A CLASS OF IMPLICIT TRANSMISSION TECHNIQUES FOR THROUGHPUT ENHANCEMENT

by

Elahe Rezaei

APPROVED BY SUPERVISORY COMMITTEE:

____
Dr. John. P. Fonseka, Chair

____
Dr. Naofal Al-Dhahir

____
Dr. Hlaing Minn

____
Dr. M. Saquib

to my parents,

with whom life begins and

love never ends

....

A CLASS OF IMPLICIT TRANSMISSION TECHNIQUES

FOR THROUGHPUT ENHANCEMENT

by

ELAHE REZAEI, BS, MS

DISSERTATION

Presented to the Faculty of

The University of Texas at Dallas

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY IN

ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT DALLAS

December 2018

ACKNOWLEDGMENTS

I would like to express my deepest appreciation and gratitude to my PhD advisor, Professor John P. Fonseka, for his priceless advice and encouragement. My PhD has been an astonishing experience and I thank John wholeheartedly, not only for his tremendous academic support, but also for giving me so many various opportunities. I greatly appreciate his consistent help, patience and guidance to grow as a researcher and teacher simultaneously. I would also like to thank my committee members, Professor Naofal Al-Dhahir, Professor Hlaing Minn, and Professor Mohammad Saquib for their valuable comments and suggestions to improve my work. Also, I thank greatly my colleagues and friends, Yu Bo and Merve Apalak, with whom I had a great time working together.

Finally, but by no means least, special thanks to my mum and dad for almost unbelievable support. I cannot find any words to express how grateful I am to you for all of the sacrifices you've made for me.

To my sisters and my friends, thank you for listening, offering me suggestions, and supporting me through this journey.

I dedicate this accomplishment to them with my deepest gratitude and love.

August 2018

A CLASS OF IMPLICIT TRANSMISSION TECHNIQUES

FOR THROUGHPUT ENHANCEMENT

Elahe Rezaei, PhD
The University of Texas at Dallas, 2018

Supervising Professor: Dr. John. P. Fonseka, Chair

Throughput enhancing techniques are very valuable to keep up with the fast increasing data rates in communication systems. Implicit transmission is particularly attractive as it can transmit information without physically transmitting them over a channel. In this study, two separate throughput enhancing techniques using implicit transmission are investigated.

First a multi-constellation signaling (MCS) technique that selects one out of $N(>1)$ constellations based on a set of implicit bits during every interval is introduced. The overall constellation used by a MCS scheme is a NM-ary constellation formed by replacing every point of a M-ary constellation by a cluster with N constellation points. Further, the size of clusters is reduced by employing multi-dimensional mapping. It is demonstrated that a properly designed MCS scheme can double and triple the throughput and also perform better than a scheme that employs a single constellation, and MCS schemes can perform better than turbo coded signals in the long term evolution (LTE).

In contrast to MCS schemes, throughput enhancing concatenated codes (TECCs) schemes transmit bits implicitly without expanding the overall constellation. In a TECC, the coded sequence of a code C transmitted over a channel is altered according to the coded bits of a second coded sequence of an implicit code C'. In this study, TECCs select one bit in every segment of n coded bits of C based on $n_s \leq log_2 n$ coded bits of C', and flip that selected coded

bit of C before transmission. It is shown that using iterative decoding between codes C and C', the receiver can decode the coded bits of both the explicit code C and the implicit code C' jointly. TECCs that can increase the throughput of C by 25% to 37.5% without sacrificing the performance of C are reported, however, at the expense of increased complexity.

TECCs are extended to form a new class of TECC-2 schemes to include a second uncoded implicit stream. It is shown here that TECC-2 schemes can significantly increase the throughput enhancing capability of TECCs at high signal to noise ratio (SNR) with only a modest increase in complexity. The tradeoff between the low SNR performance and throughput expansion is also discussed.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Wireless and Internet Protocol (IP) traffic is expanding rapidly as the smart phones and Internet-oriented devices become detachable part of human lives. For example, it is estimated that the IP traffic would increase threefold by 2022 (Cisco, 2017b). Also the number of smart devices is estimated to be 1.5 mobile devices per capita by 2021 (Cisco, 2017a). In order to address the expanding demand, the wireless communication has evolved into the current fourth-generation Long-Term Evolution (4G LTE) system. Similarly, on the optical backbone transmission, the optical transport network (OTN) has evolved starting from 10 Gbit/s transmission to 100 Gbit/s (Bartelt et al., 2017). The transmission technologies further require significant changes to meet the demands in near future. With that in mind, the wireless technology is currently moving into fifth-generation (5G) systems while the OTN is targeting 1 Tbit/s systems in the near future (Agiwal et al., 2016).

The digital transmission faces limitation in power, frequency spectrum and complexity. To deal with these limitations, the digital transmission technology has been focusing on techniques to efficiently use the available spectrum and to employ various multiplexing techniques. These multiplexing techniques include techniques such as time-division multiple access (TDMA), frequency-division multiple access (FDMA), orthogonal frequency-division multiple access (OFDMA) and code-division multiple access (CDMA) (Feng et al., 2013). The currently employed OFDM, multiple-input and multiple-output (MIMO) and cognitive radio (CR) have evolved from efficient use of spectrum and multiplexing (Tragos et al., 2013). In addition, advances in digital technology and signal processing have helped to better design of high speed efficient transmitters and receivers.

This dissertation is focused on throughput enhancing methods that can be used to increase the data rate of a communication link. The class of throughput enhancing techniques introduced in this dissertation is derived primarily using the implicit transmission of information which was first introduced in spatial modulation (SM) (Mesleh et al., 2008).

1.1 Literature Review

1.1.1 Spatial Modulation

Traditionally, in digital transmission, the throughput can be increased by employing higher order modulation (Proakis and Salehi, 2008). For example, 64-QAM can increase the throughput by a factor 3 compared with QPSK transmission without expanding bandwidth (Meyr et al., 1998). Lately, multiple antennas at the transmitter and receiver have been emerged as a method to increase the throughput (Mietzner et al., 2009). For example, MIMO schemes that employ n_t transmitting antennas can increase throughput by factor n_t by employing at least n_t receiving antennas. MIMO technology has been used in V-BLAST and D-BLAST MIMOs for the same purpose of increasing throughput (Choi and Murch, 2004). Multiple antennas have also been used in spatial modulation (SM), MIMO-OFDM and generalizes spatial modulation (GSM) (Di Renzo et al., 2011).

Compare with MIMO, SM transmits only from a single antenna which is selected based on up to $log_2 n_t$ additional bits. These additional bits are transferred to the destination implicitly along with the actual bits transmitted explicitly over the channel by the selected antenna. SM was initially introduced as Space Shift Keying (SSK) which transmits only implicit bits without any explicit bits, by simply transmitting an RF tone from the selected antenna (Jeganathan et al., 2009). The principles of MIMO and SM have been combined in the literature to introduce GSM technique, which is also referred to as GSM-MIMO and SM-MIMO. In GSM, instead of selecting one antenna during each interval as in SM, a subset of $n_a (\leq n_t)$ antennas is selected based on implicit bits and transmitting additional information from all selected n_a antennas. As a result, compared with SM, GSM can increase both the number of implicit bits and explicitly transmitted bits in any interval thereby further increasing the throughput. However compared with SM, both GSM and MIMO schemes have higher complexity due to the simultaneous transmission from multiple antennas (Di Renzo et al., 2011).

1.1.2 Concatenated Codes

Concatenated codes can achieve very high coding gains and they are used in many applications such as convolutional turbo codes (TCs) and turbo product codes (TPCs) in mobile wireless communications and deep space applications (Pyndiah, 1998). In Aitsab and Pyndiah (1997), the authors have suggested a concatenated coding structure that can significantly improve performance over individual codes at reasonable increase in complexity. It consists of the cascade of an inner code and an outer code, where the inner code is usually a short convolutional code and the outer code is a high rate code such as a Reed Solomon code. In Benedetto et al. (1997), the concatenated codes have been decoded using soft iterative decoding by exchanging reliability information between the inner and outer codes by following belief propagation approach (McEliece et al., 1998). Concatenated codes were later developed serial concatenated codes (SCCs) and parallel concatenated codes (PCCs) by using an additional interleaver and making the block length of the concatenation large (Benedetto et al., 1998). Turbo codes, which are commonly used in applications such as in the 4G LTE, are PCCCs. It has been shown that in both SCCs and PCCs, the performance can be improved by increasing the interleaver size. As a result, these codes provide very high coding gains and achieve performance close to the Shannon limit (Berrou et al., 1993).

SCCs and PCCs can be decoded using soft iterative decoding. In each iteration, individual component codes are soft decoded and soft information is exchanged through the interleaver to the other component code. Convolutional component codes can be decoded either by using Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm or using soft output Viterbi algorithm (SOVA) (Lin and Costello, 2004). Similarly block component codes can be soft decoded using optimal decoding for short codes or by using sub-optimal simplified approaches presented in Pyndiah (1998) for long codes.

1.1.3 Multilevel Codes

A Multilevel code (MLC) consists of different streams for transmission by employing a code on each level, specifically MLC with L levels can employ to 2^L array constellation by taking one coded bit from each level to form a symbol on the constellation. MLCs can employ any type of high-order constellation such as ASK, PSK and QAM. MLCs were first introduced by Imai and Hirakawa (1977) according to balanced distance rule to target the same minimum Euclidian distance for all levels. Thereby achieving similar performance at all levels of MLCs. Later MLCs have been designed according to the capacity rule and coding exponent rule (Wachsmann et al., 1999), (Ungerboeck, 1982). In Huber and Wachsmann (1994) the MLCs have been analyzed from Information Theory point of view. Huber and Wachsmann (1994) and Kofman et al. (1994) independently proved that the capacity of the modulation scheme can be achieved by multilevel codes together with multistage decoding if and only if the individual rate of each level is appropriately selected. MLCs can be sub-optimally decoded using multistage-decoding (MSD) where each component code is decoded separately starting from the lowest level code. In MSD, during decoding of a component code at every level, all prior decisions are assumed to be correct (Calderbank, 1989).

1.2 Technical Description

In this section, we provide the technical details of topics relevant to the dissertation.

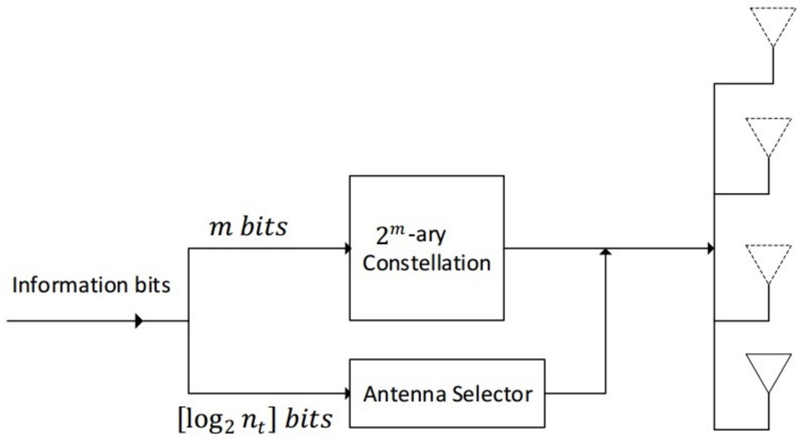

Figure 1.1: The model of transmitter for spatial modulation system.

1.2.1 Spatial Modulation

Let us consider a SM scheme that employs n_t transmitting antennas and 2^m-ary signal constellation. In SM, one transmitting antenna is selected for transmission based on $\lfloor \log_2 n_t \rfloor$ bits, which are referred here as implicit bits, where $\lfloor\rfloor$ denotes the standard floor function. During every interval only the selected antenna transmits while all remaining antennas are kept silent. As a result SM does not experience any interference from other channels as in MIMO. In addition, m bits, which are referred to as explicit bits here, are transmitted by the selection of a specific constellation point. As a result, SM scheme transmits a total of $\lfloor \log_2 n_t \rfloor + m$ bits during every interval. Since the fading channel characteristics are different for different antennas, the SM receiver is capable of identifying the specific antenna, the signal was transmitted from along with the transmitted signal constellation point. In order to extract the information of the transmitting antenna, the receiver however requires the channel information of all channels corresponding to all transmitting antennas during every interval. SM signals can be decoded by either jointly decoding both implicit and explicit bits, or by first detecting implicit bits by correctly identifying the transmitting antenna and then detecting the explicit bits by decoding the transmitted signal (Mesleh et al., 2006), (Younis et al., 2010). The joint decoding performs better however at the expense of complexity.

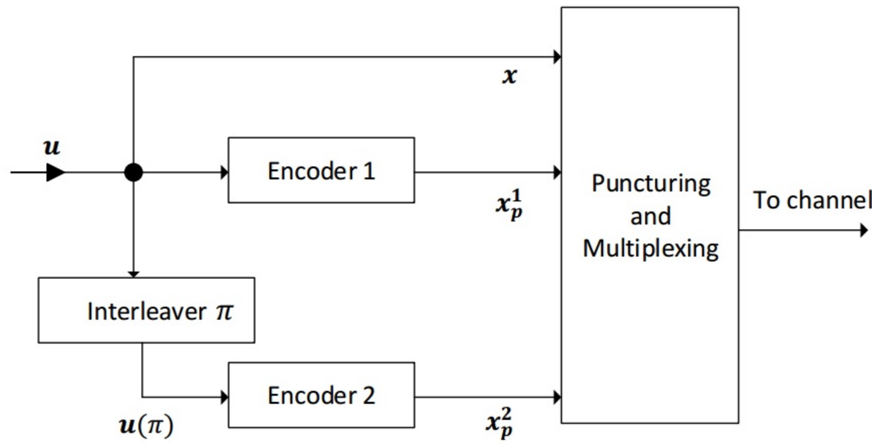

Figure 1.2: Turbo encoder structure.

1.2.2 Turbo Codes

Following the Shannon's channel coding theorem, block codes and convolutional codes can only reach the Shannon limit when their block length becomes very large (Shannon and Weaver, 1998). A concatenated code, such as a Turbo code, constructed with short component codes along with an interleaver can effectively be very long code. Therefore such concatenated codes when properly designed can approach the Shannon limit.

For illustration, Fig. 1.2 shows a rate 1/3 turbo code constructed with two rate 1/2 recursive systematic convolutional (RSC) codes and an interleaver. The first component code generates the first parity bit based on the transmitted information, while the second component code generates the second parity bit according to an interleaved version of the transmitted information. The overall turbo coded sequence is obtained by multiplexing the information sequence, and the two parity sequences. As a result, for every single input bit, the turbo code outputs three bits thereby making it a rate 1/3 code. To have any other desired rates, puncturing can be used to adjust the rate of turbo code.

The interleaver design has a significant effect on code performance, as it provides the interleaver gain. It is known that the interleaver gain increases with the frame size. Turbo codes are usually designed to achieve higher interleaver gain rather than trying to increase the minimum Hamming distance (MHD) of concatenation. However a low weight coded sequences of a turbo code determine the error floor of a turbo code (Sadjadpour et al., 2001).

Turbo codes can be decoded using iterative soft-input soft-output (SISO) decoding. The demodulation begins by extracting the log likelihood ratio (LLR) values of each bit from the received signal. The LLR is a measure of the probability that the transmitted bit u_l is +1 or -1, given the received signal r and is defined as:

$$L(u_l) \triangleq log\left[\frac{P(u_l = +1|r)}{P(u_l = -1|r)}\right] \tag{1.1}$$

The demodulator output which contains the LLR values of the sequence and the first and second parity bits are fed to the SISO components as illustrated in Fig. 1.3. SISO module gets two input probability distributions $P(u, I)$ and $P(c, I)$ and generates two output probability distributions $P(u, o)$ and $P(c, o)$ based on the code constraints (Benedetto et al., 1997). In an iterative SISO decoding, each component code is soft decoded and the extrinsic information which obtained from SISO is sent into the decoder of the other code as apriori information. Convolutional component codes are usually soft decoded using BCJR or SOVA decoding. During any soft decoding of a component code, the extrinsic information $L_e(u_l)$ is calculated by subtracting the apriori information of u (the input of SISO) $L(u_l)$ and information of r from channel from the aposteriori log likelihood ratio $L(u_l|r)$ (the output of SISO). $L_e(u_l)$ is called as extrinsic information as it is generated from only codeword parity bits. This apriori information is exploited in calculation of LLRs as a second input of SISO module for the other component code. For soft decoding of each component convolutional code, maximum a posteriori probability (MAP) based decoder such as BCJR algorithm or

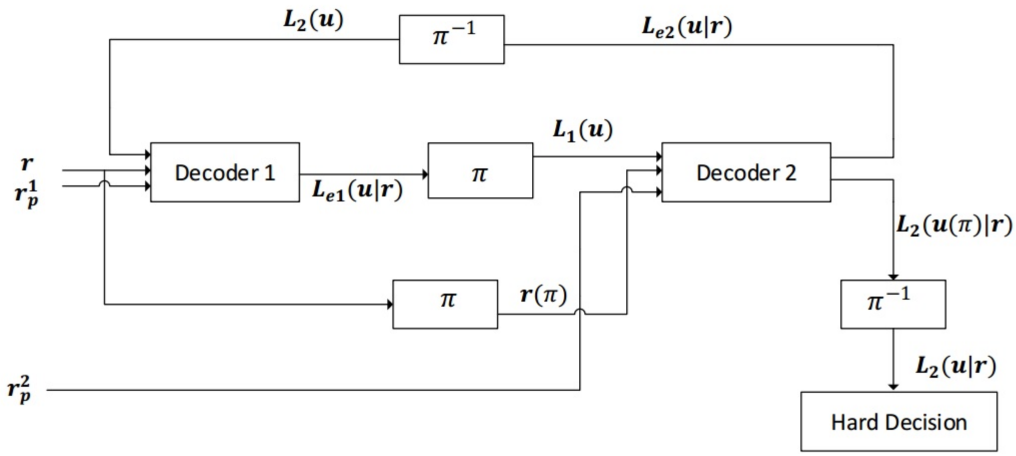

Figure 1.3: Turbo decoder structure.

maximum likelihood (ML) based decoder such as SOVA can be employed. BCJR tries to find the most likely symbol received, while SOVA looks for the most likely sequence. BCJR decoding is known to perform better than SOVA. BCJR decoding can employ either MAP decoding, log-MAP decoding or max-log-MAP decoding. The log-MAP implementation of BCJR avoids numerical issues related to MAP-based BCJR while max-log-MAP based BCJR is a simplified version of log-MAP BCJR with slight sacrifice in performance (Perişoară and Stoian, 2008). The difference between log-MAP and max-log-MAP decoding can be described using the following equation:

$$ln(e^x + e^y) = \begin{cases} max(x,y) + ln(1 + e^{-|x-y|}) & \textbf{log-MAP} \\ max(x,y) & \textbf{max-log-MAP} \end{cases} \tag{1.2}$$

where e^x and $ln(x)$ denote exponential and natural logarithm function of x, respectively.

1.2.3 Interleaving and Constellation Mapping

Interleaver is a key component of concatenated codes which is responsible in scrambling the inputs in a particular fashion. For example, TPCs usually use a row/column interleaver which guarantees the maximum achievable MHD. By reviewing the literature, we can find various interleaver design techniques for concatenated codes. The literature is mostly focused on design of interleaver for PCCs and SCCs with inner recursive codes. This is because the interleaver gain in PCCs and SSCs is the most important performance gain and increases with increasing frame size (Hu et al., 2017). Random interleavers such as uniform interleavers are simple to implement however they provide good gain (lower the error coefficients) only for large frame size.

Deterministic interleavers have been introduced for shorter frame size such as quadratic which can provide similar performance as random interleaver at higher frame sizes (Sun and Takeshita, 2005). It is also known that deterministic interleavers have better error floor variations (which is determined by the low-weight coded sequences). There are many different classes of deterministic interleavers based on permutation polynomials (Sadjadpour et al., 2001). A well-known deterministic interleaver is Quadratic permutation polynomial (QPP) which is currently employed in LTE standard of turbo codes (Nimbalker et al., 2008). In addition to designing interleavers based on MHD and interleaver gain, interleavers can also be designed based on iterative decoding suitability (IDS) (Sadjadpour et al., 2001). IDS measures effectiveness of iterative decoding based on the correlation between the extrinsic information obtained from decoder and the input information. The lower IDS values provide lower frame error rates. Another class of interleavers design is called constrained interleaving which try to jointly optimize the minimum distance and interleaver gain. Constrained interleavers can achieve a high minimum distance while at the same time providing an interleaver gain similar to those of uniform interleavers. As a result constrained interleavers can perform better than traditional row-column interleaving and random interleaving.

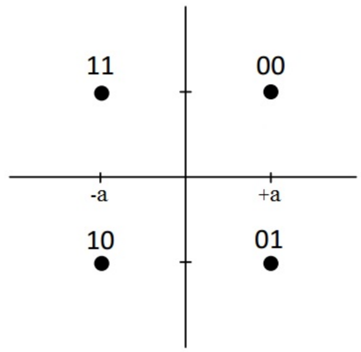

Figure 1.4: Signal constellations of 4-PSK with RGC mapping.

Interleavers also used to map coded bits onto a signal constellation. In order to achieve good performance, interleaver design should be done along with the mapping policy used for constellation. Different forms of joint interleaver design and mapping have been considered in the literature for bit interleaved coded modulation (BICM) (Chindapol and Ritcey, 2001). CICM has been introduced to systematically construct the interleaver and select the constellation mapping for a given signal constellation and frame size (Hu et al., 2017). CICM employs a reverse Gray coding (RGC) technique. RGC mapping policy ensures that as many one bit differences as possible on the constellation achieve the highest possible squared Euclidean distance (SED). Further, RGC gradually decreases the SED between the constellation points as the number of bit differences increases. For illustration, Fig. 1.4 shows a RGC mapping policy for 4-ary PSK that maintains a SED of $8a^2$ for all one bit differences in the most significant bit (MSB) position. However, in that constellation, the SED for all single bit differences that occur in the LSB is still $4a^2$. It has been shown in Hu et al. (2017) that a CICM interleaver can be systematically designed with RGC mapping to achieve the highest possible minimum squared Euclidean distance (MSED) and minimum symbol Hamming distance (MSHD).

1.2.4 Information Theory Principles

In this section, we present the principles used in Information Theory for the calculation of channel capacity and determining the achievable rate regions. These principles are later used with some of the techniques presented in this dissertation.

It is known that the channel capacity C of a communication system consisting of an input alphabet X and output alphabet Y and a probability transition matrix $p(y|x)$ is given by Gallager (1968)

$$C = max_{p(x)} I(X;Y) \tag{1.3}$$

where $p(x)$ denotes the probability distribution of x and $I(X;Y)$ is the mutual information of X and Y.

Extending (1.3) to a communication system with n inputs $X_1, X_2, ..., X_n$, and one output Y, the corresponding capacity can be written as :

$$C = max_{p(x_1,x_2,...,x_n)} I(X_1, X_2, ..., X_n; Y) \tag{1.4}$$

Following the chain rule of mutual information, the mutual information in (1.4) can be written as:

$$C = max_{p(x_1,x_2,...,x_n)} I(X_1, X_2, ..., X_n; Y) = max_{p(x_1,x_2,...,x_n)} \sum_{k=1}^{n} I(X_k; Y|X^{k-1}) \tag{1.5}$$

where X^k denotes $(X_1, X_2, ..., X_k)$.

According to Shannon's channel coding theorem, $R < I(X,Y)$ for rate R of a single input source in order to lower the error rate below any desired value (Shannon and Weaver, 1998). Extending the above statement to n inputs, the achievable rate regions have to satisfy the following conditions:

$$\sum_{i \in s} R_i \leq I(X^s; Y | X^{s^c}) \quad \forall s \subset \{1, 2, ..., n\} \tag{1.6}$$

where the complement of set s is denoted by s^c.

It is also know that the capacity of Gaussian channel (Sason, 2004) with power constraint P and noise variance σ^2, derived from (1.3), is given by

$$C = max_{p(x)} I(X; Y) = \frac{1}{2} log(1 + \frac{P}{\sigma^2}) \quad where \quad E[X^2] \leq P \tag{1.7}$$

where the maximum is attained when X has a normal distribution with variance P.

Similarly, the capacity C of the Gaussian channel with a uniform distribution on the input X with alphabet size K can be found by 1.3 as

$$C = log(K) - 1/K \sum_{x_k} \int_{\ell} log[\sum_{x'_k} - \frac{(\ell - x'_k + x_k)^2 - \ell^2}{2\sigma^2}] d\ell \tag{1.8}$$

1.3 Motivation and Contribution

The communication technology is forced to search for novel techniques to keep up with the rapidly increasing consumer demand. For example, according to Cisco (2017b), it is predicted that global mobile data traffic would sevenfold between 2016 and 2021. Internet service provider companies pursue solutions to increase network capacity and drastically reduce transmission cost over the network. Traditional methods such as allocating more spectrum or use of small cells while quite efficient, require new infrastructure and incur capital cost. On the other hand, by comparing the processing power of electronic devices from 1956 to 2015, we observe 1 trillion-fold increase in performance over the past six decades (Expert, 2017). This extensive growth in computation power could be exploited in advanced digital communication techniques which improve throughput without expansion of spectrum or transmitted power.

In this dissertation, we present a novel class of transmission techniques to increase the throughput on any digital communication systems. These techniques are flexible and can be applied to any existing schemes however at the expense of some complexity. In this study, first, we present a multi-constellation signal scheme (MCS) (Fonseka and Rezaei, 2017) that transmits higher order constellations in a much more efficient manner. Specifically, MCS employs higher order mapping along with clustered signal constellation thereby significantly improving performance over traditional higher order signal. For example, 16-ary and 64-ary MCS constellations that have average symbol energy close to that of QPSK signals and perform significantly better than the traditional 16QAM and 64QAM constellations respectively have been reported in Fonseka and Rezaei (2017). We also present a novel throughput enhancing concatenated coding (TECC) technique that can transmit a second coded data stream implicitly when transmitting a first coded data stream explicitly. As MCS, TECC can be employed in any digital transmission scheme. TECC schemes that can increase the throughput by 25 to 37% are reported without sacrificing performance in turbo codes employed in LTE.

CHAPTER 2

IMPLICIT TRANSMISSION OF CODED INFORMATION USING MULTI-CONSTELLATION SIGNALING (MCS) [1]

2.1 Introduction

In this chapter, a multi-constellation signaling (MCS) technique is proposed to transmit a portion of a coded information sequence implicitly along with the remaining portion of that sequence transmitted explicitly over the channel thereby increasing the throughput. As stated before, spatial modulation (SM) is a technique that transmits information implicitly from the selection of the transmitted antenna among several available antennas (Mesleh et al., 2006). However, SM requires multiple antennas at the transmitter to offer the system the freedom to choose one transmitter each interval based on the implicitly transmitted bits. Specifically, a SM scheme with n_t transmitting antennas implicitly transmits $m_{Im} = \lfloor \log_2 n_t \rfloor$, number of bits per interval from the selection of the antenna along with $m_{Ex} = log_2 M$ number of explicitly transmitted bits using a M-ary signal constellation (Mesleh et al., 2006), where $\lfloor . \rfloor$ denotes the standard floor function. As a result, a SM scheme transmits $(m_{Im} + m_{Ex})$ number of bits in total by only transmitting m_{Ex} number of bits physically over the channel (Mesleh et al., 2006). In contrast to SM, MCS technique presented here employs only a single transmitting antenna. However, similar to SM, it offers the transmitter the freedom to select a constellation among a bank of constellations at the transmitter based on a set of implicitly transmitted bits. The selection of the constellation in MCS is similar to the selection of an antenna in SM.

A. General Description: Consider a MCS scheme that employs a set of constellations, $S = \{C_1, C_2, ..., C_N\}$, and selects one constellation from S during every interval based on a set of implicitly transmitted bits. In this study, the constellations $C_2, C_3, ..., C_N$ are considered to be either scaled or rotated versions of a standard M-ary constellation C_1. Specifically, in this study, any C_t , $t = 2, 3, ..., N$, is considered as either $C_1 * r_t$ or $C_1 * e^{j\theta_t}$, where r_t and θ_t are a set of scaling factors and phase shifts respectively, $t = 2, 3, ..., N$. As a result, all constellations $C_1, C_2, ..., C_N$ considered here are M-ary constellations capable of transmitting $\log_2 M$ bits per interval explicitly. Let us consider the transmission of such a MCS scheme over a block of L intervals. Since each interval has N options to select a constellation, the entire block of L intervals has N^L possible permutations of the constellations. All permutations discussed here are permutations with replacement. Hence, the specific permutation of the constellations employed for the block of L intervals, which defines a particular constellation for each interval within the block, can be selected based on $N_{PS} = \lfloor \log_2(N^L) \rfloor$ number of permutation selection (PS) bits. As a result, the above described MCS scheme is capable of transmitting $m_{Im} = N_{PS}/L$ number of bits implicitly in addition to $m_{Ex} = \log_2 M$ number of bits transmitted explicitly during every interval. Hence, the above MCS scheme increases the throughput by a throughput expansion factor $\eta = (m_{Im} + m_{Ex})/m_{Ex}$ compared with a standard scheme that employs a single M-ary constellation. Note that m_{Im} is independent of L when N is an integer power of 2 and otherwise it depends on L. Throughout the paper, "transmission rate" is used for the actual symbol rate of explicitly transmitted bits, while "data transfer rate" is used for the actual data transfer rate which includes both implicit and explicit bits.

Fig. 2.1 depicts the structure of a MCS transmitter considered in this study. A message sequence $\boldsymbol{u}$ is encoded by a (n, k) outer block code (OBC) with minimum Hamming distance (MHD) d_m to form a coded bit sequence $\boldsymbol{v}$. The sequence $\boldsymbol{v}$ is divided by a sequence divider into two sequences, an explicit sequence, $\boldsymbol{v_{Ex}}$, and an implicit sequence, $\boldsymbol{v_{Im}}$. The MCS transmitter then transmits only the explicit sequence $\boldsymbol{v_{Ex}}$ using a constellation during each interval which is selected based on a group of implicit bits on $\boldsymbol{v_{Im}}$. A MCS scheme can

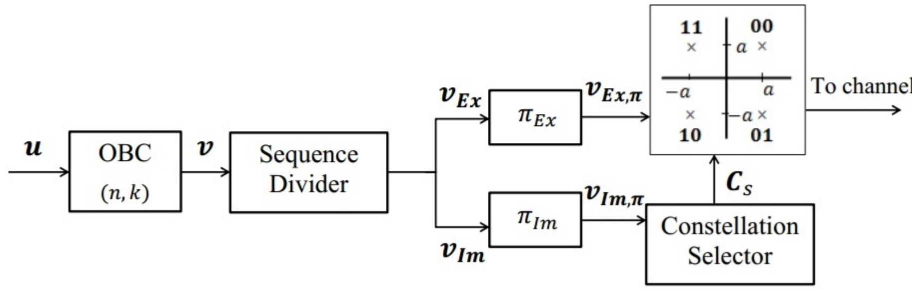

Figure 2.1: Generation of MCS signals during block of L intervals.

be best designed to make the selection of a constellation during an interval on the basis of blocks of L intervals as opposed to an interval by interval basis. It is demonstrated here that the performance of a MCS scheme can be improved by increasing L, however, at the expense of complexity.

A MCS scheme for a given OBC and a given throughput expansion factor η is best designed to achieve the highest possible minimum squared Euclidean distance (MSED), D^2_{min}, of the signals. The first step in maximizing D^2_{min} is to properly design the sequence divider in Fig. 2.1 to divide the sequence $\boldsymbol{v}$ into sequences $\boldsymbol{v_{Ex}}$ and $\boldsymbol{v_{Im}}$. This should be done uniformly (or as close to it as possible) across all codewords of the OBC. In other words, every codeword of the OBC should ideally feed the same number of bits into sequences $\boldsymbol{v_{Ex}}$ and $\boldsymbol{v_{Im}}$. Then the sequences $\boldsymbol{v_{Ex}}$ and $\boldsymbol{v_{Im}}$ need to be separately interleaved using two interleavers π_{Ex} and π_{Im} respectively to maximize D^2_{min}. The interleaved version of the sequence $\boldsymbol{v_{Ex}}$, $\boldsymbol{v_{Ex,\pi}}$, is transmitted by mapping every group of m_{Ex} bits of $\boldsymbol{v_{Ex,\pi}}$ on to a symbol of the selected M-ary constellation during each interval. Fig. 2.1 illustrates the transmission of any i^{th} block of L intervals. Based on the N_{PS} number of permutation bits from the interleaved implicit sequence $\boldsymbol{v_{Im,\pi}}$ of that block, the constellation sequence selector selects a sequence of constellations $\boldsymbol{C_i} = (C_{i1}, C_{i2}, ..., C_{iL})$. Then the constellation C_{il} ($l = 1, 2, ..., L$) is used for the transmission of the l^{th} symbol of the i^{th} block. The interleaver π_{Ex} and the mapper to map m_{Ex} bits of $\boldsymbol{v_{Ex,\pi}}$ on to the constellation C_{iL} can be designed as described in Hu et al. (2017).

B. Processing of Sequence $\boldsymbol{v_{Im}}$: In a block of L intervals, every combination of N_{PS} bits is assigned to a unique permutation of constellations. Hence, the $\boldsymbol{v_{Im}}$ sequence can be best processed by first designing π_{Im} to ensure that each coded bit of every codeword on $\boldsymbol{v_{Im}}$ is fed into different blocks of L intervals. The interleaver π_{Im} is designed to ensure that (a) no two coded bits of the same codeword of the OBC are placed in the same block of N_{PS} PS bits of the interleaved sequence $\boldsymbol{v_{Im,\pi}}$ and (b) no more than a pre-selected number of blocks n_b (≥ 0) of PS bits can include coded bits from the same two codewords of the OBC. Similar to the construction of π_{Ex} (Hu et al., 2017), an interleaver π_{Im} can be constructed as a constrained interleaver with N_{PS} rows by (i) feeding coded bits of every codeword that are sent on $\boldsymbol{v_{Im}}$ along rows, (ii) interleaving along rows to satisfy condition (b), and (iii) reading the interleaved sequence $\boldsymbol{v_{Im,\pi}}$ along columns. As a result, the interleaver π_{Im} guarantees that every coded bit of a codeword of the OBC transmitted implicitly makes a separate contribution to D^2_{min}. In addition to designing the intereleaver π_{Im}, the mapping of N_{PS} PS bits on to a permutation of constellations should be done to maximize the contribution made by the PS bits to D^2_{min}. Similar to the coding policy used to map blocks of m_{Ex} bits from $\boldsymbol{v_{Ex,\pi}}$ on to a constellation point, different combinations of N_{PS} bits from $\boldsymbol{v_{Im,\pi}}$ are mapped onto different permutations of constellations to maximize the SED for all one bit differences of the N_{PS} bits, and decreases the SED with increasing number of bit differences in the N_{PS} bits. A standard optimization software package such as IBM ILOG CPLEX Optimization Studio (IBM, 2016) that is widely known in the integer programming community can be used to determine the best mapping policy for higher values of N_{PS}. The above steps ensure that, the portion of any codeword transmitted on the sequence $\boldsymbol{v_{Im}}$ makes the highest possible SED contribution to that codeword. Hence, the above MCS design achieves the highest possible SED for every codeword thereby achieving the highest possible D^2_{min}. The values of n_a and n_b used in the interleavers π_{Ex} and π_{Im} respectively can be selected to ensure that the SED generated by two or more non-zero codewords of the OBC is guaranteed to be more than D^2_{min}, thereby ensuring that contributions from only single non-zero codewords of the OBC dominate the performance of a MCS scheme.

C. Selection of Constellations: Even though it is possible to numerically search for the best set of constellations S of a MCS, three classes of MCS schemes, MCS-A, MCS-P and MCS-AP are introduced here. MCS-A schemes employ $\theta_t = 0$ for $t = 2, 3, .., N$ and vary the amplitude r_t uniformly by maintaining the same separation, Δr, between any two adjacent amplitudes. Similarly, MCS-P schemes maintain $r_t = 1$ for $t = 2, 3, .., N$ and vary θ_t uniformly maintaining the same separation, $\Delta\theta$, between any two adjacent phases. Hence, the optimal MCS-A (MCS-P) schemes can be found by searching for the optimal Δr ($\Delta\theta$) that maximizes D^2_{min} subject to the constraint that the average symbol energy of the MCS scheme is equal to that of the constellation C_1. A MCS-A scheme with $N = N_1$ constellations and a MCS-P scheme with $N = N_2$ constellations are then combined to form a hybrid MCS-AP scheme with $N = N_1 \times N_2$ constellations.

D. Exemplary MCS Scheme: Consider a MCS-A scheme constructed with an $(8, 4)$ extended Hamming code ($d_m = 4$) as the OBC and a QPSK constellation C_1 with points $\{\pm a, \pm a\}$ and the mapping shown in Fig. 2.1 when $N = 3$, $L = 3$, $n_a = 0$, $n_b = 2$ and $N_{PS} = 4$. Among every set of five codewords of the OBC, four codewords feed 5 coded bits each to $\boldsymbol{v_{Ex}}$ and the three coded bits each to $\boldsymbol{v_{Im}}$ while the remaining codeword feeds 4 bits each to $\boldsymbol{v_{Ex}}$ and $\boldsymbol{v_{Im}}$ to achieve $\eta = 5/3$. Denoting the three amplitudes employed as $r_1 = \sqrt{2}a$, $r_2 = (1 - \Delta r)r_1$, and $r_3 = (1 + \Delta r)r_1$, where the value of Δr and a are found to maximize D^2_{min} subject to the average energy constraint $[1 + (1 - \Delta r)^2 + (1 + \Delta r)^2] = 3/(2a^2)$, which results in $\Delta r = 0.472$ and $a = 0.659$. It was numerically found that four or more sets of five codewords are sufficient to design interleavers π_{Ex} and π_{Im} that satisfy all conditions described in sections II-B and II-C. Table 2.1 lists the constellation sequences used for eight different combinations of $N_{PS} = 4$ implicit bits that achieve the highest possible SED for one bit differences. For every implicit bit combination $\boldsymbol{v_{Im}}$ mapped to the corresponding sequence of constellations $\boldsymbol{C(v_{Im})}$ in Table 2.1 , the implicit bit combination $[\boldsymbol{v_{Im}} \oplus (1000)]$ is mapped to $\boldsymbol{\hat{C}(v_{Im})}$, where $\boldsymbol{\hat{C}(v_{Im})}$ is obtained by swapping any $C_2(C_3)$ with $C_3(C_2)$ in $\boldsymbol{C(v_{Im})}$, thereby providing the sequence of constellations corresponding to the remaining eight combinations of PS bits, where $\oplus$ denotes modulo 2 addition. It is known that when mapped based on multiple intervals ($L > 1$), a coded stream can be properly interleaved and mapped to shrink the constellation without sacrificing performance (Hu et al., 2017). Hence, by employing large enough value of L on $\boldsymbol{v_{Im}}$, it is possible to employ all r_i values of MCS-A schemes close to one and all θ_i values of MCS-P schemes close to zero. Hence, the overall constellation used by a MCS scheme is a NM-ary constellation formed by replacing every point of the M-ary constellation by a cluster with N constellation points. Further, the clusters can be shrunk by increasing L. Hence, due to the use of mapping with $L > 1$ on $\boldsymbol{v_{Im}}$, a MCS scheme employs a more compact overall NM-ary constellation than an evenly spaced regular NM-ary constellation.

2.2 Detection and Performance Analysis

MCS signals can be decoded using soft iterative decoding by passing extrinsic information between the OBC and the modulator. The extrinsic information of the explicitly and implicitly transmitted bits from the OBC is obtained by soft decoding the OBC as in Lin and Costello (2004). The extrinsic information of explicit bits from the demodulator are extracted similar to the decoding of bit interleaved coded modulation with iterative decoding (Tran and Nguyen, 2006). The extrinsic information of implicitly transmitted bits can be extracted from the demodulator by (a) finding the likelihood of each constellation C_t during every l^{th} interval within a block of L intervals, $t = 1, 2, ..., N$, $l = 1, 2, ..., L$, (b) calculating the likelihood of every sequence of constellations $\boldsymbol{C_s} = (C_{s1}, C_{s2}, ..., C_{sL})$, $s = 1, 2, ..., 2^{N_{Ps}}$ and (c) calculating the likelihood of every implicit bit according to the mapping of PS bits on to sequences of constellations.

The asymptotic bit error probability (BEP) of a MCS scheme is derived from its MSED, D^2_{min}. It is assumed that the MCS scheme is constructed as described in section II with proper selection of n_a and n_b values which allows the BEP derivation to only consider the impact of single codewords of the OBC with weight d_m. Let d_{Ex} and $d_{Im} = (d_m - d_{Ex})$ respectively denote the number of explicitly and implicitly transmitted non-zero coded bits of any codeword with weight d_m. Since each explicitly transmitted bit can make a different SED contribution on the selected constellation, the contribution made by all d_{Ex} explicitly transmitted bits can be written as $D^2_{Ex} = \sum_{j=1}^{d_{Ex}} {D_j}^2$ where ${D_j}^2$ is the SED contribution made by the j^{th} explicit bit of that codeword on the constellation. In order to find the contribution made by the implicitly transmitted bits, first note that each implicitly transmitted bit makes a separate contribution to D^2_{min} because, by design, each implicit bit is placed in a different block of L intervals. The contribution made by any single implicit bit is bounded by the minimum SED between any two permutations of constellations $\boldsymbol{C_p} = (C_{p1}, C_{p2}, ..., C_{pL})$ and $\boldsymbol{C_q} = (C_{q1}, C_{q2}, ..., C_{qL})$ corresponding to two sequences of implicitly transmitted bits, $\boldsymbol{p} = (p_1, p_2, ..., p_{N_{PS}})$ and $\boldsymbol{q} = (q_1, q_2, ..., q_{N_{PS}})$ respectively, that differ only in one bit position, can be written as

$$G = \min_{\{\boldsymbol{p},\boldsymbol{q}\}} \sum_{l=1}^{L} |C_{pl} - C_{ql}|^2 \tag{2.1}$$

By considering all $(d_m - d_{Ex})$, the contribution made by explicitly transmitted bits of that codeword, D^2_{Im}, can be bounded by

$$D^2_{Im} \geq G(d_m - d_{Ex}). \tag{2.2}$$

Hence, the overall SED of the codeword in consideration can be bounded by the sum of (2.1) and (2.2), and hence the MSED of the MCS scheme is bounded by

$$D^2_{min} \geq \min[G + D^2_{Im}] \tag{2.3}$$

where, the minimization in (2.3) is taken over all codewords of the OBC with weight d_m. Denoting the average number of message bits in a codeword of the OBC with weight d_m by m_a and the number of codewords that generate a transmitted signal with SED=D^2_{min} by n_c, the BEP of a MCS scheme over and additive white Gaussian channel with power spectral density $N_0/2$ is approximately given by Lin and Costello (2004)

$$P_e \approx \frac{m_a n_c}{k} Q\left(\sqrt{\frac{D^2_{min}}{2N_0}}\right) \tag{2.4}$$

where, $Q(.)$ is the standard Q-function (Lin and Costello, 2004). It is interesting to note that with proper mapping of implicit bits on to sequences of constellations and proper design of π_{Im}, G in (2.1) can be increased by increasing L which in turn increases D^2_{min}. Hence, the performance of a MCS scheme can be improved by increasing L, however, at the expense of complexity. In general, constellations for MCS schemes and the block length L can be best selected to maintain a good balance between the likelihood values extracted from the demodulator for explicit and implicit bits. It is interesting to see from (2.2) that for any set of constellations, $C_1, C_2, ..., C_N$ that maintains a non-zero SED contribution between any two constellations, a value of L can be found to provide likelihood values for implicit bits comparable to those of explicit bits.

MCS schemes that enhance the throughput however requires increased decoding complexity. Compared with a coded scheme that employs a single M-ary constellation with iterative decoding. Specifically, the decoding complexity of a MCS scheme is increased due to (a) increased complexity in the extracting/updating of bit metrics by a magnitude of N due to the expansion of the overall constellation to a NM-ary constellation, and (b) due to the calculation of extrinsic information of implicit bits. The complexity of the additional step (b) can be deduced from the known complexity of decoding a (n, k) block code, which is $(2^k * n + 6n^2)$ per codeword (Hu et al., 2017), by setting $k = log_2(NML)$ and $n = L * m_{Im}$ as $O(\frac{NML(\eta-1)}{\eta} + 6L\frac{(\eta-1)^2}{\eta} log_2 M)$ per single decoding of implicit bits in a block of L intervals.

Table 2.1: Mapping of the implicit bits in the example.

$\boldsymbol{v_{Im}}$	$\boldsymbol{C(v_{Im})}$	$\boldsymbol{v_{Im}}$	$\boldsymbol{C(v_{Im})}$
[0000]	$C_2C_2C_2$	[1100]	$C_1C_2C_2$
[1010]	$C_2C_1C_2$	[1001]	$C_2C_2C_1$
[0101]	$C_3C_2C_2$	[0011]	$C_2C_3C_2$
[1110]	$C_2C_2C_3$	[0111]	$C_1C_2C_3$

2.3 Numerical Results

In this section, MCS-A, MCS-P and MCS-AP schemes are constructed as described in section II using an (8,4) extended Hamming code ($d_m = 4$) as the OBC and a QPSK constellation shown in Fig. 2.1 as C_1. Figs. 2.2 and 2.3 show the BEP variation of MCS-A and MCS-P schemes respectively with E_t/N_0 when $N = 3$ and $L = 2$ and 4, where E_t is the transmitted bit energy which is the bit energy of the explicit bits. Figs. 2.2 and 2.3 also show the corresponding theoretical bound with $L = 4$ given by (4) and the performance of a QPSK scheme in Fig. 2.1 with no implicit bits ($N = 1, \eta = 1$). As expected, Figs. 2.2 and 2.3 demonstrate that the performance of MCS schemes can be improved by increasing L, and also the theoretical bound matches well with the simulations. It is also seen that MCS schemes perform similar to a QPSK scheme that does not transmit any implicit bits. Hence, the MCS technique can increase the throughput by a factor η without any performance degradation. The BEP performance of all MCS schemes presented here have been found with iterative decoding described in section III with 10 iterations.

Fig. 2.4 compares the BEP variation of attractive MCS schemes with the bit SNR, $E_b/N_0 = \eta * E_t/N_0$. For comparison, BEP variation of rate 1/2 turbo coded schemes with 16-QAM and 64-QAM signaling employed in the LTE standard are also plotted. The MCS-AP scheme with $N = 16$, $L = 2$ and $\eta = 3$ shown in Fig. 2.4 is constructed by combining MCS-A and MCS-P schemes with $N = 4$ and $L = 2$. It is clear from Fig. 2.4 that the MCS schemes constructed with a simple (8,4) extended Hamming code can perform better than the turbo coded scheme employed in the LTE that use traditional higher order constellations

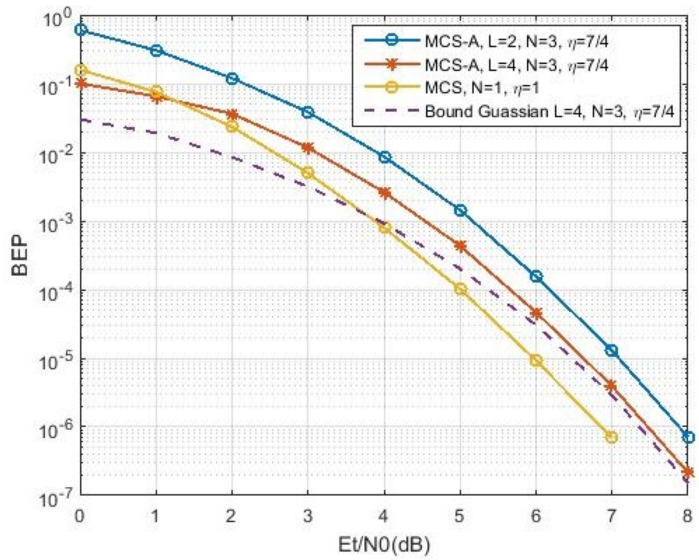

Figure 2.2: BEP variation of MCS-A schemes when $N = 3$.

for all practical BEP values down to 10^{-5}. Comparing the two MCS schemes, it is noted that the MCS-AP scheme with $\eta = 3$ performs better than the MCS-A scheme with $\eta = 2$ primarily due to its higher value of η. However if Fig. 2.4 was plotted with E_t/N_0 (instead of E_b/N_0) both those MCS schemes would perform almost the same. Comparing the decoding complexity for each data bit of MCS schemes with the LTE schemes (Hu et al., 2017) in Fig. 2.4, it is found that the MCS-A scheme requires 240 computations while LTE with 16-QAM requires 832 computations per iteration. Similarly, the MCS-AP scheme in Fig. 4 and LTE with 64-QAM require 880 and 916 computations per iteration respectively. Hence, in addition to performing better, the MCS schemes in Fig. 2.4 have lower decoding complexity in comparison with turbo decoding used in the LTE.

2.4 Conclusions

A multi-constellation signaling (MCS) technique has been proposed to implicitly transmit coded bits in addition to explicitly transmitted bits over the channel thereby increasing the throughput. A MCS scheme selects one constellation among a bank of constellations

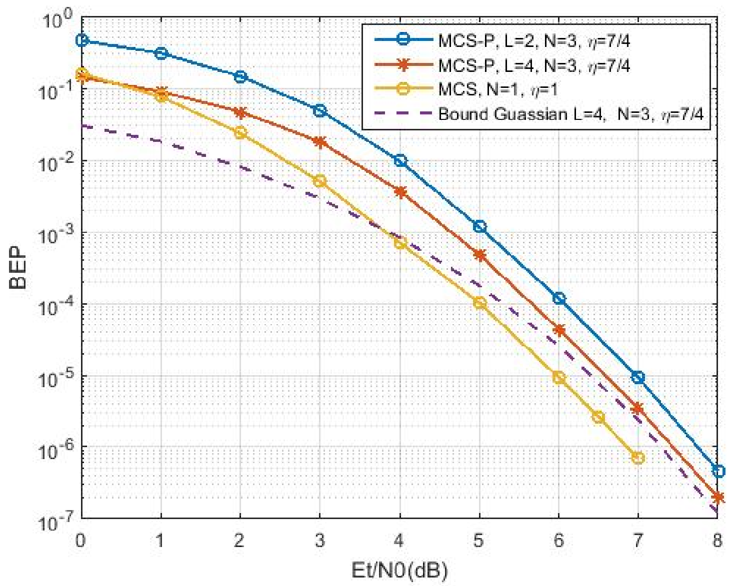

Figure 2.3: BEP variation of MCS-P schemes when $N = 3$.

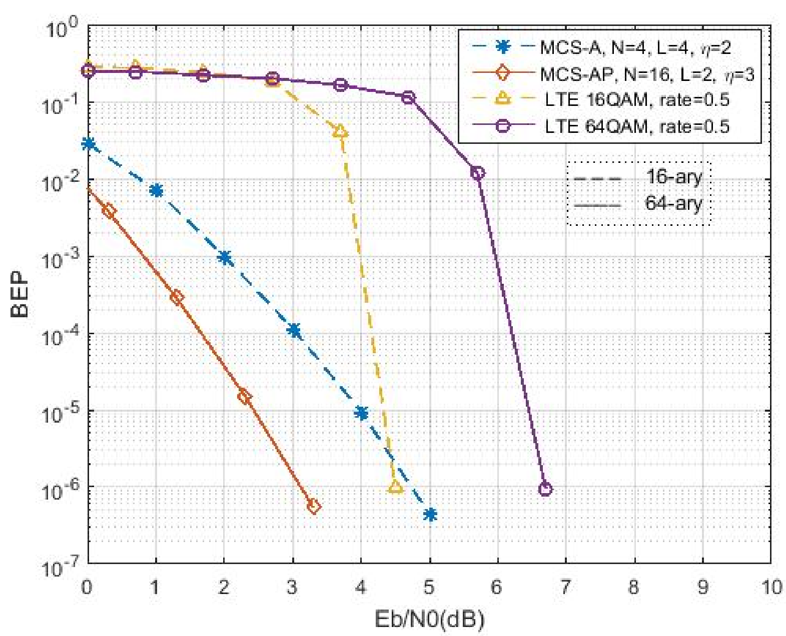

Figure 2.4: Comparison of attractive MCS with LTE.

during every interval based on a set of implicit bits. MCS schemes have been designed, analyzed and compared with standard signaling schemes that employ a single constellation. It has been demonstrated that a properly designed MCS scheme can double and triple the throughput over the corresponding traditional schemes that employ a single constellation without sacrificing performance. Compared with turbo coded signaling used in the LTE, MCS schemes that perform better with a lower decoding complexity have been presented.

CHAPTER 3

THROUGHPUT ENHANCING CONCATENATED CODES (TECCS)

3.1 Introduction

Concatenated codes have been considered in the literature to generate powerful codes using much simpler component codes. Concatenations can be parallel concatenations (Benedetto and Montorsi, 1996) or serial concatenations (Benedetto et al., 1998). In traditional concatenations the goal is to improve coding gain and generate powerful resulting concatenated codes. However, by inserting additional component codes, the rate and the overall throughput of the concatenation decrease unless the added code is a full rate code. In this chapter, a class of throughput enhancing concatenated codes (TECCs) is presented using the principle of concatenation to increase the throughput of a coded system. It has been demonstrated that TECCs can increase the throughput while maintaining performance close to or even better than the performance of individual component codes. TECC schemes that can increase the throughput of turbo code used in the 4G LTE standard by 25% to 37.5% without sacrificing its performance are reported.

3.2 Throughput Enhancing Concatenated Codes (TECCs)

3.2.1 General Description

In order to describe the generation of TECCs, let us consider a single code C. Traditionally, when C is used in isolation, coded bits of C are transmitted using a selected 2^m-ary signal constellation possibly with a channel interleaver prior to mapping groups of m coded bits onto a constellation point (Lin and Costello, 2004). In contrast, a TECC scheme modifies coded bits of C according to coded bits of a second code C' prior to transmission. However, this modification is done without increasing the original length of the coded sequence of C.

Figure 3.1 illustrates the encoding structure of a TECC. In a TECC, two message sequences, an explicit message sequence m_{Ex} and an implicit message sequence m_{Im}, are separately processed as illustrated in Fig. 3.1. The coded bits of C are processed on the basis of blocks of n bits. The value of n can be pre-selected based on the explicit code C. If C is a (n, k) block code, n can be chosen as the block length n. If C is a long code such as a turbo code or a low-density parity-check code (LDPC) (Chung et al., 2001), n can be chosen numerically to achieve good performance. Fig. 3.1 focuses on the processing of a general i^{th} block of n bits, $v_i = (v_{i1}, v_{i2}, , ..., v_{in})$, $i = 1, 2, ..., \rho$, among a total of ρ blocks that form the frame of $N = n\rho$ number of coded bits of C. The implicit sequence is separately processed by passing the implicit message sequence m_{Im} through the code C' to generate the coded stream v'_{Im}. The coded sequence v'_{Im} is interleaved by an interleaver π to form the interleaved sequence $v' = \pi(v'_{Im})$. The coded interleaved sequence v' is divided into the same number of blocks ρ with n_s number of bits in each block. Hence, any i^{th} block of the sequence v' originated by the code C' can be denoted by $v'_i = (v'_{i1}, v'_{i2}, ..., v'_{in_s})$, $i = 1, 2, ..., \rho$. As shown in Fig. 3.1, the block v'_i is then mapped onto a unique n-bit long error sequence $e_i = (e_{i1}, e_{i2}, , ..., e_{in})$, $i = 1, 2, .., \rho$. The selected error sequence e_i is then added to the coded block v_i of C to form the modified block $v_{s,i} = (v_{s,i1}, v_{s,i2}, , ..., v_{s,in})$ which is transmitted over the channel, where, $v_{s,ik} = v_{ik} \oplus e_{ik}$, $k = 1, 2, ..., n$, and $\oplus$ denotes modulo-2 addition. In Fig.3.1, it is illustrated how $N = n\rho$ coded bits of C are transmitted on the explicit sequence by transmitting $c = n\rho/m$ symbols on a 2^m-ary constellation while transmitting $N' = n_s\rho$ number of coded bits of C' implicitly from the implicit sequence. As a result, every transmitted symbol carries m coded bits of C over the explicit channel and $n_s m/n$ number of coded bits of C' during every interval. Hence, in total $N = n\rho$ coded bits of C are transmitted explicitly while transmitting $N' = n_s\rho$ coded bits of C' implicitly in a frame.

The addition of e_i to v_i is equivalent to inverting the non-zero bit positions of e_i on v_i to form $v_{s,i}$ prior to transmission. Note that the length of v_s is the same as that of v, and as a result no coded bits of C' are transmitted explicitly over the channel. However, through the selection of the error sequences e_i, coded bits of C' influence the sequence $v_s = (v_{s,1}, v_{s,2}, ..., v_{s,\rho})$. This influence of C' on the transmitted signal allows the recovery of coded bits of both C and C' from the received version of v_s. Hence, a TECC transmits altered coded bits of C explicitly while transmitting coded bits of C' implicitly through the selection of the error sequences. As a result, throughout this study, C is referred to as the explicit code while C' is referred to as the implicit code of the TECC. It is demonstrated here that TECCs can be properly designed to decode coded bits of both codes C and C' using the received version of v_s by employing iterative decoding at the receiver. In fact, it is demonstrated here that the performance of coded bits of C of a TECC is very close to that of traditional transmission of coded bits of C in isolation without any involvement of a second code C'. For comparison, the coded scheme with only the code C is referred to as "original C scheme" and its decoding as "original C decoding" throughout this study. Hence, a TECC is capable of increasing the throughput by transmitting coded bits of C' implicitly without sacrificing the performance of the original C decoding. Note that, the role of the second code C' in a traditional concatenation of two codes C and C' is to improve the original coding power of C which however lowers the overall rate and the throughput of the concatenation (Benedetto et al., 1996). In contrast, the function of C' in a TECC is not to try to improve the power of C, but instead to increase the overall rate and the throughput of the concatenation by using the coding power of C. It is important to note that the addition of e to v to form v_s expands the code space of C by a factor N_e, however, without increasing the code length n, where, N_e is the total number of error sequences used by the TECC. As a result, there are $2^k N_e$ valid sequences of v_s within any single block of n transmitted bits of the concatenation which allows the transmission of n_s additional coded bits of C' implicitly, where $k = R_1 n$ is the number of explicit message bits in the segment of n coded bits of C and R_1 is the rate of the explicit code.

3.2.2 Impact of the Error Sequence

In the design of a TECC, it is important to properly select the set of all allowed error sequences, S_e. One particular error sequence e_i is chosen among the N_e number of total error sequences in S_e based on $n_s = \lfloor log_2 N_e \rfloor$ number of coded bits of C', where $\lfloor . \rfloor$ denotes the standard floor function. Hence, it is desirable to increase N_e as much as possible to increase n_s and to increase the throughput of the concatenation. Specifically, if R_1 and R_2 are the rates of C and C' respectively, the effective rate of the TECC is calculated by realizing that the code C transmits $R_1 n$ number of message bits explicitly and C' transmits $R_2 n_s$ number of message bits implicitly by transmitting n bits of v_s. Hence, the effective rate of the TECC, R, is given by

$$R = (R_1 n + R_2 n_s)/n \tag{3.1}$$

As a result, the rate of C is increased by a rate enhancement factor $\eta = R_2 n_s / R_1 n$.

In order to extract useful information about coded bits of C' from the received signal r during every interval, it is necessary to choose the set of error sequences S_e carefully. Specifically, error sequences in S_e should be chosen so that the code C can still recover its coded sequence v even after the addition of the error sequence e. Clearly, the coding power of C decreases due to the addition of the error sequence e. However, if the weight of e is small enough and the code C is reasonably powerful, then C is generally able to still decode v fairly accurately even in presence of e. For example, consider C to be the (8,4) extended Hamming code that has minimum Hamming distance (MHD) 4 and is capable of correcting all single errors. Hence, in this case, eight distinct weight one error sequences, $e_1 = (10000000), e_2 = (01000000), ..., e_8 = (00000001)$, can be chosen in S_e. Further, when any of the error sequences e_i, $i = 1, 2, ..., 8$, is added to v, the code C would become a weight 2 code and it can still recover v as the signal to noise ratio (SNR) increases. As a result, selecting the specific error sequence e_i based on $n_s = 3$ bits of v', it is possible to construct a TECC with $n = 8$, $n_s = 3$ both (n, k) block code C and (n', k') block code C' as the (8,4) code. The resulting TECC is capable of transmitting eight coded bits of C explicitly from v and three coded bits of C' implicitly from v'. When a (n, k) block code that has error correcting capability t is used as C, error sequences with weight $1, 2, ..., t$ can be used in S_e. The number of error sequences N_e, and thereby the number of implicit bits n_s, can be increased by employing error sequences with all possible weights. However, it is noted here that as the weight of the employed error sequences increases, the code C gets weaker and hence, the TECC would not perform well at lower SNR values.

However, when C is a long code like a turbo code or a LDPC code that does not have short codewords, the selection of n and the set of error sequences can be modified. The easiest way to design a TECC with such a long code C is to pre-select a value of n and select only the set of all weight one error sequences in every segment of n bits. A value of n is selected to ensure that the explicit code C can perform reasonably well even with the addition of the error sequence e. Since there are n number of error sequences with weight one in a segment of n bits, such a TECC can transmit an additional $n_s = \lfloor log_2 n \rfloor$ number of coded bits from v' implicitly when a block of n bits coded bits of v are transmitted explicitly over the channel. Preferably, n can be chosen as an integer power of 2. Since the objective is to increase n_s/n as much as possible, it is desirable to choose the smallest possible value of n. Following the above description, TECC encoding of a long code C is done by: (a) dividing the coded sequence of C into segments of n bits, (b) selecting one coded in each such segment based on $n_s = \lfloor log_2 n \rfloor$ number of implicitly encoded bits of v', and (c) inverting the selected bit in each segment before transmission.

3.2.3 Error rate variation of TECCs

As stated before, the TECC signals are decoded by running iterations between the explicit code C and the implicit code C'. The details of the decoder are presented later in section

IV. In order to recover useful information about the implicit coded sequence v' from the decoding of C, it is necessary for C to decode v reasonably well even in the presence of the error sequence e and noise of channel without any apriori knowledge of e. Therefore, compared with regular C coded signals, TECCs require a higher SNR to decode C reasonable well to produce reliable soft information about e. However, once reliable information about e becomes available, the implicit code C' and the mapper M can almost correctly estimate the error sequence within each segment of n bits of the sequence v. Therefore, at higher SNR, the performance of the explicitly transmitted sequence v of a TECC approaches that of the corresponding original C decoding. Since the information about the error sequence e improves as the decoding of the explicitly coded sequence v improves, decoding of the implicitly coded sequence v' improves as the decoding of the sequence v improves too. Hence, as decoding of C' improves decoding of C approaches original C decoding. Similarly, as decoding of C improves, decoding of C' improves as well. Therefore, the implicit code C' also can perform well at higher SNR values, and further, C' can be chosen to perform even better than both original C coding and original C' decoding. In fact, the numerical results presented later demonstrate that when both C and C' are the same turbo code, the performance of v' is slightly better than that of v which is very close to that of original C coding.

As a result the overall performance of explicit and implicit sequences is slightly better than that of original C decoding at high SNR when both C and C' are the same. This also suggests that the implicit code C' can be slightly weaker than the explicit code C and maintain the overall performance of the TECC close to that of the corresponding original C decoding. However, at lower SNR, the explicitly coded sequence v is worse than regular C coding as the SNR needs to be higher for C to start producing useful information about e. Since the decoding of C' relies on the decoding of C, the performance of the implicit sequence v' is also worse at lower SNR. Therefore, the overall performance of a TECC is worse at lower SNR values than regular C decoding. However, when the SNR is increased until the explicit code C starts to function fairly well in the presence of the error sequence e, the performance of C can become close to that of regular C decoding, while the performance of C' can be even better than that of regular C decoding depending on the selection of C'.

The SNR at which the performance of C starts to be almost the same as that of the corresponding regular C decoding gets higher as the weight of the error sequences in S_e gets higher and/or n gets smaller. Depending on the desired bit error rate (BER), the value of n and the highest weight of the error sequences can be selected to ensure that the TECC scheme performs almost the same as the corresponding original C decoding at the desired error probability. However, the value of n for a short block code can be chosen as the length of the codewords. The maximum allowed weight of the error sequences in S_e of such a short block code can be selected based on the error correcting capability of the code and the desired operating error probability. In this study, we consider TECCs constructed by selecting an appropriate value of n and employing only the set of all weight one error sequences in S_e.

3.3 Construction of TECC Schemes

As shown in Fig. 3.1, a TECC can be constructed by selecting component codes C and C' and designing the interleaver π and the mapper M. In order to achieve similar performance for both explicit and implicit message sequences, both C and C' can be chosen as the same code. However, the numerical results presented later indicate that C' can be slightly less powerful than C. In applications where the implicit sequence can afford to maintain slightly a higher error rate, the code C' can be made significantly less powerful than the code C thereby making C' a higher rate code than C. According to (3.1), when C and C' are the same, $\eta = n_s/n$, while if C' is a higher rate code ($R_2 > R_1$), $\eta > n_s/n$.

3.3.1 Design of the Implicit Interleaver π

As shown in Fig. 3.1, every block of n_s bits at the output of the interleaver π is used to generate a n-bit long error sequence. The function of the interleaver is to generate blocks of n_s bits at the output of the interleaver so that any non-zero coded sequence of C' would influence as many error sequences as possible. This can be done by maximizing the minimum number of n-bit long error sequences influenced by a coded sequence of C'. Therefore, the interleaver π can be designed as a standard row-column interleaver with n_s rows and ρ columns by feeding low weight sequences of v' along rows of π and reading the interleaved sequence along columns. The i^{th} column of π decides the n-bit long error sequence e_i for the i^{th} block of coded bits of C, v_i, $i = 1, 2, ..., \rho$. For a block code C', the low weight sequences are the low weight codewords of C'. However, when π is a standard row-column interleaver, every combination of low weight sequences on different rows that occupy the same columns impact the same difference in the Euclidean distance in the transmitted sequence v_s. This increases the multiplicity of the minimum Euclidean distance event and thereby degrading the performance of the implicit sequence. Therefore, the interleaver π can be further improved by introducing an additional constraint as in constrained interleaving in Hu et al. (2017) and ensuring that no two low weight sequences in different rows share more than a pre-selected number of columns $n' < d'_{min}$, where, d'_{min} is the MHD of C'. The value of n' can be chosen to achieve a good performance for the implicit sequence and the overall sequence.

However, when C' is a long code, such as a turbo code or an LDPC code, the value of n_s is much smaller than the length of a low weight sequence. Therefore, in such TECCs, the interleaver π can be eliminated in the design. All TECCs constructed in this study with turbo component codes are constructed without an interleaver π. It is demonstrated here that such TECCs without an interleaver π can still perform close to that of the original C coded scheme.

3.3.2 Design of the Mapper M

The function of the mapper is to map each of the 2^{n_s} number of n_s-tuples onto a n-bit long error sequence. When S_e contains only the set of all single weight error sequences, this can be done easily by assigning every n_s-tuple sequence of v' to an n-bit long error sequence with weight one in any order. However, if the maximum weight of error sequences allowed in S_e is more than one, the mapping of n_s-tuples onto error sequences should be done to achieve the best performance. Specifically, mapping can be done so that low weight n_s-tuples are mapped to the highest weight error sequences to create the maximum possible Euclidean distance for all low weight coded sequences of v'.

3.4 Decoding of TECC Schemes

In this section, the detection of the two message sequences, the explicit message sequence m_{Ex} and the implicit message sequence m_{Im} using the received version of the transmitted sequence v_s is described. Let (a) the length of the transmitted sequence v_s, which is transmitted using the symbol sequence $s = (s_1, s_2, ..., s_c)$, be $N = n\rho = mc$, where each s_k, $k = 1, 2, ..., c$, is a symbol on the 2^m-ary constellation used for transmission, and (b) the corresponding received sequence $r = (r_1, r_2, ..., r_c)$, where $r_k = s_k + n_k$, $k = 1, 2, ..., c$, and n_k is the additive white Gaussian noise (AWGN) with two-sided power spectral density (PSD) $N_0/2$.

Fig. 3.2 shows the general structure of the soft iterative decoder. The decoding starts from extracting the log-likelihood ratio (LLR) value of each transmitted bit $v_{s,k}$, $L(v_{s,k})$, $k = 1, 2, ..., N$, given by

$$L(v_{s,k}) = ln[\frac{P(r_k|v_{s,k} = 1)}{P(r_k|v_{s,k} = 0)}] + ln[\frac{P(v_{s,k} = 1)}{P(v_{s,k} = 0)}] \tag{3.2}$$

where, $ln(.)$ denotes the natural logarithm. The first term of (3.2) is the channel information, denoted by $L_{ch}(.)$, extracted from the received signal r as in Imai and Hirakawa (1977), which is done only once, and the second term of (3.2) is the apriori information of v_s, denoted by $L_{apr}(v_s)$, which is zero in the first iteration and is updated during the iterations.

Even though the code space of C expands due to the addition of the error sequence e, the decoding presented here does not operate on the expanded coded space. The proposed decoding algorithm is developed to (a) soft decode C, (b) soft decode C', (c) use the mapper M and the demapper M^{-1} to find the LLR value of each e_i using the LLR value of each v'_j, and vice versa, $i = 1, 2, ..., N$ and $j = 1, 2, ..., Nn_s/n$, (d) uses the relationship $v_{s,i} = v_i \oplus e_i$, to calculate the LLR value of each $v_{s,i}$ using those of the corresponding v_i and e_i. Since e_i is generated based on an independent implicit stream which is independent of v_i, $L_{apr}(v_{s,i})$ can be written as

$$\begin{aligned} L_{apr}(v_{s,i}) = ln[\frac{P(v_{s,i}=1)}{P(v_{s,i}=0)}] \\ &= ln[\frac{P(v_i=1,e_i=0)+P(v_i=0,e_i=1)}{P(v_i=0,e_i=0)+P(v_i=1,e_i=1)}] \\ &= ln[\frac{P(v_i=1)P(e_i=0)+P(v_i=0)P(e_i=1)}{P(v_i=0)P(e_i=0)+P(v_i=1)P(e_i=1)}] \\ &= ln[\frac{exp(L_{apr}(v_i))P(v_i=0)P(e_i=0)+exp(L_{apr}(e_i))P(v_i=0)P(e_i=0)}{P(v_i=0)P(e_i=0)+exp(L_{apr}(v_i))exp(L_{apr}(e_i))P(v_i=0)P(e_i=0)}] \\ &= ln[\frac{exp(L_{apr}(v_i))+exp(L_{apr}(e_i))}{1+exp(L_{apr}(v_i)+L_{apr}(e_i)}] \end{aligned} \tag{3.3a}$$

$$\approx -sign(L_{apr}(v_i).L_{apr}(e_i)).min(|L_{apr}(v_i)|,|L_{apr}(e_i)|) \tag{3.3b}$$

where, $L_{apr}(v)$ and $L_{apr}(e)$ are apriori information of v and e respectively. Equation (3.3b) has been obtained using the standard approximation of $ln[\frac{(exp(a)+exp(b))}{(exp(c)+exp(d))}] \approx [max(a,b) - max(c,d)]$ (Li et al., 2004), (Perişoară and Stoian, 2008).

Even though the pair $v_{s,i}$ and v_i (or $v_{s,i}$ and e_i) are not bitwise independent, $v_{s,i}$ and extrinsic information of v_i (or e_i) which is calculated based on other v_j (or e_j) $j \neq i$, can be assumed independent. Therefore, while working with extrinsic information of v_s and v (or v_s and e), the (3) can still be used to calculate the apriori information of e (or v).

Based on the above approach, TECC decoding begins from $L(v_{s,l})$, $l = 1, 2, ..., N$, values calculated according to (3.2) and following the steps listed below:

1. Using $L_{ch}(v_{s,l})$ values and any available information of the error sequence $e(l)$, $L_{ext}(e_l)$, find the information of each bit $v(l)$, $l = 1, 2, ..., N$, according to the (3) to form the sequence $L_i(v) = (L_i(v_1), L_i(v_2), ..., L_i(v_N))$.

2. Using the sequence $L_i(v)$ found in step 2, soft decode C and obtain the extrinsic information $L_{ext}(v_l)$ of each coded bit v_l, $l = 1, 2, ..., N$, according to the code C to form the extrinsic information sequence $L_{ext}(v) = (L_{ext}(v_1), L_{ext}(v_2), ..., L_{ext}(v_N))$.

3. For each l, $l = 1, 2, ..., N$, use $L_{ch}(v_{s,l})$ values and $L_{ext}(v_l)$ found in step 2, find the information sequence of e according to (3) to form the sequence $L_i(e) = (L_i(e_1), L_i(e_2), ..., L_i(e_N))$.

4. Similar to soft decoding a block code, for every j^{th} block of n_s bits of v'_j, $j = 1, 2, ..., \rho$, do the following two steps: (a) using $L_i(e)$, calculate a metric M_k, $k = 1, 2, ..., 2^{n_s}$, for each valid error sequence e_k, and (b) using the demapping policy M^{-1} used to demap each combination of n-bit long error sequence e_k, $k = 1, 2, ..., 2^{n_s}$, onto a corresponding n_s-bit long block of v', v'_k, calculate the LLR value of each i^{th} coded bit of v'_j, $i = 1, 2, ..., n_s$. The steps (a) and (b) above are denoted by $M^{-1}(L)$ in Fig. 2. Upon completion of step 4, all LLR values bits of v' are available to form the sequence $L_i(v') = (L_i(v'_1), L_i(v'_2), ..., L_i(v'_{N'}))$, where $N' = Nn_s/n$.

5. De-interleave $L_i(v')$ sequence according to π^{-1} to obtain the sequence $L_i(v'_{Im}) = (L_i(v'_{Im,1}), L_i(v'_{Im,2}), ..., L_i(v'_{Im,N'}))$.

6. Soft decode C' and find the extrinsic information of each coded bit $L_{ext}(v'_{Im})$, to form the sequence $L_{ext}(v'_{Im}) = (L_{ext}(v'_{Im,1}), L_{ext}(v'_{Im,2}), ..., L_{ext}(v'_{Im,N'}))$.

7. Interleave the sequence $L_{ext}(v'_{Im})$, according to π to form the sequence $L_{ext}(v') = \pi(L_{ext}(v'_{Im}))$.

8. Perform the reverse operation of step 4 to obtain the extrinsic information of each error bit of any n-bit long error sequence e_j, $j = 1, 2, ..., \rho$, based on the code C'. Similar to step 4, this is done on the basis of blocks. For every block j, perform the following two steps: (a) using $L_{ext}(v')$ found in step 7, calculate a metric M_k, $k = 1, 2, ..., 2^{n_s}$, for every combination v' bits in that block, and (b) using the mapping policy M used to map each combination of n_s-bit long sequence of v'_k onto the corresponding n-bit long error sequence e_k, calculate the LLR value of each i'^{th} error bit of e_j, $i' = 1, 2, ..., n$. The steps (a) and (b) above are denoted by $M(L)$ in Fig. 2. Upon completion, the extrinsic information of each error bit e_l, $L_{ext}(e_l)$, is found to form the sequence $L_{ext}(e) = (L_{ext}(e_1), L_{ext}(e_2), ..., L_{ext}(e_N)))$.

   Using the extrinsic information obtained for each e_l, $l = 1, 2, ..., N$ in step 8, go back to step 1 for the next iteration. Upon completion of the iterations, the explicit message stream m_{Ex} and the implicit message stream m_{Im} are found as illustrated in Fig.3.2.

Note that after running several initial TECC decoding iterations according to steps 1 through 8 listed above, the decoding of C' in step 6 would most likely have very few or no errors. Therefore, at that point, the above algorithm can be modified to obtain a hard decoded sequence v'_{Im} from the LLR values of v'_{Im} obtained in step 6. Then use that hard decoded sequence v'_{Im} in steps 7 and 8. As a result, step 8 would simply reduce to selecting the n-bit long error sequence according to the output of the mapper M corresponding to each of the n_s-bit long segments of v' obtained in step 6. This modified version of the decoding algorithm according to steps 1 through 8 above is referred to as the "modified TECC decoding algorithm". Hence, TECC decoding considered here runs a preselected N_1 number of initial iterations followed by a preselected N_2 number of iterations according to the modified decoding algorithm. The values of N_1 and N_2 can be chosen depending on the component codes C and C', and the frame length to achieve best performance. It was numerically found that by employing few iterations of the modified TECC decoding after several initial TECC decoding iterations improves performance over using the same initial TECC decoding algorithm according to steps 1 through 8 throughout the iterations.

3.5 Numerical Results

In this section, the simulated BER variations of TECC schemes are presented over an additive white Gaussian noise channel with two-sided power spectral density $N_0/2$. All TECC schemes are constructed as described in section III and decoded as described in section IV. Throughout this section QPSK modulation is considered for the transmission of the TECC transmitted sequence v_s. The performance of TECC schemes are compared with those of original C and original C' decoding.

Numerical results are first presented when both C and C' are the $(8,4)$ extended Hamming code (Lin and Costello, 2004). In this case $n = 8$, $n_s = 3$, and one bit out of each eight-bit long codeword is selected based on $n_s = 3$ implicit coded bits and flipped before transmission. The interleaver π is designed as described in section III-A. Fig. 3.3 shows the BER variations of the explicit and implicit message sequences along with that of the overall BER variation. The BER variations in Fig.3.3 have been obtained using $N_1 = 5$ initial decoding iterations followed by $N_2 = 2$ modified decoding iterations. For comparison, the BER variation of the original $(8,4)$ code using soft decoding is also shown in Fig. 3.3. It is seen that the performance of the explicit and implicit sequences follow the description presented in II-C. Further, it is seen that the performance of the TECC is almost the same as that of original $(8,4)$ decoding at high SNR values. As a result, the TECC scheme shown in Fig. 3.3 is capable of transmitting three additional coded bits of C' implicitly when eight coded bits of C are transmitted explicitly. Therefore, the TECC scheme can increase the throughput of the original $(8,4)$ code by a throughput expansion factor $\eta = 0.37$ without sacrificing performance at high SNR.

Figs. 3.4 through 3.7 show the BER variations when both C and C' are rate 1/3 turbo codes, each of which is constructed using two eight state component codes as in the current long term evolution (LTE) standard (Martin and Taylor, 2001). As stated in section II-B, all TECCs with component turbo codes presented here are constructed by selecting $n = 16$ and flipping one out of 16 bits selected based on $n_s = 4$ coded bits of C'. As a result these TECCs have a throughput expansion factor $\eta = 0.25$, and their transmission rate on the implicit sequence is 25% of that on the explicit sequence. Therefore, every four frames transmitted on the explicit sequence, the implicit sequence can transmit one frame of the turbo code thereby increasing the throughput of the original turbo code C by 25%. All results with component turbo codes have been obtained using $N_1 = 3$ initial decoding iterations followed by $N_2 = 2$ modified decoding iterations. Further, as stated before in section III-A, all TECC schemes with component turbo codes have been constructed without an interleaver π. It is seen from Figs. 3.4 through 3.7 that the BER variations of the explicit and implicit sequences and that of the overall sequence follow the description in section II-C, and the overall performance of the TECC is very close to that of the original turbo code for all BER values below 10^{-4}. Therefore, the 25% throughput enhancement provided by the TECC comes with no penalty in performance at practical BER values. It is also seen from Figs. 3.4 through 3.7 that the BER variation of the implicit sequence and the overall BER variation are slightly better than that of the BER variation of the explicit sequence. Therefore, as stated before, C' can be made slightly weaker and still maintain the overall TECC performance close to that of the original turbo code. In order to justify the above statement, Fig. 3.8 shows the BER variation of the TECC constructed with a rate 1/3 turbo code C and a rate 1/2 turbo code C' when the frame size of both turbo codes is 6144. As a result, according to (3.1), this TECC increases the throughput of the original C code by $\eta = 0.375$. It is seen from Fig. 3.8 that the overall TECC is close to that of the original C code, and therefore, the 37.5% throughput enhancement comes with minimal performance loss. For comparison,

Fig. 3.8 also shows the performance of the original C' code. It is interesting to note that the performance of the implicit sequence that employs the code C' is significantly better than that of original C' decoding. This is because the implicit sequence gets help from the more powerful code C during decoding. As a result the performance of a weaker code can be significantly improved by using it as the implicit code C' of a TECC and employing a more powerful code C as the explicit code.

It is important to note that the latency of a TECC is higher compared with that of the original C code. For example, in TECC schemes presented in Figs. 3.4 through 3.8, the explicit code needs to decode four of its frames with one frame of the implicit code. Therefore, the first frame of the explicit sequence needs to wait until all four of its frames to arrive before starting the TECC decoding as described in section IV with the completed implicit frame. In order to eliminate that delay, the code C' can be selected to have a smaller frame size. For example, C can be chosen as a rate 1/3 turbo code with frame size 1024 message bits, C' can be chosen as a rate 1/3 turbo code with frame size 256 message bits. With that selection, as soon as the explicit code completes transmission of each of its frames, the implicit code also completes its corresponding frame. Therefore, TECC iterations can start at the end of every frame on the explicit sequence eliminating the increase in delay. Fig. 3.9 shows the BER variations of the above scheme that employs a rate 1/3 turbo code with frame size 1024 bits on the explicit sequence and a rate 1/3 turbo code with frame size 256 bits on the implicit sequence. It is seen from Fig. 3.9 that the overall performance of the TECC is slightly worse but is close to that of original C decoding. The above TECC increases the throughput by 25% without increasing the latency and with a slight degradation in performance. However, the gain in throughput offered by TECCs can only be achieved at the expense of decoding complexity (Hassan et al., 2012). The decoding complexity of a TECC scheme is comparable to that of the corresponding original C and original C' codes due to the fact that in each iteration, TECC decoder decodes each component code C and

C', and performs calculations in the mapper and demapper (steps 4 and 8 of the decoding algorithm) as described before.

3.6 Information Theory Point of View

3.6.1 Capacity of a TECC

So far, it has been demonstrated that TECCs can be constructed to additionally transmit an implicit coded sequence implicitly without sacrificing performance by transmitting only an altered explicit coded sequence over the channel. The capacity of a TECC is calculated by observing the similarity of a TECC with a multi-user system (Ahlswede and Han, 1983) that transmits multiple sequences simultaneously similar to a TECC transmitting an explicit sequence and implicit sequence simultaneously. In order to be consistent with the information theory literature (Gallager, 1968), we denote a TECC subjected to a power constraint P, by $\mathfrak{I} :< X_{Ex}, X_{Im}, n >$ where, X_{Ex} is the explicit codeword with binary symbols $\{-\sqrt{P}, +\sqrt{P}\}$ and X_{Im} is the implicit codeword that is used to flip one bit out of n-bit long explicit codeword. Since the power P remains the same even with flipping, the transmitted sequence $X : (x_1, x_2, ..., x_n)$ satisfies the following power constraint

$$\frac{1}{n}\sum_i E[x_i^2] = E[X^2] = E[X_{Ex}^2] \leq P \tag{3.4}$$

Let $C(\mathfrak{I}) = maxI(X_{Ex}, X_{Im}; Y)$ be the information capacity of a memory-less Gaussian channel with a discrete input X and output Y, achieved by TECC $\mathfrak{I}$ with power constraint P and noise variance σ^2 where, $I(X_{Ex}, X_{Im}; Y)$ denotes the mutual information of (X_{Ex}, X_{Im}) and Y. Therefore, in order to calculate the information capacity of a TECC, we first find the above mutual information. Using the chain rule (Gallager, 1968), $I(X_{Ex}, X_{Im}; Y)$ can be written as

$$I(X_{Ex}, X_{Im}; Y) = I(X_{Ex}; Y) + I(X_{Im}; Y|X_{Ex}) = I_{Ex} + I_{Im} \tag{3.5}$$

where, I_{Ex} is the mutual information of the X_{Ex} and Y and I_{Im} is the conditional mutual information of X_{Im} and Y conditioned on X_{Ex}, when optimal decoding is applied for decoding of X_{Ex}. The mutual information of a memory-less Gaussian channel with a discrete uniform input alphabet X with K symbols is derived by Csiszar and Körner (2011)

$$I(X;Y) = H(Y) - H(Y|X) = log(K) - 1/K \sum_{x_k} \int_y log[\sum_{x'_k} - \frac{(y - x'_k + x_k)^2 - y^2}{2\sigma^2}]dy \tag{3.6}$$

Similarly, I_{Ex} can be derived as

$$I_{Ex} = H(Y) - H(Y|X_{Ex}) = 1/2 \sum_{x_{Ex,k}} \int_y log[\frac{f(y|x_{Ex,k})}{\sum_{x_{Ex,k'}} \frac{1}{2} f(y|x_{Ex,k'})}]dy \tag{3.7}$$

where $x_{Ex,k}$ can be either $\{-\sqrt{P}, +\sqrt{P}\}$ with equal probability. The channel noise in (3.6) and (3.7) is modeled by a Gaussian probability density function (pdf) with zero mean and standard deviation σ, $N(0,\sigma)$.

Noticing that (a) one out of n bits of x_{Ex} is flipped, (b) y is $N(x_{Ex},\sigma)$ when x_{Ex} is not flipped, and (c) y is $N(-x_{Ex},\sigma)$ when x_{Ex} is flipped, the conditional pdf $f(y|x_{Ex} = \pm\sqrt{P})$ in (3.7) can be written as

$$f(y|x_{Ex} = \pm\sqrt{P}) = \frac{1}{\sqrt{2}\pi\sigma^2}[\frac{1}{n}exp(-\frac{(y \pm \sqrt{P})^2}{2\sigma^2}) + \frac{n-1}{n}exp(-\frac{(y \mp \sqrt{P})^2}{2\sigma^2})] \tag{3.8}$$

Then by following Costa and El Gamal (1987), the mutual information I_{Im} in (3.5) can be found as

$$I_{Im} = \sum_{x_{Ex,\ell}} \sum_{x_{Im,k}} Pr(x_{Im,k}, x_{Ex,\ell}) \int_y log[\frac{f(y|x_{Im,k}, x_{Ex,\ell})}{\sum_{x_{Im,k'}} Pr(x_{Im,k'}, x_{Ex,\ell}) f(y|x_{Im,k'}, x_{Ex,\ell})}]dy \tag{3.9}$$

where, $Pr(x_{Im,k}, x_{Ex,\ell})$ is the joint probability distribution of explicit and implicit sequences which can be written as

$$Pr(x_{Im}, x_{Ex}) = \begin{cases} \frac{1}{2n} & x_{Im} = \pm 2\sqrt{P}, x_{Ex} = \mp\sqrt{P} \\ 1 - \frac{1}{2n} & x_{Im} = 0, x_{Ex} = \pm\sqrt{P} \end{cases} \tag{3.10}$$

Similarly, $f(y|x_{Im}, x_{Ex})$ in (3.9) is the conditional probability density of y conditioned on explicit and implicit sequences which can be written as

$$f(y|x_{Im}, x_{Ex}) = \begin{cases} N(\pm\sqrt{P}, \sigma) & x_{Im} = \pm 2\sqrt{P}, x_{Ex} = \mp\sqrt{P} \\ N(x_{Ex}, \sigma) & x_{Im} = 0, \forall x_{Ex} \end{cases} \tag{3.11}$$

According to (3.7) through (3.11), I_{Ex} and I_{Im} can be evaluated numerically for given values of n and $Eb/N_0 = P/\sigma^2$. Fig. 3.10 shows I_{Ex}, I_{Im} and $I(X_{Ex}, X_{Im}; Y)$ variations of a TECC with E_b/N_0 over an AWGN channel for different values of n of a TECC scheme. For comparison, Fig. 3.10 also shows the capacity of the original BPSK C_{BPSK}, and the Shannon capacity $C_{Shannon}$, over an AWGN channel. As seen from Fig. 3.10, even though I_{Ex} is slightly lower than C_{BPSK}, $I(X_{Ex}, X_{Im}; Y)$ is ultimately higher than C_{BPSK} thereby demonstrating the throughput enhancement that can be achieved by a TECC. It is also seen from Fig. 3.10 that (a) I_{Ex} increases with n, (b) I_{Im} decreases with n, and (c) $C(\mathfrak{I})$ decreases with n as we expected.

3.6.2 Achievable rate region of a TECC

Following the capacity calculation of a multi-user scheme (Wachsmann et al., 1999), (Ahlswede and Han, 1983), any pair of (R_{Ex}, R_{Im}) which respectively are the rates of explicit and implicit sequences in a TECC, can be achieved if R_{Ex} and R_{Im} satisfy the following conditions: a) $R_{Ex} \leq I(Y; X_{Ex})$, b) $R_{Im} \leq I(Y; X_{Im}|X_{Ex})$, and c) $R_{Ex} + R_{Im} \leq I(Y; X_{Ex}, X_{Im})$. Fig. 3.11 illustrates the achievable rate region of a TECC obtained by the three aforementioned conditions. Point 1 highlighted in Fig. 3.11 , where $R_{Ex} = I(Y; X_{Ex})$ and $R_{Im} = (Y; X_{Im}|X_{Ex})$, corresponds to the case when X_{Ex} is optimally decoded as in the single-user case without any knowledge about X_{Im}. On the other hand, point 2 highlighted in Fig. 3.11 corresponds to interchanging X_{Ex} and X_{Im} demonstrating the case when decoding starts from the implicit sequence. The line between points 1 and 2, is derived from the time sharing solution and achievable rate region is the closure of the set of achievable rate pairs (R_{Ex}, R_{Im}) due to the convexity of rate region (Sason, 2004).

3.7 Conclusion

Throughput enhancing concatenated codes (TECCs) have been constructed to increase the throughput of a code C. In a TECC, the coded bit sequence of an explicit code C is altered according to the coded bits of an implicit code C'. TECCs that select one bit in every segment of n coded bits of C based on $n_s = log_2 n$ coded bits of C', and flip that selected bit before transmission have been constructed. The selection of the bit and flipping of that selected bit can be viewed as mapping n_s coded bits of C' onto a n-bit long error sequence that is added to the n bit long coded segment of C before transmission. Even though none of the coded bits of C' is transmitted explicitly over the channel, the information of coded bits of C' is transferred to the destination through the flipped bits. When C is a short code, TECCs can employ an interleaver to improve performance of the implicit sequence, however, when C is a long code, no interleaver is required. It has been shown that iterative decoding between code C and C' can resolve both the coded bits of the explicit code C and the implicit code C'. The simulated results demonstrate that when C is a turbo code used in the LTE standard, a turbo code C' can be chosen to increase the throughput of C by 25% to 37.5% without sacrificing the performance of C.

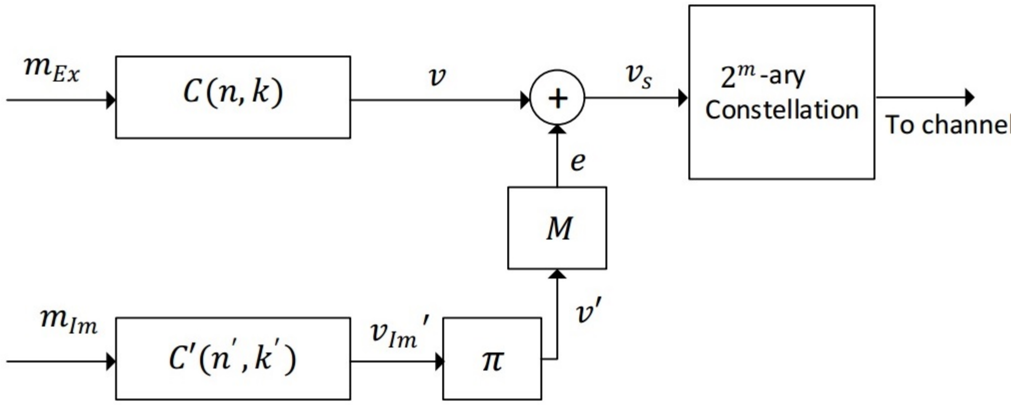

Figure 3.1: TECC encoder structure.

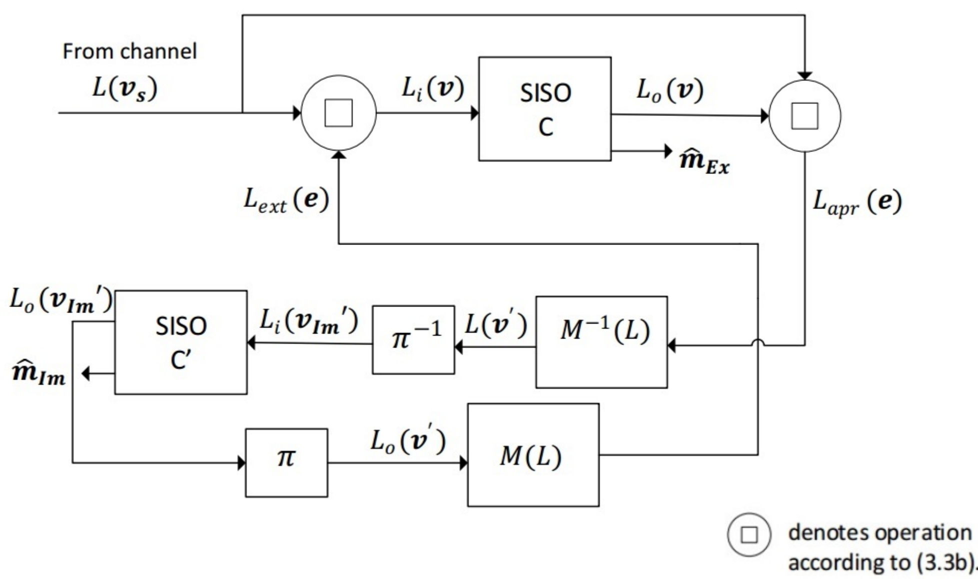

Figure 3.2: TECC decoder structure.

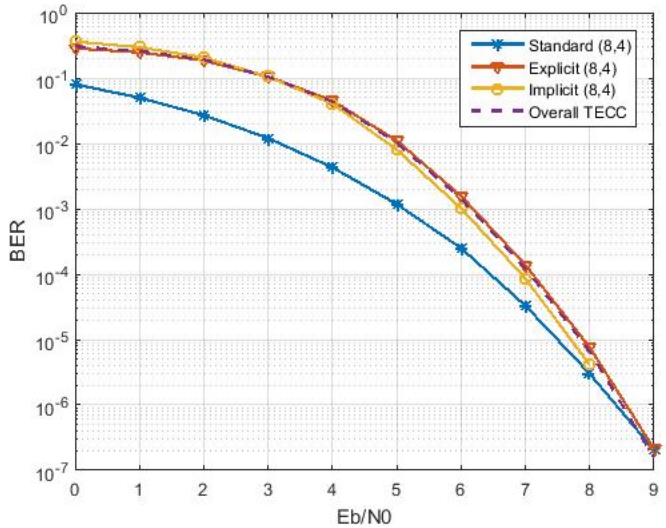

Figure 3.3: The performance of explicit sequence, implicit sequence and overall TECC, constructed using a (8,4) extended Hamming code as C and C'.

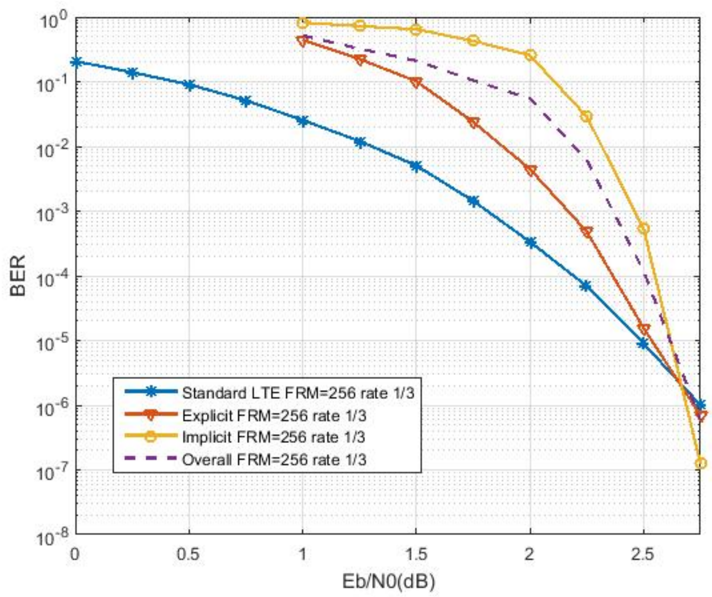

Figure 3.4: The performance of explicit sequence, implicit sequence and overall TECC, constructed using a rate 1/3 LTE turbo code as C and C' with frame size 256 bits.

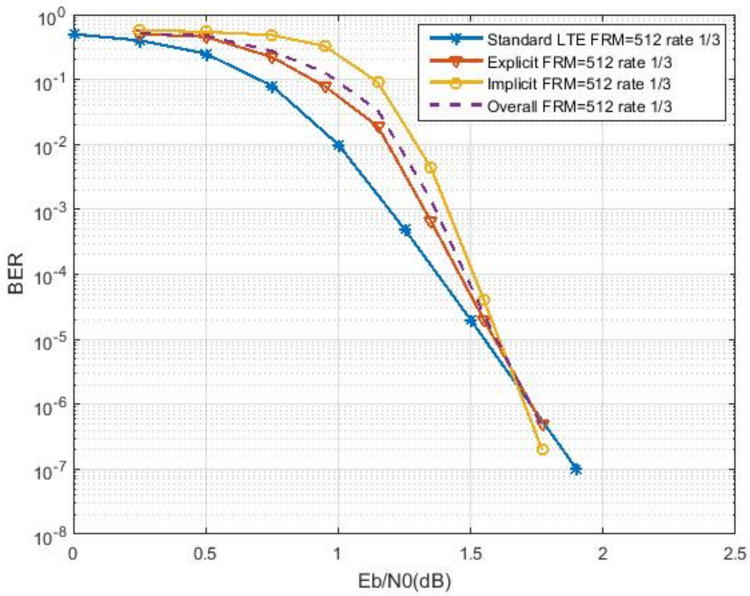

Figure 3.5: The performance of explicit sequence, implicit sequence and overall TECC, constructed using a rate 1/3 LTE turbo code as C and C' with frame size 512 bits.

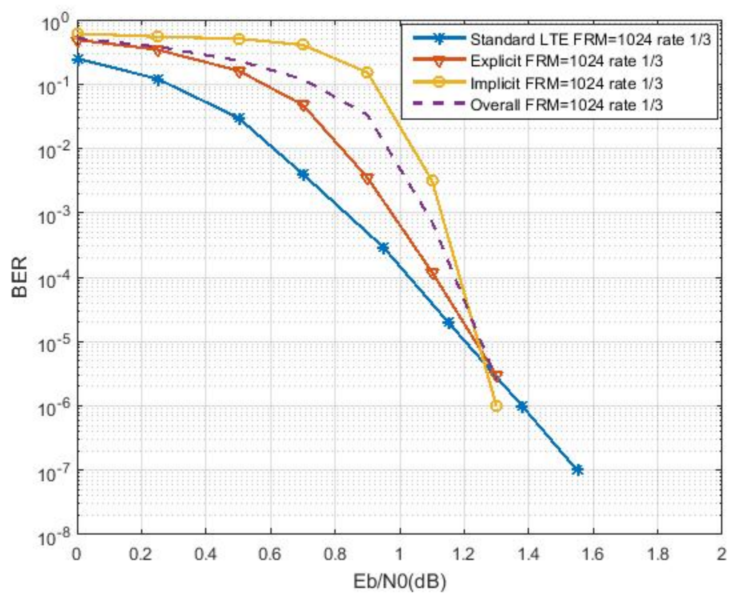

Figure 3.6: The performance of explicit sequence, implicit sequence and overall TECC, constructed using a rate 1/3 LTE turbo code as C and C' with frame size 1024 bits.

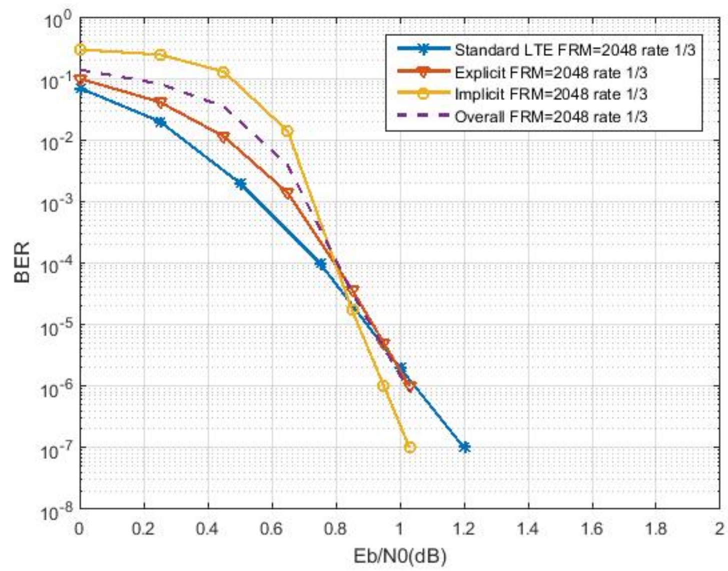

Figure 3.7: The performance of explicit sequence, implicit sequence and overall TECC, constructed using a rate 1/3 LTE turbo code as C and C' with frame size 2048 bits.

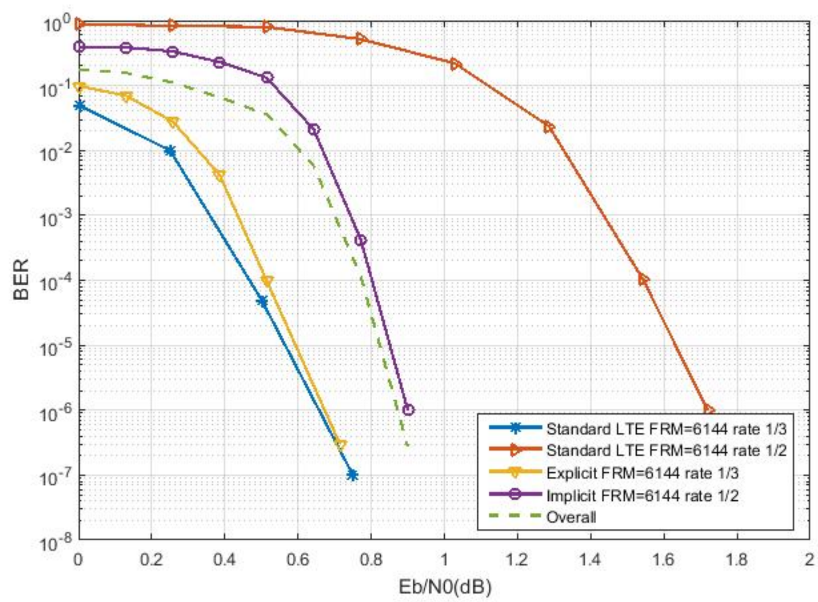

Figure 3.8: The performance of explicit sequence, implicit sequence and overall TECC, constructed using a rate 1/3 LTE turbo code as C and rate 1/2 LTE Turbo code as C' with frame size 6144 bits.

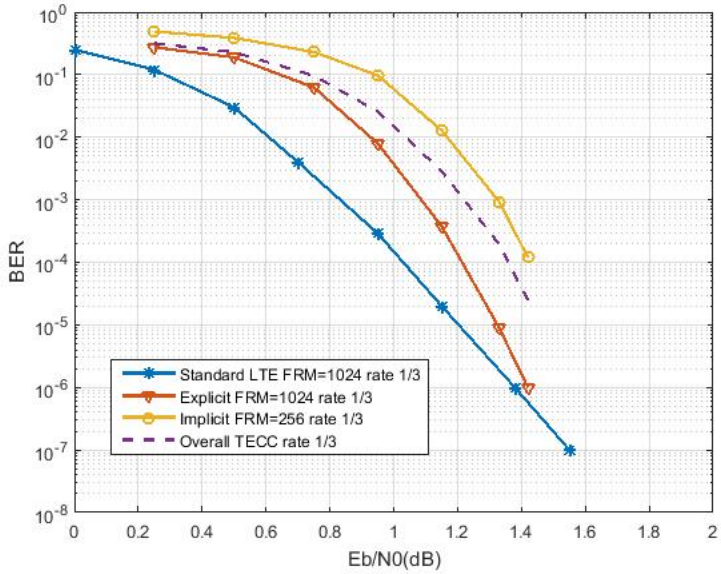

Figure 3.9: The performance of explicit sequence, implicit sequence and overall TECC, constructed using a rate 1/3 LTE turbo code as C with frame size 1024 bits and C' with frame size 256.

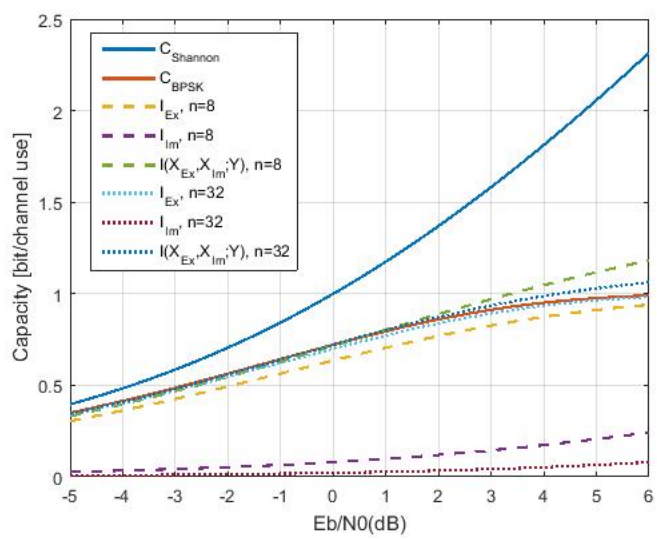

Figure 3.10: Information capacity of explicit sequence, implicit sequence and overall TECC for n=8 and n=32.

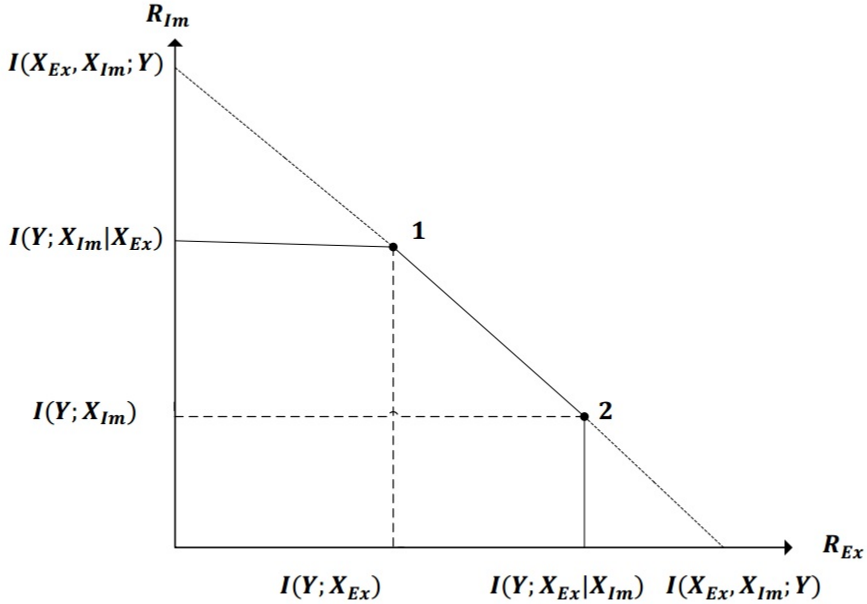

Figure 3.11: Illustration of achievable rate region of TECC schemes.

CHAPTER 4

THROUGHPUT ENHANCING CONCATENATED CODES WITH A SECOND UNCODED IMPLICIT STREAM

4.1 Introduction

In Chapter 3 it was demonstrated that TECC schemes can enhance the throughput by implicitly transmitting a separate data stream. In this chapter, the possibility of employing multiple implicit streams in a TECC scheme is studied. Specifically, two TECC configurations that can be used to employ more than one implicit steam are examined.

In particular, TECC schemes with two implicit streams, which are referred here as TECC-2 schemes, are constructed by extending TECC schemes. For comparison, TEEC schemes in Chapter 3 that employ a single implicit stream are referred to as TECC-1 schemes. It is shown here that TECC-2 schemes can significantly increase the throughput expansion that can be achieved by TECC-1 while maintaining the same performance. Interestingly, the second implicit stream of a TECC-2 scheme can be kept uncoded to maximize its impact on the throughput expansion and to minimize the increase in decoding complexity. In addition, the tradeoff between the low signal to noise ratio (SNR) performance and throughput expansion is discussed by adjusting different design parameters of TECC-2 schemes.

4.2 Review of TECCs

In Chapter 3 throughput enhancing concatenated codes (TECCs) have been introduced to increase the throughput of a coded stream. This has been done by (a) selecting one bit out of every n coded bits, v, of a code C (referred to as the explicit code) based on $\lfloor log_2 n \rfloor$ coded bits, v', of a second cod C' (referred to as the implicit code), and (b) flipping (inverting) the selected bit in (a) before transmission. It has been discussed that flipping the selected bit is equivalent to mapping v' on to an n-bit long weight one error sequence e and adding e to

v to form the transmitted sequence v_s as $v_s = v \oplus e$, where $\oplus$ denotes modulo-2 addition. It has been shown in Chapter 3 that both sequences v and v' can be extracted at the receiver using the received version of v_s and employing iterative decoding. As a result the sequence v' (implicit stream) can be recovered without actually transmitting any bits of it over the channel by use of extrinsic information provided by explicit streams. As a result, throughput of the code C can be increased by transmitting a second sequence implicitly. The numerical results presented here when both C and C' are LTE turbo codes demonstrate that the throughput of C can be increased by 25% to 37.5% without sacrificing performance at high SNR. Throughout this study, TECCs presented in Chapter 3 that employ a single implicit stream are referred to as TECC-1 schemes.

4.3 Direct Extension

Fig. 4.1 illustrates a structure of a TECC scheme that employs N implicit coded streams by directly extending TECC-1 technique. This TECC scheme with N implicit streams selects one coded bit out n'_i coded bits of coded interleaved sequence of the i^{th} implicit branch is selected based on $\lfloor log_2 n_i \rfloor$ of the coded interleaved of the $(i{+}1)^{th}$ implicit stream and flips that selected bit, for all $1 \leq i < N$. As with TECC-1 schemes, the interleavers in Fig. 4.1 can be eliminated when the implicit codes, C$_i$', $i = 1, 2, ..., N$, are all long codes such as turbo codes or LDPC codes. It can be seen from Fig. 4.1, such a TECC scheme with N implicit streams can transmit additional coded bits implicitly from all implicit streams thereby increasing the throughput expansion beyond that of TECC-1. However, the throughput increase of any i^{th} implicit stream decreases as i increases limiting the achievable throughput expansion. Since each implicit code C$_i$', $i = 1, ..., N$, needs to be decoded in each TECC iteration during decoding, the decoding complexity of the TECC configuration shown in Fig. 4.1 grows rapidly with i. Since the structure in Fig. 4.1 is not attractive in practice, it is not further investigated but is used only for comparison purpose in this study.

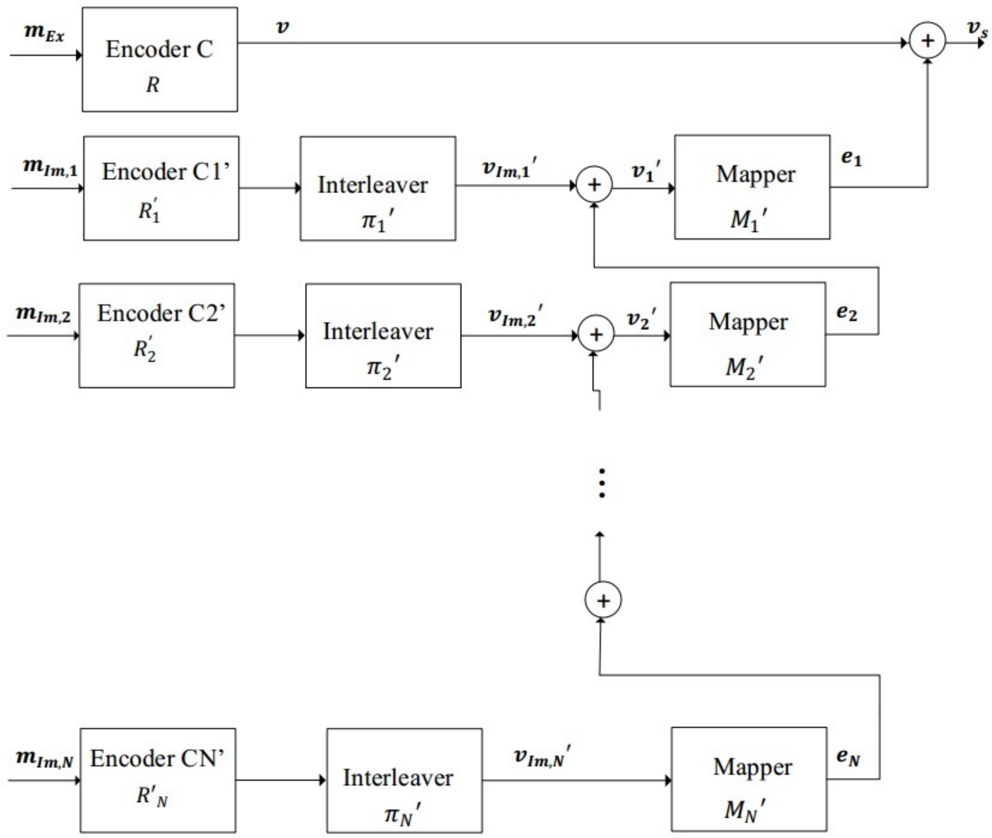

Figure 4.1: A structure of a TECC encoder that employs N implicit coded streams by directly extending TECC-1 technique.

4.4 TECC-2 Schemes

Instead of using the structure shown in Fig. 4.1, a much simpler TECC transmitter structure with only two implicit streams as shown in Fig. 4.2 is proposed and studied. The structure shown in Fig. 4.2 always use only two implicit streams, and it is referred to as TECC-2 in this study. Further, it differs from the structure in Fig. 4.1 due to the fact that its second implicit stream m_{Im2} is uncoded and hence it does not employ a code or an interleaver. During encoding of TECC-2, the implicit stream v'_{Im} in Fig. 4.2 is altered by (a) selecting one bit out of n' bits according to $n'_s = \lfloor log_2 n' \rfloor$ bits of the second implicit

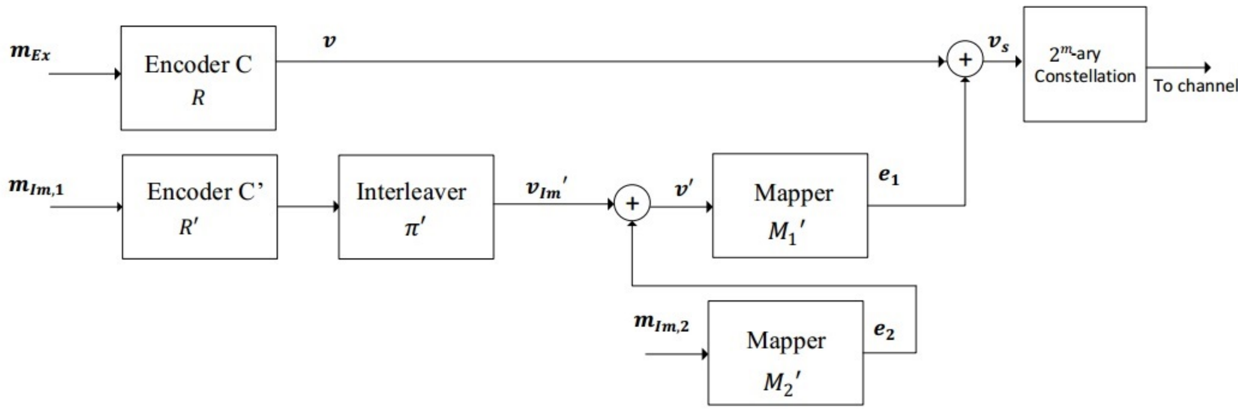

Figure 4.2: TECC-2 encoder structure.

stream m_{Im2} and (b) flipping that selected bit in (a) to form the stream v_1'. As illustrated in Fig. 4.2, this operation is equivalent to mapping every n_s' bit long block of m_{Im2}, $m_{Im2}(k) = (m_{Im2,1}(k), m_{Im2,2}(k), ..., m_{Im2,n_s'}(k))$, onto a n' bit long weight one error sequence $e_2(k) = (e_{2,1}(k), e_{2,2}(k), ..., e_{2,n'}(k))$ and adding that error sequence e_2 to the corresponding n'-bit long block of $v_I m'$, $v_{Im}'(k)$, to form a n'-bit long block of v_1', $v_1'(k)$, as $v_1'(k) = v_{Im}'(k) \oplus e_2(k)$, where $\oplus$ denotes modulo 2 addition. Even though any one to one mapping policy can be used in the mapper, one easy way to implement the mapper M_2' in Fig. 4.2 is to place the single non-zero digit of $e_2(k)$ at the position given by the $dec(m_{Im2}(k) + 1)$, where, $dec(x)$ denotes the decimal value of the binary sequence x. Similarly, one bit out of every block of n bits of the explicit stream v is selected based on n_s bits of v_1' and flipped to form the transmitted sequence v_s. Again this can be implemented by mapping every j^{th} block of n_s bits of v_1', $v_1'(j) = (v_{1,1}'(j), v_{1,2}'(j), ..., v_{1,n_s}'(j))$ onto an n-bit long weight one error sequence $e_1(j) = (e_{1,1}(j), e_{1,2}(j), ..., e_{1,n}(j))$ and adding it to the corresponding n-bit long block of v, $v(j)$, to form the corresponding block of the transmitted sequence $v_s(j)$ as $v_s(j) = v(j) \oplus e_1(j)$. Hence, in the transmitter m_{Im2} and v_{Im}' are independent, but the effect of m_{Im2} and v_{Im}' is felt by v_1' which is responsible for altering the transmitted sequence

v_s. Therefore, the information of the explicit sequence v, first implicit sequence v'_{Im} and second implicit sequence m_{Im2} are all contained in some form in the transmitted sequence v_s. As a result, iterative decoding of the received version of v_s can be used to recover all three sequences v, v'_{Im} and m_{Im2} simultaneously. Throughput expansion offered by the TECC-2 structure shown in Fig. 4.2 over original C decoding can be found by considering the additional bits transmitted by the two implicit streams in addition to the information carries by the explicit stream. Considering a n-bit block of v_s, the throughput expansion can be expressed as

$$\eta = \frac{log_2(n)(R' + \frac{log_2(n')}{n'})}{nR} \tag{4.1}$$

where R and R' are the rates of C and C' respectively. It is interesting to compare the throughput expansion of TECC structures shown in Figs. 4.1 and 4.2.

It is interesting to compare the throughput expansion of structures 4.1 and 4.2 in the special case when $n'_i = n'$ and $R'_i = R'$ where R'_i is the rate of the code used by the code on the implicit stream i for $i = 1, ..., N$. In order to achieve the highest throughput expansion the values of n'_i in the structure shown in Fig. 4.1 should be kept as small as possible, however, n'_i usually increases with i. Therefore, the throughput expansion of the direct extension η_{DE} in Fig. 4.1 when $n'_i = n'$ and $R'_i = R'$, $i = 1, .., N$, η_{DE} can be bounded by

$$\eta_{DE} \leq \frac{log_2 n}{nR} R'(1 + \frac{log_2 n'}{n'} + (\frac{log_2 n'^2}{n'}) + ... + (\frac{log_2 n'}{n'})^{N-1}) \tag{4.2}$$

Comparing 4.1 with 4.2, it can be easily shown that $\eta > \eta_{DE}$ when $R' < (n' - log_2 n')/n'$ even at the limit $N \to \infty$. Since the above condition for R' is satisfied for most practical values of n', for example in cases shown in the numerical results $n' = 64$, TEEC-2 structure shown in Fig. 4.2, which is much simpler than the direct structure shown in Fig. 4.1, offers a higher throughput expansion too.

4.4.1 TECC-2 Decoding

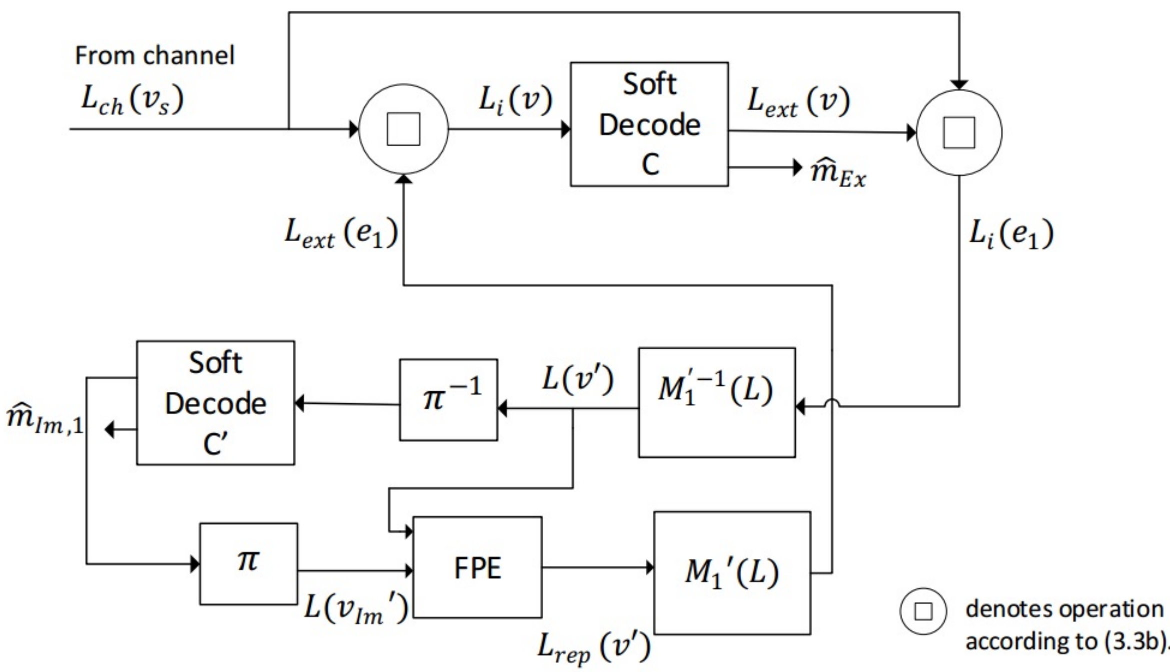

Figure 4.3: TECC-2 decoder structure.

Fig. 4.3 shows the decoder structure that employs iterative decoding to simultaneously decode the sequences v, v' and m_{Im2}. This decoder structure is only slightly more complex than the normal TECC-1 decoder for the decoding of a TECC scheme with a single implicit stream (see Fig. 3.2). The only addition from the Fig. 3.2 made in Fig. 4.3 is an operation highlighted in Fig. 4.3 that estimates the position of v'_{Im} that has been flipped according to m_{Im2}. This operation, which is referred to as the flipped position extraction (FPE), is performed on the basis of blocks of n' bits. It is done by comparing the extrinsic information at the input, $L(v')$, and the output, $L(v'_{Im})$, of the decoder of C' during iterations as illustrated in Fig. 4.3. Note that $L(v') = (L(v'_1), L(v'_2), ..., L(v'_n))$ carries information of v' where as $L(v'_{Im}) = (L(v'_{Im,1}), L(v'_{Im,2}), ..., L(v'_{Im,n'}))$ carries information of v'_{Im}. Since the decoder knows that one out of n' bits of v'_{Im} is flipped to form v', the position in which

$L(v')$ and $L(v'_{Im})$ has changed the sign and/or has the biggest difference is most likely to has been flipped before transmission. Since the TEEC iterations are likely to make $L(v')$ and $L(v'_{Im})$ mostly reliable, the second implicit stream can most likely be correctly extracted even when it is left uncoded. However, note that the first implicit stream cannot be kept uncoded because the input to the decoder of C is severely affected by channel noise. In this study, the following steps can be used in the FPE to estimate the flipped position in a soft manner and to update LLR values of v', $L_{upd}(v')$:

$$L_{upd}(v'_i) = L(v'_i) - sign(L(v'_i))w(i).(|L(v'_i) - L(v'_{Im,i})|), \qquad \forall i = 1 : n' \tag{4.3}$$

where $w(i)$ is weighting parameter for the i^{th} bit which is computed as follows:

$$w(i) = \frac{|L(v'_i) - L(v'_{Im,i})|}{\sum_{j=1}^{n'} |L(v'_j) - L(v'_{Im,j})|} \tag{4.4}$$

In order to simplify, the above estimation can be alternatively done in a hard sense, however, at the expense of some performance. In a hard estimation of FPE, the following steps are used to update $L(v')$:

$$L_{upd}(v'_{i*}) = -L(v'_{i*}) \quad where \quad i* = max_i |L(v'_i) - L(v'_{Im,i})| \tag{4.5}$$

In practice, soft estimation can be preferably used during the first N_1 TECC iterations and then hard estimation can be during the last N_2 iterations with a total of $N = (N_1 + N_2)$ iterations to reduce complexity. The values of N_1 and N_2 can be chosen depending on the application to maintain good performance. The remaining operations in the decoder are described in Chapter 3. Summarizing, each iteration requires soft decoding of C and C', passing of soft information through the mapper and the demapper which can be done similar to soft decoding of a linear block code, and calculating the soft information transfer through the modulo 2 addition operation. Specifically, the log-likelihood ratio (LLR) value of $z = x \oplus y$, $L(z)$ can be expressed in terms of the LLR values of x, $L(x)$, and y, $L(y)$, as in 3.3b.

Noticing that if $z = x \oplus y$, then $y = x \oplus z$, and $x = z \oplus y$, 3.3b can be used to calculate the soft information of any of the three variables x, y or z, when those of the other two are known.

4.5 Low SNR Performance and Throughput Expansion Tradeoff

As stated before TECC-1 schemes can increase the throughput significantly by transmitting a second coded data stream implicitly while transmitting a first coded data stream explicitly. Further, this increase in throughput is achieved without sacrificing performance at high SNR values. For example, Fig. 3.6 shows the BER variation of a TECC-1 when C and C' are both LTE turbo codes with rate 1/3 and frame size 1024, the TECC-1 scheme that increases the throughput by 25% achieves the same performance as standard LTE code (which is also referred to as original C signaling) for error rates below about 10^{-5}. One drawback of TECC-1 schemes is that they suffer in performance at lower SNR values.

Since there are applications that can afford to use slightly higher error rates, such as voice over IP or video stream (Chen et al., 2004), it would be desirable to find a way to maintain TECC performance similar to the standard original C decoding even at lower SNR values. Hence, in this section, methods that can be used to maintain performance of TECC schemes similar to that of the original C coding at lower SNR values are discussed. Specifically, the following three methods can effectively improve lower SNR performance of a TECC scheme compared with original C decoding, however, at the expense of some throughput expansion: (a) increasing n, thereby flipping one bit out of a higher number of bits n selected according to up to the $log_2 n$ bits of v in Fig. 3.1 , or (b) adjusting the frame sizes of component codes C and C', or (c) adjusting the rates of the component codes C and C'. It is noted that these three methods can be applied in both TECC-1 and TECC-2 schemes. Among the above three methods, method (a) is the easiest to implement as it only requires to change the value of n. Method (b) is suitable only in applications where packets with different frame sizes are used for transmission. Similarly, method (c) forces to use different code rates for the two codes which increases the complexity of the decoder. Hence, in this study, only method (a) is examined. Numerical results presented in 4.5 for selected TECC-2 schemes, show that by increasing n, TECC performance at low SNR can be improved however, by lowering the throughput expansion.

4.6 Numerical Results

In this section, bit error rate (BER) variation of TECC schemes described before are presented and compared when both C and C' are turbo codes adopted in the 4G long term generation (LTE) standard. Fig. 4.4 shows the BER variation of the explicit stream, first and second implicit streams, and the overall BER variations of a TECC-2 scheme when both C and C' are rate (1/3) LTE turbo codes (Martin and Taylor, 2001) when $n = 16, n' = 64$ and with frame size 2048. For comparison, the BER variation of a standard LTE turbo code of the same rate and frame size (which is referred to as the original C scheme in Chapter 3) and the BER variation of the corresponding TECC-1 scheme discussed in Chapter 3 are also plotted in Fig. 4.4. These BER variations have been obtained using $N_1 = 4$ and $N_2 = 2$ with a total of $N = 6$ iterations. It is seen that the proposed TECC-2 scheme performs similar to the original C decoding and the corresponding TECC-1 scheme. It is noted that the TECC-1 and TECC-2 schemes respectively achieve 25% and 32% throughput expansion over the original C scheme. In other words the TECC-2 scheme offers an additional 7% throughput expansion over the corresponding TECC-1 by employing a second implicit stream, while maintaining about the same decoding complexity of TECC-1.

In order to demonstrate the tradeoff between the low SNR performance and throughput, Fig. 4.5 shows the BER variation of a TECC-2 scheme using $N_1 = 4$ and $N_2 = 2$ when both

C and C' are LTE turbo codes with rate (1/3) LTE turbo codes and frame size 2048, when $n = 16, 64$, and 256 and $n' = 64$. It is easily seen from Fig. 4.5 that the TECC-2 performance at lower SNR improves as n increases, however, at the expense of of throughput expansion. Specifically, the throughput expansion of the TECC-2 scheme in Fig. 4.5 according to 4.1 is η=32%, 12% and 4% when $n = 16, 64$ and 256, respectively. Therefore, in practice, the value of n can be carefully chosen to maintain TECC performance similar to that of original C decoding at the desired BER value.

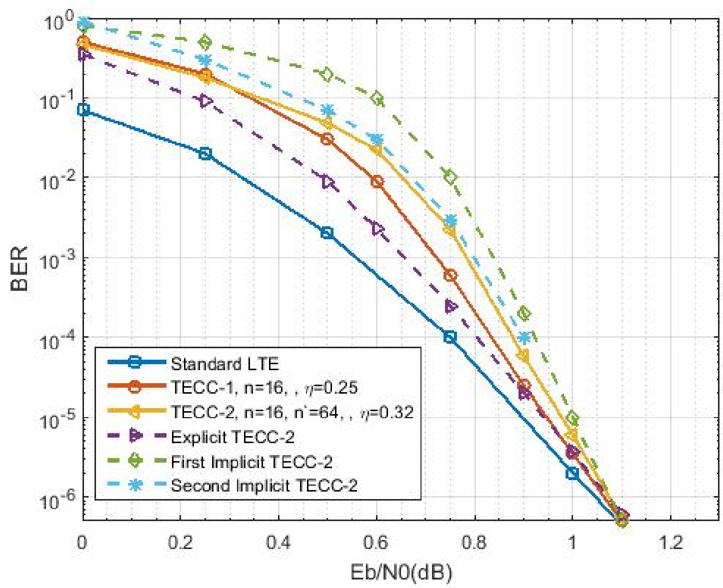

Figure 4.4: BER variations of TECC-2 when C and C' are LTE Turbo code with rate 1/3 and frame length=2048.

4.7 Conclusions

In this chapter, TECC schemes presented in Chapter 3 have been extended to include two implicit streams to form TECC-2 schemes. The second implicit stream flips one bit of every n' bits of the first implicit stream based on up to $log_2 n'$ bits of the second implicit stream, while one bit out of every n coded bits of the explicit code is flipped based on up to $log_2 n$ bits of the first implicit steam. Importantly, the second implicit stream can be left uncoded.

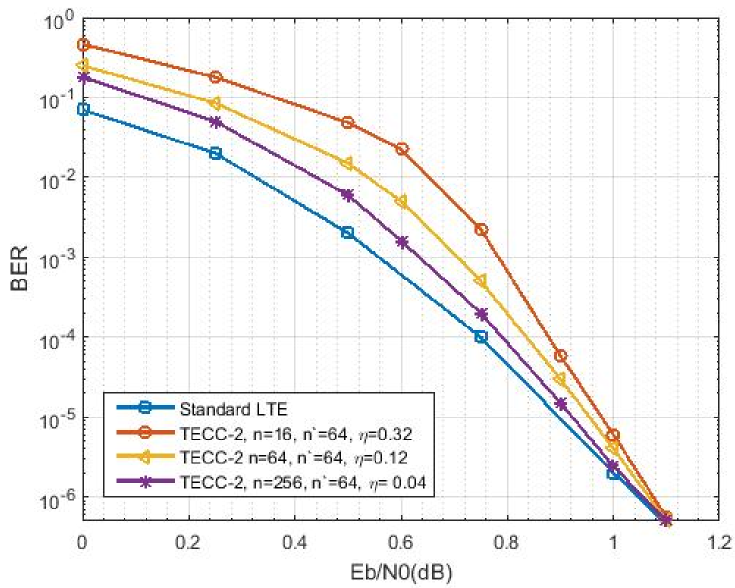

Figure 4.5: BER variations of TECC-2 for different values of n to improve the low SNR performance when C and C' are LTE Turbo code with rate 1/3 and frame length=2048.

It has been demonstrated here that TECC-2 schemes can increase the throughput enhancing capability of TECCs significantly with a minimal increase in complexity. It has also been demonstrated that low SNR performance of TECC schemes can be improved by adjusting the design parameters.

CHAPTER 5

CONCLUSION

A multi-constellation signaling (MCS) technique has been introduced to implicitly transmit coded bits in addition to explicitly transmitted bits over the channel thereby increasing the throughput. A MCS scheme selects one constellation among a bank of constellations during every interval based on a set of implicit bits. MCS schemes have been designed, analyzed and compared with standard signaling schemes that employ a single constellation. It has been demonstrated that a properly designed MCS scheme can double and triple the throughput over the corresponding traditional schemes that employ a single constellation without sacrificing performance. Compared with turbo coded signaling used in the LTE, MCS schemes that perform better with a lower decoding complexity have been presented.

Throughput enhancing concatenated codes (TECCs) have been constructed to increase the throughput of a code C. In a TECC, the coded bit sequence of an explicit code C is altered according to the coded bits of an implicit code C'. TECCs that select one bit in every segment of n coded bits of C based on $n_s \leq log_2 n$ coded bits of C', and flip that selected bit before transmission have been constructed. The selection of the bit and flipping of that selected bit can be viewed as mapping n_s coded bits of C' onto an n-bit long error sequence that is added to the n bit long coded segment of C before transmission. Even though none of the coded bits of C' is transmitted explicitly over the channel, the information of coded bits of C' is transferred to the destination through the flipped bits. When C is a short code, TECCs can employ an interleaver to improve performance of the TECC schemes, however, when C is a long code, no interleaver is required. It has been shown that iterative decoding between code C and C' can resolve both the coded bits of the explicit code C and the implicit code C'. The simulated results demonstrate that when C is a turbo code used in the LTE standard, a turbo code C' can be chosen to increase the throughput of C by 25% to 37.5% without sacrificing the performance of C.

An extension of TECC known as TECC-2 has been introduced by employing a second uncoded implicit stream. A TECC-2 schemes flips one bit out of every n' bits of the first implicit stream based on up to $log_2 n'$ bits of the second implicit stream, while one bit out of every n coded bits of the explicit code is flipped based on the $log_2 n$ bits of the first implicit steam. It has been demonstrated here that TECC-2 schemes can increase the throughput enhancing capability of TECCs significantly with a minimal increase in complexity. It has also been demonstrated that low SNR performance of TECC schemes can be improved by adjusting the TECC design parameters. The TECC that has been discussed so far, selects one bit out of n bits and flips it before transmission. Even though this is a simple way to get the influence of an implicit stream on to the transmitted stream, there can be better alternate ways to introduce that influence. Finding such methods to transmit bits implicitly is one potential future study that stems from this dissertation. In addition application of TECCs in different technologies such as optical communications and Biomedical applications are also valuable extension of this study.

REFERENCES

Agiwal, M., A. Roy, and N. Saxena (2016). Next generation 5g wireless networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials 18*(3), 1617–1655.

Ahlswede, R. and T. Han (1983). On source coding with side information via a multiple-access channel and related problems in multi-user information theory. *IEEE Transactions on Information Theory 29*(3), 396–412.

Aitsab, O. and R. Pyndiah (1997). Performance of concatenated reed-solomon/convolutional codes with iterative decoding. In *Global Telecommunications Conference, 1997. GLOBECOM'97., IEEE*, Volume 2, pp. 934–938. IEEE.

Bartelt, J., N. Vucic, D. Camps-Mur, E. Garcia-Villegas, I. Demirkol, A. Fehske, M. Grieger, A. Tzanakaki, J. Gutiérrez, E. Grass, et al. (2017). 5g transport network requirements for the next generation fronthaul interface. *EURASIP Journal on Wireless Communications and Networking 2017*(1), 89.

Benedetto, S., D. Divsalar, G. Montorsi, and F. Pollara (1996). A soft-input soft-output maximum a posteriori (map) module to decode parallel and serial concatenated codes. *TDA progress report 42*(127), 1–20.

Benedetto, S., D. Divsalar, G. Montorsi, and F. Pollara (1997). A soft-input soft-output app module for iterative decoding of concatenated codes. *IEEE communications letters 1*(1), 22–24.

Benedetto, S., D. Divsalar, G. Montorsi, and F. Pollara (1998). Serial concatenation of interleaved codes: Performance analysis, design, and iterative decoding. *IEEE Transactions on information Theory 44*(3), 909–926.

Benedetto, S. and G. Montorsi (1996). Design of parallel concatenated convolutional codes. *IEEE Transactions on Communications 44*(5), 591–600.

Berrou, C., A. Glavieux, and P. Thitimajshima (1993). Near shannon limit error-correcting coding and decoding: Turbo-codes. 1. In *Communications, 1993. ICC'93 Geneva. Technical Program, Conference Record, IEEE International Conference on*, Volume 2, pp. 1064–1070. IEEE.

Calderbank, A. R. (1989). Multilevel codes and multistage decoding. *IEEE Transactions on Communications 37*(3), 222–229.

Chen, Y., T. Farley, and N. Ye (2004). Qos requirements of network applications on the internet. *Information Knowledge Systems Management 4*(1), 55–76.

Chindapol, A. and J. A. Ritcey (2001). Design, analysis, and performance evaluation for bicm-id with square qam constellations in rayleigh fading channels. *IEEE Journal on Selected Areas in communications 19*(5), 944–957.

Choi, L.-U. and R. D. Murch (2004). A transmit preprocessing technique for multiuser mimo systems using a decomposition approach. *IEEE Transactions on Wireless Communications 3*(1), 20–24.

Chung, S.-Y., G. D. Forney, T. J. Richardson, and R. Urbanke (2001). On the design of low-density parity-check codes within 0.0045 db of the shannon limit. *IEEE Communications letters 5*(2), 58–60.

Cisco (2017a, 05). Cisco visual networking index: Forecast and methodology, 20162021.

Cisco (2017b, 05). Cisco visual networking index: Global mobile data traffic forecast update, 20162021 white paper.

Costa, M. H. and A. El Gamal (1987). The capacity region of the discrete memoryless interference channel with strong interference. *IEEE Transactions on Information Theory 33*(5), 710–711.

Csiszar, I. and J. Körner (2011). *Information theory: coding theorems for discrete memoryless systems*. Cambridge University Press.

Di Renzo, M., H. Haas, and P. M. Grant (2011). Spatial modulation for multiple-antenna wireless systems: A survey. *IEEE Communications Magazine 49*(12).

Expert (2017, 10). Processing power compared, experts-exchange.

Feng, D., C. Jiang, G. Lim, L. J. Cimini, G. Feng, and G. Y. Li (2013). A survey of energy-efficient wireless communications. *IEEE Communications Surveys & Tutorials 15*(1), 167–178.

Fonseka, J. P. and E. Rezaei (2017). Implicit transmission of coded information using multi-constellation signaling. *IEEE Communications Letters 21*(4), 718–721.

Gallager, R. G. (1968). *Information theory and reliable communication*, Volume 2. Springer.

Hassan, A. E. S., M. Dessouky, A. A. Elazm, and M. Shokair (2012). Evaluation of complexity versus performance for turbo code and ldpc under different code rates. In *Proc. SPACOMM 2012: The fourth international conference on advances in satellite and space communication*, pp. 93–98.

Hu, Y., J. P. Fonseka, Y. Bo, E. M. Dowling, and M. Torlak (2017). Constrained turbo product and block-convolutional codes in wireless applications. *IEEE Transactions on Vehicular Technology 66*(5), 4491–4495.

Hu, Y., J. P. Fonseka, and E. M. Dowling (2017). Constrained interleaved coded modulation. *IEEE Transactions on Vehicular Technology 66*(4), 3501–3506.

Huber, J. and U. Wachsmann (1994). Capacities of equivalent channels in multilevel coding schemes. *Electronics Letters 30*(7), 557–558.

IBM (2016, 01). Processing power compared, experts-exchange ibm ilog cplex solver(version 12) [software].

Imai, H. and S. Hirakawa (1977). A new multilevel coding method using error-correcting codes. *IEEE Transactions on Information Theory 23*(3), 371–377.

Jeganathan, J., A. Ghrayeb, L. Szczecinski, and A. Ceron (2009). Space shift keying modulation for mimo channels. *IEEE Transactions on Wireless Communications 8*(7), 3692–3703.

Kofman, Y., E. Zehavi, and S. Shamai (1994). Performance analysis of a multilevel coded modulation system. *IEEE Transactions on Communications 42*(234), 299–312.

Li, J., K. R. Narayanan, and C. N. Georghiades (2004). Product accumulate codes: a class of codes with near-capacity performance and low decoding complexity. *IEEE Transactions on Information Theory 50*(1), 31–46.

Lin, S. and D. J. Costello (2004). *Error control coding*, Volume 2. Prentice Hall Englewood Cliffs.

Martin, P. A. and D. P. Taylor (2001). On multilevel codes and iterative multistage decoding. *IEEE Transactions on Communications 49*(11), 1916–1925.

McEliece, R. J., D. J. C. MacKay, and J.-F. Cheng (1998). Turbo decoding as an instance of pearl's" belief propagation" algorithm. *IEEE Journal on selected areas in communications 16*(2), 140–152.

Mesleh, R., H. Haas, C. W. Ahn, and S. Yun (2006, Oct). Spatial modulation - a new low complexity spectral efficiency enhancing technique. In *2006 First International Conference on Communications and Networking in China*, pp. 1–5.

Mesleh, R. Y., H. Haas, S. Sinanovic, C. W. Ahn, and S. Yun (2008). Spatial modulation. *IEEE Transactions on Vehicular Technology 57*(4), 2228–2241.

Meyr, H., M. Moeneclaey, and S. A. Fechtel (1998). *Digital communication receivers: synchronization, channel estimation, and signal processing*. Wiley Online Library.

Mietzner, J., R. Schober, L. Lampe, W. H. Gerstacker, and P. A. Hoeher (2009). Multiple-antenna techniques for wireless communications-a comprehensive literature survey. *IEEE communications surveys & tutorials 11*(2).

Nimbalker, A., Y. Blankenship, B. Classon, and T. K. Blankenship (2008). Arp and qpp interleavers for lte turbo coding. In *Wireless communications and networking conference, 2008. WCNC 2008. IEEE*, pp. 1032–1037. IEEE.

Perişoară, L. A. and R. Stoian (2008). The decision reliability of map, log-map, max-log-map and sova algorithms for turbo codes. *International Journal of Communications 2*(1), 65–74.

Proakis, J. G. and M. Salehi (2008). *Digital communications*. McGraw-Hill.

Pyndiah, R. M. (1998). Near-optimum decoding of product codes: Block turbo codes. *IEEE Transactions on communications 46*(8), 1003–1010.

Sadjadpour, H. R., N. J. Sloane, M. Salehi, and G. Nebe (2001). Interleaver design for turbo codes. *IEEE Journal on Selected Areas in Communications 19*(5), 831–837.

Sason, I. (2004). On achievable rate regions for the gaussian interference channel. *IEEE transactions on information theory 50*(6), 1345–1356.

Shannon, C. E. and W. Weaver (1998). *The mathematical theory of communication*. University of Illinois press.

Sun, J. and O. Y. Takeshita (2005). Interleavers for turbo codes using permutation polynomials over integer rings. *IEEE Transactions on Information Theory 51*(1), 101–119.

Tragos, E. Z., S. Zeadally, A. G. Fragkiadakis, and V. A. Siris (2013). Spectrum assignment in cognitive radio networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials 15*(3), 1108–1135.

Tran, N. H. and H. H. Nguyen (2006, May). Design and performance of bicm-id systems with hypercube constellations. *IEEE Transactions on Wireless Communications 5*(5), 1169–1179.

Ungerboeck, G. (1982). Channel coding with multilevel/phase signals. *IEEE transactions on Information Theory 28*(1), 55–67.

Wachsmann, U., R. F. Fischer, and J. B. Huber (1999). Multilevel codes: Theoretical concepts and practical design rules. *IEEE Transactions on Information Theory 45*(5), 1361–1391.

Younis, A., N. Serafimovski, R. Mesleh, and H. Haas (2010). Generalised spatial modulation. In *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on*, pp. 1498–1502. IEEE.

BIOGRAPHICAL SKETCH

Elahe Rezaei was born in Iran. She received a Bachelor of Science degree in Electrical and Electronics Engineering and a Master of Science with a major in Communications from Sharif University of Technology in May 2012 and 2014, respectively. Currently, she is a PhD candidate and works as a graduate teaching/research assistant in the Erik Jonsson School of Engineering and Computer Science at The University of Texas at Dallas. She is interested in the classical problems of communication and information theory and also intersection of the new-emerging applications of communication and Artificial Intelligence, particularly the theoretical frameworks in multi-agent learning and decision making problems.

CURRICULUM VITAE

Elahe Rezaei

August 15, 2018

Contact Information:

Department of Electrical and Computer Engineering
The University of Texas at Dallas
800 W. Campbell Rd.
Richardson, TX 75080-3021, U.S.A.

Fax: (972) 883-2349
Email: elahe.rezaei@utdallas.edu

Educational History:

B.S., Electrical and Electronics Engineering, Sharif University of Technology, 2012
M.S., Electrical and Electronics Engineering, Sharif University of Technology, 2014
Ph.D., Electrical and Computer Engineering, University of Texas at Dallas, 2018

A Class Of Implicit Transmission Techniques For Throughput Enhancement
Ph.D. Dissertation
Electrical and Computer Engineering, University of Texas at Dallas
Advisors: Professor Jonh. P. Fonseka

Artificial Intelligence For Real-time Applications in Information and Communication Technologies.
Master Thesis
Electrical Engineering, Sharif University of Technology
Advisor: Professor Babak Khalaj

Employment History:

Graduate Research Assistant, The University of Texas at Dallas, September 2015 – present
Research Scientist, Koc University, January 2015 – August 2015
Senior Technical Advisor, Monenco Iran Consulting Engineers, August 2014 –January 2015

Professional Recognitions and Honors:

Admitted for graduate program, University of Texas at Dallas, 2015
Awarded Tubitak Scholarship, Koc University, 2014
Dean's Honorary Award, Sharif University of Technology, 2014
21^{st} ranked among 270,000 participants in the nationwide entrance examination of Iranian universities, 2007
Awarded in National Mathematics Competition, 2006

Professional Memberships:

Institute of Electrical and Electronics Engineers (IEEE), 2010–present
Member of Iranian National Elite Foundation (INEF), 2007
Honorary member of Sharif University of Technology Association (SUTA), 2017