# A Comparison of Turbo Equalization Techniques for Stationary Intersymbol Interference Channels

by

**Nelson Lin, B. Sc**

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of

**Master of Applied Science**

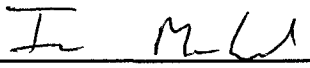
Ottawa-Carleton Institute for Electrical Engineering
Faulty of Engineering
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario, Canada, K1S 5B6
September 19, 2002

The undersigned recommend to the Faculty of Graduate Studies and Research acceptance of the thesis

# A Comparison of Turbo Equalization Techniques for Stationary Intersymbol Interference Channels

submitted by Nelson Lin, B. Sc,
in partial fulfillment of the requirements for
the degree of Master of Applied Science

**Chair, Department of Systems and Computer Engineering**



**Thesis Supervisor**

Carleton University
September 18, 2002

# Abstract

In wireless communications, one main obstacle to reliable data transmission is intersymbol interference (ISI). With ISI, the transmitted signal experiences non-trivial impairment in which the received sample depends not only on the transmitted signal, but also on delayed versions of the transmitted signal. To suppress the ISI, equalization techniques are generally used, where the maximum *a posteriori* probability (MAP) equalizer, maximum likelihood sequence estimation (MLSE), decision feedback equalizer (DFE), linear equalizer (LE), and interference canceller (IC) are the popular techniques. However, none of these techniques are able to perfectly remove the ISI, so there is a substantial ISI penalty compared to transmission over an ISI-free channel. To overcome this penalty, various turbo equalization strategies have been proposed. Turbo equalizers combine an equalizer and a channel decoder and operate using an iterative decoding strategy, similar to decoders for turbo codes.

In this thesis, we extensively compare the performance of six different turbo equalizer structures over several stationary ISI channel models. MAP-based and DFE-based, and LE-based turbo equalizers are considered, with both soft- and hard-decision feedback. In general, the MAP-based turbo equalizer provides the best performance, coming very close to the coded

AWGN channel performance without ISI; however, the high complexity limits its versatility. We observe that there is little difference between LE-based and DFE-based turbo equalizers. We also show that there is only a slight reduction in performance when hard-decision feedback is used instead of soft-decision feedback, with the advantage of lower complexity.

# Table of Contents

# List of Figures

# Abbreviations

| | |
|---|---|
| BER | bit error rate |
| dB | decibel |
| DFE | decision feedback equalizer |
| IC | interference canceller |
| ISI | intersymbol interference |
| LE | linear equalizer |
| MAP | maximum *a posteriori* probability |
| MLSE | maximum likelihood sequence estimation |
| PSK | phase-shift keying |
| SNR | signal-to-noise ratio |

# Chapter 1 Introduction

Today, digital communication plays a significant role in modern systems such as wireless, satellite and computer communications. In these systems, information is represented as a sequence of binary digitals named the information sequence. By using one of different modulation schemes, the information sequence is mapped to analog signal waveforms which are transmitted over a communication channel.

In a channel with *additive white Gaussian noise* (AWGN), the transmitted signals will be corrupted by random noise signals. At the receiver end, the corrupted transmitted signals are mapped back into bits; however, bit errors may occur due to the amount of noise in the channel. To protect the information from the noise, channel coding is applied by adding redundant bits into binary information.

In most band-limited channels such as wireless and satellite channels, AWGN is not the only perturbation to corrupt the transmitted signals. The transmitted signal also undergoes multi-path distortion. The received sample at any time instant depends not only on the transmitted signal at that time instant, but also on previously transmitted signals. The result is *intersymbol interference* (ISI). ISI can have a profound effect on the system performance, to the

point where reliable performance is not possible even with sophisticated channel coding. Some method for suppressing the ISI must be employed at the receiver. Methods for suppressing the ISI are generally referred to as equalizers.

*Maximum likelihood sequence estimation* (MLSE) is one robust equalization technique. By modeling the channel as a finite state machine, the Viterbi algorithm [1] can be used to suppress the ISI. In the absence of channel coding, MLSE is an optimal equalization technique, in the sense that the output of the equalizer is the information sequence that was most likely to have been transmitted. An alternative to MLSE is *maximum a posteriori* (MAP) equalization, as proposed by Chang and Hancock [2]. In the absence of channel coding, this technique is optimal, but in the sense that it minimizes the probability of a bit error instead of sequence error. Although over twice as complex as MLSE, MAP equalization has the distinct advantage in that it can be configured to produce soft-output, which will permit soft-output decoding of the channel code. Both MLSE and MAP equalization have very high complexity, so simpler equalization techniques are often preferred.

One alternative is *linear equalization* (LE), which uses a simple linear filter. However, it may perform poorly in severe ISI channels due to excessive noise enhancement at the frequencies

corresponding to large attenuation [3]. *Decision feedback equalization* (DFE) is another method, which has similar complexity to the LE, but with less noise enhancement [4]. In 1981, Gersho and Lim [5] and Mueller and Salz [6] proposed an equalizer combined with a matched filter and a linear interference canceller. This equalizer is able to totally eliminate the intersymbol interference with the criterion that the past and/or future transmitted signals are known; otherwise, a linear equalizer estimates the transmitted signals previously. Because of this reason, the receiver performance highly relies on the performance of the linear equalizer.

In 1988, Eyuboglu [7] proposed the *Noise-predictive DFE* with interleaving. This noise-predictive DFE is composed of a DFE, a channel decoder, and a periodic interleaver. In ideal DFE, the delay-free decisions are required for feedback; however, in a coded system, such decisions are not reliable. Therefore, by using the periodic interleaver, it provides some delay for feedback in order to generate delayed reliable decisions. In such a receiver, the equalizer can use hard decisions provided by the channel decoder. In [7], it showed that the noise-predictive DFE with interleaving can approach the performance of ideal DFE when a long delay can be tolerated. Note that this receiver is dependent on the channel decoder.

In 1993, Berrou, Galvieux and Thitmajshima [8] first introduced the "turbo concept" as an

iterative decoding algorithm for parallel concatenated convolutional codes ("turbo code"). With this algorithm, two disjoint decoders, implemented using the soft-output MAP algorithm, repeatedly exchange soft information in an iterative fashion. Although sub-optimal[1], this algorithm is nonetheless very effective. After the discovery of the turbo concept for decoding turbo codes, the algorithm has been applied to several other problems, including joint equalization and decoding.

In 1995, the "turbo concept" was proposed in a receiver called a *turbo-equalizer* [9], which combined a MAP equalizer with a MAP decoder through an iterative process. In many wireless channels, the performance of this turbo-equalizer is nearly optimal; however, both the MAP equalizer and the MAP decoder require extremely high computational complexity, which limits its versatility. To reduce the high computational complexity, the MAP equalizer can be replaced by an *interference canceller* (IC) during the second and subsequent iterations [10]. When the transmitted signals are known previously, the IC can completely eliminate the ISI. To further reduce the complexity, the MAP equalizer used for the first iteration can be replaced by the LE or DFE while the performance degradation is expected. For decoding, both the MAP algorithm and the Viterbi algorithm can be employed. In an iterative process, the MAP decoder can provide

[1] An optimal decoder would treat the concatenated codes as a single code, instead of two separate ones. Optimal decoders are prohibitively complex in this case.

soft-decision output for use in the next iteration, whereas the Viterbi decoder only provides hard-decision output.

In this thesis, we are going to extensively compare the performance of the following previously proposed turbo equalization schemes in both coded QPSK and coded 8PSK:

1) MAP-based turbo equalization (MAP-based)

2) Soft-decision feedback interference canceller based turbo equalization with linear equalizer used at the first iteration (LE/IC-soft)

3) Hard-decision feedback interference canceller based turbo equalization with linear equalizer used at the first iteration (LE/IC-hard)

We also present and compare the following slight modification of the above schemes:

4) Soft-decision feedback interference canceller based turbo equalization with MAP equalizer used at the first iteration (MAP/IC-soft)

5) Soft-decision feedback interference canceller based turbo equalization with decision feedback equalizer used at the first iteration (DFE/IC-soft)

6) Hard-decision feedback interference canceller based turbo equalization with decision feedback equalizer used at the first iteration (DFE/IC-hard)

The main goal of this thesis is to broaden the readers' understanding on how the modulation schemes, symbol rates, error control coding schemes, interleaving, block sizes, and channel models affect the performance of the previously invested turbo equalization schemes and of the slightly modified turbo equalization schemes. We will mainly focus on the following three

different elements: block size, channel model, and interleaving. In general, we know that changing the size of transmitted blocks can have an impact on the overall system performance. Therefore, to examine the impact on the performance of turbo equalization we select three different block sizes, which are 200, 1000, and 5000. The performance of six different turbo equalization schemes will be simulated and compared over five different channel models where some of the models are also used in [9] [12], and [13]. For the interleaving, both binary- and symbol-based interleaving will be considered in the turbo equalization such that the performance difference of presented turbo equalization schemes will be drawn based on different interleaving. In addition, for coded 8PSK, there are three different convolutional codes are employed in the turbo equalization. Two codes are used with trellis coded modulation and one with Gray mapping. Therefore, we can investigate the performance of turbo equalization when trellis modulation scheme is used with coded 8PSK and also when a typical convolutional code is used. This thesis is organized as follows: Chapter 2 describes the system model, Chapter 3 presents the turbo equalization scheme, Chapter 4 illustrates the simulation results and the conclusion of this thesis will be drawn in Chapter 5.

# Chapter 2 System Model

A block diagram of the system model considered in this thesis is shown in Figure 2-1. The transmitter consists of a convolutional encoder, an interleaver, and a symbol mapper. The receiver consists of an equalizer, a de-interleaver, and a channel decoder. These components, along with the ISI channel model, are described in this chapter. A description of turbo equalization and various turbo-equalizer structures is given in Chapter 3.

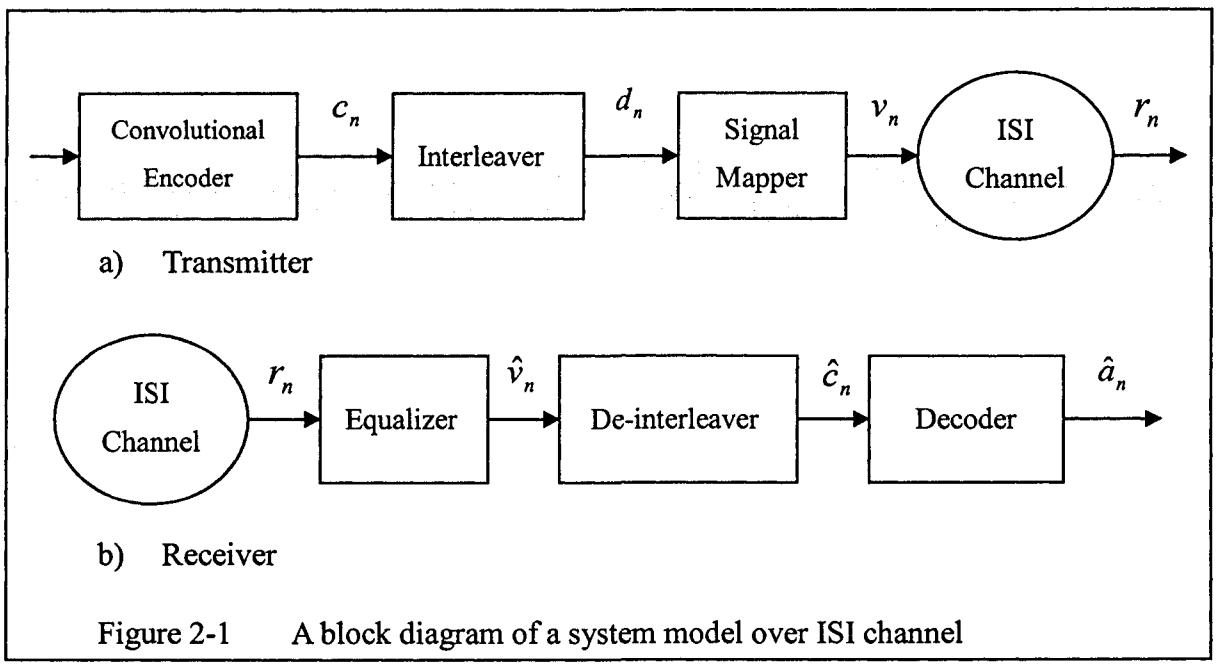



Figure 2-1 A block diagram of a system model over ISI channel

## 2-1 Convolutional Encoder

Error control coding is a useful tool to improve the system performance by allowing lower $E_b/N_o$ than could be attained by the same modulation scheme without coding at a fixed BER. Figure 2-2 shows a system model over an AWGN channel using an error control coding scheme.

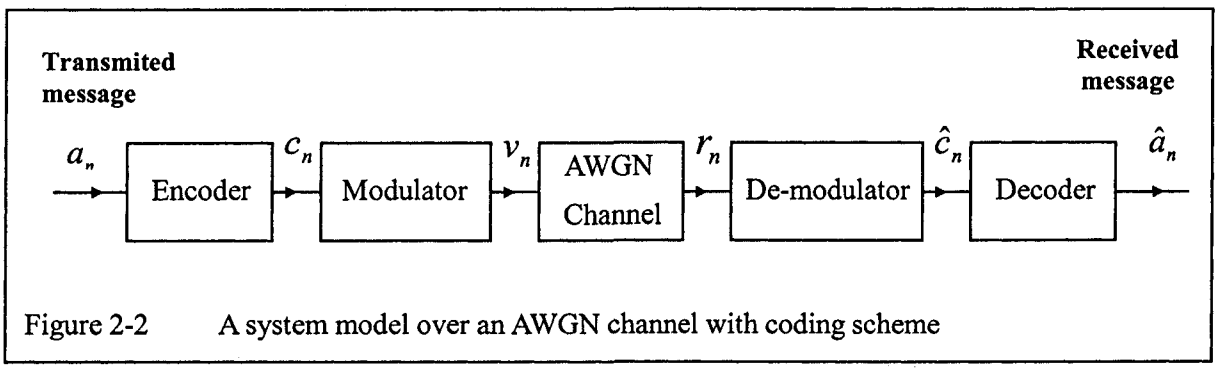



Figure 2-2 A system model over an AWGN channel with coding scheme

Convolutional code is one of error control coding schemes. It accepts bits in stream and produces a codeword with memory. In this thesis, we only consider convolutional codes since they are easily implemented with soft-output decoding which is useful for the turbo equalization techniques. Figure 2-3 depicts a convolutional encoder with $n_c$=3, $k_c$=2, and $K_c$=3 where $k_c$ is the number of message bits that enter the encoder at a time, $n_c$ is the total number of bits that are generated by the encoder at a time, and $K_c$ is the encoder constraint length. The output of the encoder depends not only on the current input message symbol, but it also depends on the previous $K_c$-1 input message symbols. A convolutional code can be represented by the trellis diagram and

the state diagram, which are two important analytical techniques.

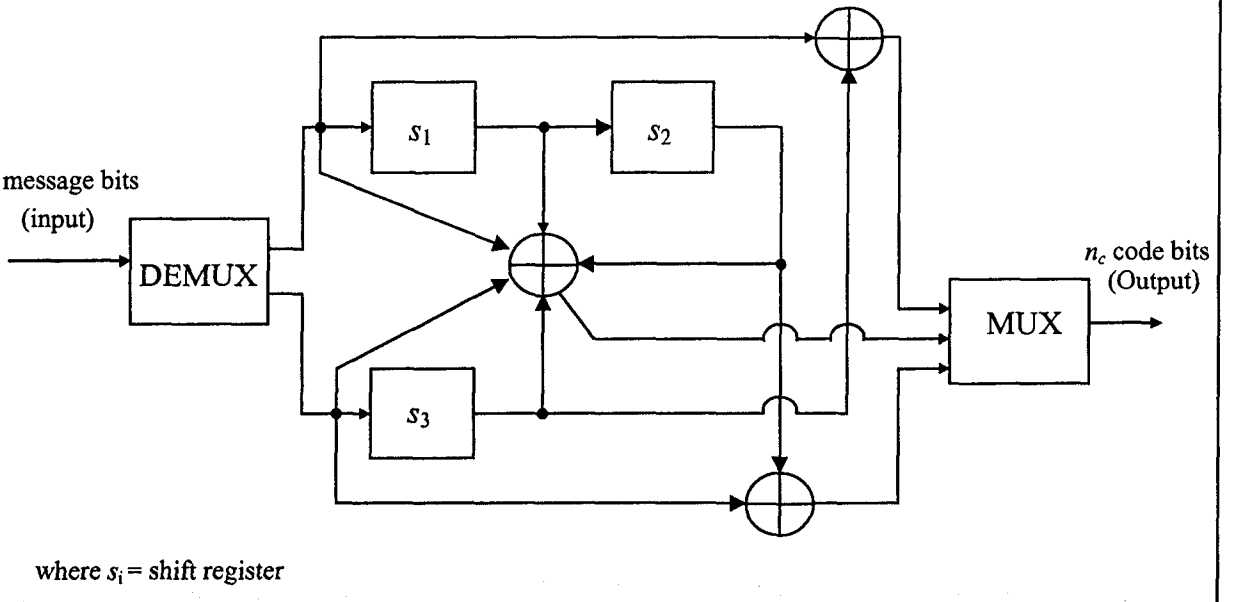



Figure 2-3 A structure of (2,3) convolutional encoder with constraint length of 3

## 2-2 Interleaving

Most coding schemes are designed to combat statistically independent errors caused by memoryless channels such as the AWGN channel; however, the channel characterized by intersymbol interference is a channel with memory. As a result, convolutional codes, which are designed to compensate for random independent errors, not error bursts, are not particularly effective. The idea of interleaving is to separate the code symbols in time in a way that the burst error channel appears like a memoryless channel, thereby restoring the effectiveness of convolutional codes.

There are different types of interleavers such as block interleavers, convolutional interleavers, and random interleavers. For the sake of simplicity, we are only interested in random interleaver in our analysis. A random interleaver uses a fixed random permutation and rearranges the input sequence according to the permutation order. Interleavers can be designed to operate on a bit-by-bit basis or a symbol-by-symbol basis. In a binary-based interleaver, the individual bits of the input sequence are rearranged, whereas in a symbol-based interleaver, the symbols are rearranged, but the bits within each symbol remain intact. Figure 2-4 shows examples of binary-based and symbol-based interleaver. For the binary-based interleaver, if the

input sequence with the length of 10 bits is [1000111011], the output of the random interleaver is [0110011101]. For a two-bit symbol interleaver, the output for the same input sequence would be [1101110001].

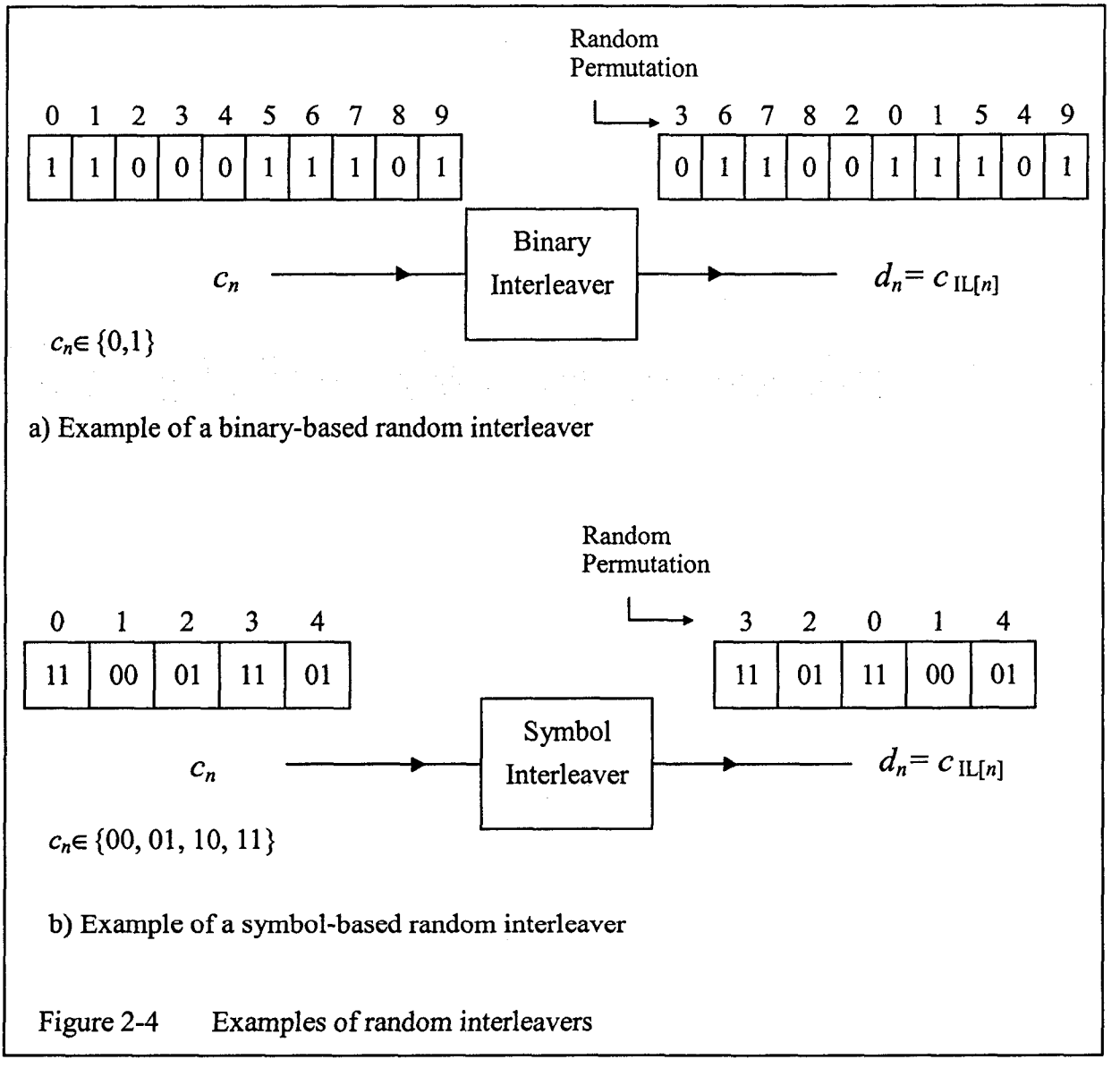



Figure 2-4 Examples of random interleavers

## 2-3 Modulation

Digital modulation is the process by which information symbols are mapped into analogue waveforms. The mapping is done by taking blocks of $k = \log_2 M$ binary digits at a time from the code sequence $\{c_n\}$ and selecting one of $M=2^k$ waveforms $\{s_m(t), m=0,1,2,3,\ldots, M-1\}$ for transmission over the channel. The most common digital modulation schemes are Phase Shift Keying (PSK), Frequency Shift Keying (FSK) and Amplitude Shift Keying (ASK). In modern commercial communications system, PSK is widely used; therefore, in our analysis, we focus on $M$-ary PSK.

The general signal waveforms for PSK can be expressed as

$$s_m(t) = \sqrt{\frac{2E_s}{T_s}} \cos[2\pi f_c t + \phi_m] \qquad \begin{matrix} 0 \le t < T_s \\ m = 0,1,2,\ldots,M-1 \end{matrix} \qquad (2\text{-}1)$$

where

$$\phi_m = \frac{2\pi m}{M} \qquad m = 0,1,2,\ldots,M-1$$

$T_s$ = symbol duration

$E_s$ = symbol energy

$f_c$ = carrier frequency

Furthermore, the signal waveforms can also be represented as two-dimensional vectors, such as

$$s_m = \left[\sqrt{E_s}\cos\frac{2\pi m}{M} \quad \sqrt{E_s}\sin\frac{2\pi m}{M}\right] \qquad i = 0,1,2,\ldots,M-1 \qquad (2\text{-}2)$$

and can be shown in a signal space diagram. Figure 2-5 illustrates some examples of signal space diagrams for $M$-ary PSK where $M$=2, 4, and 8. Note that the energy of each symbol is the square of the distance between the origin and the symbol and the average energy per symbol is

$$E_{s,ave} = \frac{1}{M}\sum_{m=0}^{M-1}|s_m|^2 \qquad (2\text{-}3)$$

For the $M$-ary PSK, the energy of each symbol is the same as the average energy since all symbols have the same distance from the origin, i.e. $E_s = E_{s,ave}$.

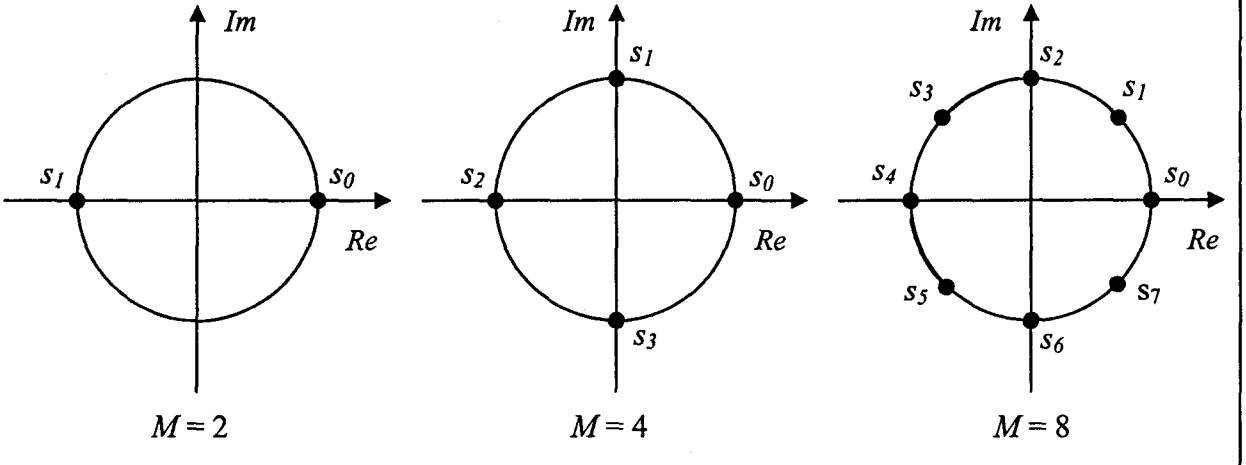



Figure 2-5 Signal space diagrams for $M$-ary PSK for $M$ = 2, 4, and 8

The $2^k$ $k$-bit code symbol have to be mapped to $M = 2^k$ possible symbols in the signal constellation. There are different ways of mapping. One of the ways is to have one bit difference between the information bits assigned to one symbol and the adjacent symbols. This method is called Gray Mapping. In this way, usually only a single bit error occurs when a symbol error occurs. Figure 2-6 illustrates the Gray mapping for 8-PSK. For example, the information bits 000 is assigned to symbol $s_0$ where the information bits assigned to adjacent symbols, (i.e. $s_1$ and $s_4$), differ by one bit from 000, (i.e. $s_1 = 001$ and $s_4 = 100$).

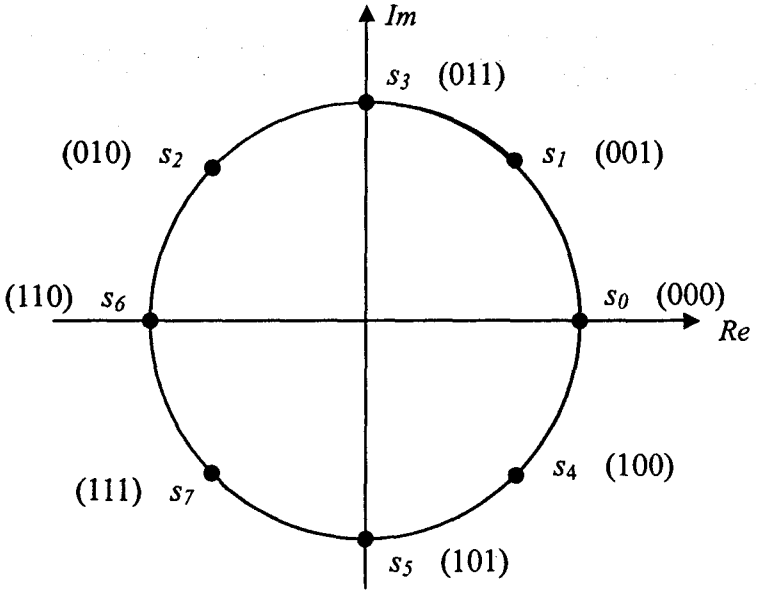



Figure 2-6 Gray mapping for 8-PSK

## 2-4 Intersymbol Interference (ISI)

In the presence of both additive white Gaussian noise (AWGN) and intersymbol interference that spans $L+1$ symbols on the channel, each transmitted symbol experiences non-trivial impairment due to channel distortion. The resulting signal produced by the channel is the addition of the current and past transmitted symbols weighted with different channel coefficients, plus the additional white Gaussian noise. Since the ISI channel has a similar nature to a convolutional encoder, the channel can also be viewed as a convolutive channel.

In an analog system, the transmitted signal is expressed as [11]

$$v(t) = \sum_{n=0}^{N-1} v_n h_T(t - nT)$$

where $\{v_n\}$ represents the discrete information-bearing sequence of symbols and $h_T(t\text{-}nT)$ is a pulse shape that is assumed to have a band-limited frequency response $H_T(f)$, i.e. $H_T(f)=0$ for $|f| > W$. When this signal is transmitted over a channel with a frequency response $C(f)$, also limited to $|f| \leq W$. The received signal can be represented as

$$r_i(t) = \sum_{n=0}^{N-1} v_n g(t - nT) + \eta(t)$$

where

$$g(t) = \int_{-\infty}^{\infty} h_T(\tau)\, c(t-\tau)\, d\tau$$

and $\eta(t)$ is the additive white Gaussian noise with single-sided noise power spectral density of $N_0$. At the receiver, the received signal $r_i(t)$ is sent to a matched filter, $g^*(-t)$, and sampled at a rate of $1/T$ samples per second which corresponds to the symbol rate of $1/T$ at the transmitter. Therefore, the sampled output of the matched filter can be expressed as

$$y_n = \sum_k v_k z_{n-k} + p_n$$

where

$$b_n = b(nT) = \int_{-\infty}^{\infty} g^*(t) g(t+nT) dt$$

and $p_n$ is the additive noise sequence of the output of the matched filter, i.e.,

$$p_n = \int_{-\infty}^{\infty} \eta(t) g^*(t-kT) dt$$

with the auto-variance of $N_0 b_{j-k}$ ( $|j-k| \leq L$).

Since the noise sequence $\{p_n\}$ is correlated, a noise-whitening filter, $1/F^*(z^{-1})$, is required in order to simplify the calculation of error performance where

$$B(z) = F(z) F^*(z^{-1}) \qquad (z\text{-transform of } b_n)$$

Hence, the output of the noise-whitening filter is

$$r_n = \sum_{l=0}^{L} f_l v_{n-l} + w_n \qquad (2\text{-}4)$$

where

$r_n$ = $n$-th received sample

$v_n$ = $n$-th transmitted symbol

$w_n$ = $n$-th additive white gaussian noise sample

$f_l$ = $l$-th channel taps

$L$ = channel ISI length

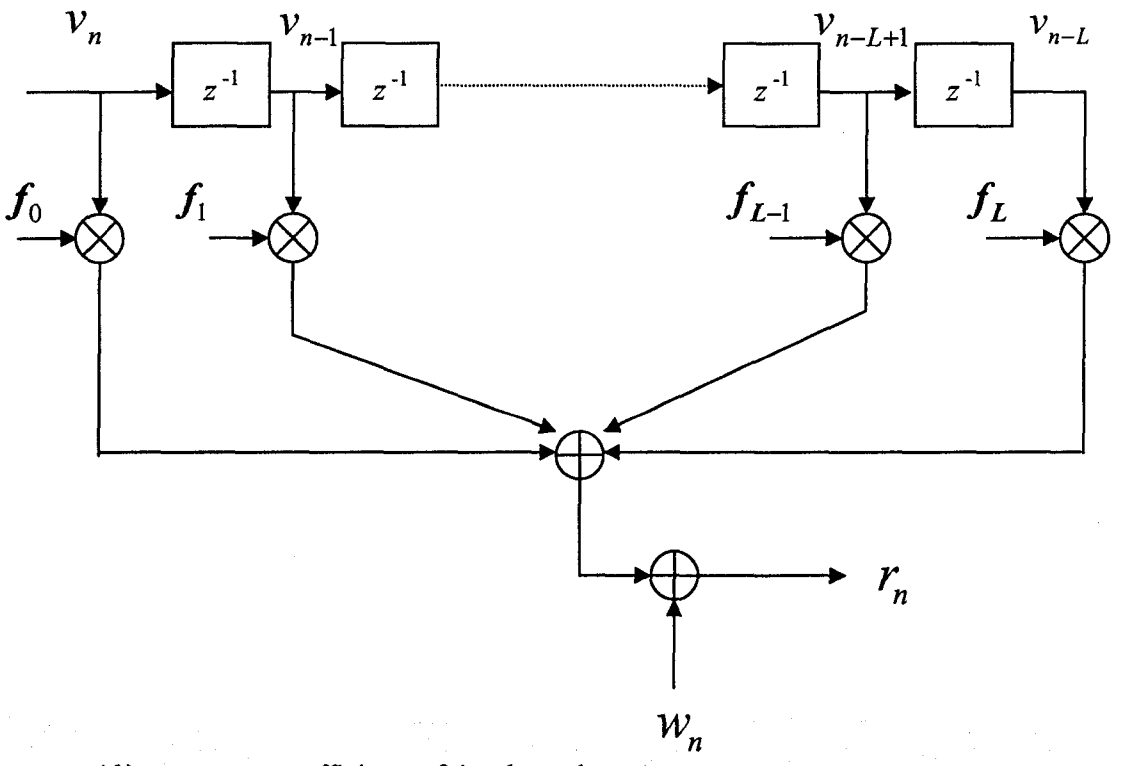
and the set of $\{f_k\}$ is the tap of the equivalent discrete-time transversal filter that is the cascade of the transmitting pulse shape filter $h_T(t)$, the channel $c(t)$, the matched filter $g^*(-t)$, the sampler, and the noise-whitening filter, $1/F^*(z^{-1})$. Figure 2-7 depicts the equivalent discrete-time channel model.

The noise sample, $w_n$, is a complex zero mean, stationary, white Gaussian noise sequence with variance $E[|w_n|^2] = N_o$, and is independent of $v_n$ where $N_o$ is the single-sided noise power spectral density.

In most communication systems, the channel coefficients are not known and vary over time; therefore, the receiver must first estimate the channel and then make adjustments to the equalizer coefficients with respect to the time-variant channel. However, for the stationary ISI channel, the

channel coefficients $\{f_l\}$ are assumed to be constant. In this thesis, the channels are assumed to be stationary ISI channels and the channel coefficients are also assumed to be known perfectly. Without loss of generality, we further assume that the channel neither amplifies nor attenuates the transmitted symbol, so

$$\sum_{l=0}^{L} |f_l|^2 = 1. \tag{2-5}$$




$\{f_l\}$ = tap coefficients of the channel
$z^{-1}$ = delay of a symbol duration $T_s$

Figure 2-7 Discrete-time model of ISI channel with AWGN

## 2-5 Equalization

To eliminate or suppress the ISI, an equalizer is required. Ideally, so as to eliminate the ISI effect totally, the equalizer has to concentrate the energy of the transmitted symbol, $v_n$, and reduce the energy from other transmitted symbols, e.g. ..,$v_{n-2}$,$v_{n-1}$, $v_{n+1}$, $v_{n+2}$,...,etc. Thus, an ideal equalizer produces a symbol, $\hat{v}_n$, which is the same as the transmitted symbol, $v_n$, plus the additive white Gaussian noise.

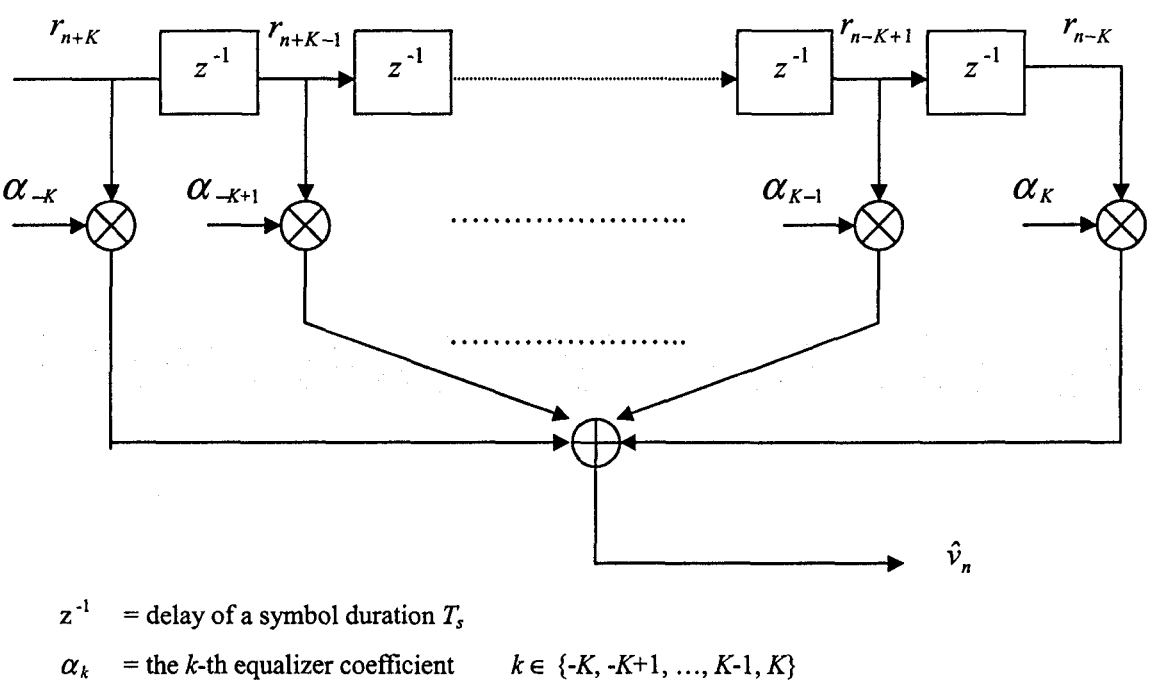
$$\hat{v}_n = v_n + w_n' \tag{2-6}$$

where $w_n'$ is additive white Gaussian noise with the same variance as $w_n$.

To compensate for the intersymbol interference, there are popular equalizers which were proposed in the past and are widely used, such as the *Linear Equalizer* (LE), the *Decision Feedback Equalizer* (DFE), *Minimum-Likelihood Sequence Estimation* (MLSE), and the *Maximum A Posteriori Probability* (MAP) equalizer. There is also an equalizer named the *Interference Canceller*. The interference canceller can eliminate the intersymbol interference perfectly if the transmitted symbols are known a priori [6, 7]. In this chapter, we are going to describe those techniques extensively.

## 2-5-1 Linear Equalizer

A linear equalizer is implemented using a linear transversal filter. It is an easily adjustable equalizing filter which is composed of a tapped delay line with $T_s$-second taps and equalizer coefficients. Figure 2-8 depicts a general form of a linear transversal filter with $2K+1$ taps.




$z^{-1}$ = delay of a symbol duration $T_s$

$\alpha_k$ = the $k$-th equalizer coefficient $k \in \{-K, -K+1, \ldots, K-1, K\}$

Figure 2-8 Linear equalizer with $2K+1$ coefficients

The current, past, and future samples of the received signal are linearly weighted by the equalizer coefficients and summed up to produce the estimated transmitted symbols. The $n$-th

estimated transmitted signal, $\hat{v}_n$, produced by the equalizer with $2K+1$ equalizer coefficients can be expressed as

$$\hat{v}_n = \sum_{k=-K}^{K} \alpha_k r_{n-k} \tag{2-7}$$

Although this filter is non-causal, causality can be achieved by introducing a delay of $K$ symbols. In the linear equalizer, the coefficients $\{\alpha_k\}$ have to be optimized so that the estimated symbol is as close to the transmitted symbol as possible.

The equalizer coefficients are selected to meet various optimization criteria. The most widely used criteria are the zero forcing (ZF) criterion and minimum mean square error (MMSE). The peak distortion criterion is optimal in the sense that it minimizes the peak distortion. If the equalizer length is considered to be infinite, the ISI components can be completely removed at the output of the equalizer. However, in practice, the length of the equalizer must be finite. As a result, there is some residual intersymbol interference that remains. Moreover, the equalizer has the disadvantage when the channel contains a spectral null in its frequency response. In this situation, the additive noise may be excessively enhanced at that frequency since the equalizer attempts to compensate the distortion at that frequency. Due to the limitations of peak distortion criterion, in what follows, we will not consider this criterion, and will instead focus on the MMSE

criterion.

In the MMSE criterion, the equalizer coefficients $\{\alpha_k\}$ are adjusted so that the expected value of the squared error between the actual equalizer output and the desired output is minimized. The error, $\varepsilon_n$, in this scope is defined as the difference between the $n$-th transmitted symbol and $n$-th estimated transmitted symbol. The error and the expected value are expressed in the following respectively

$$\varepsilon_n = v_n - \hat{v}_n \tag{2-8}$$

and

$$J = \mathrm{E}\,[\,|\varepsilon_n|^2\,], \tag{2-9}$$

which, for stationary ISI, does not depend on $n$.

In [11], it shows that $J$ can be minimized when the $2K+1$ equalizer coefficients are given by

$$\begin{bmatrix} \alpha_{-K} \\ \alpha_{-K+1} \\ \alpha_{-K+2} \\ \vdots \\ \alpha_{-2} \\ \alpha_{-1} \\ \alpha_{0} \\ \alpha_{1} \\ \vdots \\ \alpha_{K-1} \\ \alpha_{K} \end{bmatrix} = X^{-1} \cdot \begin{bmatrix} 0 \\ \vdots \\ 0 \\ f_L^* \\ \vdots \\ f_1^* \\ f_0^* \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \begin{matrix} \left.\begin{matrix} \\ \\ \end{matrix}\right\} K\text{-}L \text{ zeros} \\ \\ \\ \\ \\ \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} K \text{ zeros} \end{matrix} \tag{2-10}$$

where $\{f_l \mid l = 0, 1, \ldots, L\}$ are the known channel taps, * denotes complex conjugation, and the $(2K+1)$ by $(2K+1)$ matrix X is given by

$$X = \begin{bmatrix} x_0+N_o & x_{-1} & x_{-2} & \cdots & x_{-L+1} & x_{-L} & 0 & \cdots & 0 & 0 & 0 \\ x_1 & x_0+N_o & x_{-1} & x_{-2} & \cdots & x_{-L+1} & x_{-L} & 0 & \cdots & 0 & 0 \\ x_2 & \cdots & x_0+N_o & x_{-1} & \cdots & \cdots & x_{-L+1} & x_{-L} & 0 & \cdots & 0 \\ \vdots & & & \ddots & & & & & & & \vdots \\ x_{L-1} & & & & x_0+N_o & & & & & & 0 \\ x_L & & & & \cdots & x_0+N_o & \cdots & & & & x_{-L} \\ 0 & & & & & & x_0+N_o & & & & x_{-L+1} \\ \vdots & & & & & & & \ddots & & & \vdots \\ 0 & & & & & & & & x_0+N_o & & x_{-2} \\ 0 & & & & & & & & & x_0+N_o & x_{-1} \\ 0 & 0 & 0 & \cdots & 0 & x_L & x_{L-1} & \cdots & x_2 & x_1 & x_0+N_o \end{bmatrix} \tag{2-11}$$

with[1]

$$x_k = \sum_{l=0}^{L} f_l * f_{l-k} \qquad \forall\ k \in \{-L, \ldots, L\}, \tag{2-12}$$

and $N_o$ is the noise power spectral density. When the coefficients are calculated according to Eq. (2-10), the minimum mean square error [11] is

$$\begin{aligned} J_{\min} &= E\left(\left| v_n - \hat{v}_n \right|^2\right) \\ &= E\left(\varepsilon_n v_n^*\right) \\ &= 1 - \sum_{k=-K}^{0} c_k f_{-k} \end{aligned} \tag{2-13}$$

Linear equalization is a simple technique in terms of complexity; however, the main drawback is noise enhancement. The equalizer performs poorly in severe ISI channels, even at high $E_b/N_o$. An alternative equalization technique which can reduce the problem of noise enhancement is decision feedback equalization (DFE).

[1] Assume $f_l = 0 \quad \forall \quad l < 0, \quad l > L$.

## 2-5-2 Decision Feedback Equalizer (DFE)

A decision feedback equalizer is a non-linear equalizer that consists of two filters, a feedforward filter and a feedback filter as shown in Figure 2-9. The feedforward filter and the feedback filter are both transversal filters as described in section 2-5-1. The filters have $K_1$+1 and $K_2$ coefficients respectively. The idea of decision feedback equalization is to use previously detected symbols, $\{\tilde{v}_{n-1}, \tilde{v}_{n-2}, \tilde{v}_{n-3}, ..., \text{etc}\}$, made by the decision device to estimate the ISI produced by those symbols in the present symbol.

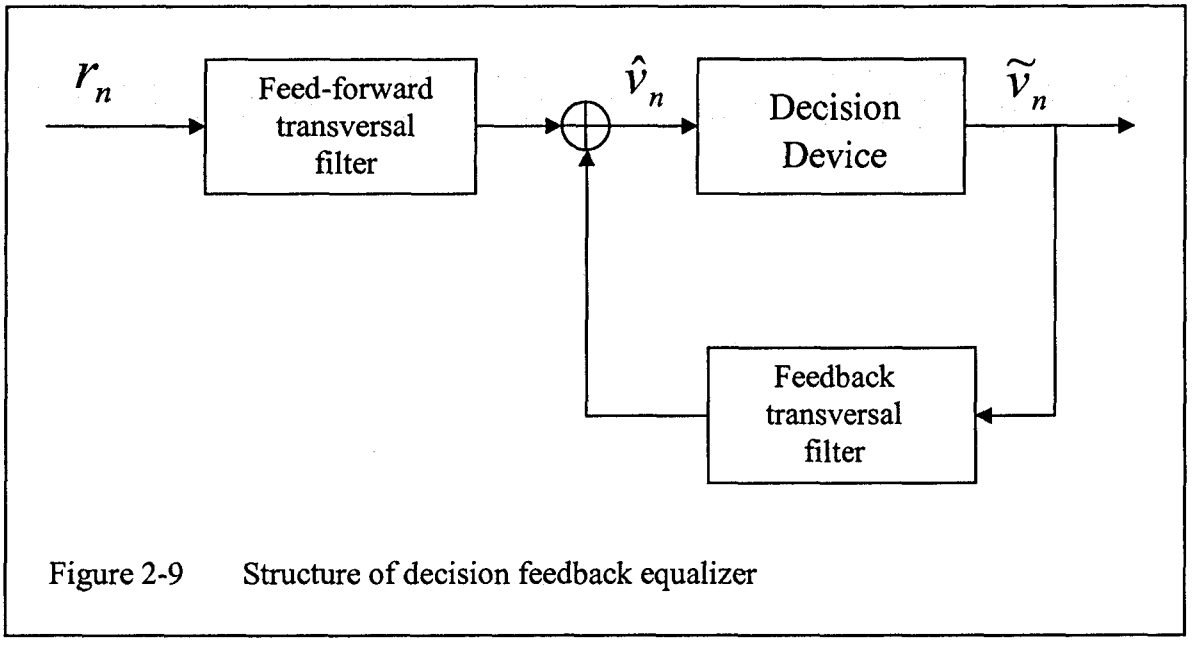



Figure 2-9 Structure of decision feedback equalizer

The output of the equalizer can be expressed as

$$\hat{v}_n = \underbrace{\sum_{k=-K_1}^{0} \alpha_k r_{n-k}}_{\text{Feedforward filter}} + \underbrace{\sum_{k=1}^{K_2} \alpha_k \tilde{v}_{n-k}}_{\text{Feedback filter}} \tag{2-14}$$

For the feedforward filter, the $K_1$+1 equalizer coefficients are calculated in the same way as the linear equalizer whereas the linear equalizer has $2K_1$+1 coefficients. Since the feedback filter is used to eliminate the ISI from the previously detected samples, the coefficients of the feedback filter can be expressed as [11]

$$\alpha_k = -\sum_{j=-K_1}^{0} \alpha_j \, f_{k-j}, \qquad k = 1, 2, \ldots, K_2 \tag{2-15}$$

where the coefficients $\{\alpha_{-K_1}, \alpha_{-K_1+1}, \ldots, \alpha_{-1}, \alpha_0\}$ are provided by the feedforward filter.

The minimum mean square error between $v_k$ and $\hat{v}_k$ is [11]

$$\begin{aligned} J_{\min} &= E[|v_k - \hat{v}_k|^2] \\ &= 1 - \sum_{j=-K_1}^{0} \alpha_j f_{-j}. \end{aligned} \tag{2-16}$$

### 2-5-3 Maximum Likelihood Sequence Estimation (MLSE)

The maximum likelihood sequence estimation (MLSE) equalizer is an optimal equalizer in the sense that its output is the sequence of code symbols that is most likely to have been transmitted. The idea of MLSE equalization is to characterize the channel as a finite state machine. The state at time index $n$ is given by the $L$ previous transmitted symbol and can be expressed as

$$S_n = (v_{n-L}, v_{n-L+1}, \ldots, \ldots, v_{n-2}, v_{n-1}). \tag{2-17}$$

If $M$-ary PSK modulation is used, the symbols can be expressed as

$$v_n \in \left\{ e^{j\frac{2\pi}{M}m} \,\middle|\, \mathrm{m} = 0,1,2,3,\ldots,M-2,M-1 \right\} \tag{2-18}$$

and the channel has $M^L$ states. The branch metric from state $s'_n = (v'_{n-L}, v'_{n-L+1}, \ldots, v'_{n-1})$ in response to input $v'_n$ at time index $n$ is given by

$$f(r_n \,|\, v'_{n-L}, v'_{n-L+1}, v'_{n-L+2}, \ldots, v'_{n-1}, v'_n) = \left| r_n - \sum_{l=0}^{L} f_l \, v'_{n-l} \right|^2. \tag{2-19}$$

Since the relationship between the received samples and the transmitted symbols is similar to the relationship between the output and input of a convolutional code, the Viterbi algorithm can be used to determine the most likely transmitted sequence.

### 2-5-4 MAP Equalization

As an alternative to the MLSE equalization using the Viterbi algorithm, symbol-by-symbol MAP equalization can be performed using the MAP algorithm [2]. The output of this equalizer is the *a posteriori probabilities* (APP)

$$\Pr\{v_n = v \,|\, \underline{r}\} \qquad \forall \quad v \in \left\{ e^{j\frac{2\pi}{M}m} \,\middle|\, m = 0,1,2,\ldots,M-1 \right\} \tag{2-20}$$

for each $n$, where $\underline{r} = (r_0, r_1, \ldots, r_{N-1})$ is the whole block of received samples. By selecting

$$\hat{v}_n = \arg\max_{v} \Pr\{v_n = v \,|\, \underline{r}\}, \tag{2-21}$$

the output of this equalizer is the symbol that was most likely to have been transmitted during each time index $n$. Whereas the MLSE equalizer minimizes the probability of a sequence error, the MAP equalizer minimizes the probability of a symbol error.

The MAP equalizer has two important advantages over the MLSE equalizers. Firstly, it does not require equal a-priori probabilities for the input symbols, and secondly, by omitting the decision device, the a posteriori probabilities can be used as soft-input into subsequent decoding stages. Both advantages are useful in turbo equalizer structures, as will be discussed in Chapter 3.

The drawback to the MAP equalizer is that it is more complex than the MLSE equalizers, which in turn is much more complex than the LE and DFE. The interference canceller (IC) described next, has lower complexity than all of these techniques.

## 2-5-5 Interference Canceller

It is claimed that the interference canceller can completely remove the ISI if the transmitted symbols are known a priori [6,7]. If we consider the $n$-th transmitted signal, $v_n$, over the ISI channel spanning $L$+1 spanning symbols, the $n$-th received sample can be expressed as

$$\begin{aligned} r_n &= \sum_{l=0}^{L} f_l v_{n-l} + w_n \\ &= f_0 v_n + f_1 v_{n-1} + f_2 v_{n-2} + \cdots + f_L v_{n-L} + w_n \end{aligned} \tag{2-22}$$

In Eq. (2-22), we notice that the $n$-th received signal contains some of the energy of $v_n$. Moreover, the $L$ future received samples, $\{ r_{n+1}, r_{n+2}, \ldots, r_{n+L-1}, r_{n+L} \}$, also contain some of the energy of $v_n$. If all received samples containing the energy of $v_n$ are weighted with the conjugate of channel coefficients and are summed up, the summed signal, $x_n$, is

$$x_n = \sum_{k=0}^{L} f_k^* r_{n+k} \tag{2-23}$$

$$\begin{aligned} &= f_0^* r_n + f_1^* r_{n+1} + f_2^* r_{n+2} + \cdots + f_{L-1}^* r_{n+L-1} + f_L^* r_{n+L} \\ &= |f_0|^2 v_n + f_0^* f_1 v_{n-1} + f_0^* f_2 v_{n-2} + \cdots + f_0^* f_{L-1} v_{n-L+1} + f_0^* f_L v_{n-L} + f_0^* w_n \\ &+ f_1^* f_0 v_{n+1} + |f_1|^2 v_n + f_1^* f_2 v_{n-1} + \cdots + f_1^* f_{L-1} v_{n+2-L} + f_1^* f_L v_{n+1-L} + f_1^* w_{n+1} \\ &\vdots \\ &+ f_L^* f_0 v_{n+L} + f_L^* f_1 v_{n+L-1} + f_L^* f_2 v_{n+L-2} + \cdots + f_L^* f_{L-1} v_{n+1} + |f_L|^2 v_n + f_L^* w_{n+L} \end{aligned}$$

Grouping the terms for each transmitted symbol in Eq (2-23) gives

$$x_n = v_n \sum_{l=0}^{L} |f_l|^2 + \sum_{k=1}^{L} q_k^* v_{n+k} + \sum_{k=1}^{L} q_k v_{n-k} + w_n' \tag{2-24}$$

where

$$q_k = \sum_{l=k}^{L} f_l f_{l-k}^* \tag{2-25}$$

and

$$w_n' = \sum_{k=0}^{L} f_k^* w_{n+k} \tag{2-26}$$

Without loss of generality, assume the channel neither attenuates nor amplifies the symbols such that the channel taps are normalized, i.e.

$$\sum_{l=0}^{L} |f_l|^2 = 1$$

Then,

$$x_n = v_n + \sum_{k=1}^{L} q_k^* v_{n+k} + \sum_{k=1}^{L} q_k v_{n-k} + w_n' \tag{2-27}$$

The term $w_n'$ is coloured Gaussian noise and the variance is

$$\begin{aligned} E[w_n' w'^*_n] &= \sum_{k_1=0}^{L} \sum_{k_2=0}^{L} f_{k_1}^* f_{k_2} E[w'_{n+k_1} w'^*_{n+k_2}] \\ &= \sum_{k=0}^{L} |f_k|^2 \, \delta(n) N_o \\ &= N_o \delta(n) \end{aligned}$$

From Eq. (2-27), we obviously see that the coefficient of the $v_n$ term is unity, and to recover $v_n$ from $x_n$, the other terms in Eq. (2-27) have to be eliminated. Theoretically, if we have the knowledge of the past and future samples of the transmitted signal, e.g. $\{v_m | m \neq n\}$, all interfering symbols, $\{v_{n-L}, \ldots, v_{n-1}, \ldots, v_{n+1}, \ldots, v_{n+L}\}$, can be eliminated.

Then, the $n$-th estimated symbol produced by the interference canceller is

$$\begin{aligned} \hat{v}_n &= x_n - \sum_{k=1}^{L} q_k^* v_{n+k} - \sum_{k=1}^{L} q_k v_{n-k} \\ &= v_n + w_n' \end{aligned} \tag{2-28}$$

The limitation of using the Interference Canceller is that the knowledge of past and future transmitted symbols is required. However, in practice, they are not known a priori, causing a problem with the use of the Interference Canceller. This problem can be solved by utilizing the "Turbo Equalization" technique which will be discussed in the next chapter. Moreover, the noise samples ($w_n'$) at the Interference Canceller output are correlated. This problem can be solved by placing an interleaver in between the encoder and the modulator, and a corresponding de-interleaver between the interference canceller and the decoder. Thus, the correlated noise appears at the decoder input as uncorrelated.

Table 2-1 shows the complexity of different equalization techniques where $M$ is the $M$-ary

signaling level, $L$ is the ISI length, and $K$ is total of taps (DFE and LE only). It shows the number of adders and multipliers (in real) required by different equalizers. Obviously, the MAP equalizer required more adders and multipliers than the other equalizers. The number of adders and multipliers grows exponentially with the ISI length and the M-ary signaling level. On the other hand, for DFE, LE, and IC, the complexity increases linearly.

| | Adders/ received sample (real) | Multipliers / received sample (real) |
|---|---|---|
| MAP Equalizer | $M^{L+1}(M^{L+1}+M-1)+2M$ | $M^{L+1}(3M^{L+1}+2M+3)+2M$ |
| Decision Feedback Equalizer | $4K+3M-2$ | $4K+2M$ |
| Linear Equalizer | $4K-2$ | $4K$ |
| Interference Canceller | $12L+2$ | $8L+4$ |

Table 2-1 Complexity of various equalization techniques

## 2-6 De-interleaver

When a random interleaver is utilized at the transmitting end, the sequence of transmitted symbols is shuffled according to the random permutation. Therefore, a de-interleaver is needed at the receiving end to re-shuffle the equalized samples to the proper order. Since the equalizer tends to produce error bursts, the de-interleaver can shuffle the received samples such that the error bursts appear as random independent errors at the output of the de-interleaver.

The output of the de-interleaver is

$$\hat{c}_n = \hat{v}_{\mathrm{IL}^{-1}[n]}$$

where $\mathrm{IL}^{-1}[\bullet]$ is defined such that $\mathrm{IL}^{-1}[\,\mathrm{IL}[n]\,] = n$ for all $n$.

When the symbol-based interleaving is used, $\hat{c}_n$ is sent to the decoder to find the branch metric which is required for the convolutional decoding. When the $(n_c, k_c)$ convolutional code is used, the branch metric, $\mu_n(c)$, is expressed as

$$\begin{aligned}\mu_n(c) &= f(\hat{c}_n | c_n = c) \\ &= \frac{1}{\pi N_o}\exp\left\{\frac{-1}{N_o}\left|\hat{c}_n - \mathrm{SM}[c]\right|^2\right\}\end{aligned} \tag{2-29}$$

where $c \in \{0,1,2,\ldots,M-1\}$ is the coded symbol.

However, when the binary-based interleaving is used, $\hat{c}_n$ cannot be used directly by the decoder to find the branch metrics since the coded symbols, $c_n$, are shuffled in bit-by-bit basis by

the binary-based interleaver at the transmitter. Hence, bit metrics are required so that they can be re-shuffled in bit-by-bit basis. Figure 2-10 shows a receiver with binary-based interleaving used.

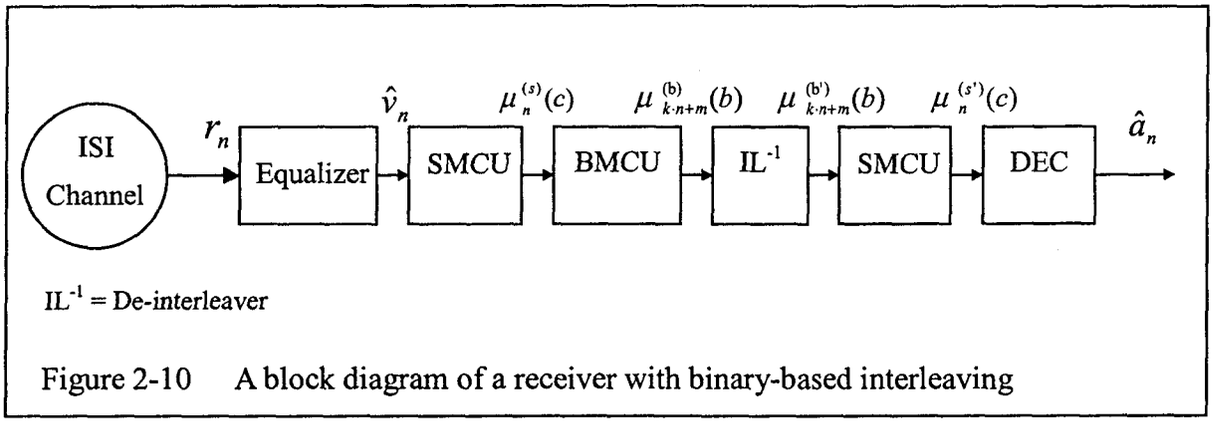



Figure 2-10 A block diagram of a receiver with binary-based interleaving

By the use of a metric calculation unit (MCU), the branch metrics, $\mu_n^{(s)}(c)$, can be found and then they are used for the bit metric calculation unit (BMCU) to produce the bit metrics, $\mu_{k\cdot n+m}^{(b)}(b)$. With the bit metrics, the binary-based de-interleaver can be performed which generates the re-shuffled bit metrics, $\mu_{k\cdot n+m}^{(b')}(b)$. Since the decoder can only make use of the branch metrics, the re-shuffled bit metrics are required to convert back to the branch metrics, $\mu_n^{(s')}(c)$.

The expressions for the bit metrics and the branch metrics are shown in the following.

$$\mu_n^{(s)}(c) = \exp\{\frac{-1}{N_o}\left|\hat{v}_n - SM[c]\right|^2\} \quad (2\text{-}30)$$

$$\mu_{k\cdot n+m}^{(b)}(b) = \sum_{c \mid \text{bit } m \text{ of } c=b} \mu_n^{(s)}(c) \quad (2\text{-}31)$$

$$\mu_{k\cdot n+m}^{(b')}(b) = \mu_{\text{IL}^{-1}[k\cdot n+m]}^{(b)}(b) \quad (2\text{-}32)$$

$$\mu_n^{(s')}(c) = \prod_{m=0}^{k-1} \mu_{k\cdot n+m}^{(b')}(\text{bit } m \text{ of } c) \quad (2\text{-}33)$$

where

$k$ = total number of bits per symbol

$c \in \{0, 1, 2, \ldots, 2^k - 1\}$

$b \in \{0, 1\}$

$m \in \{0, 1, 2, \ldots, k-1\}$

## 2-7 Decoding

Section 2-1 introduces a general concept in how the convolutional encoder operates. The next issue of coding scheme we need to consider is the decoding of a convolutional code. There are two popular decoding algorithms, Viterbi decoding and MAP decoding.

The Viterbi decoding algorithm performs maximum likelihood sequence decoding which was discovered by Viterbi in 1967. Both hard and soft input decoding can be implemented for a convolutional code; however, soft input decoding is generally superior by about 2 dB in an AWGN channel. Viterbi decoding is able to find the most likely message sequence to have been transmitted based on the received sequence ( $\underline{r}$ ). It minimizes the probability of a message sequence error by means of finding $\underline{a}$ which maximizes the likelihood function

$$f(\underline{r}|\underline{a}) \tag{2-34}$$

Similar to the Viterbi decoding, MAP decoding algorithm is also used to minimize the error probability. But this time, it minimizes the probability of a message bit error, such as

$$P_r\{ a_i^{(j)} | \underline{r}\} \qquad \text{where } a_i^{(j)} \in \{0,1\} \tag{2-35}$$

where $a_i^{(j)}$ is the $j$-th bit of $i$-th message symbol

Let us consider each data symbol $a_i$ with $k$ bits, i.e. $j \in \{0,1,2,\ldots, k\text{-}1\}$

If

$$P_r\{ a_i^{(j)}=0 \mid \underline{r}\} \geq P_r\{ a_i^{(j)}=1 \mid \underline{r}\}, \tag{2-36}$$

then the MAP decoder produces

$$\hat{a}_i^{(j)} = 0$$

as its output for the $j$-th bit of the $i$-th data symbol. Note that $P_r\{ a_i^{(j)} \mid \underline{r}\}$ is the a posteriori probability (APP).

# Chapter 3 Turbo Equalization

Now, let us pay attention to the turbo equalization technique which is our main focus in this thesis. The idea of turbo equalization is to jointly perform the equalization and the decoding through an iterative process in order to better remove the intersymbol interference. In this thesis, we consider six different turbo equalizer structures. These structures are described here.

## 3-1 Maximum A Posteriori Probability (MAP) Based Turbo Equalization

Maximum A Posteriori Probability (MAP) Based turbo equalizer [9] has proven that its performance can be near optimum. It is composed of a *soft-in soft-output* (SISO) MAP equalizer and a SISO MAP decoder. The extrinsic information generated by the MAP equalizer and the MAP decoder ($E_n^{(1)}$ and $E_n^{(0)}$) are passed back and forth in an iterative manner; in such a way, the overall performance is dramatically improved. Figure 3-1 shows the structure of the MAP-based turbo equalizer. Note that both extrinsic information, $E_n^{(1)}$ and $E_n^{(0)}$, are modified for each iteration.

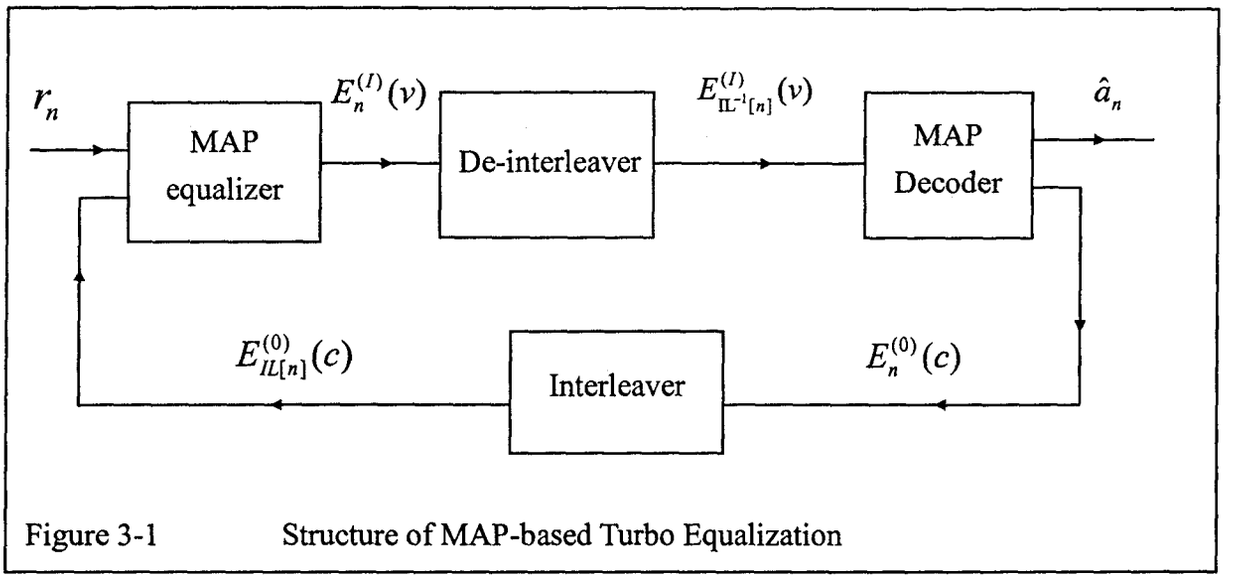



Figure 3-1 Structure of MAP-based Turbo Equalization

The extrinsic information ($E_n^{(I)}$) at the output of the MAP equalizer is defined as

$$E_n^{(I)}(v) = \frac{\Pr\{v_n = v \mid \underline{r}\}}{\Pr\{v_n = v\}}$$

where $\Pr\{v_n = v \mid \underline{r}\}$ is the a posteriori probability (APP) that the $n$-th transmitted symbol is $v$, given that samples $\underline{r}$ are received, and $\Pr\{v_n = v\}$ is a priori probability that the $n$-th transmitted symbol is $v$. The de-interleaver output, $E_{IL^{-1}[n]}^{(I)}(v)$, is provided to the MAP decoder, where it is used as the branch metric $f(r_n \mid c_n = v)$.

At the decoder output, the extrinsic information ($E_n^{(0)}$) is defined as

$$E_n^{(0)}(c) = \frac{P_r\{c_n = c \mid \underline{r}\}}{f(r_n \mid c_n = c)}$$

and $E_{IL[n]}^{(0)}(c)$ is used by the MAP equalizer as the probability of the transmitted symbols, $\Pr\{v_n = c\}$. Note that M values of $E_n^{(I)}$(v) and $E_n^{(0)}$(c) are calculated at time index $n$

since both $c$ and $v$ have $M$ possibilities. A detailed description of the MAP algorithm is given in Appendix A.

In this thesis, we are going to use the *Maximum A Posteriori Probability* (MAP) based turbo equalizer as our optimum case, since with the structure composed of MAP equalizer and MAP decoder (shown in Figure 3-1), the output at the decoder after a number of iterations should be near optimum. However, computational complexity is a shortcoming for using the MAP-based turbo equalizer. For both MAP equalizer and MAP decoder, the number of computation grows exponentially with the ISI length and the encoder memory respectively. An alternative solution which reduces the ISI and keeps the computation low is the interference canceller (IC) based turbo equalizer.

## 3-2 Hard-decision Feedback Interference Cancellation (IC) Based Turbo Equalization

A hard-decision feedback IC-based turbo equalizer consists of a DFE or LE, a de-interleaver, a Viterbi decoder, a convolutional encoder, a signal mapper, an interleaver, and an interference canceller. The process can be done in an iterative fashion. For the first iteration, the DFE or LE is used. From the second and following iterations, the interference canceller is used so that the computational complexity can be reduced significantly comparing to the MAP equalizer. Figure 3-2 depicts the structure of the hard-decision feedback IC-based turbo equalizer.

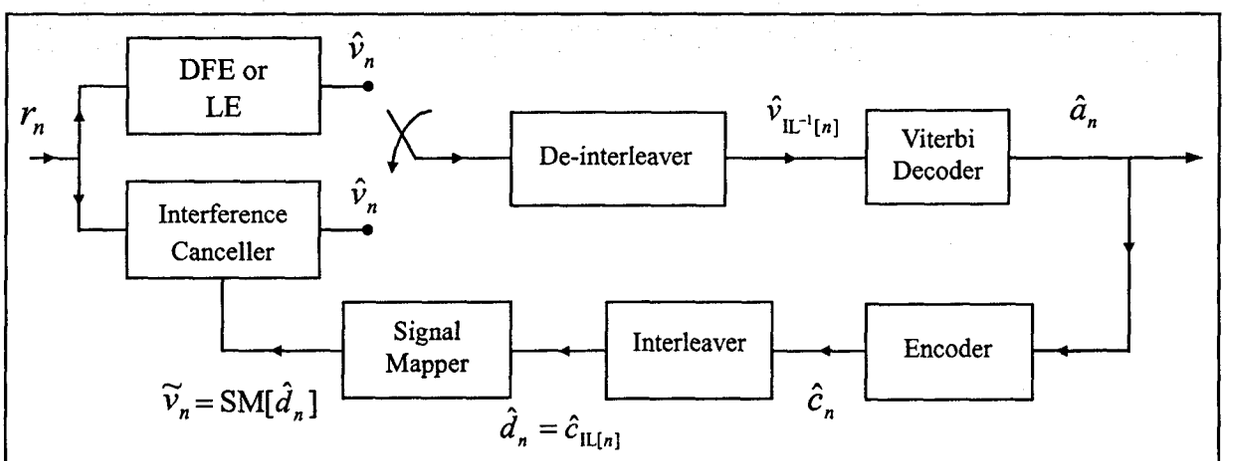



Figure 3-2 Structure of the hard-decision feedback IC-based turbo equalizer

A block of received samples, $r_n$, is sent to an equalizer, either a linear equalizer (LE) or decision feedback equalizer (DFE), then the estimated transmitted samples, $\hat{v}_n$, will be produced and enter the de-interleaver and decoder to estimate the transmitted message, $\hat{a}_n$. Up to this

point, the signal processing is just the same as the conventional process with an equalizer and a decoder. However, the major difference from the conventional process is that the estimated message are fed back to the encoder again and then to the modulator such that the estimated transmitted samples, $\tilde{v}_n$, can be produced. Based on this estimated transmitted samples, the interference canceller can be used and its outputs, $\hat{v}_n$, are treated as the estimated transmitted samples and sent to the de-interleaver. Eventually the re-estimated messages will be generated by the decoder. This process can be performed iteratively by feeding back the re-estimated messages to the encoder and repeat the entire process again. Therefore, the estimated messages are re-calculated again. We note that the overall system performance becomes better for every iteration. Since the Viterbi decoder generates the hard-decision outputs, $\hat{a}_n$, which are fed back to the encoder, this process is named as "Hard-decision Feedback IC-Based Turbo Equalization".

As seen in Eq. (2-26), the noise is not only the additive white Gaussian noise. It is also convoluted with the channel coefficients to form coloured noise. Therefore, the noise sequence is correlated at the decoder input which reduces the performance of the decoding. In practice, the noise correlation is limited to adjacent samples. Hence, the random interleaving is essential such that the noise appears as white noise at the input of the decoder at the expense of time delay.

The implementation of the hard-decision feedback IC-based turbo equalization technique can be summarized as followed:

1. Equalize the received samples using a decision feedback equalizer or linear equalizer
2. Decode the convolutional code, yielding $\hat{a}_n$
3. Re-encode, then interleave and symbol map the result, giving $\tilde{v}_n$
4. Cancel the interference in $r_n$ with an interference canceller using $\tilde{v}_n$ in place of $v_n$

$$\hat{v}_n = \sum_{l=0}^{L} f_l^* r_{n+l} - \sum_{k=1}^{L} q_k^* \tilde{v}_{n+k} - \sum_{k=1}^{L} q_k \tilde{v}_{n-k}$$

5. Use $\hat{v}_n$ to re-decode the convolutional code
6. Repeat from step 3 for more iteration

## 3-3 Soft-decision Feedback IC-Based Turbo Equalization

On the hard-decision feedback IC-based turbo equalizer, the input to the decoder are typically soft-input and the output from the decoding process of the decoder is hard-output, e.g. Viterbi decoder. The hard-output is then sent to the encoder again in iterative fashion. Because of the hard-output decoder, the degradation of the system performance is occurred. Therefore, by using the SISO decoder, the overall performance can be improved. Such a decoder can be employed in the IC-based turbo equalization jointly with the equalizer in order to provide soft-decision feedback for the subsequent iterations. This process can be named as the "Soft-decision Feedback IC-based Turbo Equalization" [10].

To generate the soft-decision output from the decoder, the a posteriori probability (MAP) algorithm can be utilized as mentioned in the Section 2-7. The MAP decoder computes the probability of all possible code symbols given the received samples. The probability of a code symbol, $c$, can be expressed as

$$\Pr\{c_n = c \,|\, \underline{r}\} \qquad c \in \{0, 1, 2, \ldots, M-1\} \tag{3-1}$$

The algorithm uses the branch metrics

$$\mu_n^{(s)}(c) = \frac{1}{\pi N_o} \exp\{\frac{-1}{N_o}|\hat{v}_n - \mathrm{SM}[c]|^2\} \tag{3-2}$$

in its calculation.

If we find all probabilities of each possible code symbol, then the estimated transmitted samples can be obtained. The $n$-th estimated transmitted samples, $\tilde{v}_n$, generated by the signal mapper can be expressed as

$$\tilde{v}_n = \sum_{d=0}^{M-1} \mathrm{SM}[d] \cdot \Pr\{d_n = d \,|\, \underline{r}\} \tag{3-3}$$

where

$$\Pr\{d_n = c \,|\, \underline{r}\} = \Pr\{c_{\mathrm{IL}[n]} = c \,|\, \underline{r}\} \qquad \forall\ c \in \{0, 1, 2, \ldots, M-1\}$$

Note that $\tilde{v}_n$ is not found based on one code symbol only. It is calculated based on the signal mapping of all possible code symbols, SM[$d$], weighted with its probability. The $\tilde{v}_n$ can be considered as an averaged value of all possible transmitted symbols, $v_n$. It will be sent to the interference canceller.

By utilizing the soft-decision feedback IC-based turbo equalization, the overall system performance can be enhanced. Figure 3-3 shows the structure of the soft-decision feedback IC-based equalizer. Unlike the hard-decision feedback IC-based turbo equalizer, the

soft-decision feedback IC-based turbo equalizer does not require the convolutional encoder and since the probabilities of the code symbols, $\Pr\{c|\underline{r}\}$, can be directly found by the MAP decoder.

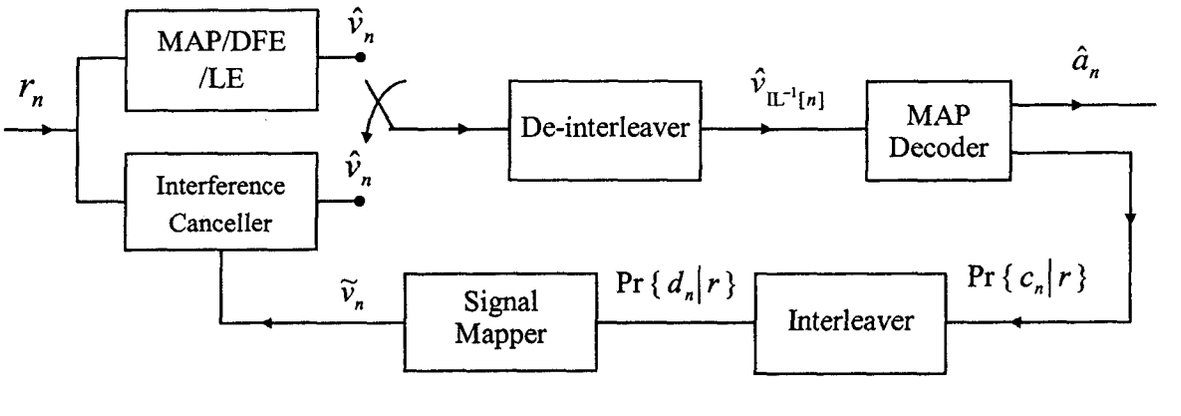



Figure 3-3 Structure of the soft-decision feedback IC-based turbo equalizer

For the first iteration, a sub-optimal equalizer, i.e. a linear equalizer or decision feedback equalizer, can be used for estimating the transmitted symbols which are required for the interference canceller after the first iteration. Since a sub-optimal equalizer still suffers loss in performance, there is a better approach which replaces the sub-optimal equalizer by the optimal equalizer, i.e. MAP equalizer. However, the computational complexity grows exponentially with the ISI length, $L$.

## 3-4 Turbo Equalizations with binary-based interleaving

With binary-based interleaving, the hard-decision feedback IC-based turbo equalizer [13] requires additional components which are two symbol metric calculation units (SMCU) and a bit metric calculation unit (BMCU). Those components are used to find the branch metrics from the equalized received samples, and then convert the branch metrics to the bit metrics. Finally, the SMCU will be used to find the branch metrics from the re-shuffled bit metrics. Figure 3-4 depicts the structure of hard-decision feedback IC-based turbo equalizer with binary-based interleaving.

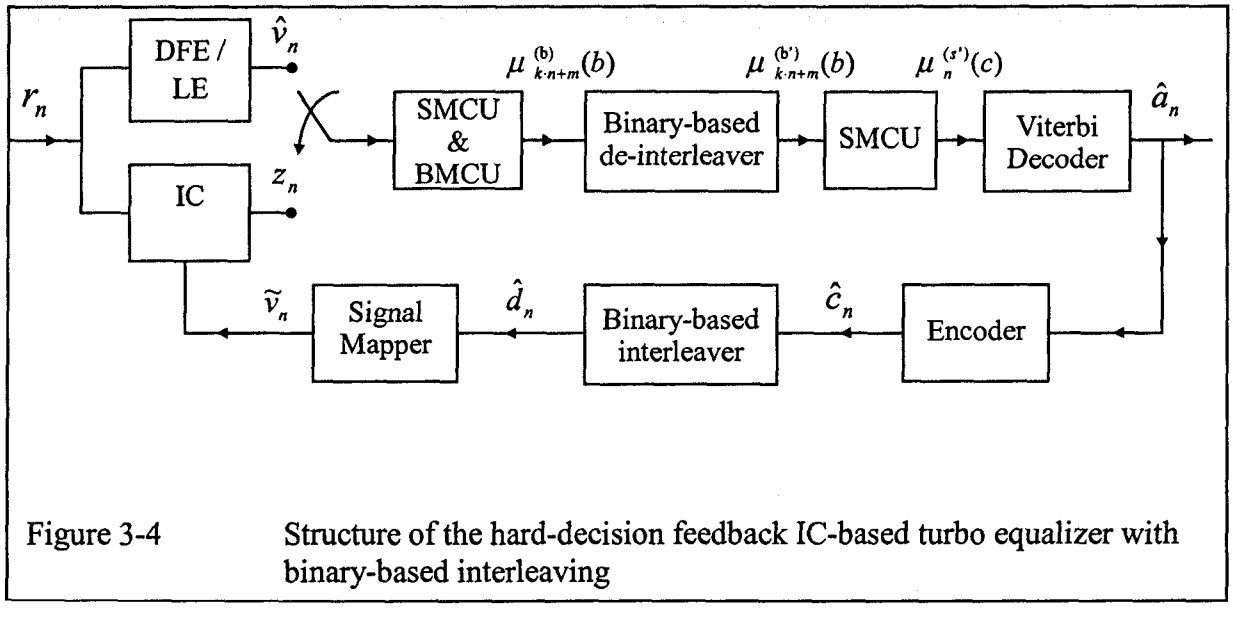



Figure 3-4 Structure of the hard-decision feedback IC-based turbo equalizer with binary-based interleaving

For the soft-decision feedback IC-based turbo equalizer, more components must be added comparing to the hard-decision feedback IC-based turbo equalizer. Since the MAP decoder can provide the probability of the coded symbols, $\Pr\{c_n | r\}$, and the binary-based interleaver is only allowed to perform on a bit-by-bit basis, a bit probability calculation unit is needed to convert $\Pr\{c_n | r\}$ to the probability of the coded bits, $\Pr\{c_n^{(b)} | r\}$, which is expressed as

$$\Pr\{c_{k\cdot n+m}^{(b)}(b)\} = \sum_{c \,|\, \text{bit } k \text{ of } c=b} \Pr\{c_n = c \,|\, r\}$$

where

$k$ = total number of bits per symbol

$m \in \{0, 1, 2, \ldots, k-1\}$,

$c \in \{0, 1, 2, \ldots, 2^k - 1\}$, and

$b \in \{0, 1\}$

Then, $\Pr\{c_n^{(b)} | r\}$ is sent to the binary-based interleaver to generate

$$\Pr\{d_{k\cdot n+m}^{(b)}(b)\} = \Pr\{c_{\mathrm{IL}[k\cdot n+m]}^{(b)}(b)\}.$$

Since the signal mapper can only process with the symbol probability, $\Pr\{d_{k\cdot n+m}^{(b)}(b)\}$ must be converted to the symbol probability by the use of symbol probability calculation unit (SPCU), which produces

$$\Pr\{d_n = d \mid r\} = \prod_{m=0}^{k-1} \Pr\{d_{k\cdot n+m}^{(b)} (\text{bit } k \text{ of } d)\}.$$

Figure 3-5 describes a structure of the soft-decision feedback IC-based turbo equalizer with binary-based interleaving.

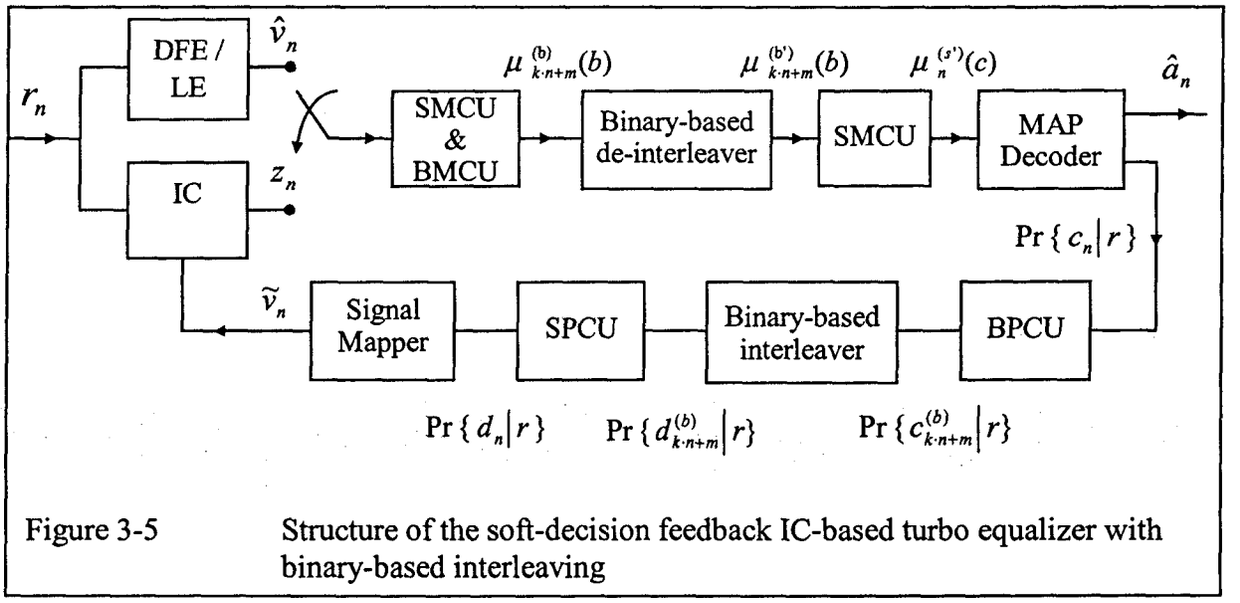



Figure 3-5 Structure of the soft-decision feedback IC-based turbo equalizer with binary-based interleaving

# Chapter 4 Simulation Results

In this chapter, we present simulation results that illustrate the performance of the various turbo equalizers. The performance of all the turbo equalization techniques mentioned in the Chapter 3 will be evaluated for both 4-PSK and 8-PSK modulation schemes, with different coding schemes over several intersymbol interference (ISI) channel models.

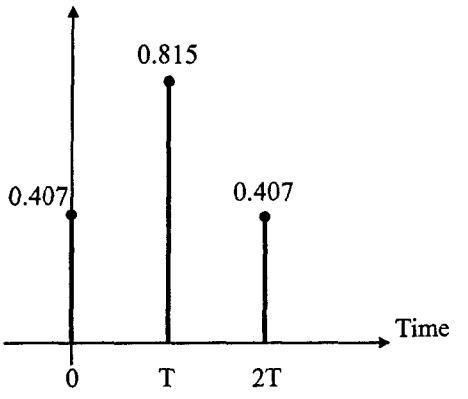
## 4-1 Time-invariant AWGN Channel with ISI

In a stationary ISI channel with AWGN, a transmitted signal at the receiver end will be smeared over another $L$ transmitted signals depending on the channel characteristics. Figure 4-1 shows five different discrete-time equivalent channel models which are used in [9], [12], and [13]. In our thesis, these channels will be considered. Note that all of the channels are normalized, i.e.

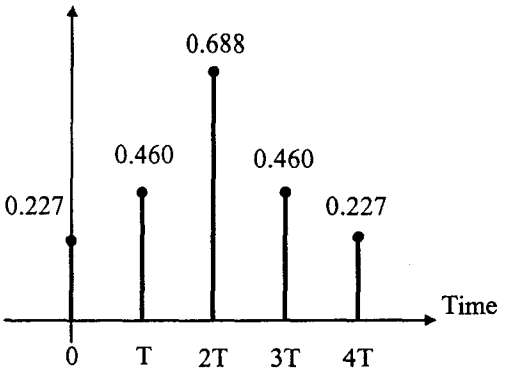
$$\sum_{l=0}^{L} |f_l|^2 = 1$$

where $f_l$ is the $l$-th channel tap coefficient.

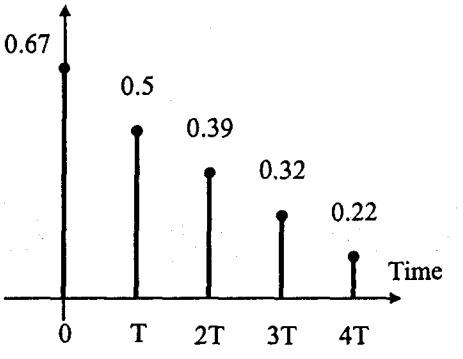
Figure 4-2 shows the spectral characteristics for those five channel models. Channel 2 is shown to have the worst spectral characteristic whereas Channel 3 and 5 have better spectral characteristic and do not have spectral nulls. Therefore, the performance of the LE over Channel 3, and 5 are better than over Channel 1, 2, and 4.



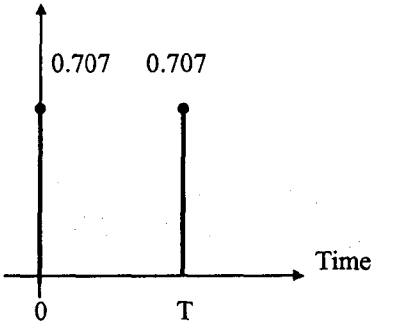

(a) Channel 1; L=2



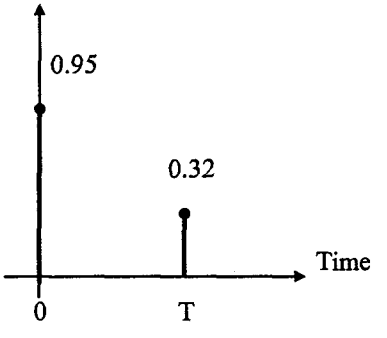

(b) Channel 2; L=4




(c) Channel 3; L=4

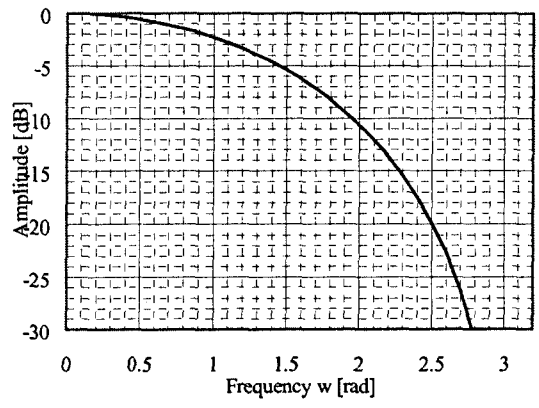



(d) Channel 4; L=1

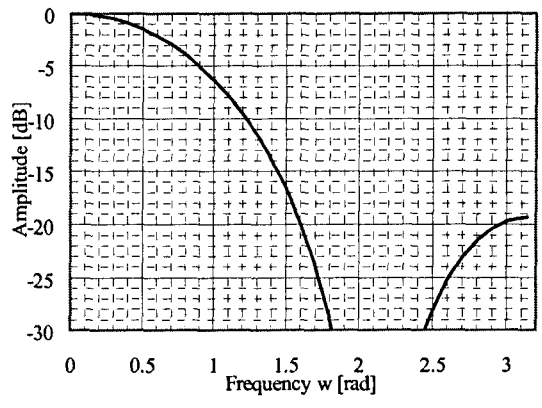



(e) Channel 5; L=1

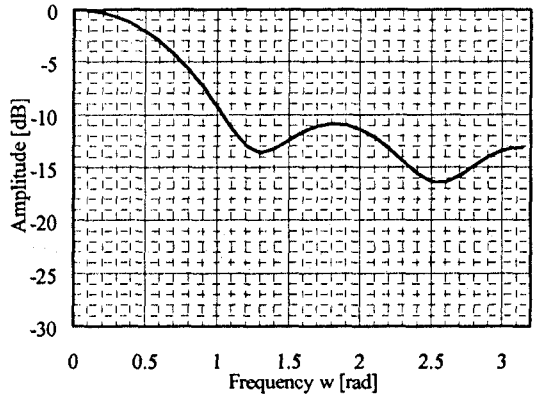
Figure 4-1 Channel Models



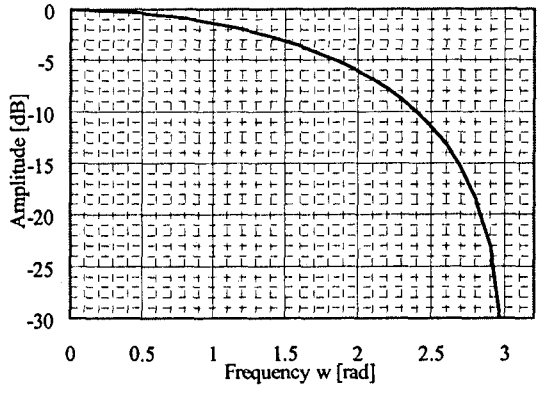

(a) Channel 1; L=2



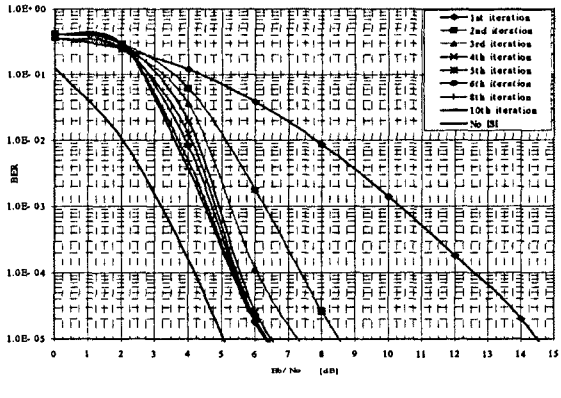

(b) Channel 2; L=4




(c) Channel 3; L=4




(d) Channel 4; L=1



(e) Channel 5; L=1

Figure 4-2 Amplitude spectra for the channel models shown in Figure 4-1

## 4-2 System Description

For all simulations, the number of message bits for each block (or block size) is implicitly chosen to be 1000 bits. If binary-based interleaving is employed, the size of the interleaver is fixed to $1000/R_c$, where $R_c = k_c/n_c$ is the rate of the convolutional code. The size of the interleaver is $1000/k_c$ if symbol-based interleaving is used. Because of the simplicity, random interleaving will be employed in our simulations. For the convolutional encoders, extra $K_c$-1 symbols must be encoded after the block has been encoded such that the encoder state can be set to zero.

Moreover, for LE-MMSE, the number of equalizer coefficients is chosen to be 31. For the DFE-MMSE, the feed-forward filter has $31-L$ equalizer coefficients while the feedback filter has $L$ coefficients such that the total number of useful equalizer coefficients in the DFE-MMSE is the same as the LE-MMSE, where $L$ is the length of ISI.

In this chapter, we will present the performance of six different turbo equalizers as follow:

System 1: Soft-decision feedback MAP-based turbo equalizer (MAP-based).

System 2: Soft-decision feedback IC-based turbo equalizer with MAP equalizer used at the first iteration (MAP/IC-soft).

System 3: Soft-decision feedback IC-based turbo equalizer with DFE used at the first iteration (DFE/IC-soft).

System 4: Soft-decision feedback IC-based turbo equalizer with LE used at the first iteration (LE/IC-soft).

System 5: Hard-decision feedback IC-based turbo equalizer with DFE used at the first iteration (DFE/IC-hard).

System 6: Hard-decision feedback IC-based turbo equalizer with LE used at the first iteration (LE/IC-hard).

## 4-3 Performance of the Turbo Equalizers with Coded QPSK

In this section, the performance of turbo equalizers in a coded QPSK system will be revealed. For the coded QPSK system with symbol-based random interleaver, the 16-state rate-1/2 convolutional encoder with generator $(27,35)_8$ is the only coding scheme that we consider.

In a coded system, the error performance over an AWGN channel with no ISI is generally realized as the ideal performance for any coded communication system in the presence of ISI. Indeed, the goal of the turbo equalizers is to reach the performance in an AWGN ISI-free channel. In our simulation examples, this performance will also be displayed in order to compare with the performance of the turbo equalizers.

### 4-3-1 Performance of the Turbo Equalizers over Channel 1

Figure 4-3 shows the performance of the MAP-based turbo equalizer over Channel 1. This figure displays the performance from the first to the tenth iterations. The performance after the first iteration corresponds to an optimal disjoint receiver, based on a MAP equalizer and MAP decoder, without any turbo processing. As can be seen, this optimal disjoint receiver suffers from a 3.2 dB ISI penalty at a BER of $10^{-5}$ (as compared to an system using the same code and modulation scheme in an ISI-free environment). This penalty can be substantially reduced by using turbo processing. In fact, after just three iterations the penalty drops to 0.1 dB, corresponding to turbo processing gain of 3.1 dB. Clearly the MAP-based turbo equalizer is very effective at suppressing the ISI. Unfortunately, it is also complex.

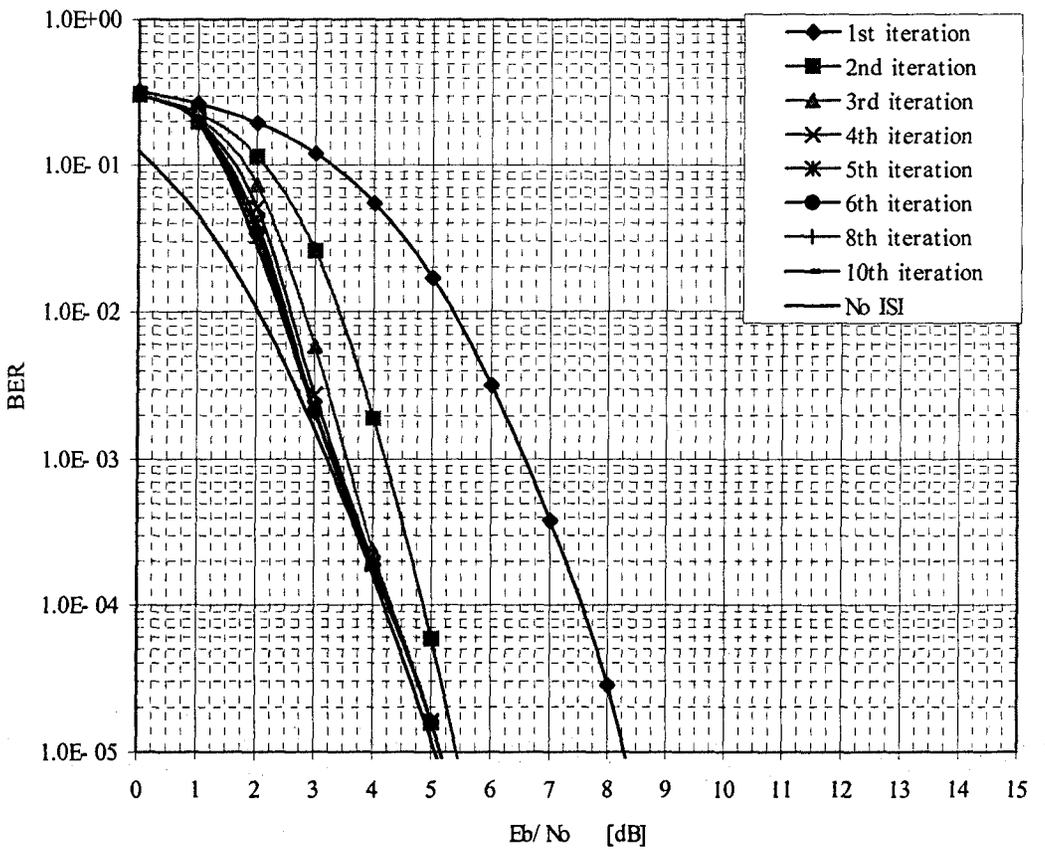



Figure 4-3 QSPK - MAP-based turbo equalizer over Channel 1

To reduce the complexity, an interference canceller can be used in place of the MAP equalizer for the second and the following iterations[1]. Figure 4-4 shows the performance of the MAP/IC-soft over Channel 1. Compared to the coded QPSK system with no ISI, the performance loss of the MAP/IC-soft at the fifth iteration is only 0.3 dB at the bit error rate of $10^{-5}$. In other words, the penalty of 0.2 dB is suffered comparing to the performance of the MAP-based turbo equalizer; however, the computational complexity has been reduced for the second and following iterations.

In fact, the computational complexity of the MAP equalizer grows exponentially with the ISI length. If a channel model has a long ISI length, the MAP equalizer may not be possible to implement even if it is only used for the first iteration. Therefore, sub-optimal equalizers could be employed such as DFE and LE. Figure 4-5 and 4-6 show the performance of the DFE/IC-soft and the LE/IC-soft over Channel 1, respectively. With the DFE or LE used at the first iteration, the complexity can be reduced significantly but the performance can still be close to the performance of the coded QPSK system with no ISI. From both figures, we observe that the performance of the soft-decision feedback IC-based turbo equalizers is superior over the conventional equalizers. For the DFE/IC-soft, it provides a 6.6 dB gain after five iterations at the BER of $10^{-5}$ and only a 0.63 dB ISI penalty. For the LE/IC-soft, the performance gain is much greater. A turbo processing gain of 9.25 dB is obtained resulting on an ISI penalty of only 0.2 dB.

[1] Note that the MAP equalizer is still used for the first iteration, since the IC cannot be used until after the convolution code has been decoded.

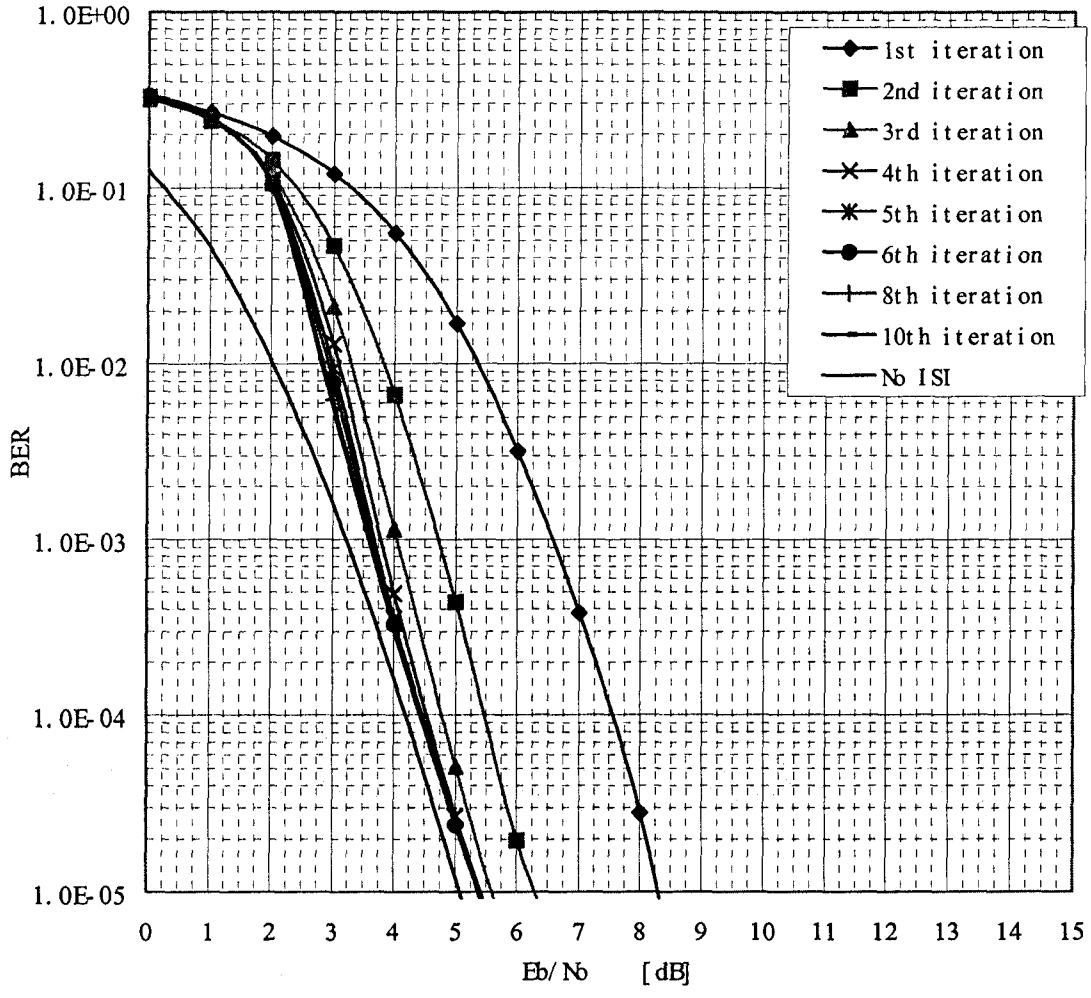



Figure 4-4 QPSK - MAP/IC-soft over Channel 1

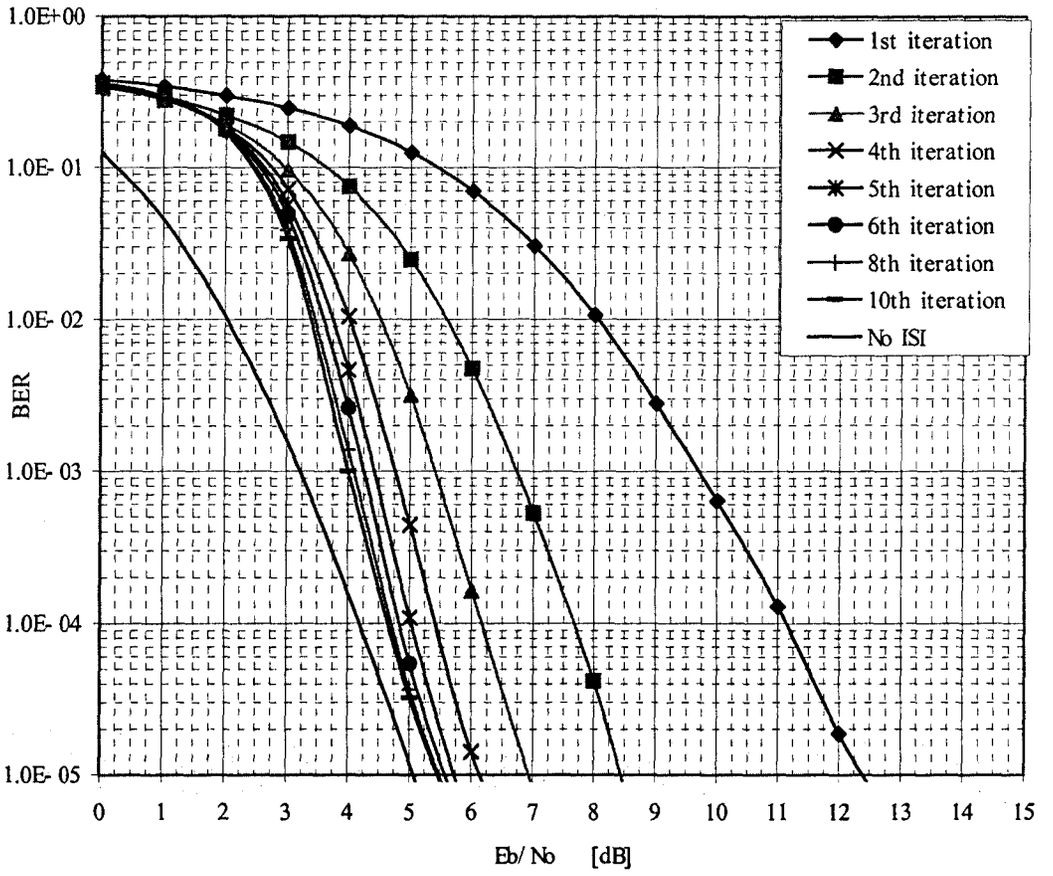



Figure 4-5 QPSK - DFE/IC-soft over Channel 1

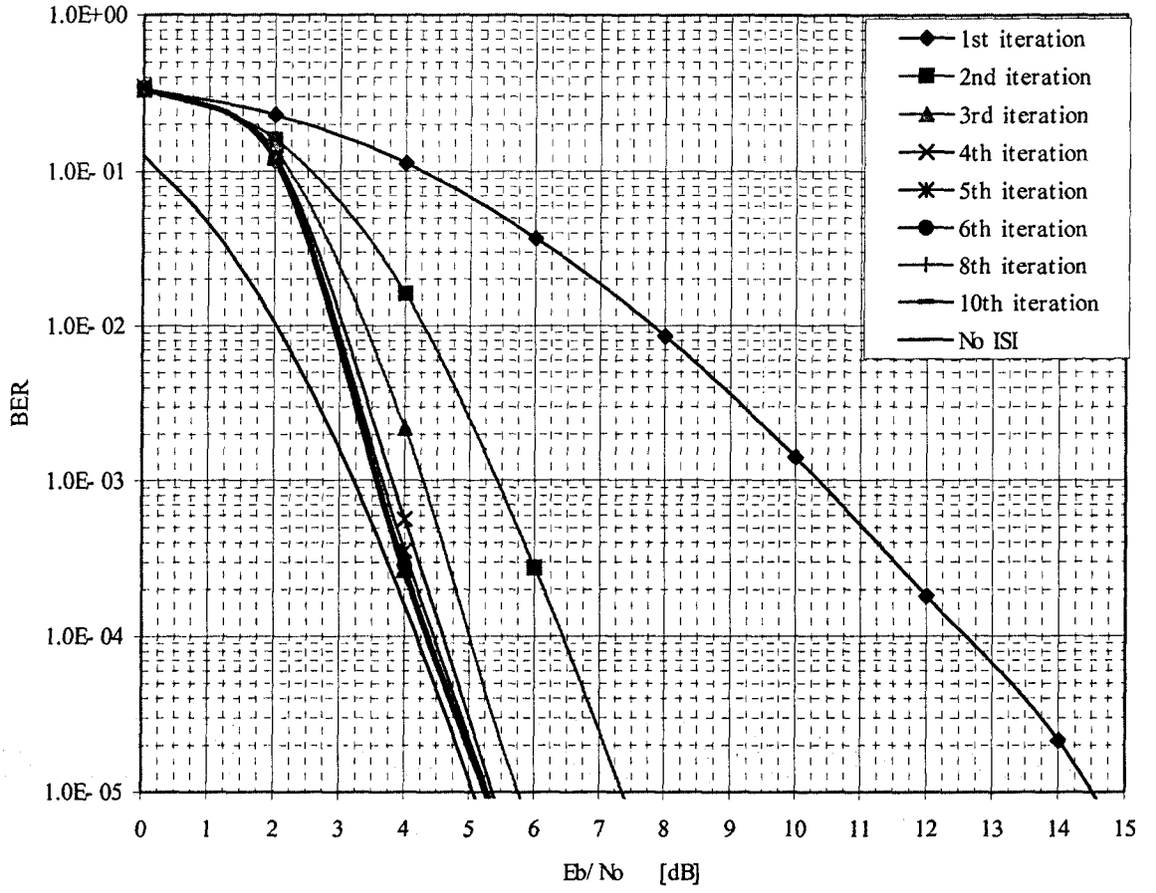



Figure 4-6 QPSK - LE/IC-soft over Channel 1

We observe that the more iteration taken, the smaller the performance gain relative to the previous iteration. For example, the performance gain between the first and the second iterations is higher than between the fifth and the sixth iterations. From Figure 4-6, the performance gain between the first and the second iteration is 7.2 dB; however, the performance gain between the fifth and the sixth iterations is less than 0.1 dB.

It is surprising that this LE-based turbo equalizer is better at suppressing the ISI than the DFE-based one (as shown in Figure 4-7), since in general the DFE is more effective than the LE in a stand-alone (non-iterative) environment. An explanation for this peculiar behaviour stems from the fact that the performance of the turbo equalizer highly depends on the performance of the equalizer during the first iteration. Although neither the LE nor the DFE are particularly effective at low $E_b/N_o$, the LE provides a slightly lower BER, which translates into larger improvement after turbo processing. As an example, at $E_b/N_o = 4$ dB after the first iteration, the DFE-based system gives a BER of 0.19 and the LE gives a BER of 0.11. This slight difference is amplified by the turbo processing. After four iterations, the LE/IC-soft system is able to reduce the BER to $6\times10^{-4}$ whereas the DFE/IC-soft system could only achieve a BER of 0.01.

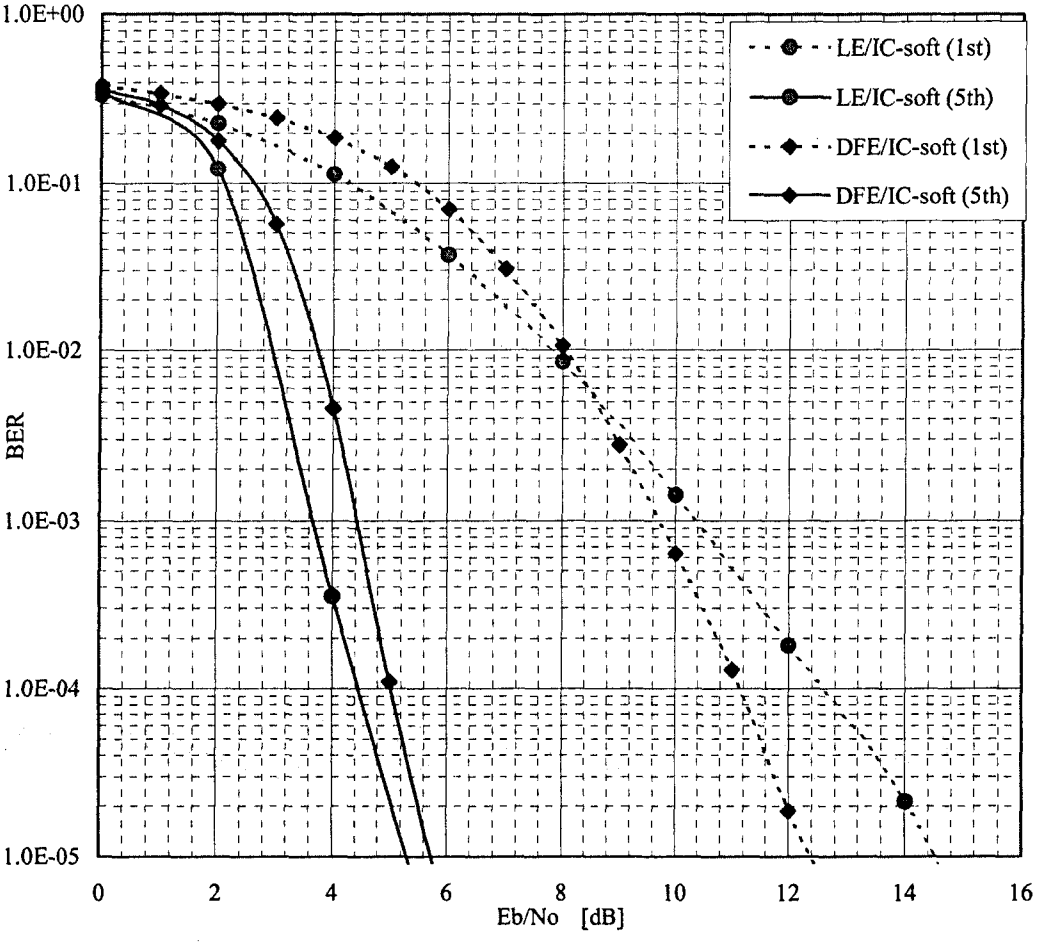



Figure 4-7 QPSK – Performance comparison of DFE/IC-soft and LE/IC-soft over Channel 1

By replacing the complex MAP equalizer with simpler LE or DFE, the overall system complexity can be reduced without significantly degrading performance. However, the complexity of the MAP decoder still remains an issue. If the computational complexity is a critical consideration, a Viterbi decoder could be used since it is less complex than the MAP decoder. Figure 4-8 and 4-9 show the performance of LE/IC-hard and DFE/IC-hard over Channel 1 respectively. Although these systems suffer a degradation compared to the soft-decision feedback IC-based turbo equalizers, a significant turbo processing gain can still be achieved. For the LE/IC-hard, a gain of 8.5 dB is achieved at the BER of $10^{-5}$, leaving a 1.4 dB ISI penalty. At the expense of 1.1 dB relative to the LE/IC-soft, LE/IC-hard can significantly reduce the computational complexity. For all the systems described in Figure 4-3 to 4-9, gains are negligible after the fifth iteration. In the following, we therefore restrict our analysis on the performance after the fifth iteration.

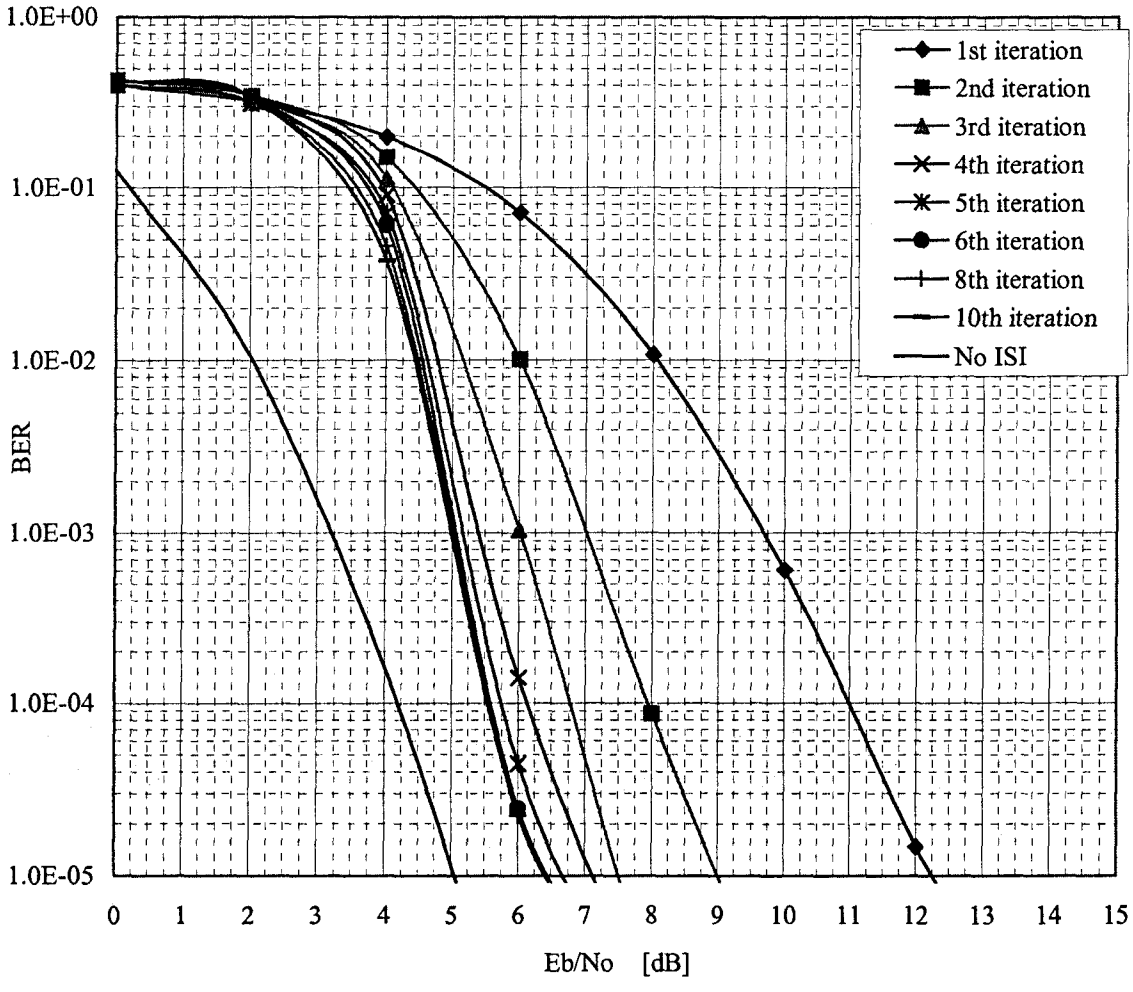



Figure 4-8 QPSK - DFE/IC-hard over Channel 1

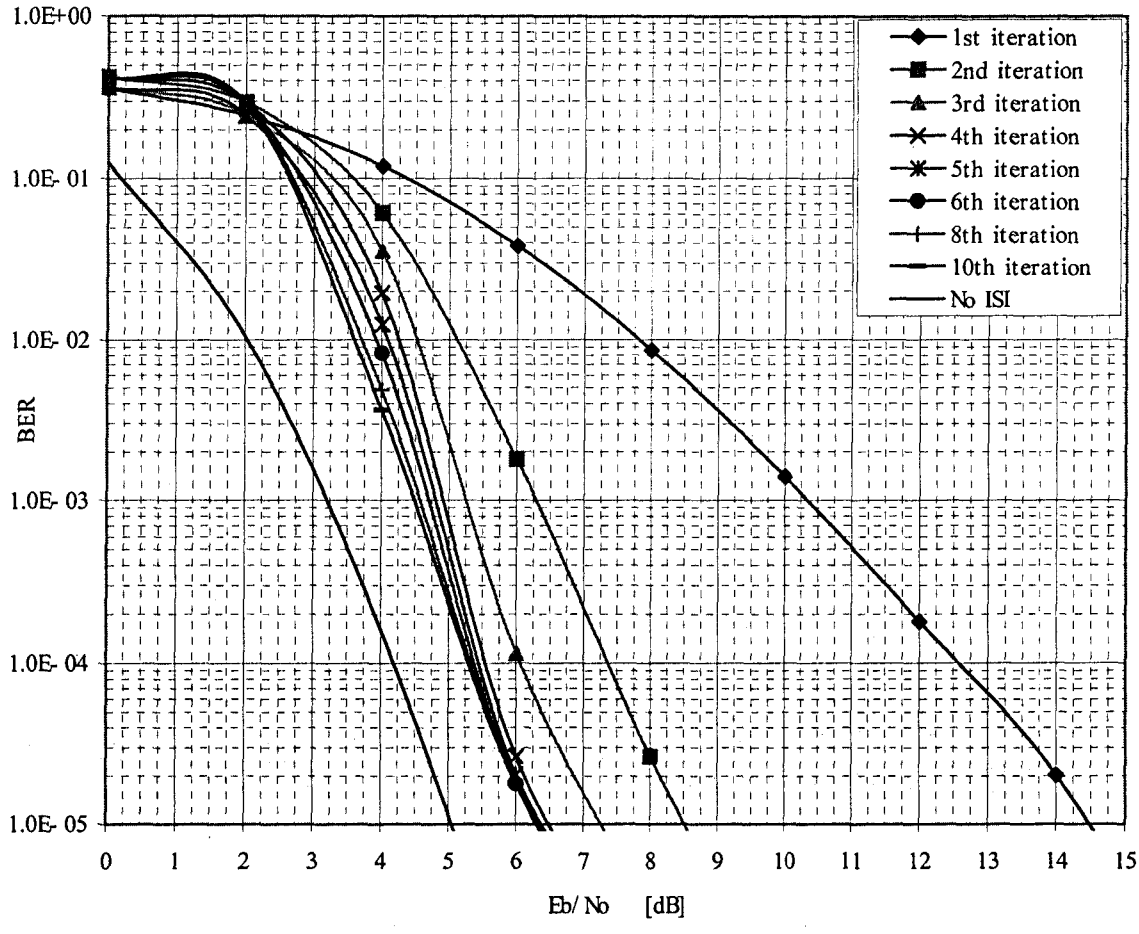



Figure 4-9 QPSK - LE/IC-hard over Channel 1

Figure 4-10 shows the performance comparison of the various turbo equalizers achieved after the fifth iteration in a coded QPSK system over Channel 1. From this figure, it is apparent that MAP-based turbo equalizer almost completely eliminates the ISI with sufficient $E_b/N_o$ (i.e. starting from 3 dB). On the other hand, the BER performance achieved by the soft-decision feedback IC-based turbo equalizers are nearly as good but require much less computation than the MAP-based turbo equalizer. The hard-decision feedback IC-based turbo equalizers do suffer from a performance penalty, but require much lower complexity.

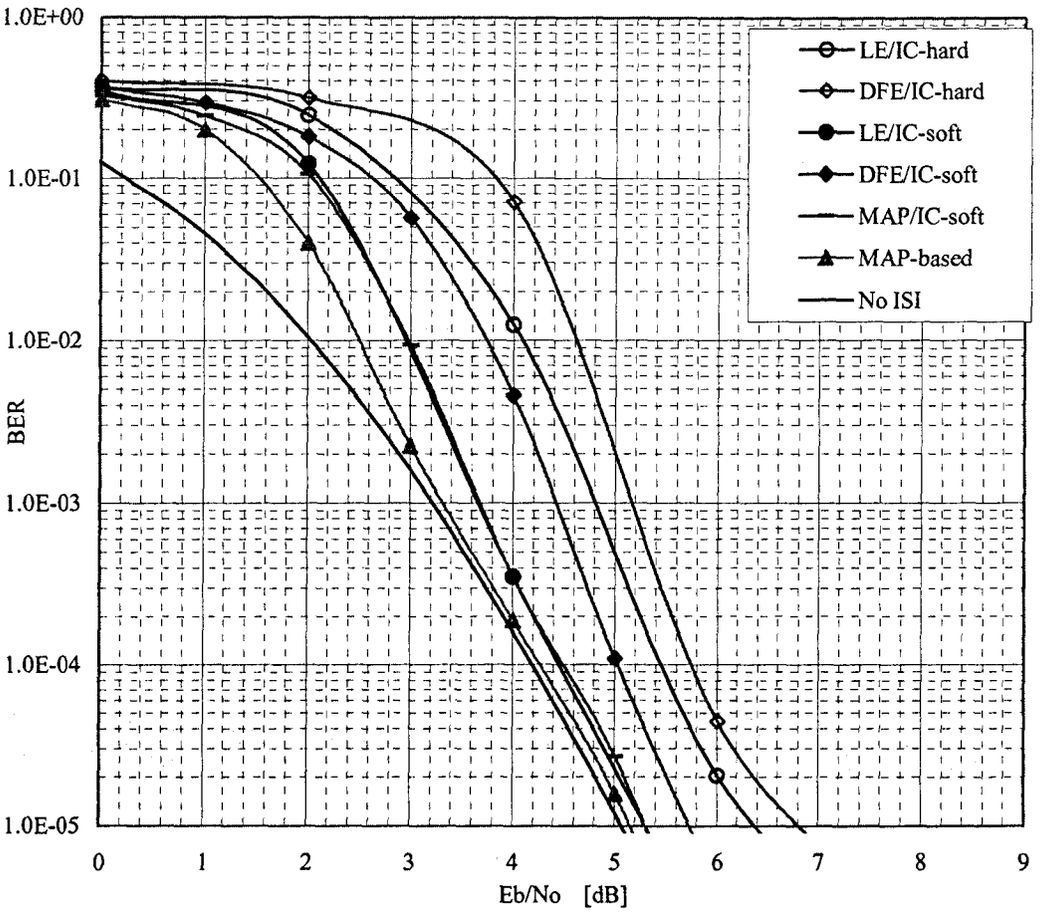


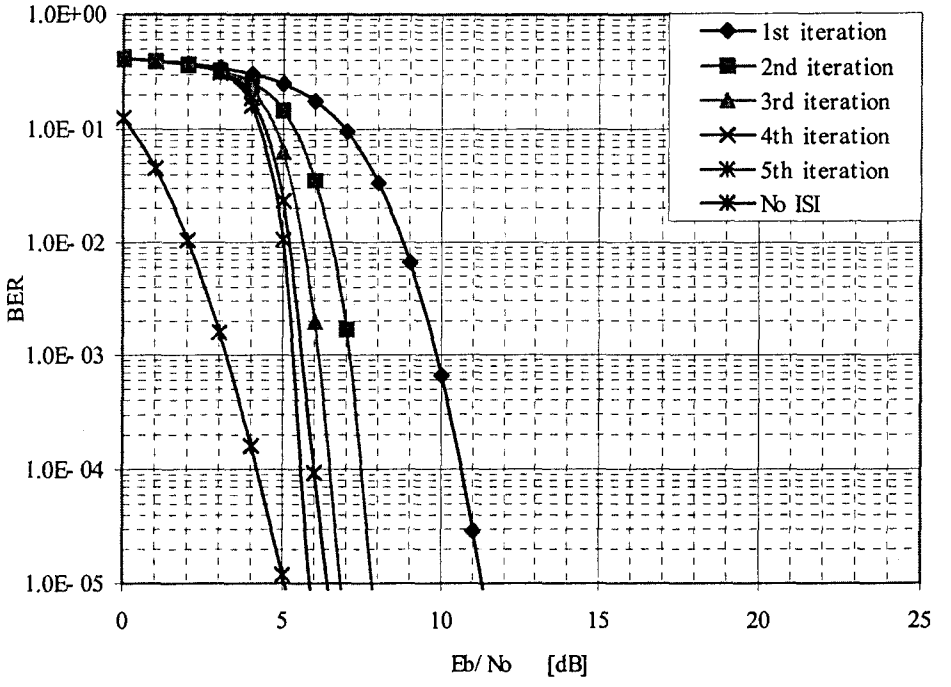

Figure 4-10 QPSK – Performance comparison among different turbo equalizer over Channel 1 after the fifth iteration

### 4-3-2 Different Channel Models

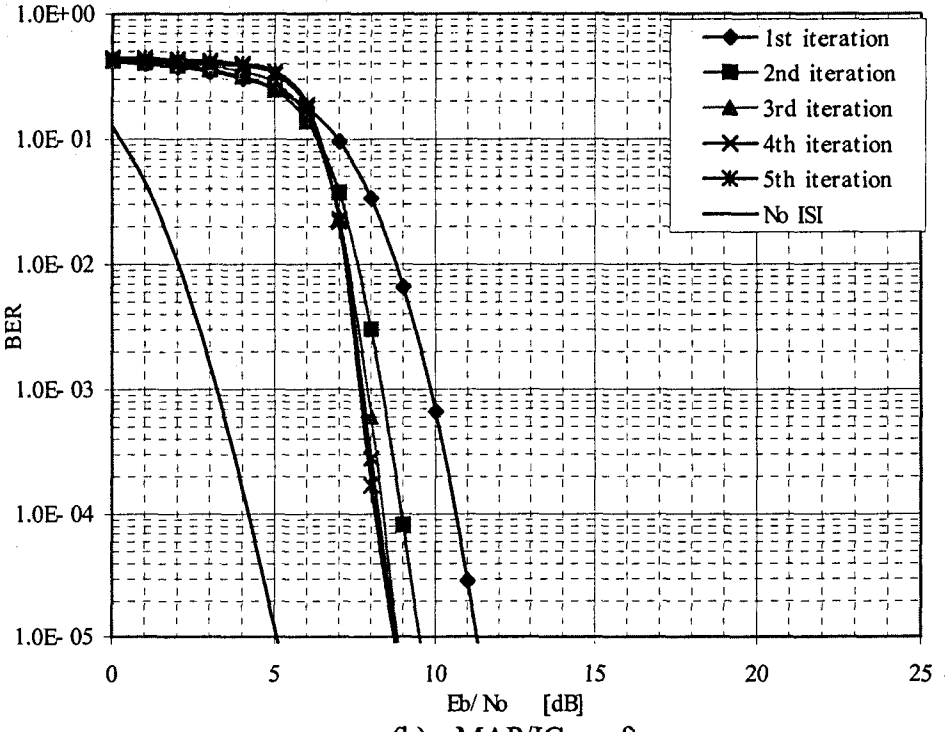
Next we consider the turbo equalizer performance over different channel models. The performance of each of the six turbo-equalizers over Channel 2 are shown in Figures 4-11 (a)-(f), and compared after the fifth iteration in Figure 4-12. All systems exhibit sustained turbo processing gains, but also suffer from a large residual ISI penalty. The IC-based systems in particular are ineffective at low $E_b/N_0$. From Figure 4-11, it is apparent that if the BER after the first iteration is higher than about 0.15, the information fed back to the interference canceller is too unreliable for the IC to work properly, causing error propagation and system failure. On the other hand, if the BER is below this trigger point, turbo processing can improve performance. This effect is less pronounced in Channel 3, as shown in Figure 4-13.

When the ISI is less severe than Channel 1 (e.g. Channel 4), all six systems are very effective, with a residual ISI penalty of less than 0.1 dB, as shown in Figure 4-14. In fact, the MAP-based turbo equalizer even appears to outperform the ISI-free system. Although this surprising result may be due to experimented uncertainty, this surprising result was explored further by considering the even milder Channel 5. At $E_b/N_0 = 5$dB, the ISI-free system gave a BER of $1.2\times10^{-5}$ and the MAP-based turbo equalizer gave a BER of $9.89\times10^{-6}$ after five iterations. In calculating these results, 100000 frames of 1000 bits were simulated, and 388 frames were detected with bit errors. Although exact confidence intervals are difficult to determine (since the bit errors do not occur independently), these results suggest that experimented uncertainty is not the cause and that the MAP-based turbo equalizer is performing better than the ISI-free system. If this is in fact the case, one possible explanation is that the turbo system is able to exploit the memory introduced by the channel to improve performance. This is similar to a serially concatenated convolutional

code, where the inner code is a very poor code, but better than nothing.

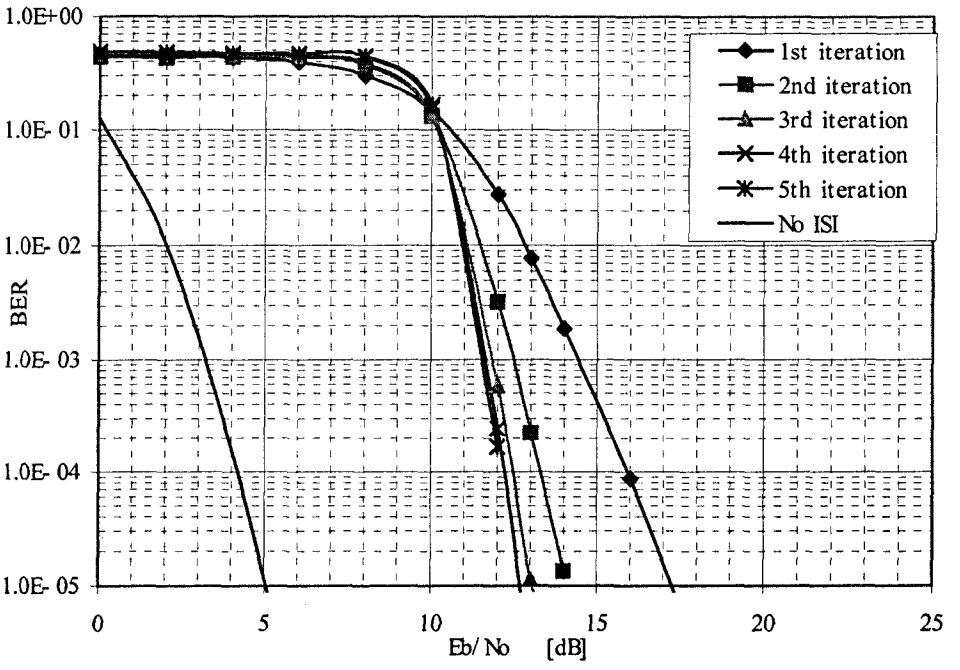



(a) MAP-based turbo equalizer

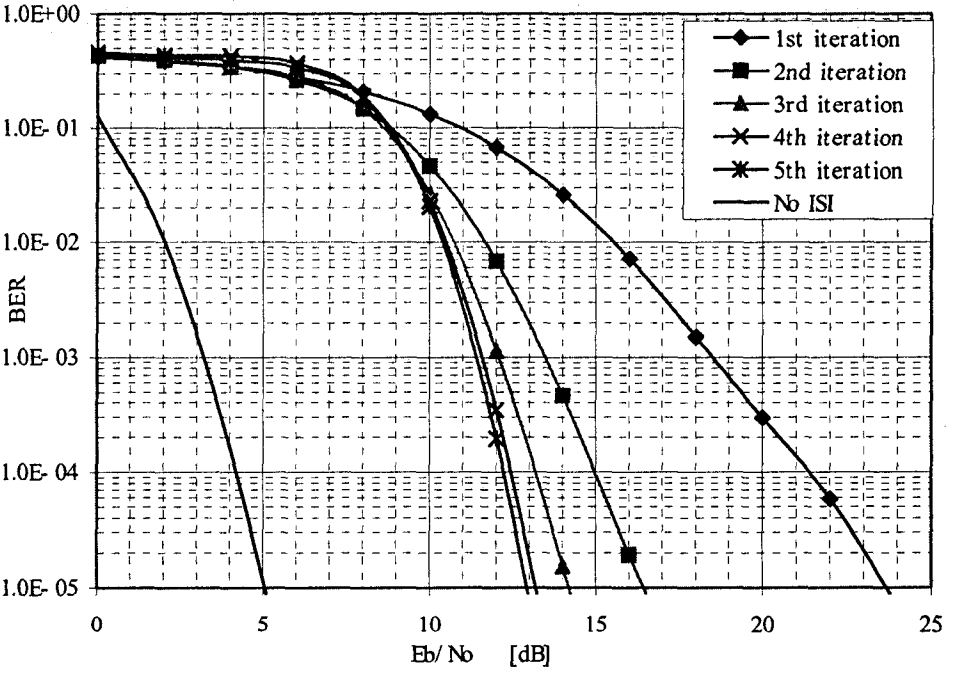



(b) MAP/IC - soft

Figure 4-11 QPSK - Performance of various turbo equalizers over Channel 2

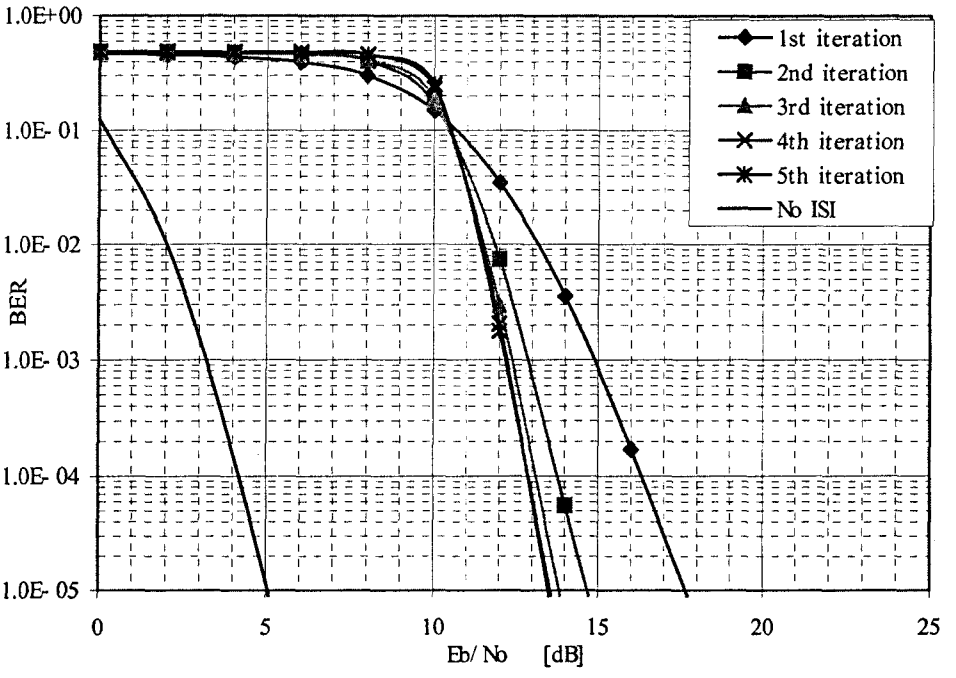



(c) DFE/IC-soft

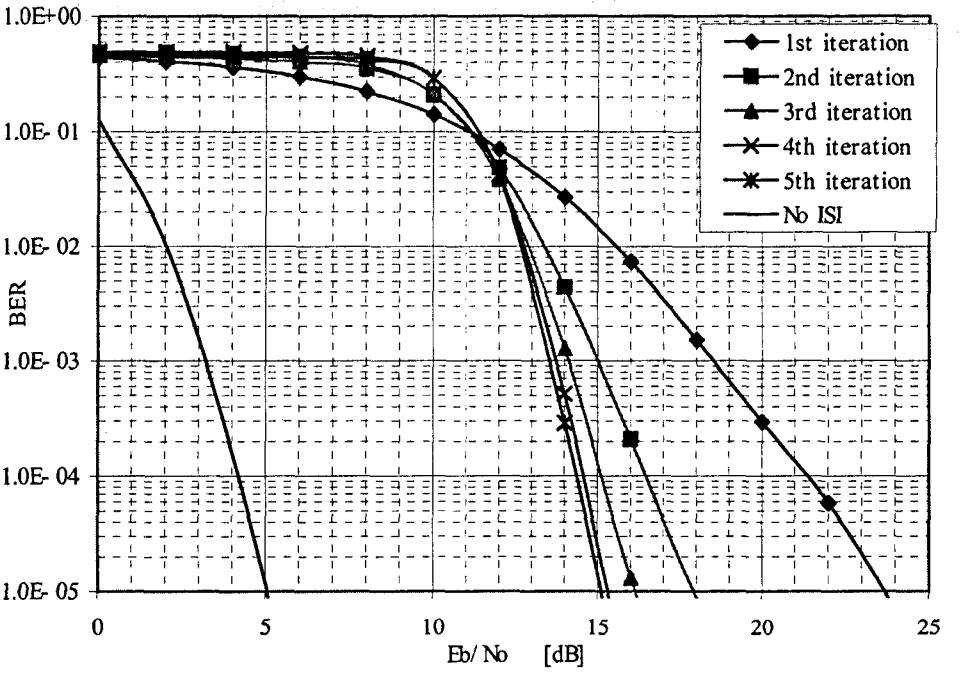



(d) LE/IC-soft

Figure 4-11 (Con't)




(e) DFE/IC-hard




(f) LE/IC-hard

Figure 4-11 (Con't)

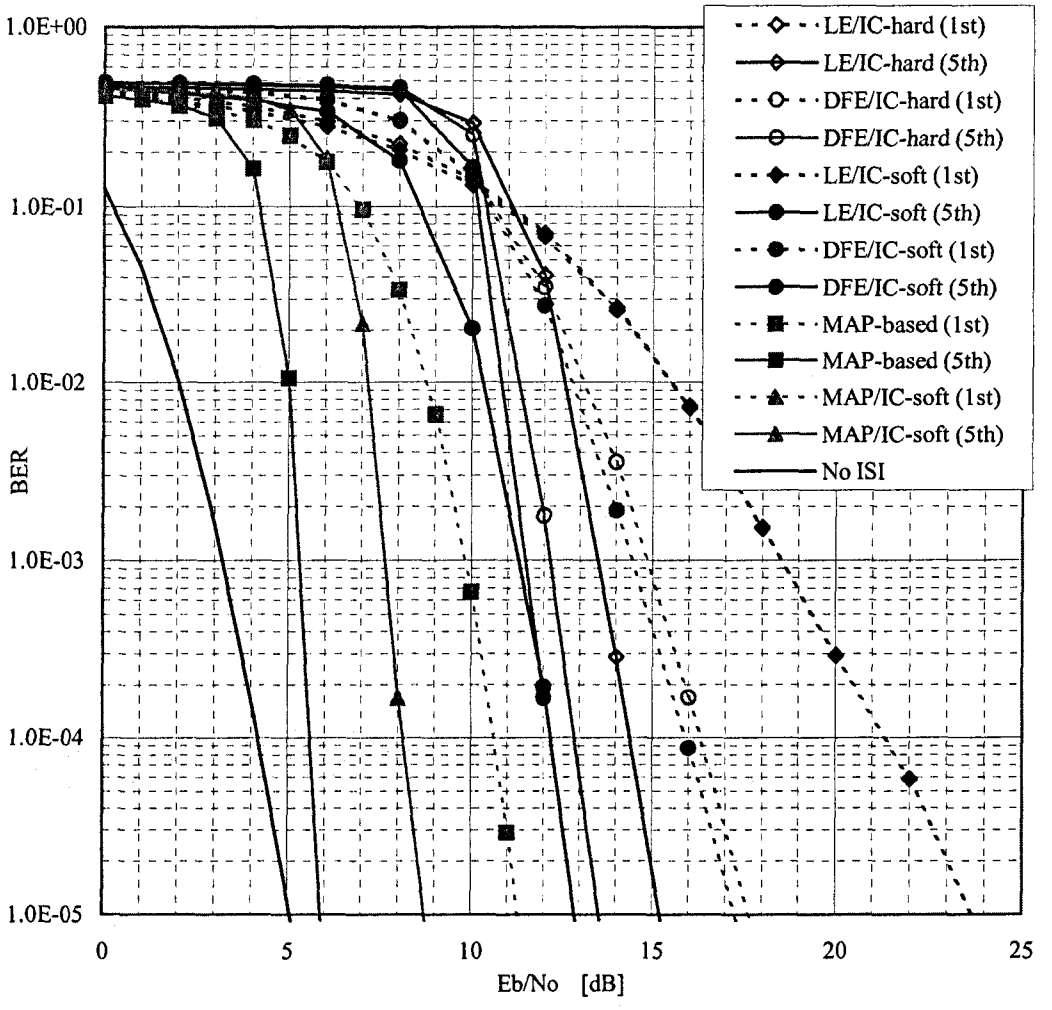



Figure 4-12 QPSK – Performance comparison of various turbo equalizers after fifth iteration over Channel 2

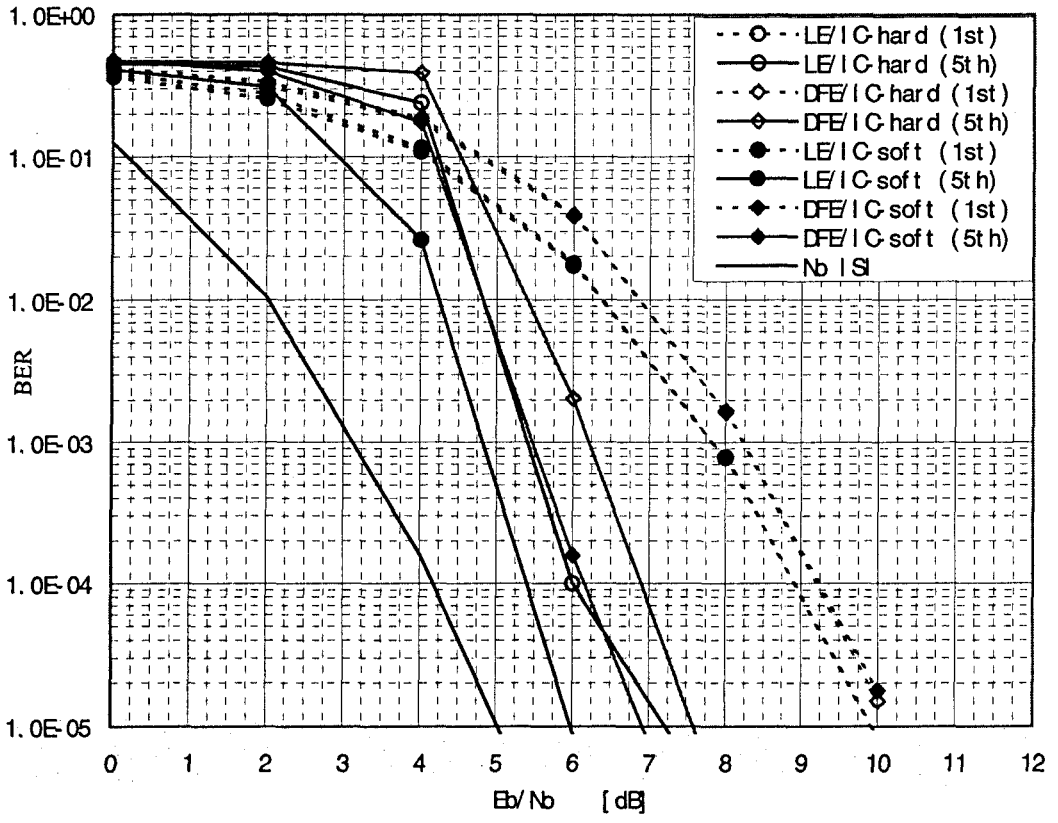



Figure 4-13 QPSK - Performance of various turbo equalizers over Channel 3

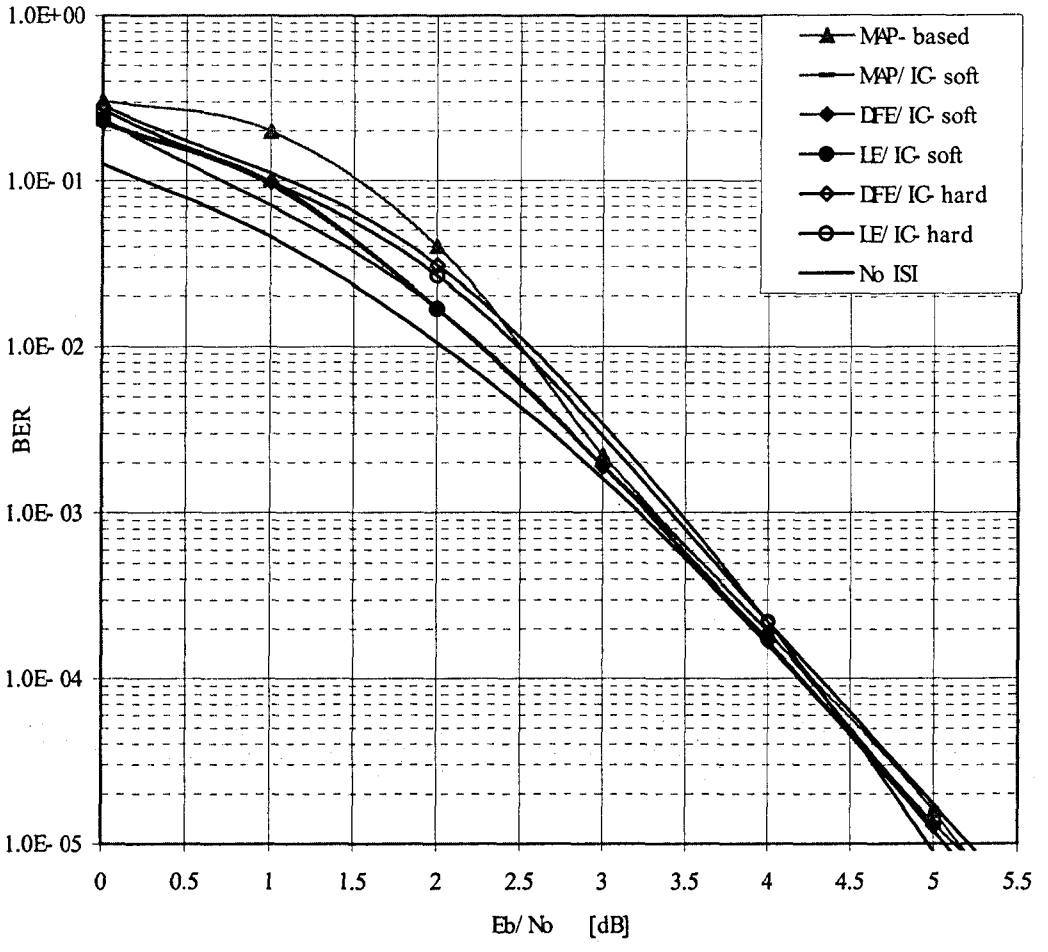
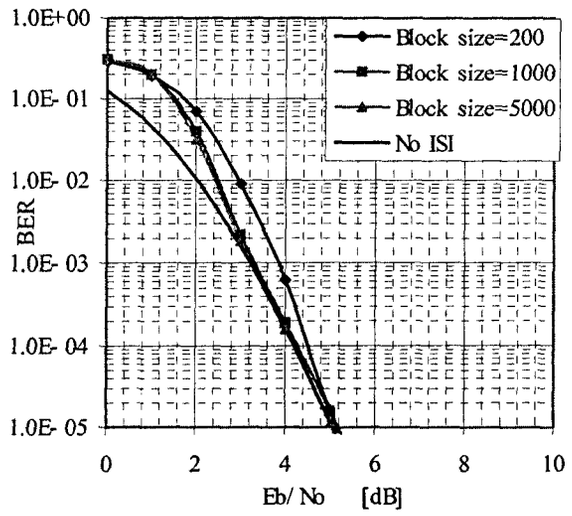



Figure 4-14 QPSK-Performance comparison among different turbo equalizers over Channel 4

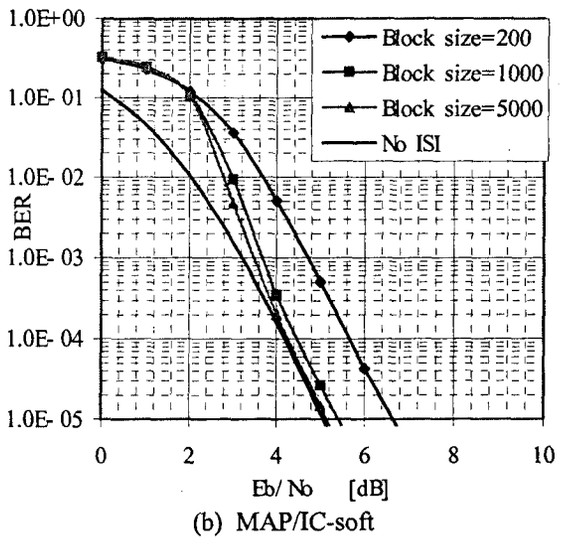
### 4-3-3 Different Block Sizes

In the previous simulation results, the number of message bits per block is selected to be 1000 bits. This size allows the symbol-based random de-interleaver to shuffle the sequence of the equalized samples such that the correlated noise in the samples at the input of the de-interleaver can be appeared as the random noise at the output of the de-interleaver. By changing the block size, the performance of the turbo equalizers will also be affected. Figure 4-15 depicts the turbo equalizer performance when different block sizes are considered.

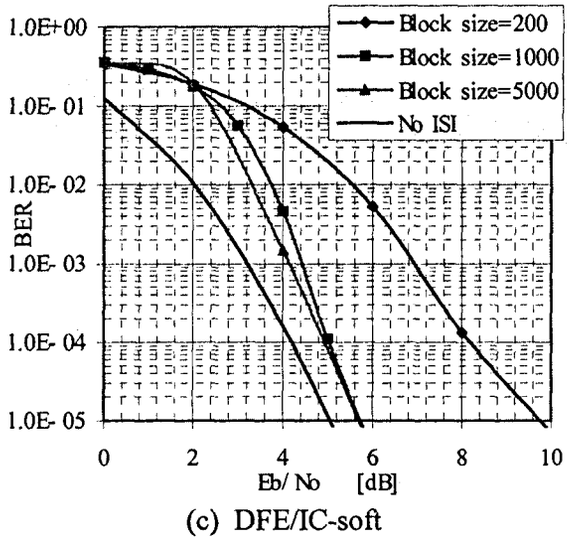
In Figure 4-15, it shows that the performance of the turbo equalizers with block size of 200 are worse than that with block size of 1000. The LE and DFE-based systems suffer more when the block size is reduced than the MAP-based systems. However, the performance of turbo equalizers with block size of 5000 has slightly advantage than that with block size of 1000 since the noise appeared at the output of the de-interleaver with block size of 5000 is much more randomness than with block size of 1000. We can conclude that larger block size can allow the turbo equalizer to perform better; however, a smaller block results in a shorter decoding delay.



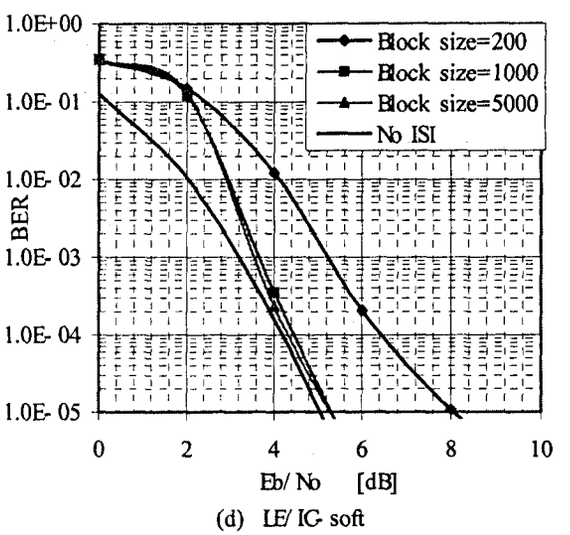

(a) MAP-based turbo equalizer




(b) MAP/IC-soft

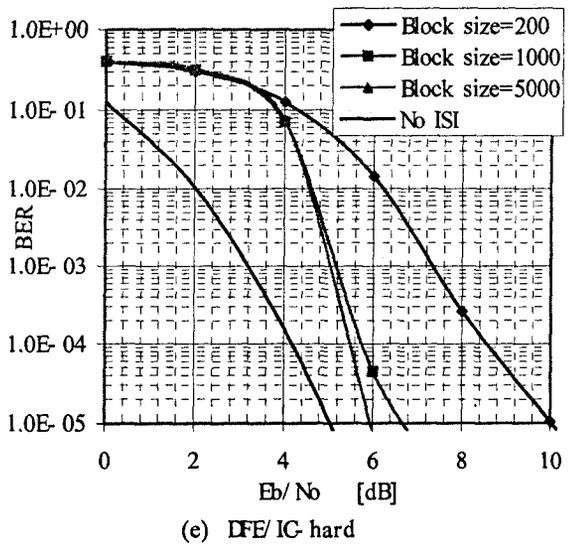



(c) DFE/IC-soft

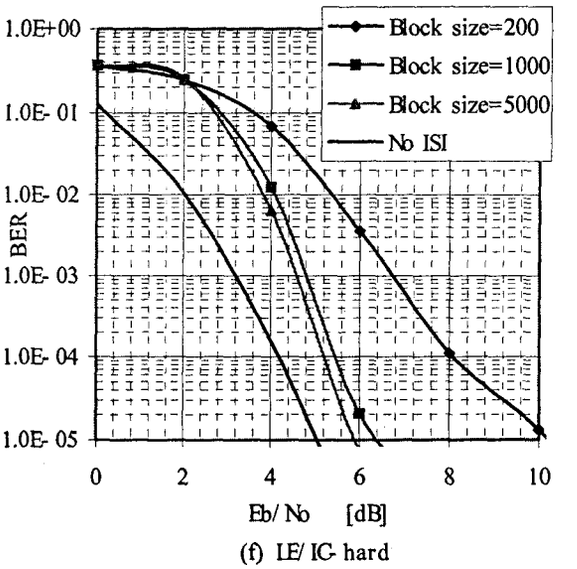



(d) LE/ IC- soft

Figure 4-15 QPSK - Performance of different turbo equalizers achieved at the fifth iteration with different block sizes over Channel 1




(e) DFE/ IC- hard




(f) LE/ IC- hard

Figure 4-15 (con't)

### 4-3-4 Binary-based Interleaving

We have seen that the size of interleaver and de-interleaver are curial components in the turbo equalizers which affect the overall performance. Now, let us consider the case of binary-based random interleaving. For the binary-based interleaver, the sequence of code bits (instead of code symbols) are shuffled, and then sent to the modulator for mapping. Figure 4-16 shows the performance of the turbo equalizers when binary-based interleaving is used instead of symbol-based interleaving. Compared with Figure 4-10, it is seen that the MAP-based and soft-decision feedback IC-based turbo equalizers perform similarly when either binary or symbol-based interleaving is used. Moreover, the hard-decision systems perform worse with binary-interleaving. Since binary-based interleaving is harder to implement, there is no advantage to its use in this case.

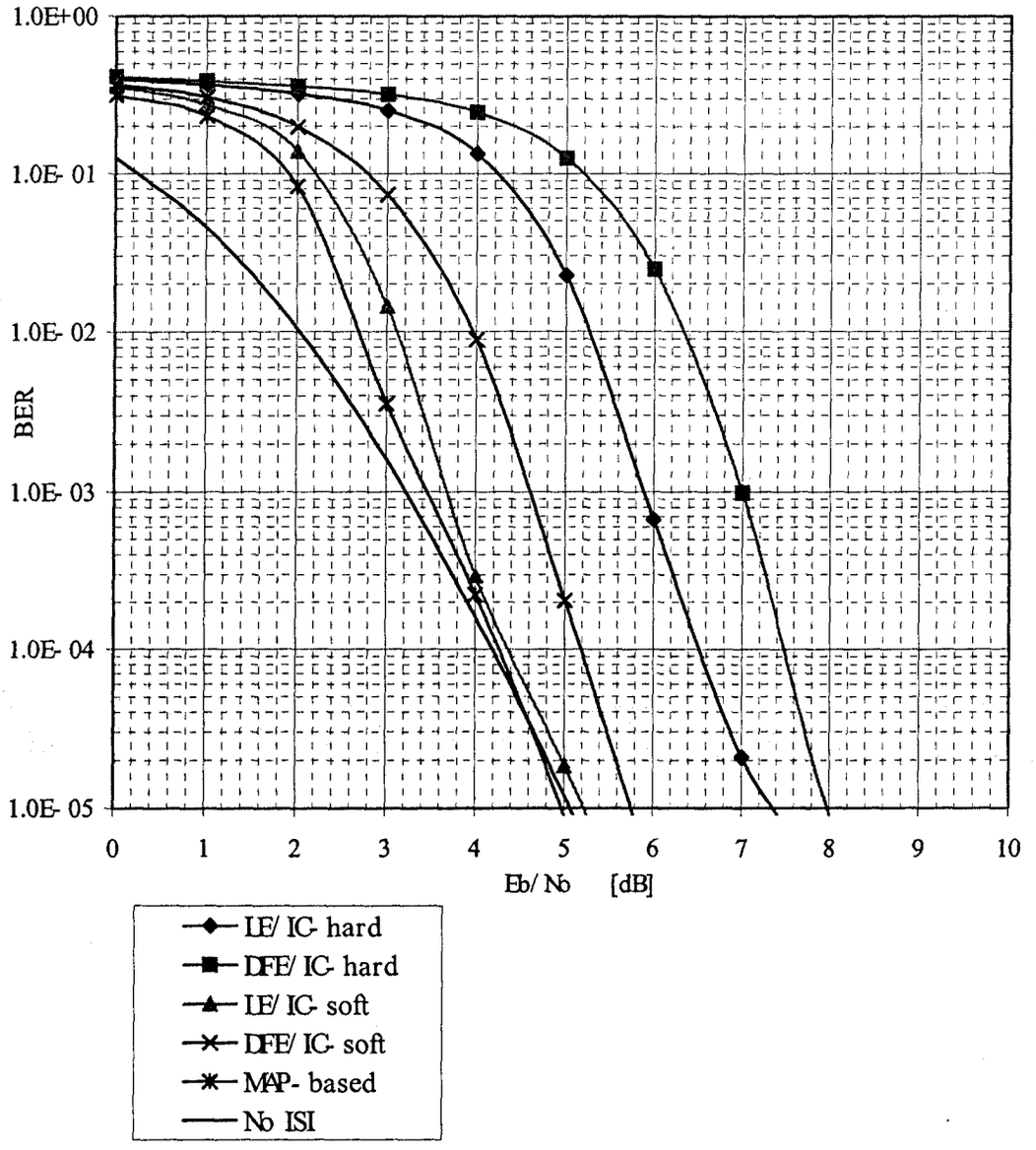



Figure 4-16 QPSK - Performance comparison of different turbo equalizers with binary-based interleaving over Channel 1

In a coded QPSK system, we have shown that the soft-decision feedback IC-based turbo equalization technique can provide reliable performance; however, the symbol rate is kept to be one information bit/symbol. In practice, especially in wireless communications, bandwidth is very limited; therefore, to conserve the bandwidth a larger symbol rate is required. In the next section, we will examine the performance of the turbo equalizers over a coded 8-PSK system with a 16-state rate-2/3 convolutional code in which the information data rate can be increased to two information bits/symbol.

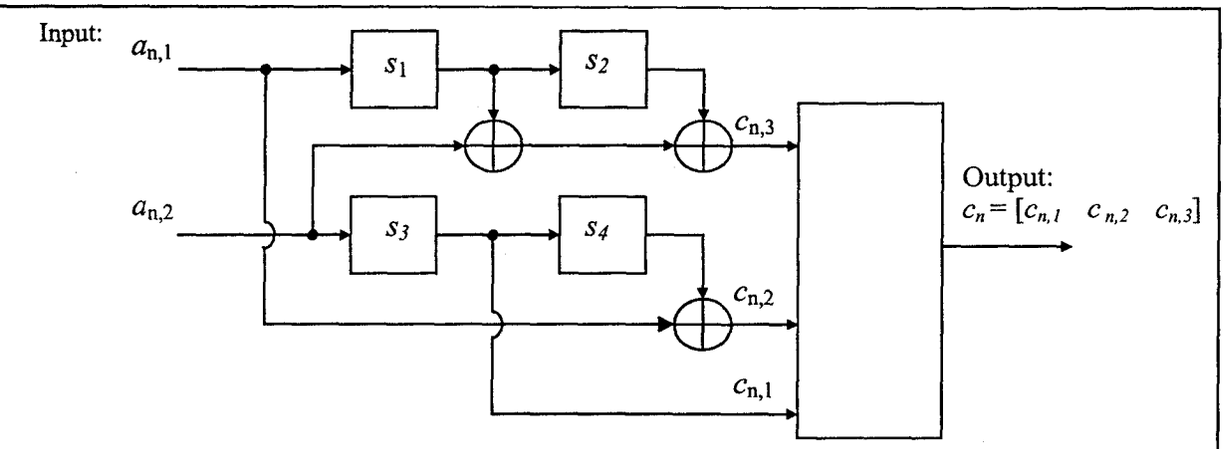
## 4-4 Performance of the Turbo Equalizers with Coded 8-PSK

In a coded 8-PSK system, we will illustrate the performance of turbo equalization techniques with different coding schemes. Figure 4-17 presents the three different convolutional encoders which will be considered in this thesis. The first two convolutional codes can be also represented by the generator matrices such as

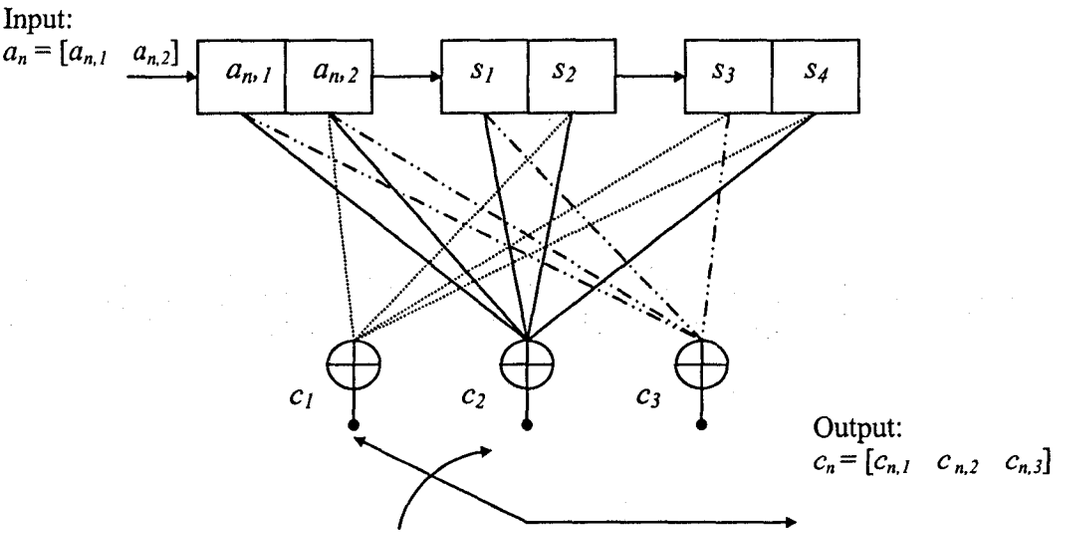
Encoder 1: Generator = $(45,22,10)_8$ in octal form, and

Encoder 2: Generator = $(27,75,72)_8$ in octal form.

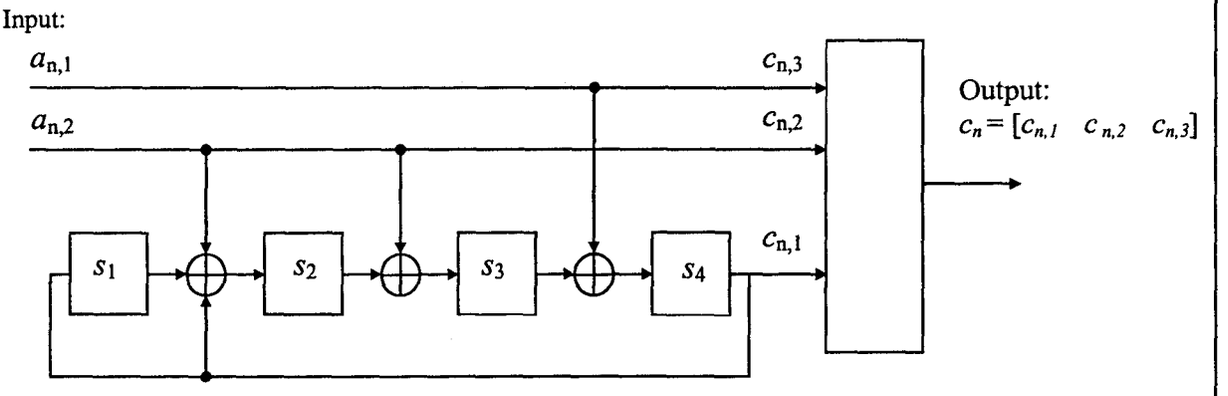
And the last convolutional encoder (Encoder 3) is a systematic convolutional encoder with feedback. Note that Code 1 and Code 3 employ the trellis-coded modulation (TCM) whereas Code 2 is used with Gray mapping.




(a) Convolutional encoder with generators $(45,22,10)_8$




(b) Convolutional encoder with generators $(27,75,72)_8$




(c) Systematic convolutional encoder with feedback

Figure 4-17 Structure of different convolutional encoders with rate-2/3

Figure 4-18 presents the performance of the described encoders. Obviously, Code 1 provides better performance than the other two encoders. The coding gain is about 0.63 dB relative to the Code 2 and a gain of 0.12 dB relative to the Code 3. Those convolutional encoders will be considered in our simulations to see how they affect the overall performance of the turbo equalizers.

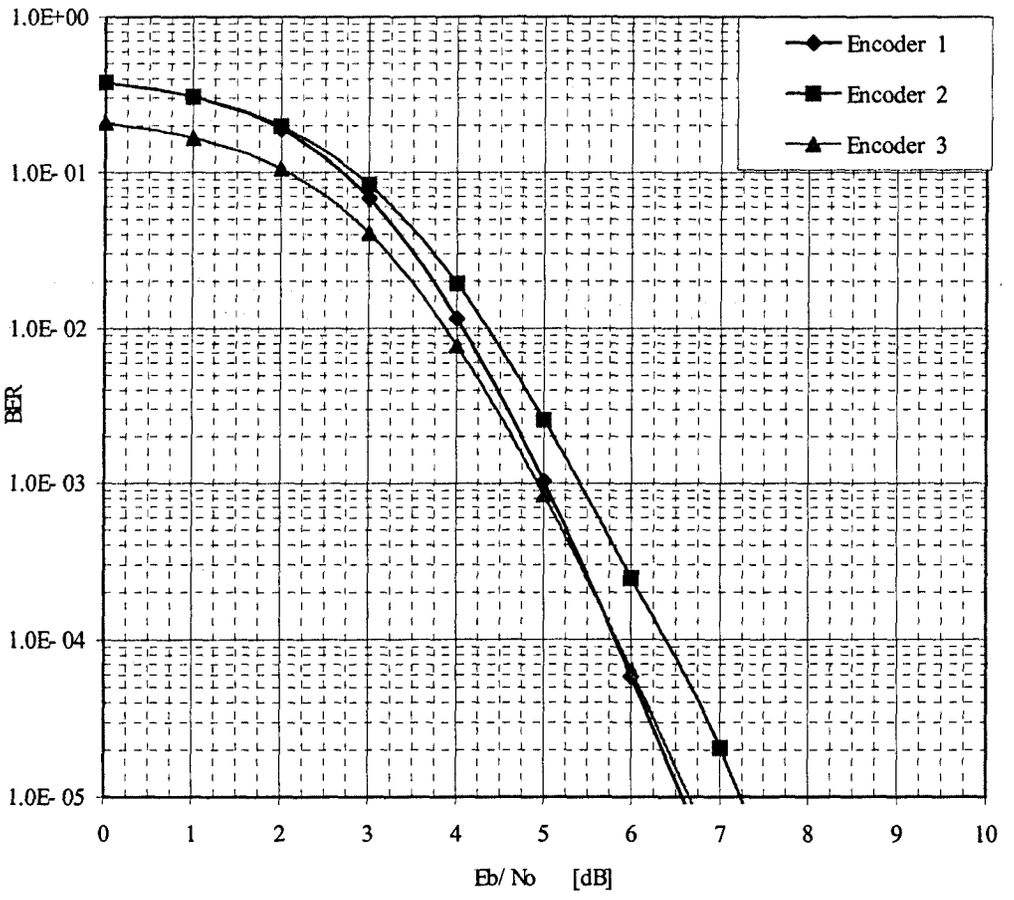



Figure 4-18 8-PSK - Performance of different convolutional codes over an AWGN channel without ISI

### 4-4-1 Performance of the Turbo Equalizers over Channel 1

In this section, symbol-based random interleaving is used in our simulations with a 16-state rate-2/3 convolutional code with the generator $(45,22,10)_8$. Figure 4-19 shows the performance of the LE/IC-soft over Channel 1 in a coded 8-PSK system for the first to fifth iterations.

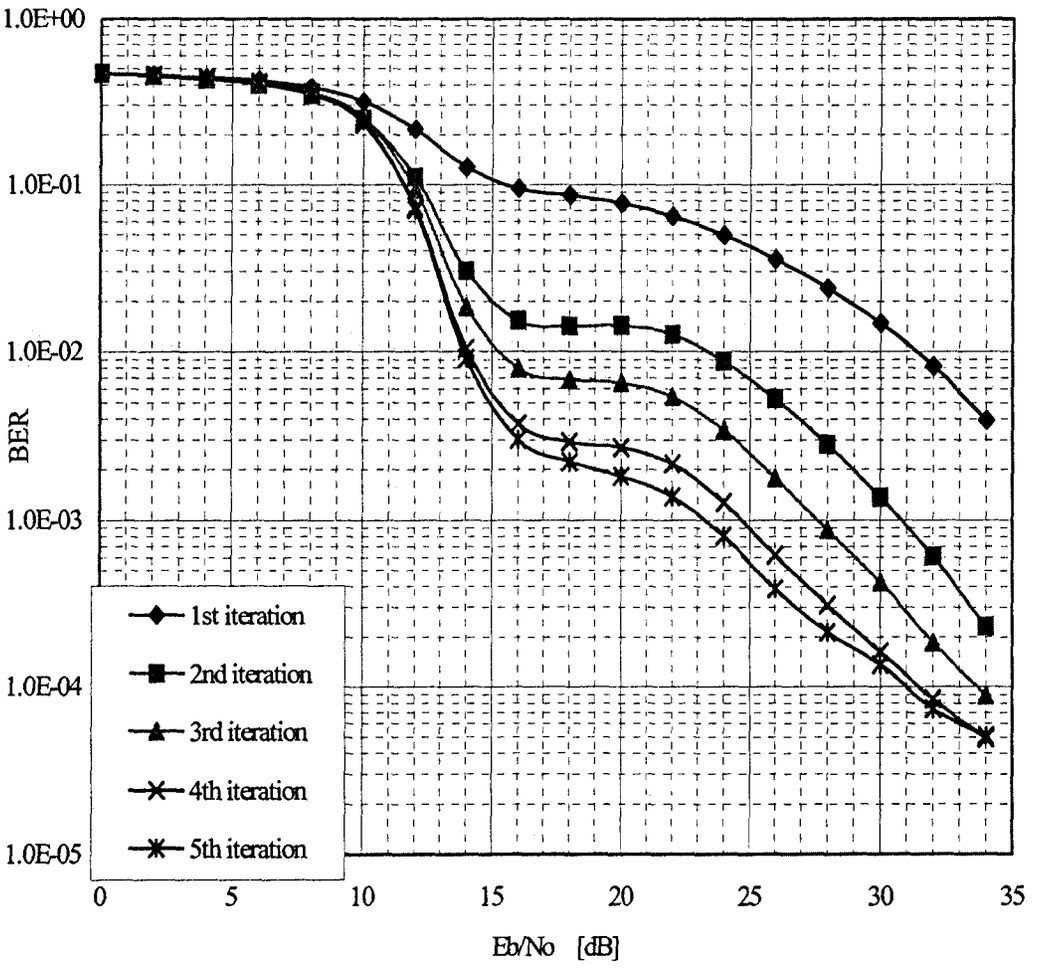



Figure 4-19 8PSK-Performance of LE/IC-soft over Channel 1

The most startling observation from this figure is that the LE/IC-soft has extremely poor performance, in which the BER has the flat behavior as $E_b/N_o$ increases. This is because the MAP decoder is not working properly. In calculating the branch metrics for the MAP decoder we used $N_o$ as the noise variance in Eq. (3-2). While this is appropriate if the equalizer was able to completely suppress the ISI, if some residual ISI remained the noise variance would be wrong. Although the MAP algorithm can tolerate some mismatch in the noise variance, it fails if the noise variance is significantly incorrect, as it is in this case. To better reflect the residual ISI, the decoder should use the minimum mean square error ($J_{min}$) for the noise variance. For the LE, the $J_{min}$ is expressed as [11]

$$J_{min} = 1 - \sum_{j=-K}^{0} c_j f_{-j}$$

Figure 4-20 shows the performance of LE/IC-soft over Channel 1 when $J_{min}$ is used by the MAP decoder during the first iteration. Obviously, the noise problem can be corrected.

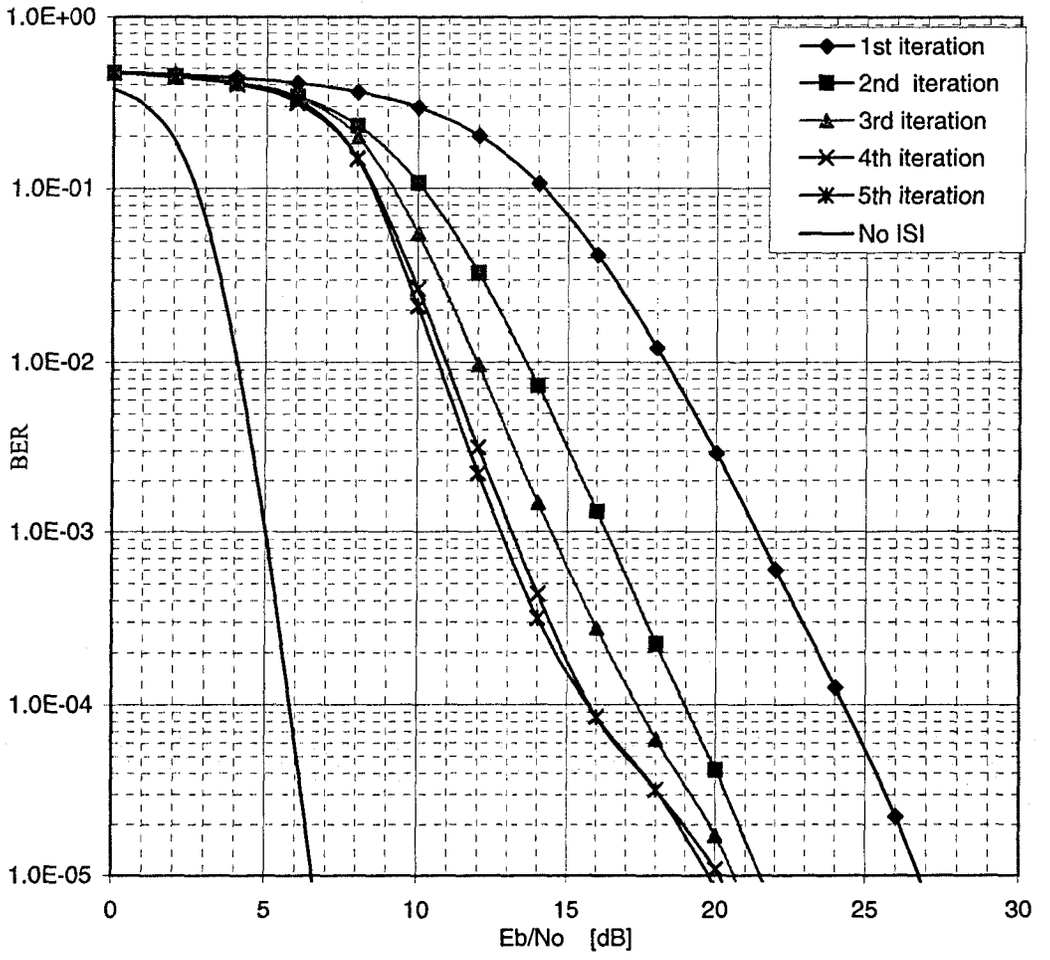



Figure 4-20 8PSK – LE/IC-soft over Channel 1 while $J_{min}$ is used to calculate the branch metrics

In order to shorten that flat behavior, the MAP equalizer can be used at the first iteration. Figure 4-21 shows the performance comparison of various turbo equalizers over Channel 1 with Encoder 1. In this figure, an improvement of 6dB can be achieved by the MAP/IC-soft relative to the DFE/IC-soft and the gain of 10 dB relative to the LE/IC-soft. On the other hand, the IC-based turbo equalizers have advantages over the performance of LE or DFE with MAP decoder. For the LE/IC-soft, it can provide a 7 gain relative to the performance of LE. For the DFE/IC-soft, it provides 3.5 dB gain relative to the performance of DFE.

With the MAP-based turbo equalizer, the performance can be close to the performance of the coded 8PSK system without ISI which the turbo equalizer intends to reach. The ISI penalty is only about 0.25 dB at the bit error of $10^{-5}$. Certainly, the number of computations for the MAP-based turbo equalizer is higher than the IC-based turbo equalizers.

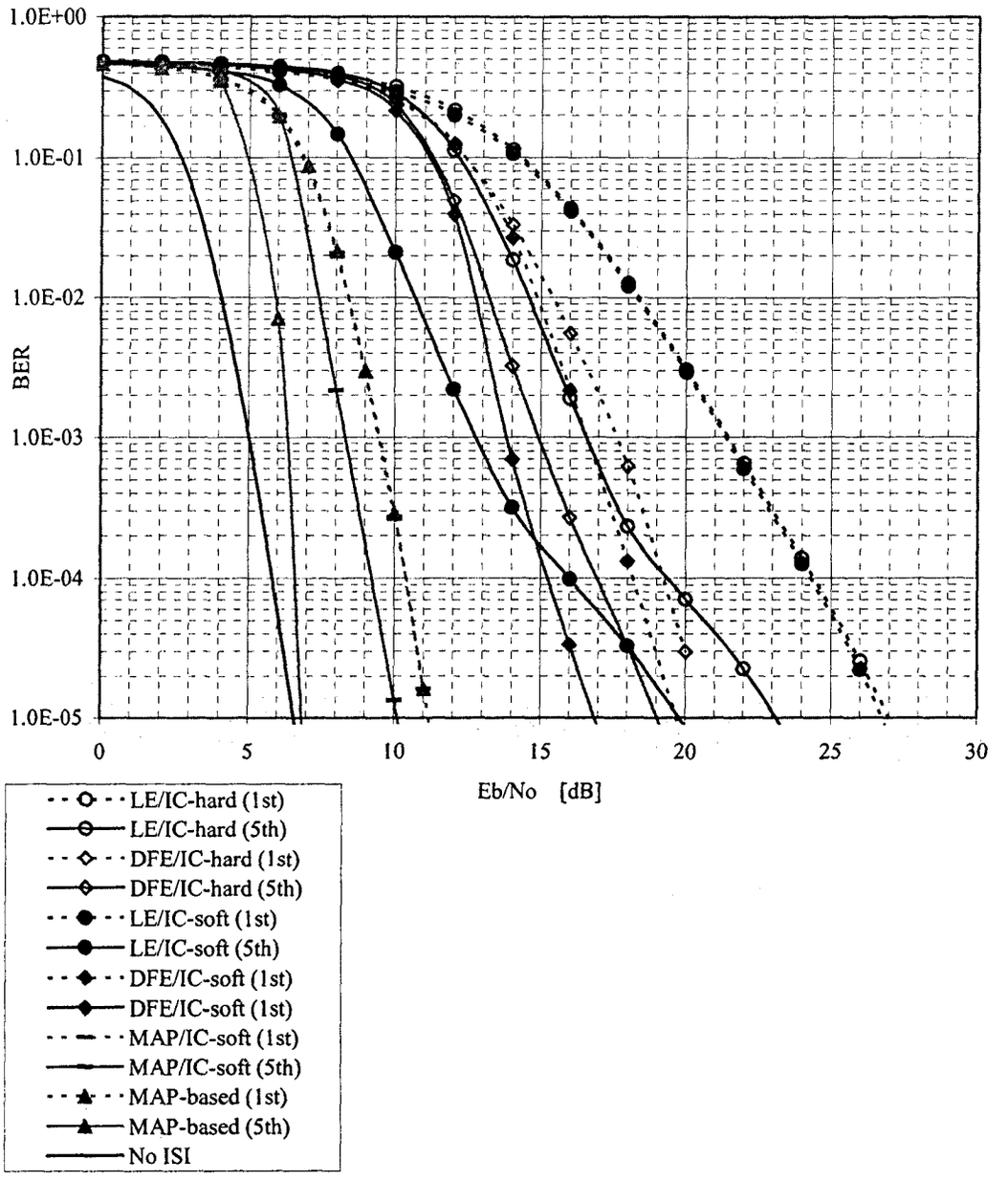
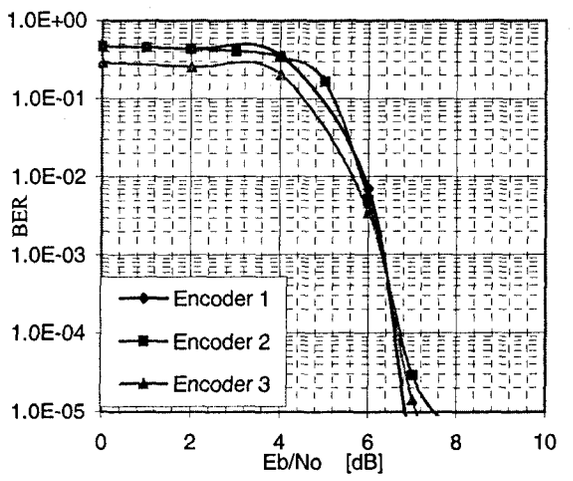


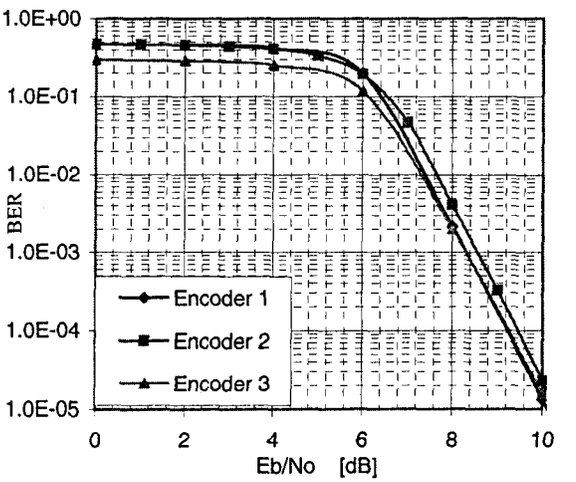

Figure 4-21 8PSK – Performance comparison of different turbo equalizers over Channel 1, after five iterations

### 4-4-2 Different Convolutional Codes

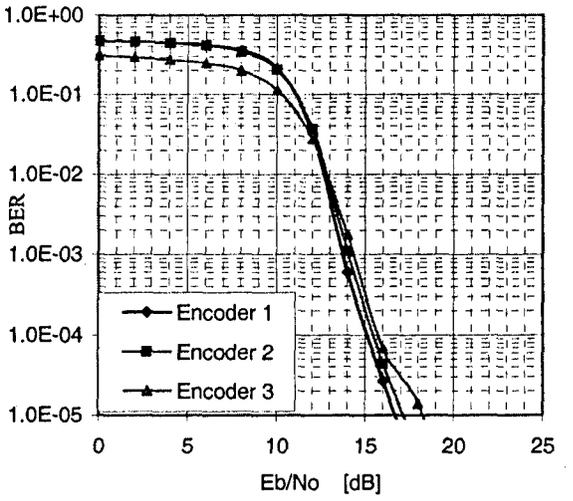
Figure 4-22 shows the turbo equalizers performance with Encoder 1, 2 and 3 respectively over Channel 1. For all six turbo equalizers, we observe that they are sensitive to the convolutional codes. They have similar performance behavior as the performance of the coded 8PSK without ISI, in which Encoder 1 provides better performance than the other two convolutional codes. Hence, an appropriate convolutional code should be chosen such that the reliability of the turbo-equalizer performance can be remained.



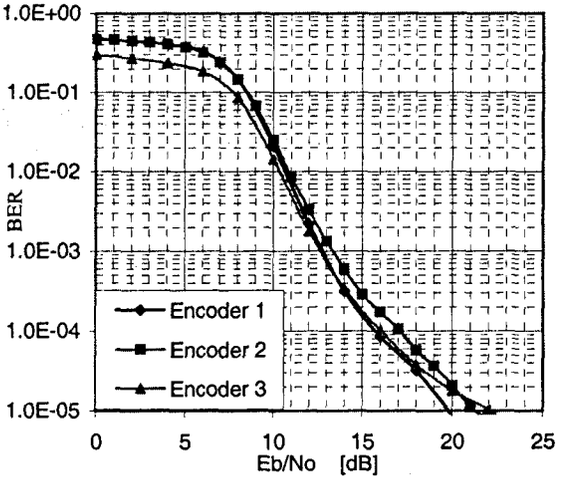

(a) MAP-based turbo equalizer




(b) MAP/IC-soft

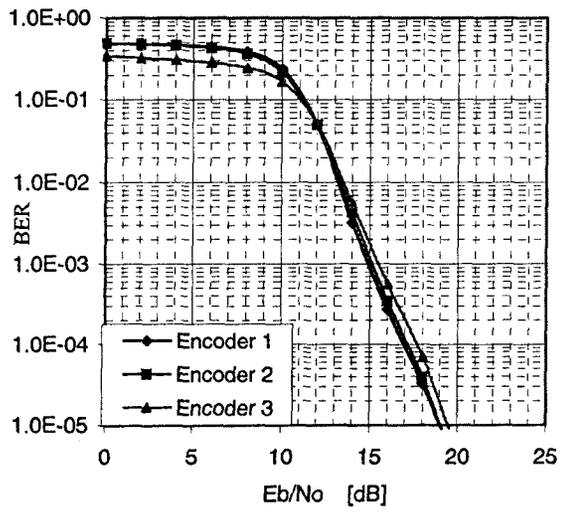



(c) DFE/IC-soft

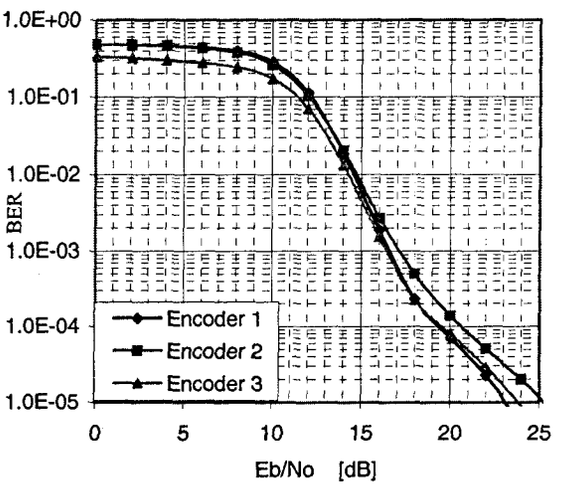



(d) LE/IC-soft

Figure 4-22 8PSK - Performances of different turbo equalizers achieved at the sixth iteration with different convolutional codes over Channel 1




(e) DFE/IC-hard

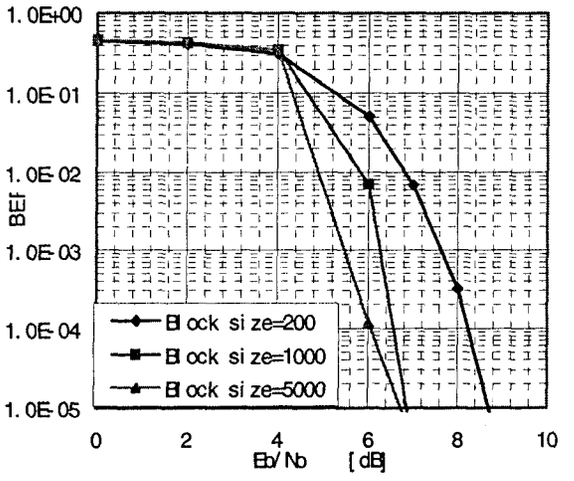



(f) LE/IC-hard

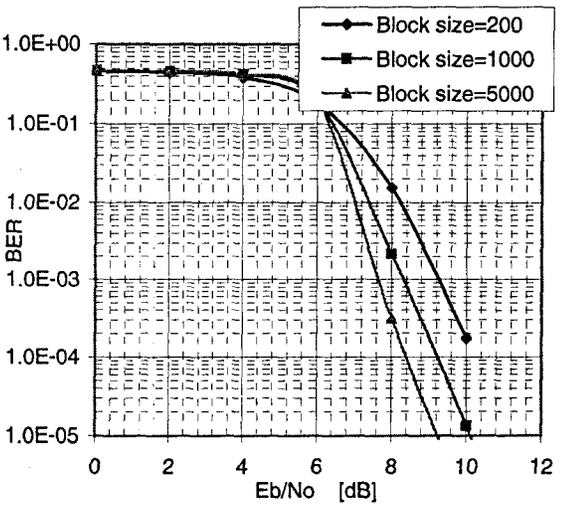
Figure 4-22 (con't)

### 4-4-3 Different Block Sizes

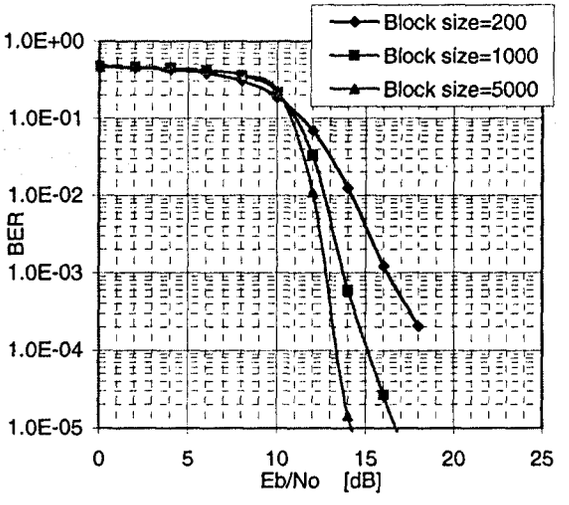
Let us now investigate the turbo equalizers performance with generators (45,22,10) while different block sizes are assumed. Figure 4-23 depicts the performance of the turbo equalizers with the block size of 200, 1000 and 5000. Similar to the coded-QPSK system, the block size can affect the performance of the turbo equalizers. With the block size of 5000, the soft-decision feedback IC-based turbo equalizers provides at least 2 dB gain relative to the block size of 1000. In terms of decoding delay, the turbo equalizers with the block size of 1000 are preferable since they give minimal decoding delay and reliable performance.



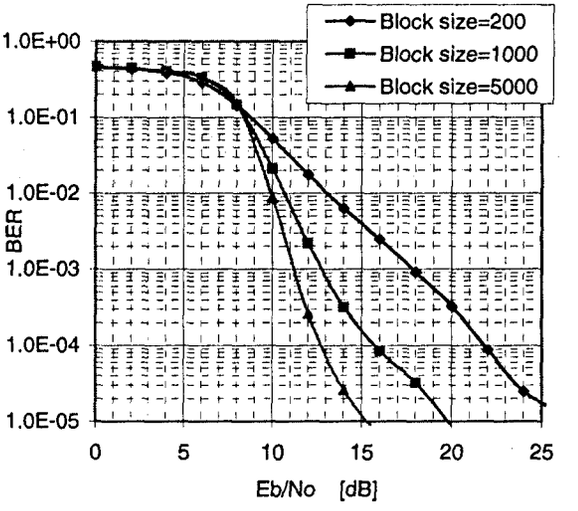

(a) MAP-based turbo equalizer




(b) MAP/IC-soft

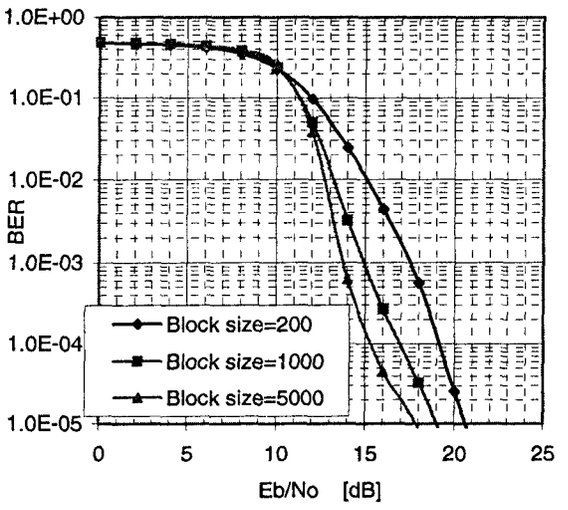



(c) DFE/IC-soft

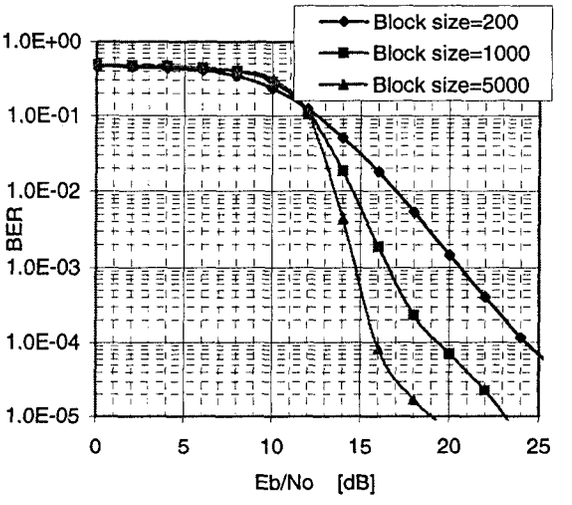



(d) LE/IC-soft

Figure 4-23 8PSK - Performances of different turbo equalizers after five iterations with different block sizes over Channel 1




(e) DFE/IC-hard

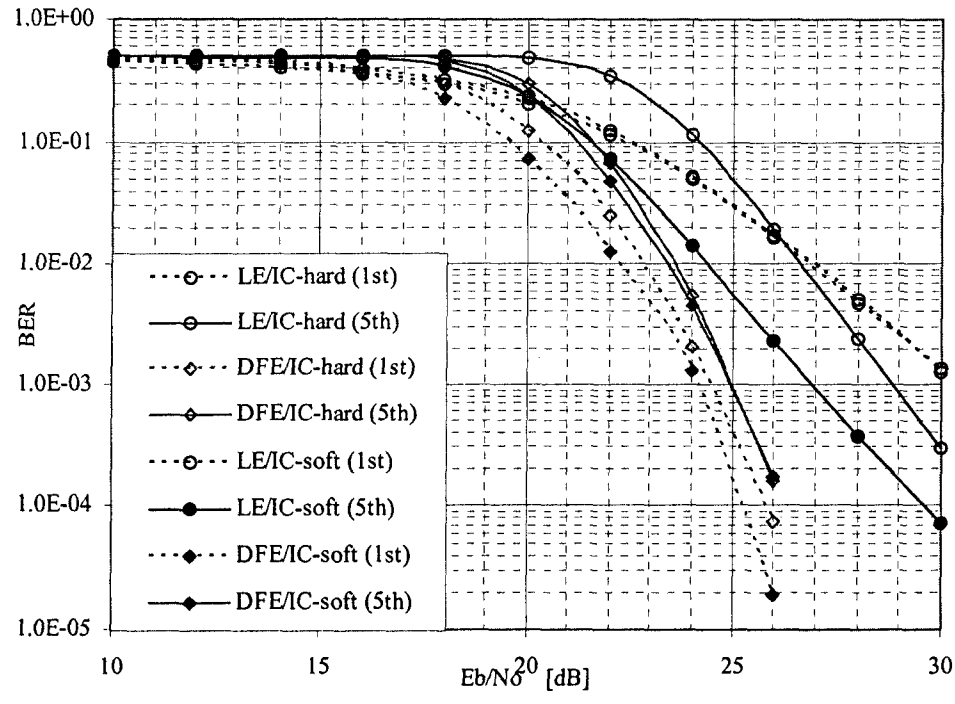



(f) LE/IC-hard

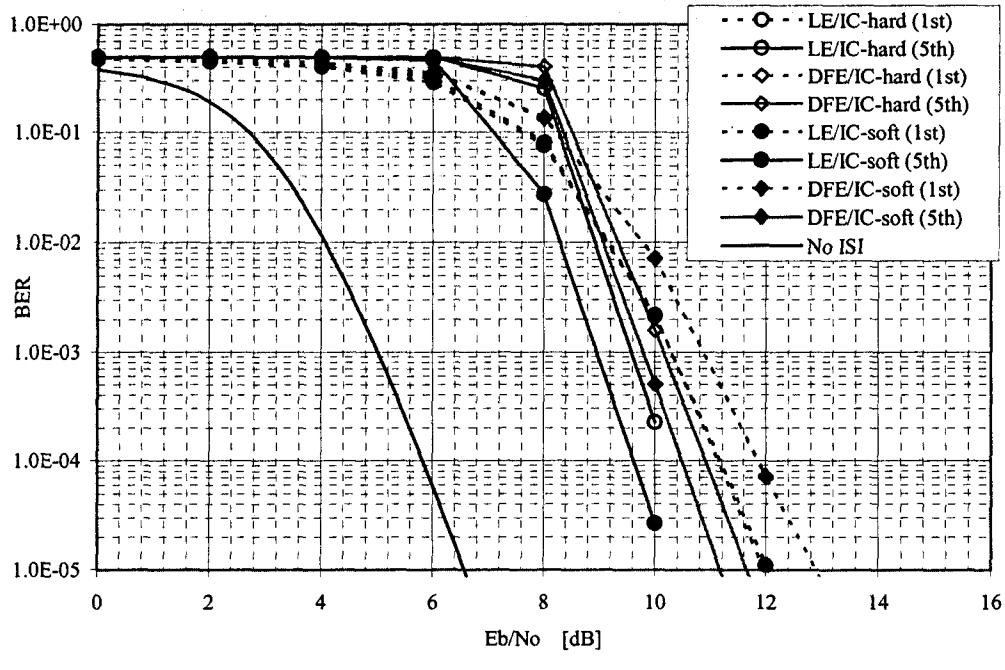
Figure 4-23 (con't)

### 4-4-4 Different Channel Models

To broadly evaluate the performance of the turbo equalizers, let us show the performance over different channel models. (shown in Figure 4-24) Over a severe channel models, i.e. Channel 2, the IC-based turbo equalizers are not allowed to perform effectively since the performance of the equalizers used at the first iteration is extremely poor. Over Channel 3 and 4, the performance of the turbo equalizers can give some degree of improvement.

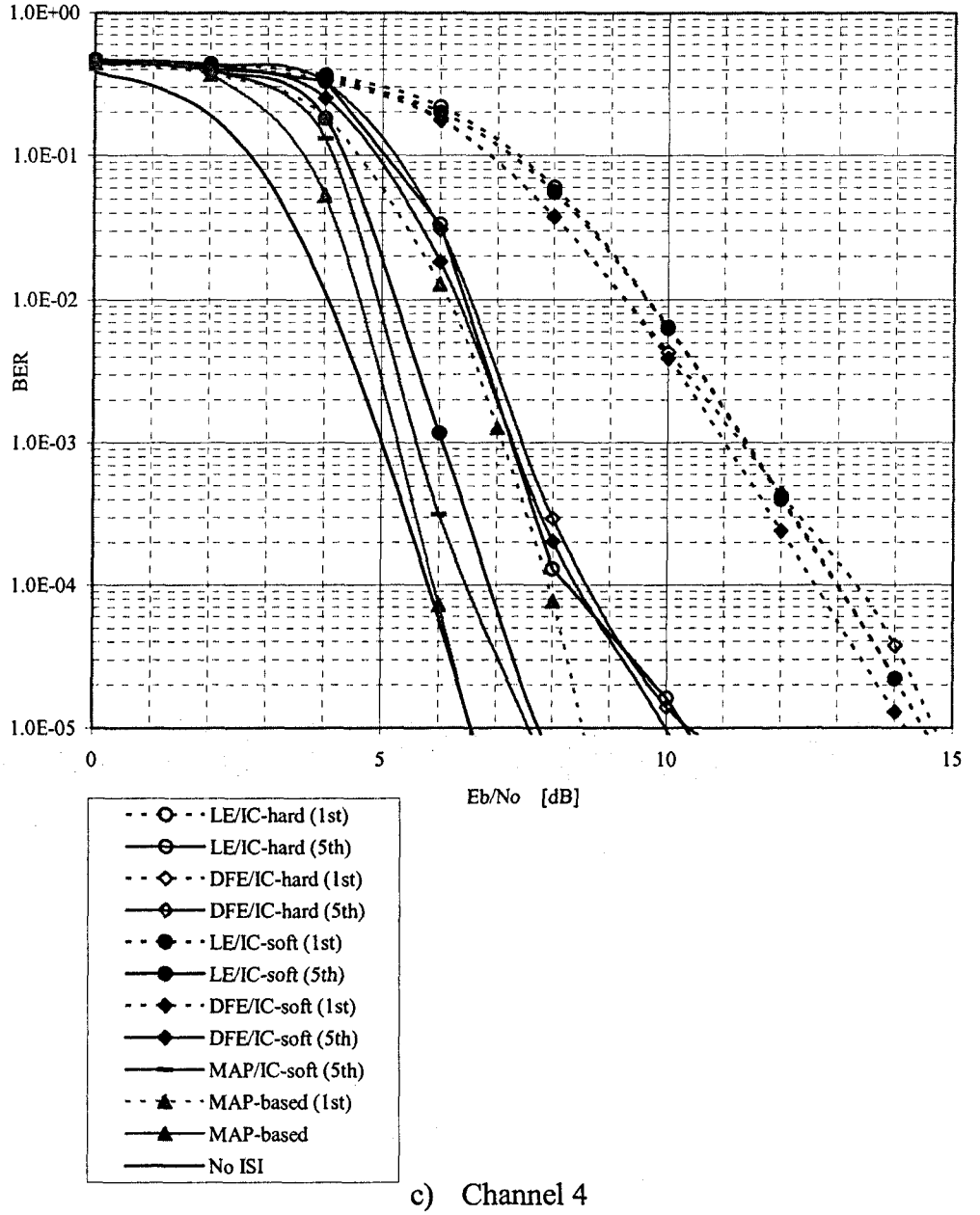



a) Channel 2




b) Channel 3

Figure 4-24 8PSK - Performance of various turbo equalizers with Code 1 over different channel models




c) Channel 4

Figure 4-24 (Con't)

### 4-4-5 Binary-based Interleaving

Comparing with the coded QPSK system, it is observed that the performance of the IC-based turbo equalizers in the coded 8-PSK system has not much significant improvement which they suffer at least 4 dB of ISI penalty at the bit error rate of $10^{-5}$. This is because the minimum distance between transmitted symbols in the coded 8-PSK system is much less than that in the coded QPSK system; therefore, the noise components can easily corrupt transmitted symbols. When a transmitted symbol is corrupted, three consecutive code bits may be in error. Since the decoder can only handle independent errors but not burst errors, the performance of the turbo equalizers is degraded. Thus, binary-based interleaving should be considered.

Figure 4-25 displays the performance comparison of various turbo equalizers with binary-based interleaving and Encoder 2 over Channel 1. It is shown that the performance gain of 8 dB can be obtained by the LE/IC-soft comparing to the LE/IC-soft with symbol-based interleaving. In addition, it is observed that the LE/IC-soft with binary-based interleaving can outperform the DFE/IC-soft with binary-based interleaving. With the advantage of binary-based interleaving, the linear equalizer provides better performance at the low $E_b/N_o$ where the trigger point can be reached. Hence, the LE/IC-soft can perform effectively after the trigger point.

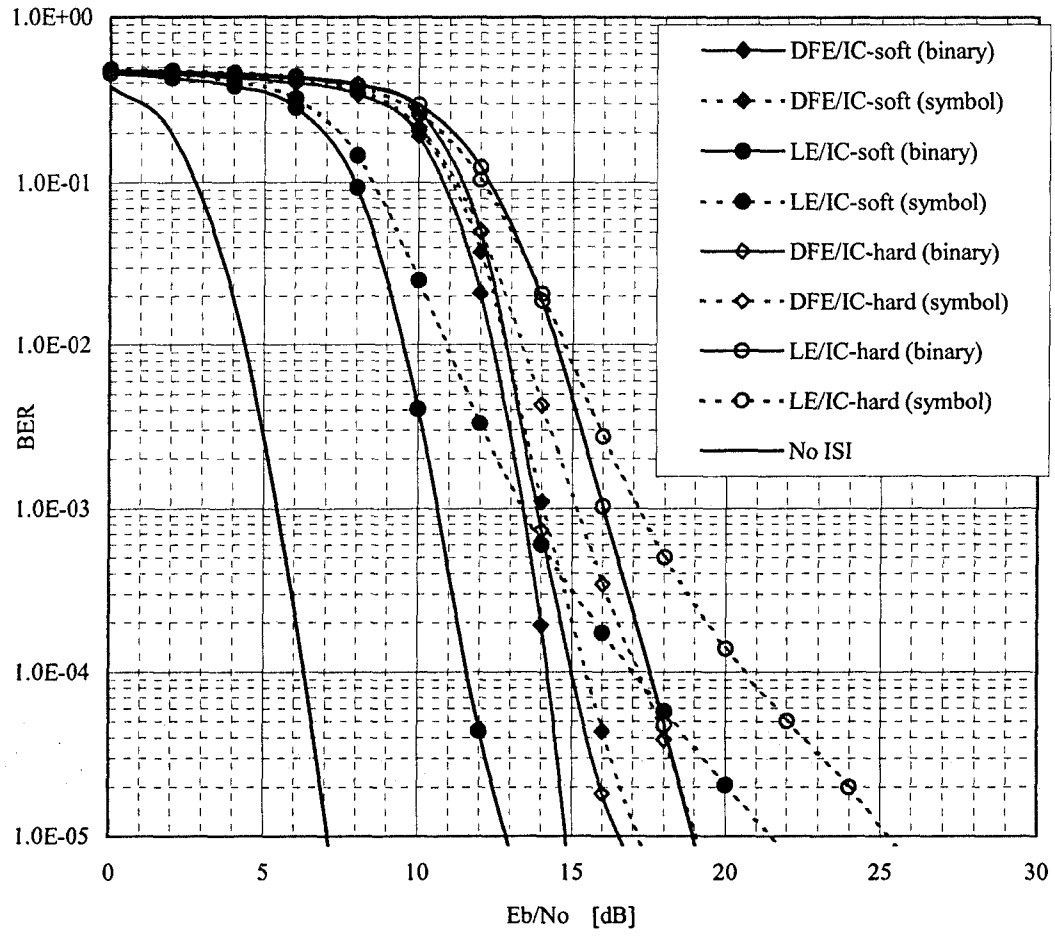



Figure 4-25 8PSK – Performance comparison of various turbo equalizers with Encoder 2 and binary-based interleaving achieved at the fifth iteration over Channel 1

# Chapter 5 Conclusions

In wireless communication, one main obstacle that we have to be solved is the intersymbol interference (ISI) caused by the multi-path distortion on the wireless channels.

In this thesis, we have investigated the performance of various turbo equalization techniques with different channel models and encoders. The idea of the turbo equalization technique is to perform the equalization and the channel decoding iteratively to improve the overall performance. As more iterations are taken, the performance of the turbo equalizer improves. In general, the turbo equalization technique can provide reliable and promising performance. With mild channel distortion, the turbo equalizer can attain the performance of the same system without ISI. On severely distorted channel, it is shown that the use of turbo equalization may be limited since they have a high loss compared to the performance of the same system without ISI. Nonetheless, they far surpass the performance of the traditional non-iterative techniques.

For the MAP-based turbo equalizer, it is combined with a MAP equalizer and a MAP decoder. Since the MAP algorithm is based on minimizing the probability of a bit error, the MAP-based turbo equalizer can provide a reliable performance. Over many channel models, the MAP-based turbo equalizer only suffers a minimum ISI penalty. However, the high computation of the MAP

algorithm leads the MAP-based turbo equalizer to be infeasible in many practical situations. If the ISI length is high, the MAP equalizer is prohibitively complex.

There is an alternative approach, namely IC-based turbo equalizer, which can reduce the computational complexity since an interference canceller is used after the first iteration. The IC-based turbo equalizer is generally divided into two categories: soft-decision feedback IC-based turbo equalizer and hard-decision feedback IC-based turbo equalizers.

The soft-decision feedback IC-based turbo equalizer is formed by a LE/DFE, an interference canceller, a de-interleaver and a MAP decoder. As the MAP decoder provides soft-decision output for turbo processing, the soft-decision feedback IC-based turbo equalizer gives reliable performance with less computational complexity comparable to the MAP-based turbo equalizer. In the coded QPSK system, the loss of the soft-decision feedback IC-based turbo equalizer compared to the MAP equalizer is often less than 1 dB. In the coded 8-PSK system, the penalty is higher than in the case of the coded QPSK system; however, substantial turbo processing gain can still be achieved.

Since the MAP algorithm is still used for channel decoding, the computational complexity of the soft-decision feedback IC-based turbo equalizer is still quite high. To minimize the

complexity, a Viterbi decoder can be used in the place of the MAP decoder, so the complexity is reduced by about a half. Since the Viterbi algorithm only generates hard-decision output, such a turbo equalizer can be referred as hard-decision feedback IC-based turbo equalizer which is consisted of a LE/DFE equalizer, an interference canceller, a de-interleaver, and a Viterbi decoder. Indeed, the loss of the hard-decision feedback IC-based turbo equalizer is higher than both MAP-based turbo equalizer and soft-decision feedback IC-based turbo equalizer but a turbo processing gain can be still achieved. When computational complexity is a crucial consideration, the hard-decision feedback IC-based turbo equalizer should be considered.

We also observe that the LE-based turbo equalizer provides better performance than the DFE-based one because the LE provides a slightly low BER at low $E_b/N_0$ than the DFE, which leads into larger improvement after turbo processing. Moreover, the LE gives a little advantage over the DFE in terms of complexity since the LE does not require an internal decision device. Thus, the LE should be used along with the IC-based turbo equalizer. In a coded 8-PSK system, binary-based interleaving is suggested if LE is chosen.

For the future research on the turbo equalization technique, time-varying channels can be considered such that the turbo equalizers can perform adaptively. Different modulation schemes

can also be considered in the turbo equalizations, such as M-ary QAM, and M-ary PAM. Also, block coding may be employed in the turbo equalizers. Moreover, in this thesis, we consider the noise component at the output of the IC is AWGN and the noise variance of $N_0$ is used; however, it may not be true since the noise component is not uncorrelated at the output. Hence, a modification of the noise variance may be required.

# References

[1] Bernard Sklar, *Digital Communications*. New Jersey: Prentice-Hall, 2001.

[2] R. W. Chang, and J. C. Hancock, "On Receiver Structures for Channels Having Memory," *IEEE Transactions on Information Theory*, vol. 12, pp. 463-468, Oct. 1966.

[3] K. Zhou, J. G. Proakis, F. Ling, "Decision-Feedback Equalization of Time-Dispersive Channels with Coded Modulation, " *IEEE Trans. Commun.*, vol. 38, pp. 18-24, Jan. 1990.

[4] C. A. Belfiore and J. H. Park, Jr., "Decision-feedback equalization," Proc. IEEE, vol. 67, pp. 1143-1156, Aug. 1979.

[5] A. Gersho and T.L. Lim, "Adaptive cancellation of intersymbol interference for data transmission," *Bell Syst. Tech. J.*, vol. 60, no.11, pp.1997-2021, Nov. 1981.

[6] M.S. Muller and J. Salz, "A unified theory of data-aided equalization," *Bell Syst. Tech. J.*, vol. 60, no.9, pp.2023-2038, Nov. 1981.

[7] M. Vedat Eyuboglu, "Detection of coded modulation signals on linear, severely distorted channels using decision feedback noise prediction with interleaving," *IEEE Trans. Commun*, vol. 36, pp.401-409, Apr. 1998.

[8] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near optimum error correcting coding and decoding: Turbo-codes," *IEEE Trans. Commun.*, vol. 44, pp.1262-1271, Oct. 1996

[9] C. Douillard, A. Picart, P. Didier, M. Jezequel, C. Berrou, and A. Glavieux, "Iterative correction of intersymbol interference: Turbo Equalization," *Eur. Trans. Telecommunications*, vol. 6, no. 5, Sept./Oct. 1995.

[10] A. Glavieux, C. Laot, and J. Labat, "Turbo-equalization over a frequency selective channel," in *Int. Symp. On Turbo-Codes*, pp. 96-102, Brest, France, Sept. 1997.

[11] J. G. Proakis, *Digital Communications*, McGraw-Hill, 2nd ed., 1989.

[12] P. Magniez, P. Duhamel, "Turbo-Equalization applied to Trellis-Coded Modulations," *IEEE VTC 1999*, pp. 2556-2560.

[13] A. Glavieux, C. Laot, and J. Labat, "Turbo Equalization: Adaptive Equalization and Channel Decoding Jointly Optimized," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 1744-1752, Sept. 2001.

[14] I. D. Marsland, P. T. Mathiopoulos, and S. Kallel, "Noncoherent Turbo-Equalization for Frequency Selective Rayleigh Fast Fading Channels," *International Symposium on Turbo Codes*, pp. 196-199, Brest, France, Sept. 1997.

[15] A. Roumy, I. Fijalkow, D. Pirez, "Joint Equalization and Decoding : why choose the iterative solution?," *IEEE 50th Vehicular Technology Conference,* pp. 2989-2993, Amsterdam, The Netherlands, September 1999.

[16] Zigang Yang and Xiaodong Wang, "Turbo Equalization for GMSK Signaling Over Multipath Channels Based on the Gibbs Sampler," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 1753-1763, Sept. 2001.

[17] Xiaodong Wang and Rong Chen, "Blind Turbo Equalization in Gaussian and Impulsive Noise," *IEEE Trans. On Vehicular Technology*, vol. 50, July 2001.

[18] I. Fijalkow, A. Roumy, S. Ronger, D. Pirez, P. Vila, "Improved Interference Cancellation for Turbo Equalization," *IEEE Int. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Turkey, June 2000.

[19] J. Nelson, R. Koetter, and A. Singer "Evolution of Prior Information in SISO Equalization," *IEEE International Symposium on Information Theory*, 2001.

[20] M. Tuechler, R. Koetter, and A. Singer, "Iterative correction of Intersymbol interference via equalization and decoding with priors," Proceedings of the 2000 International Symposium on the Information Theory, p. 100, Jun. 2000, Sorrento, Italy

# Appendix A – MAP algorithm

## Calculation of A Posterior Probability (APP)

The A Posterior Probability (APP) that message symbol $a_i$ is equal to some a∈**A** is

$$\Pr\{a_i = a | \underline{r}\}$$

where

$i \in \{0, 1, 2, 3, \ldots, N_a\text{-}1\}$

$\underline{a} = a_0, a_1, a_2, \ldots, a_{Na-1}$ $\quad a_i \in \boldsymbol{A}$

$\underline{s} = s_0, s_1, s_2, \ldots, s_{Nc}$ $\quad s_i \in \boldsymbol{S}$

$\boldsymbol{S} = \{0, 1, 2, \ldots, N_S\text{-}1\}$

$\boldsymbol{A} = \{0, 1, 2, \ldots, 2^k\text{-}1\}$

The APP can be expressed as

$$\Pr\{a_i = a | \underline{r}\} = \sum_{s \in S} \Pr\{s_i = s, a_i = a | \underline{r}\}$$

$$= \sum_{s \in S} \Pr\{a_i = a\} \underbrace{\Pr\{s_i = s | \underline{r}_0^{i-1}\}}_{\alpha_i(s)} \underbrace{\frac{\overbrace{\mu_i(SG[s,a])}^{Branch\ metric}}{f(r_i | \underline{r}_0^{i-1})}}_{\Omega_i} \underbrace{\frac{f(\underline{r}_{i+1}^{Nc-1} | s_{i+1} = ST[s,a])}{f(\underline{r}_{i+1}^{Nc-1} | \underline{r}_0^{i})}}_{\beta_{i+1}(ST[s,a])}$$

where

$\mu_i(SG[s,a]) = f(r_i | c_i = SG[s_i, a_i])$ Branch metric

$$\Pr\{a_i = a\} = \frac{1}{2^k}$$

$$\beta_{i+1}(s') = \frac{f(\underline{r}_{i+1}^{Nc-1} | s_{i+1} = s')}{f(\underline{r}_{i+1}^{Nc-1} | \underline{r}_0^{i})}$$

$i \in \{0, 1, 2, \ldots, N_a\text{-}1\}$

Initiations:

$s_0 = 0$

$s_{Nc} = 0$

## Forward Recursion:

$$\alpha_i(s) = \Pr\{s_i = s | \underline{r}_0^{i-1}\}$$

where

$s \in S = \{0, 1, 2, \ldots, Ns\text{-}1\}$

$i \in \{0, 1, 2, \ldots, Nc\}$

$Ns$ = number of states

Initialization:

$$\Pr\{s_0 = s\} = \begin{cases} 1 & \text{if } s = 0 \\ 0 & \text{if } s \neq 0 \end{cases} \quad \Rightarrow \quad \alpha_0(s) = \begin{cases} 1 & \text{if } s = 0 \\ 0 & \text{if } s \neq 0 \end{cases}$$

Assume $\alpha_i(s)$ is known for all $s \in S$

Then

$$\alpha_{i+1}(s') = \Pr\{s_i = s | \underline{r}_0^{i-1}\} \quad \text{where } s' \in S$$

$$= \sum_{s \in S} \sum_{a \in A} \Pr\{s_i = s, a_i = a, s_{i+1} = s' | \underline{r}_0^i\}$$

$$= \sum_{s \in S} \sum_{a \in A} \Pr\{s_{i+1} = s' | s_i = s, a_i = a, \underline{r}_0^i\} \Pr\{s_i = s, a_i = a | \underline{r}_0^i\}$$

**where**

$$\Pr\{s_{i+1} = s' | s_i = s, a_i = a, \underline{r}_0^i\} = \Pr\{s_{i+1} = s' | s_i = s, a_i = a\}$$

$$= \begin{cases} 1 & \text{if } s' = \mathrm{ST}[s, a] \\ 0 & \text{if } s' \neq \mathrm{ST}[s, a] \end{cases}$$

By using Bayes' Rule

$$\Pr\{s_i = s, a_i = a \,|\, \underline{r}_0^i\} = \frac{\overbrace{f(r_i \,|\, s_i = s, a_i = a, \underline{r}_0^{i-1})}^{\mu_i(SG[s,a])} \Pr\{s_i = s, a_i = a \,|\, \underline{r}_0^{i-1}\}}{\underbrace{f(r_i \,|\, \underline{r}_0^{i-1})}_{\Omega_i}}$$

$$\Pr\{s_i = s, a_i = a \,|\, \underline{r}_0^{i-1}\} = \Pr\{a_i = a\}\, P_r\{s_i = s \,|\, \underline{r}_0^{i-1}\}$$

$$= \Pr\{a_i = a\}\ \alpha_i(s)$$

$$\boxed{\alpha_{i+1}(s') = \sum_{s \in S} \sum_{a \in A} \Pr\{s_{i+1} = s' \,|\, s_i = s, a_i = a\}\ \frac{1}{\Omega_i} \mu_i(SG[s,a]) \Pr\{a_i = a\}\, \alpha_i(s)}$$

$\because$ $\alpha_{i+1}(s') = \sum_{s \in S} \sum_{a \in A} \Pr\{s_{i+1} = s' \,|\, s_i = s, a_i = a\} \frac{1}{\Omega_i} \mu_i(SG[s,a]) \Pr\{a_i = a\}\, \alpha_i(s)$

we know that

$$\sum_{s' \in S} \alpha_{i+1}(s') = \sum_{s'=S} \Pr\{s_{i+1} = s' \,|\, \underline{r}_0^i\} = 1$$

Therefore

$$\Omega_i = \sum_{s' \in S} \sum_{s \in S} \sum_{a \in A} \Pr\{s_{i+1} = s' \,|\, s_i = s, a_i = a\}\, \mu_i(SG[s,a]) \Pr\{a_i = a\}\, \alpha_i(s)$$

$$\boxed{\Omega_i = \sum_{s \in S} \sum_{a \in A} \mu_i(SG[s,a]) \Pr\{a_i = a\}\, \alpha_i(s) \qquad i \in \{0, 1, 2, \ldots, Nc\text{-}1\}}$$

## Reverse Recursion → to determine $\beta_i(s)$

$$\beta_i(s) = \frac{f(\underline{r}_i^{Nc-1} \mid s_i = s, \underline{r}_0^{i-1})}{f(\underline{r}_i^{Nc-1} \mid \underline{r}_0^{i-1})}$$

Initialization: when i = Nc

$$\beta_{N_c+1}(s') = \begin{cases} 1 & \text{if } s' = 0 \\ 0 & \text{if } s' \neq 0 \end{cases}$$

Suppose that $\beta_{i+1}(s')$ has already been computed for all $s' \in S$

$$\beta_i(s) = \sum_{a \in A} \frac{f(a_i = a, \underline{r}_i^{Nc-1} \mid s_i = s, \underline{r}_0^{i-1})}{f(\underline{r}_i^{Nc-1} \mid \underline{r}_0^{i-1})}$$

$$= \sum_{a \in A} \frac{f(\underline{r}_i^{Nc-1} \mid s_i = s, a_i = a, \underline{r}_0^{i-1})}{f(\underline{r}_i^{Nc-1} \mid \underline{r}_0^{i-1})} \underbrace{\Pr\{a_i = a \mid s_i = s, \underline{r}_0^{i-1}\}}_{\Pr\{a_i = a\}}$$

$$= \sum_{a \in A} \Pr\{a_i = a\} \underbrace{\frac{f(\underline{r}_{i+1}^{Nc-1} \mid s_i = s, a_i = a, \underline{r}_0^{i-1})}{f(\underline{r}_{i+1}^{Nc-1} \mid \underline{r}_0^{i-1})}}_{\beta_{i+1}(ST[s,a])} \underbrace{\frac{\overbrace{f(r_i \mid s_i = s, a_i = a, \underline{r}_0^{i-1})}^{\mu_i(SG[s,a])}}{f(r_i \mid \underline{r}_0^{i-1})}}_{\Omega_i}$$

$$\boxed{\beta_i(s) = \frac{1}{\Omega_i} \sum_{a \in A} \Pr\{a_i = a\} \, \mu_i(SG[s,a]) \, \beta_{i+1}(ST[s,a])}$$

## Calculation of $\Pr\{c_n = c|\underline{r}\}$

$$\Pr\{c_n = c|\underline{r}\} =$$

$$\frac{1}{\Omega_n}\mu_n(c)\sum_s\sum_a \Pr\{a_n = a\}\,\alpha_n(s)\,\beta_{n+1}(ST[s,a])\Pr\{c_n = c|\,c_n = SG[s,a]\}$$

Branch metric $\mu_n(c)$

$$\mu_n(c) \quad = \quad \frac{1}{\pi N_o}\exp\left\{\frac{-1}{N_o}\left|r_n - SM[c]\right|^2\right\}$$

$$\Pr\{c_n = c|\underline{r}\} = \sum_s\sum_a \Pr\{s_n = s, a_n = a, c_n = c|\underline{r}\}$$

$$= \sum_s\sum_a \frac{f(\underline{r}, s_n = s, a_n = a, c_n = c)}{f(\underline{r})}$$

$$= \sum_s\sum_a \frac{f(\underline{r}_0^{n-1}, s_n = s, a_n = a, c_n = c)\; f(\underline{r}_n^{N-1}|\,\underline{r}_0^{n-1}, s_n = s, a_n = a, c_n = c)}{f(\underline{r}_0^{n-1})\; f(\underline{r}_n^{N-1}|\,\underline{r}_0^{n-1})}$$

$$= \sum_s\sum_a \Pr\{s_n = s, a_n = a, c_n = c|\,\underline{r}_0^{n-1}\}\;\frac{f(r_n|\,\underline{r}_0^{n-1}, s_n = s, a_n = a, c_n = c)}{f(r_n|\,\underline{r}_0^{n-1})}$$

$$\times\frac{f(\underline{r}_{n+1}^{N-1}|\,s_n = s, a_n = a, c_n = c, \underline{r}_0^{n})}{f(\underline{r}_{n+1}^{N-1}|\,\underline{r}_1^{n})}$$

$$= \sum_s\sum_a \Pr\{s_n = s|\,\underline{r}_0^{n-1}\}\;\Pr\{a_n = a, c_n = c|\,\underline{r}_0^{n-1}, s_n = s\}\;\frac{\overbrace{f(r_n|\,c_n = c)}^{\mu_n(c=SG[s,a])}}{\Omega_n}$$

$$\times\frac{f(\underline{r}_{n+1}^{N-1}|\,s_{n+1} = ST[s,a])}{f(\underline{r}_{n+1}^{N-1}|\,\underline{r}_0^{n})}$$

$$= \sum_{s}\sum_{a} \alpha_n(s)\ \Pr\{c_n = c | s_n = s, a_n = a, \underline{r}_0^{n-1}\}\ \underbrace{\Pr\{a_n = a | s_n = s, \underline{r}_1^{n-1}\}}_{\Pr(a_n=a)}$$

$$\times \frac{\mu_n(c)}{\Omega_n} \beta_{n+1}(ST[s,a])$$

$$\Pr\{c_n = c | \underline{r}\} = \frac{1}{\Omega_n} \mu_n(c) \sum_{s}\sum_{a} \alpha_n(s)\, \beta_{n+1}(ST[s,a])\, \Pr\{c_n = c | s_n = s, a_n = a\}$$

# Appendix B – Complexity of different equalization techniques

## B.1 Interference canceller

For the IC, its output can be expressed as

$$\hat{v}_n = x_n - \sum_{k=1}^{L} q_k^* v_{n+k} - \sum_{k=1}^{L} q_k v_{n-k}$$

where

$$q_k = \sum_{l=k}^{L} f_l f_{l-k}^*$$

$$x_n = \sum_{k=0}^{L} f_k^* r_{n+k}$$

$v_n$ = the $n$-th transmitted symbol

L = ISI length

The number of adders and multipliers required for each term is shown in the following.

| | Adders | Multiplier |
|---|---|---|
| $x_n$ | $4L+2$ | $4L+4$ |
| $\sum_{k=1}^{L} q_k^* v_{n+k}$ | $4L-2$ | $4L$ |
| $\sum_{k=1}^{L} q_k v_{n-k}$ | $4L-2$ | 4L |

If all the terms are added together, 4 extra adders are required. Therefore, the total number of adders required by the IC is $12L+2$ and the number of multiplier is $12L+4$

## B.2 Linear Equalization (LE) and Decision Feedback Equalization (DFE)

For the LE, a total number of taps ($K$) has to be set and its complexity depends on the taps. The output of a linear equalizer can be expressed as

$$\hat{v}_n = \sum_{k=-K_1}^{K_2} \alpha_k r_{n-k}$$

where

$\{\alpha_m\}$ = a set of equalizer taps

$r_n$ = the n-th received sample

$K$ = $K_1 + K_2 + 1$

Therefore, the linear equalizer requires $4K$-2 adders and $4K$ multipliers.

In term of complexity, the only difference between the DFE and LE is the decision device which is used to estimate the samples from the output of the feedforward filter. If the signaling level is assumed to be $M$, $3M$ adders and $2M$ multipliers are required. In total, $4K+3M$ -2 adders and $4K+2M$ multipliers are needed for the DFE.

## B.3 MAP equalization

In the MAP equalizer, the *a posterior* probability (APP) has to be calculated for each symbol at a time. For example, at time index $n$, the APP can be expressed as

$$\Pr\{v_n = v \mid \underline{r}\} = \sum_{s \in S} \Pr\{v_n = v\} \underbrace{\Pr\{s_i = s \mid \underline{r}_0^{i-1}\}}_{\alpha_i(s)} \underbrace{\frac{\overbrace{\mu_i(v')}^{Branch\ metric}}{f(r_i \mid \underline{r}_0^{i-1})}}_{\Omega_i} \underbrace{\frac{f(\underline{r}_{i+1}^{Nc-1} \mid s_{i+1})}{f(\underline{r}_{i+1}^{Nc-1} \mid \underline{r}_0^{i})}}_{\beta_{i+1}(s_{i+1})} \tag{B-3}$$

where $\underline{r}$ is the received samples, $v$ is the transmitted symbol, $s$ is a state in the finite-state machine. With the ISI length of $L$, and $M$-ary signaling level, there are $M^L$ states in the finite-state machine.

For the term $\alpha_i$, $3M^{L+1}$ multipliers and $M^{L+1}$-1 adders are required and, for the term $\beta_i(s_i)$, it requires $2M$ multipliers and $M$-1 adders. Thus, $M^L(2M + 3M^{L+1}+3)$ multipliers and $M^L(M^{L+1}+M-1)-1$ adders are needed to calculate the APP for each transmitted symbol. Note that $\Omega_i$ can be ignored in the calculation since it is a constant. Because there are $M$ transmitted symbols and the branch metric requires $2M$ multipliers and $3M$ adders, $M^{L+1}(2M+3M^{L+1}+3)+2M$ multipliers and $M^L(M^{L+1}+M-1)+2M$ adders are required for the APP at each time instant.