University of Mississippi

eGrove

Electronic Theses and Dissertations

Graduate School

2010

Distributed Turbo Product Coding Techniques Over Cooperative Communication Systems

Esam Ali Obiedat

Recommended Citation

Obiedat, Esam Ali, "Distributed Turbo Product Coding Techniques Over Cooperative Communication Systems" (2010). *Electronic Theses and Dissertations*. 212.
https://egrove.olemiss.edu/etd/212

To the Graduate Council:

I am submitting herewith a dissertation written by **Esam Ali Obiedat** entitled "**Distributed Turbo Product Coding Techniques Over Cooperative Communication Systems**." I have examined the final copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Engineering Science.

Lei Cao, Chair
Associate Professor of Electrical Engineering

We have read this dissertation
and recommend its acceptance:

John N. Daigle, Member
Professor of Electrical Engineering

Paul M. Goggans, Member
Associate Professor of Electrical Engineering

Bahram Alidaee, Member
Professor of Operations Management

Accepted for the Council:

Dean of the Graduate School

STATEMENT OF PERMISSION TO USE

In presenting this dissertation in partial fulfillment of the requirements for a Doctor of Philosophy degree at The University of Mississippi, I agree that the Library shall make it available to borrowers under rules of the Library. Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of the source is made.

Permission for extensive quotation from or reproduction of this dissertation may be granted by my major professor or, in his absence, by the Head of Interlibrary Services when, in the opinion of either, the proposed use of the material is for scholarly purposes. Any copying or use of the material in this dissertation for financial gain shall not be allowed without my written permission.

Signature ________________________________

Date ________________________________

Distributed Turbo Product Coding Techniques Over Cooperative Communication Systems

By

Esam Ali Obiedat

M.Sc., Jordan University of Science and Technology, Jordan, 2005

B.Sc., Jordan University of Science and Technology, Jordan, 2003

A Dissertation
Submitted to the Faculty
of the University of Mississippi
in Partial Fulfilment of the Requirements
for the Degree of Doctor of Philosophy
with a Major in Engineering Science
in the School of Engineering

The University of Mississippi
August 4, 2010

To my parents, Ali and Amneh...

thank you for all your continuous support.

To my wife, Raghad...

it would not be possible without you.

Acknowledgements

وَمَا تَوْفِيقِي إِلَّا بِاللَّهِ

First and foremost, I thank Allah for all countless blessings that he bestowed on me, for guiding me all the way until this point and for facilitating the work on this dissertation on me.

I would like to express my sincere gratitude to my advisor, Dr. Lei Cao, for his support, guidance and continuous encouragement throughout my doctoral studies at the University of Mississippi. He made my research and studies an enjoyable and rewarding experience. I would also like to thank the other members of my dissertation committee for many useful interactions and for contributing their broad perspectives and expertise to this dissertation.

I gratefully acknowledge the support of the department of Electrical Engineering, chaired by Prof. Allen Glisson, for the continuous assistantships and funding for my research.

I am also very grateful to all of my friends, my current and former colleagues, for inspiring technical and nontechnical discussions that made studying and living in the University of Mississippi much more pleasant.

Lastly, I am grateful to my family in Jordan for all their support and encouragement. My deep appreciation and thanks for my life companion, Raghad, for her extreme patience, emotional support, and sacrifices. May Allah reward her for helping me achieving this work.

University, Mississippi — Esam Obiedat
August 2010

Abstract

In this dissertation, we propose a coded cooperative communications framework based on Distributed Turbo Product Code (DTPC). The system uses linear block Extended Bose-Chaudhuri-Hochquenghem (EBCH) codes as component codes. The source broadcasts the EBCH coded frames to the destination and nearby relays. Each relay constructs a product code by arranging the corrected bit sequences in rows and re-encoding them vertically using EBCH as component codes to obtain an Incremental Redundancy (IR) for source's data.

Under this frame, we have investigated a number of interesting and important issues. First, to obtain, independent vertical parities from each relay in the same code space, we propose circular interleaving of the decoded EBCH rows before re-encoding vertically.

We propose and derive a novel soft information relay for the DTPC over cooperative network based on EBCH component codes. The relay generates Log-Likelihood Ratio (LLR) values for the decoded rows are used to construct a product code by re-encoding the matrix along the columns using a novel soft block encoding technique to obtain soft parity bits with different reliabilities that can be used as soft IR for source's data which is forwarded to the destination.

To minimize the overall decoding errors, we propose a power allocation method for the distributed encoded system when the channel attenuations for the direct and relay channels are known. We compare the performance of our proposed power allocation method with the fixed power assignments for DTPC system. We also develop a power optimization algorithm to check the validity of our proposed power allocation algorithm. Results for the power allocation and the power optimization prove on the potency of our proposed power allocation criterion and show the maximum possible attainable performance from the DTPC cooperative system.

Finally, we propose new joint distributed Space-Time Block Code (STBC)-DTPC by generating the vertical parity on the relay and transmitting it to the destination using STBC on the source and relay. We tested our proposed system in a fast fading environment on the three channels connecting the three nodes in the cooperative network.

Contents

List of Tables x

List of Figures xiv

Abbreviations xv

Preface xviii

1 Introduction 1

1.1 Background and Literature Survey 2

1.2 Motivation and Contributions 4

1.3 Summary of Contributions 8

2 Background Information 10

2.1 Cooperative Communication 10

2.1.1 Cooperative Diversity 12

2.1.2 Cooperative Communication Types 13

Amplify And Forward Relaying 15

Decode And Forward Relaying 16

Compression And Forward Relaying 17

Coded cooperation 18

2.2 Turbo Product Codes 20

2.2.1 Parallel Concatenation of Product Codes 20

2.2.2 Performance of Product Codes with BCH Component Codes . 27

2.2.3 Soft Decoding of Block Codes 28

Optimal Block Decoding Decision 30

Soft Decoding of Block Codes (Chase Algorithms) 32

2.2.4 Iterative Decoding of Product Turbo Codes 36

2.2.5 SISO Decoding of a Block Code using a Competing Codeword (CC) 37

2.2.6 SISO Decoding Based on the Destructive Euclidean Distance . 45

3 Distributed Turbo Coding with Hard Detection 52

3.1 Introduction 52

3.2 System Model 54

3.2.1 Turbo Product Codes (TPC) 55

3.2.2 Simulation Network Model 56

3.3 Distributed Product Turbo Codes DTPC 58

3.3.1 Generating Multiple Vertical Parities (MVP) 59

3.3.2 Decoding at the Destination 60

3.4 Simulation Results 63

3.4.1 Single Relay 63

3.4.2 Multiple Relays 66

3.5 Conclusions 69

4 Distributed Turbo Product Codes with Soft Incremental Redundancy 71

4.1 Introduction 71

4.2 System Model 72

4.3 Soft Decode and Forward 76

4.4 Decoding at Destination 81

4.5 Simulation Results 83

4.6 Conclusions 85

5 Power Allocation to Alleviate the Error Propagation on the DTPC 87

5.1 Introduction 87

5.2 System Model and Problem Formulation 90

5.3 Power Allocation (PA) 95

5.4 Simulations and Results 97

5.5 Conclusions 100

6 Power Optimization for the DTPC 102

6.1 Introduction 102

6.2 System Model 104

6.3 Power Optimization 107

6.3.1 Optimization Algorithm Requirements 108

6.3.2 Optimization Algorithm 110

6.4 Results and Discussion 113

6.4.1 Optimization Results 113

6.4.2 Performance of DTPC System with Power Optimization . . . 115

6.4.3 Comparison with Power Allocation Criterion 116

6.5 Conclusions 119

7 Joint Distributed Space-Time Block Coding with Distributed Turbo Product Code (DSTBC-DTPC) 121

7.1 Introduction 121

7.2 System Model 124

7.3 Results and Discussion 129

7.4 Conclusions 133

8 Concluding Remarks and Future Work 137

8.1 Conclusions 137

8.2 Future Work 139

Bibliography 141

VITA 150

List of Tables

2.1 Look-up table for ϕ as a function of d_{DED}. 47

3.1 Four different setup for the two relays for simulating two nodes acting as relays in Multiple Vertical Parities (MVP)-DTPC 67

6.1 Signal to Noise Ratio (SNR) values used on the optimization algorithm for each λ step 114

6.2 The 7th degree polynomial coefficients found using polynomial fitting 115

List of Figures

2.1 The three-terminal network with a source ‘S’, a relay ‘R’ and a destination ‘D’ node. 11

2.2 Example of three relays comprising a MIMO cooperative transmitter. 14

2.3 Simple cooperation model, with source and relay transmission energy/symbol set to E_1 and E_2 respectively. 14

2.4 Example of coded cooperation in which the relay transmits the IR when CRC finds no errors in the decoded sequence by the CRC. . . . 19

2.5 Concatenation of two codes in (a) Parallel and (b) Serial form 21

2.6 The output matrix for serial concatenation of two block codes (TPC) 22

2.7 The incomplete turbo product code. 25

2.8 Example of a TPC matrix having all zero rows and columns except for $\delta_2 = 5$ columns and $\delta_1 = 3$ rows, in groups A and B, respectively. . 26

2.9 Finding the most probable codewords within the sphere of radius $\sqrt{4(\delta - 1)}$. 33

2.10 Single stage SISO decoder based on competing codeword CC algorithm 44

2.11 Simulation results for the confidence value ϕ versus the the destructive Euclidean distance DED d_{DED} 48

2.12 Single stage SISO decoder based on DED algorithm 50

2.13 BER performance comparison between CC and distance based decoding DBD algorithms plotted versus E_b/N_0 for TPC $(64, 51, 6)^2$ with binary phase shift keying BPSK signaling in a Gaussian channel after 4 iterations 51

3.1 Structure of a Turbo Product Code 56

3.2 The simplified three terminals line network topology 56

3.3 The relays aid the source by forming a transmit cluster 58

3.4 Cyclically interleaving of $\mathcal{C}^1$ codewords matrix 59

3.5 Modified one stage Soft Input Soft Output (SISO) Turbo Product Codes (TPC) decoder 61

3.6 Multiple vertical parities DTPC decoder 62

3.7 Bit Error Rate (BER) performance of DTPC in Additive White Gaussian Noise (AWGN) channel with free space propagation after 4 iterations 64
3.8 BER performance of DTPC in AWGN channel with free space propagation after 4 iterations versus the position of the relay 65
3.9 BER performance of DTPC in AWGN channel with relatively lossy environment after 4 iterations 66
3.10 BER performance of DTPC with multiple relays in AWGN after 4 iterations in line network scenario 68
3.11 BER performance vs. relay 1 position λ_1 of DTPC with multiple relays in 2 dB AWGN SNR_{sd} channel after 4 iterations in line network scenario 69
3.12 BER performance vs. $SNR_{rd} - SNR_{sd}$ of DTPC with multiple relays in AWGN channel after 4 iterations in transmit cluster network scenario for different Δ SNR $=SNR_{sr} - SNR_{sd}$ 70

4.1 System Model for a general cooperative system 73
4.2 (a) The simplified three terminals in line model. (b) SNR of the inter-user and relay channels versus the position λ of the relay in line model. 75
4.3 Modified one stage SISO TPC decoder 82
4.4 BER performance of SDF-TPC in AWGN channel after 4 iterations for different λ compared to the regular Decode and Forward (DF)-TPC and non-cooperative TPC 84
4.5 BER performance of Soft Decode and Forward (SDF)-DTPC in AWGN channel after 4 iterations versus the position of the relay λ compared to the regular DF-TPC and non-cooperative TPC 85

5.1 The Structure of TPC matrix 91
5.2 The simplified three terminals line network topology 92
5.3 SNR of the inter-user and relay channels versus the position λ of the relay in line model. 93
5.4 Normalized energy/bit for the source and the relay using the proposed power allocation 97
5.5 BER performance versus γ_{sd} of the DTPC system with DF relaying and power allocation employed compared to the regular system with constant power assignments 98

5.6 BER performance versus relay position λ of the DTPC system with DF relaying and power allocation compared to the system with constant power assignments 98
5.7 BER performance versus γ_{sd} of the DTPC system with SDF relaying and power allocation compared to the system with constant power assignments 99
5.8 BER performance versus relay position λ of the DTPC system with SDF relaying and power allocation compared to the system with constant power assignments 100

6.1 Cooperative system model with one source, one relay and one destinations 104
6.2 The principle of sliding ball used in designing the optimization algorithm 110
6.3 Illustration to show how the optimization algorithm finds new search segment 113
6.4 Power Optimization Results for α in (6.2.4). 114
6.5 The BER performance of the DTPC with SDF protocol using Optimal Power Allocation versus the SNR. 116
6.6 Using curve fitting to plot α for the Power Optimization results and compare it to the earlier proposed Power Allocation and Constant Power Allocation for the Soft Information Relaying (SIR)-DTPC system. 117
6.7 Comparison between the BER performance of the DTPC with SDF protocol using Optimal Power Allocation, Proposed Power Allocation and Constant Power Allocation versus the location of the relay. 119

7.1 Proposed Joint Distributed STBC Distributed TPC configuration . . 125
7.2 BER performance of the joint Distributed STBC DTPC encoding using hard decoded and forward DF protocol for relay positions $3 \leq \lambda \leq 6$ 130
7.3 BER performance of the joint Distributed STBC DTPC encoding using hard decoded and forward DF protocol for relay positions $7 \leq \lambda \leq 9$ 131
7.4 BER performance of the joint Distributed STBC DTPC encoding using soft decoded and forward SDF protocol for relay positions $3 \leq \lambda \leq 6$ 132

7.5 BER performance of the joint Distributed STBC DTPC encoding using hard decoded and forward SDF protocol for relay positions $7 \leq \lambda \leq 9$ 133

7.6 BER performance of the two distributed systems, the system with DTPC only and the system with joint Distributed Space-Time Block Codes (DSTBC) and DTPC when the relay is using DF protocol plotted against the position of the relay λ 134

7.7 BER performance of the two distributed systems, the system with DTPC only and the system with joint DSTBC and DTPC when the relay is using SDF protocol plotted against the position of the relay λ 135

List of Abbreviations

AF Amplify and Forward

ARQ Automatic repeat request

AWGN Additive White Gaussian Noise

BCH Bose-Chaudhuri-Hochquenghem

BER Bit Error Rate

BPSK Binary Phase Shift Keying

BTC Block Turbo Codes

CC Competing Codeword

CF Compress and Forward

CRC Cyclic Redundancy Check

CSI Channel State Information

CTC Convolutional Turbo Codes

DBD Distance Based Decoding

DED Destructive Euclidean Distance

DF Decode and Forward

DTPC Distributed Turbo Product Code

DSTBC Distributed Space-Time Block Codes

DSTTC Distributed Space-Time Trellis Code

EBCH Extended Bose-Chaudhuri-Hochquenghem

FEC Forward Error Correction

GF Galois Field

IR Incremental Redundancy

iid independent and identically distributed

LDPC Low Density Parity Check

LLR Log-Likelihood Ratio

LRB Least Reliable Bits

LTE Long Term Evolution

MAP Maximum A-posteriori Probability

MIMO Multi-Input Multi-Output

ML Maximum-Likelihood

MRC Maximal Ratio Combiner

MVP Multiple Vertical Parities

OSD Ordered Statistics Decoding

PA Power Allocation

PDA Personal Digital Assistant

PSK Phase Shift Keying

QAM Quadrature Amplitude Modulation

RCPC Rate Compatible Punctured Convolutional

RS Reed-Solomon

RSC Recursive Systematic Convolutional

SDF Soft Decode and Forward

SER Symbol Error Rate

SISO Soft Input Soft Output

SIR Soft Information Relaying

SM Spatial Multiplexing

SNR Signal to Noise Ratio

ST Space-Time

STB Space-Time Block

STC Space-Time Code

STBC Space-Time Block Code

TPC Turbo Product Codes

UMB Ultra Mobile Broadband

WEF Weight Enumerator Function

Preface

Growing demand on the use of the Internet services and wireless communications in recent years has introduced new problems and challenges for wireless industry and wireless providers. This big demand on wireless services is not limited to cell phones, text-messaging devices, wireless-enabled laptops, Personal Digital Assistant (PDA), but is expanding to new services and devices, increasing the need for more reliable and faster services. High data rate is desirable for many other wireless applications, and some of these applications would have been impossible without transmission having certain quality of service, in terms of, for example, transmission rate, delay, and error rate.

In order to meet the desired quality of service requirements for the next generation of wireless devices, good channel coding and higher signal-to-noise ratio should be available to support the operation of the system. Channel coding not only provide the desired quality of service of the system, but also improves the data transmission rates over the available bandwidth and provides increased battery life, which translates to power optimal systems with high spectral efficiency.

The new wireless technologies pose significant technical challenges unlike the wired telecommunication technologies. These challenges are due to the scare resources of the power and frequency, the channel variation in fading and noisy environment and the high cost of equipment which is required for sophisticated processing of of wireless signals.

Consequently, research for new techniques is in demand to meet these requirements and to solve the underlying challenges and further expand the growth of current wireless technologies. The conventional wireless point-to-point communication technologies showed significant limitations on the maximum possible transmission rates and on handling channel impairments. Some of these limitations are related to the limited resources of the power, or available frequency, while others are related to the size of the wireless devices or the cost of hardware. For instance, although the Multi-Input Multi-Output (MIMO) technology is very promising for systems requiring high transmission data rates, it is impractical to be implemented in every wireless device because it requires an array of antennas with adequate separation to be placed at the transmitter and the receiver. Wireless sensor network give another example of such limitations, since most of the applications require vast distribution of the wireless elements, and without relay this requires high transmission power to convey the signal to the collecting node, which would drain the battery faster.

Motivated by the above limitations on the current communication systems, the idea of cooperative communication systems has evolved to provide two main schemes: (i) Use of relays (or multi-hop) to provide spatial diversity in a fading environment or to provide distributed processing of source's message. (ii) A collaborative scheme where the relay also has its own information to send so both terminals help one another to communicate by acting as relays for each other.

In this dissertation, we explore the possible incorporation of the powerful turbo product codes in the cooperative systems to provide enhanced bit error rate and less complexity.

Chapter 1

Introduction

Cooperative communication and coding have been found to be an effective way to increase the data rates or decrease the transmission power. Wireless networks are inherently broadcast, and any message sent out by a source node is heard by all surrounding nodes within communication range listening in the same frequency band. If an idle node in the system dedicates it's idle channel to retransmit the originally transmitted message it could help the initial transmission by providing diversity at destination node and thus improves the system performance while maintaining fairness among nodes and better channel utilization. This simplified example of cooperation shows the great opportunities for rich forms of cooperation among the wireless nodes.

Cooperative wireless communication is a promising technology for future communication systems, and is expected to provide solutions to many limitations on the next generation technologies such as wireless sensor networks and ad-hoc networks. In the last decade there has been a large ongoing research effort in this field to come up with more efficient networks in terms of higher power and frequency efficiency. Most of the research focused on the processing done at the intermediate node (the relay), and the outcome were plenty of relaying protocols that range from simply amplifying the received signal to very intelligent adaptive relaying techniques that involve higher coordination among participating relays.

1.1 Background and Literature Survey

The first appearance for the concept of cooperative communications can be found on an early work of Cover and Gamal on achievable capacity of a relay network in 1979. However, since the introduction of cooperative communications 30 years ago, only recently was a great potential rediscovered in applications of cellular and wireless sensor networks [1, 2], and many others. The distributed structure of wireless networks provides a unique opportunity for cooperation and distributed signal processing. Moreover, the wireless nodes have broadcasting nature, and every transmission can reach multiple idle nodes without any loss in spectral efficiency or power. Thus, a better system utilization can be accomplished if one or more of these idle nodes interact and participate to improve the system performance and guarantee fairness among nodes.

Design of efficient cooperative protocols and distributed signal processing techniques has been an important issue in the recent research to implement cooperative communications in wireless networks. Therefore, most of the recent research on wireless cooperative communication has focused on designing relaying protocols, signaling, and distributed coding and decoding. Specifically, the design of efficient relaying protocols and distributed coding schemes has attracted great attention, where a number of novel relay protocols [1–3] and distributed coding schemes [4–14] have been developed in the past several years. Some of the proposed distributed coding schemes have achieved capacity-approaching performance.

The literature reports many implementations of channel codes on cooperative coding. Most of the current cooperative coding techniques are based on dividing the whole channel code into two parts: the first part is broadcasted from the source at high code rate while the second part is generated at the relay from the first part. Examples of cooperative coding schemes in the literature are abundant, for instant, Hunter *et. el.* [15] presented simulation results for a two-user cooperative system scenario using a *Rate Compatible Punctured Convolutional* (RCPC) in which the source transmits a punctured codeword of convolutional code with a high code rate, while the relay generates and transmits the second part of the punctured codeword after correctly decoding the first part transmitted by source.

Distributed turbo coding examples can also be found in the literature, in which the destination performs turbo decoding on the multiple parts of the code received over multiple links. This can be carried out by encoding the source message using a component code having a higher code rate than the total code rate of the overall system. Then the source broadcasts the encoded message to the destination and the listening relays. The relays try to correct the received version of source's transmission over the noisy channel and re-encode the original message using the second component code after interleaving. It is often assumed that the relay can successfully decode the original source's message, re-encode and transmit the incremental redundancy to the destination. At the destination, a full turbo code matrix can be constructed to recover the source message via turbo decoding to recover the source's message. An example of such implementation can be found in [10], which applies turbo decoding at the destination to decode the convolutional turbo code received via the two channels. The source broadcasts punctured convolutional code sequence that is received by the relay and the destination. If the Cyclic Redundancy Check (CRC) check on the decoded signal at the relay indicates error free decoding, then the relay interleaves the original message and produces parity bits. The latter is punctured and transmitted to the destination as the parity of the second component code.

Similarly, sum and product decoding is used to decoded the two received parts of the Low Density Parity Check (LDPC) code through the direct link and the relay link in [16]. Indeed, there are many distributed encoding strategies and different decoding techniques depending on the constructed code, e.g. [7, 9, 10, 16–19], have been reported in the literature. Specifically, LDPC codes are employed in [9, 16–18] while the distributed turbo codes principle is used in [7, 10, 19].

1.2 Motivation and Contributions

Given all the research that has been conducted in this area recently, many issues in the design and implementation of cooperative communications still have not been fully addressed, and there are still many issues in both theory and practical implementation that have not been treated. For instance, most of the existing distributed coding schemes are based on conventional channel coding schemes, such as Space-Time Code (STC), turbo coding, and LDPC coding, nevertheless there are many other existing channel coding schemes very suitable for some applications have not been studied in cooperative coded systems.

Turbo Product Codes (TPC) have shown high decoding performance with a very low decoding complexity can be encoded or decoded via an algebraic methods. In addition to that, TPC have high code rates, making them very appropriate for some wireless applications that require very simple low-powered electronic hardware and require high spectral efficiency such as sensor networks [20]. Minimum hardware can be added to relay nodes that are participating in the distributed encoding process. However, the powerful Bit Error Rate (BER) performance and its high code rate of the TPC have not seen studied yet in coded cooperation. The literature has not shown any work that investigates the integration of TPC in cooperative coding. We aim in this dissertation to investigate and study the implementation of TPC in coded cooperation. Specifically, we propose a coded cooperation communications framework based on distributed TPC, on single and multiple relays. We study the conjunction of Distributed Turbo Product Code (DTPC) in cooperative network in many aspects, such as the effect of varying channel conditions on the attainable coding gain under assumptions of decoding errors at the relay. Based on these results, we propose solutions to mitigate the error propagation at the destination resulted from the erroneous data received from the relay.

Most of the previously presented coded cooperation strategies are formulated for the classical three-node relay channel model, i.e., the transmitter-receiver network with only one relay. However, diversity gain theoretically increases with the number of relay nodes [21]. In this dissertation, we apply the concept of distributed encoding for the source's message over multiple relay nodes and use a modified iterative turbo product decoding at the destination to decode the received distributed TPC over multiple channels. We investigate the performance of distributed TPC in Additive White Gaussian Noise (AWGN) channel using simple network topologies.

Distributed coding schemes proposed on the literature rely on some assumptions, such as error free decoding at relays. However, only a few works have investigated the effect of detection errors and a modeling of detection errors in the Detect and Forward protocol has been done by Wang *et. al.* in [3]. A Distributed Space-Time Trellis Code (DSTTC) system has been constructed in [22] using the detection error modeling presented in [3]. These two works have shown that detection errors do have some effects in constructing practical distributed coding. In this dissertation, we investigated the effect of relay decoding and re-encoding errors on the performance of the turbo decoder. We studied the BER performance of the overall system and suggested solutions to alleviate the effect relay erroneous decisions on the turbo decoding.

The Decode and Forward (DF) protocol benefits from error correction capability of the code and thus can correct some or all of the errors caused by the source-relay (inter-user) channel. However, the reported literature on coded cooperation assumes error-free for the inter-user channel or it is assumed that the inter-user channel is reliable enough to have correctable number of errors. Otherwise, the overall cooperative system will lead to serious error propagation as relay will forward erroneous data and will mislead the destination receiver. Another reason for the error propagation at the destination is that the relay channel (relay-destination channel) usually have higher Signal to Noise Ratio (SNR) than the direct channel, and thus the destination decoder considers the code part received over the relay channel to be more reliable and more trustworthy than the code part received over the direct channel. This would cause no problem if the relay's decoder makes no errors. However if the relay forwards erroneous data as result of decoding errors, the decoding process would fail severely as it bases it decisions on erroneous data that lead to error propagation.

In contrast to DF protocol, the Amplify and Forward (AF) keeps itself from any premature decision or decoding errors, because it forwards the soft information content of the received signal to the destination as is. However this scheme does not take advantage from error correction capabilities of the code at the relay. In addition it amplifies and forwards the noise and distortion, was well as the message signal. From the implementation complexity side, the DF is much more complex than AF. To gain both advantages of the DF and AF protocols, we propose in this dissertation, a modification to the distributed TPC to alleviate the error propagation due to the relay's erroneous decoding, by forwarding soft information generated from soft-decoding and then soft-encoding process at the relay. The signs of forwarded signal represent the hard decisions, while the magnitudes represent the decision's reliability.

Most existing distributed coding schemes are constructed based on fixed code rates and power allocations. Furthermore, the adaptive power allocation problem has not been addressed. Cooperative network is constituted of several independent channels, which implies that different channels have different channel conditions, for example, one channel could be suffering from shadowing while another experiencing higher SNR. Under these conditions, it is better to allocate the power among the nodes participating on the cooperative process, by loading more power on the links having higher SNR or the nodes transmitting more significant information. We investigate in this dissertation the further improvement that can be achieved in the DTPC cooperative coding system proposed previously, when the channel attenuations for the direct and relay channels are also known for the source and the relay by allocating the transmission power among the two nodes, the source and the relay. Rather than assigning equal power to the source and the relay, we use the relative locations of the source and relay to the destination (and thus the channel attenuations in free space propagation environment) to find the power allocation that results in the desired SNR requirements at the destination's receiver. We investigate the effect that the positioning of the relay has on a relaying system and derive power allocation expressions depending on the comparative distances for the two transmitting nodes in the three-terminal model with respect to the destination, which is described in detail in the system model. To investigate the effectiveness of our proposed power allocation, we investigate the extent of attainable improvements that can be achieved in the DTPC cooperative coding system by searching for the optimal power allocations based on the position of the relay. We propose a black-box optimization algorithm that is based on the principle of a sliding ball and use it to find the power allocation that results in the BER performance at the destination's receiver.

To improve the performance of the DTPC system even more by enhancing the conditions of the relay channel, we employ transmit diversity to transmit the second phase of data, which are the parity information about the first transmission phase as generated by the relay. We propose to use the distributed Space-Time Block (STB) coding to transmit the second part of the distributed TPC on the relay and direct channels. The source and the relay share their single antennas to create a virtual transmit array to transmit the generated parity on the second phase toward the destination.

1.3 Summary of Contributions

To summarize, the contributions of this dissertation are that we propose:

1. A framework for distributed turbo block codes, for which we present simulation results under the assumption that the relay makes decoding errors and forward erroneous incremental parity to the destination. Under this framework, we propose solutions to enhance the BER performance under different channel conditions.

2. A method to generate multiple vertical parities for the turbo block codes using a cyclic interleaver. We use cyclic interleavers on each relay and forward the result parities to the destination. The destination performs a joint turbo decoding for all the received vertical parities.

3. A soft information relaying technique in which the relay decodes the source's message and re-encodes it across columns using a novel soft block encoding technique to obtain soft parity bits with different reliabilities that can be used as soft incremental redundancy that is forwarded to the destination.

4. To overcome the error propagation at the destination we proposed also a power allocation method and verified the effectiveness of this simple method by comparing it's results to system with optimized power at the source and the relay.

For the same purpose, we proposed a power optimization algorithm for the distributed coded system that is based on a sliding ball principle.

5. We also proposed a joint distributed STBC-TPC system that aims to enhance the BER performance by transmitting the second part of the turbo product code over virtual transmit antenna using the source and the relay.

Chapter 2

Background Information

2.1 Cooperative Communication

The main concept behind wireless cooperative communications is to allow the independent non-cooperative users of a network to share their scarce resources in order to improve the overall performance or guarantee fairness. Cooperation can be of primitive form that only require very simple cooperative protocols, where the cooperative communication is considered as inherent part of the wireless communication protocol, if the protocol applies rules for medium sharing among users (e.g. TPC, ALOHA). In this type of cooperation, the participants focus on the fair sharing of the given resource without gaining anything else. On the other hand, cooperative communication may require advanced cooperative protocols to be pre-established, i.e. it should be allowed and supported by the design of the communication system. Examples of such advanced cooperation include relaying techniques, coded cooperation, etc.

Cooperation terminology is used mainly for the relaying process, which aims to extend the coverage range of the communication systems. The three independent nodes/terminals, as depicted in Fig. 2.1 represent the simplest cooperative topology. One of these three terminals acts as the source 'S' of the information, another terminal acts as a relaying terminal 'R', which conveys the signal to the last terminal which is the destination 'D'. Henceforth, we will use the terms inter-user, relay and direct

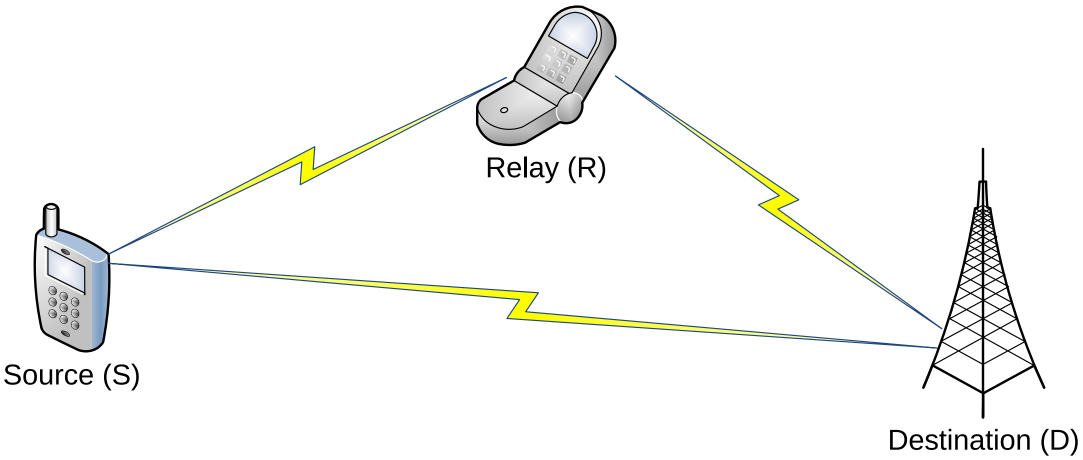


Figure 2.1. The three-terminal network with a source 'S', a relay 'R' and a destination 'D' node.

channels to indicate the channel connecting the source with the relay, the channel connecting the relay with the destination and the channel connecting the source with the destination, respectively. This fundamental topology of cooperation is further described and used in chapters 3, 4.

The relay channel can be thought of as a supporting channel for the direct channel between the source and destination. A key feature of the cooperative communication process is the processing of the signal received from the source node at the relay. These different processing schemes at the relay result in different cooperative communications types (relaying protocols).

The processing at the relay differs with the cooperation protocol used. In an AF relaying protocol, the relay simply amplifies the received version of the signal without additional processing and transmits the scaled version to the destination. Another processing of the received signal at the relay is in the DF relaying protocol where the relay decodes the received signal, re-encodes it and then retransmits it to the receiver. In selective relaying protocol, if the SNR of the signal received at the relay exceeds a certain threshold, the relay performs DF operation on the received signal, otherwise, if the channel between the source and the relay has severe fading such that the SNR is below the threshold, the relay performs nothing. Moreover, if Automatic repeat request (ARQ) protocol is implemented in the system and feedback channels are available from the destination to the source and the relay, the source may re-transmit the information to the destination or the relay may help by forwarding additional information about the received signal. The latter case is also termed as incremental relaying.

2.1.1 Cooperative Diversity

The idea of using multiple antennas for transmission and reception in wireless communication systems has been a hot research area with the aim to increase transmission rate and system capacity. A major issue in these researches is how to develop proper transmission techniques to exploit all of the diversities available in the space, time, and frequency domains while maintaining the design complexity to minimum. For the narrow-band wireless communications, any two adjacent frequency channels are considered frequency non-selective “flat”, thus frequency diversity among this narrow band is not possible and the only available diversities are in the space and time domains. For this scenario, the modulation and coding approach adopted is called Space-Time (ST) coding, exploiting the available spatial and temporal diversity.

Cooperative diversity is the set of techniques used to achieve spatial diversity in cooperative networks when the devices have restrictions on size and complexity. These techniques were initially introduced to achieve spatial diversity between correlated antennas in MIMO systems. Current and future high data rate wireless systems, such as Ultra Mobile Broadband (UMB), Long Term Evolution (LTE), and IEEE 802.16e (WiMAX) provide very high data rates per user over high bandwidth channels (5, 10, and 20 MHz). For example, the next fourth generation wireless networks which are under development and will be possibly deployed in few years, high date rates of

260 Mbps on the downlink and 60 Mbps on the uplink are promised [23]. However, these data rates can only be achieved by full-rank MIMO system. Moreover, a full-rank MIMO mobile user must have multiple antennas, and these antennas must see independent channel fades (uncorrelated) from the base station.

In practice, the small size of mobile devices does not allow to have multiple antennas, or the propagation environment cannot support MIMO because, for example, there is not enough scattering and therefore the separation distances required between the antennas should be very large. These limitations in implementing a MIMO system can be overcome by exploiting the distributed nature of other mobile devices (nodes) in the vicinity of the mobile device. One can view these nodes as a set of antennas distributed in the wireless system. The mobile device which is trying to communicate with the base station can be considered as a broadcast device because of the broadcast nature of the wireless channel. From this prospective, nodes in the wireless network can cooperate together for distributed transmission and processing of the source's information. The cooperating nodes act as a relay nodes for the source node. By this way, independent paths between the user and the base station are generated by introducing single or multiple relay channel(s). An example of such cooperative system is shown in Fig. 2.2. By employing more than one cooperative transmitter or more than one cooperative receiver one can either obtain transmit diversity or receive diversity, respectively.

2.1.2 Cooperative Communication Types

A typical cooperation strategy is accomplished over two orthogonal phases, either in time domain or frequency domain, to avoid interference between the two phases:

- In the first phase, the source broadcasts information to its destination, and the information is also received by the relay at the same time.

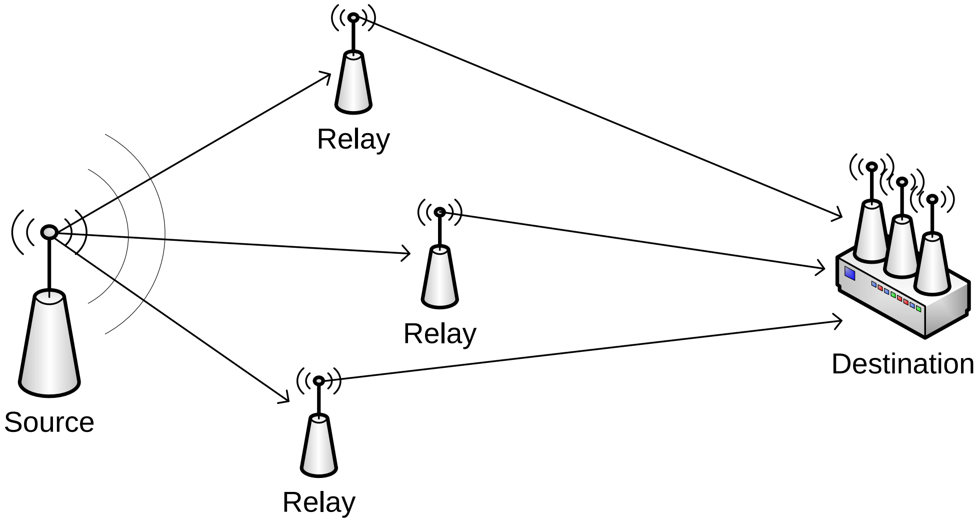


Figure 2.2. Example of three relays comprising a MIMO cooperative transmitter.

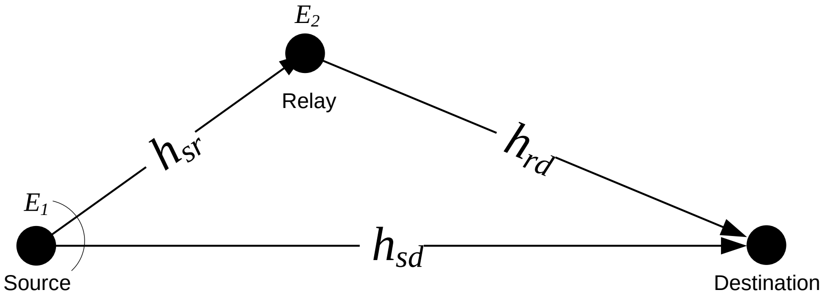


Figure 2.3. Simple cooperation model, with source and relay transmission energy/symbol set to E_1 and E_2 respectively.

- In the second phase, the relay can help the source by forwarding or retransmitting the information to the destination.

it is referred to this transmission strategy as orthogonal transmission.

Fig. 2.3 depicts a general relay channel, where the source and the relay transmits with energy/symbol E_1 and E_2 respectively. In chapters 3 and 4, we will consider the case where the source and the relay transmit with equal energy/symbol E. In the first phase, the source broadcasts its signal message to both the destination and the relay. The received signals y_{sd} and y_{sr} at the destination and the relay, respectively, can be written as:

$$y_{sd} = \sqrt{E_1} h_{sd} x + n_{sd} \tag{2.1.1}$$

$$y_{sr} = \sqrt{E_1} h_{sr} x + n_{sr} \tag{2.1.2}$$

where E is the transmission energy per symbol at the source, x is the transmitted information symbol and h_{sd} and h_{sr} are the channel attenuations on the direct and inter-user channels, respectively. n_{sd} and n_{sr} are zero mean AWGN on the direct and inter-user channels, respectively.

In the second phase, the relay processes the received signal and forwards the output to the destination. The transmitted signal from the relay could be an amplified version of the received signal or another variation of the received information from the source. The signal received at the destination via the relay channel can be modeled as:

$$y_{rd} = \sqrt{E_2} h_{rd} f(y_{sr}) + n_{rd} \tag{2.1.3}$$

where $f(\cdot)$ indicates the processing done at the relay for the received signal from the source. As mentioned in the introduction of this chapter, the processing done at the relay depends on the type of protocol used for cooperation. In the following sections, more detailed description of the cooperations protocols are presented.

Amplify And Forward Relaying

In an AF protocol, the relay simply scales the received signal over the inter-user channel and transmits the amplified noisy signal to the destination as soft output. In this protocol, the amplification at the relay aims to compensate for the effect of the channel attenuation (propagation loss and/or fading) between the source and the relay. The scaling factor of the amplification at the relay is usually calculated from the energy of the received signal as:

$$G_r = \frac{\sqrt{E_2}}{\sqrt{E_1 |h_{sr}|^2 + N_0}} \tag{2.1.4}$$

so the signal received at the destination over the relay channel is equal to:

$$y_{rd} = G_r h_{rd} y_{sr} + n_{rd} \tag{2.1.5}$$

where y_{sd} is the signal presented in (2.1.2).

There are different techniques at the destination to combine the two received copies of the signal x over the source link and relay link. It is well-known from the literature that the optimal diversity technique to maximize the overall SNR is the Maximal Ratio Combiner (MRC) [24]. The MRC method requires a coherent detector and knowledge of all channel coefficients. The SNR at the output of the MRC is equal to the sum of the two SNRs of the two received signals from both branches.

Decode And Forward Relaying

If the relay can employ more advanced processing of the sources signal, then decode-and-forward is implemented. This further processing of the source's signal include decoding, re-encoding, and then retransmitting it to the receiver. The transmitted signal from the relay to the destination is an estimate of the source's transmitted signal. For this technique, the relay is usually located in a place between the source and the destination, or chosen from the relays pool such that the two channels connecting the relay to the source and the destination having relatively higher SNR than the direct channel.

When the relay correctly decodes to the original source's message, the signal forwarded to the destination can help to boost the performance at the destination since the received signal over the relay channel usually have higher SNR. However, when the decoding and re-encoding process at the relay results in erroneous estimates, an incorrect signal is forwarded to the destination, so that the decoding at the destination may result in a degraded performance.

Compression And Forward Relaying

While the DF forwards the "decoded and then re-encoded" message from the signal that it received, the AF forwards the re-scaled received signal as is in a soft information form without any immature decisions about the received data. Since the DF method is based on the assumption that the relay can correctly decode the received signal from the source, this assumption is no more valid when the inter-user channel is experiencing low SNR and so the recovered message has many errors. At these channel conditions, the AF would result in a better performance although it does not benefit from the error correction capability of the received signal at relay.

In the Compress and Forward (CF), the relay quantizes and then compresses the received version of the original message before re-transmission. This is performed to maintain the soft state of the received bits at the relay and forward sufficiently minimum information about the received version at the relay to the destination. The soft channel output symbol generated at the relay is represented by a minimum sequence of bits that can help the destination to determine the most probable transmitted symbol with simple decoding. If a Soft Input Soft Output (SISO) decoder can be implemented at the relay, the decoded soft output of the SISO decoder is quantized and compressed for transmission. The latter, is studied under coded cooperation which is presented in the next section. The destination node combines the received message from the source node with the additional compressed information received from the relay about it to determine the actual transmitted signal. At the destination node, an estimate of the quantized and compressed message is obtained by decoding the received sequence of bits. This decoding operation involves mapping the received bits back into a set of values that estimate the transmitted message.

Coded cooperation

The difference between coded cooperation and the previous cooperation schemes is that the former is implemented at the level of the channel coding. The earlier presented schemes, i.e. AF and DF, are based on diversity combining at the distention. On other words, the relay retransmits estimates of the bits sent by the source. On the other hand, in coded cooperation scheme the relay sends Incremental Redundancy (IR) (additional code bits), which, when combined at the receiver with the codeword sent by the source, would result in a codeword with larger redundancy.

The encoder in a non-cooperative system (i.e., point-to-point communications), applies channel coding to the source bits to add additional redundant bits (parity) that can help in increasing the recovery chances of original source message (codeword containing source bits and parity bits) transmitted through the noisy channel to the destination. In coded cooperation, the codeword is divided in two parts, one part is transmitted through the direct channel, and the other part is generated and transmitted through the relay channel. Usually, the latter will have better SNR at the destination and therefore it can result in better forward error correction at the destination decoder. The second part which is received over the relay channel is generated from the first part. One simple way to do this is by puncturing the codeword generated at the source's encoder before transmission, and then the relay can guess the punctured bits by using Forward Error Correction (FEC). These punctured bits are regenerated and transmitted to the destination as the second part of the codeword.

Fig. 2.4 shows an example for coded cooperation. In this example, the source uses CRC to add a small sequence of bits to be used at the relay as error correction check bits after decoding the channel code. There are two puncturing schemes in this example and they are chosen such that only the bits removed from the source's transmission are the bits transmitted from the relay. In this specific example, the

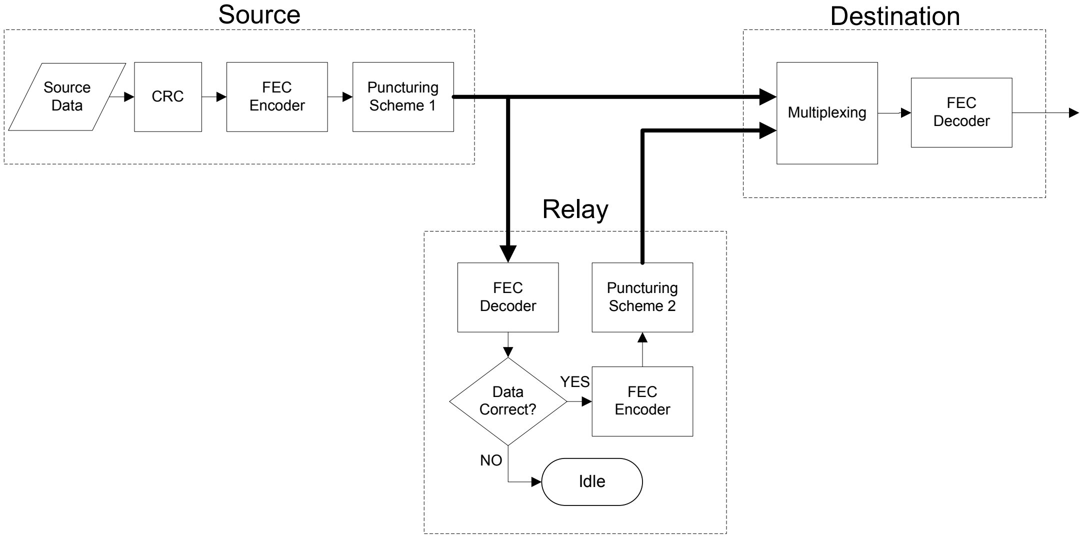


Figure 2.4. Example of coded cooperation in which the relay transmits the IR when CRC finds no errors in the decoded sequence by the CRC.

relay transmits only when the decoder at the relay results in error free decoding.

In the example of Fig. 2.4, the source bits of length K are encoded to a codeword of length N. Before transmission, the code rate is increased by puncturing the output codeword to reduce it's length to N_1, where $K/N_1 < 1$. The relay uses the received codeword of length N_1 to recover the source K bit message and check for errors using the included CRC sequence. If the decoding result is error free, the relay re-encodes the bits with the same code space used at the source and then punctures it to obtain a sequence of bits of length N_2, where $N = N_1 + N_2$. The destination receives the two parts of the code over the two channels and performs decoding for the overall codeword of code rate $R = K/N$

The whole coded cooperation process can be compared to the non-cooperative case where the source node performs channel encoding at a code rate R and then transmits the output to the destination without the assistance of relays. However, in the coded cooperation one part of the codeword is received over a better channel, which means a better performance at the destination. The codeword broadcasted by the source node belongs to a code that is weaker than the code used at the receiver. The code at the receiver is stronger from that at the source node since it combines the N_2 incremental redundancy bits received from the relay.

2.2 Turbo Product Codes

The Turbo principle was first introduced by C. Berrou in 1993 [25] and achieved for the first time an error correcting code within 0.7 dB of the Shannon limit [26]. This principle consists of iteratively decoding of two parallel concatenated Recursive Systematic Convolutional (RSC) codes through a random interleaver. The iterative decoding is based on SISO decoding of the received sequence and on the optimal transfer of the decoding information from one decoding stage to the next. After the first introduction of the results of the convolutional turbo codes, the turbo principle have been applied to block codes to obtain performances comparable to the convolutional turbo codes sine the block codes exhibit lower complexity. The first results for the Block Turbo Codes (BTC) were presented in 1994 [20] where the authors proposed a new SISO decoder based on Chase II decoding of the received sequence to find the Maximum-Likelihood (ML) codeword and a competing codeword that are used to calculate the Log-Likelihood Ratio (LLR) of ML codeword.

2.2.1 Parallel Concatenation of Product Codes

The primary principle of turbo decoding is to iteratively decode two or more connected component codes made from the same systematic bits. For BTC, the component (constituent) codes are constructed from elementary block codes such as Bose-Chaudhuri-Hochquenghem (BCH) or Reed-Solomon (RS), etc. Connection between codes is gained by parallel or serial concatenation for two or more codes. Fig. 2.5 shows the two types of concatenations possible for the turbo codes. In addition, the design of BTC also depends on the nature of the interleaver, denoted as "$\prod$" in Fig.

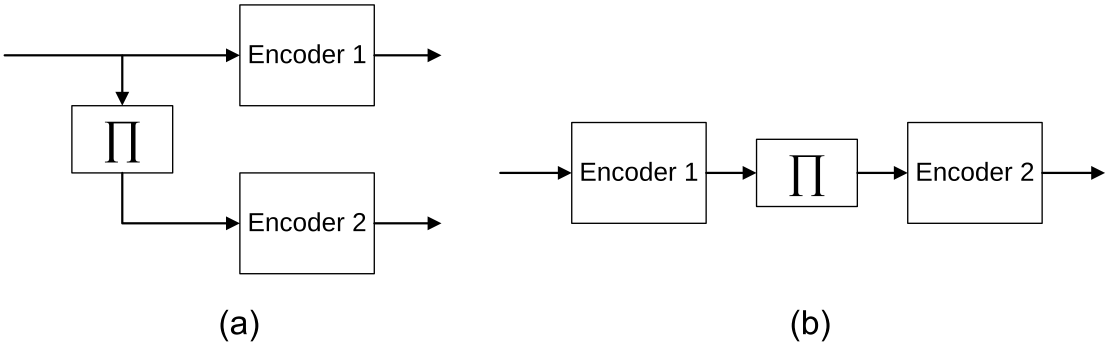


Figure 2.5. Concatenation of two codes in (a) Parallel and (b) Serial form

2.5(a) and (b), whether it is uniform or pseudo-random interleaver.

The first introduction for serial concatenation of block code was in 1954 by Elias [27]. In this thesis, the serial concatenation is referred to as TPC. The interleaver used for the TPC is uniform (matrix) which transforms the rows into columns and vice versa, where the data is written in the matrix row by row and read from the matrix column by column.

Consider two block codes $\mathcal{C}^1$ and $\mathcal{C}^2$ with parameters (n_1, k_1, δ_1) and (n_2, k_2, δ_2), respectively, where n_i, k_i and δ_i are the length of the codeword, the dimension of the code space (input information length) and the minimum Hamming distance for the code space $\mathcal{C}^i$ $i = 1, 2$, respectively. When the two codes $\mathcal{C}^1$ and $\mathcal{C}^2$ are serially concatenated as shown in Fig. 2.5(b), the data will be first placed in a matrix of dimension $k_2 \times k_1$ before entering Encoder 1 which uses the code space $\mathcal{C}^1$. The encoder will encode the matrix row by row to produce the row's checks (parity) by adding $(n_1 - k_1)$ parity bits to the end of all k_2 rows. The dimension of the output encoded matrix will be $(k_2 \times n_1)$. This matrix contains k_2 codewords of length n_1 arranged in rows. The interleaver performs a transpose for the matrix by converting the columns of the encoded matrix to columns so that the input matrix for Encoder 2 has dimension equal to $(n_1 \times k_2)$. Encoder 2, which uses the code space $\mathcal{C}^2$, encodes

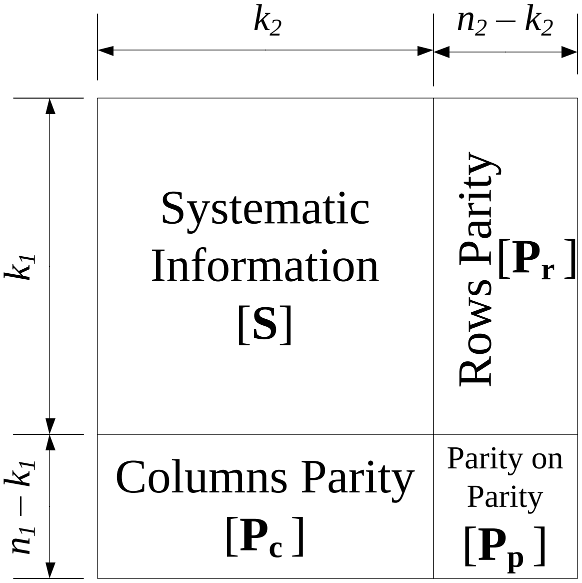


Figure 2.6. The output matrix for serial concatenation of two block codes (TPC)

the rows of the input matrix by adding $(n_2 - k_2)$ parity bits to the end of each row. Therefore, at the output of the encoder, the new dimension of the matrix will be $n_1 \times n_2$. The latter matrix, which is shown in Fig. 2.6, is used for transmission to the receiver after modulation.

What distinguishes the serial concatenation from parallel concatenation, is the fact that the second code $\mathcal{C}^2$ in serial concatenation is applied to the binary parity bits generated by the first code $\mathcal{C}^1$ as well as the systematic binary bits as can be seen in Fig. 2.6. The parity resulted from applying the second code on the first code's parity is called checks on checks (or parity on parity).

The resultant output code from the TPC encoder, using the two component codes $\mathcal{C}^1$ and $\mathcal{C}^2$, can be presented by the parameters (n, k, δ), where $n = n_1 \times n_2$ and $k = k_1 \times k_2$, respectively. The rate for the new code is $R = R_1 \times R_2$, where $R_i = k_i/n_i$ is the code rate for the $\mathcal{C}^i$ code. One attractive feature in the TPC is that the

$\delta = \delta_1 \times \delta_2$; the minimum Hamming distance is the multiple of the two Hamming distances of the two constituent codes. In the following, this important aspect of the serially concatenated codes with uniform interleaver is proved.

To establish this result, it is important to show first that all the n_2 columns are in the code space of $\mathcal{C}^1$ and all the n_1 rows are in the code space of $\mathcal{C}^2$. A vector is in the code space of a certain code if it is a codeword for that code with 1:1 relation with a unique information input sequence. This means that a codeword in the code space can be decoded to a unique original information sequence. It is straight forward to show that the n_1 rows of the coded matrix are in the code space $\mathcal{C}^2$ since they are generated using this component code. This rule also applies for the first k_2 columns of the coded matrix to show that they belong to the code space $\mathcal{C}^1$ because they are generated using this code. What remains is to show that the last $(n_2 - k_2)$ columns of the coded matrix belong to the code space of $\mathcal{C}^1$.

To show this, let $\mathbf{G}^i$, $i = \{1, 2\}$, be the generator matrix for the code $\mathcal{C}^i$. A linear code generator matrix is any matrix whose rows are vector representations for the base of the code space . By definition of the generator matrix, the dimension of $\mathbf{G}^i$ is $k_i \times n_i$. For a systematic block codes, this matrix can be written in the form:

$$\begin{aligned}
\mathbf{G}^i &= \left[\begin{array}{cccc|cccc}
1 & 0 & \cdots & 0 & p_1^1 & p_2^1 & \cdots & p_{n_i-k_i}^1 \\
0 & 1 & & 0 & p_1^2 & p_2^2 & & p_{n_i-k_i}^2 \\
\vdots & & \ddots & \vdots & \vdots & & \ddots & \vdots \\
0 & 0 & \cdots & 1 & p_1^{k_i} & p_2^{k_i} & \cdots & p_{n_i-k_i}^{k_i}
\end{array}\right] && (2.2.1) \\
&= \left[\mathbf{I}_{k_i \times k_i} \middle| \mathbf{P}^i{}_{k_i \times (n_i - k_i)}\right] && (2.2.2)
\end{aligned}$$

where the first k_i columns of the generator matrix compose an identity matrix of size k_i. These columns are responsible for the systematic bits in the output codeword. The remaining columns of $\mathbf{G}^i$, which forms a matrix $\mathbf{P}^i$ of size $(k_i \times (n_i - k_i))$ that generates the parity bits.

The sub-matrices $\mathbf{P_r}$, $\mathbf{P_c}$ and $\mathbf{P_p}$ of the TPC matrix in Fig. 2.6 can be expressed in terms of the systematic matrix $\mathbf{S}$ and the corresponding generator matrices $\mathbf{G}^1$ and $\mathbf{G}^2$ of codes $\mathcal{C}^1$ and $\mathcal{C}^2$, respectively, in the following way:

$$\begin{aligned} \mathbf{P_c} &= \left[\mathbf{S}^T\mathbf{P}^1\right]^T \\ &= \mathbf{P}^{1^T}\mathbf{S} \qquad (2.2.3) \\ \mathbf{P_r} &= \mathbf{S}\ \mathbf{P}^2 \qquad (2.2.4) \\ \mathbf{P_p} &= \mathbf{P_c}\mathbf{P}^2. \qquad (2.2.5) \end{aligned}$$

The objective here is to show that $\mathbf{P_p}$ can be written also in the form:

$$\mathbf{P_p}^T = \mathbf{P_r}^T\mathbf{P}^1 \qquad (2.2.6)$$

or

$$\begin{aligned} \mathbf{P_p} &= \left[\mathbf{P_r}^T\mathbf{P}^1\right]^T \\ &= \mathbf{P}^{1^T}\mathbf{P_r} \qquad (2.2.7) \end{aligned}$$

i.e. the last $(n_2 - k_2)$ columns of the TPC matrix belongs to the code space of $\mathcal{C}^1$. To get to this result, equation (2.2.5) can be rewritten after substituting for $\mathbf{P_c}$ from equation (2.2.3):

$$\begin{aligned} \mathbf{P_p} &= \mathbf{P_c}\mathbf{P}^2 \\ &= \mathbf{P}^{1^T}\mathbf{S}\ \mathbf{P}^2 \qquad (2.2.8) \\ &= \mathbf{P}^{1^T}\mathbf{P_r} \qquad (2.2.9) \end{aligned}$$

where (2.2.4) is substituted in (2.2.8). This is a very important result for the decoding of TPC because it enables the transfer of extrinsic information about the parities of the rows and the columns to the consecutive decoding stage unlike the decoding of the parallel concatenated Convolutional Turbo Codes (CTC) where only the extrinsic information about the systematic bits can be transferred to the next decoding stage.

The TPC resulted from serial concatenation with uniform interleaver is sometimes called the complete TPC to distinguish it from the incomplete TPC in which only the

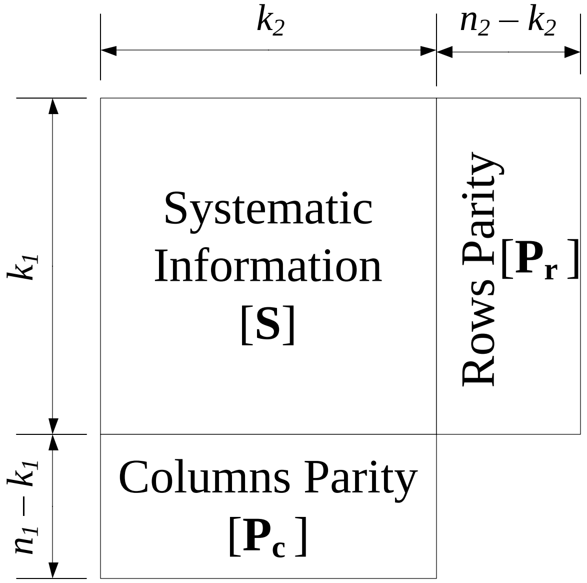

k_2
$n_2 - k_2$
k_1
$n_1 - k_1$
Systematic Information [S]
Rows Parity [$\mathbf{P_r}$]
Columns Parity [$\mathbf{P_c}$]

Figure 2.7. The incomplete turbo product code.

parities of rows and columns are produced excluding the parity of parity as show in Fig. 2.7. The incomplete TPC may result from parallel concatenation of of the two encoders with uniform interleaver. The decoding of the parity bits in the incomplete TPC does not benefit from the extrinsic information of the previous decoding stage.

In conclusion, the case of serial concatenation of two linear and systematic block codes (e.g. BCH, RS, etc.) with uniform interleaving, the n_2 columns can be decoded using $\mathcal{C}^1$ code, and the n_1 rows can be decoded using the $\mathcal{C}^2$ code. Moreover, if the coding order is changed, rows followed by columns instead of columns followed by rows, the output matrix will be the same. The iterative decoding exploits this property by decoding the rows after columns and the columns after rows and using the output extrinsic information as *a-priori* information for the stage that follows.

Having proved that all the rows and columns of complete TPC are codewords, we will show now that for the same code, the minimum distance of the code is given by

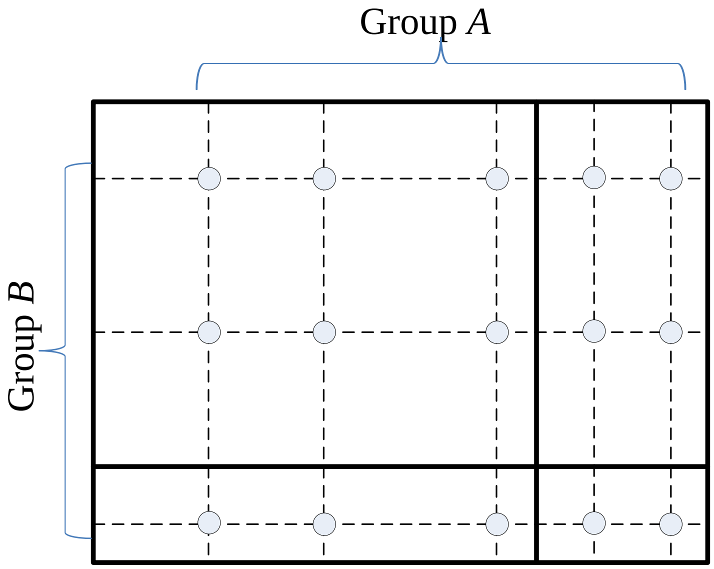


Figure 2.8. Example of a TPC matrix having all zero rows and columns except for $\delta_2 = 5$ columns and $\delta_1 = 3$ rows, in groups *A* and *B*, respectively.

$\delta = \delta_1 \times \delta_2$. For this purpose, the linear properties of the component codes $\mathcal{C}^1$ and $\mathcal{C}^2$and it's weights are used. Since the component codes are linear block codes, we can say that the concatenated serial coding is also linear and accepts linear operations.

The minimum distance of a linear code is defined as the minimum non-zero weight that a codeword of that code could have. Assuming that the coded matrix has only the codewords of minimum weights in its columns and rows and all the other columns and rows are "zero-codewords". i.e, this coded matrix has δ_2 non-zero columns (Group A) having the minimum weight δ_1 of the code $\mathcal{C}^1$, and it has δ_1 non-zero rows (Group B) all of them have the lowest possible weight of $\mathcal{C}^2$ codeword which is δ_2. Fig. 2.8 shows an example of this matrix where $\delta_1 = 3$ and $\delta_2 = 5$. The total non-zero elements of the matrix is thus equal to $\delta = \delta_1 \times \delta_2$ and is equal to 15 in the example of Fig. 2.8. Therefore, the minimum possible weight of the non-zero TPC is equal to to the multiple of the weights of the component codes.

The choice of interleaving for a serial concatenation with non-uniform interleaving is very complex and has been barely studied until now. For this type of block turbo codes and using the same component codes like the TPC, the resulting code will have the same $n = n_1 \times n_2$, $k = k_1 \times k_2$ and the same rate $R = R_1 \times R_2$. However, the minimum distance of this code is not guaranteed to be equal to $\delta_1 \times \delta_2$ of the two component codes and it depends mainly on the type of interleaving used. In the worst case $\delta = \sup(\delta_1, \delta_2)$, this is because the last $n_2 - k_2$ columns of the coded matrix no longer belong to the code space of C^1 [28]. In addition, this fact will have negative consequences on the operation of iterative decoding thereafter. Lastly, on a practical level, the implementation of pseudo-random interleaving can lead to large complexity. Therefore, we can deduce that serial concatenation of block codes with uniform interleaving (or product codes) makes up the best concatenated code for the BTC.

2.2.2 Performance of Product Codes with BCH Component Codes

The first Product Code introduced by Elias in 1954 was based on Hamming codes. However, the encoding process described in the previous section applies to any systematic linear block code (e.g. BCH, RS, etc.). The properties discussed in the previous section also remain true when the number of elementary codes is higher than two. In this section the case when BCH codes are used as component codes of the TPC is further studied for performance analysis, nevertheless, the results shown for the performance can apply for any type of systematic linear block code.

BCH codes were invented independently by two separate research teams, namely by Hocquenghem, and by Bose and Ray-Chaudhuri, in years 1959 and 1960, respectively. The main advantage of BCH codes is that they can be easily decoded using an algebraic method known as syndrome decoding. Therefore, a very simple electronic hardware can perform the task. This means that both encoder and decoder may be made small and low-powered device. On the other hand, they are also highly flexible as a class of codes, allowing vast range of possible block length and the allowed error thresholds, meaning that a custom code can be designed to a given specification.

Technically, a BCH code is a cyclic code over a finite field with a particularly chosen generator polynomial. It is also able to correct multiple random error patterns. The length of primitive BCH code is given by $n = 2^m - 1$, where m is positive integer. The minimum distance of the code is odd and the number of maximum correctable bits is given by:

$$t = \lfloor \frac{\delta - 1}{2} \rfloor. \tag{2.2.10}$$

where $\lfloor \cdot \rfloor$ returns the floor of argument.

2.2.3 Soft Decoding of Block Codes

Decoding of block codes can be carried out using two criteria depending on the nature of the input bits and the bearable complexity that can be afforded at the decoder. In the hard decoding criterion, when the input at the decoder is considered binary, optimal decoding is based on finding the codeword with the minimum Hamming distance from the received input vector. One of the main contributions in the area of hard decoding of blokc codes is for Berlekamp [29] and Massey [30] in the case of cyclic codes. These decoders have low complexity but they yield lower coding gain than soft decoding methods.

Soft decoding is often used in the case of availability of channel soft output at the input of the decoder and the decoder complexity is tolerable. Soft channel output are available when the codewords are transmitted by linear modulation(e.g. Phase Shift Keying (PSK) or Quadrature Amplitude Modulation (QAM)) through noisy channels. For simplicity, the case of Binary Phase Shift Keying (BPSK) transmission over a Gaussian channel and reception by a coherent receiver is considered. The observations at the input of the decoder have the form:

$$y_i = x_i + n_i \tag{2.2.11}$$

where x_i is the transmitted binary BPSK symbol $\in \{-1, +1\}$ and n_i is the additive noise with σ^2 variance. It was found that the optimal decoding of the input soft vector is based on finding the codeword with the minimum Euclidean distance from the input vector. With this decoder, it possible to significantly improve the coding gain compared to hard decoding.

For the description of the iterative decoding based on soft decoding in this section, we will consider the case of a product code obtained by serial concatenation of two BCH codes. Let $\mathcal{C}^1$ be the code applied along the columns and $\mathcal{C}^2$ along the rows. One decoding iteration consists of two decoding stages, i.e. iterative decoding is performed by decoding the columns using a SISO decoder based on the code $\mathcal{C}^1$ followed by a SISO decoding for the rows based on the code $\mathcal{C}^2$ and then restart the process for the next iteration. When the columns are decoded, codewords from $\mathcal{C}^1$ code space results along the columns, but in case there are errors in the decoding, the rows are not necessarily $\mathcal{C}^2$ codewords. On the other hand, decoding the rows will lead to $\mathcal{C}^2$ codewords in the rows, nevertheless, columns are not necessarily $\mathcal{C}^2$ codeword. By repeating the process, the decoding process may converges towards a product code codeword, such that all the columns and all the rows are codewords in the $\mathcal{C}^1$ and $\mathcal{C}^2$ code spaces, respectively.

The iterative decoding of product codes using hard decoders instead of SISO decoder has been studied in [31, 32]. It was found that this process is suboptimal compared to the soft decoding process. The receiver truncates the received soft channel output and returns binary bits sequence before carrying out the decoding process. The truncation process (applying threshold to the received input bits), which simply returns the sign of observation, leads to a loss of soft information which would be of great help in the decoding process. Experimental results shows that the first hard decoding of the rows led to a loss ranging of 1.0-2.0 dB. Thus, For the rest of the dissertation we will only use soft decoding of the elementary codes as the basic component for the iterative decoder. In the following sections, two famous soft decoding techniques of block codes are presented.

Optimal Block Decoding Decision

Consider the transmission of a binary codeword $C^t = \{c_1^t, c_2^t, \cdots, c_n^t\}$, where $c_i \in \{0, 1\}$, having the parameters (n, k, δ) after BPSK modulation over a Gaussian channel. This codeword belongs to a code space $\mathcal{C}$ of dimension 2^k, where $\mathcal{C} = \{C^1, C^2, \cdots, C^{2^k}\}$. The output vector after BPSK modulation is $X = \{x_1, x_2, \cdots, x_n\}$, where $x_i \in \{-1, +1\}$.

The received bits for the transmitted codeword is obtained by (2.2.11), so at the decoder input we have an observation vector $R = \{r_1, r_2, \cdots, r_n\}$. The decoder will search for the optimal decision based on search criterion which is discussed later. The optimal search criterion is based on minimization of the probability of error per bit information symbol (P_b) or the probability of error per codeword or block (P_c).

The soft decoding for block codes presented in this section is proposed by Pyndiah *et. el.* in 1994. This decoding method targets to minimizing the block P_c. The minimization of P_c is achieved by using the *Maximum A-posteriori Probability* (MAP) method as follows:

$$\begin{aligned} D &= \arg\max_{C^i \in \mathcal{C}} \left\{ P(C^t = C^i | R) \right\} && (2.2.12) \\ &= \arg\max_{C^i \in \mathcal{C}} \left\{ \frac{P\{R | C^t = C^i\} \, P\{C^t = C^i\}}{P\{R\}} \right\} && (2.2.13) \end{aligned}$$

where $D = \{d_1, d_2, \cdots, d_n\}$ is the decision codeword. This decision rule finds the codeword that is most probably transmitted given the received sequence R. The second equality follows after using Bayes rule. Since a k-bit information block is mapped by a one-to-one relation to a codeword, all the codewords have the same probability of $1/2^k$ since all data bits are mutually independent with equal probabilities, consequently, the term $P\{C^t = C^i\}$ is equal for all C^i as well as the term $P\{R\}$ because the channel is independent of the transmitted codeword. Therefore, the relation (2.2.13) reduces to:

$$D = \arg\max_{C^i \in \mathcal{C}} \left\{ P\left\{ R | C^t = C^i \right\} \right\} \tag{2.2.14}$$

In AWGN channel, conditional Gaussian probability density of can be substituted for $P\{R|C^t = C^i\}$ as follows:

$$\begin{aligned} P\left\{R|C^t = C^i\right\} &= \prod_{j=1}^{n} \left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(r_j - c_j^i)^2}{2\sigma^2}} \right) && (2.2.15) \\ &= \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right)^n e^{\frac{-\sum_{j=1}^{n} (r_j - c_j^i)^2}{2\sigma^2}} && (2.2.16) \end{aligned}$$

Substituting (2.2.16) in (2.2.14) we get:

$$\begin{aligned} D &= \arg\max_{C^i \in \mathcal{C}} \left\{ \left(\frac{1}{\sqrt{2\pi\sigma^2}} \right)^n e^{\frac{-\sum_{j=1}^{n} (r_j - c_j^i)^2}{2\sigma^2}} \right\} && (2.2.17) \\ &= \arg\max_{C^i \in \mathcal{C}} \left\{ e^{\frac{-\sum_{j=1}^{n} (r_j - c_j^i)^2}{2\sigma^2}} \right\} && (2.2.18) \\ &= \arg\max_{C^i \in \mathcal{C}} \left\{ -\sum_{j=1}^{n} (r_j - c_j^i)^2 \right\} && (2.2.19) \\ &= \arg\min_{C^i \in \mathcal{C}} \left\{ \sum_{j=1}^{n} (r_j - c_j^i)^2 \right\} && (2.2.20) \\ &= \arg\min_{C^i \in \mathcal{C}} \left\{ ||R - C^i||^2 \right\} && (2.2.21) \end{aligned}$$

where $|| \cdot ||$ returns the Euclidean distance of it's argument. In conclusion, the Maximum A-posteriori Probability (MAP) decoding try to minimize the distortion introduced by the transmission channel, called square of the Euclidean distance, between the received vector and the codeword C^i by assuming that the most probable transmitted codeword is the closest in Euclidean distance to the received vector.

Optimal decoding presented in (2.2.21) uses exhaustive search to find the most probable codeword, hence the complexity of decoding is given by the number of all codewords in the code space, which is equal to 2^k. So the complexity of decoding grows exponentially with the length of information block. Therefore, the decoding complexity is considered reasonable for codes with a small size, i.e. $k \leq 8$. However, the extended BCH code (64,51,6) contains $2.25 \times 10^1 5$ codewords, so an exhaustive search is impossible to implement. Block codes used for turbo codes are often large in order to obtain high coding outputs. Exhaustive search for the most probable codeword is impractical for these codes. Alternative sub-optimal decoding algorithms were proposed as a solution for block codes with large lengths to reduce the complexity to a tolerable level.

Soft Decoding of Block Codes (Chase Algorithms)

Chase proposed a sub-optimal algorithm of lower complexity to carry out the soft decoding of block codes [33]. Instead of searching all the code space for the optimal codeword which have the minimum Euclidean distance from the observation, this algorithm searches the optimal codeword from a set of binary sequences that are close to the observation in the sense of Euclidean distance. This is done by generating a set of test patterns with the shortest Euclidean distance from the observation. This subset contains the optimal codeword with the minimum distance with a probability close to one.

The code space is considered of dimension n and contains 2^k codewords $\mathcal{C} = \{C^1, C^2, \cdots, C^{2^k}\}$. We can also consider the observation vector R as a point in this space. Each element of the vector R is a component in a specific dimension (axis) of space, whose value indicates the projection of the observation vector on this axis. Any codeword C^i is regarded as a point in the space with n components where each component has a value in -1,+1. Moreover, not all the points in the space of size n

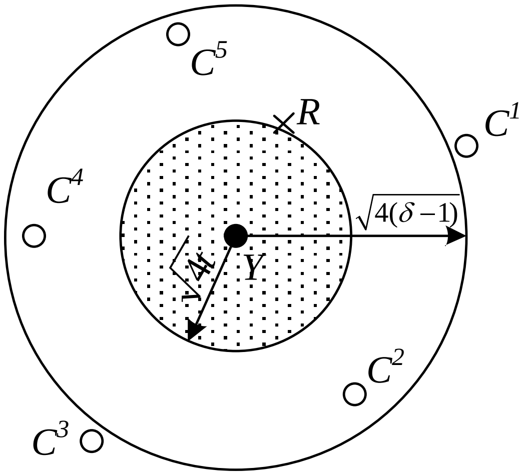


Figure 2.9. Finding the most probable codewords within the sphere of radius $\sqrt{4(\delta - 1)}$.

and with component values in -1,+1 are codewords and there are only $2^k \ll 2^n$ of them are considered codewords.

The first Chase decoding algorithm considers the most probable codewords within the Hamming distance of the code. Chase defines a zone containing a subset of codewords Ω closest to the received vector within a sphere of radius less than the Hamming distance. The center of the sphere is given by

$$Y = \{y_1, y_2, \cdots, y_n\}, \quad \text{where} \quad y_i = sign(r_i), \tag{2.2.22}$$

where this point represents the closest point the the n-dimensional space to R. The Euclidean distance is related to the Hamming distance by the relation:

$$d_E(C^1, C^2) = \sqrt{4 \times d_H(C^1, C^2)} \tag{2.2.23}$$

So the radius of the sphere is equal to $\sqrt{4(\delta - 1)}$. Therefore, this algorithm is capable of of correcting at most $(\delta - 1)$ bits, and if the noise displacement moves the received vector by more than $(\delta - 1)$ bits, the algorithm will result in an erroneous decision.

To find the subset of codewords Ω contained within the sphere, the decoder constructs another sphere of radius $\sqrt{4t}$, as shown in Fig. 2.9, and decode all the possible points in the sphere to codewords using hard decoder. This way, the decoder scans the $\sqrt{4(\delta-1)}$ from the inner sphere because $\sqrt{4(\delta-1)} = \sqrt{4(2t)} = \sqrt{4t+4t}$, i.e. each point in the inner sphere is center for another sphere of radius $\sqrt{4t}$. Therefore, the complexity of this algorithm is determined by the number of points in the inner sphere which determines the number of hard decodings performed. Thus, the latter is given by:

$$N_d = 1 + \sum_{i=1}^{t} \left(\begin{array}{c} n \\ i \end{array} \right) \sim \frac{n^t}{t!} \qquad (2.2.24)$$

this restricts the algorithm to codes with short length and low correction capacity.

The first Chase algorithm (Chase I) is presented in Algorithm 1. For a (63, 57, 3) code, MAP algorithm requires exhaustive search among $2^{57} \simeq 1.5 \times 10^{17}$ codewords, on the other hand, Chase I requires search among only 63 codewords (t=1) with 63 hard decodings to find the most probable codeword within the sphere.

```
Input: Observation vector $R = \{r_1, r_2, \cdots, r_n\}$
Calculate $Y = \{y_1, y_2, \cdots, y_n\}$, $y_i = sign(r_i)$;
Calculate $\mathcal{S}$: Sphere of radius $\sqrt{4(\delta-1)}$ centered at $Y$;
Set $\Omega = \Phi$;
foreach $S_i$ Point in $\mathcal{S}$ do
    $C^i$ = Hard decoding($S_i$);
    if $C^i$ is a codeword then
        $\Omega = \Omega \cup C^i$;
        Calculate $d_i = d_E(R, C^i)$;
    end
end
Output: Decision codeword from $\Omega$ with lowest $d_i$
```

Algorithm 1: First Chase Algorithm for Soft Decoding

Chase proposed another algorithm, known as Chase II, to reduce the complexity even more than his first algorithm by reducing the number of hard decodings performed. A small subset of the points included inside the sphere of radius $\sqrt{4t}$ is considered for hard decoding in order to construct the subset Ω. This is done by first measuring the reliability of the bits of the vector Y by defining a log likelihood probability ratio for each bit y_i. Assuming BPSK modulation in AWGN channel, the channel reliability for bit y_i is defined as:

$$\begin{aligned}
L(y_i) &= \ln\left(\frac{P(x_i=+1|r_i)}{P(x_i=-1|r_i)}\right) && (2.2.25)\\
&= \ln\left(\frac{P(r_i|x_i=+1)}{P(r_i|x_i=-1)}\right) + \ln\left(\frac{P(x_i=+1)}{P(x_i=-1)}\right) && (2.2.26)\\
&= \ln\frac{\frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(r_i-1)^2}{2\sigma^2}}}{\frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{-(r_i+1)^2}{2\sigma^2}}} && (2.2.27)\\
&= \ln\left(e^{\frac{-(r_i-1)^2+(r_i+1)^2}{2\sigma^2}}\right) && (2.2.28)\\
&= \ln\left(e^{\frac{4r_i}{2\sigma^2}}\right) = \frac{2}{\sigma^2}r_i && (2.2.29)
\end{aligned}$$

where the second part of (2.2.26) is equal to zero since the two probabilities in the numerator and the denominator are both equal to $\frac{1}{2}$ when there are no *a-priori* information at the start of decoding. In a channel experiencing attenuation, where each bit is attenuated by the value a_i, the result in (2.2.29) becomes:

$$L(y_i) = \frac{2a_i}{\sigma^2}r_i \qquad (2.2.30)$$

The magnitude of the LLR value in (2.2.29) and (2.2.30) indicates the reliability of the decision; when $L(y_i) \to 0$, the two probabilities that y_i is either -1 or +1 are almost equal, and when $L(y_i) \to \pm\infty$, the decoder is more certain about it's decision.

From the reliability values for the vector Y, the lowest q positions are selected and called the Least Reliable Bits (LRB). The decoder permutes -1's and +1's in the LRB positions to get a set of test patterns of size 2^q that contains vectors in the space within the sphere of radius $\sqrt{4t}$ centered at Y. This reduces the number of decodings required to get the candidate codewords Ω from $n^t/t!$ to 2^q. However, the reduction of search zone leads to increasing the probability that the transmitted word is outside the search zone, and so the performance of the decoder is degraded. A compromise between the reduction of the search zone and degradation of the performance is proposed by Chase by choosing the optimal value of q using the following empirical relation:

$$q = \left\lfloor \frac{\delta}{2} \right\rfloor \tag{2.2.31}$$

```
Input: Observation vector $R = \{r_1, r_2, \cdots, r_n\}$
Calculate $Y = \{y_1, y_2, \cdots, y_n\}$, $y_i = sign(r_i)$;
foreach $y_i$ in $Y$ do
    Calculate $L(y_i)$: the reliability of vector $Y$ elements;
end
Find the LRB positions in the $Y$ vector;
Construct the test patterns set by permuting -1 and +1 in the LRB positions
of $Y$;
Set $\Omega = \Phi$;
foreach Vector $S_i$ in test patterns do
    $C^i$ = Hard decoding($V_i$);
    Calculate $d_i = d_E(R, C^i)$;
    if $C^i$ is a codeword then
        $\Omega = \Omega \cup C^i$;
        Calculate $d_i = d_E(R, C^i)$;
    end
end
Output: Decision codeword from $\Omega$ with lowest $d_i$
```

Algorithm 2: Second Chase Algorithm for Soft Decoding

The second Chase algorithm is summarized in Algorithm 2. The complexity of this algorithm does not depends on the length of codeword n, thus there are no more restrictions on the length of codeword. Moreover, the degradation of performance is relatively low due to the considerable reduction in the number of decodings required.

2.2.4 Iterative Decoding of Product Turbo Codes

In this section, the main principle of TPC, which is the iterative decoding of concatenated block codes, is demonstrated. It was shown is section 2.2.1 that the serial concatenation of block codes with uniform interleaving yields more asymptotic coding gain than parallel concatenation, since the former configuration leads to larger minimum hamming distance than the any configuration for the code. Moreover, it was shown that using the extended version of the primitive codes will increase the asymptotic coding gain. As a conclusion, this code obtained by serial concatenation of extended block codes with uniform interleaving makes the best choice for the block turbo codes, which is the same code proposed by Elias in 1954 [27].

For the decoding of this code, which consists of iteratively decoding the rows and the columns of the TPC matrix and repetition of the process, we presented in section 2.2.3 some of the main soft decoding algorithms for the block codes proposed by Chase, which is preferred to be used in decoding of block codes due to the asymptotic coding gain of 1.5-2.0 dB more than hard decoding. As it well known for it's counterparts, the convolutional turbo codes, for iterative decoding, the soft decoder should deliver soft output for the next decoding stage in order to attain the maximum coding gain. However, the second Chase algorithm, which makes the best available choice for soft decoding of block codes while maintaining a good performance-complexity compromise, provides a hard decision output instead of the required soft output for the iterative decoding. In order to have an effective iterative decoding, it is important to construct a SISO decoder which assigns a reliability values for it's decision. This will be the main theme of following sections. Two SISO decoding algorithms based on Chase-II algorithm will be discussed to find the decision codeword and assigns a reliability values for it's bits.

2.2.5 SISO Decoding of a Block Code using a Competing Codeword (CC)

Pyndiah *et. el.* proposed in 1994 a SISO decoding algorithm for turbo block codes [20]. This decoding algorithm is to find a decision codeword, and generating the soft output for each decision bit d_i. First a Competing Codeword (CC) "C" with minimum Euclidean distance from R such that $d_j \neq c_j$. Then the difference between the two Euclidean distances is used as a measure of reliability for the decision bit.

To demonstrate this algorithm, consider transmitting BPSK modulation of (n, k, δ) codeword $C^t = \{c_1, c_2, \cdots, c_n\}$ over a Gaussian channel. The received signal at the relay is denoted as $R = \{r_1, r_2, \cdots, r_n\}$, where r_i is defined in (2.2.11)in terms of x_i which is the BPSK modulated symbol for c_i.

Recall the LLR for the decoder's decision as defined in (2.2.25) and redefine it in terms of the observed vector R:

$$\Lambda_i = \ln\left(\frac{P(x_i = +1|R)}{P(x_i = -1|R)}\right), \ \forall\, i \tag{2.2.32}$$

where the magnitude of Λ_i provides the reliability of the decision made by decoder. The probabilities in the numerator can be expanded using the sum of probabilities of those codewords that all have $x_i = 1$ or $x_i = -1$:

$$\begin{aligned} P(x_i = +1|R) &= \sum_{C^i \in \mathcal{C}} P(x_i = +1, C^t = C^i|R) && (2.2.33) \\ &= \sum_{C^i \in \mathcal{C}} P(x_i = +1|C^t = C^i, R) \times P(C^t = C^i|R) && (2.2.34) \end{aligned}$$

the first probability in the summation can be rewritten in the form:

$$\begin{aligned} P(x_i = +1|C^t = C^i, R) &= P(x_i = +1|C^t = C^i) && (2.2.35) \\ &= \begin{cases} 1 & \text{if } c^i_j = +1 \\ 0 & \text{if } c^i_j = -1 \end{cases} && (2.2.36) \end{aligned}$$

So (2.2.34) reduces to:

$$P(x_i = +1|R) = \sum_{C^i \in \mathcal{C}^{j+}} P(C^t = C^i|R) \tag{2.2.37}$$

where $\mathcal{C}^{j+}$ is the set of codewords in the $\mathcal{C}$ space that have $+1$ in the j-th position. Using Bayes rule, this probability can be rewritten in the form:

$$P(E = C^i|R) = \frac{P(R|C^t = C^i) \times P(C^t = C^i)}{P(R)} \tag{2.2.38}$$

The same procedure can be performed to the denominator of (2.2.32) to finally get:

$$P(x_i = -1|R) = \sum_{C^i \in \mathcal{C}^{j-}} \frac{P(R|C^t = C^i) \times P(C^t = C^i)}{P(R)} \tag{2.2.39}$$

here $\mathcal{C}^{j-}$ is the set of codewords in the $\mathcal{C}$ space having -1 in the j-th position.

Since the probability of all codewords are equal for independent and identically distributed messages, i.e.:

$$P(C^t = C^i) = \frac{1}{2^k}, \ \forall\, i$$

and because the denominator part $P(R)$ is common for all the terms in (2.2.38) and (2.2.38), the LLR of in (2.2.32) can be represented in the form:

$$\Lambda_i = \ln\left(\frac{\sum\limits_{C^i \in \mathcal{C}^{j+}} P(R|C^t = C^i)}{\sum\limits_{C^i \in \mathcal{C}^{j-}} P(R|C^t = C^i)}\right) \tag{2.2.40}$$

For a Gaussian channel, (2.2.40) can be written in terms of it's elements:

$$\begin{aligned}
\Lambda_i &= \ln\left(\frac{\sum\limits_{C^i \in \mathcal{C}^{j+}} P(R|C^t = C^i)}{\sum\limits_{C^i \in \mathcal{C}^{j-}} P(R|C^t = C^i)}\right) && (2.2.41)\\
&= \ln\left(\frac{\sum\limits_{C^i \in \mathcal{C}^{j+}} \prod\limits_{l=1}^{n} P(r_l|c_l^t = c_l^i)}{\sum\limits_{C^i \in \mathcal{C}^{j-}} \prod\limits_{l=1}^{n} P(r_l|c_l^t = c_l^i)}\right) && (2.2.42)\\
&= \ln\left(\frac{\sum\limits_{C^i \in \mathcal{C}^{j+}} \prod\limits_{l=1}^{n} \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(r_l - c_l^i)}{2\sigma^2}}}{\sum\limits_{C^i \in \mathcal{C}^{j-}} \prod\limits_{l=1}^{n} \frac{1}{\sqrt{2\pi\sigma}} e^{\frac{-(r_l - c_l^i)}{2\sigma^2}}}\right) && (2.2.43)
\end{aligned}$$

eliminating the common parts, and using the definition of Euclidian distance, (2.2.43) reduces to:

$$\Lambda_i = \ln\left(\frac{\sum\limits_{C^i \in \mathcal{C}^{j+}} e^{\left(-\frac{\left\|R - C^i\right\|^2}{2\sigma^2}\right)}}{\sum\limits_{C^i \in \mathcal{C}^{j-}} e^{\left(-\frac{\|R - C^i\|^2}{2\sigma^2}\right)}}\right) \tag{2.2.44}$$

Let $D^{j+} = \{d_1^{j+}, d_2^{j+}, \cdots, d_n^{j+}\}$ be the codeword in $\mathcal{C}^{j+}$ with minimum Euclidean distance from R, and let $D^{j-} = \{d_1^{j-}, d_2^{j-}, \cdots, d_n^{j-}\}$ be the codeword in $\mathcal{C}^{j-}$ with minimum Euclidean distance from R. Isolating D^{j+} and D^{j-} in the numerator and in the denominator the LLR can be rewritten as:

$$\Lambda_j = \frac{1}{2\sigma^2}\left(\left\|R - D^{j-}\right\|^2 - \left\|R - D^{j+}\right\|^2\right) + \ln\left(\frac{\sum\limits_i A_i}{\sum\limits_i B_i}\right) \tag{2.2.45}$$

where

$$A_i = \exp\left(\frac{\left\|R - D^{j+}\right\|^2 - \left\|R - C^i\right\|^2}{2\sigma^2}\right) \leq 1; \text{ with } C^i \in \mathcal{C}^{j+} \tag{2.2.46}$$

and

$$B_i = \exp\left(\frac{\left\|R - D^{j-}\right\|^2 - \left\|R - C^i\right\|^2}{2\sigma^2}\right) \leq 1; \text{ with } C^i \in \mathcal{C}^{j-} \tag{2.2.47}$$

the inequality in (2.2.46) and (2.2.47) follows from the fact that the argument of the exponential function are always negative since the two quantities $\|R - D^{j+}\|$ and $\|R - D^{j-}\|$ are defined to be the smallest for all $C^i \in \mathcal{C}^{j+}$ and $C^i \in \mathcal{C}^{j-}$, respectively.

Assuming that the codewords are distributed uniformly in the $\mathcal{C}$ space, the two quantities $\sum\limits_i A_i$ and $\sum\limits_i B_i$ will be almost equal and the ratio in (2.2.45) becomes:

$$\ln\left(\frac{\sum\limits_i A_i}{\sum\limits_i B_i}\right) \approx 0$$

Therefore, the LLR for the output of the decoder at bit j can be expressed as:

$$\begin{aligned}
\Lambda_j &= \frac{1}{2\sigma^2}\left(\left\|R - D^{j-}\right\|^2 - \left\|R - D^{j+}\right\|^2\right) && (2.2.48)\\
&= \frac{1}{2\sigma^2}\left(\sum_{l=1}^{n}\left(r_l - d_l^{j-}\right)^2 - \sum_{l=1}^{n}\left(r_l - d_l^{j+}\right)^2\right) && (2.2.49)\\
&= \frac{1}{2\sigma^2}\left(\sum_{l=1}^{n}\left(r_l - d_l^{j-}\right)^2 - \left(r_l - d_l^{j+}\right)^2\right) && (2.2.50)\\
&= \frac{1}{2\sigma^2}\left(\sum_{l=1}^{n}\left(-2r_l d_l^{j-}\right) - \left(-2r_l d_l^{j+}\right)\right) && (2.2.51)
\end{aligned}$$

taking r_j out of the summation (note that $d_j^{j-} = -1$ and $d_j^{j+} = +1$), we get:

$$\Lambda_j = \frac{2}{\sigma^2}\left(r_j + \sum_{l=1, l\neq j}^{n} \left(r_l d_l^{j+} p_l \right) \right) \tag{2.2.52}$$

where

$$p_l = \begin{cases} 0 & \text{if } d_l^{j+} = d_l^{j-} \\ 1 & \text{if } d_l^{j+} \neq d_l^{j-} \end{cases} \tag{2.2.53}$$

normalizing (2.2.52) with respect to $2/\sigma^2$, we get:

$$\begin{aligned} \frac{\sigma^2}{2}\Lambda_j &= r_j + \sum_{l=1, l\neq j}^{n} \left(r_l d_l^{j+} p_l \right) && (2.2.54) \\ &= r_j + w_j && (2.2.55) \\ &= r'_j && (2.2.56) \end{aligned}$$

where r' is the final soft output of the SISO decoder that is used for the following decoding stage as input. w_j in (2.2.54) is called the extrinsic information which is the information collected about the bit from the neighboring bits and does not depend on r_j. Like convolutional turbo codes, extrinsic information play very important rule in turbo decoding of block codes since it is used to deliver reliability information about the decoded bit to the decoder.

The value of w_j in (2.2.55) depends on the two codewords D^{j+} and D^{j-} with a minimum Euclidean distance from vector R with +1 and -1 in the j-th position, respectively. The term w_j have information on the sign of the transmitted symbol c_j^T that are contained in the other transmitted symbols of the codeword because of the correlation between these symbols introduced by coding.

The above derivation of the soft output information assumes that the decoder can find the codeword with minimum Euclidean distance from R as appears in equations (2.2.45), (2.2.48), etc. However, as discussed earlier in section 2.2.3, finding the optimal decision with the minimum Euclidean distance from the observation vector is a complex procedure and requires huge computations. Luckily, this procedure can be simplified by using one of Chase algorithms that discussed earlier in section 2.2.3 to generate a set of codewords Ω having minimum distance from R.

The decision codeword D can be found from this set as discussed in the Algorithms 1 and 2. However, to find the reliability of the bit output bit j in equation (2.2.48), two codewords D^{j+} and D^{j-} are required and Chase algorithm provides only one decision codeword. The algorithm proposed by Pyndiah suggests to use the decision D found by Chase-II algorithm as either D^{j+} or D^{j-}, depending on the value of the j-th bit. Assuming that there exist another codeword C in Ω having minimum Euclidean distance from R such that $c_j \neq d_j$, this codeword will serve as the second codeword of D^{j+} or D^{j-} beside D. By reusing (2.2.48) with normalization by $\sigma^2/2$ and substituting C and D for D^{j-} and D^{j+} we obtain:

$$r'_j = \left(\frac{\|R - C\|^2 - \|R - D\|^2}{4} \right) \times d_j \tag{2.2.57}$$

note that the output takes the sign of Chase decision d_j and the magnitude of the output depends exclusively on the difference between the two Euclidean distances between R and the two competing codewords. Since D is defined to have to minimum Euclidean distance then the difference in the numerator is always positive.

Note that when $\|R - C\|^2 \to \|R - D\|^2$, then the two codewords C and D tend to have equal probability, meaning that the reliability of the decoders decision about d_j is close to zero. On the other hand, when the difference between those two metrics become larger, the reliability value of the decision bit d_j become larger.

The assumption that there exists a codeword in Ω necessitate the generation of a very large set of codewords in Ω which increases the decoding complexity. The algorithm proposes to use a fixed number of test patterns (which determines the size of Ω), and to use an empirical relation when the competing codeword does not exist in Ω. This empirical relation as proposed by Pyndiah *et. el.* is:

$$r'_j \;=\; \beta \times d_j. \tag{2.2.58}$$

This assumption is based on the fact that when C does not exist in the search zone defined by Chase algorithm, this means that the codeword C is relatively distant from R and the Euclidean distance $\|R - C\|^2$ is relatively large compared to $\|R - D\|^2$, hence the decision is relatively reliable. The choice for the value of β is not easy, since an erroneous decision could propagate errors to the rest of bits in the decoding if a high β value is assigned which corresponds to a high reliability of the error bit. A small value for a correct decision, on the other hand, will slow the convergence to the correct codeword. Pyndiah *et. el.* proposed to use the probability of errors at the SISO output as factor to determine the value of β [34]:

$$\beta \propto \ln\left(\frac{P(d_j = c_j^t)}{P(d_j \neq c_j^t)}\right). \tag{2.2.59}$$

By using this relation, the complexity of the decoding will be significantly reduced due to the reduction of the number of required decodings and reduction in the search zone.

Iterative decoding process of serially concatenated block codes is composed of rows decoding and columns decoding. Each decoding is performed using SISO decoder on the observed vector in addition to the the extrinsic information provided by the previous decoding stage. This process is repeated several times until fixed number of iterations. A general one decoding stage including a SISO decoder structure is presented in Fig. 2.10. The SISO decoder takes single input and returns single output. The input is a summation of the extrinsic information matrix $\mathbf{W}[m]$ (*a-priori* input) from the previous decoding stage $(m-1)$ with the normalized observation matrix $\mathbf{R}$, where m is the decoding stage number. For the first decoding stage $\mathbf{W}[m]$ is initialized to zero. The output of the SISO decoder is the new extrinsic information obtained as the difference between the normalized LLR of the decoded bit and soft input as given in (2.2.55).

A single decoding stage (across the columns or the rows) that is used to produce

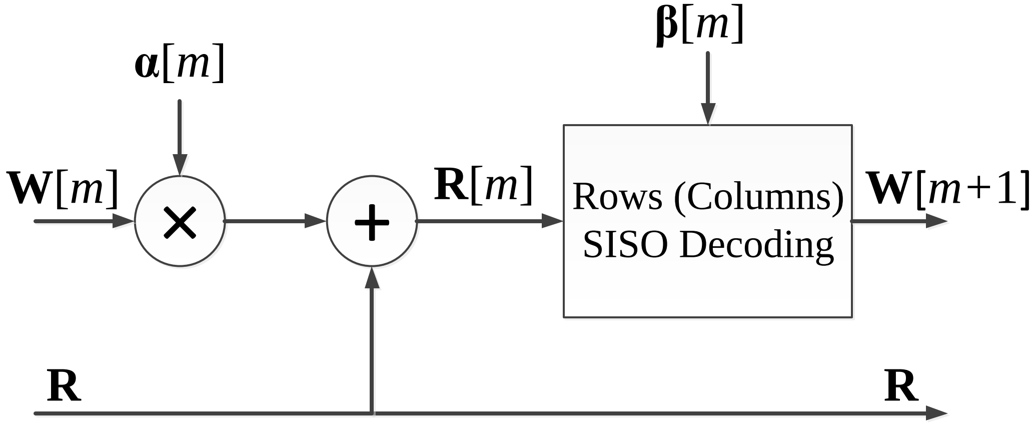


Figure 2.10. Single stage SISO decoder based on competing codeword CC algorithm

the soft output in the iterative decoding process of a product code is illustrated in Fig. 2.10. The input matrix for the SISO decoder is given by:

$$\mathbf{R}[m] = \mathbf{R} + \alpha[m]\mathbf{W}[m] \qquad (2.2.60)$$

where multiplicative factor $\alpha[m]$ are constants depend on the type of channel . Although the two random variables $\mathbf{W}[m]$ and $\mathbf{R}$ have the same distribution (elements of $\mathbf{W}[m]$ are combination of different observations from $\mathbf{R}$), they have different average and different standard deviation, therefore, $\alpha[m]$ are used to adjusts the level of $\mathbf{W}[m]$ to the level of $\mathbf{R}$ after each iteration. The elements of $\alpha[m]$ are optimized by successive approximation. For the first iterations, the elements of $\alpha[m]$ are chosen very close to zero since the BER at the output of the decoder is relatively high and therefore the output extrinsic information have low reliability. On the other hand, the coefficients of $\alpha[m]$ reaches values close to one for the last iterations since the BER of the decoding process decreases in general and therefore the reliability of the decisions increases. The coefficients of $\beta[m]$ are chosen in a similar manner, where they take small values close to zero for the first iterations and increase with m. The values of $\alpha[m]$ and $\beta[m]$ are optimized according to the code used and the transmission channel. In his paper, Pyndiah, proposed a method to reduce the dependency between the two parameters that reduces the optimization complexity [35].

2.2.6 SISO Decoding Based on the Destructive Euclidean Distance

The decoding algorithm discussed in the previous section is suitable for high code rates, and becomes inefficient when the Hamming distance of the code increases. List based algorithms which uses Ordered Statistics Decoding (OSD) algorithm for decoding are proposed in [36–38] to decode BTC with high code rates. These algorithms orders the soft inputs and produce a subset of all possible k errors patterns which are encoded to obtain a large number of codewords. The performance of such algorithms is very close to optimal maximum likelihood decoder at the cost of high computational complexity.

To reduce the computational complexity, an alternate solution was proposed by Le *et. el.* in [39]. Unlike the list based SISO decoding algorithms, the approach in [39] uses the Destructive Euclidean Distance (DED) for obtaining the soft information for the decision codeword. The DED is used as a measure of reliability of the decision obtain by Chase algorithms.

This algorithm based on the DED has less complexity than the algorithm proposed by Pyndiah *et. el.* since it does not require the search for competing codeword. The decoding is performed in two steps: First, a decision codeword $D = \{d_1, d_2, \cdots, d_n\}$ is obtained by using Chase II algorithm, which is explained in section 2.2.3. Second, the soft output are calculated using the DED between the decision D and the observation vector R as a measure of reliability.

It was shown in (2.2.48) that the reliability of a decision bit d_j is given by the magnitude of the LLR of the decision as:

$$\Lambda_j = \frac{1}{2\sigma^2}\left(\|R - C\|^2 - \|R - D\|^2\right) \qquad (2.2.61)$$

where $C = \{c_1, c_2, \cdots, c_n\}$ with $c_j \in \{-1, +1\}$ is a competing codeword such that

$d_j \neq c_j$ and minimum Euclidean distance from R. This Equation is normalized and from it the normalized extrinsic information w_j is obtained by

$$w_j = \frac{\sigma^2}{2}\Lambda(d_j) - r_j \tag{2.2.62}$$

In the algorithm based on the DED the LLR is calculated based on the destructive distance of D unlike using the CC as in (2.2.61). From the DED of the decision a confidence value is calculated for the codeword which is used to derive the soft information about the decision bit. The confidence value is defined as the probability that the decoder makes correct decision given received sequence R, which can be written as

$$\phi = P(D = X|R) \tag{2.2.63}$$

where X is the BPSK transmitted sequence.

The measure of confidence, i.e. ϕ can be estimated by the decoder in terms of the Euclidean distance d_E. However, this would only work for the first iteration when there are no extrinsic information added to the observation vector R. On other words, for this to work, the side effects of the extrinsic information have to be removed from the calculation of d_E in the following iterations. Consider the Euclidean distance at the j-th bit position $d_{E_j} = (r_j + w_j - d_j)^2$, when $(r_j - d_j)\dot{w}_j < 0$ we say that the extrinsic information have a positive effect on the Euclidean distance, i.e. decreasing d_{E_j}. On the other hand, w_j has negative effect on the Euclidean distance when $(r_j - d_j)\dot{w}_j > 0$ since it increases the Euclidean distance for this bit position. The latter case, when w_j has negative effect on d_{E_j}, is referred to the side effects on the extrinsic information on the Euclidean distance and it should not be removed from the estimation of the confidence value of the decision. Thus, for the estimation of ϕ, this algorithm suggests the use of the DED rather than the Euclidean distance so that only negative effect of w_j is considered on calculations.

d_{DED}	<9	9	10	11	12	13	14	> 14
ϕ	0.99	0.93	0.9	0.82	0.82	0.42	0.21	0

Table 2.1. Look-up table for ϕ as a function of d_{DED}.

The destructive Euclidean distance is defined as the sum of Euclidean distance for the bit positions where the observed bit has a different sign than the decision bit, i.e. if we denote DED as d_{DED}, then

$$d_{DED} = \sum_{j \in \{j|(r_j - d_j).d_j < 0\}} (r_j - d_j)^2. \tag{2.2.64}$$

Note that d_j replace w_j in the calculation of d_{DED} since they have the same sign most of time.

To obtain the relationship between the DED and the confidence value ϕ the authors use software simulation. The simulation generates 10000 samples of decoding results from BTC with parameters $(64, 51, 6)^2$ as an example. The confidence value is estimated using (2.2.63), while the DED is calculated from (2.2.64). Since ϕ may depend on the iteration number, E_b/N_0 and the number of LRB p, different cases were plotted in Fig. 2.11. Since all the cases in Fig. 2.11 are similar, the effect of the iteration number, E_b/N_0 and p can be neglected, and therefore the confidence value can be represented as only a function of d_{DED}:

$$\phi \;\; = \;\; \mathrm{f}(d_{DED}) \tag{2.2.65}$$

this would also reduce the implementation complexity. The curve which corresponds to the best performance is selected from 2.11 and a discrete look-up table 2.1 is used instead of continuous representation. Consequently, the computational complexity is significantly reduced with the pre-assigned look-up table.

To compute the soft output for the decision codeword D found by Chase algorithm, it is essential to relate the confidence value to the log-likelihood ratio of the decision

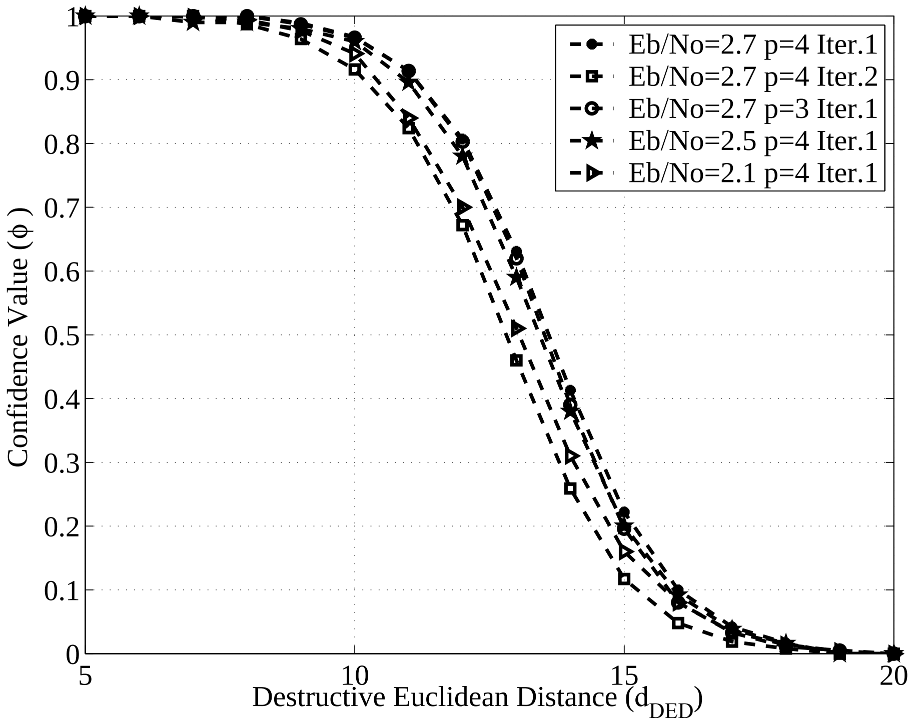


Figure 2.11. Simulation results for the confidence value ϕ versus the the destructive Euclidean distance DED d_{DED}

bits. The LLR of the decision bit d_j is given by

$$\Lambda(d_j) = \ln \frac{P(x_j = +1|R)}{P(x_j = -1|R)}. \tag{2.2.66}$$

Assuming the probability that $x_j \in X$ with confidence ϕ given that the codeword X has been transmitted, then the numerator and denominator of the relation in (2.2.66), which represent the probability of x_j given R was received, can be written as

$$P(x_j = \pm 1|R) = P(x_j = \pm 1, D = X|R) + P(x_j = \pm 1, D \neq X|R). \tag{2.2.67}$$

Applying Bayes' rule to the first term of (2.2.67), which corresponds to the decoder making correct decision, yields

$$\begin{aligned} P(x_j = \pm 1, D = X|R) &= P(x_j = 1|D = X, R) \cdot P(D = X|R) && (2.2.68) \\ &= P(x_j = 1|D = X, R) \cdot \phi && (2.2.69) \\ &= \begin{cases} \phi & \text{if } \ d_j = x_j \\ 0 & \text{if } \ d_j \neq x_j \end{cases} && (2.2.70) \end{aligned}$$

where (2.2.70) follows since the decision bit d_j is know.

For the second term in (2.2.67), which corresponds to the probability of x_j when the decoder makes wrong decision, Bayes' rule is also applied to obtain

$$\begin{aligned} P(x_j = \pm 1, D \neq X|R) &= P(x_j = \pm 1|D \neq X, R) \cdot P(D \neq X|R) && (2.2.71) \\ &= \frac{\exp(\pm 2r_i/\sigma^2)}{1 + \exp(\pm 2y_i/\sigma^2)} \cdot (1 - \phi), && (2.2.72) \end{aligned}$$

where in (2.2.72) it is assumed that the x_j was transmitted in a Gaussian noise channel.

Rewriting (2.2.67) by combining the results in (2.2.70) and (2.2.72) yields

$$P(x_j = \pm 1|R) = \begin{cases} \phi + (1 - \phi)\frac{\exp(\pm 2r_j/\sigma^2)}{1+\exp(\pm 2r_j/\sigma^2)} & \text{if } \ d_j = x_j \\ (1 - \phi)\frac{\exp(\pm 2r_j/\sigma^2)}{1+\exp(\pm 2r_j/\sigma^2)} & \text{if } \ d_j \neq x_j \end{cases} \qquad (2.2.73)$$

To get the numerator and denominator results, (2.2.73) is expanded to

$$P(x_j = +1|R) = \begin{cases} \phi + (1 - \phi)\frac{\exp(+2r_j/\sigma^2)}{1+\exp(+2r_j/\sigma^2)} & \text{if } \ d_j = +1 \\ (1 - \phi)\frac{\exp(+2r_j/\sigma^2)}{1+\exp(+2r_j/\sigma^2)} & \text{if } \ d_j = -1 \end{cases} \qquad (2.2.74)$$

and

$$P(x_j = -1|R) = \begin{cases} (1 - \Phi)\frac{\exp(-2r_j/\sigma^2)}{1+\exp(-2r_j/\sigma^2)} & \text{if } \ d_j = +1 \\ \Phi + (1 - \Phi)\frac{\exp(-2r_j/\sigma^2)}{1+\exp(-2r_j/\sigma^2)} & \text{if } \ d_j = -1 \end{cases} \qquad (2.2.75)$$

Finally, the two results in (2.2.74) and (2.2.75) are substituted in the LLR equation (2.2.66) for the decoded output bit d_j to get

$$\Lambda(d_j) = \ln\left(\frac{\phi + \exp(2r_j d_j/\sigma^2)}{1 - \phi}\right) \qquad (2.2.76)$$

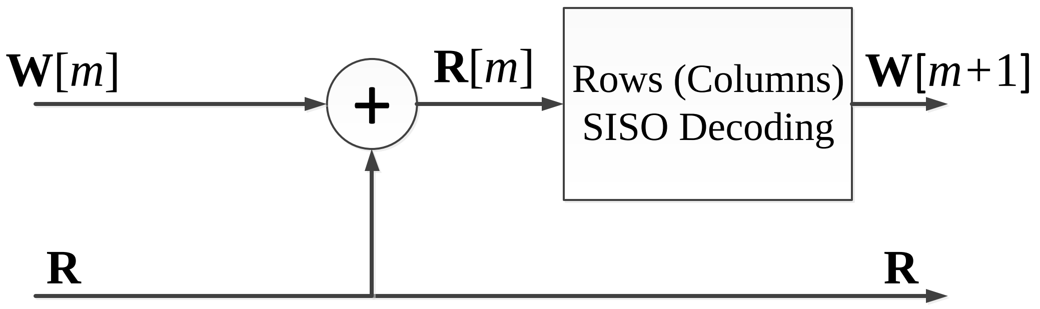


Figure 2.12. Single stage SISO decoder based on DED algorithm

Then the extrinsic output for this decoding stage can be obtained as [39]:

$$w_j = \frac{\sigma^2}{2}\Lambda(d_j) - r_j \tag{2.2.77}$$

$$= d_j\left(\frac{\sigma^2}{2}\ln\left(\frac{\phi + \exp(2r_jd_j/\sigma^2)}{1-\phi}\right) - r_jd_j\right) \tag{2.2.78}$$

where w_j is the final soft output for the j-th bit which is the normalized log extrinsic information output.

The block diagram for the SISO decoder based on the DED decoding algorithm is shown in Fig. 2.12. Unlike finding CC presented in the previous section, there is no need for scaling by weighting factors $\alpha[\mathbf{m}]$ and $\beta[\mathbf{m}]$ for each decoding stage. For comparison, the performance of the two algorithms, namely the the soft decoding based on the CC algorithm and for the soft decoding based on the DED algorithm, are shown in Fig. 2.13. The BER performance shown is for the TPC with parameters $(64, 51, 6)^2$ after 4 iterations with BPSK signaling in Gaussian channel.

For implementation purposes, Le *et. el.* proposed to pre-calculate the soft output for w_j which is a function of only r_j, d_j and d_{DED}, then store the results in a look-up table indexed with the quantized values for these three variables for faster processing.

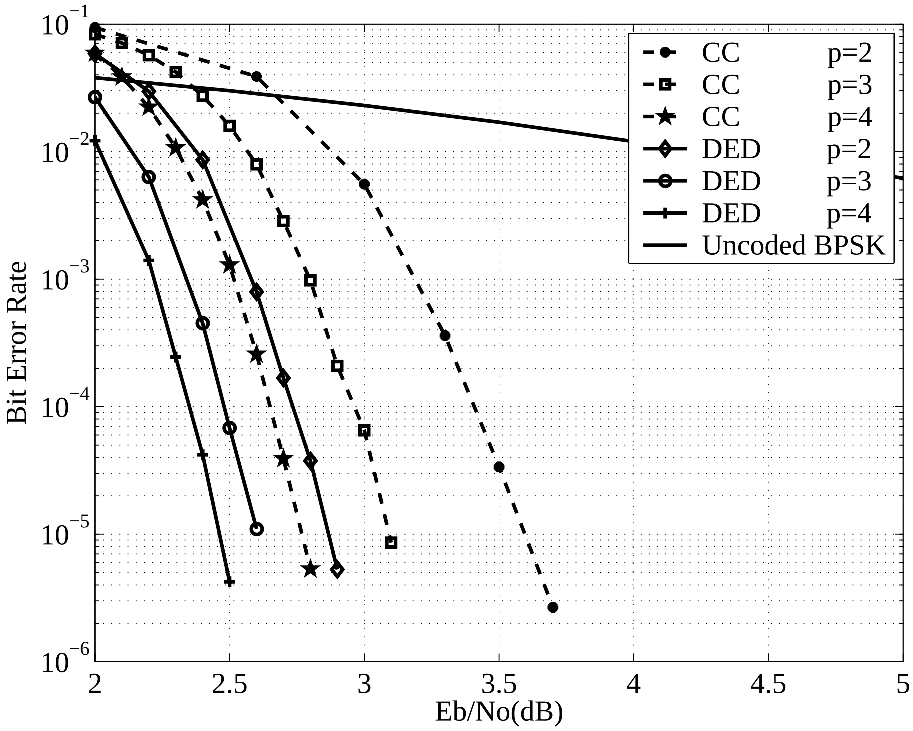


Figure 2.13. BER performance comparison between CC and distance based decoding DBD algorithms plotted versus E_b/N_0 for TPC $(64, 51, 6)^2$ with binary phase shift keying BPSK signaling in a Gaussian channel after 4 iterations

Chapter 3

Distributed Turbo Coding with Hard Detection

3.1 Introduction

As mentioned in section 2.1.2, the coded cooperation techniques can improve the overall system capacity for proper setup of the relay position compared to the original non-cooperative system. In the coded cooperation protocol, in which the relay forwards an incremental redundancy to the destination about the recovered message from the source's transmission, the destination uses the two parts of the code received about the source transmission via the direct path and the relay channel to conduct channel decoding.

Coded cooperation can be implemented on a wide variety of channel codes, convolutional codes, Rate Compatible Punctured Convolutional (RCPC), LDPC, Turbo codes, etc. Moreover, the configuration of the cooperative network, the forwarding protocol at the relay and the construction of the distributed code can take many different forms depending on the structure of the constituting code and the decoding method at the destination or the practical application intended. Properly designed distributed coding can effectively approach the capacity of cooperative wireless networks.

Most of the previously presented coded cooperation strategies are formulated for the classical three-node relay channel model, i.e., the transmitter-receiver network with only one relay. For these systems, the relay should be pre-assigned and positioned between the source and the destination such that the interuser and relay channels exhibit sufficiently good channel conditions as required for the reliable decoding at the relay and the destination. However, diversity gain theoretically increases with the number of relay nodes [21]. The low device costs associated with these networks (such as wireless sensors) allows the coverage area to be covered with dense deployment of devices. The performance of these densely deployed wireless networks can be improved by exploiting the spatial diversity due to the presence of multiple devices in the area between any source-destination pair [40].

In this chapter, the implementation of distributed product turbo codes on coded cooperation is studied and the performance of this system investigated and compared to the non-cooperative mode for TPC which involves direct transmission from the source to the destination. We apply the concept of distributed encoding for the source's message over multiple relay nodes and use a modified iterative turbo product decoding at the destination to decode the received distributed TPC over multiple channels. We investigate the performance of distributed TPC in AWGN channel using simple network topologies. The data source broadcast BCH encoded codewords to the destination. The preassigned relays in the network, residing in a collaborative region, detect and decode the received block of codes transmitted from the source. In the second time slot, The relays from the collaboration nodes transmit incremental redundancy by BCH encoding the corrected block codewords vertically and transmitting the generated parity block. A process of cyclically interleaving the decoded data is proposed so that each relay from the collaborative nodes can produce dissimilar vertical parity which are still on the same code space. The receiver then uses the bits received from the source and the relays to conduct a modified turbo product decoding process to cope with Multiple Vertical Parities (MVP).

3.2 System Model

We assume that the source and the relay employ very simple code for their input data, a block BCH code with (n, k, δ) encoder, which appends $n-k$ parity bits to the input block bits. The user "source" transmits the BCH encoded block of codewords in the first time slot to a specific destination and a relay in broadcast mode. Subsequently, the relay detects and tries to correct the received vector of bits and then encodes the decoded block vertically by appending parity bits to the columns of the decoded block. In the second time slot, the destination receives the estimated columns parity bits generated by the relay for the BCH encoded blocks received in the first time slot.

The code bits are assumed to be BPSK modulated before transmission. A codeword of a (n, k, δ) linear block code is transmitted from node i as $\mathbf{X}_i = \{x_i^1, x_i^2, \cdots, x_i^n\}$, where $x_i^l \in \{-1, +1\}, l = 1, 2, \cdots, n$, over AWGN channel. In this chapter, we use boldface capital letters $\mathbf{M}$ to denote matrices and the $[\cdot]^T$ denotes the matrix transpose operation. Denote the received vector at node j as $\mathbf{Y}_j = \{y_j^1, y_j^2, \cdots, y_j^n\}$. The received signals at the destination and the relay can be expressed as:

$$\mathbf{Y}_d[2k-1] = \alpha_{sd}[2k-1]\mathbf{X}_s[2k-1] + \mathbf{Z}_{sd}[2k-1] \quad (3.2.1)$$

$$\mathbf{Y}_r[2k-1] = \alpha_{sr}[2k-1]\mathbf{X}_s[2k-1] + \mathbf{Z}_{sr}[2k-1] \quad (3.2.2)$$

$$\mathbf{Y}_d[2k] = \alpha_{rd}[2k]\mathbf{X}_r[2k] + \mathbf{Z}_{rd}[2k] \quad (3.2.3)$$

where $k \in \{1, 2, \cdots\}$ represents the time slot; the source transmits at odd time slots and the relay transmits at even time slots. α_{ij} is the channel block fading coefficient. $\mathbf{Z}_{ij} = \{z_{ij}^1, z_{ij}^2, \cdots, z_{ij}^n\}$ are zero mean i.i.d AWGN on the channel between the nodes i and j, $i \in \{s, r\}$, $j \in \{r, d\}$

3.2.1 Turbo Product Codes (TPC)

The basic concept of TPC is to iteratively decode across rows and columns of the product codes using SISO decoders and passing the soft information from one decoding stage to the next. Product codes which were introduced by Elias [27] are obtained by the serial concatenation of two systematic linear block codes $\mathcal{C}^1$ and $\mathcal{C}^2$ with parameters (n_1, k_1, δ_1) and (n_2, k_2, δ_2), respectively, where n_i, k_i, and δ_i stand for code length, code dimension and minimum Hamming distance of the code, respectively. Data bits are placed in a $k_1 \times k_2$ matrix, then rows are encoded by $\mathcal{C}^1$ code to produce horizontal parity P_h. Columns of the matrix (including the columns of P_h) are then encoded by $\mathcal{C}^2$ code to produce vertical parity P_v. The basic constituent blocks of the $n_1 \times n_2$ matrix resulted from TPC encoding are shown in Fig.3.1. The blocks S, P_h and P_v refer to the systematic, horizontal parity and vertical parity blocks respectively. The parameters of the resultant product code (n, k, δ) are given by: $n = n_1 \cdot n_2$, $k = k_1 \cdot k_2$ and $\delta = \delta_1 \cdot \delta_2$ and the code rate is given by $R = R_1 \cdot R_2$ where R_i is the code rate of code $\mathcal{C}^i$. We assume here that the two component codes have identical parameters. For the simulations conducted using this system model, we use the $(64, 51, 6)$ extended BCH code as a component code in the TPC.

The extended BCH code is obtained by adding the overall parity check to expand the minimum hamming distance from δ to $\delta + 1$ [41]. We denote this code as a TPC $(n, k, \delta)^2$. The primary advantage of BCH codes is the ease with which they can be decoded via an algebraic method known as syndrome decoding. Very simple electronic hardware is required to perform the task, meaning that a decoding device may be made small and low-powered, making it a perfect choice for applications including wireless sensor networks. As a class of codes, BCH codes are also highly flexible, with the ability to control over block length and acceptable error thresholds, meaning that a custom code can be designed to a specific channel conditions [42]. Another unique

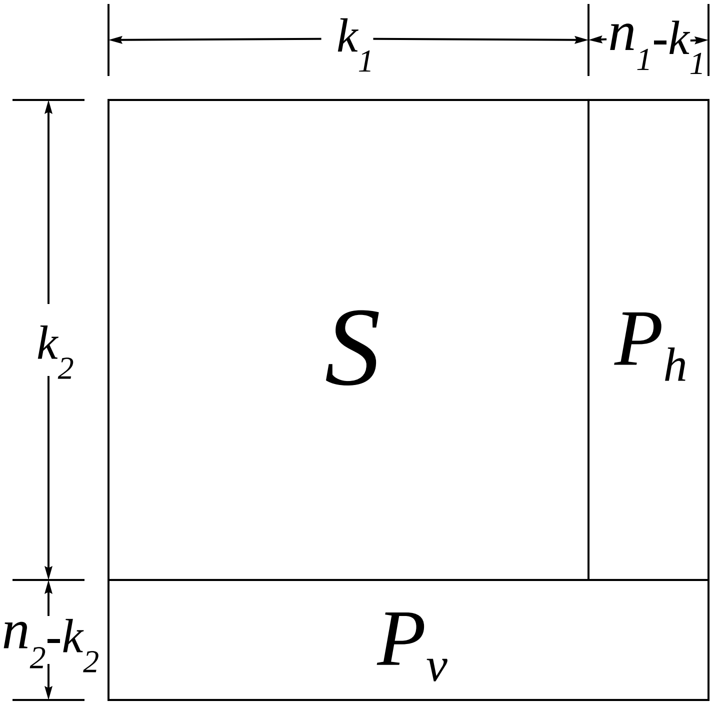


Figure 3.1. Structure of a Turbo Product Code

property of BCH codes is the inherent error detection capability which can be of great importance in cooperative communication.

3.2.2 Simulation Network Model

To simplify the simulation model, we assume different scenarios for the location of the intermediate relays. In the first scenario, we assume that the relays are located on

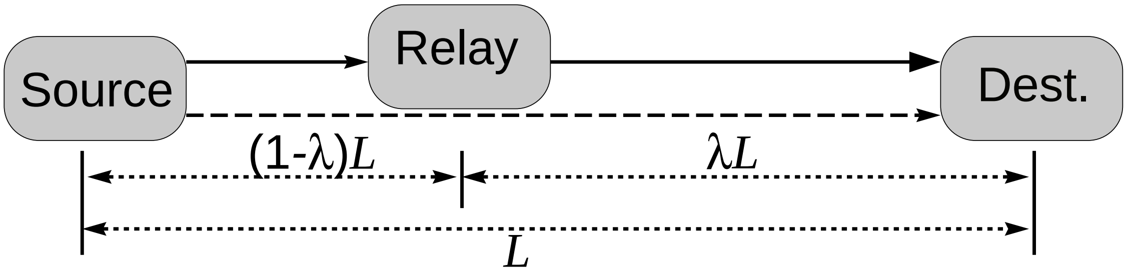


Figure 3.2. The simplified three terminals line network topology

the line connecting the two ends [9, 43–46]. Fig. 3.2 shows the line network topology used in testing the proposed DTPC, where $0 \leq \lambda \leq 1$ represents the position of the relay between the source and the destination.

Given free space propagation model defined as [47]:

$$P_r = \frac{P_t G_t G_r c^2}{(4\pi L f)^2},$$

where c, f, P_t,G_t,G_r and L are the speed of light, frequency, transmission power, transmission antenna gain, receiving antenna gain and the distance, respectively, and assuming that the transmit power from the source and the relay are equal, we obtain the values of SNR_{sr} and SNR_{rd} with respect to the reference value SNR_{sd} as:

$$SNR_{sr} = \frac{SNR_{sd}}{(1-\lambda)^2} \tag{3.2.4}$$

$$SNR_{rd} = \frac{SNR_{sd}}{\lambda^2} \tag{3.2.5}$$

where SNR_{ij} refers to the SNR at receiver j for a transmission from transmitter i.

From (3.2.4) and (3.2.5), the variances of the $z_{ij}^l, l = 1, 2, \cdots, n$ i.i.d AWGN in (3.2.2) and (3.2.3) can be expressed as:

$$\begin{aligned} \sigma_{sr}^2 &= (1-\lambda)^2 \sigma_{sd}^2 \\ \sigma_{rd}^2 &= \lambda^2 \sigma_{sd}^2 \end{aligned}$$

where $\mathrm{var}(z_{sd}^l) = \sigma_{sd}^2$, $\mathrm{var}(z_{rd}^l) = \sigma_{rd}^2$ and $\mathrm{var}(z_{sr}^l) = \sigma_{sr}^2$.

We assume in the second scenario that the source and relay nodes are located in geographically small region forming a transmit cluster, and thus the quality of the channels from the source to relays is a few dBs better than the direct link channel. This network model is used to simulate the case when the direct channel is experiencing shadowing effect. In this scenario, the relay nodes are assumed to be assigned from the available intermediate relay pool according to a minimum value of SNR of the interuser channel (the channel connecting the source and the relay). This network

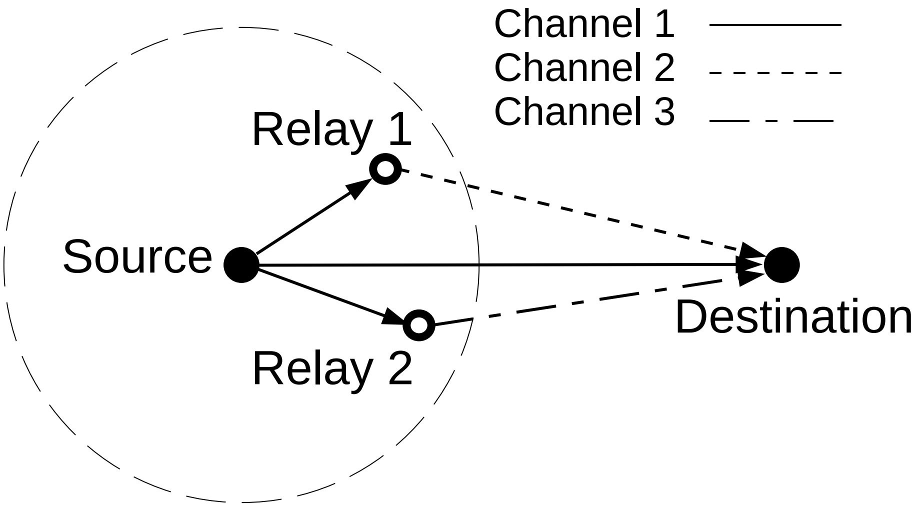


Figure 3.3. The relays aid the source by forming a transmit cluster

scenario is depicted in Fig. 3.3 which shows that the source and the relays form a transmit cluster in which the interuser channels have higher SNR compared to the direct link channel. The condition on minimum acceptable SNR for the interuser channel is shown as a circle surrounding the source. In this chapter, we evaluate the performance of the DTPC when interuser channel is experiencing different SNRs.

3.3 Distributed Product Turbo Codes DTPC

The information bits are formed into a $k \times k$ block S and then encoded with a (n, k, δ) BCH code to produce rows parity P_h. The output $k \times n$ block of bits is broadcasted from the source. The relay decode the received sequences with a simple BCH decoding algorithm (e.g. Berlekamp-Massey [48]), then it encodes the estimated blocks $\hat{S}, \hat{P}_h$ along the columns and produces columns parity $\hat{P}_v$. The latter is transmitted to the destination in the second time slot. The three blocks S, P_h and $\hat{P}_v$ are received at the destination over two time slots (not necessarily of equal length); S and P_h blocks are received from the direct link between the source and the destination, while the

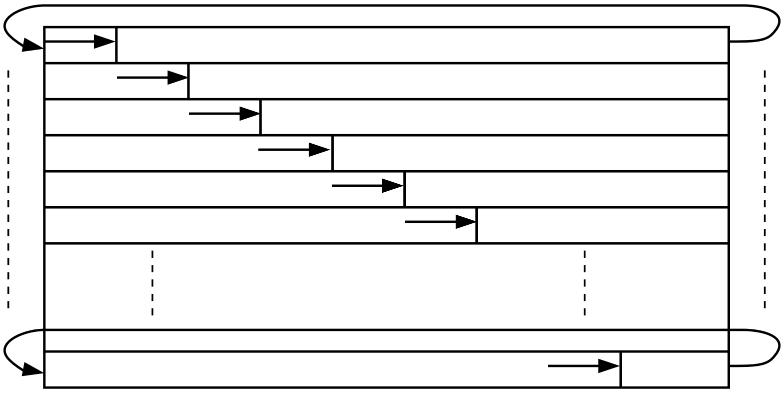


Figure 3.4. Cyclically interleaving of $\mathcal{C}^1$ codewords matrix

block $\hat{P}_v$ is received from the relay channel.

3.3.1 Generating Multiple Vertical Parities (MVP)

As described in section 3.2, and as shown in Fig. 3.1, the vertical parity is obtained by encoding the rows of $\mathcal{C}^1$ codewords $(k \times n)$ matrix along the columns with $\mathcal{C}^2$ space. The product code produced by this method is called "complete product code", and the resultant new vertical parity rows $((n-k) \times k)$are also codewords in $\mathcal{C}^1$ space [41], due to the linear property of the constituent codes. However, if the $(k \times n)$ matrix is randomly interleaved before being vertically encoded using $\mathcal{C}^2$, the resulting new rows are not in the $\mathcal{C}^1$ space. The latter is usually referred to as "incomplete turbo product code".

In this section, we propose an interleaver to generate MVP from horizontally encoded rows to produce multiple complete TPC. The proposed interleaver is based on the linear and cyclic prosperities of the constituent codes$\mathcal{C}^1$ and $\mathcal{C}^2$. The cyclic property of the code implies that the codeword can be cyclically rotated and the result is another codeword in the same code space. Suppose that $c_k \in \mathcal{C}^1$ and $c_k = [b_1 b_2 \cdots b_n]$, for any rotation of the code by l bits will result in $[b_{n-l+1} b_{n-l+2} \cdots b_n b_1 b_2 \cdots b_{n-l}]$ which is also a codeword in $\mathcal{C}^1$.

To obtain different vertical parities of a complete TPC from the same horizontally encoded rows, we use different rotation shift l for each row of the k rows as illustrated in Fig. 3.4. The result of interleaving is a different $(k \times n)$ matrix with rows in the $\mathcal{C}^1$ space. The resulted cyclically interleaved matrix can be reused to produce different vertical parity which will be used in the decoding process of the TPC. The code rate of the new TPC is:

$$\frac{k^2}{n^2 + M(n-k)n}$$

where M is the number of additional vertical parity obtained for the $(k \times n)$ matrix.

When the second relay receives the broadcast of the source of S and P_h, it decodes the received vectors using simple decoding algorithm, then cyclically interleaves the resulted $(k \times n)$ matrix. The relay encodes the columns of the resulting block using the code $\mathcal{C}^2$. Then generated parity is transmitted to the destination.

3.3.2 Decoding at the Destination

For each generated vertical parity there are two decoding stages at the destination, first through rows and then through columns. The basic component of the turbo product decoder is the SISO decoder used to decode the rows and columns. First, the SISO decoder uses Chase II algorithm [33] which searches for the p LRB in the received vector and creates 2^p test patterns by permuting with '0' and '1' in the p LRB positions. The decoding complexity is reduced by considering only the 2^p most probable code words of all the codewords. A decision codeword $D = \{d_1, d_2, \ldots, d_n\}$ with $d_j \in \{-1, +1\}$ is chosen from the list with the minimum Euclidean distance from the received vector $\mathbf{Y}$.

Once a decision codeword D is found, its confidence value ϕ will be evaluated. The confidence value is defined as the probability that the decoder makes a correct decision given received sequence $\mathbf{Y}$. The value ϕ is defined in (3.3.1) as a function of destructive

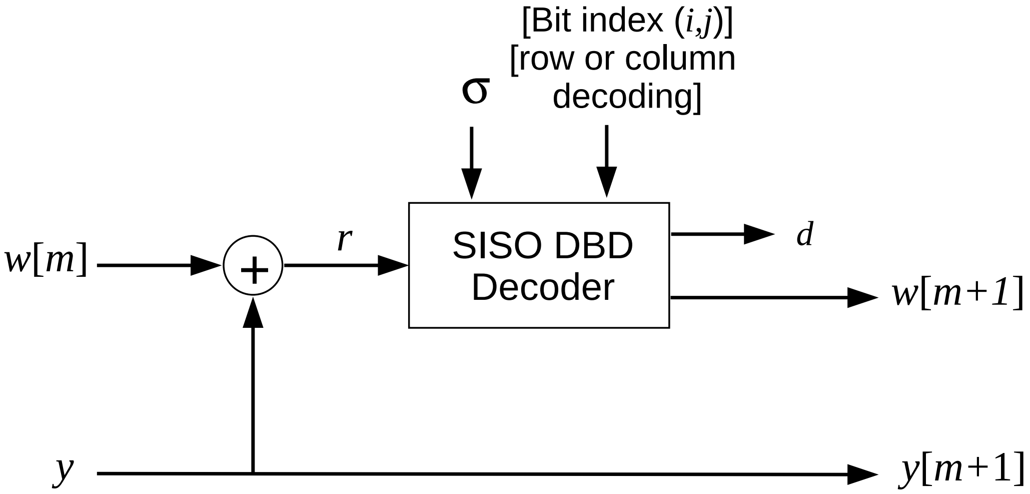


Figure 3.5. Modified one stage SISO TPC decoder

Euclidean distance between the received vector and the decision codeword [39]:

$$\phi = f\left(\sum_{j\in\{j|(r_j-d_j).d_j<0\}} (r_j - d_j)^2\right) \tag{3.3.1}$$

The destructive Euclidean distance is defined as the sum of Euclidean distance where the noise vector has a different polarity from the decision vector. The soft output bits for received bits $\mathbf{Y}$ are then calculated by using the Distance Based Decoding (DBD) method for decoding product turbo codes in [39]. The final soft output for the j-th bit is given by:

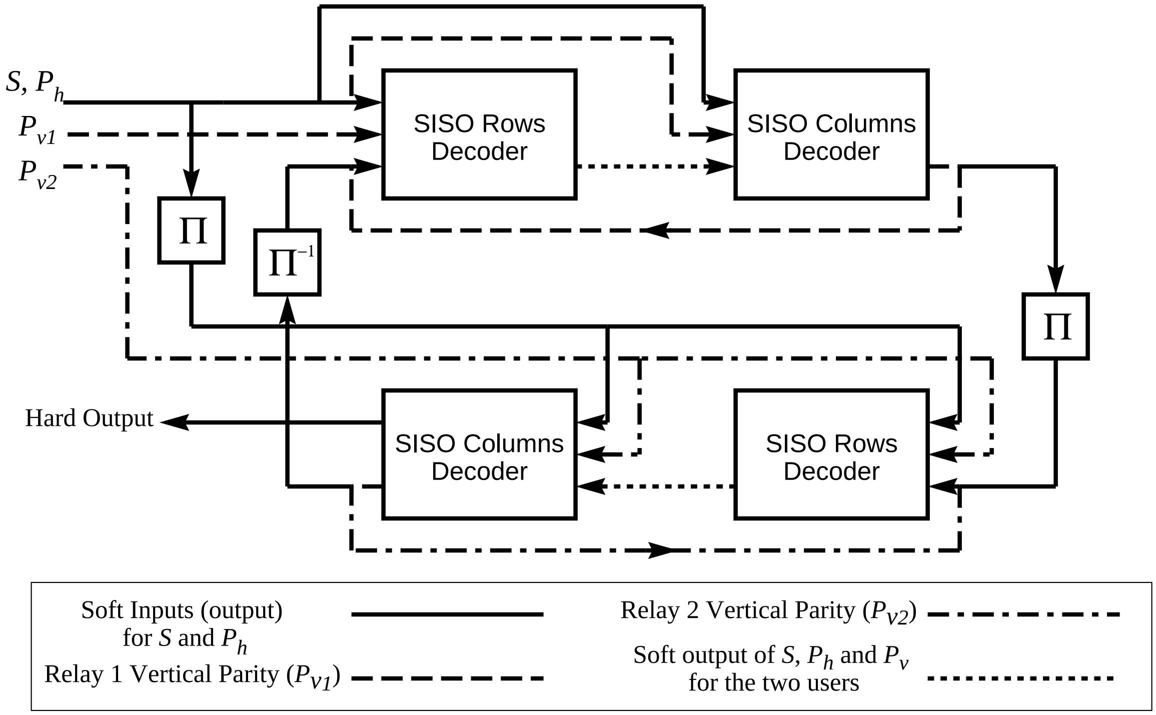
$$w_j = d_j\left(\frac{\sigma^2}{2}\ln\left(\frac{\phi + \exp(2r_jd_j/\sigma^2)}{1-\phi}\right) - r_jd_j\right) \tag{3.3.2}$$

where w_j is the normalized log extrinsic information output, d_j is the element of the decision codeword, r_j is the soft input bit to the decoder.

Fig. 3.5 depicts the modified decoder implementation for the cooperative based decoder. The diagram represents one stage of decoding (along the rows or the columns); m denotes the m-th decoding stage, d is the hard decoded output and y is the channel output. The input bit to the decode r is the summation of the channel output and the normalized log extrinsic information of the previous decoding stage.

For this decoder, the value of soft output depends on the channel standard deviation σ, so this value must be provided at the input of the decoder. The value of σ is given by the following relation according to the indices (i, j) of the input bit in the TPC matrix of Fig. 3.1:

$$\sigma = \begin{cases} \sigma_{sd} & \text{Blocks}\{S, P_h\} \\ \sigma_{rd} & \text{Block}\{P_v\} \end{cases} \tag{3.3.3}$$



S, P_h
P_{v1}
P_{v2}
SISO Rows Decoder
SISO Columns Decoder
Π
Π^{-1}
Π
Hard Output
SISO Columns Decoder
SISO Rows Decoder
Soft Inputs (output) for S and P_h
Relay 1 Vertical Parity (P_{v1})
Relay 2 Vertical Parity (P_{v2})
Soft output of S, P_h and P_v for the two users

Figure 3.6. Multiple vertical parities DTPC decoder

When the destination receives more than one vertical parity it uses multiple decoding stages for each iteration. The proposed MVP-TPC decoder is illustrated in Fig. 3.6 for the case when two vertical parities are received at the destination. Each SISO decoder in the figure is the one decoding stage shown in Fig. 3.5. Only the soft information for S and P_h blocks are passed through the four decoding stages and are cyclically interleaved or deinterleaved between the first two and the last two decoding stages. The soft information for the vertical parities P_{vi} are only rotated between two decoding stages, the first and the second for P_{v1}, the third and the fourth for P_{v2}. This guarantees maximum transfer of soft information between any two iterations.

3.4 Simulation Results

In this section we discuss simulations results based on the system models and network scenarios presented in the previous sections. For simplicity, perfect channel estimates are assumed at the receiver, the block fading coefficients in (3.2.1), (3.2.2) and (3.2.3) are normalized to 1. For the line model network scenario, simulations are carried on for 0.1 steps of λ in the range $0 \leq \lambda \leq 1$; where $\lambda = 0$ means that the relay is placed at the destination and $\lambda = 1$ means that the relay is placed at the source (non-cooperative mode). In the second simulations for the transmit cluster network scenario, the SNR of the relay channel is assumed to be X dBs greater than the direct link channel. The performance of the DTPC decoding strategy is evaluated when the direct link channel is experiencing different values of SNR.

3.4.1 Single Relay

The results for simulation under free space propagation conditions (path loss exponent is equal to 2) in AWGN channel are displayed in Fig. 3.7, where the three channels are assumed to be mutually independent, and the values of the SNR for interuser and the relay channel are given in (3.2.4) and (3.2.5), respectively. The non-cooperative case is when the source transmits the three blocks of the turbo block code to the destination over the direct link channel without using cooperation.

For values of $SNR_{sr} \leq 4SNR_{sd}$ or $SNR_{sr}[\text{dB}] \leq 6 + SNR_{sd}[\text{dB}]$, i.e when $\lambda \leq 0.5$, the probability of miss-detection in the decoding process at the relay becomes significant enough to dominate and propagate more errors on the turbo decoding at

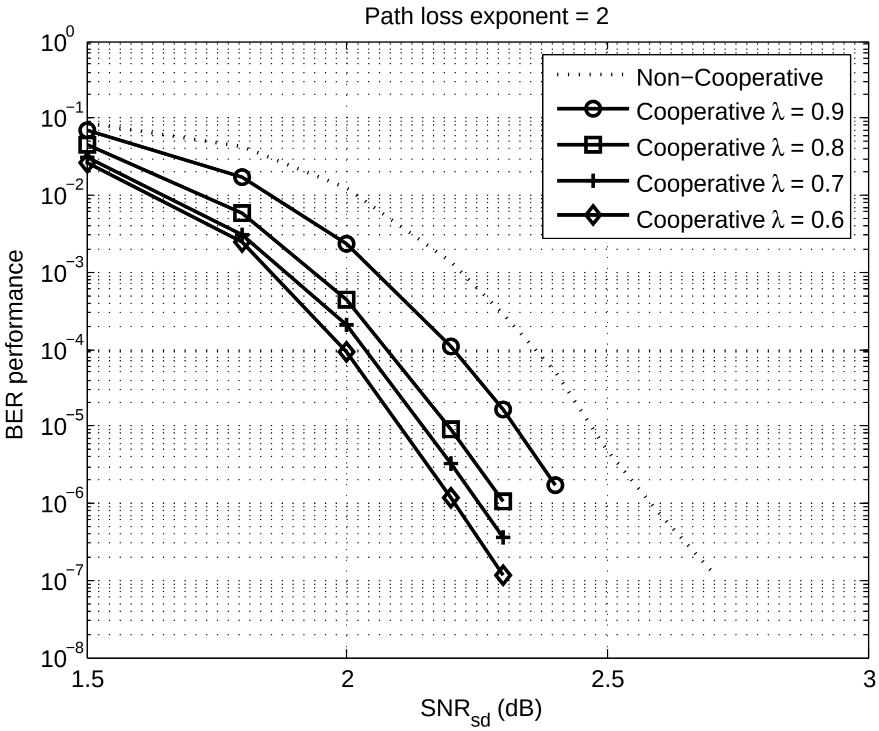


Figure 3.7. BER performance of DTPC in AWGN channel with free space propagation after 4 iterations

the destination since the transmitted block $\hat{P}_v$ has more estimation errors and yet the SISO decoder grants it more confidence than the bits of blocks S and P_h according to (3.3.2) and (3.3.3). The results for these cases were omitted from Fig. 3.7 due to relatively high error floors.

The curves in Fig. 3.7 show that at least 0.3 dB enhancement in the BER performance over the non-cooperative case is gained, that is when the relay is only $0.1L$ from the source ($SNR_{rd} = 1.2SNR_{sd}$), and reaches up to 0.75 dB when $\lambda = 0.6$, i.e. when $SNR_{rd} = 2.8SNR_{sd}$ and $SNR_{sr} = 6.3SNR_{sd}$. After this value of λ, the SNR_{sr} starts to drop below the $6 + SNR_{sd}$[dB] and error propagation starts at the decoder.

Fig. 3.8 presents another view of the relation between the BER performance and the position of the relay (namely λ), which determines the SNR of the two channels; the interuser channel and the relay channel. Again, the curves demonstrate that the peak BER performance is around the $\lambda = 0.6$ position and the SNR threshold for the

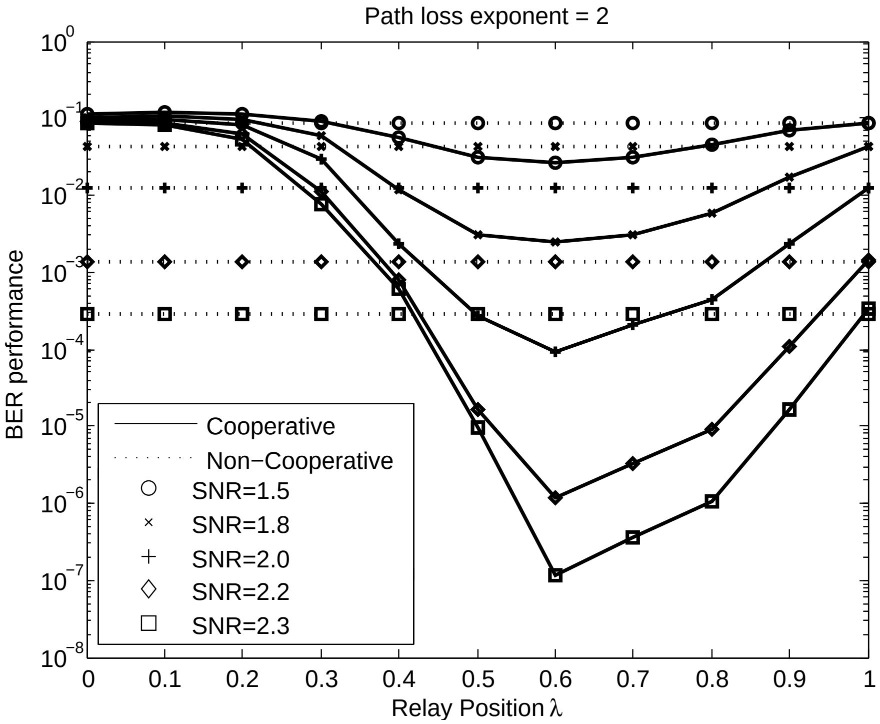


Figure 3.8. BER performance of DTPC in AWGN channel with free space propagation after 4 iterations versus the position of the relay

interuser channel before the BER performance starts to degrade is $6 + SNR_{sd}$[dB].

In some environments, such as buildings, stadiums and other indoor environments, the path loss exponent can reach values in the range of 4 to 6. By doing similar simple analysis as done to (3.2.4) and (3.2.5) to general value of path loss exponent n we get the following:

$$SNR_{sr} = \frac{SNR_{sd}}{(1-\lambda)^n} \tag{3.4.1}$$

$$SNR_{rd} = \frac{SNR_{sd}}{\lambda^n} \tag{3.4.2}$$

We have tested the cooperative system in a relatively lossy environment by setting the path loss exponent to 4. The results for this case are shown in Fig. 3.9. The results in the figure show similar tendency to BER performance enhancement while moving the relay away from the source. The BER performance reaches a peak when

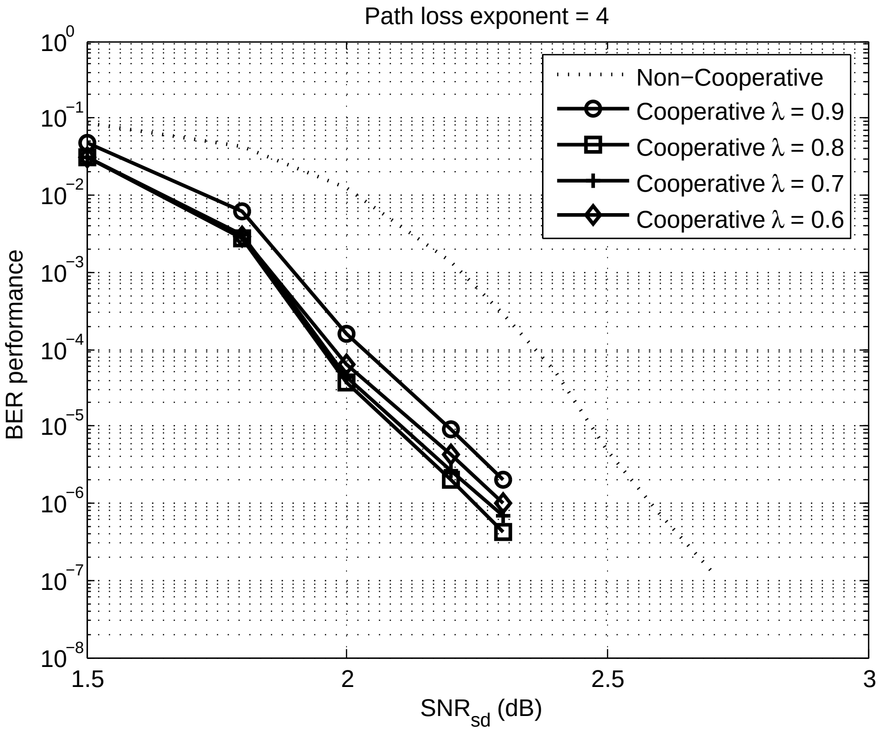


Figure 3.9. BER performance of DTPC in AWGN channel with relatively lossy environment after 4 iterations

the relay is about $0.8L$ from the destination, and starts to degrade gradually for less values of λ. This degradation in performance remains very small until the relay is closer to the destination when $\lambda \simeq 0.3$, where the performance degrades sharply for values of λ less than 0.3.

3.4.2 Multiple Relays

To evaluate the performance of the proposed system, only the two relay system is considered as demonstration of cooperative system with multiple relays, the source broadcasts the blocks S and P_h which will be detected by the two preassigned relays. The two relays, acting independently, decode the received data and use the code space $\mathcal{C}^2$ to generate the vertical parities as described in section 3.3 using cyclic interleavers on one relay.

To evaluate the performance of the cooperative system with more than one relay,

Deployment	(λ_1,λ_2)
DTPC(1)	(0.3,0.5)
DTPC(2)	(0.6,0.6)
DTPC(3)	(0.8,0.7)
DTPC(4)	(0.5,0.5)

Table 3.1. Four different setup for the two relays for simulating two nodes acting as relays in MVP-DTPC

we compare the performance of different relay distributions within the line model scenario and also compare the performance of the system with multiple relays when the transmit cluster network scenario is used. In the two cases, the direct link channel SNR_{sd} is used as a baseline to monitor the performance. While positions of the relays are changed in the line network scenario, the SNR_{sr} and SNR_{rd} are varied independently in the second scenario.

The results for the performance of the two relays cooperative system in line network scenario are shown in Fig. 3.10 and Fig. 3.11. In Fig. 3.10 four different deployments for the two relays are selected and compared. These deployments are denoted as : where λ_1 and λ_2 are the position of the first relay and the second relay respectively. We compare the performance of cooperative systems with multiple relays to the performance of non-cooperative system where the source produce the three parities, P_h, P_{v1}, P_{v2} from user's data S, while the destination applies the multiple parities decoding method illustrated in Fig. 3.6.

The above results show that under cooperative coding of DTPC with two relays outperforms the equivalent non-cooperative TPC case by about $0.3 \sim 0.5$ dB gain at 10^{-3} BER. The obtained coding gain depends on the positions of the two relays and the relative separation between them. The more the nodes are located closer to $0.5 - 0.7$ the better the coding gain.

To investigate more in the influence of the locations of the two relays, Fig. 3.11 shows the BER performance of the two relays cooperative network versus the location

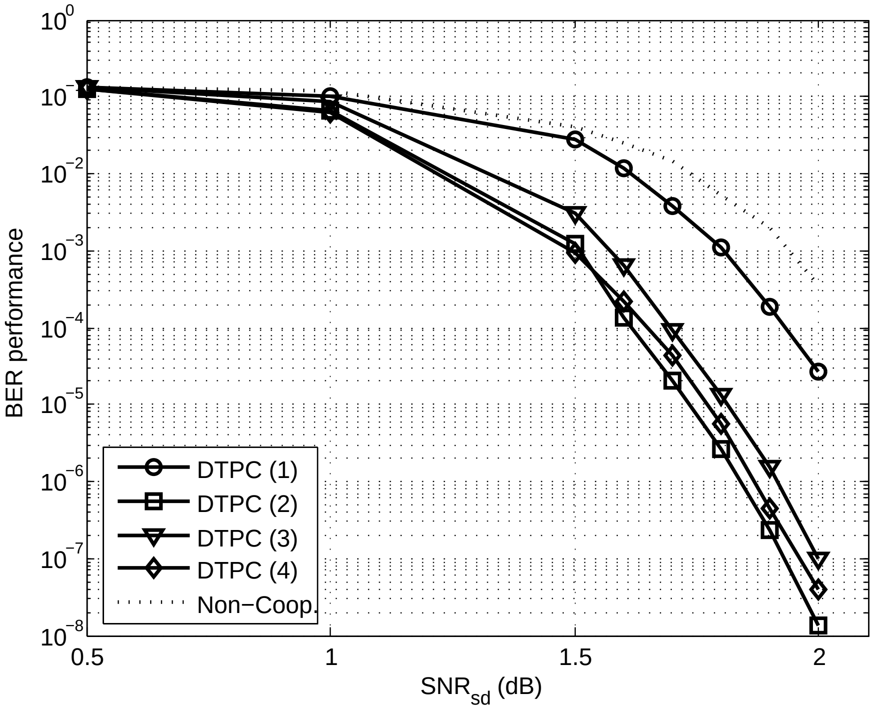


Figure 3.10. BER performance of DTPC with multiple relays in AWGN after 4 iterations in line network scenario

of one relay along the line connecting the source and the destination while the second relay is fixed at certain point. Also, the performance is tested when the two relays are located at the same position. For lower values of λ_1 (< 0.5) the performance of the configuration with one relay is fixed on the region (> 0.5) is higher than the performance of other configurations when the two relays are located on the region closer to the destination.

The transmit cluster network scenario is used to evaluate the performance of the two relays cooperative network in Fig. 3.12 in AWGN channel when $SNR_{sd} = 2$. In the figure, $SNR_{rd} - SNR_{sd}$ for the two relays is varied while Δ SNR, which refers to the difference between the SNRs of interuser channel and direct link channel ($SNR_{sr} - SNR_{sd}$) for the two relays, is kept constant. It is very clear from the results of this figure that the BER of the overall cooperative system is limited by the SNR of the direct link channel SNR_{sd}. The results also demonstrate that the interuser

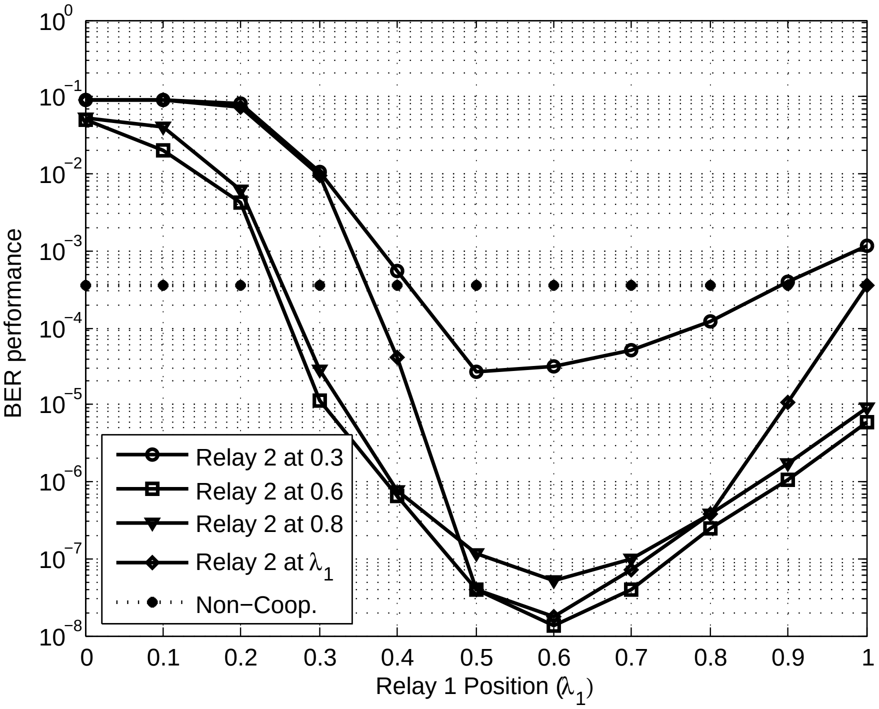


Figure 3.11. BER performance vs. relay 1 position λ_1 of DTPC with multiple relays in 2 dB AWGN SNR_{sd} channel after 4 iterations in line network scenario

channel for the two relays should be at least 5 dBs higher than the direct link channel so that the relays can help in enhancing BER the performance.

3.5 Conclusions

The results presented in this chapter illustrate that a cooperative coding system based on DTPC performs better than the non-cooperative TPC. With the same spectral and power efficiency, the new distributed TPC can give BER performance that is closer to the channel capacity. The line network topology is used only for simplicity; any position for the relay can aid in enhancing the BER performance, as long as the following two conditions are satisfied: the SNR of the interuser channel is at least 6 dBs higher than the direct link channel,and SNR of the relay channel is higher than the SNR of the direct link channel. The overall performance of the system is enhanced when the direct channel is experiencing shadowing effects, but

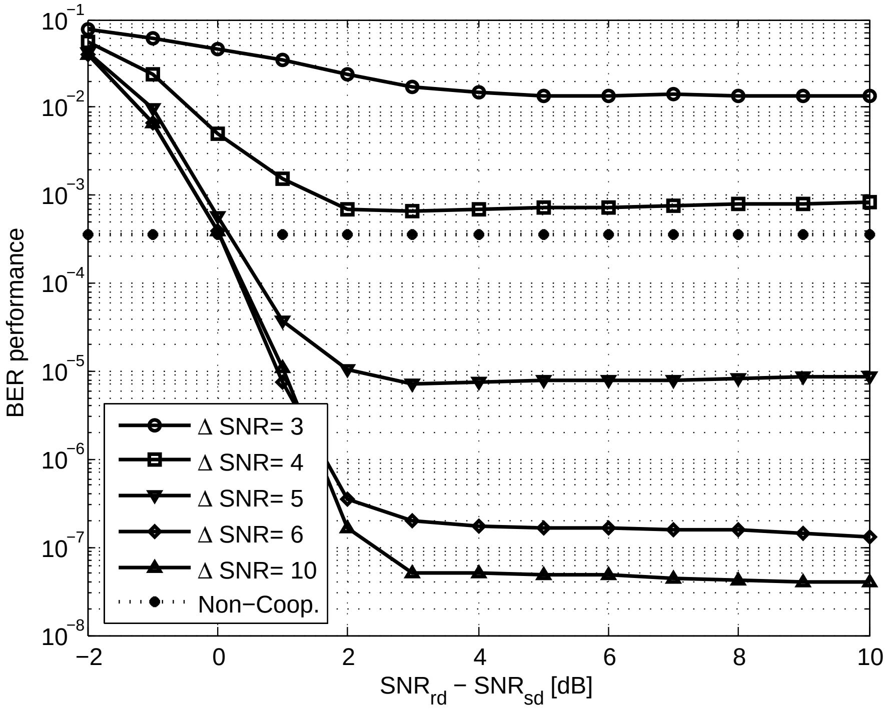


Figure 3.12. BER performance vs. $SNR_{rd} - SNR_{sd}$ of DTPC with multiple relays in AWGN channel after 4 iterations in transmit cluster network scenario for different Δ SNR $=SNR_{sr} - SNR_{sd}$

the performance is limited by the SNR of the direct channel. The results for single and multiple relays show that the BER performance can be improved by increasing the number of relays and enhancing the interuser channel, but it is limited by the SNR of the direct link channel.

Chapter 4

Distributed Turbo Product Codes with Soft Incremental Redundancy

4.1 Introduction

The inter-user channel in the DF techniques is always assumed to be so good, so the decoding at the relay is considered to be reliable. However, in real communication systems, channels are susceptive to variations and SNR of the inter-user channel might fall below the reliability level. In this case, error free assumption at the relay is no longer valid, and therefore the performance of the conventional DF is significantly degraded since the relay is forwarding erroneous IR.

The DF hard-slices the received signal at relay using a comparator and thus discards the soft information, which would help otherwise in the decoding process at the destination node. Since AF forwards amplified version of the received signal it keeps itself from any incorrect decision and conserves the soft information of the received signal. However this scheme does not benefit from error correction capability of the received signal at relay and also amplifies and forwards the front end noise at the relay. In order to preserve the soft information of the received signal at the relay a new third cooperation scheme based on soft decoding the received signal has been introduced recently, e.g. [6,7,49,50]. In this approach, which is called Soft Decode and Forward (SDF), the soft output bits are constructed from soft values of the decoded bits. It is shown in previous works that even when the relay fails in recovering the correct information from the source, it is still helpful to the cooperation by sending to the destination the quantized version of the posteriori soft information it gets from the source.

In this chapter, we propose a DTPC system by applying the concept of distributed encoding for the source's message over relay network and use a modified iterative turbo product decoding at the destination to decode the received distributed TPC over multiple channels. For maximum transfer of information from the relay, we propose and derive a soft parity generation method to be used at the relay. The soft information relaying method uses the Chase II decoding algorithm [33] for decoding and DBD to calculate the LLR values of the decoded bits, then uses the LLR values to generate soft parity bits that is forwarded to the destination. We investigate the performance of the SDF-TPC in AWGN channel using simple network topologies and compare the results with the DF-TPC proposed in a previous work [51] and with the non-cooperative TPC.

4.2 System Model

We consider three nodes network system, consisting of one source 's', one destination 'd' and one relay 'r' as shown in the system model depicted in Fig. 4.1. The three terminals communicate in a half-duplex mode, and any transmission from a source to a destination requires two time slots. The source employ a simple component code for the input data using (n, k, δ) Extended Bose-Chaudhuri-Hochquenghem (EBCH) encoder. The EBCH code is obtained by adding the overall parity check to the conventional BCH codeword to expand the minimum hamming distance from δ to $\delta + 1$ [41]. The source transmits block of EBCH encoded codewords in the first time slot to the destination and the relay in broadcast mode. Subsequently, the relay

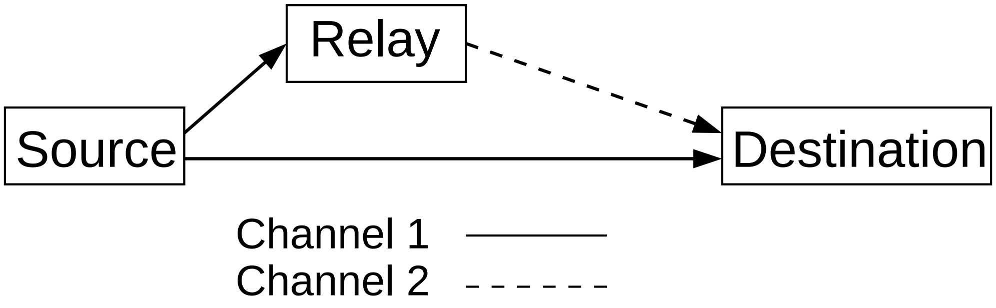


Figure 4.1. System Model for a general cooperative system

correct the received sequence and produce vertical parity by arranging the decoded codewords in rows and encoding them vertically (column-wise) with EBCH codes. The relay generates soft information for the vertical parity bits by using the LLRs of the corrected bits. In the second time slot, the relay transmits quantized version of the soft information to compact the signal within the specified bandwidth.

The channels between the three nodes are assumed to be mutually independent with block Rayleigh fading coefficients perfectly known to the corresponding receiver, where the block length is larger than the the two time slot. For BPSK modulation, transmitted codeword from node i is denoted as $\mathbf{X}_i = \{x_i^1, x_i^2, \cdots, x_i^n\}$, where $x_i^l \in \{-1, +1\}, l = 1, 2, \cdots, n$. Codeword's bits x_i^l, $0 \leq l \leq k$, are systematic bits, whereas the bits x_i^l, $k+1 \leq l \leq n$, are the parity bits generated for the systematic bits using the EBCH (n, k, δ) encoder. The received sequence of corrupted bits corresponding to the transmitted codeword $\mathbf{X}_i$ at node j over block faded AWGN channel denoted $\mathbf{Y}_j = \{y_j^1, y_j^2, \cdots, y_j^n\}$. Where the received signals at the destination and the relay in terms of the transmitted signals from the source and relay over two time slots can be expressed as:

$$\begin{aligned}
\mathbf{Y}_d[2k-1] &= \alpha_{sd}[2k-1]\mathbf{X}_s[2k-1] + \mathbf{Z}_{sd}[2k-1], && (4.2.1)\\
\mathbf{Y}_r[2k-1] &= \alpha_{sr}[2k-1]\mathbf{X}_s[2k-1] + \mathbf{Z}_{sr}[2k-1], && (4.2.2)\\
\mathbf{Y}_d[2k] &= \alpha_{rd}[2k]\mathbf{X}_r[2k] + \mathbf{Z}_{rd}[2k], && (4.2.3)
\end{aligned}$$

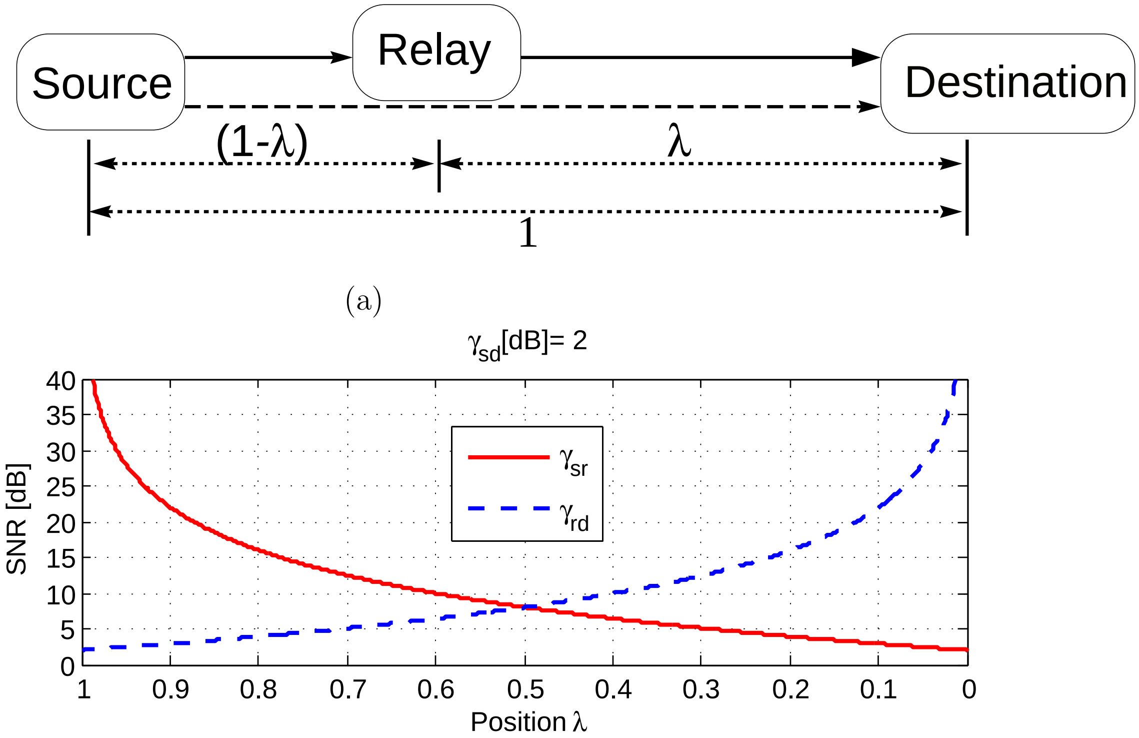
where $k \in \{1, 2, \cdots\}$ represents the time slot; the source transmits at odd time slots and the relay transmits at even time slots. α_{ij} is the channel block fading coefficient. $\mathbf{Z}_{ij} = \{z_{ij}^1, z_{ij}^2, \cdots, z_{ij}^n\}$ are zero mean i.i.d AWGN on the channel between the nodes i and j, $i \in \{s, r\}$, $j \in \{r, d\}$.

To simplify the simulation model, different scenarios for the location of the intermediate relays are assumed. To investigate the performance of our proposed system under different inter-user and relay channel conditions with minimum simulations we assume that the relays are located on the line connecting the two ends as shown in Fig. 4.2(a), [9, 43–46], where $0 \leq \lambda \leq 1$ represents separation between the relay and the destination when the distance between the source and the destination is normalized to one.

Assuming that the transmit power from the source and the relay are equal, we obtain the values of λ_{sr} and λ_{rd} with respect to the reference value γ_{sd} as:

$$\begin{aligned}
\gamma_{sr} &= \frac{\gamma_{sd}}{(1-\lambda)^2}, && (4.2.4)\\
\gamma_{rd} &= \frac{\gamma_{sd}}{\lambda^2}, && (4.2.5)
\end{aligned}$$

where γ_{ij} refers to the SNR at receiver j for a transmission from transmitter i. These relations for γ_{sr} and γ_{rd} are plotted as a function of λ in Fig. 4.2(b) when the relay position λ is changed from 1 to 0 when $\gamma_{sd} = 2$ dB. From (4.2.4) and (4.2.5), the variances of the $z_{ij}^l, l = 1, 2, \cdots, n$ i.i.d AWGN in (4.2.2) and (4.2.3) can be expressed



(a)

(b)

Figure 4.2. (a) The simplified three terminals in line model. (b) SNR of the inter-user and relay channels versus the position λ of the relay in line model.

as:

$$\begin{aligned}\sigma_{sr}^2 &= (1-\lambda)^2\sigma_{sd}^2,\\ \sigma_{rd}^2 &= \lambda^2\sigma_{sd}^2,\end{aligned}$$

where $\mathrm{var}(z_{sd}^l) = \sigma_{sd}^2$, $\mathrm{var}(z_{rd}^l) = \sigma_{rd}^2$ and $\mathrm{var}(z_{sr}^l) = \sigma_{sr}^2$.

The information bits are formed into a $k \times k$ matrix S and then encoded with a (n, k, δ) EBCH code to produce rows parity P_h. The output $k \times n$ matrix of bits is broadcasted from the source. At the relay, a SISO decoder is used to obtain the LLR values of the decoded codewords for the received sequences from the source in the first time slot. The LLR values are used to obtain the soft information for the vertical parity bits which are obtained by arranging the received codewords in rows and encoding the columns with EBCH code to produce $\hat{P}_v$. The method used to obtain the soft information from the LLR values of the input bits is explained in section 4.3.

The generated soft information are quantized and transmitted to the destination in the second time slot. The three matrices S, P_h and $\hat{P}_v$ are received at the destination over two time slots (not necessarily of equal length); S and P_h matrices are received from the direct link between the source and the destination, while the matrix $\hat{P}_v$ is received from the relay channel.

4.3 Soft Decode and Forward

In this section, the proposed method for obtaining soft information about the generated parity bits at the relay is discussed. As discussed in the introduction, the SDF cooperative technique combines the advantages of both DF and AF techniques by using error correction and sending soft information about the parity bits instead of using hard decision decoding. The received sequence of bits are corrected using SISO decoder which takes the channel output as input and delivers extrinsic information about the decoded bits. Afterward, the LLR values of the decoded bits are calculated which is used to generate the LLR values for the parity bits. From the latter, soft information about the parity bits is obtained and forwarded to the destination.

Block encoding takes k bits information word, and generates n bits codeword. Furthermore, systematic block codes are constructed by appending the $n-k$ parity bits to the end of the k bits input information word. The product codes are formed by serial concatenating the two systematic linear block codes $\mathcal{C}^1$ and $\mathcal{C}^2$ and having an intermediate matrix interleaver. The two component codes $\mathcal{C}^1$ and $\mathcal{C}^2$ have parameters (n_1, k_1, δ_1) and (n_2, k_2, δ_2), respectively, where n_i, k_i, and δ_i stand for code length, code dimension and minimum Hamming distance of the code, respectively. As discussed before, the source broadcasts k EBCH codewords with length n, so a $k \times n$ matrix of bits is received at the relay and the destination. This represents the first encoding stage for the complete TPC. At the relay, matrix interleaving is applied to the received matrix to obtain an $n \times k$ matrix. On other words, the received matrix is transposed so that the rows become columns and the columns become rows. Let the transmitted $k \times n$ matrix of bits from the source be expressed as:

$$\mathbf{X} = [\overline{\mathbf{x}}_1, \overline{\mathbf{x}}_2, \overline{\mathbf{x}}_3, \cdots, \overline{\mathbf{x}}_k]^T,$$

where $\overline{\mathbf{x}}_i = [x_i^1, x_i^2, x_i^3, \cdots, x_i^n]^T$ is a codeword of n bits. Also, let the output of the AWGN channel at the relay be written as:

$$\mathbf{Y} = [\overline{\mathbf{y}}_1, \overline{\mathbf{y}}_2, \overline{\mathbf{y}}_3, \cdots, \overline{\mathbf{y}}_k]^T,$$

where $\overline{\mathbf{y}}_i = [y_i^1, y_i^2, y_i^3, \cdots, y_i^n]^T$, $y_i^j = x_i^j + n_i^j$, $1 \leq i \leq k$, $1 \leq j \leq n$ and $n_i^j \sim N(0, \sigma^2)$, σ^2 is the variance of the AWGN channel.

Upon the receiving of sources transmission, Chase II decoding algorithm is used to decode the received transposed matrix $\mathbf{Y}^T$ to the ML decision $\mathbf{D}$ (matrix of dimension $n \times k$), where

$$\mathbf{D} = [\overline{\mathbf{d}}_1, \overline{\mathbf{d}}_2, \overline{\mathbf{d}}_3, \cdots, \overline{\mathbf{d}}_k],$$

and $\overline{\mathbf{d}}_i = [d_i^1, d_i^2, d_i^3, \cdots, d_i^n]^T$, with $d_i^j \in \{-1, +1\}$. Chase II algorithm searches for the p LRB in the received vector $\overline{\mathbf{y}}_i$ and creates 2^p test patterns by permuting with '0' and '1' in the p LRB positions. A decision codeword $\overline{\mathbf{d}}_i$ is chosen from the vaild codewords in the 2^p list with the minimum Euclidean distance from the received vector $\overline{\mathbf{y}}_i$.

The first step in obtaining the soft parity bits is to find the LLR of the decoded bits in the matrix $\mathbf{D}$. The normalized extrinsic output of the decoder for jth bit in the ith input vector can be expressed in terms of LLR of the decision as:

$$w_i^j = \frac{\sigma^2}{2} L(d_i^j) - y_i^j, \tag{4.3.1}$$

where

$$L(d_i^j) = \ln\left(\frac{P(x_i^j = +1|\overline{\mathbf{y}}_i)}{P(x_i^j = -1|\overline{\mathbf{y}}_i)}\right),$$

is the LLR of transmitted bit x_i^j given the received sequence $\overline{\mathbf{y}}_i$, d_i^j is the decoder decision.

Once a decision codeword $\overline{\mathbf{d}}_i$ is found, its confidence value ϕ_i will be evaluated. The confidence value is defined as the probability that the decoder makes a correct decision given received sequence $\overline{\mathbf{y}}_i$. The value ϕ_i is defined in (4.3.2) as a function of destructive Euclidean distance between the received vector and the decision codeword [39]:

$$\phi_i = f\left(\sum_{j\in\left\{j|(y_i^j - d_i^j).d_i^j < 0\right\}} (y_i^j - d_i^j)^2\right), 1 \le i \le k. \tag{4.3.2}$$

where the function $f(\cdot)$ is pre-defined by a lookup table to reduce the computational complexity of implementation.

Using the DBD proposed in [39] and rewriting the LLR in terms of the normalized extrinsic information and channel output, the LLR of decoder output bit d_i^j is:

$$L(d_i^j) = d_i^j \ln\left(\frac{\phi_i + \exp(2y_i^j d_i^j/\sigma^2)}{1 - \phi_i}\right). \tag{4.3.3}$$

It was found in [52] that the LLR of a parity bit for two statically independent random bits u_1 and u_2 can be obtained as:

$$\begin{aligned} L(u_1 \quad \oplus \quad u_2) = \log \frac{1 + e^{L(u_1)} e^{L(u_2)}}{e^{L(u_1)} + e^{L(u_2)}} \\ \approx \quad \text{sign}(L(u_1) \cdot L(u_2)) \cdot \min(|L(u_1)|, |L(u_2)|). \end{aligned} \tag{4.3.4}$$

Using induction, this relation is generalized to k bits. Assuming that $u_{\mathcal{X}}$ is the parity bit for a set of bits $\mathcal{X} \in \{u_1, u_2, \cdots, u_k\}$, expressed as:

$$u_{\mathcal{X}} = \sum_{u_i \in \mathcal{X}}^{\oplus} u_i,$$

then the LLR of bit $u_{\mathcal{X}}$ given the LLR of set $\mathcal{X}$ is obtained as:

$$\begin{aligned} L(u_{\mathcal{X}}) &= L\left(\sum_{u_i \in \mathcal{X}}^{\oplus} u_i\right) \\ &= \quad \log \frac{\prod_{u_i \in \mathcal{X}} (e^{L(u_i)} + 1) + \prod_{u_i \in \mathcal{X}} (e^{L(u_i)} - 1)}{\prod_{u_i \in \mathcal{X}} (e^{L(u_i)} + 1) - \prod_{u_i \in \mathcal{X}} (e^{L(u_i)} - 1)} \\ &= \quad 2 \cdot \tanh^{-1}\left(\prod_{u_i \in \mathcal{X}} \tanh(L(u_i)/2)\right) \\ &\approx \quad \text{sign}\left(\prod_{u_i \in \mathcal{X}} L(u_i)\right) \cdot \min_{u_i \in \mathcal{X}} |L(u_i)|. \end{aligned} \tag{4.3.5}$$

where the third equality follows from using the two following relations:

$$\begin{aligned} \tanh(u/2) &= \frac{e^u - 1}{e^u + 1}, \\ 2 \cdot \tanh^{-1}(u) &= \log \frac{1 + u}{1 - u}. \end{aligned}$$

The parity bits for a linear block code can be obtained by using the generator matrix for this code. A linear code generator matrix is any matrix whose rows are vector representations of a base of the code. For EBCH code, this matrix is written in systematic form as $\mathbf{G} = [\mathbf{I}_k|\mathbf{P}]$,where $\mathbf{I}_k$ is the identity matrix of rank k, $\mathbf{P} = [\bar{\mathbf{p}}_{k+1}, \bar{\mathbf{p}}_{k+2}, \cdots, \bar{\mathbf{p}}_{n-1}]$ is a $k \times (n-k-1)$ matrix responsible for generating the $n-k-1$

parity bits for the information bits, where $\bar{\mathbf{p}}_i$ is a k-bit vector in Galois Field (GF)(2). The last parity bit n in the EBCH codeword (the overall parity bit) is generated such that the overall number of 1's in the codeword is odd. Encoding for linear codes is done by multiplying the k-bit information input by the generator matrix $\mathbf{G}$. By definition of systematic encoding, the first k bits of the output codeword are the same as input bits, and the other $n-k$ bits are the parity bits. At the relay we are only interested in generating the parity bits for the decoded matrix for the received matrix over the inter-user channel (the channel between the source and the relay). The decoded matrix $\mathbf{D}$, which is composed of estimates of S and P_h, is considered as systematic information at the relay. Let $\mathbf{E}$ $(n \times n)$ be the result from EBCH encoding of decoded matrix $\mathbf{D}$:

$$\begin{aligned}
\mathbf{E} &= [\mathbf{DG}|\ \bar{\mathbf{e}}_n] \\
&= [\mathbf{DI}_k|\mathbf{DP}|\ \bar{\mathbf{e}}_n] \\
&= [\mathbf{D}\quad|\mathbf{DP}|\ \bar{\mathbf{e}}_n] \\
&= [\bar{\mathbf{e}}_1, \bar{\mathbf{e}}_2, \bar{\mathbf{e}}_3, \cdots, \bar{\mathbf{e}}_k, \bar{\mathbf{e}}_{k+1}, \cdots, \bar{\mathbf{e}}_{n-1}, \bar{\mathbf{e}}_n].
\end{aligned}$$

The resulting matrix $\mathbf{E}$ is composed of two parts, the systematic and the parity parts, $\mathbf{E_s} = [\bar{\mathbf{e}}_1, \bar{\mathbf{e}}_2, \bar{\mathbf{e}}_3, \cdots, \bar{\mathbf{e}}_k] = [\overline{\mathbf{d}}_1, \overline{\mathbf{d}}_2, \overline{\mathbf{d}}_3, \cdots, \overline{\mathbf{d}}_k] = \mathbf{D}$ and $\mathbf{E_p} = [\bar{\mathbf{e}}_{k+1}, \cdots, \bar{\mathbf{e}}_{n-1}, , \bar{\mathbf{e}}_n]$ respectively, where $\bar{\mathbf{e}}_i = [e_i^1, e_i^2, \cdots, e_i^n]^T$ is n bits vector. The latter which we refer to as the estimate of the vertical parity $(\hat{P}_v)$ is transmitted along with it's soft information to the destination. To obtain the soft information for the generated parity $\mathbf{E_p}$ we use the result in (4.3.5). The LLR for the parity bit e_i^j, $k+1 \leq i \leq n-1, 1 \leq j \leq n$

is given by:

$$\begin{aligned} L(e_i^j) &= L\left(\left(\bar{\mathbf{d}}^j \cdot \bar{\mathbf{p}}_i\right)_2\right) \\ &= L\left(\sum_{l \in \mathcal{X}_i} \oplus\, d_l^j\right), \mathcal{X}_i = \{l | p_i^l = 1\} \\ &\approx \operatorname{sign}\left(\prod_{l \in \mathcal{X}_i} L(d_l^j)\right) \cdot \min_{l \in \mathcal{X}_i} \left|L(d_l^j)\right|, \end{aligned} \tag{4.3.6}$$

where $\bar{\mathbf{d}}^j$ is the jth row in $\mathbf{D}$, $\mathcal{X}_i$ refers to the set of indices in which the vector $\bar{\mathbf{p}}_i$ has 1's. The subscript '2' in the first equality means the dot product operation is on GF(2). The values of $L(d_l^j), l \in \mathcal{X}_i$ are found from (4.3.3). The LLR of the last column of $\mathbf{E}$, i.e. $\bar{\mathbf{e}}_n$, that is composed of rows' overall parity bits, is obtained by setting $\mathcal{X}_n = \{1, 2, 3, \cdots, n-1\}$ in (4.3.6).

Let the matrix $\mathbf{Q}$ whose element $q_i^j, k+1 \leq i \leq n, 1 \leq j \leq n$ equals to the soft information for the parity bit e_i^j:

$$q_i^j = \sqrt{E_p}\frac{2}{\sigma^2}L(e_i^j), \tag{4.3.7}$$

where E_p is a constant such that the total transmitted power from the relay $\leq P_p = \frac{n(n-k)}{n^2}P$, P is the maximum transmission power for all the TPC bits. To investigate the maximum achievable performance, the output soft information is sent in the AWGN channel without quantization. This is equivalent to quantize the soft information with relatively high quantization levels. The authors in [49] proposed a quantization and compression method that can be used at the relay to forward the soft information by taking into account the distribution of the output of the decoder.

4.4 Decoding at Destination

The destination receives the two parts of the TPC matrix $\{S, P_h\}$ and $\{P_h\}$ via the direct and relay AWGN channels with different SNRs on time slots $2k-1$ and $2k$, respectively, as in (4.2.1) and (4.2.3). After receiving the two parts, the destination

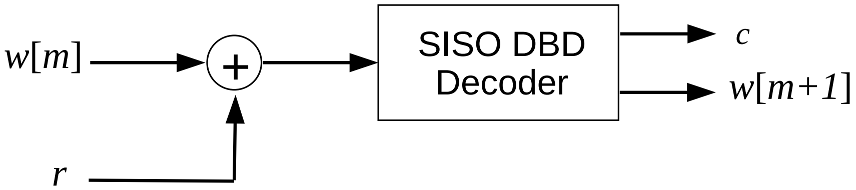


Figure 4.3. Modified one stage SISO TPC decoder

arranges the received data matrices S, P_h and $\hat{P}_v$ as in Fig. 3.1 in order to start rows and columns iterative decoding process.

The basic component of the turbo product decoder at the destination is the SISO decoder used to decode the rows and columns. Similarly to the soft decoding performed at the relay, the SISO decoder at the destination uses Chase II algorithm [33] to form 2^p test patterns after finding the p LRB bits. Decoding complexity is reduced by considering only the codewords in the 2^p list which are the most probable codewords of all the codewords. For each decoded row or column, a decision codeword $C = \{c_1, c_2, \ldots, c_n\}$ with $c_j \in \{-1, +1\}$ is chosen from codewords in the list with the such that it has the minimum Euclidean distance from the input vector. The confidence value ϕ of the decision codeword C will be evaluated using (4.3.2). The final normalized log extrinsic soft output for the jth bit $(1 \leq j \leq n)$ of the decoded codeword is given by [39]:

$$w_j = c_j \left(\frac{\sigma^2}{2} \ln \left(\frac{\phi + \exp(2r_j c_j/\sigma^2)}{1 - \phi} \right) - r_j c_j, \right) \tag{4.4.1}$$

where w_j is the normalized log extrinsic information output, c_j is the element of the decision codeword, r_j is the soft input bit to the decoder.

Fig. 4.3 shows the SISO decoder implementation for the cooperative based decoder. The diagram represents one stage of decoding along the rows (columns); m denotes the mth decoding stage, c is the hard decoded output and r is the channel output. The input bit to the decode is the summation of the channel output and the normalized log extrinsic information of the previous decoding stage. For the first decoding stage, w_j $(1 \leq j \leq n)$ is set to zero for all the decoded rows.

According to (4.4.1), the value of soft output depends on standard deviation σ of the channel. In our simulations, we assume that the two values of σ for the two received parts at the destination are provided at the input of the decoder. In rows decoding of the TPC matrix, the value of σ for the first k rows is equal to σ_{sd} of the direct channel, whereas this value is set to σ_v for the remaining $n-k$ rows of the TPC matrix shown in Fig. 3.1. σ_v is taken from channel measurements about $\hat{P}_v$ available at the input of destination. Since the columns are composed of k bits received over direct channel and $n-k$ bits received over the relay channel special processing is required before decoding. To find the p LRB in the n bits vector, the bits are first normalized by multiplying the first k bits with $\frac{2}{\sigma_{sd}^2}$ and multiplying the remaining $n-k$ bits with $\frac{2}{\sigma_v^2}$. Also, to find the normalized log extrinsic output information, the value of σ in (4.4.1) must be substitute for σ_{sd} and σ_v for the first k bits and the last $n-k$ bits, respectively.

4.5 Simulation Results

In this section we show the numerical results obtained for simulating the proposed distributed encoding system with soft information forwarding from the relay. As presented in the previous sections, we used line model to test the system by placing the relay in the line connecting the two end terminals. Simulations are carried out for steps of 0.1 of λ in the range $0 \leq \lambda \leq 1$; where $\lambda = 0$ means that the relay is placed at the destination and $\lambda = 1$ means that the relay is placed at the source (non-cooperative mode). Fig. 4.4 shows the performance of the proposed soft incremental redundancy SDF-TPC cooperative coding technique compared to regular hard decision based DF-TPC and the non-cooperative case. Results show that SDF-TPC

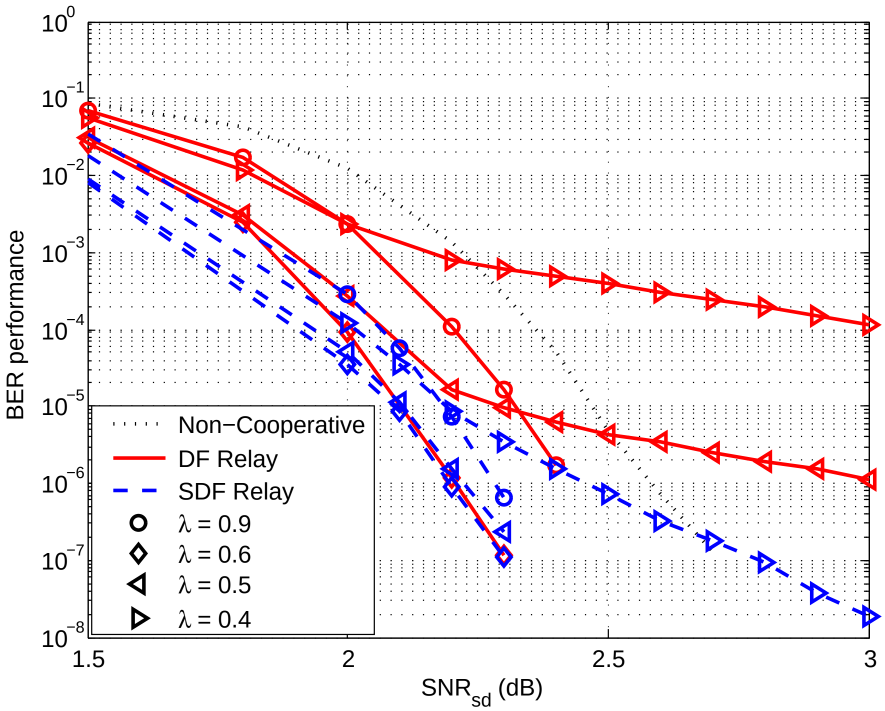


Figure 4.4. BER performance of SDF-TPC in AWGN channel after 4 iterations for different λ compared to the regular DF-TPC and non-cooperative TPC

cooperative coding method has the best performance over the other two systems, namely DF-TPC and non-cooperative TPC.

The BER performance is plotted in Fig. 4.5 as a function of the relay position (λ) between the source and the destination. The relay position determines the SNR of the two channels; the inter-user channel and the relay channel. From this figure, one can infer the minimum required SNR values for the inter-user channel that can yield an enhancement in BER performance over the non-cooperative case. When $\gamma_{sd} = 2$, λ should be greater than 0.25 to get an increase in BER performance compared to non-cooperative TPC. This value for λ corresponds to $\gamma_{sr} > 4$ dB as can be noted from Fig. 4.2(b). As was show in our previous work, the DF distributed encoding method fails for values of $\lambda < 0.5$, in contrarily to SDF method which maintain the good BER performance for less values of λ ($\lambda > 0.3$); i.e. less values of SNR for inter-user channel. In addition to that, the SDF cooperation method exhibits wider range

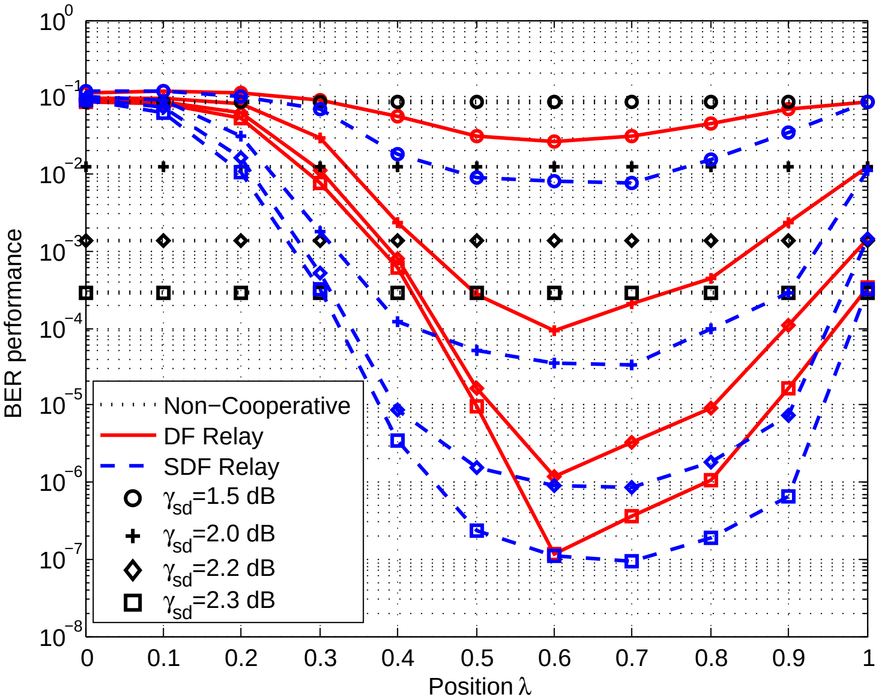


Figure 4.5. BER performance of SDF-DTPC in AWGN channel after 4 iterations versus the position of the relay λ compared to the regular DF-TPC and non-cooperative TPC

of λ for which the relay can boost the performance, in which the performance is less affected by the position of λ (i.e. SNR values for inter-user and relay channels), where the main factor affecting the BER performance is the SNR of the direct channel.

4.6 Conclusions

The results presented in this chapter show that a cooperative coding system based on DTPC with soft incremental redundancy outperforms its two counterparts; the DTPC based on DF and the non-cooperative TPC. With the same spectral and power efficiency, the distributed TPC can give BER performance that is closer to the channel capacity. The line network topology is used only for simplicity; any position for the relay can aid in enhancing the BER performance, as long as the the SNR of the inter-user channel is at least 5 dBs higher than the direct link channel, and SNR of the relay channel is higher than the SNR of the direct link channel. The overall performance of the system is enhanced when the direct channel is experiencing shadowing effects, but the performance is limited by the SNR of the direct channel. In the SDF cooperative coding technique, the relay can help in the distributed encoding process even when the inter-user channel having low SNR. With the SDF cooperative coding technique the relay can boost the performance for the two-ways communication system between any two nodes unlike the conventional DF.

Chapter 5

Power Allocation to Alleviate the Error Propagation on the DTPC

5.1 Introduction

Cooperative communication techniques, specifically cooperative coding, provide the wireless users with higher quality of service and large power savings. With the help of the relay, the performance of data transmission on a channel with deep fading can be improved, and power consumption can be consequently reduced [53, 54].

By observing the performance of the destination's decoder in the DTPC and the Soft Information Relaying (SIR)-DTPC techniques presented in chapters 3 and 4, it is noted that the performance severely degrades when the relay is closer to the destination. There are three main reasons that can justify this behavior as mentioned earlier in previous chapters: Firstly, the two parts of the DTPC matrix are received over two or more different channels which implies that different parts of the code have different SNRs. Secondly, the important part of the code (the systematic bits) is received over a lower SNR (over the direct channel). Thirdly, any SISO decoder will grants the bits of higher SNR more confidence (as can be noticed from equation (2.2.76)). So if the relay makes decoding errors, then the erroneous part of the TPC matrix will have more effect on the direction of the decoding resulting in error propagation. Therefore, an efficient power allocation amongst the source and relay nodes is a necessity to compact the aforementioned causes of performance degradation.

Power Allocation (PA) is considered to have an important role in wireless networks for combating the effect of channel fluctuations especially in multiple carriers systems and increasing the system capacity [55, 56]. A natural question risen in cooperative systems with distributed coding is that how much power should be allocated for source-information transmission and how much for relay-information transmission in order to reach the relay channel capacity.

Most of the existing reported distributed coding schemes are constructed based on fixed power allocations, e.g. [10]. Furthermore, the adaptive power allocation problem under the context of cooperative coding has not been addressed. Cooperative network is constituted of several independent channels, which implies that different channels exhibit different channel conditions, for example, one channel could be suffering from shadowing while another is experiencing higher SNR. Under these circumstances, it is wiser to allocate the power among the active nodes on the cooperative process, by, for instance, loading more power on the links having lower SNR or the nodes transmitting more significant information. At system level, capacity has been shown to be greatly increased when an efficient PA in cooperative relaying is employed [57, 58].

The problem of PA in cooperative networks have been studied usually as a joint problem with the relay selection, e.g. [59–61], that is because the relay selection criterion is seen to depend on both the channel condition information and the the residual power information on the relay and the source. After selecting the relay, the allocated power is chosen to satisfy the required SNR at destination node while taking into consideration the residual power at the relay, because a high SNR value requires more power consumption. However, this scheme for power allocation limits the possible attained performance, and it is only practical when the relays pool is big enough to allow multiple selection of relays.

The power allocation criterion can be solved by optimizing the power consumption in terms of a convex function, such as minimizing the BER, the outage probability or to maximize the capacity, etc. The second approach, the sub-optimal approach, when the objective function is not convex is to allocate the power subject to a given requirement that maximizes the performance. An example of the power allocation algorithm that is based on a certain requirement is in [62,63], where an efficient power allocation strategy is proposed to AF networks to satisfy the target SNR requirement at the destination. In [64], a position dependant power optimization for AF and DF relaying protocols is formatted that depends on the relative distances between the three nodes.

To the best of our knowledge, most of the available power allocation algorithms for three terminals network focus on specific relaying protocols, such as AF or DF, and none has studied the power allocation for distributed coding systems. Moreover, most of these algorithms assumes perfect inter-user channel, i.e. zero decoding errors at the relay. Furthermore, the existing power allocation algorithms have their own drawbacks, for instance, the iterative or extensive search for the extreme values may improve the power allocation algorithm performance, but at the expense of both search latency and algorithm complexity. In addition to that, the existing optimization methods require the function to be convex to find the extreme values, and even if the objective function is convex, the optimization algorithm may return no result or a local extreme value. Alternatively, the power can be simply allocated to yield a target SNR requirement at the destination that diminishes the error propagation in the decoding process.

In this chapter, we investigate the further improvement that can be achieved in the DTPC cooperative coding system proposed previously, when the channel attenuations for the direct and relay channels are also known for the source and the relay. Rather than assigning equal power to the source and the relay, as done previously in chapters 3 and 4, we use the relative locations of the source and relay to the destination (and thus the channel attenuations in free space propagation environment) to find the power allocation that results in the desired SNR requirements at the destination's receiver. As discussed before, these SNR requirements are set such that the turbo product decoder has less error propagation. We investigate the effect that the positioning of the relay has on a relaying system and derive power allocation expressions depending on the comparative distances for the two transmitting nodes in the three-terminal model with respect to the destination, which is described in detail in the system model. Contrary to the existing power allocation methods, the method presented here is not based on optimization subject to decreasing the BER at the destination, however it is based on delivering the SNR values at the destination that minimize the error propagation. Practically, the performance of any system ultimately depends on the SNRs, the derived expressions for the SNRs using our model allocate the power per bit based on that, with the final aim of achieving improved BER.

5.2 System Model and Problem Formulation

Consider two EBCH systematic linear block codes $\mathcal{C}_1$ and $\mathcal{C}_2$ with parameters (n_1, k_1, δ_1) and (n_2, k_2, δ_2), where n_i, k_i, and δ_i are codeword length, input information block length, and minimum hamming distance of the linear code $\mathcal{C}_i$, respectively. The complete product code, as depicted in Fig. 5.1, is obtained by serial concatenation of the two linear block codes and by having a matrix transpose in between the two encoders. This is done by arranging the information bits in $k_2 \times k_1$ matrix and then encoding the k_2 rows using code $\mathcal{C}_1$ and then encoding the resulting n_2 columns using code $\mathcal{C}_2$ to finally obtain a $n_2 \times n_1$ matrix. The resulting product code has the new parameters $N = n_1 \times n_2$, $K = k_1 \times k_2$, $\Delta = \delta_1 \times \delta_2$. In this chapter we assume that the two

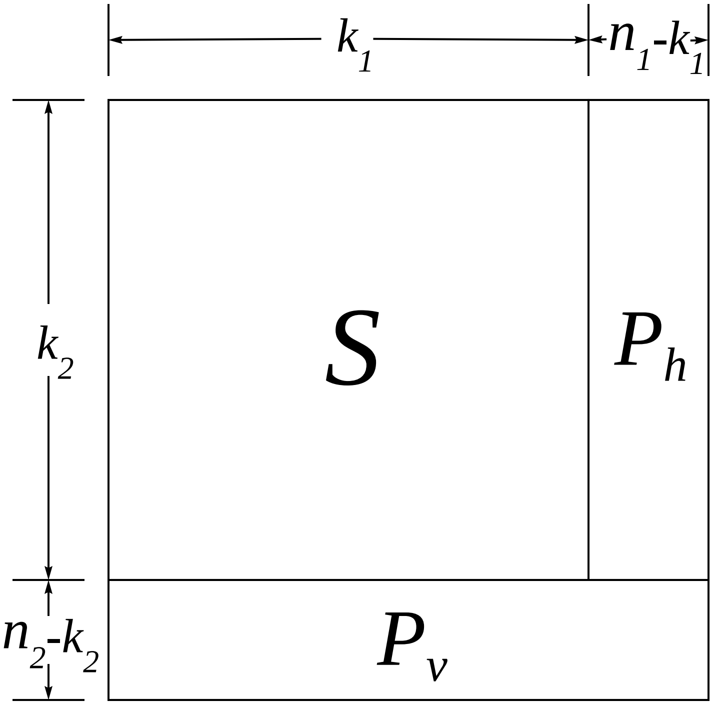


Figure 5.1. The Structure of TPC matrix

component codes are identical and they have the parameters $n_1 = n_2 = n$, $k_1, k_2 = k$ and $\delta_1 = \delta_2 = \delta$.

To establish the distributed encoding for the TPC, the source broadcasts the $k_2 \times n_1$ matrix resulted from the first encoding stage by the $\mathcal{C}_1$ code to the destination and the neighboring relay nodes. One pre-selected relay corrects the received message and uses the second code $\mathcal{C}_2$ to encode the columns of the decoded bits to obtain the $n_2 \times n_1$ matrix of the complete TPC. The relay then transmits only the generated parity bits $((n_2 - k_2) \times n_1)$ from the last encoding process to the distention. The transmitted bits from the relay can be in one of two forms: hard bits DF or soft bits SDF, depending on whether the relay employs SISO decoding and encoding or not, as explained in details chapters 3 and 4. After receiving the two parts of the code, the destination constructs a complete TPC by joining the two received parts and then

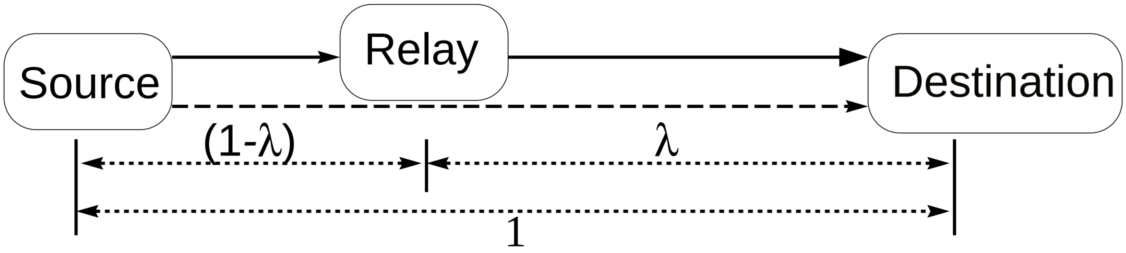


Figure 5.2. The simplified three terminals line network topology

conducts a turbo product decoding.

To simulate the system, it is necessary to test it against all possible SNR values for the three channels, namely, direct, inter-user and relay channels in order to obtain a fully comprehensive understanding of the system's behavior. However. this will extremely increase the simulation complexity and will cover points that are not possible to obtain a real practical system. Alternately, we can use the line simulation model which assumes that the relay is located on the line connecting the source with the destination as in Fig. 5.2. This model is more practical in real systems, when the relay is usually located between the two terminals and the separation distances are relatively large. This model which returns the simulation problem from three-dimensional to two-dimensional problem has been used in many other works, e.g. [9, 43–46].

The received signals at the relay and the destination during the two time slots for the line model can be be generally expressed as follows:

$$y_d[2k-1] = \sqrt{E_s}\alpha_{sd}[2k-1]x_s[2k-1] + n_{sd}[2k-1] \tag{5.2.1}$$

$$y_r[2k-1] = \frac{\sqrt{E_s}\alpha_{sr}}{(1-\lambda)^2}[2k-1]x_s[2k-1] + n_{sr}[2k-1] \tag{5.2.2}$$

$$y_d[2k] = \frac{\sqrt{E_r}\alpha_{rd}}{\lambda^2}[2k]x_r[2k] + n_{rd}[2k] \tag{5.2.3}$$

where y_j denotes the received signal at node j while x_i is the transmitted signal from node j and k is the time slot. The channel between the two nodes i and j has AWGN noise n_{ij}, and channel attenuation α_{ij}. E_s and E_r are the transmit energy/bit for the

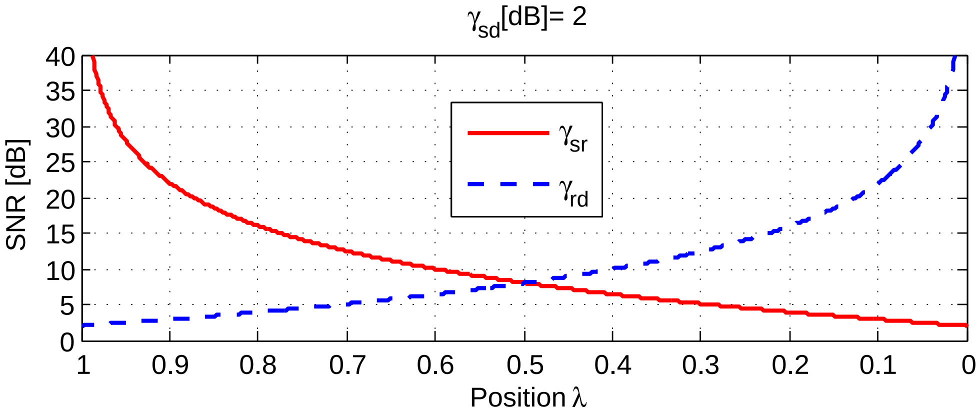


Figure 5.3. SNR of the inter-user and relay channels versus the position λ of the relay in line model.

source and relay, respectively. Henceforth, the subscripts ‘s’, ‘r’ and ‘d’ will be used to denote the source, the relay and the destination nodes, respectively.

Using free space propagation on line model and assuming *fixed transmission energy per bit*, the SNR values for the three channels, viz. γ_{sd}, γ_{sr} and γ_{rd} for the direct, inter-user and relay channel respectively, are related by the following expressions [51]:

$$\gamma_{sr} = \frac{\gamma_{sd}}{(1-\lambda)^2} \tag{5.2.4}$$

$$\gamma_{rd} = \frac{\gamma_{sd}}{\lambda^2} \tag{5.2.5}$$

where $0 \leq \lambda \leq 1$ indicates the position of the relay with respect to the destination when the distance between the source and the destination is normalized to 1, with $\lambda = 0$ when the relay is at the destination. Fig. 5.3 displays how the values of SNR at the destination and the relay change when the relay is moved across the source-destination line for a fixed γ_{sd}.

As can be revealed from the figure, the value of γ_{rd} (dotted-line) reaches very high levels when the relay is closer to the destination compared to the source. This also applies for models other than the line model when the relay is positioned at a distance very small compared to the source-destination separation. At these conditions, when λ is very small, the SNR of the direct channel is very small compared to the SNR of the relay channel. If the relay makes no errors, then it will boost the performance of the cooperative system. However, if it does make erroneous decisions, then these decisions will lead the iterative decoding process farther away from the correct decision, leading to more errors, i.e. error propagation. Unfortunately, since the source's transmission is protected with weaker code, the latter case is very probable when the relay is at relatively smaller distance to destination compared the source-destination separation. The main reason for error propagation in this case is that the destination's decoder grants the bits with higher SNR more reliability, i.e. when the SNR is higher, the variance σ^2 of the AWGN channel will be lower, and consequently the reliability of the decoder's decision given by [65]:

$$\lambda(d_j) = \ln\left(\frac{\phi + \exp(2r_j d_j/\sigma^2)}{1-\phi}\right) \tag{5.2.6}$$

will be higher. And therefore, the decoder's decision will be more biased to the bits sent by the relay, which are, however, more probable to be erroneous.

Power allocation optimization between the source and the relay does not only enhance the decoding performance by processing all the bits of the code equally but also guarantees that the systematic part of the code is received over reasonable SNR. However, to perform the power optimization, many parameters should be considered, such as the BER at the relay, etc. Furthermore, an objective transfer function that takes the variable allocated power at the source and the relay as it's inputs and considers the BER as it's output. However, an exact expression for the BER of the DTPC in terms of the relay network SNR values is very hard to attain. Even for the simple block BCH codes, the only available BER expressions are in union upper bounds form, which does not exactly describe the transfer function of the system. An alternate method for optimization is to conduct an extensive search for the optimal point on which the performance is maximized. This is done by trying all the possible power allocations on the DTPC system on all relay positions and then selecting the allocation at which the BER is minimum. The latter method will find the optimal solution at the expense of complexity and search latency, and the accuracy of the solution will depend on the number of points considered.

In this chapter we propose an alternative way to allocate the power based on avoiding the aforementioned conditions that degrades the performance. All these conditions are caused by the fact that the two parts of the code have different SNRs. Therefore, by simply assigning the power such that all the bits received at the destination have equal SNRs, we eliminate the main factors that causes the decoder to propagate errors. Therefore, the proposed power allocation in this chapter is based on the condition that the received energy/bit for all the parts of the code are equal or in terms of the SNR values:

$$\gamma_{sd} = \gamma_{rd} \tag{5.2.7}$$

which would guarantee that the all the decoded bits are received over the same channel quality.

5.3 Power Allocation (PA)

The requirement for power allocation in (5.2.7) sets the transmit power for the relay and the source for any relay positioning or channel conditions, i.e. it can also apply to channels with shadowing or fading. The total power required to transmit the full TPC matrix is divided into two parts such that:

$$P_s + P_r \leq P \tag{5.3.1}$$

where P is the total power required to transmit one TPC matrix, P_s and P_r are the total power required to transmit the two parts of the code from the source and the relay, respectively.

From the construction of the transmitted TPC matrix in Fig. 5.1 and using the constraint in (5.3.1) in the case of equality, we can obtain the relation for energy per bit using $E = P/n^2$:

$$n^2E = knE_s + (n-k)nE_r \tag{5.3.2}$$

where E is the energy/bit for the non-cooperative case, in which all the TPC matrix is sent over one link. E_s and E_r are the energy/bit for source and relay transmissions in the cooperative case.

Using the free space propagation model, and assuming that the relay and the destination are separated by a fraction λ of the source-destination distance, and using (5.2.7), the energy per bit at the source and the relay can be related by:

$$E_s = \frac{E_r}{\lambda^2} \tag{5.3.3}$$

Substituting (5.3.3) in (5.3.2) and solving for E_s and E_r in terms of E we obtain:

$$E_s = \frac{n^2}{kn + \lambda^2(n-k)n}E \tag{5.3.4}$$

$$E_r = \frac{n^2}{\frac{kn}{\lambda^2} + (n-k)n}E \tag{5.3.5}$$

compared to $E_s = E_r$ in the case for fixed power allocation as described in chapters 3 and 4. Fig. 5.4 shows how the source and relay energy/bit levels changes as the relay-destination separation approaches the distance between the source and the destination. On other words, this is equivalent to power allocation between the source and the relay when direct and relay channels are experiencing different attenuations. Moreover, the results in (5.3.4) and (5.3.5) also apply for models other than the line model.

In the case of line model, the relations in (5.2.4) and (5.2.5) become:

$$\gamma_{sr} = \frac{\gamma_{sd}}{(1-\lambda)^2} \tag{5.3.6}$$

$$\gamma_{rd} = \gamma_{sd} \tag{5.3.7}$$

However, all the SNR values in the two previous equations now depends on the position of the relay λ, even for γ_{sd}, unlike th fixed power allocation.

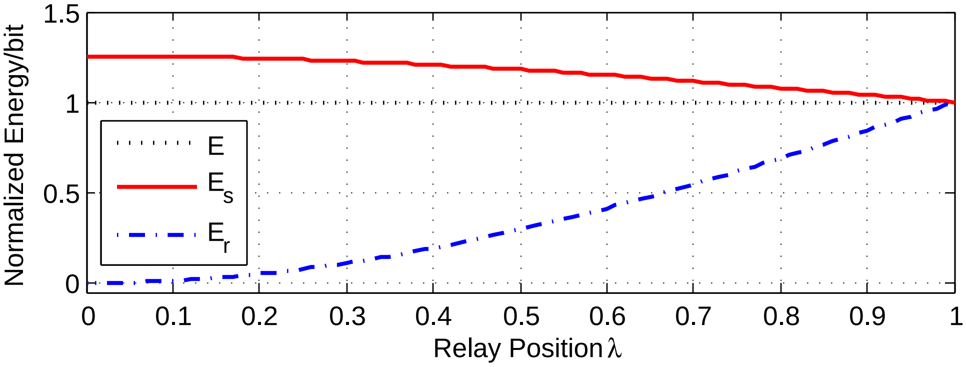


Figure 5.4. Normalized energy/bit for the source and the relay using the proposed power allocation

5.4 Simulations and Results

All the EBCH encoded n-bits codewords from the source and the relay are BPSK modulated and sent to the destination. All the three channels are considered to be orthogonal and have AWGN and the transmitted signals are considered to decay according to free space propagation model, where the path loss exponent is 2. The two component codes used in the DTPC simulations have the same parameters, where $n = 64$, $k = 51$ and $\delta = 6$.

The TPC decoding at the destination is based on the DBD described on section 2.2.6, where channel statistics are assumed to be available for the decoding process. The turbo decoding consists of four iterations, each iteration has two decoding stages. Chase II decoding used at the destination (and the relay in SDF case) uses $p = 4$ LRB to search for the ML codeword.

The simulations are carried over 0.1 steps of λ from 0 up to 1 using the line model, and for every step the system is tested against different values of γ_{sd} to obtain the system performance in terms of the BER. The results for the proposed power allocation method using the DF relaying protocol is shown in Fig. 5.5 plotted versus γ_{sd} for selected values of λ. The solid lines in the figure represent the BER performance

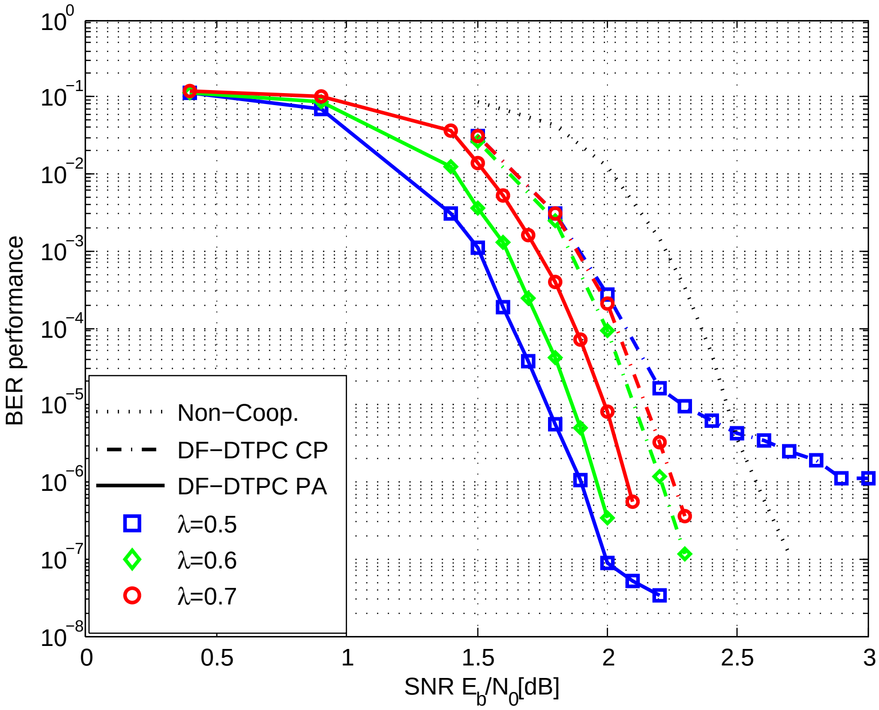


Figure 5.5. BER performance versus γ_{sd} of the DTPC system with DF relaying and power allocation employed compared to the regular system with constant power assignments

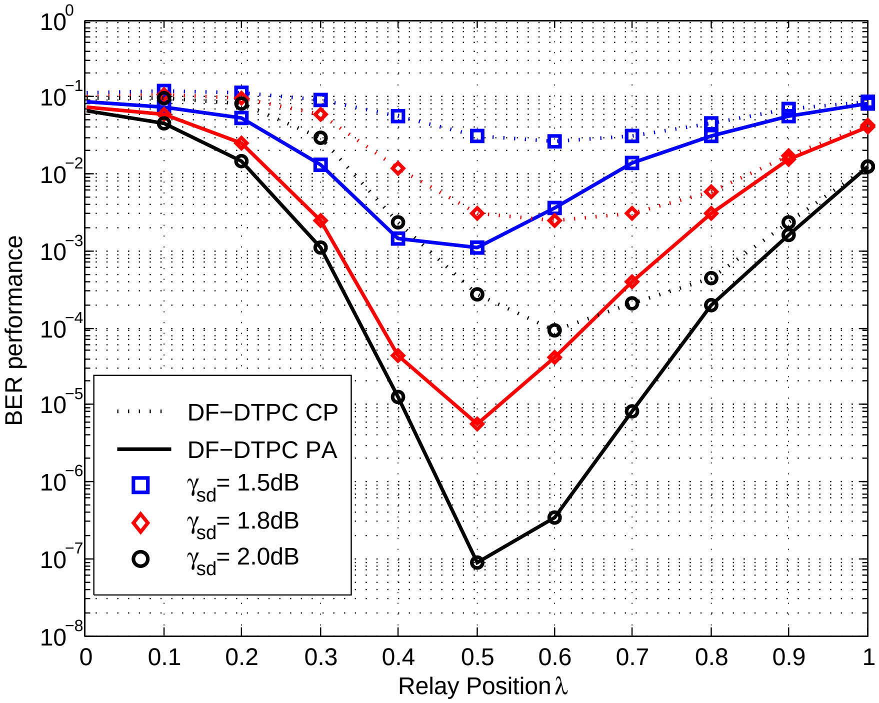


Figure 5.6. BER performance versus relay position λ of the DTPC system with DF relaying and power allocation compared to the system with constant power assignments

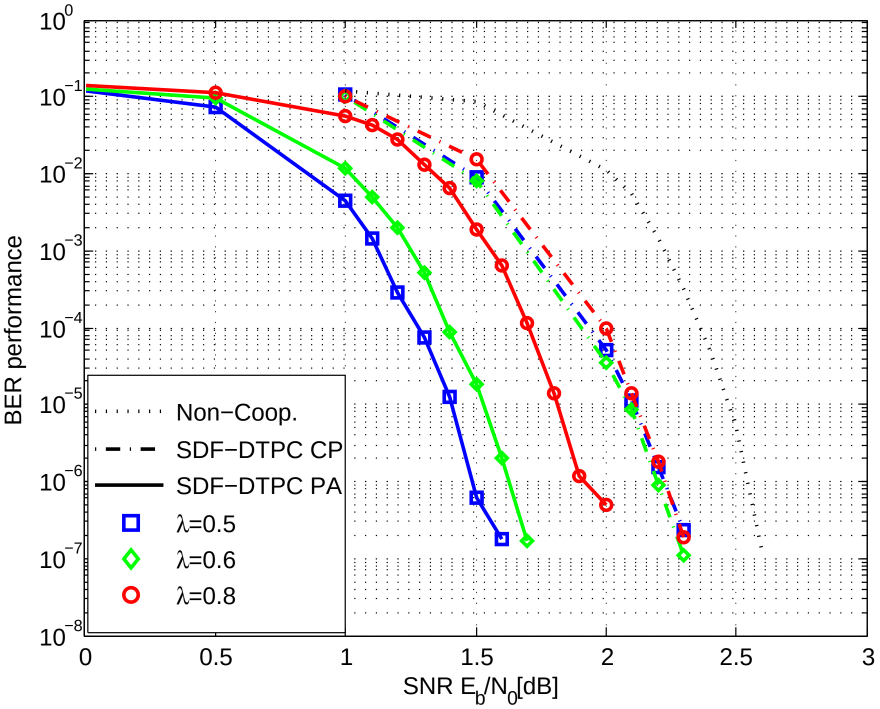


Figure 5.7. BER performance versus γ_{sd} of the DTPC system with SDF relaying and power allocation compared to the system with constant power assignments

curves for the DTPC with DF relaying and with power allocation between the source and the relay, whereas the dashed-dotted lines represent the same system but with constant power assignments. A BER gain of up to 0.5 dB is obtained at by applying power allocation between the two transmitter over the conventional DTPC system with fixed power assignments. Compared to the non-cooperative case, this cooperative gain can exceed 0.7 dB. The same results are plotted against the position of the relay in Fig. 5.6 to verify the applicability of the power allocation for different relay positions. The peak performance is now shifted to a point around 0.5 and the performance is almost symmetric around this point.

The results for the BER performance the DTPC system with power allocation for the case when the relay uses the SDF protocol is depicted in Fig. 5.5 plotted against the SNR of the direct channel for selected relay positions. Compared to the

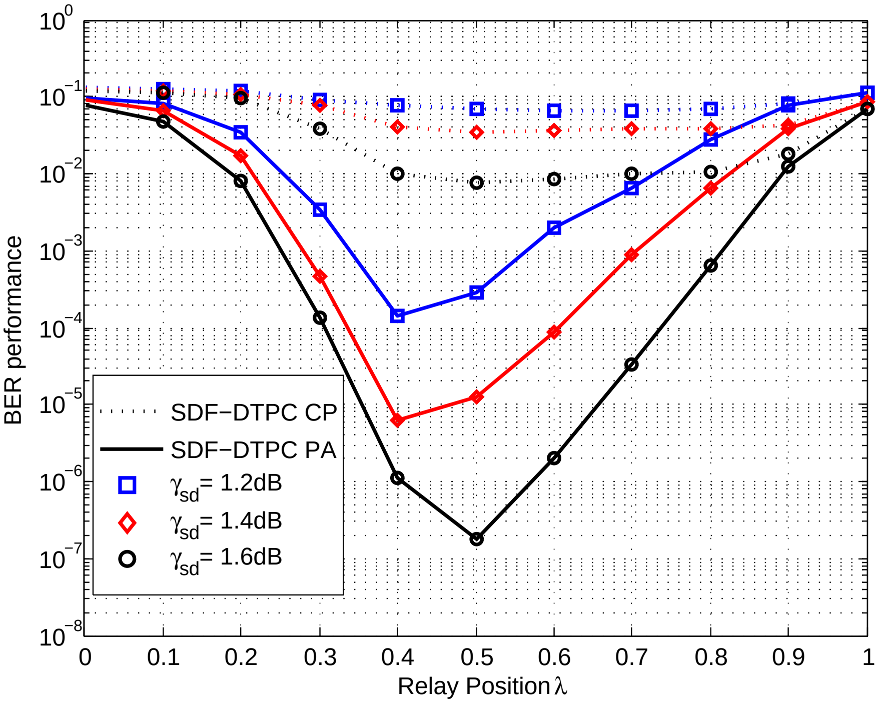


Figure 5.8. BER performance versus relay position λ of the DTPC system with SDF relaying and power allocation compared to the system with constant power assignments

DF relaying technique, the power allocation showed more effectiveness in the case of SDF relaying. For instance, the gain at 10^{-4} BER from using power allocation is 0.7 dB more than the equivalent system with constant power assignments. When compared to the non-cooperative case, the gain from power allocation is about 1.2 dB when the relay is positioned at the mid point between the source and the destination. Fig. 5.8 shows the BER performance as function of the relay position for the two systems with power allocation and with constant power assignments.

5.5 Conclusions

In this chapter, we investigated the error propagation that can appear on the DTPC systems especially when the relay is closer to the destination. Observations show that an approbate power allocation is required for the distributed coding systems to alleviate the error propagation resulted in the iterative decoding. The optimal power allocation can be found using optimization but this, however, requires extensive research for optimal power assignments. Alternately, we proposed a criterion for power allocation and derived expressions for source and relay transmission power. The results obtained from applying the proposed power allocation method show large gain in BER performance and therefore showed effectiveness in allocating power between the transmitting nodes. Finally, it worths saying that power optimization for distributed coding system is a necessity and the results we obtained is just the explorative for the potency of the conjunction of power allocation with distributed coding.

Chapter 6

Power Optimization for the DTPC

6.1 Introduction

In this chapter, we continue our investigation in the power allocation problem for the distributed TPC. Here, we are going to apply the optimization techniques to find the optimal solution for power allocation between the source and the relay that would result on the optimal BER performance. In the previous chapter, we used a different approach for power allocation which is based on studying the SISO decoding process at the destination's decoding process and allocating the power to avoid the causes of error propagation.

It was found in the previous chapter that the SISO TPC decoder performs better when all the bits of the EBCH codewords have equal SNRs. We used this observation to derive this simple power allocation such that the received SNRs at the destination of the two parts of the code are equal. Then we used the channel attenuations between the two transmission nodes and the destination to calculate the require bit transmission energy that result on the SNR balance under the condition that the total transmission energy from the two nodes does not exceed the available transmission power per TPC codeword.

The result were obtained on the previous chapter shows a tremendous improvement on the BER performance at the destination. The BER performance at the relay was also improved because more power is allocated to the source and therefore the inter-user channel quality was improved.

The power allocation criterion can be solved by optimizing the power consumption in terms of a convex function, such as minimizing the BER, the outage probability or to maximize the capacity, etc. On other words, the optimal power allocation is based on an optimization problem that depends mainly on an objective convex function. Examples of power optimization can be found in [66,67], where the objective function is a derived approximate Symbol Error Rate (SER) for cooperative network with AF relaying. The objective function can also be to minimize the outage probability or to maximize the capacity. In [68], optimal power allocation is solved subject to minimizing the outage probability for DF system with diversity. The optimum power allocations in AWGN channel for AF and DF relaying protocols in parallel relay networks were derived in [69] subject to increasing the capacity, whereas, the papers [70,71] used a Rayleigh fading channel model, where the instantaneous Channel State Information (CSI) is available at the transmitters, to develop various allocation problems for a three-nodes network with cooperative diversity system.

In this chapter, we investigate the extent of attainable improvements that can be achieved in the DTPC cooperative coding system proposed previously, by searching for the optimal power allocations based on the position of the relay. Rather than assigning equal power to the source and the relay, as done previously in chapters 3 and 4, we use optimization algorithms to find the power allocation that results in the BER performance at the destination's receiver. We also compare in this chapter the performance of the DTPC system with optimal power allocation to the performance of the DTPC system employing the power allocation criterion that is proposed in chapter 5 and with the non-cooperative system.

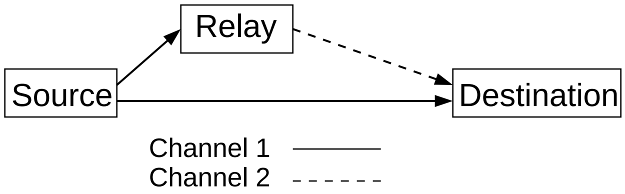


Figure 6.1. Cooperative system model with one source, one relay and one destinations

6.2 System Model

We will consider the same system model used in chapter 5 for testing the proposed power allocation. However, in this chapter, we use variable power allocation between the source and the relay at each location of the relay, and use an optimization method to search for the optimal power allocation that results on the lowest BER.

Without loss of generality, we consider a wireless network consisting of one source ‘s’, one destination ‘d’ and one relay ‘r’. The channels between the three terminals are assumed block faded. The three nodes (source, relay and destination) are assumed to have one antenna and work in half duplex mode, i.e., they can either receive or transmit at any instant of time. Therefore, a complete transmission of TPC codeword can be conducted in two phases. A simple illustration of the system model used in this chapter is shown in Fig. 6.1. All the channels connecting the three nodes are assumed independent. We assume that the attenuations on these three channels remain constant during at least the transmission of a frame, and the receivers have full knowledge of this attenuation, so that the channels can be assumed to be AWGN. The channel are characterized by the variance of Gaussian noise of the channel, and the attenuation between any two nodes, which is governed by the free space propagation of the signals. We assume that any receiving node has perfect knowledge of the channel attenuation and variance via channel estimation.

The TPC code matrix used in this chapter is constructed using two EBCH systematic linear block codes $\mathcal{C}_1$ and $\mathcal{C}_2$ with parameters (n_1, k_1, δ_1) and (n_2, k_2, δ_2), where $n_i, k_i,$ and δ_i are codeword length, input information block length, and minimum Hamming distance of the linear code $\mathcal{C}_i$, respectively. The matrix of the product code is obtained by serial concatenation of the two linear block codes and having an intermediate matrix transpose between the two encoders. This is done by arranging the information bits in $k_2 \times k_1$ matrix and then encoding the k_2 rows using code $\mathcal{C}_1$ and then encoding the resulting n_2 columns using code $\mathcal{C}_2$ to finally obtain a $n_2 \times n_1$ matrix. The resulting product code has the new parameters $N = n_1 \times n_2$, $K = k_1 \times k_2$, $\Delta = \delta_1 \times \delta_2$. In this paper, we assume that the two component codes are identical and they have the parameters $n_1 = n_2 = n$, $k_1 = k_2 = k$ and $\delta_1 = \delta_2 = \delta$.

As in the previous chapters, the source generates the EBCH block coded sequences and transmits them to the destination and the relay in the first phase of transmission. In our work, we considered both the DF and SDF protocols at the relay. The relay receives and decodes the transmitted sequences over the inter-user channel and arranges k-codewords in rows. Then the relay uses the second component code to encode the transpose of the row matrix (encoding across the columns). The parity of the second encoding, formed in a matrix of dimension $(n - k) \times n$, is separated and transmitted to the destination in the second transmission phase. Based on the used forwarding protocol, the relay will either transmit soft encoded information or hard encoded information if the SDF or DF relaying protocol used, respectively. The destination constructs a complete TPC by matrix concatenation of the two received matrices from the source and the relay and then conducts a turbo product decoding.

We aim to find in this chapter the optimal power allocation for the cooperative system when the attenuations on the direct and the relay channels are not equal. The channel attenuations can be caused by the block fading coefficients or by the free space propagation. The two channels are assumed to be statically independent, and therefore, the fading coefficients are independent. Moreover, the relay and the source can be randomly placed with respect to the destination, meaning that the separation distances between the destination and the two transmitting nodes can be different, and therefore, the attenuations can also be different.

One way to find the optimal power allocations for the cooperative system is by conducting a comprehensive search at all possible SNR values for the three channels, namely, direct, inter-user and relay channels. However, this will extremely complicate the optimization problem. An Alternate simple method is to use the line experimental setup to model the system. This model assumes the relay is located on the line connecting the source with the destination as in Fig. 5.2.

The destination receives two signals from the source and the relay during the two time slots. The two signals can be generally expressed as follows:

$$\begin{aligned} y_d[2k-1] &= \sqrt{E_s}\alpha_{sd}[2k-1]x_s[2k-1] + n_{sd}[2k-1] && (6.2.1)\\ y_r[2k-1] &= \frac{\sqrt{E_s}}{(1-\lambda)^2}\alpha_{sr}[2k-1]x_s[2k-1] + n_{sr}[2k-1] && (6.2.2)\\ y_d[2k] &= \frac{\sqrt{E_r}}{\lambda^2}\alpha_{rd}[2k]x_r[2k] + n_{rd}[2k] && (6.2.3) \end{aligned}$$

The subscripts 's', 'r' and 'd' are used to denote the source, the relay and the destination nodes, respectively. The received signal at node j is y_j while x_i is the transmitted signal from node i. The variable k denotes the time slot. The channel between the two nodes i and j has AWGN noise n_{ij}, and channel attenuation α_{ij}. E_s and E_r are the transmit energy/bit for the source and relay, respectively.

The proposed power allocation from chapter 5 suggests that the two parts of the DTPC matrix should be received with equal SNRs at the destination to minimize the error propagation in the decoding process, i.e.

$$SNR_{sd} = SNR_{rd}$$

As appears in the results, this simple method to allocate the power proves that higher BER performance can be attained from the distributed coding system, and the capacity of the relay channel is more efficiently utilized than the fixed power allocation. Since the total power should be divided between the source and the relay as $P_s + P_r = P$, where P_s and P_r are the source and the relay total transmission powers, respectively, and P is the total transmission power for one TPC codeword. Therefore, it was found in the last chapter that the energy/bit allocation for the relay (E_r) and the source (E_s) should be given by:

$$\begin{aligned} E_r &= E\frac{n^2}{\frac{kn}{\lambda^2} + (n-k)n} \\ E_s &= E\frac{n^2}{kn + \lambda^2(n-k)n} \end{aligned}$$

to meet the power allocation criterion give above so that the SNR of the two received parts of the code are equal. Where in this equation, E is the energy/bit for the original non-cooperative system, λ is the relay location on the line model.

If we use the factor α to represent the portion of the power allocated to the source from the total power, then

$$P_s = \alpha(\lambda)P, \qquad P_r = (1-\alpha(\lambda))P \tag{6.2.4}$$

Therefore, for the proposed power allocation in chapter 5, α can be given by:

$$\alpha_{PA} = \frac{kn}{kn + \lambda^2(n-k)n} \tag{6.2.5}$$

6.3 Power Optimization

In this section we apply optimization rules to the DTPC cooperative system with SDF relaying protocol presented in [72] to search for the maximum attainable performance using the line experimental setup. Our main target in this chapter is to minimize the final BER for the three transmission channels via optimal power allocation between the source and the relay. It is well known that the performance of any communication system, specifically cooperative systems, can be improved by relaying with optimal power allocations [56]. Therefore, we assume that a maximal overall transmit power from the source and the relay is fixed and is equal to the same power required to transmit the complete TPC codeword from the source to the destination in the non-cooperative scenario. Then, the overall total transmitting power is to be optimally shared between the source and the relay, so that the power is efficiently utilized to gain the maximum performance possible for the DTPC. Basically, we optimize the power allocation by minimizing bit-error probability at the destination.

Since the main target for power optimization is to reduce the probability of error at the destination after the turbo product decoding stage, the target function for our optimization problem is therefore the BER after the decoder. However, there is no exact expression available to model the probability of error after the decoder, but one way to characterize this unknown function is by the empirical function given by:

$$BER = f(\alpha, \lambda, P) \tag{6.3.1}$$

This relation is monotonically decreasing function with respect to the power P, so to optimize the power at location λ, we have to find the values of α that results in the minimum BER. Thus, the optimization problem is reduced to a one dimensional problem with only one variable parameter α.

6.3.1 Optimization Algorithm Requirements

In this chapter, we assume that the target function in (6.3.1) has a global minimum with respect to α. This assumption is based on the fact α can take values between 0 and 1, and the BER at the the two extreme values of α, i.e. 0 and 1 is much higher than the BER for any arbitrary value of α between the two limits. When $\alpha = 1$, this means that all the power is allocated to the source, and therefore, the vertical parity is transmitted with 0 power. On the other hand, when $\alpha = 0$, all the power is allocated to the relay, while the source has 0 transmission power. In the two extreme cases, the performance of the cooperative system is far worse than the performance of the system with equal energy/bit allocation (the scheme used in chapters 3 and 4), consequently, we assume that the function has the most global minimum located between the two extreme points.

We want to determine the power allocation (the value of α) that optimizes the systems performance with respect to the power. Since the BER performance of the system depend on the statistics of the channels, the results from simulating the system is known to have margin error, hence, to overcome this limitation, we use repetition and then curve fitting to find the approximate power allocation that has the least square errors with respect to the observation points.

Another issue to consider when designing the optimization algorithm is that simulations errors could also lead the optimization algorithm to a wrong solution. This limitation is solved by the proposed algorithm using the sliding ball principle on a slope as in Fig. 6.2. If a ball slides from any peaks of the slope it will slid and will exceed the lowest point on the curve and will traverse more distance upward beyond the solution until it will stops and reverse it's direction of movement. This continues until the ball reaches it's steady state at the lowest point on the slope. The numbers on the balls in Fig. 6.2 indicates the positions of the ball when it reverses the sliding direction, where the number 1' indicates the starting point. Note that if a sliding ball passes across a small bulge, it pass this bulge and continue sliding until it reaches an uphill.

The required optimization algorithm should take in consideration the aforementioned characteristics of the DTPC simulator and should be designed to minimize the run time and the complexity of the algorithm. We set the optimization algorithm to

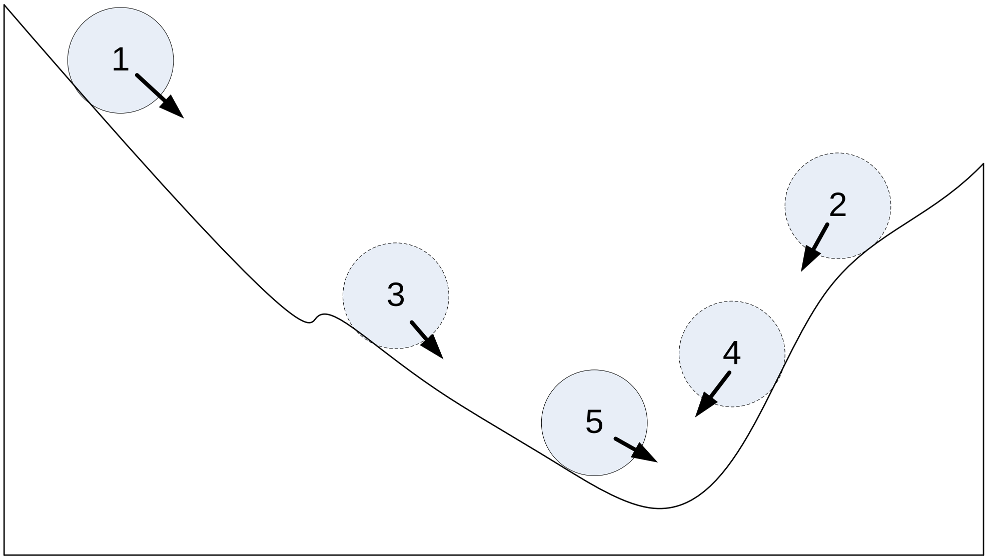


Figure 6.2. The principle of sliding ball used in designing the optimization algorithm

work on bit error rate level close to 10^{-3} bit errors/frame to have accurate results with lower number of repetitions (the number of transmitted and received frames for a single SNR and λ pair).

6.3.2 Optimization Algorithm

For every relay location λ and for a fixed SNR that meets the target BER level (as explained in the previous paragraph), the optimization algorithm starts from one of the boundary points of the search segment. In our optimization analysis, we use smaller range for α instead of the (0-1) range to avoid complexity and long running time, since the performance of the cooperative system is known experimentally to degrade severely when $\alpha < 0.7$. The algorithm calculates the step size and the sign based on the length search segment and the number of steps. Similarly to the sliding ball principle, for each step the algorithm compares the current bit error rate of the decoding result with the previous step. If this BER is smaller than the previous one, then it continues to the next simulation step. If the decoding error rate is larger than result in the last step, then the algorithm compares the previous step result with the result two steps back: if the one step back result is also larger than the BER result in the two steps back, then it sets new boundaries (search segment) and step size, otherwise, if the result one step back is smaller than the result two steps back, then it continues to the next simulation step.

The optimization algorithm continues on steps until it reaches two consecutive points on the upward direction of the curve (i.e. last two results of BER are larger then previous step) or until it reaches the boundary of the curve segment. In both cases, new search segment is determined from the length cover by the two steps before the current step. The step size is calculated from the length of the search segment and the required number of steps.

The optimization algorithm used to find the value of $\alpha(\lambda)$ for each relay position λ across the line model is shown in 3. We used the accuracy threshold Th as stopping criterion to determine when the algorithm has approached to the solution with a predetermined accuracy level. The number of steps $NSteps$ is the number of of sections that the search segment is divided to. As noted from the algorithm, the step size reduces every time when a new search segment is found. The new search segment is determined to be the the last two sections coming before the current segment at which the condition to find new search segment is satisfied. This is illustrated in Fig. 6.3, in which the current search segment is divided in 6 sections (number of steps $NSteps$). When the optimization algorithm reaches the 5th step on the curve, the condition of new search segment is satisfied and therefore it breaks at this step and finds the new search segment to be the two sections 2 and 3. Therefore, the new α_{start} and α_{end} are the points 4 and 2 on the previous search segment, respectively. If the algorithm fails to find the solution and reaches one of segment boundaries, then the new search segment will be updated to be the last two sections in the current search

```
Input: DTPC simulator with inputs $\alpha$, $\lambda$ and $SNR$ and output $BER$
for $\lambda : 0$ to $1$ do
    Set $SNR$ such that the BER is around $10^{-3}$, search segment boundaries $\alpha$:
    $\alpha_{end}$ and $\alpha_{start}$, Accuracy threshold $Th$, number of steps to $NSteps$;
    Calculate step size $Step = (\alpha_{start} - \alpha_{end})/NSteps$;
    while $\alpha_{start} - \alpha_{end} > Th$ do
        Set $\alpha = \alpha_{start}$;
        for $j = 1$ to $NSteps$ do
            repeat
                Simulations given the inputs: $\alpha$, $\lambda$ and $SNR$;
            until Maximum Number of Frames;
            The output is $BER[j]$;
            if $j > 2$ then
                if $BER[j] > BER[j-1]$ AND $BER[j-1] > BER[j-2]$ then
                    Set $\alpha_{start} = \alpha + Step$,
                    Set $\alpha_{end} = \alpha + 3 * Step$,
                    Set $Step = (\alpha_{start} - \alpha_{end})/NSteps$;
                    Break;
                end
            end
            if $j < NSteps$ then
                Set $\alpha = \alpha - Step$
            end
            if $j = NSteps$ then
                Set $\alpha_{start} = \alpha$,
                Set $\alpha_{end} = \alpha + 2 * Step$,
                Set $Step = (\alpha_{start} - \alpha_{end})/NSteps$;
            end
        end
    end
    Increment $\lambda$ by 0.1;
end
Output: $\alpha(\lambda)$ at which the simulator yields minimum BER
```

Algorithm 3: Power optimization algorithm

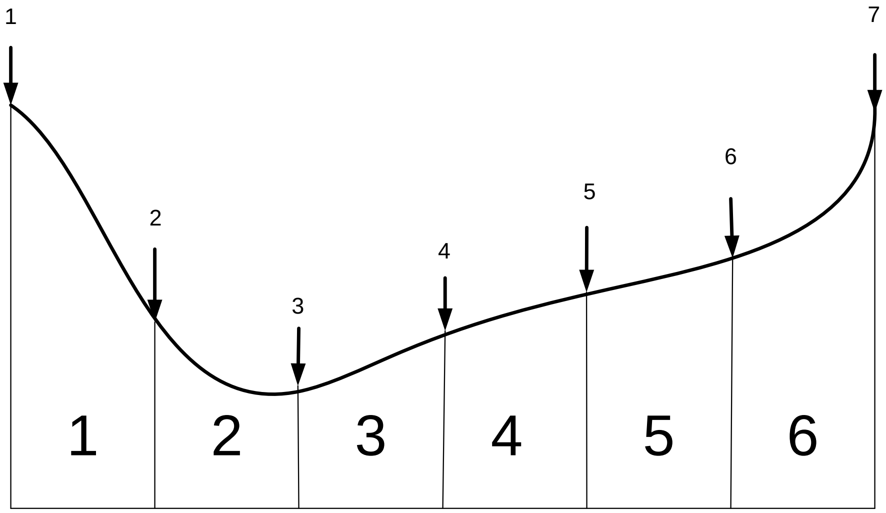


Figure 6.3. Illustration to show how the optimization algorithm finds new search segment

segment.

6.4 Results and Discussion

In this section, we first present the results of the optimization algorithm, i.e. $\alpha(\lambda)$, then we test the power optimization parameters on the DTPC system to see the effectiveness of the optimal power allocation. In the last part of this section, we compare the result obtained in this chapter to the results obtained on chapter 5 to check the validity of our proposed power allocation algorithm.

6.4.1 Optimization Results

For our optimization algorithm, the number of steps is set as $NSteps = 6$, maximum number of transmitted frames = 2000, accuracy threshold $Th = 1 \times 10^{-5}$, starting value for $\alpha_{start} = 1.0$, and for $\alpha_{end} = 0.7$, and variable step size that is always equal to

$$\frac{\alpha_{start} - \alpha_{end}}{NSteps}$$

λ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
SNR [dB]	2.5	2.2	2.0	1.3	1.1	1.1	1.3	1.4	1.6	1.8	2.2

Table 6.1. SNR values used on the optimization algorithm for each λ step

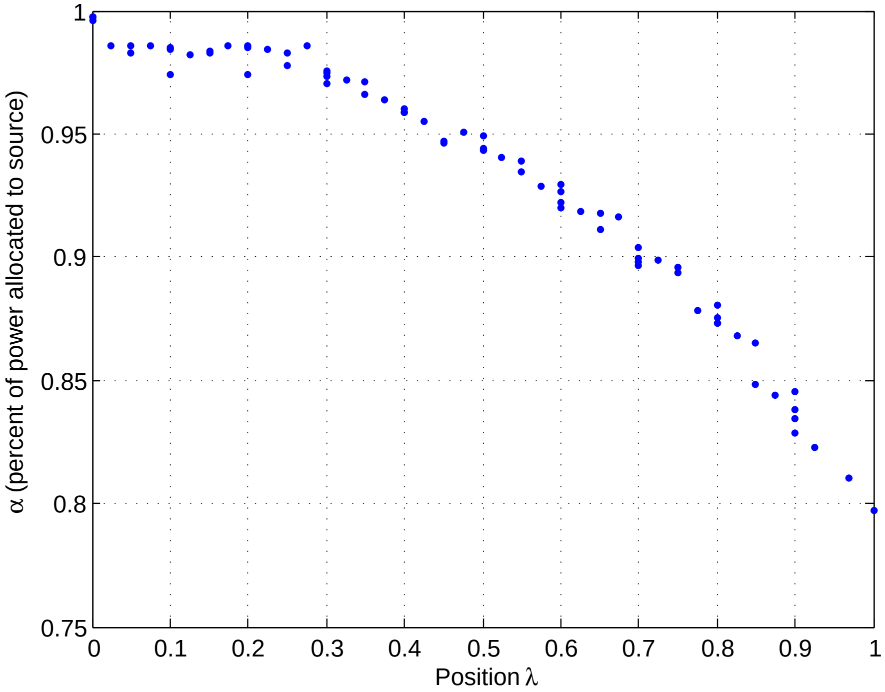


Figure 6.4. Power Optimization Results for α in (6.2.4).

We also set the SNR values at each λ step to the values in table 6.1 to ensure that the optimization algorithm works around the 10^{-3} BER level.

After running the optimization algorithm, it approaches the solution within about 50 iterations. The results for several runs of the simulations for the SDF relaying protocol is shown in Fig. 6.4. The figure shows the location of the relay on 0.025 steps versus the power optimization factor α, which is the percent of total power allocated to the source.

The higher value of α means that more power is allocated to the source. For low values of λ, when the relay is closer to the destination, it is better to allocate most of the transmission power to the source as the optimization results suggest in Fig.

6.4. This result agrees with our expectation that less power is needed for the relay transmission, because of it's proximity to the destination. On the other hand, when the relay get closer to the source, more power is required to transmit the signal to the destination to compensate for the attenuation loss resulted from the free space propagation.

6.4.2 Performance of DTPC System with Power Optimization

We used polynomial fitting to approximate the resulted points in figure 6.4 to a k-th degree polynomial. We choose to approximate the results to the polynomial with a degree $k = 7$ because it is the lowest degree that will result in least-squares errors less than or equal to 10^-3. The polynomial fitting uses the least square errors as a criterion to best fit the observation points to a k-th degree polynomial:

$$y_k(\lambda) = a_0 + a_1\lambda + a_2\lambda^2 + \cdots + a_k\lambda^k$$

The coefficients a_i for $0 \leq i \leq k$ for the DTPC system with SDF relaying protocol are found to be as shown in table 6.2.

a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7
0.9966	-0.4151	4.3207	-19.8401	43.3002	-49.2816	27.8726	-6.1564

Table 6.2. The 7th degree polynomial coefficients found using polynomial fitting

The resulted polynomial function is plotted in Fig. 6.6 against the parameter α, together with the actual observation points when λ changes from 0 to 1. We used the optimal power allocation values for every location of λ using the polynomial that has the coefficients in table 6.2 in the DTPC cooperative system. The BER performance results for several relay positions is plotted in Fig. 6.5 against the reference SNR of the direct channel. We also plotted in the same figure the non-cooperative case for

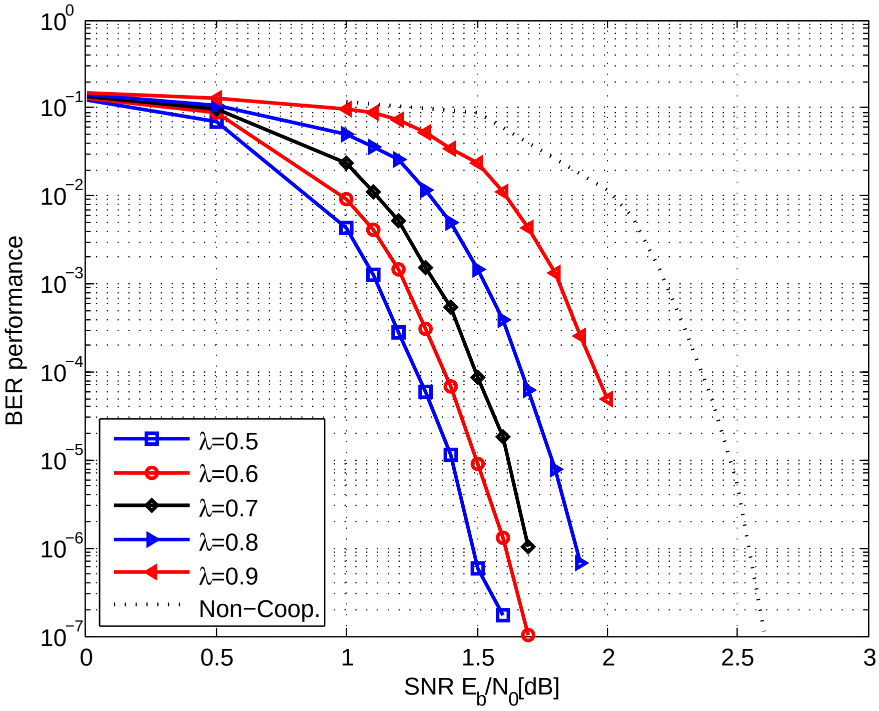


Figure 6.5. The BER performance of the DTPC with SDF protocol using Optimal Power Allocation versus the SNR.

comparison with the power optimized distributed coding system. The system shows performance gain ranges between 0.5 dB to 1.2 dB when the relay is placed between the source and the mid-point $0.5 \leq \lambda \leq 1$ on the line model that is connecting the source and the destination. When the relay is closer to the destination $\lambda \leq 0.3$, the distributed system still exhibits a performance gain but with error floor when $SNR \to \infty$.

6.4.3 Comparison with Power Allocation Criterion

In this section we compare between the optimal power allocation and the proposed power allocation in chapter 5. The criterion for power allocation as proposed in the previous chapter is based on receiving all the code parts at the same SNR. To do so, we allocated the excess power (the surplus) from the relay to the source, therefore, we decreased the SNR on the relay channel and increased the SNR on the direct channel

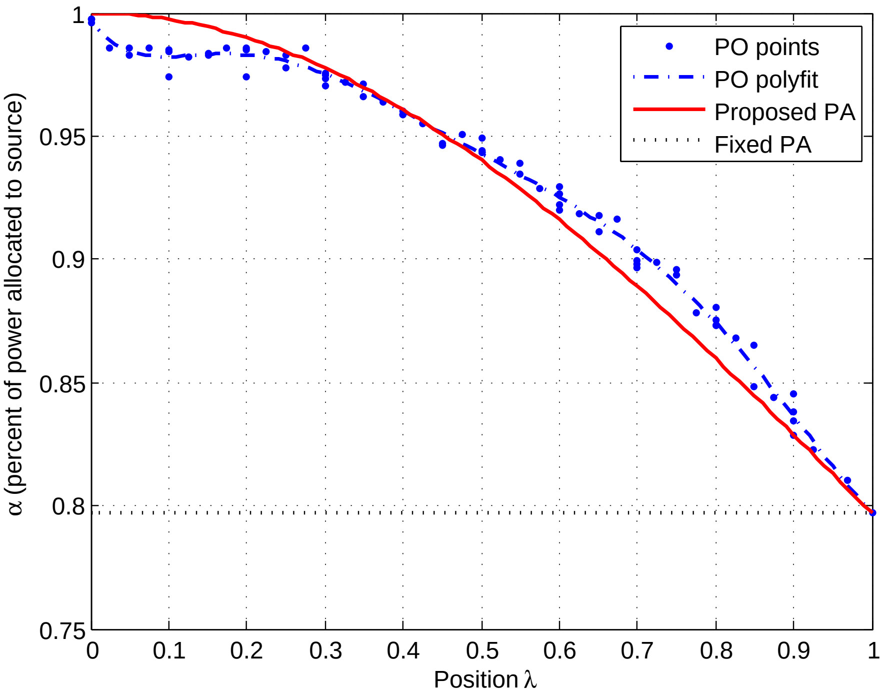


Figure 6.6. Using curve fitting to plot α for the Power Optimization results and compare it to the earlier proposed Power Allocation and Constant Power Allocation for the SIR-DTPC system.

to have them equal to each other. This power allocation criterion shows to be a very simple and effective solution for the distributed systems, however it did not consider the errors made by the relay on the decoding process before re-encoding the message signal and transmitting the resultant parity. Since the decoder of the destination handles all the bits according to their SNR, it assumes that all the bits have equal confidence when it receives the two parts of the code over the same SNR. Since the second part received from the relay is not perfect estimate of the vertical parity, it has some decoding errors and the decoder should not consider the confidence of this part to be equal to the confidence of the first part. Alternately, the power allocation algorithm should take this errors in consideration and allocate the power in a way that decreases the errors at the relay and increases the SNR of the first part of the code to be decoded at the destination with more confidence.

In Fig. 6.6, we compare the percent of power allocated to the source in the

SIR-DTPC system from the available transmission power for the three schemes: the constant power allocation (dotted black line), the equal-received energy power allocation scheme (solid red line) and the optimized power scheme (dashed-dotted blue line). The blue dots are the observed points from the optimization processes, which are approximated to the dashed-dotted line using curve fitting. The constant power allocation uses equal energy/bits for all the bits transmitted from the source and the relay. Therefore, the source share is equal to the percent of bits transmitted from the source, which is equal to kn to the total number of bits transmitted from the both the source and the relay n^2, which is equal to:

$$\alpha_{CP} = \frac{kn}{n^2} = 0.797$$

where we substituted for k and n their values that we used here in our simulation model which are 51 and 64, respectively. For the power allocation scheme that was proposed in the previous chapter, α was derived in (6.2.5).

It can be clearly seen from the results in Fig. 6.6 that the optimization result agrees with the equal-received-energy criterion power allocation proposed before. The general behavior of the power optimization scheme is approximately lead to an equal received energy at the destination for the two code parts. Moreover, the optimal power allocation for the source when $0.4 \leq \lambda \leq 0.9$ is slightly larger than equal-received-energy line. On this range, the cooperative coded system exhibits the highest gains over the non-cooperative system, and therefore as aforementioned the optimal power allocation is when SNR of the direct channel is higher than the SNR of the relay channel to make it with more confidence on the decoding process.

We also compare the BER performance of the three schemes: the constant power allocation (dashed-dotted black line), the equal-received-energy power allocation (dotted blue line) and the optimized energy power allocation methods (solid red line) when

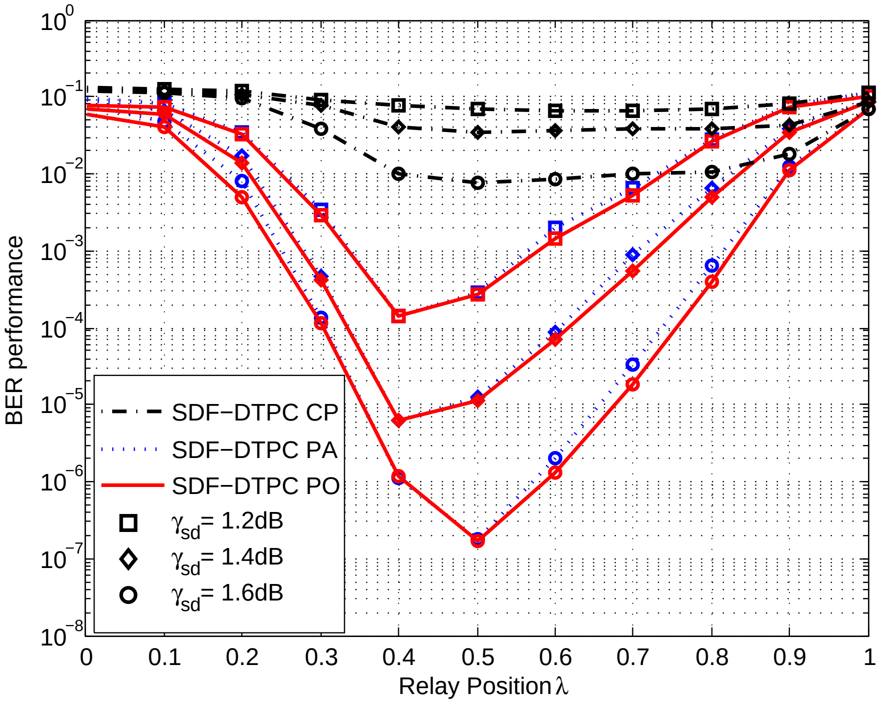


Figure 6.7. Comparison between the BER performance of the DTPC with SDF protocol using Optimal Power Allocation, Proposed Power Allocation and Constant Power Allocation versus the location of the relay.

the relay position on the line model changes from 0 to 1 for different values of direct channel SNR of the reference system. It is noted that the performance of the optimized power allocation is slightly better than the proposed power allocation for high and low values of λ. The results obtained for the optimized power allocation for the DTPC system proves the validity of the earlier proposed power allocation scheme in the previous chapter, and shows that performance attainable from this allocation scheme is very close to the maximum performance possible for the DTPC system.

6.5 Conclusions

We aimed in this chapter to find the optimal power allocations for the SDF-DTPC cooperative system. For this purpose we proposed and used a new optimization algorithm that is based on the principle of a sliding ball on an inclined surface. The results obtained from the optimization algorithm were very close to the power allocation criterion that is proposed in chapter 5 which is based on having equal received SNRs for all the parts of the distributed code. We tested the outcomes of the optimization algorithm on the distributed coded system for every location of the relay on the line model. The BER performance results obtained from plugging the optimization results in the DTPC system shows a very small improvement over the equal-received-energy based power allocation proposed earlier. The results obtained in this chapter emphasize on the validity of our proposed power allocation criterion on the previous chapter and show the maximum possible attainable performance from the DTPC cooperative system.

Chapter 7

Joint Distributed Space-Time Block Coding with Distributed Turbo Product Code (DSTBC-DTPC)

7.1 Introduction

Simulation results from the previous chapters show that the DTPC cooperative system is largely affected by the relay channel errors. Even when the relay makes no errors, the performance is seem to degrade when the SNR of the relay channel is relatively low. This can be clearly seen when the relay is positioned closer to the source, i.e. $0.5 < \lambda < 1.0$. Therefore, we aim in this chapter to improve the performance of the DTPC system even more by enhancing the conditions of the relay channel. The SNR of the relay channel can be improved by employing diversity to transmit the second phase of data, which are the parity information about the first transmission phase as generated by the relay. However, the use of diversity usually means that the system will be less spectral efficient, in the case of time or frequency diversity, or have more hardware cost, e.g. additional antennas at the receiver in the case of spatial diversity. An alternative solution is to use the Space-Time Block Code (STBC) to have transmit diversity at the second transmission phase. This solution fits perfectly on the design of DTPC code and does not evolve the need of additional antennas or using the spectrum less effectively, as can be explained more in details in this chapter.

Diversity techniques are known to offer an efficient solution for combating the fading in wireless communication environments. Time, frequency, and spatial diversity are the three main forms of diversity techniques [73–76]. Time diversity is a form of channel coding, and used mostly with time interleaving when the channel is on slow fading conditions. The receiver receives replicas of the transmitted signal in the form of redundancy in time domain. In frequency diversity, the same signal is transmitted at different frequency channels that have different fadings. Therefore, a better quality of the signal can be obtained if the different versions are combined. Antenna diversity (i.e., space diversity) is the most practical and widely used diversity technique in wireless telecommunication systems. The MIMO systems with multiple receiver and transmit antennas have proven to improve the received signal quality through diversity to a great extent [36]. Each transmitter-receiver antenna pair may provide an independent path from the transmitter to the receiver. Multiple independent faded replicas of the transmitted signal are obtained at the receiver side using proper processing, thus creating spatial diversity. Therefore, antenna diversity is able to achieve higher spectral efficiency in MIMO systems as compared with SISO systems through the use of Spatial Multiplexing (SM). However, these improvements come at the cost of requiring multiple radio frequency front ends at both the transmitter and the receiver. Furthermore, the size of mobile devices may limit the number of antennas that can be deployed.

As an alternative, ST coding is used in wireless communications to transmit multiple replicas of a signal across a number of antennas and to exploit the various received versions of the data by using additional processing at the receiver to gain spacial transmit diversity [77]. The advantage of ST coding is that there is no need to have multiple antenna at the destination to have different versions of the signal. Therefore, reducing the cost of the handset devices, and averting the design of the mobile device from being bulky.

Previous works have proposed a distributed ST coding schemes on cooperative networks, e.g. [78, 79], by using the spatially distributed of relay nodes as antennas to assist the transmitter on providing transmit diversity. In order to construct the distributed ST coding, the source has to transmit the message signal to the relays in the first phase, and then the relays together with the source transmit the message again in the second phase using STBC coding [78]. While their results shows an improved performance compared to conventional systems, sending the message twice in the first and second transmission phases decreases the spectral efficiency. Moreover, they have not shown a solution to the increase power requirements, when more relays are participating in the transmission process.

In this chapter, we propose to use the distributed STB coding to transmit the second part of the distributed TPC on the relay and direct channels. The source and the relay share their single antennas to create a virtual transmit array to transmit the generated parity on the second phase toward the destination. The main concentration of this chapter is on enhancing the channel conditions for the relay channel. The relay on our proposed distributed system may use two forwarding strategies, the DF and the SDF protocols, which are based on decoding the message signal and re-encoding using a second component code to produce vertical parity. In the second transmission phase, the relay and the source working synchronized using space-time coding to transmit the second part of the TPC matrix, the vertical parity. The destination receives the message signal transmitted over the direct channel and the two versions of the second part of the TPC matrix. The two signals received on the second transmission phase then are STB decoded and the parity bits are detected and sent to the TPC decoder to join it with the first part of the TPC code.

The main feature of the proposed architecture is twofold. Firstly, the spectral efficiency is improved. The second transmission phase from the source and relay nodes is done simultaneously for the parity of the first transmission phase. Hence, no bandwidth is wasted to re-transmit the same message again. Secondly, the gain of transmission diversity is obtained without the need of additional external antennas.

7.2 System Model

Coded cooperation is performed by sending two codewords via two independent channel paths. The basic idea is to improve the overall performance of cooperative systems through diversity. The proposed configuration for the Joint distributed STBC with DTPC is shown in Fig. 7.1. In this model, instead of using a centralized turbo product coded system originated from the source node only, we propose a distributed coding scheme at both the source and the relay nodes, as we have illustrated before in the previous chapters. The encoding process is divided over two transmission phases, so the overall code received at the destination is constructed from the two parts received over the direct and the relay channels. To further improve the overall performance through diversity, the coded cooperation operates by sending the second part of the TPC matrix using STBC encoding. The source generates the exact vertical parity by simply transposing the horizontally encoded message signal and EBCH encoding to obtain the vertical (columns) parity. On the other hand, the relay uses the corrected message signal, which was transmitted by the source during the first transmission phase, to generate the vertical parity for the encoded message signal. The vertical parity generated at the relay is an estimate for vertical parity generated by the source. The relay uses the same encoding methods as explained in chapters 3 and 4 to obtain either hard or soft incremental redundancy, respectively.

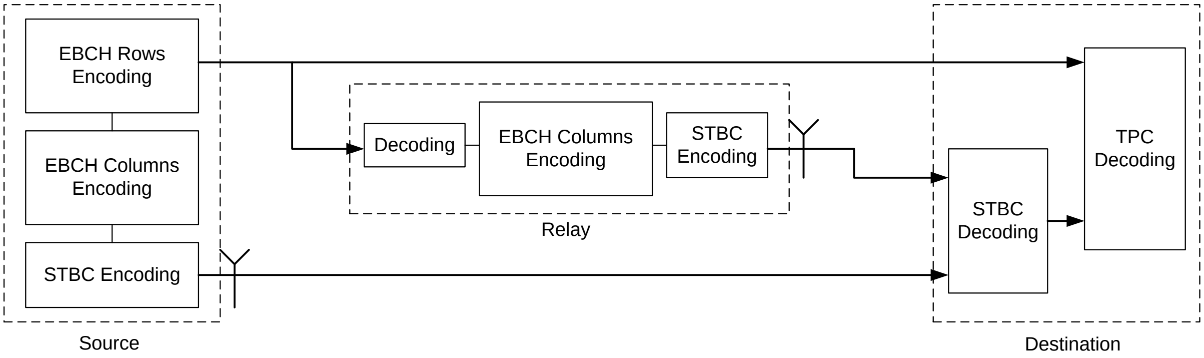


Figure 7.1. Proposed Joint Distributed STBC Distributed TPC configuration

Assuming that the source and the relay are perfectly synchronized, the second part of the code is transmitted using Alamouti encoding from both the source and the relay. The STBC encoding benefits from the single antennas available at the two nodes, the source and the relay, that makes up the virtual two antennas array required to transmit the STBC signal using the Alamouti coding [77]. Both the source and the relay are transmitting using Half-Duplex mode, where the source uses it's single antenna to transmit the first block of the code, and then to transmit the second block of the code using the STBC encoding.

The source transmits to the relay and the destination during the first transmission phase, EBCH encoded codewords, which are received at the destination and the relay, after passing through fast faded with AWGN channel, can be modeled as:

$$Y_{sd1}[k] = \sqrt{E_s} h_{sd1}[k] X_1[k] + n_{sd1}[k] \tag{7.2.1}$$

$$Y_{sr1}[k] = \frac{\sqrt{E_s}}{(1-\lambda)^2} h_{sr1}[k] X_1[k] + n_{sr1}[k] \tag{7.2.2}$$

where $Y_{sd1}[k]$, $Y_{sr1}[k]$ are the received signals at the destination and the relay, respectively, during the first transmission phase, E_s is the energy/bit used to transmit the source bits, k is the time index for each transmitted symbol, $X_{s1}[k]$ is the source modulated symbol for the sequence of codewords in the first transmission phase, $h_{ij1}[k]$ is the complex Rayleigh coefficient with mean and variance equal to 1. The additive noise n_{ij} is complex Gaussian with zero mean and variance of 0.5 in each dimension. It is assumed that the channel remains constant for any two consecutive time slots. The parameter λ represents the relay position on the line model. It is assumed that the relay is located on the line connecting the source with the destination, so the line model propagation equations apply to the transmitted signals from the source and the relay. The relay receives the source's transmission and performs a channel compensation for the received signal, assuming that it has perfect channel estimates:

$$\hat{X}_{r1}[k] = \mathrm{Re}\left\{Y_{sr1}[k]h^*_{sr1}[k]\right\} \tag{7.2.3}$$

In the second transmission phase, both the source and the relay generates their own versions of the parity, and use the Alamouti code to transmit to the destination. We use the simple two-transmit antennas system and has the coding matrix:

$$C_2 = \begin{bmatrix} s_1 & -s_2^* \\ s_2 & s_1^* \end{bmatrix} \tag{7.2.4}$$

where $*$ denotes complex conjugate of the symbol. The columns in this matrix represent the time slot, and the rows represent the antenna used to transmit the symbols. In our work, we assume that the source is the virtual antenna 1, and the relay is the virtual antenna 2. So the complex symbols coming out of the source and relay antennas can be written as follows:

$$X_{s2} = [s_1 \quad - s_2^* \qquad s_3 \quad - s_3^* \qquad s_5 \quad - s_6^* \qquad \cdots \qquad] \tag{7.2.5}$$

$$X_{r2} = [r_2 \qquad r_1^* \qquad r_4 \qquad r_3^* \qquad r_6 \qquad r_5^* \qquad \cdots \qquad] \tag{7.2.6}$$

where the symbol s_j indicates the source generated symbol using the space-time coding for the vertical parity, whereas r_j is the relay generated symbol corresponding to the estimate of the vertical parity. Every time slot k, the source and the relay transmit simultaneously their STBC encoded symbol to the destination, i.e. the symbols $X_{s2}[k]$ and $X_{r2}[k]$ from the source and the relay, respectively. Therefore, on the second transmission phase, the received signals at the destination via a fading and AWGN channel can be expressed as:

$$Y_{d2}[k] \quad = \quad \sqrt{\frac{E_s}{2}} h_{sd2}[k] X_{s2}[k] + \sqrt{\frac{E_r}{2}} h_{rd2}[k] X_{r2}[k] + n_{d2}[k] \tag{7.2.7}$$

$$\tag{7.2.8}$$

where E_r is the energy/bit for the relay node, and $n_{d2} \sim N(0,1)$. The coefficients h_{id2} here also are fast Rayleigh fading complex coefficients with mean and variance equal to 1. The assumption that the channel coefficients remain constant for two successive time slots is also used for the second transmission phase.

The destination will receive the transmissions in the two phases, from the source only in the first phase, and a combined signal from the source and the relay on the second phase. We assume that the destination can obtain a perfect knowledge about the channel conditions between it and the two other nodes, i.e. the source and the relay. So the destination uses channel tap-compensation for the first part by applying the following for every received symbol:

$$\hat{X}_{d1}[k] = \operatorname{Re}\left\{ Y_{sd1}[k] h^*_{sd1}[k] \right\} \tag{7.2.9}$$

For the second part of the distributed code, the destinations applies Alamouti decoding for the two consecutively received composite signals to separate the two received symbols. In other words, to extract the two symbols on time slots $2k-1$ and $2k$, the destination has to apply the Alamouti decoding to the received signals in time slots $2k-1$ and $2k$. One particularly attractive feature of orthogonal STBC is that maximum likelihood decoding can be achieved at the receiver with only linear processing. To perform the STBC decoding we define the matrix H:

$$H = \begin{bmatrix} h[2k-1] & h[2k] \\ h^*[2k] & -h^*[2k-1] \end{bmatrix} \tag{7.2.10}$$

To solve for $\hat{X}_2[2k-1]$ and $\hat{X}_2[2k]$, we need to find the inverse of H. For a general matrix of size $m \times n$, the pseudo-inverse for the matrix H is defined as:

$$H^+ = \left(H^H H\right)^{-1} H^H \tag{7.2.11}$$

where ^{H} here represent the Hermitian of the matrix, which is equal to the conjugate transpose of the matrix. The term:

$$\begin{aligned} H^H H &= \begin{bmatrix} h^*[2k-1] & h[2k] \\ h^*[2k] & -h[2k-1] \end{bmatrix} \begin{bmatrix} h[2k-1] & h[2k] \\ h^*[2k] & -h^*[2k-1] \end{bmatrix} \\ &= \begin{bmatrix} \left|h[2k-1]\right|^2 + \left|h[2k]\right|^2 & 0 \\ 0 & \left|h[2k-1]\right|^2 + \left|h[2k]\right|^2 \end{bmatrix} \end{aligned} \tag{7.2.12}$$

Which is a diagonal matrix, therefore, the inverse is just the inverse of the diagonal elements, i.e.:

$$\left(H^H H\right)^{-1} = \begin{bmatrix} \frac{1}{\left|h[2k-1]\right|^2 + \left|h[2k]\right|^2} & 0 \\ 0 & \frac{1}{\left|h[2k-1]\right|^2 + \left|h[2k]\right|^2} \end{bmatrix} \tag{7.2.13}$$

Therefore, the estimates of the transmitted symbols at time slots $2k-1$ and $2k$ can be obtained as follows:

$$\begin{bmatrix} \hat{X}_2[2k-1] \\ \hat{X}_2[2k] \end{bmatrix} = \left(H^H H\right)^{-1} H^H \begin{bmatrix} Y_{d2}[2k-1] \\ Y_{d2}^*[2k] \end{bmatrix} \tag{7.2.14}$$

$$= \left(H^H H\right)^{-1} H^H \left(H \begin{bmatrix} X_2[2k-1] \\ X_2[2k] \end{bmatrix} + \begin{bmatrix} n_{d2}[2k-1] \\ n_{d2}^*[2k] \end{bmatrix} \right) \tag{7.2.15}$$

$$= \begin{bmatrix} X_2[2k-1] \\ X_2[2k] \end{bmatrix} + \left(H^H H\right)^{-1} H^H \begin{bmatrix} n_{d2}[2k-1] \\ n_{d2}^*[2k] \end{bmatrix} \tag{7.2.16}$$

Here, we used the symbol X_2 to represent the transmitted symbol from the source and the relay instead of the symbols X_{s2} and X_{r2}, assuming that the relay estimates of the parity bits are correct. We, however, do not use this assumption in simulations and estimate the parity bits by means of decoding and re-encoding at the relay, then the estimates of the parity in either hard (DF) or soft (SDF) form are transmitted to the destination.

7.3 Results and Discussion

To test the proposed system with the addition of distributed STBC encoding to our earlier proposed DTPC cooperative system, we also added multiplicative fading coefficients instead of the LOS channel used in previous chapters. Fast fading channel model with Rayleigh complex coefficients are used for all the channels connecting the three nodes. In our simulations, the multiplicative complex Rayleigh coefficient is formed as follows:

$$h = \frac{1}{\sqrt{2}}(W + jZ) \tag{7.3.1}$$

where both W and Z are independent and identically distributed (iid) random variables with normal distribution with zero mean and unity variance, i.e. $N(0, 1)$, so that h has Rayleigh distributed amplitude with variance equal to 1.

In our simulations, we assumed that the transmission powers from both the relay and the source are equal, i.e. $E_s = E_r$. We also assumed that the power required for the STBC encoding, is the same power required to transmit the second part of the code. On other words, the power required to transmit the vertical parity from the source (or the relay in the cooperative coding case), is divided by 2 to guarantee fair comparison with the non-cooperative case and with the distributed coded system without the distributed STBC encoding stage. This is also shown in (7.2.7)

In this chapter, we assume that all the channels experience fast fading with AWGN at the receiver's front end. We also assume that the channel remains constant for the two successive time slots in the second transmission phase when the distributed STBC encoding is used in the system. We compare the performance of our proposed system to the cooperative system employing the DTPC without the distributed STBC and to the non-cooperative system in which the source performs all the TPC encoding (rows and columns) and then transmits the encoded symbols on the direct channel only without using relay to support the transmission.

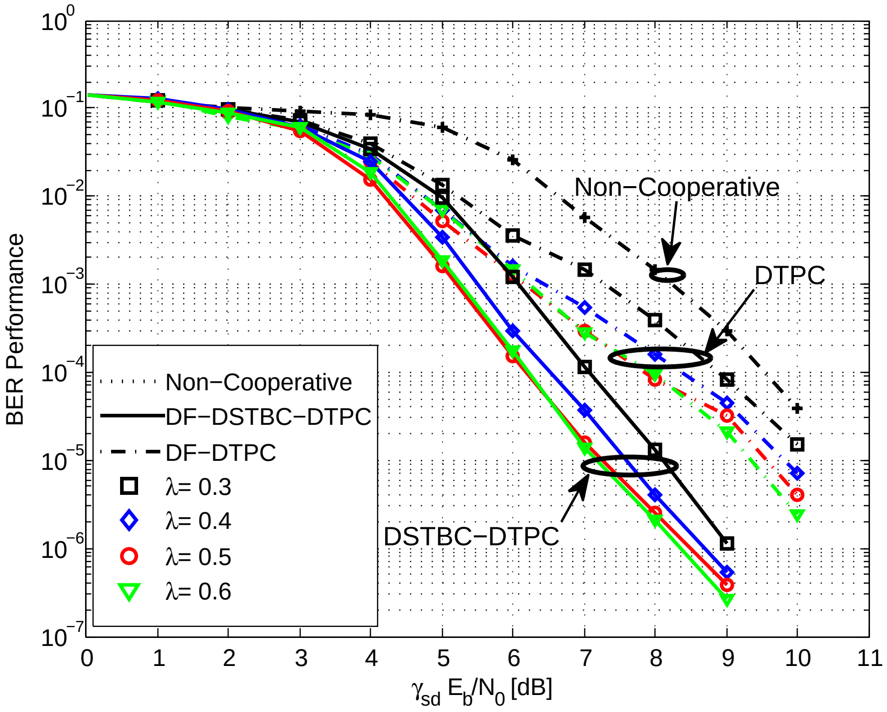


Figure 7.2. BER performance of the joint Distributed STBC DTPC encoding using hard decoded and forward DF protocol for relay positions $3 \leq \lambda \leq 6$

On the cooperative system employing only DTPC encoding, we assume that the relay and the destination have full knowledge about the channel conditions using channel estimation. The relay and the destination perform channel compensation for the symbols transmitted over the first transmission phase using the method in (7.2.3) and (7.2.9), respectively. In the second transmission phase, only the relay transmits to the destination, and thus the destination performs channel compensation on the received symbols from the relay using:

$$\hat{X}_{d2}[k] = \mathrm{Re}\left\{Y_{rd2}[k]h^{*}_{rd2}[k]\right\} \tag{7.3.2}$$

Also in the non-cooperative mode, the destination performs channel tap compensation by using the channel estimates to compensate for attenuations on the received symbols from the source in a similar manner to (7.2.3), (7.2.9) and (7.3.2).

The results for the proposed distributed system using DF forward protocol at the

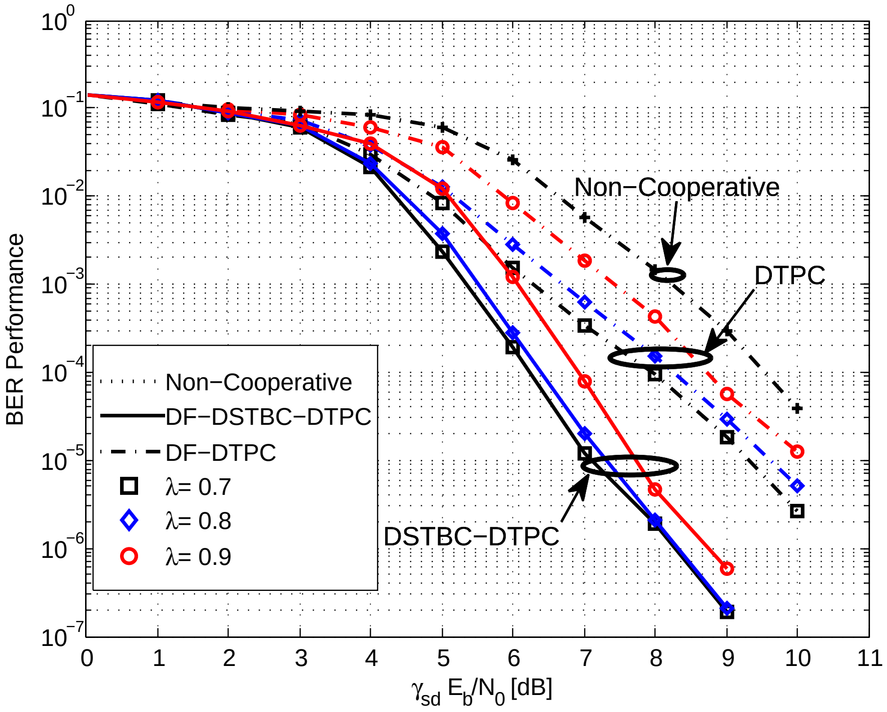


Figure 7.3. BER performance of the joint Distributed STBC DTPC encoding using hard decoded and forward DF protocol for relay positions $7 \leq \lambda \leq 9$

relay is plotted in Fig. 7.2 and Fig. 7.3 for the two ranges of λ. The proposed Distributed Space-Time Block Codes (DSTBC) cooperative communication system shows steady performance improvement in the two λ ranges. The performance enhancement due to the addition of the distributed STBC encoding on the second decoding stage on the two figures is about 2.5 dB at 10^{-5} BER at the destination. It is also noted that the addition of the distributed STBC encoding to the cooperative system improves the slope of the BER curve. The total improvement of the joint distributed system over the non-cooperative communication can go up to 3 dB at the same bit-error-rate. Although the distributed STBC encoding is only applied during the second transmission phase only to the second part of the code (which is the vertical parity), the proposed configuration shows large improvement over the system without the addition of the STBC encoding stage and over the non-cooperative

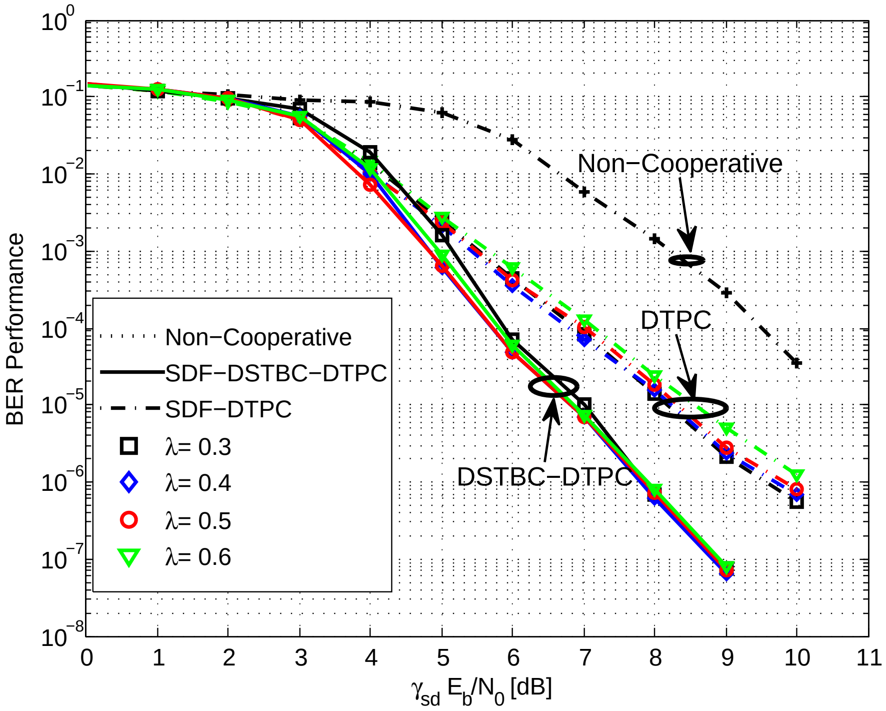


Figure 7.4. BER performance of the joint Distributed STBC DTPC encoding using soft decoded and forward SDF protocol for relay positions $3 \leq \lambda \leq 6$

reference systems.

The two figures in 7.4 and 7.5 shows the BER performance of the three systems: the non-cooperative system, the cooperative system employing SDF-DTPC only and the system employing the joint distributed STBC and SDF-DTPC encoding system. Here the proposed system also shows similar performance improvement compared to the other two systems for the second relaying protocol. The transmit diversity gain over the system with DTPC only is about 2.2 dB and this gain can reach 3.5 dB over the non-cooperative communication as can be seen at the 10^{-5} BER level.

To show the effect of the relay position on the performance of the proposed distributed coding system on a fast fading channel we have plotted the BER performance versus the relay position λ on Fig. 7.6 and Fig. 7.7 for the DF and SDF relaying protocols, respectively. Each curve in the two figures represents the BER performance

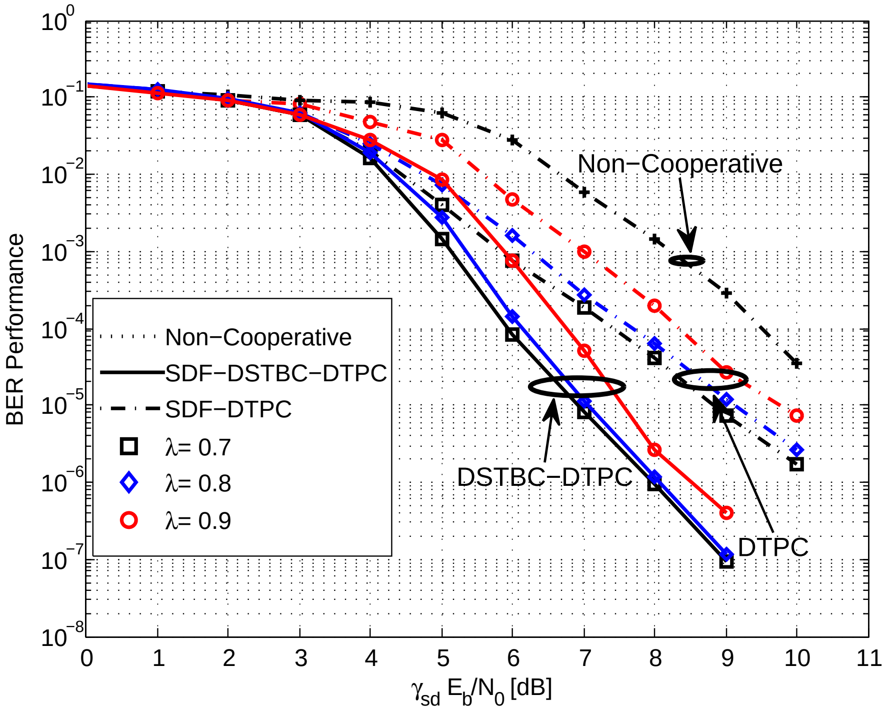


Figure 7.5. BER performance of the joint Distributed STBC DTPC encoding using hard decoded and forward SDF protocol for relay positions $7 \leq \lambda \leq 9$

for the distributed system when the SNR of the direct channel link is fixed and the position of the relay is changes between 0 and 1. It is noted from the two figures that as the SNR of the direct path increases the improvement of the joint distributed system increases over the distributed system with DTPC only. It is also noted due to the STBC diversity that the relay could also help in improving the performance even if the relay channel experience the same SNR as the direct path channel, as can be seen for the case when $\lambda = 1$. The improvement over the DTPC system increases when the SNR of the inter-user channel increases, this can be seen by comparing the two BER curves at a certain SNR when the λ increases from 0 to 1.

7.4 Conclusions

We have proposed in this chapter a new joint distributed system that joins the features of distributed TPC encoding system with the features of distributed STBC

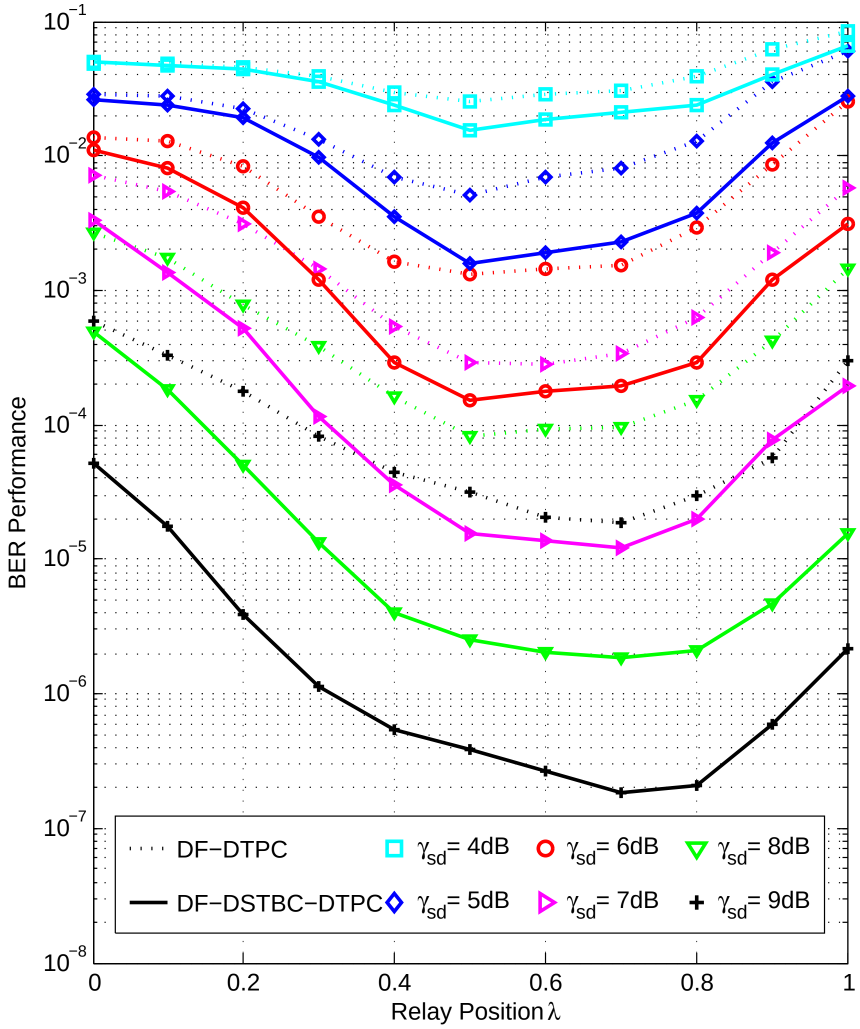


Figure 7.6. BER performance of the two distributed systems, the system with DTPC only and the system with joint DSTBC and DTPC when the relay is using DF protocol plotted against the position of the relay λ

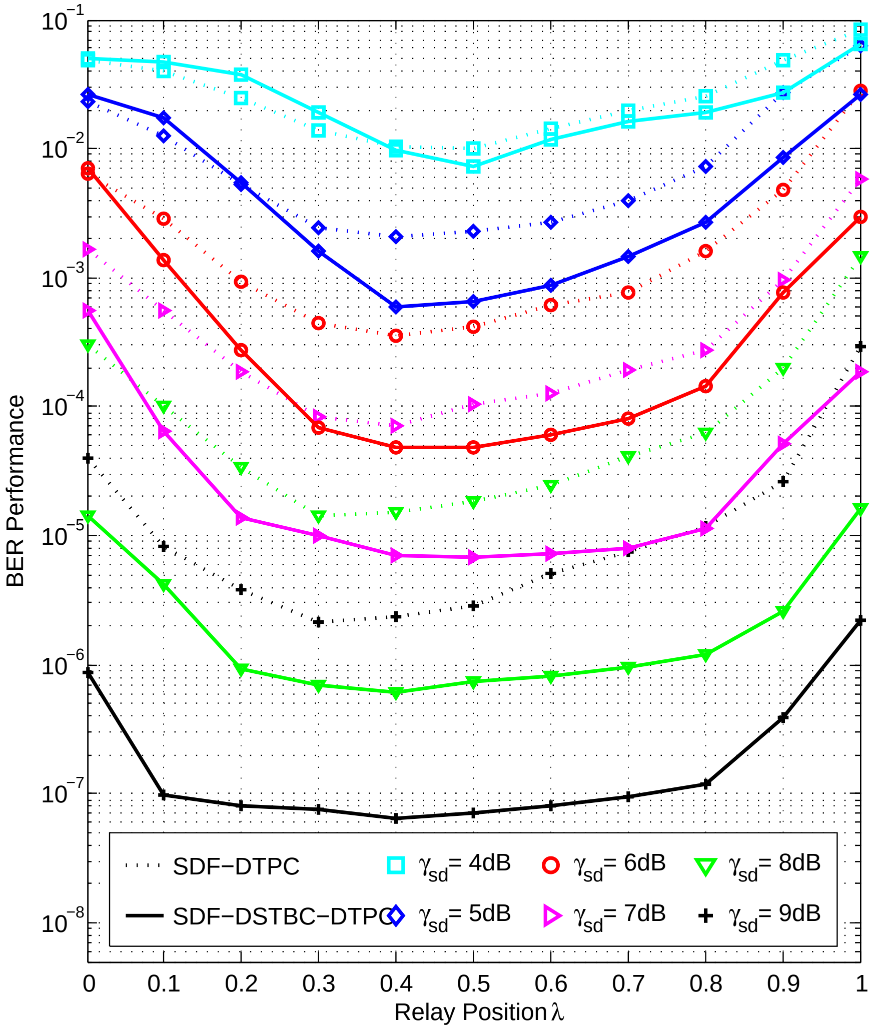


Figure 7.7. BER performance of the two distributed systems, the system with DTPC only and the system with joint DSTBC and DTPC when the relay is using SDF protocol plotted against the position of the relay λ

encoding system. We tested our proposed system in a fast fading environment on the three channels connecting the three nodes in the cooperative network. The simulation results for the two relaying protocols, the DF and SDF, show at least 2 dB improvement over the earlier proposed system based on distributed TPC only and an overall gain that can reach 3.5 dB over the non-cooperative encoded system without STBC. The design of the distributed system is very simple and does not require any additional radios or external antennas. The three systems used for comparison utilize the same bandwidth and consume the same amount of power.

Chapter 8

Concluding Remarks and Future Work

8.1 Conclusions

We presented on this dissertation a framework for distributed turbo block codes, for which we present simulation results under the assumption that the relay makes decoding errors and forward erroneous incremental parity to the destination. Under this framework, we propose solutions to enhance the BER performance under different channel conditions. We first proposed a method to generate multiple vertical parities for the turbo block codes using a cyclic interleaver. This cyclic interleavers is employed on multiple relays which forward the result parities to the destination. The destination performs a joint turbo decoding for all the received vertical parities. The results for single and multiple relays show that the BER performance can be improved by increasing the number of relays and enhancing the interuser channel, but it is limited by the SNR of the direct link channel.

We also proposed a soft information relaying technique in which the relay decodes the source's message and re-encodes it across columns using a novel soft block encoding technique to obtain soft parity bits with different reliabilities that can be used as soft incremental redundancy that is forwarded to the destination. The results presented for the SIR technique show that a cooperative coding system based on DTPC with soft incremental redundancy outperforms its two counterparts; the DTPC based on DF and the non-cooperative TPC. With the same spectral and power efficiency, the distributed TPC can give BER performance that is closer to the channel capacity. In the SDF cooperative coding technique, the relay can help in the distributed encoding process even when the inter-user channel having low SNR. With the SDF cooperative coding technique the relay can boost the performance for the two-ways communication system between any two nodes unlike the conventional DF.

To overcome the error propagation at the destination we proposed also a power allocation method and verified the effectiveness of this simple method by comparing it's results to system with optimized power at the source and the relay. The results obtained from applying the proposed power allocation method show large gain in BER performance and therefore showed effectiveness in allocating power between the transmitting nodes. For the same purpose, we proposed a power optimization algorithm for the distributed coded system that is based on a sliding ball principle. The main target for the power optimization is to minimize the final BER for the three transmission channels via optimal power allocation between the source and the relay. The goal is to optimally share overall total transmitting power from the source and the destination, so that the power is efficiently utilized to gain the maximum performance possible for the DTPC. The results obtained from the optimization algorithm emphasizes on the validity of the proposed power allocation criterion and show the maximum possible attainable performance from the DTPC cooperative system.

Finally, we proposed a joint distributed STBC-TPC system that aims to enhance the BER performance by transmitting the second part of the turbo product code over virtual transmit antenna using the source and the relay. We tested our proposed system in a fast fading environment on the three channels connecting the three nodes in the cooperative network. The simulation results for the two relaying protocols, the DF and SDF, show at least 2 dB improvement over the earlier proposed system based on distributed TPC only and an overall gain that can reach 3.5 dB over the non-cooperative encoded system without STBC. The design of the distributed system is very simple and does not require any additional radios or external antennas. The three systems used for comparison utilize the same bandwidth and consume the same amount of power.

8.2 Future Work

We plan to continue our research in the distributed coding systems to cover the research areas that has not been fully covered by this dissertation or any previous work. In the future work, we plan to:

1. Study the power allocation and optimization for the joint distributed STBC-TPC encoding.

2. Study the power allocation problem for channels with block fading.

3. Study the use of multiple vertical parities in non-cooperative scenarios to lower the code rate and increase the performance inspired by the belief propagation decoding method.

4. Analytically study the performance of the DF and SDF-DTPC systems by obtaining the Weight Enumerator Function (WEF) and then using the union bounds to obtain an approximate expression for the overall BER of the distributed coded system. We plan to use the result to design more efficient turbo decoding.

BIBLIOGRAPHY

Bibliography

[1] A. Sendonaris, E. Erkip, and B. Aazhang. User cooperation diversity. part i. system description. *IEEE Trans. Commun.*, 51(11):1927–1938, 2003.

[2] J.N. Laneman, D.N.C. Tse, and G.W. Wornell. Cooperative diversity in wireless networks: Efficient protocols and outage behavior. *IEEE Transactions on Information theory*, 50(12):3062–3080, 2004.

[3] T. Wang, A. Cano, G.B. Giannakis, and J.N. Laneman. High-performance cooperative demodulation with decode-and-forward relays. *IEEE Transactions on Communications*, 55(7):1427, 2007.

[4] Yonghui Li and B. Vucetic. On the performance of a simple adaptive relaying protocol for wireless relay networks. In *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pages 2400–2405, May 2008.

[5] F.A. Onat, A. Adinoyi, Y. Fan, H. Yanikomeroglu, and J.S. Thompson. Optimum threshold for SNR-based selective digital relaying schemes in cooperative wireless networks. In *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, 2007.

[6] HH Sneesens and L. Vandendorpe. Soft decode and forward improves cooperative communications. In *Computational Advances in Multi-Sensor Adaptive Processing, 2005 1st IEEE International Workshop on*, pages 157–160, 2005.

[7] Y. Li, B. Vucetic, T. F. Wong, and M. Dohler. Distributed Turbo Coding With Soft Information Relaying in Multihop Relay Networks. *IEEE J. Sel. Areas Commun.*, 24(11):2040–2050, 2006.

[8] S. YIU, R. SCHOBER, and L. LAMPE. Distributed space-time block coding. *IEEE transactions on communications*, 54(7):1195–1206, 2006.

[9] A. Chakrabarti, A. Baynast, A. Sabharwal, and Behnaam Aazhang. Low density parity check codes for the relay channel. *IEEE J. Sel. Areas Commun.*, 25(2):280–291, 2007.

[10] B. Zhao and M. C. Valenti. Distributed turbo coded diversity for relay channel. *Electronics Letters*, 39(10):786–787, 2003.

[11] M. Janani, A. Hedayat, T.E. Hunter, and A. Nosratinia. Coded cooperation in wireless communications: space-time transmission and iterative decoding. *IEEE Transactions on Signal Processing*, 52(2):362–371, 2004.

[12] X. Bao and J. Li. Adaptive network coded cooperation (ANCC) for wireless relay networks: matching code-on-graph with network-on-graph. *IEEE Transactions on Wireless Communications*, 7(2):574–583, 2008.

[13] Y. Li, B. Vucetic, and J. Yuan. Distributed turbo coding with hybrid relaying protocols. In *IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, 2008. PIMRC 2008*, pages 1–6, 2008.

[14] L. Fei, L. Qinghua, L. Tao, and Y. Guangxin. Impact of Relay Location According to SER for Amplify-and-Forward Cooperative Communications. In *2007 IEEE International Workshop on Anti-counterfeiting, Security, Identification*, pages 324–327, 2007.

[15] T. E. Hunter and A. Nosratinia. Coded cooperation under slow fading, fast fading, and power control. In *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 118–122 vol.1, 2002.

[16] Hong Wen, Liang Zhou, and Zhongpei Zhang. Cooperative Coding Using Parallel Concatenated LDPC Codes. *Information Theory Workshop, 2006 IEEE*, pages 395–398, Oct. 2006.

[17] Jeremie Ezri and Michael Gastpar. On the Performance of Independently Designed LDPC Codes for the Relay Channel. In *Proc. IEEE International Symposium on Information Theory*, pages 977–981, 2006.

[18] P. Razaghi and Wei Yu. Bilayer Low-Density Parity-Check Codes for Decode-and-Forward in Relay Channels. *IEEE Trans. Inf. Theory*, 53(10):3723–3739, 2007.

[19] M. C. Valenti and Bin Zhao. Distributed turbo codes: towards the capacity of the relay channel. In *Proc. VTC 2003-Fall Vehicular Technology Conference 2003 IEEE 58th*, volume 1, pages 322–326 Vol.1, 2003.

[20] R. Pyndiah, A. Glavieux, A. Picart, and S. Jacq. Near optimum decoding of product codes. In *Proc. IEEE Global Telecommunications Conference GLOBECOM '94. Communications: The Global Bridge*, pages 339–343 vol.1, 1994.

[21] K. Azarian, H. El Gamal, and P. Schniter. On the achievable diversity-multiplexing tradeoff in half-duplex cooperative channels. *Information Theory, IEEE Transactions on*, 51(12):4152–4172, Dec. 2005.

[22] L. Chu, J. Yuan, and Y. Li. Performance analysis and code design of distributed space-time trellis codes for a detection-and-forward system. In *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pages 1172–1176, 30 2007-Oct. 3 2007.

[23] J. Hwang, R.R. Consulta, and H. Yoon. 4G MOBILE NETWORKS–TECHNOLOGY BEYOND 2.5 G AND 3G. *PTC*, January 2007.

[24] G. Turin. On optimal diversity reception. *Information Theory, IRE Transactions on*, 7(3):154–166, 1961.

[25] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding and decoding: Turbo-codes. In *IEEE International Conference on Communications, 1993. ICC 93. Geneva. Technical Program, Conference Record*, volume 2, 1993.

[26] Shannon CE. and W. Weaver. A mathematical theory of communication. *Bell Syst. Tech. J*, 27:379–423, 1948.

[27] P. Elias. Error-free Coding. *Transactions of the IRE Professional Group on Information Theory*, 4(4):29–37, 1954.

[28] A. Glavieux. Channel Coding in Communication Networks. *London: Hermes Science Pub. Ltd*, 2007.

[29] E.R. Berlekamp. Algebraic Coding Theory. *New York*, 1968.

[30] J. Massey. Shift-Register Synthesis and BCH Decoding. *IEEE Transactions on Information Theory*, 15(1):122–127, 1969.

[31] S. Reddy. On decoding iterated codes. *IEEE Transactions on Information Theory*, 16(5):624–627, 1970.

[32] S. Reddy and J. Robinson. Random error and burst correction by iterated codes. *IEEE Transactions on Information Theory*, 18(1):182–185, 1972.

[33] D. Chase. Class of algorithms for decoding block codes with channel measurement information. *IEEE Trans. Inf. Theory*, 18(1):170–182, 1972.

[34] R. Pyndiah. Iterative Decoding of Product Codes: Block Turbo Codes. *Proc. Int. Symp. Turbo Codes and Related Topics*, pages 71–79.

[35] R. Pyndiah. Near-optimum decoding of product codes: block turbo codes. *Communications, IEEE Transactions on*, 46(8):1003–1010, Aug 1998.

[36] M.P.C. Fossorier and Shu Lin. Soft-input soft-output decoding of linear block codes based on ordered statistics. In *Global Telecommunications Conference, 1998. GLOBECOM 98. The Bridge to Global Integration. IEEE*, volume 5, pages 2828–2833 vol.5, 1998.

[37] P.A. Martin, D.P. Taylor, and M.P.C. Fossorier. Soft-input soft-output list-based decoding algorithm. *Communications, IEEE Transactions on*, 52(2):252–262, Feb. 2004.

[38] P.A. Martin, A. Valembois, M.P.C. Fossorier, and D.P. Taylor. On soft-input soft-output decoding using "box and match" techniques. *Communications, IEEE Transactions on*, 52(12):2033–2037, Dec. 2004.

[39] N. Le, A.R. Soleymani, and Y.R. Shayan. Distance-based-decoding of block turbo codes. *Communications Letters, IEEE*, 9(11):1006–1008, Nov. 2005.

[40] A. Sendonaris, E. Erkip, and B. Aazhang. Increasing uplink capacity via user cooperation diversity. In *Proc. IEEE International Symposium on Information Theory*, pages 156–, 1998.

[41] L. Hanzo, T.H. Liew, and B.L. Yeap. *Turbo coding, turbo equalisation and space-time coding for transmission over fading channels.* Wiley-IEEE Press, 2002.

[42] Shu Lin and Daniel J. Costello. *Error Control Coding, Second Edition.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2004.

[43] Chuxiang Li, Guosen Yue, M. A. Khojastepour, Xiaodong Wang, and M. Madihian. LDPC-coded cooperative relay systems: performance analysis and code design. *IEEE Trans. Commun.*, 56(3):485–496, 2008.

[44] Zhixin Liu, V. Stankovic, and Zixiang Xiong. Wyner-Ziv coding for the half-duplex relay channel. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, volume 5, pages v/1113–v/1116 Vol. 5, 2005.

[45] V. Stankovic, A. Host-Madsen, and Zixiang Xiong. Cooperative diversity for wireless ad hoc networks. *Signal Processing Magazine, IEEE*, 23(5):37–49, 2006.

[46] A. Chakrabarti, A. de Baynast, A. Sabharwal, and B. Aazhang. Codes for Half Duplex Relay Channels. In *Proc. International Zurich Seminar on Communications*, pages 158–161, 2006.

[47] J.G. Proakis. *Digital Communications.* McGraw-Hill, 1995.

[48] J.C. Interlando and Jr. Palazzo, R. Modified Berlekamp-Massey algorithm for decoding BCH codes defined over rings. pages 94–, Jun-1 Jul 1994.

[49] Ruiyuan Hu and Jing Ti. Practical Compress-Forward in User Cooperation: Wyner-Ziv Cooperation. pages 489–493, July 2006.

[50] Y. Hairej, A. Darmawan, and H. Morikawa. Cooperative Diversity using Soft Decision and Distributed Decoding. pages 1–5, July 2007.

[51] EA. Obiedat, G. Chen, and L. Cao. Distributed Turbo Product Codes with Multiple Vertical Parities. volume 1193, pages 405–412. AIP, 2009.

[52] J. Hagenauer, E. Offer, and L. Papke. Iterative decoding of binary block and convolutional codes. *Information Theory, IEEE Transactions on*, 42(2):429–445, Mar 1996.

[53] Yonghui Li. Distributed coding for cooperative wireless networks: an overview and recent advances - [topics in ad hoc and sensor networks]. *Communications Magazine, IEEE*, 47(8):71–77, August 2009.

[54] A. Sendonaris, E. Erkip, and B. Aazhang. User cooperation diversity. Part II. Implementation aspects and performance analysis. *IEEE Trans. Commun.*, 51(11):1939–1948, 2003.

[55] LH Brandenburg and AD Wyner. Capacity of the Gaussian channel with memory: The multivariate case. *Bell Syst. Tech. J*, 53(5):745–778, 1974.

[56] T.M. Cover and J.A. Thomas. *Elements of information theory.* Wiley, 2006.

[57] A. Host-Madsen and J. Zhang. Capacity bounds and power allocation for wireless relay channels. *Information Theory, IEEE Transactions on*, 51(6):2020–2040, June 2005.

[58] C.T.K. Ng and A.J. Goldsmith. Capacity and power allocation for transmitter and receiver cooperation in fading channels. In *Communications, 2006. ICC '06. IEEE International Conference on*, volume 8, pages 3741–3746, June 2006.

[59] Q. Lin, Z. Yong, S. Mei, and W. Danzhi. Cross-Layer Design for Relay Selection and Power Allocation Strategies in Cooperative Networks. In *Proceedings of the 2009 Seventh Annual Communication Networks and Services Research Conference-Volume 00*, pages 232–237. IEEE Computer Society Washington, DC, USA, 2009.

[60] G. Xiang and H.K. Park. Relay selection and power allocation scheme for cooperative transmission. In *Advanced Technologies for Communications, 2008. ATC 2008. International Conference on*, pages 158–161, 2008.

[61] B. Wang, Z. Han, and K.J.R. Liu. Distributed relay selection and power control for multiuser cooperative communication networks using stackelberg game. *IEEE Transactions on Mobile Computing*, 8(7):975–990, 2009.

[62] D.R. Brown III. Resource allocation for cooperative transmission in wireless networks with orthogonal users. In *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, volume 2, pages 1473–1477, 2004.

[63] Z. Qi, Z. Jingmei, S. Chunju, W. Ying, Z. Ping, and H. Rong. Power allocation for regenerative relay channel with Rayleigh fading. In *Proc. IEEE VTC*, pages 1167–1171, 2004.

[64] SD Gupta and D. Reynolds. Position Dependant Power Allocation Strategies in Cooperative Relay Networks. In *Military Communications Conference, 2006. MILCOM 2006*, pages 1–7, 2006.

[65] E. Obiedat, W. Xiang, J. Leis, and Cao L. Soft Incremental Redundancy for Distributed Turbo Product Codes. In *7th Annual IEEE Consumer Communications and Networking Conference*. IEEE, 2010.

[66] C. Pirak, ZJ Wang, and KJR Liu. An Adaptive Protocol for Cooperative Communications Achieving Asymptotic Minimum Symbol-Error-Rate. In *2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings*, volume 4, 2006.

[67] J. Adeane, M.R.D. Rodrigues, and I.J. Wassell. Centralised and distributed power allocation algorithms in cooperative networks. In *Proc. IEEE 6th Workshop on Signal Processing Advances in Wireless Communications*, pages 333–337, 2005.

[68] MO Hasna and M.S. Alouini. Optimal power allocation for relayed transmissions over Rayleigh-fading channels. *IEEE Transactions on Wireless Communications*, 3(6):1999–2004, 2004.

[69] I. Maric and R. Yates. Forwarding strategies for Gaussian parallel-relay networks. In *IEEE INTERNATIONAL SYMPOSIUM ON INFORMATION THEORY*, pages 269–269. Citeseer, 2004.

[70] N. Ahmed, M.A. Khojastepour, and B. Aazhang. Outage minimization and optimal power control for the fading relay channel. In *IEEE Information Theory Workshop*, pages 458–462, 2004.

[71] E.G. Larsson and Y. Cao. Collaborative transmit diversity with adaptive radio resource and power allocation. *IEEE Communications Letters*, 9(6):511–513, 2005.

[72] E.A. Obiedat and L. Cao. Soft Information Relaying for Distributed Turbo Product Codes (SIR-DTPC). *IEEE Signal Processing Letters*, 17:363–366, 2010.

[73] AR Calderbank. The art of signaling: Fifty years of coding theory. *IEEE Transactions on information theory*, 44(6):2561–2595, 1998.

[74] S. Verdu. Wireless bandwidth in the making. *IEEE Communications Magazine*, 38(7):53–58, 2000.

[75] AF Naguib, N. Seshadri, and AR Calderbank. Increasing data rate over wireless channels. *IEEE Signal Processing Magazine*, 17(3):76–92, 2000.

[76] V. Tarokh, N. Seshadri, and AR Calderbank. Space-time codes for high data rate wireless communication: Performance criterion and code construction. *IEEE transactions on information theory*, 44(2):744–765, 1998.

[77] S.M. Alamouti. A Simple Transmit Diversity Technique for Wireless Communications. *IEEE Journal on Selected Areas in Communications*, 16(8), 1998.

[78] J.N. Laneman and G.W. Wornell. Distributed space-time-coded protocols for exploiting cooperative diversity in wireless networks. *IEEE Transactions on Information theory*, 49(10):2415–2425, 2003.

[79] Y. Jing and B. Hassibi. Distributed space-time coding in wireless relay networks. *IEEE Transactions on Wireless Communications*, 5(12):3524–3536, 2006.

VITA

Esam Ali Obiedat, born in Irbid Jordan in 1980. He finished his B.Sc. and M.Sc. from Jordan University of Science and Technology, in 2003 and 2005, respectively. Esam joined Ole Miss PhD program in Fall 2006 and defended his dissertation in Spring 2010 under the supervision of Dr. Lei Cao. During his study in the University of Mississippi he was awarded the graduate school fellowship. Before starting in the PhD program he was working as Spectrum Planning Engineer in the Telecommunication Regulatory Commission of Jordan. Esam published more than 13 journal and conference papers in the area of coding theory and cooperative communication systems. Some of his main research interest includes cooperative communication, OFDMA, space time coding, turbo coding techniques.