectory for MSK.



Figure 3.7: Time variant phase trellis for MSK.

that interval. The phase trajectory expressed as modulo 2π gives the MSK trellis shown in Fig. 3.7.

Fig. 3.8 shows the power spectral densities for BPSK, QPSK and MSK. MSK exhibits significantly smaller side-lobes relative to QPSK and BPSK. The small side lobes are a direct consequence of *avoiding abrupt phase transitions*. MSK is simply a particular example of a wider class of modulation which exhibit phase continuity known as *continuous phase modulation* (CPM).

3.3 Continuous Phase Modulation

The MSK phase trajectory in Fig. 3.6 can also be obtained by expressing the signal phase over the interval $kT \le t \le (k+1)T$ as [14]

$$\varphi(t, \mathbf{a}) = \frac{\pi}{2} \sum_{i=0}^{k-1} a_i + \pi a_k \left(\frac{t - kT_b}{T_b}\right). \tag{3.10}$$



Figure 3.8: Power spectral densities in dB of BPSK, QPSK and MSK.

This can also be written as

$$\varphi(t, \mathbf{a}) = \frac{\pi}{2} \sum_{i=0}^{k-1} a_i + \pi a_k q(t - kT_b)$$

= $\frac{\pi}{2} \sum_{i=0}^{k-1} a_i + \pi a_k \int_0^t g(\tau - kT_b) d\tau$
= $\pi \sum_{i=0}^{k-1} a_i \int_0^t g(\tau - iT_b) d\tau + \pi a_k \int_0^t g(\tau - kT_b) d\tau,$ (3.11)

where

$$q(t) = \int_0^t g(\tau) d\tau.$$
(3.12)

g(t) and q(t) are shown in Fig. 3.9. The CPFSK phase from (3.11) can also be written as

$$\varphi(t, \mathbf{a}) = 2\pi h \sum_{i=0}^{k} a_i q(t - iT_b), \qquad (3.13)$$

where h is the modulation index. When expressed as (3.13), the MSK signal becomes a special case of continuous phase modulation (CPM), i.e. MSK is binary CPM with h = 1/2 and rectangular pulse shaping (Fig. 3.9).



Figure 3.9: Rectangular pulse shape g(t) and its integral q(t).

In general, the bandpass M-ary CPM signal during $kT \le t \le (k+1)T$ can be written as

$$x(t, \mathbf{a}) = \sqrt{2\mathcal{E}_s/T} \cos\left(2\pi f_c t + \varphi(t, \mathbf{a})\right), \qquad (3.14)$$

where T is the symbol period, \mathcal{E}_s is the symbol energy and **a** is the symbol sequence. Depending upon the choice of M, h and the pulse shape g, a large variety of CPM signals can be generated. Besides the rectangular (REC) pulse shape shown in Fig. 3.9, the raised cosine (RC) Fig. 3.10 and Gaussian (GFSK) pulse shapes Fig. 3.11 are also widely used. If g(t) = 0 for t > T, the CPM signal is called *full response* CPM. If $g(t) \neq 0$ for t > T, the CPM signal is called *full response* CPM. If $g(t) \neq 0$ for t > T, the CPM signal is called *full response* CPM. If $g(t) \neq 0$ for t > T, the CPM signal is called *full response* CPM. If $g(t) \neq 0$ for t > T, the CPM signal is called *full response* CPM. If $g(t) \neq 0$ for t > T, the CPM signal is called *full response* CPM. If $g(t) \neq 0$ for t > T, the CPM signal is called *full response* CPM. If $g(t) \neq 0$ for t > T, the CPM signal is called *full response* CPM. If $g(t) \neq 0$ for t > T, the CPM signal is called *full response* CPM. If $g(t) \neq 0$ for t > T, the CPM signal is called *full response* CPM. If $g(t) \neq 0$ for t > T, the CPM signal is called *full response* CPM. If $g(t) \neq 0$ for t > T, the CPM signal is called *full response* CPM. If $g(t) \neq 0$ for t > T, the CPM signal is called *full response* CPM and L > 1 for partial response CPM. The phase continuity constraint results in the CPM signal having memory. Partial response signaling introduces additional memory (ISI), which in turn results in smoother phase transitions from one symbol interval to another. As shown in [15], partial response signaling results in a more compact power spectrum. The phase from (3.13) can be generalized to partial response CPM by as follows [14]

$$\varphi(t, \mathbf{a}) = 2\pi h \sum_{i=0}^{\infty} a_i q(t - iT). \qquad (3.15)$$

While the symbol vector **a** has indices that range from 0 to $N_a - 1$, the limits in the summation of (3.15) are infinite to capture the effect of possible preceding and subsequent



Figure 3.10: Raised cosine pulse shape g(t) and its integral q(t).



Figure 3.11: Gaussian pulse shape g(t) and its integral q(t).

coded frames. If there is a sufficient guard time between subsequent frames, as in a packet radio system, then the limits of this summation can be narrowed to $0 \le i \le N_a - 1$. The time varying CPM signal phase can now be written as

$$\varphi(t, \mathbf{a}) = \pi h \sum_{i=0}^{k-L} a_i + 2\pi h \sum_{i=k-L+1}^{k} a_i q(t - iT)$$

= $\theta_k + 2\pi h \sum_{i=k-L+1}^{k} a_i.$ (3.16)

As with MSK, CPM signals can be represented using a trellis. The trellis representation assumes that the modulation index in a rational number of the form $h = m_h/p_h$, where m_h and p_h are relatively prime integers. The trellis for full response CPM has the following p_h states [14]

$$\Theta = \left\{ 0, \frac{\pi m_h}{p_h}, ..., \frac{(p_h - 1)\pi m_h}{p_h} \right\},$$
(3.17)

when m_h is even and the following $2p_h$ states

$$\Theta = \left\{ 0, \frac{\pi m_h}{p_h}, \dots, \frac{(2p_h - 1)\pi m_h}{p_h} \right\},$$
(3.18)

when m_h is odd. Since the partial response CPM introduces additional memory, the number of states in the trellis would also be higher to account for this ISI. The number of states for partial response CPM is [14]

$$N_s = \begin{cases} p_h M^{L-1} & \text{even } m_h \\ 2p_h M^{L-1} & \text{odd } m_h. \end{cases}$$

$$(3.19)$$

3.4 Tilted Phase Representation of CPM

Observe that the MSK phase trellis in Fig. 3.7 is time variant, since the trellis states for the even symbol intervals are different from those for the odd symbol intervals. In general, the CPM phase represented by (3.16) yields a time variant trellis. However, if one measures the phase relative to the lowest phase in the trajectory (Fig. 3.6), then the resulting MSK phase trellis is time invariant as shown in Fig. 3.12. The time invariant MSK trellis was introduced in [71]. Rimoldi generalized the time invariant representation for arbitrary CPM signals in [68] and described the resulting phase trellis (trajectory) as a *tilted phase* trellis (trajectory).

The CPM tilted phase is given by [68]

$$\psi(t, \mathbf{a}) = \varphi(t, \mathbf{a}) + \pi h(M - 1)t/T.$$
(3.20)

From [68], the tilted phase (taken modulo 2π) can be written as

$$\psi(t, \mathbf{a}) = \left[2\pi h\phi_k + 4\pi h\sum_{i=0}^{L-1} \alpha_{k-i}q(t - (k-i)T) + W(t - kT)\right] \mod 2\pi,$$
(3.21)

where

$$W(t - kT) = \frac{\pi h(M - 1)(t - kT)}{T} - 2\pi h(M - 1) \sum_{i=0}^{L-1} q(t - (k - i)T) + (L - 1)(M - 1)\pi h, \qquad (3.22)$$



Figure 3.12: Time invariant (*tilted*) phase trellis for MSK.

and $\mathbf{a} \in \{0, 1, ..., M - 1\}$. The accumulated phase is

$$\phi_k = \sum_{i=0}^{k-L} a_i \mod p_h.$$
 (3.23)

An important consequence of the CPM tilted phase is that the modulation can be decomposed into a continuous phase encoder (CPE) followed by a memoryless modulator (MM). The MM generates the consequently the CPM signal waveform. From (3.21), MM output can be completely described be specifying $[a_k, ..., a_{k-L+1}, \phi_k]$ which is in fact the MM input. The CPE's task is to update the MM input at every interval such that

$$\phi_{k+1} = [\phi_k + a_{k-L+1}] \mod p_h. \tag{3.24}$$

CPM decomposed into a CPE followed by a MM is shown in Fig. 3.13. Observe that the CPE is a linear (over modulo p_h) recursive convolutional encoder.



Figure 3.13: CPM as a continuous phase encoder (CPE) followed by a memoryless modulator (MM).

3.5 Bandwidth of CPM Signals

The power spectral density (PSD) $\Phi_x(f)$ of the CPM signal is given in [14]. Using the PSD, the 99% power bandwidth B_{99} is found as

$$\int_{-B_{99}/2}^{B_{99}/2} \Phi_x(f) df = 0.99 \int_{-\infty}^{\infty} \Phi_x(f) df.$$
(3.25)

The 99% power bandwidth is a function of M, h, pulse shape and the symbol rate $R_s = 1/T$. The bandwidth efficiency is often quantified by the two-sided normalized bandwidth, defined to be $B = B_{99}T_b$ Hz/bps, where $T_b = T/\log_2 M$. Fig. 3.14 shows the fractional out-of-band power i.e. the fraction of the signal power lying outside a certain frequency, for MSK, QPSK and BPSK. For MSK, B = 1.18 as against B = 8.2 for QPSK and B = 17 for BPSK, which makes MSK more suitable for bandwidth constrained applications. Fig. 3.15 shows the fractional out-of-band power for binary full response CPM with rectangular pulse shaping (CPFSK) at different values of h. Observe that B decreases with decreasing h. Fig. 3.16 shows the CPFSK 99% power bandwidth as a function of h, at different M. Observe that for a particular h, the bandwidth increases with the alphabet size M

The bandwidth efficiency of CPM can be further improved by partial response signaling, i.e. increasing the width of the pulse shape. Fig. 3.17 shows the fractional out-of-band power for quaternary CPM with *L*RC pulse shaping, where L = 1, 2 and 4. As *L* increases, *B* decreases, thus making the system more spectrally efficient.



Figure 3.14: Fractional out-of-band powers in dB for BPSK, QPSK and MSK versus normalized frequency fT_b .



Figure 3.15: Fractional out-of-band power for binary CPFSK at different values of h.



Figure 3.16: Normalized 99% power bandwidth as a function of h, for different CPFSK alphabet sizes.



Figure 3.17: Fractional out-of-band power for quaternary CPM with RC pulse shaping at different values of L.



Figure 3.18: Fractional out-of-band power for uncoded MSK and coded MSK with r = 1/2.

3.6 Bandwidth of Coded CPM Signals

As is well known, channel coding often increases the energy efficiency at the expense of bandwidth efficiency. As an example, an MSK signal with a rate 1/2 code i.e r = 1/2 has a B = 2.36 as against B = 1.18 for the uncoded signal. This is shown in Fig. 3.18.

In bandwidth limited systems, coding must be done without bandwidth expansion i.e. B must remain unchanged. However, with a rate r code, $B_{coded} = B/r$. Suppose we require B = 1.18 while using a rate 1/2 code. Hence relative to uncoded MSK, we can either lower h or increase L or both (M as well as the pulse shape can also be varied, but for ease of exposition we assume M = 2 and REC pulses). This is illustrated in Fig. 3.19, which shows the fractional out-of-band power for uncoded MSK, rate 1/2 coded CPM with M = 2, h = 0.125, 1REC and rate 1/2 coded CPM with M = 2, h = 0.5 and 5REC. Lowering h results in the signal begin nonorthogonal, while increasing L increases the ISI.



Figure 3.19: Fractional out-of-band power for uncoded MSK, coded CPM with r = 1/2. The uncoded and coded signals have the same 99% bandwidth.

3.7 Chapter Summary

In this chapter, a class of non-linear, constant envelope modulation known as continuous phase modulation was discussed. The memory in the CPM signal causes phase continuity, which in turn yields a well behaved power spectrum. Due the memory, CPM can be represented as a finite state machine. Hence the optimum detector for CPM is one that in addition to having perfect estimates of the channel state information, including the reference signal phase, performs a sequence based detection on the trellis representing the CPM signal. The optimal detector requires that the modulation index be rational, and its value greatly impacts the complexity. Sequence detection can be performed using either the Viterbi algorithm or the BCJR algorithm. Rimoldi's tilted phase representation of CPM allows for a time invariant trellis, which considerably simplifies sequential detection. Detection of CPM signals is discussed in detail in Chapter 4.

The spectral characteristics of the CPM signal were also investigated. It is seen that to keep the spectral efficiency unchanged for a coded system, the modulation parameters need
to be scaled (by decreasing the spacing between the tones, increasing the signal memory or changing the alphabet size). The nonlinear nature of CPM makes it difficult to gauge the effects of this scaling on the coded system performance. Optimizing the code and modulation parameters for CPM under spectral efficiency constraints is a challenging problem and is discussed in detail in Chapter 6.

Chapter 4

Detector Designs For CPM

When a CPM waveform (Chapter 3) is transmitted through a channel, it encounters noise, fading, interference and a *random phase shift* in the transmitted carrier [14]. The detector on receiving this degraded signal produces estimates of the transmitted symbols after some signal processing. Depending on how the detector copes with the carrier phase shift, detectors for CPM can be broadly categorized as 1) *Coherent*, 2) *Differential*, or 3) *Noncoherent*. The detector's output can either be hard or soft estimates of the transmitted symbols. Detector design for CPM is an extensively investigated topic and there exists a considerably body of work devoted to the subject [15, 20, 72, 73, 74, 21, 27, 75, 76, 77, 78, 30, 29, 79, 80].

The goal in this chapter is to describe the structure and investigate the performance of the detectors most relevant to our work. The chapter begins by introducing our system model. Next, the popular sequence based coherent detector [15] is described. In particular, we focus on a soft-output version of this detector (SO-Coherent) [29]. The error rate performance of an uncoded system using coherent detection is studied for select modulation parameters. This is followed by a description of the differential phase detector (DPD) [20]. Building upon the DPD and Fonseka's soft-decision differential phase detector with Viterbi decoding (SDDPD-VD) [81], we introduce a novel soft-output differential detector for CPM which we term the *soft-output, soft-decision differential phase detector* (SO-SDDPD). Our predilection for soft-output detectors is in part because it is a necessary requirement in modern communication systems employing powerful channel codes and sophisticated signal processing. Another important reason is that our capacity calculations require soft-outputs



Figure 4.1: Transmitter block diagram.

in the form of log-likelihood ratios (LLRs) from the detector (Chapter 5). We compare the error rate performance of SO-SDDPD with the soft-output coherent detector and conclude the chapter with a discussion on detector complexity.

4.1 System Model

4.1.1 Transmitter

The transmitter model is shown in Fig. 4.1. A vector $\mathbf{u} = [u_0, u_1, ..., u_{N_u-1}] \in \{0, 1\}^{N_u}$ of information bits is passed through a linear binary encoder to produce the codeword $\mathbf{b}' \in \{0, 1\}^{N_b}$. The code rate is $r = N_u/N_b$. The row vector \mathbf{b}' is then interleaved by multiplying by a permutation matrix $\mathbf{\Pi}$ to produce the bit-interleaved codeword $\mathbf{b} = \mathbf{b}'\mathbf{\Pi}$. The vector \mathbf{b} is arranged in a $\log_2 M \times N_a$ matrix \mathbf{B} with (i, k) element $B_{i,k} = b_{k \log_2 M+i}$. The number of M-ary symbols to be transmitted is $N_a = \lceil N_b/\log_2 M \rceil$. Each column of \mathbf{B} is mapped to one of M symbols to produce the vector $\mathbf{a} = [a_0, a_1, ..., a_{N_a-1}]$ which represents the sequence of coded symbols to be transmitted. The symbol sequence is used to create the signal phase $\varphi(t, \mathbf{a})^{-1}$ (details in Chapter 3). The transmitter generates the bandpass signal

$$x(t, \mathbf{a}) = \sqrt{2P_x} \cos\left(2\pi f_c t + \varphi(t, \mathbf{a})\right), \qquad (4.1)$$

which is then transmitted over the radio frequency (RF) channel with power P_x and center frequency f_c .

4.1.2 Channel

The modulated signal $x(t, \mathbf{a})$ is transmitted through a frequency nonselective, Rician fading channel. The effects of the channel are most conveniently described in terms of the

¹For a tilted phase representation, the CPM phase is $\psi(t, \mathbf{a})$ instead of $\varphi(t, \mathbf{a})$.

complex envelope of the signal. The complex envelope of the transmitted signal is

$$\tilde{x}(t, \mathbf{a}) = \sqrt{2P_x} \exp\left(j\varphi(t, \mathbf{a})\right).$$
(4.2)

This signal is multiplied by a complex flat-fading process $\tilde{c}(t)$ and added to white Gaussian noise $\tilde{n}_w(t)$, producing the complex envelope of the signal at the input to the receiver

$$\tilde{r}(t, \mathbf{a}) = \tilde{c}(t)\tilde{x}(t, \mathbf{a}) + \tilde{n}_w(t), \qquad (4.3)$$

where $\tilde{n}_w(t)$ is a complex white Gaussian process with power spectral density N_0 . The fading process $\tilde{c}(t)$ can be written as

$$\tilde{c}(t) = \sqrt{P_s} + \sqrt{P_d}\xi(t), \qquad (4.4)$$

where P_d is the power gain of the diffused component, P_s is the power gain of the specular component, and $\xi(t)$ is a circularly symmetric unit power complex Gaussian process. P_s and P_d are normalized such that $P_s + P_d = 1$. The Rician K-factor is given by $K = P_s/P_d$. When K = 0, the channel is said to be Rayleigh and when $K = \infty$, the channel is said to be AWGN. The process $\xi(t)$ is usually characterized by an autocorrelation function, which, as an example could be [82]

$$R_{\xi}(\tau) = J_0(2\pi f_d \tau), \qquad (4.5)$$

where J_0 is the zero-order Bessel function of the first kind and f_d is the maximum Doppler frequency shift.

4.1.3 Receiver

The receiver block diagram is shown in Fig. 4.2. The detector consists of two basic blocks 1) a *front-end*, which translates the bandpass signal to baseband and generates sufficient statistics for the transmitted symbols, 2) a *back-end* which forms decisions on the transmitted code bits, using the statistics provided by the front-end. The structure of the front-end depends on the choice of the detector (e.g. coherent, noncoherent or differential). The decisions produced by the back-end can either be hard (hard-decision detector) or soft (softdecision detector), which after deinterleaving are passed to the channel decoder. The decoder produces estimates of the data bits.



Figure 4.2: Receiver block diagram.



Figure 4.3: Receiver with soft-output coherent detection.

4.2 Coherent Detection of CPM

A coherent detector (Fig. 4.3) has accurate estimates of the carrier phase, which is obtained using carrier recovery circuitry [14] at the detector front-end. The optimum coherent detector performs sequence detection on a trellis describing the CPM modulation and its performance (especially with ML sequence detection) is very well documented in the literature [15].

4.2.1 Coherent Front-End

The coherent front-end consists of the carrier recovery circuit and a bank of complex correlators.

4.2.2 Soft-Output Coherent Detector

Trellis detection requires that the underlying modulation be represented as a finite state machine. Using Rimioldi's decomposition approach to CPM (Chapter 3), the state at time instant t = kT is given by $S_k = (a_{k-L+1}, ..., a_{k-1}, \phi_k)$, where L is the width of the CPM pulse shape g(t) and

$$\phi_k = \sum_{i=0}^{k-L} a_i \mod p_h, \tag{4.6}$$

where $h = m_h/p_h$. Note that at any given time interval, there are $p_h M^{L-1}$ possible states, with M branches emerging out of each state. Since the CPM signal is completely specified by the continuous phase encoder's output, the signal corresponding to the state transition $\{S_k = s'\} \rightarrow \{S_{k+1} = s\}$ is $x_{s' \rightarrow s}(t, [a_{k-L+1}, ..., a_{k-1}, \phi_k, a_k])$. The sufficient statistics required by the soft-output coherent detector are provided by a bank of $p_h M^L$ correlators (one for each branch in the trellis), such that the output of the correlator corresponding to $\{S_k = s'\} \rightarrow \{S_{k+1} = s\}$ is

$$\rho_{s' \to s} = \int_{kT}^{(k+1)T} r(t, \mathbf{a}) x^*_{s' \to s}(t, [a_{k-L+1}, ..., a_{k-1}, \phi_k, a_k]) dt.$$
(4.7)

Soft-output coherent detection is performed by executing the BCJR algorithm on the trellis and was first employed in a BICM-ID framework in [29, 83] (the authors use the term serially concatenated CPM (SCCPM) with iterative decoding instead of BICM-ID). The LLR can be decomposed using the BCJR algorithm as

$$z_{k} = \log \frac{P[b_{k} = 1 | r(t, \mathbf{a})]}{P[b_{k} = 0 | r(t, \mathbf{a})]}$$

=
$$\log \frac{\sum_{\mathcal{S}^{(1)}} \alpha_{k}(s') \gamma_{k+1}(s', s) \beta_{k+1}(s)}{\sum_{\mathcal{S}^{(0)}} \alpha_{k}(s') \gamma_{k+1}(s', s) \beta_{k+1}(s)},$$
(4.8)

$$\mathcal{Z}_{i,k} = \log \frac{P[B_{i,k} = 1 | r(t, \mathbf{a})]}{P[B_{i,k} = 0 | r(t, \mathbf{a})]} \\
= \log \frac{\sum_{\mathcal{S}^{(1)}} \alpha_k(s') \gamma_{k+1}(s', s) \beta_{k+1}(s)}{\sum_{\mathcal{S}^{(0)}} \alpha_k(s') \gamma_{k+1}(s', s) \beta_{k+1}(s)},$$
(4.9)

where $\mathcal{S}^{(1)}$ is the set of state transitions $\{S_k = s'\} \to \{S_{k+1} = s\}$ corresponding to $B_{i,k} = +1$, $\mathcal{S}^{(0)}$ is defined similarly for $B_{i,k} = 0$, and α , β and γ are the metrics in the BCJR algorithm. Because Chapter 2 already fully describes how to recursively calculate α and β , all that remains to completely describe the demodulator is the calculation of γ . From [10], $\gamma_{k+1}(s', s)$ can be written as

$$\gamma_{k+1}(s',s) = P\left[S_{k+1} = s | S_k = s'\right] P\left[r(t,\mathbf{a}) | (S_k \to S_{k+1}) = (s' \to s)\right].$$
(4.10)

For BICM, there is no feedback from the decoder and $P[S_{k+1} = s | S_k = s']$ cancels out from (4.9) under the assumption of equally likely symbols. The likelihood evaluates to

$$P[r(t, \mathbf{a})|(S_k \to S_{k+1}) = (s' \to s)] = \exp\left(\Re\{\rho_{s' \to s}\}/N_0\right).$$
(4.11)

The demodulator executes the BCJR algorithm producing LLR $\mathcal{Z}_{i,k}$ for each bit *i* of each symbol *k*. The LLRs are then placed into a row vector \mathbf{z} such that $z_{k \log_2 M+i} = \mathcal{Z}_{i,k}$. The vector is then deinterleaved and the resulting sequence \mathbf{z}' fed to the channel decoder for soft-decision decoding. The channel decoder uses \mathbf{z}' to form estimates of the data bits ($\hat{\mathbf{u}}$).

4.2.3 Bit Error Rates for Uncoded Systems with Coherent Detection

We present select simulation results to illustrate the performance of the SO-Coherent detector. Fig. 4.4 shows BER curves for uncoded binary CPFSK at different values of h in AWGN. Binary CPFSK with h = 1/2 is the popular MSK modulation. The error rate can be lowered by increasing h, as seen in Fig. 4.4²

We have seen in Chapter 3 that partial response signaling yields a more compact power spectrum relative to full response signaling. This implies that for a particular 99% power bandwidth B, partial response signaling would allow one to use a larger value of h, which, in turn could yield a lower BER. This is illustrated in Fig. 4.5. Binary 3 RC with h = 4/5has a 99% power bandwidth $B \approx 1.18$ (which is the 99% power bandwidth for MSK) and provides approximately a 2 dB improvement in energy efficiency over MSK. However, the detector for partial response signaling is more computationally intensive since it requires a

²Strictly speaking, the relationship between the BER and h is not truly monotonic, for instance the optimum modulation index for uncoded binary CPFSK is h = 0.715 [15]. But since the complexity of the coherent detector is proportional to p_h where $h = m_h/p_h$, this modulation index is unlikely to be used in practical systems.



Figure 4.4: BER for uncoded 2-CPFSK with different h and coherent detection in AWGN.

 $p_h M^{L-1} = 20$ state trellis instead of a 2 state trellis for MSK. 20 frame errors were logged at every \mathcal{E}_b/N_0 to generate the BER curves shown in Fig. 4.4 and Fig. 4.5.

4.3 Differential Phase Detection of CPM

Since a coherent receiver requires accurate estimates of the carrier phase, it is sensitive to phase estimation errors caused by phase-locked loops (PLL) used for carrier phase recovery. Additionally, such carrier recovery circuits add to the complexity of the receiver.

A pragmatic alternative is to circumvent the above two problems by using *differentially coherent* detectors. Differential detectors base their decisions on the phase differences calculated over multiple symbol intervals. The underlying assumption is that the random phase rotation introduced by the channel remains constant during the interval over which the phase difference is calculated. It therefore cancels out of the decision metric.

The principle of differential phase detection was applied to Gaussian Minimum Shift Keying (GMSK) in [20]. GMSK is a class of CPM using Gaussian pulse shapes and h = 0.5. Differential phase detection with decision feedback for GMSK was investigated in [72]. Korn



Figure 4.5: BER for comparison between MSK and 3 RC CPM with h = 1/2 and coherent detection in AWGN.

derives expressions for the error probability of GMSK using differential phase detection, with and without decision feedback in [73]. The detectors without feedback do not account for the inter-symbol interference (ISI) caused by partial response signaling. Instead of cancelling the ISI using decision feedback, it can be exploited for performance gains using trellis-based detection. Differential phase detection with ML sequence detection was investigated in [74], [75].

In this section, we investigate different differential phase detectors. Papers on differential phase detectors typically focus on GMSK. We have developed our system model in a more general fashion that is agnostic to the choice of the pulse shaping function. First, the low-complexity differential phase detector (DPD) from [73] is described. Next, ML sequence detection using Fonseka's soft-decision differential phase detection (SDDPD) [81] and our proposed innovation the soft-output SDDPD (SO-SDDPD) are described. Bit error rate simulations are used to compare the DPD and the SO-SDDPD. The SO-SDDPD is applied to Bluetooth [16] detection and results are presented demonstrating significant improvements in the error rate and throughput over conventional Bluetooth receivers.



Figure 4.6: Receiver with hard-decision differential phase detection.

4.3.1 Differential Front-End

As shown in Fig. 4.6, the received signal $r(t, \mathbf{a})$, whose complex envelope is given by (4.3), is passed through a bandpass receive filter that removes the out-of-band noise. It is assumed that the equivalent noise bandwidth of the filter, B_n , is greater than the signal's 99% power bandwidth, and that the passband of the filter is relatively flat. Thus, the informationbearing portion of the signal is essentially undistorted by the filter. Under this assumption, the signal $y(t, \mathbf{a})$ at the output of the bandpass filter has complex envelope

$$\widetilde{y}(t, \mathbf{a}) = \widetilde{c}(t)\widetilde{x}(t, \mathbf{a}) + \widetilde{n}_{c}(t)
= \sqrt{2P_{y}(t)} \exp\left(j\phi(t, \mathbf{a})\right),$$
(4.12)

where $\tilde{n}_c(t)$ is complex lowpass noise with power $P_n = N_0 B_n$, $P_y(t)$ is the power of the received signal, and $\phi(t, \mathbf{a})$ is the phase of the received signal. Due to fading and noise, $P_y(t)$ is time-varying, even though the transmitted signal power P_x is constant.

The received phase may be expressed as

$$\phi(t, \mathbf{a}) = \varphi(t, \mathbf{a}) + \upsilon(t), \tag{4.13}$$

where v(t) is the phase noise as derived in [73]

$$v(t) = \arctan \frac{\mu'_Q(t, \mathbf{a})}{\mu'_I(t, \mathbf{a}) + \sqrt{2\rho}}, \qquad (4.14)$$

where

$$\rho = \frac{P_s}{P_d + P_n},$$

$$\mu'_I(t, \mathbf{a}) = \Re \left\{ \mu'(t, \mathbf{a}) \right\},$$

$$\mu'_Q(t, \mathbf{a}) = \Im \left\{ \mu'(t, \mathbf{a}) \right\},$$

$$\mu'(t, \mathbf{a}) = \frac{1}{P_d + P_n} \mu(t, \mathbf{a}) \exp \left(-j\varphi(t, \mathbf{a}) \right),$$

$$\mu(t, \mathbf{a}) = \sqrt{P_d} \xi(t) \exp \left(j\varphi(t, \mathbf{a}) \right) + \tilde{n}_c(t),$$
(4.15)

where $\Re\{z\}$ and $\Im\{z\}$ are the real and imaginary parts of z, respectively.

As is shown in Fig. 4.6, the output of the bandpass filter $y(t, \mathbf{a})$, whose complex envelope is given by (4.12), is multiplied by a delayed and phase shifted version of itself. The resulting bandpass signal is

$$d(t, \mathbf{a}) = -2\sqrt{P_y(t)P_y(t-T)}\cos\left(2\pi f_c t + \phi(t, \mathbf{a})\right)\sin\left(2\pi f_c(t-T) + \phi(t-T, \mathbf{a})\right). \quad (4.16)$$

Using the well known trigonometric identity $\cos \alpha \sin \beta = \frac{1}{2} \sin(\alpha + \beta) - \frac{1}{2} \sin(\alpha - \beta)$, and dropping the double frequency term, $d(t, \mathbf{a})$ after low pass filtering becomes

$$d_b(t, \mathbf{a}) = \sqrt{P_y(t)P_y(t-T)} \times \sin\left(\phi(t, \mathbf{a}) - \phi(t-T, \mathbf{a}) + 2\pi f_c T\right).$$
(4.17)

The signal is then sampled at time t = (k+1)T, resulting in

$$d_b((k+1)T, \mathbf{a}) = \sqrt{P_y(kT+T)P_y(kT)} \times \sin(\phi(kT+T, \mathbf{a}) - \phi(kT, \mathbf{a}) + 2\pi f_c T). \quad (4.18)$$

As is common in the literature [72], we assume that the design parameter f_cT is selected to be an integer ³, in which case the received phase differences can be found by computing

$$\Delta \phi_k = \angle d_b((k+1)T, \mathbf{a})$$

= $\phi(kT + T, \mathbf{a}) - \phi(kT, \mathbf{a}),$ (4.19)

where $\angle \kappa \sin z = z$ for any arbitrary constant κ .

³Actually, all that is required is that the design parameter $f_c T$ be known, in which case the term $2\pi f_c T$ can be subtracted from $\angle d_b((k+1)T, \mathbf{a})$

4.3.2 Hard-Decision Differential Phase Detector

In traditional differential phase detectors (DPD) such as those described in [20], [72] and [73] hard symbol decisions are made by comparing $\Delta \phi_k$ as given in (4.19) to some threshold. While this detector is adequate for full-response signaling, it is unable to adequately compensate for the ISI induced by the partial response CPM signaling. Additionally, the DPD phase produces hard estimates of the modulated symbols.

4.3.3 Soft-Output, Soft-Decision, Differential Phase Detector

The memory in the modulation can be exploited in a manner analogous to the optimal coherent CPM demodulator. As the memory in the modulation can be accurately modelled as a finite-state machine, the Viterbi algorithm can be used to perform maximum-likelihood demodulation even when driven by phase differences. Maximum-likelihood detection using soft phase differences (4.19) and the Viterbi algorithm was proposed by Fonseka in [81], where it is termed *soft-decision, differential phase detector with Viterbi decoding (SDDPD-VD)*. The underlying structure is that of a DPD, but instead of making symbol-by-symbol hard decisions, the SDDPD-VD exploits the CPM induced ISI to produce the ML estimate of the transmitted symbol sequence. However, as with symbol-by-symbol DPD, the bit decisions made with SDDPD-VD are hard.

By replacing the Viterbi algorithm in the SDDPD-VD algorithm with the BCJR (or MAP) algorithm of [10], the demodulator will output the log-likelihood ratios of the code bits

$$\mathcal{Z}_{i,k} = \log \frac{P[B_{i,k} = 1 | \boldsymbol{\Delta} \phi]}{P[B_{i,k} = 0 | \boldsymbol{\Delta} \phi]},$$
(4.20)

where $\Delta \phi = {\Delta \phi_k, 0 \leq k \leq N_a - 1}$ is the received sequence of phase differences. In this paper, we use the term *soft-output*, *soft-decision*, *differential phase detector (SO-SDDPD)* (Fig. 4.7) to describe our formulation of the soft-differential detector based on the BCJR algorithm, which is the subject of the present subsection and was previously proposed in [84].

Both the Viterbi and BCJR algorithms require that the underlying modulation be described in terms of a finite-state machine. For differential detection, the finite states manifest



Figure 4.7: Receiver with soft-output, soft-decision differential phase detection.

themselves as a finite set of *transmitted* phase differences

$$\Delta \varphi_k(\mathbf{a}) = \varphi(kT + T, \mathbf{a}) - \varphi(kT, \mathbf{a})$$

= $2\pi h \sum_{i=-\infty}^{\infty} a_{k-i} p_i,$ (4.21)

and

$$p_i = \int_{iT}^{iT+T} g(t) dt.$$
 (4.22)

When the pulse g(t) is not time limited, the number of distinct phase differences will generally be (countably) infinite. However, for pulses of interest the CPM induced ISI terms are generally limited to only extend from Z_p past symbols to Z_f future symbols. In this case, (4.21) can be written as

$$\Delta \varphi_k(\mathbf{a}) = 2\pi h \sum_{i=-Z_f}^{Z_p} a_{k-i} p_i, \qquad (4.23)$$

and $\Delta \varphi_k(\mathbf{a})$ will assume one of only $M^{Z_p+Z_f+1}$ values. If the Gaussian pulse shape g(t) with 3 dB bandwidth B_g

$$g(t) = \frac{1}{2T} \left[Q \left(-cB_g t \right) - Q \left(-cB_g (t-T) \right) \right], \qquad (4.24)$$

where c = 7.546 and $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty \exp(-y^2/2) dy$, or the RC pulse shape

$$g(t) = \begin{cases} \frac{1}{2LT} \left[1 - \cos\left(2\pi \frac{(t + ((L-1)/2)T)}{LT}\right) \right] & -\frac{(L-1)T}{2} \le t \le \frac{(L+1)T}{2} \\ 0 & \text{otherwise,} \end{cases}$$
(4.25)

ſ	$B_g T$	θ_{-3}	θ_{-2}	θ_{-1}	θ_0	θ_1	θ_2	θ_3
ſ	0.2	0	1.6687	20.3893	45.8415	20.3893	1.6687	0
	0.25	0	0.5443	17.9411	53.0273	17.9411	0.5443	0
	0.5	0	0	9.5153	70.9684	9.5153	0	0
	∞	0	0	0	90	0	0	0

Table 4.1: θ_i in degrees at different $B_g T$ for GFSK with h = 0.5.

Table 4.2: θ_i in degrees at different L for RC pulse shape with h = 0.5.

L	θ_{-3}	θ_{-2}	θ_{-1}	$ heta_0$	θ_1	$ heta_2$	θ_3
5	0	4.3771	23.2035	58.6027	23.2035	4.3771	0
3	0	0	17.5951	54.8098	17.5951	0	0
1	0	0	0	90	0	0	0

are used, then the ISI will be centered such that $Z_f = Z_p = Z$. Table 4.1 and Table 4.2 list the values of $\theta_i = \pi h p_i$ for GFSK and RC pulse shapes respectively with h = 0.5. As an example, for GFSK with $B_g T = 0.5$, $Z = Z_f = Z_p = 1$.

Soft output demodulation is performed by running the BCJR algorithm on a M^{2Z} state trellis. The state at time instant t = kT is given by $S_k = (a_{k-2Z}, a_{k-2Z+1}, ..., a_{k-1})$, with Mbranches emerging out of each state. Once the trellis is so defined, the LLR given by (4.20) can be decomposed using the BCJR algorithm as

$$\mathcal{Z}_{i,k} = \log \frac{\sum_{\mathcal{S}^{(1)}} \alpha_k(s') \gamma_{k+1}(s', s) \beta_{k+1}(s)}{\sum_{\mathcal{S}^{(0)}} \alpha_k(s') \gamma_{k+1}(s', s) \beta_{k+1}(s)},$$
(4.26)

where $\mathcal{S}^{(1)}$ is the set of state transitions $\{S_k = s'\} \rightarrow \{S_{k+1} = s\}$ corresponding to $B_{i,k} = +1$, $\mathcal{S}^{(0)}$ is defined similarly for $B_{i,k} = 0$, and α , β and γ are the metrics in the BCJR algorithm (Chapter 2).

As in [10], the metric $\gamma_{k+1}(s', s)$ is defined as

$$\gamma_{k+1}(s',s) = P[S_{k+1} = s, \Delta \phi_k | S_k = s']$$

= $P[S_{k+1} = s | S_k = s'] P[\Delta \phi_k | (S_k \to S_{k+1}) = (s' \to s)],$ (4.27)

where $\Delta \phi_k$ is the received phase difference given in (4.19). Because the interleaved code bits are equally likely, so are the steady-state branch transition probabilities, and thus $P[S_{k+1} = s|S_k = s']$ is a constant that can be dropped from (4.27) without affecting the overall LLR (4.26). Note that if *a priori* information were delivered from the decoder back to the demodulator, then the *a priori* information could be used to update the value of $P[S_{k+1} = s|S_k = s']$. Such an operation is contemplated in [84].

Let $\Delta \varphi_k(s' \to s)$ be the transmitted phase difference associated with a transition from state s' to s, calculated by substituting the symbols associated with the two states into (4.23). Because of this one-to-one correspondence between state transitions and transmitted phase differences, (4.27) may be rewritten as

$$\gamma_{k+1}(s',s) = P\left[\Delta\phi_k | \Delta\varphi_k(s' \to s)\right], \qquad (4.28)$$

which is the conditional probability that phase difference $\Delta \phi_k$ was received given that phase difference $\Delta \varphi_k(s' \to s)$ was transmitted. Because the received phase difference $\Delta \phi_k$ is a continuous random variable, the probability given in (4.28) is actually zero. It is more appropriate to instead evaluate the probability that $\Delta \phi_k$ is within a small region (ϱ^-, ϱ^+)

$$\gamma_{k+1}(s',s) = P\left[\varrho^{-} \leq \Delta \phi_k < \varrho^{+} | \Delta \varphi_k(s' \to s) \right].$$
(4.29)

and then let $\epsilon = \varrho^+ - \varrho^- \to 0$. As ϵ gets small, the above probability becomes

$$\gamma_{k+1}(s',s) = \epsilon p_{\Delta\phi} \left(\Delta\phi_k | \Delta\varphi_k(s' \to s) \right), \qquad (4.30)$$

where $p_{\Delta\phi}(\cdot|\Delta\varphi_k(s'\to s))$ is the conditional pdf of the received phase difference given a transmitted phase difference of $\Delta\varphi_k(s'\to s)$. Because ϵ cancels in the overall LLR (4.26), it can be dropped and the branch metric $\gamma_{k+1}(s',s)$ can be determined by substituting the received phase difference $\Delta\phi_k$ into the conditional pdf.

In a practical receiver, one would not want to directly evaluate the true conditional pdf because it is not easily expressed in closed form. Alternatively, one could precalculate and store the pdf in a lookup table with a finite number of entries and read out the entry closest to the received phase difference. This would be accomplished by partitioning the phase region between 0 and 2π into R phase subregions $D_i = (\varrho_i^-, \varrho_i^+), 0 \le i \le R - 1$, where adjacent regions share a common boundary, $\varrho_i^+ = \varrho_{i+1}^-$. The table is read by first determining in which phase region the received phase difference lies, and then outputting the value stored in the table.

While the above described technique accurately approximates the true conditional pdf as the size of the phase regions gets small, it is suboptimal when a coarser table is used with fewer entries. This is because for wide regions, the conditional pdf within the region is no longer a constant and therefore the probability that the received phase difference lies in this region can no longer be approximated by the area of a rectangle, as given by (4.30). Instead, the probability given by (4.28) should be calculated by integrating the pdf over the region,

$$P\left[\varrho_i^- \le \Delta\phi < \varrho_i^+ | \Delta\varphi_k(s' \to s) \right] = \int_{\varrho_i^-}^{\varrho_i^+} p_{\Delta\phi}\left(\lambda | \Delta\varphi_k(s' \to s)\right) d\lambda.$$
(4.31)

The solution to the above integral is given in [73] to be

$$P\left[\varrho_i^- \le \Delta\phi < \varrho_i^+ | \Delta\varphi_k(s' \to s) \right] = 1 + F(\varrho_i^- | \Delta\varphi_k(s' \to s)) - F(\varrho_i^- | \Delta\varphi_k(s' \to s))$$

$$(4.32)$$

when $\varrho_i^- \leq \Delta \varphi_k(s' \to s)) < \varrho_i^+$ and

$$P\left[\varrho_i^- \le \Delta\phi < \varrho_i^+ | \Delta\varphi_k(s' \to s) \right] = F(\varrho_i^- | \Delta\varphi_k(s' \to s)) - F(\varrho_i^- | \Delta\varphi_k(s' \to s)) (4.33)$$

otherwise. The function F is described by the following set of equations [73]

$$F(\psi|\nu) = (2\pi)^{-1} \int_0^{\pi/2} \left[\exp(-E(\nu-\psi,\delta)) I'(\nu-\psi,\delta) \right]^{\log_2 M} \frac{\sin(\nu-\psi)}{1-\cos(\nu-\psi)\cos(\delta)} d\delta, (4.34)$$

$$E(\alpha, \delta) = \rho E_1(\alpha, \delta) / E_2(\alpha, \delta), \qquad (4.35)$$

$$E_1(\alpha, \delta) = 1 - \cos(\alpha)\cos(\delta), \qquad (4.36)$$

$$E_2(\alpha, \delta) = 1 - |\gamma| \cos(\alpha) \cos(\delta), \qquad (4.37)$$

$$I'(\alpha, \delta) = \frac{1 - |\gamma|}{1 - |\gamma| \cos(\alpha) \cos(\delta)},\tag{4.38}$$

$$|\gamma| = \frac{\Gamma}{1+K+\Gamma} R_{\xi}(T), \qquad (4.39)$$

$$\Gamma = \frac{P_x}{P_n} = \frac{\mathcal{E}_s}{N_0 B_n T} = \frac{\mathcal{E}_b r \log_2 M}{N_0 B_n T},\tag{4.40}$$

where $\mathcal{E}_s = P_x T$ is the energy per modulated symbol, \mathcal{E}_b is the energy per data bit, and the local variables α and γ used above are not to be confused with the BCJR branch metrics. Because the function F depends on the Rician K factor and the ratio \mathcal{E}_s/N_0 , it is assumed that these quantities are known to the receiver or that that receiver can estimate them. The SDDPD does not, however, require estimates of the fading amplitudes.

The SDDPD works by first pre-calculating $P\left[\varrho_i^- \leq \Delta \phi < \varrho_i^+ | \Delta \varphi_k(s' \to s)\right]$ for each state transition $(s' \to s)$ and each phase region D_i using (4.32) and (4.33). The results are then stored in a table with R columns (one for each phase region) and M^{2Z+1} rows, one for each state transition. The table only needs to be generated once for a particular SNR and Rician K-factor, but needs to be recreated as these quantities change. For each received phase difference $\Delta \phi_k$, the receiver determines the index of the phase region D_i in which it lies, i.e. the value of i for which $\varrho_i^- \leq \Delta \phi_k < \varrho_i^+$. This index is then used to read out the i^{th} column from the table, whose elements are used as the branch metrics $\gamma_{k+1}(s', s)$ for the corresponding state transitions $(s' \to s)$. Once the metrics are determined for every branch in the trellis, the demodulator executes the BCJR algorithm using (4.26), producing the LLR $\mathcal{Z}_{i,k}$ for each bit i of each symbol k. The LLRs are then placed into a row vector \mathbf{z} such that $z_{k\log_2 M+i} = \mathcal{Z}_{i,k}$. The vector is then deinterleaved and the resulting sequence \mathbf{z}' fed to the channel decoder for soft decision decoding.

4.3.4 Bit Error Rates for Uncoded Systems with Differential Detection

The expression for the error probability of an uncoded CPM system using DPD was derived in [73] and is briefly described here. The differential phase angles (*transmitted* phase differences) $\Delta\varphi$ and the corresponding permutation of input symbols for MSK ($B_gT = \infty$, h = 0.5) and binary GFSK with $B_gT = 0.5$ and h = 0.5 are listed in Table 4.3. Observe that when the current transmitted symbol is +1, $\Delta\varphi_k$ lies between 0 and π and between π and 2π the transmitted symbol is $a_k = -1$. On computing $\Delta\phi_k$ the detector makes a symbol

Table 4.3: Different	ntial phase ang	gles in radia	ns for binary	CPFSK at	different h
----------------------	-----------------	---------------	---------------	----------	---------------

h	a_k	$\Delta \varphi_k$
0.1	-1	5.9690
	1	0.3142
0.5	-1	4.7124
	1	1.5708
0.75	-1	3.9270
	1	2.3561

Table 4.4: Differential phase angles in radians for binary GFSK with $B_gT = 0.5$, h = 0.5.

a_{k-1}	a_k	a_{k+1}	$\Delta \varphi_k$
-1	-1	-1	4.712
-1	-1	1	5.044
-1	1	-1	0.9065
-1	1	1	1.2386
1	-1	-1	5.044
1	-1	1	5.376
1	1	-1	1.2386
1	1	1	1.5708

decision

$$\hat{a}_{k} = \begin{cases} +1, & \text{for } \Delta \phi_{k} \ge 0\\ -1, & \text{otherwise.} \end{cases}$$

$$(4.41)$$

The error probability is hence [73]

$$P[e] = \left[P[e|a=1] + P[e|a=-1] \right] / 2. \tag{4.42}$$

Due to the pulse shape, there are $2^{Z_f+Z_p} = 2^{2Z}$ symbol permutations with $a = \pm 1$ (the time indices have been dropped without loss in generality). The conditional error probabilities are hence

$$P[e|a = \pm 1] = \frac{1}{2^{2Z}} \sum_{i=1}^{2^{2Z}} P[e|a = \pm 1, i], \qquad (4.43)$$

where $(a = \pm 1, i)$ is the i^{th} combination of input symbols with $a = \pm 1$. From (4.41)

$$P[e|a = 1, i] = 1 - P[0 \le \Delta \phi \le \pi | \Delta \varphi^{i}(1)], \qquad (4.44)$$



Figure 4.8: Minimum separation between differential phase angles for symbols ± 1 (D_{min}) in degrees for binary CPFSK as a function of h.

for $i = 1, 2, ..., 2^{2Z}$ where $\Delta \varphi^i(1)$ is the differential phase angle corresponding to (a = 1, i). Similarly,

$$P[e|a = -1, i] = 1 - P[\pi \le \Delta \phi \le 2\pi | \Delta \varphi^{i}(-1)].$$
(4.45)

which can be evaluated using (4.33). By evaluating (4.43) using (4.44) and (4.45), one can find the probability of error for a symbol-by-symbol, hard decision DPD from (4.42).

Fig. 4.9 shows the BER in AWGN of binary CPFSK with DPD at different values of h found using (4.42). Observe that increasing h beyond 0.5 degrades the BER performance. This is because the performance of the detector is influenced by the (minimum) separation between the differential phase angles D_{min} for symbols ± 1 . Fig. 4.8 shows D_{min} as a function of h for binary CPFSK. This reveals D_{mins} is maximized for MSK with $D_{min} = \Delta\varphi(1) - \Delta\varphi(-1) = \pi$. Decreasing or increasing h beyond 0.5 reduces D_{min} , which results in a higher BER. For a particular D_{min} (except at $D_{min} = \pi$) one has two choices for the value of h, i.e. a value which is less than 0.5 or a value which is greater than 0.5. Our results indicate that it is better to pick h which is less than 0.5. We conjecture that this is because a larger value of h requires a front-end filter with a larger noise bandwidth B_n . Fig. 4.10



Figure 4.9: BER curves for uncoded binary CPFSK with hard decision, symbol-by-symbol DPD in AWGN at select values of h.

shows the BER for binary GFSK with $B_gT = 0.5$ at different values of h. For this scenario increasing h beyond 0.6 degrades the performance.

Increasing the pulse width introduces ISI which goes unmitigated when using the DPD. The performance can of partial response CPM can be improved using a sequence based differential detector such as the SO-SDDPD. Fig. 4.11 shows the BER for 2-GFSK at different values of h and $B_gT = 0.5$ using DPD and SO-SDDPD. The SO-SDDPD uses R = 40 uniformly spaced phase regions. The channel is AWGN. Sequence detection using the SO-SDDPD results in significant performance improvements which become more pronounced as the amount of modulation induced ISI is increased. This is illustrated in Fig. 4.12. Curves for the SO-SDDPD were generated using computer simulations with 25 frame errors recorded per \mathcal{E}_b/N_0 .

4.3.5 Applying SO-SDDPD to Bluetooth

Gaussian frequency shift keying (GFSK) is used in the Bluetooth [16] physical layer. The optimal ML coherent receiver [15], as well as low complexity coherent receivers ([77], [85],



Figure 4.10: BER curves for uncoded binary GFSK ($B_gT = 0.5$) with hard decision, symbolby-symbol DPD in AWGN at select values of h.



Figure 4.11: BER comparison between the DPD and SO-SDDPD for uncoded 2-GFSK with $B_gT = 0.5$ at different h. The SDDPD uses R = 40 uniform phase regions. The channel is AWGN.



Figure 4.12: BER comparison between the DPD and SO-SDDPD for uncoded 2-GFSK with h = 0.5 and $B_gT = 0.25$. The SO-SDDPD uses R = 26 uniform phase regions. Three different K-factors are considered $K = -\infty$ dB (Rayleigh), K = 6 dB and $K = \infty$ dB (AWGN). The maximum Doppler frequency is assumed to be $f_d = 0$.

[86]) are susceptible to phase estimation errors. Noncoherent or differential receivers are hence preferred for Bluetooth systems. The noncoherent detector often used for Bluetooth is the limiter discriminator integrator (LDI) detector [87], [88]. Since the data medium (DM)-rate packet type in Bluetooth is protected by a (15, 10) shortened Hamming channel (SHC) code [16], typically, LDI detection is followed by hard decision decoding (HDD) of the code bits (LDI-HDD). While low in complexity, these receivers have poor energy efficiency, especially in harsh mobile environments. It is hence desirable to investigate power efficient receiver designs while maintaining feasible complexity.

In this section we demonstrate that significant gains in the error rate and throughput are possible by using our proposed SO-SDDPD for Bluetooth systems. For comparison purposes, six different systems are considered: 1) LDI detection followed by hard decision decoding (HDD) of the code bits (LDI-HDD) 2) LDI-HDD with bit-interleaving 3) SDDPD followed by HDD of the SHC (SDDPD-HDD) 4)SDDPD-HDD with bit-interleaving 5) SO-SDDPD with soft-decision decoding of the SHC (because it is cyclic, the SHC code is treated as a recursive systematic convolutional (RSC) code). (SO-SDDPD-SDD) 6)SO-SDDPD-SDD



Figure 4.13: Dotted curve is the BICM capacity in Rician channel with K = 2 dB, using SISO-SDDPD. Six simulated points are shown for DM1 packets, representing minimum \mathcal{E}_b/N_0 (dB) to achieve BER = 10^{-4} , from top to bottom: (1) LDI-HDD (2) LDI-HDD with bit-interleaving (3) SDDPD-HDD (4) SDDPD-HDD with bit-interleaving (5) SO-SDDPD-SDD (6) BICM receiver. All SDDPD systems use R = 24 uniform phase regions. Modulation index h = 0.315 is assumed.

with BICM.

Bit Error Rate Comparisons

Fig. 4.13 shows the simulated minimum \mathcal{E}_b/N_o for DM1 packet types ($N_a = 240$) at BER = 10^{-4} and select receivers (BER is measured at the channel decoder's output). The bottom most curve in Fig. 4.13 is the information theoretical minimum \mathcal{E}_b/N_0 (Chapter ??) required to achieve arbitrarily low bit error rate (BER) for a BICM receiver using SO-SDDPD with R = 24 uniform phase sub-regions (i.e. width of each sub-region is $2\pi/R$). The channel is Rician with K = 2 dB and a modulation index of h = 0.315 is assumed.

A The BICM receiver performs closest to capacity. A BER gain of 8 dB and 4.35 dB over LDI-HDD, and SDDPD-HDD respectively and 0.8 dB over the SO-SDDPD-SDD is observed. BICM with iterative decoding (BICM-ID) [89] was also investigated. However,

iterating between the demodulator and decoder gave no significant gain over BICM (hence not shown in Fig. 4.13). A possible reason could be that $B_gT = 0.5$ causes only a little adjacent symbol interference. Hence during BICM-ID, extrinsic information for each modulated bit is provided by only two other bits, which appears to be insufficient to give noticeable improvement over BICM. Simulations (not shown here) reveal increasing gain using BICM-ID with decreasing values of B_gT . There is however a 9 dB gap between BICM capacity and the proposed receiver. This is primarily due to the short packet sizes and weak channel code used in the Bluetooth standard. This gap could be reduced by using capacity approaching channel codes, such as turbo or LDPC codes instead. It is observed from the capacity curve that the minimum \mathcal{E}_b/N_0 does not necessarily improve with decreasing code rate. This is partly due to the noncoherent combining penalty [90].

Packet Error Rate Comparisons

Fig. 4.14 shows the packet error rate (PER) of the DM1 packet type using different receivers. The SO-SDDPD-SDD receiver gives an \mathcal{E}_b/N_0 gain of about 8 dB over the LDI-HDD (the SHC code is treated as a single error correcting code). Bit-interleaving is seen to offer no improvement in the PER for LDI-HDD. A 3.1 dB reduction in \mathcal{E}_b/N_o is observed compared to SDDPD-HDD. Note that for the SDDPD-HDD, bit-interleaving improves the PER, especially at higher \mathcal{E}_b/N_0 . BICM gives a 0.82 dB gain over SO-SDDPD-SDD, this gain was close to 1 dB for the DM3 packet type and 1.5 dB for the DM5 packet type. BICM-ID was performed with 4 iterations carried out between the SO-SDDPD and the decoder. No significant improvement was observed over a BICM receiver. Similar gains between receivers were observed for the DM3 and DM5 packet types.

Throughput Comparison

The throughput (maximum achievable one-way data rate) for the six ACL packets using ARQ (DM1, DM3, DM5, DH1, DH3, DH5) was calculated in [91]. However, [91] assumes nonorthogonal, full response FSK which does not account for GFSK induced ISI. Here, we extend analysis in [91] to find throughput as a function of \mathcal{E}_b/N_0 for GFSK with Bluetooth specifications, taking into account both ISI and receiver implementation. We consider those



Figure 4.14: PER for DM1 packet types in Rician channel with K = 2 dB. All SDDPD systems use R = 24 uniform phase regions. Dotted curves indicate systems with bit-interleaving. Modulation index h = 0.315 is assumed.

ACL packets that use the SHC code (DM1, DM3, DM5). Let N_t be the (average) total number of times a given packet must be transmitted until it is successfully decoded. The data rate (throughput) is a function of N_t given by [91]

$$D_r = \frac{K_p}{(N_s N_t)(625 \times 10^{-6})} \tag{4.46}$$

where N_s is the number of slots occupied per round trip including one return slot, K_p is the number of data bits in the packet type. Assuming no upper limit of retransmissions,

$$N_t = \frac{1}{1 - \bar{Pe}}$$

where \bar{Pe} is the average PER.

Fig. 4.15 shows throughput performance for the different receivers for the DM1 packet type. Since relative performance between receivers for the DM3 and DM5 packet type follows a similar trend, only the best (BICM and SO-SDDPD-SDD) and worst case (LDI-HDD) throughput is shown. At high SNR, the throughput converges to the maximum possible



Figure 4.15: Throughput for DM1, DM3 and DM5 packet types in Rician channel with K = 2 dB. SDDPD systems use R = 24 uniform phase regions. Dotted curves indicate systems with bit-interleaving. Modulation index h = 0.315 is assumed.

value for each packet type i.e. 108.8 kbps for DM1, 387.2 kbps for DM3 and 477.9 kbps for DM5. The increase in data rate and energy efficiency due to SO-SDDPD-SDD is evident from Fig. 4.15. As an example, at $\mathcal{E}_b/N_0 = 20$ dB, our proposed receiver gives a 30 kbps improvement in throughput over LDI-HDD for DM1 packet types. The gain in throughput is even more significant (450 kbps) at $\mathcal{E}_b/N_0 = 20$ dB, if DM5 packet types (with SO-SDDPD-SDD/SDD/BICM) were used instead of DM1. Hence, it could be inferred that to achieve maximal throughput, the packet type should be adaptively selected to match the SNR as suggested in [92]. BICM is seen to offer a 5 kbps increase in throughput over SO-SDDPD-SDD at lower \mathcal{E}_b/N_0 for the DM1 packet type, and increasing gains are seen with DM3 and DM5 packet types.

At this point some important qualifications on our system's complexity must be made. Our noncoherent sequence detector performs MAP decoding on a M^2 -state trellis. Prior to MAP decoding, the branch metrics are calculated and stored in an $M^3 \times R$ look-up table. The metric calculations involve nonlinear functions and need to be updated once at each \mathcal{E}_b/N_0 making our system more complex than LDI and DPD. However, it has been pointed out in [77] that branch metrics calculated at BER = 10^{-4} seem optimum for all \mathcal{E}_b/N_0 . The size of the look-up table could be reduced further by careful selection of the phase subregions. For simplicity, R uniformly spaced regions are used, but as mentioned in [77], the same performance could be obtained using a smaller number of non-uniform phase regions. However, the non-uniform regions may have to be recalculated each time the modulation parameters change.

4.4 Coherent Detection versus Differential Detection

Since differential detectors base their decisions by comparing the received noisy signal with its delayed version, their performance is expected to be worse than coherent detectors. Fig. 4.16 shows the BER for uncoded binary CPFSK using the SO-Coherent detector and the DPD, in AWGN. For MSK, using the DPD results in a 2.5 dB degradation in \mathcal{E}_b/N_0 . However, the difference between the two detectors becomes less pronounced at smaller values of h, with the gap in \mathcal{E}_b/N_0 reducing to 0.7 dB at h = 1/4. Similar conclusions can also be drawn for partial response signaling (Fig. 4.17).

Regarding complexity, differential detectors benefit from a lower complexity front-end and back-end, relative to the optimum coherent detector. Since carrier phase recovery is not required, differential detectors could provide a more robust signal reception in harsh mobile environments.

The SO-Coherent detector employs a $p_h M^{L-1}$ state trellis, due to which its complexity increases exponentially with increasing M and L and linearly with increasing p_h . Additionally, with coherent detection h is constrained to be a rational number. In contrast, the complexity of the SO-SDDPD is entirely independent of the choice of h and we could set hto any value we like. While the SO-SDDPD operates on a M^{L-1} state trellis, its complexity could be reduced by constraining the detector to account only adjacent symbol interference (ISI) without incurring a significant loss in performance. In this case, the SO-SDDPD only requires a M^2 state trellis.



Figure 4.16: BER comparison between the DPD and the sequence based coherent detector for uncoded binary-CPFSK at different h in AWGN.



Figure 4.17: BER comparison between the SO-SDDPD and the sequence based coherent detector for uncoded binary 3-RC CPM in AWGN. The SDDPD used R = 40 uniform phase regions.

4.5 Chapter Summary

Differential detection is a pragmatic alternative to coherent detection when low-complexity receivers are desired. Additionally, since coherent detectors are susceptible to phase estimation errors, differential detectors are a more robust option as well. Since most modern communication systems use receivers with soft-decision decoding and because our capacity calculations require bit-wise LLRs from the detector, we have developed a novel soft-output, soft-decision differential phase detector (SO-SDDPD). The SO-SDDPD is shown to provide significant gains over DPD for partial response CPM.

The main complaint about differential detectors is their poor energy efficiency [78]. This however is true only at $h \ge 0.5$. We have demonstrated (for an uncoded system in this chapter and for coded systems in Chapter 6) that the differential detector can approach the performance of the optimum coherent detector. This is especially true for the low values hthat are required under tight bandwidth constraints. The SO-SDDPD also requires fewer states per trellis section than the coherent detector. The number of trellis states in the coherent detector increases as h is lowered, with h constrained to be a rational number. This could result in a receiver with unreasonable complexity at very high spectral efficiencies. We conjecture that the best differential design may in fact outperform the best coherent design on reasonable complexity simply because the differential detector does not need to round hto a rational number.

Another alternative to coherent detection is to use noncoherent detectors in which symbol decisions are made in the presence of the channel induced random phase rotation. A detailed treatment of noncoherent detectors for CPM is outside the scope of this dissertation, instead we refer interested readers to [21, 27, 78, 30, 79, 93] for details.

Chapter 5

Mutual Information as a Performance Measure

As discussed in Chapter 1, the main drawback in using the bit error rate (BER) as a cost function in designing sophisticated communication systems is that BER simulations are extremely time consuming for systems that use capacity-approaching coding. A mutual information based cost function is a more pragmatic and less time consuming alternative. In this chapter we delve into the details of the applications of mutual information most relevant to our work. These include 1) the Shannon capacity, 2) the modulation constrained capacity and 3) extrinsic information transfer charts. A computationally feasible method for computing the above metrics is also presented.

As a starting point, we begin with the definition of mutual information. Let X be a 1-dimensional input signal with average power P. The signal at the output of an additive Gaussian channel is hence

$$Y = X + Z,\tag{5.1}$$

where z is Gaussian noise with zero mean and variance N. The *mutual information* between X and Y is defined as [94]

$$I(X;Y) = \int \int p(x,y) \log \frac{p(x,y)}{p(x)p(y)} dxdy.$$
(5.2)

It is noted that (5.2) can be expressed in terms of the expectation

$$I(X;Y) = E[i(X;Y)]$$

= $\int \int p(x,y)i(x;y)dxdy,$ (5.3)

where the *mutual information random variable* is defined as

$$i(x;y) = \log \frac{p(x,y)}{p(x)p(y)}.$$
 (5.4)

The remainder of this chapter is organized as follows. We begin with a detailed exposition of the unconstrained Shannon capacity. Next, we derive the capacity under modulation constraints for two widely used paradigms in coded system design, namely *coded modulation* (CM) and *bit-interleaved coded modulation* (BICM) [37]. A computationally feasible method for computing the constrained capacities is described, following which we present select CM and BICM capacity results. Finally, extrinsic information transfer charts [95] (EXIT) and their application to iteratively decoded systems are discussed.

5.1 The Unconstrained Shannon Capacity

The channel capacity as defined by Shannon [1] is simply the mutual information between the channel input and output, maximized over all input distributions p(x) i.e.

$$C = \max_{p(x)} I(X;Y). \tag{5.5}$$

From [94]

$$I(X;Y) = H(Y) - H(X + Z|X) = H(Y) - H(Z).$$
(5.6)

Since Z is Gaussian, H(Z) can be written as

$$H(Z) = \frac{1}{2}\log_2 2\pi eN.$$
 (5.7)

Since the entropy of any random variable is upper bounded by the entropy of a Gaussian random variable with the same variance [94]

$$H(Y) \le \frac{1}{2}\log_2 2\pi e(P+N).$$
 (5.8)

R. Iyer Seshadri

Hence (5.6) can be written as

$$I(X;Y) \le \frac{1}{2}\log_2 2\pi e(P+N) - \frac{1}{2}\log_2 2\pi eN,$$
(5.9)

where the equality is achieved when x is a zero mean Gaussian random variable. The channel capacity in units of bits per transmission is hence

$$C = \frac{1}{2}\log_2\left(1 + \frac{P}{N}\right),\tag{5.10}$$

From Nyquist's sampling theorem, a bandlimited, continuous time signal can be accurately represented by discrete samples spaced 1/2W seconds apart. Hence the interval [0, T]consists of 2WT samples. A continuous time signal during [0, T] can hence be represented using a n = 2WT dimensional vector. The signal power per sample is hence P/2W and the noise power per sample is $N_0/2$. The channel capacity in bits per sample (dimension) is hence,

$$C = \frac{1}{2} \log_2 \left(1 + \frac{P/2W}{N_0/2} \right) = \frac{1}{2} \log_2 \left(1 + \frac{P}{N_0 W} \right).$$
(5.11)

In terms of bits per transmission (channel use), this becomes

$$C = WT \log_2 \left(1 + \frac{P}{N_0 W} \right). \tag{5.12}$$

The capacity in bits per second is hence

$$C = W \log_2 \left(1 + \frac{P}{N_0 W} \right). \tag{5.13}$$

It is often more informative to express the capacity in terms of \mathcal{E}_s/N_0 or \mathcal{E}_b/N_0 . Now,

$$P = \mathcal{E}_{s}r_{s}$$

$$= \mathcal{E}_{b}r\log_{2}Mr_{s}$$

$$= \mathcal{E}_{b}r\log_{2}M\frac{r_{b}}{r\log_{2}M}$$

$$= \mathcal{E}_{b}r_{b}.$$
(5.14)

(5.15)

where r_s is the symbol rate, r_b is the data rate and r is the code rate. The capacity in bits per second is hence

$$C = W \log_2 \left(1 + \frac{\mathcal{E}_b r_b}{N_0 W} \right).$$
(5.16)

The maximum data rate is equal to the channel capacity $r_b = C$, hence

$$C = W \log_2 \left(1 + \frac{\mathcal{E}_b C}{N_0 W} \right). \tag{5.17}$$

The minimum \mathcal{E}_b/N_0 required to achieve an arbitrarily low BER can be written as

$$\frac{\mathcal{E}_b}{N_0} = \frac{2^{C/W} - 1}{C/W}.$$
(5.18)

For a one dimensional, binary signal set, the \mathcal{E}_b/N_0 as a function of r is hence (assuming $W = r_s/2$)

$$\frac{\mathcal{E}_b}{N_0} = \frac{2^{2r} - 1}{2r}.$$
(5.19)

The above equation can be generalized for an N dimensional, M-ary signal set as

$$\frac{\mathcal{E}_b}{N_0} = \frac{2^{\frac{2r\log_2 M}{N}} - 1}{\frac{2r\log_2 M}{N}}.$$
(5.20)

5.2 Modulation Constrained Capacity

The mutual information in (5.5) is maximized when the input distribution is Gaussian. Practical communication systems however use modulation schemes (eg. BPSK, QAM, CPM, etc..) in which the transmitted input symbols are drawn from a finite set. In such scenarios, the unconstrained Shannon capacity is hence an overly optimistic indicator of the system performance. The capacity under constraints of a practical modulation scheme must be computed using the pdf of the underlying modulation.

5.2.1 Capacity of Coded Modulation

Following the introduction of *trellis-coded modulation* (TCM) by Ungerboeck [25], a popular approach to designing coded systems is to treat coding and modulation as a single entity.



Figure 5.1: Block diagram of system with coded modulation.

This process is known as *coded modulation* (CM) and is shown in Fig. 5.1. A CM scheme [37] consists of an encoder for a code defined over alphabet \mathcal{A} and a N-dimensional modulator over a signal set $\chi \subseteq \mathcal{C}^N$ separated by a *symbol interleaver*. Also, $|\chi| = |\mathcal{A}| = M$. There is a one-to-one mapping between the generated code symbol and the modulated symbol.

To determine the CM capacity, we first evaluate the symbol log-likelihood ratio (LLR) $\log p(x|y)$ by noting that

$$\log p(x|y) = \log \frac{p(x|y)}{\sum_{x_k \in \chi} p(x_k|y)}.$$
(5.21)

Applying Bayes rule, we have

$$\log p(x|y) = \log \frac{p(y|x)}{\sum_{x_k \in \chi} p(y|x_k)},$$
(5.22)

where,

$$p(y|x) = \frac{1}{(\pi N_0)^{N/2}} \exp(-||y - x||^2 / N_0)$$

= $\frac{1}{(\pi N_0)^{N/2}} \exp(f(y|x)),$ (5.23)

where

$$f(y|x) = -\|y - x\|^2 / N_0.$$
(5.24)

Equation (5.21) can now be written as

$$\log p(x|y) = f(y|x) - \max_{x_k \in \chi} * f(y|x_k),$$
 (5.25)

where the max * operator is defined as [64]

$$\max_{i} *\{x_i\} = \log \sum_{i} \exp(x_i).$$
 (5.26)

From (5.3), (5.25) and assuming that the modulated symbols are equally likely ¹ the CM capacity in nats per channel use is

$$C = \log M + E[f(y|x) - \max_{x_k \in Y} * f(y|x_k)].$$
(5.27)

To convert to bits per channel use, the above equation is divided by $\log 2$.

5.2.2 Capacity of Bit-Interleaved Coded Modulation

An alternative system design can be obtained by concatenating a binary encoder with a N-dimensional modulator defined over a signal set $\chi \subseteq \mathcal{C}^N$ such that $|\chi| = M$, through a bit-interleaver. Such a scheme is called *bit-interleaved coded modulation* (BICM) [37] and is shown in Fig. 5.2). The codeword \mathbf{b}' is bit-interleaved to produce the sequence \mathbf{b} . $\mu = \log_2 M$ bits of **b** are mapped to produce the N-dimensional modulated symbol x_k . BICM simplifies system design, since, the code alphabet and the modulation alphabet need not match. Moreover, most off-the-shelf capacity approaching codes tend to be binary and can hence be easily incorporated in a BICM framework. Assuming ideal interleaving, $\mu = \log_2 M$ bits from the channel encoder's output are transmitted using independent symbols. This is not possible in CM due to symbol-level interleaving. This implies BICM can offer better diversity than CM in fading channels [37], [39]. In essence, BICM maximizes the Hamming distance whereas CM maximizes the Euclidian distance. While maximizing the Euclidian distance is good for AWGN, it is better to maximize the Hamming distance in fading. This is because in a fully interleaved fading channel, the diversity order *is* the Hamming distance. Similar conclusions are drawn for CPM systems in [30]. The disadvantage of BICM is that its capacity is smaller than the CM capacity due to the *data processing inequality*.

¹Although assuming equally likely symbols maximizes the mutual information for modulations such as M-PSK, it is not necessarily the best distribution for modulation schemes such a M-QAM. In such scenarios, the mutual information computed under the assumption of equally likely symbols is more appropriately termed as the symmetric information rate.



Figure 5.2: Block diagram of system with bit-interleaved coded modulation.

As noted in [37], BICM transforms the channel into $\log_2 M$ parallel channels such that the capacity of the i^{th} channel in nats is

$$C_i = I(b_i; z_i)$$

= $E[i(b_i; z_i)].$ (5.28)

 z_i is now a bit-wise log-likelihood ratio

$$z_{i} = \log \frac{P[b_{i} = 1 | \mathbf{y}]}{P[b_{i} = 0 | \mathbf{y}]}$$
$$= \log \frac{\sum_{x_{k} \in \mathcal{S}^{(1)}} p(y | x_{k}) p(x_{k})}{\sum_{x_{k} \in \mathcal{S}^{(0)}} p(y | x_{k}) p(x_{k})},$$
(5.29)

where, $S^{(1)}$ represents the set of symbols whose i^{th} bit is 1 and $S^{(0)}$ is similarly defined for bit 0. In BICM, x_k are assumed equally-likely. The two events $\{z\}$ and $\{y\}$ are said to be *equivalent* in the sense that conditioning on detector output z is equivalent to conditioning on detector input y. Using the fact that b is discrete, we get

$$i(b;z) = \log \frac{1}{P[b]} + \log P[b|z],$$
 (5.30)

assuming that b is equally likely to be a one or zero

$$i(b;z) = \log 2 + \log P[b|z].$$
 (5.31)
R. Iyer Seshadri Chapter 5. Mutual Information as a Performance Measure

We now focus on the term $\log P[b|z]$. Since P[b=0|z] + P[b=1|z] = 1

$$\log P[b|z] = \log \frac{P[b|z]}{P[b=0|z] + P[b=1|z]}$$

= $-\log \left(\frac{P[b=0|z]}{P[b|z]} + \frac{P[b=1|z]}{P[b|z]} \right).$ (5.32)

The evaluation of the above expression depends on whether the transmitted bit was a zero or a one. For the case that b = 0,

$$\log P[b=0|z] = -\log\left(1 + \frac{P[b=1|z]}{P[b=0|z]}\right).$$
(5.33)

Note that

$$\frac{P[b=1|z]}{P[b=0|z]} = e^z$$
(5.34)

and thus

$$\log P[b = 0|z] = -\log (e^{0} + e^{z})$$

= -max *(0, z). (5.35)

A similar derivation when b = 1 yields

$$\log P[b=1|z] = -\max *(0, -z).$$
(5.36)

Equations (5.35) and (5.36) can be compactly represented by the single expression

$$\log P[b|z] = -\max * (0, z(-1)^b).$$
(5.37)

Since the capacities of parallel channels add, the BICM capacity is simply

$$C = \sum_{i=1}^{\log_2 M} C_i$$

=
$$\sum_{i=1}^{\log_2 M} \log(2) - E\left[\max *\{0, z_i(-1)^{b_i}\}\right].$$
 (5.38)

The BICM capacity when converted to bits per channel use is

$$C = \log_2 M - \frac{1}{\log(2)} \sum_{i=1}^{\log_2 M} E\left[\max *\{0, z_i(-1)^{b_i}\}\right].$$
(5.39)

94

5.2.3 A Computationally Feasible Method for Evaluating Capacity

The capacity in (5.27) can be written as

$$C = \log M + \sum_{i=1}^{M} \int p(x_i, y) [f(y|x) - \max_{x_k \in \chi} * f(y|x_k)] dy$$

= $\log M + \frac{1}{M} \sum_{i=1}^{M} \int p(y|x_i) [f(y|x) - \max_{x_k \in \chi} * f(y|x_k)] dy,$ (5.40)

where the above integral is evaluated over N dimensions. For low-dimensionality linear modulation such as BPSK, PSK, or QAM, numerical integration could be used to evaluate the expectation. For more complicated systems, such as the ones that we study in this work, the expectations in (5.27), (6.1) are most readily found by using Monte Carlo integration. We note that we are not the first to propose using Monte Carlo integration to evaluate capacity. Indeed, it has become a common practice in the literature [37], [43], [44]. For CM, the basic idea is to randomly generate symbols x and transmit them through a simulated channel. The receiver can be viewed as a metric calculator that computes the symbol-wise LLRs. The CM capacity can be found using (5.27) by averaging the outcomes of a large number of trials. The BICM capacity is calculated by using a detector that generates bit-wise LLRs along with equation (6.1).

Just as with conventional BER simulations, the accuracy of the capacity when evaluated through Monte Carlo methods depends on the number of trials. This is illustrated in Fig. 5.3, which shows the effect of the number of trials (in terms of the number of symbols per SNR) on the constrained CM capacity. The modulation is BPSK and channel is AWGN. It is well known that the minimum \mathcal{E}_b/N_0 required by BPSK for C = 0.5 is 0.2 dB. As expected, when the number of trials is very small, the capacity values obtained by Mote Carlo integration deviate considerably from the actual value. However, as the number of trials increases, this deviation becomes smaller and as the number of trials tends to infinity, the capacity converges to the "true" value. It is obvious that there exists a tradeoff between the accuracy of the capacity calculations (i.e. the number of trials) and the simulation time. Fig. 5.4 shows the simulation time (in seconds) as a function of the number of trials. From



Figure 5.3: BPSK capacity as a function of the number of simulated symbols per SNR.

Figs. 5.3 and 5.4, we can infer that Monte Carlo integration using a million symbols per SNR provides a good tradeoff between accuracy and simulation time. The MATLAB modules for computing the CM capacity using Monte-Carlo integration are given in Appendix B.

5.2.4 Capacity Results

Fig. 5.5 shows the modulation unconstrained capacity in bits per channel use for a 2 dimensional signal set. Also shown are the CM capacities for PSK and QAM at different alphabet sizes M. Unlike the unconstrained Shannon capacity, the CM capacity allows for a more meaningful comparison between various modulation schemes. For instance, we can infer from Fig. 5.5 that for communication systems in which power is more of a premium than bandwidth, BPSK modulation used in conjunction with a low rate powerful channel code such as the one in [13] is a very attractive option. When higher spectral efficiency is required, one could select non-binary PSK or QAM. Our capacity results (Fig. 5.5) reveal that given a choice between M-PSK and M-QAM, M-QAM provides a better tradeoff between spectral efficiency and energy efficiency. However, PSK does have the advantage of being a constant



Figure 5.4: Simulation time as a function of the number of simulated symbols per SNR for computing the BPSK capacity.

envelope modulation.

Using (6.1), the capacity under BICM can be evaluated for different modulation schemes using Monte-Carlo integration. Unlike the CM capacity, the BICM capacity is influenced by the bit-to-symbol mapping. Fig. 5.6 shows the BICM capacity (as bits per channel use) versus \mathcal{E}_s/N_0 for 16-PSK with gray and natural bit-to-symbol labelling. Also shown is the corresponding CM capacity. BICM being sub-optimum, its capacity will be either less than or equal to the CM capacity. It is observed that BICM with gray labelling performs closest to the CM capacity. This is also evident in Fig. 5.7 for 16-QAM (SP denotes the set-partitioning labelling from [37]).

It is interesting to compare the simulation time taken to generate capacity curves by Monte-Carlo integration against the simulation time required to generate error rate curves by Monte-Carlo trials. Such a comparison is illustrated in Fig. 5.8. The modulation is BPSK, in an AWGN channel. The figure shows the average simulation time per SNR point in seconds required to generate capacity and bit error rate results by Monte-Carlo trials using 5 million modulated symbols generated per SNR point. To generate the BER curves,



Figure 5.5: The unconstrained Shannon capacity for 2 dimensional signal sets and and CM capacity for different linear modulations as a function of \mathcal{E}_b/N_0 . The CM capacities were calculated using Monte-Carlo integration with 2 million symbols generated per SNR. The modulation signal set is assumed to have equally likely symbols.



Figure 5.6: The BICM capacity of 16-PSK with natural and gray labelling in AWGN. Also shown is the CM capacity. Monte-Carlo integration with 2 million symbols generated per \mathcal{E}_s/N_0 was used to compute the capacity.



Figure 5.7: The BICM capacity of 16-QAM with set partitioning and gray labelling in AWGN. Also shown is the CM capacity. Monte-Carlo integration with 2 million symbols generated per \mathcal{E}_s/N_0 was used to compute the capacity.

two different channel codes were considered: a rate 1/2, constraint length 3 (generator polynomials [7, 5] octal) convolutional code and a rate 1/2 DVB-S2 LDPC code [58] (with 100 decoder iterations). The frame size is 64800 coded bits. The simulations were run on an Intel *pentium*[®] 4 machine with a 3.4 GHz processor. The capacity simulations were written in MATLAB. The channel encoding and decoding functions were written in C and called from MATLAB using C-MEX. Observe that the capacity simulations are significantly faster since the channel code need not be incorporated in the simulations. For instance, simulating the error rate performance of BPSK with the DVB-S2 LDPC code requires about 13 minutes (on an average) per SNR point. In contrast, calculating the capacity under BPSK requires only about 0.15 minutes per SNR point for the same number of simulated symbols.

5.3 Extrinsic Information Transfer Charts

While BICM simplifies design and provides higher diversity in fading channels, its capacity is lower than coded-modulation due to the data-processing inequality, as shown in Fig.



Figure 5.8: Average time required per SNR in seconds to compute the capacity and bit error rate by Monte Carlo trials. Simulations were performed using 5 million modulated symbols per SNR point.

5.6 and Fig. 5.7. One method to mitigate the performance loss due to BICM and approach the CM capacity is by iteratively exchanging bit-wise *extrinsic information* between the detector and the decoder. Such a process is often called *bit-interleaved coded modulation with iterative decoding* (BICM-ID), a term coined by Li and Ritcey in [89]. The block diagram of a system with BICM-ID is shown in Fig. 5.9. To begin the iterations, the detector upon receiving the noisy signal generates bit LLRs, which after interleaving are fed as *a priori* information to the decoder. The decoder uses this information to generate LLRs for the code bits. Extrinsic information for the detector is obtained by subtracting the *a priori* information to the decoder from the LLRs. The extrinsic information after deinterleaving becomes *a priori* information for the detector.

As has been reported widely in the literature, the bit-error rate curves for systems with BICM-ID can be divided into three regions: 1) the *bottleneck region*, in which the error rate is unacceptably high even after a large number of iterations, 2) the *waterfall region* which is characterized by a steep drop in the error rate with increasing iterations, 3) the *error floor*



Figure 5.9: Block diagram of system with bit-interleaved coded modulation with iterative decoding.

region in which a low error rate is achieved after relatively fewer number of iterations and in which increasing the SNR brings little or no further reduction in the error rate. These regions are illustrated in Fig. 5.10. The iterative decoding is said to *converge* at the onset of the waterfall region, which causes the error rate to drop to some arbitrarily low value. The minimum \mathcal{E}_b/N_0 required to trigger the transition from the bottleneck region to the waterfall region is known as the *convergence threshold*.

A most remarkable tool for visualizing the process of iterative decoding and predicting the convergence threshold was developed by ten Brink in [95] known as *Extrinsic Information Transfer* (EXIT) charts. Since their inception, EXIT charts have been widely used in diverse applications such as analyzing the performance of turbo codes [96], designing turbo coded noncoherent-orthogonal FSK [44], [67], reducing error floors in iteratively decoded QAM systems [97], designing capacity approaching for multiple antennae systems [66], turbo equalization [98], turbo multiuser detection [99], analyzing coded CPM with iterative decoding [100] and code design for CPM ([33], [35]).

An EXIT chart consists of two curves 1) detector mutual information transfer characteristic 2) decoder mutual information transfer characteristic. The *mutual information transfer characteristic* is a plot of the mutual information of the extrinsic information at the output



Figure 5.10: BER after 20 BICM-ID iteration for 4-CPFSK with h = 1/3 and rate 1/2 NRC code with octal generators [7, 5]. Two different bit-to-symbol mappings are considered. 4-CPFSK is coherently detected. The channel is AWGN and interleaver size is 100000 bits.

of the decoder/ detector as a function of the mutual information of the *a priori* input. A key assumption is generating the transfer characteristics is that the *a priori* information (as an LLR) is Gaussian distributed with a variance (σ^2) equal to twice the mean. There is a one-to-one correspondence between the variance σ^2 and the mutual information of the *a priori* information given by [96]

$$\mathcal{J}(\sigma) = 1 - \frac{1}{\sqrt{2\pi\sigma}} \int_{-\infty}^{\infty} \exp\left(-\frac{(x-\sigma^2/2)^2}{2\sigma^2}\right) \log_2\left(1+\exp(-x)\right) dx.$$
(5.41)

Fig. 5.11 shows $\mathcal{J}(\sigma)$ as a function of the variance σ^2 .

5.3.1 Detector Transfer Characteristics

To plot the detector transfer characteristic, a long sequence of bits (**b**) are generated and mapped to symbols. The symbol sequence is modulated and transmitted through the desired channel with a certain \mathcal{E}_s/N_0 . Next, Gaussian distributed *a priori* LLRs i.e $\bar{\mathbf{v}} \sim \mathcal{N}((2\mathbf{b}-1)\sigma^2/2,\sigma^2)$ with mutual information I_v are generated using (5.41). The extrinsic



Figure 5.11: Mutual information of Gaussian distributed $a \ priori$ information as a function of the variance.

information $\bar{\mathbf{z}}$ is obtained by subtracting $\bar{\mathbf{v}}$ from the LLRs \mathbf{z} (5.29)

$$\bar{\mathbf{z}} = \mathbf{z} - \bar{\mathbf{v}}.\tag{5.42}$$

From (6.1), the mutual information at the output of the detector is

$$I_z = 1 - \frac{1}{\log(2)\log_2 M} \sum_{i=1}^{\log_2 M} E\left[\max * (0, \bar{z}_i(-1)^{b_i})\right].$$
(5.43)

The mutual information at the output of the detector is a function of the I_v and \mathcal{E}_s/N_0 . As with the constrained capacity, I_z is found using Monte-Carlo integration. Fig. 5.12 shows detector transfer characteristics for 16-QAM with different bit-to-symbol mapping in AWGN at different \mathcal{E}_s/N_0 . The value of I_z when $I_v = 0$ is the BICM capacity.

5.3.2 Decoder Transfer Characteristics

The mutual information at the output of the channel decoder I_v is a function of the mutual information of the APP input $\bar{\mathbf{z}}$ to the decoder i.e I_z . To plot the decoder transfer characteristic, a long sequence of code bits (**b**) are generated. The *a priori* input to the

104



Figure 5.12: Detector mutual information transfer characteristics for 16-QAM modulation in an AWGN channel, at different \mathcal{E}_s/N_0 and bit-to-symbol mapping.

decoder is Gaussian distributed such that $\bar{\mathbf{z}} \sim \mathcal{N}((2\mathbf{b}-1)\sigma^2/2, \sigma^2)$. The decoder produces extrinsic information $\bar{\mathbf{v}}$ from which I_v can be calculated using (5.43). Fig. 5.13 shows the decoder transfer characteristics for rate-1/2 nonrecursive convolutional (NRC) codes with constraint lengths K = 2, 3 and 5. Fig. 5.14 shows the decoder transfer characteristics for NRC codes with generators [7, 5], [7, 4] and RSC code with feedback/ feedforward generators [7, 3]. As long as the *a prioris* generated by the detector are Gaussian, the decoder transfer characteristic *does not* depend on the channel.

An EXIT chart is obtained by plotting the detector and decoder transfer characteristics on the same plot (since the extrinsic information from the detector/decoder becomes the APP input to the decoder/detector after deinterleaving/interleaving). The convergence threshold is the minimum \mathcal{E}_b/N_0 required to raise the detector curve high enough to open a tunnel between the detector and decoder trajectories. Fig. 5.10 shows EXIT curves for coherently detected 4-CPFSK and rate 1/2 [7, 5] NRC code in an AWGN channel.

The MATLAB modules for generating detector transfer characteristics are given in Appendix C.



Figure 5.13: Decoder mutual information transfer characteristics for rate 1/2, non-recursive convolutional codes with constraint lengths K = 2, 3 and 5.



Figure 5.14: Decoder mutual information transfer characteristics for rate 1/2, NRC and RSC codes.



Figure 5.15: Extrinsic information transfer chart for 4-CPFSK a rate 1/2 NRC code with octal generators [7, 5].

5.4 Chapter Summary

In this chapter, the role of mutual information as a fundamental performance metric in coded communication systems was considered. Since the Shannon capacity assumes Gaussian distributed channel inputs, it is neither the most accurate indicator of system performance, nor does it allow for a meaningful comparison between different modulations. A more appropriate performance measure is hence the modulation constrained capacity. Two popular approaches to coded system design, CM and BICM are considered and a detailed exposition of their capacities is given. Our assumption throughout this chapter (and most of the dissertation) is that the channel is *ergodic*. For non-ergodic (slow or block fading) channels, a Shannon type channel capacity does not exist; instead the performance under non-ergodic channels is characterized in terms of the *information outage probability* [101], [102] which serves as a lower bound of the frame error rate.

The complexity involved in the constrained capacity calculations precludes a closed form solution for the same under sophisticated modulations. We hence resort to computing the capacity using Monte-Carlo integration. This is a singular feature of the optimization problems considered in this dissertation wherein the cost function (i.e. capacity) is not known in closed form and its accuracy hence depends on the number of Monte-Carlo trials. This however is not a critical impairment since the capacity obtained using Monte-Carlo integration attain "steady state" (i.e. converge to their true values) when the number of trials is sufficiently large eg. 1 million symbols per SNR point. Additionally, the capacity calculations require a significantly shorter run time than error rate simulations, especially when simulating sophisticated modulation schemes such as CPM with capacity approaching codes.

EXIT charts and their use in predicting the convergence of systems with BICM-ID have been discussed. It is noted that since EXIT charts essentially measure the mutual information, they compliment a capacity-based optimization rather well should the system be extended to iterate between the detector and the decoder. As will be shown in Chapter 6, both the constrained capacity and EXIT charts are very practical indicators of the coded system performance. This, coupled with their computational feasibility makes them a very effective performance measure and design tool for modern communication systems.

Chapter 6

Optimization Results for Bit-Interleaved Coded CPM

The goal of this chapter is to use the information-theoretic framework derived in Chapter 5 to determine the minimum value of \mathcal{E}_b/N_0 required to achieve reliable signaling for a particular choice of modulation and receiver implementation. Using Monte Carlo integration, the BICM capacity C in bits per channel use can be found using

$$C = \log_2 M - \frac{1}{\log(2)} \sum_{i=1}^{\log_2 M} E\left[\max *\{0, z_i(-1)^{b_i}\}\right].$$
(6.1)

Since the capacity is found by measuring the mutual information between the modulator input and the soft-detector output, it takes into account the constraints of not only the (simulated) modulation and channel, but also the constraints imposed by the receiver formulation. This allows receiver design issues, such as the choice of phase regions for the soft-output, soft-decision differential phase detector (SO-SDDPD, Chapter 4), to be studied in terms of the impact on the constrained capacity. In fact, any receiver design (e.g. coherent, noncoherent, differential) can be characterized in terms of the constrained capacity. All that is required is for the detector to produce a LLR.

The capacity in (6.1), normalized by $\log_2 M$ evaluated at a particular value of \mathcal{E}_s/N_0 is the minimum code rate r of the binary code required to achieve reliable signaling for \mathcal{E}_s/N_0 equal to or greater than that point (since capacity is a monotonically increasing function of \mathcal{E}_s/N_0). This functional relationship could be inverted to give the minimum \mathcal{E}_b/N_0 required for a particular code rate r. The corresponding minimum \mathcal{E}_b/N_0 can then be found by using the relationship

$$\mathcal{E}_b = \frac{\mathcal{E}_s}{r \log_2 M}.\tag{6.2}$$

Fig. 6.1 shows the constrained capacity in bits per symbol C as a function of \mathcal{E}_s/N_0 for M-ary, bit-interleaved coded continuous phase frequency shift keying (BICCPFSK)¹ with different of h and coherent reception in AWGN. These curves were generated using Monte-Carlo simulations with 2 million symbols generated at every \mathcal{E}_s/N_0 . Fig. 6.2 shows \mathcal{E}_b/N_0 as a function of r for binary-CPFSK with different h and coherent reception, in AWGN. It is interesting to note that going to a lower r does not necessarily improve the energy efficiency. We conjecture that at low code rates, the value of \mathcal{E}_s/N_0 is very small. Since the energy per bit is spread out over multiple trellis sections and the energy of the received signal, per trellis section is very small, there is a high probability that the detector would stray from the correct path in the trellis. One can see that for each choice of h, there is a particular value of r that minimizes the \mathcal{E}_b/N_0 .

From the above discussion, it is apparent that given any choice of modulation parameters (h, M, and pulse shape g(t)), receiver design, channel model, and code rate r, the minimum value of \mathcal{E}_b/N_0 can be found under these constraints. As the constraints change, so does the minimum value of \mathcal{E}_b/N_0 and what we seek is to determine its global minimum for all possible values of h, M, g(t), and r for the particular channel and receiver. On one hand, we could allow the set (h, M, and g(t)) to assume any value. However, the resulting solution could have a high bandwidth (for instance, a low r or high h) or it could result in a high complexity receiver (for instance a g(t) with a large width L). To constrain the search space to reasonable solutions, we first impose a bandwidth constraint. The bandwidth constraint requires that (h, M, g(t)) and code rate) be related in a manner discussed below.

In Chapter 3, we had defined the normalized double sided 99% power bandwidth as $B = 2B_{99}T_b$ Hz/bps (T_b is the bit-period), which depends on the parameters (h, M and

¹CPFSK is full-response CPM with rectangular (REC) pulse shaping



Figure 6.1: Capacity in bits per channel use for *M*-ary BICCPFSK with h = 3/4, 1/2, 1/5, and 1/10 and coherent detection, in AWGN.



Figure 6.2: Information-theoretic \mathcal{E}_b/N_0 versus code rate for binary CPFSK with h = 3/4, 1/2, 1/5, and 1/10 with coherent detection, in AWGN.

g(t)). The spectral efficiency η of the coded system is defined to be

$$\eta = r_b / 2B_{99}$$

= $r_s r \log_2 M / 2B_{99}$
= r/B , (6.3)

in units of bits-per-second-per-Hz (bps/Hz). As is common in the literature, the bandwidth efficiency of coded CPM systems can also be quantified in terms of the normalized double sided 99% coded power bandwidth B_{coded} such that

$$B_{coded} = 1/\eta, \tag{6.4}$$

in Hz/bps. If we constrain our BICCPM system to have a spectral efficiency of *exactly* η bps/Hz, then, the minimum allowable code rate for our system is given by

$$r' = \eta B. \tag{6.5}$$

The \mathcal{E}_b/N_0 required for an arbitrarily low bit error rate at η can be found using (6.2) with r = r'. When the spectral efficiency is constrained to be *at least* η , the range of allowable code rates for our system becomes $r \in [r', 1]$. Rates r < r' cannot be considered since the resulting spectral efficiency will be lower than η . Due to the non-monotonous nature of the \mathcal{E}_b/N_0 versus code rate curves, the value of the code rate which minimizes the required \mathcal{E}_b/N_0 could be higher than r' i.e. the optimum code rate could be greater than the minimum code rate.

The next constraint is on the type of detector used for CPM. We first constrain the receiver to use the optimum coherent detector described in Chapter 4. Our motivation for using the coherent detector is that since it is widely studied in the literature, our results could serve as a design benchmark for interested researchers. However, the coherent receiver has high complexity and imposes restrictions on the search space. For instance, its complexity increases exponentially with L. Additionally, with coherent detection h must have rational form $h = m_h/p_h$ and the complexity increases with p_h . The rationality constraint greatly reduces the number of values of h that can be considered in the search space. Therefore, in addition to the bandwidth constraint, we also impose a complexity constraint on the

receiver. The complexity constraint requires that the detector be differential (soft-output, soft decision differential phase detector) and contain no more than M^2 states.

The remainder of this chapter is organized as follows. We begin this chapter by applying our capacity-based optimization to coherently detected BICCPFSK and select partial response BICCPM formats. Next, we identify the combination of modulation parameters and code rates which have the best energy efficiency at different spectral efficiencies, under constraints of differential detection using the SO-SDDPD. Bit error rate (BER) curves using some off-the-shelf binary turbo codes are presented to corroborate our capacity results. Finally, BICCPM with iterative detector and decoding is investigated as a means of mitigating performance performance loss due to bit-interleaved coded modulation.

6.1 Capacity-Based Parameter Optimization of BIC-CPM with Coherent Detection

6.1.1 Design of CPFSK under Bandwidth Constraints

We now present optimization results for coherently detected CPFSK across a reasonable representative search space. We consider the alphabet sizes $M \in \{2, 4, 8, 16\}$ and modulation indices $0.1 \le h \le 1$. For non-binary modulation, natural and gray constellation labelling is considered. The channel is assumed to AWGN.

Observe from Fig. 6.1 that the energy efficiency improves with increasing h. The capacity can be increased by increasing the alphabet size M. Increasing the alphabet size also improves the energy efficiency, especially at smaller values of h. However, increasing M also increases the signal bandwidth (Chapter 3).

For a particular spectral efficiency η , the minimum allowable code rate r' for the pair (M, h) was determined from (6.5). As an example, Fig. 6.3 shows r' as a function of h at different M, when $\eta = 3/4$ bps/Hz. As expected, increasing h necessitates an increase in r' (and consequently a possible decrease in the energy efficiency). For a particular h, a larger alphabet size (M > 4) also requires a larger r'. M = 4 is an interesting scenario. Recall from Chapter 3 (Fig.) that M = 4 is more bandwidth efficient than M = 2 (for



Figure 6.3: Minimum allowable code rate as a function of h at $\eta=3/4$ bps/Hz for M-ary CPFSK .

most values of h) and M > 4 for all h. This allows us to use a lower code rate for the same spectral efficiency, relative to the other alphabet sizes. Once the minimum code rate for a (M, h) pair is determined, the optimum code rate $r \in [r', 1]$ and the corresponding minimum \mathcal{E}_b/N_0 can be found using the constrained capacity. Fig. 6.4, Fig. 6.5, 6.6, 6.7 show the information-theoretic minimum \mathcal{E}_b/N_0 as a function of h for M = 2, 4, 8 and 16 respectively.

For a particular η there is an optimal choice of h that minimizes the required \mathcal{E}_b/N_0 . For instance, the minimum \mathcal{E}_b/N_0 for binary CPFSK at $\eta = 1/2$ is 2.39 dB, obtained using the pair (h = 3/5, r = 0.64). It is interesting to note that the popular MSK (binary CPFSK with h = 1/2) is not the optimum choice at any spectral efficiency. Fig. 6.8 shows the effect of the bit-to-symbol labelling on the performance of 4-CPFSK at different η . Gray labelling is seen to be better than natural labelling at higher spectral efficiencies. This trend is also observed for M > 4.

By finding the minimum value of \mathcal{E}_b/N_0 at different M, h and bit-to-symbol mappings over a wide range of η , one can determine the capacity of coherently detected BICCPFSK. This is simply the minimum \mathcal{E}_b/N_0 required for reliable signaling at different η and is shown in



Figure 6.4: Minimum \mathcal{E}_b/N_0 for reliable signaling required by binary BICCPFSK with coherent detection as a function of h, at different η , in AWGN.



Figure 6.5: Minimum \mathcal{E}_b/N_0 for reliable signaling required by M = 4 BICCPFSK with coherent detection as a function of h, at different η , in AWGN.



Figure 6.6: Minimum \mathcal{E}_b/N_0 for reliable signaling required by M = 8 BICCPFSK with coherent detection as a function of h, at different η , in AWGN.



Figure 6.7: Minimum \mathcal{E}_b/N_0 for reliable signaling required by M = 16 BICCPFSK with coherent detection as a function of h, at different η , in AWGN.



Figure 6.8: Effect of bit-to-symbol mapping on the energy efficiency of M = 4 BICCPFSK with coherent detection.

Fig. 6.9. The corresponding optimum modulation indices as a function of η is shown in Fig. 6.10, and the optimum code rates are shown in Fig. 6.11. The most significant improvement in energy efficiency occurs by increasing M from 2 to 4. While there is a benefit to increasing M at very low η , these benefits diminish as η is increased. It is observed that gray labelling is more energy efficient at h < 1/2, whereas natural labelling is better at larger values of h.

6.1.2 Design of Partial Response CPM under Bandwidth Constraints

Partial response CPM (PR-CPM) yields a more compact power spectrum than CPFSK. This implies that for a particular spectral efficiency, one could use a larger h (or smaller r) relative to CPFSK due to which the capacity of PR-CPM could be greater. Fig. 6.12 reveals that binary PR-CPM with 3-REC and 3-RC pulse shapes provides significant improvements in energy efficiency, especially at high spectral efficiencies. Also, observe that raised cosine pulse-shaping performs better than rectangular pulse shaping. The price to pay for this



Figure 6.9: Minimum \mathcal{E}_b/N_0 required for reliable signaling at different spectral efficiencies for coherently detected BICCPFSK, in AWGN.



Figure 6.10: Optimum code rate at different spectral efficiencies for coherently detected BICCPFSK, in AWGN.



Figure 6.11: Optimum h at different spectral efficiencies for coherently detected BICCPFSK, in AWGN.

improvement is a more complex detector. Recall from Chapter 3 that for a particular Mand h, PR-CPM increases the number of detector trellis states by a factor of M^{L-1} over CPFSK.

6.2 Capacity-Based Parameter Optimization of BIC-CPM with Differential Detection

Our optimization results in the previous section reveal that energy efficient signaling at spectral efficiencies $\eta > 1$ bps/Hz necessitates the use of a combination of partial response CPM, non-binary alphabets and small h. This results in a significant increase in the complexity of the coherent receiver. In comparison, the complexity of the differential detector can be significantly lower. Recall that the formulation of the SO-SDDPD is independent of the choice of h and it is constrained to contain no more than M^2 states. This makes differential detection an attractive alternative to coherent detection, when high spectral efficiency, low-complexity applications.



Figure 6.12: Minimum \mathcal{E}_b/N_0 required at different spectral efficiencies for coherently detected BICCPM, in AWGN. Binary partial response signaling using 3 RC and 3 REC pulse shapes give significant improvement in the energy efficiency over binary CPFSK.

6.2.1 Information Rate Loss Relative to Coherent Detector

The main concern regarding differential detection is its poor energy efficiency relative to coherent detection. We show now that at high spectral efficiencies, the information rates achievable with differential detection can approach coherent detection while enjoying a much lower complexity.

Fig. 6.13 shows the minimum \mathcal{E}_b/N_0 required at different spectral efficiencies with coherent as well as differential reception. Observe that the performance gap between the coherent receiver and differential receiver becomes less pronounced as the spectral efficiency is increased. Given how close differential and coherent receivers perform, we anticipate that the best differential design may in fact outperform the best coherent design of reasonable complexity at high spectral efficiencies.



Figure 6.13: Information-theoretic minimum \mathcal{E}_b/N_0 as a function of h at different spectral efficiencies with coherent and differential detection. The channel is AWGN. The modulation is binary CPFSK.

6.2.2 Design of Partial Response CPM under Bandwidth Constraints

We now perform a capacity-based parameter optimization of BICCPM with differential phase detection, using the SO-SDDPD. As with coherent detection, we limit the search space to the following sensible alternatives. The pulse shapes may be either GFSK with parameters $B_gT = 0.5, 0.25$ and 0.2 or RC with L = 3 or 5. We consider the code rates $r = \{6/7, 5/6, 3/4, 2/3, 1/2, 1/3, 1/4, 1/5\}$. We limit the modulation order M to be either 2 or 4 (natural bit-to-symbol mapping). The receiver is restricted to use SO-SDDPD with uniformly space phase regions with R = 40 for M = 2 and R = 26 for M = 4. Finally, the normalized bandwidth is constrained to $B_{coded} = \{0.4, 0.6, 0.8, 0.9, 1.0, 1.2\}$. The value of his determined by the choice of the other parameters along with the bandwidth constraint. The optimization is run for a Rayleigh channel (K = 0) and a Rician channel with K = 6dB.

Because the search space includes 5 pulse shapes, 8 code rates, and 2 modulation orders,

there are 80 design points to consider for each of the 6 bandwidth efficiencies and 2 channel types. For each design point, bandwidth constraint, and channel, a capacity curve was generated using Monte Carlo integration, and from the curve the minimum \mathcal{E}_b/N_0 was found by reading off the value of \mathcal{E}_s/N_0 for which $C = r \log_2 M$ and then converting to $\mathcal{E}_b/N_0 =$ $\mathcal{E}_s/N_0/C$. Then for a particular channel and bandwidth constraint, the design point with the smallest minimum \mathcal{E}_b/N_0 was selected and declared the best design for that channel and bandwidth constraint.

Because a total of 960 capacity curves were generated, a full account of the results cannot be given in this paper. Figs. 6.14 and 6.15 show representative intermediate results. In Fig. 6.14, results are shown in Rician fading (K = 6 dB) for the subset of design points that contain r = 2/3 and GFSK pulse shapes. For each of the six design points, a curve is given showing the theoretic required \mathcal{E}_b/N_0 as a function of the bandwidth constraint. Also listed is the value of h for the curve with smallest required \mathcal{E}_b/N_0 . For all but the largest bandwidth, the design point M = 4, $B_g T = 0.5$ has the lowest required \mathcal{E}_b/N_0 . In Fig. 6.15, the bandwidth is held fixed at $B_{coded} = 0.8$ in a Rayleigh fading channel, and the minimum \mathcal{E}_b/N_0 for each code rate is shown (with the design that achieves this minimum so indicated in the figure legend). From the curve, it is apparent that r = 3/4 is the best code rate for this bandwidth constraint, and that the other design parameters are M = 4, h = 0.25, and GFSK pulse shaping with $B_qT = 0.5$. Fig. 6.15 also illustrates the tradeoff between between code rate and CPM parameters at a fixed bandwidth. As an example, for GFSK, when R_c is lowered from 6/7 to 3/4, an improvement in the energy efficiency is seen due to increased coding gain. However, when r is lowered below 3/4, the scaling of CPM parameters (primarily h) not only offsets any potential coding gain, but in fact worsens the performance as indicated by the increasing required \mathcal{E}_b/N_0 .

The results of the search are tabulated in Table 6.1 for Rayleigh fading and Table 6.2 for Rician fading (K = 6). The tables indicate the design parameters that minimize \mathcal{E}_b/N_0 for each bandwidth constraint, and the corresponding value of the minimum \mathcal{E}_b/N_0 . In the case of GFSK pulse shaping, the value of B_gT is given, while in the case of RC pulse shaping, Lis given. As can be seen from the tables, GFSK outperforms RC-CPM except at the most relaxed bandwidth constraint. Further insight into the performance of the SO-SDDPD with



Figure 6.14: Minimum required \mathcal{E}_b/N_0 as a function of normalized bandwidth B_{coded} for BICCPM in Rician fading (K = 6 dB). The code rate is r = 2/3, GFSK modulation used with $M \in \{2, 4\}$, and SO-SDDPD. The numbers denote modulation indices corresponding to GFSK parameters with the lowest information-theoretic limit on \mathcal{E}_b/N_0 at different B_{coded} .

different pulse shapes can be obtained from Table 4.1 and Table 4.2 in Chapter 4. $B_gT = \infty$ and L = 1 indicate full response CPM ($Z_f = Z_p = 0$). As B_gT is lowered or L is increased, the amount of ISI increases. It is interesting to note that for a given pulse width, the amount of ISI in the RC pulse shape is greater than GFSK. As a consequence, for the same M, hand pulse width, RC-CPM would have a more compact power spectrum than GFSK. This also implies that at a fixed B_{coded} , r, M and pulse width, RC-CPM allows us to have a higher value of h. Typically (but not necessarily), larger values of h result in lower values of \mathcal{E}_b/N_0 , however since the induced ISI is also greater, the RC-CPM may not always have better energy efficiency.



Figure 6.15: Minimum required \mathcal{E}_b/N_0 as a function of code rate for BICCPM with GFSK signaling in Rayleigh fading with SO-SDDPD and under bandwidth constraint $B_{coded} = 0.8$. The legend specifies the GFSK parameters (M, h, B_gT) that achieve this minimum. Under the given constraints, the design $\{r = 3/4, M = 4, h = 0.25, B_gT = 0.5\}$ has the best energy efficiency.

Table 6.1: Combination of code rates and CPM parameters with lowest information theoretic minimum \mathcal{E}_b/N_0 under the constraint of using SO-SDDPD in Rayleigh fading at different B_{coded} .

B_{coded}	Rate	M	Pulse shape	$\{L, B_g T\}$	h	$\mathcal{E}_b/N_0 \mathrm{~dB}$
0.4	5/6	4	GFSK	0.25	0.14	20.24 dB
0.6	5/6	4	GFSK	0.25	0.26	14.92 dB
0.8	3/4	4	GFSK	0.5	0.25	12.38 dB
0.9	2/3	4	GFSK	0.5	0.24	11.99 dB
1.0	2/3	4	GFSK	0.5	0.3	11.44 dB
1.2	6/7	2	RC	3	0.73	10.97 dB

Table 6.2: Combination of code rates and CPM parameters having lowest information theoretic minimum \mathcal{E}_b/N_0 under the constraint of using SO-SDDPD in Rician fading (K = 6dB) at different B_{coded} .

B_{coded}	Rate	M	Pulse shape	$\{L, B_g T\}$	h	$\mathcal{E}_b/N_0 \mathrm{~dB}$
0.4	6/7	4	GFSK	0.5	0.097	16.52 dB
0.6	5/6	4	GFSK	0.5	0.18	$11.67 \mathrm{~dB}$
0.8	5/6	4	GFSK	0.5	0.29	9.09 dB
0.9	3/4	4	GFSK	0.5	0.285	$8.87 \mathrm{~dB}$
1.0	2/3	4	GFSK	0.5	0.3	$8.83~\mathrm{dB}$
1.2	6/7	2	RC	3	0.73	8.32 dB

6.3 Validating Design Parameters using BER Simulations

To confirm that the constrained capacity is indeed a good indicator of the performance of an actual coded system, bit error rate (BER) simulations were performed for the proposed BICCPM system using off-the-shelf binary capacity-approaching codes. Fig. 6.16 shows the bit error rate (BER) of bit-interleaved coded 2-CPFSK at modulation indices $\{1/10, 1/7, 1/2\}$ using a rate 1/2 CDMA-2000 turbo code [38], after 10 decoder iterations with an interleaver length of 24576 bits in AWGN. The vertical lines denote the information theoretic thresholds to achieve an arbitrarily low BER for the particular modulation index at r = 1/2.

BER curves were also generated using the UMTS turbo code [46] and SO-SDDPD detection. The codeword length used was $N_b = 6720$ bits, and the decoder performed 16 decoder iterations. While the mother code rate is r = 1/3, rate matching was performed to obtain higher code rates. Enough trials were run at each \mathcal{E}_b/N_0 to record 30 frame errors.

Simulations were performed for both Rayleigh fading and Rician fading with K = 6 dB. In each case, the optimal parameters were chosen from Tables 6.1 and 6.2 for the bandwidth constraint $B_{coded} = 0.9$. The BER in Rayleigh fading for 4-GFSK with parameters h = 0.24, $B_gT = 0.5$, and r = 2/3 is shown in Fig. 6.17. From this curve, it is seen that the \mathcal{E}_b/N_0 required to achieve a BER of 10^{-5} is 12.93 dB. This is not far from the minimum theoretical \mathcal{E}_b/N_0 of 11.99 predicted by Table 6.1. The BER in Rician fading (K = 6 dB) for 4-GFSK with parameters h = 0.285, $B_gT = 0.5$, and r = 3/4 is shown in Fig. 6.2. In this case, the \mathcal{E}_b/N_0 required to achieve a BER of 10^{-5} is 9.52 dB, which is close to the theoretical



Figure 6.16: Bit error rate in AWGN for bit-interleaved coded, 2-CPFSK with h = 1/10, 1/7and 1/2 using a rate 1/2 CDMA 2000 code after 10 turbo decoder iterations. The vertical lines denote the information theoretic \mathcal{E}_b/N_0 in dB to achieve an arbitrarily low BER for the respective h and r = 1/2. The interleaver size is 24576 bits.

minimum of 8.87 dB (from Table 6.2). These results confirm that constrained capacity is a practical measure of BICCPM performance, since it is possible to signal within 1 dB of the capacity by using off-the-shelf binary, capacity approaching codes.

It is also informative to compare the performance of the coded system against that of an uncoded system with the same spectral efficiency. Additional simulations of uncoded CPM using SDDPD detection and binary GFSK with h = 0.5 and $B_gT = 0.3$ are shown in Fig. 6.2 and 6.17 for the Rayleigh and Rician channels, respectively. These parameters were chosen because they are what is used in the GSM standard and because they also result in $B_{coded} = 0.9$. From these figures, it can be seen that the coding gain at BER = 10^{-5} is 16 dB in Rayleigh fading and 14 dB in Rician fading.



Figure 6.17: BER of coded (solid line) and uncoded (dotted line) GFSK in Rayleigh fading under bandwidth constraint $B_{coded} = 0.9$ using SO-SDDPD. The coded (BICCPM) system system uses a rate r = 2/3, length $N_b = 6720$ turbo code, 16 decoder iterations, R = 26phase regions, and GFSK parameters M = 4, h = 0.24, and $B_gT = 0.5$. The uncoded system uses R = 40 phase regions and GFSK parameters M = 2, h = 0.5 and $B_gT = 0.3$.

6.4 Design of Coded CPM with Iterative Detection and Decoding

While bit-interleaved coded modulation (BICM) is convenient to design and implement and provides higher diversity in fading channels, its capacity is lower than coded modulation (CM) due to the data processing inequality [103] (Appendix A). One method to mitigate the performance due to BICM and approach the CM capacity is by iteratively exchanging bit-wise *extrinsic information* between the detector and the decoder. Such a process is often called *bit-interleaved coded modulation with iterative decoding* (BICM-ID) [89] and here we refer to its extension to CPM as *BICCPM-ID*. Our goal in this section to extend our



Figure 6.18: BER of coded (solid line) and uncoded (dotted line) GFSK in Rician fading (K = 6 dB) under bandwidth constraint $B_{coded} = 0.9$ using SO-SDDPD. The coded (BICCPM) system system uses a rate r = 3/4, length $N_b = 6720$ turbo code, 16 decoder iterations, R = 26 phase regions, and GFSK parameters M = 4, h = 0.285, and $B_gT = 0.5$. The uncoded system uses R = 40 phase regions and GFSK parameters M = 2, h = 0.5 and $B_gT = 0.3$.

optimization to BICCPM-ID under bandwidth constraints. Since iterative decoding does not change the spectral efficiency, the minimum code rate r required at a specific η for a particular (M, h, g(t)) is unchanged relative to BICM. The problem now boils down to identifying the set of M, h, g(t), constellation labelling and outer code which has the lowest \mathcal{E}_b/N_0 required for reliable signaling at some η . A similar optimization is attempted for partial response CPM with convolutional codes in [41] using a union bound analysis based on evaluating the distance spectrum of the coded system. However, such analysis assumes ML decoding and is hence not the most appropriate tool for predicting the performance of an iterative decoding system. The authors of [41] make the same observation later in [100]. We instead use extrinsic information transfer (EXIT) charts (Chapter 5) [95] which have been widely used to analyze system performance with iterative decoding [104], [44], [97].



Figure 6.19: Minimum \mathcal{E}_b/N_0 in dB required for reliable signaling as a function of h for coherently detected CPFSK at $\eta = 1/2$ bps/Hz. The channel is AWGN. The dotted curves denote BICCPFSK convergence thresholds found from the constrained capacity. The dashed curves denote BICCPFSK-ID convergence thresholds with a NRC [7, 5] convolutional code, predicted by EXIT chart analysis. The alphabet sizes are $M \in \{2, 4, 16\}$ with natural bit-to-symbol labelling.

Since EXIT charts work by measuring the mutual information at the detector and decoder's output, it compliments our BICM capacity analysis, such that modulation parameters with a higher BICM capacity typically need a lower \mathcal{E}_b/N_0 to converge (i.e. attain an arbitrarily low error rate).

It is noted that EXIT chart analysis has been previously considered for coherently detected CPM with rate 1/2 outer convolutional codes in [100]. EXIT charts and density evolution [55] are used to design rate 1/2 convolutional codes and LDPC codes respectively for coherently detected MSK in [33]. More recently, irregular repeat-accumulate codes (IRA) [35] and ring convolutional codes [42] have also been applied in an iterative (coherent) detection-decoding framework for CPM.

6.4.1 Design of BICCPFSK-ID with Coherent Detection

Rather than begin our optimization from scratch, we use our BICM optimization results as a starting point, since intuitively, the pair of M and h with the lowest \mathcal{E}_b/N_0 for BICM would also lift the detection characteristic proportionally, which might result in a lower convergence threshold. This is illustrated in Fig. 6.19.

For the outer code, we consider constraint length 2 and 3 non-recursive convolutional (NRC) codes and a constraint length 3 recursive convolutional (RSC) code (see Chapter ?? for details). The constraint length 2 NRC codes use generators polynomials (octal) [2,3]. The two constraint length 3 NRC codes use the polynomials [7,5] and [7,4] respectively. The RSC code employs the feedback/ feedforward polynomials [7,3]. Table 6.4 lists the modulation parameters and outer codes which have the lowest \mathcal{E}_b/N_0 at $\eta \in \{1,3/4,1/2\}$. It is observed that the minimum code rate r' and hence h tends to be lower than the corresponding BICM parameters. As an example, the pair of h and r' for 4-CPFSK and $\eta = 1$ with the lowest \mathcal{E}_b/N_0 for BICM is (3/10, 0.92) with gray labelling, and (1/4, 0.8) with gray labelling for BICM-ID.

We have observed that for a particular (M, h, r'), the optimum bit-to-symbol mappings tend to be different for BICCPFSK and BICCPFSK-ID. As reported in [29], the combination of certain values of M, h and mapping leads to an early error floor. Hence, while h = 1/2with natural mapping is the optimum choice for M = 8 at $\eta = 1/2$ (Table 6.3), it is not a good choice for BICM-ID as it would lead to an early error floor.

It was reported in [100] and [33] that lowering the code constraint length lowers the convergence threshold. While this is true for the rate 1/2 codes considered in the papers, we have found that lowering the constraint length does not guarantee a lower convergence threshold at all code rates. The [2, 3] performs better in the bottleneck region of the curve. However, since its transfer characteristic is steeper, it could potentially perform worse than than the [7, 5] code at higher code rates. This is seen in Fig. 6.20. At $\mathcal{E}_s/N_0 = 2.2$ dB, the transfer characteristics for the [2, 3] NRC code and the [7, 4] code fail to exit the bottleneck region, while the [7, 5] code manages to emerge out of the bottleneck region. Our results also indicate that the performance of the RSC [7, 3] code is very similar to the [7, 4] NRC


Figure 6.20: EXIT curves for 4-CPFSK with h = 1/2, gray labelling and various outer convolutional codes. The code rate is r = 0.64. The channel is AWGN. The figure indicates that for the particular r, the NRC [7, 5] code yields the lowest convergence threshold.

code at higher code rates.

Finally, select bit error rate curves (BER) were generated to corroborate the EXIT chart analysis. Fig. 6.21 shows BER curves for 4-CPFSK with h = 3/7 and natural labelling using NRC [2, 3] and NRC [7, 5] convolutional codes. With a code rate of r = 0.6, the spectral efficiency for the system is $\eta = 1/2$ bps/Hz. The interleaver size is 100000 bits. The curves show BER after 25 iterations. The vertical lines indicate the convergence thresholds, predicted using EXIT charts. Note that the system with the [2, 3] code shows an error floor due to the poor free distance of the punctured code. 20 frame errors were recorded per \mathcal{E}_b/N_0 (fewer errors were logged at the highest SNR).

6.4.2 Design of BICCPM-ID with Differential Detection

Fig. 6.22 shows the mutual information transfer characteristics for binary GFSK with $(h = 0.315 B_g T = 0.5)$ and soft-output, soft-decision differential phase detection. The channel is Rician, with K = 2 dB. Observe that these are essentially straight lines, with



Figure 6.21: BER after 25 iterations for 4-CPFSK with h = 3/7, natural labelling and two outer convolutional codes. The code rate is r = 0.6. The channel is AWGN. The vertical lines indicate convergence thresholds, predicted using EXIT charts.

Table 6.3: Information theoretic minimum \mathcal{E}_b/N_0 in dB for non-iterative BICM in AWGN at different η .

η	M	h	Mapping	r'	$\mathcal{E}_b/N_0 \mathrm{~dB}$
1	2	1/4	Natural	0.9	7.91
	4	3/10	Gray	0.92	4.32
	8	1/5	Gray	0.75	4.24
	16	1/7	Gray	0.76	5.03
3/4	2	3/7	Natural	0.847	3.69
	4	3/7	Gray	0.846	2.62
	8	3/10	Gray	0.77	2.79
	16	1/5	Gray	0.74	3.09
1/2	2	3/5	Natural	0.64	2.39
	4	4/7	Natural	0.72	2.06
	8	1/2	Natural	0.77	1.87
	16	1/3	Gray	0.75	1.93

η	M	h	Mapping	r'	Code	$\mathcal{E}_b/N_0 \mathrm{~dB}$
1	2	1/4	Natural	0.9	[7, 5]	6.86
	4	1/4	Gray	0.8	[7, 5]	2.958
	8	1/5	Natural	0.75	[2, 3]	2.38
	16	1/10	Natural	0.6	[2, 3]	2.99
3/4	2	3/7	Natural	0.847	[7, 5]	2.46
	4	2/5	Natural	0.81	[2, 3]	1.204
	8	1/4	Natural	0.67	[7, 4]	1.177
	16	1/5	Natural	0.74	[2, 3]	1.428
1/2	2	4/7	Natural	0.62	[7, 4]	0.476
	4	3/7	Natural	0.6	[2, 3]	0.508
	8	1/3	Natural	0.56	[2, 3]	0.346
	16	1/5	Gray	0.5	[2, 3]	0.389

Table 6.4: Minimum \mathcal{E}_b/N_0 in dB for BICM-ID in AWGN with outer convolutional codes at different η .

two points of interest 1) When $I_{\mathbf{v}} = 0$ (no a priori information at the SO-SDDPD), hence $I_{\mathbf{z}}$ is the BICM capacity, 2) $I_{\mathbf{z}} = 1$, i.e. perfect a priori knowledge on all bits except a_k , which implies that if the slope of the line is steep, BICM-ID would give significant gains over BICM. In our case, the line is almost horizontal hence implying little/ no benefit over BICM. A possible reason could be that $B_gT = 0.5$ causes only a little adjacent symbol interference. Hence during BICM-ID, extrinsic information for each modulated bit is provided by only two other bits, which appears to be insufficient to give noticeable improvement over BICM.

The transfer characteristics for the cyclic code used in the Bluetooth standard is also shown in Fig. 6.22. Note that the curve passes through the point (0.5, R') where R' = 10/15is the code rate. The EXIT chart is obtained by plotting the SO-SDDPD and decoder characteristics together. The threshold is simply the minimum \mathcal{E}_b/N_0 at which both curves progress all the way to the right without intersecting. Fig. 6.22 reveals that the threshold for our receiver is at $E_b/N_0 = 21$ dB.

6.5 Chapter Summary

We have investigated the problem of designing bit-interleaved coded CPM systems under bandwidth constraints. Our goal was to identify the combination of CPM and channel code parameters that have the best energy efficiency at some required bandwidth efficiency. We



Figure 6.22: EXIT chart for the proposed BICM receiver for Bluetooth specifications (h = 0.315, $B_gT = 0.5$). SO-SDDPD EXIT curves assume Rician channel with K = 2 dB, R = 24 uniform phase regions. Note that the decoder's EXIT curve intersects (0.5, R'), where R' = 10/15.

treat this as an optimization problem, with the modulation constrained capacity as the cost function driving our optimization. Since the entire modulation and code search space has infinite parameters, we instead perform optimization over a smaller search space which includes some popular modulation formats such as CPFSK, partial response GFSK and raised cosine CPM. Interested researchers can just as easily apply our methodology to their choice of CPM parameters.

We have extended the energy-bandwidth analysis to bit-interleaved CPM systems employing iterative extrinsic information exchange between the detector and the decoder. EXIT charts are used to identify the combination of modulation parameters and outer convolutional codes with the best energy efficiency at different spectral efficiencies. We have shown that by a careful selection of modulation parameters and code puncturing pattern, close to 2 dB gains in energy efficiency is possible over BICM by using low-complexity convolutional codes. Though not considered in this work, a more elaborate system design could be considered using LDPC codes [33] or IRA codes [35], in which the degree distribution of the code is optimized with respect to the modulation parameters.

Chapter 7

Conclusion

7.1 Summary

Continuous phase modulation has long been known to be an excellent modulation choice for bandwidth constrained systems. Its use has also been advocated for systems desiring high amplifier power efficiency (e.g. satellite systems). In this dissertation, we have investigated the problem of designing coded CPM systems under bandwidth constraints. This problem has several points of interest. First, the code rate cannot be arbitrarily lowered in order to improve the energy efficiency, since it is possible that doing so will require a change in modulation parameters that can result in a coding loss. Second, the search for coded CPM parameters with the best energy efficiency at different bandwidth efficiencies spans a very large search space. Hence, any performance metric or cost function that we use to perform this search must be feasible to compute for the different modulation parameters, code rates, channel conditions, and receiver formulations considered. Additionally, such a cost function should also be a realistic indicator of the system performance.

Previous attempts at addressing the above problem have focused on coherently detected systems, using a cost function which is based on the distance spectrum of the concatenated system. The resulting bounds predict the performance of the coded-system at high SNR (typically the error floor region), under the assumption of ML reception. However, it is often of practical interest to optimize with respect to the SNR required to trigger the waterfall region of the error rate curve. While this operating point can be determined by running error rate simulations, such a process is extremely time consuming, especially for sophisticated systems.

In this dissertation, we have outlined a methodology for solving the above problem using the constrained capacity and more generally the mutual information, as the cost function. The benefits of a cost function based on the capacity have been discussed in detail in previous chapters. Briefly, capacity sets the fundamental limits on the performance of a coded system, i.e. it determines the information theoretic limit on the SNR for reliable signaling. As we have demonstrated, it is a very practical indicator of system performance due to advances in code design resulting in the availability of "off-the-shelf" capacity-approaching codes. The capacity also best quantifies the tradeoff between CPM parameters and the code rate. Since most CPM systems and their associated demodulators are too complex to admit closed-form solution, a method for determining the constrained capacity using Monte Carlo integration has been presented. Monte Carlo integration with a large number of trials offer a computationally feasible method for determining the capacity for a wide range of coded CPM parameters, which is significantly faster than running error rate simulations. The constrained capacity also takes into account the detector design.

We have focused our attention on systems employing bit-interleaved coded modulation (BICM) since it has become a standard feature in satellite, wireless and cellular systems. This is because, not only does BICM simplify system design, but it also increases the temporal diversity of the system. Our proposed methodology has been used to perform optimization over a trial search space, for coherently detected CPM.

The complexity of the coherent detector renders it unfeasible for high spectral efficiency applications. Furthermore, it is also suspectable to phase estimation errors. It is hence of practical interest to investigate low complexity alternatives such as noncoherent and differential phase detectors. In this work, we have developed a low complexity, high performance CPM detector, which we call the soft-output, soft-decision differential phase detector (SO-SDDPD). Our detector overcomes the two main drawbacks associated with differential phase detectors, namely unmitigated ISI and hard-symbol decisions. We have shown that the low complexity and excellent performance, make the SO-SDDPD a very attractive alternative to coherent detection, especially at high spectral efficiency. The optimization of coded-CPM systems using the SO-SDDPD has also been undertaken.

The sub-optimality of BICM can be overcome by iteratively exchanging extrinsic information between the detector and the decoder, a process called BICM with iterative decoding (BICM-ID). In fact, BICM-ID is often used as a pragmatic alternative to true ML reception. We have also considered the optimization of convolutionally coded CPM with BICM-ID. EXIT charts have been used to identify CPM parameters and outer convolutional codes with the best energy efficiency at different spectral efficiencies. Since EXIT charts essentially measure the mutual information, the BICM capacity results provide a good starting point for the design since intuitively, the CPM parameters with the lowest \mathcal{E}_b/N_0 for BICM, could yield a lower BICM-ID convergence threshold.

7.2 Ideas for Future Research

7.2.1 Optimization in Non-Ergodic Channels

The block fading channel model, is often used to characterize wireless systems. For such non-ergodic channels, a Shannon-sense capacity does not exist and a more appropriate performance measure is the information outage probability [105]. The outage probability under linear modulation constraints was found in [101] and [102].

In block fading, the codeword **b** is broken into F blocks each of which is transmitted over an independent channel such that c(t) = c remains constant for the entire duration of the block. The instantaneous SNR of the b^{th} block is now, $\lambda_b = |c|^2 \mathcal{E}_s / N_0$. Assuming code combining is used at the receiver, the instantaneous capacity for the entire codeword is

$$C(\lambda_1, \lambda_2, ..., \lambda_F) = \frac{1}{F} \sum_{b=1}^F C(\lambda_b).$$
(7.1)

An information outage is said to occur when the instantaneous capacity $C(\lambda_1, \lambda_2, ..., \lambda_F) < r \log_2 M$. The outage probability is hence [105]

$$p_o[F] = P[C(\lambda_1, \lambda_2, ..., \lambda_F) < r \log_2 M].$$
 (7.2)

The constrained capacities in (7.1) are evaluated as described in Chapter 5. Fig. 7.1 shows the information outage probability using code combining in block fading (F = 1 and F =



Figure 7.1: Information outage probability with code combining in block fading at F = 1 and F = 100 for BICCPM using SO-SDDPD. The combination of code rates and GFSK parameters are selected such that B = 0.9.

100) with SO-SDDPD based BICCPM at B = 0.9. When F = 1, r = 3/4 with M = 4, h = 0.285 and $B_gT = 0.5$ has the lowest outage probability as well as the lowest \mathcal{E}_b/N_0 in AWGN (from Fig. 7.2). Similarly, when F = 100, r = 2/3 with M = 4, h = 0.24 and $B_gT = 0.5$ gives not only the lowest outage probability in block fading, but also yields the lowest \mathcal{E}_b/N_0 in ergodic Rayleigh fading. Additionally, the trends exhibited by the outage probability curves at different r and modulation parameters when F = 1 are identical to the trends exhibited by the ergodic capacity in AWGN and in Rayleigh fading at large F.

These initial results demonstrate that our capacity-based approach in selecting the most energy efficient combination of CPM parameters and code rates in ergodic fading also helps in identifying the coded parameters with the lowest outage probability in block fading, at a desired spectral efficiency. A more extensive search over different spectral efficiencies and detector designs remains a topic for future work.



Figure 7.2: GFSK parameters with the lowest information theoretic minimum \mathcal{E}_b/N_0 (dB) for various code rates at B = 0.9 in AWGN and Rayleigh channels for BICCPM with SO-SDDPD.

7.2.2 Effect of Finite Codeword Lengths on Capacity

Fig. 7.3 shows BER curves for coherently detected MSK using a rate 1/2 CDMA 2000 turbo code at different (data) interleaver lengths , in AWGN. Also shown is the information theoretic limit on \mathcal{E}_b/N_0 obtained by computing the BICCPM capacity of coherent MSK (Chapter 5). Observe that as the codeword length decreases, the BICCPM threshold becomes an increasingly optimistic performance predictor. This is because signaling at the Shannon capacity-limit requires an infinitely long codeword. However, latency requirements in many applications restrict the size of the codewords, for instance, the WiMAX LDPC code uses blocklengths ranging from 576 bits to 2304 bits. It is therefore of practical interest to investigate performance bounds that take into account the effect of the codeword length on the performance.

The effect of codeword length on the performance was quantified using the sphere packing bound in [106]. Given, a certain codeword error probability P_e , capacity C (bits per channel use) and the codelength N_b , the sphere packing bound can be used to determine the minimum



Figure 7.3: BER curves for coherently detected MSK using a CDMA 2000 turbo code, in AWGN.

value of \mathcal{E}_b/N_0 required to signal at P_e . It is more convenient to instead use a sphere packing bound approximation (SPBA) [107], such that the minimum \mathcal{E}_b/N_0 required to attain an error probability of P_e , using a code of length N_b bits is given by [107]

$$(\mathcal{E}_b/N_0)_{SPBA} = (\mathcal{E}_b/N_0)_C + \Delta, \tag{7.3}$$

where

$$\Delta = \sqrt{\frac{20C(2^C - 1)10\log_{10}[1 - P_e]}{N\log_e 10(2^C - 1)}},$$
(7.4)

and \mathcal{E}_b/N_0 is the information theoretic limit on the SNR required to signal reliably at C bits per channel use with infinitely large codewords. Fig. 7.4 shows the \mathcal{E}_b/N_0 required for $P_e = 10^{-4}$ at blocklengths $N_b = 1024$ and $N_b = 16384$, found using (7.3), for 16-QAM in AWGN. Also shown is the CM capacity for 16-QAM. The extension of the sphere packing bound to CPM remains a topic for future work.



Figure 7.4: Minimum \mathcal{E}_b/N_0 required for $P_e = 10^{-4}$ at blocklengths $N_b = 1024$ and 16384 for 16-QAM in AWGN. Also shown is the CM capacity of 16-QAM.

7.2.3 Optimization in the Face of Uncertainty

A singular feature of our capacity-based optimization is that the cost function is evaluated using Monte Carlo simulations due to unavailability of a closed-form expression. The capacity C^i is a function of the SNR \mathcal{E}_s^i/N_0 and can be written as

$$C^i = f(\mathcal{E}^i_s/N_0). \tag{7.5}$$

However, since the function in (7.5) cannot be evaluated in closed form, C_i is not directly observable. The uncertainty associated with the Mote Carlo trials implies that C_i is instead observed over a noisy channel. For any particular \mathcal{E}_s/N_0 , a set of N observations is collected such that

$$\mathbf{c} = f(\mathcal{E}_s/N_0) + \mathbf{n},\tag{7.6}$$

where **n** is a vector of independent and identically distributed Gaussian random variables with zero mean and variance σ^2 . The unbiased estimator for C^i is obtained by finding the sample mean

$$\hat{C}^{i} = \frac{1}{N} \sum_{k=1}^{N} c_{k}$$

$$= f(\mathcal{E}_{s}/N_{0}) + \frac{1}{N} \sum_{k=1}^{N} n_{k}$$

$$= f(\mathcal{E}_{s}/N_{0}) + v, \qquad (7.7)$$

where v is zero-mean Gaussian, with variance σ^2/N . Thus as the observation size increases, the variance of the estimate decreases. The accuracy of the capacity-based cost function hence depends on the observation size. Instead of running a large number of trials over a wide range of SNRs, a more computationally efficient approach would be pick an initial set of SNRs, $SNR_0 \in \{\mathcal{E}_s^1/N_0, \mathcal{E}_s^2/N_0, ..., \mathcal{E}_s^M/N_0\}$ and run a few number of initial trials $\mathcal{N}_0 \in \{N_1, N_2, ..., N_M\}$. We can then "zoom-in" on the range of SNRs interest and run further Monte Carlo trials. Alternatively, we could apply *simulated annealing* [108] by treating the constrained capacity as the internal energy and defining the state as the combination of code rate and CPM parameters.

A Review of Relevant Information Theoretic Concepts

A.1 Entropy of Random Variables

The entropy of a continuous random variable (RV) X with pdf p(X) is

$$H(X) = -\int \log p(x)p(x)dx.$$
 (A-8)

If X is a discrete RV with pmf p(X) and |X| = M,

$$H(X) = -\sum_{x} \log p(x)p(x).$$
(A-9)

If the logarithm has base e, the entropy has units of nats. Alternatively, if the logarithm has base 2, H(X) has units of bits. The entropy can also be expressed as an expectation

$$H(X) = -E[\log p(X)]. \tag{A-10}$$

The conditional entropy between RVs X_i and X_j is

$$H(X_{i}|X_{j}) = -E[\log p(X_{i}|X_{j})] \\ = -\int_{x_{i},x_{j}} \log p(x_{i}|x_{j})p(x_{i},x_{j})dx_{i},dx_{j}.$$
 (A-11)

If X_i and X_j are independent

$$H(X_i|X_j) = H(X_i). \tag{A-12}$$

The joint entropy of a collection of RVs $X_1, X_2, ..., X_n$ is

$$H(X_{1}, X_{2}, ..., X_{n}) = -E[\log p(X_{1}, X_{2}, ..., X_{n})]$$

$$= -\int_{x_{1}, x_{2}, ..., x_{n}} \log p(x_{1}, x_{2}, ..., x_{n}) p(x_{1}, x_{2}, ..., x_{n}) dx_{1}, dx_{2}, ..., dx_{n}$$

$$= -\int_{x_{1}, x_{2}, ..., x_{n}} [\log p(x_{1}) + ... + \log p(x_{n}|x_{n-1}, ..., x_{1})] p(x_{1}, x_{2}, ..., x_{n}) dx_{1}, ..., dx_{n}$$

$$= \sum_{i=1}^{n} H(X_{i}|X_{i-1}, ..., X_{1})$$
(A-13)

If $X_1, X_2, ..., X_n$ are independent

$$H(X_1, X_2, ..., X_n) = \sum_{i=1}^n H(X_i).$$
 (A-14)

A.2 Mutual Information

The mutual information between RVs X and Y is

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = I(Y;X).$$
(A-15)

If X and Y are independent

$$I(X;Y) = 0.$$
 (A-16)

The conditional mutual information I(X; Y|Z) is

$$I(X;Y|Z) = H(X|Z) - H(X|Y,Z).$$
 (A-17)

The chain rule for mutual information is

$$I(X_{1}, X_{2}, ..., X_{n}; Y) = H(X_{1}, X_{2}, ..., X_{n}) - H(X_{1}, X_{2}, ..., X_{n}|Y)$$

$$= \sum_{i=1}^{n} H(X_{i}|X_{n-1}, ..., X_{1}) - H(X_{i}|X_{n-1}, ..., X_{1}, Y)$$

$$= \sum_{i=1}^{n} I(X_{i}; Y|X_{n-1}, ..., X_{1}).$$
(A-18)

A.3 Data Processing Inequality

The random variables $X,\,Y,\,Z$ form a Markov chain $(X\to Y\to Z$) if X and Z are conditionally independent, i.e.

$$p(x, z|y) = p(x|y)p(z|y).$$
(A-19)

Using the chain rule for mutual information (A-18) we have

$$I(X;Y,Z) = I(X;Z) + I(X;Y|Z) = I(X;Y) + I(X;Z|Y).$$
(A-20)

From (A-19)

$$I(X;Z|Y) = 0.$$
 (A-21)

From (A-20)

$$I(X;Z) + I(X;Y|Z) = I(X;Y),$$
 (A-22)

which implies

$$I(X;Y) \ge I(X;Z) \tag{A-23}$$

(A-23) is known as the data processing inequality. In particular, if Z is a function of Y i.e. Z = g(Y), then

$$I(X;g(Y)) \le I(X;Y). \tag{A-24}$$

A.4 Capacity of Independent Parallel Channels

Let $(X_1, X_2, ..., X_n)$ and $(Y_1, Y_2, ..., Y_n)$ be the inputs and outputs, respectively, of n independent parallel channels. The mutual information between the parallel channel input and output is

$$I(X_1, X_2, ..., X_n; Y_1, Y_2, ..., Y_n) = H(X_1, X_2, ..., X_n) - H(X_1, X_2, ..., X_n | Y_1, Y_2, ..., Y_n).$$
(A-25)

Since the n channels are independent, from (A-14)

$$I(X_1, X_2, ..., X_n; Y_1, Y_2, ..., Y_n) = \sum_{i=1}^n H(X_i) - \sum_{i=1}^n H(X_i|Y_i)$$

=
$$\sum_{i=1}^n I(X_i; Y_i).$$
 (A-26)

This implies that the capacity of n independent parallel channels is the sum of the capacities of the individual channels.

B MATLAB Module for Computing CM Capacity using Monte Carlo Integration

The MATLAB code for computing the CM capacity of 16-QAM using Monte Carlo integration is shown below. The same code can be easily generalized for other linear modulation formats, by changing the alphabet size and the variable *symbols*.

```
M=16; % Alphabet size
Ns=10000; % Number of symbols
s(1,1:4)=[(-3:2:3)+sqrt(-1)*3];
s(1,5:8)=[(-3:2:3)+sqrt(-1)*1];
s(1,9:12)=[(-3:2:3)-sqrt(-1)*1];
s(1,13:16)=[(-3:2:3)-sqrt(-1)*3];
symbols=s/sqrt(mean( abs(s).^2 ) );
EsNodB = [-10:1:20]; % Range of SNR
EsNo=10.^(EsNodB/10);
vari=1./(2*EsNo);% Noise variance
[a,datapoints]=size(EsNodB);
maxtrials = 1000*ones(1,length(EsNodB));% Number of Monte Carlo trials
trials = zeros( size( EsNodB ) );
capacity = zeros( size( EsNodB ));
capacity_sum=zeros(size( EsNodB ));
randn( 'state', cputime);
for snrpoint = 1:length(EsNodB);
    while trials(snrpoint) < max_trials(snrpoint)</pre>
          trials(snrpoint) = trials(snrpoint) + 1;
          for i=1:M
             data=symbols(i)*ones(1, Ns);
             noise=sqrt(vari(snrpoint))*(randn(1,Ns)+sqrt(-1)*randn(1,Ns));
             r=data+noise;
             f=-(abs(r-data).^2)/(2*vari(snrpoint));
             max_str=-10000000;
             for k=1:M
                data1= symbols(k)*ones(1, Ns);
                f1=-(abs(r-data1).^2)/(2*vari(snrpoint));
                maxstar=log(exp(f1)+exp(max_str));
```

```
max_str=maxstar;
end
metric=f-maxstar;
capacity_sum(snrpoint)=capacity_sum(snrpoint)+sum(metric);
end
end
capacity(snrpoint)=log2(M)+capacity_sum(snrpoint)/(M*trials(snrpoint)*Ns*log(2));
end
EbNodB = EsNodB - 10*log10(capacity);
plot(EbNodB, capacity);
```

C MATLAB Modules for Generating Detector Transfer Characteristics

The MATLAB code for computing the detector transfer characteristic is given below. 16-QAM is used as an example, but the code can be easily generalized for other linear modulations. The same module can also be used to determine the BICM capacity. Other detector designs can also be incorporated, so long as they produce bit-wise LLRs.

```
M=16; bps=log2(M);
Ns=10000;
Nb=bps*Ns;
EsNodB = [6.8];
EsNo=10.^(EsNodB/10);
nvari=1./(2*EsNo);
s(1,1:4)=[(-3:2:3)+sqrt(-1)*3];
s(1,5:8)=[(-3:2:3)+sqrt(-1)*1];
s(1,9:12)=[(-3:2:3)-sqrt(-1)*1];
```

```
symbols=s/sqrt(mean( abs(s).^2 ) );
if (map ==0)%SP labelling
    mapper = [1 \ 0 \ 0 \ 1; \dots]
               1 1 0 0 ;...
               1 1 0 1;...
               1 0 0 0;...
              1 1 1 0;...
               1 0 1 1;...
              1 0 1 0;...
              1 1 1 1;...
              0 1 0 1;...
              0 0 0 0;...
              0 0 0 1;...
              0 1 0 0;...
              0 0 1 0;...
              0 1 1 1;...
              0 1 1 0;...
              0 0 1 1];
 elseif (map ==1)% Gray labelling
    mapper =[ 1 1 1 0;...
               1 0 1 0;...
              0 0 1 0;...
              0 1 1 0;...
              1 1 1 1;...
              1 0 1 1;...
              0 0 1 1;...
              0 1 1 1;...
              1 1 0 1;...
               1 0 0 1;...
              0 0 0 1;...
```

```
0 1 0 1;...
                1 1 0 0;...
                1 0 0 0;...
                0 0 0 0;...
                0 1 0 0];
   end
  MI_i=[0:0.01:0.99];
  [a,points]=size(MI_i);
  load('Jtable');
  vari(1)=0;
  vari(2:points)=interp1(mutual_info,sigma_square_vector,MI_i(2:points),'spline');
  for k=1:bps
     [a1(:,k)]=find(mapper(:,k)==1);
     [a0(:,k)]=find(mapper(:,k)==0);
  end
for snr=1:length(EsNodB)
   max_trials =[1000*ones(1,points)]; randn('state', cputime);
   test = zeros(2, Nb);
   capacity_sum = zeros( size( vari ) );
   trials = zeros( size( vari ) );
   MI_output = zeros( size( vari ) );
   for point = 1:length(MI_input)
       while trials(point) < max_trials(point)</pre>
             trials(point) = trials(point) + 1;
             cin=round(rand(1, Nb));
             data=bit_to_sym_mapper(cin, symbols, mapper, bps);
             noise= sqrt(nvari(snr))*(randn(1, Ns)+sqrt(-1)*randn(1, Ns));
             r=data+noise;
             for k=1:M
                 data1= symbols(k)*ones(1, Ns);
```

```
f(k,:)=-(abs(r-data1).^2)/(2*nvari(snr));
  end
APP =zeros(1,Nb);
APP =sqrt(vari(point))*randn(1,Nb)+(2.*cin-1).*(vari(point)/2);
bit_extrinsic=zeros(1,Nb);
p(1,:)=1./(1+exp(APP));
p(2,:)=1-p(1,:);
cnt=1;
cnt2=1;
sym_prob=zeros(M,1);
for i=1:Ns
    for j=1:M
      tmp=1;
      for k=1:bps
         tmp =tmp*p(mapper(j,k)+1,cnt2+k-1);
      end
      sym_prob(j)=tmp;
   end
   for k=1:bps
      num=f(a1(:,k),i);
      den=f(a0(:,k),i);
      x1=sym_prob(a1(:,k));
      x0=sym_prob(a0(:,k));
      maxstar1=log(sum(exp(num).*x1));
      maxstar0=log(sum(exp(den).*x0));
      LLR(1,cnt)=maxstar1-maxstar0;
      cnt=cnt+1;
   end
  cnt2=cnt2+bps;
```

 end

```
bit_extrinsic=LLR-APP;
        test(2,:)= (1-2*cin).*bit_extrinsic;
        maxstar=log(exp(test(1,:))+exp(test(2,:)));
        capacity_sum(point) = capacity_sum(point) +sum(maxstar);
  end
  MI_output(point) = 1 - capacity_sum(point)/trials(point)/(Ns)/log(2)/log2(M);
  end
end
plot(MI_i, MI_output);
function data = bit_to_sym_mapper(cin, symbols, mapper, bps) )
       [rows, cols]=size(mapper);
      k=1;
      for i=1:bps:length(cin)
          ind(1,1:bps)=cin(1,i:i+bps-1);
          for j=1:rows
            if ((ind(1,1:bps)==mapper(j,1:bps)))
              sym=j;
              break;
            end
          end
         data(k)=symbols(sym);
         k=k+1;
```

```
end
```

References

- C. E. Shannon, "A mathematical theory of communication," Bell Syst. Tech. J., vol. 27, pp. 379–423,623–656, July, Oct. 1948.
- [2] R. W. Hamming, "Error detecting and error correcting codes," Bell Syst. Tech. J., vol. 29, pp. 147–160, Apr. 1950.
- [3] M. J. E. Golay, "Notes on digital coding," in Proc., IRE, Jun. 1949, vol. 37, p. 657.
- [4] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," SIAM J., vol. 8, pp. 300 – 304, Jun. 1960.
- [5] S. Lin and D. J. Costello Jr., *Error Control Coding*, Prentice-Hall, Inc., Upper Saddle River, NJ, second edition, 2004.
- [6] Z. Peng, Y. F. Huang, and D. J. Costello Jr., "Turbo codes for image transmission-A joint channel and source decoding approach," *IEEE J. Select Areas Commun.*, vol. 18, no. 6, pp. 868–879, June. 2000.
- [7] R. G. Gallager, *Low-Density Parity-Check Codes*, Ph.D. thesis, Massachusetts Institue of Technology, Cambridge, MA, 1960.
- [8] P. Elias, "Error-free coding," IRE Trans. Inform. Theory, vol. 4, pp. 29–37, Sept. 1954.
- [9] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory*, vol. 13, pp. 260–269, Apr. 1967.
- [10] L. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, pp. 284–287, Mar. 1974.
- [11] C. Berrou, A. Glavieux, and P. Thitimasjshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes(1)," in *Proc.*, *IEEE Intl. Conf. on Commun. (ICC)*, Geneva, Switzerland, May 1993, pp. 1064–1070.
- [12] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Letters*, vol. 33, no. 6, pp. 457 – 458, Mar. 1997.

- [13] S. Chung, G. David Forney Jr., T. Richardson, and R. Urbanke, "On the design of Low-density parity-check codes within 0.0045 dB of the Shannon limit," *IEEE Commun. Letters*, vol. 5, pp. 58–60, Feb. 2001.
- [14] J. Proakis, *Digital Communications*, McGraw-Hill, Inc., New York, fourth edition, 2001.
- [15] J. B. Anderson, T. Aulin, and C. E. Sundberg, *Digital Phase Modulation*, Plenum Press, New York, 1986.
- [16] Bluetooth Special Interest Group, "Specification of the Bluetooth system," Core Version 1.2, Nov. 2003.
- [17] T. S. Rappaport, Wireless Communications: Principles and Practice, Prentice-Hall, Inc., Upper Saddle River, NJ, second edition, 2002.
- [18] C. Liang, J. Jong, W. E. Stark, and J. R. East, "Nonlinear amplifier effects in communications systems," *IEEE Trans. Microwave Theory and Techniques*, vol. 47, no. 8, pp. 1461–1466, Aug. 1999.
- [19] D. Torrieri, S. Cheng, and M. C. Valenti, "Robust frequency-hopping system for channels with interference and frequency-selective fading," in *Proc.*, *IEEE Intl. Conf.* on Commun. (ICC), Glasgow, Scotland, Jun. 2007.
- [20] M. K. Simon and C. C. Wang, "Differential detection of Gaussian MSK in a mobile radio environment," *IEEE Trans. Veh. Technol.*, vol. 33, no. 4, pp. 307–320, Nov. 1984.
- [21] M. K. Simon and D. Divsalar, "Maximum-likelihood block detection of noncoherent continuous phase modulation," *IEEE Trans. Commun.*, vol. 41, no. 1, pp. 90–98, Jan. 1993.
- [22] S. V. Pizzi and S. G. Wilson, "Convolutional coding combined with continuous phase modulation," *IEEE Trans. Commun.*, vol. 33, no. 1, pp. 20–29, Jan. 1985.
- [23] B. E. Rimoldi, "Design of coded CPFSK modulation systems for bandwidth and energy efficiency," *IEEE Trans. Commun.*, vol. 37, no. 9, pp. 897–905, Sept. 1989.
- [24] B. Rimoldi and Q. Li, "Coded continuous phase modulation using ring convolutional codes," *IEEE Trans. Commun.*, vol. 43, no. 11, pp. 2714–2720, Nov. 1995.
- [25] G. Ungerboeck, "Channel coding with multilevel/phase signals," IEEE Trans. Inform. Theory, vol. 28, pp. 56–67, Jan. 1982.
- [26] R. W. Kerr and P. J. McLane, "Coherent detection of interleaved trellis encoded CPFSK on shadowed mobile satellite channels," *IEEE Trans. Veh. Technol.*, vol. 41, no. 2, pp. 159 – 169, May. 1992.

- [27] L. Yiin and G. L. Stüber, "Noncoherently detected trellis-coded partial response CPM on mobile radio channels," *IEEE Trans. Commun.*, vol. 44, no. 8, pp. 967–975, Aug. 1996.
- [28] T. L. Tapp and R. L. Mickelson, "Turbo detection of coded continuous phase modulation," in *Proc., Military Commun. Conf. (MILCOM)*, Atlantic City, NJ, Nov. 1999, vol. 1, pp. 534 –537.
- [29] P. Moqvist and T. Aulin, "Serially concatenated continuous phase modulation with iterative decoding," *IEEE Trans. Commun.*, vol. 49, no. 11, pp. 1901–1915, Nov. 2001.
- [30] K. R. Narayanan and G. L. Stüber, "Performance of trellis-coded CPM with iterative demodulation and decoding," *IEEE Trans. Commun.*, vol. 49, no. 4, pp. 676–687, Apr. 2001.
- [31] M. R. Shane and R. D. Wesel, "Parallel concatenated turbo codes for continuous phase modulation," in *Proc.*, *IEEE Wireless Commun. Network. Conf. (WCNC)*, Chicago, IL, Sep. 2000, vol. 1, pp. 147–152.
- [32] E. Cura and J. P. Fonseka, "Soft-decision differential phase detection of turbo-coded M-ary CPFSK signals over Ricean channels," *IEEE Trans. Veh. Technol.*, vol. 54, no. 5, pp. 1722 – 1728, Sep. 2005.
- [33] K. R. Narayanan, I. Altunbaş, and R. S. Narayanaswami, "Design of serially concatenated MSK schemes based on density evolution," *IEEE Trans. Commun.*, vol. 51, no. 8, pp. 1283–1295, Aug. 2003.
- [34] H. Jin, A. Khandekar, and R. McEliece, "Irregular repeat accumulate codes," in Proc., Intl. Symp. Turbo Codes, Brest, France, Sep. 2000, pp. 1–8.
- [35] M. Xiao and T. Aulin, "Irregular repeat continuous phase modulation," IEEE Commun. Letters, vol. 9, no. 8, pp. 723–725, Aug. 2005.
- [36] E. Biglieri, "Digital transmission in the 21st century: conflating modulation and coding," *IEEE Commun. Mag.*, pp. 128–137, May. 2002.
- [37] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 927–946, May. 1998.
- [38] Third Generation Partnership Project 2 (3GPP2), "Physical layer standard for cdma2000 spread spectrum systems, release C," 3GPP2 C.0002-C Version 1.0, pp. 115–122, May 2002.
- [39] E. Zehavi, "8-PSK trellis codes for a Rayleigh channel," *IEEE Trans. Commun.*, vol. 40, no. 5, pp. 873 –884, May 1992.
- [40] T. Svensson and A. Svensson, "On convolutionally encoded partial response CPM," in Proc., IEEE Veh. Tech. Conf.(VTC), Amsterdam, Sept. 1999, pp. 663–667.

- [41] P. Moqvist and T. Aulin, "Power and bandwidth efficient serially concatenated CPM with iterative decoding," in *Proc.*, *Global Telecommun. Conf. (GLOBECOM)*, San Francisco, CA, Nov. 2000, vol. 2, pp. 790 – 794.
- [42] M. Xiao and T. Aulin, "Serially concatenated continuous phase modulation with convolutional codes over rings," *IEEE Trans. Commun.*, vol. 54, no. 8, pp. 1387–1396, Aug. 2006.
- [43] S. J. MacMullan and O. M. Collins, "The capacity of orthogonal and bi-orthogonal codes on the Gaussian channel," *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 1217 1232, May 1998.
- [44] M. C. Valenti and S. Cheng, "Iterative demodulation and decoding of turbo coded M-ary noncoherent orthogonal modulation," *IEEE J. Select Areas Commun.*, vol. 23, no. 9, pp. 1739 – 1747, Sep. 2005.
- [45] R. De Gaudenzi, A. Guillén i Fàbregas, and A. Martinez, "Design of turbo-coded APSK modulations for satellite broadband communications," Wiley Intl. J. of Satellite Commun. and Netw., vol. 24, no. 4, pp. 261–281, Aug. 2006.
- [46] European Telecommunications Standard Institute, Universal mobile telecommunications system, "Multiplexing and channel coding (FDD)," 3GPP2 TS 125.212 Version 6.6.0, Sep. 2005.
- [47] European Telecommunications Standards Institue, "Digital video broadcasting second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications," Draft EN 302 307 DVBS2-74r15, 2003.
- [48] IEEE Std 802.16e 2005, "Standard for local and metropolitan area networks, Part 16: Air interface for fixed broadband wireless access systems, amendment for physical and medium access control layers for combined fixed and mobile operation in licensed bands," Apr. 2005.
- [49] B. Vucetic and J. Yuan, Turbo codes: principles and applications, Kluwer Academic Publishers, Norwell, MA, second edition, 2001.
- [50] Y. Bian, A. Popplewell, and J. J. O'Reilly, "New very high rate punctured convolutional codes," *Electron. Letters*, vol. 30, no. 14, pp. 1119–1120, Jul. 1994.
- [51] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inform. Theory*, vol. 27, pp. 533–547, Sep. 1981.
- [52] R. Gallager, "Low-density parity-check codes," IRE Trans. Inform. Theory, vol. 8, pp. 21–28, Jan. 1962.
- [53] W. E. Ryan, "An introduction to LDPC codes," CRC Handbook for Coding and Signal Processing for Recording Systems, B. Vasic ed. CRC Press, 2004.

- [54] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs and belief propogation," in *Proc.*, *Intl. Symp. on Inf. The. (ISIT)*, Cambridge, MA, Aug. 1998, p. 117.
- [55] T. Richardson, A. Shokrollahi, and R. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 619–637, Feb. 2001.
- [56] T. Richardson and R. Urbanke, "Efficient encoding of Low-Density Parity-Check codes," *IEEE Trans. Inform. Theory*, vol. 47, pp. 683–656, Feb. 2001.
- [57] M. Yang, W. E. Ryan, and Li. Yan, "Design of efficiently encodable moderate-length high-rate irregular LDPC codes," *IEEE Trans. Commun.*, vol. 52, pp. 564–571, April. 2004.
- [58] M. C. Valenti, S. Cheng, and R. I. Seshadri, Turbo Code Applications: A Journey from a Paper to Realization, chapter 12, Springer, New York, NY, 2005.
- [59] D. J. C MacKay, "Good error-correcting codes based on very sparse matrices," IEEE Trans. Inform. Theory, vol. 45, no. 2, pp. 399–431, Mar. 1999.
- [60] F. Jiang, E. Posta, and L. C. Pérez, "The generator and parity-check matrices of turbo codes," in Proc., Conf. on Info. Sci and Sys. (CISS), Princeton, NJ, Mar. 2006.
- [61] A. Nimbalker, Y. Blankenship, and B. Classon, "Turbo-like decoding algorithm for structured LDPC codes," in *Proc.*, *Intl. Symp. on Inf. The. (ISIT)*, Seattle, WA, Jul. 2006, pp. 1708 –1712.
- [62] J. P. Odenwalder, Optimal decoding of convolutional codes, Ph.D. thesis, University of California, Los Angeles, 1970.
- [63] C. Jones, E. Vallés, M. Smith, and J. D. Villasenor, "Approximate-Min* constraint node updating for LDPC code decoding," in *Proc.*, *Military Commun. Conf. (MIL-COM)*, Boston, MA, Nov. 2003, pp. 157–162.
- [64] A. Viterbi, "An intuitive justification and a simplified implementation of the MAP decoder for convolutional codes," *IEEE J. Select Areas Commun.*, vol. 16, pp. 260–264, Feb. 1998.
- [65] M. C. Valenti and J. Sun, "The UMTS turbo code and an efficient decoder implementation suitable for software defined radios," *Intl. J. on Wireless Info. Netw.*, vol. 8, no. 4, pp. 203–216, Oct. 2001.
- [66] S. ten Brink and G. Kramer, "Design of repeat-accumulate codes for iterative detection and decoding," *IEEE Trans. Signal Processing*, vol. 51, no. 11, pp. 2764 –2772, Nov. 2003.
- [67] A. Guillén i Fàbregas and A. Grant, "Capacity approaching codes for non-coherent orthogonal modulation," *IEEE Trans. Wireless Commun.*, 2006.

- [68] B. E. Rimoldi, "A decomposition approach to CPM," IEEE Trans. Inform. Theory, vol. 34, no. 2, pp. 260–270, Mar. 1988.
- [69] S. Pasupathy, "Minimum shift keying: A spectrally efficient modulation," IEEE Commun. Mag., pp. 14–22, Jul. 1979.
- [70] S. A. Gronemeyer and A. L. McBride, "MSK and offset QPSK modulation," *IEEE Trans. Commun.*, vol. 24, no. 8, pp. 809–820, Aug. 1976.
- [71] F. Amoroso and J. A. Kivett, "Simplified MSK signaling technique," *IEEE Trans. Commun.*, vol. 25, pp. 433–441, Apr. 1977.
- [72] A. Yongaçoğlu, D. Makrakis, and K. Feher, "Differential detection of GMSK using decision feedback," *IEEE Trans. Commun.*, vol. 36, no. 6, pp. 641–649, Jun. 1988.
- [73] I. Korn, "GMSK with differential phase detection in the satellite mobile channel," *IEEE Trans. Commun.*, vol. 38, pp. 1980–1986, Nov. 1990.
- [74] G. K. Kaleh, "Differential detection via the Viterbi algorithm for offset modulation and MSK-type signals," *IEEE Trans. Veh. Technol.*, vol. 41, pp. 401–406, Nov. 1992.
- [75] J. P. Fonseka, "Noncoherent detection with Viterbi decoding for GMSK signals," *IEE Proc. Commun.*, vol. 143, pp. 373–379, Dec. 1996.
- [76] A. Mathis, "Differential detection of GMSK signals with low B_tT using the SOVA," *IEEE Trans. Commun.*, vol. 41, pp. 428–430, Apr. 1998.
- [77] J. P. Fonseka, "Soft-decision phase detection with Viterbi decoding for CPM signals," *IEEE Trans. Commun.*, vol. 47, no. 12, pp. 1802–1810, Dec. 1999.
- [78] G. Colavolpe and R. Raheli, "Noncoherent sequence detection of continuous phase modulations," *IEEE Trans. Commun.*, vol. 47, no. 9, pp. 1303–1307, Sept. 1999.
- [79] L. H. J. Lampe, R. Schober, G. Enzner, and J. B. Huber, "Coded continuous phase modulation with low-complexity noncoherent reception," *IEEE Trans. Commun.*, vol. 50, no. 4, pp. 517–520, Apr. 2002.
- [80] R. Yang, S. Chern, and Z. Hsu, "Differential detection of serially concatenated precoded GMSK with iterative decoding," in *Proc.*, *IEEE Wireless Commun. Network. Conf. (WCNC)*, Hong Kong, China, Mar. 2007, pp. 698–703.
- [81] J. P. Fonseka, "Soft-decision differential phase detection with Viterbi decoding in satellite mobile systems," J. Commun. and Netw., pp. 265–272, Sept. 2001.
- [82] W. C. Jakes, *Microwave Mobile Communications*, Wiley, New York, 1974.
- [83] P. Moqvist, Multiuser Serially Concatenated Continuous Phase Modulation, Ph.D. thesis, Chalmers University of Technology, 2002.

- [84] R. I. Seshadri and M. C. Valenti, "Soft-in/ Soft-out noncoherent sequence detection for Bluetooth: Capacity, error rate and throughput analysis," in *Proc.*, *IEEE Wireless Commun. Network. Conf. (WCNC)*, Las Vegas, NV, Apr. 2006.
- [85] A. Soltanian and R. E. Van Dyck, "Performance of the Bluetooth system in fading dispersive channels and interference," in *Proc.*, *Global Telecommun. Conf. (GLOBE-COM)*, San Antonio, TX, Nov. 2001, pp. 3499–3503.
- [86] R. Schiphorst, F. Hoeksema, and C. H. Slump, "A (simplified) Bluetooth maximum a posteriori probability (MAP) receiver," in *Proc.*, *SPAWC*, Rome, Italy, Jun. 2003, pp. 160–164.
- [87] L. Lampe, M. Jain, and R. Schober, "Improved decoding for Bluetooth systems," *IEEE Trans. Commun.*, vol. 53, no. 1, pp. 1–4, Jan. 2005.
- [88] T. Scholand, A. Waadt, and P. Jung, "Max-log-ML symbol estimation postprocessor for intermediate frequency LDI detectors," *Electron. Letters*, vol. 40, no. 5, pp. 183– 184, Feb. 2004.
- [89] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding," *IEEE Commun. Letters*, vol. 1, pp. 169–171, Nov. 1997.
- [90] W. E. Stark, "Capacity and cutoff rate of noncoherent FSK with nonselective Rician fading," *IEEE Trans. Commun.*, vol. 33, pp. 1153–1159, Nov. 1985.
- [91] M. C. Valenti, M. Robert, and J. H. Reed, "On the throughput of bluetooth data transmissions," in *Proc.*, *IEEE Wireless Commun. Network. Conf. (WCNC)*, Orlando, FL, Mar. 2002, pp. 119–123.
- [92] M. C. Valenti and M. Robert, "Custom coding, adaptive rate control, and distributed detection for bluetooth," in *Proc.*, *IEEE Veh. Tech. Conf. (VTC)*, Sep. 2002, pp. 927– 931.
- [93] J. Sykora, "Factor graph framework for serially concatenated coded CPM with limiter phase discriminator receiver," in *Proc.*, *IEEE Veh. Tech. Conf.(VTC)*, Baltimore, MD, Oct. 2007.
- [94] T. Cover and J. Thomas, *Elements of Information Theory*, Wiley, New York, 1991.
- [95] S. ten Brink, "Convergence of iterative decoding," *Electron. Letters*, vol. 35, no. 13, pp. 1117–1118, Jun. 1999.
- [96] S. ten Brink, "Convergence behavior of iteratively decoded parallel concatenated codes," *IEEE Trans. Commun.*, vol. 49, no. 10, pp. 1727–1737, Oct. 2001.
- [97] S. Pfletschinger and F. Sanzi, "Error floor removal for bit-interleaved coded modulation with iterative decoding," *IEEE Trans. Wireless Commun.*, vol. 11, no. 5, pp. 3174 – 3181, Nov. 2006.

- [98] M. Tuchler, R. Koetter, and A. C. Singer, "Turbo equalization: principles and new results," *IEEE Trans. Commun.*, vol. 50, no. 5, pp. 754 –767, May 2002.
- [99] K. Li and X. Wang, "EXIT chart analysis of turbo multiuser detection," IEEE Trans. Wireless Commun., vol. 4, no. 1, pp. 300 –311, Jan. 2005.
- [100] P. Moqvist and T. Aulin, "Convergence analysis of SCCPM with iterative decoding," in *Proc., Global Telecommun. Conf. (GLOBECOM)*, San Antonio, TX, Nov. 2001, vol. 2, pp. 1048 – 1052.
- [101] A. Guillén i Fàbregas and G. Caire, "Coded modulation in the block-fading channel: Coding theorems and code construction," *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 91 – 114, Jan. 2006.
- [102] T. Ghanim and M. C. Valenti, "The throughput of hybrid-ARQ in block fading under modulation constraints," in *Proc.*, Conf. on Info. Sci and Sys. (CISS), Princeton, NJ, Mar. 2006.
- [103] T. Cover and J. Thomas, *Elements of Information Theory*, Wiley Interscience, New York, 1991.
- [104] S. ten Brink, J. Speidel, and R. H. Yan, "Iterative demapping and decoding for multilevel modulation," in *Proc.*, *Global Telecommun. Conf. (GLOBECOM)*, Sydney, Australia, Nov. 1998, pp. 579–584.
- [105] A. Knopp and P. A. Humblet, "On coding for block fading channels," IEEE Trans. Inform. Theory, vol. 46, no. 1, pp. 189 – 205, Jan. 2000.
- [106] S. Dolinar, D. Divsalar, and F. Pollara, "Code performance as a function of block size," JPL TDA, 1998.
- [107] K. M. Chugg, "Summary of performance limits," Lecture Notes, 2004.
- [108] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.

Contributions

Contributions to the field of coded communication system design with CPM:

- R. Iyer Seshadri and M. C. Valenti, "Capacity-based parameter optimization of bitinterleaved coded CPM with differential detection," to appear in *IEEE Transactions* on Vehicular Technology (accepted on Feb. 19, 2007).
- R. Iyer Seshadri, S. Cheng and M. C. Valenti, "The BICM capacity of coherent continuous-phase frequency shift keying," in *Proc. IEEE Vehicular Technology Conference (VTC)*, (Baltimore, MD), Oct. 2007.
- S. Cheng, R. Iyer Seshadri and M. C. Valenti, "The capacity of noncoherent coherent continuous-phase frequency shift keying," in *Proc. Conference on Info. Sci. and Sys.* (CISS), (Baltimore, MD), Mar. 2007.
- R. Iyer Seshadri and M. C. Valenti, "A capacity-based approach for designing bitinterleaved coded GFSK with noncoherent detection," in *Proc. International Sympo*sium on Information Theory (ISIT), (Seattle, WA), July 2006.
- R. Iyer Seshadri and M. C. Valenti, "Soft-in/ Soft-out noncoherent sequence detection for Bluetooth: capacity, error rate and throughput analysis," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, (Las Vegas, NV), Apr. 2006.
- R. Iyer Seshadri, D. Lao, C. Kwan and J. P. Fonseka, "Bandwidth constrained, low complexity, noncoherent CPM with ML soft-decision differential phase detection", in *Proc. IEEE Military Communications Conference (MILCOM)*, (Atlantic City, NJ), Nov. 2005. (Invited Paper).

Contributions to the field of channel coding, especially low-density parity-check codes:

- M. C. Valenti, S. Cheng and R. Iyer Seshadri, "Turbo and LDPC codes for digital video broadcasting," Chapter 12 of Turbo Code Applications: A Journey from a Paper to a Paper to Realization, Springer, 2005.
- M. C. Valenti and R. Iyer Seshadri, "Turbo and LDPC codes: Implementation, simulation, and standardization," Virginia Tech. Symposium on Wireless Personal Communications, (Blacksburg, VA), Jun. 2006.

Contributions to the Iterative Solutions Coded Modulation Library:

- 1. Encoder and support for the DVB-S2 LDPC code, Dec. 2005.
- 2. Encoder and support for the WiMAX LDPC code, Jun. 2007.
- 3. Correlated fading simulator, Aug. 2007.

Miscellaneous contributions:

- B. Zhao, R. Iyer Seshadri, and M. C. Valenti, "Geographic random forwarding with hybrid-ARQ for ad hoc networks with rapid sleep cycles," in *Proc. IEEE Global Telecommunications Conference (GLOBECOM)*, (Dallas, TX), Dec. 2004.
- R. Iyer Seshadri and D. J. Miller, "An iterative joint-source channel decoder using turbo codes," MS. Thesis, Department of Electrical Engineering, Pennsylvania State University, May 2003.