Optimal Frame Synchronization for Continuous and Packet Data Transmission

Patrick Robertson

June 12, 1995

Contents

Chapter 1

Introduction

Frame synchronization is an essential component of all reliable digital communication systems, from high data rate computer links to satellite and deep space communications. Essentially, it can be formulated as the task of knowing where frames or packets of information begin, in other words the correct association at the receiver of the received symbols to blocks such as words, bytes or data-frames. For this reason, frame synchronization is often referred to as word, burst or packet synchronization. It is of little use to the information sink in a communications system if this association is even one symbol off; therefore, frame synchronization is very much a 'hit or miss' problem. Tasks falling into the category 'Frame synchronization' may be necessary at various levels in a data transmission system, depending on the organization of the transmitted data, see for example Fig. 1.1. On the other hand, some packet transmission schemes may only require *one* such stage, this being the determination of the location of the packet in a time-slot, see Fig. 1.2, for example. There is an inherent danger of confusion that results from the widespread use of such a generic term as frame synchronization; and since this work will treat both frame synchronization of data frames and packets, we shall refer to the former as the **traditional frame synchronization problem** and the latter as the **frame synchronization problem for packets**.

Over the last decades, a number of different measures have been introduced that allow a receiver to accomplish frame synchronization. These usually include both special measures at the transmitter, as well as a dedicated algorithm at the receiver. Such measures include the use of block-codes that can detect small shifts of the detected bits [CS88], or the very popular **marker concept** introduced by Barker [Bar53]. In the case of the latter we differentiate between the 'maximization' and 'threshold' techniques (see chapter 2). Optimal -in the sense of minimizing the synchronization failure probability- is the former, since it can be derived using the MAP (Maximum A-Posteriori) principle, based on a well defined problem definition [Mas72]. The models defining the problem have the advantages of simplicity and universality: we shall introduce the first one shortly. Based on these models we will derive optimal and close-to-optimal frame synchronization techniques and analyze their performance. It is in the

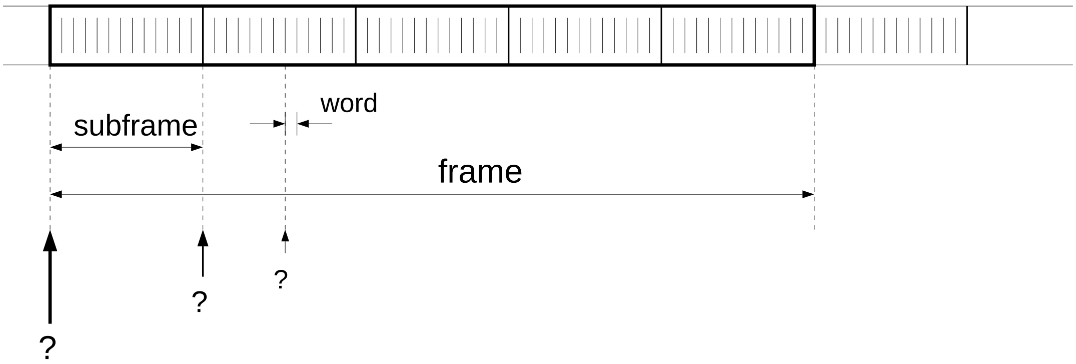


Figure 1.1: *Example of three different hierarchies in a digital transmission system that require some kind of frame synchronization. Shown are word, sub-frame and frame boundaries that might all require independent synchronization, although knowledge of frame synchronization might imply knowledge of sub-frame and word synchronization.*

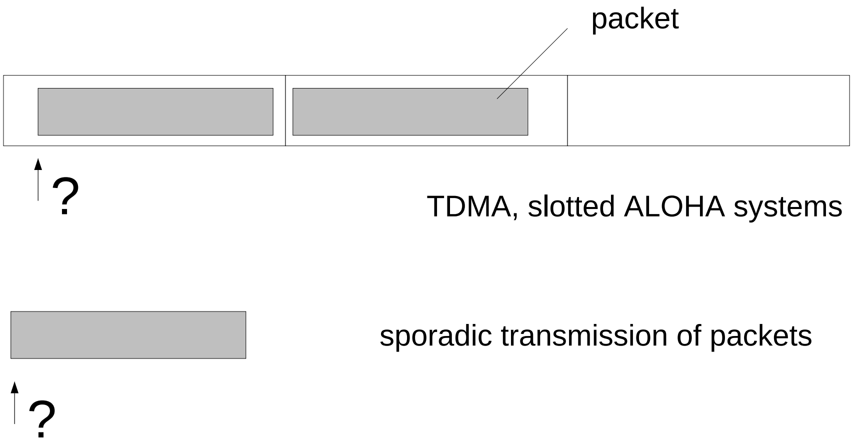


Figure 1.2: *Packet frame synchronization. Packets may either lie within time-slots, or arrive sporadically. In both cases reliable detection of the packet start is required at the receiver.*

spirit of Massey's pioneering work on frame synchronization [Mas72], that we shall proceed.

1.1 Overview

This work is organized into two parts. In the first we shall address the traditional frame synchronization problem, in the second frame synchronization for packets without preambles. We have chosen this order for several reasons. Firstly, the former is historically the older and more thoroughly covered by previous research, secondly the techniques developed in Part I can be extended to the more demanding frame synchronization task for packets.

In Part I, 'Traditional Frame Synchronization', we shall proceed as follows:

Chapter 2: After introducing the traditional frame synchronization problem based on the marker concept and maximization, we shall give an overview of the literature on this topic, available prior to the commencement of this work. This shall lay the foundations for the derivation of likelihood functions and performance analysis.

Chapter 3: The kernel of an optimal frame synchronization technique is the knowledge of the most suitable **likelihood function** that must be evaluated by the receiver for each possible frame starting position. These can be derived systematically for different channels, modulation formats and data encoding techniques based on the MAP principle, hence the synchronizers are optimal in the MAP sense (and minimize the synchronization failure probability). The MAP principle is used in many parameter estimation and detection problems where an observer has to make a decision on the (unknown) value(s) of a parameter(s) based on perturbed observations (direct or indirect) of the parameter(s). In many cases, this decision process can be reduced to the evaluation of a likelihood function for each possible value of the parameter(s), followed by maximum selection. We shall begin by re-deriving the likelihood function for phase coherent detection of linear modulation (QAM and PSK), and present suboptimal likelihood functions, in particular the virtually optimal high signal-to-noise ratio approximations. By suboptimal we mean likelihood functions which can be seen as simplifications of, or approximations to the optimal (in the MAP sense) likelihood function. This principle will be extended to demodulation with phase ambiguity and differential demodulation of binary phase shift keying (DBPSK). Subsequently, we shall address coded data instead of independent data comprising the frames, and introduce the list synchronizer and the synchronizer using trellis termination information. Finally, we shall show that if channel state information is available to a synchronizer in a receiver confronted with communications over a time-varying, frequency non-selective fading channel, then this can also be used in the optimal and close-to optimal likelihood function.

Chapter 4: One reason why Massey's optimal synchronizer has enjoyed relatively little practical use, is that hitherto, its performance analysis was limited to simulations. Only the grossly suboptimal **correlation rule** has yielded to an analysis so far -in the form of a union upper bound on the synchronization failure rate [LT87] of one synchronization attempt. After presenting an expression for the synchronization rate in the noiseless case [Nie73] we shall re-derive the union bound for the correlation rule in a Gaussian channel and later extend it to the high signal-to-noise ratio approximation of the optimal synchronizer. Subsequently, we shall investigate the performance of the list synchronizer (as far as possible) and after a brief section on the choice of suitable synchronization words (markers) we confirm our analysis by simulations. The latter will also be our tool to demonstrate the benefits of the more complex synchronizers that evade analytical description.

Chapter 5: We have already hinted at several points of interaction between coding and frame synchronization. However, the important question of whether to encode the synchronization word or not, has never been fully examined. We shall present a simple model to describe the two competing approaches (encode the synchronization word or not), and use it to provide a very simple tool to estimate the performance difference. The conclusion is that under certain circumstances, encoding the synchronization word may be beneficial.

Part II, 'Frame Synchronization for Preamble-less Packets', is somewhat more concise and presumes knowledge from Part I.

Chapter 6: This will give an introduction to packet communications in general, and more specifically to some applications of short packets transmitted over noisy and/or fluctuating radio channels that are particularly demanding as far as synchronization (including frame synchronization) are concerned.

Chapter 7: Here the foundations for the rest of the work will be laid. Two different packet receiver 'philosophies' will be compared, one being the more traditional approach, essentially treating packets as short 'long data sequences', requiring a preamble to achieve carrier and symbol timing recovery. The other, more recently proposed and developed, is the often called 'preamble-less' concept: the packet is first sampled, then stored and processed. The latter approach allows a host of 'holistic' processing techniques to be employed, such as FFT carrier recovery and one shot timing synchronization (for the latter we will present simulations confirming the suitability of the digital square and filter algorithm). Finally, we will describe a very simple model of a preamble-less packet within a time-slot for which we will develop the optimal synchronizers.

Chapter 8: In analogy to Part I, some optimal and close-to-optimal likelihood functions for the just presented model will be derived. They include coherent detection of linear modulation schemes (QAM and PSK) and DBPSK. Furthermore, these synchronizers can, in addition, be used as an indication of whether a packet had been sent or not. Finally, a simple realization possibility of such synchronizers is presented, making use of a recursive (and hence easily computable) definition of the likelihood function.

Chapter 9: Again we follow the structure of Part I and indulge in analytical performance evaluation. We shall limit our analysis to BPSK for reasons of complexity. Using our newly developed union bounds as an optimization criterion, we employ computer search to find new binary synchronization words suitable for packet frame synchronization. Simulation results will also form a major part of the chapter.

Chapter 10: A simple design formula based on simulations and analysis is given, which allows the length of the synchronization word to be chosen, given the signal-to-noise ratio and the required synchronization performance. Finally, two system examples - both for packet communications over severe channels- are taken to demonstrate the effectiveness of the newly derived algorithms.

Chapter 11: The main achievements will be summarized, followed by a list of further possible work and open questions on the topic, that are seen to be potentially fruitful.

Part I

Improvements for Traditional Frame Synchronization

Chapter 2

Problem Definition and State-of-the-Art

In this and the following chapters we will look into detail at the **traditional frame synchronization problem** as formulated in [Bar53] [Mas72] [Sch80], for example. **Frame synchronization** is defined as determining the start of each frame of data in a stream of continuously transmitted frames. It can, but need not, be achieved by inserting a known synchronization word or sequence -from now on: sync word, sometimes abbreviated SW- into the data to be transmitted. This technique is referred to as the **marker concept**, and we shall focus exclusively on it.

The receiver has two options to search for the sync word. The first is to perform a 'sliding' evaluation over discrete time μ, of a **likelihood function** $L(\mu)$ (or an approximation to the likelihood function, e.g. correlation with the sync word) and to compare the obtained value with a **threshold**. If it is exceeded, then that position is declared to be the start of the sync word (Fig. 2.1 a). The other method, which yields a lower synchronization failure rate because it can be designed to be optimal in the sense of minimizing the probability of incorrect synchronization, is to examine a sub-sequence of symbols that is as long as one frame and which will thus include the boundary between two frames, then to select that position in time μ that maximizes the likelihood function -we shall call this technique the **marker concept with maximum selection**. It is illustrated in Fig. 2.1 b and Fig. 2.2. The underlying problem is a typical example of a parameter detection problem where the unknown parameter is the frame starting position (see Section 3.1 for a brief introduction to detection theory).

We will treat only the latter approach in this work since it is more closely related to the 'one shot' synchronization philosophy for packet reception which we will come to later and because detection theory allows this optimal synchronizer to be derived systematically using the MAP principle [Mas72].

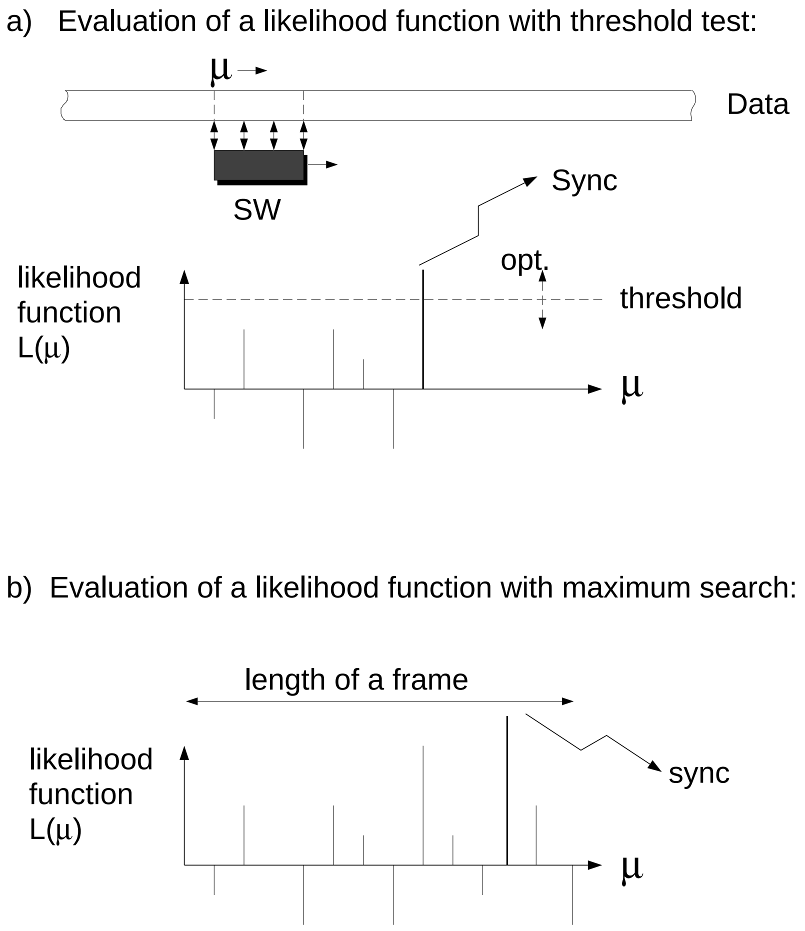

a) Evaluation of a likelihood function with threshold test:
μ
Data
SW
Sync
opt.
threshold
likelihood function L(μ)
μ
b) Evaluation of a likelihood function with maximum search:
length of a frame
likelihood function L(μ)
sync
μ

Figure 2.1: *Two principle methods of performing frame synchronization. Method a) relies on the comparison of a likelihood function with an optimized threshold and is simpler to implement than method b) which performs a maximum selection over an interval of the length of one frame, but which results in a lower synchronization failure rate and can be derived from first principles.*

In practice, the receiver may make its final decision for frame alignment based on one such observation and search, or it may extend the search over several frames [Mas72], finally making, for example, a **majority decision** [LT87] or by employing more elaborate techniques (describable through state diagrams) [KL84] [JAS85]. In either case, it is possible to restrict the analysis to the case where the receiver observes *one* sub-sequence of the length of a frame, by first defining the frame structure and channel model (this chapter), then deriving optimal and suboptimal likelihood functions to be evaluated by the receiver (chapter 3). Before we will outline previous work in the area of frame synchronization, we shall present the underlying frame structure based on the use of the marker concept and formulate the task of frame synchronization precisely.

2.1 Frame Synchronization Model

In the following, we will define the frame structure and transmitter/ receiver model up to, but not including, the frame synchronizer for the case of an AWGN channel, random data and coherent demodulation with perfect symbol timing. Later deviations from this simple model will be explained where they become necessary.

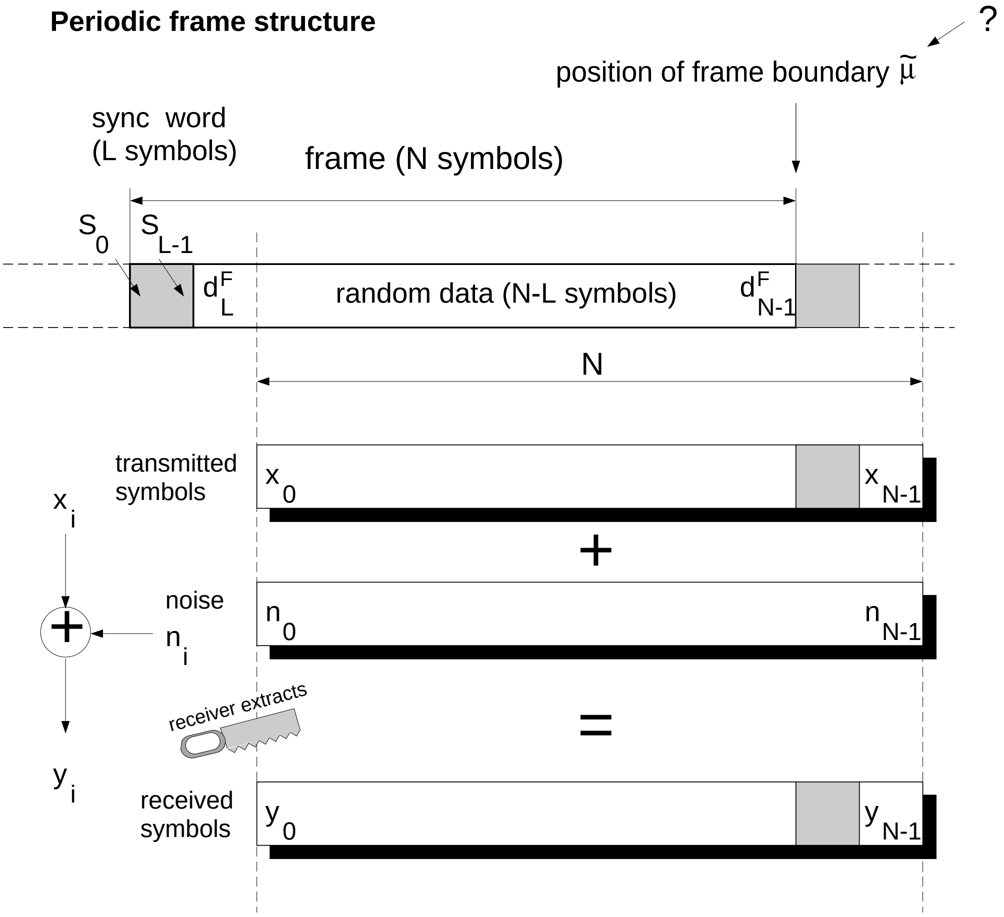


Figure 2.2: The model for traditional frame synchronization. The receiver extracts -at random- a sequence of symbols of length N; $\vec{y} = (y_0, y_1, ..., y_{N-1})$ from the data stream. $\vec{y}$ is the realization of the random variable $\vec{\mathbf{y}}$ (see text). The receiver has no knowledge of the transmitted sequence corresponding to the extracted sequence $\vec{x}$ or the noise sequence $\vec{n}$. The goal of the synchronizer is to determine the frame starting position $\tilde{\mu}$. Known to the receiver is the sync word $\vec{S} = (S_0, S_1, ..., S_{L-1})$ comprising the start of each frame.

- We have used an equivalent complex baseband model to describe the channel and modulation/demodulation, hence the transmitted and received symbols of all linear quadrature modulation schemes such as QAM and PSK, can be represented by complex numbers [Pro89].

- Consider the situation depicted in Fig. 2.2: **Frames** have a known **synchronization word** inserted at the beginning of each frame.

- The length of each frame is N symbols, and the underlying frame structure is periodic[1].

- The first L symbols are the known sync word $\vec{S} = (S_0, S_1, ..., S_{L-1})$.

- The remaining $N - L$ symbols are data, and are chosen randomly and independently from the M-ary symbol set $\{W_j, 1 \leq j \leq M\}$. We define the average symbol energy, E_s, as $E_s = \frac{1}{M} \sum_{j=1}^{M} ||W_j||^2$ assuming equally likely symbols.

- These constitute the **frame data sequence $\vec{\mathbf{d}}^{\mathbf{F}}$** the actual realization of which in any particular frame is the sequence $\vec{d}^F = (d^F_L, d^F_{L+1}, ..., d^F_{N-1})$.

- The sync word is assumed to be chosen from the same symbol set $\{W_j\}$; although a choice from a sub-set is often used for practical reasons.

- The transmitted symbols (data and sync word) are disturbed by **additive white Gaussian noise**: the real and imaginary components of the noise vectors are i.i.d. with zero mean and variance $N_0/2$, where N_0 is the one-sided power spectral density. The signal-to-noise ratio is defined as E_s/N_0.

- The receiver extracts an N-long, perfectly demodulated sequence of symbols $\vec{\mathbf{y}}$ -at random from the data stream- on which it will operate. The index of the symbol in the frame corresponding to $\mathbf{y_0}$ is the **true frame starting position** $\tilde{\mu}$.

- $\vec{\mathbf{y}}$ is a random variable, since it depends on the random data, random noise and random $\tilde{\mu}$. Let the *actual* value of $\vec{\mathbf{y}}$ in any one synchronization attempt be the sequence $\vec{y} = (y_0, y_1, ..., y_{N-1})$. In our model, $\vec{y}$ is constructed as follows: the transmitted sequence of symbols corresponding to the sequence $\vec{y}$ shall be called $\vec{x} = (x_0, x_1, ..., x_{N-1})$, it is added to the noise sequence $\vec{n} = (n_0, n_1, ..., n_{N-1})$ to yield the sequence $\vec{y}$ with:

$$y_i = x_i + n_i. \tag{2.1}$$

- We also define the **observation data sequence $\vec{\mathbf{d}}$**, the actual realization of which in any particular observation is the sequence $\vec{d} = (d_L, d_{L+1}, ..., d_{N-1})$ $= (x_{(\tilde{\mu}+L)\text{mod}N}, ..., x_{(\tilde{\mu}+N-1)\text{mod}N})$. The observation data sequence is the concatenation of two partial frame data sequences, unless $\tilde{\mu} = 0$ (in which case $\vec{d} = \vec{d}^F$.)

[1] this does not mean that the data is the same from frame to frame.

- Our synchronizer must generate an estimate of $\tilde{\mu}$: $\hat{\mu} \in \{0, 1, ..., N-1\}$; **synchronization is defined as being correct when $\tilde{\mu} = \hat{\mu}$.** The estimate shall be based on the evaluation of a likelihood function $L(\mu)$ (derived from the MAP principle) for all possible $\mu \in \{0, 1, ..., N-1\}$. We shall choose that $\hat{\mu}$ that maximizes the likelihood function.

2.2 State-of-the-Art

We shall now briefly outline important previous work on optimal and close-to optimal frame synchronization that has been carried out prior to the commencement of this work (1990). In Barker's pioneering work dating back to 1953 [Bar53], frame synchronization for the binary symmetric channel was proposed using the marker (or sync word) concept. Henceforth, this principle has been widely applied to a variety of situations. Several decades later, Stiffler recognized that the approach -at the receiver side- proposed by Barker (correlation of the data stream with the sync word) was not optimal, the random data part of the frame should somehow be used in the optimal synchronization rule; he concluded, however, that the analysis would be too involved [Sti71].

The breakthrough for optimal frame synchronization came in 1972 [Mas72]. Massey presented a relatively simple derivation of the optimal frame synchronizer for binary modulation for the frame structure and receiver model we have introduced in the previous section. Massey concluded that there exist three frame synchronizers of interest:

1. The optimal synchronizer (in the Maximum Likelihood sense).
2. A high signal-to-noise ratio approximation to 1.
3. A low signal-to-noise ratio approximation to 1.

We already have the formalism to express these three synchronizers. We would like to present them at this stage, and the reader is referred to section 3.2, for details on their derivation. Following the same order as above, our frame synchronizers should generate the value $\hat{\mu}$ of μ in $(0, 1, ..., N-1)$, which maximizes the following likelihood functions:

1. $L(\mu) = \sum\limits_{i=0}^{L-1} y_{i+\mu} \cdot S_i - \frac{N_0}{2} \sum\limits_{i=0}^{L-1} \ln\ \cosh(2\sqrt{E_s} y_{i+\mu}/N_0)$.
2. $L_H(\mu) = \sum\limits_{i=0}^{L-1} y_{i+\mu} \cdot S_i - \sum\limits_{i=0}^{L-1} |y_{i+\mu}|$.
3. $L_L(\mu) = \sum\limits_{i=0}^{L-1} y_{i+\mu} \cdot S_i - 1/N_0 \cdot \sum\limits_{i=0}^{L-1} {y_{i+\mu}}^2$.

The commonalities and differences are as follows; all three synchronizers share the first term in common, it is called the **correlation term** because the sync word is correlated with the received data stream. The optimal synchronizer modifies, or augments this correlation component by a correction term, which has its origin in the random data part of the frame. It is a non-linear function of each received symbol, summed over the length of the sync word. For an interpretation of the correction term, see Fig. 3.1 in section 3.2.4. For high signal-to-noise ratios the values of y_i/N_0 will generally be high, as $y_i = x_i + n_i$. The high signal-to-noise ratio approximation to the optimal synchronizer approximates the individual correction term by the absolute value of each symbol, whereas the low SNR synchronizer uses a true energy correction. The soft correlation rule, hitherto often thought optimal [Mas72], comprises just the first (correlation) term.

Simulations by Nielsen [Nie73] showed that only the first two rules above were of technical relevance; in fact, an astounding observation was that the high SNR rule that was designed for high SNR, performed well at low SNR also. 'Being optimistic' about the channel proved not to be a disadvantage at lower SNR [Mas93b]. The high SNR rules have become the synchronizers of practical interest throughout previous work and the work presented here. Nielsen was also the first to provide an analysis for the frame synchronization rate of the frame synchronizer using the maximization technique for a noise-less channel, he named his upper bound on the correct synchronization probability P_{RDL}, **RDL** stands for Random Data Limited; it takes into account the possibility of the sync word occurring in the data itself.

In a tutorial paper on the topic, Scholtz reviewed the literature up to 1980 [Sch80], but focussed mainly on the threshold algorithm. He gave some analytical expressions for the synchronization failure rate for this technique with a binary symmetric channel (BSC). Bi, in 1983, heuristically applied Massey's 'optimal' synchronizer to the threshold technique (instead of correlation) for Gaussian channels [Bi83]. It outperformed both the soft and hard correlation rules, and interestingly, Bi was the first to indicate that soft correlation was inferior (at high SNR) to hard correlation. He also applied the central limit theorem to approximate the performance of the threshold algorithm, although the method presented is cumbersome.

Lui and Tan derived optimal synchronization rules for various optical modulation schemes (OOK, PPM), and extended the random data limited bound P_{RDL} to higher order signaling formats [LT86]. One year later they extended Massey's work to coherent and non-coherent phase, M-ary signaling (in other words including PSK, QAM, FSK and PPM) [LT87]. For the correlation rule they derived a union lower bound on the synchronization rate but concluded that the optimal rule or high SNR approximation did not yield to analysis.

A lot of work has been done in the field of sync word choice, both for binary and polyphase modulation, for example [Bar53] [Sch80] [TS61] [GS65] [ZG90]. We have very briefly reviewed some of the more important work in section 4.4. Put very briefly, one has searched for sync words that minimize some function of the synchronization failure rate -usually by simple brute-force search. Essentially, this boils down to evaluating the partial auto-correlation function of the sync word, and finding those sequences with minimal partial auto-correlation.

There have also been some approaches for joint coding and frame synchronization. Chang and Sollenberger used cyclic block codes to correct small frame timing inaccuracies in TDMA systems [CS88]. However, such approaches are unsuitable if the time shifts are larger (e.g. acquisition) or when the SNR is very low so that channel errors occur (they used an error detection code). The question of whether to encode the sync word or not (and in the former case to search for it in the decoded data), has not been analyzed in depth. One systems is known where the sync word is in fact encoded, this is the NASA deep space standard [Con87], although this decision has been criticized by Paaske in [Paa90], on the grounds that optimal synchronization could only be achieved with the rules developed by Massey, since they require soft decisions. We shall address just this problem in chapter 5, in particular motivated by the recent availability of soft decisions at the decoder output, [HH89] [HR90].

It is the philosophy of optimal frame synchronization promulgated by Massey, Nielsen and Lui and Tan that is the main motivation for this work, and shall be the subject of Part I for the traditional frame synchronization problem. In Part II we shall address the problem of optimal frame synchronization for preamble-less packets and, for a review of the literature on this topic the reader is referred to chapter 7.

Chapter 3

Traditional Frame Synchronization - Derivation of Likelihood Functions

In the previous chapter we introduced the frame model and defined the problem to be solved. We will now derive the optimal synchronization strategy for several modulation schemes and data structures and also for the non-frequency selective fading channel. The underlying principle of these synchronizers is to evaluate a so-called likelihood function for each possible starting position of the sync word [Mas72]. The likelihood function is a measure for the probability that the true starting position of the sync word was at the corresponding position in the observation sequence. It can be derived in a systematic manner and can be approximated by simpler to evaluate sub-optimal likelihood functions.

Unfortunately, the correlation rule (using soft or hard decisions) is still widely implemented in situations where the optimal rule and simplifications thereof could provide improvement. This is, perhaps, due to the fact that although the signal-to-noise ratio gain over the correlation rule(s) amounts to several dB, the number of sync word symbols that can be saved is not that large. However, in systems where the sync word has been fixed in length, but synchronization performance is poor, the benefits of using the optimal or nearly optimal synchronizer can be significant. Furthermore, the 'soft' correlation rule can perform catastrophically under certain circumstances and we will later arrive at the interesting result that the 'hard' correlation rule actually *outperforms* the soft correlation rule contrary to widespread belief [Sch80] (this has hitherto only been pointed out in [Bi83]). One must also not forget that for terrestrial scenarios, bandwidth restrictions are becoming just as tight an issue as power constraints, well known from satellite communications, imposing strong bounds on the redundancy that system designers can spend on overhead such as the sync word. We will present synchronizers that if compared to the correlation rules show such an improvement that hopefully engineering practice will implement them.

Although we will later focus on synchronization of packets (Part II), the similarity of the two frame sync tasks is so great that a deeper understanding of the well-known traditional frame sync problem will help us. For this reason, we will briefly mention some of the differences between the two tasks at appropriate places, already in this chapter.

3.1 Short Introduction to Detection

Since this work is based on the derivation of synchronizers derived from the MAP principle, we shall briefly introduce the concepts **detection**, **estimation**, **Maximum A-Posteriori** and **Maximum Likelihood detection**. The problem of detection has been addressed first by Bayes well over two centuries ago [Bay64] and the reader is referred to [Tre68] for a detailed coverage of the wide topic.

Stated briefly in words, **detection** can be viewed as the task of *deciding on the cause of an output that is random in character, based on the observation of this output* -for instance the 'cause' can be a binary symbol transmitted over a channel, resulting in an observation after the channel. The detection may be binary (two possible causes) or N-ary, (N possible causes). In contrast, **estimation** requires the estimation of *the value of a parameter that has influenced our -random- observation.* The latter problem might be seen as the continuous extension of the former. Frame synchronization, discrete in nature, falls into the detection category, although we will sometimes use the word 'estimate' when referring to the discrete decision outcome, for linguistic reasons.

Estimation and detection theory are based on the concept of risk or cost minimization, the idea being that false decisions (or estimates deviating from the true value of the parameter) incur risks or costs. The goal is now to reduce the *average* risk, and the so-called Bayes test in a detection problem is a decision rule that minimizes the average risk. Because of its 'hit or miss' nature, frame synchronization can be treated as a uniform cost problem, which simply means that the cost of a false decision (frame synchronization failure) is the same regardless of the nature of the false decision. In this case, the detection problem can be called **Maximum A-Posteriori detection** (MAP detection), see [Tre68], page 57.

Let us formalize this at this stage, and present a one-dimensional detection problem where we have to decide on a discrete cause $\tilde{A}$. Our decision will be based on the observation (one or multi-dimensional) r which is a realization of a random variable (or vector) $\mathbf{r}$. The **MAP rule** can be written as:

- Select that A, that maximizes:

$$Pr\{\tilde{A} = A|\mathbf{r} = r\}. \tag{3.1}$$

In words, we should find the maximum of the probability of $\tilde{A}$ given that we observed r. Using the mixed Bayes rule [WJ65], we can write,

$$Pr\{\tilde{A}=A|\mathbf{r}=r\} = f_{\mathbf{r}}(r|\tilde{A}=A)\cdot\frac{Pr\{\tilde{A}=A\}}{f_{\mathbf{r}}(r)}, \tag{3.2}$$

and ignoring the term $f_{\mathbf{r}}(r)$ which will not affect our detection outcome, the MAP rule can be rewritten as

- Select that A, that maximizes:

$$f_{\mathbf{r}}(r|\tilde{A}=A)\cdot Pr\{\tilde{A}=A\}. \tag{3.3}$$

Now $Pr\{\tilde{A}=A\}$ is the a-priori probability of the event $\tilde{A}=A$. This a-priori knowledge affects our decision as well; not only the observation r. If all non-zero values of $Pr\{\tilde{A}=A\}$ are identical for all A, then the MAP rule is identical to the **Maximum Likelihood rule** (ML):

- Select that A, that maximizes:

$$f_{\mathbf{r}}(r|\tilde{A}=A). \tag{3.4}$$

Often, we are not able to directly express this PDF because some other unknown event(s) that are present in our model will also influence the observation $\mathbf{r}$. For example, we might only be able to determine the PDF

$$f_{\mathbf{r}}(r|\tilde{A}=A,\tilde{B}=B), \tag{3.5}$$

for all possible B. If we are interested in $\tilde{B}$ as well, then our search will become a two-dimensional one (over A and B). If, on the other hand, we are not interested in detecting $\tilde{B}$ at all, and $\tilde{A}$ and $\tilde{B}$ are statistically independent, one can simply integrate (or sum) over all possible B, yielding

$$f_{\mathbf{r}}(r|\tilde{A}=A) = \sum_{\forall B} f_{\mathbf{r}}(r|\tilde{A}=A,\tilde{B}=B)\cdot Pr\{\tilde{B}=B\}. \tag{3.6}$$

In a technical environment we are often only interested in the outcome of the maximization process, so it need not be necessary to maximize over the PDFs themselves, but instead over a strictly monotone increasing function of the PDF. For example, it is possible to take the logarithm of the PDF; the resulting function of A is often referred to as a **log-likelihood function**. We shall adopt the simple notation of calling any monotone increasing function of the PDF a **likelihood function**, and it shall be denoted by $L(A)$. Here and in the future we explicitly assume the dependence of $L(A)$ on r. Since we will also encounter approximations to a true likelihood function (approximations in the sense that they are no longer strictly monotone increasing functions of the PDF), we shall use indices with $L(A)$ to denote this fact. However, we have also yielded to the popular temptation of calling them likelihood functions, despite the fact that they are only approximations to the true likelihood function. Let us now address the frame synchronization problem.

3.2 Likelihood Functions for Uncoded, Coherently Demodulated Frames Transmitted over an AWGN Channel

3.2.1 The Optimal Likelihood Function

We shall now replace $\tilde{A}$ by the true frame starting position $\tilde{\mu}$, and the observation $\mathbf{r}$ by the observation vector $\vec{\mathbf{y}}$. The MAP approach requires that a frame synchronizer must choose $\hat{\mu}$ as that μ that maximizes the conditional probability [Mas72]

$$Pr\{\tilde{\mu} = \mu | \vec{\mathbf{y}} = \vec{y}\}.$$

Remember that $\vec{\mathbf{y}}$ is an N dimensional random vector, the realization of which is $\vec{y}$, where $y_i = x_i + n_i$. Assuming that all $\tilde{\mu}$ are equally likely, this corresponds to the Maximum Likelihood (ML) rule where we maximize $f_{\vec{\mathbf{y}}}(\vec{y}|\mu)$. Clearly we can expand this PDF and write,

$$f_{\vec{\mathbf{y}}}(\vec{y}|\mu) = \sum_{\forall\, \vec{d}} f_{\vec{\mathbf{y}}}(\vec{y}|\mu, \vec{d}) \cdot Pr\{\vec{d}\}. \tag{3.7}$$

Splitting this into two individual products, we obtain

$$f_{\vec{\mathbf{y}}}(\vec{y}|\mu) = \frac{1}{(\pi N_0)^N} \prod_{i=0}^{L-1} e^{\frac{-||y_{i+\mu} - S_i||^2}{N_0}} \cdot \sum_{\forall\, \vec{d}} Pr\{\vec{d}\} \cdot \prod_{i=L}^{N-1} e^{-\frac{||y_{i+\mu} - d_i||^2}{N_0}}, \tag{3.8}$$

where $||x||$ is the Euclidean norm of x, because the noise samples are Gaussian distributed and i.i.d. The sync word symbols are S_i, the observation data symbols are d_i. The first sum takes into account the Euclidean distance between the sync word and the portion of the sequence $\vec{y}$ where is is expected to occur, had $\tilde{\mu}$ been equal to μ; the second takes into account the distance between any observation data sequence $\vec{d}$ and the corresponding symbols in $\vec{y}$. Note that addition of indices is modulo N because of the periodic frame structure. Since we wish to maximize this conditional probability over μ, we can equivalently maximize a likelihood function that must be a strictly monotone increasing function of $f_{\vec{\mathbf{y}}}(\vec{y}|\mu)$. Leaving out irrelevant factors and by making use of the periodic frame structure we obtain

$$L_1(\mu) = \prod_{i=0}^{L-1} e^{\frac{2}{N_0}\langle y_{i+\mu}, S_i\rangle} \cdot \sum_{\forall\, \vec{d}} Pr\{\vec{d}\} \cdot \prod_{i=L}^{N-1} e^{\frac{2}{N_0}\langle y_{i+\mu}, d_i\rangle - \frac{\|d_i\|^2}{N_0}}, \tag{3.9}$$

in which $\langle , \rangle$ denotes the inner product: $\langle a, b\rangle = \mathrm{Re}\{a\}\mathrm{Re}\{b\} + \mathrm{Im}\{a\}\mathrm{Im}\{b\}$. In appendix B it has been shown that

$$\sum_{\forall\, \vec{d}} \prod_{i=0}^{N-1} e^{\frac{2}{N_0}\left[\langle y_{i+\mu}, d_i\rangle - \frac{\|d_i\|^2}{2}\right]} = \prod_{i=0}^{N-1} \sum_{j=1}^{M} e^{\frac{2}{N_0}\left[\langle y_{i+\mu}, W_j\rangle - \frac{\|W_j\|^2}{2}\right]}, \tag{3.10}$$

where W_j are the elements of the M-ary symbol set. At this stage we make an important assumption: the data sequences are all equally likely, so we can pull the a-priori probability of $\vec{d}$, $Pr\{\vec{d}\}$, in front of the sum $\sum_{\forall\, \vec{d}}$. This allows us to divide $L_1(\mu)$ in (3.9) by the sum (3.10) and arrive at the likelihood function (ML rule in [LT87]):

$$L(\mu) = \sum_{i=0}^{L-1} \langle y_{i+\mu}, S_i\rangle - \frac{N_0}{2} \sum_{i=0}^{L-1} \ln \sum_{j=1}^{M} e^{\frac{2}{N_0}\left[\langle y_{i+\mu}, W_j\rangle - \frac{\|W_j\|^2}{2}\right]}. \tag{3.11}$$

For binary signalling where $W_i = \pm\sqrt{E_s}$ and $S_i \in \{\pm\sqrt{E_s}\}$, this reduces to Massey's result:

$$L(\mu) = \sum_{i=0}^{L-1} y_{i+\mu} \cdot S_i - \frac{N_0}{2} \sum_{i=0}^{L-1} \ln\ \cosh(2\sqrt{E_s} y_{i+\mu}/N_0). \tag{3.12}$$

The receiver will make the best choice for $\hat{\mu}$ by choosing that μ that maximizes the likelihood function (3.11). Evaluation of just the first term in (3.11) corresponds to the soft correlation rule. The second term is a 'correction term' [Mas72] that actually takes into account the random data following the sync word, *although the data portion of $\vec{y}$ is not explicitly used to evaluate* (3.11) *for one* μ, this is due to the division of $L_1(\mu)$ by (3.10). In the sequel we will gain insights into the meaning and significance of this correction term.

3.2.2 High Signal-to-Noise Ratio Approximation of the Optimal Likelihood Function

Evaluation of (3.11) requires knowledge of N_0 (implying estimation of the signal-to-noise ratio of the received signal) and needs either many complex arithmetic operations or a digital memory that stores pre-calculated values for the correction term for, perhaps, 16 possible values of y_i, and this for several N_0. However, a useful approximation to (3.11) can be derived for high SNR that needs no value for N_0. In [LT87] an approximation to the ML rule is given by

$$L_H(\mu) = \sum_{i=0}^{L-1} \left(\langle y_{i+\mu}, S_i - W_{\hat{\jmath}} \rangle + \frac{||W_{\hat{\jmath}}||^2}{2} \right), \tag{3.13}$$

where $\hat{\jmath}$ is that j which maximizes $\langle y_{i+\mu}, W_j \rangle - \frac{||W_j||^2}{2}$. This can easily be explained by inspecting the correction term of the optimal likelihood function (3.11): the second sum will be dominated by one value of j, here denoted by $\hat{\jmath}$. The correction term we encountered in the optimal likelihood functions (3.11) and (3.12) is still there, now it is simply the subtraction of $W_{\hat{\jmath}}$ in the inner-product term. For PSK signalling, all values $||W_j||^2$ are equal to the symbol energy E_s and the term can be omitted. For BPSK in particular we obtain the very simple result

$$L_H(\mu) = \sum_{i=0}^{L-1} \left(y_{i+\mu} \cdot S_i - \sqrt{E_s} |y_{i+\mu}| \right), \tag{3.14}$$

previously given by Massey in [Mas72].

3.2.3 Correlation Rules

The correlation rule is defined as just the first term of (3.11):

$$L_C(\mu) = \sum_{i=0}^{L-1} \langle y_{i+\mu}, S_i \rangle. \tag{3.15}$$

The correlation rule is 'soft' or 'hard' depending on the quantization of the y_i. Assuming no quantization we are dealing with the soft correlation rule, hard decision at the demodulator output yields the hard correlation rule.

3.2.4 Interpretation

Although we will treat the benefit of using the ML rule (or the high SNR approximation) over the correlation rules in more detail in the next chapter, we would still like to provide the reader with at least an intuitive reason or an example for why and when the ML rule outperforms the soft correlation rule.

Consider the case depicted in Fig. 3.1. The value of the correlation term at any position μ is determined by the similarity of the data sub-sequence of length L beginning at μ with the real sync word and the noise samples affecting those symbols. In our example, the first graph shows the value of the soft correlation; the noise was such that the energy at the receiver (and hence correlation term) of the sync word (starting at $\tilde{\mu}$) was low. On the other hand, a part of the frame data sequence of the frame starting at μ', differed from the true sync word in only one symbol, but the noise made the correlation higher because the energy of the received sub-sequence was higher. The soft correlation term would lead to synchronization failure here, but the optimal likelihood function would take account of the energy of the symbols and result in correct synchronization. The hard correlation rule's performance actually lies in between the soft correlation and optimal rules, as we will see later.

3.3 Derivations for Differentially Encoded Phase Modulation and Differentially Coherent Detection

So far, a very important modulation and demodulation technique has not been investigated as far as optimal frame synchronization is concerned: differentially coherent phase modulation [Pro89] [LS73]. Information is not represented by the phase of each symbol, but by the phase difference between two successive symbols. For example, in binary differentially encoded PSK (often abbreviated DBPSK or 2DPSK), a 1 might be represented by a phase transition of 180^o; and no phase change implying a -1 having been transmitted. The receiver can now demodulate a successive stream of symbols coherently, the differential encoding making a systematic 180^o phase ambiguity on the demodulated symbols without effect. Alternatively, decisions on the transmitted phase difference and hence information are made directly by observing *two successive symbols*, then subtracting their phases to estimate the corresponding information symbol. In the latter case no phase coherent demodulation is necessary, even a small frequency offset between receiver and transmitter oscillators can be tolerated. The advantages of not requiring a separate phase synchronizer has made DPSK a very attractive scheme in a wide number of applications, especially when phase synchronization is difficult, e.g. packet communication or on mobile channels [SD87] [Neu89] [WRL$^+$91].

Differential encoding (sometimes called differential modulation) can be defined as follows: the transmitted sequence corresponding to the N-long sequence extracted by the receiver is $\vec{x}^D$, where[1]

$$x_i^D = x_{i-1}^D \cdot x_i / \sqrt{E_s}. \tag{3.16}$$

Hence we have differentially encoded the sync word symbols and also the observation data symbols. Now, at the receiver side, we receive $\vec{y}^D = (y_0^D, ..., y_{N-1}^D)$ where

$$y_i^D = x_{i-1}^D + n_i. \tag{3.17}$$

[1] The division through $\sqrt{E_s}$ is necessary in order to ensure that the dimension of x^D is always $\sqrt{\text{dim(Energy)}}$.

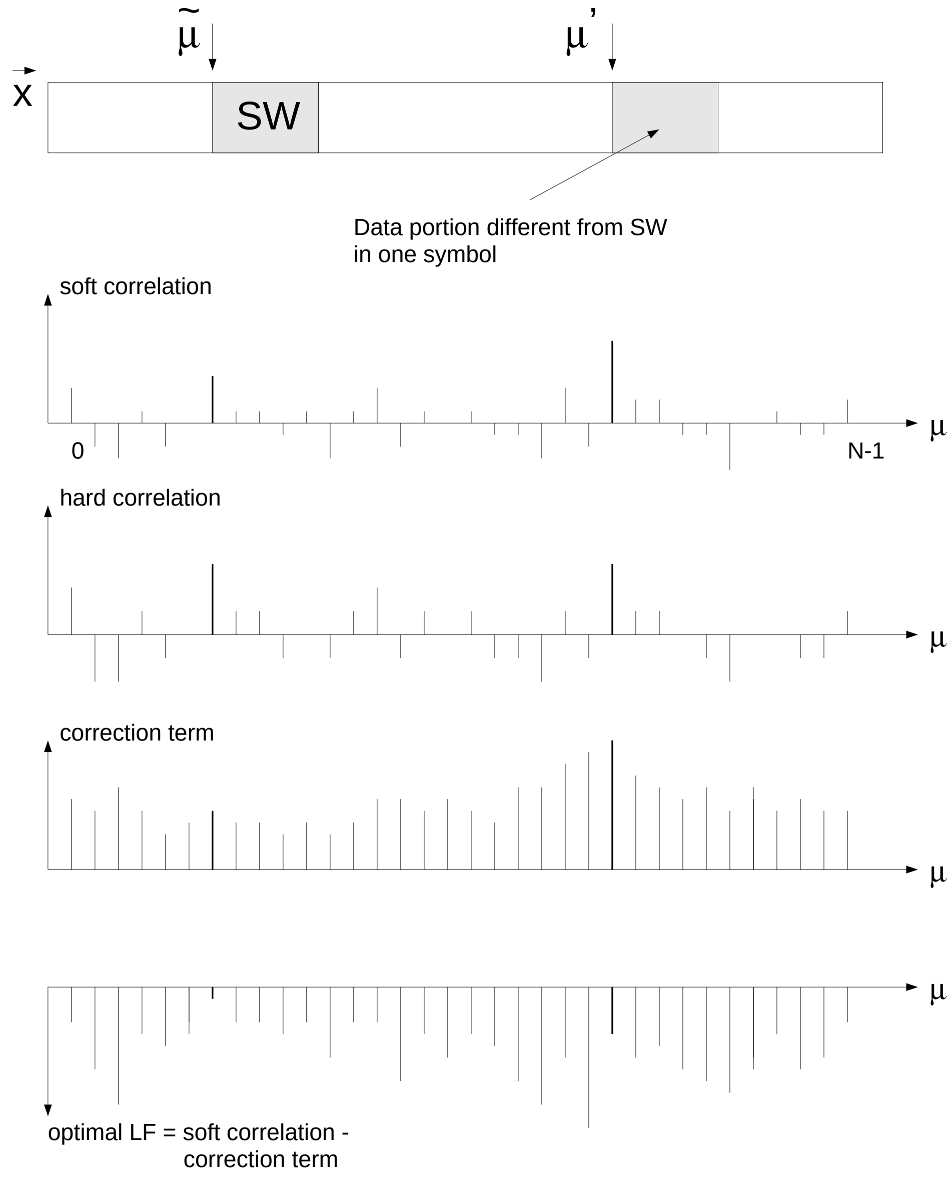


Figure 3.1: *Interpretation of the ML and correlation rules. We compare two 'competing' frame starting positions, $\tilde{\mu}$ (real start of sync word) and μ' (beginning of data sub-sequence similar to sync word). The soft correlation terms (top graph) are affected by noise so that μ' has a higher value than $\tilde{\mu}$, because the energy of some of the L received symbols following μ' was higher than those following $\tilde{\mu}$. The soft correlation term augmented by the correction term (third graph) leads to the ML rule (last graph). Interestingly, the hard correlation rule is on average worse than the ML rule (or high SNR approximation) but outperforms the soft correlation rule at high SNR.*

We now need to modify our definition of $\vec{y}$, our decision vector. As before, the synchronizer will operate only on this sequence. The choice for the calculation of $\vec{y}$ must be such that one is able to determine it without the need of phase synchronization. The decision variable normally chosen for binary DPSK is:

$$y_i = 2\mathrm{Re}\{y_i^D (y_{i-1}^D)^*\}, \tag{3.18}$$

($()^*$ denotes the complex conjugate). In the sequel, it is important to bear this modification of the definition of $\vec{y}$ in mind. It is clear that an identical phase error affecting both y_i^D and y_{i-1}^D does not influence y_i.

We must be careful later when working with $\vec{y}$: y_{-1}^D and hence y_0 has not been defined. To overcome this difficulty, one can make the observation interval $N+1$ symbols long, making y_{-1}^D available. The vector $\vec{y}$, however, is still of length N.

But how do we optimally perform frame synchronization if the data is differentially encoded and we assume differential detection?

In [Neu89], Neul derived the maximum likelihood metric for Viterbi decoding of differentially detected sequences. It was necessary to derive the conditional probability density function of the decision variable y_i, given that either a phase change ($x_i = -\sqrt{E_s}$) or no phase change ($x_i = +\sqrt{E_s}$) had been transmitted. For the AWGN channel the following result is obtained:

$$f_{\mathbf{y}}(y_i| + \sqrt{E_s}) = \begin{cases} A(y_i)Q(\sqrt{E_s/N_0}, 2\sqrt{y_i}/N_0) & \text{for} \quad y_i > 0 \\ A(y_i) & \text{for} \quad y_i \leq 0 \end{cases} \tag{3.19}$$

$$f_{\mathbf{y}}(y_i| - \sqrt{E_s}) = f_{\mathbf{y}}(-y_i| + \sqrt{E_s}), \tag{3.20}$$

(see [Neu89], (5.12); for AWGN), where $A(D) = \frac{1}{2N_0}\exp(-E_s/N_0 + D/N_0)$, E_s is the symbol energy and $Q(a,b)$ is the Marcum Q-function:

$$Q(a,b) = \int_b^\infty x \cdot e^{\frac{-(x^2+a^2)}{2}} \cdot I_0(ax) \cdot dx, \tag{3.21}$$

[Par80]. With [Neu89], (5.12) it can be shown that $f_{\mathbf{y}}(y_i| \pm \sqrt{E_s})$ can be approximated to

$$f_{\mathbf{y}}(y_i| \pm \sqrt{E_s}) \approx \frac{1}{2N_0} e^{-E_s/N_0 \pm y_i/N_0}. \tag{3.22}$$

Let us now return to the frame synchronization problem where we can write $f_{\vec{\mathbf{y}}}(\vec{y}|\mu)$ in terms of the products of $f_{\mathbf{y}}(y_i| \pm \sqrt{E_s})$ (neglecting the statistical dependence of the y_i):

$$f_{\vec{\mathbf{y}}}(\vec{y}|\mu) = \prod_{i=0}^{L-1} f_{\mathbf{y}}(y_{i+\mu}|S_i) \cdot \sum_{\forall\, \vec{d}} Pr\{\vec{d}\} \cdot \prod_{i=L}^{N-1} f_{\mathbf{y}}(y_{i+\mu}|d_i). \tag{3.23}$$

Inserting (3.22) into (3.23) and following a similar derivation as before leads to the close-to optimal likelihood function

$$L_A(\mu) = \sum_{i=0}^{L-1} \frac{1}{2\sqrt{E_s}} \cdot S_i \cdot y_{i+\mu} - \frac{N_0}{2} \sum_{i=0}^{L-1} \ln \sum_{j=1}^{2} \exp\left[\frac{1}{N_0\sqrt{E_s}} W_j y_{i+\mu}\right]. \tag{3.24}$$

As before, we can approximate (3.24) by a high SNR rule

$$L_H(\mu) = \sum_{i=0}^{L-1} \left(\frac{1}{\sqrt{E_s}} S_i \cdot y_{i+\mu} - |y_{i+\mu}|\right), \tag{3.25}$$

the even simpler soft correlation rule comprising just the first term. Equation (3.25) corresponds exactly to the binary coherent case (except for the $\sqrt{E_s}$ normalization term).

3.4 Synchronizers for Demodulation with Phase Ambiguity

So far, we have assumed coherent detection or differentially coherent detection when determining the optimal likelihood functions. However, as illustrated in [Mas72] it is important to treat the case of a remaining phase ambiguity after phase recovery and coherent detection, because one will often resolve this ambiguity by using the sync word itself (see 3.4.1) or by making use of properties of the error protection coding [MS90]. The phase ambiguity after demodulation is a result of the *symmetry* of the signaling constellation. For example, for BPSK there is an ambiguity of π radians, since the phase recovery unit may lock onto one of two possible phases.

Massey gave the result for BPSK and we will now extend the principle to other signalling formats. We presume that there are M_a possible phase references remaining after carrier recovery and demodulation; for PSK modulation M_a will be equal to the number of phase points in the constellation, i.e. M. For QAM modulation schemes M_a will generally be 4 -due to the rectangular structure of the QAM constellation. We denote the possible discrete phase errors due to the ambiguity by $\phi_a(k) = k \cdot 2 \cdot \pi / M_a$, with $k \in \{0, ..., M_a - 1\}$. Our one dimensional detection problem has now been extended to a two dimensional one: we have to estimate $\tilde{\phi}_a$, the actual phase error, as well as $\tilde{\mu}$. In order to eliminate one parameter, we shall integrate over all possible ϕ_a (just as we have done with the observation data sequences $\vec{d}$), then determine our estimate of $\tilde{\mu}$. Derivation of the likelihood functions is straightforward, we will proceed as before with the ML approach

$$f_{\vec{\mathbf{y}}}(\vec{y}|\mu) = \sum_{k=0}^{M_a-1} \sum_{\forall\, \vec{d}} f_{\vec{\mathbf{y}}}(\vec{y}|\mu, \vec{d}, \phi_a(k)) \cdot Pr\{\vec{d}\} \cdot Pr\{\phi_a(k)\}. \tag{3.26}$$

The probability of any phase error ϕ_a is $\frac{1}{M\,a}$, we assume them to be equally likely, hence

$$f_{\vec{\mathbf{y}}}(\vec{y}|\mu) = \frac{1}{M_a \cdot (\pi N_0)^N} \sum_{k=0}^{M_a-1} \prod_{i=0}^{L-1} e^{\frac{-||e^{-j\phi a(k)} \cdot y_{i+\mu} - S_i||^2}{N_0}} \cdot$$

$$\sum_{\forall\, \vec{d}} Pr\{\vec{d}\} \cdot \prod_{i=L}^{N-1} e^{-\frac{||e^{-j\phi a(k)} \cdot y_{i+\mu} - d_i||^2}{N_0}}. \qquad (3.27)$$

It is easy to show that

1. we can pull the sum $\sum\limits_{\forall\, \vec{d}}$ in front of the first sum (because the data is random), and

2. if the value of $\sum\limits_{i=0}^{L-1} \langle e^{-j\phi_a(k)} \cdot y_{i+\mu}, S_i \rangle$ (the correlation term) is large for only one value of k then the first exponent will be dominated by this one k.

The high SNR rule can, therefore, be written as

$$L_H(\mu) = \sum_{i=0}^{L-1} \langle e^{-j \cdot \hat{k} \cdot 2 \cdot \pi / M} \cdot y_{i+\mu}, S_i - W_{\hat{j}(i+\mu)} \rangle + \frac{||W_{\hat{j}}||^2}{2}, \qquad (3.28)$$

where $\hat{k}$ is that k that maximizes $\sum\limits_{i=0}^{L-1} \langle e^{-j\phi_a(k)} \cdot y_{i+\mu}, S_i \rangle$. For binary PSK this simply becomes ([Mas72]),

$$L_H(\mu) = \left| \sum_{i=0}^{L-1} y_{i+\mu} \cdot S_i \right| - \sum_{i=0}^{L-1} |y_{i+\mu}| \cdot \sqrt{E_s}. \qquad (3.29)$$

In other words, before the correlation term is added to the correction term we must take the absolute value of the former, which is intuitively pleasing -the correction term on the other hand, is independent of the phase ambiguity if the data is random (premise 1 above).

3.4.1 Resolving the Phase Ambiguity

A technique often employed in order to resolve the phase ambiguity is to simply store the value $\hat{k}$ above. This can be justified because the ML detector for ϕ_a should maximize

$$f_{\vec{\mathbf{y}}}(\vec{y}|k) = \sum_{\forall\, \vec{d}} f_{\vec{\mathbf{y}}}(\vec{y}|\tilde{\mu}, \vec{d}, \phi_a(k)) \cdot Pr\{\vec{d}\}, \qquad (3.30)$$

which becomes,

$$f_{\vec{\mathbf{y}}}(\vec{y}|k) = \frac{1}{M^{N-L} \cdot (\pi N_0)^N} \prod_{i=0}^{L-1} e^{\frac{-||e^{-j\phi a(k)} \cdot y_{i+\tilde{\mu}} - S_i||^2}{N_0}} \cdot \sum_{\forall\, \vec{d}} \prod_{i=L}^{N-1} e^{-\frac{||e^{-j\phi a(k)} \cdot y_{i+\tilde{\mu}} - d_i||^2}{N_0}}. \qquad (3.31)$$

We have postulated that the synchronizer (3.28) can be seen as an detector for $\tilde{\phi}_a$ *if* $\hat{\mu}$ *is correct*, where $\hat{k}$ is chosen so that it maximizes

$$\sum_{i=0}^{L-1} \langle e^{-j\phi_a(k)} \cdot y_{i+\tilde{\mu}}, S_i \rangle. \tag{3.32}$$

But because the data term of (3.31) is not a function of ϕ_a, it is irrelevant, hence the likelihood function based on (3.31) deteriorates to (3.32). Q.E.D.

3.5 The Case where Data is Coded and the List Synchronizer

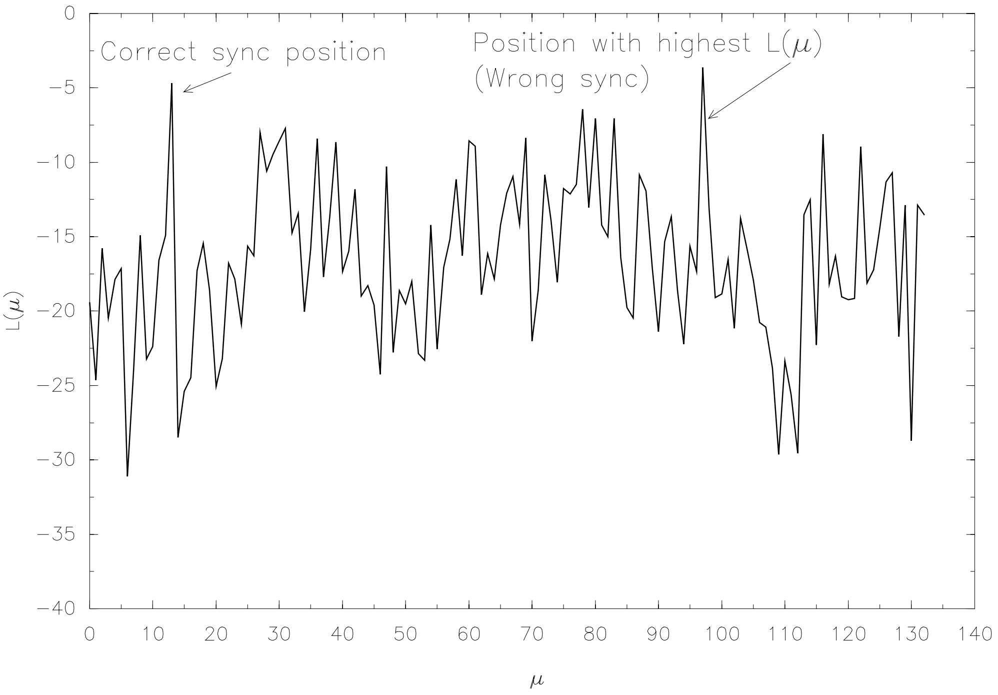


Figure 3.2: *Motivation for the list synchronizer: The Value of the likelihood function as a function of* μ*.* $\tilde{\mu} = 13$*,* $\hat{\mu} = 97$*. Note that* $L(\tilde{\mu})$ *is almost as large as* $L(\hat{\mu})$*. Example of a simulation with BPSK,* $N = 133$*,* $L = 13$ *and a SNR of* -2 *dB.*

Looking at Fig. 3.2, one can see the values of $L(\mu)$ obtained in a simulation. In this special case $\tilde{\mu} \neq \hat{\mu}$, i.e. sync was not achieved. The true sync position had a likelihood function not much lower than the largest, in fact it was the second largest. This situation is quite typical for a synchronization failure event. By nature, traditional frame synchronization is a hit or miss problem, there is nothing in-between false and correct sync. We thus propose that it would make sense to make available say, the two or three most likely positions to the next stage in the receiver. This can only be an advantage if the next stage has some way of eliminating those positions which are wrong, which could be done by a decoding stage following the frame sync unit. There is, of course, an overhead introduced through several processings (e.g. decodings) becoming necessary at times. If this technique can provide improved performance, then why is evaluation of the optimal likelihood function not sufficient to find the value of $\tilde{\mu}$ that will most often be correct, since we have called it the 'optimal' synchronization rule?

In 3.2 we assumed that all observation data sequences $\vec{d}$ are equally likely. This allowed us to eliminate the sum over $\vec{d}$, and construct a computable correction term. However, if the data is coded[2] then the data symbols are *not* independently chosen from the symbol set and hence not all sequences are equally likely (some might never occur). Let us look again at the last term in (3.8) which has to be evaluated for each $\vec{d}$,

$$Pr\{\vec{d}\} \cdot \prod_{i=L}^{N-1} \exp(-\frac{||y_{i+\mu} - d_i||^2}{N_0}). \tag{3.33}$$

The situation is somewhat complicated, because the observation data sequence of $\vec{x}$, the sequence $\vec{d}$, is usually composed of the concatenation of the frame data sequences $\vec{d^F}$ from *two* frames (only if $\tilde{\mu} = 0$, is $\vec{d^F} = \vec{d}$.) If the data symbols are chosen independently, this is of no importance, but now there will be frame data sequences $\vec{d^F}$ which the coder (or source) can produce (or are more likely), and others which it will not (or which are less likely). We are now faced with the dilemma of determining *how* this extends to the observation data sequence of $\vec{x}$ -the transmitted sequence corresponding to our observed sequence $\vec{y}$. This will depend very much on the nature of the dependencies. Fortunately, this problem does not apply to the 'one shot' synchronization case of packets (see chapter 8), where we consider just one packet being transmitted.

Let us assume for a moment that $\tilde{\mu} = 0$, or the packet synchronization case, i.e. $\vec{d^F} = \vec{d}$. Furthermore, the data be coded, such that some frame data sequences (=observation data sequences) occur (='legal' $\vec{d}$), and others can not occur. Evaluating (3.33) for all $\vec{d}$ will give very small values when the expected data portion of $\vec{y}$, $(y_{L+\mu}, ..., y_{N-1+\mu})$, lies far away (large Euclidean distance) from any legal $\vec{d}$. Conversely, if $(y_{L+\mu}, ..., y_{N-1+\mu})$ is close to any $\vec{d}$, then (3.33) will be larger. Note that the sequence $(y_{L+\mu}, ..., y_{N-1+\mu})$ is a function of μ and that the value of (3.33) will depend greatly on μ. Choosing that $\vec{d}$, given $\vec{y}$ and $\mu = \tilde{\mu}$ is exactly what is performed by a 'correctly synchronized' MAP decoder. The influence of (3.33) in (3.8) can be very much larger than that of the first term (corresponding to the sync word). An optimal receiver would evaluate (3.8) -i.e. also decode $(y_{L+\mu}, ..., y_{N-1+\mu})$- for all $\vec{d}$ *and* μ. Frame synchronization and decoding has to be seen as a *joint detection* problem. The number of decodings necessary, however, is equal to N. In fact, if the structure of the code is such that for only *one* μ and *one* $\vec{d}$ will (3.33) become large, then the sync word will no longer be needed!

[2] Statistical independence of the data symbols may also be due to the nature of the transmitted information - digitally encoded audio or video signals for instance.

A way to reduce the number of decodings is the following: a small number $\nu << N$ of likely frame starting positions be selected using the likelihood function derived *assuming equally likely data sequences.* The final choice is made by evaluating (3.33) (or (3.8)) just for these ν positions. In order to necessitate only evaluation of (3.8), the decoder must accept only one data sequence and one μ as correct (i.e. error detection). We will see later that with suitable error detection coding this scheme improves frame synchronization performance considerably, because the probability that the correct position $\tilde{\mu}$ is contained in the ν-list provided by the simple, one dimensional frame synchronizer is high even for ν as small as two or three. The relationship between the optimal synchronizer for the coded case, and the suboptimal realization using a list has been illustrated in Fig. 3.3.

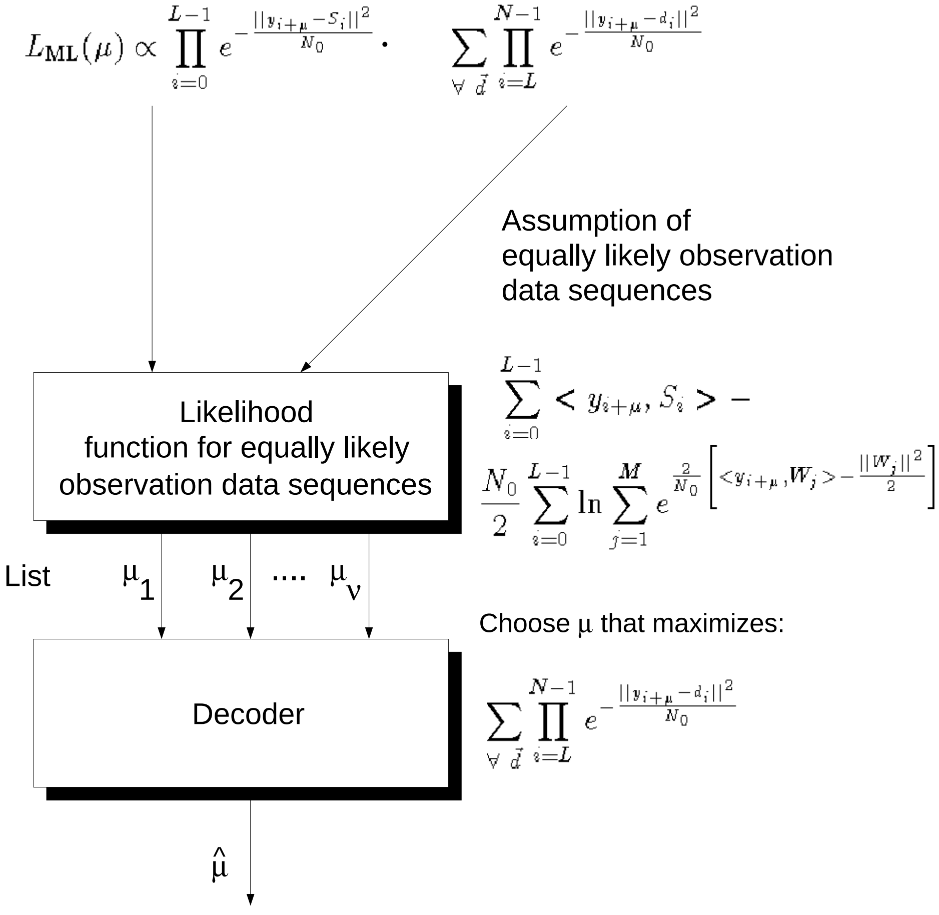

$L_{\mathrm{ML}}(\mu) \propto \prod_{i=0}^{L-1} e^{-\frac{\|y_{i+\mu}-S_i\|^2}{N_0}} \cdot \sum_{\forall \vec{d}} \prod_{i=L}^{N-1} e^{-\frac{\|y_{i+\mu}-d_i\|^2}{N_0}}$
Assumption of equally likely observation data sequences
Likelihood function for equally likely observation data sequences
$\sum_{i=0}^{L-1} < y_{i+\mu}, S_i > -$
$\frac{N_0}{2} \sum_{i=0}^{L-1} \ln \sum_{j=1}^{M} e^{\frac{2}{N_0}\left[<y_{i+\mu},W_j> - \frac{\|W_j\|^2}{2}\right]}$
List
μ_1
μ_2
....
μ_ν
Choose μ that maximizes:
Decoder
$\sum_{\forall \vec{d}} \prod_{i=L}^{N-1} e^{-\frac{\|y_{i+\mu}-d_i\|^2}{N_0}}$
$\hat{\mu}$

Figure 3.3: Comparison of the optimal likelihood function for the coded data case and the likelihood function for random data. Our list synchronizer can be interpreted as an approximation of the synchronizer that should evaluate the top line for all μ. Note that we have assumed dominance of (3.33) in (3.8). Otherwise the "decoder" must evaluate (3.8) completely.

Since we have intuitively motivated our list synchronizer by employing perhaps a decoder to select that frame starting position that leads to frames being decode-able, we shall get around the dilemma of $\vec{d^F} \neq \vec{d}$, in practice, by assuming that the decoder that does the final selection of $\hat{\mu}$, has available a complete frame and not just a subsection of two frames enclosed in the N-symbol observation available to the synchronizer.

Using a decoder to select $\hat{\mu}$ is not the only application of the list synchronizer. The likelihood functions derived so far are rather straightforward, or can be approximated well by simple likelihood functions (high SNR approximations). However, we will come across likelihood functions whose evaluation require some computational effort, for instance those proposed in the next section. It is not necessary to calculate these likelihood functions for each μ, but for only a small subset (ν-list); this list can be generated by a simpler, suboptimal likelihood function, see Fig. 3.4. As we will see later, it is even possible to cascade list synchronizers, starting with a simple synchronizer, the last stage being a decoder.

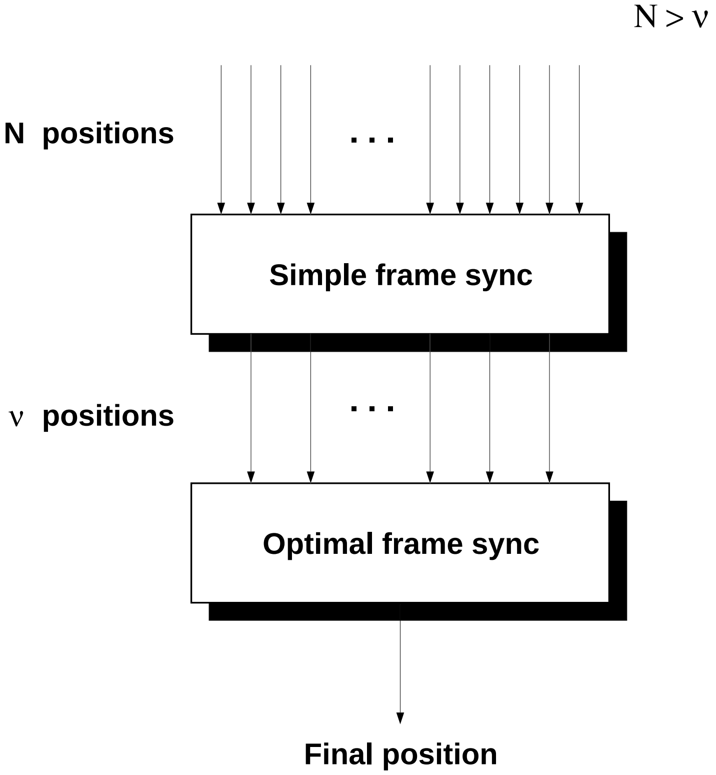


Figure 3.4: *Principle of concatenating frame synchronizers. Here the first synchronizer outputs a list of length ν of positions -the result of evaluating a simple likelihood function. The more optimal synchronizer chooses the best position from this list.*

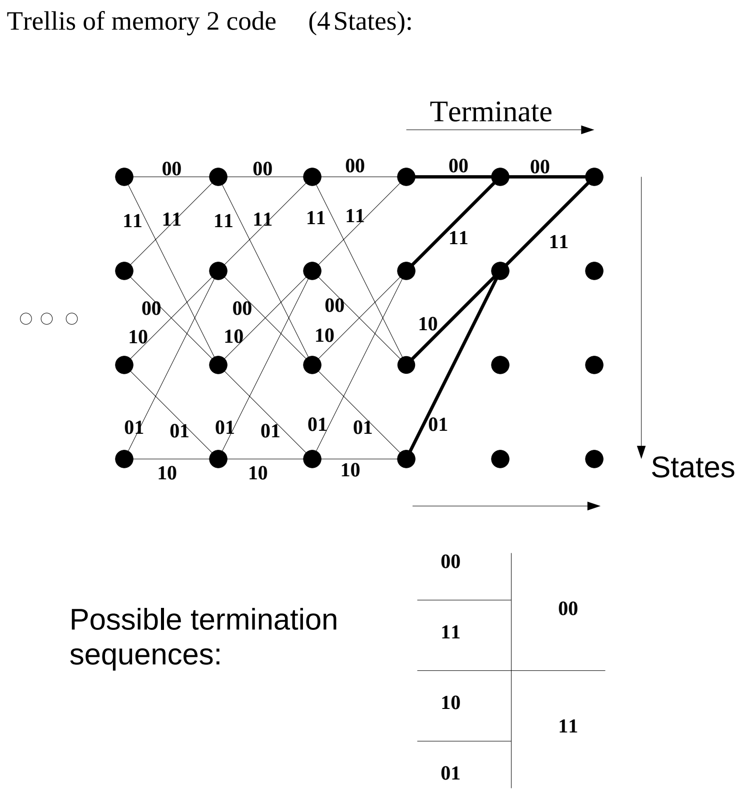
A similar technique has been employed in [SS89] by cascading decoders: The first decoder is the Viterbi algorithm that outputs not just one sequence of symbols (the ML sequence), but several likely sequences to some subsequent stage -for instance a source decoder, or error detection decoder. This is in contrast to the more sophisticated "Soft Output" Viterbi algorithm (SOVA) that provides reliability information for *each symbol* individually [HH89] [HR90] [NS93]. Returning to our frame sync problem, there is unfortunately little point in supplying the value of $L(\hat{\mu})$ -the reliability- with every decision, because of the difficulty of making use of such information since we are usually only interested in a single $\hat{\mu}$, not a sequence thereof. Nevertheless, the list synchronizer is a way of 'softening' the output of the otherwise incomparably 'hard' frame synchronization process [Rob92a].

Let us complete the discussion of the list synchronizer at this point by making a few remarks about the implementation thereof. Basically, the only modification to a normal frame synchronizer is the replacement of the maximum searching unit (to find $\hat{\mu}$) by a unit that finds the ν largest $L(\mu)$'s. These can be kept as a sorted list that is continuously updated as the $L(\mu)$'s are evaluated. If the stage following the list synchronizer is able to immediately detect a correct sync (e.g. error detection decoding), then the synchronizer can start by supplying μ_1 -corresponding to the highest $L(\mu)$- to the decoder. In the case of a sync failure, the next position μ_2 is output (corresponding to the second highest $L(\mu)$), and so on; until μ_ν. If this finally leads to no correct sync, then a sync failure must be declared. ν is the largest number of times the next stage has to be activated.

3.6 Synchronization of Terminated Convolutionally Encoded Sequences

In the previous section we looked at the case where data is encoded, i.e. the symbols of the observation data sequence $\vec{d}$ are no longer independent. A special kind of coding for error protection that is often used in environments where the signal-to-noise ratio is low, is convolutional coding [Pro89] [VO79]. A convolutional code can be described through its trellis; a trellis is simply the representation of the different states that an encoder can be in (the encoder can be represented as a finite state machine) but re-drawing all the states after each transition. An example of a four-state trellis is given in Fig. 3.5. Only certain transitions are allowed, and for simplicity we will restrict ourselves to two transitions per state in this introduction. The number of states is equal to 2^m where m is the memory of the encoder, here $m = 2$. In our example the code rate is 1/2, since two output bits are generated for every input bit. The transitions are labeled with the output of the encoder $\in \{0, 1\}$; a path going up represents a 0 as input bit, it going down means a 1 was at the input. The output is modulated and transmitted across the channel. In the following derivations we shall treat codes of rate k/n, i.e. for k information bits, n bit are actually transmitted. One popular way to increase the rate other than increasing k, is by puncturing the output of a low rate coder: certain symbols are simply not transmitted -an advantage is that the same decoder can be used as for the lower rate; an example of such a class of codes are the Rate Compatible Punctured Convolutional codes described in [Hag88]. The pattern of deletion is deterministic and must be known to the receiver which will replace the deleted symbols by zero (equivalent to an erasure) before decoding. At the receiver, one can use the Viterbi algorithm to find the most likely sequence of input bits of the encoder [For73].

When using convolutional codes to protect relatively short data sequences the effective code rate is reduced through the use of terminating bits which force the encoder into a certain state (usually the all-zero state) at the end of the frame data sequence. The number of data bits needed for this is equal to the memory of the code. Not transmitting the corresponding output bits of the coder without loss of decoding performance can only be achieved through tail-biting which increases decoding complexity quite considerably [MW86]. In this section we suggest a way of indirectly using these 'obsolete' symbols for frame synchronization [Rob93]. Let us look at Fig. 3.5 which shows the termination of a 4-state trellis. We observe that the possible output sequences corresponding to the two terminating bits are: 00 00, 11 00, 10 11, or 01 11. If we were to look at a long stream of frames of such encoded data, we would be able to make a good estimate at where the tail of the trellis is located, since only one of the above four sequences would occur at the end of each frame. A similar argument holds for the starting symbols of the trellis, assuming that the trellis starts in a known state. We could thus try to make use of this knowledge to aid frame synchronization, our sync word becoming effectively longer.


Trellis of memory 2 code (4 States):
Terminate
00
11
10
01
States
Possible termination sequences:

00	00
11	
10	11
01	

Figure 3.5: *Termination of a four state trellis. Below we see the set of possible termination sequences at the output of the encoder.*

In the following, we derive the optimal likelihood function (for the AWGN channel) that uses both a known sync word and the starting/terminating information. We also give some suboptimal algorithms which are easier to implement and reduce computational complexity.

3.6.1 The Likelihood Function

The derivation is fundamentally similar to the uncoded coherent case discussed earlier. The $N-L$ data symbols (*not* including the sync word!) are convolutionally encoded (and terminated) information. The memory of the rate k/n code is m, thus the last $n \cdot m$ symbols (of $N-L$) are required for trellis termination. The frame structure is shown in Fig. 3.6 (in practice, the position of the sync word and starting/terminating portions are arbitrary).

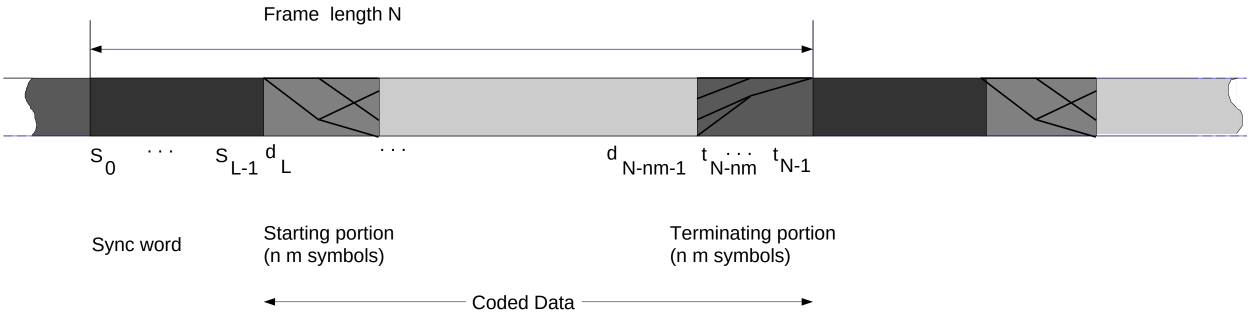

Frame length N
s0 . . . sL-1 dL . . . dN-nm-1 tN-nm . . . tN-1
Sync word
Starting portion (n m symbols)
Terminating portion (n m symbols)
Coded Data

Figure 3.6: Structure of trellis encoded and terminated frames.

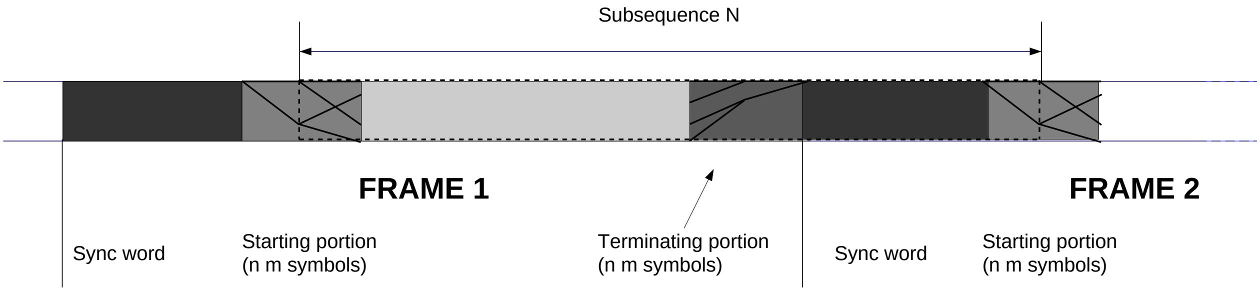

Subsequence N
FRAME 1
FRAME 2
Sync word
Starting portion (n m symbols)
Terminating portion (n m symbols)
Sync word
Starting portion (n m symbols)

Figure 3.7: Situation where trellis termination aided synchronization can degrade: Our receiver has chosen $\vec{y}$ to cut through a trellis starting portion of a frame.

Because the frames are presumed to be transmitted in a continuous stream, the observed sequence contains one complete synchronization word and *at least* either a legal complete termination portion or a legal complete starting portion of the trellis. Unfortunately, it is possible that the receiver has chosen a sequence $\vec{y}$ whose termination or starting trellis-portion does not completely belong to the same frame by 'cutting' through either of the former, see Fig. 3.7. This effect will degrade the performance slightly, and has been taken into account in the simulations presented later. It will not apply, however, when performing synchronization of a single packet, since then a complete packet is available to the receiver. For simplicity, our derivation will only take into account the trellis termination; trellis opening can be dealt with in exactly the same way. For $f_{\mathbf{y}}(\vec{y}|\mu)$ we can write

$$f_{\mathbf{y}}(\vec{y}|\mu) = \frac{1}{(\pi N_0)^N} \sum_{\forall\ \vec{d}} Pr\{\vec{d}\} \cdot \left(\prod_{i=0}^{L-1} e^{-\frac{||y_{i+\mu}-S_i||^2}{N_0}} \cdot \prod_{i=L}^{N-nm-1} e^{-\frac{||y_{i+\mu}-d_i||^2}{N_0}} \cdot \prod_{i=N-nm}^{N-1} e^{-\frac{||y_{i+\mu}-t_i||^2}{N_0}} \right). \tag{3.34}$$

Note that we have chosen to split the observation data sequence into two sub-sequences, the latter nm long sub-sequence corresponding to those symbols in the termination portion of the observation data sequence of $\vec{x}$. The symbol $t_i = d_{N-nm+i}$, is the i-th terminating symbol (modulator output) belonging to the terminating sequence (of length $n \cdot m$); t_i is a function of the frame data sequence $\vec{d^F}$ of the frame to which the terminating sequence $\vec{t}$ belongs. If we ignore the dependency of $\vec{t}$ on that part of the corresponding frame data sequence $\vec{d^F}$ preceding it, then we can rewrite

$$f_{\mathbf{y}}(\vec{y}|\mu) \approx \frac{1}{2^m \cdot (\pi N_0)^N} \prod_{i=0}^{L-1} e^{-\frac{||y_{i+\mu}-S_i||^2}{N_0}} \cdot \sum_{g=0}^{2^m-1} \prod_{i=N-nm}^{N-1} e^{-\frac{||y_{i+\mu}-t_i(g)||^2}{N_0}} \cdot \sum_{\forall\ \vec{d}} Pr\{\vec{d}\} \cdot \prod_{i=L}^{N-nm-1} e^{-\frac{||y_{i+\mu}-d_i||^2}{N_0}}. \tag{3.35}$$

There are 2^m possible terminating sequences. $t_i(g)$ is now the i-th terminating symbol (modulator output) belonging to the g-th possible terminating sequence which we have labeled arbitrarily from 0 to $2^m - 1$ and which we have assumed equally likely. In our example see Table 3.1.

In the second term in (3.35) we sum the conditional probabilities corresponding to all of the 2^m possible terminating sequences. To find $\hat{\mu}$ we can equivalently maximize the approximate likelihood function

$$L_{A1}(\mu) = \prod_{i=0}^{L-1} e^{\frac{2}{N_0}\langle y_{i+\mu},S_i\rangle} \cdot \sum_{g=0}^{2^m-1} \prod_{i=N-nm}^{N-1} e^{\frac{2}{N_0}\langle y_{i+\mu},t_i(g)\rangle-\frac{||t_i(g)||^2}{N_0}} \cdot \sum_{\forall\ \vec{d}} Pr\{\vec{d}\} \cdot \prod_{i=L}^{N-nm-1} e^{\frac{2}{N_0}\langle y_{i\mu},d_i\rangle-\frac{||d_i||^2}{N_0}}, \tag{3.36}$$

If the data were uncoded and all observation data sequences were equally likely, then we could proceed as in section 3.2. However, since the data is coded, we would ideally have to

Table 3.1: *Values of termination sequences (taken at the modulator output)* $\vec{t}(g)$ *for our example of a four state convolutional code with octal generators 5 and 7. There are four such termination sequences.*

g	$\vec{t}(g)/\sqrt{E_s}$
0	$-1\ -1\ -1\ -1$
1	$+1\ +1\ -1\ -1$
2	$+1\ -1\ +1\ +1$
3	$-1\ +1\ +1\ +1$

decode the data N times as discussed in the forgoing section[3], implying huge computational complexity (list synchronizer with $\nu = N$). For this reason, we assume equally likely $\vec{d}$ and using similar arguments as before, we finally obtain the suboptimal likelihood function:

$$L_A(\mu) = \sum_{i=0}^{L-1}\langle y_{i+\mu}, S_i\rangle + \frac{N_0}{2}\ln\sum_{g=0}^{2^m-1}\prod_{i=N-nm}^{N-1} e^{\frac{2}{N_0}\langle y_{i+\mu},t_i(g)\rangle-\frac{||t_i(g)||^2}{N_0}} -$$
$$\frac{N_0}{2}\sum_{i=0}^{L-1}\ln\sum_{j=1}^{M} e^{\frac{2}{N_0}\left[\langle y_{i+\mu},W_j\rangle-\frac{||W_j||^2}{2}\right]} - \frac{N_0}{2}\sum_{i=N-nm}^{N-1}\ln\sum_{j=1}^{M} e^{\frac{2}{N_0}\left[\langle y_{i+\mu},W_j\rangle-\frac{||W_j||^2}{2}\right]}. \qquad (3.37)$$

The first and third terms are the correlation and correction terms for the sync word and correspond to the sync word in the frame, the second and fourth terms take into account the trellis termination. Similarly, consideration of the starting portion of the trellis introduces another two sums.

3.6.2 Simplifications of Likelihood Functions

To simplify the suboptimal likelihood function, we can make the following approximation if the terminating sequence $\{t_{N-nm}(\hat{g}), ..., t_{N-1}(\hat{g})\}$ is close to $\{y_{N-nm+\mu}, ..., y_{N-1+\mu}\}$ for any one $\hat{g}$:

$$\frac{N_0}{2}\ln\sum_{g=0}^{2^m-1}\prod_{i=N-nm}^{N-1} e^{\frac{2}{N_0}\langle y_{i+\mu},t_i(g)\rangle-\frac{||t_i(g)||^2}{N_0}} \approx \sum_{i=N-nm}^{N-1}\langle y_{i+\mu}, t_i(\hat{g})\rangle - \frac{||t_i(\hat{g})||^2}{2}, \qquad (3.38)$$

and hence $\hat{g}$ is that g $(0 \le g < 2^m)$ that maximizes $\sum_{i=N-nm}^{N-1}\left[\langle y_{i+\mu}, t_i(g)\rangle - \frac{||t_i(g)||^2}{2}\right]$. To avoid evaluation of all 2^m correlation terms in our search for $\hat{g}$, we can make use of the tree structure of the terminating portion of the trellis. It is sufficient to use a tree searching algorithm that

[3] Also the separation of the termination sequence from the foregoing frame data sequence would be unnecessary.

calculates the value of the best sub-correlation at each trellis node by making a decision at each node; this reduces the computational complexity; in section 3.6.4 this process is explained in more detail. Applying the high signal-to-noise ratio approximation of before to the correction terms, together with the approximation (3.38), leads to a high SNR approximation of (3.37),

$$L_H(\mu) = \sum_{i=0}^{L-1} \left[\langle y_{i+\mu}, S_i - W_{\hat{j}} \rangle + \frac{||W_{\hat{j}}||^2}{2} \right] + \\ \sum_{i=N-nm}^{N-1} \left[\langle y_{i+\mu}, t_i(\hat{g}) - W_{\hat{j}} \rangle + \frac{||W_{\hat{j}}||^2 - ||t_i(\hat{g})||^2}{2} \right], \tag{3.39}$$

The term $||t_i(\hat{g})||^2$ can be omitted if $\sum_{i=N-nm}^{N-1} ||t_i(g)||^2$ are equal for all g. $||W_j||^2$ and $||t_i(\hat{g})||^2$ can be omitted when all $||W_j||^2$ are equal (e.g. PSK signaling). The selection of $\hat{g}$, i.e. the selection of the largest correlation to a termination sequence has been illustrated in Fig. 3.8. Intuitively satisfying is the fact that the process boils down to two stages, where first the most likely 'sync word' is selected from the 2^m possible ones, then it is treated exactly in the same way as an extension of the the conventional sync word $\vec{S}$.

We can also simplify the suboptimal likelihood function further and give a correlation rule (soft or hard)

$$L_C(\mu) = \sum_{i=0}^{L-1} \langle y_{i+\mu}, S_i \rangle + \sum_{i=N-nm}^{N-1} \left[\langle y_{i+\mu}, t_i(\hat{g}) \rangle - \frac{||t_i(\hat{g})||^2}{2} \right], \tag{3.40}$$

The high SNR rule for DBPSK can be derived similarly.

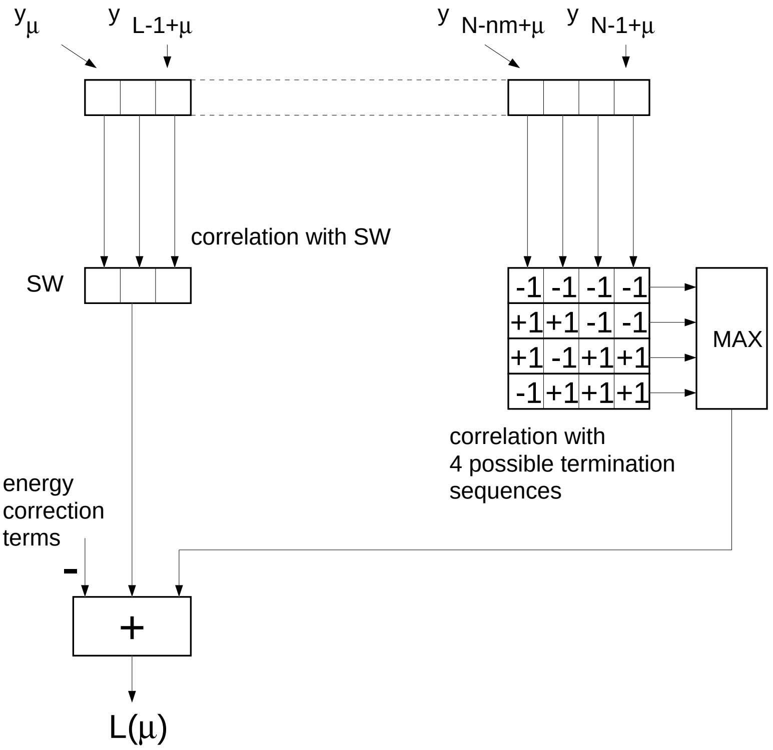
3.6.3 Synchronization of Terminated Trellis Encoded Sequences for Demodulation with Phase Ambiguity

We proceed in the same way as in section 3.4 where we will average over all the possible phase errors due to ambiguity, $\phi_a(k)$, when determining the (one dimensional) likelihood function. We rewrite (3.36) as

$$L_{A1}(\mu) = \sum_{k=0}^{M_a-1} \left\{ \prod_{i=0}^{L-1} e^{\frac{2}{N_0} \langle e^{-j\phi_a(k)} \cdot y_{i+\mu}, S_i \rangle} \cdot \sum_{g=0}^{2^m-1} \prod_{i=N-nm}^{N-1} e^{\frac{2}{N_0} \langle e^{-j\phi_a(k)} \cdot y_{i+\mu}, t_i(g) \rangle - \frac{||t_i(g)||^2}{2}} \right. \\ \left. \cdot Pr\{\phi_a(k)\} \right\} \cdot \sum_{\forall \vec{d}} Pr\{\vec{d}\} \cdot \prod_{i=L}^{N-nm-1} e^{\frac{2}{N_0} \langle y_{i+\mu}, d_i \rangle - \frac{||d_i||^2}{2}}, \tag{3.41}$$

again assuming equally likely data sequences. Now to find an approximation we first make use of the fact that

$$\sum_{g=0}^{2^m-1} \prod_{i=N-nm}^{N-1} e^{\frac{2}{N_0} \langle e^{-j\phi_a(k)} \cdot y_{i+\mu}, t_i(g) \rangle - \frac{||t_i(g)||^2}{2}} \approx \prod_{i=N-nm}^{N-1} e^{\frac{2}{N_0} \langle e^{-j\phi_a(k)} \cdot y_{i+\mu}, t_i(\hat{g}) \rangle - \frac{||t_i(\hat{g})||^2}{2}}, \tag{3.42}$$



-1	-1	-1	-1
+1	+1	-1	-1
+1	-1	+1	+1
-1	+1	+1	+1

Figure 3.8: Suboptimal synchronizer using trellis termination information for $m = 2$. The 2^m possible termination sequences are correlated with the received sub-sequence and the largest correlation is chosen. Note that $\hat{g}$ is determined implicitly in the selection of the greatest correlation, and the value of the correction term is independent of the outcome of this maximization process.

where $\hat{g}(k)$ is that g which maximizes

$$\sum_{i=N-nm}^{N-1} \langle e^{-j\phi_a(k)} \cdot y_{i+\mu}, t_i(g)\rangle - \frac{||t_i(g)||^2}{2} \qquad (3.43)$$

-exactly the same assumption as (3.38) only before taking logarithms. Now again assuming that one of the summands will dominate the sum $\sum_{k=0}^{M_a-1}$ in (3.41), we arrive at the high SNR approximation

$$L_H(\mu) = \sum_{i=0}^{L-1}\langle e^{j\cdot\hat{k}\cdot 2\cdot\pi/M} \cdot y_{i+\mu}, S_i - W_{\hat{j}}\rangle + \frac{||W_{\hat{j}}||^2}{2} + \sum_{i=N-nm}^{N-1} \langle e^{j\cdot\hat{k}\cdot 2\cdot\pi/M} \cdot y_{i+\mu}, t_i(\hat{g}(\hat{k})) - W_{\hat{j}}\rangle + \frac{||W_{\hat{j}}||^2 - ||t_i(\hat{g}(\hat{k}))||^2}{2}, \qquad (3.44)$$

where $\hat{k}$ is that k which maximizes

$$\sum_{i=0}^{L-1}\langle e^{j\cdot k\cdot 2\cdot\pi/M} \cdot y_{i+\mu}, S_i\rangle + \sum_{i=N-nm}^{N-1} \langle e^{j\cdot k\cdot 2\cdot\pi/M} \cdot y_{i+\mu}, t_i(\hat{g}(k))\rangle - \frac{||t_i(\hat{g}(k))||^2}{2}. \qquad (3.45)$$

This rather complicated relation between $\hat{g}$ and $\hat{k}$ can best be explained with the aid of BPSK as an example. Here, $\hat{k} \in \{0, 1\}$ is chosen to be 0 if

$$\begin{aligned}\sum_{i=0}^{L-1} y_{i+\mu} \cdot S_i + \sum_{i=N-nm}^{N-1} y_{i+\mu} \cdot t_i(\hat{g}(k=0)))\rangle \\ \sum_{i=0}^{L-1} -y_{i+\mu} \cdot S_i + \sum_{i=N-nm}^{N-1} -y_{i+\mu} \cdot t_i(\hat{g}(k=1), \end{aligned} \tag{3.46}$$

and vice-versa. $\hat{g}(k=0)$ is that g that maximizes

$$\sum_{i=N-nm}^{N-1} y_{i+\mu} \cdot t_i(g), \tag{3.47}$$

and $\hat{g}(k=1)$ is that g that maximizes

$$\sum_{i=N-nm}^{N-1} -y_{i+\mu} \cdot t_i(g). \tag{3.48}$$

From this example and also from (3.43) we see that the search for $\hat{g}$ must be performed M_a times -once for each phase ambiguity. Each $\hat{g}$ is then used in (3.45) (BPSK: (3.46)) to find the most likely phase ambiguity $\hat{\phi}_a = \hat{k} \cdot 2 \cdot \pi / M_a$. These selection processes must be carried out *for each position* μ. The value of $\hat{\phi}_a$ corresponding to $\hat{\mu}$ can be used as a phase reference to resolve the ambiguity before further processing (e.g. decoding) as illustrated in 3.4.1.

3.6.3.1 Code invariance to phase ambiguity

One special case still remains to be treated, and that is if the code is invariant to a phase rotation of one or more $\pm\phi_r = \pm\frac{k_r 2\pi}{M}$. If this is the case, then one can write

$$\forall k_r,\ \forall g\ ;\ \{0 \leq g \leq 2^m\},\ \exists\ h^{\pm}(g) \neq g\ :\ t_i(g) = e^{\pm\frac{jk_r 2\pi}{M}} \cdot t_i(h^{\pm}(g)), \tag{3.49}$$

since every termination sequence rotated by $\pm\frac{k_r 2\pi}{M}$ remains a 'legal' termination sequence, $t_i(h^{\pm}(g))$, hence

$$\forall k_r,\ \sum_{i=N-nm}^{N-1} \langle e^{\pm\frac{jk_r 2\pi}{M}} \cdot y_{i+\mu}, t_i(h^{\pm}(g))\rangle = \sum_{i=N-nm}^{N-1} \langle y_{i+\mu}, t_i(g)\rangle. \tag{3.50}$$

The left and right hand side above correspond to three of the possible cases over which (3.43) must be maximized, namely $k = 0$ (right hand side above) and $k = k_r$ resp. $k = M_a - k_r$ (left hand side above) -bearing in mind that the normalization term $-\frac{\|t_i(g)\|^2}{2}$ in (3.43) will be the same in all three cases. Thus (3.43) need only be evaluated for those $k \in \mathcal{E}$, where the set $\mathcal{E} \subset \{0, 1, ..., M_a - 1\}$ is constructed such that

$$\forall k_r,\ \forall k_i \in \mathcal{E},\ \sim \exists k_j \in\ \mathcal{E}\ :\ k_i = \begin{cases} (k_j + k_r)\ \mathrm{mod} M_a\ ;\ \text{or} \\ (k_j - k_r + M_a)\ \mathrm{mod} M_a\ . \end{cases} \tag{3.51}$$

and

$$\forall k_r,\ \forall k_l \in \{0, ..., M_a - 1\},\ \exists k_j \in \mathcal{E}\ :\ k_l = \begin{cases} (k_j + k_r)\ \mathrm{mod} M_a\ ;\ \text{or} \\ (k_j - k_r + M_a)\ \mathrm{mod} M_a\ . \end{cases} \tag{3.52}$$

Example: Let $M = M_a = 8$ (e.g. 8PSK) and $k_r = 4$ ($\equiv \pm 180^o$ phase invariance) then we must evaluate (3.43) $\forall\ k \in \mathcal{E} = \{0, 1, 2, 3\}$.

3.6.4 Implementation of the High SNR Rule for BPSK

As mentioned in section 3.6.2, one can make use of the tree structure of the terminating (or starting) portion of the trellis. This leads to quite a simple implementation for small code memories. In Figs. 3.9 and 3.10 we have drawn the structure of the high SNR synchronizers for BPSK that makes use of just the termination portion of the frame for rate $\frac{1}{2}$ codes with memory two and four respectively.

We note the higher complexity of the memory four scheme (21 adders, 9 maximizers compared to 5 adders, 3 maximizers for the memory two scheme). In general, for rate $\frac{1}{2}$, we can calculate the number of maximizers, n_M as

$$n_M = 1 + 2 + 4 + ... + 2^{m-2} + 2 = \sum_{i=0}^{m-1} 2^i + 2 = 2^{m-1} + 1, \tag{3.53}$$

and the number of adders, n_A as

$$n_A = (n_M - 2) \cdot 2 + (m - 1) \cdot 2 + 1 = 2^m + 2 \cdot m - 3. \tag{3.54}$$

As we would expect, the complexity increases exponentially with the memory m; with the brute force solution of (3.37) it would increase with order $m \cdot 2^m$ -order m operations for each of 2^m possible g. One must not forget, that two such modules are needed if both the starting and termination portions are to be utilized; if processing speed is not critical one unit may be time-shared by changing the coefficients of the multipliers to take account of the difference between starting and termination sequences. Nevertheless, the complexity may still be justified if one wants to shorten the sync word or operate at a lower signal-to-noise ratio. Furthermore, use of the list synchronizer can reduce the computational burden, especially in a DSP implementation. Performance evaluation results will be given in section 4.5.

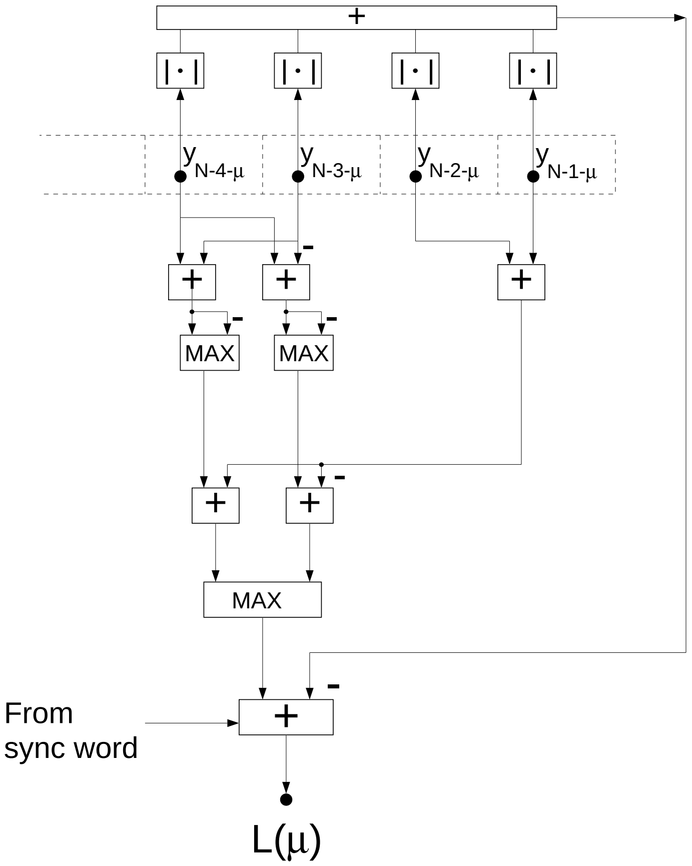


Figure 3.9: *Structure of the synchronizer using trellis termination (starting portion is treated analogously) for memory two and octal generator matrix* $\binom{5}{7}$. *The two maximization units on the top left calculate the highest correlation of the two symbol long sequence comprised of the symbols* $y_{N-4-\mu}$ *and* $y_{N-3-\mu}$ *with the two pairs of sub-sequences* $\{-1,-1\}$, $\{1,1\}$ *and* $\{1,-1\}$, $\{-1,1\}$. *The final maximization unit calculates the highest correlation with all (four) possible termination sequences, by adding to the two results of the first maximizations, the correlation of the two symbol long sequence (comprised of the symbols* $y_{N-2-\mu}$ *and* $y_{N-1-\mu}$*) with either the sub-sequence* $\{-1,-1\}$ *or* $\{1,1\}$. *From the correlation value we then subtract the sum of the absolute values of the received sub-sequence -the correction term characteristic of the high SNR rule. The associated trellis is shown in Fig. 3.5.*

3.7 Synchronization in the Non-Frequency Selective Fading Channel

Up to now, we have assumed an AWGN channel when deriving our likelihood functions. It is interesting to investigate how the optimal synchronizer must be constructed for different channel models. Recently, Moon and Soliman derived the ML rule for a time invariant AWGN

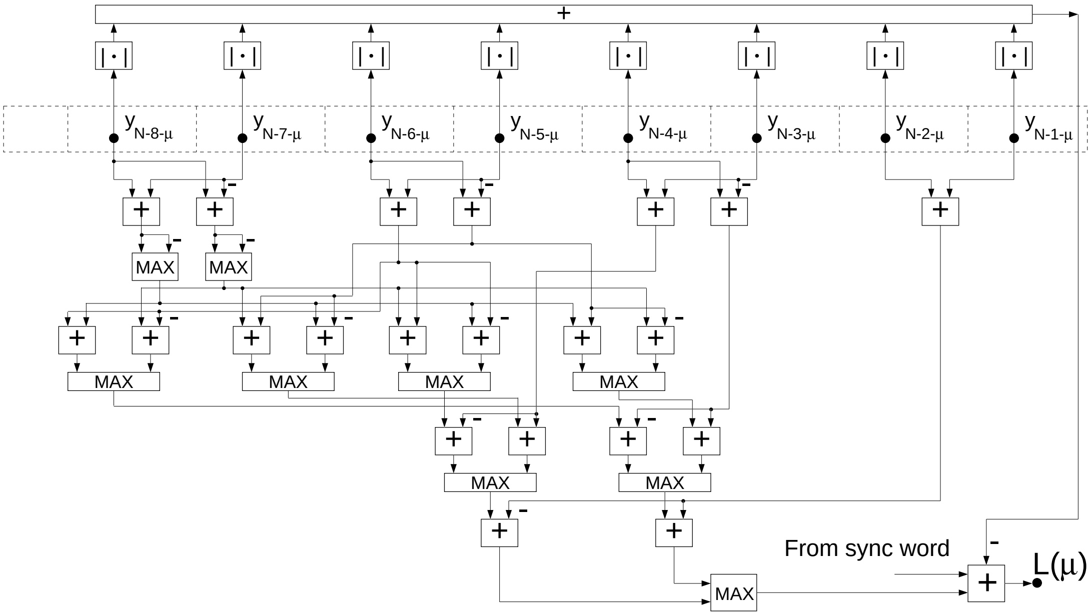


Figure 3.10: *Structure of the synchronizer using trellis termination (starting portion is treated analogously) for memory four and octal generator matrix* $\left(\begin{smallmatrix}23\\35\end{smallmatrix}\right)$. *The number of maximization units decreases by a factor of two at each stage -corresponding to the halving of the number of branches.*

channel with known intersymbol interference, together with two approximations [MS91]. One is confronted with this type of channel when transmitting over telephone lines or in many mobile communications environments [Pro89] [Hoe90].

Here we look towards the derivation of the ML rule for a non-frequency selective fading channel with M-ary phase coherent signaling. By non-frequency selective we mean that no intersymbol interference occurs, i.e. the frequency transfer function of the channel is constant over the frequency range occupied by the transmitted signal. Only the phase and amplitude of the channel are time variant. We encounter such channels in mobile communications when the path delay between signal arriving at the receiver have a time difference much smaller than the symbol duration [Pro89]. The 60 GHz mobile channel [Sch91] is a good example, furthermore, the fading is quite rapid with respect to the frame (or packet length), as presumed in the following.

We assume that channel state information (CSI) is available for each received symbol (which is the case, for instance, when a subsequent ML Viterbi decoder uses CSI to improve decoding performance; CSI perhaps generated using a Kalman filter [HM88], or might be derived from an AGC circuit). As will be explained later, it is assumed that the sync word is spread over the data frame. We derive a high SNR rule, and later we will show that there is no useful equivalent of the soft correlation rule.

The use of channel state information to improve the performance of a ML (Viterbi) decoder in the case of fading was proposed by Hagenauer [Hag80]. The new metric is hardly more complex than the common soft decision metric for the AWGN channel, but gives significant improvement in terms of power efficiency.

3.7.1 Likelihood Functions

We assume a multiplicative non-frequency selective fading channel followed by an additive white Gaussian noise channel (AWGN). Fig. 3.11 shows the discrete equivalent channel model used, together with the structure of a possible receiver employing a Viterbi decoder that also uses CSI. By assuming fast fading and spreading the sync word across the frame we have independent channel states for each symbol, see Fig. 3.13. The complex fading value (channel state) for each symbol i is represented by a_i and here we assume perfect channel estimation i.e. $a_i = \hat{a}_i$. Furthermore, a_i is assumed to be constant over a symbol interval. Our channel model is thus:

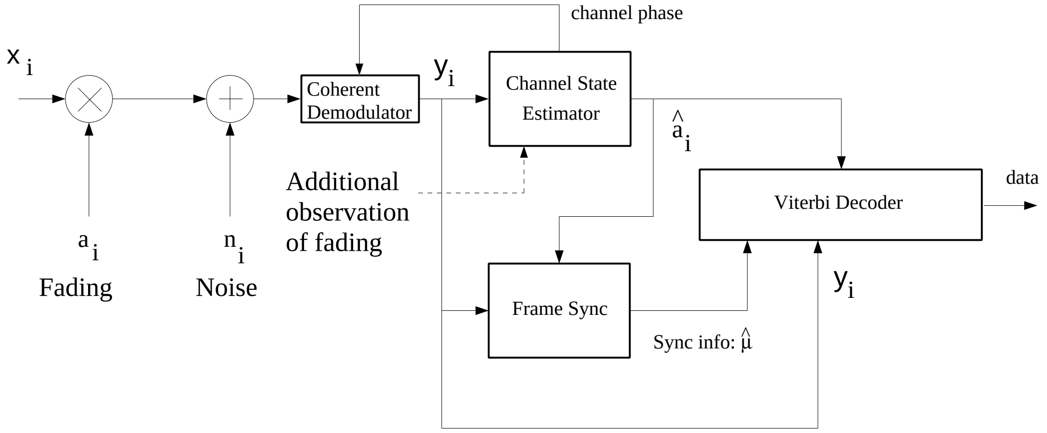


Figure 3.11: *Possible receiver structure with a synchronizer using CSI: CSI is generated for both a ML Viterbi decoder and the frame synchronizer. Coherent detection is assumed, hence the channel states a_i are real valued as far as the frame synchronizer and decoder are concerned. In practice, channel estimation may be obtained with the aid of a Kalman filter.*

$$y_i = x_i \cdot a_i + n_i, \tag{3.55}$$

which is illustrated in Fig. 3.12 for the case of BPSK. The CSI estimator now has to estimate

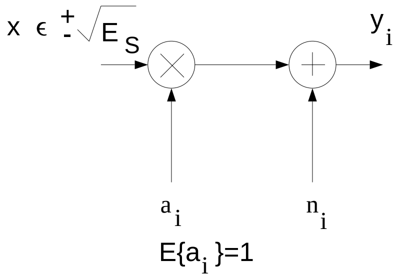


Figure 3.12: *Channel model for the non-frequency selective fading channel in the case of BPSK. We have assumed phase coherent demodulation, hence the fading values a_i are real valued.*

both the phase and amplitude of a_i in oder to enable coherent detection. If we assume that the phase estimation is possible, and that the received symbols are re-rotated (phase correct demodulation) before being passed to the frame synchronizer, then we can replace our a_i in the model by just the amplitude of a_i. For simplicity, we shall still denote the real-valued channel state by a_i.

Because the sync word is spread across the frame, the synchronizer must know which symbols of a frame are sync symbols. The sequence $\mathbf{g} = (g(0), g(1), ..., g(L-1))$ denotes the positions of the L sync symbols; similarly, $\mathbf{h} = (h(0), h(1), ..., h(N-L-1))$ denotes the data positions (see Fig. 3.13).

We begin by writing the PDF

$$f_{\vec{\mathbf{y}}}(\vec{y}|\mu) = \frac{1}{(\pi N_0)^N} \prod_{i=0}^{L-1} e^{\frac{-||y_{g(i)+\mu} - S_i a_{g(i)+\mu}||^2}{N_0}} \cdot \sum_{\forall\ \vec{d}} Pr\{\vec{d}\} \cdot \prod_{i=L}^{N-1} e^{-\frac{||y_{h(i)+\mu} - d_i a_{h(i)+\mu}||^2}{N_0}}, \tag{3.56}$$

as before. After a derivation similar to those presented in the previous sections we obtain the likelihood function [Rob92b]:

$$L(\mu) = \sum_{i=0}^{L-1} \left(a_{g(i)+\mu} \langle y_{g(i)+\mu}, S_i \rangle - \frac{1}{2} ||S_i a_{g(i)+\mu}||^2 \right) -$$
$$\frac{N_0}{2} \sum_{i=0}^{L-1} \ln \sum_{j=1}^{M} e^{\frac{2}{N_0} \left[a_{g(i)+\mu} \langle y_{g(i)+\mu}, W_j \rangle - ||W_j||^2 \frac{a^2_{g(i)+\mu}}{2} \right]}. \tag{3.57}$$

If we compare this likelihood function with (3.11) we notice that there is an extra term, and that the channel state information is used in several places in (3.57). Note that even if all $||W_j||^2$ are equal, the last term in the exponent of (3.57) cannot be deleted. The first term is the correlation rule with CSI. The equivalent high SNR approximation of (3.57) is

$$L_H(\mu) = \sum_{i=0}^{L-1} \left[a_{g(i)+\mu} \langle y_{g(i)+\mu}, S_i - W_{\hat{j}} \rangle + \frac{a^2_{g(i)+\mu}}{2} \left(||W_{\hat{j}}||^2 - ||S_i||^2 \right) \right], \tag{3.58}$$

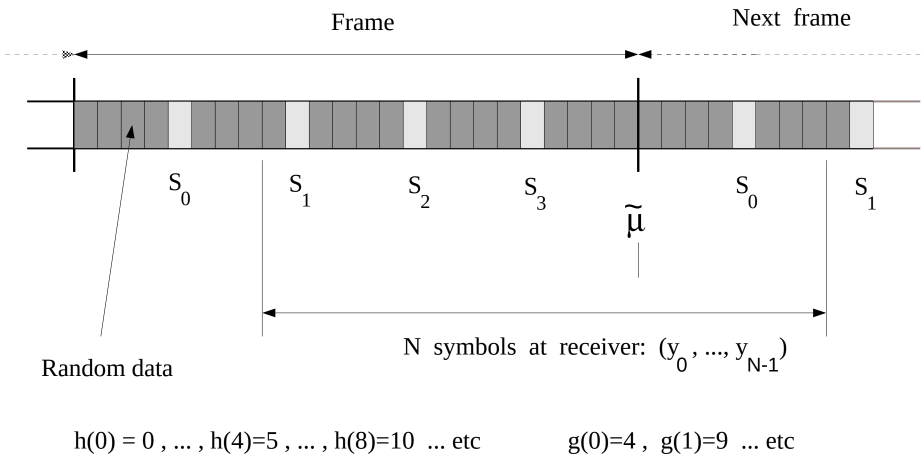

Frame
Next frame
S0
S1
S2
S3
$\tilde{\mu}$
S0
S1
N symbols at receiver: (y0 , ..., yN-1)
Random data
h(0) = 0 , ... , h(4)=5 , ... , h(8)=10 ... etc
g(0)=4 , g(1)=9 ... etc

Figure 3.13: *Frame structure for the non-frequency selective fading channel: by interleaving the sync word in the frame, we ensure independent fading values for each sync word symbol. Here we show the indices h and g that are needed to address each symbol of the sync word and data.*

where $\hat{j}$ is that j which maximizes $a_{g(i)+\mu}\langle y_{g(i)+\mu}, W_j\rangle - ||W_j||^2\frac{a^2_{g(i)+\mu}}{2}$. For PSK signalling $||W_j||^2$ are all equal and can be omitted:

$$L_H(\mu) = \sum_{i=0}^{L-1}\left[a_{g(i)+\mu}\langle y_{g(i)+\mu}, S_i - W_{\hat{j}}\rangle\right]. \tag{3.59}$$

This is very similar to the high SNR rule (3.13) derived earlier, except for the multiplication with the CSI factor corresponding to each symbol, and reminds us of the metric to be used by a Viterbi decoder utilizing CSI [Hag80].

We must, however, look critically at situations where the likelihood functions using CSI will be employed. We have assumed perfect estimation of the phase (to achieve coherent detection) and amplitude (a_i's) of the fading value. In a traditional frame sync environment, the first tasks of a receiver will be to achieve timing and phase synchronization i.e. in this case channel estimation; only then will frame sync be attempted: in such a case assuming CSI could be realistic. However, in the 'one shot' packet frame sync problem which we will concentrate on later, the number of symbols available for a Kalman filter, for example, to reach acquisition is very small. It might even be necessary to use information about known symbols (e.g. the sync word!) to estimate the channel at all: here, CSI cannot be assumed available.

Chapter 4

Performance of Traditional Frame Synchronizers

In this chapter the performance of some of the different synchronizers from the previous chapter will be compared, both analytically and with approximations. In the former case it is possible to exactly determine the synchronization rate at infinite SNR. Subsequently, we will make use of the union bound to derive an upper bound for the synchronization failure rate of the correlation rules and the technically important and quasi-optimal high SNR rule. Since the evaluation of the union bound involves convolution of PDFs -which in the case of the soft correlation rule are Gaussian- if they are not Gaussian, we apply the central limit theorem when treating the synchronization rate of the high SNR rule and will give a simple to calculate bound for BPSK with and without phase ambiguity and also QPSK (without phase ambiguity).

Simulation results will confirm these findings and will be essential to describe the more complex synchronizers (ML rule, list synchronizer, trellis termination and the synchronizers for the fading channel). Also, we shall very briefly introduce the problem of sync word choice which, of course, has an effect on the performance of the synchronizers. In the following, we will denote the frame synchronization failure rate by $Pr\{f\}$.

4.1 Performance of the ML, High SNR and Correlation Rules for the Noiseless Case

In [Nie73], [LT86] and [LT87] the synchronization rate of a frame synchronizer is analyzed, whose performance is limited by the possible repetition of the sync word in the random data surrounding the true sync word. We can express their argument briefly as follows: Assuming that the frames are not perturbed by noise, the value of the likelihood function corresponding to the correct frame starting position will always be the same, and the highest value it can obtain. Furthermore, the choice of likelihood function is arbitrary (correlation rules, ML rule and its high SNR approximation), as can be trivially shown by inspection of the corresponding likelihood functions. If the sync word is repeated *once* anywhere in the observation data portion of $\vec{y}$, then the receiver must make a random choice between the two possible positions; likewise, when two repetitions occur it must make a choice out of three. The synchronization failure probability $Pr\{f|\mathrm{RDL}\}$ (RDL: Random Data Limited) can be calculated exactly, and with the necessary premise that no equally long prefix and suffix of the sync word shall be equal, i.e.

$$\forall s\ ,\ 1 \leq s \leq L-1\ :\ (S_0, S_1, \ldots, S_{s-1}) \neq (S_{L-s}, S_{L-s+1}, \ldots, S_{L-1}) \tag{4.1}$$

one obtains (see [Nie73], [Mas93a] and Appendix B):

$$Pr\{f|\mathrm{RDL}\} = \sum_{i=1}^{\lfloor N/L-1 \rfloor} \frac{(-1)^i}{i+1} \binom{N-L-(L-1)i}{i} M^{-Li}. \tag{4.2}$$

Remember that N is the frame length, L the sync word length and M the number of symbols of the modulation format. In Fig. 4.1 we show the values of $Pr\{f|\mathrm{RDL}\}$, evaluated as a function of the frame length N and sync word length L, for $M = 2$ (i.e. binary signaling). We observe that increasing L slightly leads to a marked synchronization improvement, increasing N only degrades performance slightly, for moderate to large N. $Pr\{f|\mathrm{RDL}\}$ can be used as an initial help when designing communications systems, as it serves as a loose upper bound on performance.

4.2 Union Upper Bound on the Synchronization Failure Rate for the Correlation and High SNR Rules

An upper bound to the synchronization failure probability of the soft correlation rule for coherent demodulation without phase ambiguity has been presented in [LT87]. Unfortunately, the derivation is hard to follow and details are partly erroneous in the paper. For this reason we will again derive the same bound in a different way, but initially confine our analysis to BPSK (see 4.2.6 for extensions to PSK in general). The framework of our analysis will -in principle- enable different likelihood functions to be tackled. This is valuable for the extension of the bound to the high SNR synchronization rule which we will direct our attention to in 4.2.2. Furthermore, the steps of our derivation can be illustrated graphically, which will lead to a deeper understanding of *why* the high SNR rule originally proposed by Massey [Mas72] outperforms the simple soft correlation rule.

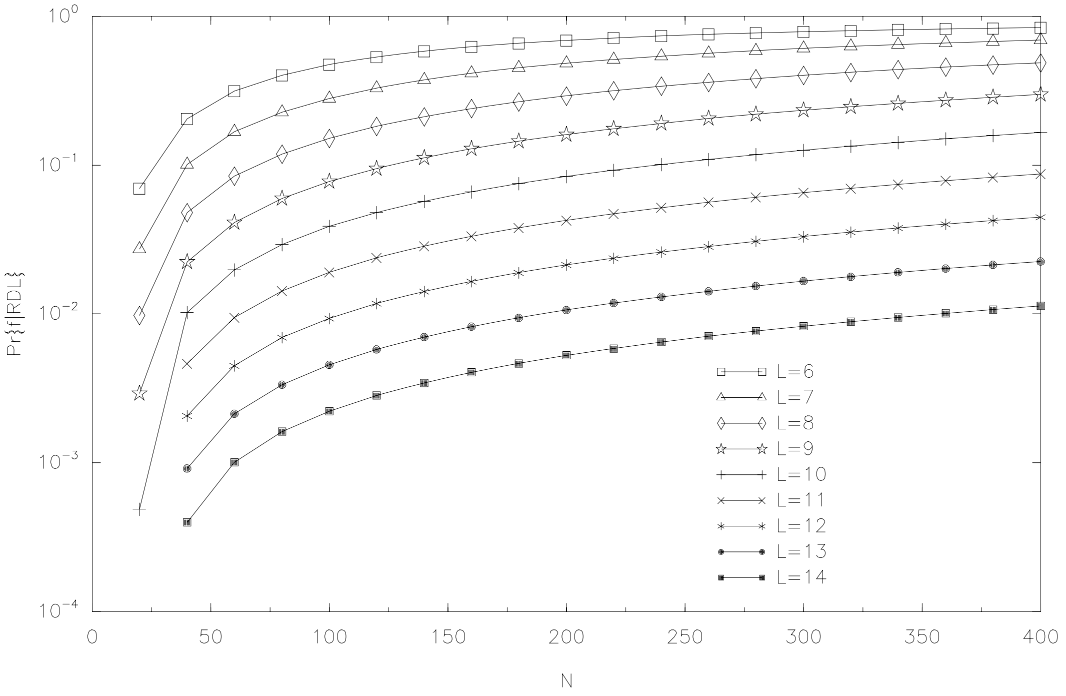


Figure 4.1: Values of $Pr\{f|\text{RDL}\}$ as a function of L and N, $M = 2$.

Because of symmetry and equally likely $\tilde{\mu}$, we can limit our analysis to the case $\tilde{\mu} = 0$. The synchronizer will chose that μ that has the highest value of $L(\mu)$. Applying the union bound an introducing the variable μ' to denote a 'competing' frame starting position, we can write:

$$Pr\{\text{f}\} \leq \sum_{\mu'=1}^{N-1} \left[Pr\{L(\mu') > L(\tilde{\mu} = 0)\} + \frac{1}{2} \cdot Pr\{L(\mu') = L(\tilde{\mu} = 0)\} \right]. \tag{4.3}$$

Note that the accuracy of the bound is influenced by the definition of a sync failure event. There are synchronization rules that can lead to discrete values of $L(\mu)$ occurring -specifically the hard correlation and high SNR rules. In these cases, the receiver will have the chance of randomly choosing from all positions yielding equal valued likelihood functions, we made use of this when deriving $Pr\{f|\text{RDL}\}$. To account for this in our bound, we shall assume a sync failure probability of 0.5 if $L(\mu') = L(\tilde{\mu} = 0)$ which is included in the bound (4.3). It is easy to show that this does not affect the validity of the bound -on the contrary, it makes it looser for very high SNR.

Let us now define

$$\Delta L = L(\tilde{\mu} = 0) - L(\mu'). \tag{4.4}$$

We must be aware that ΔL **is a random variable** that depends on many parameters (noise, data, sync word etc). Assuming statistical independence[1] of $L(\mu')$ and $L(0)$ we can write the

[1] we shall discuss this restriction shortly

PDF of ΔL as

$$f_{\mathbf{\Delta L}}(\Delta L) = f_{L(0)}(L(0)) * f_{L(\mu')}(-L(\mu')), \tag{4.5}$$

where $*$ denotes the convolution. Thus we can write our union bound (4.3) as

$$\begin{aligned} Pr\{\mathrm{f}\} &\leq \sum_{\mu'=1}^{N-1} \left[\lim_{\epsilon \to 0} \int_{-\infty}^{-\epsilon} f_{\mathbf{\Delta L}}(\Delta L) d\Delta L + \frac{1}{2} \cdot Pr\{\Delta L = 0\} \right] \\ &= \sum_{\mu'=1}^{N-1} \left[\lim_{\epsilon \to 0} \int_{-\infty}^{-\epsilon} f_{L(0)}(L(0)) * f_{L(\mu')}(-L(\mu')) dL + \frac{1}{2} \cdot Pr\{L(\mu') = L(\tilde{\mu} = 0)\} \right], \end{aligned} \tag{4.6}$$

where $\epsilon > 0$. We must now find a way of expressing the PDFs of $L(0)$ and $L(\mu')$. We should at this stage remember that the likelihood functions are generally a sum of individual components. Arriving at these PDFs is quite easy if the following applies:

- the PDFs of the components forming these likelihood functions can be expressed, and can be easily convoluted.

We will tackle this restriction as well as the assumption of independence of $L(0)$ and $L(\mu')$ in the sequel.

4.2.1 Union Upper Bound on the Synchronization Failure Rate for the Soft Correlation Rule and BPSK

The soft correlation rule for BPSK is (equation (3.15))

$$L_C(\mu) = \sum_{i=0}^{L-1} y_{i+\mu} \cdot S_i = \sum_{i=0}^{L-1} C_i. \tag{4.7}$$

The PDF of $L_C(\mu)$ is given by the convolution

$$f_{\mathbf{L_C}}(L_C) = f_{\mathbf{C_0}} * f_{\mathbf{C_1}} * \ldots * f_{\mathbf{C_{L-1}}}, \tag{4.8}$$

as the C_i are independent (because of white noise). Assuming that the elements of data and sync word are chosen from $\{-1 \cdot \sqrt{\text{Energy}}, 1 \cdot \sqrt{\text{Energy}}\}$, (i.e. unit symbol energy for simplification), we can write

$$f_{\mathbf{C_i}}(C_i) = \frac{1}{\sqrt{\pi N_0}} \cdot e^{\frac{(C_i - S_i x_{i+\mu'})^2}{N_0}} = \mathcal{N}_{C_i}(S_i x_{i+\mu'}, N_0/2), \tag{4.9}$$

where x_i is the i-th element of the actually transmitted sequence $\vec{x} = \vec{y} - \vec{n}$. We have introduced $\mathcal{N}_a(m, \sigma^2)$ to denote a Gaussian distribution of the random variable a with mean

m and variance σ^2. Note that in the following we will omit the dimension $\sqrt{\text{Energy}}$ in our illustrations and examples.

Assuming that $L \leq \mu' \leq N - L$, i.e. the elements c_i of $L_C(0)$ and $L_C(\mu')$ are independent, as our 'competing sync word' starting at μ' does not overlap the real one, we can insert (4.9) into (4.5) and simply convolute the pertinent $f_{\mathbf{C_i}}$ by adding means and variances. At this stage it should become evident that the first moment of $C_i = S_i x_{i+\mu'}$, is itself a random variable that can take on the values ± 1, depending on the similarity of S_i and $x_{i+\mu'}$. We now introduce κ_L to denote the **cross-correlation between the random sequence** $(x_{\mu'}, ..., x_{\mu'+L-1})$ **of length** L **with the sync word** $(S_0, ...S_{L-1})$, that is

$$\kappa_L = \sum_{i=0}^{L-1} S_i x_{i+\mu'}. \tag{4.10}$$

κ_L is a random variable since the data symbols are random, with discrete PDF $f_{\kappa_{\mathbf{L}}}(\kappa_L)$; for its calculation, see Appendix C. This discrete PDF can be expressed simply as the probabilities of each κ_L: $Pr\{\kappa_L\}$. Very large κ_L are unlikely, since they correspond to a *strong similarity* between the sync word and the random sequence; similarly, very small κ_L are unlikely. The distribution of κ_L is shown in the first curve of Fig. 4.6.

Now the mean of $L_C(0) = \sum_{i=0}^{L-1} S_i x_i = L$, hence the PDF of $L_C(0)$ becomes:

$$f_{\mathbf{L_C(0)}}(L_C(0)) = \mathcal{N}_{L(0)}(L, L\frac{N_0}{2}), \tag{4.11}$$

since there are L contributions from noise terms with variance $\frac{N_0}{2}$, leading to a variance of $L \cdot \frac{N_0}{2}$. Similarly,

$$f_{\mathbf{L_C}(\mu')}(L_C(\mu')) = \mathcal{N}_{L(\mu')}(\kappa_L, L\frac{N_0}{2}). \tag{4.12}$$

So inserting the last two equations into (4.5), we obtain the desired PDF of ΔL_C:

$$f_{\mathbf{\Delta L_C}(\kappa_\mathbf{L})}(\Delta L_C) = \mathcal{N}_{L(0)}(L, L\frac{N_0}{2}) * \mathcal{N}_{L(\mu')}(-\kappa_L, L\frac{N_0}{2}) = \mathcal{N}_{\Delta L_C}(L - \kappa_L, LN_0). \tag{4.13}$$

One should be aware that the PDF of ΔL is itself a function of a random variable, namely κ_L. We can now express the integral in (4.6) as

$$\lim_{\epsilon \to 0} \int_{-\infty}^{-\epsilon} f_{\mathbf{\Delta L_C}(\kappa_\mathbf{L})}(\Delta L_C) d\Delta L_C = \int_{-\infty}^{0} \mathcal{N}_{\Delta L_C}(L - \kappa_L, LN_0) d\Delta L_C = \frac{1}{2} \cdot \operatorname{erfc}\left(\frac{L - \kappa_L}{\sqrt{2LN_0}}\right), \tag{4.14}$$

where we have made use of the complementary error function

$$\operatorname{erfc}(a) = \frac{2}{\sqrt{\pi}} \int_a^{\infty} e^{-t^2} dt. \tag{4.15}$$

Let us now look at the case when the 'competing sync word' starting at μ' overlaps the real sync word, i.e. when $0 < \mu' < L$ or $N - L < \mu' < N$. In this case the components C_i belonging to $L_C(0)$ and $L_C(\mu')$ *in the overlap region* are no longer independent. This is because the noise samples that have perturbed the x_i 's in the overlap region and which effect both $L_C(\mu')$ and $L_C(0)$ are, of course, identical. We must look at the effect this has on the resulting value of ΔL_C. This can best be illustrated using an example: Let the sync word $\vec{S} = (1, -1, 1, 1)$, and our competing position be $\mu' = 2$; see Fig. 4.2.

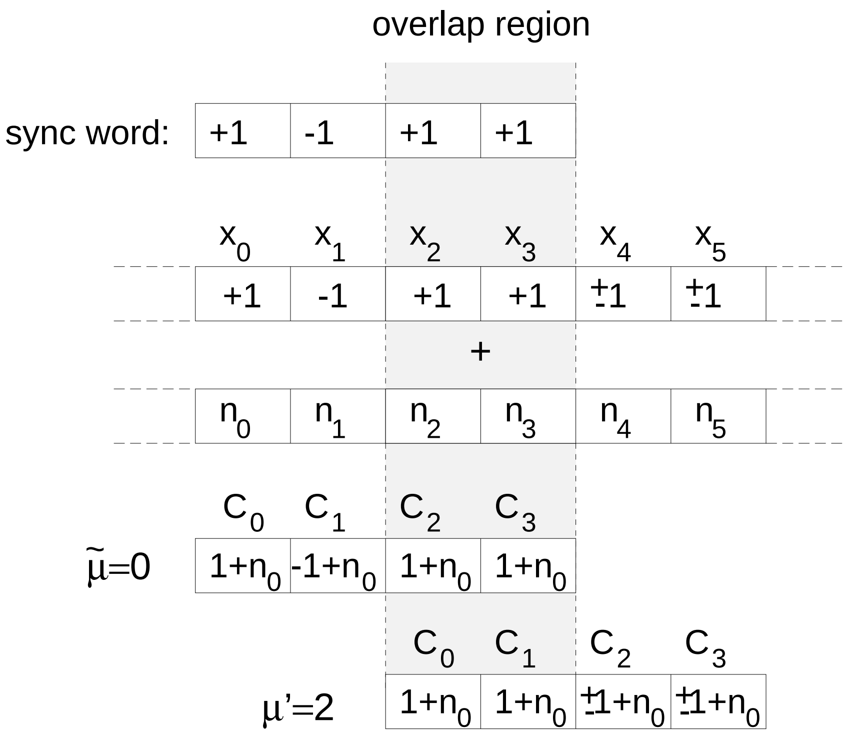


Figure 4.2: *Illustration of the overlap case. Shown are the transmitted symbols, the noise and the values of* C_i *for the two cases:* $\mu = \tilde{\mu} = 0$ *and the competing* $\mu = \mu' = 2$.

Thus

$$L_C(0) = (1 + n_0) + (1 - n_1) + (1 + n_2) + (1 + n_3), \tag{4.16}$$

$$L_C(\mu' = 2) = (1 + n_2) + (-1 - n_3) + (\pm 1 + n_4) + (\pm 1 + n_5). \tag{4.17}$$

The signs of the noise terms affecting the individual C_i depend on the sign of S_i, and the ± 1 indicate the unknown data in the non-overlap part. Let us now evaluate $\Delta L_C = L_C(0) - L_C(\mu')$,

$$\Delta L_C = (1 + n_0) + (1 - n_1) + 0 + 2(1 + n_3) - (\pm 1 + n_4) - (\pm 1 + n_5). \tag{4.18}$$

We observe that the noise term n_2 has disappeared, and the term n_3 is doubled in amplitude. In general, it is clear that, in the overlapping region, transmitted symbols of the 'competing sync word' starting at μ' that are equal to the corresponding real sync word symbol (i.e. when $S_{i+\mu'} = x_{i+\mu'} = S_i$) have no influence on ΔL_C. On the other hand, if $S_{i+\mu'} = x_{i+\mu'} \neq S_i$ (again only in the overlapping region) ΔL_C is increased by $2 + 2n_{i+\mu'}$. The number of each of these two cases is given by the partial auto-correlation function of the sync word at shift μ', defined as

$$R_{\mu'} = \sum_{i=0}^{L-1-\mu'} S_i \cdot S_{i+\mu'}. \tag{4.19}$$

Now:

- cases where: $x_{i+\mu'} = S_i$; occurs $\frac{L-\mu'+R_{\mu'}}{2}$ times,
- cases where: $x_{i+\mu'} \neq S_i$; occurs $\frac{L-\mu'-R_{\mu'}}{2}$ times.

The contribution to the PDF of ΔL_C by the remaining symbols lying outside the overlap region is similar to the case before (equation (4.13)), only now the number of these symbols is no longer L, but μ'. Similarly, we must replace κ_L by $\kappa_{\mu'}$ which denotes the partial cross-correlation between the random sequence of length μ' with the μ' long sub-sequence of the sync word.

The mean of ΔL_C is now

$$m_{\Delta L} = (\mu' - \kappa_{\mu'}) + 2 \cdot \frac{L - \mu' - R_{\mu'}}{2} = L - (R_{\mu'} + \kappa_{\mu'}), \tag{4.20}$$

where the first term $(\mu' - \kappa_{\mu'})$ is from the non-overlap region. The variance of ΔL_C is

$$\sigma^2_{\Delta L} = 2 \cdot \mu' \cdot \frac{N_0}{2} + 4 \cdot \frac{N_0}{2} \cdot \frac{L - \mu' - R_{\mu'}}{2} = (L - R_{\mu'}) \cdot N_0. \tag{4.21}$$

The factor $4 \cdot \frac{N_0}{2}$ takes into account the doubling of the amplitude of the noise terms where $x_{i+\mu'} \neq S_i$. So in analogy to (4.14) we can write

$$\lim_{\epsilon \to 0} \int_{-\infty}^{-\epsilon} f_{\Delta L_C(\kappa_{\mu'}, R_{\mu'})}(\Delta L) d\Delta L_C =$$

$$\int_{-\infty}^{0} \mathcal{N}_{\Delta L_C}(L - (R_{\mu'} + \kappa_{\mu'}), (L - R_{\mu'})N_0) d\Delta L_C = \frac{1}{2}\text{erfc}\left(\frac{L - (R_{\mu'} + \kappa_{\mu'})}{\sqrt{2(L - R_{\mu'})N_0}}\right). \tag{4.22}$$

The evaluation of the union bound has been illustrated graphically in Fig. 4.3. The general principle holds for all possible likelihood functions (not, however, those for phase ambiguity).

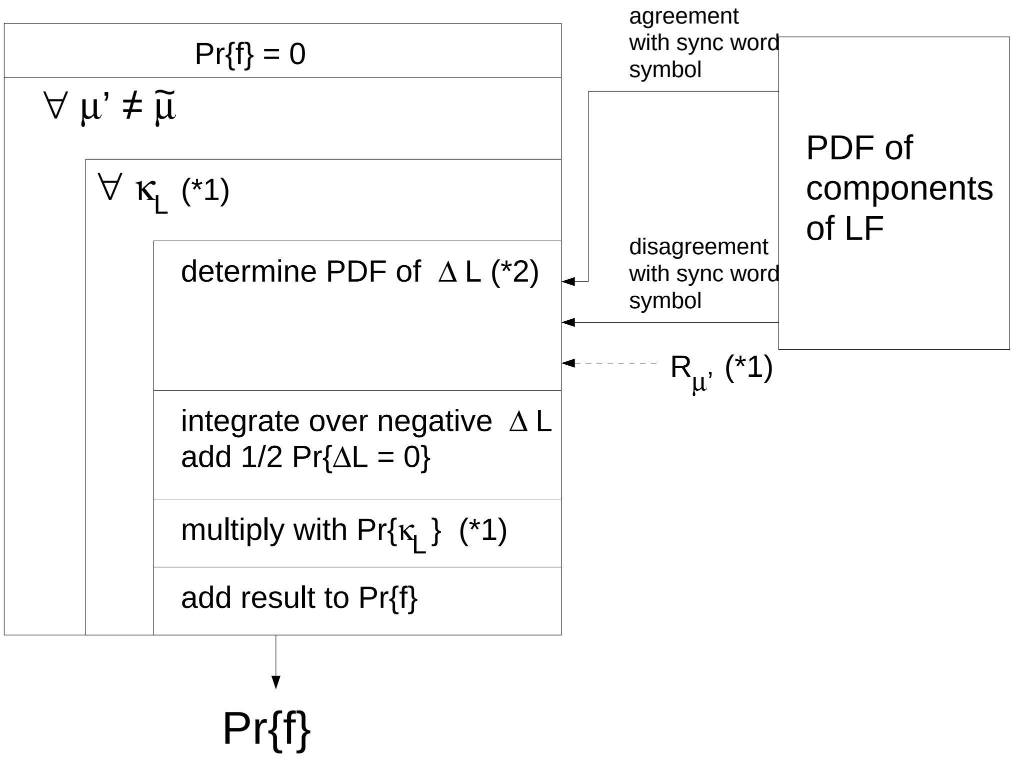

Pr{f} = 0
∀ μ' ≠ μ̃
∀ κL (*1)
determine PDF of Δ L (*2)
integrate over negative Δ L
add 1/2 Pr{ΔL = 0}
multiply with Pr{κL} (*1)
add result to Pr{f}
Pr{f}
agreement with sync word symbol
disagreement with sync word symbol
PDF of components of LF
Rμ' (*1)

Figure 4.3: *Illustration of the evaluation of the union bound. The principle holds (with exceptions) for all likelihood functions (correlation rule, high SNR rule, etc.). Those cases where the competing sync word overlaps the true sync word have to be treated specially: in particular, (*1) denotes that κ_L be replaced by $\kappa_{\mu'}$ and the partial auto-correlation $R_{\mu'}$ be taken into account. Also, (*2) indicates that dependencies have to be treated when determining the PDF of ΔL.*

Central to the procedure is the evaluation of the PDF of the (additive) components of the likelihood function -in the case of the soft correlation rule they are Gaussian.

To evaluate (4.6), it is necessary to compute the expectations over κ_L and $\kappa_{\mu'}$, since they are random variables, yielding

$$Pr\{f\} \leq \sum_{\mu'=1}^{N-1} \int_{-\infty}^{0} f_{\Delta\mathbf{L_C}}(\Delta L_C)d\Delta L_C = \sum_{\mu'=L}^{N-L} \sum_{\forall\kappa_L} Pr\{\kappa_L\} \cdot \frac{1}{2} \cdot \mathrm{erfc}\left(\frac{L-\kappa_L}{\sqrt{2LN_0}}\right) +$$

$$\sum_{\mu'=1}^{L-1} \sum_{\forall\kappa_{\mu'}} Pr\{\kappa_{\mu'}\} \cdot \frac{1}{2} \cdot \mathrm{erfc}\left(\frac{L-(R_{\mu'}+\kappa_{\mu'})}{\sqrt{2(L-R_{\mu'})N_0}}\right) + \tag{4.23}$$

$$\sum_{\mu'=N-L+1}^{N-1} \sum_{\forall\kappa_{N-\mu'}} Pr\{\kappa_{N-\mu'}\} \cdot \frac{1}{2} \cdot \mathrm{erfc}\left(\frac{L-(R_{N-\mu'}+\kappa_{N-\mu'})}{\sqrt{2(L-R_{N-\mu'})N_0}}\right). \tag{4.24}$$

For reasons of symmetry, the last two sums -both accounting for overlap cases- are identical, and can be combined. Also note that ΔL equals zero with probability zero. The first sum can be written as a multiplication with $N-2L+1$, giving us the result previously found in

[LT87]:

$$Pr\{f\} \leq (N - 2L + 1) \cdot \sum_{\forall \kappa_L} Pr\{\kappa_L\} \cdot \frac{1}{2} \cdot \mathrm{erfc}\left(\frac{L - \kappa_L}{\sqrt{2LN_0}}\right) +$$
$$\sum_{\mu'=1}^{L-1} \sum_{\forall \kappa_{\mu'}} Pr\{\kappa_{\mu'}\} \cdot \mathrm{erfc}\left(\frac{L - (R_{\mu'} + \kappa_{\mu'})}{\sqrt{2(L - R_{\mu'})N_0}}\right). \tag{4.25}$$

For the calculation of $Pr\{\kappa_L\}$ and $Pr\{\kappa_{\mu'}\}$ see Appendix C. The result we have obtained is identical to that in [LT87] (save their mistake in equation (28)), however, the derivation here is intuitively better understandable, and can be extended to other likelihood functions more easily.

Let us give an example of what the PDFs of the individual components of the likelihood functions $L_C(0)$ and $L_C(\mu')$ look like. We restrict ourselves to any piece of random data not overlapping the real sync word $\vec{S} = (1, 1, -1, -1, 1)$ of length $L = 5$. The observation data sequence beginning anywhere at $\mu' \geq 5$ be $(1, 1, 1, -1, 1)$, thus $\kappa_L = 3$ (i.e. one symbol does not match the sync word). In Fig. 4.4 we can see the PDFs of the y_i and C_i for $\mu = \tilde{\mu} = 0$. Similarly, the corresponding PDFs for $\mu = \mu'$ are illustrated in Fig. 4.5.

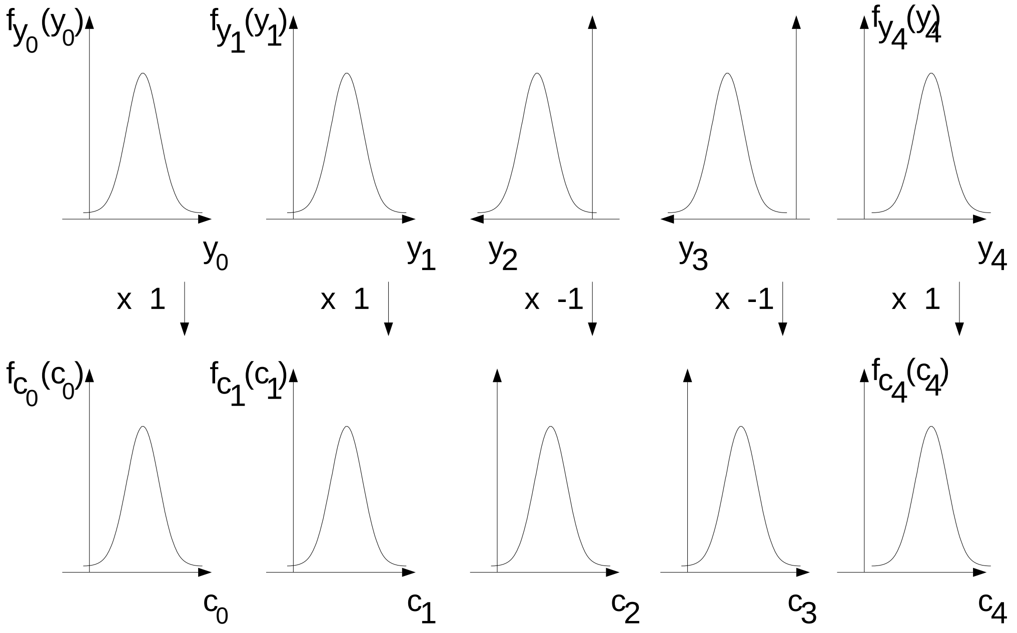


Figure 4.4: *PDFs of the received components and elements of the likelihood function for the real position, $\tilde{\mu} = 0$. The received symbols y_i are Gaussian distributed, the means correspond to the values of $\vec{S} = (1, 1, -1, -1, 1)$.*

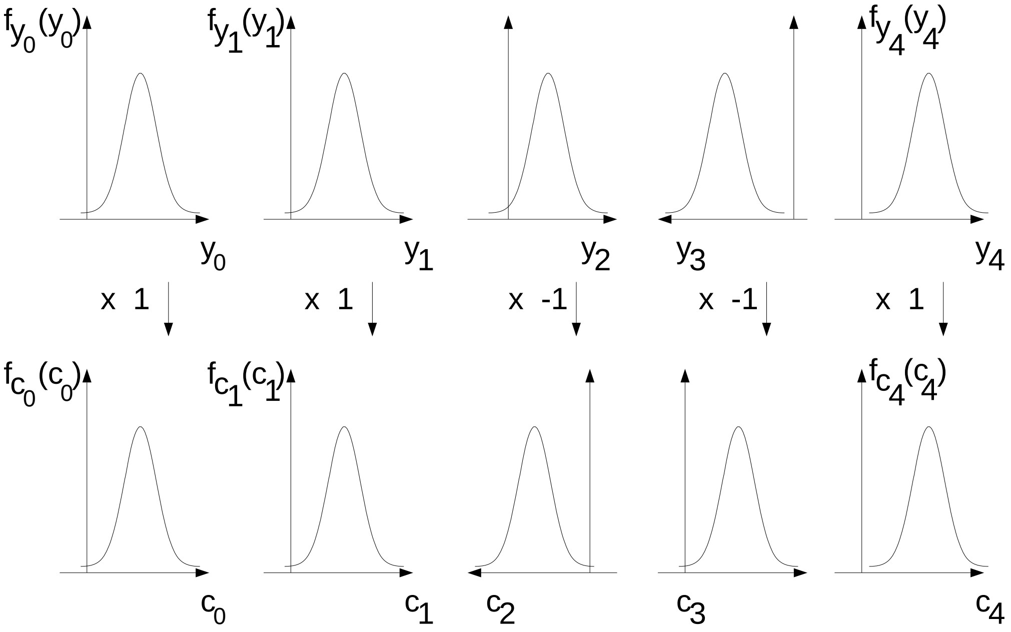


Figure 4.5: PDFs of the received components and elements of the likelihood function for the competing position, $\mu = \mu'$. The received symbols y_i are Gaussian distributed, the means correspond to the values of $(x_{\mu'}, ..., x_{\mu'+4}) = (1, 1, 1, -1, 1)$.

The convolution of all the f_C leads to the PDF of ΔL which has an average of $L - \kappa_L = 5 - 3 = 2$ and variance $2L \cdot N_0/2 = LN_0$. We will later compare this with the corresponding result for the high SNR rule.

It is interesting to shed some light upon the components in the first summation of (4.25) (non overlapping case, the other terms behave similarly). In Fig. 4.6 we have plotted $Pr\{\kappa_L\}$, $\frac{1}{2} \cdot \operatorname{erfc}\left(\frac{L-\kappa_L}{\sqrt{2LN_0}}\right)$ (equation (4.14)) and the product of the two as a function of κ_L. Note that the sync word is longer now: $L = 13$. Equation (4.14) is the probability of sync failure at the position of a data sub-sequence with a particular κ_L, this must be weighted with the probability of that κ_L; we arrive at the product. The greatest contribution to the sync error is from $\kappa_L = 9$ or 11, depending on the SNR. For very high SNR only $\kappa_L = 13$ will contribute, leading approximately[2] to the random data limited bound discussed in section 4.1.

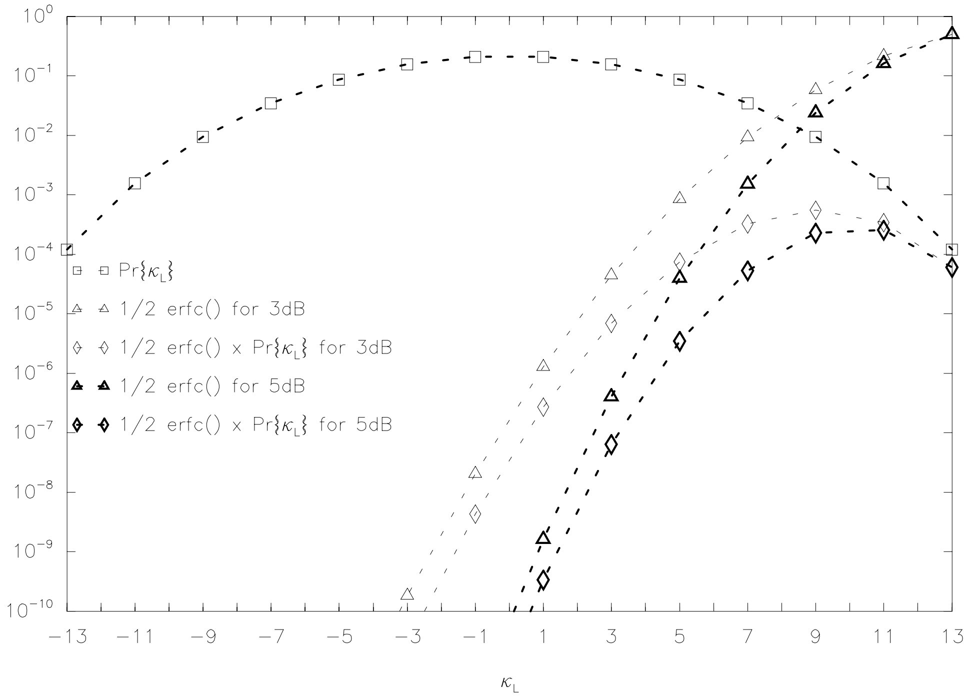


Figure 4.6: *Components of the union bound for the soft correlation rule.* $L = 13$, *no overlapping.* SNR=3 dB *and* 5 dB. *As a function of* κ_L *we see:* $Pr\{\kappa_L\}$, $\frac{1}{2} \cdot \operatorname{erfc}\left(\frac{L-\kappa_L}{\sqrt{2LN_0}}\right)$ *equation (4.14) and the product of the two. Notice at which* κ_L *the greatest contribution to the sync error occurs:* $\kappa_L = 9$ *for* 3 dB, *and* 11 *for* 5 dB.

We can make a simplification to the above union bound by treating the overlap cases in the same way as the 'random data' case. The simplification leads to

$$Pr\{f\} \approx (N-1) \cdot \sum_{\forall \kappa_L} Pr\{\kappa_L\} \cdot \frac{1}{2} \cdot \operatorname{erfc}\left(\frac{L-\kappa_L}{\sqrt{2LN_0}}\right). \tag{4.26}$$

[2] The value of the union bound for infinite SNR is *not* the same as the value of $Pr\{f|\text{RDL}\}$, equation (4.2), since the former does not take into account: 1) the fact that the sync word will never occur again in data partly overlapping itself, 2) the exact probability of choosing the correct position even when two or more repetitions of the sync word occur in the random data.

We have thus ignored the beneficial effect of a low partial auto-correlation of the sync word, $R_{\mu'}$ that helped to keep the second term of (4.25) small (by keeping the argument of the erfc function, $\frac{L-(R_{\mu'}+\kappa_{\mu'})}{\sqrt{2(L-R_{\mu'})N_0}}$, high).

4.2.2 Numerical Upper Bound for the High SNR Rule and Coherent BPSK Signalling

In the last section, we were fortunate in being able to express all the PDFs that have to be convoluted, as Gaussian distributions. This enabled us to restrict the evaluation to a series of table-lookups of the erfc function. However, the high SNR rule which we will investigate in the following leads to PDFs which are non-Gaussian and, therefore, have to be convoluted explicitly. This is a substantial task, even when relying on fast convolution techniques using the fast Fourier transform (FFT). It can be argued that the computation necessary approaches that needed by a Monte-Carlo simulation of the frame synchronizer. Fortunately, as we will see in the next section, one can make use of the central limit theorem when determining the distribution of ΔL. We intend to show that the exact calculation using the correct PDFs produces results very close to the approximation, so that the technique discussed in this section can be neglected in practice.

The derivation of the union bound for the high SNR rule can best be understood if we again pick up our example ($L = 5$) of the previous section. The synchronizer now subtracts the absolute value of $y_{i+\mu}$ from the individual correlation terms,

$$L_H(\mu) = \sum_{i=0}^{L-1} (y_{i+\mu} \cdot S_i - |y_{i+\mu}|) = \sum_{i=0}^{L-1} H_i. \tag{4.27}$$

This non-linear function of $y_{i+\mu}$ influences H_i in the following way:

For all positive values of $y_{i+\mu} \cdot S_i$, H_i will become zero;

for all negative values of $y_{i+\mu} \cdot S_i$: $\quad H_i = 2 \cdot y_{i+\mu} \cdot S_i$.

There are now again two cases to be distinguished (not to be confused with the two cases above!):

1. If $x_{i+\mu} = S_i$, then the main area of $f_{\mathbf{H_i}}(H_i)$ will be concentrated on a δ at zero -since usually $y_{i+\mu} \cdot S_i$ will be positive- weighted by the area of the Gaussian distribution with mean 1 and variance $N_0/2$, right of the y-axis; with only a small negative 'tail' remaining (when $y_{i+\mu} \cdot S_i$ is negative):

$$f^{+}_{\mathbf{H_i}}(H_i) = \left\{ \begin{array}{lcl} \mathcal{N}_{H_i}(2, 2N_0) & : & H_i < 0 \\ 0 & : & H_i \geq 0 \end{array} \right\} + \left(1 - \frac{1}{2}\mathrm{erfc}\left(\frac{1}{\sqrt{N_0}}\right)\right) \cdot \delta(H_i), \tag{4.28}$$

2. If, on the other hand, $x_{i+\mu'} = -S_i$, the situation is reversed and the main area of $f_{\mathbf{H_i}}(H_i)$ will be to the left of the y-axis, and a small δ remains at zero:

$$f_{\mathbf{H_i}}^{-}(H_i) = \left\{ \begin{array}{lcl} \mathcal{N}_{H_i}(-2, 2N_0) & : & H_i < 0 \\ 0 & : & H_i \geq 0 \end{array} \right\} + \left(\frac{1}{2}\mathrm{erfc}\left(\frac{1}{\sqrt{N_0}}\right)\right) \cdot \delta(H_i). \tag{4.29}$$

The sign $\pm$ in $f_{\mathbf{H_i}}$ denotes the sign of the product $x_{i+\mu'} \cdot S_i$. Assuming that $L \leq \mu' \leq N - L$, that is, our 'competing sync word' starting at $\mu = \mu'$ does not overlap the real one, we can express $f_{\mathbf{\Delta L_H(\kappa_L)}}$ as

$$f_{\mathbf{\Delta L_H(\kappa_L)}}(\Delta L_H) = \mathop{*}_{i=0}^{L-1} f_{\mathbf{H_i}}^{+}(H_i) * \mathop{*}_{i=0}^{\frac{L+\kappa_L}{2}-1} f_{\mathbf{H_i}}^{+}(-H_i) * \mathop{*}_{i=0}^{\frac{L-\kappa_L}{2}-1} f_{\mathbf{H_i}}^{-}(-H_i). \tag{4.30}$$

Note that we have denoted L-fold convolution by $\mathop{*}_{i=0}^{L-1}$. The PDFs $f_{\mathbf{H_i}}^{\pm}$ and $f_{\mathbf{\Delta L_H(\kappa_L)}}$ are illustrated in Fig. 4.7. In the example, the distribution $f_{\mathbf{\Delta L_H}}$ is dominated by the distribution of H_2 -corresponding to that position where the data differed from the sync word. The majority of H_i *do not* contribute much to the variance of $L_H(\mu')$.

The overlap cases can be treated similarly, except that again the noise dependencies have to be considered. Since we will not continue using the results of this subsection, we will just give the simplified bound; similar to (4.26),

$$Pr\{f\} \approx \sum_{\forall \kappa_L} Pr\{\kappa_L\} \cdot \lim_{\epsilon \to 0} (N-1) \int_{-\infty}^{-\epsilon} f_{\mathbf{\Delta L_H(\kappa_L)}}(\Delta L_H) d\Delta L_H + \frac{Pr\{\Delta L_H = 0\}}{2}. \tag{4.31}$$

Note that in the evaluation of (4.31), we let a δ in the PDF of ΔL at zero contribute to the integral with half its value. Let us look again at the PDF of ΔL_H, equation (4.30) (third line of Fig. 4.7) over which we must integrate, for $\kappa_L = 3$. Compare this with the corresponding PDF from the previous section, equation (4.13) (bottom line of Fig. 4.7). In our case, $\kappa_L = 3$ means that the data at the competing position comes quite close (before transmission) to the real sync word; four symbols are equal to S_i, one unequal. We notice that for the soft correlation rule, the area to the left of the y-axis is much larger, because the corresponding Gaussian distribution has a larger variance than that of the high SNR rule whereas the means of the distributions are actually identical. For smaller κ_L, the difference in variance is not so pronounced, but these κ_L do not contribute so greatly to the frame error events. It seems that the introduction of the correction term in the high SNR rule protects from synchronization failure events for large, and hence threatening, κ_L. We shall return to the comparison between soft correlation and high SNR rules shortly.

In Fig. 4.8 we have plotted $Pr\{\kappa_L\}$, $\lim_{\epsilon \to 0} \int_{-\infty}^{-\epsilon} f_{\mathbf{\Delta L_H(\kappa_L)}}(\Delta L_H) d\Delta L_H + Pr\{\Delta L_H = 0\}/2$ and the product of the two as a function of κ_L. As before, $L = 13$. The quantity

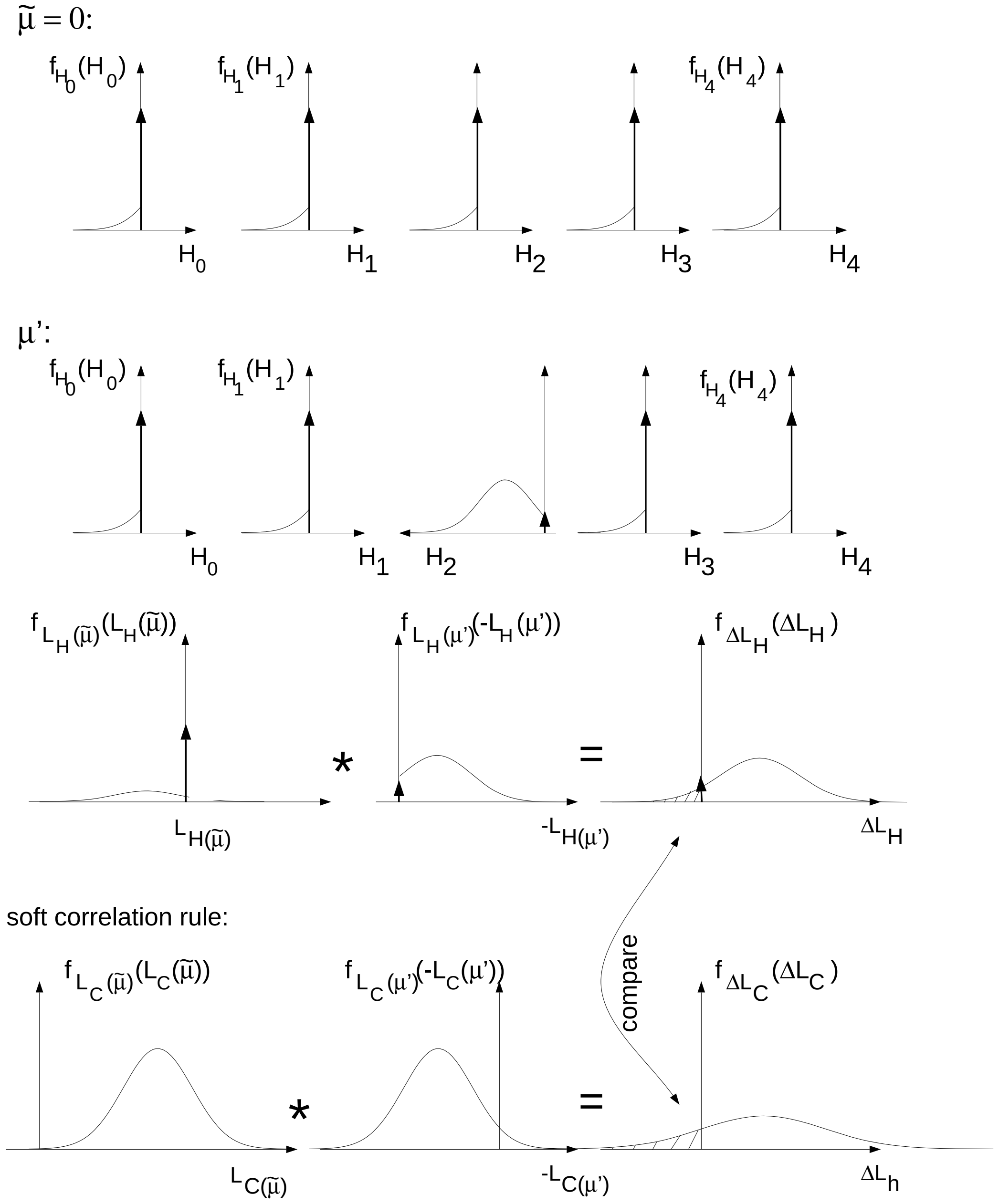


Figure 4.7: *Qualitative illustration of the PDFs of the received components and elements of the likelihood function for the high SNR rule, shown in the same manner in the bottom line are the resulting PDFs for the soft correlation rule. The advantage of the high SNR rule becomes evident: the area (shaded) contributing to a sync failure event is smaller.*

$\int_{-\infty}^{0} f_{\mathbf{\Delta L_H}(\kappa_\mathbf{L})}(\Delta L_H) d\Delta L_H + Pr\{\Delta L_H = 0\}/2$ is the probability of sync failure resulting from a data sub-sequence with a particular κ_L, this is weighted with the probability of that κ_L;

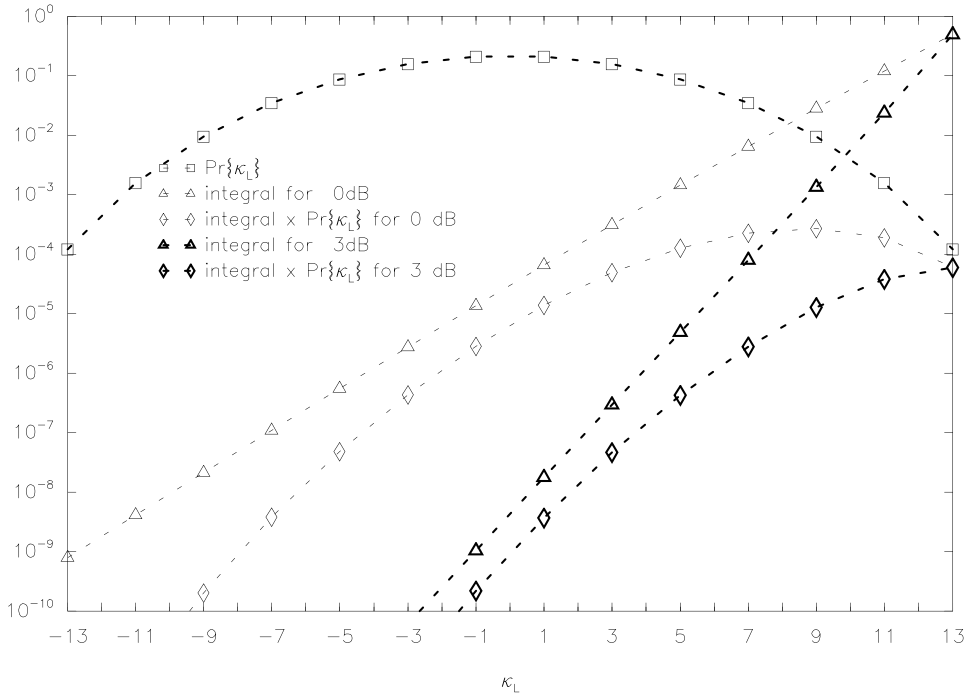


Figure 4.8: *Components of the union bound for the high SNR rule.* $L = 13$, *no overlapping.* SNR=0 dB *and* 3 dB. *As a function of* κ_L *we see:* $Pr\{\kappa_L\}$, $\lim_{\epsilon\to 0} \int_{-\infty}^{-\epsilon} f_{\Delta \mathbf{L_H}(\kappa_\mathbf{L})}(\Delta L_H) d\Delta L_H + Pr\{\Delta L_H = 0\}/2$ *and the product of the two. Notice now at which* κ_L *the greatest contribution to the sync error occurs:* $\kappa_L = 9$ *for* 0 dB, *and* 13 *for* 3 dB.

we arrive at the product. We notice from Fig. 4.8 that the greatest contribution to the sync error is in the case $\kappa_L = 13$ for 3 dB. The high SNR rule has successfully reduced the danger of those sync failure occurrences with $\kappa_L < L$, especially at higher SNR.

Finally, we return to the comparison of the soft correlation rule and the high SNR rule in Fig. 4.9. The product for the soft correlation rule is much higher than for the high SNR rule, leading to a higher synchronization failure probability of the soft correlation rule. For $\kappa_L = 13$ both rules result in the same integral, the difference becomes greater for those threatening κ_L that are more common. This is because the variance of ΔL becomes greater for the correlation rule.

4.2.3 Central Limit Theorem Approximation to the Upper Bound for the High SNR Rule and Coherent BPSK Signalling

We would like to find an approximation to the distribution of ΔL allowing us to integrate over it to determine the probability of a sync failure event. A similar problem was encountered in [Bi83] where the sync failure rate of the threshold frame synchronizer using Massey's high

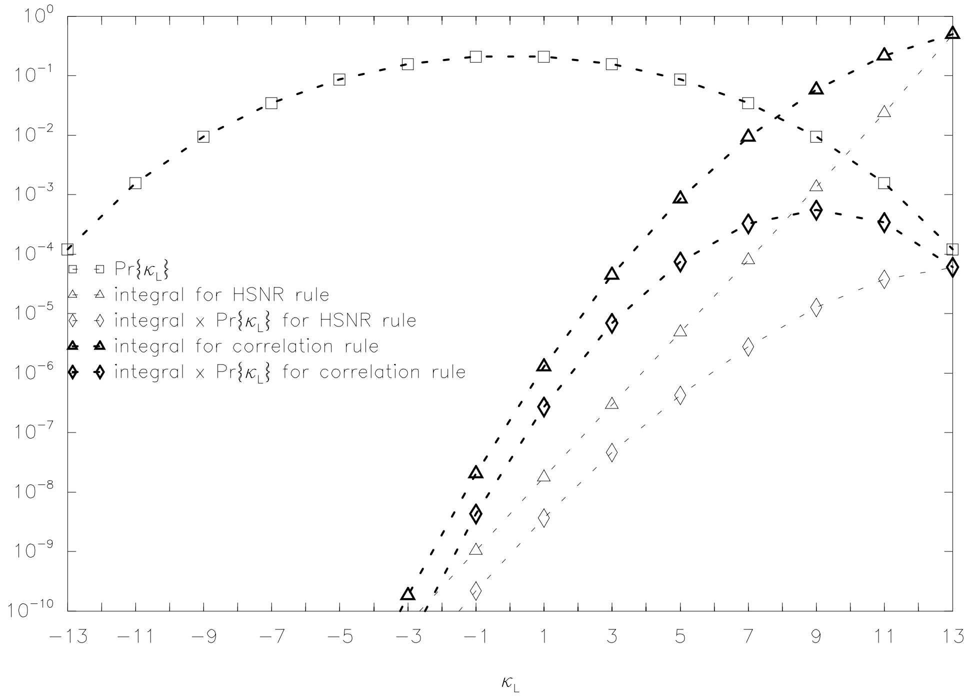


Figure 4.9: *Components of the union bound for the high SNR rule and soft correlation rule. $L = 13$, no overlapping. SNR $= 3$ dB. As a function of κ_L we see: $Pr\{\kappa_L\}$, $\lim_{\epsilon\to 0}\int_{-\infty}^{-\epsilon} f_{\mathbf{\Delta L}(\kappa_{\mathbf{L}})}(\Delta L)d\Delta L + Pr\{\Delta L = 0\}/2$ and the product of the two. Notice the marked improvement of the high SNR rule over the soft correlation rule: those $\kappa_L < 13$ (with a higher probability of occurrence) do not have such a detrimental influence on the sync rate.*

SNR rule instead of the soft correlation rule was analytically approximated. The underlying conjecture is that if the length of the sync word is adequately large, then the distributions of the likelihood functions will become Gaussian (central limit theorem). We adopt a similar strategy, but with some important differences. In particular, we present a simple method that allows us to exactly compute the required means and variances of the (approximately Gaussian) distributions of ΔL with less computational effort. In [Bi83] the means and variances could only be approximated and the resulting expressions were very complex.

4.2.3.1 Non-overlap cases

For simplicity let us first treat the case where $L_H(0)$ and $L_H(\mu')$ are independent, i.e. $L \leq \mu' \leq N - L$. To obtain the variance and mean of $L_H(0)$ and $L_H(\mu')$ which we need in order to determine the mean and variance of ΔL_H, we need means and variances of the components of $L_H(\mu)$: H_i for the two cases $x_{i+\mu} = \pm S_i$. We shall call these m_{H+}, m_{H-}, σ^2_{H+} and σ^2_{H-} respectively.

The mean of $L_H(0)$ is

$$m_{L_H(0)} = L \cdot m_{H+}. \tag{4.32}$$

Similarly, *since all H_i are independent*, the variance of $L_H(0)$ is

$$\sigma^2_{L_H(0)} = L \cdot \sigma^2_{H+}. \tag{4.33}$$

The mean and variance of $L_H(\mu')$ are

$$m_{L_H(\mu')} = \frac{L+\kappa_L}{2} \cdot m_{H+} + \frac{L-\kappa_L}{2} \cdot m_{H-}. \tag{4.34}$$

$$\sigma^2_{L_H(\mu')} = \frac{L+\kappa_L}{2} \cdot \sigma^2_{H+} + \frac{L-\kappa_L}{2} \cdot \sigma^2_{H-}. \tag{4.35}$$

The four necessary means and variances $m_{H\pm}$ and $\sigma^2_{H\pm}$ are calculated in Appendix D. With $m_{L_H(0)}$, $\sigma^2_{L_H(0)}$, $m_{L_H(\mu')}$ and $\sigma^2_{L_H(\mu')}$ we can easily determine the mean and variance of ΔL_H:

$$\begin{aligned} m_{\Delta L} &= L \cdot m_{H+} - \frac{L+\kappa_L}{2} \cdot m_{H+} - \frac{L-\kappa_L}{2} \cdot m_{H-} \\ &= \frac{L-\kappa_L}{2} \cdot m_{H+} - \frac{L-\kappa_L}{2} \cdot m_{H-} \\ &= L - \kappa_L, \end{aligned} \tag{4.36}$$

where we have made use of the fact that $m_{H+} - m_{H-} = 2$ (see Appendix D). Similarly,

$$\sigma^2_{\Delta L} = \sigma^2_{L_H(0)} + \sigma^2_{L_H(\mu')} = \frac{3 \cdot L + \kappa_L}{2} \cdot \sigma^2_{H+} + \frac{L-\kappa_L}{2} \cdot \sigma^2_{H-}. \tag{4.37}$$

4.2.3.2 Overlap cases

Fortunately, the treatment of the overlap cases is quite straightforward. This is because *in the overlap region* the energy correction terms $|y_i|$ comprising $L_H(0)$ and $L_H(\mu')$ are identical and do not affect ΔL_H (they cancel each other out!), only the correlation terms are important -we have treated them in 4.2.1 already. The only correction terms affecting ΔL_H are from *outside the overlap region* and can be treated in the same way as in the previous section. The mean of ΔL_H is thus

$$\begin{aligned} m_{\Delta L} &= \left(\mu' - \frac{\mu'+\kappa_{\mu'}}{2}\right) \cdot m_{H+} - \frac{\mu'-\kappa_{\mu'}}{2} \cdot m_{H-} + \frac{L-\mu'-R_{\mu'}}{2} \cdot 2 \\ &= \frac{\mu'-\kappa_{\mu'}}{2} \cdot m_{H+} - \frac{\mu'-\kappa_{\mu'}}{2} \cdot m_{H-} + L - \mu' - R_{\mu'} \\ &= L - \kappa_{\mu'} - R_{\mu'}. \end{aligned} \tag{4.38}$$

Taken together, the first two terms of the top line comprise correction and correlation terms outside the overlap region: we have simply replaced L in equations (4.32) and (4.34) by μ'. The last term comprises the correlation term inside the overlap region and is taken from the second term of the middle equality in equation (4.20).

A similar argument yields,

$$\sigma^2_{\Delta L} = \frac{\kappa_{\mu'} + 3\mu'}{2} \cdot \sigma^2_{H+} + \frac{\mu'-\kappa_{\mu'}}{2} \cdot \sigma^2_{H-} + N_0(L - \mu' - R_{\mu'}). \tag{4.39}$$

4.2.3.3 The complete approximation

Let us now put the results of the previous two sections together. We have exactly determined mean and variance of ΔL_H, the distribution of which we are now approximating as being Gaussian. To evaluate the union bound (4.3) we approximate

$$Pr\{f\} \le \sum_{\mu'=1}^{N-1} \int_{-\infty}^{0} f_{\mathbf{\Delta L_H}}(\Delta L_H) d\Delta L_H \approx \sum_{\mu'=1}^{N-1} \int_{-\infty}^{0} \mathcal{N}_{\Delta L_H}(m_{\Delta L}, \sigma^2_{\Delta L}) d\Delta L_H, \tag{4.40}$$

and insert equations (4.36), (4.37), (4.38) and (4.39) for the mean and variance that are functions of μ' and, therefore, depend on which of the cases overlap/non-overlap applies.

This, then, is the very important and hitherto unavailable approximate union bound for the synchronization failure rate:

$$Pr\{f\} \approx (N-2L+1) \cdot \sum_{\forall \kappa_L} Pr\{\kappa_L\} \cdot \frac{1}{2} \cdot \text{erfc}\left(\frac{L-\kappa_L}{\sqrt{(3L+\kappa_L)\cdot\sigma^2_{H+} + (L-\kappa_L)\cdot\sigma^2_{H-}}}\right) + \\ \sum_{\mu'=1}^{L-1} \sum_{\forall \kappa_{\mu'}} Pr\{\kappa_{\mu'}\}\text{erfc}\left(\frac{L-(\kappa_{\mu'}+R_{\mu'})}{\sqrt{(3\mu'+\kappa_{\mu'})\sigma^2_{H+} + (\mu'-\kappa_{\mu'})\sigma^2_{H-} + 2N_0(L-\mu'-R_{\mu'})}}\right). \tag{4.41}$$

Let us compare the components of the approximate bound with the correct bound obtained through numerical convolution. They are shown in Fig. 4.10. The deviation from the correct bound becomes larger for decreasing κ_L, but these contribute little to the overall synchronization failure rate. The deviation can be explained as follows: we are not interested in the actual distribution of ΔL_H, but in the area to the left $\Delta L_H = 0$. For $\kappa_L = L$ the distribution will be symmetrical (w.r.t. y axis), so no matter how accurate the central limit approximation may be, the area will be 0.5. Now as κ_L decreases, the contribution to the area to the left $\Delta L_H = 0$ will come more and more from the 'tail' of the distribution of ΔL_H -and exactly here the central limit approximation is less accurate. The deviation also increases with higher signal-to-noise ratios, this is because the PDF $f^+_{\mathbf{H_i}}(H_i)$ becomes more and more a Dirac delta at zero, this in turn reduces the number of 'dominant' (non-delta) PDFs in the convolution reducing the central limit theorem's validity [Pap84]. In Figures 4.14, 4.15 and 4.17 we can see the closeness of the bound for lower sync failure rates.

4.2.4 Union Bound for the Hard Correlation Rule and BPSK

The hard correlation rule's performance has been analyzed by several authors, for example for the threshold algorithm by Scholtz in [Sch80]. We will employ some of the results of this paper to derive a union bound for the maximum search method investigated in this work.

The hard correlation rule differs from the soft correlation rule in that the correlation is performed on hard decisions. We are, therefore, able to use simple combinatoric manipulations to arrive at the desired bound.

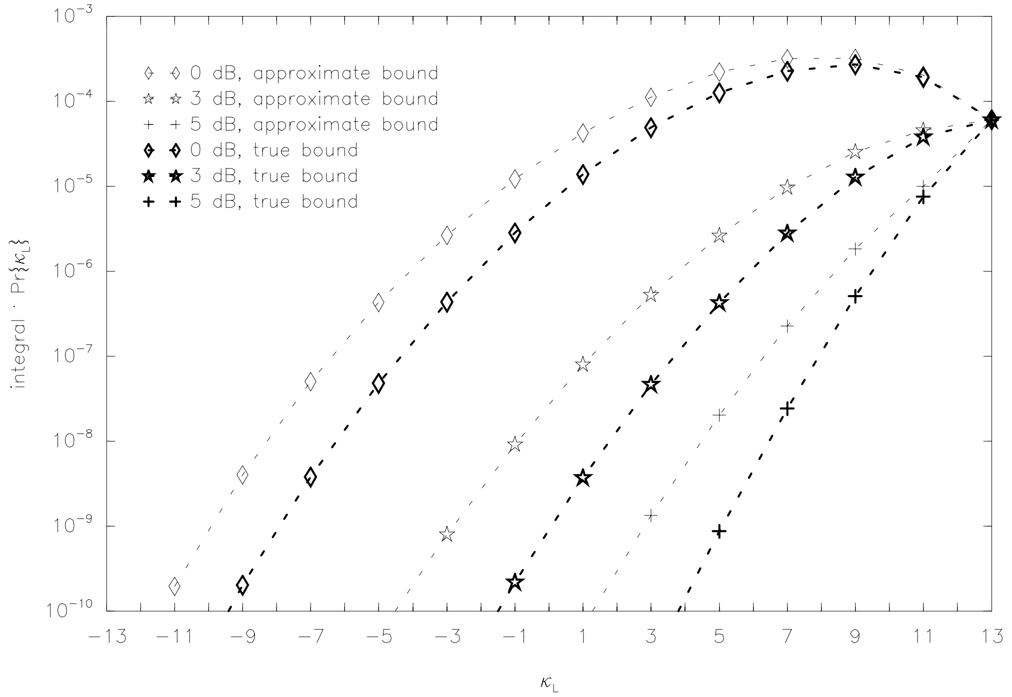


Figure 4.10: *Components of the approximate and true union bounds for the high SNR rule. $L = 13$, no overlapping. SNR = 0, 3 and 5 dB. As a function of κ_L we see $Pr\{\kappa_L\} \cdot \lim_{\epsilon \to 0} \int_{-\infty}^{-\epsilon} f_{\mathbf{\Delta L_H}(\kappa_\mathbf{L})}(\Delta L_H) d\Delta L_H$. The deviation from the correct bound becomes larger for decreasing κ_L, but these contribute little to the overall synchronization failure rate.*

4.2.4.1 Non-overlap cases

A sync error is definitely made if the correlation with the random data is higher than that with the real sync word. The probability of a sync error for any $L \le \mu' \le N - L$ is

$$Pr\{L_C(\mu') > L_C(\tilde{\mu} = 0)\} + \frac{1}{2} \cdot Pr\{L_C(\mu') = L_C(\tilde{\mu} = 0)\} =$$
$$\sum_{i=0}^{L} (1 - p_e)^{L-i} \cdot {p_e}^i \binom{L}{i} \cdot \frac{1}{2^L} \cdot \left(\sum_{k=0}^{i-1} \binom{L}{k} + \frac{1}{2}\binom{L}{i} \right), \qquad (4.42)$$

where we have abbreviated $p_e = \frac{1}{2} \cdot \operatorname{erfc}\left(\frac{1}{\sqrt{N_0}}\right)$ to denote the probability of a channel error (BSC). The first sum (over i) is over all possible number of channel errors in the real sync word. The second sum is over all possible number of sync word disagreements, less than i in number, in the L-long random data portion beginning at μ' -these cases will certainly lead to a sync error. The $\binom{L}{i}$ cases where $k = i$ are treated separately and lead to a sync error of 0.5 (fifty percent chance of still choosing $\hat{\mu} = 0$).

4.2.4.2 Overlap cases

Again these cases have to be treated separately. A similar argument as in section 4.2.1 leads to the following result:

$$
\begin{aligned}
&Pr\{L(\mu') > L(\tilde{\mu}=0)\} + \frac{1}{2}\cdot Pr\{L(\mu') = L(\tilde{\mu}=0)\} = \\
&\sum_{i=0}^{\mu'} (1-p_e)^{\mu'-i}\cdot {p_e}^{i}\binom{\mu'}{i}\cdot \sum_{j=0}^{\frac{L-\mu'-R_{\mu'}}{2}} (1-p_e)^{\frac{L-\mu'-R_{\mu'}}{2}-j}\cdot {p_e}^{j}\binom{\frac{L-\mu'-R_{\mu'}}{2}}{j}\cdot \\
&\cdot\begin{cases}
0 & \text{if} \quad n_r < 0 \\
\frac{1}{2} & \text{if} \quad n_r = \mu' \\
1 & \text{if} \quad n_r > \mu' \\
\frac{1}{2^{\mu'}}\cdot\left[\sum_{k=0}^{n_r-1}\binom{\mu'}{k} + \frac{1}{2}\binom{\mu'}{n_r}\right] & \text{if} \quad 1 \le n_r \le \mu'-1.
\end{cases}
\end{aligned}
\tag{4.43}
$$

The terms can be explained as follows: the first sum is over the possible channel errors in the real sync word in the μ' long non-overlap portion; the second sum is over the possible channel errors in the real sync word in the $L-\mu'$ long overlap portion, but only for those bits where $x_{i+\mu'} = S_{i+\mu'} \neq S_i$: the cases $S_{i+\mu'} = S_i$ do not influence ΔL_C. The final term takes into account all the possibilities of disagreements between the random bits and the sync word, in the μ' random data sub-sequence following the sync word, *that can still lead to a sync error*. The maximum number of such disagreements that can still lead to a sync error is $n_r = i + 2j - \frac{L-R_{\mu'}-\mu'}{2}$. The number n_r can be interpreted as follows: on the '$\tilde{\mu}$ side' of a pair of scales that weighs 'sync word disagreements with hard decisions' (Fig. 4.11), we put i errors that account for the errors in the real received sync word outside of the overlap region; then add to i the number of errors j in the overlap region and where $S_{i+\mu'} = -S_i$. On the 'μ' side' of the pair of scales we have to put the remaining $\frac{L-R_{\mu'}-\mu'}{2} - j$ symbols (in the overlap region, and where $S_{i+\mu'} = -S_i$) that are error-free -they are sync word disagreements when seen from μ'. To the 'μ' side' we finally add the number of disagreements between the random bits and the sync word, in the μ' random data sub-sequence following the sync word: q. The scales will tip to the $\tilde{\mu}$ side or remain horizontal (actual or possible sync error) if $i + j \geq \frac{L-R_{\mu'}-\mu'}{2} - j + q$, i.e. if $q \leq n_r$. Of course, if n_r is less than zero, there will never be a sync error. If n_r is greater than or equal to μ', there might be a sync error, since q will always be less than or equal to μ'.

Equations (4.42) and (4.43) can now be inserted into the basic union bound (4.3).

4.2.5 Phase Ambiguity

Up to now, our bounds have treated coherent detection without phase ambiguity. We can, however, find a simple modification to the preceding bounds for BPSK to take account of

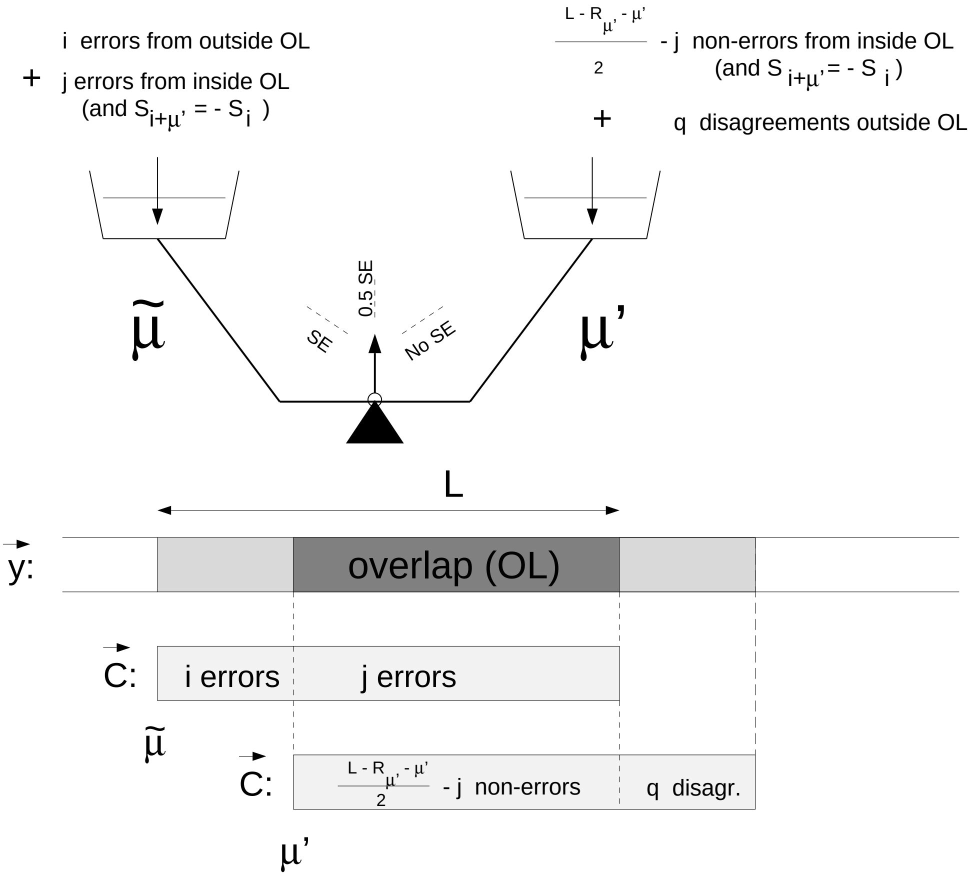


Figure 4.11: *Illustration of the overlap case for the hard correlation rule. On a pair of scales we weigh 'sync word disagreements with hard decisions'. A sync error (SE) occurs if there are more disagreements on the '$\tilde{\mu}$ side' than on the 'μ' side'. If both sides are equal, there is a 50 % chance of a sync error. See text in section 4.2.4.2 for a further description.*

this. The high SNR rule for BPSK with π radians phase ambiguity is (3.29):

$$L_H(\mu) = \left| \sum_{i=0}^{L-1} y_{i+\mu} \cdot S_i \right| - \sum_{i=0}^{L-1} |y_{i+\mu}|. \tag{4.44}$$

The synchronizer effectively computes two correlations, one with the sync word, and one with the inverted sync word. It selects the correlation term with the highest value, i.e. it implicitly decides on the phase error due to the ambiguity. The value of the correction term is unaffected by this process. For large (in magnitude) values of κ_L (and $\kappa_{\mu'} + R_{\mu'}$ in the overlap cases) this selection process can be viewed as being equivalent to κ_L and $\kappa_{\mu'} + R_{\mu'}$ being replaced by $|\kappa_L|$ and $|\kappa_{\mu'} + R_{\mu'}|$, respectively. Let us show this for the case κ_L. If κ_L and hence $\sum_{i=0}^{L-1} y_{i+\mu} \cdot S_i$ is large in magnitude, then

$$\left| \sum_{i=0}^{L-1} y_{i+\mu} \cdot S_i \right| \approx \text{sign} \left(\sum_{i=0}^{L-1} x_{i+\mu} \cdot S_i \right) \cdot \sum_{i=0}^{L-1} y_{i+\mu} \cdot S_i = \text{sign}(\kappa_L) \cdot \sum_{i=0}^{L-1} y_{i+\mu} \cdot S_i \tag{4.45}$$

$$\approx \sum_{i=0}^{L-1} y_{i+\mu} \cdot S'_i, \tag{4.46}$$

where $\vec{S'} = \text{sign}(\kappa_L) \cdot \vec{S}$.

Since κ_L can be positive or negative, we need to examine two cases:

1. $\vec{S'} = \vec{S}$. No change needs to be made in the evaluation of the mean and variance of ΔL.
2. $\vec{S'} = -\vec{S}$. The sync word with which we are 'comparing' the random data sub-sequence is inverted. Hence for each former symbol-similarity with the non-inverted sync word we now have a non-similarity, and vice-versa, i.e. we must replace κ_L by $-\kappa_L$ in the equations describing mean and variance of ΔL.

A similar argument holds for the overlap cases, here it is $\text{sign}(\kappa_{\mu'} + R_{\mu'})$ that determines the sign of $\vec{S'}$.

To modify the bounds (or approximations) we simply need to replace

$$\kappa_L \text{ by } |\kappa_L|, \tag{4.47}$$

and

$$\kappa_{\mu'} + R_{\mu'} \text{ by } |\kappa_{\mu'} + R_{\mu'}|. \tag{4.48}$$

When $R_{\mu'}$ or $\kappa_{\mu'}$ stand alone, they will just be multiplied by $\text{sign}(\kappa_{\mu'} + R_{\mu'})$ -the receiver's estimate of the phase error $(\hat{\phi}_a)$ due to phase ambiguity. We have assumed that κ_L and $\kappa_{\mu'} + R_{\mu'}$ are large: the approximation (4.45) will not hold otherwise. If, on the other hand, they are small, then the chance of a sync error will be very small anyhow, as is evident from Figs. 4.6 and 4.8. Hence, the approximate bound for phase ambiguity should be quite accurate.

4.2.6 Extension to Other Modulation Formats

The union bound for the soft correlation rule is, in fact, correct for PSK in general, if the energy of a symbol E_s is normalized to 1. It is easily extended to QAM (in fact only the sync word energy has to be determined) [LT87].

However, if the modulation format is two dimensional (complex), the partial auto-correlation function of the sync word at shift μ' must now be defined as

$$R_{\mu'} = \sum_{i=0}^{L-1-\mu'} \langle S_i, S_{i+\mu'} \rangle. \tag{4.49}$$

Similarly, $\kappa_{\mu'} = \sum\limits_{i=0}^{\mu'-1} \langle S_i, x_{i+\mu'} \rangle$; the PDF of which is given in Appendix C, for MPSK.

The approximate union bound for the high SNR rule is also valid for QPSK, as becomes clear when inspecting the likelihood function (equation (3.13) for MPSK),

$$L_H(\mu) = \sum_{i=0}^{L-1} \langle y_{i+\mu}, S_i - W_{\hat{\jmath}(i+\mu)} \rangle, \tag{4.50}$$

where $\hat{\jmath}(i+\mu)$ is that j which maximizes $\langle y_{i+\mu}, W_j \rangle$. For QPSK, (4.50) becomes

$$\begin{aligned} L_H(\mu) = \sum_{i=0}^{L-1} (&\mathrm{Re}\{y_{i+\mu}\} \cdot \mathrm{Re}\{S_i\} + \mathrm{Im}\{y_{i+\mu}\} \cdot \mathrm{Im}\{S_i\} \\ &-|\mathrm{Re}\{y_{i+\mu}\}| - |\mathrm{Im}\{y_{i+\mu}\}|) \,. \end{aligned} \tag{4.51}$$

Hence the approximation for the bound (central limit theorem) can be applied to both real and imaginary components, paying attention that m_{H+} and m_{H-} are to be multiplied by $1/\sqrt{2}$, since real and imaginary symbol components share the total symbol energy; whereas σ^2_{H+} and σ^2_{H-} are unchanged, although they enter the bound twice -for real and imaginary components. For example, $m_{L_H(\mu')}$ in the non-overlap case becomes

$$\begin{aligned} m_{L_H(\mu')} = &\left\{ \frac{L + 2\kappa_L^r}{2} + \frac{L + 2\kappa_L^i}{2} \right\} \cdot \frac{m_{H+}}{\sqrt{2}} \\ &+ \left\{ \frac{L - 2\kappa_L^r}{2} + \frac{L - 2\kappa_L^i}{2} \right\} \cdot \frac{m_{H-}}{\sqrt{2}}, \end{aligned} \tag{4.52}$$

where $\kappa_L^r = \sum\limits_{i=0}^{L-1} \mathrm{Re}\{S_i\} \cdot \mathrm{Re}\{x_{i+\mu'}\}$, and $\kappa_L^i = \sum\limits_{i=0}^{L-1} \mathrm{Im}\{S_i\} \cdot \mathrm{Im}\{x_{i+\mu'}\}$, so $\kappa_L = \kappa_L^r + \kappa_L^i$. Therefore,

$$m_{L_H(\mu')} = \frac{L + \kappa_L}{2} \cdot 2 \cdot \frac{m_{H+}}{\sqrt{2}} + \frac{L - \kappa_L}{2} \cdot 2 \cdot \frac{m_{H-}}{\sqrt{2}}. \tag{4.53}$$

Applying these and similar modifications to the approximate union bound (4.41), results in no change other than those necessary in the evaluation of $R_{\mu'}$ and the PDF of $\kappa_{\mu'}$ (for QPSK). Other modulation types (e.g. 8PSK, QAM) would require new analysis of the distributions of correlation term minus the energy correction term *for all possible deviations of the transmitted symbol from the sync word symbol.* Furthermore, the extension to QPSK does not include phase ambiguity, this case remains an interesting exercise for the ardent reader.

4.3 Performance of the List Synchronizer

We defined the list synchronizer to be a frame synchronizer that supplies the ν best frame starting positions $(\mu_1, ..., \mu_\nu)$ to either another (more reliable) frame synchronizer or a decoding unit, that is able to select the correct position from the list. Let the probability that the correct frame starting position, $\tilde{\mu}$ is *not* in the ν-long list supplied by the list frame synchronizer be $Pr\{f\}(\nu)$. $Pr\{f\}(\nu)$ is a monotone decreasing function with ν: in other words, the list synchronizer will become more and more reliable the longer the list is.

4.3.1 Performance of ML, High SNR and Correlation Rules in a List Synchronizer for the Noiseless Case

We extend the results of section 4.1 by denoting the probability of correct synchronization in the noiseless case ($Pr\{f\}(\nu)$ for infinite SNR) by $Pr\{f|\mathrm{RDL}\}(\nu)$. In Appendix B it is shown that

$$Pr\{f|\mathrm{RDL}\}(\nu) = 1 - \sum_{j=0}^{Q} \min(1, \frac{\nu}{j+1}) D_j M^{-(N-L)}, \tag{4.54}$$

where $Q = \lfloor (N-L)/L \rfloor$. The value D_j is the number of possible observation data sequences in which exactly j occurrences of the sync word occur. The term $M^{-(N-L)}$ is simply the probability of any one observation data sequence, and $\min(1, \frac{\nu}{j+1})$ is the probability that the correct position is in the set of ν positions chosen out of $j+1$ competing positions. Note that in the case of $j+1 < \nu$, the correct position will always be in the list. D_j can be defined recursively by

$$D_j = \binom{N-L-(L-1)j}{j} M^{(N-L-Lj)} - \sum_{i=j+1}^{Q} D_i \binom{i}{j}, \tag{4.55}$$

with

$$D_Q = \binom{N-L-(L-1)Q}{Q} M^{(N-L-LQ)}. \tag{4.56}$$

Because numerical inaccuracies were observed when evaluating (4.55) and (4.54), we rewrite

$$\frac{D_j}{M^{N-L}} = \binom{N-L-(L-1)j}{j} M^{-Lj} - \sum_{i=j+1}^{Q} \frac{D_i}{M^{N-L}} \binom{i}{j}, \tag{4.57}$$

which can be directly inserted into (4.54).

In Fig. 4.12 we see the values of $Pr\{f|\mathrm{RDL}\}$ as a function of N, L and ν for $M = 2$.

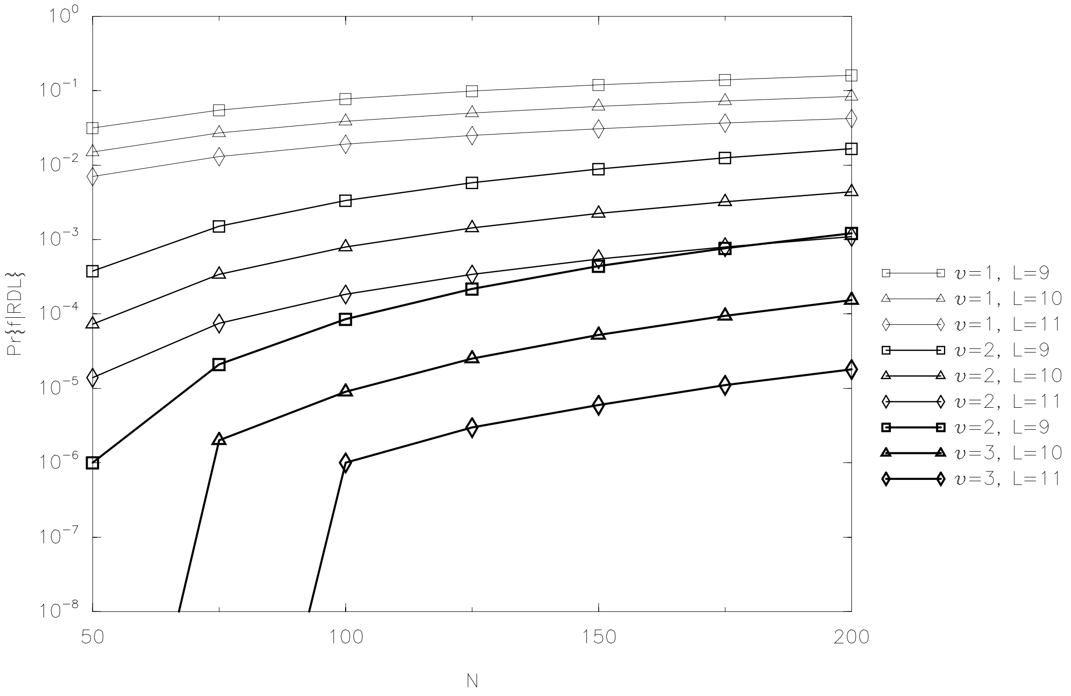


Figure 4.12: Values of $Pr\{f|\text{RDL}\}$ as a function of L, N and ν, $M = 2$

4.3.1.1 Applications of list synchronizers

As noted earlier, the price to pay for this improved sync performance is the requirement of overhead in a subsequent processing stage. This could be a channel decoder; or a source decoder that checks the validity of the final data (perhaps using some kind of extrapolation from previous data, e.g. in voice transmission); or a supervising frame sync unit that makes majority decisions based on many N length observations. Ideally, this subsequent stage would perform optimally in all cases, i.e. it would never declare a frame correctly synchronized to be false and vice-versa. In real applications when for instance the next stage is a decoder, decoding errors lead to these situations. In the following, we will neglect the possibility of incorrectly synchronized frames being declared correct.

Let $Pr\{\text{EC}\}$ denote the probability of declaring a correctly synchronized frame incorrect (e.g. frame decoding error), note that $Pr\{\text{EC}\}$ is relevant only for our discussion of the list synchronizer. Assuming $Pr\{\text{f}\}(\nu)$ and $Pr\{\text{EC}\}$ to be independent, we can express the probability that any one frame will eventually not be received correctly, $Pr\{\text{FE}\}$, to be

$$Pr\{\text{FE}\} = 1 - [(1 - Pr\{\text{EC}\})(1 - Pr\{\text{f}\}(\nu))] . \tag{4.58}$$

We can also calculate the average number of activations of the subsequent processing stage, denoted by $\bar{n_s}$,

$$\begin{aligned}\bar{n_s} = {} & \nu Pr\{\text{EC}\} + (1 - Pr\{\text{EC}\})\,\{\nu Pr\{\text{f}\}(\nu) + (1 - Pr\{\text{f}\}(1)) + \\ & 2\,([1 - Pr\{\text{f}\}(2)] - [1 - Pr\{\text{f}\}(1)]) + ... +\end{aligned}$$

$$\begin{aligned}&\nu\left([1-Pr\{\mathrm{f}\}(\nu)]-[1-Pr\{\mathrm{f}\}(\nu-1)]\right)\}\\&=\nu-\left[(1-Pr\{\mathrm{EC}\})\sum_{i=1}^{\nu-1}(1-Pr\{\mathrm{f}\}(i))\right].\end{aligned}\tag{4.59}$$

For example, let the subsequent processing stage be a Viterbi decoder followed by a block decoder which we presume can detect errors in the Viterbi decoder's output (for instance a CRC code), see Fig. 4.13. $Pr\{\mathrm{EC}\}$ is the probability of at least one Viterbi decoding error occurring in a block and can be upper bounded by

$$Pr\{\mathrm{EC}\} \leq 1-\left[1-\sum_{d=d_{min}}^{\infty} c_d P_d\right]^{R(N-L)}, \tag{4.60}$$

[Hag88], where d_{min} and the c_d's are given by the convolutional code with the code rate R [CC81]. P_d is equal to $\frac{1}{2}\mathrm{erfc}\sqrt{dE_s/N_0}$. For estimates of $Pr\{\mathrm{f}\}(\nu)$ we can use simulation results. The number $\bar{n_s}$, of course, is the average number of decodings per frame.

4.4 Choice of Sync Words

In this section we will briefly mention the criterion used to select good sync words. For a more detailed discussion of this topic the reader is referred to [Sch80], which is good review of the literature covering this topic.

Usually, techniques such as the minimax or min-average method are employed to select sync words. These involve finding a function that is related to the synchronization failure probability in the portion where data and sync word overlap [Wil62] [MS64] [NH71] [Tur68]. The measure used can be the value of the partial auto-correlation function [Lin75], which should be as small as possible (and close to zero if phase ambiguities are expected after demodulation). In Barker's pioneering work [Bar53], sync words (Barker sequences) of lengths $L = 2, 3, 4, 5, 7, 11, 13$ were given that have partial auto-correlation functions always less or equal to one in magnitude, i.e. $|R_\mu| \leq 1\ ,\ \forall\mu\ :\ 1 \leq \mu \leq L-1$. Storer and Turyn showed that there exist no other Barker sequences of odd lengths [TS61], and there are constraints on even lengths that indicate that other even Barker sequences are unlikely [ZG90]. Note here that we should not confuse those sequences with good *partial* auto-correlation, with sequences having a good *cyclic* auto-correlation function, such as those used in DS/PN modulation.

The result of the search for binary sync words for channels without phase ambiguity by Maury and Styles for lengths $L = 7$ to 30 and with phase ambiguity by Turyn for $L = 7$ to 34 are given in tables E.1 and E.2 in Appendix E. Multiphase markers called generalized Barker sequences have better partial auto-correlation properties than binary sync words of the same length due to greater flexibility in design, they can be used in the case of higher order modulation [GS65]

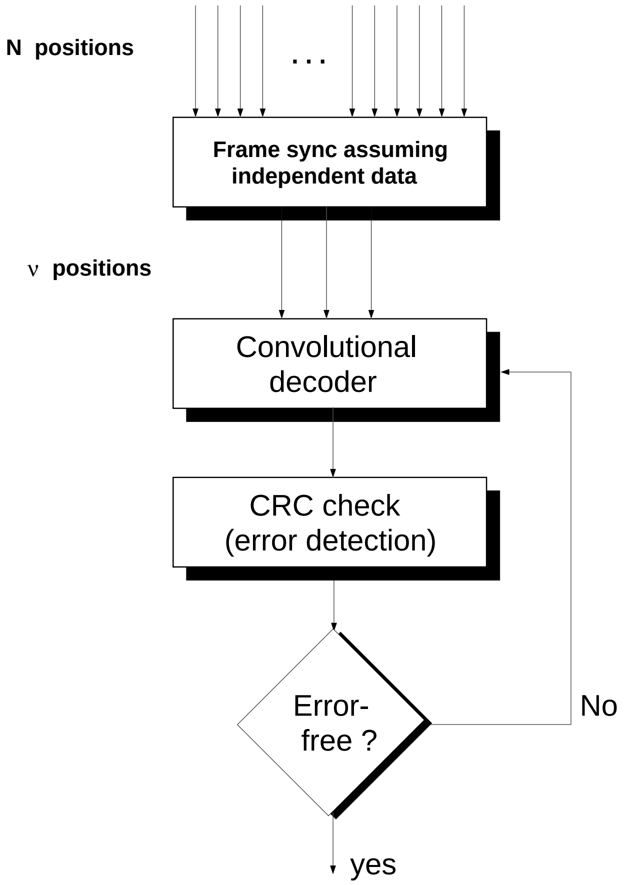

N positions
...
Frame sync assuming independent data
ν positions
Convolutional decoder
CRC check (error detection)
Error-free ?
No
yes

Figure 4.13: *Example of a list synchronizer in conjunction with error correction and detection coding. The frame synchronizer, suboptimal since the frame data sequence, $\vec{d}^F$, is coded, provides ν 'candidate' positions $\mu_1, ..., \mu_\nu$ to a convolutional decoder and subsequent CRC check. The data is decoded up to ν times, until the CRC check is successful.*

[Tur74], an algorithm for finding M-phase Barker sequences is given in [ZG90]. In practice, however, one can choose the sync word from a subset of the modulation symbols, for instance use a binary sync word in a QPSK modulation.

In section 9.5, we shall use the approximate union bound for the high SNR rule and packet transmission as an optimality criterion for developing sync words; we will see that in this case, the values of the side-lobes of the auto-correlation function are themselves not the only governing parameters, but additionally their exact position.

4.5 Monte Carlo Simulation Results

In order to compare the frame synchronization performance of the different synchronization rules and to confirm the accuracy of the upper bounds and approximations, we performed simulations of the frame synchronization process. In the case of traditional frame synchronization (and no trellis termination information is used, see Fig. 3.7) we can rely on the fact that correct results are obtained when restraining $\tilde{\mu}$ to 0 (for reasons of symmetry, see also 4.2.1). So only one frame has to be randomly generated, with the sync word at the correct position. This frame is perturbed by additive white Gaussian noise (preceded by fading if necessary). The 'receiver' then evaluates the likelihood function for all potential frame starting positions, $0 \leq \mu \leq N-1$. The position with the highest likelihood function is chosen. The list synchronizers are implemented by sorting the ν highest likelihood functions and their corresponding μ's; subsequent synchronizers (or decoding processes) operate on this reduced list of positions. Care was exercised to assure a sufficient number of simulation trials, and to use a reliable random number generator[3]. The latter problem is not at all insignificant; to quote [PFTV88]: *If all scientific papers whose results are in doubt because of bad rand()s* (random number generating subroutines, the author) *were to disappear from library shelves, there would be a gap on each shelf about as big as your fist.* In all simulations for BPSK (and analytical bounds) we have used the sync words given in Appendix E in Tables E.1 and E.2 for no phase ambiguity and π radians phase ambiguity, respectively.

4.5.1 Uncoded Coherently Demodulated Frames Transmitted over an AWGN Channel

We will first look at the performance of the ML, high SNR and correlation rules, for several different L, N and M. The results are identical to those found in [LT87] -we have chosen the same parameters. Fig. 4.14 shows the result for $L = 7$, $N = 35$ and $M = 2$ (BPSK). One can see the low maximum synchronization rate reached for infinite SNR, this is due to $Pr\{f|\text{RDL}\} = 0.0835$, compared with Fig. 4.1; indeed, the sync word length $L = 7$ would not generally be used in practice, as it is too short. Furthermore, the soft correlation rule performs several dB worse than the other rules, and surprisingly, the hard correlation rule lies between the soft correlation rule and the high SNR approximation of the ML rule for high SNR, but is the worst choice for very low SNR. What is the reason for this? In Chapter 3 we have given an interpretation for why the soft correlation rule is worse than the ML rule: the missing correction term that corrects differences in the energy of the received symbols. The *hard* correlation rule does not suffer from the problem of 'over-weighting' higher energy components, since it operates on hard decisions. On the other hand, if the SNR is extremely low, then soft correlation actually becomes the optimal sync word search strategy (the noise dominates over the observation data portion) and the correction term is of little importance: in this case the ML rule and soft correlation rule perform almost identically. The ML rule is only marginally better than the high SNR approximation -a very important fact from a practical point of view. In the following, we will not always give the results for all types of

[3] We chose the `ran1` function of [PFTV88]

synchronizers, but will often omit the ML rule for the above reason. For later comparison

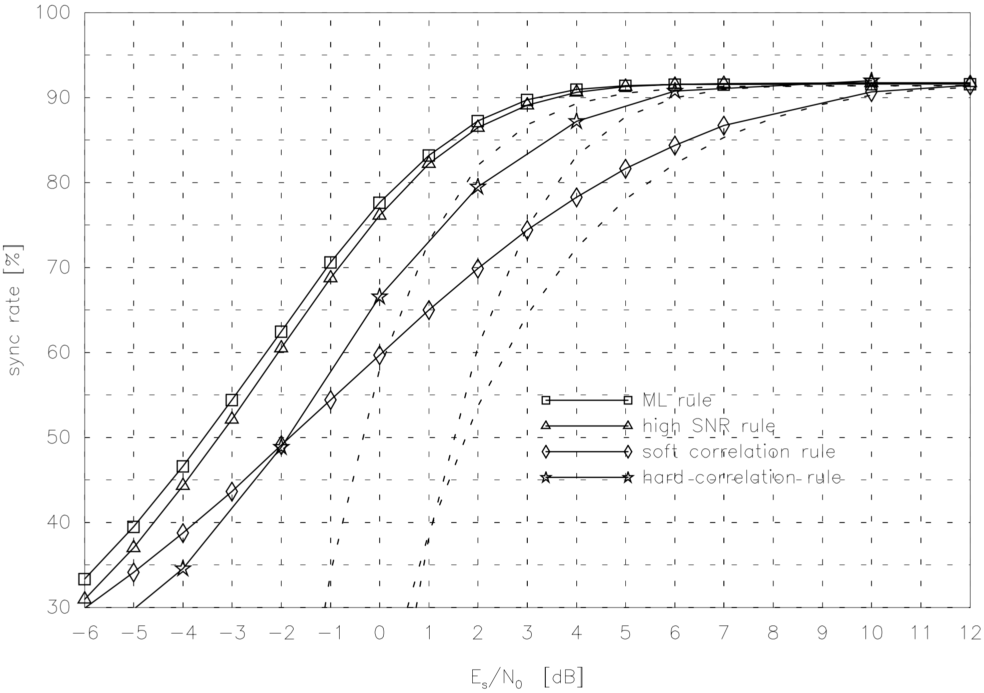


Figure 4.14: *Simulated frame synchronization performance. $L = 7$, $N = 35$, BPSK. We compare the ML, high SNR and correlation rules. The dotted lines show the results of the bounds (approximation for high SNR rule).*

with the analytical solutions of the previous sections, we include another example for BPSK modulation and larger values of N and L (Fig. 4.15). From these figures we can see that our bounds (or approximation) are quite accurate for larger SNR (typical for a union bound).

4.5.2 Simulation of Synchronizers with Differential BPSK

In Fig. 4.16 we show the sync failure rate of the high SNR, soft- and hard-correlation rules for DBPSK (random data). The length of the frame data sequence is again $N - L = 117$, and the sync word length is $L = 13$. There is approximately a 3 dB penalty to be paid compared to coherent demodulation, but only for the high SNR and hard correlation rules, the soft correlation rule suffers a higher penalty. It seems that the energy correction term becomes even more important when using differential demodulation than with coherent demodulation.

4.5.3 Variation of the Sync Word Length

In Fig. 4.17 we see the sync failure rate of the high SNR rule for BPSK (random data) without phase ambiguity and $N - L = 117$, for various sync word lengths L. One can achieve a marked

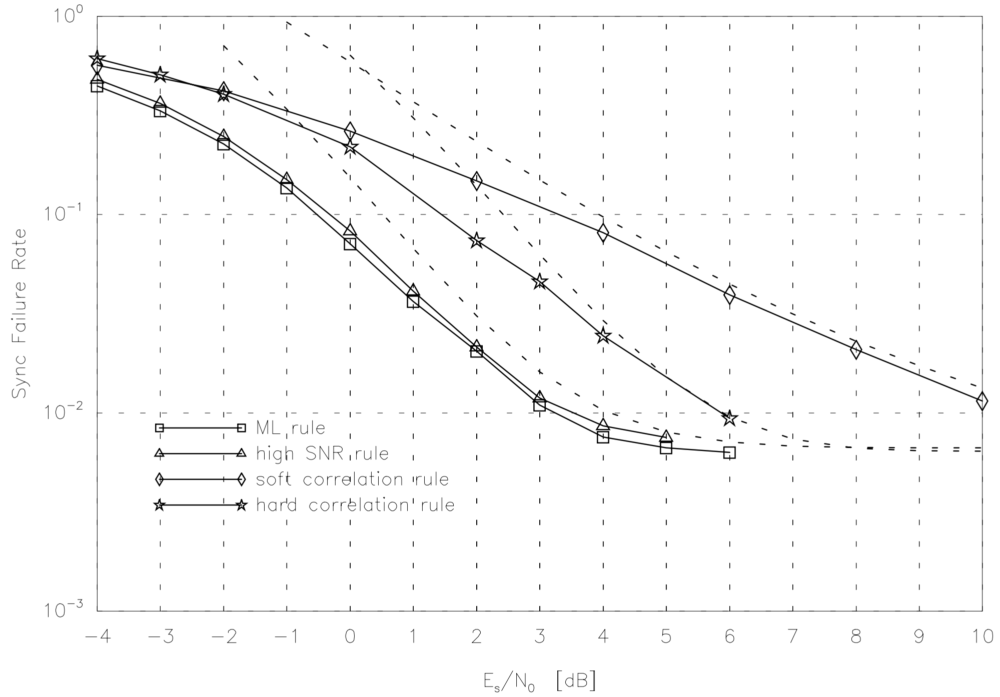


Figure 4.15: *Simulated frame synchronization performance.* $L = 13$, $N = 130$, *BPSK. We compare the ML, high SNR and correlation rules. The dotted lines show the results of the bounds (approximation for high SNR rule).*

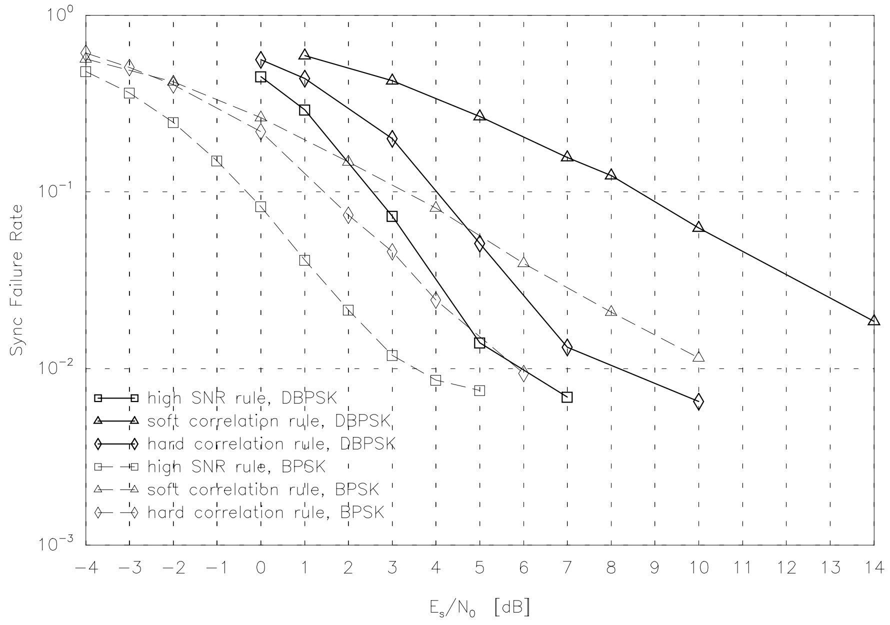


Figure 4.16: *Simulation of the sync failure rate for DBPSK. Parameters:* $N - L = 117$, $L = 13$; *random data. There is approximately a 3 dB penalty to be paid compared to coherent demodulation, but only for the high SNR and hard correlation rules, the soft correlation rule suffers a higher penalty.*

synchronization improvement by only increasing the sync word slightly. The deviation of the bounds for larger L is due to the approximate bound not being equal to $Pr\{f|\text{RDL}\}$ for infinite SNR.

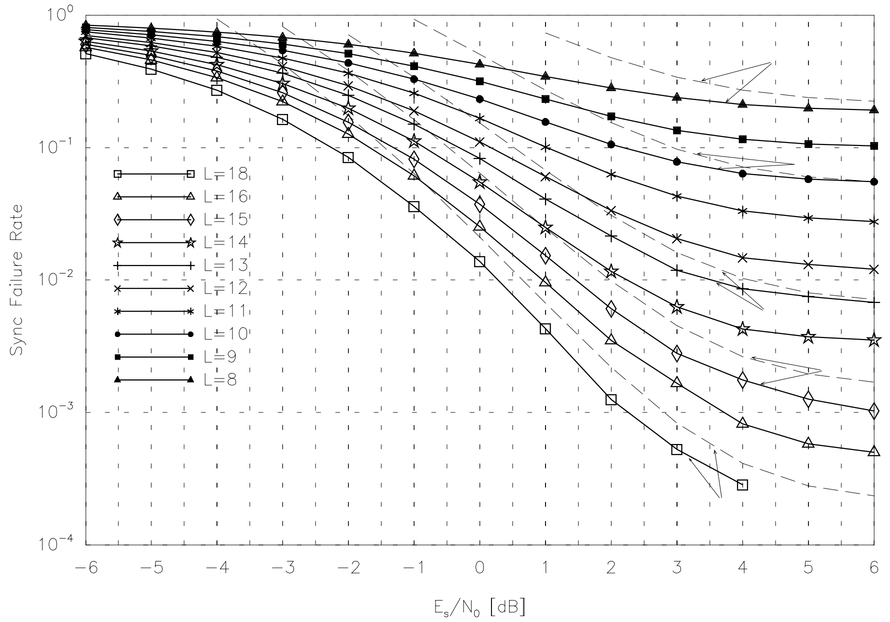


Figure 4.17: *Sync failure rate of the high SNR rule for various sync words lengths L. Parameters: $N - L = 117$, BPSK (random data). No phase ambiguity after demodulation. Thick lines are simulations, dashed lines show the approximate union bound that appears to deviate more from the simulations at higher SNR and for larger L. One can achieve a marked synchronization improvement when only increasing the sync word slightly.*

4.5.4 Simulation of Synchronizers with Phase Ambiguity

In Fig. 4.18 we show the sync failure rate of the high SNR rule for BPSK (random data) with π radians phase ambiguity and $N - L = 117$, for various sync word lengths L. There is approximately a factor of two between these results and those shown in Fig. 4.17.

4.5.5 The List Synchronizer

4.5.5.1 Simulation results of the basic list synchronizer

Before we discuss applications of the list synchronizer in more detail, we present simulation results and evaluation of the bound for the noiseless case (4.54). The likelihood function used in the simulations is the high signal-to-noise ratio rule for BPSK.

Our measure of performance is the probability that the correct sync word position does not correspond to the best ν likelihood functions: $Pr\{\mathrm{f}\}(\nu)$. In Fig. 4.19 we see the performance using BPSK modulation on the AWGN channel with frame length $N = 42$. Note that the gain is especially high for higher SNR, this is because of the decrease in $Pr\{f|\mathrm{RDL}\}(\nu)$, for larger ν. Similar results are obtained for larger N as can be observed in Fig. 4.20. The greatest improvement is achieved when ν is increased from 1 to 2.

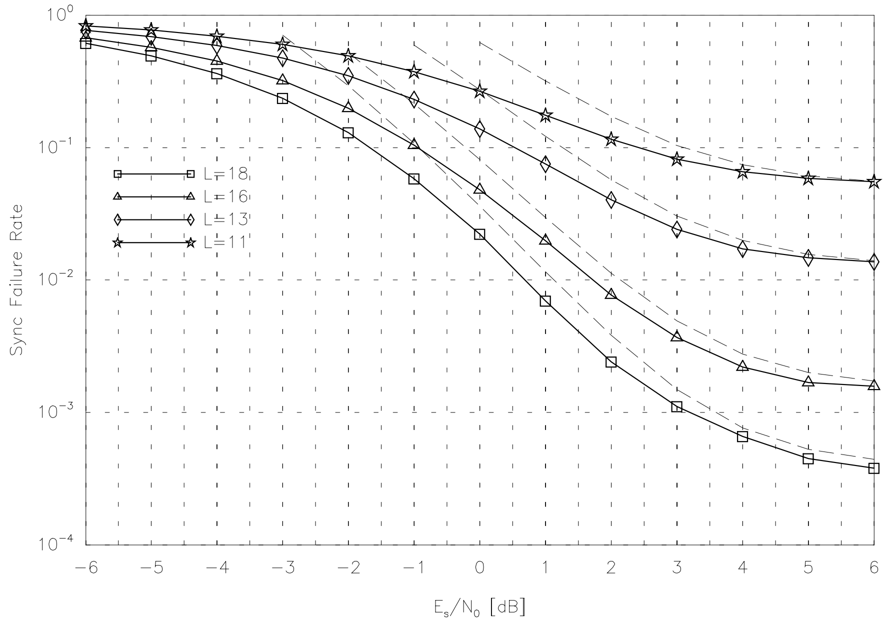


Figure 4.18: *Sync failure rate of the high SNR rule for various sync words lengths L. Demodulation with remaining π radians phase ambiguity. Parameters: $N - L = 117$, BPSK (random data). Thick lines are simulations, dashed lines show the approximate union bound that appears to deviate more from the simulations at higher SNR and for larger L. There is approximately a factor of two between these results and those shown in Fig. 4.17*

4.5.5.2 Concatenation of frame synchronizer and Viterbi decoder and error detection decoding

As an example for a list synchronizer that is followed by a decoder, let us consider the concatenation of frame synchronizer, convolutional decoder and error detection decoder (that is assumed perfect). The performance of such a system using a memory 6, rate $\frac{1}{2}$ code with the octal generator matrix $\binom{133}{171}$ is shown in Fig. 4.21. The values of $Pr\{\mathrm{f}\}(\nu)$ used in the calculations are taken from the simulation results shown in Fig. 4.20. We observe that the benefit of the combined system becomes smaller as the decoder produces more errors (low SNR). The upper bound (4.60) is not very tight for low E_s/N_0, thus $Pr\{\mathrm{FE}\}$ and $\bar{n_s}$ are not very accurate in this region. Note that the average decoding overhead $\bar{n_s}$ (see Fig. 4.22) becomes very small when the performance improves (high SNR). The value of $\bar{n_s}$ could be significant when designing systems with frame processing spread amongst several processors, or when power consumption of the receiver plays a role. The maximum number of decodings can, of course, reach ν per frame.

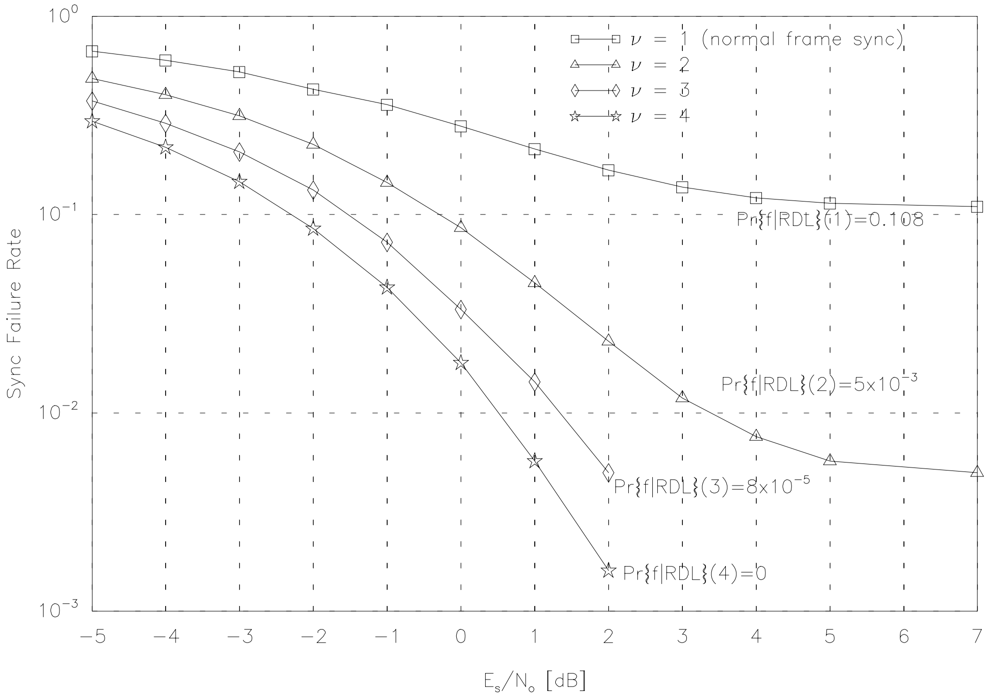


Figure 4.19: *Simulated frame synchronization performance of the list synchronizer with short frames. BPSK modulation,* $N = 42$*,* $L = 7$*. Note the dramatic gain from using* $\nu \geq 2$ *especially for high SNR, this is linked to the decrease in* $Pr\{f|\mathrm{RDL}\}$.

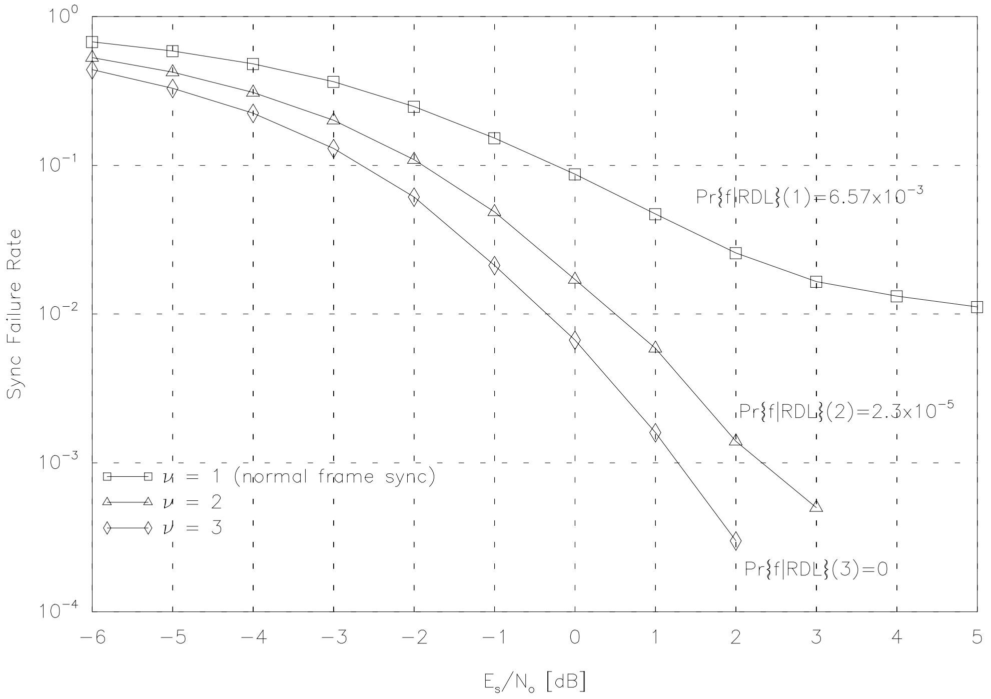


Figure 4.20: *Simulated synchronization performance of the list synchronizer with longer frames. BPSK modulation,* $N = 133$*,* $L = 13$*. Again note the large gain from using* $\nu \geq 2$ *especially for high SNR, this is linked to the decrease in* $Pr\{f|\mathrm{RDL}\}$.

4.5.6 The Synchronizer using Trellis Termination

In the following, we present Monte-Carlo simulations of the synchronizer that uses both trellis opening and termination information to aid performance. Two frames are randomly generated

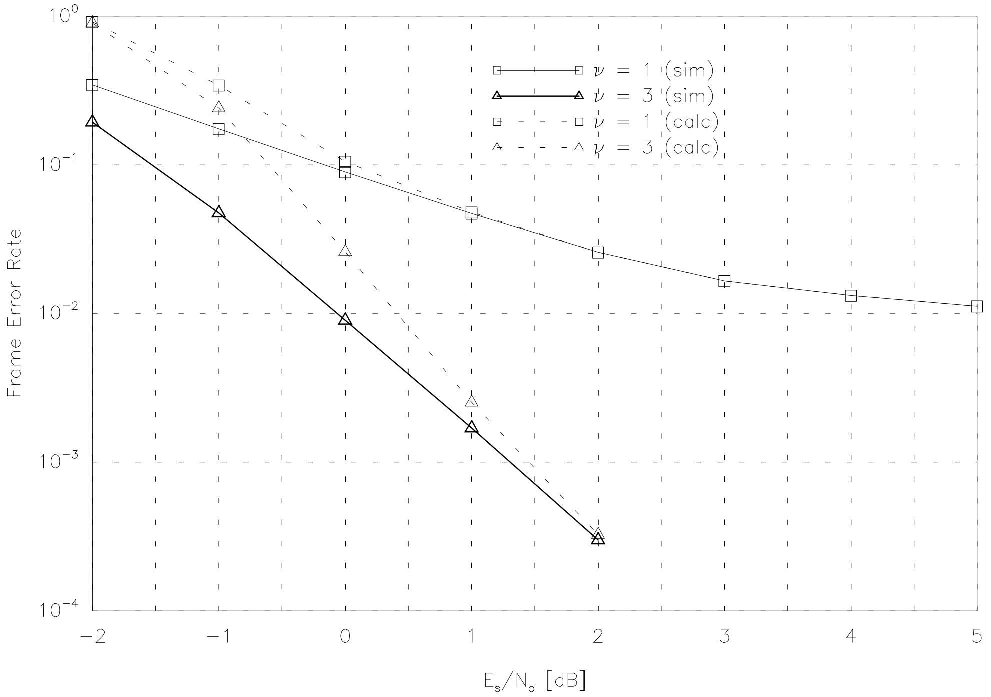


Figure 4.21: *Total frame error rate of a system using a list synchronizer. BPSK modulation, $N = 133$, $L = 13$, code rate $R = 1/2$, memory 6. We see both simulations of the complete system as well as evaluation of (4.58). Note how for small SNR the list synchronizer provides no advantage as the decoder failures dominate over the synchronization errors.*

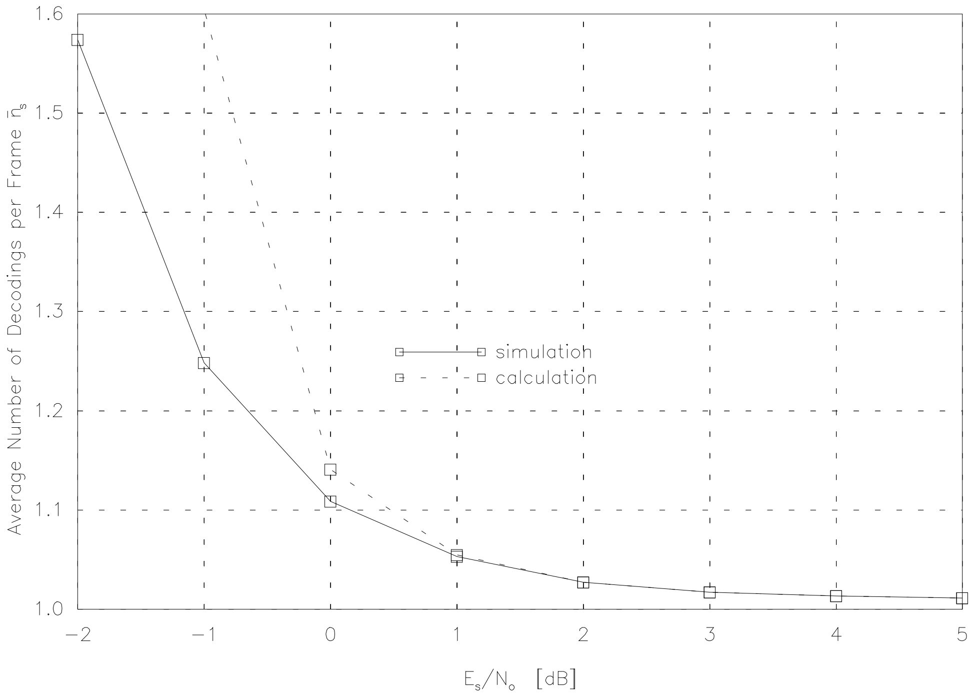


Figure 4.22: *Average decoding overhead of a system using a list synchronizer. Parameters as Fig. 4.21. We see how the simulation matches (4.59) closely.*

and concatenated, from this $2N$ long sequence an N long sub-sequence is chosen at random, in order to take into account the situation depicted in Fig. 3.7. Care has to be taken when choosing the sync word. Because the data is coded, not all sync words may be equally good because the decoder may or may not be able to output exactly the sync word from time to time if $L > nm$, i.e. the sync word is longer than the memory of the code times the number of symbols per branch. In the first case the frame sync performance will be degraded. If the code has the property that an inverted encoder output sequence is not a valid code sequence we can simply chose either the sync word or the inverted sync word depending on which of the two is a valid code sequence or not. If one chooses the incorrect sync word, then one will suffer a slight penalty at higher SNR.

4.5.6.1 Suboptimal rule

We begin by showing the performance of (3.37), but with both trellis termination *and* starting portions being used. The rate 1/2 code's constraint length was 4 ($m = 3$) with the octal generator matrix $\binom{15}{17}$ [CC81]. The length of the frame was 133 bits, with $L = 13$. The simulation results are shown in Fig. 4.23. We observe that the trellis aided scheme performs about 1 to 1.5 dB better than the traditional frame synchronizer. In Fig. 4.24 we show the same scheme but with a more powerful code. Here, $m = 5$ and the generator matrix $= \binom{65}{57}$. The performance enhancement is now improved (up to 2.5 dB), since this code has a longer termination sequence.

4.5.6.2 Simplifications to the suboptimal rule

In the same figure we have plotted the sync performance of the high SNR rule (3.39) with and without the trellis information. Both curves are slightly worse, the degradation being greater using trellis information. This can be explained by our use of the approximation (3.38) *as well as* the traditional high SNR approximation, in the derivation of (3.39).

4.5.6.3 Short frames with suboptimal high SNR rule and with preceding list frame synchronizer.

For very short frame lengths the performance increases even more as the ratio of termination symbols to frame length increases. We simulated: $N = 47$, $L = 7$ and the same $m = 5$ code. The improvement in the sync rate is dramatic (Fig. 4.25), especially at high SNR, when the traditional frame synchronizer fails due to the random appearance of the sync sequence in the data. A list synchronizer that outputs a list of $\nu = 4$ candidate positions to the frame synchronizer using termination information which were selected using the traditional high SNR rule (no trellis information), was likewise examined. We observe that $\nu = 4$ is adequate to ensure almost exactly the same performance as $\nu = 47$, although the computational complexity has been reduced by a factor 4/47.

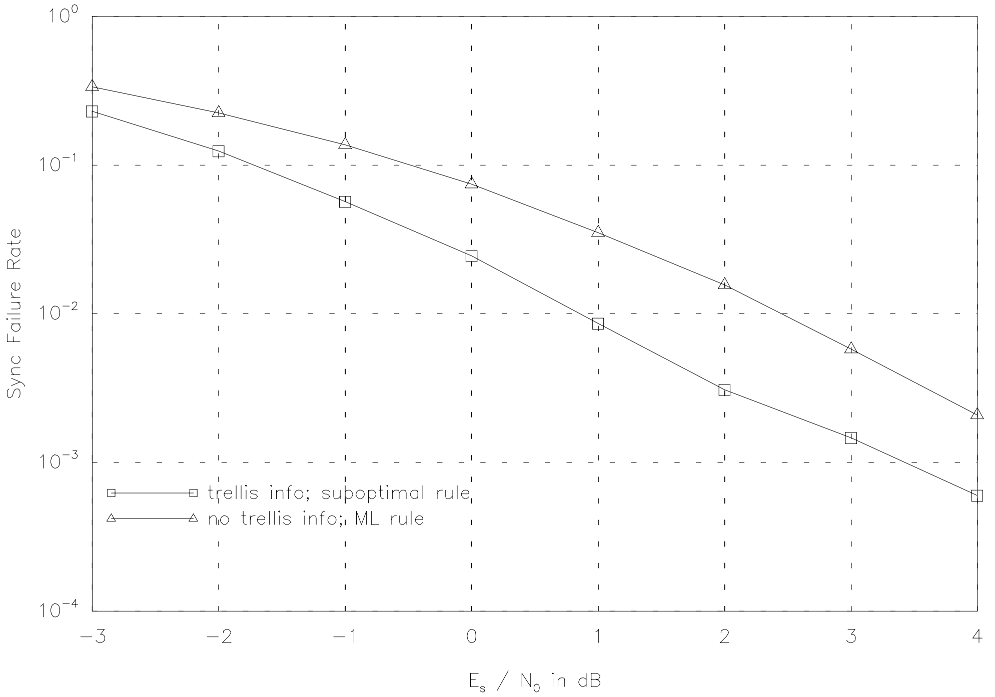


Figure 4.23: *Simulated synchronization performance of (3.37) for a code with memory $m = 3$. BPSK modulation on the AWGN Channel with coherent detection. Code rate $R = 1/2$, 60 info bits, $L = 13$, $N = 133$. Using trellis termination/opening information improves synchronization by about 1 to 1.5 dB.*

4.5.7 The Synchronizer for Non-Frequency Selective Fading

Monte Carlo computer simulation results are shown for BPSK and QPSK modulation schemes in Figs 4.26 and 4.27. Following the ordering of the legends, we start by showing the performance of the ML and high SNR rules for the AWGN channel, the difference between the curves is, of course, low. Next we see the ML and high SNR rules with and without CSI, on the Rayleigh distributed fading channel. Finally the performance of the correlation rules without CSI are given.

We observe that the ML and high SNR rules with CSI perform up to 1 dB better than those without CSI. We also notice that the high SNR rules approach the ML rules at high SNR. Interesting is that the difference between the ML and high SNR rules at low SNR, is larger than for just the AWGN channel. The greatest improvement that can be reached when starting with the AWGN high SNR rule is almost 1.5 dB (ML rule with CSI). But the low additional complexity of the high SNR rule with CSI (assuming CSI is available) makes it seem very attractive in practice.

Notice that the soft correlation rule performs significantly worse than the other rules. Looking at the performance at high SNR, it appears that the soft correlation rule never gives the same performance predicted by $Pr\{f|\text{RDL}\}$. The explanation -in the case of no noise- is simple: The soft correlation term (without CSI) at the correct position, does not necessarily equal the

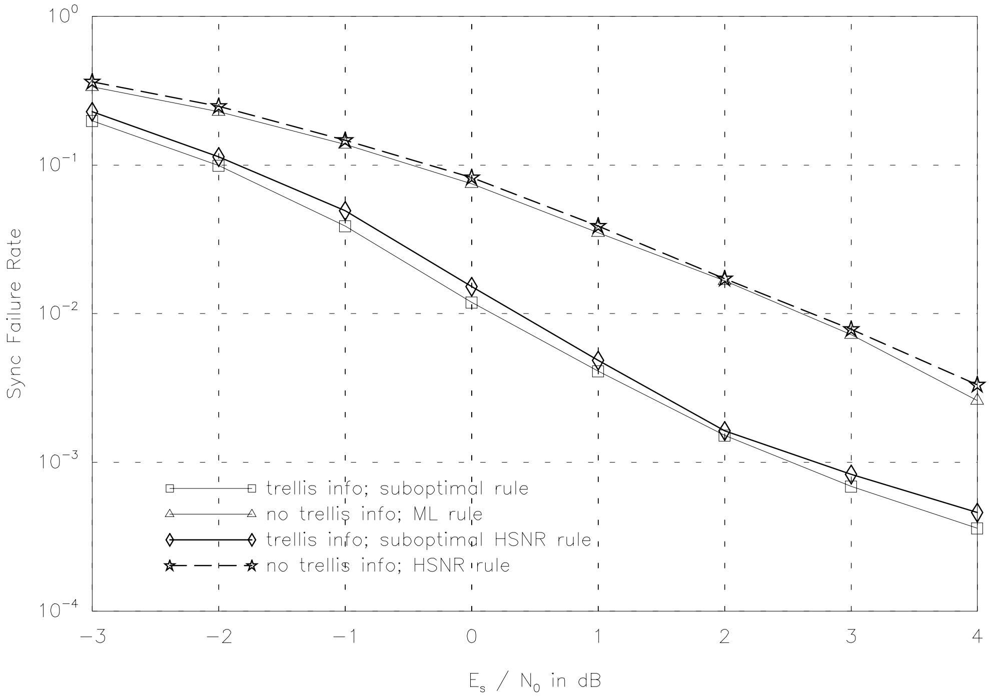


Figure 4.24: *Simulated synchronization performance of (3.37) and (3.39) for a code with memory $m = 5$. BPSK modulation on the AWGN Channel with coherent detection. Code rate $R = 1/2$, 60 info bits, $L = 13$, $N = 133$. The high SNR approximation performs almost as well as the (near) ML rule. Using trellis termination/opening information now improves synchronization by about 2 to 2.5 dB.*

energy of the sync word, but will depend strongly on the fading values. At other positions in the frame, which closely resemble the sync word, the correlation term might be higher than at the true position; this leads to a sync failure. The situation is rather similar to the illustration of Fig. 3.1, except that here the fading leads to the changes in the correlation. The same also applies to the correlation rule using CSI, but even to a greater extent: using CSI in the correlation rule degrades performance significantly [Rob92b].

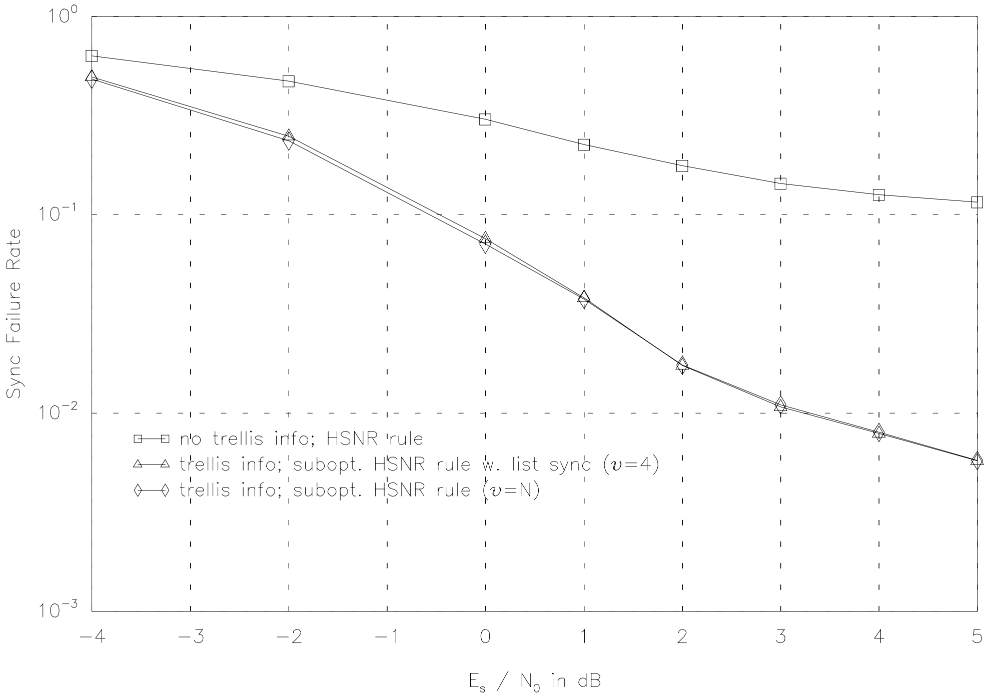


Figure 4.25: *Simulated synchronization performance of a trellis aided synchronizer using the high SNR approximation with a preceding list frame synchronizer (with list-length* $\nu = 4$*) for* $m = 5$*. BPSK on the AWGN Channel, code rate* $R = 1/2$*, 20 info bits (*$N - L = 40$*), sync word length* $L = 7$*.*

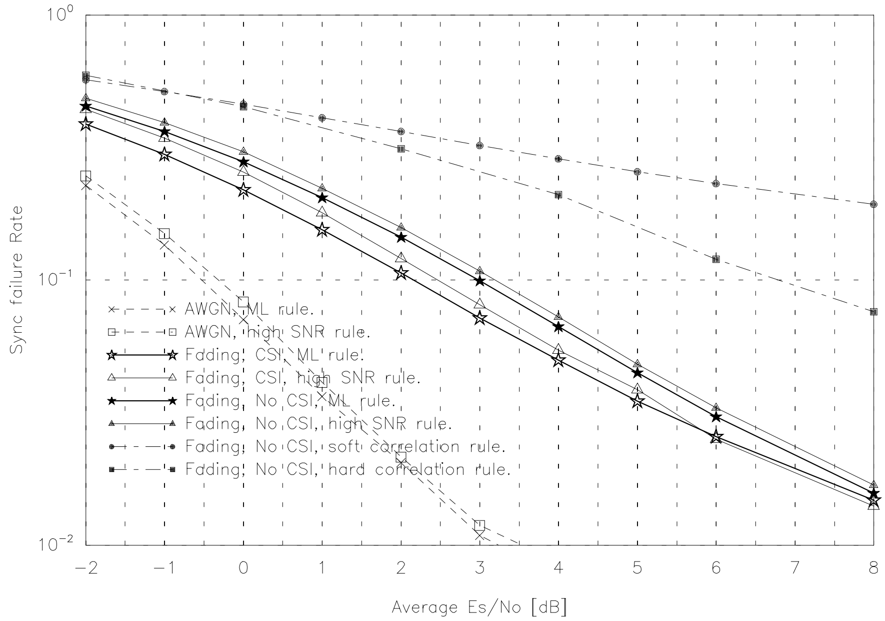


Figure 4.26: *Simulated synchronization rate of various synchronizers in a non-frequency selective Rayleigh fading fading environment. Parameters:* $N = 130$*,* $L = 13$*, BPSK modulation. Note how badly the soft correlation rule performs, its synchronization rate never even reaches* $1 - Pr\{f|\mathrm{RDL}\}$*. One can gain slightly less than 1 dB by using CSI in the high SNR and ML rules.*

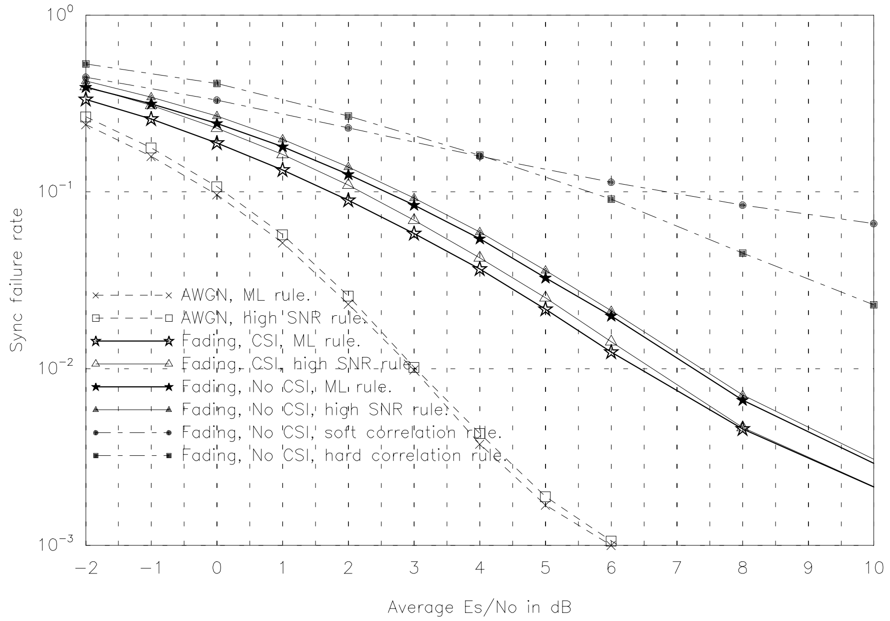


Figure 4.27: *Simulated synchronization rate of various synchronizers in a Rayleigh fading fading environment. Parameters:* $N = 28$, $L = 7$, *QPSK modulation. Again the soft correlation rule performs badly. One can gain slightly less than 1 dB by using CSI in the high SNR and ML rules.*

Chapter 5

Coded Frame Synchronization; Should the Sync Word be Added Before or After Coding?

In this chapter we will try to answer the question concerning the relative locations of the frame synchronizer and decoder. The work is described in more detail in the Masters thesis by Jacobs [Jac92] (supervised by the author). The motivation for performing decoding first (i.e. adding the sync word *before* coding) is that the decoder will hopefully reduce the number of errors in the sync word thus aiding the search.

5.1 The Two System Structures

In Fig. 5.1 we have shown the transmission chain for the two systems. The one on the left hand side we will denote by DAF (Decoding After Frame synchronization), the one on the right by DBF. In the DAF case, the sync word is inserted after coding, we deem this to be the more traditional case as opposed to inserting it before coding (DBF). Before discussing the individual receiver elements we have to (re-)introduce and modify some terminology that will enable us to compare the two approaches.

- R is the code rate, in the example of Fig. 5.1 it is 0.5.
- I is the number of information bit that are sent in one frame, in our example $I = 200$.
- L is the length of the sync word *at that place in the receiver where it is inserted and looked for*. The latter amendment is very important as we shall see. Here we have chosen $L^{DBF} = 13$ bit, $L^{DAF} = 26$ symbols.

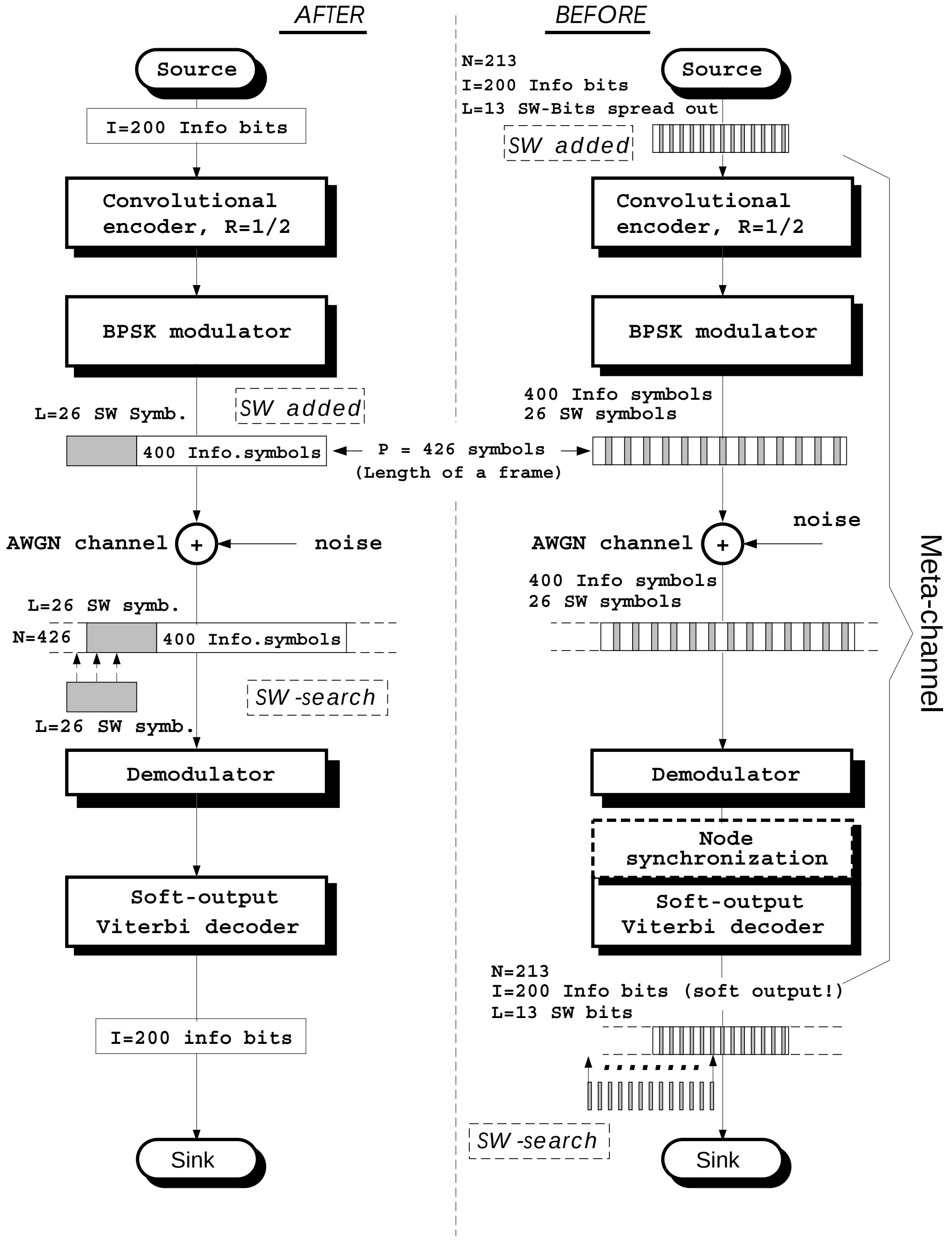


Figure 5.1: Structures of the DAF (Decoding After Frame synchronization) system on the right hand side and DBF (Decoding Before Frame synchronization) system on the left. On the right we have bracketed the meta-channel, which we will describe in 5.3. Note that the in the DBF system we attempt to guarantee independence of the disturbance on the sync word bits by spreading them across the information bit sequence.

- N denotes the length of the sequence in which the sync word has just been inserted and is equal to the length of the sequence in which it will be searched for. We have: $N^{DBF} = I + L^{DBF} = 213$ bits and $N^{DAF} = I/R + L^{DAF} = 426$ symbols.

- P is the length of the complete frame. Here $P = 426$ symbols in both cases.

At this stage some comments are due. Although it may appear confusing to re-define the length of the frame as P (no longer N), it can be justified when we state that in the sequel we wish to retain our previous concept so far associated with N when dealing with traditional frame synchronization: the length of the sequence in which we search for the sync word. This becomes important when we introduce the concept of the meta-channel shortly. The situation can be made a little less contradictory if we adhere to the use of superscripts DBF and DAF for N. In order to have a fair comparison, we shall require that P and I are equal for both schemes (hence they carry no superscripts), in other words the same number of information bits are transmitted with the same number of channel symbols.

Let us now proceed with our discussion of Fig. 5.1. The DAF structure needs little further explanation, but two points must be made on the DBF scheme: The sync word is added not in one piece but is 'spread between' the information bits, this is to try to reduce the effect of burst errors made by the decoder, on the frame synchronizer. Secondly, the block entitled 'node synchronization' fulfills the function of duly aligning the received symbols to a branch of the trellis. It is not necessary for instance, when QPSK modulation and rate 1/2 coding is employed, as in this case one complex symbol is mapped to one branch with no ambiguity. At this point we can mention that we are somewhat restricted in the coding we can employ in the DBF system; no punctured codes can be used (at least there appears no obvious solution) since the puncture matrix at the receiver would have to be aligned before frame synchronization. Furthermore, interleaving the channel symbols to protect against bursty channel errors is difficult for the same reason.

5.1.1 Advantages and Disadvantages of the Two Schemes

Before moving on to explain how the DBF system can be implemented, we wish to briefly state the advantages that we expect from each. The DAF system has the advantage that for a given code rate R, information sequence length I and frame length P, the length of the sync word L^{DAF} can be chosen to be

$$L^{DAF} = P - I/R, \tag{5.1}$$

whereas

$$L^{DBF} = R \cdot P - I = R \cdot L^{DAF}, \tag{5.2}$$

i.e. the sync word of the DBF system suffers a kind of rate penalty. This is because the absolute redundancy introduced by adding the sync word becomes increased through the coding process. On the other hand, the length of the sequence in which we search for the sync word is only half as long: $N^{DBF} = N^{DAF}/2$. Looking back at our results from 4.1, in particular Fig. 4.1, we see that for very high SNR at least, the DBF system will suffer an inherent disadvantage; $Pr\{f|\text{RDL}\}$ does not decrease as much when N is doubled as it decreases when L is halved. For lower channel signal-to-noise ratios we hope that the increase in the *effective* signal-to-noise ratio at the frame synchronizer input (a concept that we will indulge in shortly) will provide a sufficient improvement to increase the frame synchronization performance beyond that of the DAF system.

5.2 The Interface Between the Decoder and the Frame Synchronizer

So far in this work, we have considered optimal frame synchronization that uses all available channel- and other side-information. The ML and high SNR rule in fact only make sense when soft decisions are available. In the DAF system the frame synchronizer is the first unit after demodulation (other synchronization tasks are assumed to have been perfectly accomplished), soft decisions pose no problem. Luckily, we are able to use the soft-output Viterbi algorithm (SOVA) [HH89] [HR90] which we have already mentioned in 3.5, to provide soft decisions when convolutional codes are employed. But in order to successfully link the frame synchronizer to the decoder we must consider what it is that the SOVA outputs. In [HH89] it was conjectured that the output of the SOVA, which approximately the log-likelihood-ratio of each information bit i_k

$$\Lambda(i_k) = \log \frac{Pr\{i_k = +1/\text{observation}\}}{Pr\{i_k = -1/\text{observation}\}} \tag{5.3}$$

is approximately Gaussian distributed (at least for high SNR) if the all zero sequence is sent.

If we can model the output of the SOVA as a noisy version of the *en*coder *in*put, then we can very easily apply our frame synchronization techniques of chapter 3 to the sequence of decoder outputs in which the sync word is embedded. Since we use the high SNR rule, we need not derive a frame synchronizer that takes log-likelihood-ratios (or probabilities) at the input instead of channel values, for the high SNR rule does not need to know the variance of the noise. It is also invariant to multiplication of the input by a constant (for BPSK only), which means that we can directly use the log-likelihood-ratio output by the decoder. This now brings us on to the method we shall employ to estimate the performance of the DBF system based on the above channel model.

5.3 The Meta-Channel and its Use to Approximate the Synchronization Performance of the DBF System

In Fig. 5.1 we have bracketed the units from the encoder to the decoder and labelled it a meta-channel. There is one more condition that has to be met if we are to use it to model an *independent* Gaussian channel with an SNR different to that of the physical channel as far as frame synchronization is concerned. If the bits of the sync word are spread out sufficiently so as to destroy the memory of the meta-channel, then we effectively have independent noise samples for each sync word bit.

5.3.1 Use of Previous Simulations or Calculations of Frame Synchronization Error

We have two ways of obtaining the frame synchronization rate of the high SNR rule synchronizers for AWGN: 1) simulation results from 4.5 and 2) analytical calculations from 4.2.3. All that needs to be done to estimate the performance of the DBF system is to evaluate the SNR of the meta-channel and to evaluate the parameters L^{DBF} and N^{DBF}. These will replace L and N in the above simulations/calculations.

5.3.2 The Signal-to-Noise Ratio of the Meta-Channel

The SNR of the SOVA's output can be measured and has been plotted in Fig. 5.2 as a function of the SNR of the physical channel for different code memories at rate $R = 1/2$. The octal generator matrices for the codes of memory 3, 4, 5 and 6 were $\binom{15}{17}$, $\binom{23}{35}$, $\binom{65}{57}$ and $\binom{133}{171}$ respectively.

One can observe that the value of the channel signal-to-noise ratio where there is no benefit from coding is the same for all code memories and is about -2.25 dB E_s/N_0. Below this value, the SNR of the output is in fact lower than the uncoded case -a consequence of the 'threshold effect' of convolutional (and most other) codes. But the very low SNR will not be so interesting for us, since in that case decoding of the data will be impossible too.

5.3.3 Graphical Illustration of the Procedure

In Fig. 5.3 two curves are shown. The first shows simulation results of the synchronization rate for the high SNR rule at different SNR for the parameters $N^{DBF} = 213$ and $L^{DBF} = 13$.

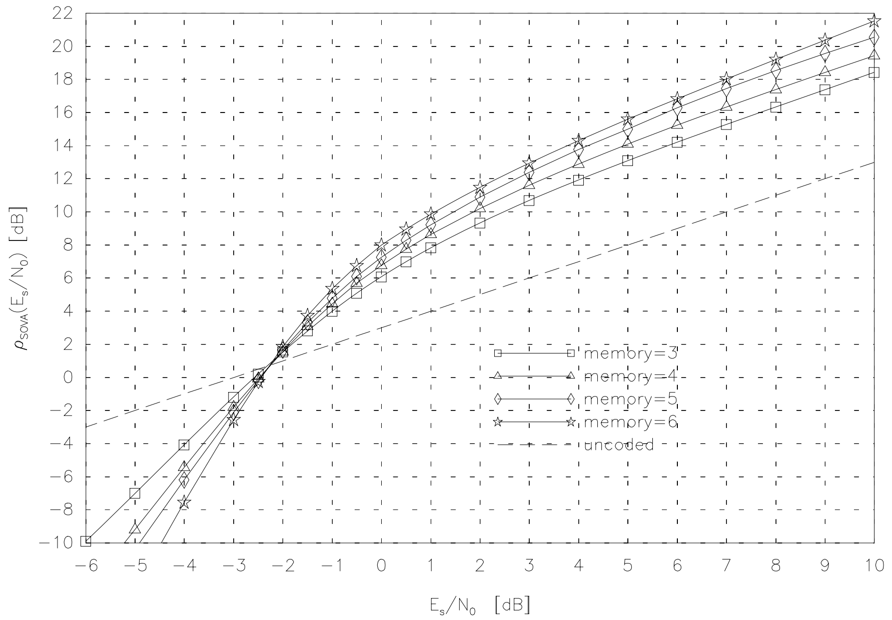


Figure 5.2: *The SNR of the output of the SOVA as a function of the SNR of the physical channel for different code memories at rate $R = 1/2$. Also shown (dotted) is the 'uncoded' $R = 1/2$ case which surpasses the coded cases below about -2.25 dB E_s/N_0. Note how all the curves for codes of different memories appear to pass through the same point, again -2.25 dB E_s/N_0.*

In order to convert these results to those of the DBF system, we must proceed as follows: For a given *channel* SNR, compute the SNR at the output of the meta-channel using Fig. 5.2; then take the synchronization rate from the original curve at this 'meta-channel SNR' and mark a point with this synchronization rate at the channel SNR. This has been visualized for a channel SNR of -2 dB E_s/N_0: the SNR of the meta-channel is 1.4 dB, the synchronization rate at this SNR is approximately 94 %, so the DBF system will have an expected synchronization rate of 94 % at -2 dB. We can express this a little more formally as

$$1 - Pr\{\mathrm{f}\}(\mathrm{D}BF, \frac{E_s}{N_0}) = 1 - Pr\{\mathrm{f}\}(N = N^{DBF}, L = L^{DBF}, \rho_{\mathrm{SOVA}}(\frac{E_s}{N_0})), \tag{5.4}$$

where $1 - Pr\{\mathrm{f}\}(\mathrm{D}BF, \frac{E_s}{N_0})$ is the synchronization rate of the DBF system at the channel SNR of $\frac{E_s}{N_0}$ and $1 - Pr\{\mathrm{f}\}(N = N^{DBF}, L = L^{DBF}, \frac{E_s'}{N_0})$ is the synchronization performance of a traditional frame synchronizer working on a frame length of $N = N^{DBF}$ with sync word length $L = L^{DBF}$ at an SNR of $\frac{E_s'}{N_0}$ (it can be a result from a simulation or from an analytical calculation). $\rho_{\mathrm{SOVA}}(\frac{E_s}{N_0})$ is the signal-to-noise ratio $\frac{E_s'}{N_0}$ of the output of the SOVA operating at $\frac{E_s}{N_0}$. The second curve shows the result of the construction. We should not forget that we have assumed perfect node-synchronization, perfect interleaving of the sync word and no deterioration due to not making use of correct termination of the code trellis or knowledge that the encoder starts in a predetermined state. The latter two assumptions are reasonable as we can let the decoder start a fraction of a frame before and run a fraction of a frame

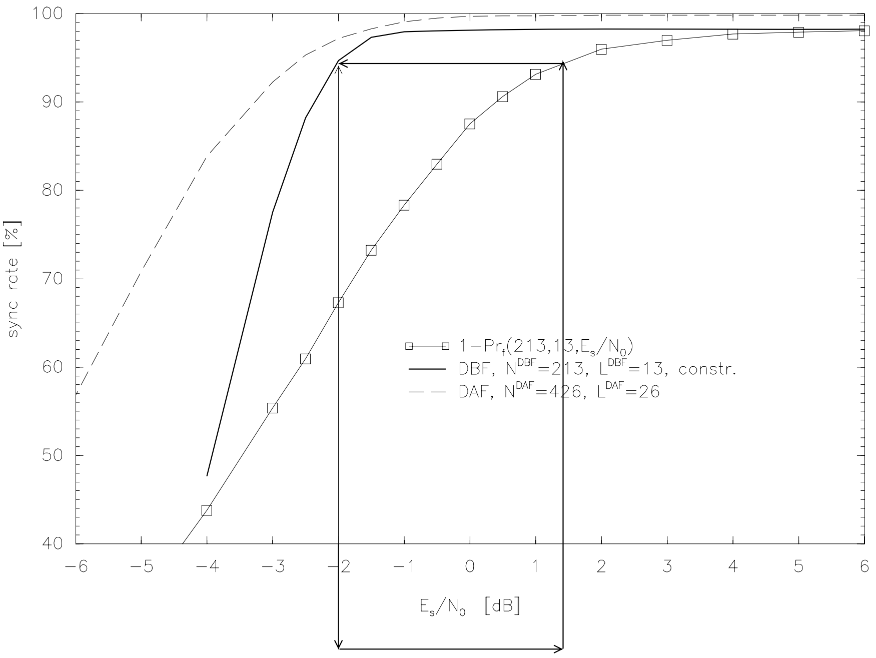


Figure 5.3: *Estimation for the Performance of the DBF system for code memory 3, $N^{DBF} = 213$ and $L^{DBF} = 13$. The bold curve is constructed from the simulation results of 4.5 (rightmost curve) by making use of the SNR of the meta-channel. See the text for a detailed description of the construction.*

beyond the normal frame length in order to increase the spacing of decoded data we will use to locate the sync word from the edges of the (unterminated) trellis.

5.4 Node Synchronization

Before we come to simulation results we have to say how node synchronization can be established. The method we have employed is very simple and is described for rate 1/2 codes: Two decoders start decoding but are offset by one symbol (half a branch) relative to each other. One decoder must be synchronized correctly, we shall take the offset of that decoder which has the largest metric associated with its best path through the trellis [VO79]. It might be possible (especially for large frames) to decode only a fraction of the frame and then to decide which was the correct offset, we have not investigated this, since node-synchronization need only be achieved or re-established when acquiring or re-acquiring frame synchronization and not during normal data decoding.

Simulations [Jac92] show that the false node synchronization rate only exceeds 1 % for signal-to-noise ratios (E_s/N_0) lower than -3 dB, so the effect on the frame sync failure rate will be negligible at interesting SNR.

5.5 Results

In Figures 5.4, 5.5, 5.6 and 5.7 we show the results for two different code memories and two set of frame and sync word length. We have plotted simulation results [1] of the systems DBF and DAF together with the results of the semi-analytical estimation described above. We are reassured that our model is accurate by the fact that the simulations come very close to our estimations. The results show that the DBF receiver structure is inferior to the DAF structure as far as frame synchronization rate in concerned: although the DBF curves rise more steeply (due to the strong increase of the SOVA output's SNR above the threshold), they do not reach the performance of the DAF system which has a larger sync word length L leading to a lower $Pr\{f|\text{RDL}\}$.

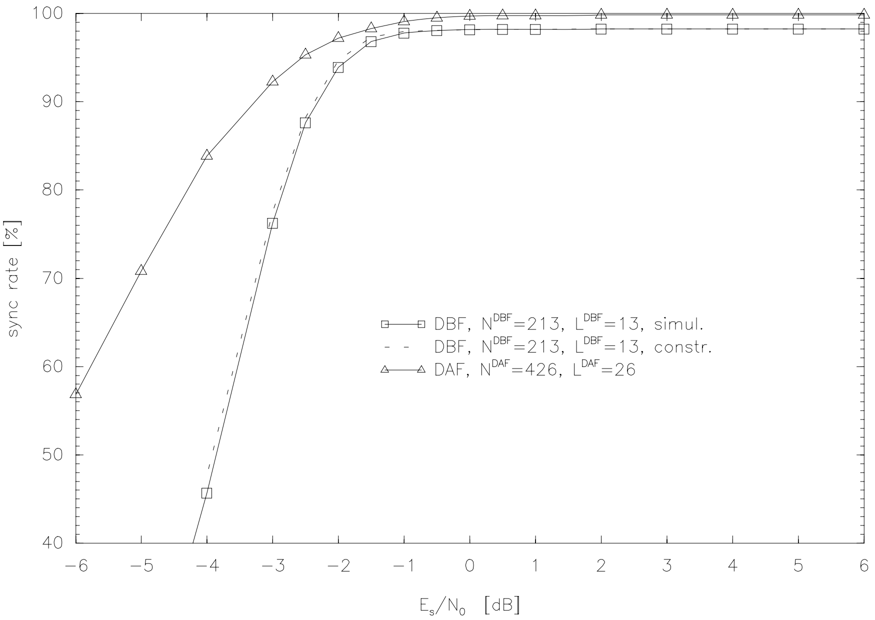


Figure 5.4: Comparison of the DBF and DAF systems. Parameters: $N^{DBF} = 213$, $L^{DBF} = 13$; $N^{DAF} = 426$, $L^{DAF} = 26$; code memory 3, $R = 1/2$. $1 - Pr\{f|\text{RDL}\}^{DBF} = 0.9886$; $1 - Pr\{f|\text{RDL}\}^{DAF} = 0.9999$.

To conclude this section let us point out that our hopes that the synchronization rate of the DBF system would -at least for some SNR- exceed that of the more traditional DAF system have not been met. However, this is not the complete picture as we will see in section 5.7.

[1] Simulations include node synchronization.

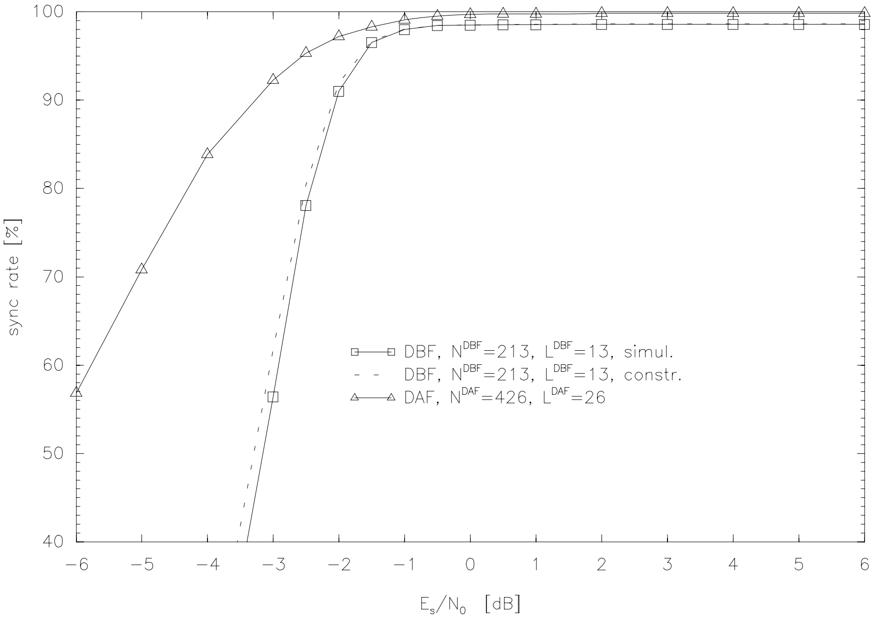


Figure 5.5: Comparison of the DBF and DAF systems. Parameters: $N^{DBF} = 213$, $L^{DBF} = 13$; $N^{DAF} = 426$, $L^{DAF} = 26$; code memory 6, $R = 1/2$. $1 - Pr\{f|\mathrm{RDL}\}^{DBF} = 0.9886$; $1 - Pr\{f|\mathrm{RDL}\}^{DAF} = 0.9999$. Note how the performance of the DBF system drops even more quickly than for memory 3 (Fig. 5.4), because the threshold effect of the decoder is more pronounced.

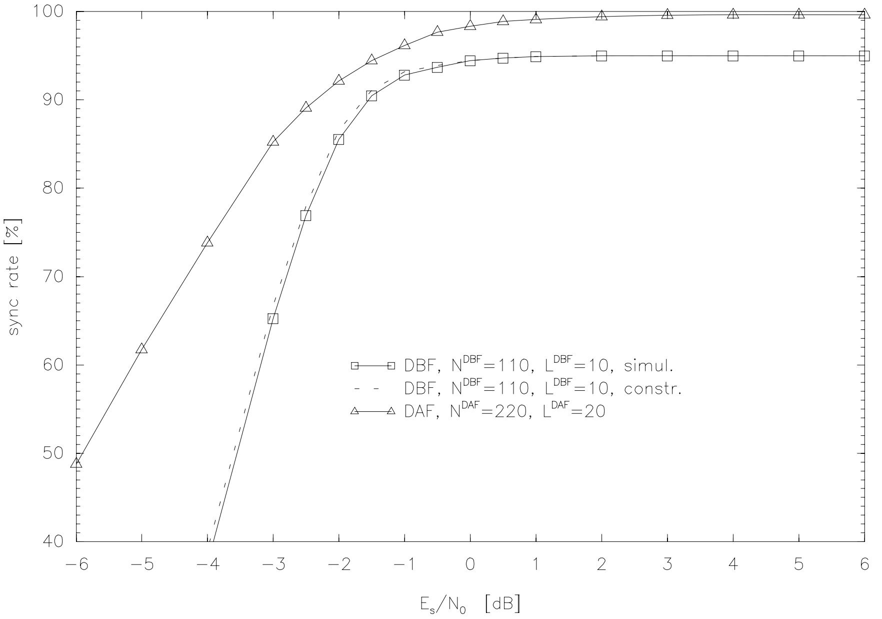


Figure 5.6: Comparison of the DBF and DAF systems. Parameters: $N^{DBF} = 110$, $L^{DBF} = 10$; $N^{DAF} = 220$, $L^{DAF} = 20$; code memory 3, $R = 1/2$. $1 - Pr\{f|\mathrm{RDL}\}^{DBF} = 0.9566$; $1 - Pr\{f|\mathrm{RDL}\}^{DAF} = 0.9999$.

5.6 Deterioration When Using the VA Instead of the SOVA

In chapter 4 we saw the surprising result that the hard decision correlation rule does not perform as badly as expected (‘hard decision is worse than soft decision’). The soft correlation

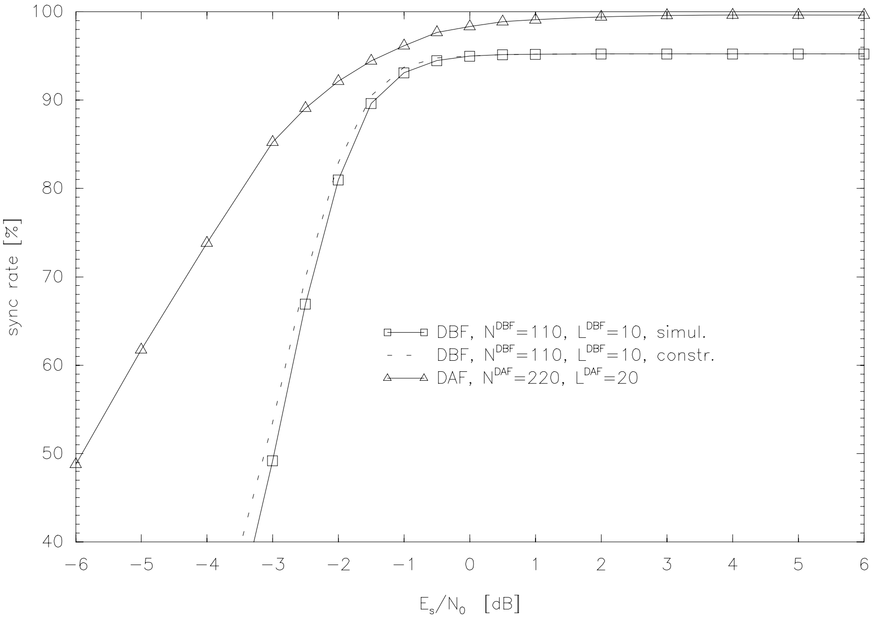


Figure 5.7: *Comparison of the DBF and DAF systems. Parameters:* $N^{DBF} = 110$, $L^{DBF} = 10$; $N^{DAF} = 220$, $L^{DAF} = 20$; *code memory 6,* $R = 1/2$. $1 - Pr\{f|\text{RDL}\}^{DBF} = 0.9566$; $1 - Pr\{f|\text{RDL}\}^{DAF} = 0.9999$. *The difference between DBF and DAF is even greater than before due to the more pronounced difference on the respective* $Pr\{f|\text{RDL}\}$ *s.*

rule was much worse for moderate and higher SNR. However, we would still like to show the gain that is achievable when using the SOVA (allowing the high SNR rule to be used) compared to employing the VA, making frame synchronization with hard decisions mandatory. In Fig. 5.8 one can see that about 0.5 to 1.5 dB can be gained by going from hard decisions to the high SNR rule, both curves are limited by $1 - Pr\{f|\text{RDL}\}^{DBF}$, of course. The price to pay (if the SOVA is not being used in any case) is an increase in decoder complexity of about factor 1.8 [JVM93].

5.7 Improving the Decoding Performance of the DBF System

So far, our results have indicated that the DBF system has a poorer synchronization performance compared to the DAF system. However, we must bear in mind that synchronization is only to be established at the start of a transmission session and occasionally when synchronization is lost (synchronization can be continually monitored) [JAS85] [Dri93].

We shall see that the inserted sync word of the DBF system (the sync word is inserted *before* coding) can actually help the convolutional decoder. This is because the bits of the sync word are intertwined with the information bits and both the positions and values of these

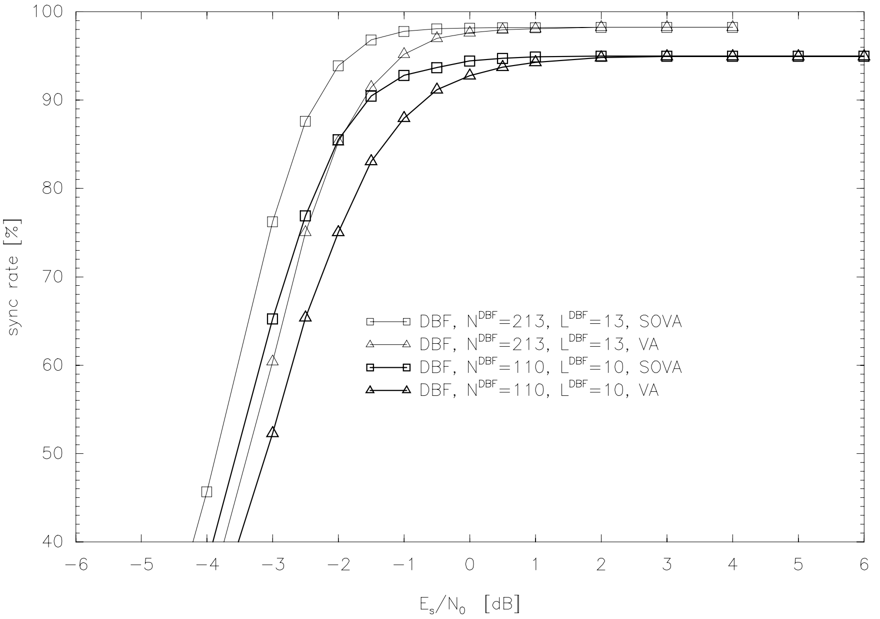


Figure 5.8: *Advantage of using the SOVA in the DBF system. Parameters:* $N^{DBF} = 110$, $L^{DBF} = 10$; *and* $N^{DBF} = 213$, $L^{DBF} = 13$; *code memory* 3, $R = 1/2$. *Note the improvement that is achievable when using the SOVA instead of the VA: the SOVA allows soft decisions to be used by the frame synchronizer.*

'sync' bits are known *when the receiver is in the 'in sync' mode of operation*, i.e. when frame synchronization is accomplished.

5.7.1 Decoding when the Trellis has Known Subsets of Transitions

In the normal decoding mode the receiver knows some of the information bits of the sequence it is trying to decode. We can best illustrate this in a diagram: in Fig. 5.9 we see a subsection of a four-state trellis with 9 transitions of which we know that the second transition is associated with an information bit 0 and the 6'th is associated with a 1. There are only two out of the possible four states that the true path can go through at steps 2 and 6, resulting in a reduction of the number of possible paths by a factor of two. This can help the decoder choose the correct path since an incorrect path with a path-metric greater than that of the true path (which would normally lead to a decoder error) might be an 'illegal' path and be rejected. In practice we can implement this by setting the metric increment of every transition associated with the opposite of the sync word symbol to a very large negative value thus penalizing the transition and all paths of which it is a branch.

We expect an improvement in the BER of the decoder output which will become more marked as the density of the known bits increases. The improvement for two examples of ratio of sync word length to frame length is shown in Fig. 5.10. One sees that the gain is only a fraction of

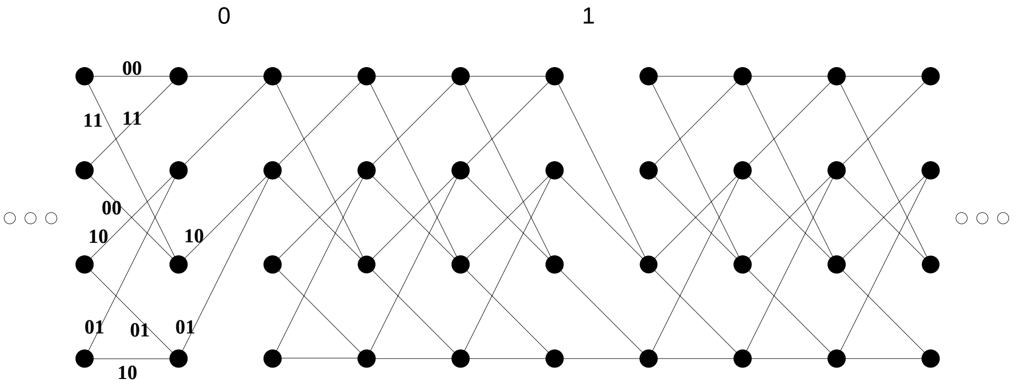


Figure 5.9: *Use of known bits to improve decoding performance. Shown is a subsection of a four state trellis with 9 transitions. We know that the second information bit is a 0 and the 6'th a 1. Thus there are only two out of the possible four states that the true path can go through at steps 2 and 6, resulting in a reduction of the number of possible paths. Knowing this can help the decoder choose the correct path.*

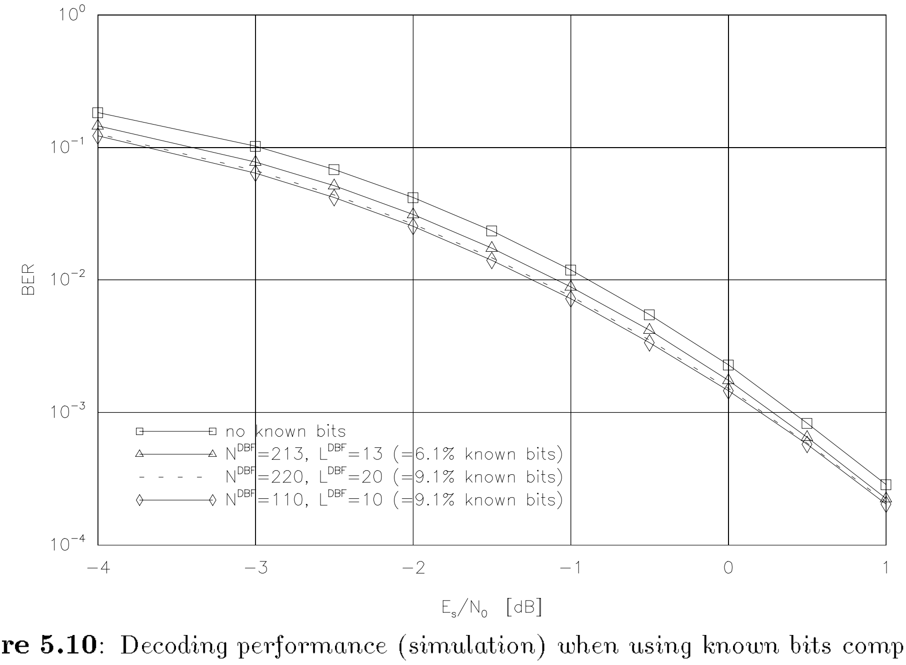


Figure 5.10: *Decoding performance (simulation) when using known bits compared to not using the known bits. We see that the benefit from using known bits increases as their density increases, but the gain is only a fraction of a dB in both cases. It should be made use of, however, as it is virtually for free. It should also be made part of our assessment of the relative performance of the DBF and DAF systems.*

a dB in both cases. It should be made use of, however, as it is virtually for free and must thus be made part of our assessment of the relative performance of the DBF and DAF systems.

5.8 Final Comparison

We are faced with the dilemma of having to compare two systems that differ in both frame synchronization rate and bit error rate. In some applications this might be enough to make the decision on which to use, but we wish to adjust one of the two systems so that one of the two criteria of performance are equal for both. If we stay with our constraint that the number of information bits I and channel symbols P per frame be equal, then one method of comparison is to lower the code rate of the DAF system until we get equal coding performance compared to the DBF system and to then compare their synchronization rate. This can be done by using punctured convolutional codes, for instance those given in [Hag88]. In order to choose the correct rate R^{DAF} we have plotted the BER performance of RCPC codes with different (but similar) rates in Fig. 5.11 together with the curves from Fig. 5.10 that show what can be gained by using the known (sync word) bits.

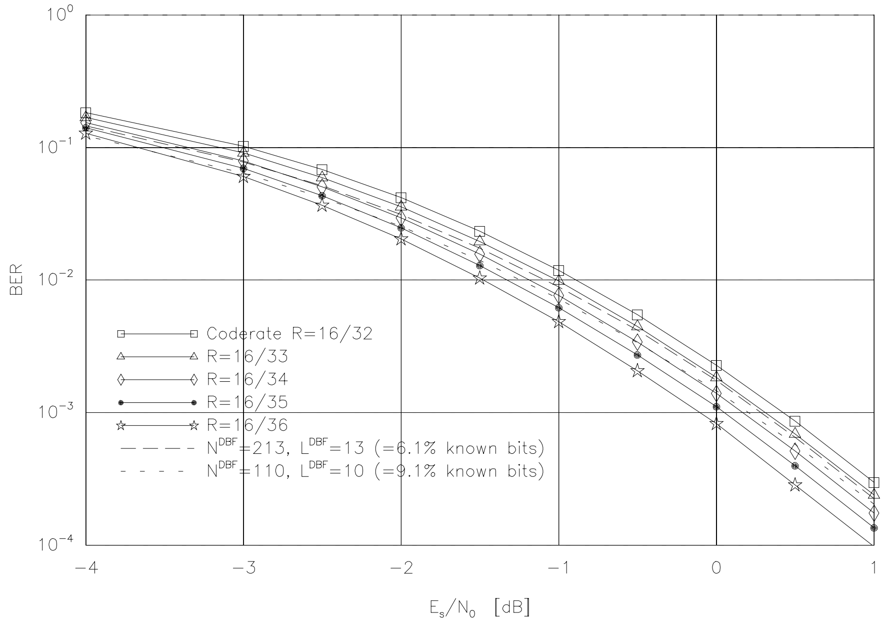


Figure 5.11: *Simulation of the decoding performance of different rate RCPC codes. We can use these results to find the code rate of the DAF system that has the same BER as the DBF system that uses known bits to help decoding.*

We can summarize the new frame structure of the DAF system for our two examples in table 5.1. From this we see that the constraint of limiting the length of the frame and reducing the code rate R^{DAF} leaves only a short sync word to be used by the DAF system if it is to have the same BER as the DBF system. As can be seen from Figs. 5.12 and 5.13 the synchronization performance is greatly reduced, especially for the smaller frame length.

It seems that to achieve a good BER is much more difficult than frame synchronization. As we have seen before, the sensitivity of the frame synchronization rate (for the traditional frame synchronization problem) on the length of the sync word is very pronounced. This

Table 5.1: *Parameters of the DAF and DBF systems with comparable bit error rates. The rate R^{DAF} is chosen such that the bit error rate of the DBF system using known bits when decoding will be the same. This enables the length of the sync word L^{DAF} to be determined (rightmost column).*

Frame length P	Info bits I	Parameters of DBF	Code rate R^{DAF}	$\frac{I}{R^{DAF}}$	L^{DAF}
426	200	$N^{DBF} = 213; L^{DBF} = 13$	$\frac{16}{33}$	413	13
220	100	$N^{DBF} = 110; L^{DBF} = 10$	$\frac{16}{34}$	213	7

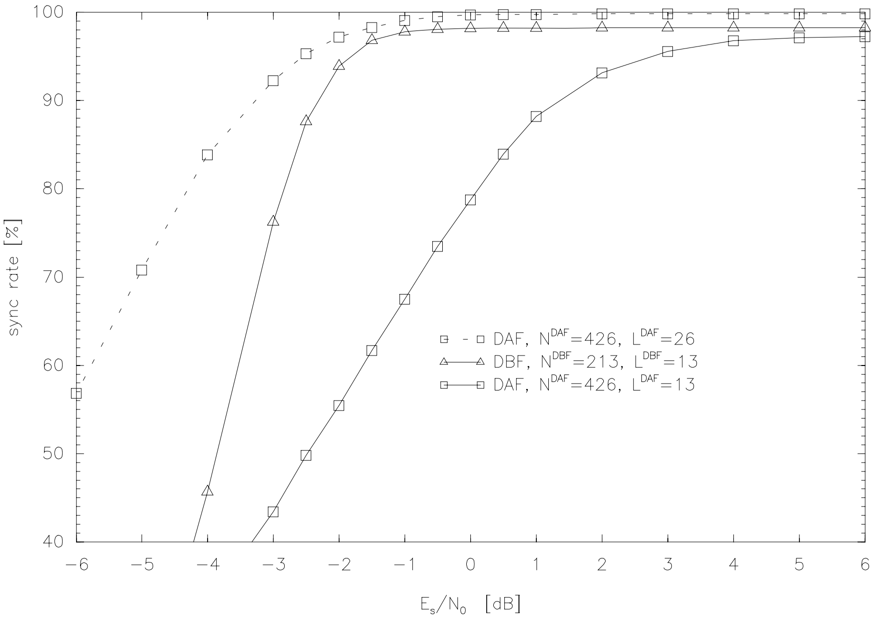


Figure 5.12: *Comparison of the DBF and DAF systems -BER being equal. Parameters:* $N^{DBF} = 213$, $L^{DBF} = 13$; $N^{DAF} = 426$, $L^{DAF} = 13$; *code memory* 3, $R^{DBF} = 1/2$. $R^{DAF} = 16/34$. $1-Pr\{f|\text{RDL}\}^{DBF} = 0.9886$; $1-Pr\{f|\text{RDL}\}^{DAF} = 0.976$. *The reduction of the length of the sync word to* 13 *has greatly reduced the synchronization performance of the DAF system.*

is especially true if the sync word is reduced below a certain threshold which is the case in the second example. Therefore, the comparison might lead to the impression that decoding before frame synchronization leads to greatly improved performance. However, when viewed from the aspect of improved decoding performance the benefits seem quite small (about 0.2 to 0.25 dB). Furthermore, if no use can be made from the sync symbols when decoding, the frame synchronization rate of the DBF scheme is worse, and in bad channel conditions will suffer even more. But in the traditional frame sync problem we are interested in maintaining synchronization during deep fades (or similar events) even when decoding is impossible.

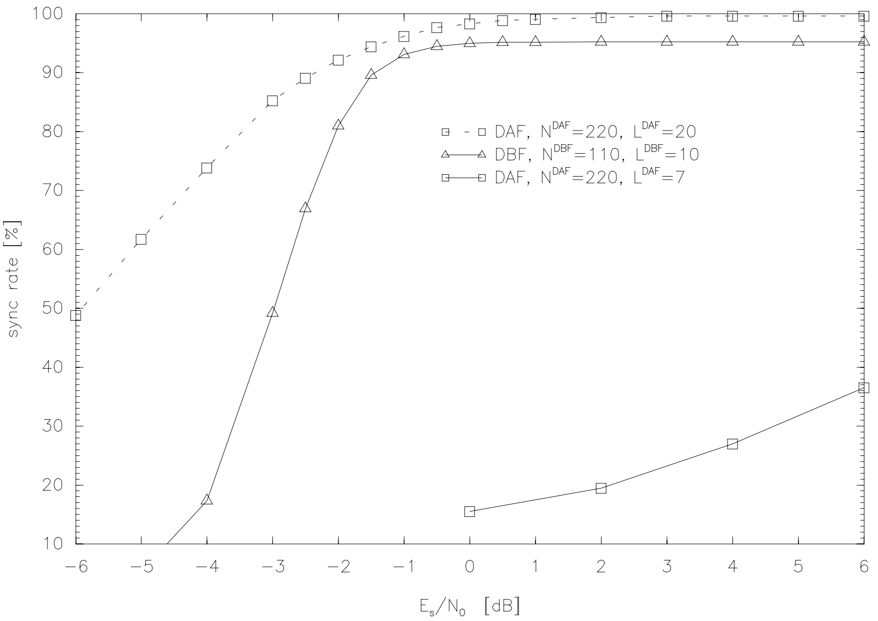


Figure 5.13: Comparison of the DBF and DAF systems -BER being equal. Parameters: $N^{DBF} = 110$, $L^{DBF} = 10$; $N^{DAF} = 220$, $L^{DAF} = 7$; code memory 3, $R^{DBF} = 1/2$. $R^{DAF} = 16/34$. $1 - Pr\{f|\mathrm{RDL}\}^{DBF} = 0.9566$; $1 - Pr\{f|\mathrm{RDL}\}^{DAF} = 0.487$. The difference between DBF and DAF is even greater than before due to the more pronounced difference on the respective P_{RDL}'s. The reduction of the length of the sync word to 7 has made the synchronization performance of DAF catastrophic.

Part II

Frame Synchronization for Preamble-less Packets

Chapter 6

Packet Communications

6.1 Introduction

6.1.1 The need for Packet Communication

We will begin by giving a brief overview of packet communication systems. Packets are used for transmitting data for two main reasons; firstly the need by several users to share a common channel, secondly when data is to be transmitted only at certain times, perhaps because only small amounts of data have to be transmitted, when for instance data arrives for transmission sporadically.

Packet communication has been used for several decades now, although it is becoming increasingly important. Applications range from high speed LAN/WAN computer networks and ATM networks [AW92], to low rate satellite communication, mobile communication and fixed terrestrial communication over radio channels. Data rates encountered range between several hundred mega bit/second, to a few hundred bit/second. Packet information content also varies greatly, from many tens of thousands of bit of information to only a few tens of bit. Similarly, the quality of the transmission medium encountered can vary between virtually noise-free to hostile multi-path fading channels with low signal-to-noise ratios.

It would be futile to try to address all areas of packet communication in one piece of work. Even reducing the field to simply radio communication leaves countless different possibilities to consider. In this work we will focus mainly on short packet communication on radio channels without inter-symbol interference. We will give examples of such systems later.

6.1.2 Access Protocols

Central to many packet communication environments is the question of access protocols. As mentioned in the introduction, several users need to share one channel, e.g. a radio frequency band. They share it on a *time-to-time* basis; hence this form of multiple access is referred to as Time Division Multiple Access (TDMA), in contrast to Frequency Division Multiple Access or the more recently developed Code Division Multiple Access. One should mention, that there are systems where two or even all these basic access techniques are combined.

6.1.2.1 Overview

For our purposes, TDMA random access protocols can be roughly classified into the following main categories

- Fixed TDMA and reservation protocols
- ALOHA protocol
- Slotted-ALOHA protocol

The first group is historically the oldest and perhaps the most widely used. Basically, each user of the common channel is assigned a time slot, often referred to as a TDMA slot or window, in which it may send its data. The slots are commonly of equal duration, and in the simplest of implementations each user may send data on a periodic, fixed basis. These protocols can be refined by dynamically allocating slots to the different users, depending on the users' needs and other factors.

The second group is quite different, there is no time slot structure. All users transmit asynchronously, on a random basis. To ensure a low collision rate, each user waits a random time before attempting to transmit. There is, therefore, the possibility of packets colliding, when two users transmit at the same time and their signals at the receiver are of such relative strength so as to make reception of either impossible. Collisions, and hence complete or partial loss of a packet must be tolerated in these types of communications systems. ALOHA systems have the advantage of not requiring slot timing. They may be enhanced by so called 'carrier sensing' in which each user listens to the channel and only transmits if the channel appears to be free. This only works, of course, if the users are able to receive one another, which is not the case for uplink satellite communications, for example.

The third group combines the ideas of both preceding strategies. Time is again divided into slots, just as in TDMA systems. However, the users transmit in these slots on a random basis. Thus we observe basically the same behavior as ALOHA systems, except that the total throughput is twice as high as that for ALOHA [Abr70] [KT75].

6.1.3 Short Packets -Why and When are they Used?

Since synchronization becomes more tough an issue to tackle and the gains in terms of bandwidth saving are most pronounced, we will in this work concentrate on improving frame synchronization for short packets.

Short packets are used when small amounts of data are to be transmitted, or data only arrives sporadically or where delay must be kept low. Some examples are:

- *Traffic management* e.g. [WRL+91] [HW90]. Data is transmitted between moving vehicles and other vehicles or roadside sites in order to pass position, speed and destination information, or handle toll collection, provide route guidance and traffic information. Data typically needs to be transmitted sporadically, and is often not long.
- *Medical applications* where patients carry with then monitoring units that transmit information to a central receiver. Power consumption may be particularly important.
- *Military uses* are numerous. Remote control, guidance and communication. Again power consumption may be important; additionally short packets are more difficult to detect and/or interfere with.
- *Environmental and measurement data* may need to be transmitted from remotely located monitoring sites to central stations. Information is usually accrued then transmitted. Data may only amount to a few tens of bit per measurement.
- *Access channels, packet acknowledgement and key (cipher) transmission* are examples of overhead needed for packet transmission that itself needs to be transmitted. Again the data length is usually very short, a few bit at the extreme. Since many packets belonging to this category may need to be transmitted for every information packet, they may degrade the total throughput and/or energy efficiency considerably.
- *Voice communication* for personal communication such as the DECT system [SSHS92] often uses short packets because of delay restrictions and because the transmission rate can vary with the duty cycle of the speech.
- *incremental redundancy* is sometimes transmitted when a receiver cannot detect a packet the first time. An example of such a scheme makes use of rate compatible punctured convolutional codes [Hag88]. The amount of incremental redundancy transmitted is typically a fraction of that of the the original data, furthermore, several such retransmissions may be necessary until the associated block of data has been received correctly.

It is only recently that short packets have been considered a useful means of transmitting data. Short packets have been neglected because they have been too hard to synchronize without greatly increasing the bandwidth overhead, until recently, with the advent of preamble-less, block-oriented algorithms which we will come to later (e.g. [CSV90] [OI92] [SC92]). By block-oriented we mean that the whole packet is used to recover the different synchronization parameters, after the packet has been sampled, quantized and stored in a digital memory.

6.1.4 Packet Structures

Hand in hand with the development of efficient block oriented packet processing algorithms went the introduction of a new packet structure. Traditionally packets consisted of a preamble, followed by a sync word for frame synchronization and the the data [Feh83] [UTMT91]. This arrangement was a requirement of the real-time processing of the packet: Initially carrier phase and frequency and also symbol timing would be established, then the output of a sliding correlator with the sync word was monitored. When the sync word was located, the data was demodulated and decoded if necessary. Basically, packets were treated in the same way as long continuous streams of data. Hence long preambles were common and discouraged the introduction of shorter packets. It was slowly realized that in many applications of packet transmission one had the enormous advantage that *packets could be sampled and stored prior to processing*. It is only this that makes it possible to synchronize short packets on hostile environments. In recent years, preamble-less packet structures have been proposed; they depend vitally on effective block-oriented synchronization techniques for symbol timing, carrier estimation and frame synchronization. The only redundancy remaining, besides coding redundancy, is the sync word. The main goal of this work is to find efficient techniques of reducing the length needed to achieve frame synchronization at a certain signal-to-noise ratio i.e. to reduce the length of the sync word.

6.1.5 Packet Receiver Structures

We have already mentioned different packet structures, those with a preamble (or mid-amble) and those without. The preamble-less structure depends on block-oriented methods of synchronization. We define the following two classes:

- No storing of the packet (except for decoding). If the packet is to be processed purely by hardware this is often adopted. A preamble becomes necessary since timing and carrier recovery need to be accomplished prior to demodulation. It has the advantage that the packet is processed in real-time, necessary for very time critical applications. We must remember, however, that any packet communication system where collisions occur will result in a possibly large delay.

- Sample then process (digitally). This is generally the most flexible approach. No preamble is required as we can recover the synchronization parameters using the whole block.

Our investigations will focus on the latter problem, since this novel processing technique seems promising and allows the greatest improvements through the use of optimal or close-to-optimal algorithms.

Chapter 7

Important Elements of Packet Receivers and Model of Packet in a Time-Slot

7.1 Coding for Packet Communications

Due to the low signal-to-noise ratios and/or fading environments effecting many packet communication systems, coding becomes necessary to protect the data against transmission errors. Furthermore, since collisions may occur, and also the synchronization tasks that need to be accomplished may not all be successful, error detection becomes necessary. The two, error prevention and detection are often combined, both are achieved by adding *redundancy* either in the form of extra symbols (resulting in a higher bandwidth) or by expanding the signal format [Ung81] -or both. We do not wish to give an extensive review of the vast literature available on this topic; furthermore, the type of coding used is very dependent on the particular application. Since we are interested in low signal-to-noise ratios, we assume that if error protection is employed, then it will often be using convolutional codes [CC81] [Bla83] [Pro89], the principle of which we have already introduced in section 3.6. Their principle advantage is their ability to use soft decisions or channel state information in a fading environment [Hag80]; this is vital if all the information about the channel is to be used and can provide a gain of several dB in signal-to-noise ratio (SNR). Additionally, soft output decoding is possible, beneficial if codes are concatenated and advantageous for source decoding [Hoe93] [Hag94]. One of the disadvantages of convolutional coding seems to be the difficulty in exact analytical performance evaluations, although this has not discouraged its use.

7.1.1 Block Coding

We will only mention another important family of codes: block codes. Most block codes share certain properties that distinguish them from convolutional codes:

- Decoding is usually only on hard decisions, producing hard decisions.
- Performance at low SNR and for small code rates is generally less than that of convolutional codes.
- The error statistics are different: convolutional codes result in bursts of errors in the decoded data stream; use of block codes may yield block errors (or -in the case of systematic block codes- blocks where the raw demodulator output is hard decided.)

7.1.1.1 Concatenated convolutional and block coding

The property of convolutional codes to work better at low signal-to-noise ratios combined with the fact that they produce bursts of errors has led to their combination with non-binary block codes. For these reasons the convolutional code is usually the 'inner' code, the block code the outer code. The purpose of the block code is to eliminate the errors left after decoding the convolutional code. Best known example of block codes to be used in such a scheme are Reed Solomon Codes [Bla83] and an example of concatenation of trellis coded modulation and RS codes is found in [Edb89]. The advantage of RS codes is that they can make use of the soft decisions from soft output Viterbi algorithm (SOVA) [HH89] through erasure decoding.

An even simpler case of concatenated convolutional and block coding is to use a cyclical redundancy check (CRC) code to determine with very high probability if there are errors remaining in the convolutional decoder's output. This is advantageous when short packets are transmitted and errors after decoding are quite rare. An application of such a construction will be given in the next section.

7.1.2 ARQ/FEC Schemes

Coding to protect against loss of data is often referred to as Forward Error Correction (FEC), redundancy enabling error correction being added *before* errors occur. Another way to deal with errors *after* they have occurred is to simply detect errors at the receiver by protecting the data with an error-detecting code and requesting retransmission of the affected block(s) if they are found to be in error, these schemes are called ARQ schemes, [Wel83] [MC85].

Both approaches can be combined, resulting in ARQ/FEC schemes that protect data in such as way that errors can be corrected with a certain probability and any remaining errors are detected. Now in this case, it is known that making use of what has been received when processing a repeated block is better than disregarding it. An elegant way of realizing this was presented in [Hag88]. A special family of convolutional codes called Rate Compatible Punctured Convolutional codes (RCPC codes) was introduced that allow the block to be transmitted with little error protection (high rate) initially; should this result in errors [1], incrementally transmitted additional redundancy can be optimally combined by the receiver. This process of retransmitting additional redundancy can be repeated several times if necessary. The advantage is a high achievable throughput measured in error-free bits per channel use (symbols). Characteristic of this scheme is that small packets must be transmitted when supplying the receiver with additional redundancy -these packets must all be synchronized reliably.

7.2 Processing of the Traditional Packet Structure - Packets with Preambles and no Storage Prior to Decoding.

The majority of present packet communications systems fall into this class. Examples are the DECT system for personal communications [SSHS92] and many mobile satellite links [Boe92].

The algorithms used to recover timing and carrier parameters are most often feedback structures such as phase-locked-loops (digital or analogue PLLs) [LC81] [Fra80] [Oer89] [Kob71]. PLLs have the disadvantage of the hang-up phenomenon occurring, whereby the estimated parameter, symbol timing offset or carrier phase is far from the correct value. In traditional communications this leads to a loss of a certain amount of data. Here, it can mean the loss of a complete packet. Closely linked with this phenomenon is the problem of acquisition of PLLs. Since processing is on-line, the preamble must be long enough to ensure stable operation of all loops. Particularly sensitive are loops where the timing acquisition depends on the carrier acquisition and vice-versa, for instance the Mueller-Müller timing recovery algorithm needs phase synchronization to be achieved [MM76].

7.2.1 Frame Synchronization

The method adopted for frame synchronization is usually to insert a known sync word between the preamble and the data portion of the packet. A correlation (usually using hard outputs) is then done with the sync word; when a threshold is exceeded the sync word is declared to have been found, hence the position of the beginning of the packet is known [Sch80] [Boe92] [Bi83].

[1] A CRC code was assumed to guarantee perfect error detection.

A refinement of this principle has been given in [SH90] that enables the number of uncertainties for the location of the sync word to be reduced by factor of two by making use of the preamble with a periodic $-1\ 1\ -1\ 1$... structure. Furthermore, the case of convolutionally encoded data is investigated: by special code choice (the encoder must remain in the all-zero state during the preamble) and using just one sync word symbol that is encoded to represent the start of the data portion, the receiver is able to monitor the decoder output and observe when it first leaves the all-zero state. This elegant method is an example of coded synchronization and is well suited to environments requiring on-line processing of the packet, however a preamble is a necessary requirement to ensure that the decoder has had time to achieve carrier, symbol timing and node synchronization.

More recently, the optimal frame synchronizer for packets with preambles has been derived [MM93]. The preamble is used as an extension of the sync word to aid frame synchronization and not just timing and carrier recovery. The algorithm presented was shown to be considerably better than the correlation rule -an observation that has often been made in the past and which we will encounter throughout this work.

7.3 Preamble-less Packet Communication -a Modern Receiver Concept

Recently, preamble-less receiver concepts have gained in popularity because of the bandwidth and power efficiency associated with them. It is only the application of digital signal processing techniques which allow the packet to be stored and subsequently be processed that make this receiver concept attractive. The processing capabilities of modern DSP-chips are increasing at a phenomenal rate, allowing more and more refined processing to be carried out for any given transmission rate.

The underlying concept is very simple and appealing. The packet is simply sampled at a sufficiently high rate and stored in a memory. Different processing stages are now invoked, usually starting with coarse frequency offset estimation, based for example on FFT techniques [Rif74] [Cow93b] or a novel technique proposed by Crozier requiring fewer operations [Cro93]; this is followed usually by symbol timing offset estimation, for example using the digital square and filter technique presented in [OM88]. After symbol timing has been established and the matched filter values at the symbol rate are determined by interpolation, processing is continued by either frame synchronization or phase and fine frequency offset estimation.

7.3.1 Problem Definition

7.3.1.1 Implication of access protocols for packet synchronization

This part of the work will deal with synchronization, frame synchronization in particular, of preamble-less packets. Let us reconsider the classification of the three main different access protocols (see section 6.1.2). If one does not deal with the problem of collisions, then it is even possible to treat slotted-ALOHA systems in the same way as TDMA systems; requiring only two classes to be distinguished. This assumption is not so far fetched when one assumes that a collision will often result in a loss of all packets involved (usually two, sometimes more).

The basic difference between the two (new) subgroups (TDMA and slotted- ALOHA; ALOHA) is that in the first, the receiver is able to make use of the fact that a packet must arrive in a designated time slot; whereas in the second case the packet arrives sporadically. These two cases can be treated as one as far as synchronization is concerned if one assumes that the receiver can detect energy on a continuous basis and then decide that a packet must have arrived if a threshold is exceeded [WRL$^+$91] [MM93]. Thus an ‘artificial window’ can be placed around the newly acquired packet prior to subsequent processing.

7.3.1.2 The model

At this stage we will introduce the model used to describe the packet and its location in a time slot and the information provided for the frame synchronizer: see Fig. 7.1. We assume that the time-slot either contains a packet or not (we do not explicitly consider collisions). The slot is either a TDMA or slotted ALOHA slot, or an artificial window placed around a packet after energy detection. The packet cannot extend beyond the boundaries of the slot.

Let us assume that a packet has actually been transmitted. The modulation used be M-ary and the set of transmitted symbols is $\{W_j, 0 \leq j \leq M-1\}$. The packets are N-symbols long, of which the first L symbols are a known sync word $\vec{S} = (S_0, S_1, ..., S_{L-1})$. The symbols in the packet data sequence, $\vec{d} = (d_L, d_1, ..., d_{N-1})$, are assumed to be chosen randomly and uniformly from the signal set. Our packet is assumed to start at the $\tilde{\mu}$-th position of the slot of length G. Typically, the value of $G-N$ will vary between 10 and 100, mainly depending on clock uncertainties, delay differences or the accuracy of the artificial window. For instance, the maximum delay difference between a geo-stationary satellite and mobile users at the end of the beam coverage area is 16 ms, for a data rate of 2.4 k bit/s this corresponds to a minimal slot over-length $G-N$ of 38 [LY84]. The preamble-less INMARSAT-C access channel has a slot over-length of 54 symbols [CSV90]. A proposal for an access protocol for EHF communication between moving vehicles assumes an over-length of 40 μs at a data rate of 500 k bit/s to 2 Mbit/s, corresponding to $G-N$ from 20 to 80 symbols [HW90].

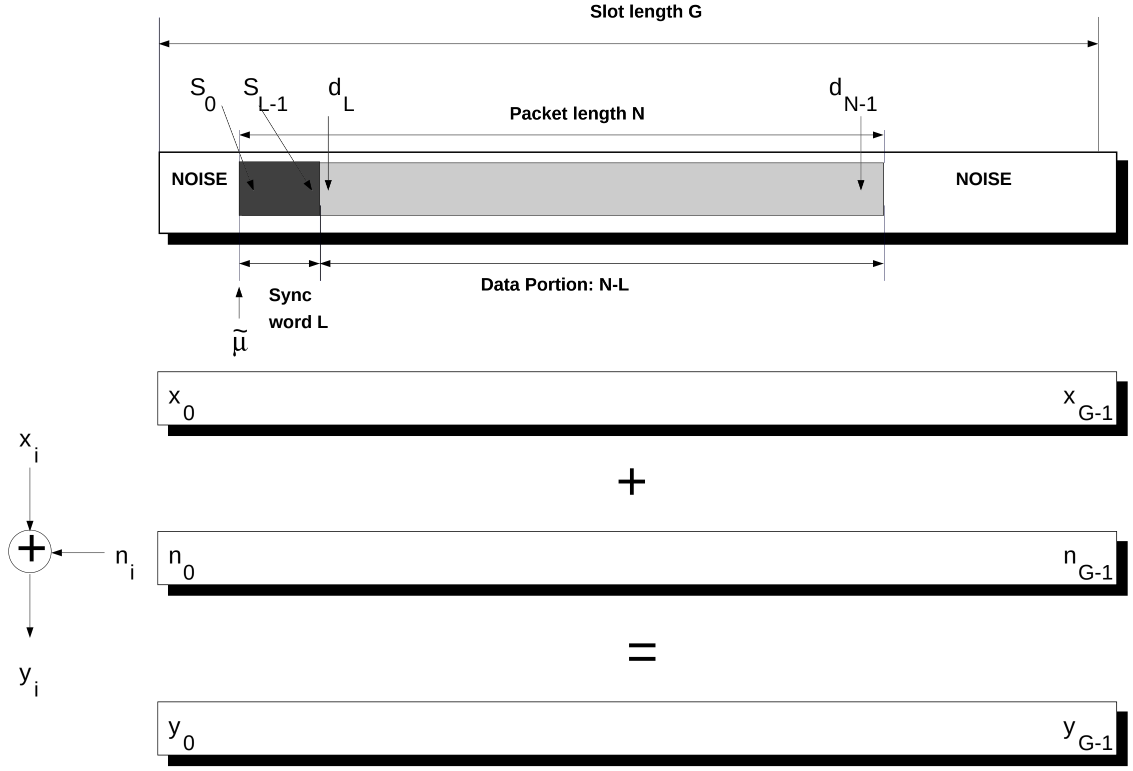


Figure 7.1: Model for frame synchronization of a packet surrounded by noise. The packet (shaded) lies in a time-slot. The receiver operates on of symbols of length G; $\vec{y} = (y_0, y_1, ..., y_{G-1})$. The receiver has no knowledge of the transmitted sequence corresponding to the extracted sequence $\vec{x}$ or the noise sequence $\vec{n}$. The goal of the synchronizer is to determine the packet starting position $\tilde{\mu}$. Known to the receiver is the sync word $\vec{S} = (S_0, S_1, ..., S_{L-1})$ comprising the start of each packet, and the length N of each packet.

7.3.2 Algorithms for Timing and Carrier Synchronization

The in-phase and quadrature baseband signals are sampled -usually at four or eight times the symbol rate [WRL$^+$91] [CSV90] after the analogue signals have been band-limited to fulfill at least the sampling theorem (taking into account a possible frequency offset). The complex samples are A/D converted (for example with 16 bit in [WRL$^+$91]) and stored in a memory. For short packet transmission the memory costs are not very significant and may amount to a few K byte only. We now look at some of the different algorithms used for processing. The discussion is not intended to be complete by any means, it serves only to present the most commonly found algorithms.

7.3.2.1 Coarse frequency acquisition

The carrier frequency offset may be a significant proportion of the symbol rate. One method that is predestined for block-oriented synchronization of sampled packets is an FFT based search procedure. After stripping the signal of modulation, commonly done by squaring the signal when using binary phase shift keying (BPSK) or taking the fourth power in the case of QPSK [2], a FFT is performed and the spectral power density is calculated, then the FFT bin is searched for that contains the largest peak hence enabling the stored signal to be corrected by this frequency offset and subsequently to be matched filtered.

7.3.2.2 Timing synchronization

One very popular block-oriented timing synchronization method is the square and filter method proposed in the digital form in [OM88]. It is the digital counterpart of the commonly known timing synchronizer that determines the phase of the spectral component (hence filter) of the square of the magnitude of the matched filter's output. In the analogue version of the method, this phase is used to generate a control signal for the sampling time of the matched filter. In the digital implementation just the Fourier component of the squared magnitude is evaluated and the phase uniquely determines the timing offset. Correct timing is then achieved by either choosing the sample nearest to the evaluated sampling time or by performing interpolation. There are several advantages gained by using this method: firstly no use is made of known data symbols, secondly the algorithm is insensitive to phase offset or even non-frequency selective fading (use is made of the square of the *magnitude* of the (complex) matched filtered signal), thirdly we have observed that it is insensitive to noise surrounding the packet. It is the latter fact which is a strong argument for performing frame synchronization *after* timing synchronization, this even holds when the packet whose position is unknown in a comparatively large slot (up to twice the packet's length). The frame synchronization algorithms are derived in this work and earlier assume perfect timing and phase synchronization. Most recently proposed block oriented phase recovery techniques [VV83] [Ohs90] [Cow93b] also require that symbol timing be established. The invariance of the digital square and filter timing method to unknown frame synchronization has not been shown hitherto, we will give shortly present simulation results for this algorithm to confirm the above statement. Finally, even some traditional timing recovery loops have been praised for their invariance to carrier-phase offsets, for example the Gardner Algorithm (GA) [Gar86] (it is also a non-data-aided technique). They can be applied prior to frame synchronization provided that the algorithms operate on a part of the received vector where the packet is guaranteed to lie. Finally, another method was proposed in [SC92] which performs similarly to the above, it too is neither data aided nor decision directed (although it can be modified

[2] Some recent work has investigated other nonlinear operations [Cow93b].

to be data-aided giving a performance improvement).

The Digital Square and Filter Timing Algorithm Let us present Oerder's and Meyr's digital reformulation of the square and filter timing algorithm [OM88]. We assume that our estimator operates on the complex valued matched filter output sequence $\vec{y}^{\mathrm{MF}} = \{y_0^{\mathrm{MF}}, ...y_{p\cdot(G-1)}^{\mathrm{MF}}\}$ sampled with p samples per symbol period T_s (p must be greater than 2 [OM88], $p = 4$ is chosen in that paper and here) corresponding to a sampled time slot of length G in which a packet may lie. Goal is to generate an estimate $\hat{\tau}$ of the optimal normalized sampling point of the matched filter, $\tilde{\tau}$, where $-1/2 < \tilde{\tau} \leq 1/2$. The estimator is indeed very simple, the absolute value of each element of the sequence $\vec{y}^{\mathrm{MF}}$ is squared, then the timing estimate is evaluated through

$$\epsilon_\tau = \frac{-1}{2\cdot\pi}\cdot \arg\left[\sum_{k=0}^{p\cdot(G-1)} |y_k^{\mathrm{MF}}|^2 \cdot e^{-2\pi jk/p}\right]. \tag{7.1}$$

In [OM88] it is shown that the estimate is unbiased, and an approximation for the variance of the estimate was derived for large G [3]. Instead of using the variance of ϵ_τ as the measure of performance we will simply look at the increase in the (uncoded) bit error rate of packets that lie in slots of different lengths where the symbol timing evaluated with (7.1), compared to the BER of packets with perfect symbol timing.

Simulation Results We investigated uncoded BPSK and QPSK modulated packets transmitted over an AWGN channel. Phase coherence was not assumed at the point of symbol timing recovery, only for demodulation. The transmission pulse is a raised cosine impulse (Nyquist system) with the roll-off parameter $\beta = 0.4$, see chapter 6 in [Pro89]. The resulting bit error rate for packet lengths of N equal to 25, 50 and 124 are shown in Figs. 7.2 and 7.3 for BPSK and QPSK, respectively. We observe that the performance degradation for the smaller packet lengths is more marked, becoming greater with a larger noise window surrounding the packet (ratio G/N larger). However, the loss is still very small and is dominated mainly by the length of the packet N. At this point it is important to observe how little of the loss is due to performing timing recovery *before* frame synchronization, i.e. the effect of having noise around the packet in addition to the noise perturbing the packet itself is small. It is known that the above algorithm performs close to the Cramer-Rao lower bound on the timing offset variance [SC92], so we cannot find an algorithm that performs non-data-aided estimation more efficiently. The alternative would thus be to perform decision directed timing acquisition after (perfect) frame synchronization ($G = N$) -with all associated hang-up problems, leaving the problem of achieving frame synchronization without timing recovery still to be solved.

[3] In that paper it is assumed that the receiver is operating on a continuous block of data of length G, i.e. in our case this is equivalent to the packet length N being equal to G.

Which ever way we see it, the loss due to imperfect timing synchronization is almost insignificant except for very large ratios G/N which might be encountered only when receiving sporadic packets. In the latter case, timing acquisition might even be done twice -a second time after frame synchronization when $G = N$ in the above equation.

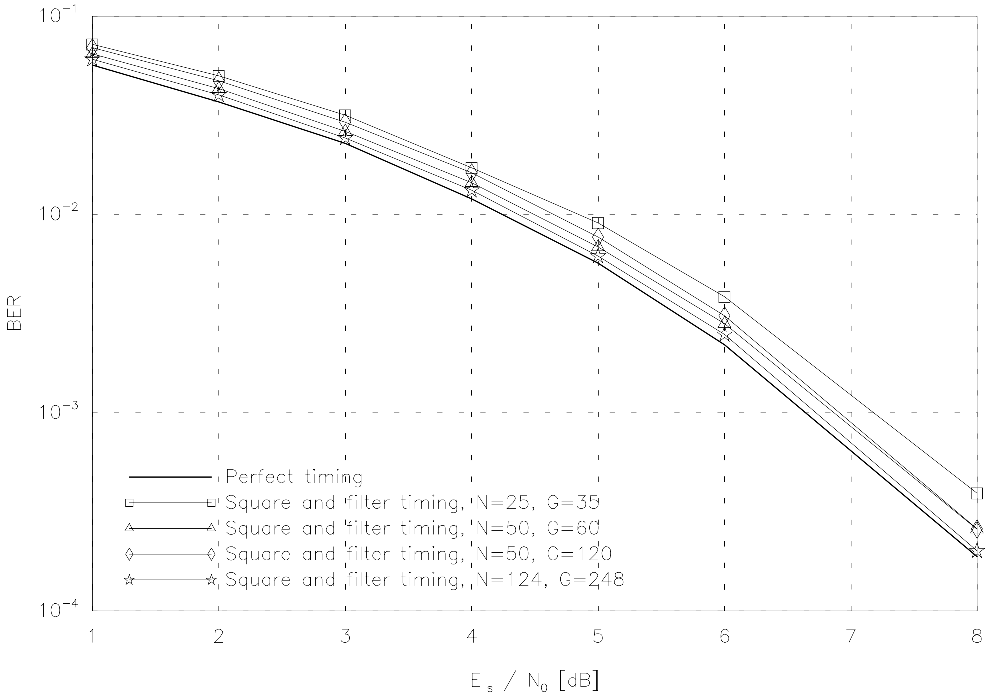


Figure 7.2: *Degradation of the uncoded bit error rate in packets due to imperfect timing recovery, for BPSK modulation. Parameter: The block size N and slot length G. Note the small degradation for larger N and the relative un-importance of the ratio G/N.*

7.3.2.3 Frequency and phase synchronization

So far, we have implied that timing synchronization has been successfully accomplished before fine-frequency and phase estimation are tackled. The task is essential for coherent detection, and frequency detection is necessary even for differentially coherent detection [SD87] [WRL$^+$91]. In addition to the vast collection of traditional phase-locked-loop type algorithms, there is increased interest in block-oriented techniques. Best known is probably that proposed by Viterbi and Viterbi [VV83], which only considers phase recovery of blocks of data such as packets. The major advantage is that is does not suffer from hang-up phenomena, however, it is sensitive to frequency offsets.

The FFT method from above can be used for fine frequency correction if sufficient padding is employed to guarantee the necessary accuracy [Cow93a] and it represents the true maximum likelihood frequency estimator if the padding is sufficiently large. Another algorithm was proposed in [Ohs90] and recently applied in [DS92]: the method of recursive least mean

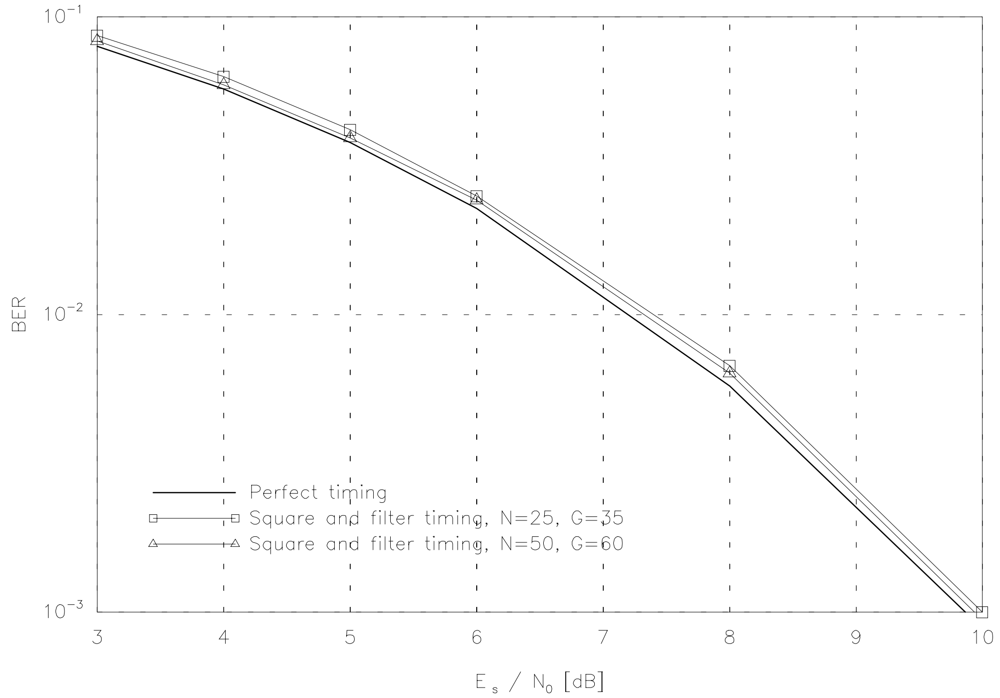


Figure 7.3: *Degradation of the uncoded bit error rate in packets due to imperfect timing recovery, for QPSK modulation. Parameter: The block size N and slot length G (QPSK **symbols**). Note how the degradation is even smaller than for BPSK. For $N = 50$ there is virtually no loss due to imperfect timing.*

squares. The algorithm is based on the maximum likelihood principle given observation of the phase of the received samples (at least for high SNR).

7.3.3 Frame Synchronization

To the author's knowledge, no work has been published so far that tackles optimum frame synchronization for block-oriented processing of preamble-less packets in a similar way as it has been done for traditional frame synchronization [Mas72] [Nie73] [LT87] [Sch80].

The only work presented at the time of writing is that of Schaub and Hansson [SH90] which we have briefly encountered earlier and the recently presented paper by Mehlan and Meyr [MM93] [SH90]. Both assume a preamble at the start of the packet, although the latter makes use of ability to store the packet. However, it is argued by those who propagate the store-and-process philosophy, that a preamble is not required. Filling this gap will be one of the major contributions of the work presented, it is the subject of chapter 8.

Chapter 8

Frame Synchronization of Packets in a Time-Slot - Derivation of Likelihood Functions

In this chapter we will derive the optimal MAP frame synchronizers for coherent demodulation as well as DPSK for packets that are transmitted in time slots. The MAP approach also yields an indication of whether or not a packet was actually present in a slot. The optimal synchronizer uses both the correlation with the sync word *and* the energy of the packet to correctly establish the position of the packet. In addition, we present a high SNR approximation to the MAP rule which, as expected performs almost as well, but is easier to implement. Finally, we illustrate a method of implementing the optimum rules that requires relatively few components by using a recursive definition of the likelihood functions.

8.1 The Likelihood Function

Fig. 7.1 shows a packet that consists of $N-L$ data symbols preceded by L sync symbols, randomly positioned in a slot of length G. Our synchronizer must make an estimate $\hat{\mu}$ of the real position $\tilde{\mu}$ within the time-slot of length G, where the packet starts.

The modulation scheme is M-ary, demodulation is again assumed to be coherent with perfect symbol timing. For the moment, we look at the case where a packet has actually been transmitted. The set of transmitted symbols is $\{W_j, 0 \leq j \leq M-1\}$. The packets are N-symbols long, of which the first L symbols are a known sync word $\vec{S} = (S_0, S_1, ..., S_{L-1})$. The symbols in the packet data sequence, $\vec{d} = (d_L, d_1, ..., d_{N-1})$, are assumed to be chosen randomly and uniformly from the signal set. The demodulator output consists of a sequence of G complex vectors, the sampled matched filter outputs. Let this sequence (a random variable) be denoted by $\vec{\mathbf{y}}$, the actual value for any one case is the sequence $\vec{y} = (y_0, y_1, ..., y_{G-1})$, and is composed of either just noise or the sum of data symbols and noise. Transmission is over an additive white Gaussian noise channel (AWGN) where the real and imaginary components of the noise samples are i.i.d with zero mean and variance $N_0/2$: N_0 is the one-sided power spectral density.

The frame synchronizer should generate the value $\hat{\mu}$ of μ in $(0, 1, ..., G-N)$, which maximizes

$$Pr\{\tilde{\mu} = \mu | \vec{\mathbf{y}} = \vec{y}\}, \tag{8.1}$$

(MAP approach). Using the mixed Bayes' rule, we can maximize

$$f_{\vec{\mathbf{y}}}(\vec{y}|\tilde{\mu} = \mu) \cdot Pr\{\tilde{\mu} = \mu\} = \sum_{\forall\, \vec{d}} f_{\vec{\mathbf{y}}}(\vec{y}|\tilde{\mu} = \mu, \vec{d}) \cdot M^{-(N-L)} \cdot Pr\{\tilde{\mu} = \mu\}. \tag{8.2}$$

The probability of any packet data sequence $\vec{d}$, is $M^{-(N-L)}$. The individual products are

$$f_{\vec{\mathbf{y}}}(\vec{y}|\tilde{\mu} = \mu) \cdot Pr\{\tilde{\mu} = \mu\} = \frac{1}{(\pi N_0)^G \cdot M^{(N-L)}} \prod_{i=0}^{\mu-1} e^{\frac{-\|y_i\|^2}{N_0}} \cdot \prod_{i=\mu}^{\mu+L-1} e^{\frac{-\|y_i - S_{i-\mu}\|^2}{N_0}} \cdot$$
$$\sum_{\forall\, \vec{d}} \prod_{i=\mu+L}^{\mu+N-1} e^{-\frac{\|y_i - d_{i-\mu}\|^2}{N_0}} \cdot \prod_{i=\mu+N}^{G-1} e^{\frac{-\|y_i\|^2}{N_0}} \cdot Pr\{\tilde{\mu} = \mu\}, \tag{8.3}$$

the first and last products take into account the noise before and after the packet; the second and third products correspond to the sync word and packet data sequence. We can equivalently maximize the likelihood function

$$L_1(\mu) = \prod_{i=0}^{G-1} e^{-\frac{\|y_i\|^2}{N_0}} \cdot \prod_{i=\mu}^{\mu+L-1} e^{\langle y_i, S_{i-\mu}\rangle \frac{2}{N_0}} \cdot \sum_{\forall\, \vec{d}} \prod_{i=\mu+L}^{\mu+N-1} e^{\frac{2}{N_0}\langle y_i, d_{i-\mu}\rangle - \frac{\|d_{i-\mu}\|^2}{N_0}} \cdot Pr\{\tilde{\mu} = \mu\}. \tag{8.4}$$

The first term is no function of μ and can be disregarded. Rearranging indices and a similar argument as in Part I (see also [Rob94]) yields

$$L_2(\mu) = \prod_{i=0}^{L-1} e^{\frac{2}{N_0}\langle y_{i+\mu}, S_i\rangle} \cdot \prod_{i=L}^{N-1} \sum_{j=1}^{M} e^{\frac{2}{N_0}\langle y_{i+\mu}, W_j\rangle - \frac{\|W_j\|^2}{N_0}} \cdot Pr\{\tilde{\mu} = \mu\}. \tag{8.5}$$

At this point we must be careful. Interestingly, (8.5) is exactly the result obtained at this stage for the traditional frame sync problem, where in the next step we are able to divide it by

$$\prod_{i=0}^{N-1} \sum_{j=1}^{M} e^{\frac{2}{N_0}\langle y_{i+\mu}, W_j\rangle - \frac{\|W_j\|^2}{N_0}}, \tag{8.6}$$

because periodic frame boundaries make the value of (8.6) independent of μ. Observe that in our case, we can only divide (8.5) by any number of the terms $z = \sum_{j=1}^{M} e^{\frac{2}{N_0}\langle y_z, W_j\rangle - \frac{\|W_j\|^2}{N_0}}$ (they are no function of μ). One can find those elements z, which if we divide into (8.5), reduce the number of $\ln\sum\cdots$ operations needed below. For the moment, however, we will not divide (8.5) at all; manipulating further and taking the logarithm leads us to the likelihood function:

$$L(\mu) = \sum_{i=0}^{L-1}\langle y_{i+\mu}, S_i\rangle + \frac{N_0}{2}\sum_{i=L}^{N-1}\ln\sum_{j=1}^{M} e^{\frac{2}{N_0}\langle y_{i+\mu}, W_j\rangle - \frac{||W_j||^2}{N_0}} + \frac{N_0}{2}ln(Pr\{\tilde{\mu} = \mu\}). \tag{8.7}$$

The terms can be explained in the following way: The first is the well known correlation term ($L_C(\mu)$). The second term accounts for the random data following the sync word. The last term accounts for the likelihood of the event that a packet was sent and began at $\tilde{\mu}$. The correlation term provides a kind of 'local protection' of the correct position due to the sharp peak of the sync word's autocorrelation function, whereas the second term that takes the rest of the packet into account ensures that only those μ have a high likelihood function which are close to $\tilde{\mu}$. This will become clear in chapter 9.

Let us now address the case that **no** packet was sent: we call this event NT. Detection theory requires that we assign costs to all possible incorrect detection cases, and then arrive at a receiver detection algorithm that minimizes the average cost. Hence, we would need to assign a cost to the false alarm case (receiver decides that a packet was sent when in fact it was not) and also the miss case (receiver decides that no packet was sent when, in fact, there was a packet). If these costs are assumed to be equal, then our detection problem can again be expressed with the MAP rule where we now need to make a decision on $G - N + 2$ possible events ($G - N + 1$ possible $\tilde{\mu}$, one event NT). We shall follow this approach and continue by calculating the PDF of the received vector conditioned on the case that no packet was sent, weighted with the a-priori probability of NT:

$$f_{\vec{\mathbf{y}}}(\vec{y}|NT)\cdot Pr\{NT\} = \frac{1}{(\pi N_0)^G}\prod_{i=0}^{G-1} e^{\frac{-|y_i|^2}{N_0}}\cdot Pr\{NT\}, \tag{8.8}$$

and following the above derivation, remembering when simplifying (i.e. multiplying by and adding constants) that we will wish to compare it with $L_{\mathrm{MAP}}(\mu)$, yields

$$L(NT) = \frac{1}{2}\sum_{i=0}^{L-1}||S_i||^2 + \frac{N_0}{2}\left[\ln(M)\cdot(N-L) + \ln(Pr\{NT\})\right]. \tag{8.9}$$

If we now wish to make a decision on the $G - N + 2$ possible events ($G - N + 1$ possible $\tilde{\mu}$, one event NT), we must simply compare the $G - N + 2$ likelihood functions (8.7) and (8.9). Alternatively, one can choose the highest $L(\mu)$ and subsequently compare the highest value with a threshold $T = L(NT)$ since (8.9) can be pre-computed as it does not depend on the received sequence $\vec{y}$. If the threshold is not exceeded, then it is most likely that no packet was sent (event NT). The higher the signal-to-noise ratio, the lower the threshold becomes, Also, the ratio of the probability of the events NT and $\tilde{\mu}$ plays a role: the more likely NT is, the higher the threshold[1].

[1] In practice, however, the term $\frac{N_0}{2}\ln(M)\cdot(N-L)$ dominates in the threshold.

8.2 Interpretation

In analogy to chapter 3 we shall now give an intuitive interpretation of why the new synchronizer is optimal, and what is 'wrong' with the correlation rule. We show the result of a simulation for both the MAP and correlation rules. In Fig. 8.1 we see the value of the likelihood functions as a function of μ. In the simulation $\tilde{\mu} = 14$, $L = 7$ and $G - N = 31$. Notice how the sync word gives a good *local protection* due to a low partial auto-correlation function, whereas the energy of the packet data sequence gives *global protection* of $\tilde{\mu}$ against sync errors, the correlation rule alone is susceptible to sync errors 'further out' from the correct position. This explains the relatively low influence of changes of $G-N$ on the performance of the MAP rule.

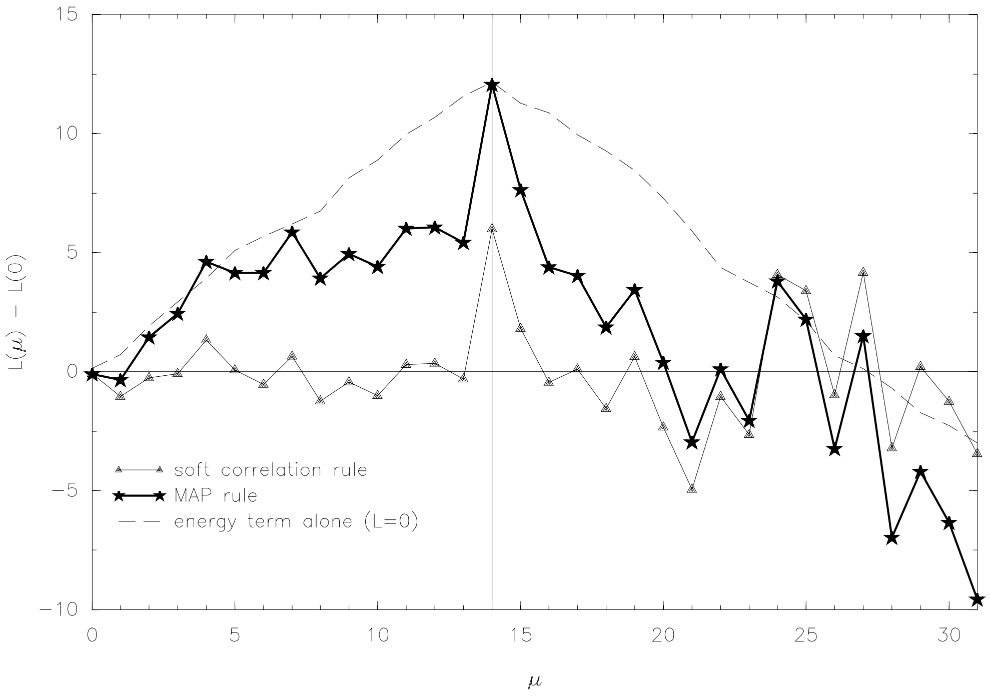


Figure 8.1: Illustration of local and global protection against sync errors. $\tilde{\mu} = 14$, $L = 7$ and $G - N = 31$. Shown are the values of the likelihood function for the MAP and correlation rules for one simulation. The dotted line shows the correction term alone (L=0). The sync word gives a good local protection due to a low partial auto-correlation function, whereas the energy of the packet data portion gives global protection of $\tilde{\mu}$.

8.2.1 Geometric Interpretation (Coding Theory Approach)

We shall now give a geometric interpretation of the frame sync task. We shall treat the problem as being the transmission of one of $G-N+1$ possible $\tilde{\mu}$ ('information' to be transmitted), see Fig. 8.2. The 'information' is coded onto a G dimensional (complex) vector -the G symbols of the packet within the time slot. The mapping is not one-to-one, but one-to-$M^{(N-L)}$, since the packets contain $N-L$ random data symbols too. The values of these $N-L$ data symbols are no function of $\tilde{\mu}$. However, the mapping is invertible as each position μ is mapped onto a unique cluster of possible points. The data symbols of the packet determine where exactly within each cluster the transmitted point lies: they determine the values of the sub-coordinates $(\tilde{\mu}+L, ..., \tilde{\mu}+N-1)$ of the G dimensional space. The values of the sub-coordinates $(\tilde{\mu}, ..., \tilde{\mu}+L-1)$ are determined by the sync word. The values of the unused coordinates are zero in all cases, since the mapping is done before the noisy transmission channel, of course. The MAP rule for synchronization now does the following: given the received G dimensional point $\vec{y}$, it simply calculates the sum of the conditional probabilities ($\equiv$ squared Euclidean distance for the ML rule) over all M^{N-L} possible data sequences within one cluster (one μ); the resulting probability is compared over all clusters and the highest is selected. We shall present the high SNR rule in the following section, this effectively determines only the correlation with the *closest* data sequence, the underlying assumption being that this one data sequence dominates over all the others in the sum over their conditional probabilities.

In Fig. 8.2 we have also tried to illustrate the underlying 'code structure': Clusters are far apart if the corresponding $\tilde{\mu}$ are far apart, this is because the data symbols are mapped to a different subset of $N-L$ coordinates. Additionally, the good partial auto-correlation of the sync word ensures that the distance between clusters corresponding to neighboring $\tilde{\mu}$ are reasonably far apart. Furthermore, since the clusters do not overlap, it becomes clear that as noise-free transmission makes the received vector identical to the transmitted vector, the mapping guarantees protection against false synchronization, in other words a MAP or ML synchronizer ('decoder!') would have a $Pr\{f|\text{RDL}\}$ of zero. In the case of traditional frame synchronization, however, the clusters *do* overlap, the degree of overlapping becoming smaller as the sync word becomes longer.

But what does the correlation rule do wrong? It simply only looks at the sub-coordinates corresponding only to the sync word: $(\mu, ..., \mu+L-1)$, of each cluster. The 'distance' is thus evaluated only along a μ-variant subset of the G dimensions of the 'code space'. The correlation rule is equivalent to the ML rule only in the trivial case when the random data sequence has zero elements.

8.3 Simpler Likelihood Function

As before, a very useful approximation to (8.7) can be given for high signal-to-noise ratios. Here we show the result only for transmission where $||W_j||$ are equal for all j (e.g. PSK):

$$L_H(\mu) = \sum_{i=0}^{L-1} \langle y_{i+\mu}, S_i \rangle + \sum_{i=L}^{N-1} \langle y_{i+\mu}, W_{\hat{j}} \rangle, \qquad (8.10)$$

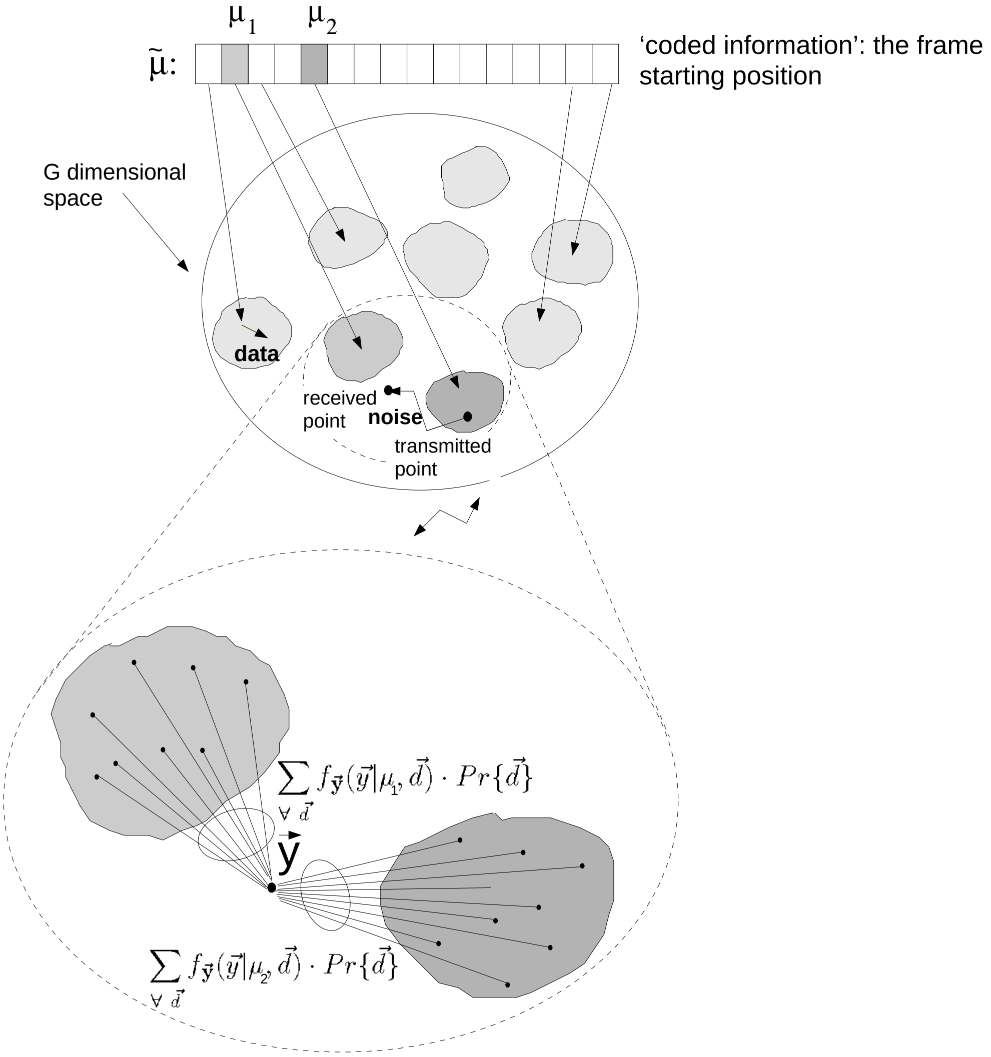

μ1
μ2
μ̃:
'coded information': the frame starting position
G dimensional space
data
received point
noise
transmitted point
$\sum_{\forall \vec{d}} f_{\vec{\mathbf{y}}}(\vec{y}|\mu_1, \vec{d}) \cdot Pr\{\vec{d}\}$
$\vec{y}$
$\sum_{\forall \vec{d}} f_{\vec{\mathbf{y}}}(\vec{y}|\mu_2, \vec{d}) \cdot Pr\{\vec{d}\}$

Figure 8.2: *Geometric illustration of the sync task. We have tried to see the problem as one similar to coding. The information to be transmitted is the frame starting position $\tilde{\mu}$ and it is coded onto non-overlapping clusters of points in a G dimensional space. It is the data symbols of the packet that determine exactly which point is transmitted. The receiver has to decide which cluster is closest (ML rule) to the received point represented by $\vec{y}$ -or a-posteriori most likely (MAP rule). Shown in the bottom is only the evaluation of two clusters.*

where $\hat{j} = \hat{j}(i)$ is that j which maximizes $\langle y_{i+\mu}, W_j \rangle$. $L_H(\mu)$ is easier to implement than $L(\mu)$, requires fewer operations and needs no estimate of N_0. However, the associated threshold -which we have omitted for brevity- still requires that N_0 be estimated.

8.4 Likelihood Function for DBPSK

Earlier in this work (section 3.3), we derived the close-to optimum frame sync rule for the traditional frame sync problem with differential binary detection. It was necessary to obtain the PDFs

$$f_{\mathbf{y}}(y_i|\pm 1), \tag{8.11}$$

of the decision variable $y_i = 2\mathrm{R}e\{y_i^D y_{i-1}^{D*}\}$. Since we need to look at that part of the slot containing no modulation, we additionally need the PDF

$$f_{\mathbf{y}}(y_i|0) = \frac{1}{2N_0}\exp(-|y_i|/N_0). \tag{8.12}$$

If we do not consider the two cases where a transition leading to a y_i falls on the border between noise and the packet, the above PDFs can be inserted into:

$$\begin{aligned} L_1(\mu) = &\prod_{i=0}^{\mu-1} f_{\mathbf{y}}(y_i|0) \prod_{i=\mu}^{\mu+L-1} f_{\mathbf{y}}(y_i|S_{i-\mu}) \cdot \sum_{\forall\ \vec{d}} \prod_{i=\mu+L}^{\mu+N-1} f_{\mathbf{y}}(y_i|d_{i-\mu}) \\ &\prod_{i=\mu+N}^{G-1} f_{\mathbf{y}}(y_i|0) \cdot Pr\{\tilde{\mu} = \mu\}. \end{aligned} \tag{8.13}$$

After taking the logarithm and adding $\sum_{i=0}^{G-1} |y_i|/N_0$ we arrive at an approximation to the optimal likelihood function since we have again neglected the statistical dependence of the y_i:

$$\begin{aligned} L_A(\mu) = &\sum_{i=0}^{L-1} \ln(f_{\mathbf{y}}(y_{i+\mu}|S_i)) + \frac{|y_{i+\mu}|}{N_0} + \sum_{i=L}^{N-1} \ln \sum_{j=1}^{2} f_{\mathbf{y}}(y_{i+\mu}|W_j) + \frac{|y_{i+\mu}|}{N_0} \\ &+ \ln(Pr\{\tilde{\mu} = \mu\}); \end{aligned} \tag{8.14}$$

observe that the first and last product in (8.13) have disappeared after the addition of $\sum_{i=0}^{G-1} |y_i|/N_0$. Fortunately, we have observed that the following high SNR approximation performs almost identically:

$$L_H(\mu) = \sum_{i=0}^{L-1} \max(0, y_{i+\mu}/2 \cdot S_i) + \sum_{i=L}^{N-1} |y_{i+\mu}|/2 + 2 \cdot N_0 \cdot \ln(Pr\{\tilde{\mu} = \mu\}), \tag{8.15}$$

motivated first through graphical inspection of the functions $\ln(f_{\mathbf{y}}(y_i|\pm 1)) + \frac{|y_i|}{N_0}$, then confirmed by simulations of (8.14) and (8.15).

8.5 Implementation

Let us now come back to the MAP synchronizer for coherent detection. If we divide (8.5) by the following term that is independent of μ:

$$\prod_{i=L}^{N-1}\sum_{j=1}^{M} e^{\frac{2}{N_0}\langle y_i,W_j\rangle-\frac{||W_j||^2}{N_0}}, \tag{8.16}$$

then we obtain a recursive rule:

$$L(\mu=0)=\sum_{i=0}^{L-1}\langle y_i,S_i\rangle+\frac{N_0}{2}ln(Pr\{\tilde{\mu}=0\}), \tag{8.17}$$

$$\begin{aligned}L(\mu)=\sum_{i=0}^{L-1}\langle y_{i+\mu},S_i\rangle+\frac{N_0}{2}\ln\sum_{j=1}^{M}e^{\frac{2}{N_0}\langle y_{N+\mu-1},W_j\rangle-\frac{||W_j||^2}{N_0}}+L^o(\mu-1)-\\ \frac{N_0}{2}\ln\sum_{j=1}^{M}e^{\frac{2}{N_0}\langle y_{L+\mu-1},W_j\rangle-\frac{||W_j||^2}{N_0}}+\frac{N_0}{2}ln(Pr\{\tilde{\mu}=\mu\}),\end{aligned} \tag{8.18}$$

where

$$L^o(\mu-1)=L(\mu-1)-\sum_{i=0}^{L-1}\langle y_{i+\mu-1},S_i\rangle-\frac{N_0}{2}ln(Pr\{\tilde{\mu}=\mu-1\}). \tag{8.19}$$

Note that it will not be possible to use our threshold T, due to the division made; i.e. we cannot make such a reliable decision *whether* a packet has been sent or not (still useful for TDMA or reservation access protocols if we know a slot contains a packet). But the number of operations needed is not very high, in fact, the correlation term is the most intensive. If we still wish to use the threshold, then we must initially evaluate

$$L(\mu=0)=\sum_{i=0}^{L-1}\langle y_i,S_i\rangle+\sum_{i=L}^{N-1}\frac{N_0}{2}\ln\sum_{j=1}^{M}e^{\frac{2}{N_0}\langle y_i,W_j\rangle-\frac{||W_j||^2}{N_0}}+\frac{N_0}{2}ln(Pr\{\tilde{\mu}=0\}). \tag{8.20}$$

In other words, we initialize the recursive formula with the *full* correction term, and our recursive definition corresponds exactly to the MAP rule.

Similar formulae can be obtained for the high SNR rules, and for DBPSK. A block diagram of the implementation of the high SNR approximation of (8.20) and (8.18) for BPSK is shown in Fig. 8.3. Although (8.18) looks involved, the high SNR synchronizer simply adds the correlation component for each μ to the recursively evaluated sum over the absolute values of the symbols of the data portion of the packet.

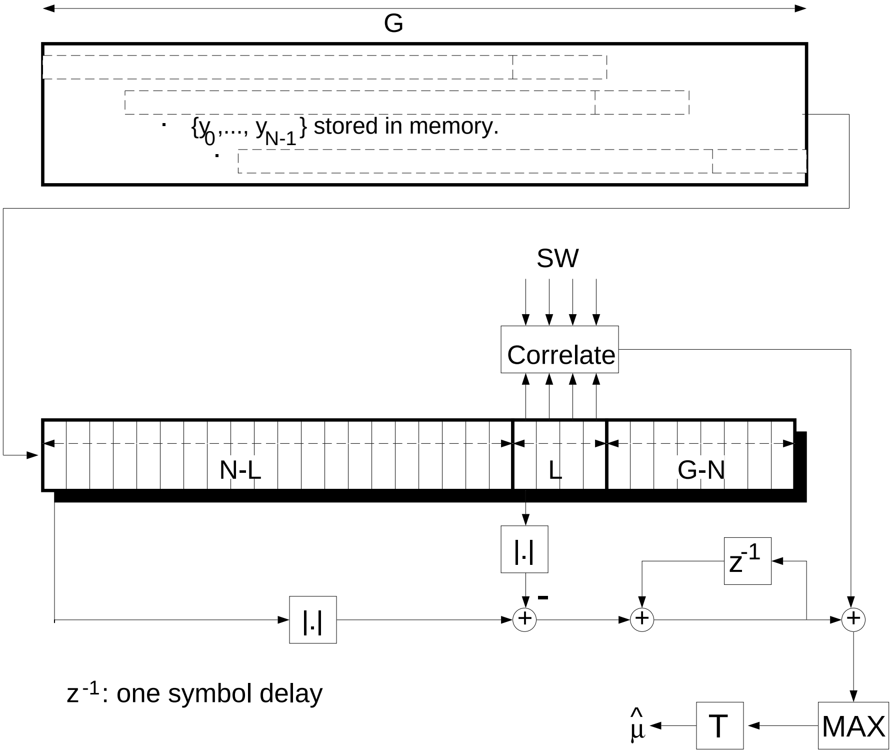


Figure 8.3: Realization of the high SNR approximation for BPSK for the recursive definition (8.20) and (8.18). The G long received vector $\vec{y}$ is applied at the input on the left. Each symbol time T_s, the memory is shifted one to the right and a new element of $\vec{y}$ is input. One can see that the 'energy' correction term from the packet data sequence is added to the correlation term. T denotes the threshold test to decide whether a packet had been sent or not.

8.6 Other Extensions

As in the case of traditional frame synchronization we can extend these likelihood functions for preamble-less packets to include:

1. Demodulation with phase ambiguity.
2. Coded data (list synchronizer).
3. Use of trellis termination information.
4. Inclusion of channel state information.

For further details, the reader is referred to sections 3.4, 3.5, 3.6 and 3.7, respectively. The extensions are straightforward, and the equations are omitted here for brevity. Case 1 above simply involves taking the largest correlation term (e.g. the absolute value of the correlation for π radians phase ambiguity). The principle of the list synchronizer (case 2) is unaltered in practice, but simplified in theory as now the packet data sequence is identical to the observation data sequence. Case 3 is easily extended to the packet sync case by shortening the random data component of the likelihood function by the termination (and/or opening) portions of the coded sequence and adding the new correlation terms corresponding to the closest termination (and/or opening) sequence, see (3.39) and Fig. 3.8. In practice, case 4 will not be of much interest, since the availability of reliable channel state information (both phase for coherent detection and amplitude) is questionable for packet transmission, especially since differentially coherent frame synchronization is possible (and quite reliable, as we shall see), possibly making channel estimation and coherent detection redundant as far as frame synchronization is concerned.

8.7 Sporadic, Preamble-Less Packets

So far, we have addressed the case of synchronizing a packet in a known slot of length G. What is, however, the optimal synchronizer for sporadic transmission of preamble-less packets? We can choose a slot of length $G > N$ at times when a packet seems likely. If we continuously monitor the energy of the matched filter output and choose a large enough artificial slot around the event of an energy threshold being exceeded, we can then perform symbol timing recovery and frame synchronization. The length of the slot chosen can be quite large, since the optimum and high SNR synchronizers do not degrade significantly when the slot length increases, as we will see in the following chapter.

Chapter 9

Performance Evaluation of the Synchronizers for Packets in a Time-Slot

In this chapter we proceed in a very similar way as in chapter 4. We begin by extending the union and random data limited bounds to the soft correlation rule. This is quite straightforward though a little more involved since we can no longer use symmetry arguments and assume $\tilde{\mu} = 0$, but must sum over all possible $\tilde{\mu}$. We will see that the high SNR and optimal synchronizers presented in the last chapter have a synchronization failure rate of zero at infinite SNR. For the high SNR synchronizer and soft correlation rule modulation we will again derive an approximate union bound for BPSK with and without phase ambiguity, that simulations will show to be very tight.

The second part of the chapter presents some simulations for various packet and slot lengths and coherent and differentially coherent detection. Furthermore, we will show what can be achieved when the data is convolutionally encoded and we use the synchronizer that makes use of the termination of the trellis. To conclude, we shall shed some light upon the 'workings within' the frame synchronizer and in particular the influence of the choice of the sync word on the performance. These intuitive findings will be substantiated by the effectiveness of sync word design using the newly developed bounds as optimality criteria.

9.1 Random Data Limited Bound for the Soft Correlation Rule

It is interesting to look at the synchronization rate for very high SNR. The soft and hard correlation rules are limited in performance for very high SNR, the optimal and high SNR rules actually never fail to synchronize correctly in a noise free environment. The latter fact becomes immediately clear when looking at the structure of the likelihood function: when $\mu = \tilde{\mu}$, the likelihood functions (8.7) and (8.10) must reach a maximum value, because the correction term (due to the random data part) reaches its maximum value for $\mu = \tilde{\mu}$ at infinite SNR. In chapter 4 the synchronization failure probability of a frame synchronizer (traditional frame sync problem) in the noiseless case was given: $Pr\{f|\mathrm{RDL}\}$ (**R**andom **D**ata **L**imited). We can easily extend the result to a synchronizer that uses the soft correlation rule to determine the starting position of a packet. We shall now show that

$$Pr\{f|\mathrm{RDL}\} = 1 - \frac{1}{G-N+1}\left[\sum_{\tilde{\mu}=0}^{G-N-L}\sum_{j=0}^{Q}\frac{1}{j+1}D_j(\tilde{\mu})M^{-(G-N-\tilde{\mu})} + L\right], \tag{9.1}$$

where $Q = \lfloor (G-N-\tilde{\mu})/L \rfloor$. $D_j(\tilde{\mu})$ is the number of possible data sub-sequences of length $G-N-\tilde{\mu}$ in which exactly j occurrences of the sync word occur, the probability of any such one sub-sequence is $M^{-(G-N-\tilde{\mu})}$. The extension that has been made is to take into account the dependence of D_j on $\tilde{\mu}$; we have to consider those $\tilde{\mu} \leq G-N-L$ where a number of 'competing' μ' can lead to a sync error. If $\tilde{\mu} > G-N-L$, there can never be a μ' that will lead to a sync error, there are exactly L such 'safe' $\tilde{\mu}$, since a potential false sync word has insufficient space in the random data segment. $\frac{1}{G-N+1}$ is the probability of each possible $\tilde{\mu}$ -we have assumed them equi-probable. As before, $D_j(\tilde{\mu})M^{-(G-N-\tilde{\mu})}$ can be defined recursively by:

$$D_j(\tilde{\mu})M^{-(G-N-\tilde{\mu})} = \binom{G-N-\tilde{\mu}-(L-1)j}{j}M^{-Lj} - \sum_{i=j+1}^{Q}\frac{D_i}{M^{G-N-\tilde{\mu}}}\binom{i}{j} \tag{9.2}$$

with

$$D_Q = \binom{G-N-\tilde{\mu}-(L-1)Q}{Q}M^{G-N-\tilde{\mu}-LQ}, \tag{9.3}$$

which can be inserted into (9.1).

9.2 Union Upper Bounds for the Synchronization Failure Rate in the Case of Noise and for BPSK

In this section we shall assume that the packet begins with the sync word. Although other cases can, in principle, be treated in the same way, the analysis becomes awkward, to say the least, because each element of the sync word must be treated differently according to its location. Let us first take a look at Fig. 9.1 which shows the possible cases we must distinguish in oder to correctly treat all the cases where the sync word of the packet overlaps that of the 'competing' position. There are four cases

I The competing sync word lies in noise.

II The competing sync word lies partly in noise and partly overlaps with the true one.

III The competing sync word partly overlaps the true sync word and partly overlaps the data.

IV The competing sync word lies just in the packet data sequence.

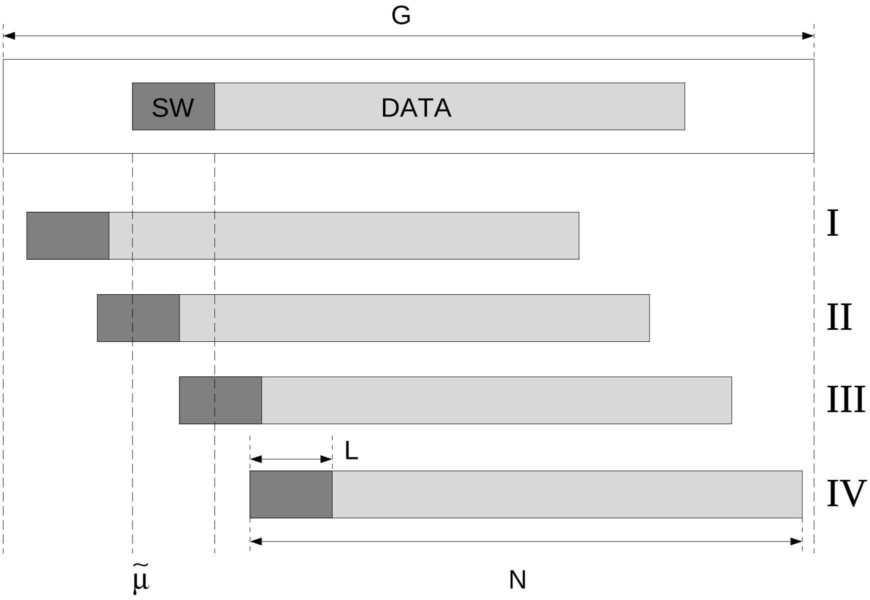


Figure 9.1: *Four possible cases where competing positions μ' may lie in relation to $\tilde{\mu}$.*

There is also a fifth case that applies to slot lengths $G \geq 2N - L$, which we do not consider in this analysis. To proceed with the derivation of the union bound we must take into account the overlap cases just as we have done for the traditional frame sync task. The situation is made more complex here, because the length of the overlap regions depends not only on $\tilde{\mu}$ but also on the competing position μ', in fact the term $\tilde{\mu} - \mu'$ is the key, the sign and magnitude of which determines which of the above four cases applies. For the following, we define $\Delta\mu = |\tilde{\mu} - \mu'|$.

9.2.1 Soft Correlation Rule

The analysis is quite straightforward and similar to the traditional frame sync problem. The difference between competing likelihood functions, ΔL_C, is again Gaussian distributed, and we have given the values of $m_{\Delta L_C}$ and $\sigma^2_{\Delta L_C}$ for each of the four cases, in Table. 9.1.

All that remains to be done is to determine the number of each possible occurring $\Delta\mu$. Since the packet is constrained to lie within the time slot it is easy to show that there are exactly

$$n_{\Delta\mu} = \max(G - N - \Delta\mu + 1, 0) \tag{9.4}$$

Table 9.1: *Means and variances needed for the union bound for the soft correlation rule.*

Case	$m_{\Delta L_C}$	$\sigma^2_{\Delta L_C}$
I	L	LN_0
II	$L - R_{\Delta\mu}$	$N_0(L - R_{\Delta\mu})$
III	$L - (R_{\Delta\mu} + \kappa_{\Delta\mu})$	$N_0(L - R_{\Delta\mu})$
IV	$L - \kappa_L$	LN_0

of each $\Delta\mu$. Assuming that all $\tilde{\mu}$ are equally likely[1], the union bound for the soft correlation rule becomes

$$\begin{aligned} Pr\{f\} \leq \frac{1}{2 \cdot (G - N + 1)} &\left\{ \sum_{\Delta\mu = L}^{G-N} n_{\Delta\mu} \left[\operatorname{erfc}\left(\frac{L}{\sqrt{2LN_0}} \right) + \right.\right. \\ &\left. \sum_{\forall\kappa_L} Pr\{\kappa_L\} \cdot \operatorname{erfc}\left(\frac{L - \kappa_L}{\sqrt{2LN_0}} \right) \right] + \\ &\sum_{\Delta\mu=0}^{L-1} n_{\Delta\mu} \left[\operatorname{erfc}\left(\frac{L - R_{\Delta\mu}}{\sqrt{2N_0(L - R_{\Delta\mu})}} \right) + \right. \\ &\left.\left. \sum_{\forall\kappa_{\Delta\mu}} Pr\{\kappa_{\Delta\mu}\} \cdot \operatorname{erfc}\left(\frac{L - (R_{\Delta\mu} + \kappa_{\Delta\mu})}{\sqrt{2N_0(L - R_{\Delta\mu})}} \right) \right] \right\}. \end{aligned} \tag{9.5}$$

The four terms correspond to the cases I, IV, II and III respectively. The factor $\frac{1}{2 \cdot (G-N+1)}$ is due to the error function and the identical a-priori probability of each $\tilde{\mu}$. In cases III and IV we have to sum over the distribution of the partial auto-correlation of the sync word with random data, $\kappa_{\Delta\mu}$ and κ_L. Intuitively satisfying is that the bound depends only on $G - N$ and not N or G alone.

9.2.2 Approximation for the High SNR Rule

Again we shall orient ourselves with the aid the four cases I to IV. The procedure of finding the union bound again involves evaluation of means and variances, resolving dependencies, and finally the application of the central limit theorem as in section 4.2.3. This is easiest if we write down the ΔL_H in such a way that we can combine or cancel dependent terms when determining the mean and variance of ΔL_H. We shall demonstrate this for case III, which is the most complicated:

$$\Delta L_H = \sum_{i=0}^{\mu' - \tilde{\mu} - 1} y_{i+\tilde{\mu}} \cdot S_i + \sum_{i=\Delta\mu}^{L-1} y_{i+\tilde{\mu}} \cdot S_i + \sum_{i=\tilde{\mu}+L}^{\tilde{\mu}+L+\Delta\mu-1} |y_i| -$$

[1] An extension to the general case is quite straightforward, although it makes the resulting bound rather cumbersome.

$$\sum_{i=\Delta\mu}^{L-1} y_{i+\tilde{\mu}} \cdot S_{i-\Delta\mu} - \sum_{i=\tilde{\mu}+L}^{\tilde{\mu}+L+\Delta\mu-1} y_i \cdot S_{i-\mu'} - \sum_{i=\tilde{\mu}+N}^{\tilde{\mu}+N-1} |y_i|. \tag{9.6}$$

The six terms correspond to A,B,C,D,E and F in Fig. 9.2. Terms B and D as well as C and E must be analyzed together to take into account dependencies. For the mean $m_{\Delta L_H}$ and variance $\sigma^2_{\Delta L_H}$ this yields

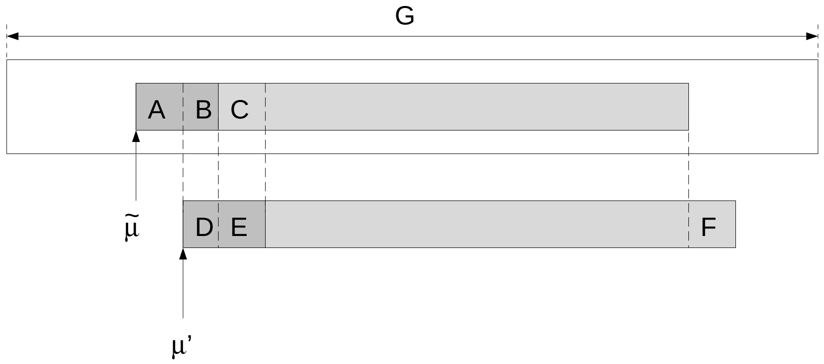


Figure 9.2: *Terms resulting from case III for the approximation for the high SNR rule. Terms B and D as well as C and E must be analyzed together to take into account dependencies.*

$$m_{\Delta L_H} = L - R_{\Delta\mu} - \left(\frac{\Delta\mu + \kappa_{\Delta\mu}}{2}\right) m_{H+} - \left(\frac{\Delta\mu - \kappa_{\Delta\mu}}{2}\right) m_{H-} - \Delta\mu \cdot m_n, \tag{9.7}$$

and

$$\sigma^2_{\Delta L_H} = \frac{N_0}{2}(2L - 2R_{\Delta\mu} - \Delta\mu) + \left(\frac{\Delta\mu + \kappa_{\Delta\mu}}{2}\right)\sigma^2_{H+} + \left(\frac{\Delta\mu - \kappa_{\Delta\mu}}{2}\right)\sigma^2_{H-} + \Delta\mu \cdot \sigma^2_n. \tag{9.8}$$

Some terms used here and also later still need explanation: m_n is the mean of the absolute value of a noise sample, $|n_i|$; similarly, σ^2_n is its variance. m_d and σ^2_d are the mean and variance of $|y_i|$ if $x_i \in \{-1, +1\}$, i.e. a data- or sync word-symbol had been transmitted.

After appropriate consideration of the remaining 3 cases, and again applying the central limit theorem to approximate the distribution of ΔL_H by a Gaussian distribution, we obtain the approximate union bound for the high SNR rule and BPSK:

$$Pr\{f\} \approx \frac{1}{2(G-N-1)} \left\{ \sum_{\Delta\mu=L}^{G-N} n_{\Delta\mu} \left[\operatorname{erfc}\left(\frac{L m_{H+} + \Delta\mu m_d - m_n(\Delta\mu - L)}{\sqrt{2L\sigma^2_{H+} + 2\Delta\mu\sigma^2_d + LN_0 + 2\sigma^2_n(\Delta\mu - L)}} \right) + \right.\right.$$

$$\sum_{\forall\kappa_L} Pr\{\kappa_L\} \cdot \operatorname{erfc}$$

$$\left.\left(\frac{L + m_d(\Delta\mu - L) - \left(\frac{L+\kappa_L}{2}\right) m_{H+} - \left(\frac{L-\kappa_L}{2}\right) m_{H-} - \Delta\mu \cdot m_n}{\sqrt{LN_0 + 2\sigma^2_d(\Delta\mu - L) + \sigma^2_{H+}(L+\kappa_L) + \sigma^2_{H-}(L-\kappa_L) + 2\Delta\mu\sigma^2_n}} \right) \right] +$$

$$\sum_{\Delta\mu=0}^{L-1} n_{\Delta\mu}\left[\operatorname{erfc}\left(\frac{L-\Delta\mu-R_{\Delta\mu}+\Delta\mu m_{H+}+\Delta\mu m_d}{\sqrt{N_0(2L-\Delta\mu-2R_{\Delta\mu})+2\Delta\mu(\sigma_{H+}^2+\sigma_d^2)}}\right)+\right. \tag{9.9}$$

$$\sum_{\forall\kappa_{\Delta\mu}} Pr\{\kappa_{\Delta\mu}\}\cdot\operatorname{erfc}$$

$$\left.\left.\left(\frac{L-R_{\Delta\mu}-\left(\frac{\Delta\mu+\kappa_{\Delta\mu}}{2}\right)m_{H+}-\left(\frac{\Delta\mu-\kappa_{\Delta\mu}}{2}\right)m_{H-}-\Delta\mu\cdot m_n}{\sqrt{N_0(2L-2R_{\Delta\mu}-\Delta\mu)+\sigma_{H+}^2(\Delta\mu+\kappa_{\Delta\mu})+\sigma_{H-}^2(\Delta\mu-\kappa_{\Delta\mu})+2\Delta\mu\sigma_n^2}}\right)\right]\right\}.$$

The terms can be interpreted as follows: The four erfc()'s correspond to the cases I, IV, II and III respectively. All that has been changed compared to the bound for the correlation rule, (9.5), is that the means and variances of ΔL_H are different. An important finding is again the fact that the bound depends only on $G-N$ and not N or G alone.

9.2.3 Extension for Demodulation of BPSK with Phase Ambiguity

In section 4.2.5 a π radians phase ambiguity after demodulation was incorporated into the bounds. We shall proceed in a similar way here, except that matters are more complicated.

For cases III and IV we proceed as before and

1. replace κ_L by $|\kappa_L|$,
2. replace $\kappa_{\mu'}+R_{\Delta\mu}$ by $|\kappa_{\Delta\mu}+R_{\Delta\mu}|$,
3. if $R_{\Delta\mu}$ or $\kappa_{\Delta\mu}$ stand alone, multiply them by $sign(\kappa_{\Delta\mu}+R_{\Delta\mu})$.

However, if the correlation term acts -wholly or partly- on noise, then the approximation (4.45),

$$\left|\sum_{i=0}^{L-1} y_{i+\mu}\cdot S_i\right| \approx sign\left(\sum_{i=0}^{L-1} x_{i+\mu}\cdot S_i\right)\cdot\sum_{i=0}^{L-1} y_{i+\mu}\cdot S_i, \tag{9.10}$$

does not hold, since $\sum_{i=0}^{L-1} y_{i+\mu}\cdot S_i$ will *not* have a high mean (in fact in case I it will have zero mean). Case I is straightforward, the correlation is only with noise, so we have to subtract from the mean of ΔL, the mean of the absolute value of a Gaussian random variable with zero mean and variance $N_0/2$: $\sqrt{LN_0/\pi}$. This then takes into account the non-zero mean of the absolute value of the 'competing' correlation term. Of course we also modify the variance of ΔL; the component LN_0 is replaced by $LN_0(1-2/\pi)$ (see appendix D). For case II, we make use of the inequality

$$|a+b|\leq|a|+|b|, \tag{9.11}$$

where $|a+b|$ is the correlation term of the competing position, and a is the correlation term from noise (like case I) and b is the correlation term with the first $L-\Delta\mu$ symbols of the real sync word. So,

$$|a| = \left|\sum_{i=0}^{\Delta\mu-1} n_{i+\mu'} \cdot S_i\right|, \tag{9.12}$$

and

$$|b| = \left|\sum_{i=\Delta\mu}^{L-1} y_{i+\mu'} \cdot S_i\right| \tag{9.13}$$

The mean and variance of $|a|$ are $\sqrt{\Delta\mu N_0/\pi}$ and $\Delta\mu N_0(1-2/\pi)$ respectively. The term $|b|$ can be treated similarly to the corresponding term of case III, by applying approximation (9.10).

The changes to the means and variances can be best examined by comparing the approximate bound without phase ambiguity (9.9) with the approximate bound with phase ambiguity (F.1) given in appendix F.

9.3 Simulation Results

9.3.1 Coherent Demodulation

In this section we shall present some results of Monte-Carlo simulations together with the evaluation of the bounds of the previous sections. The sync words are again taken from appendix E; for the correlation rules they are taken from Tables E.1 and E.2, for the high SNR and (optimal rules) from Tables E.3 and E.4 (see section 9.5).

In Fig. 9.3 we compare the optimal, high SNR and correlation rules for coherent BPSK. The parameters are $L=7$, $G-N=31$ (the bounds suggest that only $G-N$ is of significance, not N or G alone, this was confirmed by simulations). The first observation is the negligible difference between the high SNR and optimal rules. For this reason we will no longer consider the latter. Both correlation rules are clearly inferior. An interesting result is that the hard correlation rule performs *worse* than the soft correlation rule in contrast to the behavior in the traditional frame sync problem but more in accordance with our expectation that ‘soft decision is better than hard decision’. The bound (correlation rule) and approximation (high SNR rule) are shown to be quite accurate.

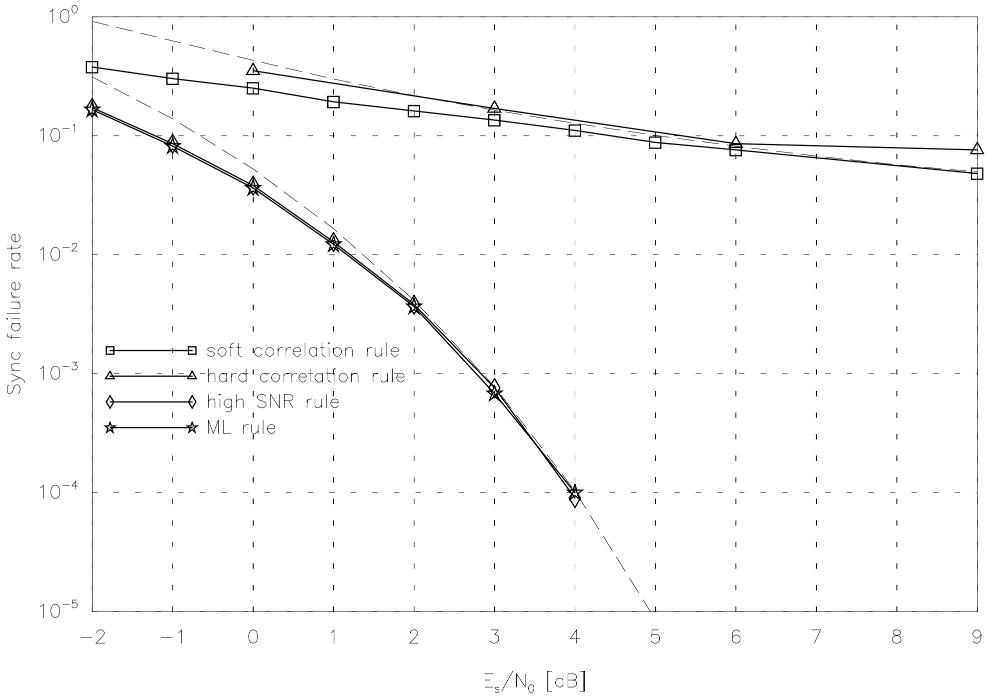


Figure 9.3: *Simulated frame synchronization performance.* $L = 7$, $G - N = 31$, BPSK. *We compare the ML, high SNR and correlation rules. The dotted lines show the results of the bounds (approximation for high SNR rule).*

9.3.1.1 Demodulation with phase ambiguity

Finally, we shall take a brief look at the loss due to synchronizing with a π radians phase ambiguity after demodulation -technically more relevant than with no phase ambiguity. The parameters are otherwise identical identical to those above. The result are given in Fig. 9.4. There is a penalty of an increase of the frame sync error rate by approximately factor two compared to BPSK without phase ambiguity.

9.3.1.2 Variation of the time-slot over-length

In Fig. 9.5 one can see the sync performance of the high SNR and soft correlation rules for BPSK (no phase ambiguity) as a function of $G - N$ for various SNR as well as $Pr\{f|\text{RDL}\}$ for the soft correlation rule. As $G - N$ gets larger, the correlation rule deteriorates in performance, whereas the new synchronizer hardly does.

This is in accordance with the intuitive interpretation given in section 8.2. The MAP and high SNR rules benefit from global protection via the energy term, and from local protection through the sharp auto-correlation of the sync word.

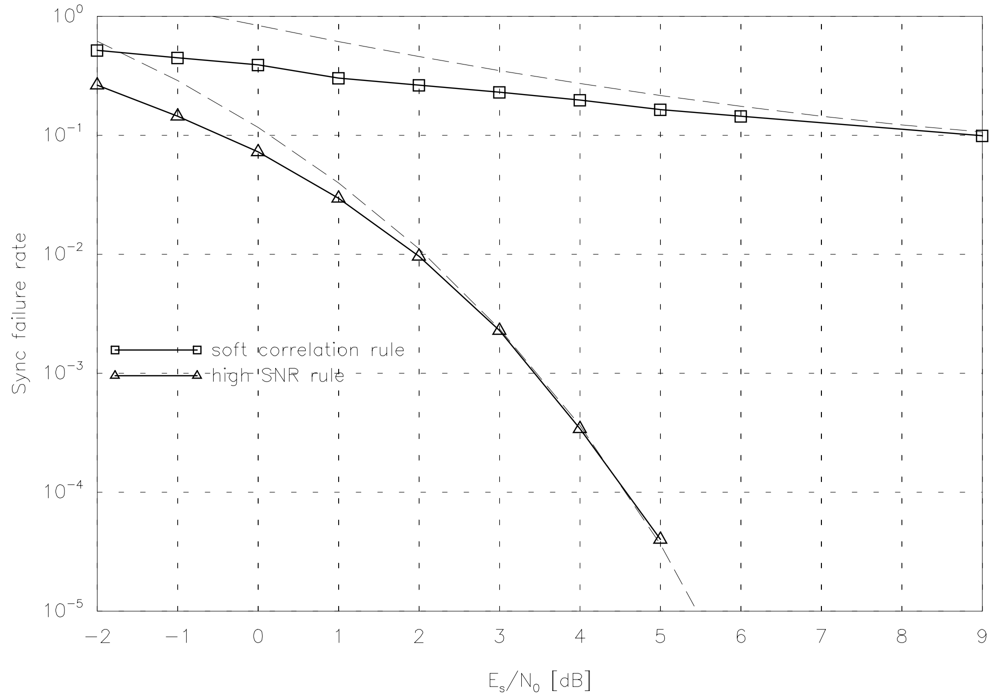


Figure 9.4: *Simulated frame synchronization performance.* $L = 7$, $G - N = 31$, *BPSK with phase ambiguity. We compare solely the high SNR and soft correlation rules. The dotted lines again show the results of the bounds (approximation for high SNR rule).*

9.3.1.3 Influence of the sync word length

The gains presented in the last sections may appear impressive, but the cautious reader will most probably suspect that the gain in terms of *actual saving in the sync word length* when replacing the soft correlation rule by the high SNR rule, is small, as is the case for the traditional frame sync problem. Fortunately, such pessimism is un-merited in the case of packet transmission, as we shall demonstrate in the sequel.

Figs. 9.6 and 9.7 show the sync failure rate (simulation) as a function of the sync word length L for different synchronizers. We have assumed coded data (convolutional code, rate 1/2, $m = 5$). Modulation is BPSK with π radians phase ambiguity, the signal-to-noise ratio is $E_s/N_0 = 2$ dB. The sync error rate includes errors due to incorrectly resolving the phase ambiguity. The various synchronizers are:

- Hard correlation rule
- Soft correlation rule.
- High SNR rule without trellis information.
- High SNR rule as list synchronizer with $\nu_1 = 7$ with subsequent synchronizer using trellis starting and terminating portions.

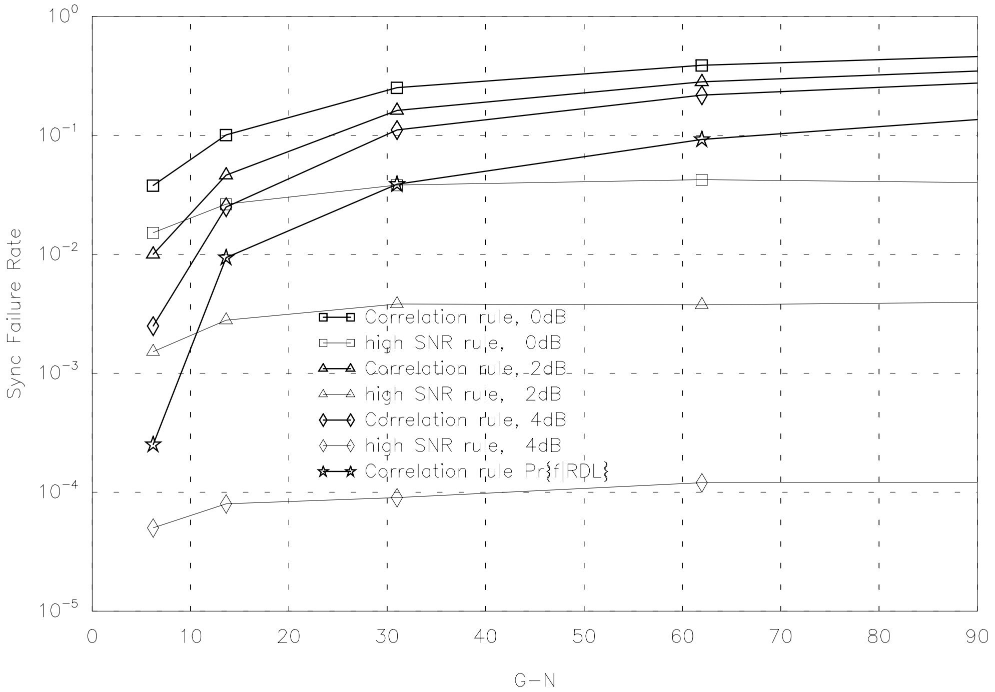


Figure 9.5: *Influence of the time-slot over-length on the frame synchronization performance (simulation).* $L = 7$, BPSK. *We compare the high* SNR *and soft correlation rules for various* SNR. *In addition we illustrate* $Pr\{f|\mathrm{RDL}\}$ *for the soft correlation rule.*

- High SNR rule as list synchronizer with $\nu_1 = 7$ with subsequent synchronizer using trellis starting and terminating portions as list synchronizer with $\nu_2 = 3$ with subsequent convolutional decoder and (assumed perfect) error detection decoder (concatenated list synchronizers).

The length of the packet data sequence [2] is $N - L = 120$ (60 info bits) and is kept constant. The slot over-length $G - N$ is 14 in Fig. 9.6 and 31 in Fig. 9.7.

Note how the savings in terms of sync word length is more pronounced when the slot over-length is greater. The savings are quite impressive if one compares the correlation rules with the concatenated list synchronizer scheme. But even the simple to implement high SNR rule without trellis information yields a considerable bandwidth and power saving.

9.3.2 Results for DBPSK

Our last simulation results consider D2PSK instead of BPSK modulation. Again we resort to our depiction of the sync failure rate as a function of the sync word length. Fig. 9.8 indicates that the savings are even more pronounced for DBPSK than they were for BPSK. This is because the loss due to differentially coherent demodulation is larger than 3 dB for the

[2] Relevant for the laset two synchronizers.

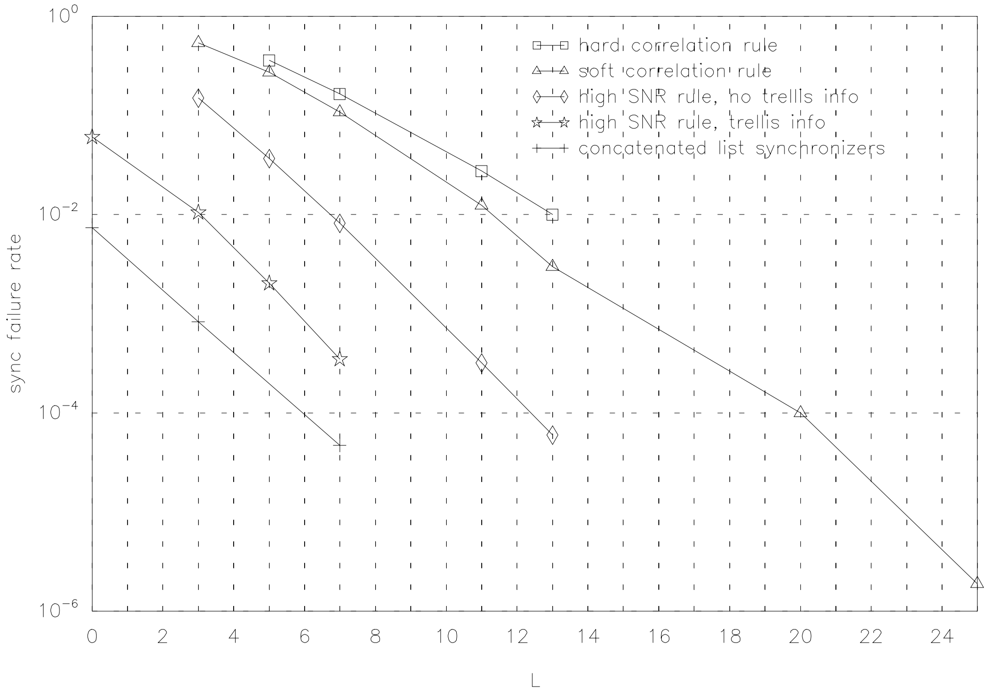


Figure 9.6: *Influence of the sync word length on the frame synchronization performance (simulation). BPSK with phase ambiguity. The slot over-length $G - N$ is 14. The signal-to-noise ratio is $E_s/N_0 = 2$ dB. For a description of the synchronizers, refer to text.*

correlation rules. Note how the concatenated list synchronizer is limited by the appearance of packets that have bit errors after decoding, even when correctly frame synchronized.

9.4 Influence of Sync Word Choice

The reader may understandably harbor doubts as to whether the union bounds proposed for the correlation rule and high SNR rule are indeed correct not merely by coincidence, but accurately reflect the true situation in the synchronizer. To provide due reassurance, the following was performed: the approximate bound (for the high SNR rule for BPSK without phase ambiguity) was used to determine the distribution of frame sync errors, $Pr\{\mu = \hat{\mu} | L_H(\hat{\mu}) > L_H(\tilde{\mu})\}$. Put in plain language, this is equivalent to asking '*if* a sync error occurs, then how far off $(\mu - \tilde{\mu})$ is the chosen position from the correct one?'. The results -both analytically and by simulation- are shown in Fig. 9.9. Following points should be noted: the simulation confirms the validity of our (approximate) bound. Furthermore, the partial autocorrelation function of the sync word seems to have a great influence on the sync error rate, for the sync word chosen, $R_\mu = (7, 0, -1, 0, -1, 0, -1)$. These values can easily be mapped onto the peaks in the diagram, both to the left and right of $\tilde{\mu}$. Interesting is also the susceptibility of the frame synchronizer to 'competing' μ immediately to the left of $\tilde{\mu}$ (case II). Comparing the results with Fig. 8.1 again exemplifies the 'local' protection by the sync word, and global

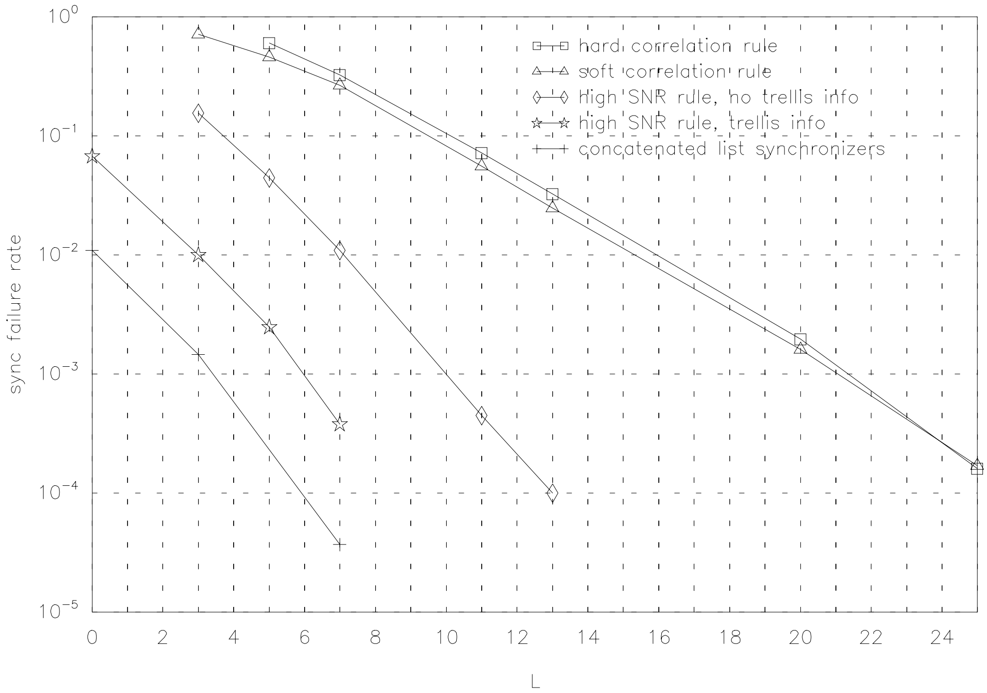


Figure 9.7: *Influence of the sync word length on the frame synchronization performance (simulation). BPSK with phase ambiguity. The slot over-length $G - N$ is 31. The signal-to-noise ratio is $E_S/N_0 = 2$ dB. For a description of the synchronizers, refer to text.*

protection from the energy correction term.

9.5 Sync Word Design Using the Union Bound as an Optimization Criterion

The results of the previous section seem to indicate a high importance of good auto-correlation properties of the sync word, especially for the high SNR rule. But because of the global protection of the energy correction term, the *position* of side-lobes in the partial auto-correlation R_μ may be of significance, side-lobes for μ closer to 0 may be more dangerous than those further away.

In section 4.4 we have indicated that sync words for the traditional frame sync problem are chosen such that they minimize a function of the sync failure rate. We adopt a similar strategy here, only for the packet sync problem, and simply apply a brute-force search of possible sync words, optimizing according to our approximate bounds (9.9) and (F.1). This search should yield the optimal binary sync words for BPSK (also DPSK) and DPSK with phase ambiguity, respectively. For several L the results are identical to those of Tables E.1 and E.2. A good example for BPSK (or DPSK) is $L = 8$, Table E.1 indicates that $\vec{S} = (1, -1, 1, 1, 1, -1, -1, -1)$ with $R_\mu = (8, 1, 0, -3, 0, -1, 0, -1)$ is a good choice, whereas

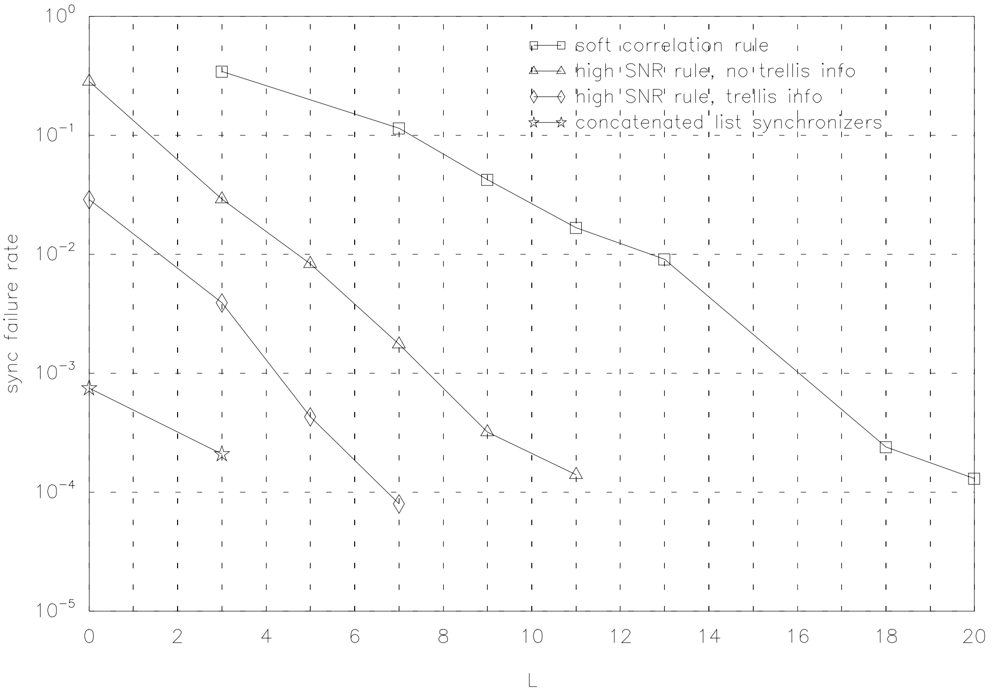


Figure 9.8: *Influence of the sync word length on the frame synchronization performance (simulation), DBPSK. The slot over-length $G - N$ is 14. The signal-to-noise ratio is $E_s/N_0 = 5$ dB. For a description of the synchronizers, refer to text in section 9.3.1.3.*

our search indicates that $\vec{S} = (-1, 1, -1, 1, 1, 1, -1, -1)$ with $R_\mu = (8, -1, 0, -3, 0, 1, 0, -1)$ is better. The only difference is that the values of R_μ are swapped for $\mu = 1$ and $\mu = 5$; the better sync word has the more favourable partial auto-correlation closer to zero.

A more pronounced example is the following: the best sync word of length $L = 22$ for BPSK with π radians phase ambiguity has the partial auto-correlation,

$$R_\mu = (22, 1, 0, -1, 0, 1, 0, -1, 2, 1, -2, -1, 0, 1, 0, -1, -2, 1, 2, 3, 2, 1, 0), \tag{9.14}$$

compared to the following partial auto-correlation for the sync word of length 22 previously found to be good for phase ambiguity, taken from Table E.2:

$$R_\mu = (22, 3, 2, 3, 2, -1, 0, -1, 2, 1, -2, 3, 0, 1, 0, -1, -2, -1, 0, -1, 0, -1, 0). \tag{9.15}$$

Our search has succeeded in producing values of R_μ less than or equal to two in magnitude for μ up to 7, whereas the other sync word has large R_μ for small μ.

The results of this brute-force search, optimizing the sync failure rate for $G - N = 50$ and $E_s/N_0 = 2$ dB are summarized in Tables E.3 and E.4 in appendix E. To illustrate that the search was worthwhile, we have applied the approximate union bound to both our new sync words and the sync words from Table E.1 for lengths $L = 7$ to 22. The results can be seen in Fig. 9.10. Particularly for larger L the benefit from using the new sync words is quite pronounced, and may save one or two sync word symbols.

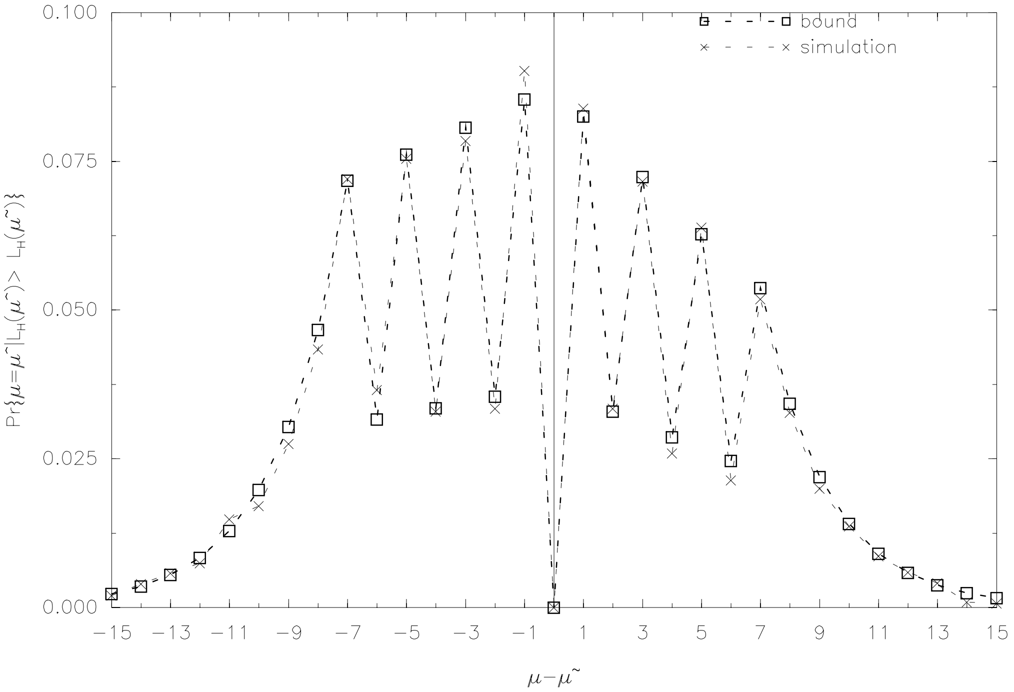


Figure 9.9: Sync error distribution for the high SNR rule. Shown is the conditional probability $Pr\{\mu = \hat{\mu} | L_H(\hat{\mu}) > L_H(\tilde{\mu})\}(\mu - \tilde{\mu})$ derived analytically, and measured (approx.) by simulations. Parameters: slot over-length $G - N = 31$, signal-to-noise ratio is $E_s/N_0 = 2$ dB. Sync word length $L = 7$. Note how well simulation and calculation match. Visible is the influence of the partial auto-correlation function of the sync word on the sync rate.

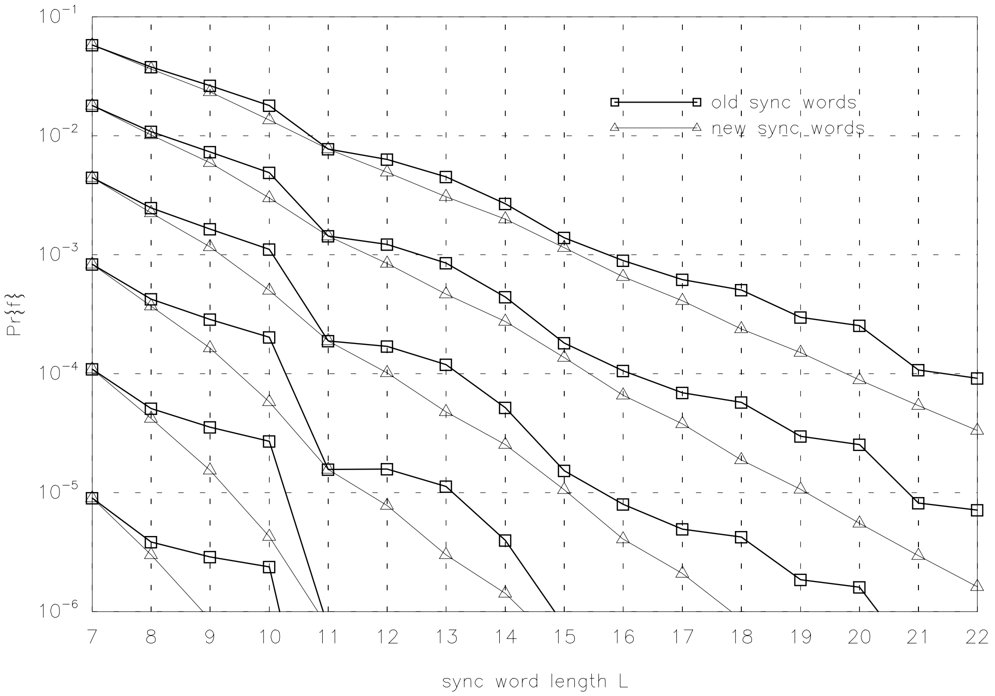


Figure 9.10: *Comparison of sync words. Sync failure rate determined with the approximate union bound for both our new sync words and the sync words from Table E.1 for lengths $L = 7$ to 22. Slot over-length $G - N = 50$. Parameter: signal-to-noise ratio in dB, from 0 dB to 5 dB in 1 dB steps. For larger L one may save one or two sync word symbols, the savings becoming greater at higher SNR.*

Chapter 10

Design Aids and System Examples

10.1 Design Aids

For the high SNR rule synchronizer for preamble-less binary modulated packets transmitted over an AWGN channel, we shall now try and construct a simple rule of thumb for the choice of the length of the sync word, based on our approximate union bounds of the previous section. Input variables in this equation should be the signal-to-noise ratio $\rho_d = 10\log(E_s/N_0)$ in dB and the tolerated frame sync error rate $Pr\{f\}$. Since the performance of the synchronizer is not very strongly dependent on the time-slot over-length (see 9.3.1.2), we have determined such a formula for $G-N=15$ and $G-N=50$ for coherent demodulation with and without phase ambiguity. The procedure is as follows: Motivated by the apparent linear relationship between the logarithm of the sync failure rate, $\log(Pr\{f\})$ and sync word length, L, (see Figs. 9.6, 9.6 and 9.8), we shall apply a linear least mean square fitting (linear regression) to the curves. Since we shall base our evaluation on the approximate union bounds, we can easily construct such curves for various L and different SNR. The linear regression technique is applied to all such curves of $\log(Pr\{f\})$ as a function of L, for 6 different E_s/N_0. One thus obtains several sets of intercept and slope values (each as a function of $\rho_d = 10\log(E_s/N_0)$). These sets again have to be duly approximated; specifically, we have chosen a quadratic approximation for the slopes, and a linear approximation for the intercepts. Let us express this process mathematically, our initial conjecture can be written as

$$\log\left(Pr\{f\}\right) \approx p_i + p_s L, \tag{10.1}$$

where $p_i(\rho_d)$ and $p_s(\rho_d)$ are the intercept and slope that are approximated by linear and quadratic approximation respectively:

$$p_i \approx A_3 + A_4 \cdot \rho_d. \tag{10.2}$$

$$p_s \approx A_0 + A_1 \cdot \rho_d + A_2 \cdot {\rho_d}^2, \tag{10.3}$$

Inserting these into (10.1), yields,

$$\log\left(Pr\{f\}\right) \approx A_3 + A_4 \cdot \rho_d + \left(A_0 + A_1 \cdot \rho_d + A_2 \cdot {\rho_d}^2\right) \cdot L, \tag{10.4}$$

and the desired inverse function

$$L \approx \frac{\log\left(Pr\{f\}\right) - A_3 - A_4 \cdot \rho_d}{A_0 + A_1 \cdot \rho_d + A_2 \cdot {\rho_d}^2} \tag{10.5}$$

The resulting approximation is very accurate for BPSK with and without phase ambiguity, values of the parameters A_0 to A_4 are given in appendix G. To demonstrate the accuracy for a value of $G - N$ different from 15 and 50, we chose $G - N = 31$ and linearly interpolated the above parameters A_0 to A_4. The comparison between true approximate bound, linear approximation (10.5) and simulation is demonstrated in Fig. 10.1. We can conclude that the linear approximation is quite accurate.

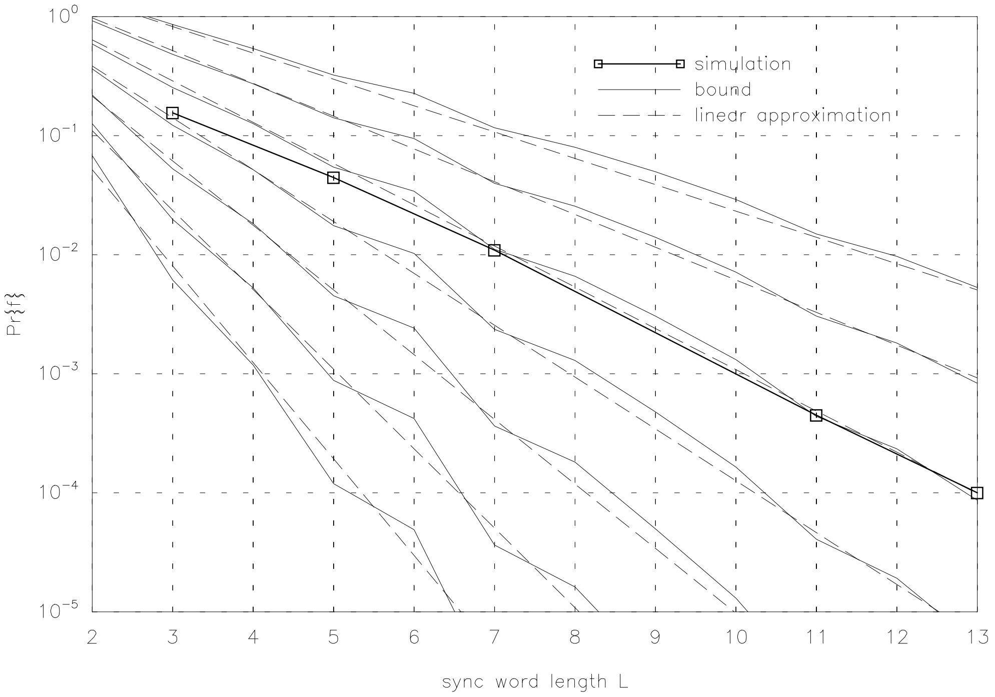


Figure 10.1: *Demonstration of linear approximation for the sync error rate of packets. BPSK with phase ambiguity. The slot over-length $G - N$ is 31. The parameter is the signal-to-noise ratio in dB, from 0 dB to 6 dB in 1 dB steps.*

In order to provide such an approximation for DPSK, we shall simply make use of our observation that binary DPSK is less than 3 dB worse than the binary synchronizer without phase ambiguity. This can be taken into account when inserting ρ_d above -one simply subtracts 3 dB.

10.2 System Examples

10.2.1 60 GHz Vehicle/Vehicle and Vehicle/Roadside Communication

Although originally not implemented as a preamble-less system, the DACAR microwave experimental link of [WRL+91] is a good example of high speed packet processing following sampling of the packet. We shall assume the following packet structure and transmission parameters, oriented towards [WRL+91] and [HW90]:

Modulation: Binary DPSK; data rate 500 KBit/s [WRL+91] to 2 MBit/s [HW90], we have chosen 1 Mbit/s.

Coding: Shortened $(n, k) = (35, 25)$ Reed Solomon code over GF 2^6 (rate 5/7). Convolutional coding was also suggested as a possibility [WRL+91].

Packet length N: 290 [WRL+91] to 500 [HW90]. In our simulations we have chosen $N = 290$.

Slot over-length $(G - N)$: 40 symbols (for 1 Mbit/s) [HW90].

Access: Slotted ALOHA [WRL+91] or reservation TDMA [HW90].

Channel: Non-frequency selective Rayleigh fading, average $E_s/N_0 = 20$ dB. Maximum doppler frequency 11 KHz [WRL+91].

Maximum tolerated synchronization error probability : 0.001, [WRL+91].

We presume that packets can be stored and processed by a DSP followed by a decoder chip, since DPSK is used, no preamble is necessary, since timing recovery, coarse and fine frequency control can be performed without known a-priori symbols using algorithms outlined in chapter 7. The fading is very fast, hence it is advantageous to spread the sync word through the whole packet, as described in section 3.7.

We have performed Monte-Carlo simulations of the following frame synchronizers for DBPSK:

1. soft correlation rule, sync word at beginning of packet,
2. hard correlation rule, sync word at beginning of packet,
3. hard correlation rule, sync word symbols uniformly spread through the whole packet,
4. high SNR rule, sync word at beginning of packet,

5. high SNR rule, sync word symbols uniformly spread through the whole packet.

The results for an average SNR of 20 dB are given in Fig. 10.2. The hard correlation rule is significantly better than the soft correlation rule, again because the soft correlation rule suffers from the missing correction term more than the hard correlation rule, see section 4.5.7. There is a marked improvement from spreading the sync word throughout the packet, especially for higher sync word lengths. For sync word length $L = 13$ the sync failure rate is three orders of magnitudes better when one compares the high SNR synchronizer and spread sync word, with the soft correlation rule without spreading the sync word. The gain in terms of throughput increase when comparing the 'best' (high SNR rule, sync word spread), with the worst synchronizer (soft correlation rule, sync word at start) is at least 23 more (coded) bits available for information transmission per packet. The hard correlation rule with the sync word spread needs a sync word about 8 bit longer than the high SNR rule (also with the sync word spread.)

However, we must be wary of over-rating the quality of the hard correlation rule. If the average SNR decreases, then the soft correlation rule will become better (compared to the hard correlation rule), as the number of bit errors (hard decision) will increase dramatically. Because *graceful degradation* is an important issue in digital communications, we have also investigated an average SNR of 15 dB, see Fig. 10.3. The hard correlation rule is now not very much better than the soft correlation rule; clearly the hard correlation rule offers less graceful degradation than the soft correlation or high SNR rules.

10.2.2 The INMARSAT-C Signalling Channel -Low Rate Ship to Satellite Communication

One of the earliest examples and also a *complete* standard of a block-oriented packet processing system is the INMARSAT Standard-C signalling channel [Int89] [CSV90] [OI92]. The channel is used for mobile terminals to make channel reservations (uplink) and operates in the L-band. The INMARSAT-C standard signalling channel specifies no preamble in the packet structure, only a 64 symbol sync word for frame synchronization. The details are summarized in the following:

Modulation: Binary PSK. Data rate: 1200 symbols/second (2nd Generation).

Coding: Memory $m = 6$ Convolutional code rate 1/2, generators 133 and 171, followed by 16 bit CRC check.

Packet length N: 316 symbols.

Proposed sync word length L: 64 symbols, in Hex: 07EACDDA4E2F28C2.

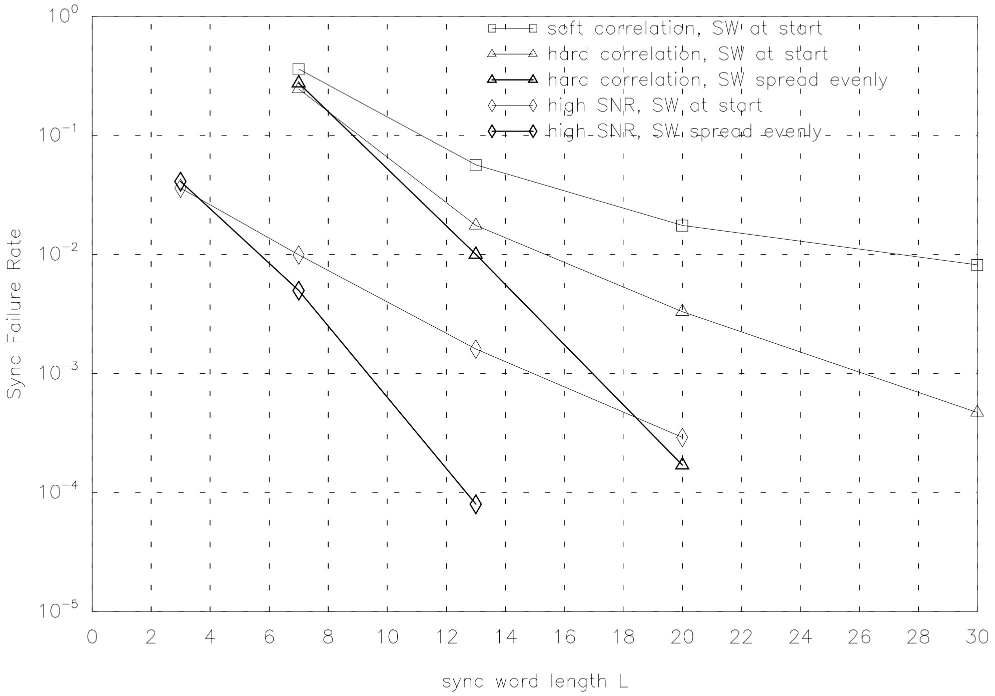


Figure 10.2: *Simulation results of frame synchronization for 60 GHz mobile packet communication. Average SNR 20 dB, DBPSK modulation. For a description of the synchronizers and the channel, see text. Notice the marked improvement achieved by spreading the sync word throughout the packet.*

Slot over-length ($G - N$): 54 symbols (2nd generation), 108 symbols(!) (1st generation).

Access: Slotted ALOHA.

Channel: Slow, non-frequency selective Ricean fading (3 dB fading bandwidth: 0.7 Hz); carrier-to-multi-path ratio: 7 dB; operating SNR: 4.7 dB E_s/N_0. Maximum short term frequency uncertainty of ±50 Hz.

Maximum tolerated synchronization error probability : 0.022.

The channel with deep slow fades affecting a complete packet, poses several problems, most severe of which seems to be reliable acquisition of the carrier frequency. However, the low data rate allows quite elaborate processing to be performed with a digital signal processing (DSP) based approach [CSV90]. After the packet has been sampled with 8 samples per symbols, an FFT based search method is used for frequency acquisition in two stages, one coarse search to within 75 Hz, then a fine search. Following frequency acquisition there comes timing recovery, accurate to within 1/8 of a symbol period [CSV90]. The sync word is searched for differentially coherently using simple correlation (hard or soft, details were not evident from [Int89], [CSV90] or [OI92]). Finally, the doppler shift is removed and coherent detection is possible; the remaining π radians phase ambiguity is resolved using the known sync word, see 3.4.1. The INMARSAT-C signalling channel has often proudly been called preambleless, although the 64 symbol overhead of the sync word (20 %) remains. We shall now see

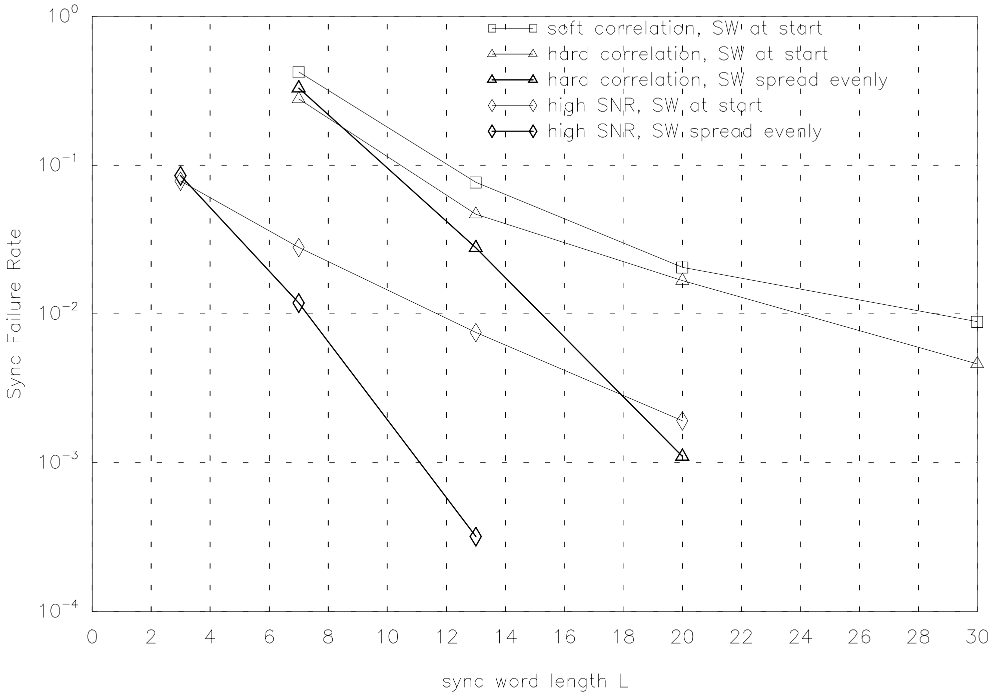


Figure 10.3: *Simulation results of frame synchronization for 60 GHz mobile packet communication. Average SNR 15 dB. For a description of the synchronizers and the channel, see text. The hard correlation rule is now not very much better than the soft correlation rule.*

how this length can be reduced by the use of our (close-to) optimal algorithms for frame synchronization of preamble-less packets with differential demodulation at the frame sync stage. We have also presumed coherent frame synchronization as an alternative technique, in addition optionally making use of trellis termination information.

When determining the frame sync rate (for 2nd generation only), we have chosen to ignore those frames in the statistic *that would lead to decoding errors, had the frame synchronization been a-priori correct.* This was done because the severe (and slow) fading results in some packets having a very bad SNR, these are neither decode-able nor synchronizable. Since it makes no sense to design a packet communications system where the synchronization is much more robust than the decoding [1], we are only interested in those packets that can be decoded but not (frame) synchronized. We shall still refer to this rate as the sync failure rate. The results for the synchronizers:

1. soft correlation rule, differential demodulation,

2. hard correlation rule, differential demodulation,

3. high SNR rule, differential demodulation,

[1] This is in contrast to continuous communication where robust synchronization is important to reduce re-acquisition time after deep fades.

4. soft correlation rule, coherent demodulation (with π radians phase ambiguity),

5. high SNR rule, coherent demodulation (with π radians phase ambiguity),

6. high SNR rule, with trellis opening and termination information, preceding list synchronizer with $\nu = 13$ to reduce computational burden, coherent demodulation (with π radians phase ambiguity),

are given in Fig. 10.4. Note that the results for the latter three synchronizers (coherent demodulation, with π radians phase ambiguity) include correct detection of the phase error due to ambiguity.

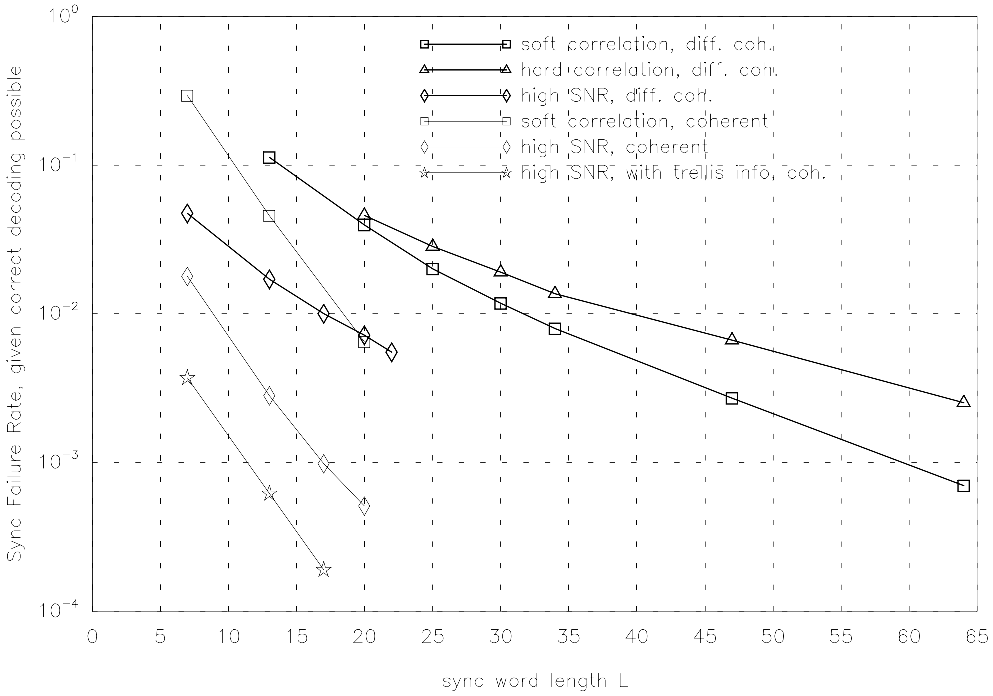


Figure 10.4: *Simulation results of frame synchronization for the INMARSAT-C signalling channel but with varying sync word length L. For a description of the parameters, see text.*

The possible reduction of the sync word length is dramatic, the proposed sync word length of 64 symbols hardly appears necessary. To achieve a sync failure rate of less than 10^{-3} the differential coherent high SNR rule needs about 22 symbols. *If* one is able to perform phase recovery before frame synchronization -i.e. the latter can be done coherently but with π radians phase ambiguity- then one only needs about 10 symbols, and only about 5 if one makes use of the trellis opening/termination information. It seems to be the case that the differentially coherent frame synchronizers suffer more severely in the deep fades than their coherent counterparts.

10.3 Remarks

To conclude this chapter let us comment on some of the main observations made. We have shown that the analysis derived in previous chapters allows us to give rough design aids for the necessary length of the sync word, assuming an AWGN channel. However, simulations were necessary to show that the DPSK high SNR rule works well in both a fast and slow time-variant non-frequency selective fading channel. In the case of the former, it is very worthwhile to spread the sync word within the packet -similar to interleaving of normal data. The correlation rules (soft and hard) are always significantly worse than the high SNR rule, but what speaks strongly against using the correlation rule is the fact that depending on the fading speed, either hard or soft correlation can be superior to the other. Using the high SNR rule makes the choice of algorithm simpler! Finally, the second example showed that differentially coherent detection is significantly worse than coherent detection with remaining phase ambiguity on a slow fading channel with deep fades, as far as the necessary sync word length is concerned.

Chapter 11

Conclusions

The work has focussed on optimal frame synchronization for continuous data transmission (Part I) and preamble-less packet communication (Part II). We have derived optimal and close-to-optimal likelihood functions that are to be evaluated by the frame synchronizers.

With both simulation and analytical methods the performance of the new synchronizers can be compared to the state-of-the-art techniques presented at the beginning of the work. In all cases, the approach 'derive optimal then close-to-optimal synchronizers' has been shown to be successful compared to using a simple synchronizer (correlation rule).

The work can be outlined graphically, see Fig. 11.1, which shows the relationship between derivation of synchronizers, simulation, analysis, technical aids and applications.

11.1 Major Achievements

The major achievements can best be summarized as follows:

1. Determination of Likelihood Functions:
   The philosophy 'use all available information at the receiver when doing frame synchronization' has been successfully applied to a wide range of scenarios. In all cases, the correlation term is augmented by a correction term that has its origin in the data following the sync word. New contributions have been made in the following areas:

   - **For different modulation schemes:**
     - Differentially demodulated BPSK.
     - Demodulation with phase ambiguity and higher order modulation.
   - **For different channels:**
     - AWGN channel.

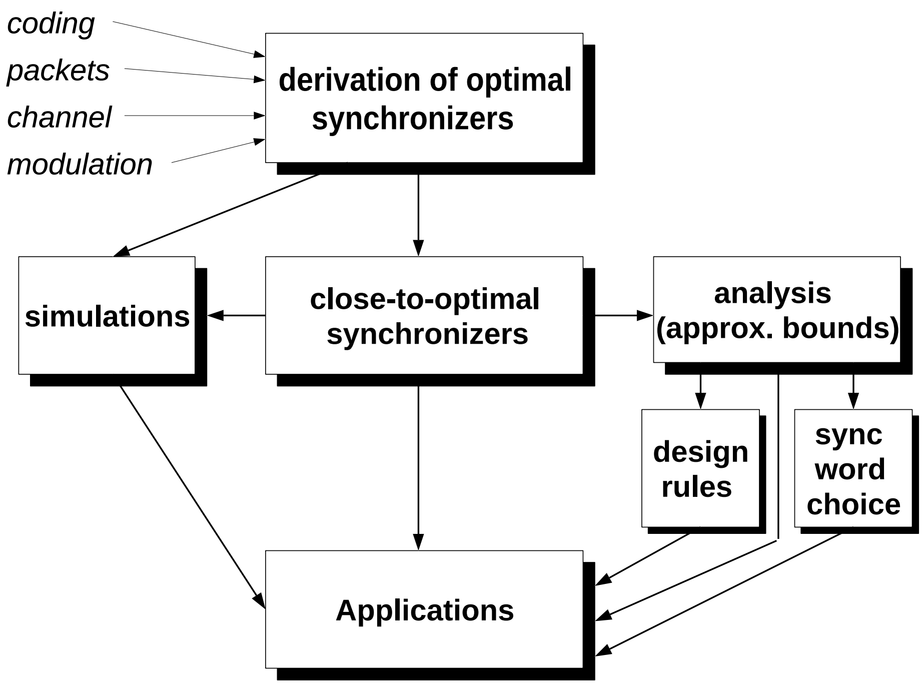

coding
packets
channel
modulation
derivation of optimal synchronizers
simulations
close-to-optimal synchronizers
analysis (approx. bounds)
design rules
sync word choice
Applications

Figure 11.1: *Illustration of the main contributions of the work. Beginning with derivation of optimal and close-to-optimal synchronizers depending on the frame sync scenario, we have incorporated simulation and analysis. The latter has become the basis of sync word design rules for packets. The suitability of the proposed methods has been illustrated with examples.*

  - Non-frequency selective time varying fading channels, (with known CSI).

- **Coding and frame synchronization:**
  - The list synchronizer that assumes that an outer instance (e.g. decoder) can select the correct frame/packet starting position from a list of possible 'candidates'.
  - The synchronizer that uses trellis termination information to improve performance (effectively lengthens the sync word).
  - 'Coded synchronization'. By using the SOVA to provide soft output information to a frame synchronizer that searches for an *encoded* sync word, we can improve the sync rate. It is also possible to use knowledge of the sync word bits to improve decoding when in the 'in sync' mode of operation.

- **Packets without preambles:**
  - Phase coherent demodulation of QAM with and without phase ambiguity. The new synchronizer uses both the correlation with the sync word *and* the energy of the data portion of the packet.
  - Differentially demodulated BPSK.

  - List synchronizer and 'trellis termination' extensions are possible.

2. Analysis:

- **Union bounds**:
  - Where possible true bounds were derived, this was the case for the hard and soft correlation rules for modulation without phase ambiguity.
  - Otherwise close approximations to bounds, were derived: most importantly for the high SNR rule for BPSK with and without phase ambiguity.
- **Random data limited bounds**:
  - For the list synchronizer.
  - For the soft correlation rule for packets.
- **A model for coded frame sync** was developed that accurately describes the achievable performance when coding the sync word.
- **A binary sync word search** was performed, suitable for packet synchronization, by using the approximate union bounds as an optimization criterion.

3. Technical relevance:

- **A significant reduction in the length of the sync word** is possible when using the new synchronizers; if the sync word length is already fixed, a major sync rate improvement.
- **The soft correlation rule is usually inferior to the hard correlation rule**, except for packets transmitted over slowly varying channels. By adhering to the new synchronizer, the question of whether to use soft or hard correlation becomes redundant.
- **The soft correlation rule can perform catastrophically** (e.g. in fading or for DBPSK).
- **The high SNR rule for packets is easily implementable**, a possible structure was given that is based on a recursive definition of the likelihood function.
- **Implementation structures** and a computational burden analysis for the 'trellis termination' synchronizer were presented.
- **Design rules** for the high SNR rule for packets, BPSK and an AWGN channel were presented.
- **New, optimal binary sync words** for packets (with and without phase ambiguity) found by computer search are tabulated.

11.2 Further Work

By no means is the topic of optimal frame synchronization complete. Further work might still be done on the following topics:

1. Analysis:
   - The union bounds may be extended to the high SNR rule for QPSK and possibly higher order signaling, with phase ambiguity, and also for packets.
   - The (approximate) union bound might be derivable for DPSK.
   - Analytical performance evaluation in fading channels (slow and fast fading) remains an open question.
2. Techniques:
   - Optimal rules for higher order differential PSK (with $M > 2$) would be useful.
   - The principles developed here and in previous work may be extended to similar problems occurring in OFDM (Orthogonal Frequency Division Multiplex) systems. In particular, timing and carrier synchronization.
   - Multiple detection DPSK [DS90] [Edb92] is a promising alternative to ordinary DPSK, offering the same advantages (no phase recovery needed) but superior performance in terms of the bit error rate. The technique could be applied to frame synchronization.
3. Packet frame synchronization in particular:
   - Behaviour with collisions, as encountered in slotted ALOHA systems for example.
   - The synchronization of variable length packets (either several possible lengths, or totally variable) remains a topic to be investigated.
   - The initial placement of the window surrounding the packet for completely sporadic reception.
   - Coupled with advances in the field of analytical performance evaluation for other modulation schemes, goes the development of further design rules (both sync word choice and sync word length).

Appendix A

Notation and Symbols

A.1 Abbreviations

The following abbreviations have sometimes been used:

- AGC: Automatic Gain Control,
- ARQ: Automatic Repeat Request,
- AWGN: Additive White Gaussian Noise,
- CRC: Cyclic Redundancy Check,
- CSI: Channel State Information,
- DBPSK: Differential Binary Phase Shift Keying,
- BPSK: Binary Phase Shift Keying,
- DAF: Decoding After Frame synchronization,
- DBF: Decoding Before Frame synchronization,
- DS/PN: Direct Sequence/Pseudo Noise,
- FEC: Forward Error Correction,
- FFT: Fast Fourier Transformation,
- FSK: Frequency Shift Keying,
- i.i.d.: Identically Independently Distributed
- LF: Likelihood Function,

- MAP: Maximum A-Posteriori,
- MF: Matched Filter,
- ML: Maximum Likelihood,
- MSK: Minimum Shift Keying,
- NT: No Transmission (of a packet),
- PSK: Phase Shift Keying,
- PDF: Probability Density Function,
- QAM: Quadrature Amplitude Modulation,
- QPSK: Quadrature Phase Shift Keying,
- RDL: Random Data Limited,
- RV: Random Variable,
- SNR: Signal-to-Noise Ratio,
- SW: Sync Word,
- SOVA: Soft Output Viterbi Algorithm,
- TDMA: Time Division Multiple Access,
- VA: Viterbi Algorithm,

A.2 Mathematical Notation

We have adopted the following notation:

- $\mathrm{Im}\{a\}$: Imaginary component of complex number a,
- $\mathrm{Re}\{a\}$: Real component of complex number a,
- a^*: Complex conjugate of complex number a,
- $\langle a, b\rangle = \mathrm{Re}\{a\}\mathrm{Re}\{b\} + \mathrm{Im}\{a\}\mathrm{Im}\{b\}$: inner product between a and b,
- $Q(x, y)$: Marcum Q-function,
- $\mathrm{erfc}()$: complementary error function,

- $\lfloor x \rfloor$: largest integer less than or equal to the real valued argument x,
- $f_{\mathbf{x}}(x)$: PDF of the random variable $\mathbf{x}$,
- $f_{\mathbf{x}}(x) * f_{\mathbf{y}}(y)$: Convolution of the PDFs of $\mathbf{x}$ and $\mathbf{y}$,
- $\mathcal{N}_a(m, \sigma^2)$: Gaussian distribution of a random variable a with mean m and variance σ^2,
- $\overset{L-1}{\underset{i=0}{*}}$: L-fold convolution,
- $Pr\{\mathcal{S}\}$: Probability of the event $\mathcal{S}$,
- $\sim \exists$: There does not exist,
- $||a||$: Euclidean norm (magnitude) of complex number a,
- $\forall\ \vec{d}$: For all data sequences $\vec{d}$,
- $\hat{x}$: receiver's estimate of true value $\tilde{x}$,
- $\vec{x} = (x_0, ..., x_{N-1})$: Sequence with N elements,
- z^{-1}: delay element,

A.3 Symbols

- A_0 to A_4: parameters for sync word length design formula,
- $C_i = y_{i+\mu} \cdot S_i$: correlation rule component,
- D_j: number of possible data sequences in which exactly j occurrences of the sync word occur,
- E_s: average energy per symbol,
- G: length of slot,
- $H_i = y_{i+\mu} \cdot S_i - |y_{i+\mu}|$: high SNR rule component,
- i_i: information bit i,
- I: number of information bits,
- L: length of sync word,
- $L(\mu)$: optimal likelihood function,
- $L_S(\mu)$: close-to optimal or suboptimal likelihood function,

- $L_H(\mu)$: Likelihood function, high SNR rule,
- $L_C(\mu)$: Likelihood function, correlation rule,
- M: number of different symbols (modulation format),
- M_a: number of possible phase references (due to phase ambiguity),
- N: length of frame or packet,
- N_0: one-sided noise power spectral density,
- P: only needed for comparison of DAF, DBF: length of frame,
- $Pr\{f\}$: frame synchronization failure probability,
- $Pr\{f|\mathrm{RDL}\}$: frame synchronization failure probability for the noiseless case,
- $Pr\{EC\}$: frame decoding error probability (only for list sync),
- $Pr\{FE\}$: total frame error probability (only for list sync),
- $Q = \lfloor (N-L)/L \rfloor$: number of times the sync word can occur in a sequence of length $N-L$,
- R: code rate,
- R_μ: partial auto-correlation function of sync word for shift μ,
- $\vec{S} = (S_0, S_1, ..., S_{L-1})$: sync word,
- T: threshold,
- T_s: symbol duration,
- $\{W_j, 1 \leq j \leq M\}$: set of transmitted symbols,
- a_i: i-th fading value,
- c_d: convolutional code parameter,
- d_{min}: minimal Hamming distance of convolutional code,
- $\vec{d} = (d_L, ..., d_{N-1})$: observation data sequence, traditional and packet sync case,
- $\vec{d^F} = (d^F_L, ..., d^F_{N-1})$: frame data sequence, traditional sync case,
- $g(i)$: integer valued function of integer valued argument i,
- $h(i)$: integer valued function of integer valued argument i,

- k: number of information bits per branch, in the trellis of convolutional code,
- m: convolutional encoder memory,
- m_d: mean of absolute value of a received symbol, given that either a sync word or data symbol was transmitted,
- $m_{H\pm}$: mean of H_i for the two cases $x_{i+\mu} = \pm S_i$,
- m_n: mean of absolute value of a noise sample, $|n_i|$,
- $m_{\Delta L}$: mean of ΔL,
- n: number of convolutional encoder output bits per trellis transition (rate $1/n$ code),
- n_A: number of adders in trellis aided synchronizer,
- n_i: noise sample i,
- n_M: number of maximizers in trellis aided synchronizer,
- $n_{\Delta\mu}$: number of each possible occurring $\Delta\mu$,
- n_r: maximum number of disagreements with the sync word, hard correlation rule,
- p: number of samples per symbol for timing estimator,
- $\bar{n}_s$: average number of subsequent decodings per frame,
- p_e: error probability of BSC,
- $\vec{t}(g)$: g-th possible terminating sequence,
- $\vec{x} = (x_0, x_1, ..., x_{N-1})$: transmitted symbols corresponding to $\vec{y}$, traditional frame sync problem,
- $\vec{x^D} = (x_0^D, x_1^D, ..., x_{N-1}^D)$: transmitted symbols corresponding to $\vec{y^D}$, traditional frame sync problem, differentially encoded,
- $\vec{x} = (x_0, x_1, ..., x_{G-1})$: transmitted symbols corresponding to $\vec{y}$, packet transmission schemes,
- $\vec{x^D} = (x_0^D, x_1^D, ..., x_{G-1}^D)$: transmitted symbols corresponding to $\vec{y^D}$, packet transmission schemes, differentially encoded,
- $\vec{y} = (y_0, y_1, ..., y_{N-1})$: base-band symbols after demodulation, traditional frame sync problem,

- $\vec{y^D} = (y_0^D, y_1^D, ..., y_{N-1}^D)$: base-band symbols before differential demodulation, traditional frame sync problem,
- $\vec{y} = (y_0, y_1, ..., y_{G-1})$: base-band symbols after demodulation, packet transmission schemes,
- $\vec{y^D} = (y_0^D, y_1^D, ..., y_{G-1}^D)$: base-band symbols before differential demodulation, packet transmission schemes,
- β: roll-off parameter,
- $\Delta L = L(\tilde{\mu}) - L(\mu')$: difference between competing values of likelihood function,
- $\Delta\mu = |\tilde{\mu} - \mu'|$,
- κ_μ: partial correlation of sync word with random data, note that κ_μ is only defined for $0 \leq \mu \leq L$,
- κ_L: correlation of sync word with random data sequence of length L,
- σ_d^2: variance of absolute value of a received symbol, given that either a sync word or data symbol was transmitted,
- $\sigma_{H\pm}^2$: variance of H_i for the two cases $x_{i+\mu} = \pm S_i$,
- σ_n^2: variance of absolute value of a noise sample, $|n_i|$,
- $\sigma_{\Delta L}^2$: variance of ΔL,
- $\tilde{\mu}$: starting position of frame or packet,
- μ': competing frame/packet starting position,
- $\hat{\mu}$: receiver's estimate of $\tilde{\mu}$,
- ν: number of frame starting positions generated by list synchronizer, additional index (e.g. ν_1, ν_2, ...) if cascaded,
- ϕ_a: error due to phase ambiguity,
- ϕ_r: phase rotation the code is invariant to,
- ρ_d: $10\log(E_s/N_0)$, signal-to-noise ratio, in dB,

Appendix B

Necessary Proofs

B.1 Proof for Derivation of Likelihood Function

We have to prove that

$$\sum_{\forall \vec{d}} \prod_{i=0}^{N-1} e^{\frac{2}{N_0}\left[\langle y_{i+\mu}, d_i\rangle - \frac{\|d_i\|^2}{2}\right]} = \prod_{i=0}^{N-1} \sum_{j=1}^{M} e^{\frac{2}{N_0}\left[\langle y_{i+\mu}, W_j\rangle - \frac{\|W_j\|^2}{2}\right]}. \tag{B.1}$$

We shall prove the general form,

$$\sum_{\forall \vec{d}} \prod_{i=0}^{N-1} f(d_i, i) = \prod_{i=0}^{N-1} \sum_{j=1}^{M} f(W_j, i), \tag{B.2}$$

where $f(a, i)$ is an arbitrary function of a and i. The proof is trivial, and we shall complete it by demonstration. The left side above can be written as,

$$\begin{aligned}
f(d_0 = W_1, 0) \cdot f(d_1 = W_1, 1) \cdot ... \cdot f(d_{N-1} = W_1, N-1) &+ \\
f(d_0 = W_1, 0) \cdot f(d_1 = W_1, 1) \cdot ... \cdot f(d_{N-1} = W_2, N-1) &+ \\
f(d_0 = W_1, 0) \cdot f(d_1 = W_1, 1) \cdot ... \cdot f(d_{N-1} = W_3, N-1) &+ \\
... &+ \\
f(d_0 = W_1, 0) \cdot f(d_1 = W_1, 1) \cdot ... \cdot f(d_{N-1} = W_M, N-1) &+ \\
f(d_0 = W_1, 0) \cdot ... \cdot f(d_{N-2} = W_2, N-2) \cdot f(d_{N-1} = W_1, N-1) &+ \\
f(d_0 = W_1, 0) \cdot ... \cdot f(d_{N-2} = W_2, N-2) \cdot f(d_{N-1} = W_2, N-1) &+ \\
... &+ \\
f(d_0 = W_M, 0) \cdot ... \cdot f(d_{N-2} = W_M, N-2) \cdot f(d_{N-1} = W_M, N-1) &,
\end{aligned} \tag{B.3}$$

and the right side as,

$$\begin{aligned}
&\{f(W_1,0)+f(W_2,0)+...+f(W_M,0)\}\\
&\cdot\{f(W_1,1)+f(W_2,1)+...+f(W_M,1)\}\\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\cdot\ldots\cdot\\
\cdot\{f(W_1,N-1)+f(W_2,N-1)+...+f(W_M,N-1)\}\,. \qquad (\mathrm{B}.4)
\end{aligned}$$

It is obvious that expansion of (B.4) yields (B.3).

B.2 Proof of Random Data Limited Bound

B.2.1 Random Data Limited Bound for the ML, High SNR and Correlation rules

Under the premise that no equally long prefix and suffix of the sync word shall be equal, i.e.

$$\forall s\ ,\ 1 \le s \le L-1\ :\ (S_0, S_1, \ldots, s_{L-1}) \ne (S_s, S_{s+1}, \ldots, S_{s+L-1}) \qquad (\mathrm{B}.5)$$

we shall prove equation (4.2):

$$Pr\{f|\mathrm{RDL}\} = \sum_{i=1}^{\lfloor N/L-1\rfloor} \frac{(-1)^i}{i+1}\binom{N-L-(L-1)i}{i} M^{-Li}. \qquad (\mathrm{B}.6)$$

The proof is found in shortened form in [Nie73]. Consider an N long frame and sync word length L. There are exactly $Q = \lfloor N/L - 1 \rfloor$ maximal possible occurrences of the sync word in the observation data sequence. Let D_j be the number of possible data sequences in which the sync word occurs exactly j times; of course, $0 \le j \le Q$. The probability of each such data sequence is thus $\frac{D_j}{M^{N-L}}$. A sync error occurs (in the noiseless case) if and only if the synchronizer chooses the 'wrong' sync word. If the sync word occurs $j+1$ times in the whole observation sequence (once as the true sync word, j times in the random data), then the probability of synchronization failure is $\frac{j}{j+1}$. Thus,

$$Pr\{f|\mathrm{RDL}\} = \sum_{j=1}^{Q} \frac{j}{j+1}\frac{D_j}{M^{N-L}}. \qquad (\mathrm{B}.7)$$

How large is D_j? We can derive a recursive formula for D_j, beginning with D_Q:

$$D_Q = \binom{N-L(L-1)Q}{Q} \cdot M^{N-L-LQ}, \qquad (\mathrm{B}.8)$$

because if the sync word occurs exactly Q times in the random data, then there are $N-L-LQ$ 'free' symbols left over, the number of such data sequences is, therefore, M^{N-L-LQ} (the other LQ symbols are 'fixed' -they equal the sync word!). These Q occurrences can begin in exactly $Q+N-L-LQ$ positions. A similar argument yields,

$$D_{Q-1} = \binom{N-L(L-1)(Q-1)}{Q} \cdot M^{N-L-L(Q-1)} - \binom{Q}{Q-1} \cdot D_Q, \tag{B.9}$$

simply by replacing Q by $Q-1$ and compensating for those $\binom{Q}{Q-1} \cdot Q$ data combinations with Q occurrences of the sync word that were also counted.

Continuing with such an argument we can give the general recursive formula,

$$D_i = \binom{N-L(L-1)i}{i} \cdot M^{N-L-Li} - \sum_{j=i+1}^{Q} \binom{j}{i} \cdot D_j. \tag{B.10}$$

Multiplying both sides of (B.10) with $\frac{(-1)^{i+1}}{i+1}$ and summing over i yields the right hand side:

$$\sum_{i=1}^{Q} \frac{(-1)^{i+1}}{i+1} \binom{N-L(L-1)i}{i} \cdot M^{N-L-Li} = \sum_{i=1}^{Q} \sum_{j=i}^{Q} \frac{(-1)^{i+1}}{i+1} \binom{j}{i} \cdot D_j, \tag{B.11}$$

since $D_i + \sum_{j=i+1}^{Q} = \sum_{j=i}^{Q} \binom{j}{i} D_j$. We can rearrange the order of summation and apply the fact that

$$\frac{1}{i+1}\binom{j}{i} = \frac{1}{j+1}\binom{j+1}{i+1}, \tag{B.12}$$

yielding,

$$\sum_{j=1}^{Q} \sum_{i=1}^{j} \frac{(-1)^{i+1}}{j+1} \binom{j+1}{i+1} \cdot D_j = \sum_{j=1}^{Q} \frac{1}{j+1} \sum_{i=2}^{j+1} (-1)^i \binom{j+1}{i} \cdot D_j, \tag{B.13}$$

for the right hand side of (B.11). Using the fact that

$$\sum_{i=0}^{j+1} (-1)^i \binom{j+1}{i} = 0, \tag{B.14}$$

we can write,

$$\sum_{i=2}^{j+1} (-1)^i \binom{j+1}{i} = j. \tag{B.15}$$

Inserting this into (B.13), and further into (B.11),

$$\sum_{i=1}^{Q} \frac{(-1)^{i+1}}{i+1} \binom{N-L(L-1)i}{i} \cdot M^{N-L-Li} = \sum_{i=1}^{Q} \frac{j}{j+1} \cdot D_j, \tag{B.16}$$

the right side of which, if multiplied by $\frac{1}{M^{N-L}}$ is exactly $Pr\{f|\mathrm{RDL}\}$; in other words,

$$\begin{aligned} Pr\{f|\mathrm{RDL}\} &= \sum_{j=1}^{Q} \frac{j}{j+1} \frac{D_j}{M^{N-L}} \\ &= \sum_{i=1}^{Q} \frac{(-1)^{i+1}}{i+1} \binom{N-L(L-1)i}{i} \cdot M^{-Li}. \end{aligned} \tag{B.17}$$

QED.

B.2.2 Extension to the List Synchronizer

We now show that for the list synchronizer (with ML, high SNR or correlation rule),

$$Pr\{f|\mathrm{RDL}\}(\nu) = 1 - \sum_{j=0}^{Q} \min(1, \frac{\nu}{j+1}) D_j M^{-(N-L)}, \tag{B.18}$$

where ν denotes the length of the list of 'candidate' positions output by the list synchronizer. $D_j M^{-(N-L)}$ is further given recursively by,

$$\frac{D_j}{M^{N-L}} = \binom{N-L-(L-1)j}{j} M^{-Lj} - \sum_{i=j+1}^{Q} \frac{D_i}{M^{N-L}} \binom{i}{j}. \tag{B.19}$$

Proof: The probability of *correct* synchronization given that the sync word occurs j times again in the random data, is $\min(1, \frac{\nu}{j+1})$, since if $\nu \geq j+1$ then the correct position $\tilde{\mu}$, will *always* be in the list. For the recursive definition of $D_j M^{-(N-L)}$ we have again employed (B.10).

Appendix C

The PDF of the Partial Auto-Correlation of the Sync Word with Random Data

In [LT87] it is shown that the PDF of

$$\kappa_{\mu'} = \sum_{i=0}^{\mu'-1} \langle S_i, x_{i+\mu'} \rangle, \tag{C.1}$$

is discrete, and thus expressible as a series of values of $Pr\{\kappa_{\mu'}\}$, for which the PDF $f_{\kappa_{\mu'}}(\kappa_{\mu'})$ is non-zero, which for MPSK can be calculated using the relation:

$$f_{\kappa_{\mu'}}(\kappa_{\mu'}) = M^{\mu'} \cdot \sum_{\forall\ p_1, p_2, \ldots, p_q\ =\ \mu'} \left\{ \frac{\mu'! \cdot 2\mu' - p_1 - p_q}{p_1! \cdot p_2! \cdot \ldots p_q!} \right. \\ \left. \cdot\, \delta \left(\kappa_{\mu'} + \sum_{i=1}^{q} E_s p_i \cos\phi_i \right) \right\}. \tag{C.2}$$

$\delta(.)$ is the Kronecker delta function, E_s is the symbol energy (we have normalized this to 1 in our analysis). We have defined q as $M/2+1$ and the phase alphabet $\{\phi_i,\ 1 \le i \le M\} = \{\arg(W_i),\ 1 \le i \le M\}$. The numbers p_1 to p_q are positive integers less than or equal to μ', that must add up to μ' in the above summation (polynomial law).

The PDF $f_{\kappa_{\mathrm{L}}}(\kappa_L)$ is calculated by replacing μ' with L above.

Appendix D

Necessary Means and Variances

D.1 Moments Needed for the Approximate Union Bound for the High SNR Rule for the Traditional Frame Sync Problem

D.1.1 Means

$$m_{H^+} = \frac{1}{\sqrt{4\pi N_0}} \cdot \int\limits_{-\infty}^{0} x \cdot e^{\frac{-1}{4N_0}(x-2)^2} dx, \tag{D.1}$$

this integral is easily solved by substituting $y = (x-2)^2$ into $\int e^{ay} dy = \frac{1}{a} e^{ay}$:

$$m_{H^+} = -\sqrt{\frac{N_0}{\pi}} \cdot e^{\frac{-1}{N_0}} + \operatorname{erfc}\left(\frac{1}{N_0}\right). \tag{D.2}$$

Similarly,

$$m_{H^-} = \frac{1}{\sqrt{4\pi N_0}} \cdot \int\limits_{-\infty}^{0} x \cdot e^{\frac{-1}{4N_0}(x+2)^2} dx = m_{H^+} - 2. \tag{D.3}$$

D.1.2 Variances

$$\sigma^2_{H^+} = \frac{1}{\sqrt{4\pi N_0}} \cdot \int\limits_{-\infty}^{0} x^2 \cdot e^{\frac{-1}{4N_0}(x-2)^2} dx - {m_{H^+}}^2, \tag{D.4}$$

with integral 7 in section 3.462 from [GR65] and substitution it can be shown that

$$\sigma^2_{H^+} = \frac{1}{\sqrt{4\pi N_0}} \cdot \left[-e^{\frac{-1}{N_0}} 4N_0 + \sqrt{\pi (4N_0)^5} \cdot \left(\frac{1}{8{N_0}^2} + \frac{1}{16 N_0} \right) \cdot \operatorname{erfc}\left(\frac{1}{N_0}\right) \right] - {m_{H^+}}^2, \tag{D.5}$$

and

$$\sigma_{H^-}^2 = \frac{1}{\sqrt{4\pi N_0}} \cdot \left[e^{\frac{-1}{N_0}} 4N_0 + \sqrt{\pi (4N_0)^5} \cdot \left(\frac{1}{8{N_0}^2} + \frac{1}{16N_0} \right) \right.$$
$$\left. \cdot \left\{ 2 - \mathrm{erfc}\left(\frac{1}{N_0} \right) \right\} \right] - {m_{H^-}}^2. \tag{D.6}$$

D.2 Moments Needed for the Approximate Union Bound for the High SNR Rule for Packet Synchronization

D.2.1 Means

Using integral 1 in section 3.381 from [GR65] and substitution, we obtain:

$$m_n = \frac{2}{\sqrt{\pi N_0}} \cdot \int_0^\infty x \cdot e^{\frac{-1}{N_0} x^2} dx = \sqrt{\frac{N_0}{\pi}}, \tag{D.7}$$

$$m_d = \frac{1}{\sqrt{\pi N_0}} \cdot \int_0^\infty x \cdot e^{\frac{-1}{N_0}(x-1)^2} dx + \frac{1}{\sqrt{\pi N_0}} \cdot \int_0^\infty -x \cdot e^{\frac{-1}{N_0}(x-1)^2} dx = 1 - m_{H^+}. \tag{D.8}$$

D.2.2 Variances

$$\sigma_n^2 = \frac{2}{\sqrt{\pi N_0}} \cdot \int_0^\infty x^2 \cdot e^{\frac{-1}{N_0} x^2} dx - {m_n}^2 = N_0 \left(\frac{1}{2} - \frac{1}{\pi} \right) - {m_n}^2, \tag{D.9}$$

$$\sigma_d^2 = \left(1 + \frac{N_0}{2} \right) - {m_d}^2. \tag{D.10}$$

Appendix E

Tables of Binary Sync Words

Table E.1: *Binary sync words for channels without phase ambiguity taken from [Sch80], result of search by Maury and Styles. They share the property of having a low (and perhaps negative) partial autocorrelation function.*

Length L	sync word	Length L	sync word
7	1011000	19	1111100110010100000
8	10111000	20	11101101111000100000
9	101110000	21	111011101001011000000
10	1101110000	22	1111001101101010000000
11	10110111000	23	10110101101011010000000
12	110101100000	24	111110101111001100100000
13	1110101100000	25	1111100101101110001000000
14	11100110100000	26	11111010011010011001000000
15	111011001010000	27	111110101101001100110000000
16	1110101110010000	28	1111010111100101100110000000
17	11110011010100000	29	11110101111001100110100000000
18	111100110101000000	30	111110101111001100110100000000

Table E.2: *Binary sync words for channels with phase ambiguity taken from [Sch80], result of search by Turyn. Note that not all lengths L have a sync word that has a partial autocorrelation function low in magnitude.*

Length L	sync word	Length L	sync word
7	1011000	23	10011001101011111000010
11	10110111000	24	111111110001101010011011
13	1111100110101	25	1110011100000010101001001
14	11111001100101	27	110110100100010001000111100
15	111110011010110	28	1101101001000100010001111000
16	1110111000010110	29	11011010010001000100011110000
17	11001111101010010	30	111111101100101011010001111001
18	111110100101110011	31	1110011000111111010101001001000
19	1111000111011101101	32	11111101110010001110100100101110
20	11111011100010110100	33	111111000100111001011100101100101
21	111111010001011000110	34	1111111000011010010110011001010101
22	1111111100011011001010		

Table E.3: *Binary sync words for packets and channels without phase ambiguity. Result of brute-force search with optimization of (9.9). Note that the search produced identical sync words to Table E.1 for $L = 7$ and 11.*

Length L	sync word	Length L	sync word
2	10	15	010010111011100
3	100	16	0100101110111000
4	0110	17	00100110101111000
5	01110	18	011001101011110000
6	101100	19	1001110010111010000
7	1011000	20	10001000111110010110
8	01011100	21	010010100110011111000
9	011110010	22	1100001110111010100100
10	0110111000	23	10000111110101100110010
11	10110111000	24	011011010101110011100000
12	010110111000	25	0111000011101110110100100
13	0111100100010	26	00010111001110101101101000
14	00111010100100	27	100001001111011101001110100

Table E.4: *Binary sync words for packets and channels with phase ambiguity. Result of brute-force search with optimization of (F.1). Note that the search produced identical sync words to Table E.2 for $L = 7, 11, 13$ and 18.*

Length L	sync word	Length L	sync word
2	00	14	01111110011010
3	100	15	101011001100000
4	1000	16	1001101011111100
5	01000	17	11000100010110100
6	001000	18	001100010110100000
7	1011000	19	1101100111101010000
8	10110000	20	10011100101110100000
9	001101000	21	011010101100111111100
10	0110000010	22	0001110010101100100000
11	10110111000	23	11000101110100100001000
12	101001100000	24	100110110101000000011100
13	0101001100000	25	1001101101010000000111000
		26	01110001111111010100100110

Appendix F

Approximate Union Bound for the high SNR rule for Packet Synchronization for BPSK and Phase Ambiguity

The approximate bound for the High SNR synchronizer and BPSK with π radians phase ambiguity is

$$
\begin{aligned}
Pr\{f\} \approx \frac{1}{2\cdot(G-N)} \Bigg\{ & \sum_{\Delta\mu=L}^{G-N} n_{\Delta\mu} \big[\mathrm{erfc} \\
& \left(\frac{Lm_{H^+} + \Delta\mu m_d - m_n(\Delta\mu - L) - \sqrt{LN_0/\pi}}{\sqrt{2L\sigma_{H^+}^2 + 2\Delta\mu\sigma_d^2 + LN_0(1-2/\pi) + 2\sigma_n^2(\Delta\mu - L)}} \right) + \\
& \sum_{\forall\kappa_L} Pr\{\kappa_L\}\cdot \mathrm{erfc} \\
& \left(\frac{L + m_d(\Delta\mu - L) - \left(\frac{L+|\kappa_L|}{2}\right) m_{H^+} - \left(\frac{L-|\kappa_L|}{2}\right) m_{H^-} - \Delta\mu\cdot m_n}{\sqrt{LN_0 + 2\sigma_d^2(\Delta\mu-L) + \sigma_{H^+}^2(L+|\kappa_L|) + \sigma_{H^-}^2(L-|\kappa_L|) + 2\Delta\mu\sigma_n^2}} \right) \Bigg] + \\
& \sum_{\Delta\mu=0}^{L-1} n_{\Delta\mu} \left[\mathrm{erfc} \left(\frac{L - \Delta\mu - |R_{\Delta\mu}| - \sqrt{\Delta\mu N_0/\pi} + \Delta\mu m_{H^+} + \Delta\mu m_d}{\sqrt{N_0(2L - 2\Delta\mu - 2|R_{\Delta\mu}|) + 2\Delta\mu(\sigma_{H^+}^2+\sigma_d^2) + \Delta\mu N_0(1-2/\pi)}} \right) + \right. \\
& \sum_{\forall\kappa_{\Delta\mu}} Pr\{\kappa_{\Delta\mu}\}\cdot \mathrm{erfc} \\
& \left. \left(\frac{L - R_{\Delta\mu} - \left(\frac{\Delta\mu+\hat{k}\kappa_{\Delta\mu}}{2}\right) m_{H^+} - \left(\frac{\Delta\mu - \hat{k}\kappa_{\Delta\mu}}{2}\right) m_{H^-} - \Delta\mu\cdot m_n}{\sqrt{N_0(2L - 2\hat{k}R_{\Delta\mu} - \Delta\mu) + \sigma_{H^+}^2(\Delta\mu+\hat{k}\kappa_{\Delta\mu}) + \sigma_{H^-}^2(\Delta\mu - \hat{k}\kappa_{\Delta\mu}) + 2\Delta\mu\sigma_n^2}} \right) \right] \Bigg] \Bigg\},
\end{aligned}
\tag{F.1}
$$

where $\hat{k} = sign(\kappa_{\Delta\mu} + R_{\Delta\mu})$.

Appendix G

Tables of Parameters for Linear Approximation of the Sync Error Rate for Packets

The design rules from chapter 10 are repeated here:

$$\log\left(Pr\{f\}\right) \approx A_3 + A_4 \cdot \rho_d + \left(A_0 + A_1 \cdot \rho_d + A_2 \cdot {\rho_d}^2\right) \cdot L, \tag{G.1}$$

with the inverse function

$$L \approx \frac{\log\left(Pr\{f\}\right) - A_3 - A_4 \cdot \rho_d}{A_0 + A_1 \cdot \rho_d + A_2 \cdot {\rho_d}^2}. \tag{G.2}$$

The necessary parameters can be taken from Table G.1. To take into account different $G-N$, a simple linear interpolation (or extrapolation) of the values for A_0 to A_4 is sufficiently accurate for reasonable $G-N$.

Table G.1: *Parameters for linear approximation of the Sync Error Rate for Packets. To be inserted into equation (10.5).*

Case	A_0	A_1	A_2	A_3	A_4
BPSK, no PA, $G-N=15$	-0.23419	-0.037167	-0.0099752	0.2096813	-0.103088
BPSK, no PA, $G-N=50$	-0.23012	-0.029291	-0.011102	0.2778632	-0.1161422
BPSK, π PA, $G-N=15$	-0.22751	-0.043348	-0.0091694	0.5498587	-0.03627786
BPSK, π PA, $G-N=50$	-0.21546	-0.04619	-0.0086269	0.6110553	-0.04511575

Bibliography

[Abr70] N. Abramson. The ALOHA system -another alternative for computer communication. In *Proc. AFIPS Fall Joint Comput. Conf.*, volume 37, pages 281–285, 1970.

[AW92] H. Armbrüster and K. Wimmer. Broadband multimedia applications using ATM networks: High-performance computing, high-capacity storage, and high-speed communication. *IEEE Journal on Selected Areas in Comm.*, 10(9):1382–1396, December 1992.

[Bar53] R. H. Barker. *Group Synchronization of Binary Digital Systems.* Butterworth, London, 1953.

[Bay64] T. Bayes. An essay towards solving a problem in the doctrine of chances. *Phil. Trans.*, 53:370–418, 1764.

[Bi83] G-G. Bi. Performance of frame sync acquisition algorithms on the AWGN-channel. *IEEE Trans. Commun.*, 31(10):1196–1201, October 1983.

[Bla83] R. E. Blahut. *Theory and Practice of Error Control Coding.* Massachusetts: Addison-Wesley Publishing Co., 1983.

[Boe92] A. Boettcher. *Vielfachzugriffsmethoden in zentralisierten Datenfunknetzen für zeitvariante, gestörte Kanäle.* PhD thesis, University of Federal Armed Forces, Munich, November 1992. Published by VDI Verlag, Reihe 10, Nr. 230.

[CC81] G. C. Clark and J. B. Cain. *Error-Correction Coding for Digital Communications.* New York: Plenum, 1981.

[Con87] Consultative Committee for Space Data Systems. *Recommendation for space data systems standard, telemetry channel coding*, ccsds 101.0-b-2, blue book, issue 2 edition, January 1987.

[Cow93a] W. Cowley. Personal Communication, June 1993.

[Cow93b] W. G. Cowley. Phase and frequency estimation for psk packets: Bounds and algorithms. Accepted for IEEE Trans. Commun., 1993.

[Cro93] S. Crozier. Theoretical and simulated performance for a novel frequency estimation technique. In *Proc. IMSC'93*, pages 423–428, June 1993.

[CS88] L. F. Chang and N. R. Sollenberger. The use of cyclic block codes for synchronization recovery in a TDMA radio system. In *Proc. 3rd Nodic Sem. Digital Land Mobile Radio Comm.*, September 1988. Paper 13.10.

[CSV90] H. Chalmers, A. Shenoi, and F. Verahrami. A digitally implemented preambleless demodulator for maritime moble data communication. In *Proc. GLOBECOM'90*, pages 729–732, December 1990.

[Dri93] P. F. Driessen. Improved frame synchronization performance for CCITT algorithms using bit erasures. Accepted for publication in IEEE Trans. Commun., 1993.

[DS90] D. Divsalar and M. Simon. Multiple-symbol differential detection of MPSK. *IEEE Trans. Commun.*, 38(3):300–308, March 1990.

[DS92] S. Denno and Y. Saito. Recursive least squares estimation for QAM and QPSK signals in the presence of frequency offset. In *Proc. ICC '89*, pages 491–495, June 1992.

[Edb89] F. Edbauer. Concatenated trellis-coded differential 8-PSK modulation for rayleigh channels. In *Proc. URSI '89*, pages 268–271, 1989.

[Edb92] F. Edbauer. Bit error rate of binary and quaternary DPSK signals with multiple differential feedback detection. *IEEE Trans. Commun.*, 40(3):457–460, March 1992.

[Feh83] K. Feher. *Digital Communications, Satellite/Earch Station Engineering.* Prentice-Hall, 1983.

[For73] G. D. Forney. The Viterbi algorithm. *Proc. of the IEEE*, 61(3):268–278, March 1973.

[Fra80] L. E. Franks. Carrier and bit synchronization in data communications- a tutorial review. *IEEE Trans. Commun.*, 28(8):1107–1121, August 1980.

[Gar86] F. M. Gardner. A BPSK-QPSK-timing-error detector for sampled receivers. *IEEE Trans. Commun.*, 34(5):423–429, May 1986.

[GR65] I. S. Gradshteyn and I.M. Ryzhik. *Tables of Integrals, Series and Products.* Academic Press, 1965.

[GS65] S. W. Golomb and R. A. Scholtz. Generalized Barker sequences. *IEEE Trans. IT.*, 11:533–537, 1965.

[Hag80] J. Hagenauer. Viterbi decoding of convolutional codes for fading- and burst-channels. In *Proc. Zurich Seminar on Digital Communications*, pages G2.1–G2.7, 1980.

[Hag88] J. Hagenauer. Rate-compatible punctured convolutional codes (RCPC codes) and their applications. *IEEE Trans. Commun.*, 36(4):389–400, April 1988.

[Hag94] J. Hagenauer. Source-controlled channel decoding. Accepted for publication in IEEE Trans. Commun., 1994.

[HH89] J. Hagenauer and P. Hoeher. A Viterbi algorithm with soft-decision outputs and its applications. In *Proc. GLOBECOM '89*, pages 1680–1686, November 1989.

[HM88] R. Häb and H. Meyr. A digital synchronizer for linearly modulated signals transmitted over a frequency-nonselective fading chanel. In *Proc. ICC '88*, pages 1012–1016, June 1988.

[Hoe90] P. Hoeher. *Kohärenter Empfang trelliscodierter PSK-Signale auf frequenceseleck-tiven Mobilfunkkanälen - Entzerrung, Decodierung und Kanalparameterschätzung.* PhD thesis, University of Kaiserslautern, July 1990. Published by VDI Verlag, Reihe 10, Nr. 147.

[Hoe93] P. Hoeher. Advances in soft output decoding. In *Proc. GLOBECOM '93*, pages 793–797, December 1993.

[HR90] J. Huber and A. Rüppel. Zuverlässigkeitsschätzung für die ausgangssymbole von trellis-decodern. *Archiv für Elekronik und Übertragung (AEÜ)*, 44:8–21, January 1990.

[HW90] T. Hellmich and B. Walke. Highly reliable channels for short-range moble radio networks. In *Proc. Third Information Conference PROC-COM*, December 1990.

[Int89] International Maritime Satellite Organization. *System Definition Manual for the Standard-C Communications System*, July 1989. Release 1.3.

[Jac92] A. Jacobs. Rahmensynchronisation und Faltungscodierung. Diploma thesis (supervised by the author), TU Vienna, November 1992.

[JAS85] E. V. Jones and M. N. Al-Subbagh. Algorithms for frame alignment -some comparisons. *IEE Proceedings*, 132(7):529–536, December 1985.

[JVM93] O. J. Joerensen, M. Vaupel, and H. Meyr. Soft-output Viterbi decoding: VLSI implementation issues. In *Proc. VTC '93*, pages 941–944, May 1993.

[KL84] S. S. Kamal and R. G. Lyons. Unique-word detection in TDMA: acquisition and retention. *IEEE Trans. Commun.*, 32(7):804–817, July 1984.

[Kob71] H. Kobayashi. Simultaneous adaptive estimation and decision algorithm for carrier modulated data transmission systems. *IEEE Trans. Commun.*, 19(3):268–280, June 1971.

[KT75] L. Kleinrock and F. A. Tobagi. Packet switching in radio channels: part I-carrier sense multiple access modes and their throughput delay characteristics. *IEEE Trans. Commun.*, 23(12):1400–1416, December 1975.

[LC81] W. Lindsey and C. Chie. A survey of digital phase-locked loops. *Proceedings of IEEE*, 69(4):410–431, April 1981.

[Lin75] J. Lindner. Binary sequences up to length 40 with best possible autocorrelation properties. *Electronic Letters*, 1975.

[LS73] W. Lindsey and M. Simon. *Telecommunications Systems Engineering*. Prentice-Hall, 1973.

[LT86] G. L. Lui and H. H. Tan. Frame synchronization for direct-detection optical communication systems. *IEEE Trans. Commun.*, 34(3):227–237, March 1986.

[LT87] G. L. Lui and H. H. Tan. Frame synchronization for Gaussian channels. *IEEE Trans. Commun.*, 35(8):818–829, August 1987.

[LY84] V. Li and T-Y. Yan. Adaptive mobile access protocol (AMAP) for the message service of a land mobile satellite experiment experiment (MSAT-X). *IEEE Journ. SAC*, 2(4):621–627, July 1984.

[Mas72] J. Massey. Optimum frame synchronization. *IEEE Trans. Commun.*, 20(4):115–119, April 1972.

[Mas93a] A. Masera. Analytische und numerische Beschreibung der Rahmensynchronisationswahscheinlichkeit für kontinuierliche Datenübertragung und Paketübertragung. Diploma thesis (supervised by the author), Fed. Univ. Armed Forces, Munich, April 1993.

[Mas93b] J. Massey. Personal Communication, March 1993.

[MC85] J. J. Metzner and D. Chang. Efficient selective repeat ARQ strategies for very noisy and fluctuating channels. *IEEE Trans. Commun.*, 33(5):409–416, May 1985.

[MM76] K. Mueller and M. Müller. Timing recovery in digital synchronous data receivers. *IEEE Trans. Commun.*, 24(5):516–531, May 1976.

[MM93] R. Mehlan and H. Meyr. Optimum frame synchronization for asynchronous packet transmission. In *Proc. ICC '93*, pages 826–830, May 1993.

[MS64] J. L. Maury and F. J. Styles. Development of optimum frame synchronization codes for Goddard Space Flight Center PCM telemetrie standards. In *Proc. 1964 Nat. Telemetering Conf.*, June 1964. paper 3-1.

[MS90] M. Moenclaey and P. Sanders. Syndrome-based Viterbi decoder node synchronization and out-of-lock detection. In *Proc. GLOBECOM'90*, pages 604–608, December 1990.

[MS91] B. H. Moon and S. S. Soliman. ML frame synchronization for the Gaussian channel with ISI. In *Proc. ICC '91*, pages 1698–1702, June 1991.

[MW86] H. H. Ma and J. K. Wolf. On tail-biting convolutional codes. *IEEE Trans. Commun.*, 34(2):104–111, February 1986.

[Neu89] A. Neul. *Modulation und Codierung im aeronautischen Satellitenkanal.* PhD thesis, University of Federal Armed Forces, Munich, June 1989.

[NH71] F. Neuman and L. Hofman. New pulse sequences with desirable correlation properties. In *Conf. Rec. 1971 Nat. Telemetering Conf.*, pages 277–282, 1971.

[Nie73] P. T. Nielsen. Some optimum and suboptimum frame synchronizers for binary data in Gaussian noise. *IEEE Trans. Commun.*, 21(6):770–772, June 1973.

[NS93] C. Nill and C-E. W. Sundberg. Viterbi algorithms with list and soft symbol outputs: extensions and comparisons. In *Proc. GLOBECOM '93*, pages 788–792, December 1993.

[Oer89] M. Oerder. *Algorithmen zur digitalen Taktsynchronization bei Datenübertragung.* PhD thesis, Aachen University of Technology, February 1989. Published by VDI Verlag, Reihe 10, Nr. 119.

[Ohs90] T. Ohsawa. Block demodulation method for PSK signal by sequential regression estimator. *Electronics and Communications in Japan, Part 1*, 73(6):41–51, June 1990.

[OI92] T. Ohsawa and M. Iwasaki. Preamble-less demodulation for IMARSAT standard-C coast earth station. In *Proc. ICC '92*, pages 783–787, June 1992.

[OM88] M. Oerder and H. Meyr. Digital filter and square timing recovery. *IEEE Trans. Commun.*, 36(5):605–612, May 1988.

[Paa90] E. Paaske. Improved decoding for a concatenated coding system recommended by CCSDS. *IEEE Trans. Commun.*, 38(8):1138–1144, August 1990.

[Pap84] A. Papoulis. *Probability, Random Variables and Stochastic Processes.* New York: McGraw Hill Book Co., 1984.

[Par80] S. Parl. A new method of calculating the generalized Q function. *IEEE Trans. Information Theory.*, 26(1):121–124, January 1980.

[PFTV88] W. H. Press, B. P. Flannery, S. S. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C -The Art of Scientific Computing.* Cambridge University Press, 1988.

[Pro89] J. G. Proakis. *Digital Communications.* New York: McGraw Hill Book Co., 1989.

[Rif74] D. C. Rife. Single-tone parameter estimation from discrete-time observations. *IEEE Trans. IT.*, 20(5):591–598, September 1974.

[Rob92a] P. Robertson. A generalized frame synchronizer. In *Proc. GLOBECOM '92*, pages 365–369, December 1992.

[Rob92b] P. Robertson. Maximum likelihood frame synchronization for flat fading channels. In *Proc. ICC '92*, pages 1426–1430, June 1992.

[Rob93] P. Robertson. Improving frame synchronizer when using convolutional codes. In *Proc. GLOBECOM '93*, pages 1606–1611, December 1993.

[Rob94] P. Robertson. Optimum frame synchronization of packets surrounded by noise with coherent and differentially coherent demodulation. In *Proc. ICC '94*, May 1994.

[SC92] L. P. Sabel and W. G. Cowley. New synchronization methods for symbol timing recovery in TDMA modems. In *Proc. Mobile and Personal Communications Systems*, pages 213–222, November 1992.

[Sch80] R. Scholtz. Frame synchronization techniques. *IEEE Trans. Commun.*, 28(8):1204–1212, August 1980.

[Sch91] W. Schäfer. Channel modelling of short-range radio links at 60 GHz for mobile intervehicle communication. In *Proc. 41 VTS*, pages 314–318, May 1991.

[SD87] M. K. Simon and D. Divsalar. Open loop frequency synchronization of MPSK with Doppler. In *Proc. ICC '87*, pages 232–238, June 1987.

[SH90] T. Schaub and A. Hansson. Frame synchronization for spontaneous transmissions. In *Proc. GLOBECOM'90*, pages 617–622, December 1990.

[SS89] N. Seshadri and C-E. W. Sundberg. Generalized Viterbi algorithms for error-detection with convolutional codes. In *Proc. GLOBECOM '89*, pages 1534–1538, December 1989.

[SSHS92] G. Schultes, A. Scholtz, M. Happel, and W. Simbürger. A testbed for DECT physical- and medium access-layer. In *Proc.*, pages 349–353, 1992.

[Sti71] J. J. Stiffler. *Synchronous Communications.* New York: Plenum, 1971.

[Tre68] H. Van Trees. *Detection, Estimation and Modulation Theory.* Wiley, New York, 1968.

[TS61] R.J. Turyn and J. Storer. On binary sequences. *Proceedings of American Mathematical Society*, 12:394–399, 1961.

[Tur68] R. Turyn. *Sequences with Small Correlation.* Wiley, New York, 1968.

[Tur74] R. Turyn. Four-phase Barker codes. *IEEE Trans. IT.*, 20:366–371, May 1974.

[Ung81] G. Ungerboeck. Channel coding with multilevel/phase signals. *IEEE Trans. IT.*, 28(1):55–67, January 1981.

[UTMT91] M. Uchishima, Y. Tozawa, T. Miyo, and S. Takenaka. Burst DSP demodulator for low Eb/No operation. In *Proc. ICC '91*, pages 226–230, June 1991.

[VO79] A. J. Viterbi and J. K. Omura. *Principles of Digital Communication and Coding.* New York: McGraw Hill Book Co., 1979.

[VV83] A. J. Viterbi and A. M. Viterbi. Nonlinear estimation of PSK-modulated carrier phase with application to burst digital transmission. *IEEE Trans. IT.*, 29(4):543–551, July 1983.

[Wel83] E. J. Weldon. An improved selective-repeat ARQ strategy. *IEEE Trans. Commun.*, 30(3):480–486, March 1983.

[Wil62] M. W. Willard. Optimum code patterns for PCM synchronization. In *Proc. 1962 Nat. Telemetering Conf.*, May 1962. paper 5-5.

[WJ65] J. M. Wozencraft and I.M. Jacobs. *Principles of Communication Engineering.* New York: John Wiley & Sons, Inc., 1965.

[WRL+91] T. Wörz, P. Robertson, H. Lang, C. Moss, and H. Godde. Final report, short range microwave communication for rti applications. Technical report, DLR/MARCONI/SEL, August 1991. Reference: SP5/P9/M665/TG/CRM/4075/BAH/B/Drive2.

[ZG90] N. Zhang and S. W. Golomb. N-phase Barker sequences. In *Proc. Int. Symposium on Information Theory and its Applications (ISITA) '90*, pages 845–848, November 1990.