Performance Evaluation of the MLSDA Decoders by Importance Sampling Techniques

Prepared by Jau-Yuan Weng

Advisory by Prof. Po-Ning Chen

In Partial Fulfillment of the Requirements

For the Degree of
Master of Science

Department of Communications Engineering

National Chiao Tung University

Hsinchu, Taiwan 300, R.O.C.

E-mail: u8813525@cc.nctu.edu.tw

June 14, 2001

Abstract

It is known that the error correction capability of the convolutional codes grows dramatically as the code constraint length increases. Yet, to employ codes with long constraint length may suffer a high decoding complexity. In [1], the authors proposed the *Maximum-Likelihood soft-decision Sequential Decoding Algorithm* (MLSDA) [1] for binary convolutional codes, and showed that its computational complexity turns out to be less affected by the code constraint length; therefore, it may apply to convolutional codes with long constraint lengths, and yield a good system performance. In order to evaluate the resultant system performance, sufficient simulation runs, usually requiring to induce hundred of errors, should be taken. This may render an unfeasibly long simulation time if the true system error is indeed low.

Unlike the brute-force *Monte Carlo* (MC) simulation that often requires a very large number of simulation trials to achieve meaningful estimates of system performance, the *Importance Sampling* (IS) [2, 3, 4, 7] simulation can achieve relatively accurate estimate with much less simulation trials. For IS technique, well-chosen channels to adapt to suitable error events can result in an efficient simulator.

In this thesis, we focused on the selection of good biased channels for Importance Sampling being applied to the MLSDA. By the simulation results, the best IS simulator among those we tried can save fairly large simulation trials, while an improperly chosen IS density may arise an IS simulator that performs even worse than an MC one.

Acknowledgements

I would like to thank Dr. Po-Ning Chen and Dr. Yunghsiang S. Han for their encouragement and guidance. This work would not have been possible without their advice and commitments.

I would like to thank my dear laboratory mates Jia-Qin, Yi-Ren and Wei-Sheng. They provide much fun on countless boring days. I would also like to thank the members of Network Technology Laboratory and my friends.

Finally, I give the greatest respect to my family for their continued support during my study.

Contents

List of Figures

Chapter 1

Introduction

It is known that the error correction capability of the convolutional codes grows dramatically as the code constraint length increases. For a convolutional code, the Viterbi decoding algorithm can provide the maximum-likelihood performance but suffer prohibitively big computational complexity, if the code constraint length is large. In contrast, the sequential algorithm decodes the convolutional codes with computational complexity independent of the code constraint length; however, it is only *asymptotically* maximum-likelihood in its performance. By substituting the Fano metric with a newly proposed metric, the conventional sequential decoding algorithm can be transformed to a maximum-likelihood decoder, as named the *maximum-likelihood soft-decision sequential decoding algorithm* (MLSDA) [1]. By retaining the characteristic of the sequential decoding, a great saving on the computational complexity is obtained by the MLSDA, if it is compared to the Viterbi algorithm [1, 18].

Although improvement on computational complexity is achieved by the MLSDA, its performance evaluation is still a tough course. When traditional Monte Carlo (MC) method is utilized to estimate the bit error rate (BER) of the MLSDA decoder, more than $100/P_e$ experiments are needed to obtain a relatively accurate estimate of P_e [3]. For example, if the error probability is about 10^{-8}, the simulation requires tens of billion of simulation runs, which is very time consuming. Some improvements on the simulation technique itself may be necessary.

In 1980, a modified Monte Carlo simulation technique was first used to estimate the error probabilities of the digital communication systems [11], and then various modifications were subsequently developed [5, 6, 9, 10, 12, 14]. The exhibited good efficiency of these estimators originated the motivation of applying such techniques, which are named the Importance Sampling (IS), to the simulations of coded systems [2, 3, 4, 7]. The basic approach of the importance sampling to achieve the reduction of the computational efforts is to transmit signals through channels being biased by variance-scaled (VS) or mean-translated (MT) noise. Equipped with the approach, one can force the important error events to occur more frequently; hence, one can obtain the desired hundreds of errors within less number of simulation runs. The computation of the error probability is then adjusted by a multiplicative weighting function, depending on the biased channel statistics. One can then justify the accuracy of the resultant error probability with the *relative precision estimate*.

As anticipated from the previous description, the sampling biased channel density is the most essential factor on the efficacy of the IS technique. A proper choice of the IS density will result in a large time saving. However, if the IS density is not well chosen, not only the computational time may elongate but the estimate accuracy may lost.

The thesis is organized in the following fashion. A brief introduction of the MLSDA is contained in Chapter 2. Chapter 3 gives some principles of the importance sampling techniques, which are used in channel model designs. Chapter 4 introduces our simulation results on the MLSDA performance under the AWGN channel. Summary and conclusion appear in Chapter 5.

Chapter 2

A Brief Review of the MLSDA

The MLSDA decoder is briefly introduced in this chapter.

I Definitions and Notations

Consider a binary (n, k, m) convolutional code $\boldsymbol{C}$, where m is the memory order and k/n is the code rate. The constraint length is defined as $(m+1)$. The codeword length is $n(L+m)$, where L is the block length of the information bits. The encoding process starts from the initial node, named origin node, and then goes forward according to the information sequence. It can be represented as a code tree, where the origin node is the leftmost node, while the leaves, known as the terminal nodes, are at the rightmost positions of the tree. Each codeword is represented as a path starting from the origin node and ending at one terminal node. For each node of the first L levels, 2^k branches emanate from it, and only one branch stems from the nodes over the remaining levels.

If the same states of each level are merged, a new structure of the codeword representation, named trellis, results. The number of the states is settled by the total number of the shift-registers of the encoder. Like the code tree, a codeword can be represented as a path from the original node to the terminal node. In a trellis, there is only one terminal node remained because of merging.

Both the structures can be used in the MLSDA decoding. Here, the MLSDA we employed searches codewords on a trellis.

II Maximum-Likelihood Soft-Decision Sequential Decoding

In the traditional stack algorithm, the code tree is searched according to the Fano metrics in order to locate an estimate of the transmitted codeword. Here, the MLSDA decodes the codeword on the code trellis by using two stacks, and the path metric is replaced by the new metric

$$M(\boldsymbol{x}_{(\ell n-1)}) \triangleq \sum_{j=0}^{\ell n-1} (y_j \oplus x_j) \left| \ln \frac{Pr(r_j|0)}{Pr(r_j|1)} \right| \tag{2.1}$$

where $\boldsymbol{x}_{(\ell n-1)} = (x_0, x_1, \ldots, x_{\ell n-1}) \in \{0,1\}^{\ell n}$ are the codewords corresponding to the current path ending at level ℓ, $r_j = (-1)^{v_j}\sqrt{\mathcal{E}} + \lambda_j$ is the jth received scalar, $(v_0, v_1, \ldots, v_{\ell n-1}) \in \{0,1\}^{\ell n}$ is the originally transmitted codewords, $\mathcal{E}$ is the transmitted signal power per code bit, λ_j is the jth noise sample, and

$$y_j \triangleq \begin{cases} 1, & \text{if } \ln \dfrac{Pr(r_j|0)}{Pr(r_j|1)} < 0; \\ 0, & \text{otherwise,} \end{cases}$$

The algorithm is described below.

<The trellis-based MLSDA>

Step 1. Load the Open Stack with the origin node whose metric is assigned to be zero.

Step 2. Compute the metrics of the successors of the top path in the Open Stack, and put the ending state and the ending level of the top path into the Closed Stack. Delete the top path from the Open Stack.

Step 3. Whenever any of the new paths (i.e., the successors of the top path in the Open Stack in Step 2) merges with a path already in the Open Stack, eliminate the one with higher metric value. If any of the new paths merges with a path already in the Closed Stack, discard it.

Step 4. Insert the remaining new paths into the Open Stack, and reorder the Open Stack according to ascending metric values.

Step 5. If the top path in the Open Stack ends at the terminal node in the trellis, the algorithm stops; otherwise go to Step 2.

With the non-negativity of the new metric, the MLSDA decoder is proved to be an ML decoder, if no limitation on the stack size is placed. As expected, its performance can be a little degraded if finite stack size is assumed. Although there is unavoidably an upper limit on the stack size in practical simulation program, such limit is never reached in our simulations. Hence, our simulations can be viewed as ones with infinite stack size.

Since both the MLSDA and the Viterbi algorithms are ML decoders, they should yield the same error performance. The only difference between is on the computational efficiency. The computational complexity of a decoding algorithm can be defined as the number of the path metrics evaluated. As shown in a previous work [18, 19], the computational complexity of the MLSDA is much less than the Viterbi Algorithm especially when the SNR is high. Despite the superiority of the computational efficiency of the MLSDA, the BER estimation is still a difficult task as SNR grows. Consequently, alternative efficient techniques for the performance evaluation is required. Note that in the investigation of the computational complexity of the MLSDA, to account for the total metric values evaluated is more informational than to record the number of simulation trials. This is because the time consumed in each simulation run for the MLSDA is not a constant, and its average grows as the SNR decreases.

Chapter 3

Importance sampling techniques and simulation models for MLSDA decoders

This chapter describes the simulation models, as well as some mathematical preliminaries, of the thesis.

I Ordinary Monte Carlo Simulations

In general, a simulation environment can be described in Figure 3.1.

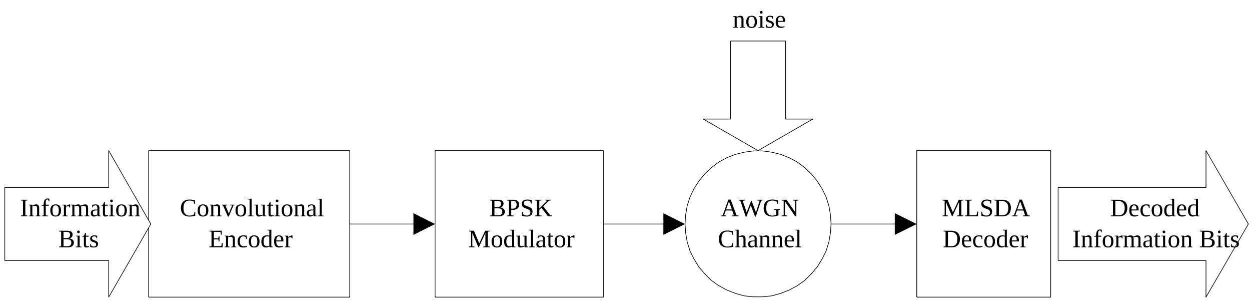

Figure 3.1: A general simulation model.

After L information bits are encoded by an (n, k, m) convolutional encoder, the binary codewords $\boldsymbol{x} = (x_0, x_1, ..., x_{n(L+m)-1})$ are then BPSK modulated and transmitted through the AWGN channel. The received vector becomes

$$r_j = (-1)^{x_j}\sqrt{\mathcal{E}} + \lambda_j, \tag{3.1}$$

where $0 \leq j \leq N-1$, $\mathcal{E}$ is the signal energy of each codeword bit, and λ_j is the jth noise sample. For mutually independent noise samples, the channel can be described by the conditional density

$$f(\boldsymbol{r}|\boldsymbol{x}) = \prod_{j=0}^{n(L+m)-1} f(r_j|x_j). \tag{3.2}$$

When a codeword $\boldsymbol{x}$ is transmitted, the probability that $\boldsymbol{x}'$ is decoded instead of $\boldsymbol{x}$ is

$$P(\boldsymbol{x}'|\boldsymbol{x}) = \int\cdots\int_{D(\boldsymbol{x}')} f(\boldsymbol{r}|\boldsymbol{x})d\boldsymbol{r} = \int\cdots\int_{\Re^{n(L+m)}} I_{\boldsymbol{x}'}(\boldsymbol{r})f(\boldsymbol{r}|\boldsymbol{x})d\boldsymbol{r} \tag{3.3}$$

where $D(\boldsymbol{x}')$ is the decoding region for $\boldsymbol{x}'$, and

$$I_{\boldsymbol{x}'}(\boldsymbol{r}) \triangleq \begin{cases} 1, & \boldsymbol{r} \in D(\boldsymbol{x}'); \\ 0, & \text{otherwise.} \end{cases}$$

Afterwards, the overall BER for transmitting $\boldsymbol{x}$ can be calculated as

$$P_b(\boldsymbol{x}) = \frac{1}{L}E[N_b|\boldsymbol{x}] = \frac{1}{L}\sum_{\boldsymbol{x}' \in \boldsymbol{C}} n_b(\boldsymbol{x}, \boldsymbol{x}')P(\boldsymbol{x}'|\boldsymbol{x}) \tag{3.4}$$

where $n_b(\boldsymbol{x}, \boldsymbol{x}')$ is the number of bit errors of decoding $\boldsymbol{x}'$ other than $\boldsymbol{x}$. Since the code considered is a linear code, the average BER is independent of the transmitted codeword; hence, $P_b = P_b(\boldsymbol{x})$. A criterion named as *relative precision estimate* (RPE) [3, 7] can be used to access the accuracy of the estimated BER. The RPE is defined as the ratio of the standard deviation of the BER divided by the BER, i.e.,

$$\frac{\sqrt{\text{Var}[P_b]}}{P_b} \times 100\%.$$

When the RPE is small, we can say the estimated BER falls around the estimated value with high probability.

Owing to the inavailability of the closed form expressions of many complex communication systems, Monte Carlo simulations is popular in estimating the system performance.

However, the MC method needs more than $100/P_b$ simulation trials to obtain an accurate evaluation of P_b. Apparently, the method is infeasible as P_b is below 10^{-8}, where about 10^{12} computational complexity is required. Therefore, modified MC simulation techniques, named *Importance Sampling* (IS) techniques are developed to combat the problem.

II Importance sampling Techniques

To achieve the purpose of reducing the computational complexity of the simulations, a small variance or RPE of the estimated P_b has to be obtained through fewer computations. By dividing and multiplying an IS density $f^*(\boldsymbol{r}|\boldsymbol{x})$, (3.3) becomes

$$P(\boldsymbol{x}'|\boldsymbol{x}) = \int \cdots \int_{\Re^{n(L+m)}} I_{\boldsymbol{x}'}(\boldsymbol{r}) \frac{f(\boldsymbol{r}|\boldsymbol{x})}{f^*(\boldsymbol{r}|\boldsymbol{x})} f^*(\boldsymbol{r}|\boldsymbol{x}) d\boldsymbol{r} = \int \cdots \int_{\Re^{n(L+m)}} I_{\boldsymbol{x}'}(\boldsymbol{r}) w(\boldsymbol{r}|\boldsymbol{x}) f^*(\boldsymbol{r}|\boldsymbol{x}) d\boldsymbol{r} \tag{3.5}$$

where

$$w(\boldsymbol{r}|\boldsymbol{x}) \stackrel{\triangle}{=} \frac{f(\boldsymbol{r}|\boldsymbol{x})}{f^*(\boldsymbol{r}|\boldsymbol{x})} \tag{3.6}$$

is the *weight function*. From equation (3.5), one can realize that for the IS technique, the original channel is first biased to a more erroneous channel f^* to lift the error occurring rate, and then the calculation of the estimated error is weighted by the weight function to recover the true BER.

For an N_{IS}-trial simulation, the estimated probability of $P(\boldsymbol{x}'|\boldsymbol{x})$ is

$$\widehat{P}^*_{N_{\text{IS}}}(\boldsymbol{x}'|\boldsymbol{x}) = \frac{1}{N_{\text{IS}}} \sum_{\ell=1}^{N_{\text{IS}}} w(\boldsymbol{r}^{(\ell)}|\boldsymbol{x}) I_{\boldsymbol{x}'}(\boldsymbol{r}^{(\ell)}) \tag{3.7}$$

and the overall BER is

$$\widehat{P}^*_b = \widehat{P}^*_b(\boldsymbol{x}) = \sum_{\boldsymbol{x}' \in \boldsymbol{C}} n_b(\boldsymbol{x}', \boldsymbol{x}) \widehat{P}^*_{N_{\text{IS}}}(\boldsymbol{x}'|\boldsymbol{x}). \tag{3.8}$$

We can examine the expectation of the estimated $\widehat{P}^*_{N_{\text{IS}}}(\boldsymbol{x}'|\boldsymbol{x})$, and obtain

$$E^*\left[\widehat{P}^*_{N_{\text{IS}}}(\boldsymbol{x}'|\boldsymbol{x})\right] = E^*\left[w(\boldsymbol{r}^{(\ell)}|\boldsymbol{x}) I_{\boldsymbol{x}'}(\boldsymbol{r}^{(\ell)})\right]$$

$$
\begin{aligned}
&= \int \cdots \int_{\Re^{n(L+m)}} I_{\boldsymbol{x}'}(\boldsymbol{r}) w(\boldsymbol{r}|\boldsymbol{x}) f^*(\boldsymbol{r}|\boldsymbol{x}) d\boldsymbol{r} \\
&= \int \cdots \int_{\Re^{n(L+m)}} I_{\boldsymbol{x}'}(\boldsymbol{r}) \frac{f(\boldsymbol{r}|\boldsymbol{x})}{f^*(\boldsymbol{r}|\boldsymbol{x})} f^*(\boldsymbol{r}|\boldsymbol{x}) d\boldsymbol{r} \\
&= P(\boldsymbol{x}'|\boldsymbol{x}). \qquad (3.9)
\end{aligned}
$$

So the estimate on $\widehat{P}^*_{N_{\mathrm{IS}}}(\boldsymbol{x}'|\boldsymbol{x})$ is unbiased.

As previously mentioned, the usage of the RPE can provide us the accuracy of the estimated BER. We calculate the RPE as:

$$
\begin{aligned}
\mathrm{Var}^*\left[\widehat{P}^*_b\right] &= \mathrm{Var}^*\left[\widehat{P}^*_b(\boldsymbol{x})\right] \\
&= \mathrm{Var}^*\left[\frac{1}{L}\sum_{\boldsymbol{x}'\in\boldsymbol{C}} n_b(\boldsymbol{x}',\boldsymbol{x})\widehat{P}^*_{N_{\mathrm{IS}}}(\boldsymbol{x}'|\boldsymbol{x})\right] \\
&= \frac{1}{L^2}\mathrm{Var}^*\left[\sum_{\boldsymbol{x}'\in\boldsymbol{C}} n_b(\boldsymbol{x}',\boldsymbol{x})\widehat{P}^*_{N_{\mathrm{IS}}}(\boldsymbol{x}'|\boldsymbol{x})\right] \\
&= \frac{1}{L^2}\mathrm{Var}^*\left[\sum_{\boldsymbol{x}'\in\boldsymbol{C}} n_b(\boldsymbol{x}',\boldsymbol{x})\frac{1}{N_{\mathrm{IS}}}\sum_{\ell=1}^{N_{\mathrm{IS}}} w(\boldsymbol{r}^{(\ell)}|\boldsymbol{x}) I_{\boldsymbol{x}'}(\boldsymbol{r}^{(\ell)})\right] \\
&= \frac{1}{L^2 N_{\mathrm{IS}}}\mathrm{Var}^*\left[w(\boldsymbol{r}^{(\ell)}|\boldsymbol{x})\sum_{\boldsymbol{x}'\in\boldsymbol{C}} n_b(\boldsymbol{x}',\boldsymbol{x}) I_{\boldsymbol{x}'}(\boldsymbol{r}^{(\ell)})\right] \\
&= \frac{1}{L^2 N_{\mathrm{IS}}}\left(E^*\left[w^2(\boldsymbol{r}^{(\ell)}|\boldsymbol{x})\left(\sum_{\boldsymbol{x}'\in\boldsymbol{C}} n_b(\boldsymbol{x}',\boldsymbol{x}) I_{\boldsymbol{x}'}(\boldsymbol{r}^{(\ell)})\right)^2\right]\right. \\
&\qquad \left. - \left(E^*\left[w(\boldsymbol{r}^{(\ell)}|\boldsymbol{x})\sum_{\boldsymbol{x}'\in\boldsymbol{C}} n_b(\boldsymbol{x}',\boldsymbol{x}) I_{\boldsymbol{x}'}(\boldsymbol{r}^{(\ell)})\right]\right)^2\right) \\
&= \frac{1}{L^2 N_{\mathrm{IS}}}\left(E^*\left[w^2(\boldsymbol{r}^{(\ell)}|\boldsymbol{x})\left(\sum_{\boldsymbol{x}'\in\boldsymbol{C}} n_b^2(\boldsymbol{x}',\boldsymbol{x}) I_{\boldsymbol{x}'}(\boldsymbol{r}^{(\ell)})\right)\right] - L^2 P_b^2(\boldsymbol{x})\right) \qquad (3.10) \\
&= \frac{1}{L^2 N_{\mathrm{IS}}}\left(\sum_{\boldsymbol{x}'\in\boldsymbol{C}} n_b^2(\boldsymbol{x}',\boldsymbol{x}) E^*\left[w^2(\boldsymbol{r}^{(\ell)}|\boldsymbol{x}) I_{\boldsymbol{x}'}(\boldsymbol{r}^{(\ell)})\right] - L^2 P_b^2\right), \qquad (3.11)
\end{aligned}
$$

where (3.10) follows since

$$
\begin{aligned}
\left(\sum_{\boldsymbol{x}'\in\boldsymbol{C}} n_b(\boldsymbol{x}',\boldsymbol{x}) I_{\boldsymbol{x}'}(\boldsymbol{r}^{(\ell)})\right)^2 &= \sum_{\boldsymbol{x}'\in\boldsymbol{C}} n_b^2(\boldsymbol{x}',\boldsymbol{x}) I_{\boldsymbol{x}'}(\boldsymbol{r}^{(\ell)}) \\
&\quad + \sum_{\boldsymbol{x}'\neq\boldsymbol{x}''} n_b(\boldsymbol{x}',\boldsymbol{x}) n_b(\boldsymbol{x}'',\boldsymbol{x}) I_{\boldsymbol{x}'}(\boldsymbol{r}^{(\ell)}) I_{\boldsymbol{x}''}(\boldsymbol{r}^{(\ell)}),
\end{aligned}
$$

and $I_{\boldsymbol{x}'}(\boldsymbol{r}^{(\ell)})I_{\boldsymbol{x}''}(\boldsymbol{r}^{(\ell)}) = 0$ whenever $\boldsymbol{x}' \neq \boldsymbol{x}''$, and equations (3.4) and (3.9) together imply that

$$P_b(\boldsymbol{x}) = \frac{1}{L}\sum_{\boldsymbol{x}'\in\boldsymbol{C}} n_b(\boldsymbol{x},\boldsymbol{x}')E^*\left[w(\boldsymbol{r}^{(\ell)}|\boldsymbol{x})I_{\boldsymbol{x}'}(\boldsymbol{r}^{(\ell)})\right].$$

In estimation, $E^*\left[w^2(\boldsymbol{r}^{(\ell)}|\boldsymbol{x})I_{\boldsymbol{x}'}(\boldsymbol{r}^{(\ell)})\right]$ and P_b in (3.11) can be replaced by:

$$\frac{1}{N_{\text{IS}}}\sum_{\ell=1}^{N_{\text{IS}}} w^2(\boldsymbol{r}^{(\ell)}|\boldsymbol{x})I_{\boldsymbol{x}'}(\boldsymbol{r}^{(\ell)})$$

and $\widehat{P}_b^*$, where $\boldsymbol{r}^{(\ell)}$ for $1 \leq \ell \leq N_{\text{IS}}$ are i.i.d. generated according to biased distribution $f^*(\cdot|\boldsymbol{x})$.

Now we are ready to design the biased channels.

III Simulation models

The convolutional code used in our simulations is a (2,1,6) code with generators 634,564 (octal) and information length 60. The free distance of the code is 10, and the number of non-zero codewords, whose Hamming weight equals the free distance, and who are encoded from the information sequences of weights between 1 and 7, is 668 [8]. For channels with weak noise power, errors barely occur even when large simulation trials are taken. Observe that the erroneously decisive codewords are those whose Hamming distance with respect to the transmitted codeword is equal to the free distance. These "*important*" error events can be determined by the *distance spectrum*. Consequently, we can derive the IS channel density to force the occurrence of these error events to obtain an efficient estimate of system performance. Both biased stationary and biased non-stationary channels are considered here.

III.1 Biased Stationary Channels

For biased stationary channel, each noise sample is stationarily biased with larger noise variance, and thus a noisier channel results. With such a channel, erroneous decoding is expected to occur more often. As graphically depicted in Figure 3.2, the channels are stationary-biased by the VS method only for the MT method would yield a small weight function. Consequently, each channel output is $-1 + N_{\mathrm{IS}}$ if an all-zero codeword is transmitted. For ease of understanding, the pdf of the biased noise sample, as well as that of the original noise sample, is illustrated in Figure 3.3.

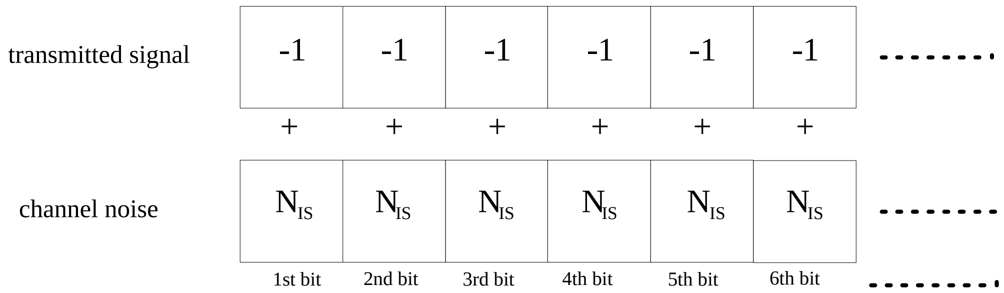

Figure 3.2: The biased stationary channel.

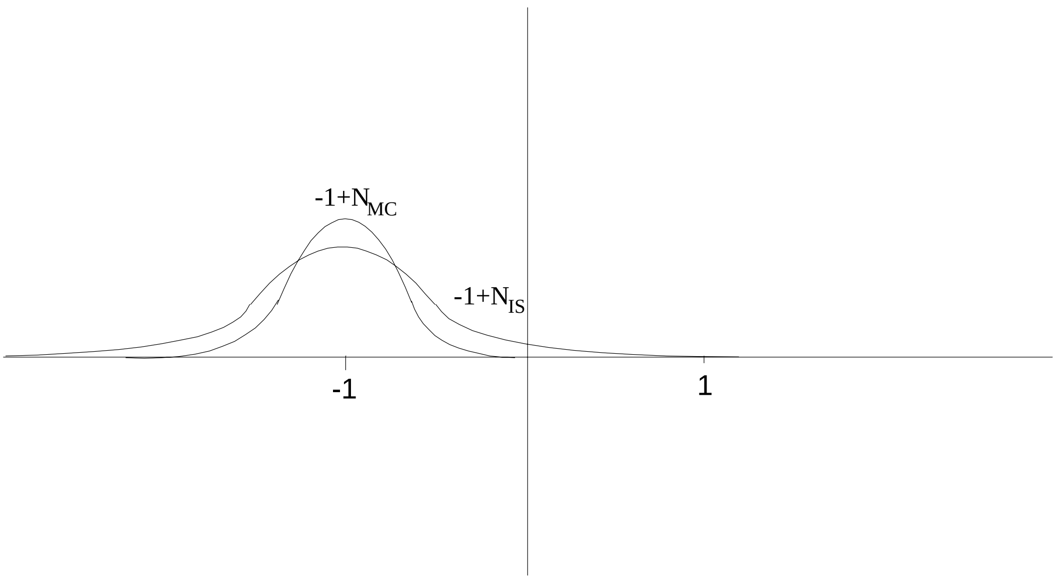

Figure 3.3: The pdf of the received signals through the original and biased stationary channels.

III.2 Biased Non-Stationary Channels

In our design, only "free-distance" codewords with respective to the transmitted codeword are considered in the simulation of the *biased non-stationary channels*. As aforementioned, the number of the "free-distance" codewords for the considered convolutional code is prohibitively large. In order to force these important error events to occur more frequently, the noises at those positions, where $d(\boldsymbol{x}, \boldsymbol{x}') \neq 0$, are amplified by means of the *VS* or *MT* methods, while noises at other positions remained unchanged or slightly enlarged. As shown in Figure 3.4, signals interfered by noises N' are likely to enter the region of $D(\boldsymbol{x}')$ if an all-zero codeword is transmitted.

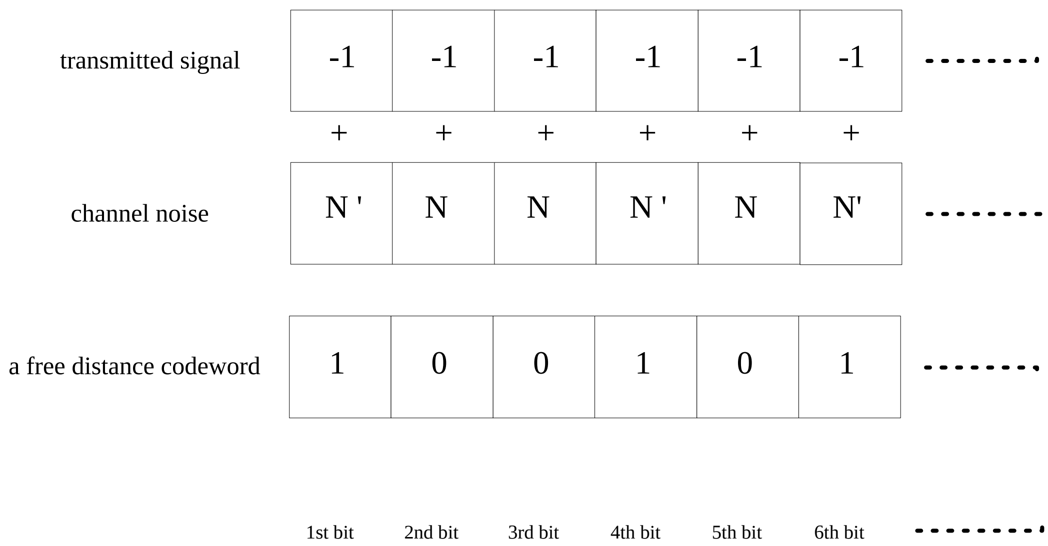

Figure 3.4: The biased non-stationary channel.

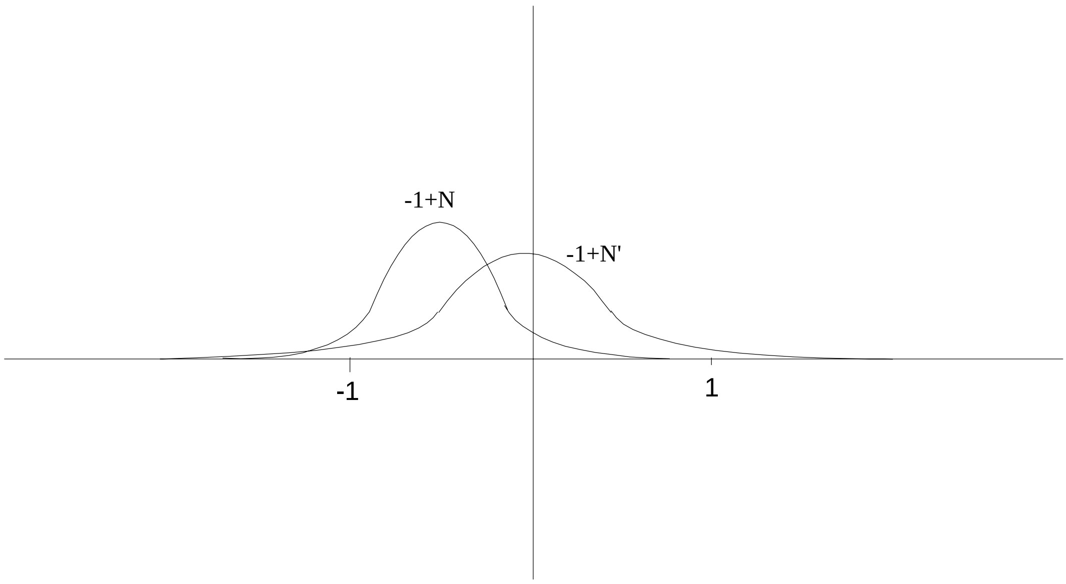

Figure 3.5: The pdf of the received signals through the original and biased non-stationary channels.

Chapter 4

Simulation Results

This chapter contains the simulation results corresponding to the two kinds of channels described in the previous chapter.

I Biased Stationary Channels

Figures 4.1 and 4.2 respectively show the BER estimates and RPE of the channel with E_s/N_0 being 3dB. Since the BER is about 10^{-3}, the superiority of the IS simulator is not as expected. Specifically, IS simulator based on biased stationary channels does not help improving the efficiency of the MC simulator. In addition, the erroneous estimated BER when biased noise variance with respect to $(E_s/N_0)^* = 0$dB is taken is due to that the error pattern has been altered to be away from the desired SNR and the weight function is too small to compensate it back. The uselessness of taking biased stationary channels can be deduced from the simulation results that higher computational complexity is rendered for the IS estimator with biased stationary channel.

Similar results can be observed when E_s/N_0 is increased to 4dB as shown in Figures 4.3 and 4.4. Although the BER estimate of the IS simulation with biased noise being $(E_s/N_0)^* = 3$dB is quite close to the MC estimator, higher computational complexity is unfortunately required.

As SNR is further increased to 5dB, Figures 4.5 and 4.6 reveal the improvement of the IS simulation with biased stationary channel at $(E_s/N_0)^* = 4$dB; but the improvement is not apparent.

Figures 4.7-4.10 describe the BER estimates and RPE for SNR=6dB and SNR=7dB with 10^6 simulating runs. Since the true BER is quite low in these cases, tremendously long simulation time is required to acquire an accurate MC estimate. When the biased stationary IS density whose resultant SNR is higher than 5dB is taken, we can predict that a more efficient simulation (than the MC simulation) can be achieved since the RPE drops more rapidly than the MC estimator whose computational complexity is indeed very high [18].

Tables 4.1-4.4 list the estimated BER and the required computational complexities subject to RPE$\leq$ 10%. Better improvements in computational complexity are exhibited when slightly lower $(E_s/N_0)^*$ than the true (E_s/N_0) is employed.

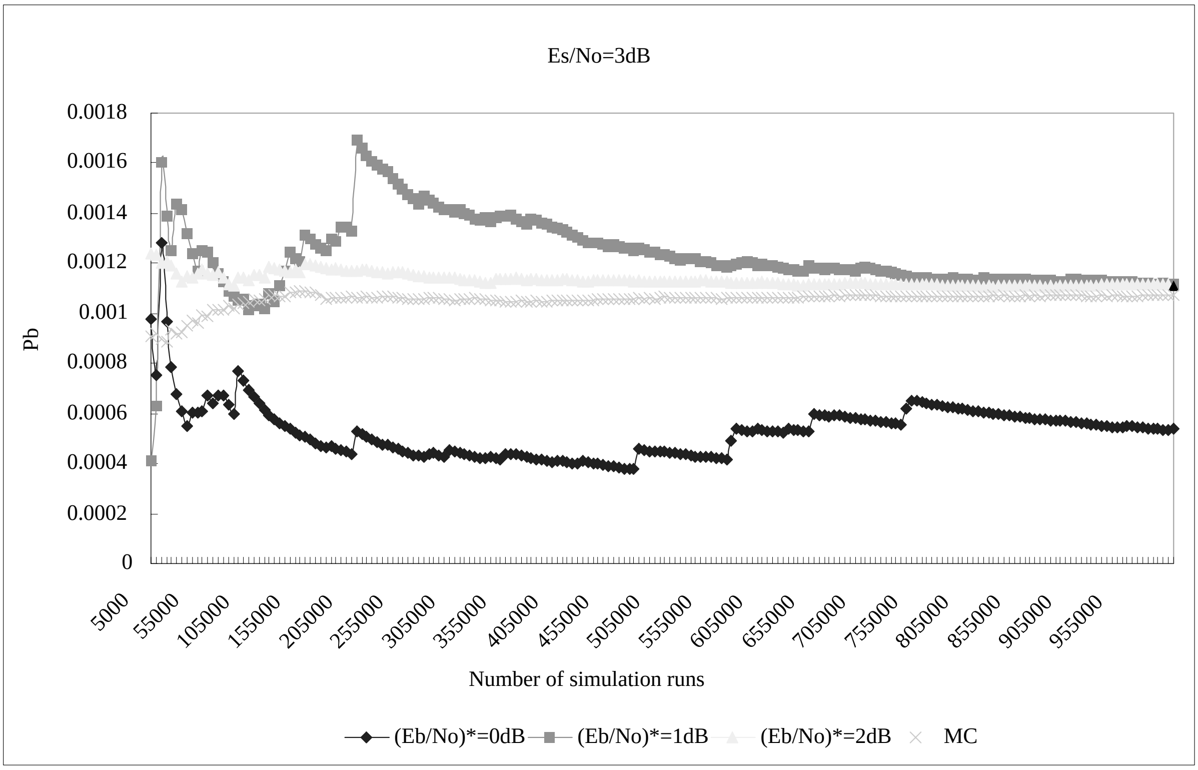

Figure 4.1: The BER estimate by IS simulations with biased stationary channels under $E_s/N_0 = 3\text{dB}$.

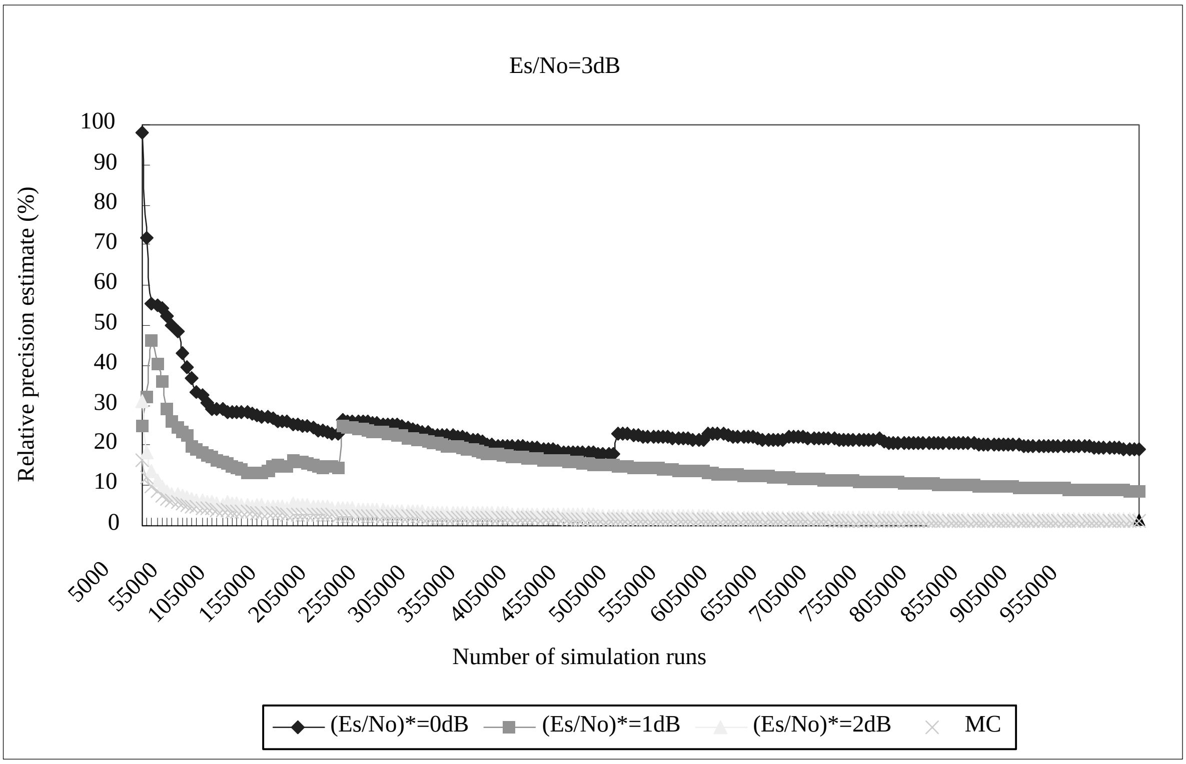

Figure 4.2: The RPE by IS simulations with biased stationary channels under $E_s/N_0 = 3$dB.

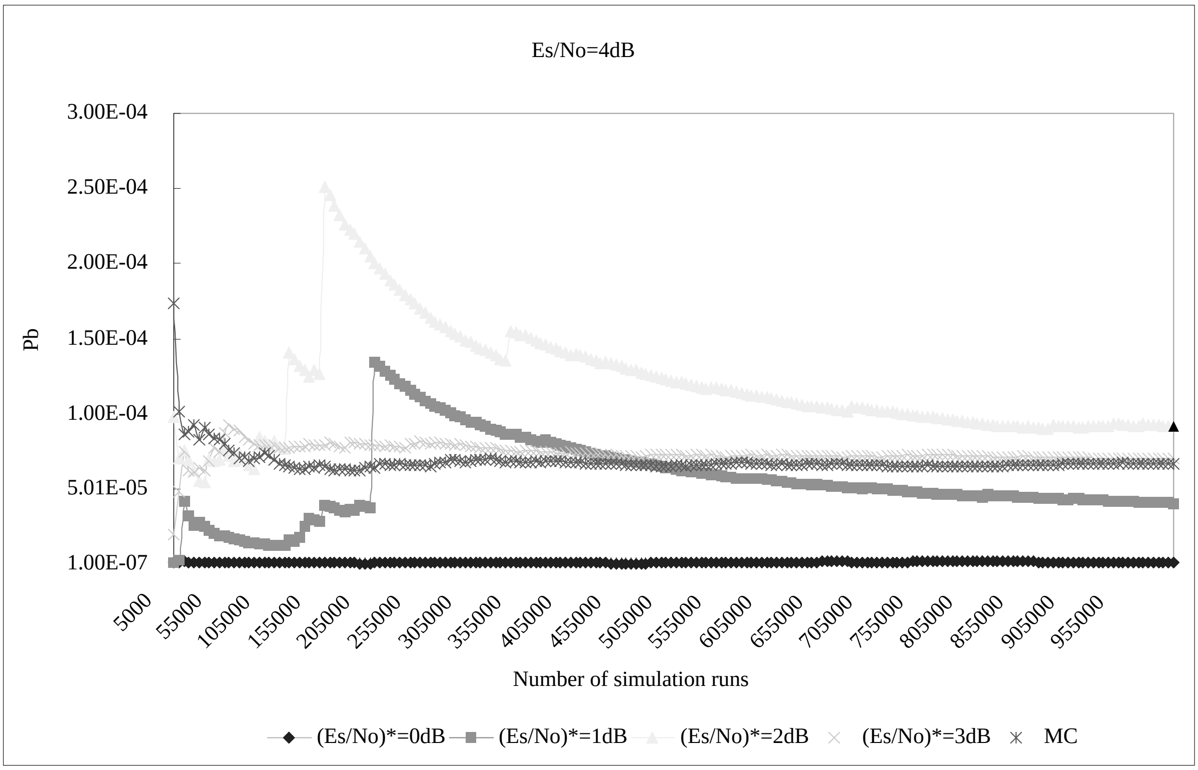

Figure 4.3: The BER estimate by IS simulations with biased stationary channels under $E_s/N_0 = 4$dB.

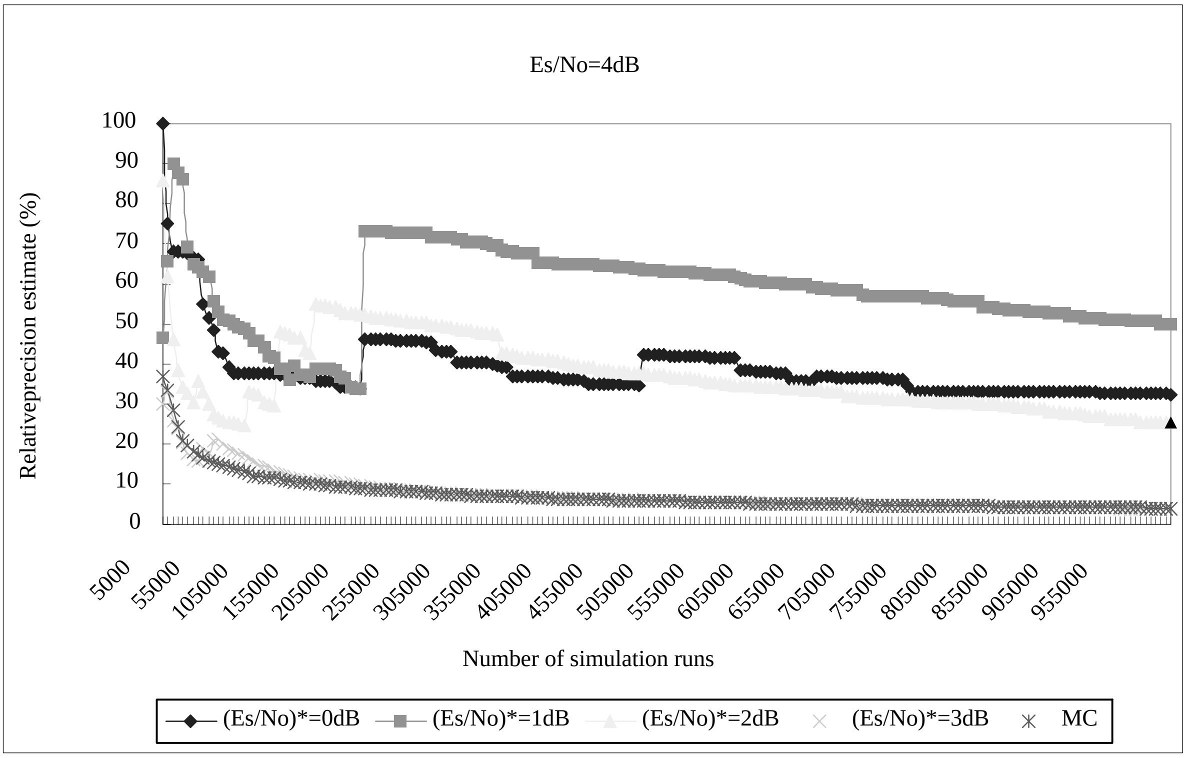

Figure 4.4: The RPE by IS simulations with biased stationary channels under $E_s/N_0 = 4$dB.

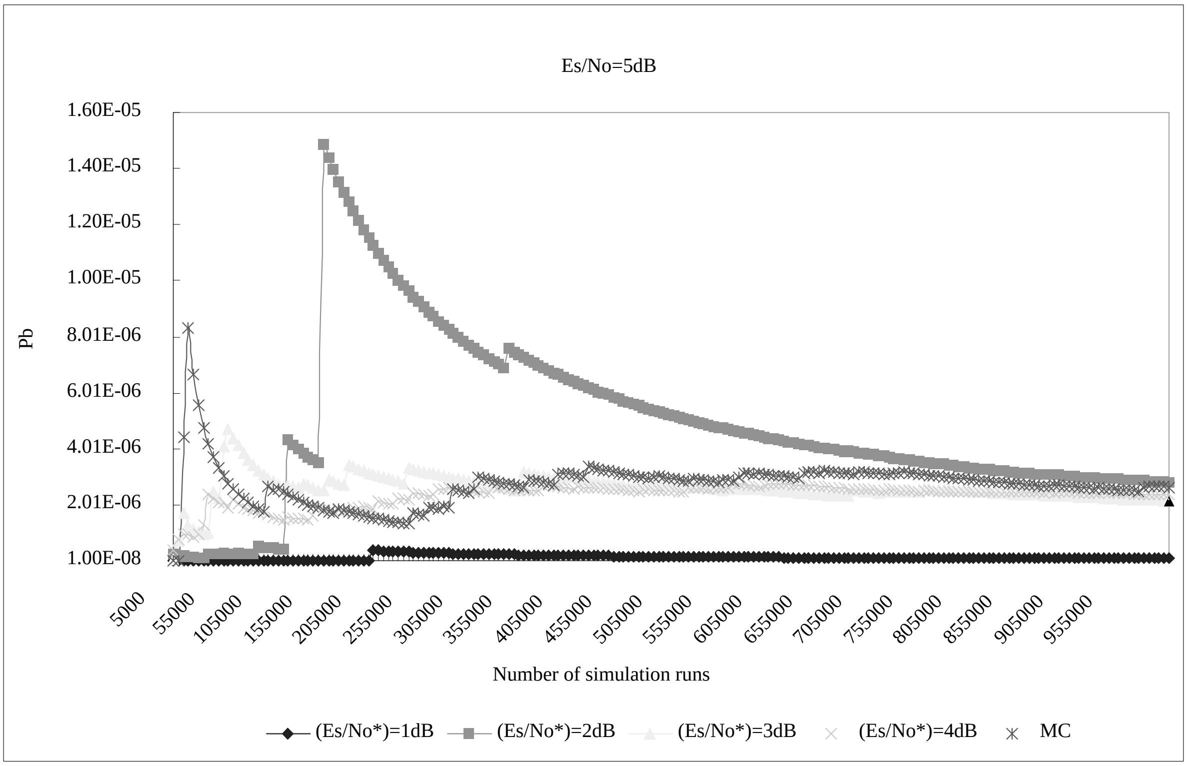

Figure 4.5: The BER estimate by IS simulations with biased stationary channels under $E_s/N_0 = 5$dB.

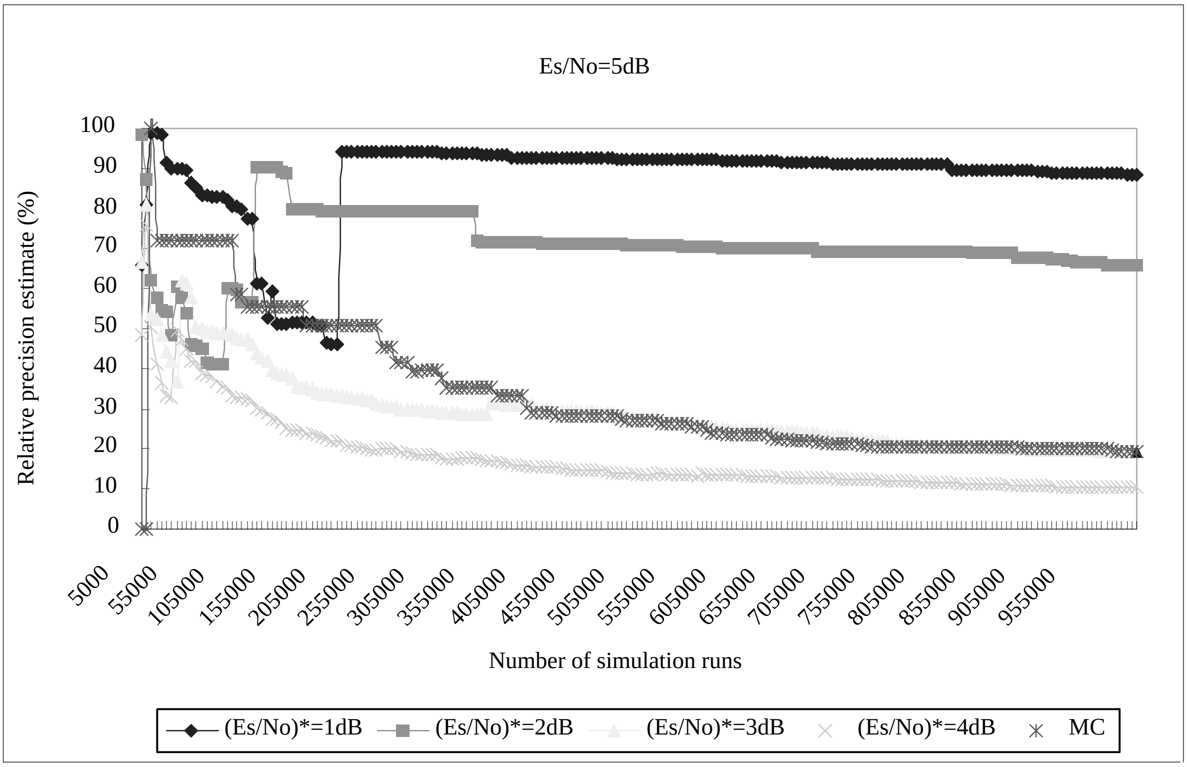

Figure 4.6: The RPE by IS simulations with biased stationary channels under $E_s/N_0 = 5$dB.

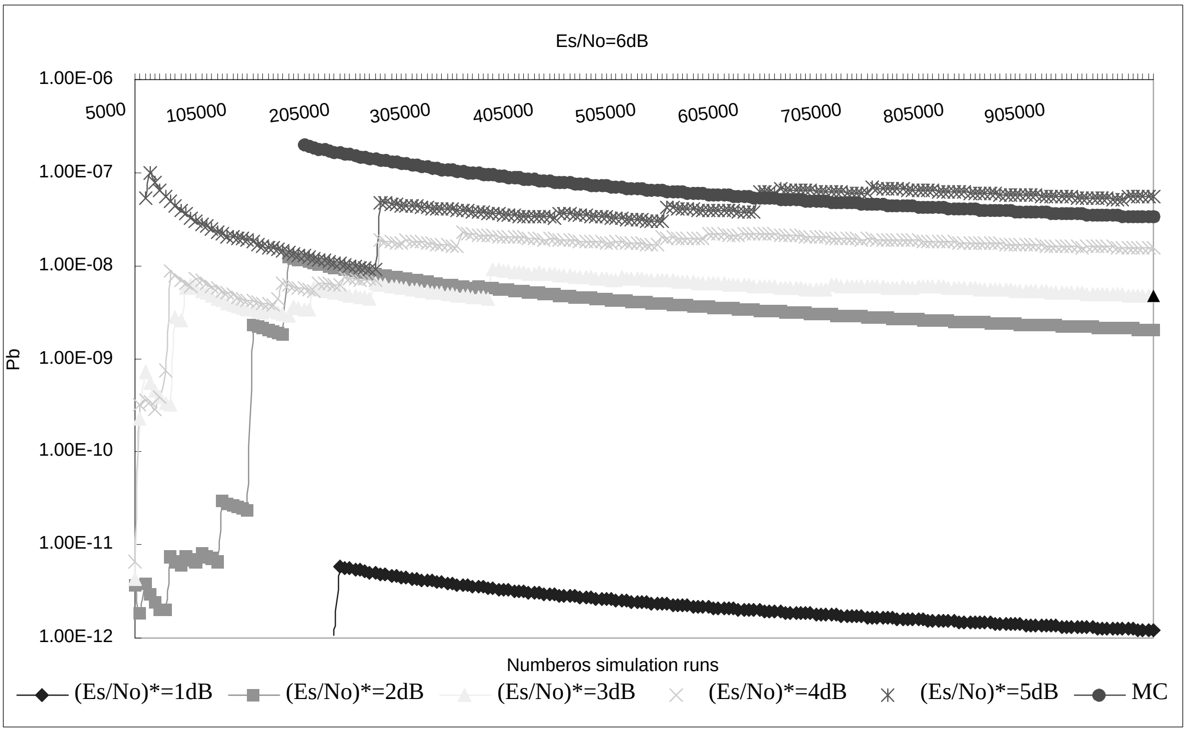

Figure 4.7: The BER estimate by IS simulations with biased stationary channels under $E_s/N_0 = 6$dB.

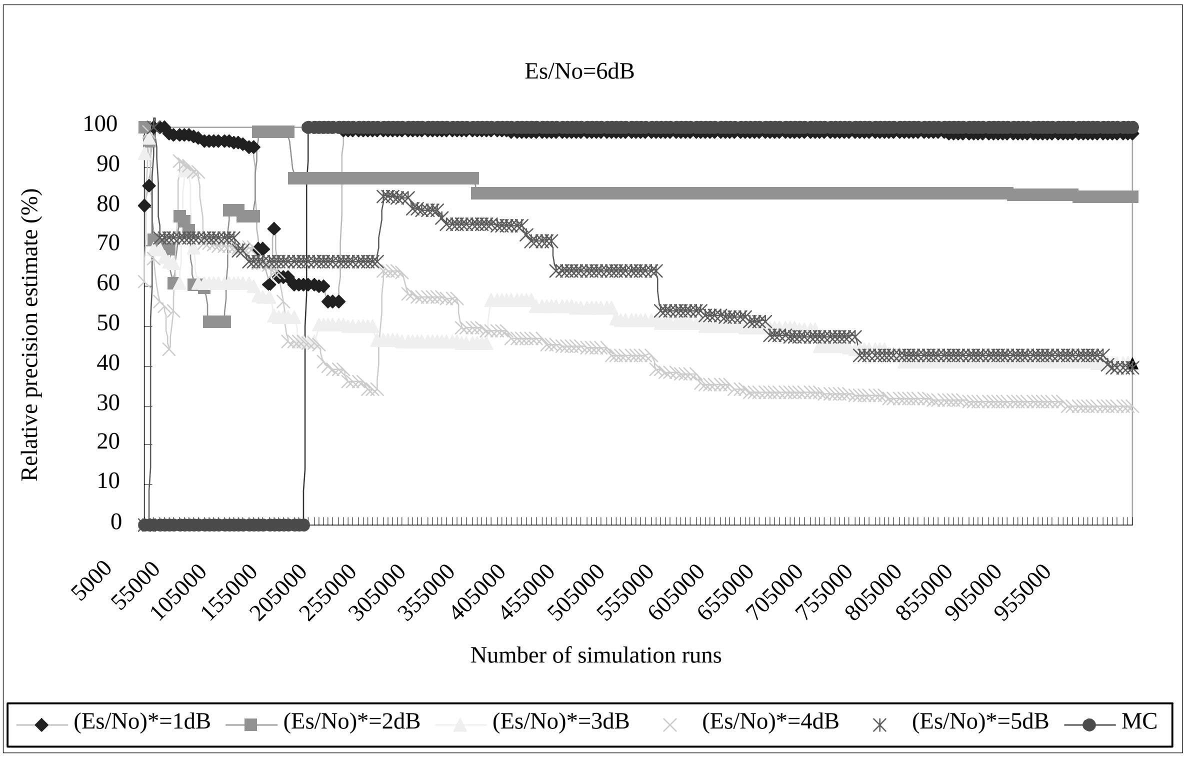

Figure 4.8: The RPE by IS simulations with biased stationary channels under $E_s/N_0 = 6$dB.

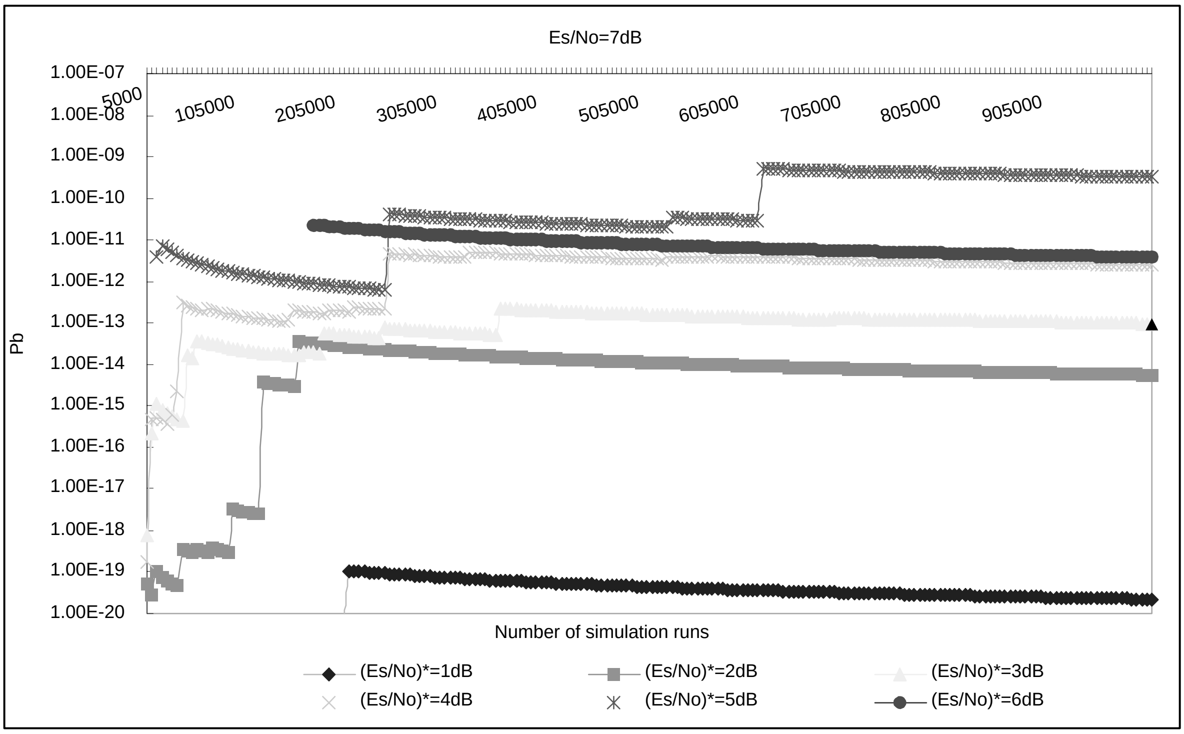

Figure 4.9: The BER estimate by IS simulations with biased stationary channels under $E_s/N_0 = 7$dB.

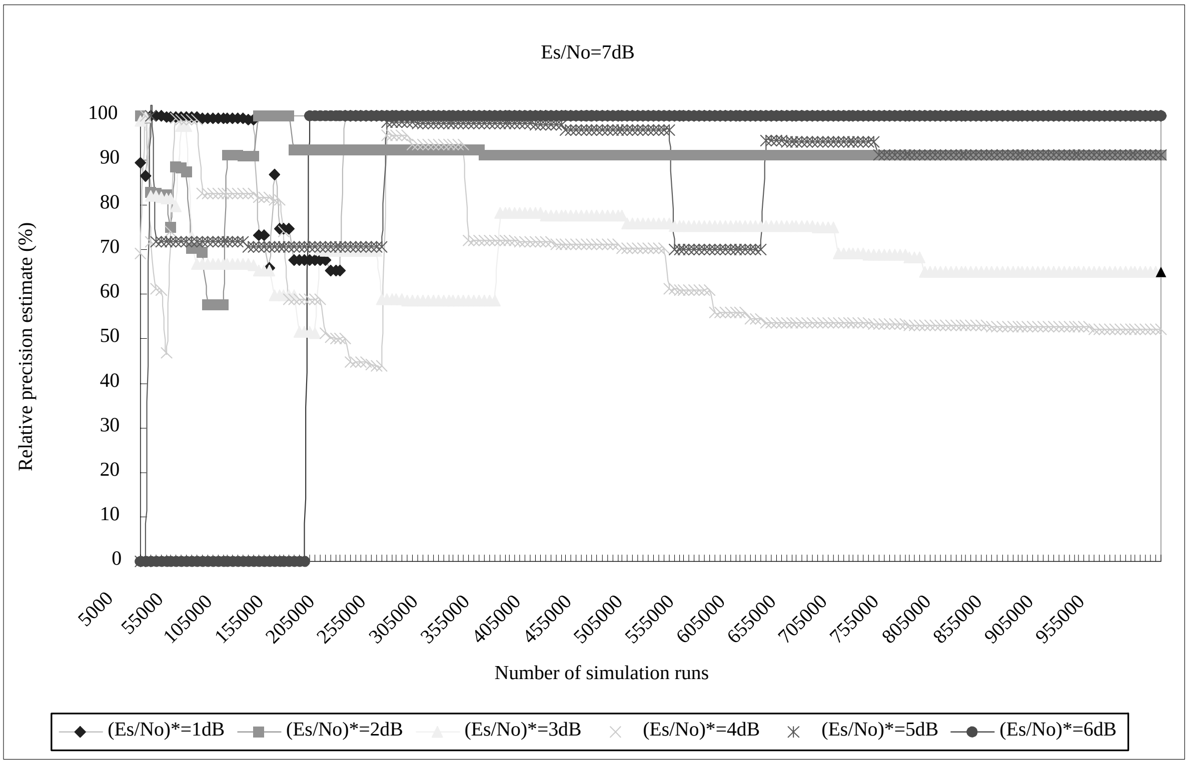

Figure 4.10: The RPE by IS simulations with biased stationary channels under $E_s/N_0 = 7$dB.

Es/No=3dB				
	pb	rpe	complexity	IS-complexity/MC-complexity
(Es/No)*=2dB	1.19E-03	9.99%	6.26E+07	3.846251182
(Es/No)*=2.5dB	8.86E-04	9.93%	1.40E+07	0.860821386
MC	8.65E-04	9.98%	1.63E+07	1

Table 4.1: The list of IS estimates for biased stationary under $E_s/N_0 =$ 3dB.

Es/No=4dB				
	pb	rpe	complexity	IS-complexity/MC-complexity
(Es/No)*=2dB	6.56E-05	10.00%	1.12E+10	201.4669211
(Es/No)*=3dB	8.03E-05	9.99%	2.18E+08	3.930713075
(Es/No)*=3.5dB	6.10E-05	9.98%	4.92E+07	0.888036214
MC	6.69E-05	9.97%	5.54E+07	1

Table 4.2: The list of IS estimates for biased stationary under $E_s/N_0 =$ 4dB.

Es/No=5dB				
	pb	rpe	complexity	IS-complexity/MC-complexity
(Es/No)*=3dB	1.76E-06	10.00%	6.76E+09	11.71058356
(Es/No)*=4dB	2.24E-06	9.99%	4.11E+08	0.71176908
(Es/No)*=4.5dB	2.70E-06	9.99%	3.54E+08	0.61317836
MC	2.68E-06	9.98%	5.77E+08	1

Table 4.3: The list of IS estimates for biased stationary under $E_s/N_0 =$ 5dB.

Es/No=6dB				
	pb	rpe	complexity	IS-complexity/MC-complexity
(Es/No)*=4dB	3.87E-08	9.98%	3.30E+10	1.467250732
(Es/No)*=5dB	4.22E-08	9.99%	6.83E+09	0.303572365
(Es/No)*=5.5dB	3.93E-08	9.99%	6.90E+09	0.306531639
MC	4.49E-08	9.97%	2.25E+10	1

Table 4.4: The list of IS estimates for biased stationary under $E_s/N_0 =$ 6dB.

II Biased Non-Stationary Channels

As discussed in the previous section, biased stationary IS density can improve the efficiency of the MC simulation when SNR is large. Only 1/3 computational efforts are required when SNR=6dB, and more are expected to save when SNR is higher. We further modify the IS simulator by means of biased non-stationary density described in the previous chapter. As simulation results indicate, when SNR is lower than 4dB, the error probability is not dominated by the "free-distance" error patterns; so the non-stationary IS technique is not suitable for such biased channels.

Figure 4.11 and 4.12 are the simulation results for estimating the BER under $E_s/N_0 =$ 4dB. In these figures, mean0 and SNR0 are the mean and resultant SNR value corresponding to $(-1+N)$ in Figure 3.5. Likewise, mean1 and SNR1 are those corresponding to $(-1+N')$ in Figure 3.5. It is noted from our simulation results that if mean1 is moved to 0, the estimated BER is not very accurate. This is because of the small weight function values to the dominant error events.

When true SNR=5dB, the simulation results are summarized in Figures 4.13 and 4.14. In the two figures, the mean is not that far from where the signal transmitted. As a result, better computational efficiency is gained.

When SNR is further raised to 6dB, the improvement is more significant, which are summarized in Figures 4.15 and 4.16. Even if we move the mean1 from -1 to -0.7, better efficiency can still be achieved. We also experimented the biased non-stationary channels with only 3 out of 10 positions are mean translated, contrary to the all-position-mean-translated cases. Despite of fewer decoded errors when being compared with other cases, the resultant RPE curves are steeper. As described in Chapter 3, the weight function is the product of the ratio, $f(r_j|x_j)/f^*(r_j|x_j)$, which is so small in such over-biased channels. Larger weight function values can be obtained by only moderately biasing most of the contributors, at the expense of fewer error occurrence. No matter what, better efficiency than the MC and stationary IS techniques is observed a when good non-stationary IS simulator is utilized as listed in Tables 4.5 and 4.6.

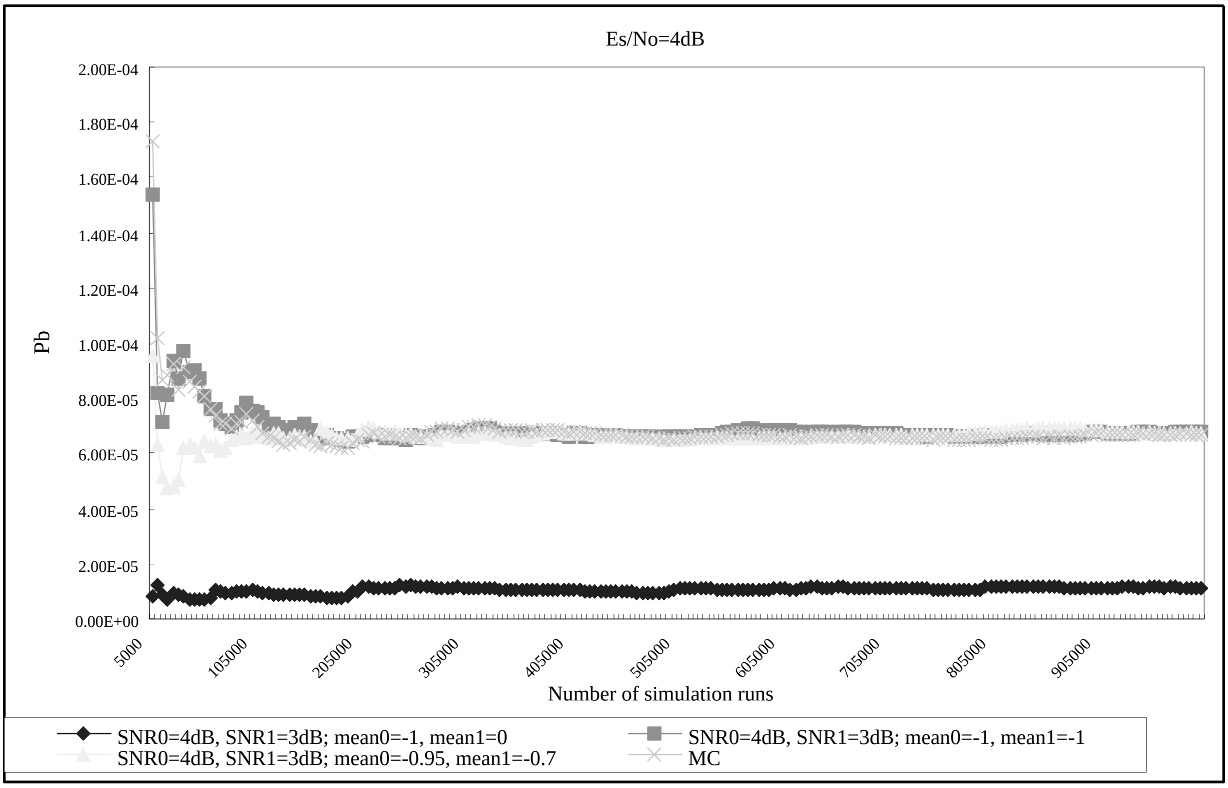

Figure 4.11: The BER estimate by IS simulations with biased non-stationary channels under $E_s/N_0 = 4$dB.

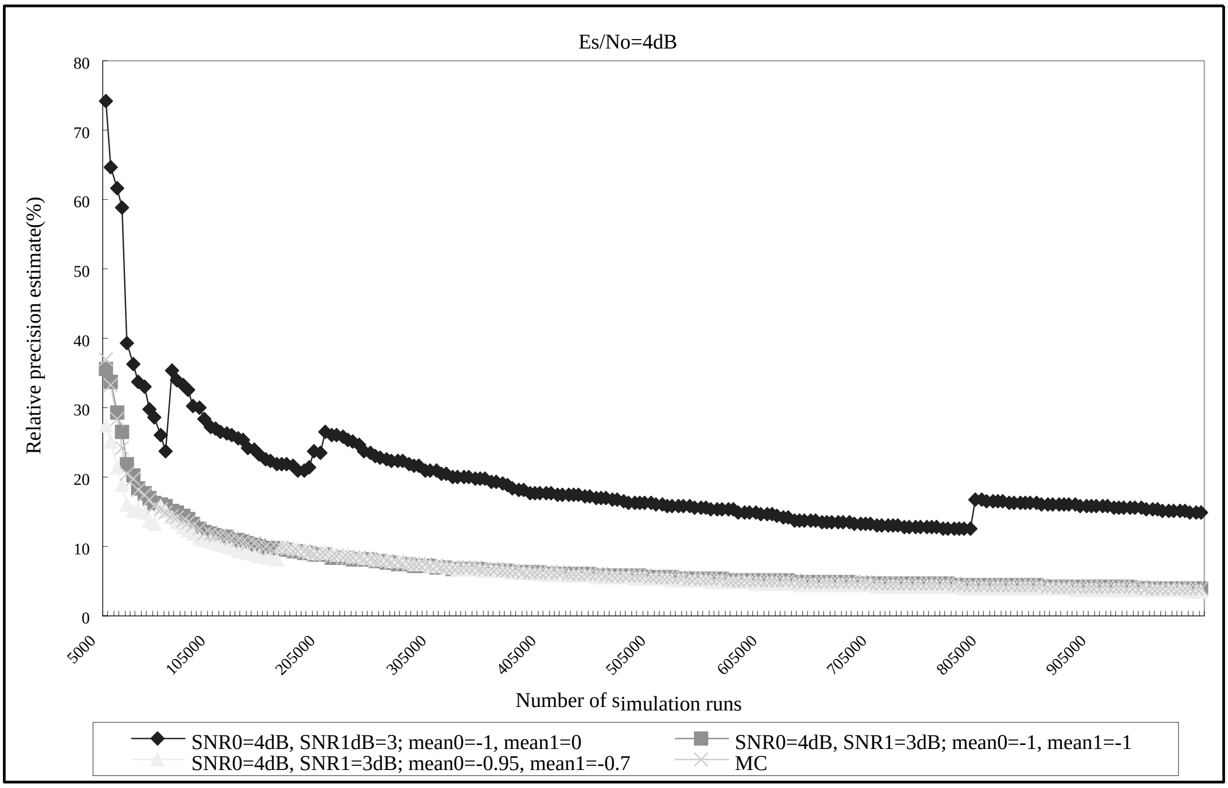

Figure 4.12: The RPE by IS simulations with biased non-stationary channels under $E_s/N_0 = 4$dB.

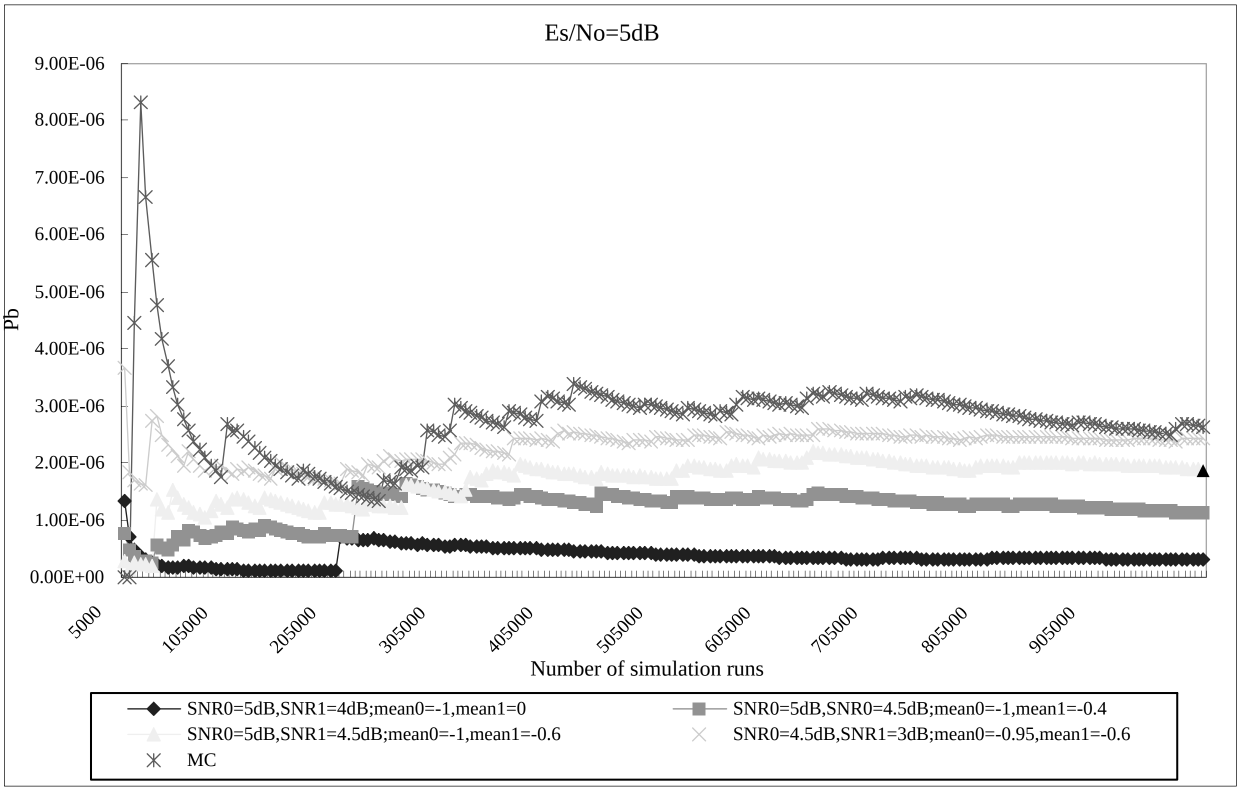

Figure 4.13: The BER estimate by IS simulations with biased non-stationary channels under $E_s/N_0 = 5\text{dB}$.

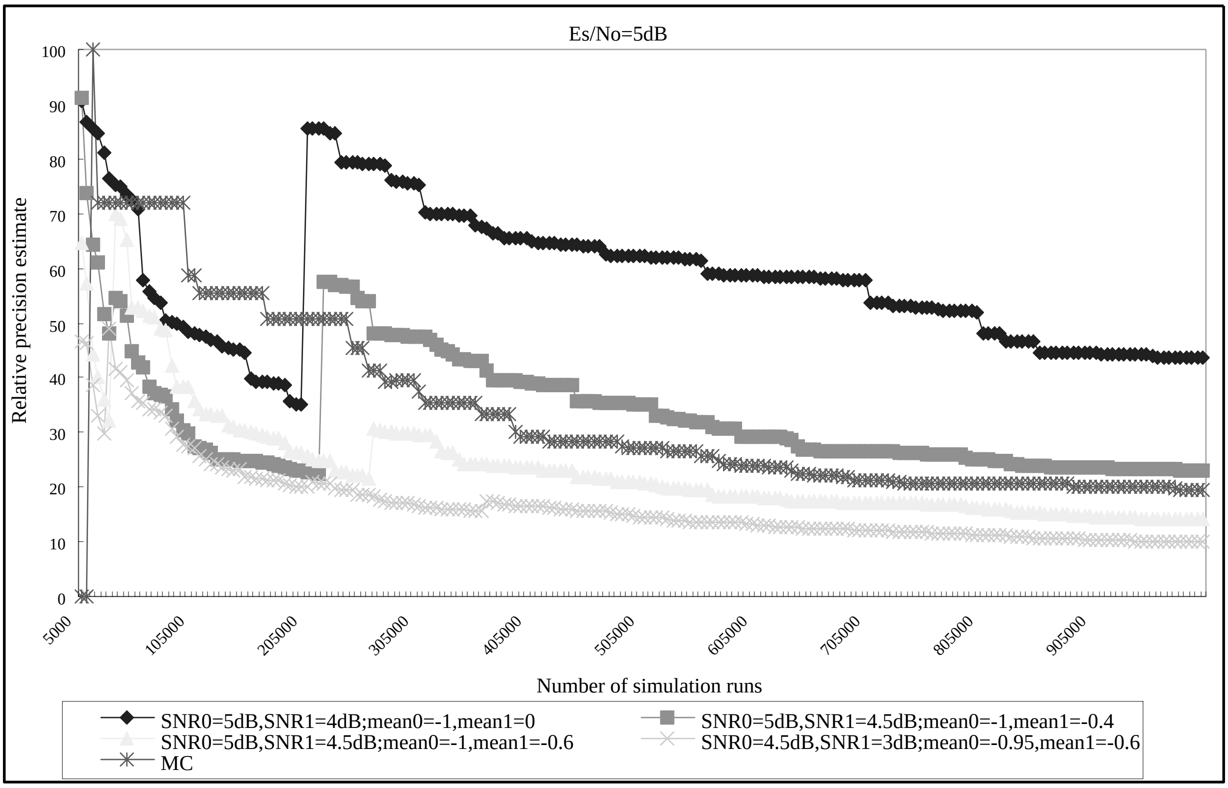

Figure 4.14: The RPE by IS simulations with biased non-stationary channels under $E_s/N_0 = 5$dB.

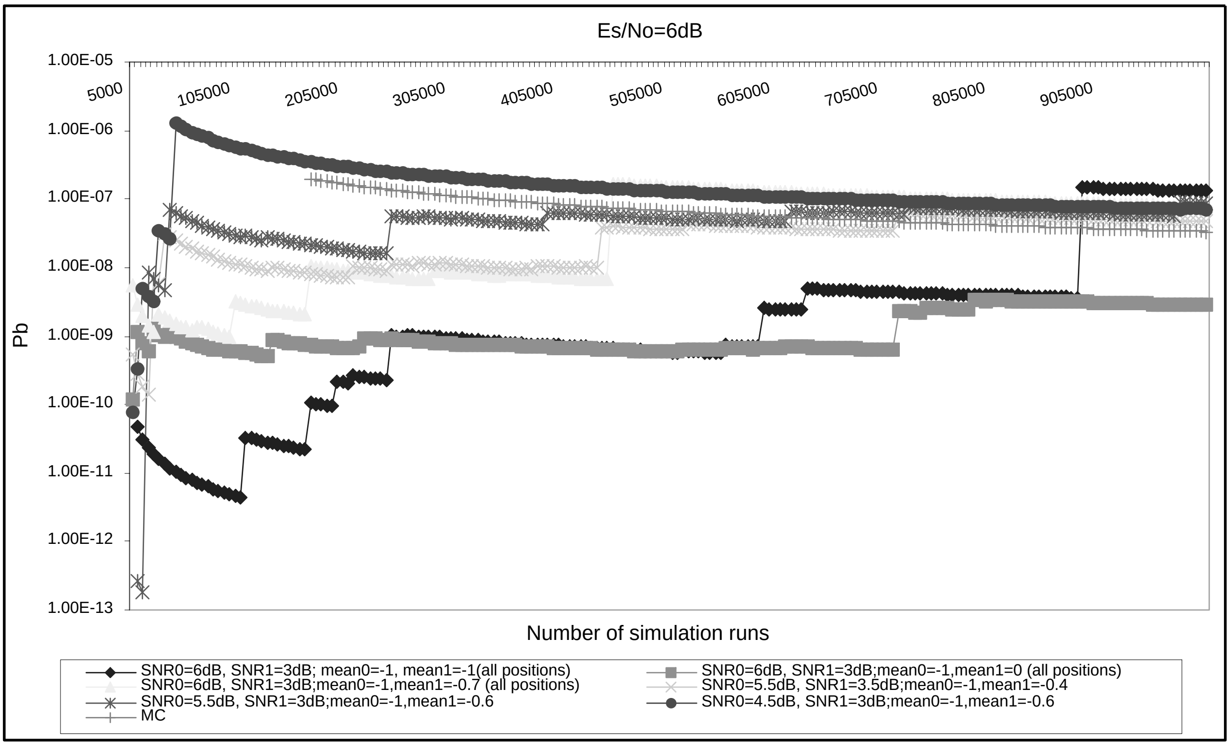

Figure 4.15: The BER estimate by IS simulations with biased non-stationary channels under $E_s/N_0 = 6\text{dB}$.

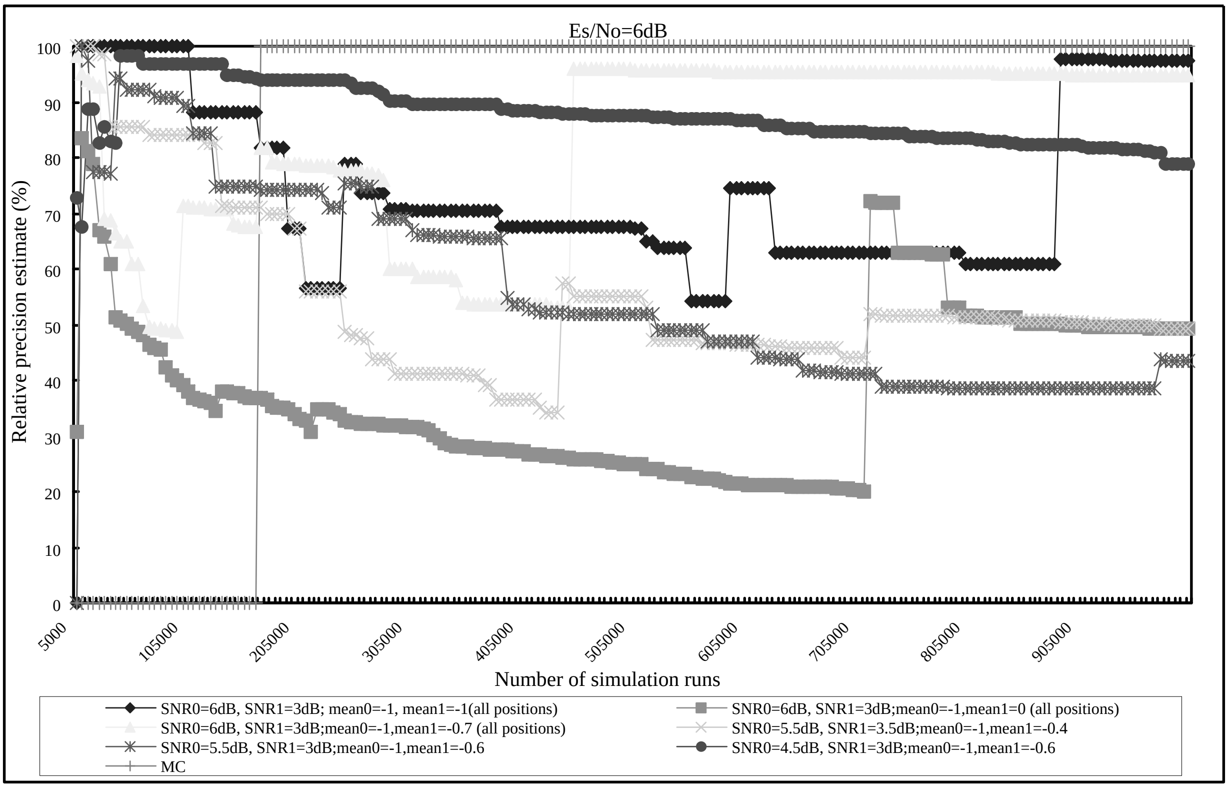

Figure 4.16: The RPE by IS simulations with biased non-stationary channels under $E_s/N_0 = 6$dB.

Es/No=5dB				
	pb	rpe	complexity	IS-complexity/MC-complexity
SNR0=4.5dB, SNR1=3dB; mean0=-0.95, mean1=-0.6	2.54E-06	9.99%	3.41E+08	0.59087876
MC	2.68E-06	9.98%	5.77E+08	1

Table 4.5: The list of IS estimates for biased stationary under $E_s/N_0 =$ 5dB (Only 3 out of 10 positions are mean-translated).

Es/No=6dB				
	pb	rpe	complexity	IS-complexity/MC-complexity
SNR0=5.5dB, SNR1=3dB; mean0=-0.95, mean1=-0.6	4.27E-08	10.00%	4.80E+09	0.213042397
MC	4.49E-08	9.97%	2.25E+10	1

Table 4.6: The list of IS estimates for biased stationary under $E_s/N_0 =$ 6dB (Only 3 out of 10 positions are mean-translated).

Chapter 5

Conclusions

As indicted in Chapter 3, the selection of the IS density determines the efficiency of the IS simulation. Worse results than the MC simulator may cause by an improperly chosen IS density. In spite of the unsuitability of the bad channel, an accurate estimate can still be obtained with huge extra computational efforts because the estimator is unbiased.

While the stationary IS density is well chosen, improvements are perceived only for E_s/N_0 is larger than 4dB. Further improvement is observed when E_s/N_0 is higher; therefore, we biased the channel to generate the major error patterns by a non-stationary IS density. The absence of those error events except for those corresponding to "free-distance" events does not affect the BER result severely when SNR is high. Also for the case of high SNR, non-stationary IS density only limitedly improves the efficiency. But if the SNR is further expanded, larger weight function values are rendered for the non-stationary IS density, and a better efficiency should be observable.

Finally, our simulations show that one should favor the selection of stationary IS density. This may always be the case for codes with large number of "free-distance" codewords.

Bibliography

[1] Yunghsiang S. Han and Po-Ning Chen, "Maximum-likelihood soft-decision sequential decoding algorithms for convolutional codes," presented at the recent results session of *the 1998 IEEE International Symposium on Information Theory*, Cambridge, MA, USA, August 1998.

[2] M. A. Herro and J. M. Nowack, "Simulated Viterbi decoding using importance sampling," *IEE Proceedings*, pp. 133–142, April 1988.

[3] John. S. Sadowsky, "A new method for Viterbi decoder simulation using importance sampling," *IEEE Trans. Commun.*, pp. 1341–1351, September 1990.

[4] Khaled Ben Letaief and Khurram Muhammad, "An efficient new technique for accurate bit error probability estimation of ZJ decoders," *IEEE Trans. Commun.*, pp. 2020–2027, June 1995.

[5] Michel C. Jeruchim, "Techniques for estimating the bit error rate in the simulation of digital communication systems," *IEEE J. Select. Areas Commun.*, pp. 153–170, January 1984.

[6] Peter J. Smith, Mansoor Shafi, and Hongsheng Gao "Quick simulation: A review of importance sampling techniques in communications systems," *IEEE J. Select. Areas Commun.*, pp. 597–613, May 1997.

[7] Khaled Ben Letaief and John S. Sadowsky, "New importance sampling methods for simulating sequential decoders," *IEEE Trans. Inform. Theory*, pp. 1716-1722, September 1993.

[8] M. Cedervall and R. Johannesson, "A fast algorithm for computing distance spectrum of comvolutional codes," *IEEE Trans. Inform. Theory*, pp. 1146–1159, November 1989.

[9] Dingqing Lu and Kung Yao, "Improved importance sampling technique for efficient simulation of digital communication systems," *IEEE J. Select. Areas Commun.*, pp. 67–75, January 1988.

[10] Peter M. Hahn and Michel C. Jeruchim, "Developments in the theory and application of importance sampling," *IEEE Trans. Commun.*, PP.706–714, July 1987.

[11] K. S. Shanmugam and P. Balaban, "A modified Monte-Carlo simulation technique for the evaluation of error rate in digital communication systems," *IEEE Trans. Commun.*, PP.1916–1924, November 1980.

[12] Bruce R. Davis, "An improved importance sampling method for digital communication system simulations," *IEEE Trans. Commun.*, pp. 715–719, July 1986.

[13] John S. Sadowsky, "On the optimality and stability of exponential twisting in Monte Carlo estimation," *IEEE Trans. Inform. Theory*, pp. 119–128, January 1993.

[14] John S. Sadowsky and James A. Bucklew, "On large deviation theory and asymptotically efficient Monte Carlo estimation," *IEEE Trans. Inform. Theory*, pp. 579–588, May 1990.

[15] J. G. Proakis, *Digital Communications*, New York: McGraw-Hill, 1995.

[16] S. Lin and Daniel J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1983.

[17] Stephen B. Wicker, *Error Control Systems for Digital Communication and Storage*, Englewood Cliffs, NJ: Prentice-Hall, Inc., 1995.

[18] Hong-Bin Wu, *The Maximum-Likelihood Soft-Decision Sequential Decoding Algorithms for Convolutional Codes*, Master Thesis, Dept. of Communications Eng., National Chiao Tung Univ., Taiwan, R.O.C., June 1999.

[19] Yunghsiang S. Han, Po-Ning Chen and Hong-Bin Wu, "A maximum-likelihood soft-decision sequential decoding algorithm for binary convolutional codes," to appear, *IEEE Trans. Commun.*, 2001.