EFFECTS OF CHANNEL ESTIMATION ON TURBO EQUALIZER PERFORMANCE

AHMED SAMI ALI AL-ISAWI

SEPTEMBER 2014

EFFECTS OF CHANNEL ESTIMATION ON TURBO EQUALIZER PERFORMANCE

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES OF
ÇANKAYA UNIVERSITY

BY
AHMED SAMI ALI AL-ISAWI

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF
ELECTRONIC AND COMMUNICATION ENGINEERING

SEPTEMBER 2014

Title of the Thesis : **Effects of Channel Estimation on Turbo Equalizer Performance.**

Submitted by **Ahmed Sami Ali AL-ISAWI**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University.

Prof. Dr. Taner ALTUNOK
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Celal Zaim ÇİL
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Orhan GAZİ
Supervisor

Examination Date: 22.09.2014

Examining Committee Members

Assoc. Prof. Dr. Fahd JARAD	(UTAA)
Assoc. Prof. Dr. Orhan GAZİ	(Çankaya Univ.)
Assist. Prof. Dr. Göker ŞENER	(Çankaya Univ.)

STATEMENT OF NON-PLAGIARISM PAGE

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name: Ahmed Sami Ali AL-ISAWI

Signature :

Date : 22.09.2014

ABSTRACT

EFFECTS OF CHANNEL ESTIMATION ON TURBO EQUALIZER PERFORMANCE

AL-ISAWI, Ahmed Sami Ali
M.Sc., Department of Electronic and Communication Engineering
Supervisor: Assoc. Prof. Dr. Orhan GAZİ

September 2014, 39 pages

Equalizer design is an important concept for the wireless communication systems suffering from multipath effects such as ISI. Turbo equalizers are one of the most powerful equalizers introduced several decades ago. For the equalizers to be efficient in removing ISI reliable channel estimation is essential. In this thesis the performance of turbo equalizers for blind and non-blind channel estimators is inspected. For blind channel estimation per-survivor processing with least mean square method is employed. For non-blind channel estimation training sequences are used for channel estimation. An improved channel estimation approach for non-blind technique was suggested and via simulation results, it was seen that the proposed method shows better performance in turbo equalizer than the classical blind and non-blind channel estimation methods do.

Keywords: Convolution Code, Turbo Code, Turbo Equalizer, Channel Estimation, Per-Survivor Processing.

ÖZ

KANAL TAHMİNİNİN TURBO DENKLEŞTİRİCİLERİN PERFORMANSLARI ÜZERİNDEKİ ETKİLERİ

AL-ISAWI, Ahmed Sami Ali
Yüksek Lisans, Elektronik ve Haberleşme Mühendisliği Anabilim Dalı
Tez Yöneticisi: Doç. Dr. Orhan GAZİ

Eylül 2014, 39 sayfa

Kablosuz iletişim sistemlerinde denkleştirici tasarımı çoklu yol iletişiminin sebep olduğu etkileri yok etmek için gerekli olan önemli bir konsepttir. Turbo denkleştirici bir kaç on yıl önce tasarlanan en etkili denkleştiricilerden birisidir. Denkleştiricilerin etkili performans sergileyebilmeleri için kanal katsayılarının düzgün olarak tahmin edilmesi zaruridir. Kanal katsayıları genel olarak iki yöntemle hesaplanırlar, bunlar, bilgisiz olarak kanal katsayılarını tahmin etme ve bilgili olarak kanal katsayılarını hesaplama yöntemleridir. Bilgisiz olarak kanal katsayılarını hesaplamak için kazanan yol yöntemi ile en az ortalama kare yöntemleri birlikte kullanılmıştır. Bilgili olarak kanal katsayılarını hesaplamak için eğitim dizileri kullanılmıştır. Bilgisiz olarak kanal katsayılarını hesaplamak için iyileştirilmiş bilgisiz kanal hesaplama yöntemi önerilmiştir ve bilgisayar benzetim sonuçlarından görüleceği üzere önerilen yöntemle hesaplanan kanal katsayıları diğer yöntemle hesaplananlara göre turbo denkleştiricide daha iyi performans göstermiştir.

Anahtar Kelimeler: Konvolusyonel Kod, Turbo Kod, Turbo Denkleştirici, Kanal Tahmini, Kazanan Patika

ACKNOWLEDGEMENTS

I would like to express my special appreciation and thanks to my advisor Assoc. Prof. Dr. Orhan GAZİ, you are a great mentor for me, you encourage me in master degree, and guide me how to make this research thesis. I will do my best to follow your advices in my career and my life. A special thanks to my family. I cannot express how grateful I am to my father, mother for all of the help and sacrifices that you have made on my behalf. Your prayer for me was what sustained me thus far. No words can express my feeling to my friends; you support me in my studying and in my life.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURES

LIST OF ABBREVIATIONS

BER	Bit Error Rate
FEC	Forward Error Correction
FIR	Finite Impulse Response
APP	A Posteriori Probabilities
SISO	Soft Input Soft Output
BCJR	Bahl, Cocke, Jelinek, and Raviv
SOVA	Soft Output Viterbi Algorithm
DFSE	Decision Feedback Sequence Estimation
MAP	Maximum A Posteriori
MLSE	Maximum Likelihood Sequence Estimation
LSE	Least Square Estimation
MMSE	Minimum Mean-Square Error
OFDM	Orthogonal Frequency-Division Multiplexing
CDMA	Code Division Multiple Access
DSL	Digital Subscriber Line
RCS	Recursive Systematic Convolutional
NSC	Non Systematic Convolutional
LLR	Log-likelihood Ratio
SNR	Signal to Noise Ratio
AWGN	Additive White Gaussian Noise
ISI	Inter Symbol Interference
PBSK	Binary Phase Shift Keying
PPSP	Principle of Per-Survivor Processing
PSP	Per-survivor Processing
RSSE	Reduced State Sequence Estimation
MSE	Mean Square Error

CIR	Channel Impulse Response
GSM	Global System for Mobile Communications
ML	Maximum Likelihood
MVU	Minimum Variance Unbiased
LMS	Least Mean Square

CHAPTER 1

INTRODUCTION

1.1. Channel Coding

Signal strength, duration, and speed are the major bases of any communication system. In this challenging communication era, all three of these resources operate in contrast to each other. The major aim is always to achieve high-quality bulk data transmission at high speed. Maintaining quality of data transmission is usually related to bit error probability likelihood at the receiver side. Mentioned below in Eq. (1.1), the Shannon-Hartley law presents the capacity of a communication channel with two values. First, it demonstrates how speed (B) and signal strength (S/N) can be utilized in an ideal system, and second, it provides a theoretical restriction for the rate of transmission of reliable, error-free data (R max) from a transmitter of a given strength, over the channel with a given speed that operates in an existing noisy atmosphere.

$$R_{\max} = B \log_2\left(1+\frac{S}{N}\right) bit\ /\ s \tag{1.1}$$

According to the Shannon-Hartley law, an appropriate coding scheme can be used to achieve theoretical limits. For example, a design engineer may intend to estimate the required data rate speed that is limited to an existing channel and with limited strength of a particular application. Error control coding is an appropriate approach for modifying the quality of data for a specific S/N. The other factor that motivates us to use this approach is that it helps to reduce the needed S/N for a specified bit. In the radio communication scenario, this reduction process of S/N can be useful in decreasing the needed transmission strength and hardware cost of acquiring a small antenna.

Therefore, with the use of error control coding, the design interchange gains a bearable error rate by also considering the speed and system difficulty. The main design purpose of channel coding (error control coding) is to raise the defense power of a digital communication system against channel noise. The received error symbols can be detected and corrected by using error control coding.

1.2. Turbo Codes

The first classic paper by Claude Shannon (published in 1949) supports a mathematical solution for the noisy communication channel. Claude Shannon measured the highest theoretical volume in his analysis of communication channel; he indicates and restricts the error correction channel coding that permits it to gain the highest capacity. The intervening years have noticed many valuable channel codes that approach the Shannon limit. However, all challengers have required great blocks of measurements to achieve a closest rate to the boundary. The resulting difficulty, price, and signal inactivity of these codes made them unfeasible within 3 to 5 dB of the edge.They were, however, capable of delivering beneficial coding gain at maximum values of Eb/No and bit error rate. A new session of density codes, known as turbo codes (invented by Berrou, Glavieux and Thitimajshima in 1993 whose presentation in terms of Bit Error Rate (BER)), were quiet similar to the Shannon limit. After a time, some other writers discovered a method to code in their supportive assessment, which provided the possibility to operate within 0.7dB of the Shannon limit. The possible action [1] presented by turbo codes has motivated both industrial and academic researchers. It is noteworthy that in the last seven years that there is a resultant explosion of study into all aspects of turbo codes. This study inspects the ideologies of turbo coding and decoding methods, focusing primarily (mainly) on coding and decoding procedures. It then observes the act of turbo codes in both simple and multilevel groups and accomplishes this by assessing the state of turbo code research and growth.

1.3. Iterative Soft Input Soft Output Algorithms

In 1993, Berrou et al. [3, 4] proposed Turbo codes such that iterative processes play a significant role because of the incredible enhancement of presentation. Further ahead, Benedetto et al. [2] revealed that consecutively joined coded systems could also approach the Shannon limit by repeating the process. However, these repetitive process approaches are extensively utilized in developing communication system receivers. Let us suppose that the idea of this iterative process leads to equalization [9, 23, 24], bit-added code alteration [5, 18, 22], an organization [2], and channel assessment [8, 16, 17, 20, 2, 24] with forward error correction (FEC) codes for numerous transmission systems. All the driven outcomes infer that the important act of enhancement could be acquired with repetition on both fading and AWGN channels. However, the difficulty in this repetition processing approach is considerable and it must be replaced by using less complex schemes. Douillard et al. [9] were the first to suggest an iterative method for turbo equalization, which is relied on a trellis arrangement. This method encouraged signal processing or communication groups with an emphasis on serial joint coded schemes and an external channel encoder as well as an internal modulator. The drawback of the trellis-based turbo equalizer is the quantity of trellis conditions that grows / increases /expands exponentially with the signal alphabet size and the channel desire reply. After that, Tüchler et al. [23, 24] offered turbo equalization structures founded on a finite impulse response (FIR) filter. In reality, an exceptional bit error, rate (BER) presentation can be gained with both the FIR-based and trellis-based equalization methods. The fundamental of the repetitive process is the exchange of lenient information, which is produced by SISO modules [1, 11, 14]. Given the consistent a priori prospect as inputs, the SISO modules assess an a posteriori probability (APP) by applying the well-known Bahl, Cocke, Jelinek and Raviv (BCJR) algorithm [3]. Subsequently, the BCJR process is mathematically rigorous due to the forward and backward recursions, and the possibility of a soft-output Viterbi algorithm (SOVA) [13] being applied to a SISO decoder. The compact search BCJR procedures, also known as T- and M-approaches, were suggested by Franz and Anderson [12].

1.4. Channel Estimation

The time deviation of a radio channel reduces the system performance, and suffers a complicated error floor because of the erroneous channel estimation. Therefore, the channel approximation in intelligible communication receivers plays a vital role to detect communicated signals. Channel estimation for intelligible demodulation can be classified into nonlinear and linear methods. The standard nonlinear estimator comprises the decision feedback sequence estimation (DFSE) [17, 21], MAP [8, 16], and MLSE [7, 19, 25] methods. In contrast, LSE and MMSE based channel estimation schemes are considered as linear approaches [6, 10, 15].

1.5. Selective Fading Channel

Selective fading, also known as frequency selective fading, is a wireless broadcast variance produced by the incomplete autonomous withdrawal of a wireless signal. When the signal reaches the receiver side via two dissimilar pathways, one of the paths either expands or shortens. This classically occurs initially at sunset or dawn, as several layers in the ionosphere change, separate and recombine. The two paths can be both a sky wave or a ground wave. Selective fading establishes itself as a sluggish, recurring disruption, the termination impact, or "null", is genuine at one specific rate of recurrence that changes without interruption, clearing [3] through the received voice. As the carrier rate of recurrence of a signal is broad, the magnitude of the alteration will differ immensely. The consistent speed assesses the split-up in frequency after which two signals will practice irrelevant fading. In flat fading, the rational speed of the channel is greater than the speed of the signal. Consequently, all frequency modules of the signal will go through a similar scale of fading. In frequency-selective fading, the rational speed of the channel is minor in comparison to the signal speed. Various diverse frequency mechanisms of the signal consequently practice uncorrelated fading. Since these frequency components of the signal are self-sufficient, it is extremely doubtful whether all portions of the signal will be concurrently affected by a deep fade. Definite variation arrangements such as orthogonal frequency-division multiplexing (OFDM) and code partition multiple accesses (CDMA) are suitable for retaining frequency variety so as to provide forcefulness to fading. OFDM splits the wideband signal into several gently curbed narrowband subcarriers, each bare to flat fading somewhat more so than frequency selective fading. This is possible to overcome by means of error encoding, easy equalization and adaptive bit loading. Inter symbol intrusion avoids this by introducing guard interval symbols. CDMA uses a RAKE receiver address for all echoes separately. A channel frequency selective fading disperses to the signal power link up with each symbol distributed in time. This is caused by the origin of natural gas symbols, which interfere with each of these channels. Equalizers are often used contiguously in time to compensate for interference. of the effects between the echo may be subject to Doppler, a varying channel model. The effect can be neutralized at the time of applying a diversity system such as OFDM, which are the two distinct receiver antennas that are a wavelength apart, or the receiver specifically for the two antennas. The receiver continuously compares the signals with the two antennas, and defines the best signal.

1.6. Equalization

The process of maintaining stability between frequency components within an electronic signal is known as equalization. The most famous use of equalization is voice recording and imitation. Numerous other applications in electronics and telecommunications do a similar job. The equipment used to accomplish equalization is called an equalizer. These devices boost or reduce the energy of particular frequency bands. In the telecommunications industry, equalizers are utilized to condense the frequency reply—for example, a telephone line—flat from end to end. As soon as a channel has been "equalized", the frequency area characteristics of the signal at the input are exclusively regenerated at the output. DSL lines within an electronic signal as well as television cables also use equalizers to organize data signals for transmission. . In the field of audio electronics, the term "equalization" has come to include the adjustment of frequency responses for practical or aesthetic reasons, often resulting in a net response that is not truly equalized. The term EQ explicitly states this variant of the term. [1] Stereos naturally have adaptive equalizers, which decreases the bass or threefold frequency. Broadcast and recording studios use refined equalizers which are more capable of more detailed adjustments, such as eliminating undesirable sounds or making particular devices or voices more prominent. Equalizers are essential to the effective operation of electronic systems such as analog broadcast television. The actual waveform in this application is the transmitted signal that must preserve not only its regular content. Equalizing filters should be capable of capping any delay response and phase delay between dissimilar frequency components.

1.7. Equalizer

The equalizer is an ideal tool to remove any distortion that occurs in transmitted signals through a channel. It helps to reverse any distortion during transmission and its main objective is to decrease the intervention of inter-symbols in order to permit the retrieval of transmitted symbols. This may be achieved with either a simplest lined filter or a more complicated process The types of equalizers that are mentioned in further sections are mostly utilized for similar purposes in wireless communication.

1.8. Thesis Outline

This thesis is organized as follows: A brief introduction of channel coding, turbo code, channel estimation, selective fading channel, equalizer, and equalization.

In Chapter 2 convolutional code, turbo code and interleavers are presented.

Chapter 3 discusses equalization techniques and channel estimations.

Chapter 4 presents the implementation, simulation, and results for the performance of turbo equalizers.

Finally, the conclusion and future work are presented in Chapter 5.

CHAPTER 2

CONVOLUTIONAL CODES AND TURBO CODES

2.1. Convolutional Codes

Error-correcting codes are used to improve the links of wireless digital communication. Convolutional codes are a type of error-correcting code that enhances the quality of wireless digital links such as radio and satellite. In 1955 [30, 29], a book by P. Elias introduced the concept of "coding for Noisy Channels". Block code is another type of error-correcting code. However, it differs from convolution code such that it provides an output at any given time unit by using memory encoders. With the dependency on "k" input, the output also depends on "m" previous input blocks. Convolutional code (n.m.k) is implemented with a k-input, a linear n-output, a sequential circuit with input memory m. In order to achieve lower probabilities of error, the memory order m should be large, while n and k are small integers where k>n. By accepting its importance as a redundant component, memory plays an essential role in correcting the capability of code. In general, R=k/n where the convolutional encoder input is in the form of a sequence of binary n-tuples as:

$$u = \cdots u_0, u_1, u_2, u_3, \cdots \tag{2.1}$$

Where $u_t = (u_t^1 ,. . . u_t^k)$ Sequence of binary–tuples is output

$$c = \cdots c_0, c_1, c_2, c_3, \cdots \tag{2.2}$$

There is a special case when k = l. where the information sequence is processing constantly and not divided into blocks.

where $c_t = (c_t^1, \ldots c_t^k)$ The sequence has to start at a finite (positive or negative) time; however, the end time is not known. The relation between information and code sequence is:

$$c = uG \tag{2.3}$$

Where

$$G = \begin{pmatrix} G_0 & G_1 & \cdots & G_m & & & \\ & G_0 & G_1 & \cdots & G_m & & \\ & & G_0 & G_1 & \cdots & G_m & \\ & & & G_0 & G_1 & \cdots & G_m \end{pmatrix} \tag{2.4}$$

The matrix is designed for semi-infinite generator and entries for sub and binary matrices are presented as:

$$G_i = \begin{pmatrix} g_{1.i}^{(1)} & g_{1.i}^{(2)} & \cdots & g_{1.i}^{(n)} \\ g_{2.i}^{(1)} & g_{2.i}^{(2)} & \cdots & g_{2.i}^{(n)} \\ \vdots & \vdots & \ddots & \vdots \\ g_{k.i}^{(1)} & g_{k.i}^{(2)} & \cdots & g_{k.i}^{(n)} \end{pmatrix} \tag{2.5}$$

The binary field IF2 is the arithmetic component of Eq. (2.3) while the areas left blank are unspecified and are filled with zeros in the generator matrix G.

The right hand side of Eq. (2.3) represents the discrete-time between u and g that is equal to (G0, G1,…,Gm) and termed convolutional codes. This presentation of sequences is suitable in case of convolution.

The delay operator D is widely using in information and coding theory, thus, the code and information sequence will be presented as:

$u(D)=\cdots+u_0+u_1D+u_2D^2+\cdots$

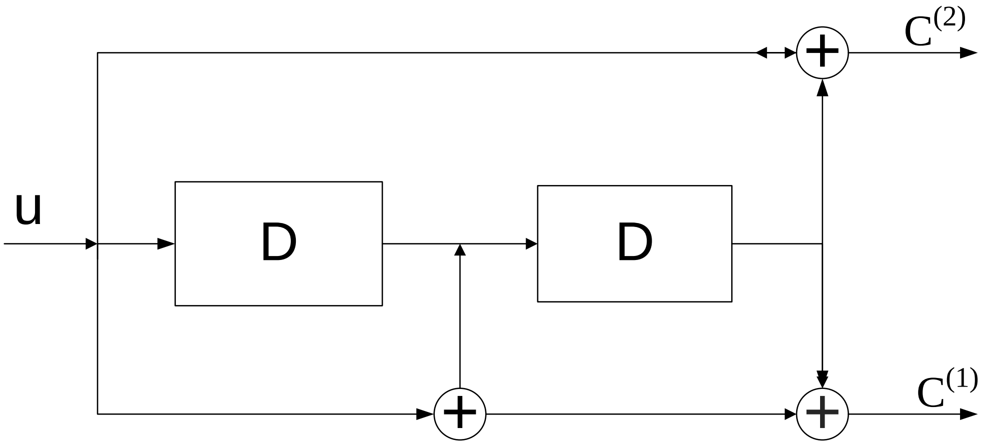

Figure 1: An encoder for the generator polynomial in Eq. 2.5.

And

$c(D)=\cdots+c_0+c_1D+c_2D^2+\cdots$

This equation is relating them in a way as

$$c(D)=u(D)G(D) \tag{2.6}$$

Where

$$G(D)=\begin{pmatrix} g_1^{(1)}(D) & g_1^{(2)}(D) & \cdots & g_1^{(n)}(D) \\ g_2^{(1)}(D) & g_2^{(2)}(D) & \cdots & g_2^{(n)}(D) \\ \vdots & \vdots & \ddots & \vdots \\ g_k^{(1)}(D) & g_k^{(2)}(D) & \cdots & g_k^{(n)}(D) \end{pmatrix}$$

is the generator matrix. It is termed as generator polynomials and it is presented as

$$g_i^{(o)}(D)=g_{i,0}^{(o)}+g_{i,1}^{(o)}D+g_{i,2}^{(o)}D^2+\cdots g_{i,m}^{(o)}D^m \tag{2.7}$$

$$for \quad i=1,2,\cdots,k \ and \ o=1,2,\cdots,n.$$

Example 2.1: Lets consider the polynomial generator matrix

$$G=(1+D+D^2,1+D^2) \tag{2.8}$$

Fig.1 depicts the encoder for generator matrix. The systematic codes have receive recognition among the subclass of convolutional codes. The systematic code functions such that the first k output sequences appear as replicas of the input sequences, i.e.,

$$c^i=u^i, i=1,2,\cdots k. \tag{2.9}$$

Definition 2.1: The Generator matrix of rate R=k/n convolutional code is known as systematic, while all the information sequences u(D) should remain unchanged in the code sequences c(D).

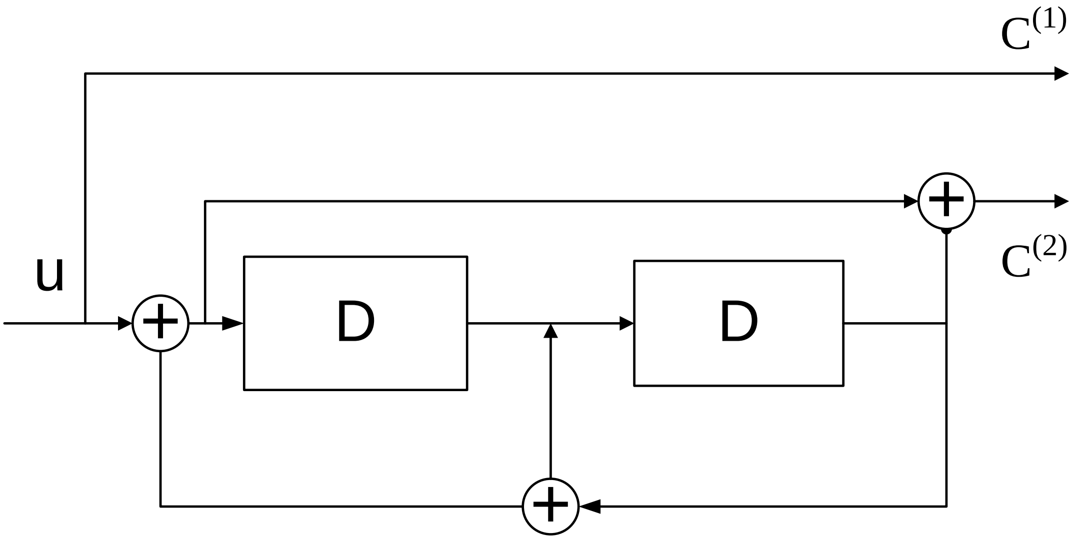

Figure 2: RCS(1,5/7) convolutional encoder.

The recursive systematic convolutional (RSC) is considered to be an important class among systematic convolutional codes. RSC plays an important role in parallel concatenation with the significance in convolutional-coupled codes.

For instance, a binary code R=1/2 RSC is obtained from the non-systematic convolutional code (NSC) by using a feedback loop where one of the two outputs () is equal to the input data bit (u_k).

Example 2.2: by following the convolutional code used in the previous example, the presentation of an equivalent recursive systematic generator mix will be

$$G_{sys} = \frac{1}{1+D+D^2} G = (1, \frac{1+D^2}{1+D+D^2}) \tag{2.10}$$

Fig.2 presents the encoder for the above generator matrix

2.1.1. State Diagram

A convolutional coder has a finite number *m* of memory and consequently a finite number 2^m of memory states σ.The coder can be realized as an automated finite state. The result of the coder at time t depends on the memory state σ_t and on the information block u_t. At time t+1 , the coder is in state σ_{t+1}. Each state modification $\sigma \rightarrow \sigma_{t+1}$ is associated with a certain input sequence and an output sequence. The graph with all possible states can be presented as the next example.

Example 2.3: The RSC (1,5/7) convolutional coder with memory $m = 2$ elements of example 2.2 is presented in the Fig. 2. The coder has $2^m = 4$_ memory states $\sigma \in \{(00),(01),(10),(11)\}$. For example, the state transition $(10) \rightarrow (01)$ which is labeled with $0/10$, means that the corresponding information block is $u_t = (0)$ and the output code symbol is $c_t = (10)$.

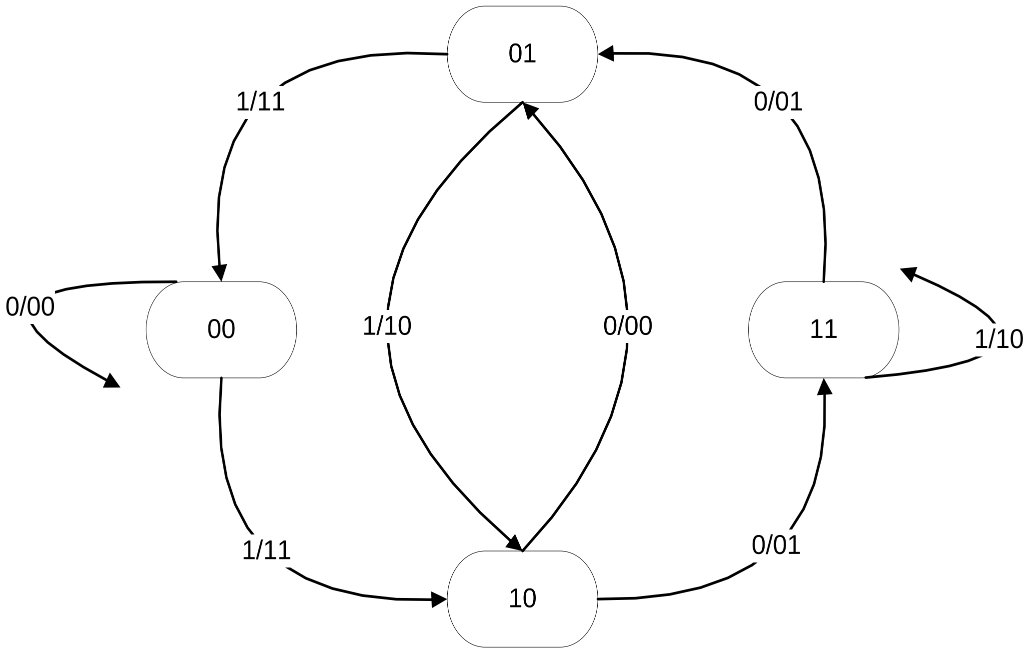

Figure 3: State diagram of a RSC(1,5/7) convolutional encoder.

2.1.2. Trellis

A convolutional code can be defined as a set of all possible code sequences that are encoded by a sequential circuit, namely the convolutional encoder. A trellis is a combination of branches and nodes where the nodes represent the encoder state while the state transitions are represented by a branch. The branches are labeled by the corresponding input/output block. The nodes are arranged horizontally in rows and the time step is arranged in columns. The path in the trellis defines the sequences of branches and nodes as well as any valid code words.

The illustration of the RCS convolutional code (1, 5/7) continues from example 2.3 and is presented in Fig.4 as a trellis. In Fig.4, we can observe that each branch is labelled with the respective output symbol. An input of one is represented by a solid line, while the dashed lines represent an input of zero. The output symbols are labeled to represent code words. All potential paths in the trellis are presented by convolutional code C. Moreover, the trellis relies on the encoder.

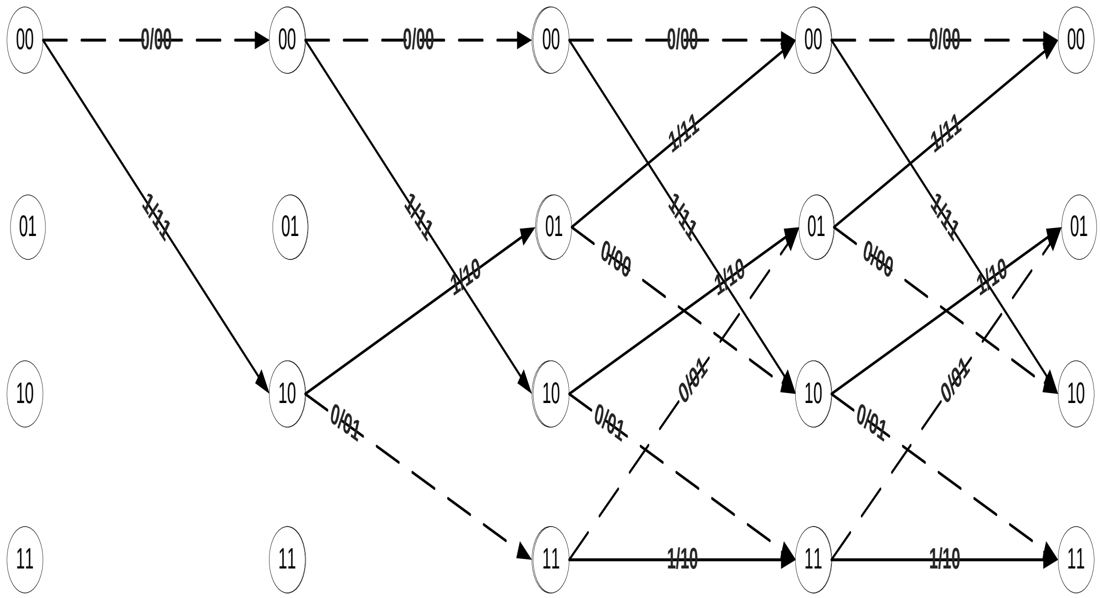

Figure 4: Trellis diagram of a RSC(1,5/7) convolutional encoder.

2.2. Turbo-Codes

There are several ways to code messages. Most simply add parity into the original code words. These code words are strengthened by the addition of a parity bit against every bit error. For example, code word 1010 may be coded in the following simple way:

1. The first and second bits are paired. The third and fourth bits will be added together.

2. Because of this pairing code word, 101011 will be attained. In case any bit encounters a bit error, the receiver will receive the incorrect result 101111. Thus, the receiver can easily identify an invalid result.

3. To correct the error in code, the receiver side has to verify the transmission. However, this process takes time. Usually this bad transmission occurs over long distances, such as deep-space communications.

2.3. Turbo-Codes Encoding

Turbo-codes can be coded in different ways, but they can be summarized in Fig. 5. Here, the scheme starts with the information, u, or the message that is to be sent. u is sent through encoder 1 and u is also interleaved, which often is denoted by π, and the result, u', is sent through encoder 2. Encoder 1 and encoder 2 are normally identical. The reason their output differs is totally dependent on the interleaving. The information, u or v0, is produced together with the result from encoder 1, v1, and the result from encoder 2, v2, concatenated.

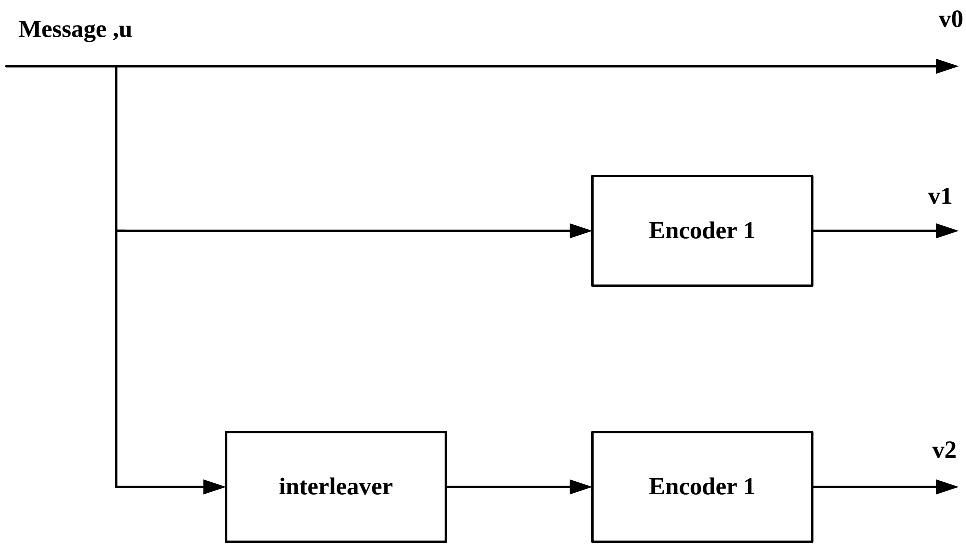

Figure 5: Encoder scheme of turbo code

2.4. BCJR or MAP Algorithm

In 1974, Bahl, Cocke, Jelinek and Raviv presented the BCJR algorithm [26]. In comparison to the Viterbi algorithm, it is considered more complicated. In Turbo codes, the BCJR algorithm is known as "Maximum a Posteriori Probability" (MAP) and has key roles due to its ability to produce an "A Posteriori Probability" (APP) for every decoded bit. The Viterbi algorithm differs from the BCJR algorithm such that it finds the most probable sequence from sent messages while the BCJR algorithm focuses on the most probable information bit within sent encoded sequences. In the early days of turbo codes [28] in 1993, a modified decoding algorithm known as Bahl algorithm was presented. This algorithm was very similar to the modified BCJR algorithm while the original version of BCJR decoder works to receive a randomly transmitted data sequence with a priori log-likelihood ratios of information bits

$$L_a(u_1), l = 0,1,\cdots,h-1$$

In this equation, h represents the block length of sequence u. For example, Fig.4 explains a transition probability $P(r_l \backslash v_l)$, in which v_l denotes the transmitted symbol and r_l is the received symbol. By using this data, the algorithm calculates the a posteriori LLRs (APP LLRs) for each information bit

$$L(u_1 \equiv \ln\left[\frac{P(u_1=+1|r)}{P(u_1=-1|r)}\right]. \qquad (2.11)$$

The decoder output is specified by hard decisions of the resulting estimated values (ˆu)

where

$$u_1 = \begin{cases} +1\, if \;\; L(u_\iota) > 0 \\ -1\, if \;\; L(u_\iota) > 0 \end{cases} \quad , \iota = 0,1,\cdots,h-1 \qquad (2.12)$$

Equation (2.11) can be reformulated after several steps [5,6] to

$$L(u_\iota) = \ln\left\{\frac{\sum_{(s',s)\in S_\iota^+} p(s_\iota = s', s_{\iota+1} = s, r)}{\sum_{(s',s)\in S_\iota^-} p(s_\iota = s', s_{\iota+1} = s, r)}\right\}, \qquad (2.13)$$

where p(.) is the probability density function and s is some state in the trellis. The set S_ι^+ indicates the set of all state pairs $s_\iota = s', s_{\iota+1} = s$ in the trellis, that corresponds to the input bit $u_\iota = +1$ at time ι, and the set S_i will be all the pairs analogous with $u_\iota = -1$. Further the probability density function can be evaluated recursively by

$$\begin{aligned} P(s',s,r) &= P(s',s,r_{t<\iota},r_\iota,r_{t>\iota}), \\ P(s',s,r) &= P(s',s,r_{t<\iota},r_\iota,r_{t>\iota}), \end{aligned} \qquad (2.14)$$

where r_ι represents the portion of the received sequence at time ι and $r_{t<\iota}$ and $r_{t>\iota}$ represents the portion received, before and after time ι, respectively. Then after using Bayes' rule given in [29]

$$\begin{aligned} P(s',s,r) &= p(r_{t>\iota}\,|\,s',s,r_{t<\iota},r_\iota)\,p(s',s,r_{t<\iota},r_\iota) \\ &= p(r_{t>\iota}\,|\,s',s,r_{t<\iota},r_\iota)\,p(s,r_\iota\,|\,s',r_{t<\iota})\,p(s',r_{t<\iota}) \\ &= p(r_{t>\iota}\,|\,s)\,p(s,r_\iota\,|\,s')\,p(s',r_{t<\iota}). \end{aligned} \qquad (2.15)$$

The last equality is splitting into three parts, as follows

$$\alpha_\iota(s') \equiv p(s', r_{t<\iota}) \qquad (2.16)$$

$$\gamma_\iota(s',s) \equiv p(s, r_\iota\,|\,s') \qquad (2.17)$$

$$\beta_{\iota+1}(s) \equiv p(r_{t>\iota}\,|\,s). \qquad (2.18)$$

Equation (2.15) can then be rewritten as

$$p(s',s,r) = \beta_{\iota+1}(s)\gamma_{\iota}(s',s)\alpha_{\iota}(s'). \tag{2.19}$$

The forward metric $\alpha_{\iota+1}(s)$ can be derived from Eq. 2.16, and this forward recursion is defined as

$$\alpha_{\iota+1}(s) = \sum \gamma_{\iota}(s',s)\alpha_{\iota}(s'), \tag{2.20}$$

Where σ_t is the set of all states at time ι, and the backward metric for $\beta_{\iota}(s')$ it can be defined in a way

$$\beta_{\iota}(s') = \sum_{s \in \sigma_{\iota+1}} \gamma_{\iota}(s',s)\beta_{\iota+1}(s). \tag{2.21}$$

The branch metric $\gamma_{\iota}(s',s)$ can be defined as

$$\gamma_{\iota}(s',s) = p(u_{\iota})p(r_{\iota} \mid \upsilon_{\iota}). \tag{2.22}$$

After modification it can be written as

$$\gamma_{\iota}(s',s) = p(u_{\iota})e^{-E_s/N_0 \|r_{\iota}-\upsilon_{\iota}\|^2}, \tag{2.23}$$

Where $\| r_{\iota} - \upsilon_{\iota} \|^2$ is the Euclidean distance between the received r_{ι} while the transmitted υ_{ι} E_S / N_0 is the Signal to Noise Ratio (SNR). Original $\alpha_0(s)$ and $\beta_k(s)$, where K = h+m6 and considered as the length of the input sequence u, can be defined as

$$\alpha_0(s) = \begin{cases} 1, s = 0 \\ 0, s \neq 0 \end{cases}, \tag{2.24}$$

And

$$\beta_k(s) = \begin{cases} 1, s = 0 \\ 0, s \neq 0 \end{cases}, \tag{2.25}$$

2.5. Log-MAP Algorithm

The Logarithmic Maximum A-posteriori Probability (Log-MAP) is applicable in systematic encoders. It is a modified form of the BCJR algorithm. Primarily, it is estimated that the received systematic data are the real value. This is the modified version of Max-Log-MAP introduced as the Log-MAP algorithm [30]. The above-defined Eq. (2.13) is used to calculate the posterior LLR in the MAP algorithm where the probabilities are replaced by Eq. (2.19) together with Eq. (2.20), (2.21) and (2.22). These equations can be simplified by application of an approximation after transfer into the log arithmetic domain

$$\ln\left(\sum_i e^{x_i}\right) = \max_i(x_i) \tag{2.26}$$

$\max_i(x_i)$ is the maximum value of x_i ,and $A_l(s), B_l(s)$ and $\Gamma_l(s', s)$ are, after using the definitions given in Eq. (2.20), (2.21) and (2.22), defined as

$$A_{l+1}(s) \triangleq \ln(\alpha_{l+1}(s)) \approx \max_{s'}(A_l(s') + \Gamma_l(s', s)) \tag{2.27}$$

$$B_l(s') \triangleq \ln(\beta_l(s')) \approx \max_{s}(B_{l+1}(s) + \Gamma_l(s', s)) \tag{2.28}$$

And

$$\Gamma_l(s', s) \triangleq \ln(\gamma_l(s', s) = \frac{u_l L_a(u_l)}{2} + \frac{L_c r_l . \upsilon_l}{2} \tag{2.29}$$

where La is the a priori log-likelihood values of the sent information bits. vl is the sent bit, rl is the received bit and Lc is the channel reliability factor. The computation of these equations can be seen in [33, 31, 32, 30].

Since the approximation in Eq. (2.26) preduces an inferior soft output compared to the MAP algorithm, the Jacobian logarithm used in the Log-MAP algorithm to fix this problem [33, 31].

$$\ln(e^{\delta_1} + e^{\delta_2}) = \max(\delta_1, \delta_2) + \ln(1 + e^{-|\delta_2 - \delta_1|}) \tag{2.30}$$

When the Eq. (2.27) and (2.28) use Eq. (2.29) instead of (2.26) the algorithm will be termed as a Log-MAP.

2.6. Interleaving

nterleavers have long been used in communication systems. The classical use of interleavers is to randomize the location of errors, allow for the use of random-error-correcting codes within channels by burst error patterns. Generally in wireless transmission, “bursty” channels also comprise fading channels. Furthermore, interleaving is used to concatenate the coding scheme where the result of inner-stage decoder shows burst error patterns (similar to the Viterbi decoder). In this scenario, the significance of the parameter of the interleaver is its ability to increase error bursts such that they find isolated errors at the outer-stage decoder. Naturally, the most favorable interleaver can achieve this with minimum memory [35]. In this case, the required parameter of the interleaver depends on any applied inner and outer codes and decoders Turbo coding; however, also introduced a further dimension to what is required from the interleaverIt involves the effects of the iterative algorithm and the passing of intrinsic information between successive decoder stages. This way, the interleaver tends to respond in order to reduce the dependency between the parity bits corresponding to the original and the interleaved data frames. In the beginner's introduction of turbo codes, Berrou et al explained the parameters that are important to achieve good a interleaver. In particular, an increase in block size improves the performance, while the interleaver has to randomize the input sequence in order to escape the low weight pattern mapping and reduce the effective free distance of the turbo codes.

Similarly to Turbo codes, interleavers prevent the formation of low-weight code words. An interleaver π is a permutation $i \to \pi(i)$ that maps a data sequence of k input symbols $(u_1, u_2, \cdots u_k)$ onto the same sequence in a new order. If the input sequence is $u = (u_1, u_2, \cdots u_k)$, then the permuted data sequence is uM_p, where M_p is an interleaving matrix with a single 1 in each row and column with all other entries being (). Every interleaver has a subsequent deinterleaver π^{-1} that acts on the interleaved data sequence and restores it to the original form. The deinterleaving matrix is simply the reverse of the interleaving matrix M_p^T.

2.6.1. Random interleaver

A random interleaver is basically a random variation π. With regard to the deinterleave arrangement, the randomness should be deterministic, such that the random table can be generated at the interleaver and the deinterleaver.

2.6.2. S-random interleaver

For the huge block size interleavers, most random interleavers need to function well. Conversely, small interleavers, which degrade the performance of the code, are essentially random interleavers. A short block length interleaver selection of semi-random interleavers can have a significant impact on the performance of the codes. The interleaver is based on sampling without replacement of whole numbers, which is less than the length of the interleaver limitations. These types of interleavers were discussed in [34].

Constraint 1: The $i-th$ randomly selected integer $\pi(i)$ must be rejected if there exists j such that:

$0 < i - j \leq S_1, and\ |\pi(i) - \pi(j)| \leq S_2.$

This constraint ensured that if two symbols i, j are within distance S_1 in the arrangement without interweaving, they cannot be planned to distance less than S_2 in the interleaved arrangement. We can also add other restraints to recover again the presentation of the interleaver. A delay of this technique reflects multiple error activities, in the command without interleaving. For example, the Fig. 5 portrays two error events that interchange module symbols. To avoid this disorder two more boundaries are defined as T_1 and T_2 and accomplish on the structure of the s-interleaver an additional restraint. Again casually select without extra integer from $1 \rightarrow k$ and if the $i-th$ selection $\pi(i)$ satisfies the restraint 1 described before, check if the following condition are also satisfied.

Constraint 2: The $i-th$ randomly selected integer $\pi(i)$ must be rejected if there exist any triple $j, t, l, j, t, l < i$ for which the conditions:

$$0 < i - j \leq T_1, \quad |\pi(i) - \pi(t)| \leq T_2$$

$$0 < |t - l| \leq T_1, \quad |\pi(j) - \pi(l)| \leq T_2$$

are fulfilled.

This constraint guarantees that the two relatively close component symbols i and j , do not have $\pi(t)$ near $\pi(i)$ and $\pi(l)$ near $\pi(j)$ in the interleaved sequence, with t and l near each other . An interleaver according to constraint 1, converges in a reasonable time if $S_1 S_2 \leq k/2$.

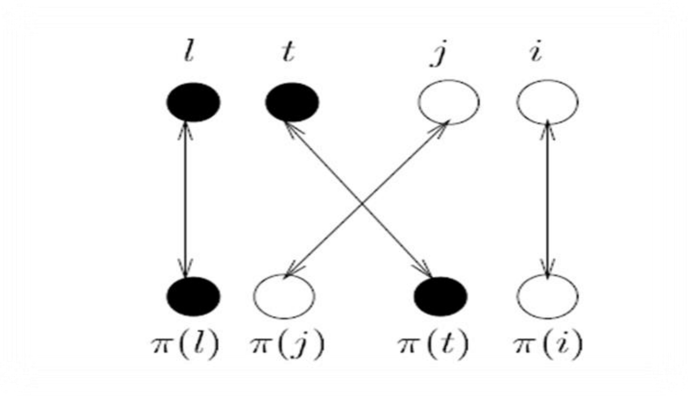

Figure 6: Error event

CHAPTER 3

EQUALIZATION AND CHANNEL ESTIMATION TECHNIQUES

3.1. Equalization Techniques for Frequency-Selective Channels

All real channels are able to diffuse time to some extent. In radio links, this diffusion can be seen in multi-path propagation, whereas in telephone channels, imperfect transferred characteristics cause time dispersion in a transmission system. In data transmission, this distortion causes successive transmitted symbols to be smeared in time and thus to overlap to the point that they may be no longer distinguishable as distinct pulses at the receiving terminal. This phenomenon is known as inter-symbol interference (ISI). In the perspective of frequency-domain, the channel transfer function exhibits frequency-dependent attenuations and delays over the transmission bandwidth; hence the frequency-selective channels having alternative designations. The impairments as a result of ISI become greater as successive symbols are spaced closer together in time in order to increase the data rate while bandwidth requirements are more stringent (thereby requiring transmit pulses with larger temporal support by virtue of the time frequency duality). ISI can cause profound impacts on pathological channels to discourage reliable communication in data systems even in the absence of noise. Nowadays, it is common to encounter highly dispersive channels exhibiting delay and to extend the spanning to a number approaching one-hundred symbol periods, as discussed in this example in context of underwater communications [38] or broadband wireless access [36]. If channel characteristics are pre-defined, theoretically, it is possible to design transmitting and receiving letters with high degrees of accuracy with the removal of ISI from sampling instants. if the transmission rate R (in symbols per second) is less than the system bandwidth W (Hz) for a band-pass transmission. This condition is known as the Nyquist criterion for zero ISI [36]. In practice, it is not possible to achieve a frequency response of the channel; therefore, appropriate methods are employed to balance the ISI and to attain reliable communication.

The transmitter side for the frequency selective channel is illustrated in Fig. 7

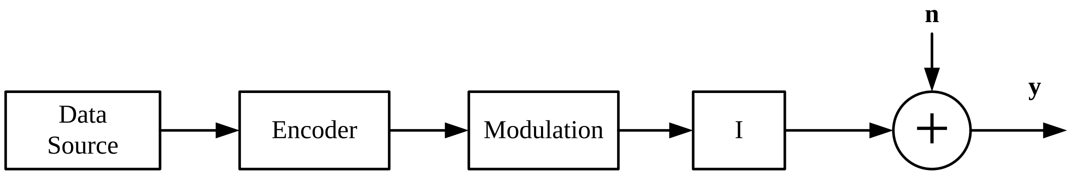

Figure 7: Transmitter of frequency selective channel.

3.2. The MAP Turbo-Equalizer

MAP turbo-equalization is specially designed to support the transmission of BPSK signals over ISI channels. The MAP turbo-equalization scheme based on simple principles and applies to PBSK. There are two receivers: the “SISO equalizer” and the “SISO decoder” which are similar to each other in their function and only differ in their interleaving operation as illustrated in Fig. 8. The following studies will emphasize the concept of Log-likelihood ratio because of its significance.

3.3. Log-Likelihood Ratios

Let us consider a coded bit $c_n^i \in \{0,1\}$. The binary log-likelihood ratio (LLR, or L-value) for c_n^i is denoted $L\left(c_n^i\right)$ and presented as

$$L\left(c_n^i\right) = \ln \frac{\Pr\left(c_n^i = 1\right)}{\Pr\left(c_n^i = 0\right)} \tag{3.1}$$

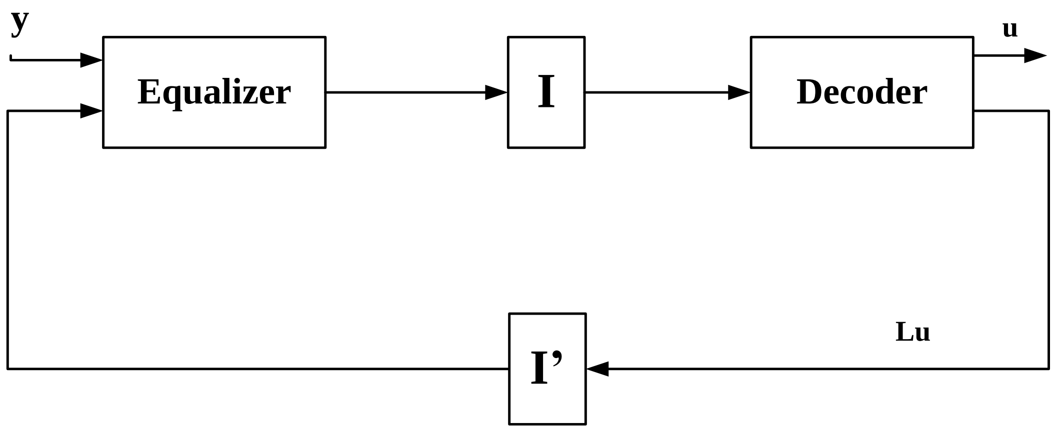

Figure 8: Turbo-equalizer scheme

This illustration laads us to

$$\Pr\left(c_n^i = b\right) = \frac{\exp\left[b.L\left(c_n^i\right)\right]}{1+\exp\left[L\left(c_n^i\right)\right]} \quad \text{, with } b = \{0,1\} \tag{3.2}$$

Generally, these definitions are quite simple in conditional probability.

Let us assume there is a signal set S with cardinality $|S| = M = 2^m$, also we assume that the sequence of m-coded bits $\{c_n^1,\cdots,c_n^m\}$ is mapped onto some symbol $x_n = s$ at time n, with $s \in S$. The coded bits c_n^i take their value in $\{0,1\}$ as a function of the considered symbol *s* and the labeling map. We define the symbol log-likelihood ratio $L_n(s)$ for *s* at time n as follows

$$L_n(s) = \ln\frac{\Pr(x_n = s)}{\Pr(x_n = s_0)} \tag{3.3}$$

where s_0 is reference symbol in the signal set M can be defined by symbol LLRs in time, one for each symbol in the signal set. Moreover, in every case $L_n(s_0) = 0$. The symbol probabilities are derived from the symbol LLRs as follows

$$\Pr(x_n = s) = K_n \times \exp\left[L_n(s)\right] \tag{3.4}$$

where the factor Kn is equal to $\Pr(x_n = s_0)$ and common to all symbol LLRs at time n.

The reference symbol s_0 can be selected randomly; however, it is suggested to select s0; therefore, $c_n^i = 0$ is yielded for all $i = 1, \cdots, m$ according to the signal labeling rule. This convention indeed raises some interesting properties. In particular, let us assume that the coded bits c_n^i are i.i.d., owing to the presence of a bit interleaver.

Then we have

$$\Pr\left(x_n = s\right) = \prod_{i=1}^{m} \Pr\left(c_n^i = l^i\left(s\right)\right) \tag{3.5}$$

Finally, the conversion from symbol LLRs to bit LLRs does not require any independent assumption, and can be obtained by noting that

$$\Pr\left(c_n^i = b\right) = \sum_{s \in S_b^i} \Pr\left(x_n = s\right) \text{ with } b = \{0,1\} \tag{3.6}$$

By Recalling the memory we be are able to say that S_b^i denotes the subset of all signals $s \in S$ whose label $l^i\left(s\right)$ has value $b = \{0,1\}$ in position i . This produces

$$L\left(c_n^i\right) = \ln \frac{\sum_{s \in S_b^1} \exp\left[L_n\left(s\right)\right]}{\sum_{s \in S_b^0} \exp\left[L_n\left(s\right)\right]} \tag{3.7}$$

3.4. Channel Estimation Using Blind Technique (PSP)

Blind channels can be estimated by using the per-survivor processing technique. The principle of Per-Survivor Processing (PPSP) is based on a general framework for the approximation of Maximum Likelihood Sequence Estimation (MLSE) algorithms. Realization of the classical Viterbi can be interrupted by the interference of unknown quantities. The principle is based on the idea of implanting data-aided estimation techniques into the structure of the Viterbi algorithm itself. Various applications have been developed to adapt MLSE and to reduce State Sequence Estimation (RSSE). Channel truncation techniques are used in RSSE to simplify the interpretation of principle. Several algorithms have been designed to estimate data sequences and unknown channels.

3.4.1. Adaptive MLSE based on MSE channel identification

Authors have widely discussed the application of Mean Square Error (MSE) channel estimation techniques in the realization of adaptive MLSE decoders [39], [40]. In every case the estimation process takes place on the basis of preface decision. At $k-th$ step, for all possible transitions $s_k \to s_{k+1}$, the following errors are calculated

$$\lambda_k(s_k, s_{k+1}) = y_k - a_k(s_k, s_{k+1}) \times h_k(s_k)^T \tag{3.8}$$

where

$$a_k(s_k, s_{k+1}) = (a_k, a_{k+1}, \cdots, a_{k+L})$$

L is the tap of channel

$$y_k = \sum_{i=0}^{L} a_{k-i} h_i + n$$

Where n represents the AWGN. One-step of the Viterbi algorithm is then performed.

$$\Gamma_{k+1}(s_{k+1}) = \min\left[\Gamma_k(s_k) + \left|e(s_k, s_{k+1})\right|^2\right] \tag{3.9}$$

The channel estimates $h_k(s_k)$ are then rationalized according to the stochastic gradient algorithm [l], by transitions $s_k \to s_{k+1}$ that satisfy Eq.(3.9). The constant β is used to cooperate between speed of convergence and stability. Each metric is related to survivor path and consists of a survivor sequence and a channel vector.

$$h_{k+1}(s_{k+1}) = h_k(s_k) + \beta e_k(s_k, s_{k+1}) a_k \tag{3.10}$$

3.5. Channel Estimation Using Non-Blind Technique (Training Sequence)

If binary data travels in blocks $u = (u_0, u_1, u_2, \cdots u_k)^T$, in presence of $u \in z^k$ with memory L then w will be the Goruareian noise i.e. $w_n = N(0, R)$ Where R is known as the noise covariance matrix. Thus for channels white Gaussian noise presented as (AWGN) $R = \sigma^2 I$ .where noise variance is $u_2 = N_0/2$. Here we assume that channel impulse response (CIR) is unknown and characterized by complex channel taps

$h_L = (h_0, h_0, \cdots h_L)$. Several codes are used to encode binary data such as $c \equiv u$, $c \in z^N$, more over interleaving schemes might b employed to destroy error clusters at the receiving side.

A known training sequence m is inserted into each block to assist channel estimation. The training sequence m consists of L + P symbols, where L is a preamble and P for mid amble symbols $m = (m_0, m_1, \cdots m_{(P+L-1)})^T$. As a result data block a of length N_b is arranged as follows $a = [c_1^T \ m^T \ c_2^T]^T$. Where $c_i = (c_0^{(i)}, c_1^{(i)}, ...c_{Nd-1}^{(i)})^T$ are coded data symbols separated into sub blocks and Nb = (2Nd+L+ P). Thus data block a is mapped into M-ary symbols and transmitted with the normalized symbol energy Es =l. According to maximum likelihood (ML) criteria the optimal receiver is written as

$$\hat{u} = \arg\max p(r \mid h, a) = \arg\max p(r \mid h, m \equiv u) \tag{3.11}$$

The most approving solution Eq. (3.11) is highly in practice, hence the that is solution Eq.(3.11) divide into parts, and each part is then considered individually.

If the training sequence is known, separate channel equalization (detection) and decoding take place in order to formulate the sub-optimal solution Eq.(3.11) shown as

$$\hat{a} = \arg\max_{h.a} p(r \mid h, a) \Rightarrow \hat{c} = \arg\max_{h.a} p(r \mid h, c) \tag{3.12}$$

$$\hat{u} = \arg\max_{u} p(\hat{c} \mid u) \tag{3.13}$$

The best solution should have the ability to search for all potential c and h, which is not possible to attain when a realistic value is L, Nd. Therefore, a typical sub-optimal solution Eq.(3.12) is divided for channel estimation and equalization as presented in Eq.(3.13). For example, one channel estimates the h in GSM from a known training sequence m, while the equalization algorithm is used to achieve e_k by removing the ISI. If the data is decoded, the transmitted information u will be recovered.

3.5.1. Channel estimation and sequence detection

Assuming a linear channel with time-invariant CIR during the transmitted block, the received block can be presented as

$$r = Ah + w = \begin{bmatrix} r_{c1} \\ r_m \\ r_{c2} \end{bmatrix} where\ A = \begin{bmatrix} A_1 \\ M \\ A_2 \end{bmatrix} \tag{3.14}$$

A a block matrix of size Nbx (L+l), M is a P x (L+l) matrix formed by the training sequence m as follows

$$M = \begin{bmatrix} m_L & m_{L-1} & .... & m_1 & m_0 \\ m_{L+1} & m_L & .... & m_2 & m_1 \\ .... & .... & .... & .... & .... \\ m_{p+L-1} & m_{p+L-2} & .... & m_p & m_{p-1} \end{bmatrix}$$

$$A_2 = \begin{bmatrix} c_0^{(2)} & m_{p+L-1} & .... & m_{p+1} & m_p \\ c_1^{(2)} & .... & .... & m_{p+2} & m_{p+1} \\ .... & .... & .... & .... & .... \\ c_{Nd-2}^{(2)} & c_{Nd-3}^{(2)} & .... & c_{Nd-L}^{(2)} & c_{Nd-L-2}^{(2)} \\ c_{Nd-1}^{(2)} & c_{Nd-2}^{(2)} & .... & c_{Nd-L-2}^{(2)} & c_{Nd-L-1}^{(2)} \end{bmatrix} \tag{3.15}$$

$$A_1 = \begin{bmatrix} c_L^{(1)} & c_{L-1}^{(1)} & .... & c_2^{(1)} & c_1^{(1)} & c_0^{(1)} \\ c_{L+1}^{(1)} & c_L^{(1)} & .... & c_3^{(1)} & c_2^{(1)} & c_1^{(1)} \\ .... & .... & .... & .... & .... & .... \\ m_{L-2} & m_{L-3} & .... & m_0 & c_{Nd-1}^{(1)} & c_{Nd-2}^{(1)} \\ m_{L-1} & m_{L-2} & .... & m_1 & m_0 & c_{Nd-1}^{(1)} \end{bmatrix}$$

When the receiver is predictable the channel estimation is based on midanible P symbols from the received sequence $r_m = Mh + w$ .For a channel with Gaussian noise likelihood function $p(r_m \mid m.h) \sim \exp(-\frac{1}{2}(r_m - Mh)^H R^{-1}(r_m - Mh))$ That gives ML channel estimation is given as

$$\hat{h}_{ML} = \arg\max_h p(r_m \mid M, h) = C(\hat{h}_{ML}) M^H R^{-1} r_m \tag{3.16}$$

Here $C(\hat{h}_{ML}) = (M^H R^{-1} M)^{-1}$ is a covariance matrix of estimation. In this case the ML channel estimate Eq.(3.16) is the minimum variance unbiased (MVU) and based on the data rm. Here h is provided, thus ML sequence can be determined by maximizing the likelihood function with respect to $\hat{c} = \arg\max_c p(r \mid \hat{h}, c)$ that can be implemented via ML -state Viterbi algorithm.

3.5.2. Iterative turbo estimation

In the turbo equalization scheme Fig.6 the iteration occurs only in between the signal detector and channel decoder whose iteration can be assumed as h. If the known training sequence is m, the channel estimate can be attained by using data r_m.

$$\hat{u} = \arg\max_u p(r \mid \hat{h}, m \equiv (u)) = \arg\max_u p(r \mid \hat{h}, a) \tag{3.17}$$

Accuracy of the channel estimate appears mostly low and depends only on a relatively short training sequence m, thus possibly leading to weakly significant performance at the receiver end. This limitation cannot be managed by the turbo equalization. This fact encouraged us to use a decision directed adaptive channel estimation method during the iteration process similar to [38]. This concept is based on the decoding of symbols to the channel estimator and update channel parameters.

The idea is to feed back the decoded symbols to the channel estimator and update the former channel estimates believing that the receiver is familiar with the whole burst Fig.12. Thus, the receiver relies on the decoded data symbols $\hat{u}$ and the known training sequence m in order to design a new channel estimate. At the first iteration, the estimated channel shows itself as Eq. (3.18).

$$\hat{h} = (A^H A)^{-1} A^H r \tag{3.18}$$

Here, a represents the training sequence in this iteration (A=M) and r represents the received sequence, while only the training sequence is present in the first iteration. In the next iterations, a represents the hole burst including the training sequence and response (estimated data). All of the received sequence is presented by the symbol r.

CHAPTER 4

TURBO EQUALIZATION PERFORMANCE, IMPLEMENTATION, SIMULATION AND RESULTS

4.1. Background

This chapter presents the performance of the turbo equalizer with the effect of channel estimation techniques. In this thesis, two techniques are employed for channel estimation: a blind technique and a non-blind technique. The non-blind technique deals with training sequences. The blind channel estimation is defined as using per-survivor processing to estimate the channel impulse response. Per-survivor processing is based on a least mean square algorithm. Consequent sections describe the proposed structure and simulation results. The results are measured by the bit error rate (BER) versus signal to noise ratio (SNR) in decibels (dB). The simulation environment is the MATLAB program.

4.2. Transmitter Description

The transmitter side presented in Fig.9 describes the data source. This source generates random data u_t , such as $u=[11001]$, which is coded by RSC (1,5/7) at a rate of $k/n = 1/2$ if the generator polynomial is $(1+D^2 \quad 1+D+D^2)$. The code c with length $2k$ is $c=[1110000110]$. Here, after the code sequence c is modulated by a binary phases shift keying (BPSK) $C=[111-1-1-1-111-1]$, the s-random interleaver rearranges the sequence. The interleaver is used in order for the data to avoid burst noise. This sequence is affected by the ISI channel according to the Eq. (4.1)

$$x_k = \sum_{i=0}^{L} h_i C_{k-i} \tag{4.1}$$

where L equals 3 (3-tap channel) and h represents non-zero samples $(h_0 = 0.407, h_1 = 0.815, h_2 = 0.407)$ as h sampling as illustrated in Fig. 10. When totaled, the sequence will form as x after the effect of the ISI channel. A trellis diagram of the ISI channel is shown in Fig.11.

$x = [-0.815,\ 0.815, 1.63\ ,\ 0.815,\ -0.815\ ,\ -1.63\ ,\ -1.63\ ,\ -0.815\ ,\ 0.815\ ,\ 0.815]$

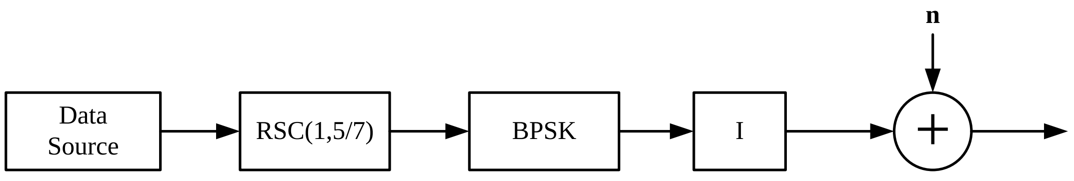

Figure 9: Transmitter scheme

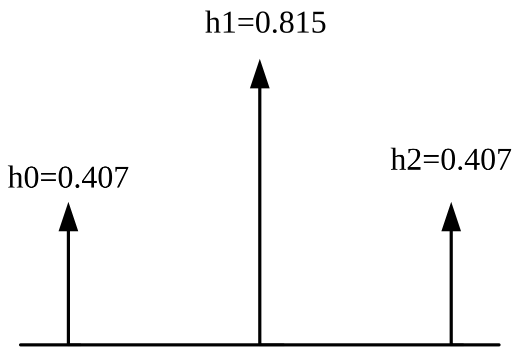

Figure 10: 3-tap channel samples

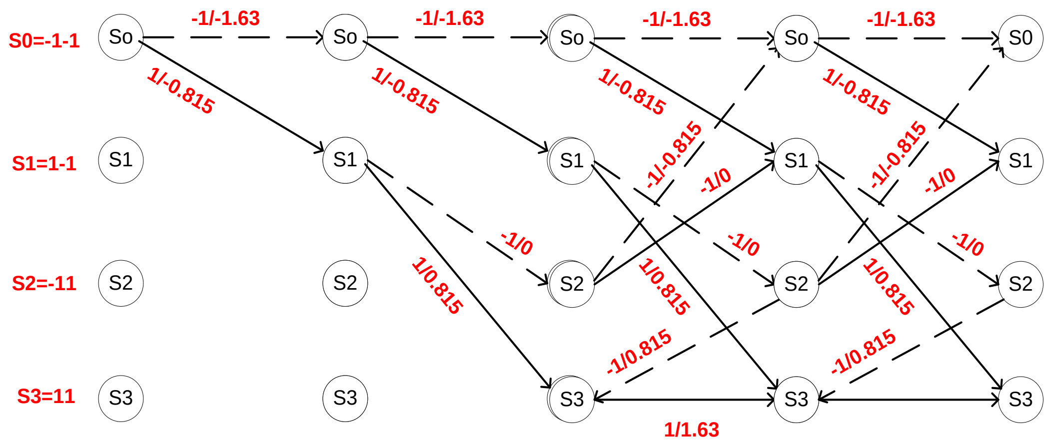

Figure 11: Trellis diagram of ISI channel

4.3. Receiver Description

The estimation of data at the receiver side is discussed in the following paragraphs. The received signal y transmits over additive white Gaussian noise (AWGN).

$$y_k = x_k + n_k \tag{4.2}$$

Fig. 12 shows the receiver block that comprises two parts: a signal equalizer and a signal detector (decoder). The signal equalizer requires information from the channel impulse response (CIR). The channel impulse response is determined by the estimator and feedback to the equalizer. The equalizer uses parameters (CIR) to compute the probabilities of data and parity bits. Thus, the probabilities proceed to the decoder through the de-interleaver. The decoder determines the data $\hat{u}$.

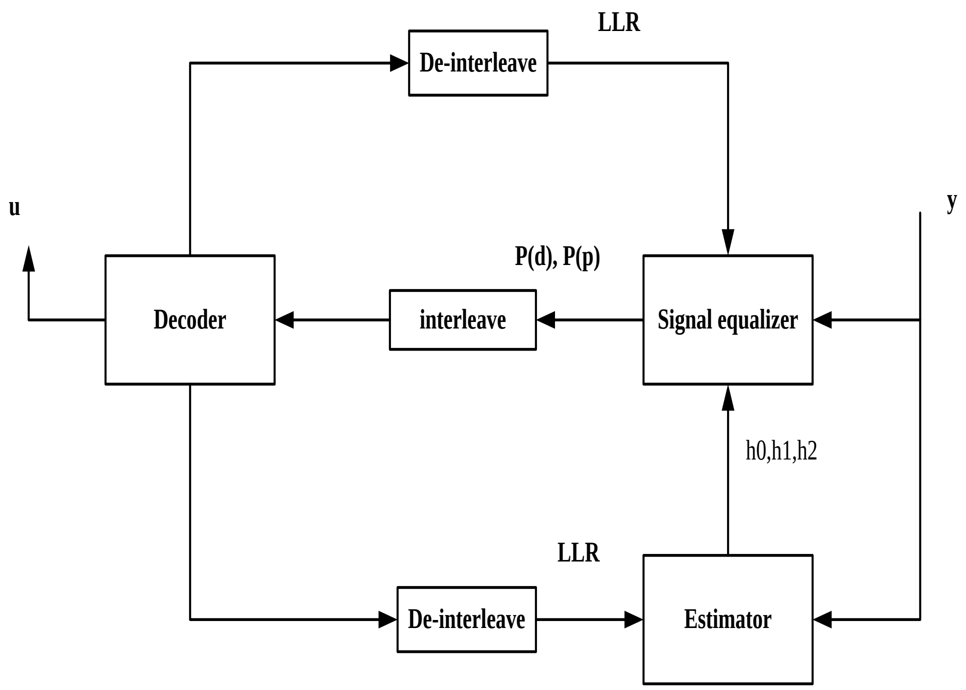

Figure 12: Receiver side scheme

4.3.1. Turbo equalizer (signal equalizer)

The principle of the turbo equalizer is discussed in the preceding chapter in 2.5 and 2.6. The log-map algorithm is applied to compute the data and parity bits. The equalizer receives the signal y and computes the transitions metric according to Eq. (4.3), (4.4).

$$p(x/y) = \frac{P(y/x)p(x)}{P(y)} \tag{4.3}$$

$$p(y/x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{(y-x)^2}{2\sigma^2}} \tag{4.4}$$

where $p(x)$ is the probability of the bit, σ^2 is the noise variance and $p(y)$ is a constant since it remains unchanged for all transitions. The log likelihood ratio

(LLR) is computed by using the log-map algorithm, while the transition metric as well as the forward and backward probabilities are computed according to Fig.11.

Hence

$$\gamma_k = p(x)p(y/x)$$

$$\gamma = p(x)\frac{1}{\sqrt{2\pi\sigma^2}}e^{\frac{(y-\hat{x})^2}{2\sigma^2}} \tag{4.5}$$

At the first iteration, $p(x)$ equals ½ since the data bits are in binary [1, -1]. $\hat{x}$ is computed from Eq. (3.1). The parameter h_i is determined by the estimator. After the computation of γ, forward and backward probabilities are computed. The (LLR) is passed to the decoder through the de-interleaver. The (LLR) is computed as per Eq.(2.8). The decoder is similar to the equalizer with a bti deference in γ_k which is computed from Eq. (4.5).

$$\gamma_k = P_d(x)p_p(x) \tag{4.6}$$

where $p_d(x)$ is the probability of the data bits, while $p_p(x)$ is the probability of parity bits.

4.4. Channel Estimator

Two techniques are discussed in this thesis for the estimation channel: the training sequence and per-survivor processing. The channel estimator provides the parameters to the equalizer.

4.4.1. Channel estimation using training sequence

A known training sequence is used in the first iteration and only one sequence is selected from the received signals. The known received signal sequence is used to estimate the channel parameters and further pass the parameters to the signal equalizer. The signal equalizer can compute the probabilities once a / the channel impulse response is attained. After that, the decoder receives the data and parity bits probability in order to determine the whole sequence. The decoder feeds the whole sequence back to the estimator. In this iteration (after first iteration), the estimator uses all sequences, including the training sequence, to determine the estimated channel impulse response as per Eq. (3.14), (3.15), and (3.16).

4.4.2. Channel estimation based on per-survivor processing

This method is known as the blind approach as it does not rely on any known order to estimate the channel impulse response. According to section 3.4, it computes the channel impulse response. Initially, it computes the error for each state using Eq. (3.9) from the trellis in Fig. 9. Then, it selects the minimum path for each state and updates the parameters according to that path, and completes the computations for all sequence until the trellis ends. After the completion of all states, four states are produced from which a minimum of one will be selected to obtain/determine the parameters of this state. These parameters will be forwarded to the signal equalizer in order to compute the probabilities. Thus, the decoder concludes the estimated sequence.

4.4.2.1. Enhancement 1

The blind channel estimation is better than the non-blind channel estimation such that the blind channel produces higher-quality output than non-blind channels. The performance of the turbo equalizer can also be increased by using the blind technique. The first optimization used the feedback sequence in the estimator to determine the parameters. A question arises as to how we can employ this technique. We have mentioned in per-survivor processing that the LMS algorithm is used for this purpose. In all transitions, computation of error occurs only in the first iteration. After the first iteration, an estimated sequence is used as a survivor path and computed the error only for this path further more updates the parameters till a sufficient number of iterations.

4.4.2.2. Enhancement 2

Optimization 2 provides better results than optimization 1. This technique depends on the feedback probability rather than the sequence. It works by inserting the probability of data and parity bits into the equation in order to update the channel impulse response as in Eq. (4.7)

$$h_{k+1}(s_{k+1}) = h_k(s_k) + \beta e_k(s_k, s_{k+1}) a_k + p(\hat{x}) \tag{4.7}$$

where

$p(\hat{x})$ is the feedback probability.

This term is added to this equation for the reason this equation will be used to evaluate the updated CIR. The feedback receives in form of probability of transmitted bit. Therefore, this probability has sufficient information to identify which path is surviving in trellis diagram.

4.5. Simulation Results

The results are measured by bit error rate (BER) versus signal to noise ratio (SNR). As compare between the results as shown in Fig. 13, the result of the second enhancement with dash red line is the best one. The blue line represents the idial performance with knowon CIR. The green line represents the performance with training sequence.

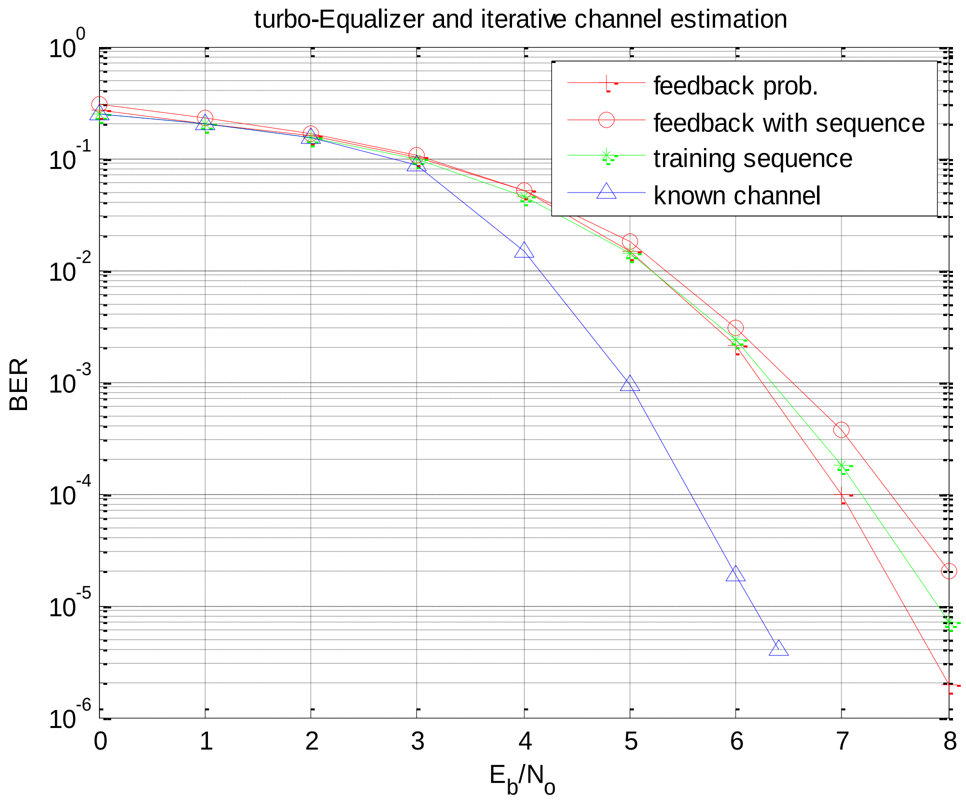

Figure 13: The effect of channel estimation on turbo equalizer performance.

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1. Conclusion

The major purpose to design this thesis was to explore Turbo equalizer and the effects of channel estimation techniques on its performance, particularly structure, performance, and the effects of channel estimation technique.

This research covers the following areas:

- Effect of non-blind channel estimation technique on performance of turbo equalizer
- Effect of blind channel estimation technique on performance of turbo equalizer
- Implemented of iterative channel estimation by using per-survivor processing (blind estimation). Feedback probability to channel estimator and use this feedback to estimate the coefficients.

In depth studies, discussion and analysis in early chapters led us to conclude that the performance of turbo equalizes with the use of non-blind technique is better than blind technique, whereas training sequence is applied in non-blind technique. The training sequence refers to extra bits in every transmitting burst. That’s why system needs high bandwidth while bandwidth may increase the cost. In blind technique, there is no any requirement to use any extra bit, resultant it does not perform well as compared to training sequence. The blind technique accompanied with feedback probability performs well comparative to training sequence and blind technique alone.

5.2. Future Work

A natural direction of future research is the extension of the turbo-equalization scheme to multiple-input multiple-output systems, where the turbo-equalizer could fully exploit the diversity offered by a rich-scattering multipath environment. Another interesting research changes the type of interleaver as mentioned in chapter two. Another interesting involves investigating the possibility of testing this research on really fading channel and use more tap channels. Finally, investigate the possibility of taking explicit advantage of a priori information and feedback sequence in the channel estimation process.

REFERENCES

1. **Benedetto S., Divsalar D., Montorsi G., Pollara F., (1997*),*** *"A Soft-Input Soft-Output APP Module for Iterative Decoding of Concatenated Codes"*, IEEE Communication Letters, vol. 1, no. 1, pp. 22–24.

2. **Benedetto S., Divsalar D., Montorsi G., Pollara F., (1998),** *"Serial Concatenation of Interleaved Codes: Performance Analysis, Design, and Iterative Decoding"*, IEEE Transactions Information Theory, vol. 44, no. 3, pp. 909–926.

3. **Berrou C., Glavieux A., Thitimajshima P., (1993*),*** *"Near Shannon Limit Error-correcting Coding and Decoding: Turbo-Codes"*, in IEEE International Conference on Communications, vol. 2, no. 10, pp. 1064–1070.

4. **Berrou C., Glavieux A., (1996),** *" Near Optimum Error Correcting Coding and Decoding: Turbo Codes",* IEEE Transactions Communication, vol. 44, no. 10, pp. 1261–1271.

5. **Caire G., Taricco G., Biglieri E.,(1998**), *"Bit-Interleaved Coded Modulation"*, IEEE Transactions Information Theory, vol. 44, no. 3, pp. 927–946.

6. **Cavers J. K., (1991),** *"An Analysis of Pilot Symbol Assisted Modulation for Rayleigh Fading Channels"*, IEEE Transactions Vehicular Technology, vol. 40, no. 4, pp. 686–693.

7. **Chugg K. M., Polydoros A., (1996),** "*MLSE for An Unknown Channel – Part I: Optimality Considerations*", IEEE Transactions Communication, vol. 44, no. 7, pp. 836–846.

8. **Davis L. M., Collings I. B., Hoeher P.,(2001),** *"Joint MAP Equalization and Channel Estimation for Frequency-Selective and Frequency-Flat Fast-Fading Channels"*, IEEE Transactions Communication, vol. 49, no. 12, pp. 2106–2114.

9. **Douillard C., Jezequel M., Berrou C., (1995),** "*Iterative Correction of Intersymbol Interference: Turbo Equalization"*, Eur. Transactions Telecommunication, vol. 6, no. 5, pp. 507–511.

10. **Edfors O., Sandell M., Van de Beek J., Wilson S. K., Borjesson P. O.,(1998**), *"OFDM Channel Estimation By Singular Value Decomposition"*, IEEE Transactions Communication, vol. 46, no. 7, pp. 931–936.

11. **El Gamal H., Geraniotis E., (2000),** *"Iterative Multiuser Detection for Coded CDMA Signals in AWGN and Fading Channels"*, IEEE Jr. Select Areas Communication, vol. 18, no. 1, pp. 30–41.

12. **Franz, V., Anderson, J. B., (1998),** "*Concatenated Decoding with A Reduced-Search BCJR Algorithm"*, IEEE Jr. Select Areas Communication, vol. 16, no. 2, pp. 186–195.

13. **Hagenauer J., Hoeher, P., (1989),** *"A Viterbi Algorithm with Soft-Decision Outputs and Its Applications"*, IEEE Global Telecommunications Conference, vol. 3, no. 5, pp. 1680–1686.

14. **Hagenauer J., Offer E., Papke L., (1996),** *"Iterative Decoding of Binary Block and Convolutional Codes"*, IEEE Transactions Information Theory, vol. 42, no. 2, pp. 429–445.

15. **Hoeher P., (1991),** "*TCM on Frequency-Selective Land-Mobile Fading Channels"*, Workshop on Digital Communication, vol. 18, no. 3, pp. 317–328.

16. **Hoeher P., Lodge J., (1999),** "*Turbo DPSK: Iterative Differential PSK Demodulation and Channel Decoding*", IEEE Transactions Communication, vol. 47, no. 6, pp. 837–843.

17. **Komninakis C., Wesel R. D., (2001**), "*Joint Iterative Channel Estimation and Decoding in Flat Correlated Rayleigh Fading*", IEEE Jr. Select Areas Communication, vol. 19, no. 9, pp. 1706–1717.

18. **Li X., Chindapol A., Ritcey J. A., (2002),** "*Bit-Interleaved Coded Modulation with Iterative Decoding and 8PSK Signaling*", IEEE Transactions Communication, vol. 50, no. 8, pp. 1250–1257.

19. **Moher M. L., Lodge J. H., (1990),** *"Rayleigh Flat- Maximum Likelihood Sequence Estimation of CPM Signals Transmitted Over Fading Channels"*, IEEE Transactions Communication, vol. 38, no. 6, pp. 787–794.

20. **Sandell M., Luschi C., Strauch P., Yan R., (1998),** *"Iterative Channel Estimation Using Soft Decision Feedback"*, IEEE Global Telecommunications Conference, vol. 6, no. 8, pp. 3728–3733.

21. **Sun Y., Yee M., Sandell M., (2003),** "*Iterative Channel Estimation with MIMO MMSE-Turbo Equalisation",* IEEE Vehicular Technology Conference, vol. 2, no. 4, pp. 1278– 1282.

22. **Tan J., Stuber G. L., (2005),** "*Analysis and Design of Interleaver Mappings for Iteratively Decoded BICM*", IEEE Transactions Wireless Communication, vol. 4, no. 5, pp. 662–672.

23. **Tuchler M., Koetter R., Singer A. C., (2002),** *"Turbo Equalization: Principles and New Results Unknown"*, IEEE Transactions Communication, vol. 50, no. 5, pp. 754–767.

24. **Valenti M. C., Woerner B. D., (2001),** *"Iterative Channel Estimation And Decoding of Pilot Symbol Assisted Turbo Codes Over Flat-Fading Channels",* IEEE Jr. Select Areas Communication, vol. 19, no. 9, pp. 1697–1705.

25. **Yu X., Pasupathy S., (1995),** *"Innovations-Based MLSE for Rayleigh Fading Channels"*, IEEE Transactions Communication, vol. 43, no. 3, pp. 1534–1544.

26. **Bahl L. R., Cocke J., Jelinek F., Raviv J., (1974),** *"Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate"*, IEEE Transactions Information Theory, vol. 20, no. 2, pp. 284–287.

27. **Banerjee A., Vatta F., Scanavino B., Costello D. J., (2005),** *"Nonsystematic Turbo Codes"*, IEEE Jr. Transactions Communications. vol. 53, no. 11, pp. 1841–1849.

28. **Berrou C., Glavieux A., Thitimajshima P., (1993),** *"Near Shannon Limit Error-Correcting Coding and Decoding: Turbo-Codes"*, Geneva, Switzerland, pp. 1064–1070.

29. **Lin S., Costello D. J., (2003),** *"Error Control Coding"*, Pearson Higher Education, 2ed edition, pp. 123-125.

30. **Robertson P., Villebrun E., Hoeher P., (1995),** *"A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in The Log Domain"*, IEEE Communications Conference, vol. 2, no. 7, pp. 1009–1013.

31. **Woodard J. P., Hanzo L., (2000),** *"Comparative Study of Turbo Decoding Techniques: An Overview"*, IEEE Transactions on vehicular Technology, vol. 49, no. 6, pp. 2208–2232.

32. **Ramsey J. L., (1970),** *"Realisation of Optimum Interleavers"*, IEEE Transactions on Information Theory, vol. 16, no. 3, pp. 338-345.

33. **Ariyavisitakul S. L., Greenstein L. J., (1997),** *"Reduced-Complexity Equalization Techniques for Broadband Wireless Channels"*, IEEE Jr. Selection Areas Communication, vol. 15, no. 1, pp. 5-15.

34. **Proakis J. G., (2000),** *"Digital Communications"*, 4th edition, New-York, McGraw-Hill, pp.765-769.

35. **Stojanovic M., (1996),** *"Recent Advances in High-Speed Underwater Acoustic Communications"*, IEEE Jr. Oceanic Engineer, vol. 21, no. 2, pp. 125-136.

36. **Proakis J. G., (1983),** *"Digital Communications"*. New York, McGraw-Hill, pp. 289-293.

37. **Chevillat P. R., Eleftheriou E., (1989),** *"Decoding of Trellis- Encoded Signals in the Presence of Intersymbol Interference and Noise"*, IEEE Transactions Communication, vol. 37, no. 7, pp. 669-676.

38. **Didier P., Glavieux A., (1997),** *"Turbo- Detection: A New Approach to Combat Channel Frequency Selectivity"*, IEEE International Conference on Communication, vol. 3, no. 9, pp. 1498-1502

APPENDICES A

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: AL-ISAWI, Ahmed Sami Ali
Date and Place of Birth: 10 May 1990, Baghdad
Marital Status: Single
Phone: +90 535 847 45 28 / +964 770 767 65 61
Email: **asaahmed_aa@yahoo.com**

EDUCATION

Degree	Institution	Year of Graduation
M.Sc.	Çankaya Univ., Electronic and Communication Engineering	2014
B.Sc.	Mustansiriya Univ.,Electrical Engineering	2012
High School	AL-Jwaheri High School	2008

WORK EXPERIENCE

Year	Place	Enrollment
2012 June	AL-Rayhana Company To Flour Milling	Electrical Engineer

FOREIN LANGUAGES
Arabic (Mother language), English, Beginner Turkish

PUBLICATIONS

1. AL-Isawi, A., "*The Performance of Turbo Equalization Using Blind and Non-Blind Channel Estimation Technique*", Progress in URSI Electronic and Communication Research, Fırat University, Turkey, (2014).

HOBBIES
Travel, Reading, Programing, Video Games.