COMPARISON OF ITERATIVE MAP AND SOVA RECEIVERS FOR THE FREQUENCY NON-SELECTIVE FADING CHANNEL

By

Yim Cheung Li

A Thesis Presented to the Faculty of Graduate Studies

In Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE

Department of Electrical and Computer Engineering

University of Manitoba

Winnipeg, Manitoba

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION PAGE

COMPARISON OF ITERATIVE MAP AND SOVA RECEIVERS FOR THE FREQUENCY NON-SELECTIVE FADING CHANNEL

BY

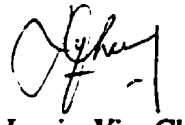
YIM CHEUNG LI

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirements of the degree

of

MASTER OF SCIENCE

I hereby declare that I am the sole author of this thesis. I authorize the University of Manitoba to lend this thesis to other institutions or individuals for the purpose of scholarly research.



Louis, Yim Cheung, Li.

I further authorize the University of Manitoba to reproduce this thesis by photocopying or by other means, in whole or in part, at the request of other institutions or individuals for the purpose of scholarly research.

Louis, Yim Cheung, Li.

ACKNOWLEDGEMENTS

My thanks first go to Professor Ed. Shwedyk for his patience and guidance throughout this thesis research. He was always willing to provide me with tremendously valuable advice and suggestions even if he had a busy time table. Special thanks also go to Dr. Hongwei Kong who offered me many ideas in the brainstorming stage of this research. I would also like to extend my sincerest gratitude to Professor G. O. Martens, Mr. Jiahuai Zhou and Mr. Ha Hoang Nguyen for their comments and helpful discussions.

Grateful acknowledgement also goes to the Natural Science and Engineering Research Council of Canada which supported this research through granting me a scholarship. Finally, I acknowledge the support and encouragement of all the members of my family and friends.

ABSTRACT

The Rayleigh frequency non-selective fading channel is a typical wireless communication channel model. When signals are transmitted over this kind of channel, the received signals experience a time-varying multiplicative gain plus the usual additive white Gaussian noise. The multiplicative gain can be modelled as a correlated Gaussian random variable.

For interleaved coded signals transmitted over fading channels, a suboptimal yet practical two stage receiver has been proposed by Vitetta and Taylor in [51]. The first stage is a channel demodulator to combat the fading distortion and the second stage is a decoder. Later research by Gertsman and Lodge in [16] has shown that error performance can be greatly enhanced when the entire detection scheme is carried out iteratively because channel equalization can benefit from the coding structure. Both demodulator and decoder should be able to deliver soft-outputs for a better error correction capability. In this thesis, two receivers, which use Bahl *et al*'s MAP algorithm [3] and Hagenauer and Hoeher's soft-output Viterbi algorithm (SOVA) [18, 24], are implemented and their error performances are compared. In the slow fading channel, the SOVA receiver exhibits negligible degradation in contrast to the MAP receiver. However, when a fast changing environment is considered the SOVA receiver is outperformed significantly. Even if more iterations are carried out on the SOVA receiver, the error performance is still not as good as the MAP receiver with one iteration. A new approach to extract soft-outputs from the standard Viterbi algorithm (VA) is also proposed and investigated. Simulation results have shown that the proposed new SOVA behaves as good as the ordinary SOVA in terms of error performance.

Complexity analysis is also made on the MAP algorithm, ordinary SOVA and the new proposed SOVA. It is found that the MAP algorithm is the most complicated method because it takes a huge amount of memory and a large number of numerical operations. The proposed SOVA is next in the number of numerical operations required but most economical in memory usage. The ordinary SOVA ranks last in computational complexity and is intermediate in memory usage.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

A/D	Analog-to-digital
AP	*A priori* probabilities
APP	*A posteriori* probabilities
AWGN	Additive white Gaussian noise
BER	Bit error rate
CIR	Channel impulse response
CPM	Continuous phase modulation
DSP	Digital signal processor
Exp	Exponential function
FIR	Finite impulse response
H-SOVA	Horizontally updated soft-output Viterbi algorithm
ICS	Ideal channel state
ISI	Intersymbol interference
LLR	Log-likelihood ratio
LVA	List Viterbi algorithm
MAP	Maximum *a posteriori*
MAP-SE	Maximum *a posteriori* sequence estimation
MAP-SS	Maximum *a posteriori* probability symbol-by-symbol detection
MLSE	Maximum likelihood sequence estimation
MMSPE	Minimum mean square prediction error

PCS	Personal communication service
PIR	Pilot insertion rate
PSAM	Pilot symbol assisted modulation
PSP	Per-survivor processing
QPSK	Quadrature-phase-shift-keying
SADD	Symbol-aided plus decision direction
SISO	Soft-input soft-output
SOVA	Soft-output Viterbi algorithm
SNR	Signal-to-noise ratio
VA	Viterbi algorithm
V-SOVA	Vertically updated soft-output Viterbi algorithm
WSS	Wide-sense-stationary
WSSUS	Wide-sense-stationary uncorrelated scattering

Chapter 1

Introduction

Nowadays, wireless technology is prevalent: from globe-spanning satellite systems to the rapidly growing Internet. As part of this trend the goal of personal communication service (PCS), which involves developing instant communication between people anywhere, will be rapidly evolving. The ever-increasing demand of PCS has increased the need for a faster, more efficient and reliable wireless communication system. To meet this goal, research in areas such as wireless network protocol design, multiple access techniques and, last and probably most importantly, schemes to detect signals reliably over the wireless channel, has been carried out.

Virtually all forms of wireless communications are becoming digital, which is constantly cited as a major reason why signals can be detected more reliably. Digital transmission is found to be capable of significantly reducing either the required transmitter power or the bandwidth of transmission, or both and yet achieve better performance than analog transmission [50]. Aided by the mature solid-state electronic technology, digital communication techniques have undoubtedly spurred wireless communication technology.

The focus of this thesis is to develop iterative receivers for reliable detection of signals transmitted within digital wireless communication systems. This chapter briefly reviews digital communication systems and existing coherent detection techniques in the wireless environment.

1.1 Digital Communications and Mobile Channels

Digital Communication usually refers to the transmission of binary integers (zeros and ones) but can be generalized to numbers of any base. However, all physical channels are analog in nature which means waveforms instead of abstract numbers are transmitted. The digital message data, whether coded or uncoded, have to modulate a carrier, typically the amplitude, frequency or phase of a sinusoid, for transmission. These digital modulation techniques are similar to analog modulation schemes except that digital modulation is done in a discrete time slot over a discrete alphabet.

The message signal, when propagated over the channel, is distorted by channel interference such as additive noise. System degradation due to interference is usually reflected by bit/symbol errors in the digital communication system. The nature of the interference is solely dependent on the characteristics of the transmission media. Wireless or so called mobile channels are characterized by a multipath phenomenon which is due to reflection, diffraction and scattering of the transmitted signal by the objects surrounding the receiver. As a result, the received signals experience effects of attenuation and possibly intersymbol interference (ISI), plus the usual additive white noise, making reception of digital signals in wireless channels very challenging. In order to achieve a better performance for the communication systems operating in wireless channels, equalization methods which combat the channel distortion is mandatory.

1.2 Existing Coherent Detection Techniques

Narrowband signals propagated in wireless channels are usually claimed to experience "frequency non-selective fading". There is another class of fading phenomenon called "frequency selective fading" which applies to the transmission of wideband signals. A detailed discussion of these concepts is given in a later chapter. The mathematical representation for the received signal in the case of non-selective fading is

$$r(t) = g(t)s(t) + n(t) \tag{1.1}$$

where $r(t)$ and $s(t)$ are the received and transmitted signals, $g(t)$ represents the channel distortion which may be viewed as a time-varying gain and $n(t)$ is the additive white Gaus-

sian noise. It is generally accepted that coherent detection is better than non-coherent detection. In order to detect the signal coherently, it is necessary to estimate the channel distortion $g(t)$. There are basically two categories of channel tracking techniques for coherent detection of narrowband signal in wireless channels [28].

Reference-based detection: A straightforward method to estimate $g(t)$ is by inserting periodically some known symbols (called pilot symbols) into the information bearing signals. By assuming perfect synchronization, the receiver can derive estimates of $g(t)$ at these time slots. These estimates can be interpolated to find the channel gain associated with the data symbols. A diagram showing the time-multiplexed data structure at a transmitter is given in figure 1.1.

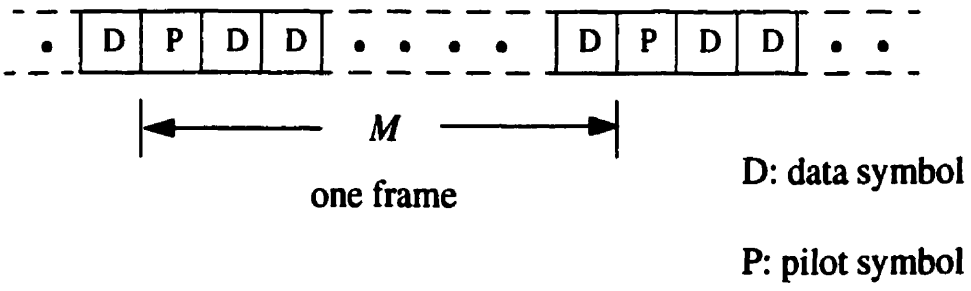


Figure 1.1: Transmission frame structure.

This channel estimation method is called pilot symbol assisted modulation (PSAM) [34] and is found to be quite effective in coherently detecting the data transmitted in the frequency non-selective wireless channel. It has been shown to provide quite robust performance over a wide range of mobile channel models and digital modulation schemes with acceptable increase in overall system complexity [2, 10, 34, 40]. Irvine and McLane [26] further improved this scheme with a version called symbol-aided plus decision directed (SADD) algorithm in which the channel gain is estimated with the help of both pilot and data symbols. The SADD scheme was found to have significant improvement over PSAM.

There exists a constraint to this reference-based detection method. The pilot insertion rate (PIR), which is interpreted as the number of data symbols between two pilot symbols, has to satisfy the Nyquist sampling theorem [8]. This means that

$$\frac{1}{MT_s} \geq 2f_d \tag{1.2}$$

where f_d is the maximum Doppler frequency which can be determined beforehand, T_s is the symbol duration and M is the length of a frame as illustrated in figure 1.1. Such a constraint imposes a severe limitation in relatively fast time-varying channels having a large f_d because a large portion of transmission power has to be allocated for pilot symbols. This results in a reduction of transmission efficiency which discourages system designers from using this method in a fast changing environment.

Memory-based detection: The memory-based detection method considers the channel estimation and signal detection jointly or as a single problem. Wireless channels, regardless whether frequency selective or non-selective, can be modelled by a trellis structure. Once such a structure is defined, existing tools designed for "solving" a trellis can be used to obtain an estimate for the transmitted symbols under some given criteria. Two well known criteria namely "maximum *a posteriori* sequence estimation" (MAP-SE) [14] and "maximum *a posteriori* probability symbol-by-symbol detection" (MAP-SS) [3] are studied in this thesis. Both are optimum but in a different sense. The former minimizes the probability of sequence error whereas the latter minimizes the symbol error rate. In [22] Hartmann and Rudolph studied the two criteria through the decoding of convolutional codes in an additive white noise (AWGN) channel and found that the two approaches provide practically the same performance measured by either bit error rate or word error rate. With respect to this example, the two criteria are equivalent.

MAP-SE is equivalent to another sequence estimation criterion "maximum likelihood sequence estimation" (MLSE) when each transmitted sequence is equiprobable. Generally speaking, the traditional algorithm designed for MLSE works for MAP-SE as well [14]. For a digital transmission and reception system, let $\vec{s}$ be a vector which represents the transmitted symbols. Also assume that a set of sufficient statistics $\vec{r}$ have been derived from the observed received signal. Consider MAP-SE first. It can be written mathematically as

$$\textbf{MAP-SE:} \qquad \vec{s}_{est} = \underset{\vec{s}(i)}{Max} Pr\{\vec{s}(i) \mid \vec{r}\} \tag{1.3}$$

where $\vec{s}(i)$ is the transmitted sequence for the i-th hypothesis and $\vec{s}_{est}$ is the estimated transmitted sequence. Equation (1.3) can be rewritten as

$$\textbf{MAP-SE:} \qquad \vec{s}_{est} = \underset{\vec{s}(i)}{Max} \frac{Pr\{\vec{r}|\vec{s}(i)\}\, Pr\{\vec{s}(i)\}}{Pr\{\vec{r}\}}$$

$$= \underset{\vec{s}(i)}{Max}\, Pr\{\vec{r}|\vec{s}(i)\}\, Pr\{\vec{s}(i)\}\,. \qquad (1.4)$$

The second equality comes from the fact that $Pr\{\vec{r}\}$ does not depend on i. $Pr\{\vec{s}(i)\}$ is referred to as the *a priori* probability of the i-th hypothesis. When the *a priori* probability (AP) is not a function of the hypothesis, i.e., every hypothesis is equally likely, (1.4) reduces to the definition of the MLSE criterion:

$$\textbf{MLSE:} \qquad \vec{s}_{est} = \underset{\vec{s}(i)}{Max}\, Pr\{\vec{r}|\vec{s}(i)\}\,. \qquad (1.5)$$

The Viterbi algorithm (VA) is an attractive method for finding the most likely path in a trellis under the criterion of MAP-SE or MLSE [14] because its complexity is proportional to the number of states in the trellis at any time instant instead of the length of the transmitted sequence. It is also regarded as a hard-output algorithm because it delivers an estimate of the transmitted sequence rather than values reflecting probabilities. Initially it was designed to decode convolutional code in memoryless noise [49]. Its application was later extended to ISI channel equalization [13] and decoding any error correction code having a trellis structure such as linear block code [56]. In order to apply the same algorithm to detect signals in mobile channels, some modifications have to be made. Basically a scheme to combat the channel disturbance has to be implemented. This can be achieved by a bank of linear prediction filters [38, 51, 52]. The prediction order can be assumed to be finite for practical purposes and this modified VA has been shown to provide promising results for different signal constellations and channel models [33, 51, 52, 58].

MAP-SS is another optimal detection strategy which aims at minimizing the probability of symbol/bit error. The concept behind it is to observe the entire received sequence $\vec{r}$ and then make a decision individually for each transmitted symbol. Mathematically,

$$\textbf{MAP-SS:} \qquad \hat{s}_k = \underset{\alpha \in Q}{Max} Pr\{s_k = \alpha | \vec{r}\} \tag{1.6}$$

where α is a member of the set of all possible symbols Q, and $\hat{s}_k$ is the estimated k-th transmitted symbol. There are two types of so called "MAP algorithm" for this purpose. Type-I MAP algorithm was proposed by Bahl *et al* [3] involving both forward and backward recursions within a trellis whereas only forward recursion is required by the type-II MAP algorithm [1]. The type-I method is more popular and has been used extensively in applications which range from decoding to wireless channel equalization [7, 16]. The MAP algorithm described in this thesis refers to the type-I method. Nevertheless, the algorithm, regardless of which type, is computationally complicated compared to VA. A simplified version is given in [39] which is always referred to as the "Max-log MAP algorithm" and can bring the complexity down to the order of VA with satisfactory performance at a high signal-to-noise ratio (SNR).

1.3 Iterative Processing

The use of iteration dates back to the development of the "turbo-code" [7]. It has been shown that such code can achieve a performance as close as 0.2 dB to Shannon's limit in an AWGN channel. Part of its success is due to the idea of iterative processing which allows the decoder in the first stage to use information obtained from the later stage. The same idea was borrowed by Gertsman and Lodge in their design of a wireless receiver [16]. For a practical receiver operating in a wireless channel, the receiver is separated into two stages [51]. The first stage is the channel demodulator which aims at combating the channel distortion and the second stage is a decoder which retrieves the information sequence. As will be mentioned in chapter 3, demodulation in the first stage has two inputs: the received samples which reflect the transmitted symbols and *a priori* information of the transmitted symbols. By applying an iterative detection strategy, the decoder can supply the *"a priori"* information back to the demodulator and hence the channel equalization can receive benefit from the coding structure. Intuitively, when more and more iterations are carried out, the feedback *"a priori"* information becomes more and more reliable. This improves the quality of the channel equalization and in turn enhances

the overall system performance. However, as more and more iterations are carried out, the gain between iterations becomes marginal.

The information circulated between the demodulator and decoder is "soft". The term "soft" refers to values which directly or indirectly reflect probabilities. In [16] Bahl *et al*'s MAP algorithm, which is optimum for the generation of soft information, was used. As mentioned, this algorithm is quite complicated. It can be replaced by the soft-output Viterbi algorithm (SOVA) [17, 18, 24] which is a modified version of the standard VA to suboptimally deliver soft information. It is of interest and important to compare the performance of different iterative wireless receivers when different detection algorithms are used.

1.4 Objectives

Even though iterative processing in a wireless channel can be done by using either the MAP algorithm or SOVA, there is no indication on how good one is compared to the other. This thesis investigates their relative performance. The author will make comparison between the two receivers based on error performance, computational complexity and the required memory. Knowing the differences in these categories will be beneficial to future receiver design for fading channels because the designer would have an idea of the trade-off between error performance and complexity.

The second goal of this thesis is related to the SOVA. In general, there are two versions of this detection method namely trace back mode and register exchange mode [20]. Originally both were designed to operate with a binary trellis. Hoeher later modified the register exchange mode so that it can function in a M-ary trellis as well [24]. Another objective of this thesis is to modify the trace back mode so it can also function in a M-ary trellis.

1.5 Outline

After this introductory chapter, the next chapter is devoted to a thorough discussion of the characteristics of wireless channels and the digital communication systems which operate in this environment. It serves as the backbone for the entire thesis and gives the

necessary detail to understand this thesis. The third chapter provides extensive coverage on the MAP algorithm and the modification required to adapt it to a mobile receiver. Chapter four examines SOVA. It first provides information on how to apply the existing algorithms to the design problem. Next it studies an alternative approach to obtain soft-outputs from the VA and its application to the same problem. With receivers developed from these three methods, a comparison is made at the end of this chapter. The last chapter, chapter five, sums up the entire thesis and gives recommendations for future studies.

Chapter 2

Digital Communications in Wireless Channels

In order to design receivers that perform effectively in wireless or mobile channels, system designers must understand the nature of the receiver's operating environment. The wireless communication channel, which is also known as a multipath fading channel, is characterized by the time-varying nature of the transmitting medium. In general, the physical medium of transmission changes continuously and these changes occur randomly. Thus, it is reasonable to view the time-varying channel as a random process. The characterization of this random process is discussed in the first section of this chapter.

Next a discussion of digital communication systems operating in a wireless environment follows. Information on transmitter and receiver models are given. In brief, this chapter serves as a background for the entire thesis.

2.1 Multipath Phenomena

A multipath fading channel is simply one where electromagnetic energy arrives via different paths, usually as a result of reflections or inhomogeneities in the physical medium that produce ray-splitting or scattering effects [45]. The scenario can be depicted as in figure 2.1 where the transmitted signal is reflected or scattered by objects such as buildings or trees. Hence the electromagnetic waves arriving at the mobile receiver are not only attenuated, but may also be shifted in frequency and have different delays. To make matters worse, differences in arrival times may even cause ISI which further degrades the receiver performance. To examine the behaviour of the channel, let $s(t)$ be the bandpass transmitted signal written as

$$s(t) = Re\{u(t)e^{j2\pi f_c t}\} \tag{2.1}$$

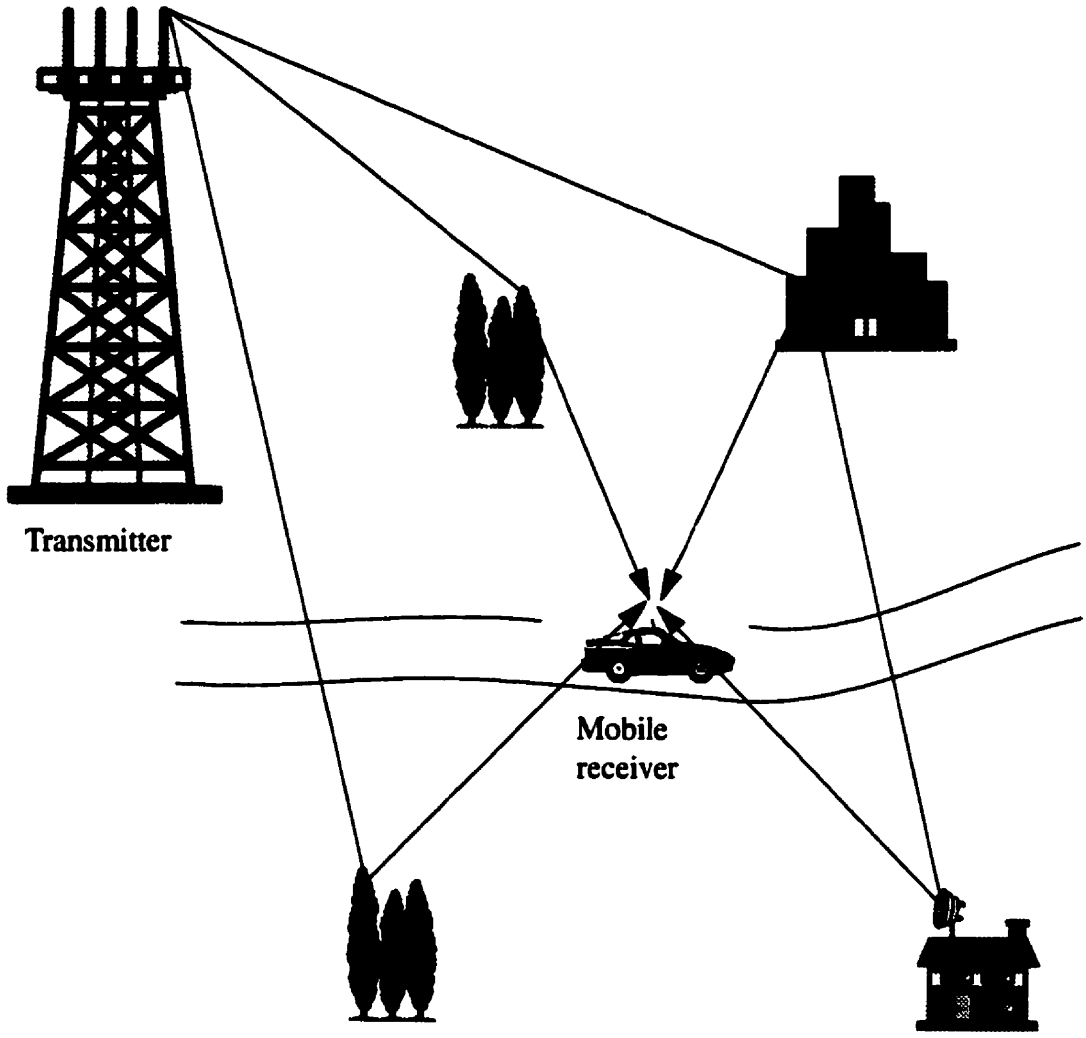


Figure 2.1 Wireless mobile communication environment.

where $u(t)$ is the complex baseband transmitted signal and f_c is the carrier frequency. Consider a situation where the mobile unit (the car in the picture) receives N waves. The n-th wave, which comes at an incidence angle $\theta_n(t)$ with respect to the direction of travel, experiences a Doppler shift and is shown in figure 2.2. The Doppler shift of this wave is

$$f_{d,n}(t) = \frac{vf_c}{c}\cos\theta_n(t) \tag{2.2}$$

where c is the speed of light and v is the instantaneous velocity of the mobile receiver.

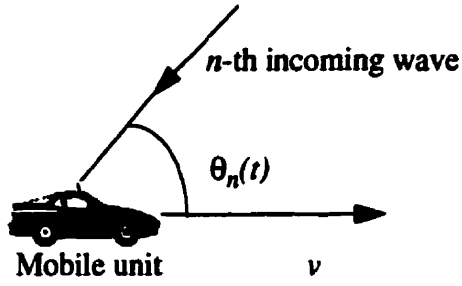


Figure 2.2 A typical wave component incident on a mobile receiver.

In addition to the Doppler shift, the multipath fading phenomenon introduces attenuation and delay to each wave. The mobile receiver picks up many waves from different directions and thus, by using superposition, the received signal can be expressed as:

$$x(t) = Re\left\{ \sum_{n=1}^{N} \alpha_n(t)\, u[t-\tau_n(t)]\, e^{j2\pi [f_c + f_{d,n}(t)]\,[t-\tau_n(t)]} \right\} \tag{2.3}$$

where $\alpha_n(t)$ and $\tau_n(t)$ are the attenuation factor and propagation delay of the received signal on the n-th path. The equivalent baseband received signal can be represented by

$$r(t) = \sum_{n=1}^{N} \alpha_n(t)\, u[t-\tau_n(t)]\, e^{j2\pi \{f_{d,n}(t)\, t - [f_c + f_{d,n}(t)]\,\tau_n(t)\}} . \tag{2.4}$$

As $r(t)$ is the channel output when the input is $u(t)$, the channel impulse response (CIR) is

$$c(t;\tau) = \sum_{n=1}^{N} \alpha_n(t)\, \delta[t-\tau_n(t)]\, e^{j2\pi \{f_{d,n}(t)\, t - [f_c + f_{d,n}(t)]\,\tau_n(t)\}} \tag{2.5}$$

where $c(t;\tau)$ is the response of the channel at time t due to an impulse applied at time t-τ. In general, the attenuation factor, delay and Doppler shift are time variant and unpredictable so that the CIR is reasonably modelled as a random process [37]. When N is large, $c(t;\tau)$ can be modelled as a complex Gaussian random process by applying the central limit theorem. The first order statistics of the CIR is characterized by its amplitude and phase distributions which depend on the transmission medium [53]. In urban areas the direct line of sight between the mobile unit and transmitter is completely obstructed. In this case the electromagnetic energy propagation is largely due to scattering and the mobile unit picks up reflected waves from all directions. The envelope of the received sig-

nal usually exhibits a Rayleigh distribution and the phase is distributed uniformly over $[0,2\pi)$ [45]. A wireless fading channel is typically named after its envelope distribution and thus the one described above is termed a Rayleigh fading channel. There exists other kinds of fading channel such as Rician and lognormal. The former applies to the channel having a direct line-of-sight to the mobile receiver whereas the latter refers to those having a shadowing effect. The Rayleigh fading channel is the model considered in this thesis.

For the Rayleigh fading channel, $c(t;\tau)$ is a zero-mean complex Gaussian process. Assume that it is wide-sense-stationary (WSS). Then its autocorrelation function is

$$\phi_c(\tau_1, \tau_2;\Delta t) = \mathcal{E}\{c^*(t;\tau_1)\, c(t+\Delta t;\tau_2)\} \tag{2.6}$$

where $(\cdot)^*$ denotes the complex conjugate and $\mathcal{E}\{\cdot\}$ is the expectation operator. Suppose the attenuation factor and phase shift associated with any two paths with delays τ_1 and τ_2 are uncorrelated. The random process becomes a wide-sense-stationary uncorrelated scattering (WSSUS) process and (2.6) can be modified to

$$\begin{aligned}\phi_c(\tau_1, \tau_2;\Delta t) &= \mathcal{E}\{c^*(t;\tau_1)\, c(t+\Delta t;\tau_2)\} \\ &= \mathcal{E}\{c^*(t;\tau_1)\, c(t+\Delta t;\tau_1)\}\,\delta(\tau_1-\tau_2) \\ &= \phi_c(\tau_1;\Delta t)\,\delta(\tau_1-\tau_2)\,. \end{aligned} \tag{2.7}$$

With $\Delta t = 0$, the function $\phi_c(\tau_1;0) \equiv \phi_c(\tau)$, which is called "multipath intensity profile" or "delay power spectrum", is obtained. It gives the average power of the impulse response as a function of τ. Figure 2.3(a) shows a typical shape of this function [37]. The range of values for τ for which $\phi_c(\tau)$ is essentially non-zero is called the "multipath spread" and is denoted by T_m. When the information symbol interval T_s, is much larger than T_m, the channel is characterized as "frequency non-selective". Otherwise, it falls into the category of "frequency selective" channel.

The same argument can be made in the frequency domain. Let $\Phi_c(\Delta f)$ be the Fourier transform of $\phi_c(\tau)$, then $\Phi_c(\Delta f)$ provides a measure of the frequency coherence of the

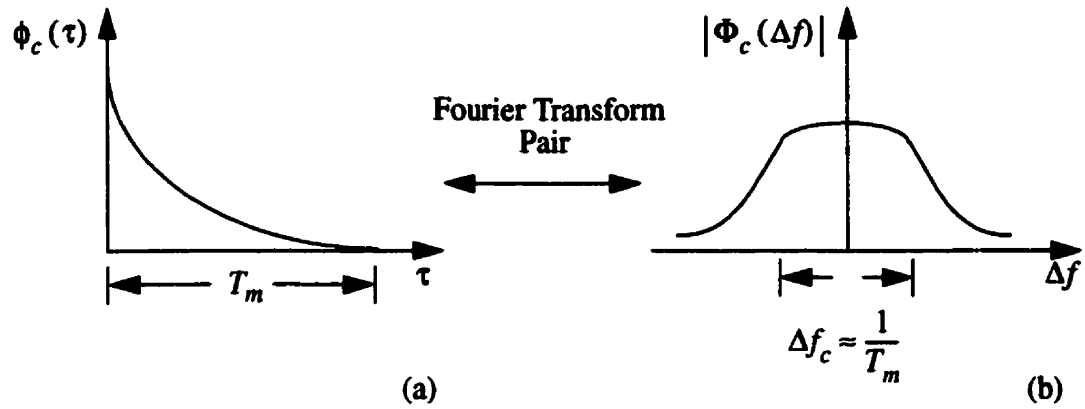


Figure 2.3 (a) Multipath intensity profile and (b) magnitude of the Fourier transform of multipath intensity profile.

channel. The range of values of f for which $\Phi_c(\Delta f)$ does not change significantly is called the "coherence bandwidth". When the transmitted signal has a bandwidth smaller than this coherence bandwidth, the signal is classified as "narrowband" and the channel behaves as a frequency non-selective channel. Otherwise, the signal is viewed as "wide-band" and the channel is frequency selective. Figure 2.3(b) shows an example of this function.

The time variation of the channel is evidenced by the Doppler effect. It can be studied by looking at the two-dimensional Fourier transform of $\phi_c(\tau;\Delta t)$, viz

$$S_c(\lambda;\Delta f) = \int_{-\infty}^{\infty} e^{-j2\pi\lambda\Delta t} d\Delta t \int_0^{\infty} \phi_c(\tau;\Delta t)\, e^{-j2\pi\Delta f\tau} d\tau. \qquad (2.8)$$

By setting $\Delta f = 0$, the function $S_c(\lambda;0) \equiv S_c(\lambda)$, which is called the "Doppler power spectrum" of the channel, is obtained. Figure 2.4 illustrates how $S_c(\lambda)$ may look. It shows the average power output as a function of Doppler frequency λ. The range of values over which $S_c(\lambda)$ is essentially non-zero is called the "Doppler spread". The latter is denoted by B_d. The reciprocal of B_d is a measure of the "coherence time"

$$(\Delta t)_c \approx \frac{1}{B_d} \qquad (2.9)$$

which is an indication of how fast the channel changes. In general, if the coherence time is much larger than the transmitted symbol duration, the symbol will experience a slow

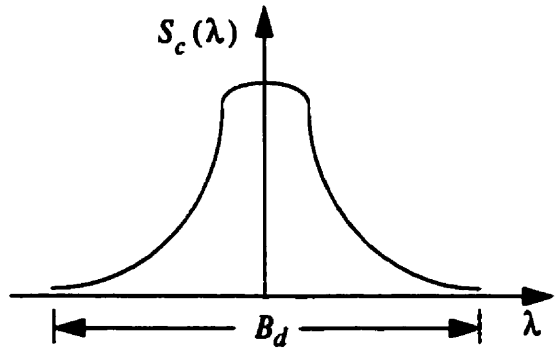


Figure 2.4 Doppler power spectrum.

change in channel parameters over time and the channel is slow fading. Otherwise, the channel is a fast fading channel.

As a short summary, a fading channel can be classified with the help of the multipath spread (T_m) and coherence time $(\Delta t)_c$, into

- Frequency non-selective ($T_m << T_s$); slow fading ($(\Delta t)_c >> T_s$)
- Frequency non-selective ($T_m << T_s$); fast fading ($(\Delta t)_c << T_s$)
- Frequency selective ($T_m >> T_s$); slow fading ($(\Delta t)_c >> T_s$)
- Frequency selective ($T_m >> T_s$); fast fading ($(\Delta t)_c << T_s$)

In this thesis, the slow frequency non-selective fading channel in considered. A frequency non-selective fading channel can be ISI free with a properly chosen signalling scheme. The same does not apply to frequency selective fading because the transmitted signal experiences cut off in this kind of channel. Next, the frequency non-selective channel will be explored to provide a mathematical representation for the received signal transmitted in this environment.

2.2 Mathematical Models

The definition of the CIR leads one to treat the channel as a time-varying filter with a baseband impulse response $c(t;\tau)$. The received baseband signal, $r(t)$, can be written as a convolution of the input signal $s(t)$ and $c(t;\tau)$

$$r(t) = \int_{-\infty}^{\infty} c(t;\tau)\, s(t-\tau)\, d\tau + n(t) \tag{2.10}$$

where $n(t)$ is usual additive complex white Gaussian noise. For the frequency non-selective fading channel, (2.10) can be simplified by replacing $c(t;\tau)$ with $g(t)\,\delta(\tau)$ [12]: i.e.

$$r(t) = g(t)\, s(t) + n(t)\,. \tag{2.11}$$

The channel thus distorts the transmitted signal with a time-varying multiplicative gain. As stated previously, $c(t;\tau)$, which is now reflected by $g(t)$, is a zero-mean Gaussian random process in a Rayleigh fading channel. The description of the random process will be completed once its variance is obtained.

Assume the mobile vehicle receives a uniform power over the range of angles $[0,2\pi)$ and each incoming wave is independent of the others. This is commonly referred to as an isotropic scattering model [27, 53]. If a sinusoidal signal is transmitted over this specific medium, the received baseband signal, $r(t)$, will have a power spectrum of (neglecting white noise)

$$S(f) = \begin{cases} \dfrac{\sigma_d^2}{2\pi f_d}\left[1-\left(\dfrac{f}{f_d}\right)^2\right]^{-\frac{1}{2}} & \text{for } |f| < f_d \\ 0 & \text{otherwise} \end{cases} \tag{2.12}$$

for both the in-phase and quadrature components [46]. This spectrum is commonly called the "Jakes' spectrum" [25]. In this equation, σ_d^2 is a known constant and f_d is the maximum Doppler frequency whose value can be calculated by using

$$f_d = \frac{v}{c} f_c\,. \tag{2.13}$$

This relationship is the same as (2.2) with $\theta_n(t)$ set to zero. The Jakes' spectrum is plotted in figure 2.5 with f_d and σ_d^2 both set to one. The autocorrelation function follows by taking the inverse Fourier transform of (2.12) which results in

$$\phi(\tau) = \frac{\sigma_d^2}{2} J_0(2\pi f_d|\tau|) \tag{2.14}$$

where $J_0(\bullet)$ is the zero-order Bessel function of the first kind. Both the in-phase and

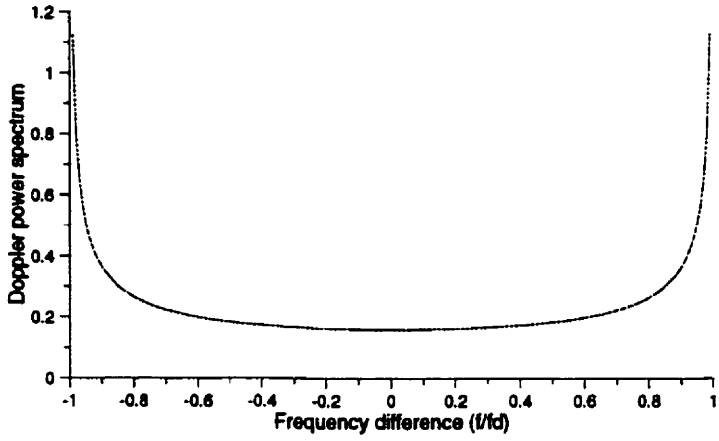


Figure 2.5 Power spectrum of the received baseband signal for an isotropic scattering channel.

quadrature components of $r(t)$ have their autocorrelation functions equal to (2.14) and they are also independent of one another because of the isotropic scattering model [46]. The autocorrelation function of $r(t)$ is

$$\phi_r(\tau) = \mathcal{E}\{r^*(t)\,r(t+\tau)\} = \sigma_d^{\,2} J_0(2\pi f_d|\tau|)\,. \qquad (2.15)$$

This is also the autocorrelation of $g(t)$ which appears in (2.11). Knowing the autocorrelation function is very important because the fading channel disturbance has to be generated when doing a simulation of a wireless system. With this knowledge the fading process can be obtained by passing white noise through a linear filter having a transfer function derived from this statistical information. This completes the description of the fading process.

2.3 Digital Communication Systems over Fading Channels

Digital communication systems nowadays are becoming more and more complex. Nevertheless, the systems can still be generally separated into three major components: namely a transmitter, receiver front-end processor and receiver. In the AWGN channel, the second component is always included as part of the receiver but in mobile channel usage, they are intentionally separated. This section describes the structure of each block and the

model of every unit used in this thesis will also be addressed.

2.3.1 Transmitter Model

In a digital communication system there exists a source which emits information in binary form with some known *a priori* probabilities. These bit sequences may represent the quantized output of an analog-to-digital (A/D) converter or some data outputs from a personal computer. Channel coding is usually credited for the superiority of digital transmission over analog transmission. It is basically a mapping that accepts information bits and adds redundancy for error correction. The addition of redundancy in the coded message implies that an increase in the transmission bandwidth is required so that the overall system becomes more complicated.

Regardless of whether the system is coded or uncoded, the bit stream has to be mapped onto a selected signal constellation for transmission. Among the many existing M-ary signalling schemes, quadrature-phase-shift-keying (QPSK) as shown in figure 2.6 with Gray coding is chosen in this thesis. It has the property of being constant amplitude which, as will be seen in chapter 3, helps in simplifying the metric calculation in the receiver. The QPSK symbols are then fed into a shaping filter with an impulse response $f(t)$. For narrowband signals, $f(t)$ can be chosen to satisfy the Nyquist criterion to achieve no ISI [37]. Without loss of generality, assume $\int_{-\infty}^{\infty} |f(t)|^2 dt = 1$. The resulting baseband signal is

$$s(t) = \sum_{k=1}^{N} a_k f(t - kT_s) \tag{2.16}$$

where a_k is a M-ary alphabet, T_s is the symbol interval and N symbols in total are assumed to be transmitted. This baseband signal is transmitted over the channel.

A coded system is highly recommended in mobile communications. With fading channels the coding gain is generally higher than the gain in an AWGN channel [19]. Coded systems in a slowly time-varying channel always employ interleaving, which is a mapping to scramble the bits/symbols. In this type of channel, the error events occur in long bursts, as illustrated in figure 2.7, rather than randomly. With a large enough interleaver size, the

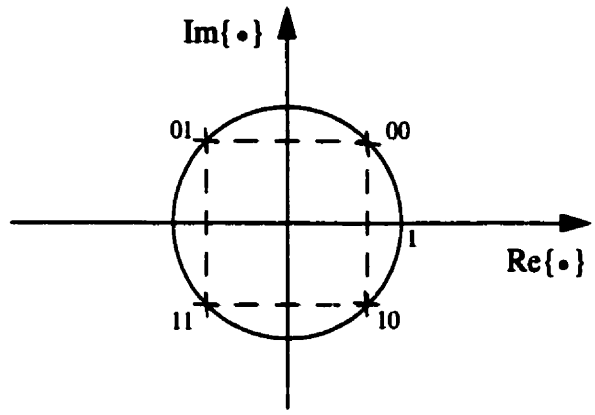


Figure 2.6 QPSK signal constellation.

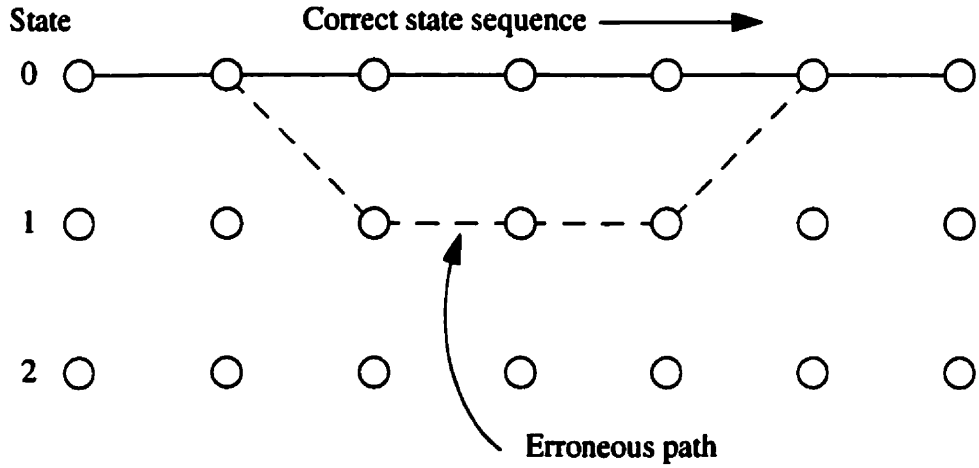


Figure 2.7 A typical error pattern in a trellis.

inverse operation of interleaving, called deinterleaving, can randomize these error events. Hence even a coding scheme with a short memory can achieve a satisfactory performance. Figure 2.8 demonstrates an n x m block interleaver structure [37] which is the interleaving scheme used in this thesis. This particular interleaving scheme is chosen because it is simple and easy to implement in simulation.

Both coded and uncoded systems are considered in this thesis. The coding scheme employed is a rate 1/2 convolutional code with constraint length five and generator polynomial

$$G(D) = \left[1 + D + D^3 + D^5, 1 + D^2 + D^3 + D^4 + D^5\right] \tag{2.17}$$

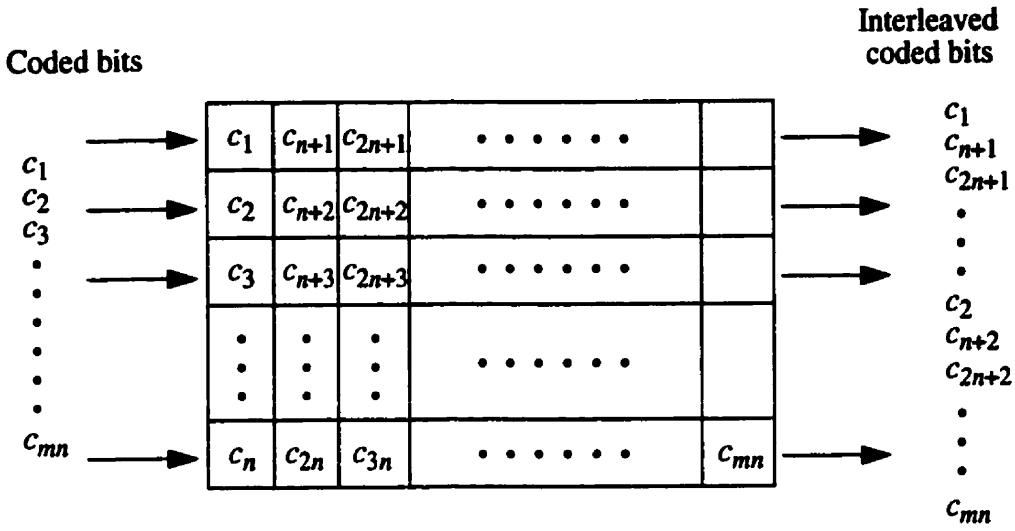


Figure 2.8 An n x m block interleaver.

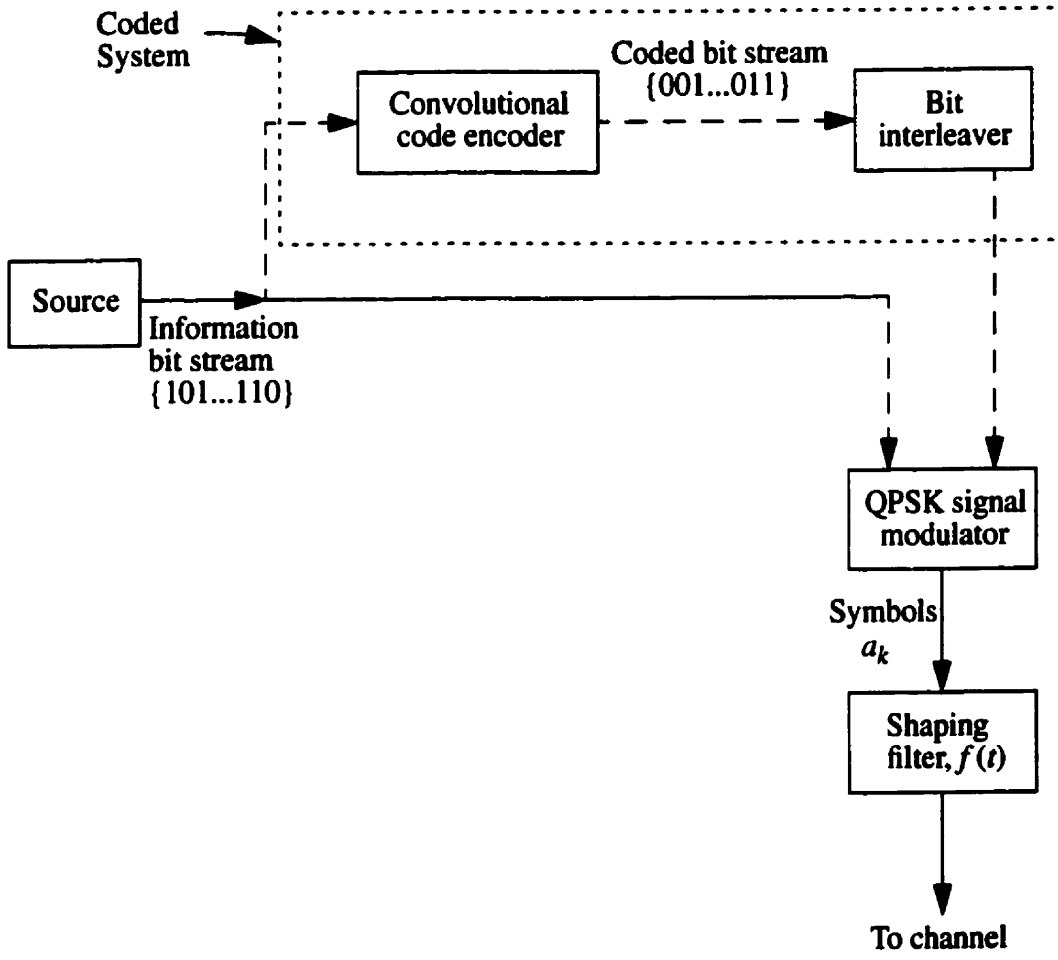


Figure 2.9 Coded and uncoded transmitter model.

where D can be interpreted as a delay operator in the encoder [29]. Thus, for every information bit that enters the encoder, two coded bits come out. If bit interleaving is used, each one of these two bits will join with some other bit to form a QPSK symbol. If symbol interleaving is used, they group together to form a symbol. Conventionally, bit interleaving spreads the effect of an erroneous symbol while symbol interleaving concentrates the error into a node in demodulation [15]. The system in this thesis chooses bit interleaving.

After mapping the coded bits onto the QPSK alphabet, the symbols are fed into the shaping filter as discussed. The resultant signal is given in (2.16). For the uncoded system, a_k is formed by information bits and, for a coded system, it is built from coded bits. A block diagram showing the transmitter model is provided in figure 2.9. Actually this is just a preliminary picture of the transmitter. A few more features will be added but they are postponed until the next chapter.

2.3.2 Front-End Processor

For a digital communication system, a front-end processor is always placed as the first stage of the receiver in order to obtain a set of sufficient statistics [58]. One example of such a processor is shown in figure 2.10. If $f(t)$ and $c(t;\tau)$ are both bandlimited, then the fading channel output is also bandlimited [12]. Accordingly, the received signal can be low-pass filtered without being distorted. Sufficient statistics can be obtained next from the filtered output using by Nyquist sampling.

Another procedure, called the matched-filtering method, is assumed in this thesis. Its block diagram is given in figure 2.11. For the frequency non-selective channel, by combining (2.11) and (2.16) the received signal is expressed as

$$r(t) = g(t)s(t) + n(t) = \sum_{k=1}^{N} a_k g(t) f(t - kT_s) + n(t) . \qquad (2.18)$$

When the channel changes slowly, the fading process $g(t)$ can be approximated by a complex constant g_k during a symbol interval $[(k\text{-}1)T_s, kT_s]$. However, it may change between successive intervals [9]. The received signal can then be represented as

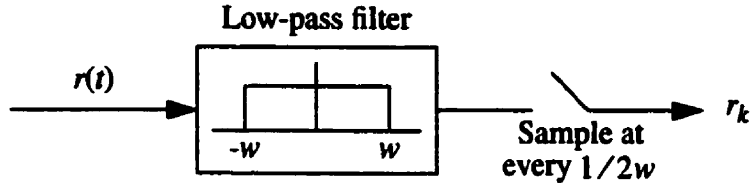


Figure 2.10 Low-pass filtering front-end processor.

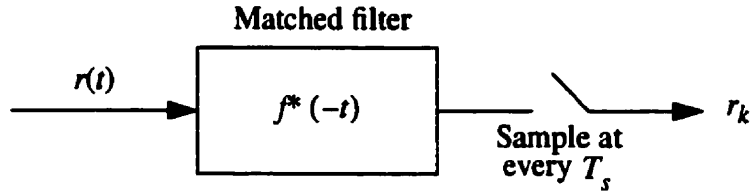


Figure 2.11 Matched-filtering front-end processor.

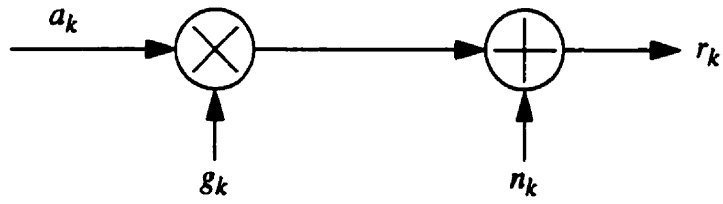


Figure 2.12 Discrete model for frequency non-selective fading channel.

Figure 2.13 Two stage receiver model.

$$r(t) = \sum_{k=1}^{N} a_k g_k f(t - kT_s) + n(t) . \qquad (2.19)$$

This signal is passed through a matched filter with an impulse response $f^*(-t)$. If $f(t)$ is chosen to satisfy the Nyquist criterion for zero ISI (which is the case here), the output can be sampled at every symbol interval to produce a set of sufficient statistics of the received continuous signal with the k-th sample written as

$$r_k = g_k a_k + n_k \qquad (2.20)$$

where $\{n_k\}$ is a set of complex white Gaussian noise samples. An equivalent discrete channel model which is implied by (2.20) is drawn in figure 2.12.

2.3.3 Demodulator and Decoder

The channel demodulator/equalizer is an integral part of the wireless receiver as it functions to combat the distortion on the received signal imposed by the fading process whose statistical characteristics are assumed to be known. Among the coherent detection strategies mentioned in chapter 1, the memory-based detection method is chosen in this thesis. The fading process is modelled as a channel with memory due to its strong correlation over time and therefore can be represented by a trellis. In order to perform coherent detection, the fading distortion has to be estimated. By using per-survivor processing (PSP) [38], the channel estimate can be evaluated on the basis of symbols belonging to the survivor path within the trellis. With these channel estimates, conventional methods such as the symbol-by-symbol MAP algorithm and the VA can be employed in the channel demodulator. More details about demodulation are given in the next two chapters.

For a coded interleaved transmission system, the two stage receiver structure shown in figure 2.13 was proposed in [28, 51]. The first stage channel demodulator tries to compensate for the fading distortion without considering any coding structure. The output of this stage can be either "hard" or "soft". Hard-output means that the demodulator makes a preliminary decision on the transmitted symbol. Soft-output refers to values which reflect the *a posteriori* probabilities (APPs) of these symbols. After the deinterleaver, the second stage channel decoder utilizes the deinterleaved soft or hard output plus the knowledge of

coding structure to make an estimation of the transmitted information sequence. It is well known that if the first stage delivers soft-output, the decoding process can achieve a larger coding gain [3, 18, 23, 24]. The focus of the following chapters in this thesis is on soft-output channel equalization techniques.

2.4 Fading Simulator

The performance of the receivers to be discussed are determined by computer simulations. Carrying out simulations of digital signalling over wireless channels requires generating the fading process $g(t)$ or equivalently its samples $\{g_k\}$. The in-phase and quadrature components of a complex g_k have the Doppler spectrum given in (2.12). In order to generate samples having the autocorrelation function (2.14), a filter with the following impulse response is required:

$$h(t) = \frac{1}{\sqrt{2}}\pi^{-\frac{1}{4}}\Gamma\left(\frac{3}{4}\right)f_d^{\frac{1}{4}}|t|J_{\frac{1}{4}}(2\pi f_d|t|)$$

and

$$\lim_{t\to 0} h(t) = \frac{1}{\sqrt{2}}\frac{\Gamma\left(\frac{3}{4}\right)f_d^{\frac{1}{2}}}{\Gamma\left(\frac{5}{4}\right)} \tag{2.21}$$

where $\Gamma(\cdot)$ is the Gamma function, $J_{\frac{1}{4}}(\cdot)$ is the 1/4-th order Bessel function of the first kind and it is assumed that σ_d^2 in (2.12) equals one. To generate discrete samples g_k, a finite impulse response (FIR) filter can be employed to approximate (2.21) with Nyquist sampling. In this thesis a 1001-tap filter is used. By passing white Gaussian noise samples with unit variance through two such FIR filters, the complex fading process samples can be obtained. The block diagram of the fading simulator is depicted in figure 2.14.

There are methods to simplify the FIR structure so that a filter with fewer taps can be used [55]. However, they are beyond the scope of this thesis.

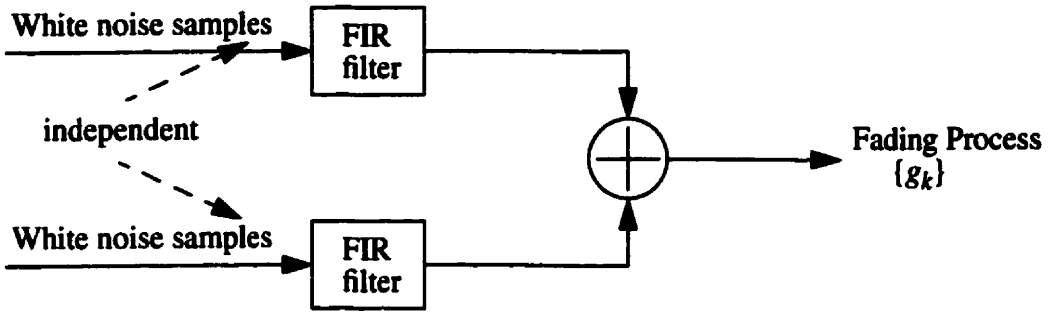


Figure 2.14 Fading process generator.

2.5 Summary

In this chapter, background information on fading channels has been given. The fading channel examined in this thesis is a Rayleigh fading channel which can be modelled as a zero-mean complex Gaussian random process with the autocorrelation function (2.15). Samples of this fading process can be generated by passing white noise with unit variance through two identical FIR filters with their impulse responses derived from the Doppler spectrum. Also presented in this chapter are the transmitter and receiver models. A convolutional code with the generator polynomial (2.17), plus bit interleaving, is the coded transmitter model. Regardless of whether a coded or uncoded system is considered, the bit data stream is mapped to QPSK symbols for transmission. At the receiver, the corrupted continuous received signal is first passed through a matched-filter to obtain a set of sufficient statistics. When the uncoded data sequence is transmitted, the receiver has one single stage only, called the channel demodulator, to compensate the fading distortion. The demodulator models the channel as a trellis so that the symbol-by-symbol MAP algorithm and the VA can be used. For the coded system the receiver becomes two stages which are the channel demodulator followed by the convolutional decoder. The succeeding chapters are devoted to soft-output channel demodulation and decoding techniques such as the MAP algorithm and SOVA. As will be seen in these chapters, the channel modelling and front-end processing are important because without them, it is almost impossible to apply these tools to the detection problem.

Chapter 3

Symbol-by-Symbol MAP Processing

Symbol-by-symbol detection for the fading channel was first discussed in [43]. Though the symbol-by-symbol MAP algorithm was originated by Bahl *et al* in 1974 [3], it did not draw much attention until the development of the turbo-code [7]. Gertsman and Lodge applied the same algorithm to iteratively detect continuous phase modulation (CPM) and QPSK signals in the frequency non-selective fading channel [16]. This chapter also considers the problem of detecting QPSK signals transmitted over the frequency non-selective channel. Receivers for both coded and uncoded systems are considered and the concepts of iterative processing are also explored. The simulation results in this chapter serve as a benchmark when it comes to comparison with the SOVA receiver in chapter four.

3.1 Symbol-by-Symbol MAP Demodulation

Communication systems in a wireless environment require channel equalization at the receiver for reliable information exchange. This section describes a MAP equalizer and its algorithmic implementation for channel demodulation. As will be seen, this is not straightforward and additional features have to be added to the transmitter model so that the demodulator can work properly.

3.1.1 MAP Demodulation

The received signal transmitted through a frequency non-selective fading channel can be expressed in discrete form as

$$r_k = g_k a_k + n_k \tag{3.1}$$

where g_k represents the fading process and is a complex zero-mean Gaussian random var-

iable with known autocorrelation function, a_k is the transmitted QPSK symbol with unit energy and n_k is a sample of the additive complex white Gaussian noise process with variance N_o. Assume N symbols are sent and define the following:

h : index of the hypothesis.

$\vec{R}$: N-vector of received samples, $\{r_1, r_2, \ldots, r_N\}$.

$\vec{A}(h)$: N-vector of transmitted symbols for the h-th hypothesis, $\{a_1(h), a_2(h), \ldots, a_N(h)\}$.

$\hat{A}$: N-vector of the estimated transmitted symbols, $\{\hat{a}_1, \hat{a}_2, \ldots, \hat{a}_N\}$

Consider the *a posteriori* probability (APP) of the h-th hypothesized symbol sequence

$$Pr\{\hat{A} = \vec{A}(h) | \vec{R}\} = \frac{f\{\vec{R} | \hat{A} = \vec{A}(h)\} Pr\{\hat{A} = \vec{A}(h)\}}{f\{\vec{R}\}}. \tag{3.2}$$

Assume that the QPSK symbols are independent of each other. The second term in the numerator of (3.2) becomes

$$Pr\{\hat{A} = \vec{A}(h)\} = \prod_{k=1}^{N} Pr\{\hat{a}_k = a_k(h)\}. \tag{3.3}$$

For the first term in the numerator, consider a linear transformation

$$\vec{Y}(h) = \vec{R}\, T\left[\vec{A}(h)\right] \tag{3.4}$$

where $T\left[\vec{A}(h)\right]$ is a diagonal matrix with the i-th element being $a_i^*(h)$. It has been shown in [30] that, for a constant amplitude signalling scheme with unit energy, the following equality of probability density functions

$$f\{\vec{R} | \hat{A} = \vec{A}(h)\} = f\{\vec{Y}(h) | \hat{A} = \vec{A}(h)\} \tag{3.5}$$

is valid and that the covariance of the density function on the right hand side does not rely on which hypothesized sequence is transmitted. By successive application of Bayes' rule the density function becomes

$$f\{\vec{Y}(h)\,|\hat{A}=\vec{A}(h)\} = \prod_{k=1}^{N} f\{y_k(h)\,|\left(y_{k-1}(h),\ldots,y_2(h),y_1(h),\vec{A}(h)\right)\} \quad (3.6)$$

where $y_k = r_k a_k^*(h)$. It is given in [30, 51] that

$$f\{y_k(h)\,|\left(y_{k-1}(h),\ldots,y_2(h),y_1(h),\vec{A}(h)\right)\}$$

$$= \frac{1}{\pi e_o v_{k-1}} \exp\left\{-\frac{1}{e_o v_{k-1}}\left|\sum_{n=0}^{k-1} b_n^{k-1} r_{k-n} a_{k-n}^*(h)\right|^2\right\} \quad (3.7)$$

where b_j^i is the j-th coefficient of the i-th order minimum mean square error linear predictor for $y_k(h)$ given previous observations with $b_0^i = -1$, v_i is the normalized squared prediction error for the i-th order linear prediction and

$$e_o = \mathcal{E}\{g_k^* g_k\} + \mathcal{E}\{n_k^* n_k\} = \sigma_d^2 + N_o. \quad (3.8)$$

Equation (3.7) describes a system whose conditional probability depends on the entire history of the transmitted data. In practice, it is reasonable to assume that the "memory" of the system is finite. As a result, the folding condition as described in per-survivor processing (PSP) holds [11] and

$$f\{y_k(h)\,|\left(y_{k-1}(h),\ldots,y_2(h),y_1(h),\vec{A}(h)\right)\}$$

$$= f\{y_k(h)\,|\,(y_{k-1}(h),\ldots,y_{k-L}(h),a_k(h),a_{k-1}(h),\ldots,a_{k-L}(h))\} \quad (3.9)$$

where L is the memory length of the system. This means that the present observation depends only on L previous observations. Under this condition, (3.7) can be rewritten as

$$f\{y_k(h)\,|\,(y_{k-1}(h),\ldots,y_{k-L}(h),a_k(h),a_{k-1}(h),\ldots,a_{k-L}(h))\}$$

$$= \frac{1}{\pi e_o v_L} \exp\left\{-\frac{1}{e_o v_L}\left|\sum_{n=0}^{L} b_n^L r_{k-n} a_{k-n}^*(h)\right|^2\right\}. \quad (3.10)$$

This last equation suggests a finite state machine model with L symbols defining one state. The "partial branch weight" for transition between states $m' = (a_{k-1}, a_{k-1}, \ldots, a_{k-L})$ and $m = (a_k, a_{k-1}, \ldots, a_{k-L+1})$ at time k is given by

$$W_k(m',m) = \frac{1}{\pi e_o v_L} \exp\left\{ -\frac{1}{e_o v_L} \left| \sum_{n=0}^{L} b_n^L r_{k-n} a_{k-n}^* \right|^2 \right\}. \tag{3.11}$$

Notice now that the weight does not depend on the hypothesis since all hypothesized sequences passing through this branch have to include this partial weight. Combining (3.2), (3.3), (3.5), (3.6) and (3.11) gives

$$Pr\{\hat{A} = \vec{A}(h) | \vec{R}\} = \frac{\prod_{k=1}^{N} W_k(m', m)\, Pr\{\hat{a}_k = a_k(h)\}}{f\{\vec{R}\}} = \frac{\prod_{k=1}^{N} \gamma_k(m', m)}{f\{\vec{R}\}} \tag{3.12}$$

where $\gamma_k(m', m) = W_k(m',m)\, Pr\{\hat{a}_k = a_k(h)\}$ and $\hat{a}_k = a_k(h)$ is the symbol associated with the state transition m' to m at time k. When looking at the trellis structure, each hypothesized data sequence can be equivalently viewed as a state sequence and each symbol in the data sequence is marked by a state transition. As there is more than one hypothesized sequence having the branch (m', m) at time k, $\gamma_k(m', m)$ is, therefore, not a function of the hypothesis. $Pr\{\hat{a}_k = a_k(h)\}$ is known as the *a priori* probability (AP) of the symbol $\hat{a}_k$ and $\gamma_k(m', m)$ can be interpreted as the "overall branch weight" corresponding to the specific state transition at the given time index.

The state transition probability can be written as

$$Pr\{s_{k-1} = m'; s_k = m | \vec{R}\} = \frac{\sum_{h \in C_k(m',m)} Pr\{\hat{A} = \vec{A}(h) | \vec{R}\}}{\sum_{\forall h} Pr\{\hat{A} = \vec{A}(h) | \vec{R}\}}$$

$$= \frac{\sum_{h \in C_k(m',m)} \prod_{k=1}^{N} \gamma_k(m', m)}{\sum_{\forall h} \prod_{k=1}^{N} \gamma_k(m', m)} \tag{3.13}$$

where $C_k(m',m)$ is the subset of the set of all hypotheses that transverse the trellis and has a branch between states $s_{k-1} = m'$ and $s_k = m$. Equation (3.13) can be studied as a problem of [16]

$$\frac{\begin{array}{c}\text{Sum of products of weights of all paths} \\ \text{which pass through the specific branch}\end{array}}{\begin{array}{c}\text{Sum of products of weights of all} \\ \text{paths passing through the entire trellis}\end{array}} . \tag{3.14}$$

The MAP algorithm is an efficient computational method for this problem. As shown in appendix A, a forward recursion is needed, i.e.

$$\alpha_k(m) = \sum_{m'} \alpha_{k-1}(m') \gamma_k(m',m) \tag{3.15}$$

where $\alpha_k(m)$ is the sum of products of the weights along all paths which terminate in state m at time k and $\alpha_0(m)$ is a known initial condition for all possible values of m. The backward recursion is

$$\beta_k(m) = \sum_{m'} \beta_{k+1}(m') \gamma_{k+1}(m,m') \tag{3.16}$$

where $\beta_k(m)$ is the sum of products of the weights of all paths going to the end of trellis starting from state m at time k. $\beta_N(m)$ is the known initial condition for all m.

Define $\sigma_k(m',m)$ as the sum of products of the weights of all paths having a branch joining state $s_{k-1} = m'$ and state $s_k = m$. It can be expressed as

$$\sigma_k(m',m) = \alpha_{k-1}(m') \gamma_k(m',m) \beta_k(m) . \tag{3.17}$$

A simple trellis diagram is given in figure 3.1 to provide a better understanding of these variables.

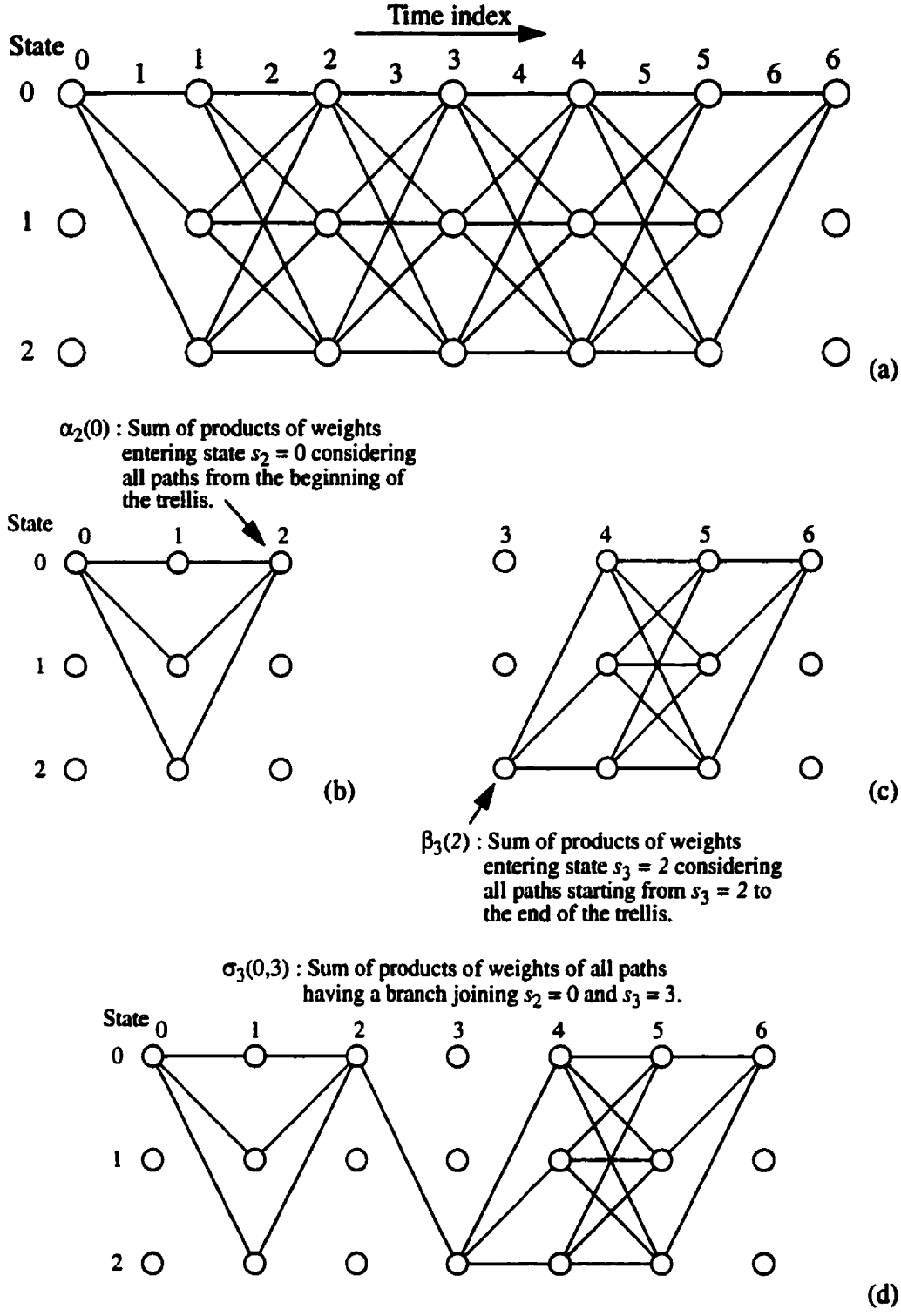


Figure 3.1 (a) A trellis diagram with three states, (b) paths needed to be considered when looking for α, (c) paths needed to be considered when looking for β and (d) an example showing the part of trellis that is associated with σ.

The demodulator output is the APPs of the symbols which, for the k-th transmitted symbol, is

$$Pr\{\hat{a}_k = q|\vec{R}\} = \frac{\sum_{(m',m)\in D} \sigma_k(m',m)}{\sum_{\forall(m',m)} \sigma_k(m',m)} \tag{3.18}$$

where D is the set of state transitions at time k corresponding to transmission of alphabet q, $0 \le q \le 3$ for QPSK symbol. The denominator is the sum of weights of all paths within the trellis.

With the above discussion, it is found that the MAP algorithm can be employed in conjunction with the linear predictor to demodulate signals transmitted over fading channels. The MAP algorithm implemented demodulator has a structure illustrated in figure 3.2.

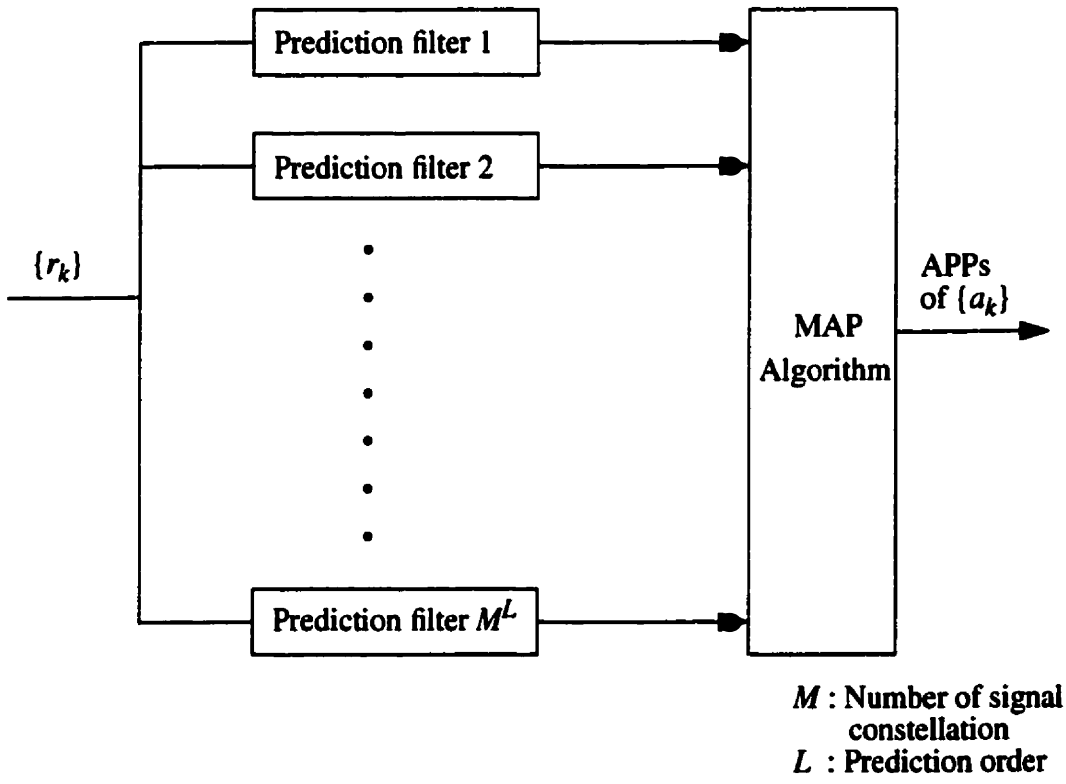


Figure 3.2 MAP algorithm based demodulator.

3.1.2 System Considerations

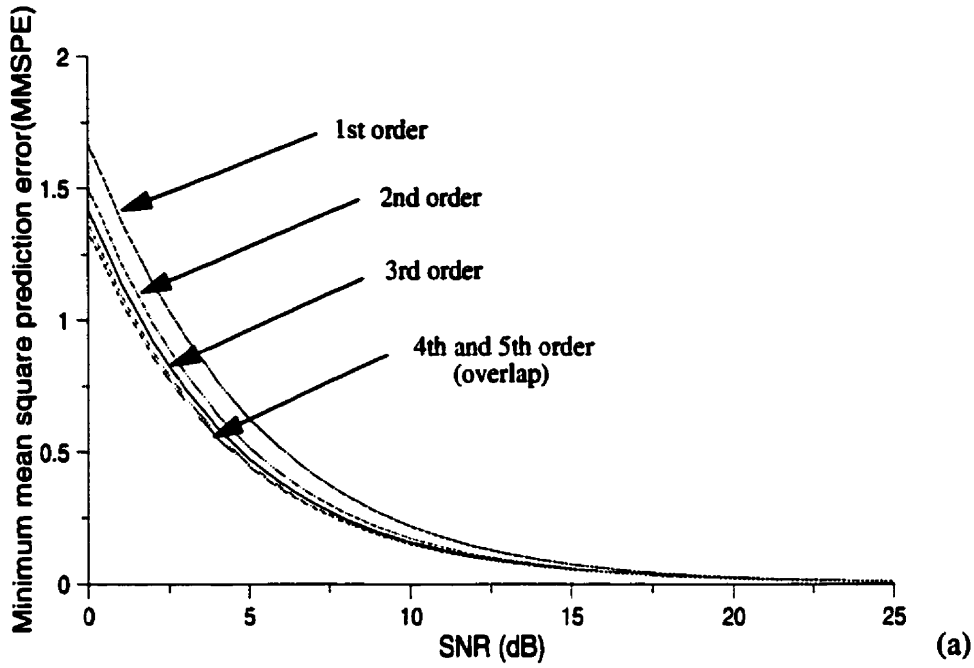
The prediction order or memory length L, which appears in (3.9) is an important parameter to be determined. The trellis of the demodulator has M^L states where M is the number of possible transmitted symbols. If L is large, the trellis becomes complicated by having an enormous number of states such that more numerical calculations are required. A small L implies an inaccurate prediction. The accuracy of prediction is commonly measured by the minimum mean square prediction error (MMSPE). Figure 3.3 shows the relationship between MMSPE and SNR under two different fading rates. The fading rate is defined as the product of the maximum Doppler frequency and the symbol interval. It is also a measure of how fast the channel changes. A large value of this parameter implies a relatively fast changing environment faced by the communication system. From figure 3.3(a), it is observed that, at a fading rate of 0.01, increasing the prediction order above three gives only marginal improvement in the prediction error over the given range of SNR. When the fading rate increases to 0.05, figure 3.3(b) reveals that increasing the prediction order beyond two does not provide a significant reduction in the prediction error. Hence, for a relatively fast time-varying channel, a linear predictor with smaller order can be employed. In this thesis, for simulation convenience, the prediction order is set to be three for all fading rates.

A 3rd-order predictor implies a 4^3=64 state trellis for QPSK signalling and each state has three symbols. It is desirable to send three known symbols at the beginning so as to start up the channel estimation. Also the same procedure is used at the end of a data block to ensure the trellis terminates at a known state. These symbols are called "framing symbols". With the "0" symbol being chosen as this known alphabet, the initial conditions for the recursions are

$$\alpha_0(0)=1 \text{ and } \alpha_0(m)=0 \text{ for } 1 \le m < 64,$$

$$\beta_N(0)=1 \text{ and } \beta_N(m)=0 \text{ for } 1 \le m < 64.$$

Equation (3.10) reveals that the branch weight depends on the magnitude of the predictor output. The problem of phase ambiguity arises when constant envelope signals, such as PSK, are transmitted. The root of the problem is the lack of phase reference in the demodulator. Two different paths, labelled A and B, may have the same path weight if path B can



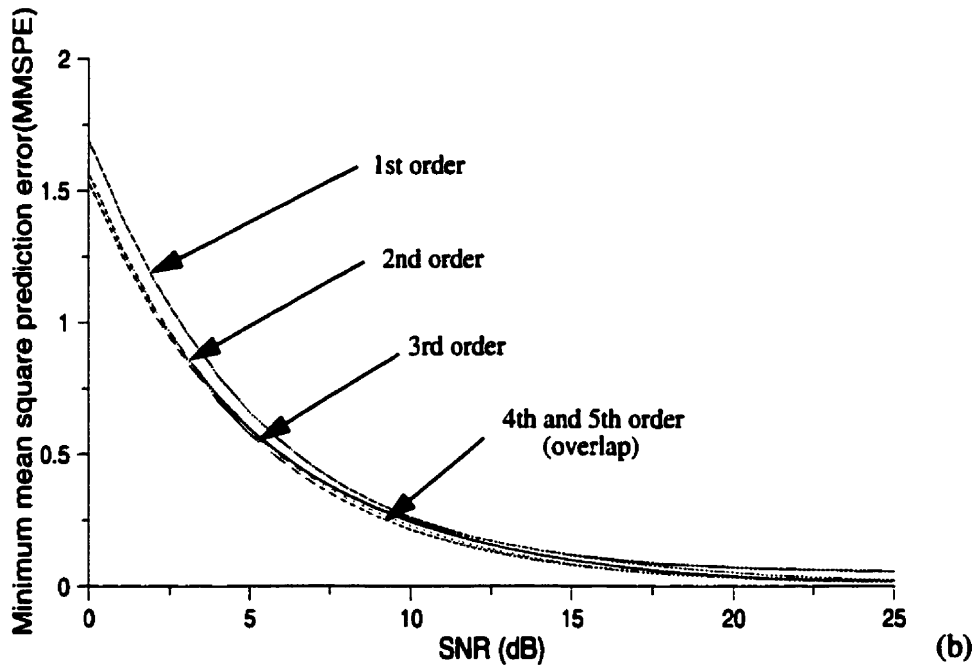


Figure 3.3 Minimum mean square error of the linear prediction of different order with (a) fading rate = 0.01 and (b) fading rate = 0.05.

be obtained by rotating each symbol of A by one of the phase symmetries of the M-ary PSK constellation [52, 58]. This problem can be solved by periodically inserting pilot symbols into the data sequence which enables the demodulator to obtain a phase reference.

The pilot symbols have to be randomized to avoid transmission of a tone. Unlike the PSAM and SADD schemes discussed in chapter 1, the purpose of pilot symbols is not to estimate the channel gain, g_k, for the data sequence and hence the restriction on PIR is not as tight as mentioned before. Similar to the reference-based detection method, a higher PIR often provides a better error performance at the expense of bandwidth efficiency. As a compromise between these two aspects, a PIR of 1 in 9 (1:8) has been chosen for all simulations.

The introduction of framing and pilot symbols invokes the need to modify the transmitter structure provided in figure 2.8. The modified model is shown in figure 3.4 in which a multiplexer is included to insert these two types of symbols into the data stream. The resultant data block will have a structure as depicted in figure 3.5.

As a final note in the discussion of the MAP demodulation, attention must be paid to reduce the risk of numerical error due to underflow when directly applying the MAP algorithm. The risk is especially high in a relatively long trellis. Simulation done during this thesis research has indicated that such an error occurs, in fact, when the length of the trellis is more than one thousand. One possible solution for this problem is provided in appendix C.

3.2 Detection of Uncoded signals

This section looks at the problem of detecting QPSK signals in an uncoded system. For an uncoded system the receiver needs only one stage which is the channel demodulator. The MAP demodulator takes the received samples and computes the APPs of the data symbols. The QPSK symbols are assumed to be equally likely which means the APs that appear in (3.12) are the same for all possible symbols at any time. As the error performance is evaluated on the basis of bit error rate (BER) under a certain signal-to-noise ratio (SNR), hard decision of the bit sequence based on the calculated symbol APPs is

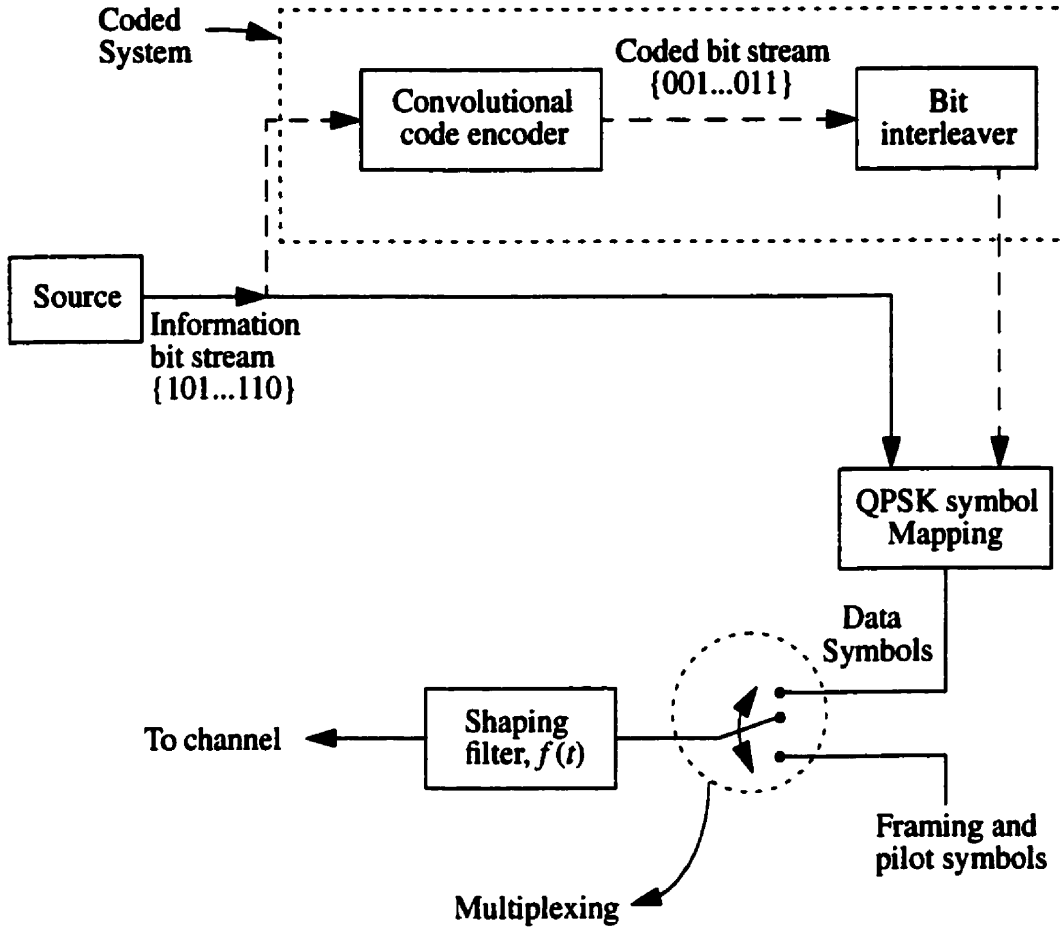


Figure 3.4 Modified transmitter model.

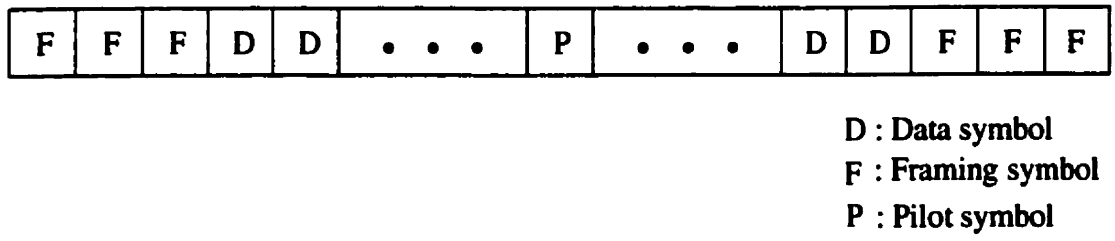


Figure 3.5 Transmitted data structure in a frame.

required. The demodulator has two approaches for this purpose. The first approach is to find the symbol with the largest APP and claim that this symbol is transmitted. By making a hard decision on the transmitted symbol, the demodulator can then proceed to the inverse mapping of the QPSK signalling scheme according to figure 2.6 to determine the corresponding two bits. This procedure is illustrated by the numerical example given below:

APP of "0" (00)	APP of "1" (01)	APP of "2" (10)	APP of "3" (11)	Symbol HD	Bit #1 HD	Bit #2 HD
0.15	0.36	0.04	0.45	3	1	1

Table 3.1 The first approach for the demodulator decision.

In the above table, the numbers inside the parentheses are the binary representation of each QPSK alphabet. The most significant bit symbolizes bit #1 and bit #2 is represented by the least significant bit.

In the second approach, the symbol hard decision is not needed. The demodulator can compute the APPs of bits #1 and #2 using the APPs of the symbols. Using the above numerical example, this approach can be summarized as follows:

APP of "0"	APP of "1"	APP of "2"	APP of "3"	APPs of bit #1		APPs of bit #2		Bit #1 HD	Bit #2 HD
				0	1	0	1		
0.15	0.36	0.04	0.45	0.51	0.49	0.19	0.81	0	1

Table 3.2 Second approach of demodulator decision.

Notice that, in this example, different decisions are made for bit #1 with these two different decision approaches. Generally speaking, it is better to fully utilize all the available soft information before making a hard decision. As the hard output being sought here is at the bit level, it is advisable to manipulate the symbol soft information into bit probabilities before making such a decision. For this reason, the second approach is considered to be more reliable.

The error performance of QPSK signals is evaluated at two fading rates: 0.01 and 0.05. Also two data alignment structures are considered for each fading rate. The first one

considers that in each frame, 256 data symbols (512 data bits) are transmitted. These data symbols, combined with the required pilot symbols and framing symbols, constitute one data block. The term "block length" is usually defined by the number of data bits embodied in a data block and thus for this case, the block length is 512. The other case studied has a data structure of block length 2048.

The demodulator uses the approach given in table 3.2 to generate a bit hard decision. BER is estimated by dividing the total number of error bits by the total number of data bearing bits transmitted. On the other hand, SNR has to include the transmission power loss due to framing and pilot symbols. The average energy required for each bit is

$$E_b = \frac{\begin{array}{c}\text{Energy required} \\ \text{for data symbols}\end{array} + \begin{array}{c}\text{Energy required for} \\ \text{framing and pilot symbols}\end{array}}{\text{Total number of data bits}}. \tag{3.19}$$

With E_b defined in this way, the average SNR per bit is

$$SNR/bit = \frac{\sigma_d^2 E_b}{N_o}. \tag{3.20}$$

For all the simulations, σ_d^2 is set to one for convenience. The error performance of the uncoded system with different fading rates and block lengths is shown in figure 3.6. It is shown in the two plots that the uncoded system demodulated with the MAP algorithm can achieve, depending on the fading rate, a BER of the order of 10^{-3} at average SNR as high as 15 to 17 dB. It is further observed that the system performs better at a lower fading rate. As demonstrated in figure 3.3, the prediction error at a fixed SNR is usually lower in the slow fading scenario. For a system without a coding scheme, a more accurate prediction is likely the main reason why the BER is lower when operating in a relatively slow fading environment.

Moreover, figure 3.6(b) indicates that this plot is just a replica of figure 3.6(a). This means that the difference in block length has no effect on the system's performance. As can be seen later, the block length becomes important when the coded interleaved system is studied.

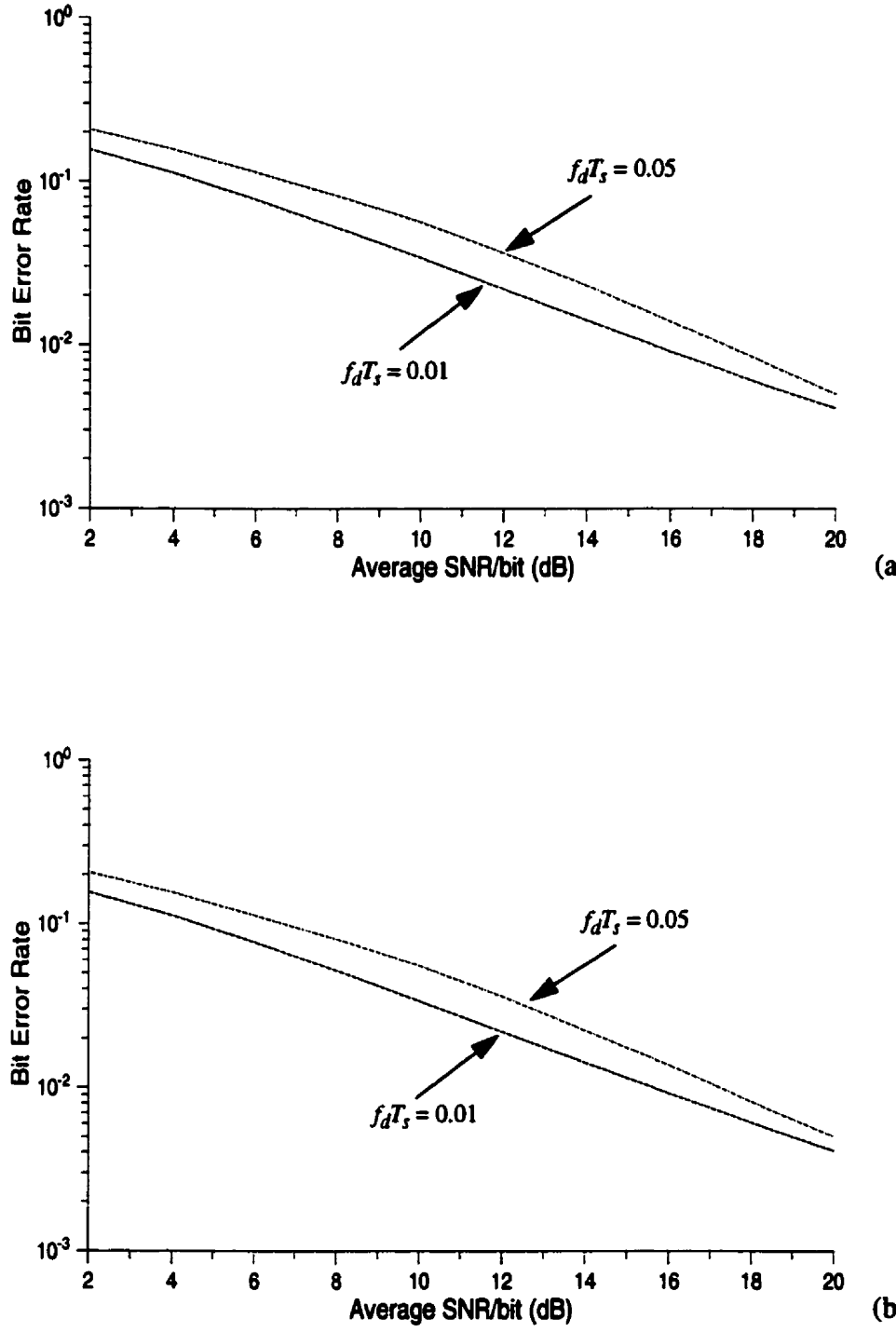


Figure 3.6 Error performances of uncoded QPSK signal with fading rates 0.01 and 0.05 at (a) block length = 512 bits and (b) block length = 2048 bits.

3.3 Detection of Coded Signals

The error performance for the uncoded system may not be satisfactory but it can be enhanced by error correction codes. This section illustrates the issue of joint channel demodulation and decoding with MAP processing.

3.3.1 Two Stage Receiver

A two stage receiver model was proposed in chapter 2 for an interleaved coded system operating in a fading channel. In brief, the first stage compensates for the channel distortion by assuming the data sequence is uncoded. The second stage decoder uses the output from the first stage, plus the coding structure, to generate an estimate for the information bit sequence.

The MAP algorithm implemented in the previous section is used again for channel demodulation. In general, although the demodulator provides APPs for framing as well as pilot and data symbols, only the probabilities of the last symbol type are useful for decoding. They can be extracted through demultiplexing. These symbol APPs reflect the reliabilities of the coded bits. Recall that the coded bits are bit interleaved and then grouped into symbols. Conversely, the symbol probabilities have to be converted back into bit probabilities followed by deinterleaving. This aligns the APPs of the coded bits in the same order as they appear at the encoder output. Figure 3.7 illustrates the relationship between interleaving and deinterleaving by using a specific 32 x 64 block interleaver. The decoder takes these probabilities of the coded bits as input and tries to recover the information bit sequence. The decoder design is again dependent on the criterion chosen for optimality. Just like the channel demodulation, minimum bit error probability is chosen as the design criterion here and the MAP algorithm is implemented. The *a priori* information of the information bit, which depends on how the source generates zeros and ones, is assumed to be known. A complete picture of the two stage receiver is given in figure 3.8.

3.3.2 Simulation Results

The receiver for the interleaved coded system of figure 3.6 was simulated by a C program. The concept of block length has been addressed in the previous section based upon

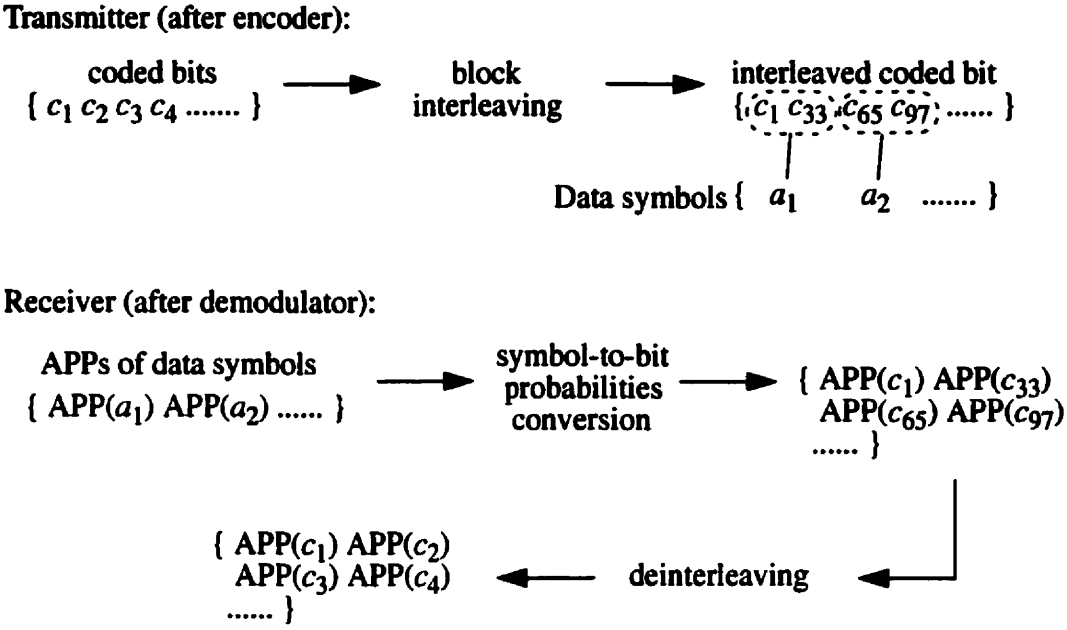


Figure 3.7 Relationships between interleaving at transmitter and deinterleaving at receiver.

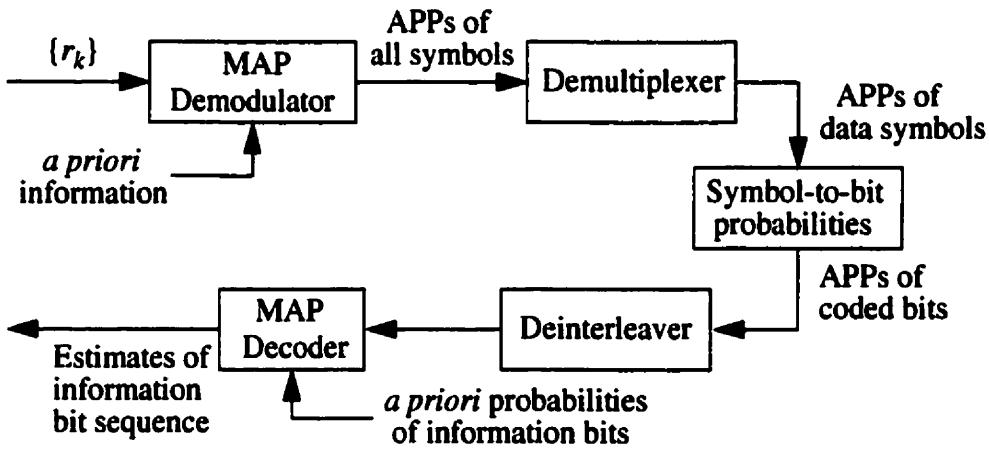


Figure 3.8 Two stage MAP receiver.

the number of data bits transmitted in each frame for an uncoded system. The same idea can be applied to the coded system where it becomes the number of coded bits transmitted within a frame. Notice that it is also equivalent to the interleaver size required because bit interleaving is assumed. Two block lengths are considered here: one has 512 coded bits and the other has 2048. By employing the convolutional coding scheme presented in section 2.3.1, the use of the two different block lengths means transmission of 256 and 1024

information bits respectively. Furthermore, just like the MAP demodulator, it is desirable to have the decoder trellis terminated at a known state. With the all-zero state being selected, five zeros are required for this purpose with reference to the generator polynomial (2.17). Hence for each frame, the "useful" number of information bits transmitted has to be reduced by five. On the other hand, the average SNR for the coded system has to take into account the loss of power in framing and pilot symbols as well as the redundancy due to the coding structure. The average energy required for each information bit is

$$E_b = \frac{\begin{array}{c}\text{Energy required} \\ \text{for data symbols}\end{array} + \begin{array}{c}\text{Energy required for} \\ \text{framing and pilot symbols}\end{array}}{\text{Total number of "useful" information bits}} \tag{3.21}$$

where data symbols refer to symbols representing the coded bits. The average SNR per bit has the same expression as in (3.20) with E_b defined in (3.21). σ_d^2 is again set to one.

The coded systems with both block lengths are assumed to work in the same fading channel having fading rates of 0.01 and 0.05. Random interleaving is applied for the system with a smaller block length whereas block interleaving with size 32 x 64 [37] is employed for the one with larger block length. The simulation results for the block length of 512 are given in figure 3.9 and those for a block length of 2048 are shown in figure 3.10. Simulation results for the ideal channel state (ICS) information are also plotted to serve as a reference.

First consider figures 3.9(a) and (b) which show the error performance for the system with a small block length. It is interesting that, despite a more accurate prediction, the system performs worse in the slow fading environment (f_dT_s=0.01) than it does in the fast fading channel (f_dT_s=0.05) for SNR larger than 10 dB. The difference in performance at BER of 10^{-4} is about 1.4 dB. This implies there must exist another factor(s), other than the prediction error, which affects the error performance. For a slowly time-varying channel, error events at the demodulator are usually very long. Constrained by block length, the small size interleaver is less effective in randomizing the error pattern. The error correction code chosen here is only efficient in correcting random errors and, even though the prediction error is less, the system working under a slow fading environment is

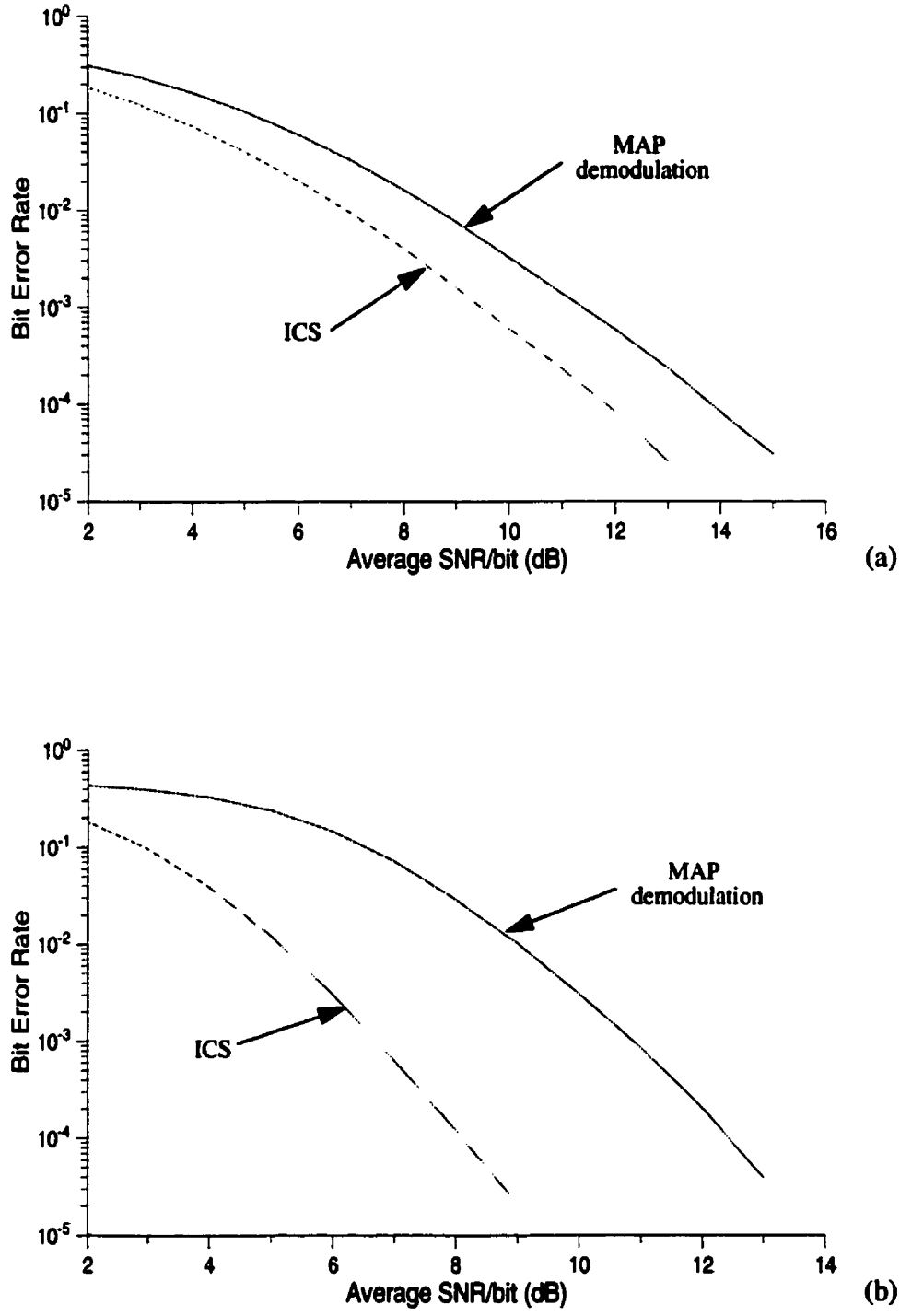


Figure 3.9 Error performances for coded interleaved system with block length 512 bits at (a) fading rate = 0.01 and (b) fading rate = 0.05.

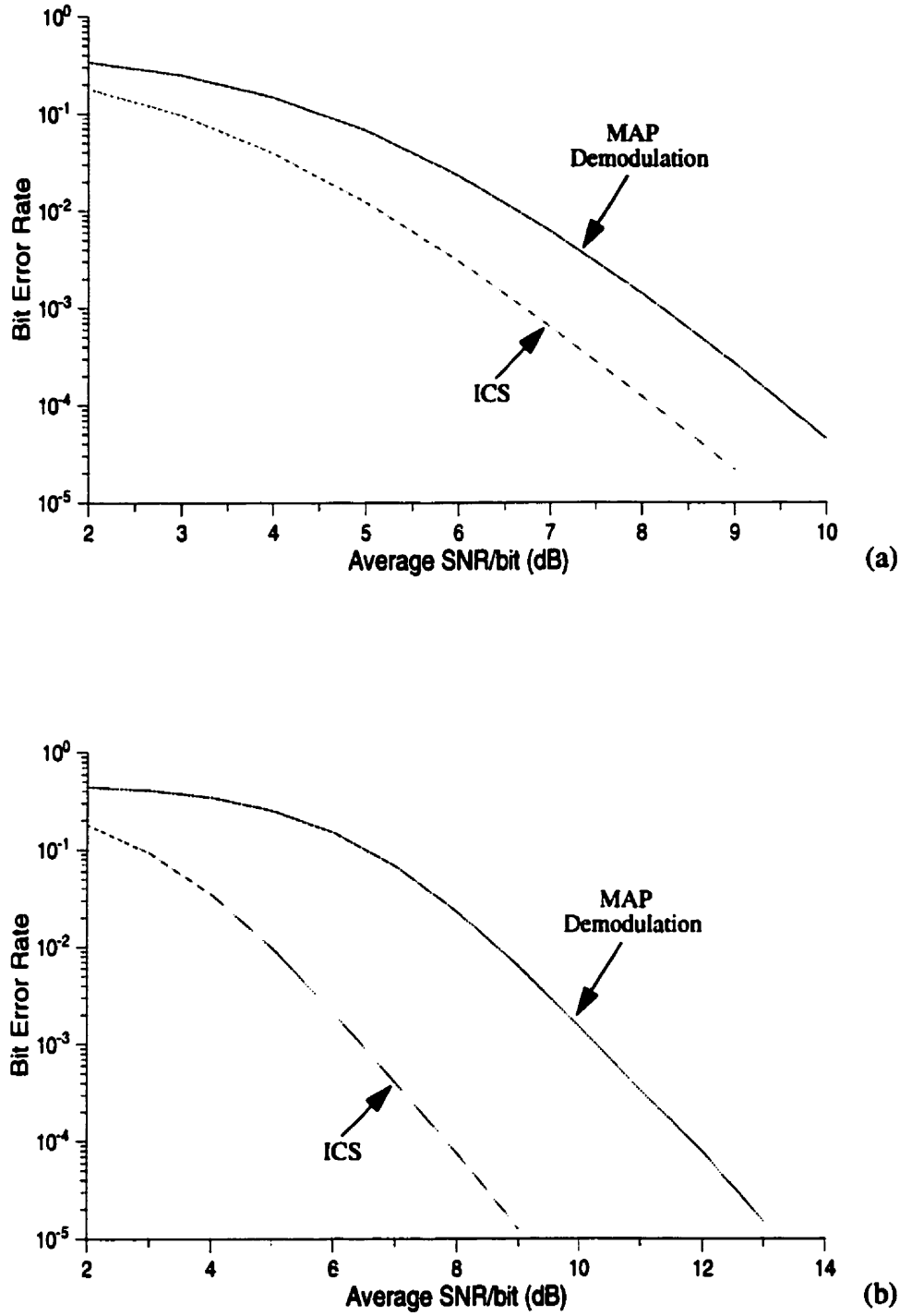


Figure 3.10 Error performances for coded interleaved system with block length 2048 bits at (a) fading rate = 0.01 and (b) fading rate = 0.05.

outperformed.

On the other hand, it is of interest to see how much worse the two stage receiver is compared to the receiver with ICS information. A smaller difference is observed for the system in a slow fading channel. There is a 2 dB loss at a BER of 10^{-4} for the slow fading channel while a loss of as much as 4.2 dB is seen at the same BER for the fast fading channel. This difference implies that the gap between the two stage receiver and the receiver with perfect channel information becomes narrower as the fading rate decreases. This is intuitively reasonable because a small prediction error (or a more "accurate" prediction) means that the receiver has a better knowledge about the channel. Hence, the performance of the receiver with ICS is the lower bound of all receivers. The receiver in slow fading environment experiences a smaller prediction error than in fast fading so that its performance should be closer to this lower bound.

For a system with a larger block length, the error performance in a slow fading channel always surpasses the performance in a fast fading channel, as illustrated in figures 3.10(a) and (b). The interleaving operation is likely to be effective here and, therefore, the lower prediction error due to slow fading leads to a superior performance.

Again, the performance of the two stage receiver is compared to the receiver with ICS. The two stage receiver, when operating in a slow fading channel, experiences a 1.4 dB loss at the BER of 10^{-4}. A larger difference of 4 dB is recorded for the same comparison for the fast fading channel. Notice that the loss becomes smaller as the block length increases.

To conclude, the two stage receiver which uses the MAP algorithm is capable of jointly demodulating and detecting coded signals transmitted over fading channels. Reliable detection depends not only on an accurate prediction but also on an effective interleaving scheme. For small fading rates, the two stage receiver has a performance which approaches a receiver with perfect channel knowledge.

3.4 Iterative MAP Processing

In general, when a coded interleaved system is considered, the optimum detection strategy is to look at the composite trellis introduced by the encoder, interleaver and fad-

ing channel and then make a decision based on a specific criterion for optimality. As mentioned, the complexity for such a receiver is prohibitively high and this is the reason why a suboptimum two stage receiver is used in practice. In order to push the performance of the suboptimum receiver closer to the optimum one, iterative processing is a possible solution. The objective of iterative processing is to provide the former stage(s) with information which is generated by the latter stage(s) in a multi-stage receiver structure. The information generated by the latter stage(s) can serve as the "updated *a priori* information" in the former stage(s).

For the two stage receiver structure, consider the branch weight calculation in channel demodulation given by (3.12). It depends on two terms: the first term is the channel output which directly affects the prediction and the second is the AP of the symbol. In the previous discussion of demodulation the symbols are assumed to be equally likely which means that the *a priori* information can be ignored. Error performance can be improved if there is a way to supply extra or updated "*a priori*" information. This can be achieved by feeding back the reliability information of the coded bits obtained from the second stage decoder to the demodulator. In doing so, channel compensation in the demodulator is allowed to enjoy the benefit of the coding structure generated from the decoder. This is the aim of iterative processing in the fading channel and the main focus of this section is on how to accomplish this goal with the MAP algorithm.

3.4.1 Background Information on Iterative Processing

Iterative processing has been applied to decoding parallel and serially concatenated codes in AWGN channels [4, 6, 7, 20]. To have a better understanding of this concept, consider the simple serially concatenated code encoder and decoder depicted in figure 3.11. It is assumed that the coding structure can be represented by a trellis such that traditional decoding strategies such as the MAP algorithm and the VA can be applied. The encoder consists of two parts, one is labelled outer (code) encoder and the other inner (code) encoder. The outer encoder accepts the information sequence $\underset{\sim}{u}^{o}$ with known *a priori* probabilities and converts it into a coded sequence $\underset{\sim}{c}^{o}$. This coded sequence then

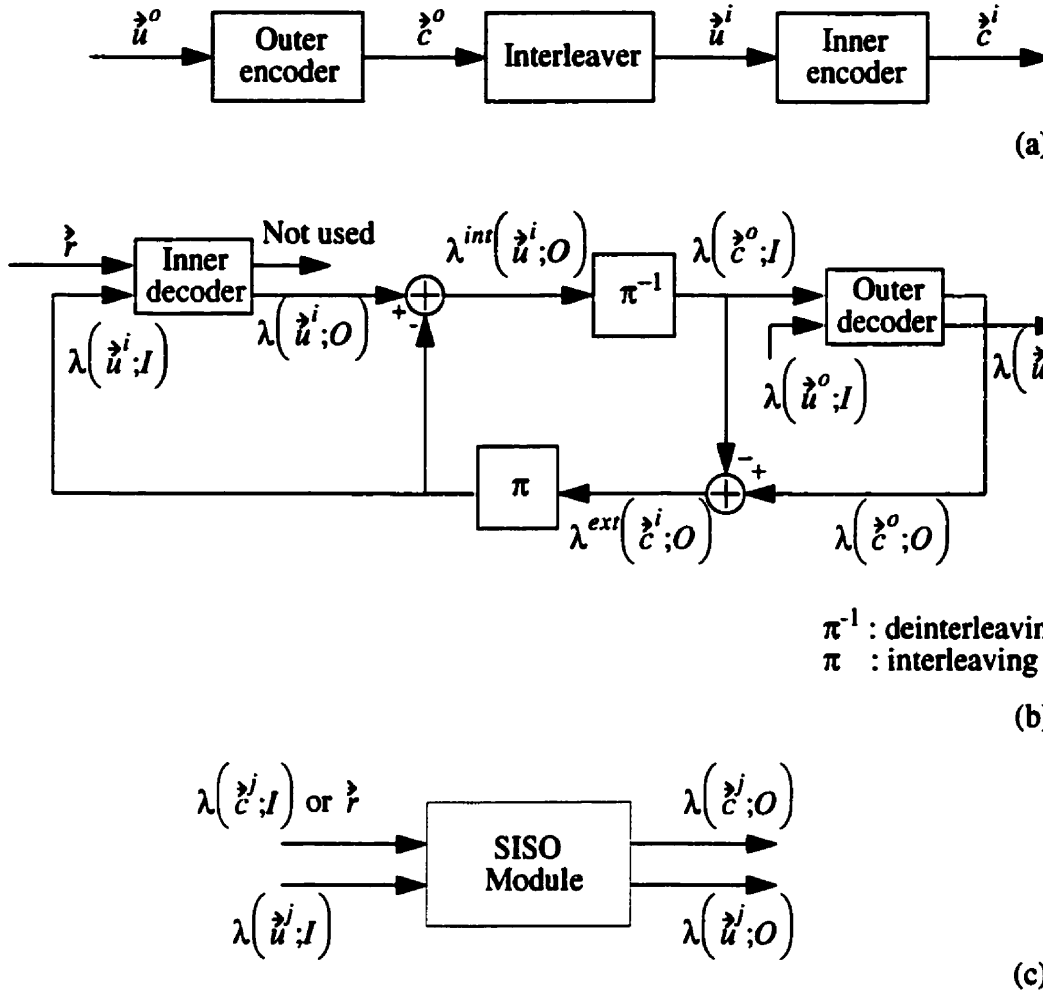


Figure 3.11 (a) Serially concatenated code encoder, (b) iterative decoder for serially cancatenated code and (c) a SISO module.

undergoes interleaving and the interleaved output becomes the information sequence $\vec{u}^i$ of the inner encoder. Finally, $\vec{u}^i$ is encoded into another coded sequence $\vec{c}^i$. As usual, $\vec{c}^i$ is mapped onto a chosen signal constellation and sent through the channel.

One key aspect for iterative decoding is the capability of the decoder to accept soft-input and soft-output. Both inner and outer decoders in figure 3.11(b) are assumed to have this "soft-input soft-output" (SISO) feature [5]. Each SISO module has two inputs and outputs as illustrated in figure 3.11(c). The outputs are APPs of the information and coded

sequences and these values are expressed in the form of a log-likelihood ratio (LLR) for convenience. Conventionally, the LLR is defined for a random variable which takes a value of either one or zero and can be expressed mathematically as

$$LLR(y) = \log\frac{Pr\{y=1\}}{Pr\{y=0\}}. \tag{3.22}$$

The *a posteriori* LLRs are denoted by $\lambda\left(\vec{u}^{j};O\right)$ and $\lambda\left(\vec{c}^{j};O\right)$ where j is the index referring to either the inner or outer decoder and "O" simply means output. The input labelled $\lambda\left(\vec{u}^{j};I\right)$ is the *a priori* LLR of the information sequence. The other input of the inner decoder is the received sample $\vec{r}$ which reflects the transmitted sequence $\vec{c}^{i}$. The inner decoder uses these samples and the *a priori* LLRs (initialized to be zeros) to compute the *a posteriori* LLR $\lambda\left(\vec{u}^{i};O\right)$. It was suggested in [20, 21] that the *a posteriori* LLR at the inner decoder output, at any time k, can be split into two terms

$$LLR\left(u_k^i\middle|\vec{r}\right) = LLR^{int}\left(u_k^i\middle|\vec{r}\right) + LLR\left(u_k^i\right)$$

or

$$\lambda\left(u_k^i;O\right) = \lambda^{int}\left(u_k^i;O\right) + \lambda\left(u_k^i;I\right) \tag{3.23}$$

where the first term on the right hand side is called the "intrinsic information" and the second term is the previously found (initialized to be zero) *a priori* information. The equation is valid only when the LLRs are independent of each other. This requirement is assumed to be met with an efficient interleaving scheme. Intrinsic information is described as the information obtained from the received samples and trellis structure without considering the *a priori* probability of u_k but possibly using *a priori* probabilities of other bits in the inner decoder/demodulator. For iterative processing, it is very important to prevent the first stage from passing information to the second stage which has been generated by the second stage in the first place. This can be carried out by subtracting the *a priori* LLR from the *a posteriori* LLR and the difference is the intrinsic information. The intrinsic information, which is also a measure of the reliability of the inner code information bits, is passed next through a deinterleaver, whose outputs are the LLRs of the coded symbols for the outer code. Thus,

$$\pi^{-1}\{\lambda^{int}(\vec{u}^{i};O)\} = \lambda(\vec{c}^{o};I). \tag{3.24}$$

The deinterleaved LLRs are then processed by the outer decoder to generate the a posteriori LLRs for both information and coded sequences, $\lambda(\vec{u}^{o};O)$ and $\lambda(\vec{c}^{o};O)$. Based on the first term, a hard decision can be made to give an estimate of $\vec{u}^{o}$. On the other hand, in order to carry on with iterative processing, it is necessary to pass *a priori* information $\lambda(\vec{u}^{i};I)$ back to the inner decoder. As $\vec{u}^{i}$ is the interleaved version of $\vec{c}^{o}$, such information can be extracted from the *a posteriori* LLR $\lambda(\vec{c}^{o};O)$. Analogous to (3.23), it is suggested that this ratio can be separated, at time k, into

$$\lambda(c_k^o;O) = \lambda^{ext}(c_k^o;O) + \lambda(c_k^o;I) \tag{3.25}$$

with the first term on right hand side called the "extrinsic information" which the decoder obtained from the coding scheme without considering the input probability of bit c_k. Again, it may consider those of other bits. This extrinsic information can be obtained by finding the difference between the *a posteriori* LLR and the input LLR of the coded bit. Furthermore, the extrinsic information is also related to the reliabilities of $\vec{u}^{i}$ through interleaving. Thus the *a priori* information for the first stage can be found from

$$\pi\{\lambda^{ext}(\vec{c}^{o};O)\} = \lambda(\vec{u}^{i};I). \tag{3.26}$$

The overall decoding cycle is restarted with this newly found *a priori* LLR for the inner decoder.

3.4.2 Iterative MAP Processing on Fading Channels

The symbol-by-symbol MAP algorithm is a technique well suited for iterative processing due to its SISO feature. It can also calculate the *a posteriori* probabilities optimally. In the problem of joint demodulation and detection of signals in the fading channel, the channel memory can be treated as imposing a coding structure to the transmitted signal. By looking at the two stage receiver model shown in figure 3.8, the channel

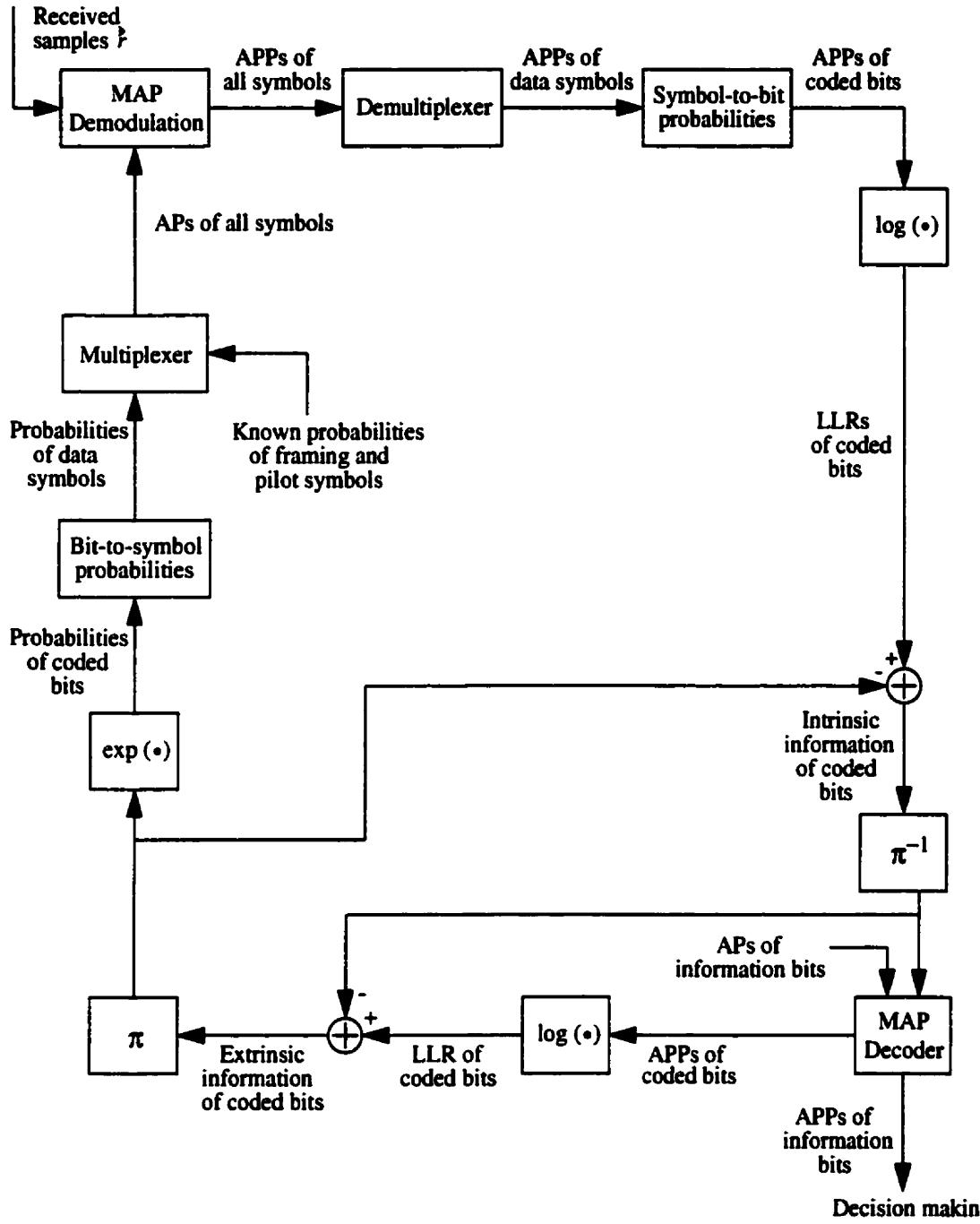


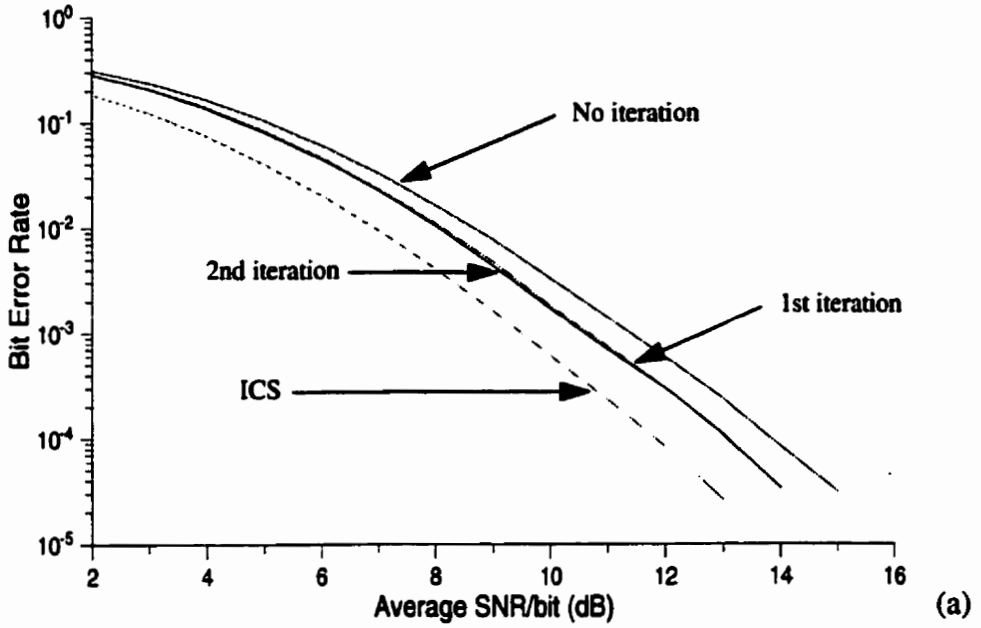
Figure 3.12 Iterative MAP processing for joint demodulation and decoding of *M*-ary coded signal transmitted in fading channel.

demodulator, which models the channel as a trellis structure, can be viewed as the inner decoder. Analogous to the outer decoder is the convolutional code decoder. Nevertheless, some precautions have to be taken. As shown in figure 3.12, demultiplexing is carried out after demodulation to get rid of the probabilities of the framing and pilot symbols. Conversely, multiplexing is required to insert these known probabilities back into the demodulator during iteration. Next, the MAP channel equalization generates APPs of coded bits and it is advisable to convert these APPs into LLRs to facilitate the calculation of the intrinsic information. Notice that there is no loss of information for the probability to LLR conversion and vice versa. A similar step is carried out at the output of MAP decoder for the purpose of extrinsic information computation. The MAP decoder takes the LLRs (intrinsic information) as input, converts them back to probabilities to generate the APPs for both information and coded bits [31, 32]. As the *a priori* information needed in the demodulator involves symbol probabilities, therefore the extrinsic information found has to be converted back into this form. By assuming that the coded bits transmitted are independent of one another, this conversion is straightforward. Such an assumption is reasonable with an efficient interleaving scheme.

3.4.3 Simulation Results

The iteration loop can be initialized by assuming the *a priori* probabilities of the QPSK data symbols entering the demodulator to be 0.25, i.e. the symbols are equally likely. After that the number of iterations to be done is decided by the system designer. Up to three iterations were done for those systems with the block lengths and fading rates mentioned in section 3.3. Nevertheless, it is found that the error performances of the first, second and third iterations for slow fading are almost completely overlapped, regardless of block length. For this reason, results of the third iteration in slow fading are not shown. Figure 3.13 shows the error performances for a system with a block length of 512 and figure 3.14 provides those for a system with a block length of 2048.

For the error performance of a small block length under slow fading, a gain of 0.8 dB is achieved at BER of 10^{-4} after the first iteration. Nevertheless only a 0.03 dB improvement is recorded between the first and second iterations. With this observation, it is likely that doing one iteration can capture most of the coding gain. The result of one iteration has



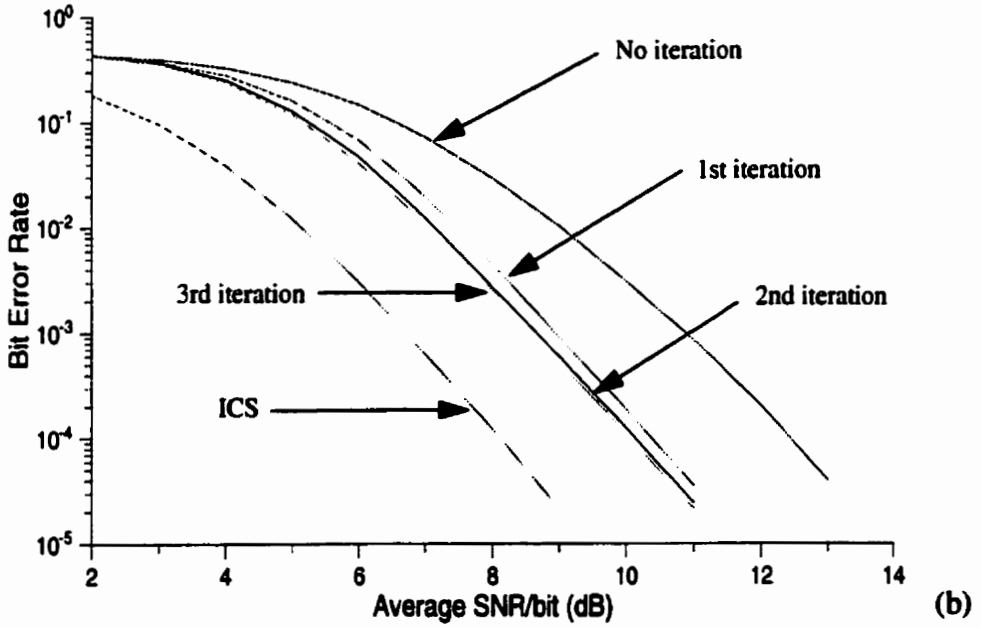


Figure 3.13 Error performances for iterative coded interleaved system with block length 512 bits at (a) fading rate = 0.01 and (b) fading rate = 0.05.

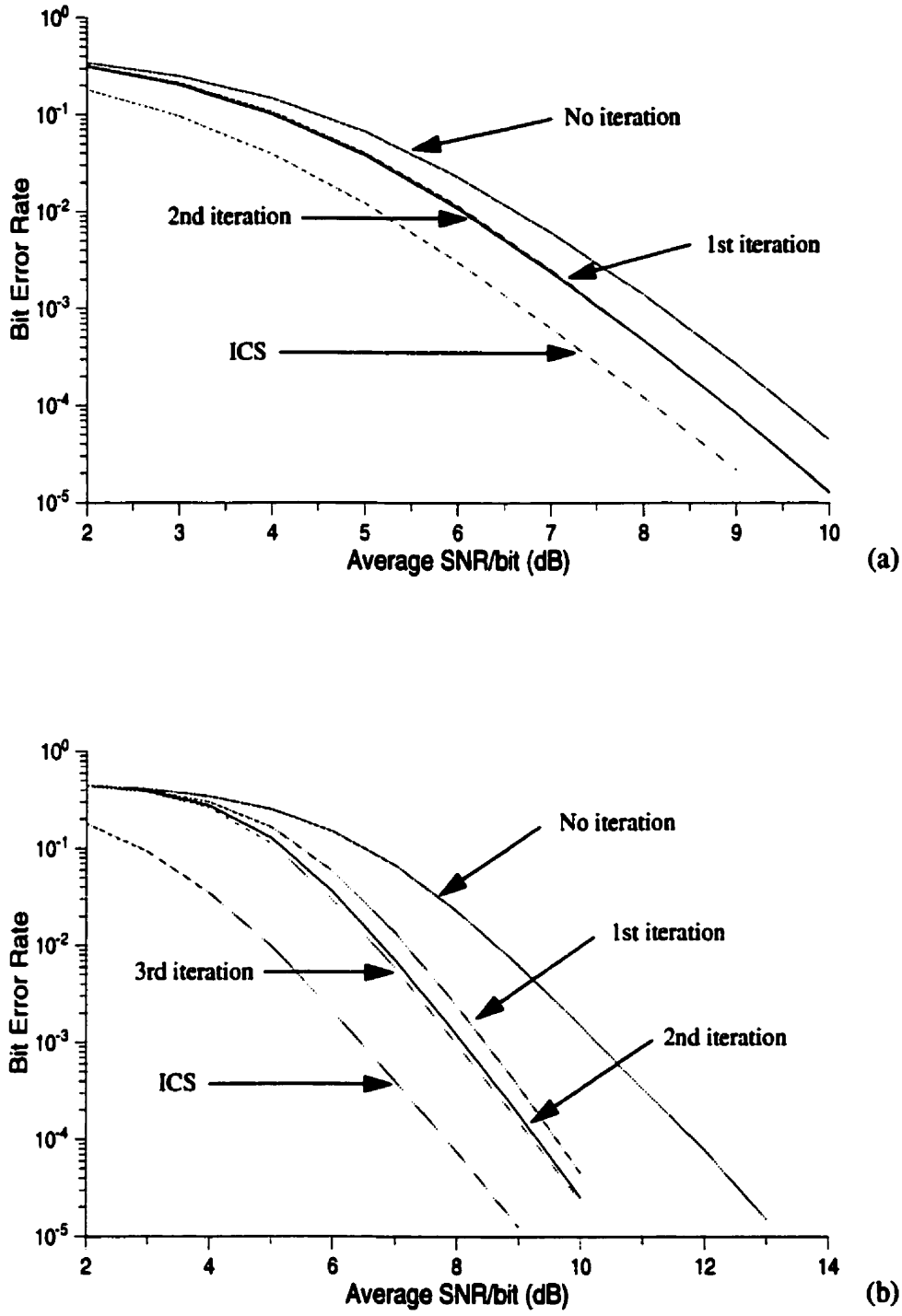


Figure 3.14 Error performances for iterative coded interleaved system with block length 2048 bits at (a) fading rate = 0.01and (b) fading rate = 0.05.

also pushed the error performance to be about 1.2 dB away from the performance of the receiver with ICS.

The coding gain of the small block length in fast fading is significantly larger. The improvement after the first iteration is 2 dB at a BER of 10^{-4}. Error performance is further enhanced by 0.25 dB and 0.06 dB after the second and third iterations respectively. This shows that, just as in slow fading, most coding gain is captured after the first iteration. Nevertheless, the gain is still visible after the second iteration. With the gain after iterations, the system is 1.9 dB away from the one with perfect channel knowledge. The difference is greatly reduced and, therefore, iterative processing is very effective in the fast fading channel.

The results for large block lengths are very similar to those for small block lengths. Under slow fading, there is only 0.65 dB gain after the first iteration and a further of 0.003 dB after the second iteration which is negligible. The difference between the performance after first iteration and the one with ICS diminishes to 0.8 dB. Again, carrying out one iteration seems to be sufficient for slow fading. For a higher fading rate, the coding gain is greater. As much as 2.2 dB is observed after the first iteration. There are improvements of 0.3 dB and 0.03 dB gains after subsequent iterations. In this case, the error performance after the third iteration is 1.45 dB from the one with ICS.

To conclude this section, it is of interest to mention some observations from the graphs. First, at all fading rates, the error performance of the system with a larger block length is usually closer to the performance of the receiver with perfect channel knowledge. Secondly, the coding gain is mostly captured after the first iteration, regardless of block lengths and fading rates. Lastly, the trend noticed for the non-iterative receiver is that the gap between the two stage receiver and the receiver with ICS is smaller in the slow fading condition is preserved. This gap is measured in terms of error performance.

3.5 Summary

In this section the MAP algorithm is implemented for both channel demodulator and decoder. The keys for this implementation are the modelling of the channel as a trellis and the derivation of state transition metrics. Both are carried out under the principle of per-

survivor processing (PSP). With the trellis structure, the MAP algorithm can be used for both the coded and uncoded system. By comparing the plots in figures 3.6, 3.9 and 3.10, it is observed that the error performance of the coded system, as expected, is a lot superior to that of the uncoded system when the SNR is larger than 4 - 5 dB. Furthermore, it is verified that both the fading rates and block length are important parameters for reliable communications in fading channels. With the coded system, iterative processing can be applied to this detection problem such that channel equalization can make use of the coding structure. Error performance is found to be enhanced significantly for fast fading and a moderate gain is observed for the slow fading scenario.

Unfortunately, the MAP processing suffers serious drawbacks in its implementation. The algorithm requires the reception of the entire frame before demodulation/decoding. Due to the necessity of both forward and backward recursions, complicated numerical calculation is involved. Another defect is the high possibility of getting into precision problems for a large trellis. Appendix C suggests one possible solution to this problem but its implementation will undoubtedly cause the method to be more complex. To deal with this complexity problem, the relatively simpler VA can be used. However, this algorithm is not able to deliver soft-outputs which makes it less favourable in concatenated or iterative decoding systems. In the next chapter, a suboptimum but simple soft-output detection algorithm, soft-output Viterbi algorithm (SOVA), will be discussed. It is a modified version of the classic VA and will be implemented to solve the joint demodulation and detection problem.

Chapter 4

MAP Sequence Estimation

The MAP sequence estimation (MAP-SE) is a detection criterion to minimize the probability of sequence error and is equivalent to the maximum likelihood sequence estimation (MLSE) when each transmitted sequence is equally likely. Under either criterion, the Viterbi algorithm (VA) is an attractive solution for the detection problem because its computational complexity varies with the number of states in a trellis rather than the length of the transmitted sequence. It was first developed as an optimal solution to decode convolutional codes transmitted over a memoryless channel [49]. Later research extended its usage to channel equalization for ISI channels [13] and fading channels [30, 33, 44, 52, 58].

Similar to the Bahl *et al*'s MAP algorithm discussed in the previous chapter, the VA accepts soft-inputs. However, the conventional VA delivers only hard-outputs. When the VA is implemented as the channel demodulator in the two stage receiver, it prohibits the decoder from using its capability to accept soft-inputs. This in turn degrades the overall performance. To eliminate this drawback, one can apply the "soft-output Viterbi algorithm" (SOVA) which is a modified version of the VA and capable of providing reliable soft information associated with the hard-outputs.

The SOVA was developed by Hagenauer and Hoeher [17, 18]. Compared to the MAP algorithm, it is a suboptimum method to estimate the symbol *a posteriori* probability (APP). However, while the MAP algorithm is prohibitively complicated, the SOVA is relatively simple. The objective of this chapter is to compare the error performance and complexity of receivers using the MAP algorithm and SOVA.

There are two versions of SOVA: register exchange and trace back modes [20]. They are assumed to have the same degree of complexity and were initially designed for a binary trellis [17]. The register exchange mode was modified later to handle a M-ary trellis

[24]. Another goal of this chapter is to provide a possible way to adapt the trace back SOVA for the M-ary trellis. Summaries for register exchange SOVA in a M-ary trellis and trace back SOVA in a binary trellis can be found in appendix B.

4.1 MAP-SE Demodulation (Standard VA)

Generally speaking, the VA can be viewed as a solution to the problem of maximum *a posteriori* probability estimation of the state sequence of a finite state discrete time Markov process observed in memoryless noise [14]. For such a problem, assume that the Markov process is characterized by a trellis with a fixed number of states, $0 \le s_k < Q$ where k is the time index and Q is the total number of states. Further assume that the initial and final states are known. As time proceeds, the process exhibits a state transition (s_{k-1}, s_k) with which a symbol x_k is associated. Without loss of generality, suppose the length of the trellis for the process is N and the state sequence $\{s_0,\ldots,s_N\}$ is represented by $\vec{S}$. There is evidently a one-to-one correspondence between $\vec{S}$ and the symbol sequence $\vec{X}$. Each possible state sequence or, equivalently, symbol sequence can be characterized by a distinct path within the trellis of the Markov process. The symbol sequence is transmitted over a memoryless channel and the observation is denoted by $\vec{Z}$.

When the detection criterion is to minimize the probability of sequence error by looking at the observation, the problem becomes that of finding the state sequence for which $Pr\{\vec{S}|\vec{Z}\}$, or equivalently $Pr\{\vec{S},\vec{Z}\} = Pr\{\vec{S}|\vec{Z}\}\,Pr\{\vec{Z}\}$, is maximum. Due to the Markov process and memoryless channel properties, $Pr\{\vec{S},\vec{Z}\}$ can be written as

$$\begin{aligned} Pr\{\vec{S},\vec{Z}\} &= Pr\{\vec{S}\}\,Pr\{\vec{Z}|\vec{S}\} \\ &= \prod_{k=1}^{N} Pr\{s_k|s_{k-1}\} \prod_{k=1}^{N} Pr\{z_k|s_k,s_{k-1}\}. \end{aligned} \tag{4.1}$$

Since $\log(\bullet)$ is a monotonic function, maximization of $Pr\{\vec{S},\vec{Z}\}$ and $\log\left[Pr\{\vec{S},\vec{Z}\}\right]$

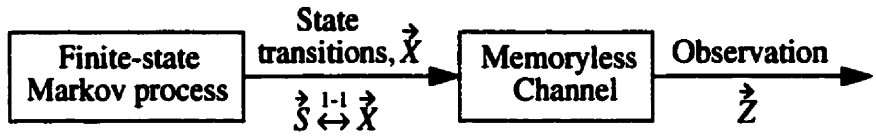


Figure 4.1 General transmission model.

are equivalent. Therefore (4.1) becomes

$$\log\left[Pr\{\vec{S},\vec{Z}\}\right] = \sum_{k=1}^{N} \log\left[Pr\{s_k|s_{k-1}\}\right] + \sum_{k=1}^{N} \log\left[Pr\{z_k|s_k,s_{k-1}\}\right]. \quad (4.2)$$

Define the state transition (branch) metric to be

$$\lambda_k(s_{k-1},s_k) \equiv \log\left[Pr\{s_k|s_{k-1}\}\right] + \log\left[Pr\{z_k|s_k,s_{k-1}\}\right] \quad (4.3)$$

where the first term on the right hand side depends on the AP of state transition and the second term depends on the channel characteristic. Each distinct path within the trellis, therefore, has a metric represented by $\sum_{k=1}^{N} \lambda_k(s_{k-1},s_k)$. The detection problem becomes that of finding the path in the trellis which has the largest metric. To accomplish this, one needs to determine the branch metric expression. For a memoryless channel such as the AWGN channel, the channel characteristic is completely known and the probability density function $f\{z_k|s_k,s_{k-1}\}$ can be found easily. Once the branch metric is defined, the VA can be applied to solve this particular maximization problem. A brief review of this detection strategy is provided in appendix B.

In the last chapter the fading channel was modelled as a finite state machine with a trellis structure. The branch weight (metric) for this trellis was well defined. When the criterion for detection is MAP-SE, the VA can be applied for channel equalization. The succeeding sections are devoted to studying the performance of this kind of receiver. Both coded and uncoded systems are explored. To be consistent with chapter 3, the design parameters, such as PIR and prediction order, are kept the same.

4.1.1 Uncoded System

For the uncoded system, the receiver will have a structure similar to the one in figure 3.2 except that the VA is used instead of a MAP algorithm. Its model is redrawn in figure 4.2. This system is evaluated for two block lengths of 512 and 2048 bits under the same two fading rates. A preliminary simulation showed that a window size of 36 symbol intervals for the VA was adequate.

Just like the results from MAP equalization, the system which operates in slow fading achieves a better error performance than in fast fading as can be seen in figure 4.3. It reaches a BER of 10^{-2} at an average SNR of about 15 or 17 dB depending on fading rate. On the other hand, there is no visible difference in the error performance when block length varies. Hence, for the uncoded system demodulated with VA, a difference in block length does not impose any effect. This statement is consistent with MAP channel equalization.

Finally, by comparing the plots in figure 4.3 and figure 3.6, it is observed that there is virtually no difference between them. This observation supports the claim of Hartmann and Rudolph who stated that detection under criterion of either MAP-SE or MAP-SS should provide the same performance [22].

4.1.2 Coded System

Again, the error performance can be improved when coding is introduced to the system. The usual two stage receiver is employed for this system with the first stage being a channel demodulator using the standard VA. The output of channel demodulation provides a hard decision for the transmitted symbol. These hard estimates of data symbols are converted to bit level by applying the inverse mapping of the QPSK signalling scheme and then deinterleaving. The deinterleaved hard-outputs, which reflect the coded bits, are next passed to the second stage to perform decoding. Minimizing the sequence error is again chosen to be the detection criterion here. Figure 4.4 demonstrates the overall receiver structure.

The coded system is simulated with block lengths of 512 and 2048 bits. Each of these blocks is transmitted under the two different fading rates. The window size for the channel

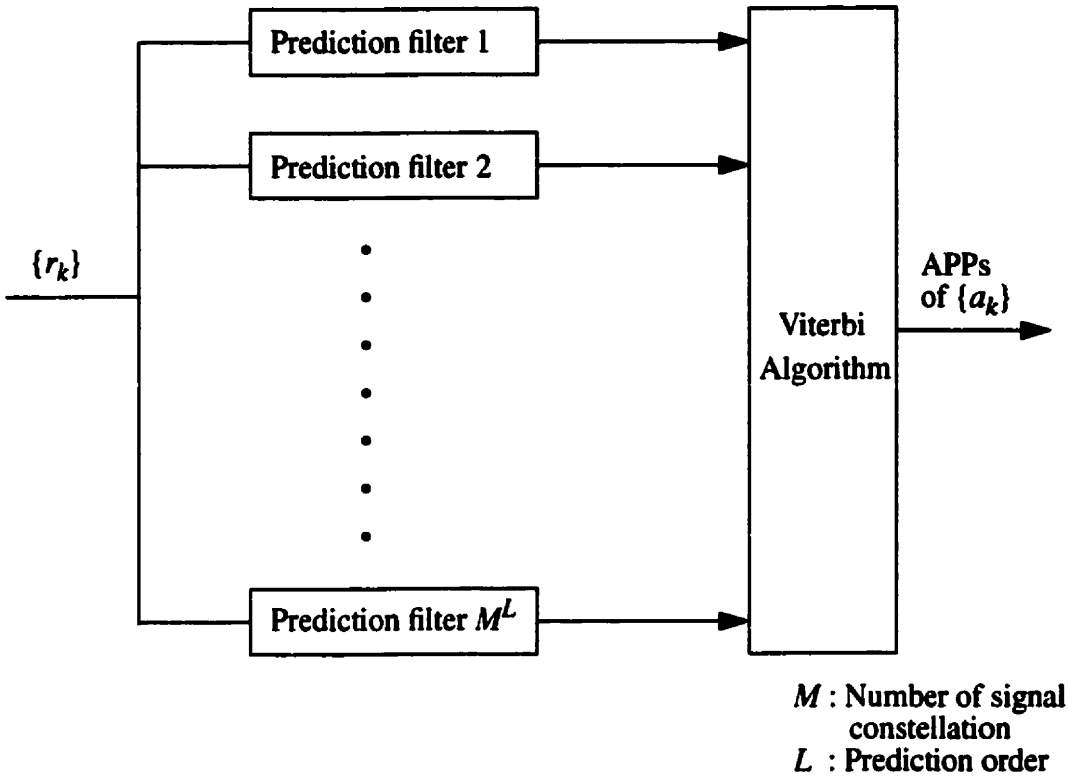
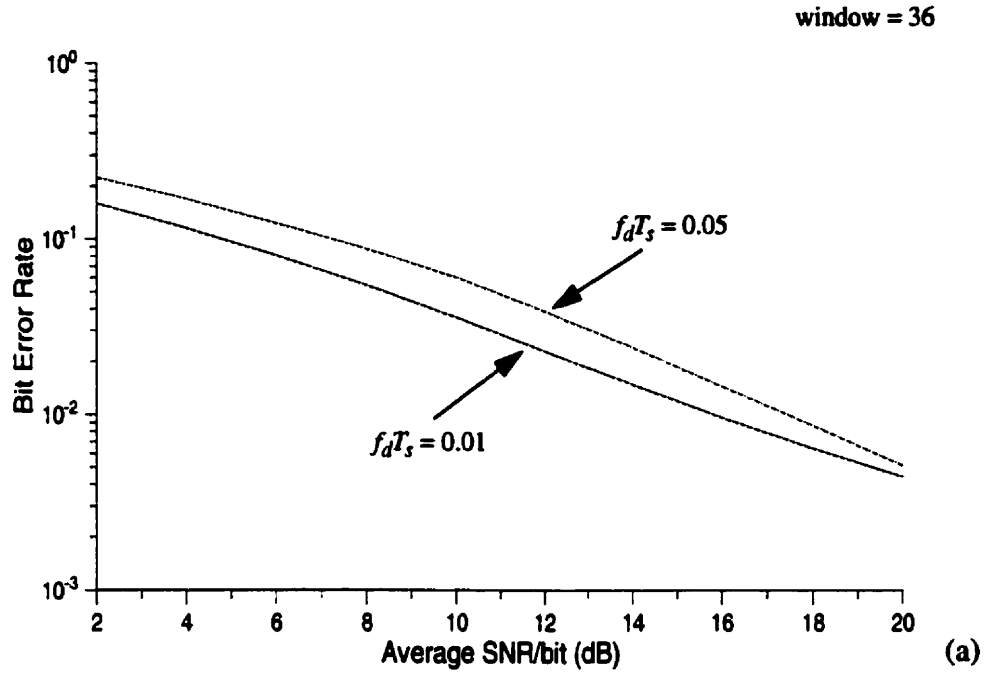


Figure 4.2 VA based demodulator.

demodulator is kept the same as for the uncoded system. To compare with hard decision demodulation, the error performance obtained in MAP detection is included in the plots drawn in figures 4.5 and 4.6. Obviously the hard decision detection strategy is outperformed badly, approximately 6 dB degradation at the BER of 10^{-4} is observed for each comparison. This result serves as sound evidence for the superiority of the soft-output detection approach. The 6 dB gain in soft-output demodulation is actually reasonable. Studies of concatenated coding situations demonstrated that applying a soft-output algorithm introduces about 1 - 4 dB gain over the conventional decoding approach in the presence of AWGN [18, 41, 54]. The reward is even higher in a multipath fading environment. In [18, 36], where SOVA is implemented in channel equalization, the gain over a hard-output demodulator is about 5 dB. In figure 4.5 the soft information is generated by the optimal MAP algorithm, thus having the larger difference (6 dB) is understandable.

In order to obtain a satisfactory performance while keeping the receiver relatively uninvolved, a simple soft-output algorithm can be employed for channel demodulation.



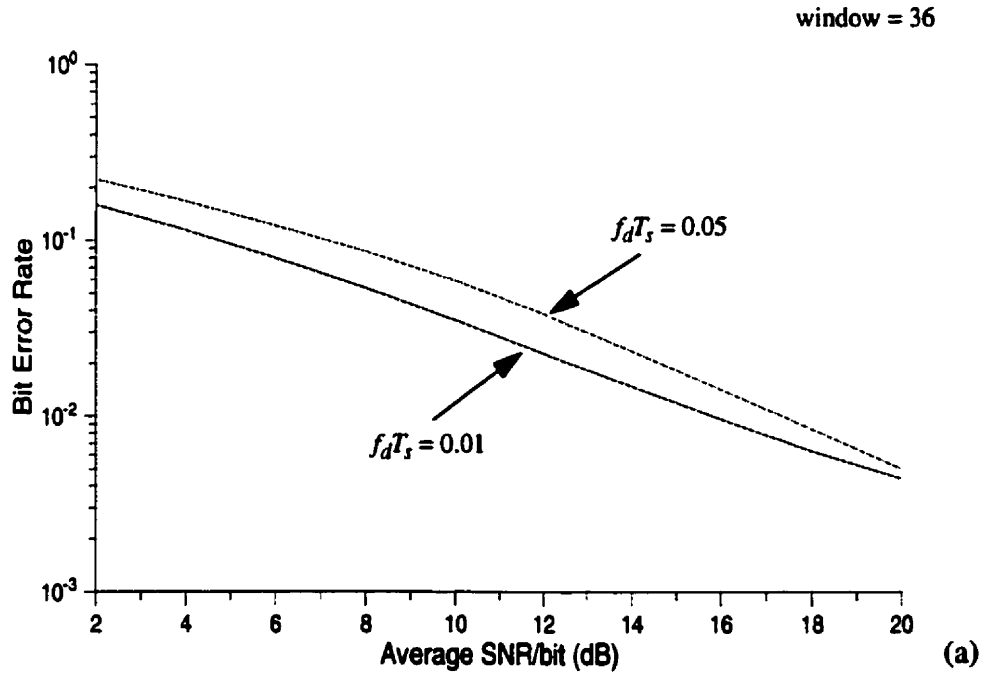


Figure 4.3 Error performances of uncoded QPSK signal with fading rates 0.01 and 0.05 at (a) block length = 512 bits and (b) block length = 2048 bits.

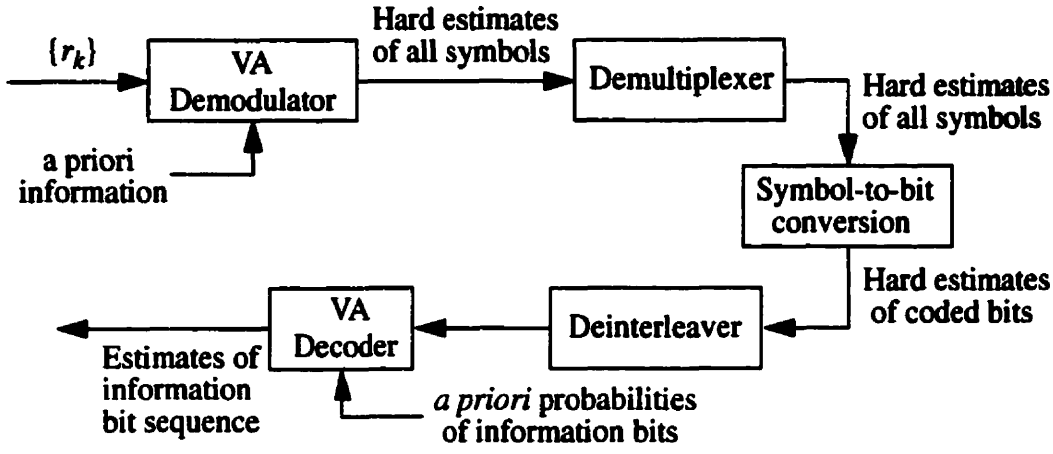


Figure 4.4 Two stage VA implemented receiver.

The next section looks at the implementation of SOVA, which is suboptimum in generating soft information, to the two stage receiver.

4.2 Joint Detection by SOVA (Type-I)

Channel demodulation in the two stage receiver involves an M-ary trellis. Of the two modes of SOVA available, only the register exchange mode was extended to deal with this kind of trellis structure [24]. This section studies receivers having this version of SOVA as the channel equalizer. Moreover, both iterative and non-iterative processing are considered. When non-iterative detection is involved, the second stage decoder does not need a SISO structure and decoding can be performed by using the classical VA. When iteration is desired, the decoder has to feed back soft information and decoding has to be carried out by another SOVA block.

4.2.1 Non-Iterative Processing

As can be seen in figure 4.6, the channel demodulator of the two stage receiver takes the sufficient statistics and APs to generate soft-outputs. When SOVA is in use, these soft-outputs are the approximation of the "true" APPs of the data symbols. Even though hard-outputs are also generated in the demodulation process, they are discarded and only the soft information is circulated. These soft-outputs are deinterleaved and passed to the second stage decoder as usual. The receiver described has been simulated and results are

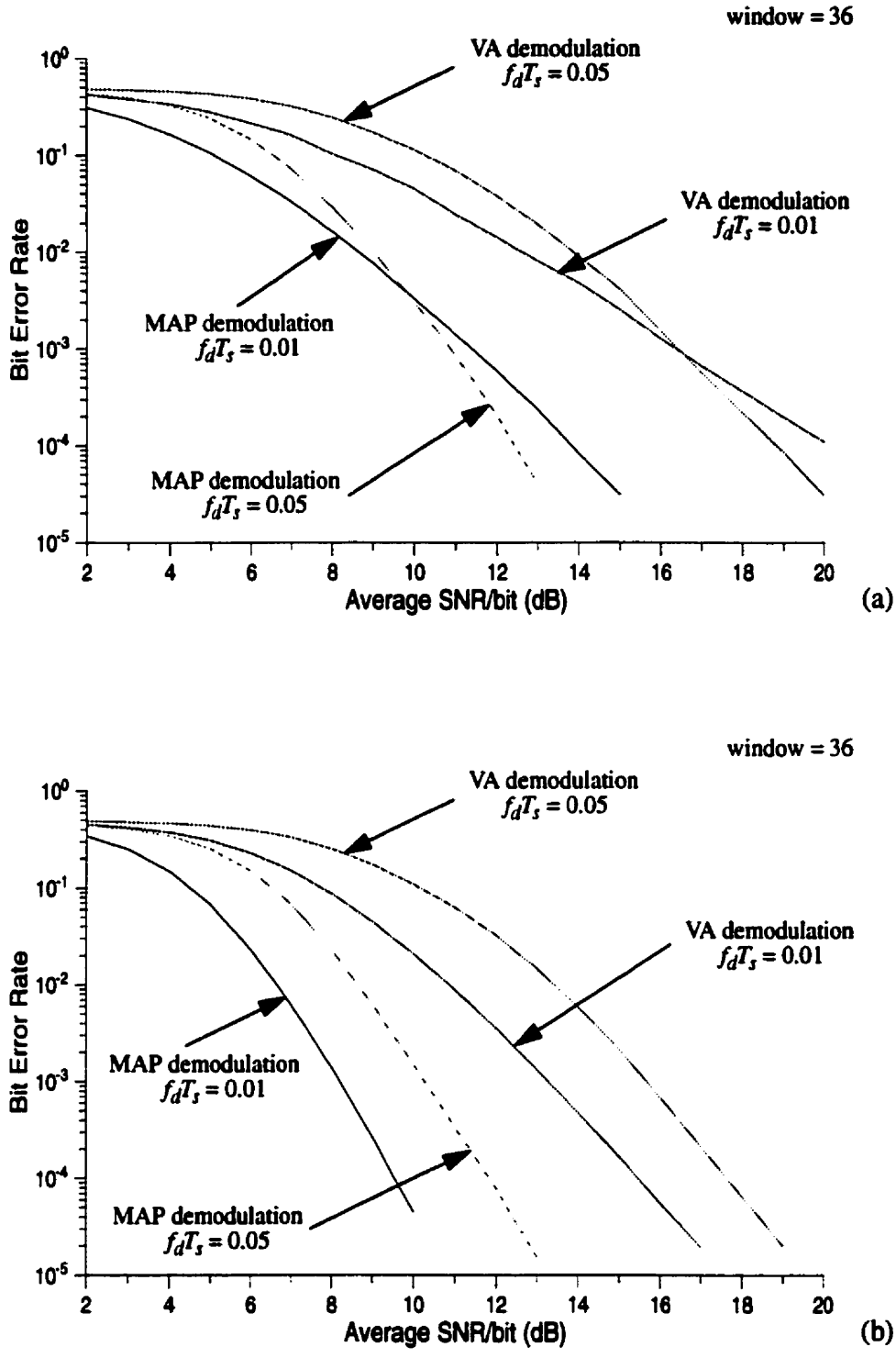


Figure 4.5 Error performances of the coded system demodulated by MAP algorithm and VA with (a) block length = 512 bits and (b) block length = 2048 bits.

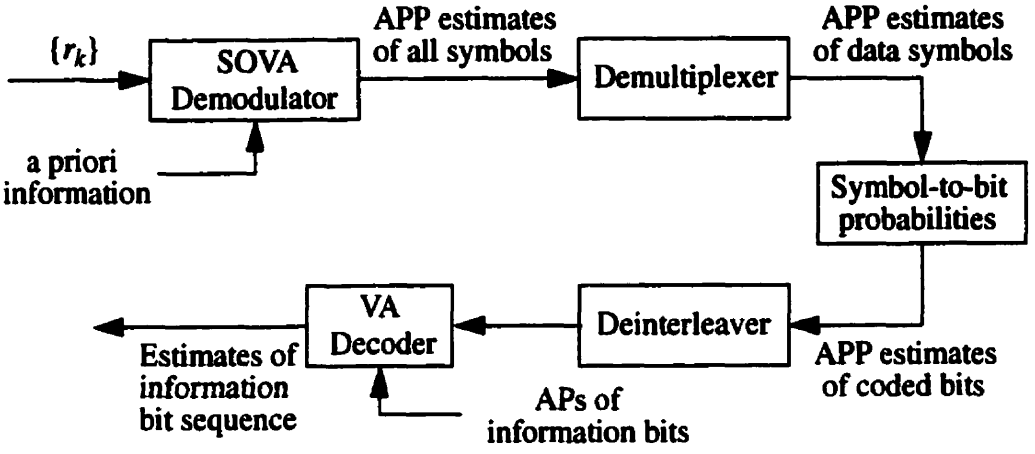


Figure 4.6 Two stage receiver with SOVA implemented at demodulator.

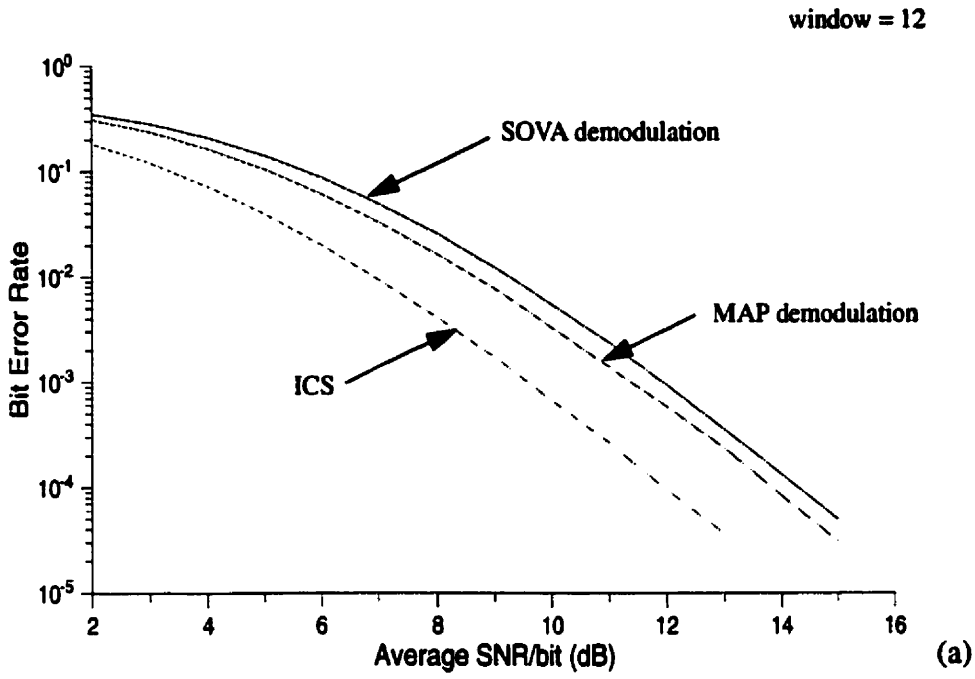
given in figures 4.7 and 4.8. Also plotted in the graphs are the receiver performance with ICS.

Under slow fading and a block length of 512, the SOVA receiver is 2.4 dB worse than the receiver with ICS. This is reduced to 1.9 dB when the block length changes to 2048, a trend that is consistent with the MAP receiver. More importantly, compared to MAP equalization, demodulation with SOVA exhibits only little degradation in performance. The loss is approximately 0.5 dB for both block lengths. This loss suggests that, under this environment, SOVA is a good substitute for the MAP algorithm.

The gap between MAP and SOVA demodulation becomes larger as the fading rate increases to 0.05. The differences are 1.9 and 2.5 dB with the former applying to the large block length. Hence, the SOVA receiver is quite sensitive to the change in fading rate. This is possibly due to the decrease in prediction accuracy as the fading rate increases. The situation improves for the system with a large block length because of more efficient interleaving. Nevertheless, the 1.9 dB difference is still significant compared to the 0.5 dB in slow fading. Hence, for a reliable system working under the fast fading scenario, applying SOVA for channel equalization is not recommended unless the system is prepared to sacrifice more power in exchange for better performance.

4.2.2 Iterative Processing

It has been mentioned that an algorithm with a SISO feature is a prerequisite for



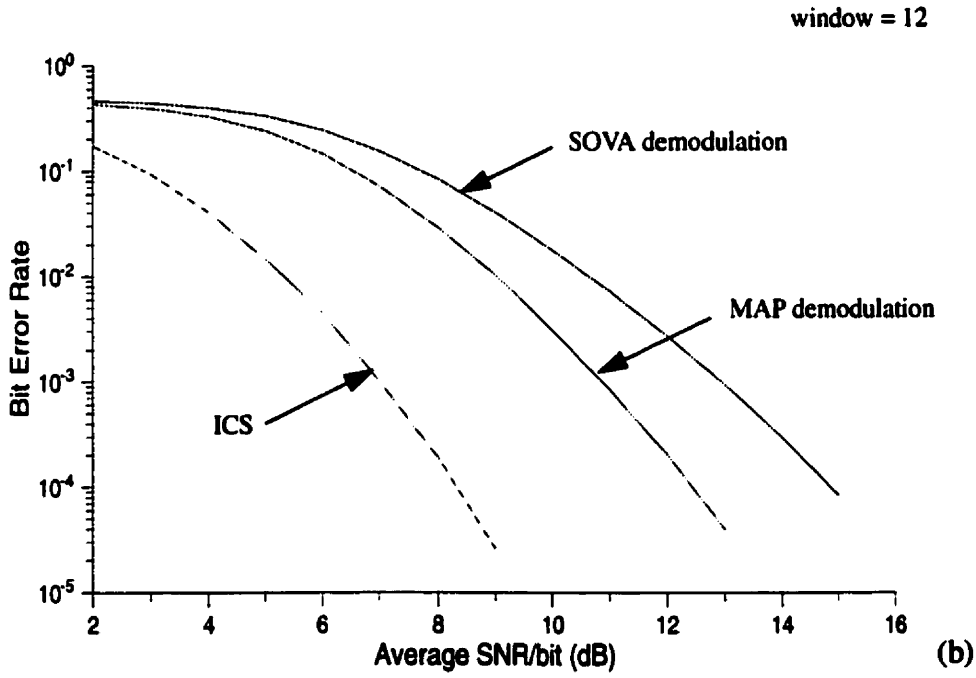
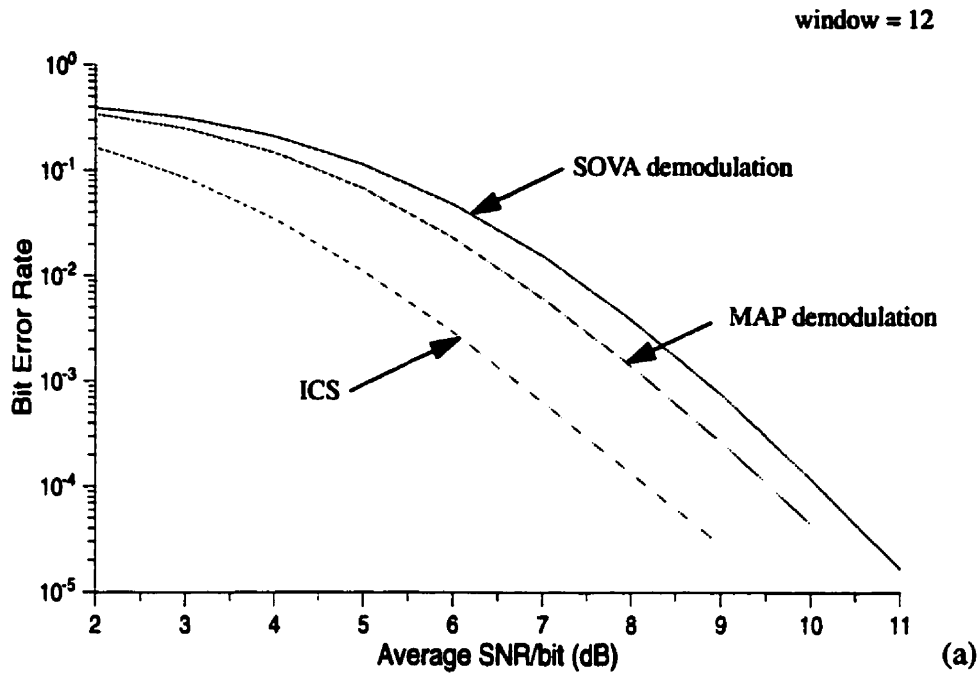


Figure 4.7 Error performances for coded interleaved system with block length 512 bits at (a) fading rate = 0.01 and (b) fading rate = 0.05.



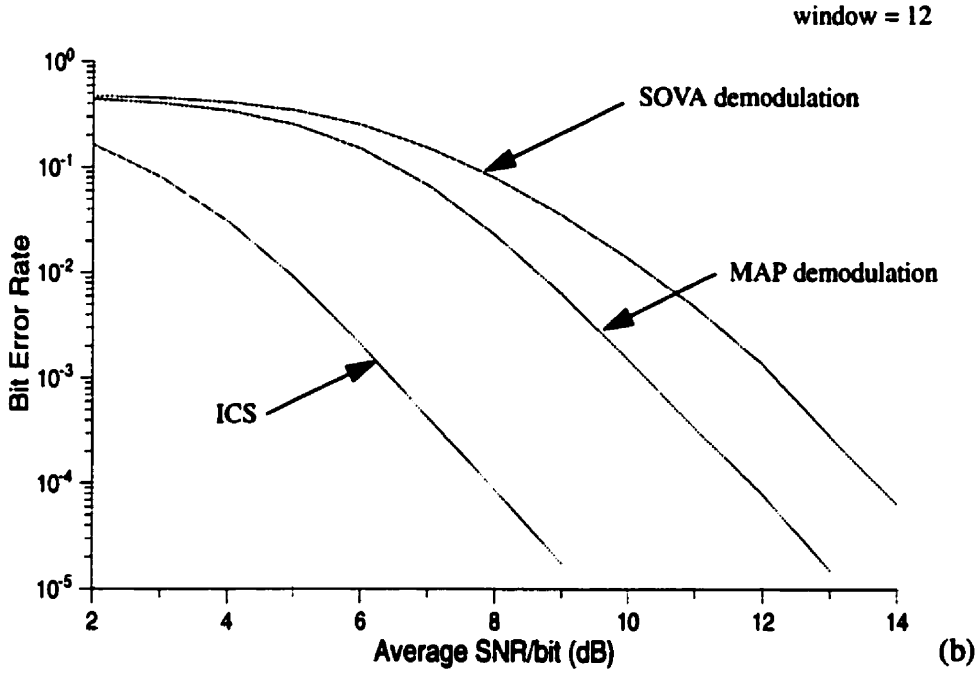


Figure 4.8 Error performances for coded interleaved system with block length 2048 bits at (a) fading rate = 0.01 and (b) fading rate = 0.05.

iterative processing. The SOVA discussed here is another legitimate candidate to fill this job. Successful iterative processing relies on efficient circulation of soft information within the loop so that for the problem studied here, an exchange of APPs of coded bits between the channel demodulator and decoder is involved. The register exchange mode SOVA is used for channel equalization.

For the convolutional code decoder, one may suspect that either mode can be applied. However, simulation results have shown that the register exchange SOVA is inappropriate. This mode of SOVA can provide a good approximation to the APPs of the information bits but not for the coded bits. One possible reason is related to the way this mode of SOVA is implemented. As discussed in appendix B, this mode of SOVA is seldom implemented in full because of its relatively high complexity. A simplified version, known as vertically updated SOVA (V-SOVA), is usually implemented. To illustrate the V-SOVA, consider the example of figure 4.9. For an arbitrary fixed state s_k, two paths enter this state. When looking back along these paths, information symbols on the respective survivor paths that cause state changes at the time slot between k-3 and k-2 always differ. As explained in appendix B, when a reduced complexity V-SOVA is considered, the APPs are estimated at this time slot. This ensures that the APP approximation for "1" and "0" are non-zero and thus some soft information is obtained for the "unlikely" bit.

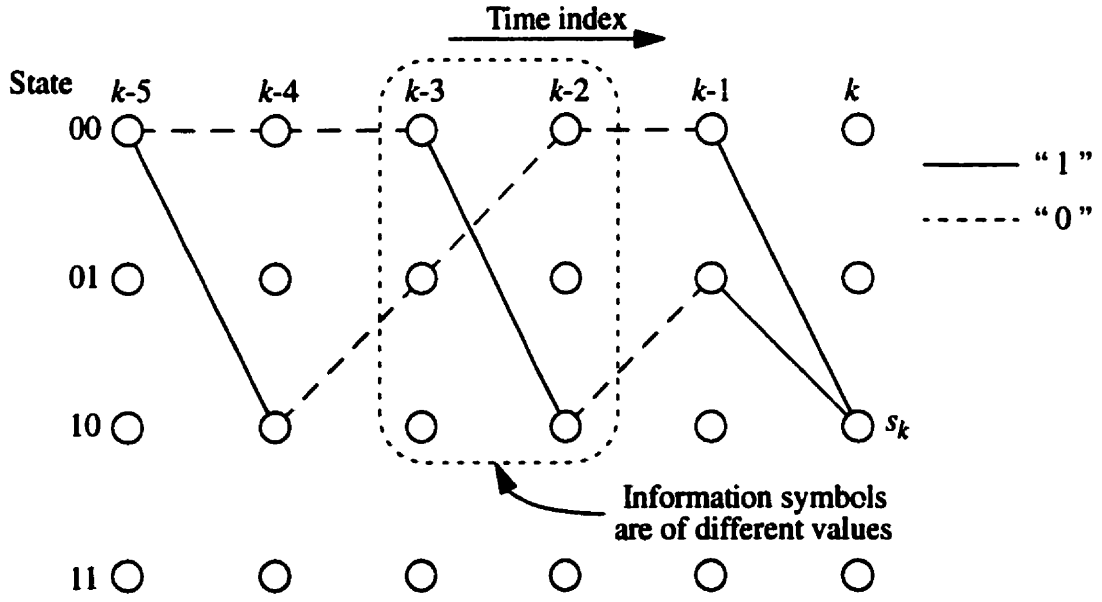


Figure 4.9 Four-state trellis with memory length two and two transitions per state.

When the APPs of coded bits are needed, this version of SOVA fails. The main reason is, at that particular time slot, there is no guarantee that the coded bits associated with the two paths are different. It is quite possible that the coded bits on these two different paths are the same (say "1"). Consequently the APP approximation becomes either one or zero, which is equivalent to hard decision detection. Thus, the register exchange mode SOVA is not capable of computing reliable soft information for coded bits.

An alternative is the trace back mode SOVA discussed in appendix B.2. The characteristic of this version is that soft information for the "unlikely bit" is always retained. Such a characteristic makes it feasible to extract soft information for the coded bits.

Just like the iterative MAP receiver, it is necessary to extract intrinsic and extrinsic information after the demodulator and decoder. As can be seen in figure 4.10, the only difference in the receiver structure is the use of SOVA as the SISO processor. The iterative receiver implemented with the mixed modes of SOVA is termed a "type-I SOVA receiver" in this thesis. Simulation was carried out with the same parameters that appeared in the last chapter. As SOVA is suboptimal in generating soft information, it is expected that more loops are required to achieve an error performance comparable to that of MAP. Up to five iterations are done for each case and the results are shown in figures 4.11 and 4.12.

Results for the system operating in slow fading indicate that about 1 dB can be gained after the first iteration. However, only a marginal improvement is achieved after further iterations. When a large block length is used, the curves corresponding to the first and fifth iterations tend to merge at a SNR of 10 dB. This observation suggests that the system may soon run into an error floor if the SNR increases beyond this point.

Coding gain obtained after iterations is greater for fast fading. Note that the gap between the ICS curve and the no iteration curve is large, which means that the initial APPs computed by SOVA in the channel equalizer are quite inaccurate. Iteration helps by supplying better a priori information to the channel demodulation so that more precise APPs can be calculated. The gain after one iteration is about 2.7 dB and a further gain of 1.3 dB is observed after five iterations. These numbers are roughly the same for both block lengths. Nevertheless, even with this amount of coding gain, the curve is still far away from the one labelled ICS. The differences are recorded to be 2.5 and 2 dB with the former occurring for the smaller block length. In contrast to the 1 dB difference for the slow

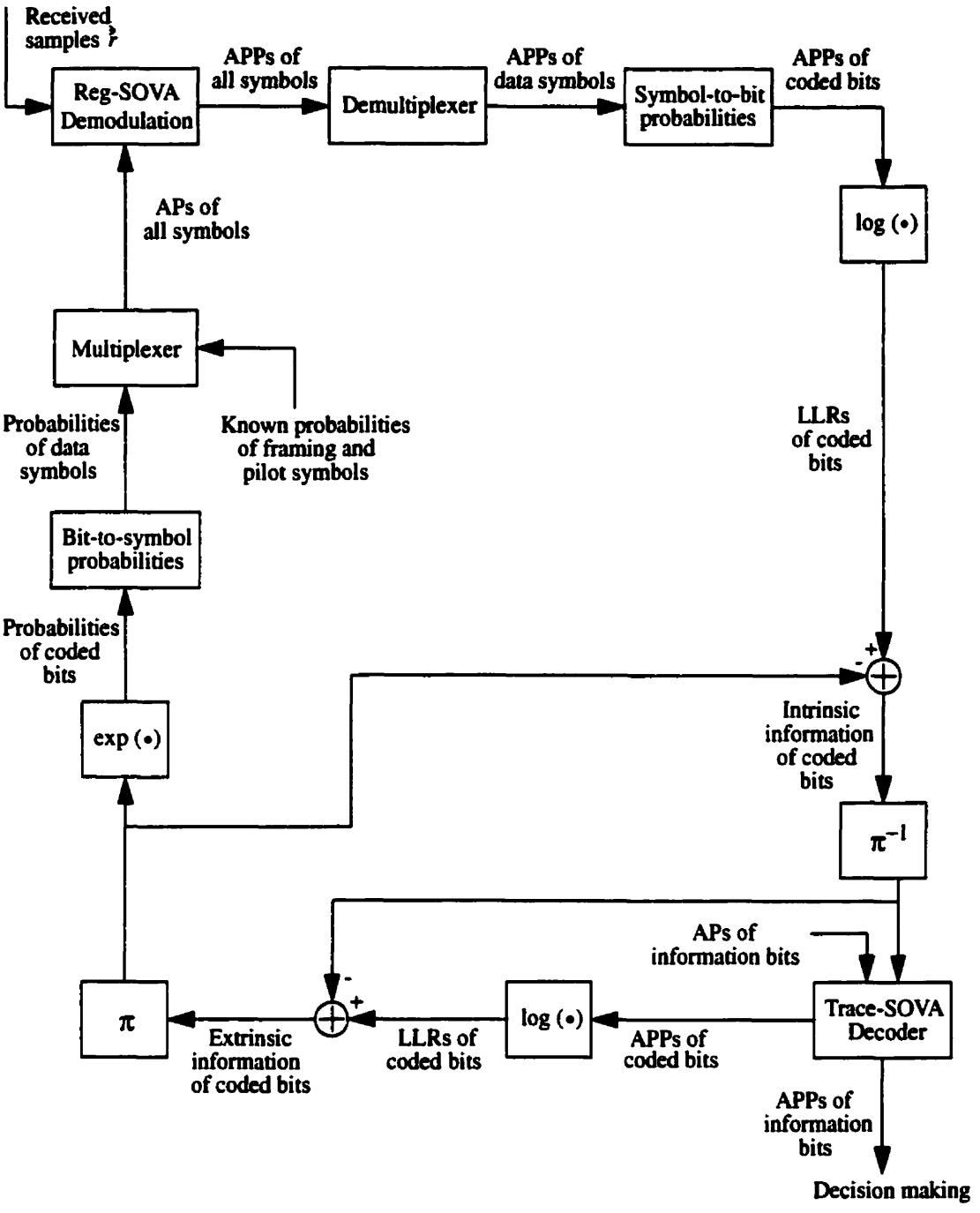
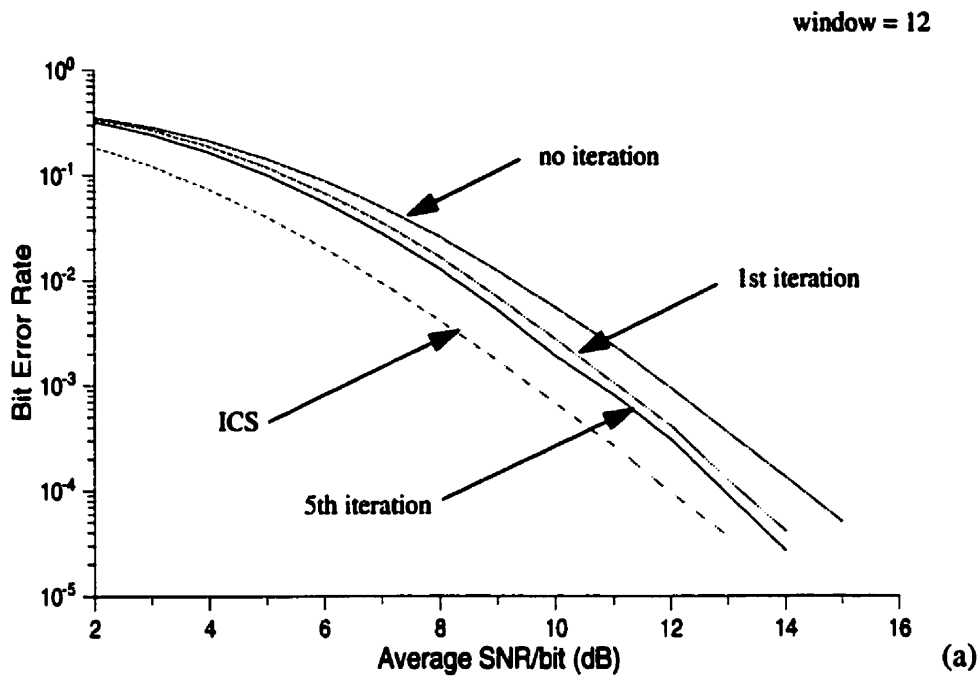


Figure 4.10 Iterative SOVA processing for joint demodulation and decoding of *M*-ary coded signal transmitted in fading channel.



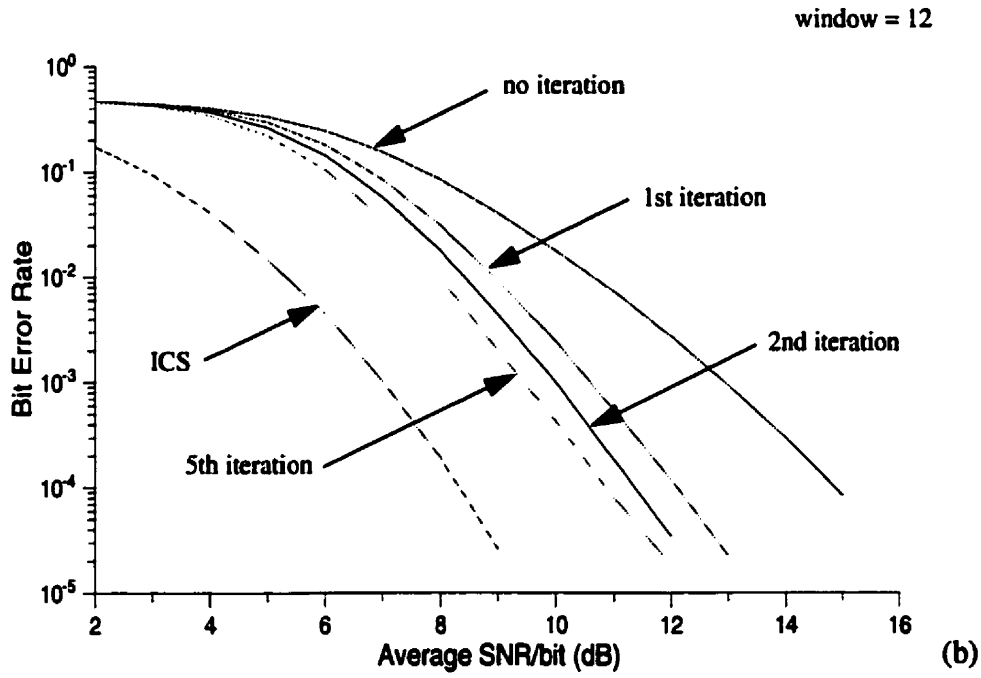
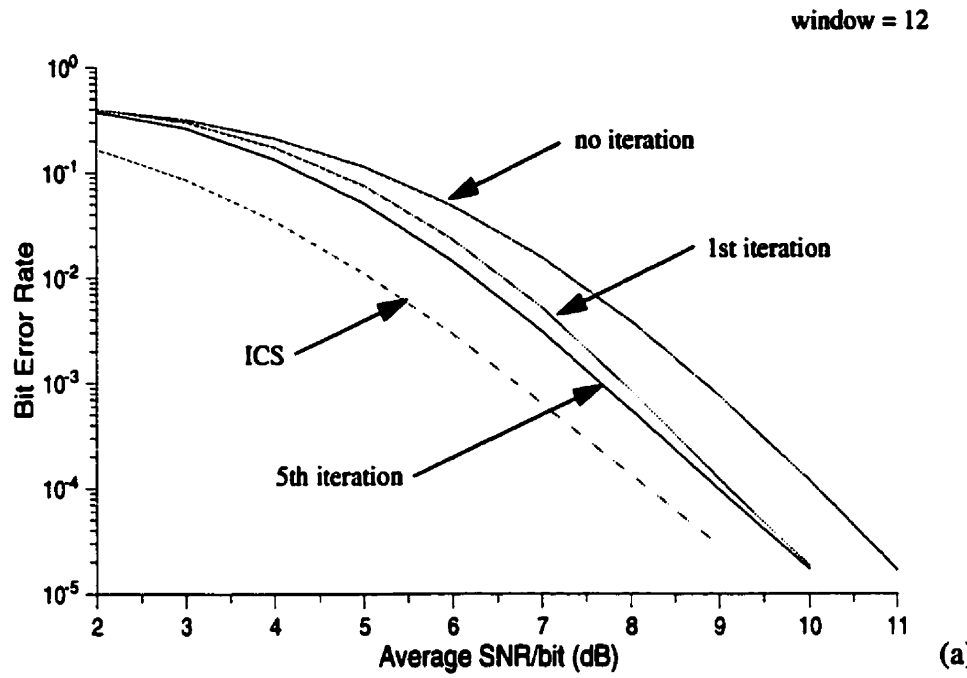


Figure 4.11 Error performances for iterative coded interleaved system with block length 512 bits at (a) fading rate = 0.01 and (b) fading rate = 0.05.



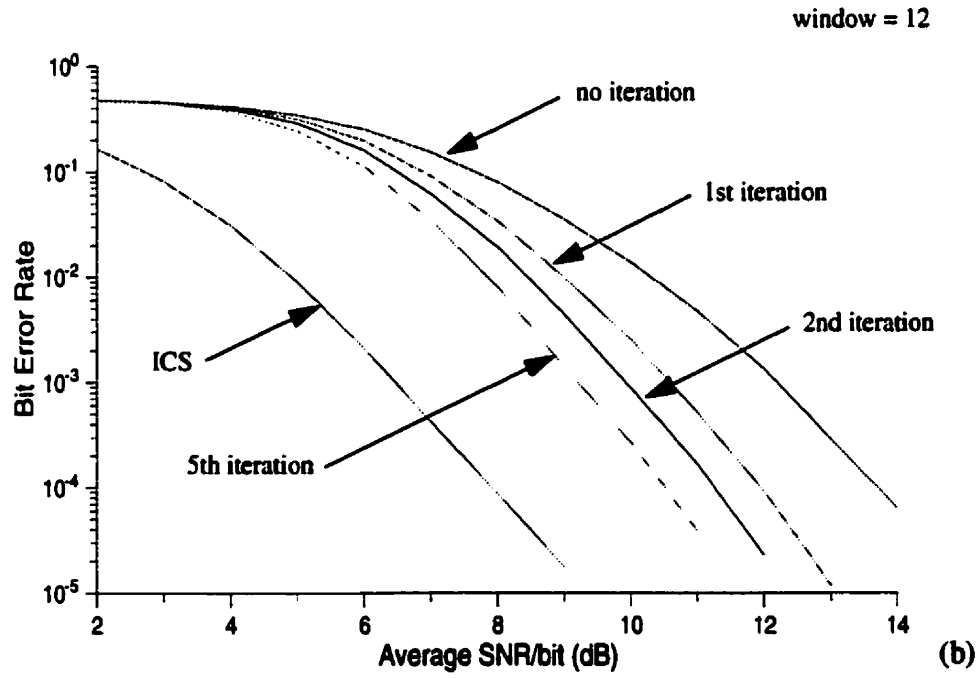


Figure 4.12 Error performances for iterative coded interleaved system with block length 2048 bits at (a) fading rate = 0.01 and (b) fading rate = 0.05.

fading scenario, this loss is quite big. As a result, although iteration is carried out with the SOVA receiver, it is still not recommended to be used in the fast fading channel.

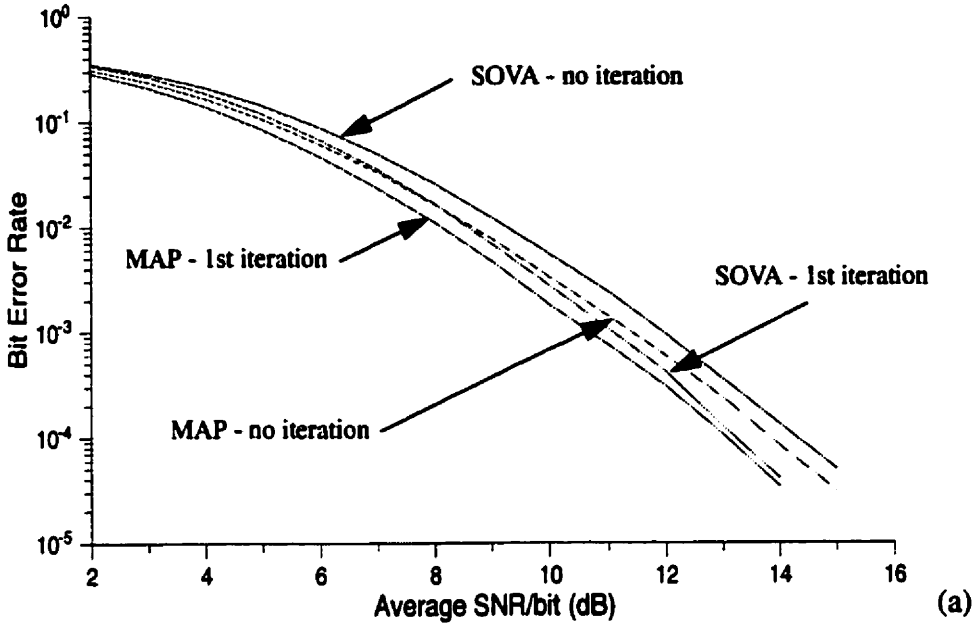
4.2.3 Error Performance Comparison

As mentioned, SOVA is suboptimal in calculating APPs. To see how much better the optimal MAP algorithm is in iterative processing, results of error performances corresponding to both are gathered in figures 4.13 and 4.14. For slow fading, the optimal method provides a superiority of about 0.5 dB when no iteration is performed. Nevertheless, the gap has been reduced to 0.1 dB after the first iteration. These figures are valid for systems under both block lengths. This observation suggests that under the slow fading environment, the key for successful detection is the accuracy of the prediction and, even if the detection method is suboptimal, an acceptable performance can still be attained.

A different behaviour holds for fast fading. With no iteration, the error performances achieved with the two detection methods are very different. Results have indicated that, in terms of the error performance, one iteration with SOVA is roughly equivalent to detecting non-iteratively with the MAP algorithm. It is also worth noting that, even if as many as five iterations are done, the performance of SOVA is still not as good as the MAP algorithm with one iteration. The difference varies from 0.5 dB to 0.75 dB depending on the block size. Simply stated, under an environment in which the prediction error is high, the system has to rely on the optimal method to compute probabilities to achieve a more reliable communication. Applying SOVA is not recommended.

4.3 Joint Detection by SOVA (Type-II)

In the preceding section, an ordinary SOVA was applied to the problem of estimating APPs for symbol sequences that can be represented by a trellis. It was found that, for the purpose of iterative processing, the receiver structure had to utilize mixed modes of SOVA. Channel equalization was done by register exchange SOVA which is the only mode available to deal with an M-ary trellis while the trace back mode was employed to generate soft information for the coded bits. This section considers an alternative approach, based on the VA, to obtain soft information from a trellis. The design objective



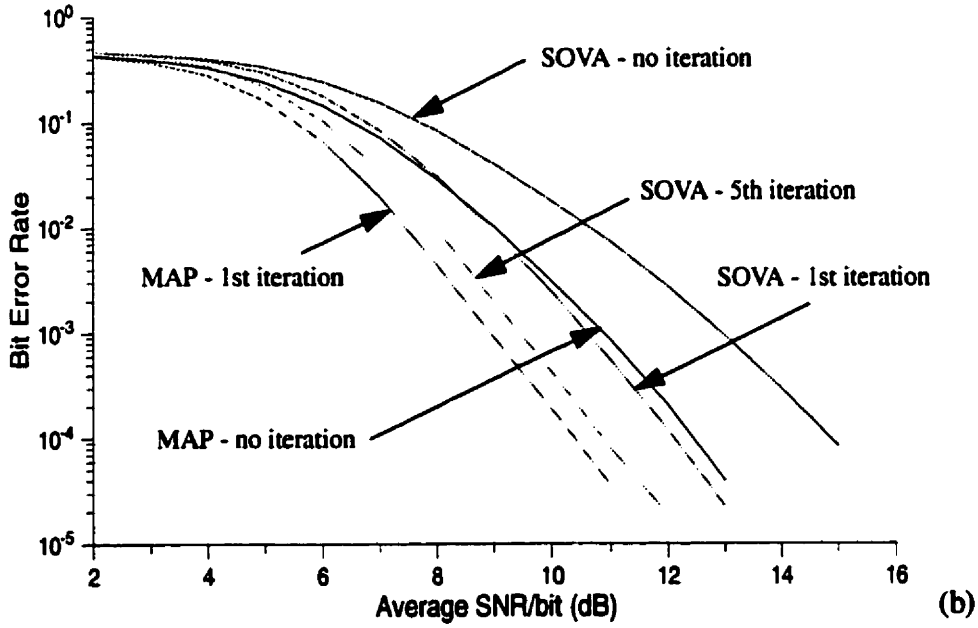


Figure 4.13 Comparison of system performances detected with MAP and type-I SOVA with block length 512 bits at (a) fading rate = 0.01 and (b) fading rate = 0.05.

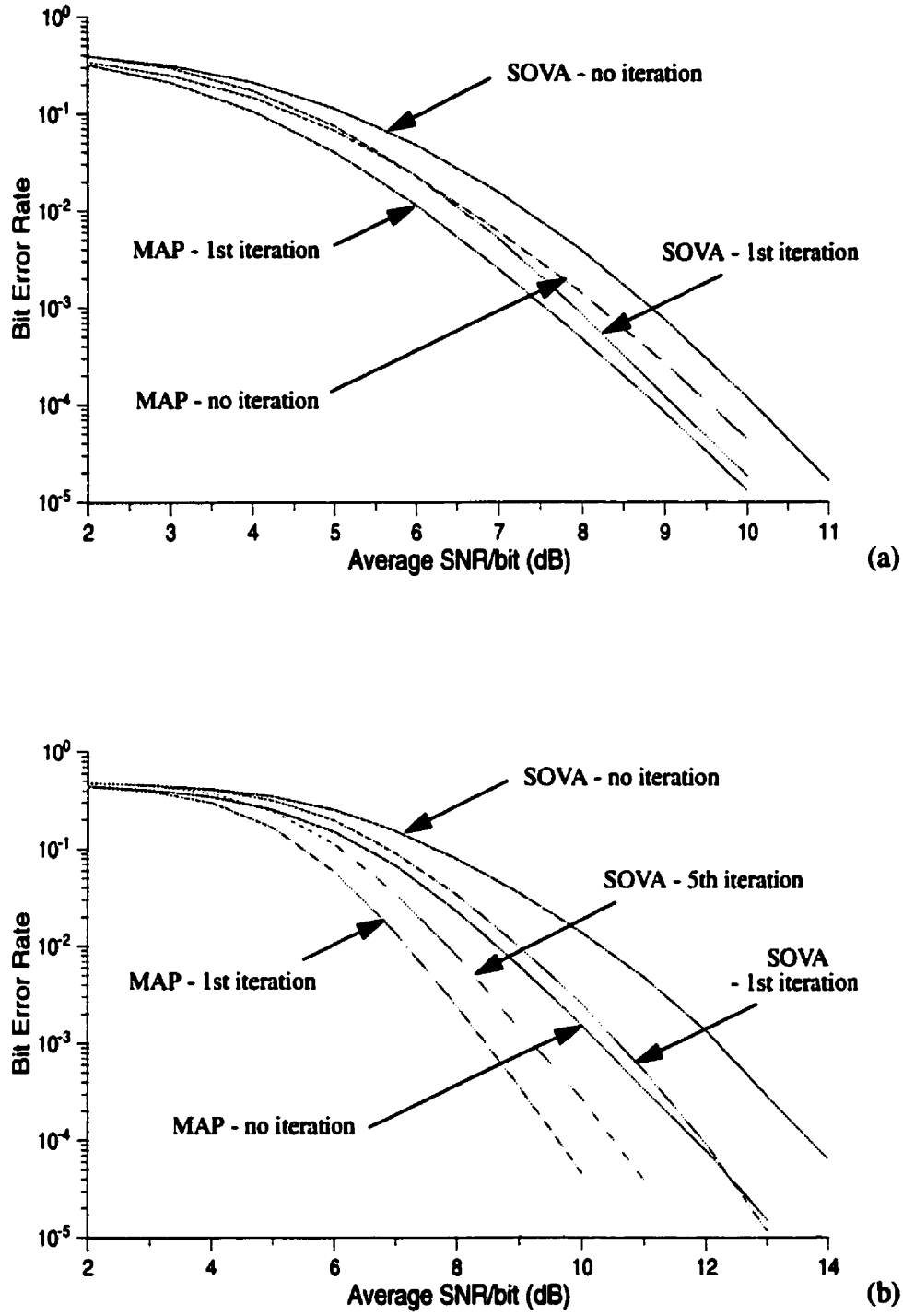


Figure 4.14 Comparison of system performances detected with MAP and type-I SOVA with block length 2048 bits at (a) fading rate = 0.01 and (b) fading rate = 0.05.

for the algorithm is to deliver soft-outputs for both information symbols and coded bits. Under this criterion, there is no need for a mixed mode such as the type-I SOVA receiver.

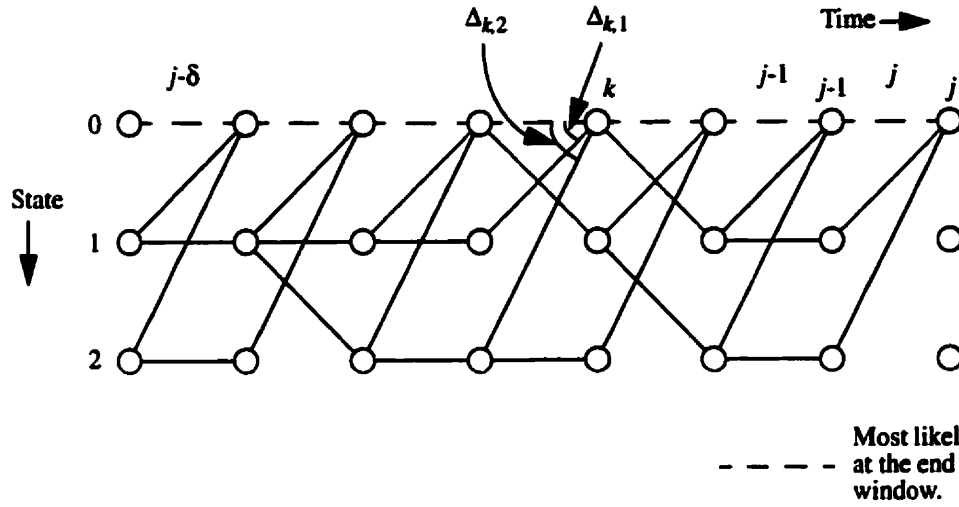
4.3.1 Detection Algorithm

The proposed method is motivated by an idea called list Viterbi algorithm (LVA) proposed by Hashimoto in [23]. LVA is just a generalization of the standard VA which produces a rank ordered list of the L globally best paths in a trellis search [35, 42]. Consider an M-ary trellis. The VA calculates the cumulative metrics of the M paths merging at state s_k, where k is the time index, and then chooses the path with the largest metric as the survivor path. The other M-1 paths, called non-survivor paths, are discarded. Besides the survivor path and its cumulative metric, the LVA also computes and stores the differences between the survivor metric and non-survivor metrics entering this state. This procedure is followed here.

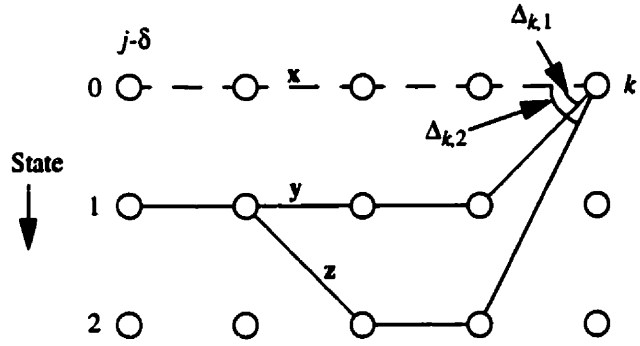
To illustrate the detection algorithm proposed here, consider figure 4.15 where part of a ternary trellis is shown. Suppose it is at the end of a decision window for the standard VA and that the all zero path has the largest metric among all survivors, which implies it is the most likely path. The difference between the survivor metric and non-survivor metric entering a specific state s_k of this most likely state sequence, at time k, is given by

$$\Delta_{k,i} = (\textit{survivor metric}) - (\textit{non-survivor metric})_i \qquad (4.4)$$

where i is the index of one of the two non-survivor paths entering s_k. As shown in the diagram, when looking back in time there are $(M\text{-}1)(\delta+1)$ non-survivor paths along the most likely path where $\delta+1$ is the number of symbols involved in a decision window. $\delta+1$ is also the length or number of symbols associated with the most likely path. When tracing back, these non-survivor paths are either unmerged or merged at most once with the most likely path. As shown in figures 4.15 (b) and (c), some of the non-survivor paths may have a length less than that of the most likely path due to the way the VA proceeds. They can be extended to the same length by joining the required portion of the most likely path, as shown in figure 4.15(c). One point is worth mentioning. For the non-survivors merging with the most likely path at time j, there is no need for an extension because they already are of the same length as the most likely path.



(a)



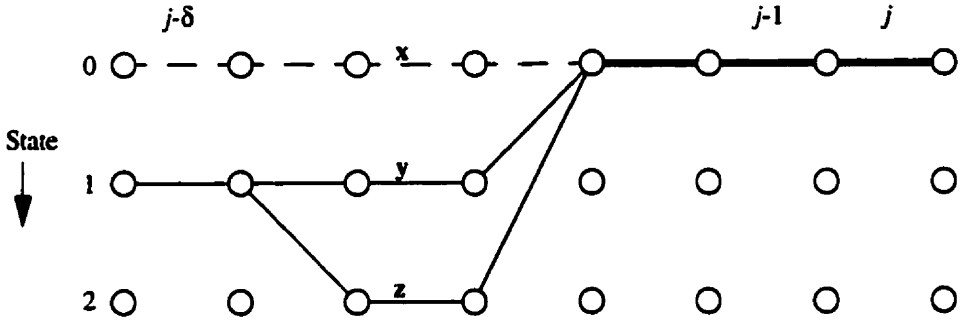
$\Delta_{k,1}$ = (path metric)$_x$ - (path metric)$_y$

$\Delta_{k,2}$ = (path metric)$_x$ - (path metric)$_z$

x is the survivor path entering state "0" at time k.

y and z are the discarded non-survivor paths entering the mentioned state.

(b)



(c)

Figure 4.15 (a) Example trellis showing the end of a decision window, (b) metric differences calculation and (c) path extension to align the non-survivor paths having length equal to the decision window.

Let p be the index of all paths where $0 \le p \le (M-1)(\delta+1)$ and $p = 0$ denotes the most likely path. Assume further that the most likely path has a cumulative metric equal to Γ_0. With reference to figure 4.15(a), any non-survivor path, after being extended to the same length as the most likely path, has a metric of

$$\Gamma_p = \Gamma_0 - \Delta_{k,l} \tag{4.5}$$

where l is the index of one of the two paths joining the most likely path at time k and $0 < p \le (M-1)(\delta+1)$. According to the path metric definition, the probability of the p-th path is given by

$$Pr\{path\ p\} \sim e^{\Gamma_p} \quad ; \quad 0 \le p \le (M-1)(\delta+1) \quad . \tag{4.6}$$

At the end of the decision window, the classic VA tries to trace back the most likely path to the beginning of the window to provide an estimate of the information symbol $u_{j-\delta}$ where the subscript is a time index. Probability estimation can be carried out in a similar manner. By tracing back the non-survivor paths along with the most likely path, the APP of the M-ary symbol q can be approximated as

$$Pr\{u_{j-\delta} = q|\vec{r}\} \approx \frac{\sum_{p \in A} e^{\Gamma_p}}{\sum_{\forall p} e^{\Gamma_p}} \tag{4.7}$$

where A is the set of the paths containing a symbol q at time j-δ and the denominator is a normalization constant. Notice that the method of choosing the set of paths to approximate the probability is the same as in the trace back mode SOVA. Also, the above calculation is not confined to computing the probability of the information symbol which causes a state change in the trellis. When it comes to finding the soft information of coded bits, $u_{j-\delta}$ can be referred to one of the coded bits transmitted at this specific time. To obtain the probabilities of symbols for the last window of a finite length trellis, one can let the time index j-δ run for the entire window length and evaluate equation (4.7) at every time slot. With this method, it is feasible to have the two stage receiver in single mode in contrast to the mixed mode type-I SOVA receiver.

A disadvantage of this method, that was found in preliminary simulation, is that it is quite sensitive to the size of the decision window. By simulating the fading channel detec-

tion problem, it was observed that the probability approximation is poor when the above method is implemented with either very large or small window sizes. Simulation trials have indicated that, when the decision delay is about three to six times the memory length, an acceptable performance can be obtained.

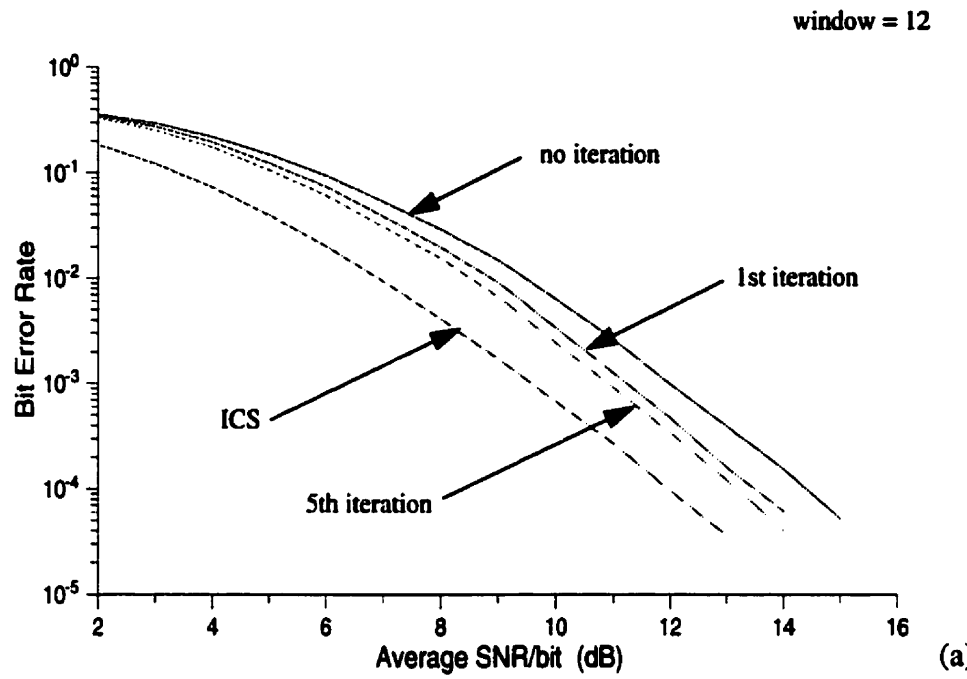
The above method can be improved further. In general, to compute the APP of a symbol requires consideration of all paths propagated in the trellis. Hence, if more paths are included in the probability approximation given in (4.7), a better estimation can be expected. Recall that the non-survivor paths are obtained along the state sequence of the globally best survivor at the end of window. The same procedure can be carried out for the other survivors. The drawback is obviously the increase in numerical computation.

As a final note, this method is actually not a straightforward modification of the trace back mode SOVA. For a problem which involves a binary trellis, the algorithm described here requires extension of the non-survivor paths which is not needed in the original algorithm given in [17]. However, because the way the set of paths is chosen is identical, this method is still considered to be an extension of the trace back SOVA.

4.3.2 Simulation Results

The above soft information calculation strategy was implemented for the two stage iterative receiver. The receiver structure is just a replica of the one shown in figure 4.10 except both the demodulator and decoder now employ the new trace back version of SOVA. Accordingly, the receiver is now in single mode and it is called a "type-II SOVA receiver". Shown in figures 4.16 and 4.17 are the error performances for the system with the same parameters as before. In slow fading, the coding gain after the first iteration is about 1 dB and the improvement between first and fifth iterations is only 0.2 dB. The same observation can be made for the system with either block length.

For a larger fading rate, the amount of coding gain becomes significant. The improvement is about 2.3 dB after the first iteration and a further 0.7 dB is recorded when five iterations are performed under small block length usage. These figures vary somewhat for the large block length but do not differ by much. By looking closely at the results of type-I and type-II SOVA receivers, it is found that they almost overlap each another. It is therefore sufficient to claim, in terms of error performance, that the two receivers are



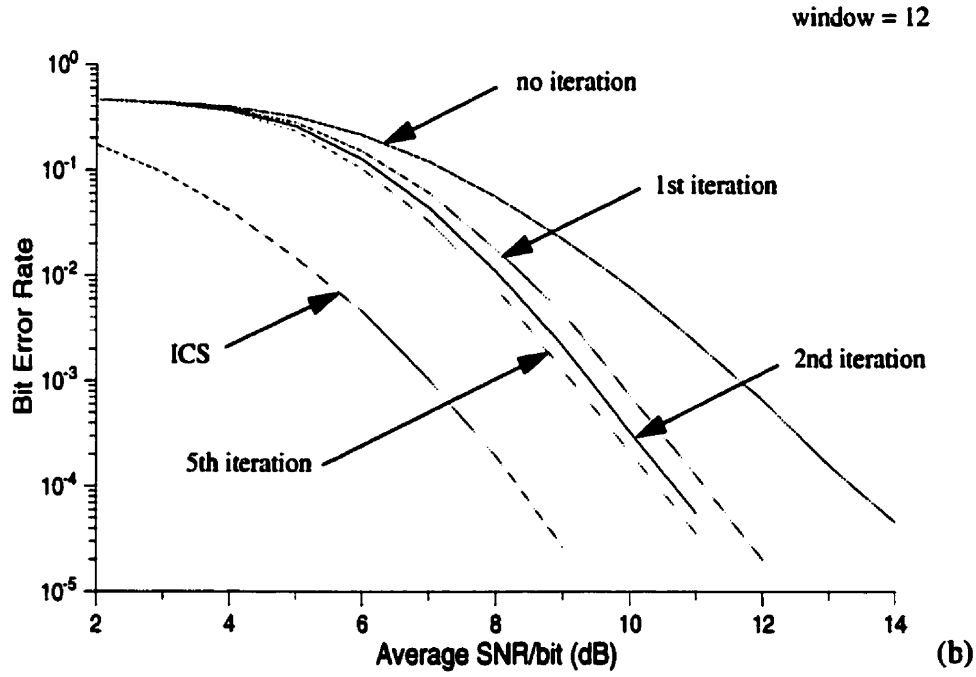
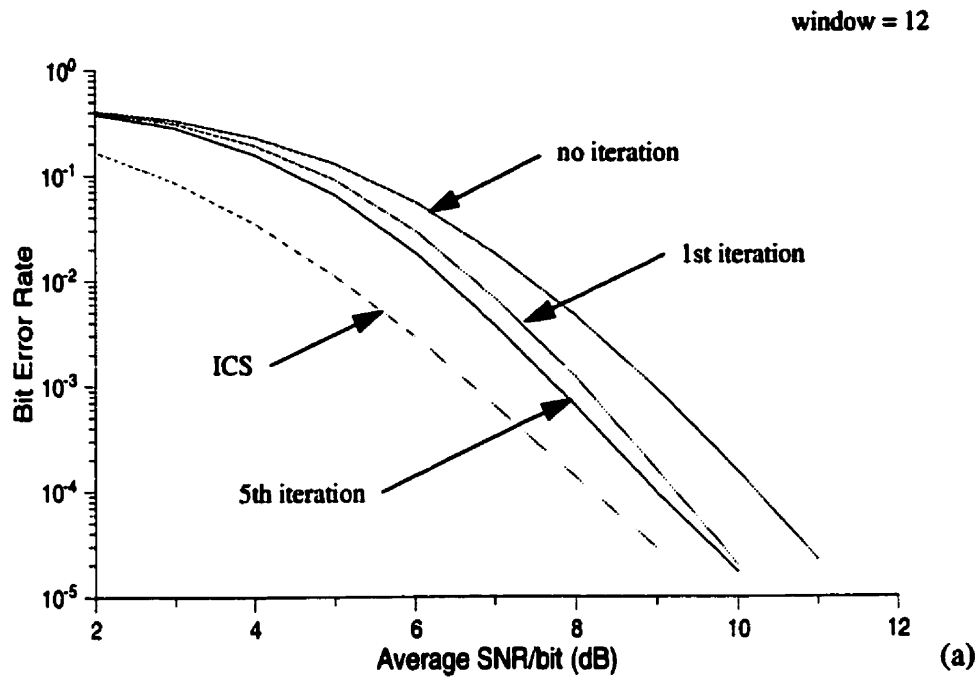


Figure 4.16 System performances of iterative type-II SOVA receiver with block length 512 bits operated at (a) fading rate = 0.01 and (b) fading rate = 0.05.



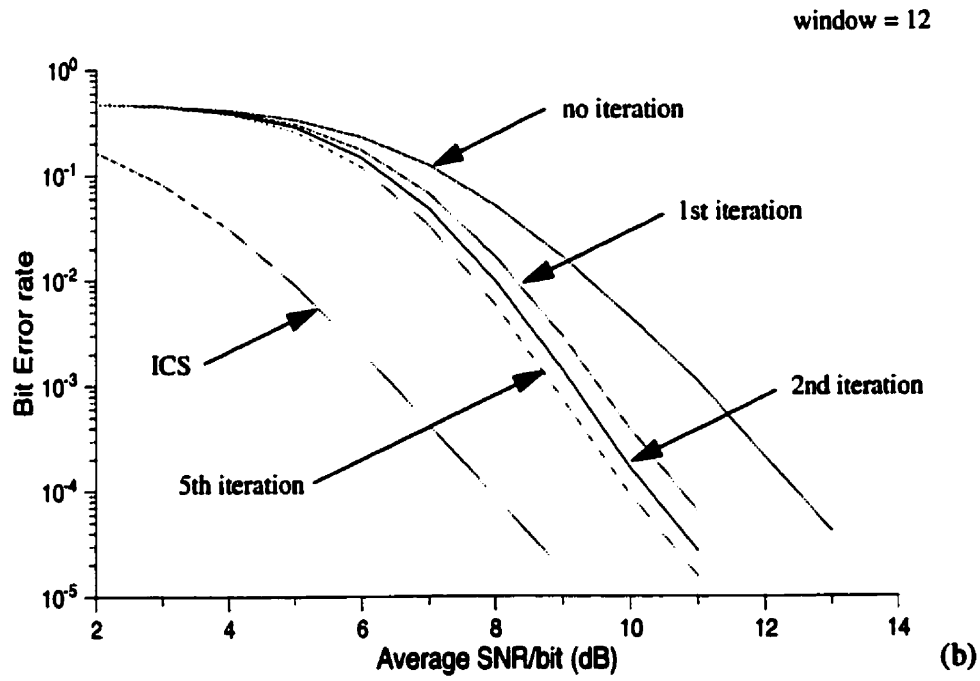


Figure 4.17 System performances of iterative type-II SOVA receiver with block length 2048 bits operated at (a) fading rate = 0.01 and (b) fading rate = 0.05.

equivalent. The comments on the type-I SOVA receiver given in section 4.2.2 and the conclusion on the error performance comparison with the MAP receiver still hold and, therefore, they are not repeated here.

4.4 Analysis of Computational Complexity

A complexity analysis can be done by considering the number of numerical operations involved in an algorithm. This section mainly focuses on comparing the three channel equalization techniques, namely MAP algorithm, register exchange mode SOVA and the new trace back version of SOVA. No attempt is made to include the ordinary trace back SOVA since it is not able to deal with a M-ary trellis. For the binary case, it is usually considered to have a degree of complexity equivalent to the other mode [17].

In view of the fact that SOVA assumes the branch weights of the trellis are in the *log* domain while the MAP algorithm does not have this requirement, it is reasonable to assume that the branch weights for each algorithm are in its desired form. It is not fair to assume that the branch weight computed from the prediction filter needs logarithmic operations and take this into the expense of the complexity of SOVA. Furthermore, the numerical calculations involved in the prediction filter are also not included simply because they are common to all three methods.

With the above assumptions, consider the general M-ary trellis illustrated in figure 4.18. As can be seen, there are M paths entering and leaving each state. This implies that $QM = M^{L+1}$ branches are present at a given time slot. Also, for the entire trellis there are $(N+1)Q$ states in total. The MAP algorithm is characterized by the forward (α) and backward (β) recursions given by equations (3.15) and (3.16) for calculating the accumulative weights at each state. Since the α for the states at time zero are pre-defined, NQ forward recursions are required. The same number applies to the backward recursion. As there are M paths entering or leaving a state, to calculate one specific α or β, the summation has M terms which means M-1 addition operations. Also, each term involves one multiplication. On the other hand, an intermediate variable σ is defined for APP calculation as given in (3.18). In order to save memory space, a direct substitution of (3.17) to the APP calculation can be assumed. The summation of the denominator has a total of QM terms and,

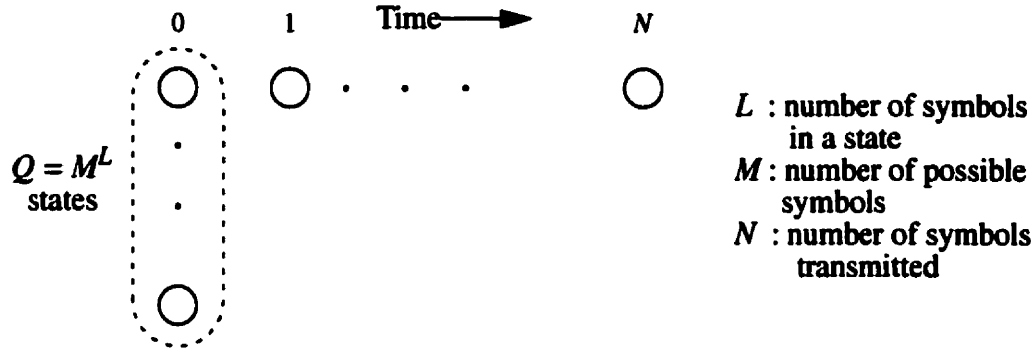


Figure 4.18 A typical M-ary trellis with length N.

because of the substitution, each term needs two multiplication operations. One can assume that the value of the numerator is calculated while computing the denominator because the terms involved are common.

The following table summarizes the above analysis when the MAP algorithm is performed to calculate the APP of all the transmitted symbols.

	Addition	Multiplication	Division	Memory required
α	$NQ(M-1)$	NQM	0	$(N+1)Q$
β	$NQ(M-1)$	NQM	0	$(N+1)Q$
APP	$N(QM-1)$	$2NQM$	MN	MN
Total	$3NQM-N(1+2Q)$	$4NQM$	MN	$2(N+1)Q+MN$

Table 4.1 Summary for the operation and memory requirements of the MAP algorithm.

Both register exchange mode and the new trace back mode SOVAs were developed on the basis of the standard VA. For this classical method, the algorithm calculates the potential survivor metrics entering a state by

Potential survivor path metric entering the state = Survivor metric from previous state + branch metric

and keeps the path with the largest value to be the survivor path, as described in appendix B. In addition to storing the survivor metrics, it is also necessary to reserve memory space for the survivor state sequences. As there are M paths entering a state, to determine a new survivor needs M additions and one maximization to choose the largest value. On the other

hand, suppose that the size of the decision window is δ. For the trellis depicted in figure 4.19, (N-δ+1) sliding windows have to be constructed. There will be Q survivors corresponding to the number of states at the end of each window. A decision is made for the symbol transmitted at the beginning of the window by tracing back the survivor path with the largest metric. This implies another maximization function which takes Q values. The trace back operation can be treated as a form of table look up and it is assumed to be easy to implement. Another advantage of using the sliding window VA is the reduction in memory space required because memory locations can be reused. The following summary is established.

	Addition	Max. function		Memory required
Survivor update	NQM	NQ (on M values)	Survivor metric	$2Q$
Sliding window	0	N-δ+1 (on Q values)	Survivor state sequence	(δ+1)Q

Table 4.2 Summary for the operation and memory requirements for the standard VA.

The simplified V-SOVA is considered here. As stated in appendix B, it is necessary to store the M path metrics entering a state to reflect the probabilities of symbols at time (k-L). The algorithm proceeds in the same way as the standard VA. Once the best survivor at the end of the window is decided, symbol probabilities at time (k-δ) are calculated by evaluating equations (B.3) and (B.4). They take M exponentiations, M-1 additions and one division. The same number of operations is needed for the symbols transmitted at other time slots. The extra number of operations and memories required for V-SOVA are as follows:

	Addition	Division	Exp.		Probability stored at each state	APP
Probability calculation	$N(M$-1)	MN	MN	Memory required	(δ+1)QM	MN

Table 4.3 Summary for the operation and memory requirement for the V-SOVA.

Finally, the new trace back version of SOVA is considered. At each state, once the survivor metric is decided, it calculates and stores the difference, Δ, between the survivor and non-survivor metrics. This process takes M-1 subtractions and memory locations. The algorithm follows the standard VA by selecting the survivor path with the largest metric at the end of a window and examines the discarded paths along this survivor. As many as $(M\text{-}1)\delta$ paths are chosen. Evaluating the probability for each of them by using (4.5) and (4.6) involves one subtraction and an exponentiation. Next, computing the APP at a discrete time slot based on this set of paths from (4.7) involves M divisions, one for each different symbol, and $(M\text{-}1)\delta$ additions. The above information is summarized in table 4.4.

	Addition	Subtraction	Exp.	Division		Memory required
Δ calculation	0	$NQ(M\text{-}1)$	0	0	Δ	$(\delta+1)Q(M\text{-}1)$
End of window probability calculation	$[(M\text{-}1)\delta]$ $(N\text{-}\delta+1)$	$[(M\text{-}1)\delta]$ $(N\text{-}\delta+1)$	$[(M\text{-}1)\delta+1]$ $(N\text{-}\delta+1)$	MN	APP	MN

Table 4.4 Summary for the operation and memory requirement for the trace back SOVA.

Notice that the numerical operations shown in the previous tables involve mostly floating point numbers. Actually double precision is preferable for the MAP algorithm to reduce the risk of singularity. In order to make a fair comparison, a standard has to be chosen to describe how complicated an operation is. For a popular general purpose digital signal processor (DSP) such as the TMS320C6000, the number of cycles required for addition and subtraction are the same. Assume that they require one operation unit. With reference to [47], the complexity of multiplication and division are about two times and three times that of addition, respectively. When it comes to exponential and maximization functions, the numerical complexity depends on how they are realized in software since they are not built-in functions of the DSP chip. The exponential function can be expressed equivalently in a power series and evaluated by summing a fixed number of terms. It can be assumed to be roughly ten times as complicated as an addition [48]. Considering maximization, the aim is to find the path metric which has the largest value and thus the task is a little more involved than just finding the maximum value from a set of numbers. Its complexity relates to the number of metrics to be compared as well. To find the maximum

value from a set of J values, the process takes roughly $k \cdot J$ equivalent addition operations where k is less than unity [48]. Here the author boldly assumes that the more difficult operation considered here needs $2J$ equivalent operation units.

Combined with the above assumptions, the memories and numerical operations required for the three channel equalization techniques are obtained and shown in table 4.5. The two types of SOVA that appear in the table have already incorporated the use of the conventional VA.

	operation units	memory sizes
MAP algorithm	$11NQM+3MN-N(1+2Q)$	$2(N+1)Q+MN$
Register exchange SOVA	$3NQM+14MN+$ $(N-\delta+1)2Q-N$	$(\delta+3+\delta M)Q+MN$
Trace back SOVA	$(12M\delta-12\delta+10+2Q)(N-\delta+1)$ $+4NQM-NQ+3MN$	$(\delta M+M+2)Q+MN$

Table 4.5 Numerical operation and memory requirements for the three techniques.

To compare the algorithms, first examine the column of the memory usage. Observe that the last term MN is the same. Assuming that $\delta \ll N$, then it is obvious that the MAP algorithm takes a lot more memory space. The trace back SOVA takes less memory because, as δ is usually larger than M, $(\delta M+M+2)Q$ should have a smaller value than $(\delta+3+\delta M)Q$. However, the difference is only marginal.

It is hard to say which algorithm needs more numerical operations just by reading the entries of the table. With $\delta \ll N$, it is reasonable to assume that $(N-\delta+1) \approx N$. Also, let the maximum delay chosen in the VA be Q such that $\delta \le Q$. Under these assumptions, the entries of the two SOVA become

- Register exchange SOVA : $3NQM+14MN+2NQ-N$
- Trace back SOVA : $16NQM-11QN+10N+3MN.$ (4.8)

An easier way to make the comparison is to look at the coefficient of NQM because the other terms are less important. Unexpectedly, the trace back SOVA involves the largest number of operations, the next one is the MAP algorithm and the register exchange SOVA needs the least. The reason is that to arrive at (4.8), δ is substituted by Q for the extreme case. Recall that the decision delay should not be kept too large or small in order to

achieve a good error performance when realizing the trace back SOVA. Having a large δ will lead to both a bad error performance and an unreasonably high complexity. In the simulation, δ is chosen to be approximately 1/5 (say 1/4 for the ease of calculation) of Q. Under this circumstance, the number of operations is

- Trace back SOVA : $7NQM-2QN+10N+3MN$. (4.8)

The number of operation units required is still more than that of the other version of SOVA but it has been significantly reduced. Now the MAP algorithm needs more numerical operations than the others.

After the complexity and performance analysis, one may be at a position to answer the question of which channel equalization technique is the best. It is completely dependent on the design specification. If complexity and memory size are not a concern, the MAP algorithm is the best candidate because it can provide the best performance. If a high speed receiver is desired, the ordinary register exchange mode V-SOVA can be employed since it needs the least number of numerical operations. If memory space is strictly limited, the newly developed trace back SOVA should be considered.

4.5 Summary

In this chapter another detection criterion MAP-SE which minimizes the probability of sequence error has been studied. One tool to achieve such criterion is the VA, which traditionally delivers hard-output only. In order to achieve a better error performance, the idea of SOVA was introduced. An effort has been made to extend the trace back mode of this algorithm so that it can handle the M-ary trellis. This new trace back SOVA and the ordinary register exchange SOVA were all simulated and the results have shown that they are equivalent. In a slow fading channel their performances are not far from the MAP algorithm. Therefore they can serve as very good substitutes for the MAP algorithm. Under a higher fading rate, the situation becomes worse. Even if five iterations are carried out for either SOVA receiver, it still cannot achieve an error performance as good as the first iteration of the MAP algorithm. In this circumstance, using these methods for detection is not recommended in terms of the error performance measure.

The system complexity is also examined for channel equalization under these three

detection methods. It turns out that the MAP algorithm is, as expected, the most complicated one because it needs more memory and more numerical operations. The newly developed trace back SOVA is the second in numerical operations and last in memory size required. For good error performance, the MAP algorithm is the best choice. When numerical complexity is the major concern, the ordinary register exchange SOVA should be used because it is roughly four times less complicated than the MAP algorithm and two times less than the trace back SOVA. If the amount of memory is very tight and the programmer does not want a mixed mode iterative receiver, the trace back mode is the one to be chosen.

Chapter 5

Conclusions and Suggestions for Further Study

5.1 Conclusions

In this thesis, the problem of detecting QPSK signals transmitted over wireless frequency non-selective fading channels has been investigated. Both maximum *a posteriori* probability symbol-by-symbol detection (MAP-SS) and maximum *a posteriori* sequence estimation (MAP-SE) are chosen as the criteria for optimality. Specifically two systems, namely uncoded and coded with interleaving, are considered. Receivers for each of them were designed and an iterative joint demodulation and decoding strategy was also studied.

For the uncoded system, channel demodulation was done by Bahl *et al*'s MAP algorithm (under the criterion of MAP-SS) and by the VA (under the criterion of MAP-SE). Simulation results in chapters 3 and 4 show that the error performances obtained are the same which implies that detection under the two criteria are equivalent. It is advisable to use VA in this situation because the VA is simpler than the MAP algorithm.

For an interleaved coded system, the complexity of the receiver is usually unreasonably high due to the interleaving structure and a suboptimal but practical two stage receiver is employed. The first stage is the channel demodulator and the second stage is the decoder. It is preferable for the first stage to pass soft information to the decoder because of higher error correction capabilities. In chapter 4 it is shown that when the demodulator uses hard-output VA, the error performance is significantly worse than with the soft-output MAP algorithm. The fading rate also affects the performance. As demonstrated in chapter 3, for a slowly changing channel with a fading rate of 0.01, the MAP receiver exhibits a system degradation of about 2 dB when compared with a receiver having ideal channel information. As much as 4 dB difference is observed when the fading rate increases to 0.05, which is considered to be relatively fast. Increasing the block length helps to reduces

these numbers because a larger block length usually implies a better interleaving scheme. All the error performances are measured at a BER of 10^{-4}.

Nevertheless, the MAP algorithm is highly complicated and it is easy to encounter a singularity problem. In view of this, it was replaced by the soft-output Viterbi algorithm (SOVA) in chapter 4. Hoeher's register exchange V-SOVA was implemented. Simulation results showed that, compared to the MAP receiver, the performance of the SOVA receiver degrades only 0.5 dB in slow fading but the difference becomes 2.5 dB in fast fading. Thus, SOVA can be a good substitute for the MAP algorithm in slow fading but not in fast fading.

The error performance can be improved by iterative processing since the first stage channel demodulation can benefit the coding structure. To achieve this, the decoder must have the SISO feature to deliver soft-outputs for the coded bits. The MAP algorithm has no problem in fulfilling this job. For the SOVA receiver, the trace back mode SOVA needs to be used. Consequently, the receiver becomes mixed mode. Simulation results showed that the difference between the SOVA receiver and the MAP receiver in slow fading is only 0.1 dB after one iteration. However, in fast fading, even if as many as five iterations are carried out on the SOVA receiver, the performance is still worse than the MAP receiver which iterates once. Even if iteration is carried out, the SOVA receiver is not recommended in the fast fading channel.

In view of the fact that the SOVA receiver has a mixed mode structure, an alternative approach to extract soft-outputs was given in chapter 4. This method combines the ideas of LVA and trace back SOVA and can deliver soft information for both information symbols and coded bits. Simulation results showed that the receiver with this new algorithm and the mixed mode SOVA receiver has practically the same error performance.

Finally, the complexity of the MAP algorithm, register exchange and the newly proposed SOVA was evaluated. By using the specifications of a particular DSP board, it was found that the simplified register exchange V-SOVA is the least complicated algorithm, followed by the newly presented method. The MAP algorithm needs the largest number of numerical operations. For memory usage, the new approach needs the least amount of memory, followed by V-SOVA. The MAP algorithm is again ranked last in this department. From this analysis, one can claim that the MAP algorithm is the most complicated

method. The V-SOVA is the fastest but needs marginally more memory size than the new technique. The design specifications: error performance, speed and available memory space, would decide which algorithm should be used.

5.2 Suggestions for Further Study

In this thesis, the presented SOVA operates only in fading channels. In order to complete the analysis of its usage, it is a good idea to look at its performance in an AWGN channel or possibly other channels as well. On the other hand, it may be possible to further improve this version of SOVA. By looking at equation (4.7), it is obvious that the denominator, which is essentially a normalization factor, takes only the probabilities of the paths along the best survivor into account. This may not be a very good normalization.

An alternative normalization method is presented in the following. Assume that the aggregated path metric for all paths within the trellis, Λ, is calculated as the VA proceeds. This is not difficult to do and needs only one extra memory space. As stated in chapter 4, there are $M^{\delta+1}$ different paths within a window of size $\delta+1$. Λ is then the total metric for all these paths. At the end of a window, the algorithm can proceed as stated in chapter 4 to choose the set of paths for the APP approximation. Using the notation in section 4.3.1, each chosen path has a metric of Γ_p where p is the path index and $0 \le p \le (M-1)(\delta+1)$. The total metric for all these [$(M$-1)(δ+1)+1] "selected paths" is

$$\Theta = \sum_{p=0}^{(M-1)(\delta+1)} \Gamma_p . \tag{5.1}$$

Since there are $M^{\delta+1}$ different paths within a window, the difference between Λ and Θ is therefore the total metric for the "non-selected paths" is

$$\Phi = \Lambda - \Theta . \tag{5.2}$$

Now assume that these "non-selected paths" are equally likely. Then each of them has a metric of

$$\Omega = \frac{\Phi}{M^{\delta+1} - [(M-1)(\delta+1)+1]} = \frac{\Phi}{H} \tag{5.3}$$

where $H = M^{\delta+1} - [(M-1)(\delta+1)+1]$ is the total number of "non-selected paths".

Since the set of the "selected paths" has only $[(M\text{-}1)(\delta+1)+1]$ members, in general this is a small number compared with $M^{\delta+1}$. Thus, it is reasonable to assume that, among those H "non-selected" paths, the number of occurrences for each M-ary symbol at time j-δ with reference to figure 4.15(a) is the same. Based on this assumption, within the set of "non-selected paths" a symbol q, $q \in \{0, 1, \ldots M-1\}$, will appear $\frac{H}{M}$ times. As each "non-selected path" has metric $\Omega = \frac{\Phi}{H}$, then based on all the above, somewhat speculative assertions, equation (4.7) can be rewritten as

$$Pr\{u_{j-\delta} = q|\vec{r}\} \approx \frac{\left(\sum_{p \in A} e^{\Gamma_p}\right) + \frac{H}{M} e^{\Omega}}{\left(\sum_{\forall p} e^{\Gamma_p}\right) + He^{\Omega}}. \tag{5.4}$$

In the above equation, the numerator includes the $\frac{H}{M}$ "non-selected paths" which have a specific symbol q at time j-δ. The denominator, which is the normalization factor, takes into account the metrics of the "non-selected paths". Notice that in this suggested change, one more exponential function and several divisions are required. Hence, this will lead to an even more complicated algorithm but it is worth studying to see if the performance improves significantly.

Finally, in [57] a method, incorporated with the VA was presented, to find the K best unmerged paths within a trellis. The advantage to finding unmerged paths is that it ensures those paths to have a different symbol at each time interval, which is an idea similar to V-SOVA. Soft-outputs can also be extracted from these unmerged path metrics. However, this method is too complicated for a trellis with many states and thus it is impractical. It is still worthwhile to set up a toy example of a small trellis and measure its performance. It maybe interesting to compare this method to the MAP algorithm, SOVA and possibly sub-optimum MAP algorithm [39] in terms of the error performance.

Appendix A

Bahl *et al*'s MAP Algorithm

The symbol-by-symbol MAP algorithm introduced by Bahl *et al* in [3] is an optimal detection method to minimize the probability of symbol/bit error. The algorithm provides a solution to the problem of calculating the *a posteriori* probabilities (APPs) of states and symbols which cause state transitions of a Markov process. A Markov source can be characterized by a trellis structure. Figure A.1 gives a simple example of a trellis diagram. Basically, if the APP of a state at a particular time is to be computed, all paths that pass through this given state have to be considered. The MAP algorithm provides an efficient solution to such a problem.

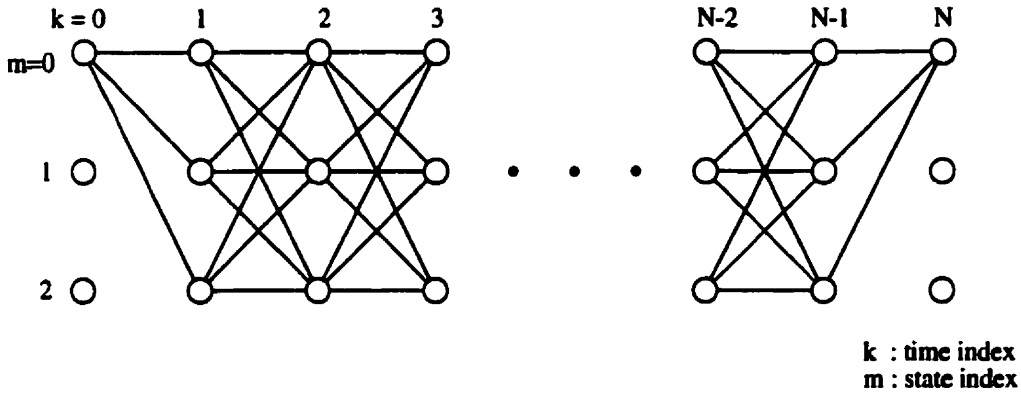


Figure A.1 Sample trellis diagram of a Markov process.

Suppose that vector $Y_1^N = \{y_1, y_2, ..., y_N\}$ is the sequence of sufficient statistics obtained at the output of a noisy channel and the vector $X_1^N = \{x_1, x_2, ..., x_N\}$ is the input sequence. Conventionally, x_j denotes the symbol which causes a state transition between states s_{j-1} and s_j. The aims of the algorithm are to find the APPs of the states and the state transitions by examining Y_1^N, i.e.

$$Pr\{s_k = m|Y_1^N\} \quad and \quad Pr\{s_{k-1} = m';s_k = m|Y_1^N\} . \tag{A.1}$$

Instead of looking at conditional probabilities, consider the joint probabilities

$$Pr\{s_k = m;Y_1^N\} \quad and \quad Pr\{s_{k-1} = m';s_k = m;Y_1^N\} . \tag{A.2}$$

Once the joint probabilities are found, calculation of the conditional probabilities are straightforward.

Consider the first term of (A.2). It can be written as

$$\begin{aligned} Pr\{s_k = m;Y_1^N\} &= Pr\{s_k = m;Y_1^k\}\, Pr\{Y_{k+1}^N \big| s_k = m;Y_1^k\} \\ &= Pr\{s_k = m;Y_1^k\}\, Pr\{Y_{k+1}^N \big| s_k = m\} \\ &= \alpha_k(m)\,\beta_k(m) \end{aligned} \tag{A.3}$$

where $\alpha_k(m) = Pr\{s_k = m;Y_1^k\}$ and $\beta_k(m) = Pr\{Y_{k+1}^N \big| s_k = m\}$. The middle equality follows from the Markov property. $\alpha_k(m)$ and $\beta_k(m)$ can be expressed as

$$\begin{aligned} \alpha_k(m) &= Pr\{s_k = m;Y_1^k\} \\ &= \sum_{m'=0}^{M-1} Pr\{s_{k-1} = m';s_k = m;Y_1^k\} \\ &= \sum_{m'=0}^{M-1} Pr\{s_{k-1} = m';Y_1^{k-1}\}\, Pr\{s_k = m;y_k \big| s_{k-1} = m';Y_1^{k-1}\} \\ &= \sum_{m'=0}^{M-1} Pr\{s_{k-1} = m';Y_1^{k-1}\}\, Pr\{s_k = m;y_k \big| s_{k-1} = m'\} \end{aligned}$$

and

$$\begin{aligned} \beta_k(m) &= Pr\{Y_{k+1}^N \big| s_k = m\} \\ &= \sum_{m'=0}^{M-1} Pr\{s_{k+1} = m';Y_{k+1}^N \big| s_k = m\} \end{aligned}$$

$$= \sum_{m'=0}^{M-1} Pr\{Y_{k+2}^{N} \big| s_{k+1} = m'; y_{k+1}; s_k = m\} \, Pr\{s_{k+1} = m'; y_{k+1} \big| s_k = m\}$$

$$= \sum_{m'=0}^{M-1} Pr\{Y_{k+2}^{N} \big| s_{k+1} = m'\} \, Pr\{s_{k+1} = m'; y_{k+1} \big| s_k = m\}$$

where M is the total number of states in the trellis at a specific time.

Define $\gamma_k(m', m) = Pr\{s_k = m; y_k \big| s_{k-1} = m'\}$. Then the above two equations can be rewritten as

$$\alpha_k(m) = \sum_{m'=0}^{M-1} \alpha_{k-1}(m') \, \gamma_k(m', m) \qquad \text{(A.4)}$$

and

$$\beta_k(m) = \sum_{m'=0}^{M-1} \beta_{k+1}(m') \, \gamma_{k+1}(m, m') \qquad \text{(A.5)}$$

which are in recursive form. In order to start the recursions, a set of initialized values must be predefined for $\alpha_0(m)$ and $\beta_N(m)$ with $0 \le m \le M-1$. Suppose the trellis starts and ends at state "0", as shown in figure A.1. The boundary conditions for the recursions are

$$\alpha_0(0) = 1 \quad and \quad \alpha_0(m) = 0 \quad \forall m \neq 0 \quad \text{and} \qquad \text{(A.6)}$$

$$\beta_N(0) = 1 \quad and \quad \beta_N(m) = 0 \quad \forall m \neq 0 \; . \qquad \text{(A.7)}$$

On the other hand, the value of $\gamma_k(m', m)$ depends on the channel model and it is assumed to be computable. The boundary conditions, plus the availability of γ, allow one to find all the α and β.

Now consider the second term of (A.2), which represents the joint state transition probability.

$$Pr\{s_{k-1} = m'; s_k = m; Y_1^N\} = Pr\{s_{k-1} = m'; Y_1^{k-1}\} \, Pr\{s_k = m; Y_k^N \big| s_{k-1} = m'; Y_1^{k-1}\}$$

$$= Pr\{s_{k-1} = m'; Y_1^{k-1}\} \, Pr\{s_k = m; Y_k^N \big| s_{k-1} = m'\}$$

$$= \alpha_{k-1}(m')\, Pr\{s_k = m; y_k | s_{k-1} = m'\}\, Pr\{Y_{k+1}^N | s_k = m; y_k; s_{k-1} = m'\}$$

$$= \alpha_{k-1}(m')\, Pr\{s_k = m; y_k | s_{k-1} = m'\}\, Pr\{Y_{k+1}^N | s_k = m\}$$

$$= \alpha_{k-1}(m')\, \gamma_k(m', m)\, \beta_k(m)$$

$$= \sigma_k(m', m)\ . \qquad \text{(A.8)}$$

This joint probability can be computed by the product of three terms which have been defined previously. The APPs of each input alphabet $x_1, x_2, \ldots, x_N$ can be found by considering the above joint probability. Assume that A is the set of transitions $s_{k-1} = m' \rightarrow s_k = m$ where the symbol corresponding to such a transition is q. Then

$$Pr\{x_k = q; Y_1^N\} = \sum_{(m', m) \in A} \sigma_k(m', m)\ , \qquad \text{(A.9)}$$

which can be normalized by $Pr\{Y_1^N\}$ to give $Pr\{x_k = q | Y_1^N\}$. Calculating these probabilities allows one to make a decision on symbol x_k. As a summary, the Bahl *et al*'s MAP algorithm uses two recursive formulae to find the probabilities of states and transition of states by observing the received sequence. The latter term in turn helps to calculate the APPs of each transmitted symbol. A decision based on the APPs will give the minimum probability of symbol error.

Appendix B

Viterbi Algorithm

The Viterbi algorithm (VA) was proposed initially as a detection method to minimize the probability of sequence error for convolutional codes [49]. In a more general form, the VA can be viewed as a solution to the problem of maximum *a posteriori* probability estimation of the state sequence for a finite state discrete time Markov process observed in memoryless noise [14]. By using the notation in chapter 4.1, the problem is to find the state sequence, $\vec{S}$, which maximizes $Pr\{\vec{S}|\vec{Z}\}$ or equivalently $Pr\{\vec{S}, \vec{Z}\}$. From equations (4.1) to (4.3), the problem statement is equivalent to finding the state sequence that maximizes

$$\log\left[Pr\{\vec{S}, \vec{Z}\}\right] = \sum_{k=1}^{N} \lambda_k(s_{k-1}, s_k) \qquad \text{(B.1)}$$

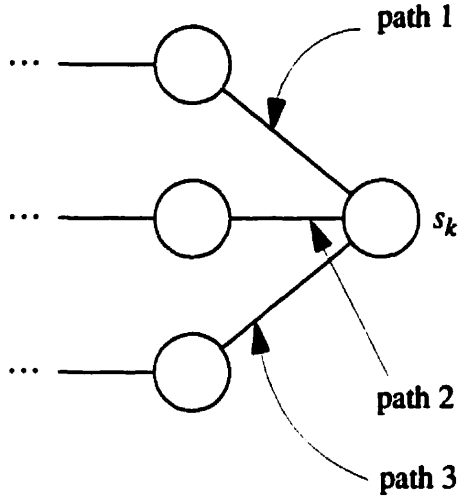
where $\lambda_k(s_{k-1}, s_k)$ corresponds to the state transition between s_{k-1} and s_k. It is called "branch metric" and is also analogous to $\gamma_k(m', m)$ in appendix A.

Denote S_0^k as a state sequence or path which resides in the trellis depicted in figure A.1. S_0^k can also be written as $\{s_0, s_1, \ldots, s_k\}$ which represents a path segment starting at s_0 and ending at s_k. Corresponding to this particular path segment, the accumulated branch metric can be expressed mathematically as

$$\Gamma(s_0, s_1, \ldots, s_k) = \Gamma\left(S_0^k\right) = \sum_{j=1}^{k} \lambda_j(s_{j-1}, s_j) \qquad \text{(B.2)}$$

where $\lambda_j(s_{j-1}, s_j)$ is the metric of the branch which joins the constituent states s_{j-1} and s_j of S_0^k. As shown in the trellis diagram, there are M possible paths (M=3 in this example)

entering a state s_k for $2 \le k \le N-1$. Observe that, if the complete state sequence which maximizes (B.1) passes through a specific s_k, then it must begin with the path segment having the largest accumulated metric among those M paths, as illustrated in figure B.1. Thus, the algorithm processes the accumulated metrics by comparing the metrics of all paths entering each state and storing the path with the largest metric (called the survivor) together with its metric [29]. Assume there are Q states in total at a given time within the trellis. Then, at any time index k, there will be Q survivor paths. Each terminates at a different state and has its own survivor metric.



Γ_i : Accumulated metric of path i entering state s_k.

Suppose Γ_1 is the largest among Γs, then the survivor path is path 1 and the algorithm stores this path and Γ_1.

Figure B.1 Deciding the survivor path from the M paths entering a state.

A summary of the algorithm is

Storage:

- k : time index
- S_0^k : survivor state sequence terminating in s_k.
- $\Gamma\left(S_0^k\right)$: accumulated survivor metric terminating in s_k.

Initialization:

With reference to the trellis of figure A1,

- $k = 0$;
- $S_0^0 = s_0$ = state "0";

- $\Gamma(s_0 = 0) = 0$ *and* $\Gamma(s_0 = m) = -\infty, \forall m \neq 0$.

Recursion:

For each state s_k at $k \geq 1$,

- calculate $\Gamma\left(S_0^{k-1}, s_k\right) = \Gamma\left(S_0^{k-1}\right) + \lambda(s_{k-1}, s_k)$ for all possible paths entering the given state;
- assign $\Gamma\left(S_0^k\right) = \underset{\forall paths}{max} \{\Gamma\left(S_0^{k-1}, s_k\right)\}$, store this value and the corresponding survivor sequence;
- increment k by 1 until $k = N$.

Instead of storing the entire survivor path vector S_0^k, an alternative is to store, at the present state, the previous state of the survivor path. There is virtually no difference between the two storage schemes.

For a very long trellis, it is necessary to truncate the survivors to a reasonable length δ which is referred to as the window size. One possible method is to have the algorithm making a decision on the state sequence at time k-δ by tracing back the survivor path with the globally best metric at time k. By choosing δ to be large enough, all Q survivors paths will merge when looking back and the effect of this windowing operation is negligible.

B.1 *M*-ary SOVA (Register Exchange Mode)

The VA described above delivers the most likely state sequence within a trellis. If the state transitions are due to some input symbols (which is usually the case), this path also represents the estimated input sequence of symbols after observing the channel output. Hagenauer and Hoeher considered binary sequences and modified the VA to generate not only the most likely path, but also a value reflecting the reliability of the decision [18]. This modified method, which is usually called the soft-output Viterbi algorithm (SOVA), can be implemented in register exchange mode or trace back mode. This section presents the register exchange mode SOVA for an *M*-ary trellis that is described in [24].

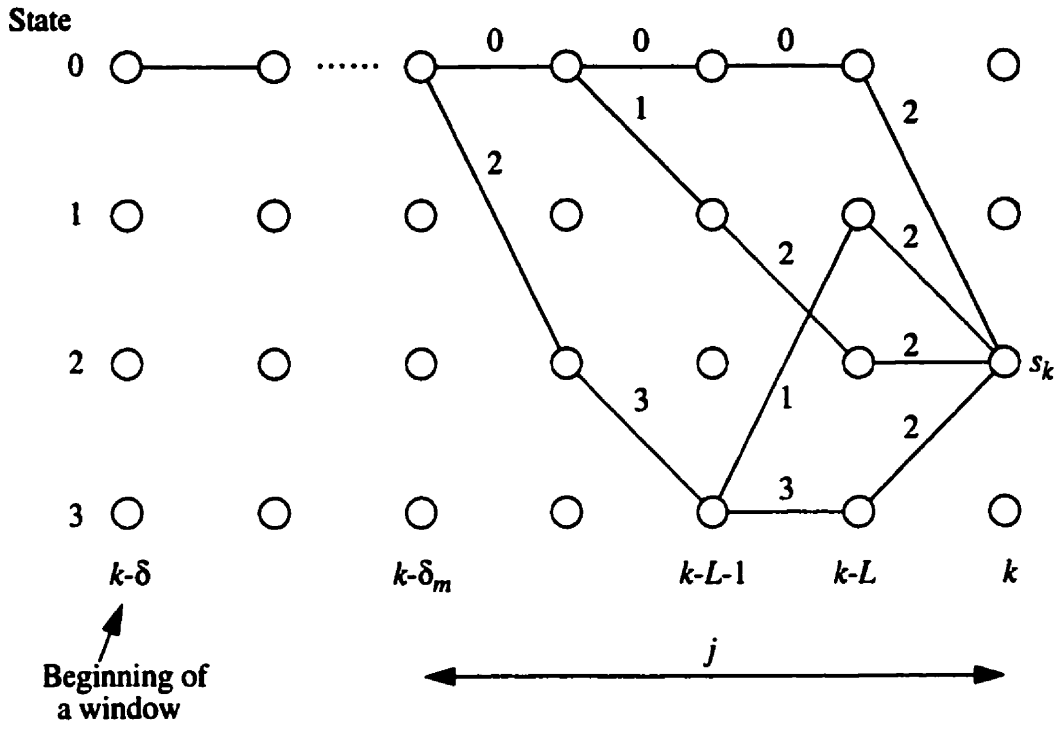


Figure B.2 A four-state trellis with memory $L = 1$ and $M = 4$ transitions per state.

An example of an M-ary trellis is shown in figure B.2. At time k, there are M paths entering state s_k and each path x, $1 \le x \le M$, is extended from a different previous state s_{k-1}. For each of these paths, the classical VA adds the branch metric $\lambda(s_{k-1}, s_k)$ to the previous accumulated metric as described in the last section. Denote the accumulated path metric entering this particular state, s_k, as $\Gamma^x(s_k)$ where the superscript is the index of the path. Each path has a probability given by:

$$Prob\{path\ x\} \propto e^{\Gamma^x(s_k)} \qquad x = 1, 2, \ldots, M. \tag{B.3}$$

Consider only these paths. The probability of j-th symbol A_j has probability

$$Prob\{A_j = q\} = \frac{\sum_{x \in C} Prob\{path\ x\}}{\sum_{\forall x} Prob\{path\ x\}} \tag{B.4}$$

where q is one of the possible M-ary alphabets transmitted at time j and C is the set of paths which contains the symbol q at that time. Define "relevant range" as the range

where, when tracing along the paths entering s_k, the symbols on these paths are different within this range. The relevant range for s_k as shown in figure B.2 is the symbol period between time k-δ_m to k. The symbols A_j, $(k-\delta_m+1) \le j \le k$, are termed "relevant symbols".

Of the M paths coming into state s_k, the classic VA chooses a survivor path. Assume the survivor which enters s_k is extended from a specific previous state s_{k-1}. As equation (B.4) has also been evaluated for this previous state, the previously calculated probability $Prob\{A_j\}$ for symbols A_j stored in s_{k-1} has a time index j between $(k-\delta_m+1)$ and $(k-1)$. The SOVA proposes a recursive update for the relevant symbols by

$$\left[Prob\{A_j\}\ at\ s_k\right] \leftarrow \left[Prob\{A_j\}\ at\ s_{k-1}\right] \cdot Prob\{A_j\} \qquad \text{(B.5)}$$

where the last term is the probability calculated by (B.4) at state s_k.

This updating procedure has to be performed within the relevant range of s_k for every relevant symbol. Normalization is required to get the sum of updated probabilities back to unity. Outside the range, the algorithm proceeds with

$$\left[Prob\{A_j\}\ at\ s_k\right] \leftarrow \left[Prob\{A_j\}\ at\ s_{k-1}\right], \quad (k-\delta_m) \le j\ . \qquad \text{(B.6)}$$

Thus, for every state, a memory space has to be reserved not only for the survivor state sequence and metric, but also for the probabilities of all the previously transmitted symbols. The summary of the entire algorithm is given as follows:

(a) Storage at each state:

- $Prob\{A_j\}$, $(k-\delta+1) \le j$

(b) For each state s_k,

- determine survivor and s_{k-1};
- initialize $Prob\{A_k\}$ at this particular state to be equally likely.

For all paths $x = \{1, 2, \ldots, M\}$,

for the relevant range (indexed j),

for each relevant symbol $A_j = \{0, 1, \ldots, M\text{-}1\}$,

- compute and update the stored probabilities according to (B.5).

for the "irrelevant range",

- copy the stored probabilities for the previous survivor.

(c) At the end of window,

- find the survivor with maximum metric.
- find the hard-output of $A_{k\text{-}\delta}$ from this survivor;
- find the soft-output of $Prob\{A_{k-\delta}\}$ which corresponds to this survivor.

The algorithm described above is too complex to implement in full. Two reduced complexity modifications are given below.

(a) By ranking the survivor metrics going into each state in descending order, the update for the relevant range at each time is done only for the best P ($P \leq Q$) states instead of all states. One special case is $P = 1$. Under this situation, this less complex method is referred to as H-SOVA ("one dimensional horizontal update").

(b) The second approach is to update only inside a fixed maximum update window of length δ_w, $1 \leq \delta_w \leq \delta_m$. Of special interest is $\delta_w = 1$. For this special case, the probability calculation is carried out on the symbol interval $[(k\text{-}L\text{-}1),k\text{-}L]$ and no further update is needed. As each state in the trellis is of the shift register type, the symbols on the M paths entering state s_k always differ at this particular time slot. Consequently, $Prob\{A_j = q\}$ for all possible q in (B.4) will be non-zero. This results in preserving the soft information of symbols, no matter how small. Furthermore, the number of numerical calculations is greatly reduced because no update is needed. Instead of storing probabilities, the algorithm can store the M paths metrics at each time because they are also a measure of probabilities. Probability conversion will be made only for the globally best survivor at the end of a window, thus reducing the computation needed to evaluate the exponential function. This version is known as V-SOVA ("one dimensional vertical

update").

B.2 Binary SOVA (Trace Back Mode)

Unlike the method described above, the trace back mode was designed to work on a binary trellis only. The reason is because this method operates on the log-likelihood ratio (LLR) that is to be defined and requires special algebraic rules which work on elements in a binary field.

Consider a binary element u which may take a value of 0 or 1 and is also a member of GF(2) that obeys modulo-2 algebra. Define the log-likelihood ratio $L(u)$ as in (3.22), viz

$$L(u) = \log\frac{Pr\{u=1\}}{Pr\{u=0\}}. \tag{B.7}$$

Suppose vector Y_1^N is some observation from the output of a noisy channel, the conditional LLR is given by

$$L\left(u|Y_1^N\right) = \log\frac{Pr\{u=1|Y_1^N\}}{Pr\{u=0|Y_1^N\}}. \tag{B.8}$$

It is worth to mention that the sign of $L\left(u|Y_1^N\right)$ gives the hard decision and its magnitude reflects the reliability of this decision. In addition, definition for some special algebraic operations on LLR are needed. Define $\boxplus$ as the addition operation with

$$L(u_1) \boxplus L(u_2) \equiv L(u_1 \oplus u_2) \tag{B.9}$$

and

$$\begin{aligned} L(u) \boxplus \infty &= L(u) \\ L(u) \boxplus -\infty &= -L(u) \\ L(u) \boxplus 0 &= 0 \end{aligned} \tag{B.10}$$

where $\oplus$ is modulo-2 addition operation. As given in [20], the LLR of the sum of K independent u_j is

$$L\{\overset{K}{\underset{j=1}{\oplus}} u_j\} = \overset{K}{\underset{j=1}{\boxplus}} L(u_j) \approx \prod_{j=1}^{K} sign\{L(u_j)\} \cdot \min_{j=1,\ldots,K} |L(u_j)| \cdot \qquad \text{(B.11)}$$

Now one is in a position to discuss the algorithm. It proceeds in the usual way as the classical VA. Other than the survivor sequence and metric, it also stores the difference in metrics between the survivor and the discarded path entering a state. Figure B.3 shows a binary trellis at the end of a decision window. The globally best path is drawn with a dashed line and from this path the VA obtains the bit hard decision $u_{j-\delta}$. Also along this path a total of $\delta+1$ non-survivor paths with indexes $t = 0, 1, \ldots, \delta$ have been discarded. The difference between the metrics is labelled Δ^t. If the bit of t-th non-survivor path $u_{j-\delta}^t$ at time j-δ equals $u_{j-\delta}$, no bit error will be made if the decision is made from this path and the reliability of the decision is ∞. However, there is a possibility of being in error if they are different. Under this circumstance the reliability value is set to Δ^t. Hence

$$L(e^t) = \begin{cases} \infty & \text{if } u_{j-\delta}^t = u_{j-\delta} \\ \Delta^t & \text{otherwise} \end{cases} \qquad \text{for } t = 0, \ldots, \delta \qquad \text{(B.12)}$$

where e^t is the reliability value for path t. Assume that the random variable e^t is statistically independent with respect to index t. In [17], it is claimed that the value of reliability or the soft-output is

$$\begin{aligned} L\{u_{j-\delta}\} &= -(2 \cdot u_{j-\delta} - 1) \overset{\delta}{\underset{t=1}{\boxplus}} L(e^t) \\ &\approx -(2 \cdot u_{j-\delta} - 1) \cdot \min_{t=0,\ldots,\delta} |L(u_j)| \cdot \end{aligned} \qquad \text{(B.13)}$$

Thus, in addition to the hard output given by the classical VA, the reliability of the decisions is also obtained. This piece of soft information can be treated as the estimated APP of bit $u_{j-\delta}$ when converting the LLR back into a probability. This completes the discussion of SOVA.

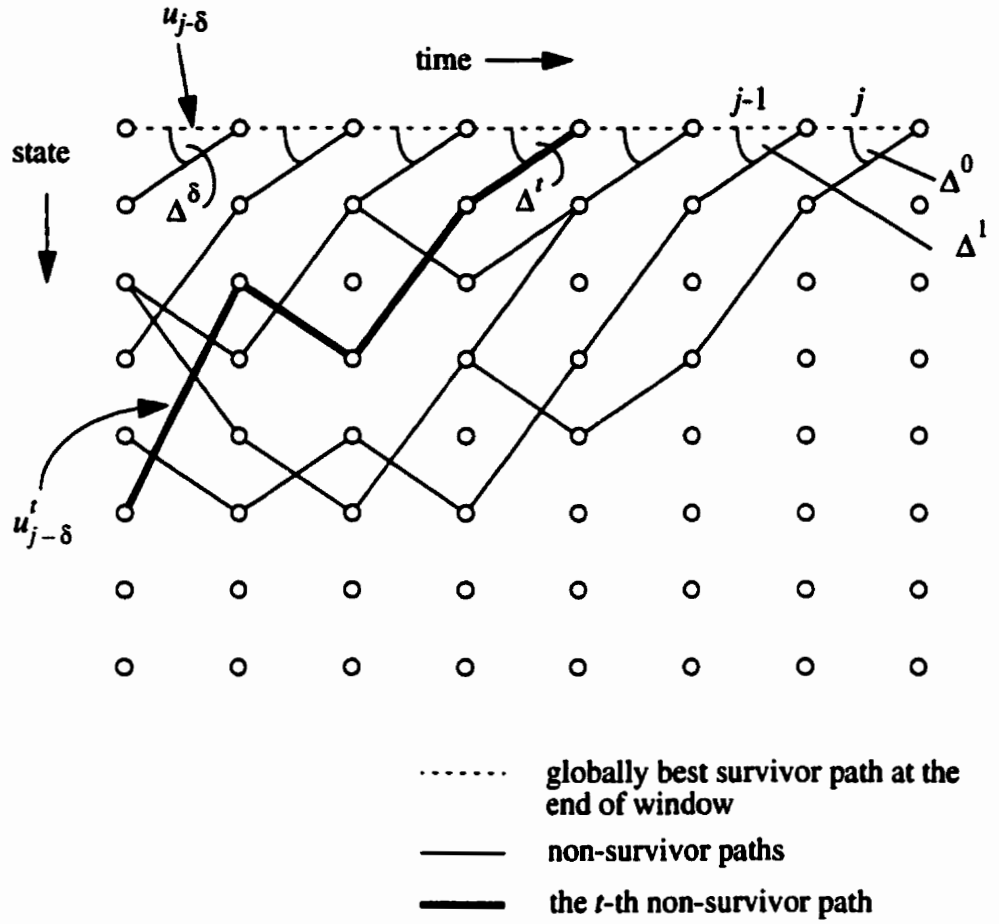


Figure B.3 Example trellis window of length (δ+1) for derivation of trace back SOVA.

Appendix C

Precaution for the MAP Algorithm

Direct implementation of the MAP algorithm has a risk of a singularity problem, i.e. running into the situation of 0/0. The risk is particularly high when the data sequence has a large block length, N. This appendix provides a feasible solution to reduce the possibility of encountering this difficulty.

The material discussed here focuses on the forward recursion equation but the same analogy applies to the backward recursion. Rewrite (3.15) as

$$\alpha_{k+1}(m) = \sum_{m'} \alpha_k(m')\,\gamma_{k+1}(m', m) \qquad \text{(C.1)}$$

where k is the time index running from 0 to N-1, m and m' represent the state index. Let M be the number of possible transmitted symbols and L be the number of symbols which define a state. Then $0 \le m, m' \le Q$ where $Q = M^L - 1$.

Suppose all γ_k are very small but they still can be described with some precision. This is actually the case encountered in this thesis. Then there exists a sufficiently large k such that α_k become very small and even using "double precision" in the computer simulation cannot describe the values precisely. Consequently these values are forced to zero and underflow error occurs. A blind substitution of equation (3.18) for APP calculation results in 0/0 and numerical error.

A straightforward solution to this problem is to enlarge the values of γ_k by multiplying them with a fixed constant. Such a solution is not recommended because the γ_k depend heavily on the sufficient statistics obtained from a random channel. When a large number of data blocks are transmitted it is hard to assign an adaptive multiplicative constant as mentioned.

For the proposed solution, assume the following:

Assumption: For $1 \le k \le N$, there exists at least one $\gamma_k(m', m)$ whose value is larger than zero and can be described with some degree of precision.

Such an assumption is important because, if for some k, all γ_k go to zero, so will the α_k and the singularity problem persists.

First, explore the α_k step-by-step starting with $k = 0$.

$k = 0$: $\alpha_1(m) = \sum_{m'} \alpha_0(m')\gamma_1(m', m)$

Assume the following numerical values are obtained:

$$\alpha_1(m) = \{\left(a_1^0 \cdot 10^{b_1^0}\right), \left(a_1^1 \cdot 10^{b_1^1}\right), \ldots, \left(a_1^Q \cdot 10^{b_1^Q}\right)\}$$

For the exponent and mantissa, the subscript refers to the time index and the superscript denotes the state index.

Define set S_1 to be

$$S_1 = \{b_1^0, b_1^1, \ldots, b_1^Q\}$$

and let $x_1 = \min\{S_1\}$. Compute

$$\overline{\alpha_1}(m) = \alpha_1(m) \cdot 10^{-x_1} \tag{C.2}$$

for all m. This completes the calculation at this time index.

$k = 1$: Instead of using the original formula, create a new relationship similar to α_k based on the function $\overline{\alpha_k}$

$$\lambda_{k+1}(m) = \sum_{m'} \overline{\alpha_k}(m')\gamma_{k+1}(m', m)\,. \tag{C.3}$$

After evaluating this equation, suppose the following numerical values are obtained:

$$\lambda_2(m) = \{\left(a_2^0 \cdot 10^{b_2^0}\right), \left(a_2^1 \cdot 10^{b_2^1}\right), \ldots, \left(a_2^Q \cdot 10^{b_2^Q}\right)\}\,.$$

Just like the case when k=0, let $S_2 = \{b_2^0, b_2^1, \ldots, b_2^Q\}$ and $x_2 = \min\{S_2\}$.

Calculate

$$\overline{\alpha_2}(m) = \lambda_2(m) \cdot 10^{-x_2}. \tag{C.4}$$

With all the above definitions, $\alpha_2(m)$ can be found to be

$$\begin{aligned}\alpha_2(m) &= \sum_{m'} \alpha_1(m')\,\gamma_2(m', m) \\ &= 10^{x_1} \sum_{m'} \overline{\alpha_1}(m')\,\gamma_2(m', m) \\ &= 10^{x_1} \lambda_2(m) \\ &= 10^{x_1 + x_2} \left[\overline{\alpha_2}(m)\right]. \end{aligned} \tag{C.6}$$

If the same procedure is repeated, the following relationship can be obtained

$$\alpha_k(m) = 10^{\sum_{j=1}^{k} x_j}\, \overline{\alpha_k}(m) \quad for\ k > 0\ . \tag{C.7}$$

A similar relationship can be found for the backward recursion equation if the same steps are followed. Then,

$$\beta_k(m) = 10^{\sum_{j=1}^{k} y_j}\, \overline{\beta_k}(m) \quad for\ k < N \tag{C.8}$$

where y_j has the same role as x_j. In order to compute the APP, equation (3.18) has to be considered

$$Pr\{\hat{a}_k = q|\vec{R}\} = \frac{\sum_{(m',m) \in D} \sigma_k(m',m)}{\sum_{\forall (m',m)} \sigma_k(m',m)} \tag{C.9}$$

where D is the set of state transition at time k corresponding to the alphabet q. Substituting equations (3.17), (C.7) and (C.8) gives

$$Pr\{\hat{a}_k = q|\vec{R}\} = \frac{\sum_{(m',m)\in D} 10^{\sum_{j=1}^{k-1} x_j} \overline{\alpha_{k-1}}(m')\gamma_k(m',m)\overline{\beta_k}(m) 10^{\sum_{j=1}^{k} y_j}}{\sum_{\forall(m',m)} 10^{\sum_{j=1}^{k-1} x_j} \overline{\alpha_{k-1}}(m')\gamma_k(m',m)\overline{\beta_k}(m) 10^{\sum_{j=1}^{k} y_j}}$$

$$= \frac{\sum_{(m',m)\in D} \overline{\alpha_{k-1}}(m')\gamma_k(m',m)\overline{\beta_k}(m)}{\sum_{\forall(m',m)} \overline{\alpha_{k-1}}(m')\gamma_k(m',m)\overline{\beta_k}(m)}. \quad \text{(C.10)}$$

Hence, storing $\bar{\alpha}$ and $\bar{\beta}$ can lead to the same solution. The advantage of storing in this format is that exponential powers of the values have been reduced at each storage, which reduces the threat of an underflow error and the risk of a singularity problem in the APP calculation. The side effect is an increase in numerical complexity as routines involving an exponential power finder and maximum value selector have to be added. The steps for the solution for the forward recursion can be summarized as:

Initialization:

- $k = 0$;
- $\overline{\alpha_0}(m)$, $0 \le m \le Q$ the same as α_0.

For each k, $1 \le k \le N$,

- use (C.3) to calculate $\lambda_k(m)$;
- find x_k which is the smallest exponent among all $\lambda_k(m)$;
- scale $\lambda_k(m)$ into $\overline{\alpha_k}(m)$ as in (C.4) and store the results.

Use $\overline{\alpha_k}$ in APP calculation.

References

[1] K. Abend and B. D. Fritchman, "Statistical Detection for Communication Channels with Intersymbol Interference", *Proc. of IEEE*, Vol. 58, No. 5, pp 779-785, May 1970.

[2] A. N. D'Andrea, A. Diglio and U. Mengali, "Symbol-Aided Channel Estimation with Nonselective Rayleigh Fading Channels", *IEEE Trans. on Veh. Tech.*, Vol. 44, No. 1, pp 41-48, Feb. 1995.

[3] L. R. Bahl, J. Cocke, F. Jelinek and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate", *IEEE Trans. on Inform. Theory*, Vol. 20, No. 2, pp 284-287, March 1974.

[4] S. Benedetto and G. Montorsi, "Iterative Decoding of Serially Concatenated Convolutional Codes", *Electronics Letters*, Vol. 32 No. 13, pp 117-120, June 1996.

[5] S. Benedetto, G. Montorsi, "A Soft-Input Soft-Output APP Module for Iterative Decoding of Concatenated Codes", *IEEE Commun. Letters*, Vol. 1, No. 1, pp 22-24, Jan. 1997.

[6] S. Benedetto, G. Montorsi, "Serial Concatenation of Interleaved Codes: Performance Analysis, Design, and Iterative Decoding", *IEEE Trans. on Inform. Theory*, Vol. 44, No. 3, pp 909-926, May 1998.

[7] C. Berrou, A. Glavieux and T. Thitimajshima, "Near Shannon Limit Error Correcting Coding and Decoding: Turbo-Codes (1)", *Proc. IEEE Int'l Conf. on Commun.*, Geneva, Switzerland, May 1993, pp 1064-1070.

[8] J. K. Cavers, "An Analysis of Pilot Symbol Assisted Modulation for Rayleigh Fading Channels", *IEEE Trans. on Veh. Tech.*, Vol. 40, No. 4, pp 686-693, Nov. 1991.

[9] J. K. Cavers, "On the Validity of the Slow and Moderate Fading Models for Matched Filter Detection of Rayleigh Fading Signals", *Canadian J. of Elec. and Comp. Eng.*, pp 183-189, Oct. 1992.

[10] K. Y. Chan and A. Bateman, "*M*-ary PSK with Trellis Coding Modulation", *IEEE Trans. on Veh. Tech.*, Vol. 41, No. 2, pp 190-198, May 1992.

[11] K. M. Chugg, "The Condition for the Applicability of the Viterbi Algorithm with Implications for Fading Channel MLSD", *IEEE Trans. on Commun.*, Vol. 46, No. 9, pp 1112-1116, Sep. 1998.

[12] Q. Dai, *Sequence Estimation for Rayleigh Fading Channels with Intersymbol Interference*, Ph. D thesis, University of Manitoba, Winnipeg, Canada, 1993.

[13] G. D. Forney Jr. "Maximum Likelihood Sequence Estimation of Digital Sequence in the Presence of Intersymbol Interference", *IEEE Trans. on Inform. Theory*, Vol. 18, No 3, pp 363-378, May 1972.

[14] G. D. Forney Jr. "The Viterbi Algorithm", *Proc. of IEEE*, Vol. 61, No. 3, pp 268-278, March 1973.

[15] M. J. Gertsman, *Symbol-by-Symbol MAP Demodulation of CPM signals on Rayleigh Flat-Fading Channels*, M. Eng. thesis, Carleton University, Ottawa, Canada, 1996.

[16] M. J. Gertsman and J. H. Lodge, "Symbol-by-Symbol MAP Demodulation of CPM and PSK signals on Rayleigh Flat-Fading Channels", *IEEE Trans. on Commun.*, Vol. 45, No. 7, pp 788-799, July 1997.

[17] J. Hagenauer, "Source-Controlled Channel Decoding", *IEEE Trans. on Commun.*, Vol. 43, No. 9, pp 2449-2456, Sep. 1995.

[18] J. Hagenauer and P. Hoeher, "A Viterbi Algorithm with Soft-Decision Outputs and its Applications", *Proc. IEEE Globecom Conf.*, Dallas, Texas, USA, Nov. 1989, pp 47.1.1-47.1.7.

[19] J. Hagenauer and E. Lutz, "Forward Error Correction Coding for fading Compensation in Mobile Satellite Channels", *IEEE J. Select. Areas on Commun.*, Vol. 5, No. 2, pp 215-225, Feb. 1987.

[20] J. Hagenauer, E. Offer and L. Papke, "Iterative Decoding of Binary Block and Convolutional Codes", *IEEE Trans. on Inform. Theory*, Vol. 42, No 2, pp 429-445, March 1996.

[21] J. Hagenauer, L. Papke and P. Robertson, "Iterative ("Turbo") Decoding of Systematic Convolutional Codes with the MAP and SOVA Algorithms", *Proc. ITG-Fachtagung 'Codierung'*, Munich, Germany, Oct. 1994, pp 21-29.

[22] C. R. P. Hartmann and L. D. Rudolph, "An Optimum Symbol-by-Symbol Decoding Rule for Linear Codes", *IEEE Trans. on Inform. Theory*, Vol. 22, No 5, pp 514-517,

Sep. 1976.

[23] T. Hashimoto, "A List-Type Reduced-Constraint Generalization of the Viterbi Algorithm", *IEEE Trans. on Inform. Theory*, Vol. 33, No 6, pp 866-876, Nov. 1987.

[24] P. Hoeher, "TCM on Frequency-Selective Fading Channels: a Comparison of Soft-output Probabilistic Equalizers", *Proc. of IEEE Globecom.* San Diego, USA, Dec. 1990, pp 401.4.1-401.4.6.

[25] P. Hoeher, "A Statistical Discrete-Time Model for the WSSUS Multipath Channel", *IEEE Trans. on Veh. Tech.*, Vol. 41, No. 4, pp 461-468, Nov. 1992.

[26] G. T. Irvine and P. J. McLane, "Symbol-Aided Plus Decision Directed Reception for PSK/TCM Modulation on Shadowed Mobile Satellite Fading Channels", *IEEE J. Select. Areas on Commun.*, Vol. 10, No. 8, pp 1289-1299, Oct. 1992.

[27] W. C. Jakes, *Microwave Mobile Communications*, IEEE Press, 1974.

[28] H. Kong, *Receiver Structures for Wireless Mobile Channels*, Ph. D thesis, University of Manitoba, Winnipeg, Canada, 1997.

[29] Shu Lin, D. J. Costello Jr., *Error Control Coding Fundamentals and Applications*, Prentice Hall Inc., 1983.

[30] J. H. Lodge and M. L. Moher, "Maximum Likelihood Sequence Estimation of CPM Signals Transmitted over Rayleigh Flat-Fading Channels", *IEEE Trans. on Commun.*, Vol. 38, No. 6, pp 787-794, June 1990.

[31] J. Lodge, R. Young and P. Guinand, "Separable Concatenated Codes with Iterate MAP Decoding for Rician Fading Channels", *Proc. Int. Mob. Sat. Conf.* June 1993.

[32] J. Lodge, R. Young, P. Hoeher and J. Hagenauer, "Separable MAP 'Filters' for the Decoding of Product and Concatenated Codes", *Proc. IEEE Int'l Conf. on Commun.*, Geneva, Switzerland, May 1993, pp 1740-1745.

[33] D. Makrakis, P. T. Mathiopoulos and D. P. Bouras, "Optimal Decoding of Coded PSK and QAM Signals in Correlated Fast Fading Channels and AWGN: A Combined Envelope, Multiple Differential and Coherent Detection Approach", *IEEE Trans. on Commun.*, Vol. 42, No. 1, pp 63-75, Jan. 1994.

[34] M. L. Moher and J. H. Lodge, "TCMP-A Modulation and Coding Strategy for Rician Fading Channels", *IEEE J. Select. Areas on Commun.*, Vol. 7, No. 9, pp 1347-1355, Dec. 1989.

[35] C. Nill and C. W. Sundberg, "List and Soft Symbol Output Viterbi Algorithms: Extensions and Comparisons", *IEEE Trans. on Commun.*, Vol. 43, No. 2/3/4, pp 277-287, Feb./Mar./April 1995.

[36] A. Picart, P. Didier and A. Glavieux, "Turbo-Detection: A New Approach to Combat Channel Frequency Selectivity", *Proc. IEEE Int'l Conf. on Commun.*, Montreal, Quebec, Canada, June 1997, pp 1498-1502.

[37] J. G. Proakis, *Digital Communications*, second edition, McGraw-Hill International Editions, 1989.

[38] R. Raheli, A. Polydoros and C. Tzou, "Per-Survivor Processing: a General Approach to MLSE in Uncertain Environments", *IEEE Trans. on Commun.*, Vol. 43, No. 2/3/4, pp 354-364, Feb./Mar./April 1995.

[39] P. Robertson, E. Villebrun and P. Hoeher, "A Comparison of Optimal and Sub-Optimal MAP Decoding Algorithms Operating in the Log Domain", *Proc. IEEE Int'l Conf. on Commun.*, Seattle, Washington, June 1995, pp 1009-1013.

[40] S. Sampei and T. Sunaga, "Rayleigh Fading Compensation Method for 16QAM in Digital Land Mobile Radio Channels", *Proc. IEEE Vech. Tech. Conf.*, San Francisco, USA, May 1989, pp 640-646.

[41] N. Seshadri and P. Hoeher, "On Post-Decision Symbol-Reliability Generation", *Proc. IEEE Int'l Conf. on Commun.*, Geneva, Switzerland, May 1993, pp 741-745.

[42] N. Seshadri and C. W. Sundberg, "List Viterbi Decoding Algorithms with Applications", *IEEE Trans. on Commun.*, Vol. 42, No. 2/3/4, pp 313-323, Feb./Mar./April 1994.

[43] J. P. Seymour and M. P. Fitz, "Near-Optimal Symbol-by-Symbol Detection Schemes for Flat Rayleigh Fading", *IEEE Trans. on Commun*, Vol. 43, No. 2/3/4, pp 1525-1533, Feb./Mar./April 1995.

[44] W. H. Sheen and G. L. Stuber, "MLSE Equalization and Decoding for Multipath-Fading Channels", *IEEE Trans. on Commun.*, Vol. 39, No. 10, pp 1455-1464, Oct. 1991.

[45] S. Stein, "Fading Channel Issues in System Engineering", *IEEE J. Select. Areas on Commun.*, Vol. 5, No. 2, pp 68-89, Feb. 1987.

[46] G. L. Stuber, *Principles of Mobile Communication*, Kluwer Academic Publishers,

1996.

[47] Texas Instrument, *TMS320C6000 CPU and Instruction Set Reference Guide*, March 1999.

[48] Texas Instrument, *TMS320C6000 Assembly Benckmarks*, 1999.

[49] A. J. Viterbi, "Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm", *IEEE Trans. on Inform. Theory*, Vol. 13, No 6, pp 260-269, April, 1967.

[50] A. J. Viterbi, "The Evolution of Digital Wireless Technology from Space Exploration to Personal Communication Services", *IEEE Trans. on Veh. Tech.*, Vol. 43, No. 3, pp. 638-643, August 1994.

[51] G. M. Vitetta and D. P. Taylor, "Maximum Likelihood Sequence Estimation of Uncoded and Coded PSK Signals Transmitted over Rayleigh Flat-Fading Channels", *Proc. IEEE Int'l Conf. on Commun.*, New Orleans, Louisiana, USA, May 1994, pp 1-5.

[52] G. M. Vitetta and D. P. Taylor, "Maximum Likelihood Decoding of Uncoded and Coded PSK Signal Sequences Transmitted over Rayleigh Flat-Fading Channels", *IEEE Trans. on Commun.*, Vol. 43, No. 11, pp 2750-2758, Nov. 1995.

[53] B. Vucetic and J. Du, "Channel Modelling and Simulation in Satellite Mobile Communication Systems", *IEEE J. Select. Areas on Commun.*, Vol. 10, No. 8, pp 1209-1218, Oct. 1992.

[54] Xiao-an Wang and Stephen B. Wicker, "A Soft-Output Decoding Algorithm for Concatenated Systems", *IEEE Trans. on Inform. Theory*, Vol. 42, No 2, pp 543-553, March 1996.

[55] K. Wesolowski, "Computer Generation of a Slowly Varying Pseudorandom Process", *IEE Proceedings-F*, Vol. 130, No. 4, pp 314-316, June 1983.

[56] J. K. Wolf, "Efficient Maximum Likelihood Decoding of Linear Block Codes Using a Trellis", *IEEE Trans. on Inform. Theory*, Vol. 24, No 1, pp 76-80, Jan. 1978.

[57] J. K. Wolf, A. M. Viterbi, G. S. Dixon, "Finding the K Best Paths Through a Trellis with Application to Multi-Target Tracking".

[58] X. Yu and S. Pasupathy, "Innovations-Based MLSE for Rayleigh Fading Channels", *IEEE Trans. on Commun.*, Vol. 43, No. 2/3/4, pp. 1534-1544, Feb./Mar./April 1995.